A General and Efficient Algorithm for the Likelihood of Diversification and Discrete-Trait Evolutionary Models

Stilianos Louca[1,2,*] & Matthew W. Pennell[3,4]

[1]*Department of Biology, University of Oregon, USA*
[2]*Institute of Ecology and Evolution, University of Oregon, USA*
[3]*Biodiversity Research Centre, University of British Columbia, Vancouver, Canada*
[4]*Department of Zoology, University of British Columbia, Vancouver, Canada*
[*]Corresponding author

Abstract

As the size of phylogenetic trees and comparative data continue to grow and more complex models are developed to investigate the processes that gave rise to them, macroevolutionary analyses are becoming increasingly limited by computational requirements. Here we introduce a novel algorithm, based on the "flow" of the differential equations that describe likelihoods along tree edges in backward time, to reduce redundancy in calculations and efficiently compute the likelihood of various macroevolutionary models. Our algorithm applies to several diversification models, including birth-death models and models that account for state- or time-dependent rates, as well as many commonly used models of discrete-trait evolution, and provides an alternative way to describe macroevolutionary model likelihoods. As a demonstration of our algorithm's utility, we implemented it for a popular class of state-dependent diversification models — BiSSE, MuSSE, and their extensions to hidden-states. Our implementation is available through the R package `castor`. We show that, for these models, our algorithm is one or more orders of magnitude faster than existing implementations when applied to large phylogenies. Our algorithm thus enables the fitting of state-dependent diversification models to modern massive phylogenies with millions of tips, and may lead to potentially similar computational improvements for many other macroevolutionary models.

Keywords: Speciation; Extinction; Macroevolution; Likelihood; Dynamical systems theory; flow

There is a vast, and ever-growing, array of statistical models that can be fit to phylogenetic trees and comparative data to investigate the historical dynamics of diversification, trait evolution, and the interaction between the two (O'Meara, 2012; Pennell and Harmon, 2013; Morlon, 2014; Ng and Smith, 2014; Harmon, 2018). These models have empowered researchers to move beyond summary statistics such as tree balance (Mooers and Heard, 1997), towards explicitly quantifying the variation in speciation and extinction rates across the Tree of Life (Magallon and Sanderson, 2001; Alfaro *et al.*, 2009; Henao Diaz *et al.*, 2019) and identifying the major drivers of this variation (Schluter and Pennell, 2017; Wiens, 2017). Concurrently, the scale of comparative data has also been growing tremendously. There are now phylogenetic trees for multiple groups that contain tens of thousands or even millions of lineages (Jetz *et al.*, 2012; Zanne *et al.*, 2014; Hinchliff *et al.*, 2015; Thompson *et al.*, 2017; Parks *et al.*, 2018; Smith and Brown, 2018) — though we are still far from having a comprehensive representation of the full Tree of Life (Mora *et al.*, 2011; Larsen *et al.*, 2017; Hinchliff *et al.*, 2015). Similarly large-scale efforts are underway to assemble trait information for many lineages, both multicellular (Cornwell *et al.*, 2018) and microbial (Mendler *et al.*, 2018).

Taken together, these developments provide tremendous opportunities for gaining new insights into macroevolutionary processes at unprecedented scales. However, as we show below, current computational procedures for fitting macroevolutionary models become practically unfeasible at the scale of modern mega-phylogenies. This greatly limits the analyses conductible with existing models and restricts the future development of even more complex models. For example, massive bacterial phylogenies could shed light on the role that the repeated loss and gain of metabolic functions, generally suspected to be dominated by horizontal gene transfer (David and Alm, 2011; Polz *et al.*, 2013; Hehemann *et al.*, 2016), has had on bacterial diversification over geological time scales (Latysheva *et al.*, 2012; Muscarella and O'dwyer, 2018). Models for state-dependent speciation and extinction (Maddison *et al.*, 2007; FitzJohn, 2012; Goldberg and Igić, 2012; van Els *et al.*, 2018; Caetano *et al.*, 2018) would be particularly suited for such an analysis, but are computationally too demanding to be applied at this scale.

To address these emerging challenges, we leverage results from dynamical systems theory, a well-established field in physics and mathematics (Meiss, 2007), and develop a novel algorithm for computing the likelihood of a large class of models for diversification and trait evolution. Dynamical systems theory investigates the behavior of time-dependent systems (including their trajectories, equilibria and stability), often described through differential equations analogous to the "equations of motion" in classical mechanics. Calculating the likelihood of macroevolutionary models often translates to calculating the solution of an equation of motion for a set of probabilities along a tree's branches, in backward time. As we show below, basic tools from dynamical systems theory can thus be used to devise an algorithm for macroevolutionary likelihood calculations that can be orders of magnitude faster than existing approaches. Our work also highlights a previously unrecognized deep similarity between seemingly distinct classes of methods; we anticipate that the recognition of this similarity will help spur the development of new types of models.

Classical approaches for calculating likelihoods of macroevolutionary models

Birth-death processes (Kendall, 1948) have long been a pillar of macroevolutionary theory (Raup, 1985), and following the pioneering work of Nee *et al.* (1994) researchers have routinely fit these models to phylogenetic data. While the simple, single-rate birth-death process has been extended to an impressive variety of models (Morlon, 2014; Harmon, 2018), these modifications broadly fall into three major classes. First, there may be variation in speciation and extinction rates through time (Rabosky *et al.*, 2007; Stadler, 2011a; Morlon *et al.*, 2011); this includes models where the rates depend on another environmental variable (e.g., Condamine *et al.*, 2013). Second a phylogeny may be partitioned by clade into several rate classes (Alfaro *et al.*, 2009;

Rabosky, 2014). And third, rates at each lineage may be associated with the current state of an evolving trait. The pioneering work of Maddison *et al.* (2007) and their BiSSE (Binary State Speciation and Extinction) model, triggered the development of a plethora of State-dependent Speciation and Extinction (SSE) models. For example, it is now possible to fit models where diversification rates vary with the state of a multi-state character (MuSSE; FitzJohn *et al.*, 2009), geographic area (GeoSSE; Goldberg *et al.*, 2011), or quantitative character (QuaSSE; FitzJohn, 2010), and character transitions may occur either along lineages (anagenetic transitions; FitzJohn *et al.*, 2009) or during speciation events (cladogenetic transitions; Magnuson-Ford and Otto, 2012; Goldberg and Igić, 2012). More recently, SSE models have been extended to include hidden states (Beaulieu and O'Meara, 2016; van Els *et al.*, 2018; Caetano *et al.*, 2018), which has been demonstrated to greatly improve the applicability of SSE-type models (Caetano *et al.*, 2018). These ways of introducing variation (by time, clade, or state) are not mutually exclusive (e.g., Rabosky and Glor, 2010; Morlon *et al.*, 2011; Cantalapiedra *et al.*, 2014), nor are they exhaustive (e.g., Etienne and Haegeman, 2012). Finally, some models merely describe the evolution of discrete characters along branches of a given phylogeny, i.e., diversification and character evolution are assumed to have occurred independently (Pagel, 1994; Lewis, 2001).

Beneath this apparent variety of models lies a deep similarity. Indeed, the likelihood of all of these models can be computed by moving down the tree postorder (tips to root), and recursively solving the Kolmogorov backward equation of the Markov chain along each edge (Kolmogorov, 1931; Feller, 1949). This works because each edge is assumed to represent a realization of a continuous-time Markov chain that is independent of all other edges, with initial state equal to the final state at the parent node. The Kolmogorov backward equation is a differential equation that describes how the likelihoods of arriving at a "target state" (the observed data) change as one moves backward in time.

In the simple case of a character-independent birth-death model, where diversification rates are either constant or depend only on time (and not on the value of an evolving state) (e.g., Morlon *et al.*, 2011), the Kolmogorov backward equation describes the likelihood $X(t)$ that a lineage alive at "age" t (time before present) would leave exactly one descending lineage in the phylogeny at some fixed later time:

$$\frac{dX}{dt} = \left[2\lambda(t)E(t) - \lambda(t) - \mu(t)\right] X(t), \tag{1}$$

where λ is the speciation rate, μ is the extinction rate, and $E(t)$ is the probability that a lineage alive at age t would be absent from the phylogeny (computed separately). We mention that this class of models includes models where the time-dependency of λ and μ also partly stems from a dependency on varying environmental conditions (Condamine *et al.*, 2013), as well as models where rates shift discontinuously over time (Stadler, 2011a). The solution to the differential equation (1), for any given initial condition at age s, is given by the simple product:

$$X(t) = \Psi(s,t) \cdot X(s), \tag{2}$$

where the factor $\Psi(s,t)$ is given by:

$$\Psi(s,t) = e^{\int_s^t [\lambda(u)-\mu(u)]du} \cdot \left[\frac{1+\rho\int_0^s e^{\int_0^\tau [\lambda(\sigma)-\mu(\sigma)]d\sigma}\lambda(\tau)d\tau}{1+\rho\int_0^t e^{\int_0^\tau [\lambda(\sigma)-\mu(\sigma)]d\sigma}\lambda(\tau)d\tau}\right]^2, \tag{3}$$

and where ρ is the sampling fraction (fraction of extant species included in the tree). Observe that Eq. (2) can be used to obtain the solution to the differential equation (1) for any arbitrary initial condition, and hence the $\Psi(s,t)$ fully encode the dynamics expressed by the differential equation. The quantity $\Psi(s,t)$ must be computed for each edge in the tree, where s is the age of the child node and t is the age of the parent node (Morlon *et al.*, 2011). We mention at this point that for any three ages t_o, s, t, the following property holds:

$$\Psi(s,t) = \Psi(t_o,t) \cdot \Psi(t_o,s)^{-1}. \tag{4}$$

Hence, if $\Psi(t_o,t)$ was known (e.g., pre-calculated) for some fixed t_o and for all t, then one could calculate $\Psi(s,t)$ for any arbitrary s, t through the simple formula in Eq. (4). As we explain below, such a relationship can be retrieved for a very general class of models, and constitutes the foundation of our proposed algorithm.

As mentioned above, a similar logic applies to the Mk models of trait evolution (Pagel, 1994), where transitions between any two states $i \to j$ occur along edges according to some fixed probability rate Q_{ij}. In Mk models, the Kolmogorov backward equation describes the evolution of the likelihoods $X_i(t)$ that a lineage, which at age t was at state i, would have a specific state at some fixed later time:

$$\frac{d\mathbf{X}}{dt} = \mathbb{Q} \cdot \mathbf{X}, \tag{5}$$

where $\mathbf{X}$ is a vector containing the likelihoods $X_1, X_2, ..$ and $\mathbb{Q}$ is the transition rate matrix. Calculating the model's likelihood involves solving the above differential equation for each edge in the tree, in postorder traversal, with the initial conditions at each node depending on the likelihoods calculated for the child edges. At the root, the final X_i are averaged to obtain an overall likelihood for the model. For any given initial condition at age s, the solution to the differential equation (5) is given by the product:

$$\mathbf{X}(t) = \Psi(s,t) \cdot \mathbf{X}(s), \tag{6}$$

where $\Psi(s,t) = e^{(t-s)\mathbb{Q}}$ is the matrix exponential. The fact that solutions to the differential equation (5) can be expressed as matrix exponentials is sometimes used for efficient computations of model likelihoods (Louca and Doebeli, 2017). As in the previous example, the matrices $\Psi(s,t)$ fully encode the dynamics expressed in the differential equation (5), and for any three ages t_o, s, t satisfy the relationship:

$$\Psi(s,t) = \Psi(t_o,t) \cdot \Psi(t_o,s)^{-1}, \tag{7}$$

where $\Psi(t_o,s)^{-1}$ is the matrix inverse.

Our final example are models where speciation and extinction rates depend on the state of an evolving discrete character. Here, and throughout the paper, we focus primarily on discrete-state speciation and extinction models (BiSSE and MuSSE; Maddison *et al.*, 2007; FitzJohn, 2012) and their extensions to including incomplete sampling (FitzJohn *et al.*, 2009), overlapping states (GeoSSE; Goldberg *et al.*, 2011), hidden variables (HiSSE, MuHiSSE, SecSSE, and GeoHiSSE; Beaulieu and O'Meara, 2016; van Els *et al.*, 2018; Caetano *et al.*, 2018), and cladogenetic state transitions (BiSSE-ness and ClaSSE; Magnuson-Ford and Otto, 2012; Goldberg and Igić, 2012), henceforth collectively "dSSE". The likelihood of a dSSE model with S diversification-modulating trait states is calculated based on a set of "extinction probabilities" $E_i(t)$ and likelihoods $X_i(t)$, defined for each state $i = 1, .., S$ and age t. More precisely, $E_i(t)$ is the probability that a lineage, which at age t was in state i, would be absent from the phylogeny either due to eventual extinction or due to incomplete species sampling. $X_i(t)$ is the likelihood that a lineage, which at age t was in state i, would evolve into the clade observed in the given phylogeny, taking into account the present-day states at the tips (if known). The variables $E_i(t)$ and $X_i(t)$ are computed by solving a system of differential equations along each edge in backward time. For dSSE models with non-cladogenetic transitions (such as BiSSE, MuSSE, SecSSE, GeoSSE, HiSSE and GeoHiSSE), these differential equations take the form:

$$\frac{dE_i}{dt} = \mu_i - (\lambda_i + \mu_i)\, E_i(t) + \lambda_i E_i(t)^2 + \sum_j Q_{ij} E_j(t), \tag{8}$$

$$\frac{dX_i}{dt} = [2\lambda_i E_i(t) - \lambda_i - \mu_i]\, X_i(t) + \sum_j Q_{ij} X_j(t), \tag{9}$$

where Q_{ij} is the (anagenetic) transition rate from state i to state j along a lineage, λ_i are the state-dependent speciation rates and μ_i are the state-dependent extinction rates. For models with cladogenetic transitions (BiSSE-ness and ClaSSE), the above differential equations are somewhat modified to accommodate state transitions during speciation events (e.g., see Goldberg and Igić, 2012, Appendix equations A1 & A2 therein). In all dSSE models, the extinction probabilities E_i can be computed regardless of the likelihoods X_j and regardless of the tree structure, by integrating the differential equation (9) from the present all the way back to the root, with initial conditions at present ($t = 0$) depending on the fraction of extant species in each state i that is included in the phylogeny ("sampling fractions"). In contrast, the X_i must be computed for each edge, traversing postorder from tips to root, with the initial conditions at each node depending on the values computed for the child edges (details in Appendix 1.). At the root, the final X_i are averaged to obtain an overall likelihood for the model. Contrary to the previous two examples, an explicit formula for the solutions of the differential equation (9) is almost never available. As we show below, however, a relationship between solutions along different time intervals as in the previous examples (Eqs. 4 and 7) can still be retrieved.

A new algorithm for the likelihood of macroevolutionary models

General description

All of the macroevolutionary models described above, and in fact many others, share the following fundamental aspect: Defining and computing the likelihood involves the calculation of one or more variables X_i at each node, based on a linear differential equation that must be solved in backward time along each edge:

$$\frac{d\mathbf{X}}{dt} = \mathbb{A}(t) \cdot \mathbf{X}(t), \tag{10}$$

where $\mathbf{X}(t)$ is a column-vector listing the variables $X_1(t), X_2(t)$.. to be computed along a specific edge and $\mathbb{A}(t)$ is some quadratic matrix. For example, in the case of Mk models Eq. (10) corresponds to Eq. (5), and in the case of MuSSE models Eq. (10) corresponds to Eq. (9). The coefficients in the matrix $\mathbb{A}(t)$, which describes the infinitesimal transitions of $\mathbf{X}$ along an edge, may depend on time, model parameters and the data at hand, but must be independent of the particular edge. As explained earlier, $\mathbf{X}(t)$ typically represents the likelihoods of some given observations depending on the state of a lineage at some age t, in which case Eq. (10) is the Kolmogorov backward equation of the underlying stochastic Markov process (Kolmogorov, 1931; Feller, 1949) and $\mathbb{A}(t)$ depends on the instantaneous probability rates of modeled events (e.g., extinction, speciation, trait changes). The initial conditions at each node are typically specified based on the solutions of $\mathbf{X}$ on the descending edges, in which case $\mathbf{X}$ must be computed in a postorder fashion (from tips to root), although in some simple models a postorder traversal is not necessary (Stadler, 2011a; Morlon *et al.*, 2011; Condamine *et al.*, 2013). For massive trees and for most models, explicitly solving the differential equation (10) for each edge can lead to impractically long computation times. Indeed, since edges (for example, in sister clades) span repeatedly overlapping time intervals, in large trees this approach exhibits a high level of redundancy. As explained below, this redundancy can be partly removed with an appropriately revised algorithm.

The linear structure of the differential equation (10) implies that it is in principle possible to find a general representation of solutions, such that any given initial condition at a node can be mapped to the corresponding solution at the parent node without explicitly solving the differential equation along the connecting edge. Before showing how such a representation can be obtained, it is useful to first highlight some of its general properties. For any two ages s and t, let $\Psi(s,t)$ be a function that maps initial conditions at age s to the corresponding solution of the differential equation (10) at time t; symbolically $\Psi(s,t) : \mathbf{X}(s) \mapsto \mathbf{X}(t)$.

That is, for any given $\mathbf{X}_1$ and any age s, let $\mathbf{X}(t) = \Psi(s,t)(\mathbf{X}_1)$ be the solution to the differential equation:

$$\frac{d\mathbf{X}(t)}{dt} = \mathbb{A}(t) \cdot \mathbf{X}(t), \tag{11}$$

for the specific initial condition $\mathbf{X}(s) = \mathbf{X}_1$ (note that the solution to the differential equation depends on the initial condition). The collection of mappings $\Psi(s,t)$, which encode the correspondence of initial conditions to solutions of a differential equation, is known in the dynamical systems literature as the "flow" of the differential equation (Olver, 2012; Arnold, 2013). The flow thus provides an alternative and complete description of the dynamics encoded by the differential equation; instead of describing changes in infinitesimal time steps, the flow describes changes across any finite time interval $s \to t$. In the typical scenario where $\mathbf{X}$ represents state-dependent likelihoods of observing the data, as in the examples discussed above, the flow $\Psi(s,t)$ becomes a "likelihood flow" that describes how the likelihoods $\mathbf{X}$ are transformed between any two ages s and t. Observe that the quantities $\Psi(s,t)$ introduced for the model examples in the previous section (Eqs. 3 and 6) constituted exactly the flow of their Kolmogorov backward equations.

A defining property of flows is that for any three ages t_o, s, t, the following relationship holds:

$$\Psi(s,t)(\mathbf{X}_1) = \Psi(t_o,t)\left(\Psi(s,t_o)(\mathbf{X}_1)\right). \tag{12}$$

That is, instead of mapping the initial condition $\mathbf{X}_1$ at age s to the corresponding solution at age t, one can first map $\mathbf{X}_1$ from age s to age t_o, and then map the obtained solution from age t_o to age t. Since $\Psi(s,t_o)$ is the inverse of $\Psi(t_o,s)$, we obtain the representation:

$$\Psi(s,t)(\mathbf{X}_1) = \Psi(t_o,t)\left(\Psi(t_o,s)^{-1}(\mathbf{X}_1)\right). \tag{13}$$

This symbolic representation forms the foundation of our algorithm: If one could somehow pre-calculate the flow $\Psi(t_o,t)$ for some fixed t_o (such as $t_o = 0$) and for all $t > t_o$, then one could obtain solutions for any initial condition defined at any other time s through the right-hand-side of Eq. (13). Conceptually, Eq. (13) provides an alternative, more abstract, description for the progression of $\mathbf{X}$ along edges and between nodes that is mathematically equivalent to the differential equation (10).

The precise nature of the flow Ψ depends on the nature of the differential equation (10). How, then, can one explicitly calculate the flow Ψ in practice? As the differential equation (10) is linear, for any two ages t_o and t the function $\Psi(t_o,t)$ must itself be linear. Hence, $\Psi(t_o,t)$ can be written in matrix format, i.e. the abstract notation $\Psi(t_o,t)(\mathbf{X}_o)$ becomes $\Psi(t_o,t) \cdot \mathbf{X}_o$, where $\Psi(t_o,t)$ is a matrix of the same size as $\mathbb{A}$. This matrix satisfies the differential equation:

$$\frac{d\Psi(t_o,t)}{dt} = \mathbb{A}(t) \cdot \Psi(t_o,t), \tag{14}$$

with initial condition $\Psi(t_o,t_o) = \mathbb{I}$, where $\mathbb{I}$ is the identity matrix. Thus, for any initial condition $\mathbf{X}(s) = \mathbf{X}_1$ at some age s, the corresponding solution at age t can be obtained algebraically:

$$\mathbf{X}(t) = \Psi(t_o,t) \cdot \Psi(t_o,s)^{-1} \cdot \mathbf{X}_1, \tag{15}$$

where $\Psi(t_o,s)^{-1}$ is the matrix inverse. The differential equation (14) can itself be solved from the present ($t = 0$) all the way to the root ($t = t_R$), regardless of tree structure and passing only once through each time point. Typically, t_o will simply correspond to the present, i.e. $t_o = 0$. Once $\Psi(t_o,t)$ is calculated for all t, one can calculate the solution of the differential equation (10) at one end of an edge, given any initial condition at its other end, without solving the differential equation along the edge from scratch. This idea is illustrated in Figure 1, where the flow $\Psi(t_o, t)$ is used to compute the variable $\mathbf{X}$ at successive nodes, traversing from tips to root.

Calculating the term $\Psi(t_o, s)^{-1}$ in Eq. (15) corresponds to inverting the matrix $\Psi(t_o, s)$, which can be computationally costly and is in fact not necessary. Indeed, the entire expression $\Psi(t_o, s)^{-1} \cdot \mathbf{X}_1$ in Eq. (15) can be replaced by a vector $\mathbf{X}_o$ that must be chosen such that it satisfies the condition:

$$\Psi(t_o, s) \cdot \mathbf{X}_o = \mathbf{X}_1. \tag{16}$$

To see why this is the case, note that the vector $\mathbf{Y}(t) := \Psi(t_o, t) \cdot \mathbf{X}_o$ satisfies the differential equation

$$\frac{d\mathbf{Y}}{dt} = \frac{d\Psi(t_o, t)}{dt} \cdot \mathbf{X}_o = \mathbb{A}(t) \cdot \Psi(t_o, t) \cdot \mathbf{X}_o = \mathbb{A}(t) \cdot \mathbf{Y}(t), \tag{17}$$

with initial condition

$$\mathbf{Y}(s) = \Psi(t_o, s) \cdot \mathbf{X}_o = \mathbf{X}_1, \tag{18}$$

which is the same differential equation and initial condition satisfied by $\mathbf{X}(t)$ in Eq. (15). Thus, it is not actually necessary to invert the entire matrix $\Psi(t_o, s)$, so long as a solution $\mathbf{X}_o$ to Eq. (16) can be found. Solving the linear system in Eq. (16) is generally easier than inverting the entire matrix $\Psi(t_o, s)$, and corresponds to mapping $\mathbf{X}_1$ “back to the future” at age t_o; the vector $\mathbf{X}_o$ is the hypothetical initial condition at t_o that would lead to $\mathbf{X}_1$ at age s according to the differential equation (10).

Here we introduced the flow Ψ as an alternative description of the differential equation (10), which in typical macroevolutionary models specifies the instantaneous rates at which likelihoods change in backward time along edges. The flow can be seen as the “macroscopic” probabilistic behavior of the model (i.e., across the finite time steps spanning adjacent nodes), emerging from the “microscopic” behavior (i.e., across infinitesimal time steps) described by the differential equation (10). This relationship is analogous to the duality between discrete-time and continuous-time population models, where the former conceptually correspond to the time-integrated version of the latter. The flow could thus enable novel interpretations of macroevolutionary processes and allow previously unrecognized model generalizations. For example, while every model with a Kolmogorov backward equation of the form in Eq. (10) admits a likelihood flow, the reverse need not be true. Indeed, one could envision models where anagenetic character transitions along edges occur discontinuously at discrete time points (e.g., due to sudden environmental change); in such scenarios the flow algorithm may be more suitable than differential equation models.

A revised algorithm for dSSE models based on “flow”

In the following, we illustrate how our flow algorithm can be used to efficiently calculate the likelihood of dSSE models. As explained above, the flow algorithm only requires that the differential equation for the likelihoods along each edge, abstracted as in Eq. (10), is linear and has coefficients that are independent of the particular edge. This condition is satisfied for dSSE models with non-cladogenetic transitions (Eq. 9) as well as for dSSE models with cladogenetic transitions (Goldberg and Igić, 2012; Magnuson-Ford and Otto, 2012). Note that all dSSE models are based on equations analogous to MuSSE (Eqs. 8 and 9), although they may differ in the interpretation of states, the initial conditions at the tips, the weighting of the likelihoods at the root, and how likelihoods are combined at each node; for models with cladogenetic transitions, additional terms are included in the differential equations. The flow algorithm exemplified below for MuSSE can thus also be applied to all other dSSE models mentioned above.

The algorithm begins by solving the differential equation (8) to obtain the trajectory of the extinction probabilities $E_i(t)$ from the present to the root age. For given computed $E_1, .., E_S$ (where S is the number of diversification-modulating states), the differential equations for $\mathbf{X}$ along any edge (Eq. 9) can then be written in matrix notation:

$$\frac{d\mathbf{X}}{dt} = \mathbb{A}(t) \cdot \mathbf{X}(t), \tag{19}$$

where $\mathbb{A}(t)$ is a time-dependent $S \times S$ matrix containing the various coefficients from Eq. (9). In the following, let $\mathbf{X}_N(t)$ denote the solution of Eq. (19) along the edge with child node N. The initial condition at the node, denoted $\mathbf{X}_N(t_N)$ where t_N is the node age, is specified based on the $\mathbf{X}_{C_1}(t_N), \mathbf{X}_{C_2}(t_N), ..$ previously computed along the child edges, as in the classical formulation of MuSSE (FitzJohn, 2012). At the tips, the initial conditions for X_i depend on the species sampling fractions as well as on the probability that a species in state i would have a known state, conditional upon being included in the tree (see Appendix 1. for details).

For any age t, let $\mathbb{G}(t)$ be an $S \times S$ matrix satisfying the following differential equation:

$$\frac{d\mathbb{G}}{dt} = \mathbb{A}(t) \cdot \mathbb{G}(t), \tag{20}$$

with initial condition $\mathbb{G}(0) = \mathbb{I}$. In the terminology of the previous section, $\mathbb{G}(t)$ corresponds to the likelihood flow $\Psi(0, t)$, that is, mapping initial conditions (initial likelihoods) at age 0 to solutions of the differential equation (19) at age t. We reiterate that Eq. (20) can be solved for all $t > 0$ regardless of tree structure and passing only once through each time point. Then, for any given edge connecting the parent node P (at age t_P) and child node N (at age t_N), and for any given initial condition $\mathbf{X}_N(t_N)$, one can directly compute the solution of the differential equation (19) at t_P using simple matrix algebra:

$$\mathbf{X}_N(t_P) = \mathbb{G}(t_P) \cdot \mathbf{X}_N^o, \tag{21}$$

where $\mathbf{X}_N^o$ is the solution to the linear system:

$$\mathbb{G}(t_N) \cdot \mathbf{X}_N^o = \mathbf{X}_N(t_N). \tag{22}$$

Equation (22) is Eq. (16) for the special case where $t_o = 0$, $s = t_N$, $\mathbf{X}_o = \mathbf{X}_N^o$, $\mathbf{X}_1 = \mathbf{X}_N(t_N)$ and $\mathbb{G}(t_N) = \Psi(0, t_N)$.

Observe that, for any given edge, we have replaced the need to solve the differential equation (19) along the edge with the need to solve a linear system of S equations (Eq. 22) and performing a matrix multiplication (Eq. 21). As a tradeoff, we need to pre-compute $\mathbb{G}(t)$ for all ages up until the root (Eq. 20). As we demonstrate below in our simulations, for large trees this approach tends to be computationally much more efficient, despite the slight initial overhead of calculating $\mathbb{G}$. Indeed, the time needed to calculate $\mathbb{G}$ scales linearly with the age span of the tree, which itself scales only sub-linearly (typically logarithmically) with tree size.

We mention that the above algorithm can in principle also be extended to quantitative-trait-dependent speciation and extinction models (QuaSSE; FitzJohn, 2010), by replacing the matrix-valued differential equation in Eq. (20) with a partial differential equation, and by replacing matrix multiplications as in Eq. (21) with convolution integrals. The situation, however, becomes complicated when writing the linear system in Eq. (22) in integral form, because deconvolutions tend to be hard inverse problems (Groetsch and Groetsch, 1993). Discretizing the continuous trait, in order to solve these differential and integral equations numerically, essentially would lead back to the case of discrete-trait SSEs. Developing efficient algorithms for this limit of large S is undoubtedly a separate challenge.

Evaluation and comparison to other dSSE implementations

We implemented the above algorithm for dSSE likelihoods in the R package castor (Louca and Doebeli, 2017), a project devoted to making established phylogenetics methods accessible to modern large datasets using redesigned algorithms. Our implementation can calculate the likelihood of a model for a specific set of parameters, but can also perform maximum-likelihood estimation of model parameters and parametric bootstrapping for estimating confidence intervals. castor supports an arbitrary number of states (S), missing and potentially biased information on tip states (Appendix 1.), incomplete and potentially biased species sampling, and an arbitrary number of concealed and observed states (when expressed in the terminology of van Els *et al.*, 2018). In contrast to existing methods, castor fully supports trees containing multifurcations, a common issue in massive phylogenies. Further, castor can fit models using multiple alternative start parameters to reduce the risk of local non-global likelihood optima, and can do so by using multiple CPU cores in parallel. For further details regarding the numerical implementation of the flow algorithm see Supplement S.1, for pseudocode see Supplement S.2 (available on Dryad at http://dx.doi.org/10.5061/dryad.6vm72sm).

To confirm that our flow algorithm is correct and that our numerical implementation is accurate, we performed simulations of BiSSE models with random parameters and compared the log-likelihoods calculated for the simulated data using castor and another popular R package, diversitree (FitzJohn, 2012) (Supplement S.4). We found that across all simulated trees the log-likelihoods were practically identical between the two methods, with relative differences always below 0.01% (Supplemental Fig. S1). We also performed maximum-likelihood estimations of model parameters for simulated BiSSE and HiSSE models, and compared the resulting parameter estimates to their true (known) values, using castor and two other implementations: diversitree (BiSSE only) and hisse (BiSSE and HiSSE) (Beaulieu and O'Meara, 2016). Using the same optimization parameters across packages, we found that parameter estimates by castor were generally similarly accurate as those of other tested packages, when measured in terms of the relative estimation error (Supplemental Figs. S2 and S3). This was true regardless of the parameter considered (λ_i, μ_i or Q_{ij}), and regardless of tree size.

To compare the computation time of castor to alternative implementations we performed benchmarks with trees and tip states simulated under the BiSSE or HiSSE model with randomly chosen parameters. The following implementations were considered: diversitree (BiSSE only), hisse (BiSSE and HiSSE) and secsse (HiSSE only) (van Els *et al.*, 2018). For each tree, we counted the time needed by each method to calculate the likelihood of the original model given the simulated data; for any given tree size, we calculated the average time needed by each method across multiple trees of that size (Figs. 2a,b, details in Supplement S.3). As becomes evident in Figs. 2a,b, on large trees castor clearly outperforms existing implementations, reducing computation time by one or more orders of magnitude, depending on the methods compared and depending on the size of the tree. For BiSSE, all tested methods (castor, diversitree and hisse) exhibit roughly asymptotically linear scaling with tree size. Towards larger trees (>200 tips), all methods differ from each other by a roughly constant speedup factor (Fig. 2a), with castor being on average 12 times faster than diversitree and about 500 times faster than hisse. Towards small trees, castor's run time does not converge to zero as fast as diversitree, and diversitree is somewhat faster than castor for trees with fewer than 200 tips. This is because for small trees castor's computation time is mostly allocated to solving the differential equations for $E_i(t)$ and $\mathbb{G}(t)$ and for preparing the interpolation of $\mathbb{G}(t)$ for the subsequent postorder traversal. castor's algorithm becomes increasingly advantageous for larger trees, where the initial preparations become less important and computation time is dominated by the postorder traversal. When compared to diversitree the speedup in castor is largely attributable to the advantages of the flow algorithm itself, whereas when compared to hisse the much greater performance of castor also partly results from castor's more efficient code (as described by Louca and Doebeli, 2017).

For HiSSE models, both hisse and secsse exhibit a super-linear scaling of computation times with tree size, whereas castor maintains linear scaling (Fig. 2b). For large trees containing hundreds of thousands of tips, hisse and secsse are about 1,000–10,000 times slower than castor; this difference further increases for larger trees. For example, hisse would require about 3 hours and secsse about 50 hours for a tree with one million tips for a single evaluation of the likelihood function, compared to castor which requires about 12 seconds. Since castor treats HiSSE internally as a variant of MuSSE, its computational complexity scales similarly to BiSSE, although castor remains faster than hisse and secsse even for small trees. We note that the dramatic speedup of castor compared to hisse and secsse, when applied to HiSSE models (Fig. 2b), only partly results from the theoretical advantages of the flow algorithm. Indeed, the computation time of the original HiSSE algorithm, in which the differential equation (19) is solved along each edge, should in principle scale roughly linearly with the number of tips (assuming that the number of tips grows exponentially with the age of the clade). Hence, the super-linear scaling of hisse's and secsse's computation times can likely be avoided with improved code.

To exemplify the application of our implementation to real data, we investigated the diversification of angiosperms depending on their woodiness (woody vs. herbaceous), using a previously published dated tree (31,749 tips) and associated trait data (Zanne *et al.*, 2014). We fitted a BiSSE model via maximum-likelihood, while allowing each λ_i, μ_i and Q_{ij} to differ from one another. To reduce the risk of local non-global likelihood maxima, fitting was repeated 20 times using random start parameters. This task took about 4 hours on our MacBook Pro laptop; the next-fastest implementation available would have taken about two days for the same task (Fig. 2a). Estimated speciation rates were $\sim 5.1\ \mathrm{Myr}^{-1}$ for herbaceous plants and $\sim 2.3\ \mathrm{Myr}^{-1}$ for woody plants, with respective extinction rates almost identical to (but slightly below) speciation rates. Estimated transition rates were $\sim\ 0.0043\ \mathrm{Myr}^{-1}$ from herbaceous to woody and $\sim\ 0.0036\ \mathrm{Myr}^{-1}$ from woody to herbaceous, suggesting that transitions between the two growth types are rare and approximately equally likely. We note that BiSSE models do not account for temporal or geographical variations in speciation/extinction rates, previously suggested to occur in angiosperms (Jansson and Davies, 2008; Crisp and Cook, 2011). The present analysis should thus only serve to illustrate the application of our method to massive trees. The complete R code used is available as Supplemental Code 1.

Efficient simulation of dSSE models

An evaluation of dSSE models for large trees (e.g., Supplement S.5), as well as complementary analyses such as parametric bootstrapping for estimating confidence intervals and Monte Carlo integration, necessitate efficient dSSE simulators for large trees, which are currently lacking. We thus also implemented an algorithm for simulating large dSSE models in forward-time, sometimes referred to as "simple sampling approach" (Stadler, 2011b) (pseudocode in Supplement S.6). Our implementation is orders of magnitude faster than any existing implementation, and can handle both anagenetic and cladogenetic transitions between states, thus covering the broad range of dSSE models discussed. In our tests castor was able to generate BiSSE trees with millions of tips in about 5 seconds (Fig. 2c, details in Supplement S.7). When assessed over varying tree sizes, it becomes apparent that our algorithm exhibits a nearly linear scaling of computation time (power-law exponent 1.1). In contrast, diversitree and hisse (to our knowledge the only other R packages able to simulate dSSE models) both exhibit super-linear scaling (exponents 2.0 and 1.5, respectively), and would require 10–100 hours to generate a tree with 1 million tips (Fig. 2c). Even when compared to existing simulators of simple uniform time-homogeneous birth-death processes, i.e. ignoring trait evolution and assuming a single constant λ and μ, castor's simulations of the more general BiSSE process are at least an order of magnitude faster than other tools (Supplemental Fig. S4). For example, to generate a tree with 1 million tips under the simple birth-death model, and based on the fitted scaling exponents, we estimate that the package geiger (Pennell *et al.*, 2014) would take on average about 73 hours, the package TESS (Höhna *et al.*, 2015) about 7 hours, the package phytools (Revell, 2012) about 32 hours and the package TreeSim (Stadler, 2011b) about 112 hours (although we caution the reader that TreeSim and TESS use a different conditioning and thus do not sample from the same exact distribution as castor; Stadler, 2011b). The correctness of our code is confirmed by comparing the distribution of generated trees to those generated by diversitree, both in terms of their lineages-through-time curves as well as the distribution of pairwise node-to-node distances (Supplement S.8 and Supplemental Fig. S5).

Similarly to most previous implementations, our implementation is essentially a Gillespie algorithm, which generates statistically correctly distributed trajectories of the stochastic model (Gillespie, 1977). In Gillespie algorithms, the waiting time until the next event — in our case speciation, extinction or transition between states — is randomly drawn from an exponential distribution according to the rates of the various processes. Variants of the Gillespie algorithm are also used in other implementations, such as diversitree, hisse, phytools and geiger. The greater efficiency of our implementation is achieved in mainly two ways. First, we use temporary auxiliary redundant data structures, which are either generated at the beginning (in linear time) or continuously updated after each event (in constant time), to accelerate certain operations and eliminate redundant calculations. This enables us to achieve the linear scaling that is theoretically predicted for the Gillespie algorithm. For example, during a simulation we keep track of lineages that are not yet extinct and in a particular state using continuously updated lookup tables (Louca and Doebeli, 2017); hence, choosing the next tip for a speciation/extinction/transition event can be done in constant time. Indeed, a common issue that we observed in other implementations is the repeated use of function calls (such as the R function which) that iterate through the entire tree at each event, thus leading to a needless super-linear scaling of overall computation time. Second, our code is almost entirely written in C++, a programming language that is especially well suited for high-performance computing. Note that the algorithms underlying TreeSim and TESS should, in theory, also scale linearly with tree size and should be comparable (if not faster) than the Gillespie algorithm, since they sample only branching times in the extant phylogeny rather than all speciation/extinction events (Stadler, 2011b; Höhna *et al.*, 2015). The lower performance and super-quadratic scaling of TreeSim and TESS (power-law exponent $\sim$2.2) thus likely result from sub-optimal code design, and could perhaps be improved using similar approaches as in castor (Louca and Doebeli, 2017).

Conclusions

An impressive number of mathematical methods have been developed for comparative phylogenetics over the last few decades (O'Meara, 2012; Pennell and Harmon, 2013; Garamszegi, 2014; Morlon, 2014; Ng and Smith, 2014; Harmon, 2018). However, existing numerical procedures for the majority of these methods — while adequate in the past — scale poorly to increasingly common large-scale phylogenies. As biology ventures into an era of massive datasets, and bottlenecks become increasingly computational, a deeper consideration of algorithmic complexity and numerical limitations is needed in order to keep these mathematical tools applicable (Freckleton, 2012; Tung Ho and Ané, 2014; Goolsby, 2017; Louca and Doebeli, 2017).

Here we present a new algorithm for calculating the likelihood of a large set of macroevolutionary models, including diversification models with time-dependent speciation and extinction rates (Morlon *et al.*, 2011; Condamine *et al.*, 2013; Rabosky, 2014), models for discrete-state-dependent diversification (Maddison *et al.*, 2007; FitzJohn *et al.*, 2009; Rabosky and Glor, 2010; Goldberg *et al.*, 2011; FitzJohn, 2012; Goldberg and Igić, 2012; Magnuson-Ford and Otto, 2012; Beaulieu and O'Meara, 2016; van Els *et al.*, 2018;

Caetano *et al.*, 2018) and Mk models for the evolution of a discrete trait along a fixed tree (Pagel, 1994; Lewis, 2001). Our algorithm makes use of the fact that the solutions to the Kolmogorov backward equation, a cornerstone of the aforementioned models, can be represented by a likelihood flow that only needs to be computed once from the present to the root. Our flow algorithm can also be applied to dSSE models where the rates λ_i, μ_i and Q_{ij} depend on time and/or on time-dependent environmental variables (Rabosky and Glor, 2010), as well as to Mk models with time-dependent transition rates. The algorithm can even be applied to cases where rate parameters vary between some taxa (Alfaro *et al.*, 2009; Rabosky, 2014), so long as a separate flow is used for each taxon-specific rate class. The flow algorithm has the potential to substantially reduce computation time whenever it is costly to solve the Kolmogorov backward differential equation for the likelihoods, as is typically the case for complex models for which no closed-form solutions are known.

Our tests showed that for large trees our algorithm, which we used to newly implement various previous dSSE models, is one or more orders of magnitude faster than existing implementations. For maximum-likelihood estimation or Bayesian MCMC, which can involve thousands of evaluations of the likelihood function and take hundreds of hours, these differences in performance become crucial determinants of the feasibility of a study. We also presented a new numerical method for simulating dSSE models, which for large trees is several orders of magnitude faster than existing methods. Our methods are provided through `castor`, an R package for efficient comparative phylogenetics on large trees (Louca and Doebeli, 2017). Practically, our methods make a large class of diversification models accessible to modern massive phylogenies, which are bound to shed new light on macroevolutionary questions. We also hope that our algorithm gives researchers a new perspective on macroevolutionary models and, in doing so, helps spur the advance of the next-generation of comparative methods.

Supplementary Material

Supplementary material, including online-only appendices, supplemental figures and example computer code, is available at the Dryad Digital Repository at: `http://dx.doi.org/10.5061/dryad.6vm72sm`

Code availability

Methods described in this article for dSSE models are available in the R package `castor`, as functions `fit_musse` and `simulate_musse`. The package `castor` is available on The Comprehensive R Archive Network (CRAN).

Funding

S.L. was supported by a startup grant by the University of Oregon, USA. M.W.P. was supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant.

Author contributions

S.L. and M.W.P. conceptualized the project. S.L. conceived the flow algorithm and wrote the computer code. Both authors contributed to the writing of the manuscript.

Competing financial interests

The authors declare that they have no competing interests.

Materials & Correspondence

Correspondence and requests for materials should be addressed to S.L.

References

Alfaro, M.E., Santini, F., Brock, C., Alamillo, H., Dornburg, A., Rabosky, D.L., *et al.* (2009) Nine exceptional radiations plus high turnover explain species diversity in jawed vertebrates. *Proceedings of the National Academy of Sciences* **106**: 13410–13414.

Arnold, L. (2013) Random Dynamical Systems. Springer Monographs in Mathematics. Springer Berlin Heidelberg.

Beaulieu, J.M. and O'Meara, B.C. (2016) Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. *Systematic biology* **65**: 583–601.

Caetano, D.S., O'Meara, B.C., and Beaulieu, J.M. (2018) Hidden state models improve state-dependent diversification approaches, including biogeographical models. *Evolution* **72**: 2308–2324.

Cantalapiedra, J.L., FitzJohn, R.G., Kuhn, T.S., Fernández, M.H., DeMiguel, D., Azanza, B., *et al.* (2014) Dietary innovations spurred the diversification of ruminants during the Caenozoic. *Proceedings of the Royal Society B: Biological Sciences* **281**: 20132746.

Condamine, F.L., Rolland, J., and Morlon, H. (2013) Macroevolutionary perspectives to environmental change. *Ecology Letters* **16**: 72–85.

Cornwell, W.K., Pearse, W.D., Dalrymple, R.L., and Zanne, A.E. (2018) What we (don't) know about global plant diversity. *BioRxiv* p. 404376.

Crisp, M.D. and Cook, L.G. (2011) Cenozoic extinctions account for the low diversity of extant gymnosperms compared with angiosperms. *New Phytologist* **192**: 997–1009.

David, L.A. and Alm, E.J. (2011) Rapid evolutionary innovation during an archaean genetic expansion. *Nature* **469**: 93–96.

Dunbar, J., White, S., and Forney, L. (1997) Genetic diversity through the looking glass: Effect of enrichment bias. *Applied and Environmental Microbiology* **63**: 1326–1331.

Etienne, R.S. and Haegeman, B. (2012) A conceptual and statistical framework for adaptive radiations with a key role for diversity dependence. *The American Naturalist* **180**: E75–E89.

Feller, W. (1949) On the theory of stochastic processes, with particular reference to applications. In Proceedings of the First Berkeley Symposium on Mathematical Statistics and Probability, pp. 403–432. Berkeley, California: University of California Press.

FitzJohn, R.G. (2010) Quantitative traits and diversification. *Systematic Biology* **59**: 619–633.

FitzJohn, R.G. (2012) Diversitree: comparative phylogenetic analyses of diversification in R. *Methods in Ecology and Evolution* **3**: 1084–1092.

FitzJohn, R.G., Maddison, W.P., and Otto, S.P. (2009) Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. *Systematic Biology* **58**: 595–611.

Freckleton, R.P. (2012) Fast likelihood calculations for comparative analyses. *Methods in Ecology and Evolution* **3**: 940–947.

Garamszegi, L.Z. (2014) Modern phylogenetic comparative methods and their application in evolutionary biology: concepts and practice. Berlin Heidelberg: Springer.

Gillespie, D.T. (1977) Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* **81**: 2340–2361.

Goldberg, E.E. and Igić, B. (2012) Tempo and mode in plant breeding system evolution. *Evolution* **66**: 3701–3709.

Goldberg, E.E., Lancaster, L.T., and Ree, R.H. (2011) Phylogenetic inference of reciprocal effects between geographic range evolution and diversification. *Systematic Biology* **60**: 451–465.

Goolsby, E.W. (2017) Rapid maximum likelihood ancestral state reconstruction of continuous characters: A rerooting-free algorithm. *Ecology and Evolution* **7**: 2791–2797.

Groetsch, C.W. and Groetsch, C. (1993) Inverse problems in the mathematical sciences, volume 52. Wiesbaden: Springer.

Harmon, L.J. (2018) Phylogenetic Comparative Methods: Learning from Trees. Self published under a CC-BY-4.0 license.
URL `https://lukejharmon.github.io/pcm/`

Hehemann, J.H., Arevalo, P., Datta, M.S., Yu, X., Corzett, C.H., Henschel, A., *et al.* (2016) Adaptive radiation by waves of gene transfer leads to fine-scale resource partitioning in marine microbes. *Nature Communications* **7**: 12860 EP.

Henao Diaz, L.F., Harmon, L.J., Sugawara, M.T.C., Miller, E.T., and Pennell, M.W. (2019) Macroevolutionary diversification rates show time dependency. *Proceedings of the National Academy of Sciences* **116**: 7403.

Hinchliff, C.E., Smith, S.A., Allman, J.F., Burleigh, J.G., Chaudhary, R., Coghill, L.M., *et al.* (2015) Synthesis of phylogeny and taxonomy into a comprehensive tree of life. *Proceedings of the National Academy of Sciences* **112**: 12764–12769.

Höhna, S., May, M.R., and Moore, B.R. (2015) TESS: an R package for efficiently simulating phylogenetic trees and performing Bayesian inference of lineage diversification rates. *Bioinformatics* **32**: 789–791.

Jansson, R. and Davies, T.J. (2008) Global variation in diversification rates of flowering plants: energy vs. climate change. *Ecology Letters* **11**: 173–183.

Jetz, W., Thomas, G., Joy, J., Hartmann, K., and Mooers, A. (2012) The global diversity of birds in space and time. *Nature* **491**: 444.

Kendall, D.G. (1948) On some modes of population growth leading to RA Fisher's logarithmic series distribution. *Biometrika* **35**: 6–15.

Kolmogorov, A. (1931) Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung. *Mathematische Annalen* **104**: 415–458.

Lagkouvardos, I., Overmann, J., and Clavel, T. (2017) Cultured microbes represent a substantial fraction of the human and mouse gut microbiota. *Gut Microbes* **8**: 493–503.

Larsen, B.B., Miller, E.C., Rhodes, M.K., and Wiens, J.J. (2017) Inordinate fondness multiplied and redistributed: the number of species on Earth and the new pie of life. *The Quarterly Review of Biology* **92**: 229–265.

Latysheva, N., Junker, V.L., Palmer, W.J., Codd, G.A., and Barker, D. (2012) The evolution of nitrogen fixation in cyanobacteria. *Bioinformatics* **28**: 603–606.

Lewis, P.O. (2001) A likelihood approach to estimating phylogeny from discrete morphological character data. *Systematic biology* **50**: 913–925.

Louca, S. and Doebeli, M. (2017) Efficient comparative phylogenetics on large trees. *Bioinformatics* **34**: 1053–1055.

Louca, S., Shih, P.M., Pennell, M.W., Fischer, W.W., Parfrey, L.W., and Doebeli, M. (2018) Bacterial diversification through geological time. *Nature Ecology & Evolution* **2**: 1458–1467.

Maddison, W.P., Midford, P.E., Otto, S.P., and Oakley, T. (2007) Estimating a binary character's effect on speciation and extinction. *Systematic Biology* **56**: 701–710.

Magallon, S. and Sanderson, M.J. (2001) Absolute diversification rates in angiosperm clades. *Evolution* **55**: 1762–1780.

Magnuson-Ford, K. and Otto, S.P. (2012) Linking the investigations of character evolution and species diversification. *The American Naturalist* **180**: 225–245.

Marchesi, J.R. and Weightman, A.J. (2003) Comparing the dehalogenase gene pool in cultivated α-halocarboxylic acid-degrading bacteria with the environmental metagene pool. *Applied and Environmental Microbiology* **69**: 4375–4382.

Meiss, J. (2007) Differential Dynamical Systems. Number v. 1 in Monographs on Mathematical Modeling and Computation. Philadelphia, USA: Society for Industrial and Applied Mathematics.

Mendler, K., Chen, H., Parks, D.H., Hug, L.A., and Doxey, A.C. (2018) Annotree: visualization and exploration of a functionally annotated microbial tree of life. *bioRxiv* .

Mooers, A.O. and Heard, S.B. (1997) Inferring evolutionary process from phylogenetic tree shape. *The Quarterly Review of Biology* **72**: 31–54.

Mora, C., Tittensor, D.P., Adl, S., Simpson, A.G., and Worm, B. (2011) How many species are there on Earth and in the ocean? *PLoS biology* **9**: e1001127.

Morlon, H. (2014) Phylogenetic approaches for studying diversification. *Ecology Letters* **17**: 508–525.

Morlon, H., Parsons, T.L., and Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record. *Proceedings of the National Academy of Sciences* **108**: 16327–16332.

Morlon, H., Potts, M.D., and Plotkin, J.B. (2010) Inferring the dynamics of diversification: A coalescent approach. *PLOS Biology* **8**: e1000493.

Muscarella, M.E. and O'dwyer, J.P. (2018) Ecological insights from the evolutionary history of microbial innovations. *BioRxiv* p. 220939.

Nee, S., May, R.M., and Harvey, P.H. (1994) The reconstructed evolutionary process. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* **344**: 305–311.

Ng, J. and Smith, S.D. (2014) How traits shape trees: new approaches for detecting character state-dependent lineage diversification. *Journal of Evolutionary Biology* **27**: 2035–2045.

Olver, P.J. (2012) Applications of Lie Groups to Differential Equations. Graduate Texts in Mathematics. Springer New York.

O'Meara, B.C. (2012) Evolutionary inferences from phylogenies: a review of methods. *Annual Review of Ecology, Evolution, and Systematics* **43**: 267–285.

Pagel, M. (1994) Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. *Proceedings of the Royal Society of London B: Biological Sciences* **255**: 37–45.

Parks, D.H., Chuvochina, M., Waite, D.W., Rinke, C., Skarshewski, A., Chaumeil, P.A., *et al.* (2018) A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nature Biotechnology* **36**: 996–1004.

Pennell, M.W., Eastman, J.M., Slater, G.J., Brown, J.W., Uyeda, J.C., FitzJohn, R.G., *et al.* (2014) geiger v2. 0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees. *Bioinformatics* **30**: 2216–2218.

Pennell, M.W. and Harmon, L.J. (2013) An integrative view of phylogenetic comparative methods: connections to population genetics, community ecology, and paleobiology. *Annals of the New York Academy of Sciences* **1289**: 90–105.

Polz, M.F., Alm, E.J., and Hanage, W.P. (2013) Horizontal gene transfer and the evolution of bacterial and archaeal population structure. *Trends in Genetics* **29**: 170–175.

Rabosky, D.L. (2014) Automatic detection of key innovations, rate shifts, and diversity-dependence on phylogenetic trees. *PLOS ONE* **9**: 1–15.

Rabosky, D.L., Donnellan, S.C., Talaba, A.L., and Lovette, I.J. (2007) Exceptional among-lineage variation in diversification rates during the radiation of australia's most diverse vertebrate clade. *Proceedings of the Royal Society of London B: Biological Sciences* **274**: 2915–2923.

Rabosky, D.L. and Glor, R.E. (2010) Equilibrium speciation dynamics in a model adaptive radiation of island lizards. *Proceedings of the National Academy of Sciences* **107**: 22178–22183.

Raup, D.M. (1985) Mathematical models of cladogenesis. *Paleobiology* **11**: 42–52.

Revell, L.J. (2012) phytools: an r package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution* **3**: 217–223.

Schluter, D. and Pennell, M.W. (2017) Speciation gradients and the distribution of biodiversity. *Nature* **546**: 48.

Smith, S.A. and Brown, J.W. (2018) Constructing a broadly inclusive seed plant phylogeny. *American Journal of Botany* **105**: 302–314.

Stadler, T. (2009) On incomplete sampling under birth–death models and connections to the sampling-based coalescent. *Journal of Theoretical Biology* **261**: 58–66.

Stadler, T. (2011a) Mammalian phylogeny reveals recent diversification rate shifts. *Proceedings of the National Academy of Sciences* **108**: 6187–6192.

Stadler, T. (2011b) Simulating trees with a fixed number of extant species. *Systematic Biology* **60**: 676–684.

Stadler, T. and Steel, M. (2019) Swapping birth and death: Symmetries and transformations in phylodynamic models. *bioRxiv* p. 494583.

Tamaki, H., Sekiguchi, Y., Hanada, S., Nakamura, K., Nomura, N., Matsumura, M., *et al.* (2005) Comparative analysis of bacterial diversity in freshwater sediment of a shallow eutrophic lake by molecular and improved cultivation-based techniques. *Applied and Environmental Microbiology* **71**: 2162–2169.

Thompson, L.R., Sanders, J.G., McDonald, D., Amir, A., Ladau, J., Locey, K.J., *et al.* (2017) A communal catalogue reveals earth's multiscale microbial diversity. *Nature* **551**: 457–463.

Tung Ho, L.S. and Ané, C. (2014) A linear-time algorithm for Gaussian and non-Gaussian trait evolution models. *Systematic Biology* **63**: 397–408.

van Els, P., Etienne, R.S., and Herrera-Alsina, L. (2018) Detecting the dependence of diversification on multiple traits from phylogenetic trees and trait data. *Systematic Biology* .

Wiens, J.J. (2017) What explains patterns of biodiversity across the tree of life? new research is revealing the causes of the dramatic variation in species numbers across branches of the tree of life. *BioEssays* **39**: 1600128.

Zanne, A.E., Tank, D.C., Cornwell, W.K., Eastman, J.M., Smith, S.A., FitzJohn, R.G., *et al.* (2014) Three keys to the radiation of angiosperms into freezing environments. *Nature* **506**: 89–92.

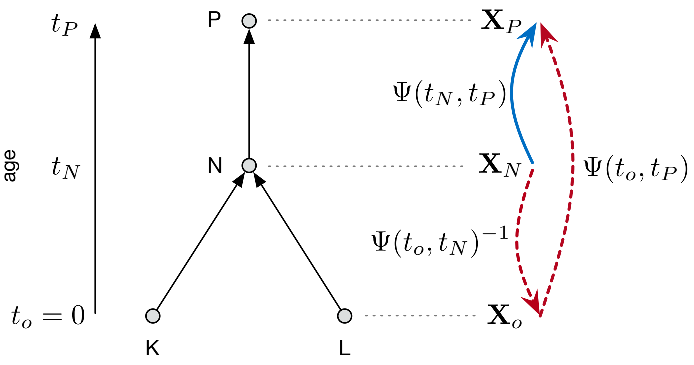

Figure 1: **Conceptual illustration of the revised algorithm.** Our proposed algorithm can be used to rapidly compute variables (e.g., state likelihoods) at nodes that are a priori defined as solutions of a linear differential equation along edges. In the example illustrated above, as in dSSE models, the initial condition for the vector-valued variable $\mathbf{X}$ at node N is determined based on the values of $\mathbf{X}$ calculated previously on the descending edges (connecting tips L and K). The goal is to obtain the solution $\mathbf{X}_P$ of the differential equation along edge $N \rightarrow P$, given the initial condition $\mathbf{X}_N$ at node N. Instead of explicitly solving this differential equation along the edge, one can calculate $\mathbf{X}_P$ as $\mathbf{X}_P = \Psi(t_o, t_P) \cdot \Psi(t_o, t_N)^{-1} \cdot \mathbf{X}_N$, where t_o is some starting time point (e.g. $t_o = 0$) for which the flow $\Psi(t_o, t)$ has been previously calculated. The direct mapping $\Psi(t_N, t_P)$ (continuous arrow) is equivalent to the product $\Psi(t_o, t_P) \cdot \Psi(t_o, t_N)^{-1}$ (dashed arrows), and hence an explicit calculation of $\Psi(t_N, t_P)$ is not necessary.

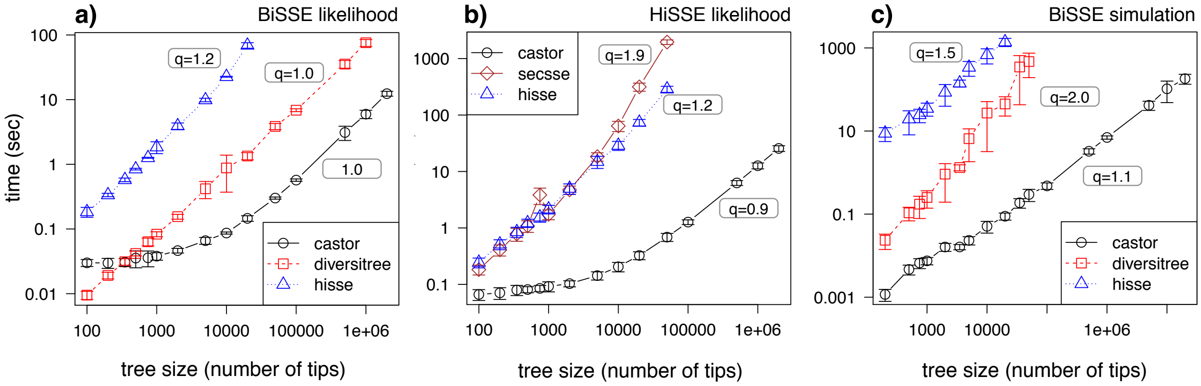

Figure 2: **Comparison of computation times.** Comparison of computation times needed for the calculation of a single BiSSE likelihood (a), the calculation of a single HiSSE likelihood (b), and the simulation of a single BiSSE model (c), using `castor` and other software packages (time T over tree size S, one curve per package). Note the logarithmic axes in all figures. Package names are listed in the legends. Fitted asymptotic power-law exponents ($T \propto S^q$) are shown next to every curve. Vertical bars show standard deviations. Compared packages include `diversitree` (FitzJohn, 2012), `hisse` (Beaulieu and O'Meara, 2016) and `secsse` (van Els *et al.*, 2018). Detailed functions and options used are explained in the Methods.

Appendix 1. An extension to existing dSSE models

In the following, we describe how existing dSSE models can be extended to account for multifurcations in the phylogeny (a common issue in massive phylogenies), as well as for potential biases in the identification of tip states, i.e. where some states are easier to detect/confirm than others. Such "reveal biases" are probably present in a multitude of traits, for example when it is easier to confirm the presence of a behavioral trait or metabolic capability than its absence. A prominent example are bacterial metabolic phenotypes, where culturing success is strongly biased towards specific phenotypes, depending on available techniques and current research interests (Dunbar *et al.*, 1997; Marchesi and Weightman, 2003; Tamaki *et al.*, 2005; Lagkouvardos *et al.*, 2017). Existing formulations of dSSE ignore such biases, that is, it has so far been assumed that either all tips have known state (Maddison *et al.*, 2007), or that all tip states have the same probability of being known (FitzJohn *et al.*, 2009; FitzJohn, 2012; Beaulieu and O'Meara, 2016; van Els *et al.*, 2018).

In dSSE, available trait data and information about sampling fractions is incorporated into a model's likelihood via the initial conditions assumed for the E_i (probability that a lineage at state i will eventually be absent from the tree) and the likelihoods X_i at the tips, i.e. at age $t = 0$. Specifically, the initial conditions for E_i at $t = 0$ depend on the fraction of extant species in state i that is included in the phylogeny, usually referred to as "sampling fraction" and denoted ρ_i:

$$E_i(0) = 1 - \rho_i. \tag{23}$$

The initial condition for X_i at $t = 0$ is defined separately for each tip, and depends on the sampling fraction of species in state i, as well as on the probability that a species in state i would have a known ("revealed") state, conditional upon being included in the tree. In existing dSSE variants, this probability — here referred to as "reveal fraction" r_i — was assumed to be independent of state (Maddison *et al.*, 2007; FitzJohn *et al.*, 2009; FitzJohn, 2012; Beaulieu and O'Meara, 2016; van Els *et al.*, 2018). For tips known to be in state i, thus:

$$X_i(0) = \rho_i \cdot r_i, \quad X_j(0) = 0 \quad \forall j \neq i, \tag{24}$$

and for tips with unknown state:

$$X_j(0) = \rho_j \cdot (1 - r_j) \quad \forall j. \tag{25}$$

Observe that even if the state of a tip is unknown, the mere fact that it is included in the phylogeny (which occurs with probability ρ_i) and the fact that its state is unknown (which occurs with probability $1 - r_i$), constitute potential information that is incorporated into the model's likelihood. In Supplement S.9 we use simulations to illustrate how ignoring or accounting for reveal biases can influence dSSE parameter estimates. Note that, unless most other model parameters are known a priori, it may not be possible to estimate the reveal fractions r_i from the phylogenetic data alone. For example, for trait-independent birth-death models it is well known that the sampling fraction ρ cannot be directly estimated from the tree when the speciation and extinction rates are unknown (Stadler, 2009; Morlon *et al.*, 2010; Stadler and Steel, 2019), and that ρ must be determined using additional information (e.g, via mark-recapture-type surveys, Louca *et al.*, 2018). It is thus possible that the r_i may also need to be determined beforehand using additional data, although more thorough investigations of parameter identifiability are required to confirm this suspicion.

At internal nodes, the initial conditions for $\mathbf{X}$ depend on the values computed for the descending clades, as explained in the following. Since we will be referring to the values of $\mathbf{X}$ at various notes, in the following we will deploy double-indices, with $X_{N,j}$ denoting the value of the j-th component of $\mathbf{X}$ (where $j = 1, .., S$) at node N, and with $\mathcal{N}$ denoting the set of all nodes (hence $N \in \mathcal{N}$). At any node $N \in \mathcal{N}$ of age t_N and having child nodes $C_1, .., C_n \in \mathcal{N}$ (with $n = 2$ in the case of bifurcating trees), the initial condition $X_{N,i}(t_N)$ is determined by the final values of $X_{C_1,i}, .., X_{C_n,i}$ on the daughter lineages:

$$X_{N,i}(t_N) = \lambda_i^{n-1} \prod_{k=1}^{n} X_{C_k,i}(t_N). \tag{26}$$

Note that classical formulations of dSSE only consider the bifurcating case ($n = 2$). The more general scaling factor λ_i^{n-1} for any $n \geq 2$ can be derived in two alternative ways. First, multifurcations can be decomposed into $n-1$ bifurcating sub-nodes in close temporal proximity, i.e., with the length of artificially introduced edges being infinitesimally small. Along these edges, the likelihoods $\mathbf{X}$ change only little, and at each sub-node a factor λ_i would be introduced as one traverses towards the root; the classical formula for dSSE in bifurcating trees would thus eventually lead to the expression in Eq. (26) for the likelihoods at the sub-node closest to the root. Second, the scaling λ_i^{n-1} corresponds to the leading-order expression for the probability that the cladogenic process generates n lineages during an infinitesimal time step ε, after rescaling to remove time units. Indeed, for a cladogenic process with some speciation rate λ_i and extinction rate μ_i, starting with a single lineage at time t, the probability of having n or more extant lineages at time $t + \varepsilon$ is given by (Nee *et al.*, 1994):

$$P(\varepsilon) = \sum_{k=n}^{\infty} \frac{\delta_i}{\lambda_i - \mu_i e^{-\delta_i \varepsilon}} \cdot [1 - u_\varepsilon]\, u_\varepsilon^{n-1}, \tag{27}$$

where $\delta_i = \lambda_i - \mu_i$ and where:

$$u_\varepsilon := \lambda_i \frac{1 - e^{-\delta_i \varepsilon}}{\lambda_i - \mu_i e^{-\delta_i \varepsilon}}. \tag{28}$$

Keeping only terms of leading order in ε yields:

$$P(\varepsilon) = \varepsilon^{n-1} \lambda_i^{n-1} + \mathcal{O}(\varepsilon^n), \tag{29}$$

and hence one recovers the scaling factor in Eq. (26).

We clarify that Eq. (26) is primarily designed to deal with poorly resolved multifurcations, i.e. where multiple bifurcations occurred in such close temporal proximity that they cannot be resolved in the phylogeny. As most existing phylogenetic software (including most existing implementations of dSSE) crash when applied to multifurcating trees, it is common practice to artificially resolve multifurcations into multiple and closely adjacent bifurcations (i.e., by introducing short artificial edges), prior to applying dSSE. Our formula essentially allows directly calculating the outcome of such an adjustment without actually modifying the input tree, in the mathematical limit where the introduced artificial edges are infinitesimally small. Note that Eq. (26) does not apply to truly multifurcating diversification models, i.e. where true n-furcations occur at some non-zero probability rate.

A General and Efficient Algorithm for the Likelihood of Diversification and Discrete-Trait Evolutionary Models
- Supplemental Information -

Stilianos Louca[1,2] & Matthew W. Pennell[3,4]

[1] *Department of Biology, University of Oregon, USA*
[2] *Institute of Ecology and Evolution, University of Oregon, USA*
[3] *Biodiversity Research Centre, University of British Columbia, Vancouver, Canada*
[4] *Department of Zoology, University of British Columbia, Vancouver, Canada*

S.1 Numerical considerations

Mathematically, our flow algorithm for dSSE models is equivalent to the original formulation of SSE. However, the following clarifying notes are warranted regarding its numerical implementation (also see Supplement S.2 for pseudocode). First, the pre-computed solution for $E_i(t)$ is inevitably stored on a finite discrete time grid, however E_i needs to be evaluated at arbitrary times during the subsequent calculation of $\mathbb{G}$. Thus, any evaluation of E_i at a time falling between grid points is obtained through linear interpolation. Similarly, the precomputed $\mathbb{G}$ is stored on a discrete time grid, and is later evaluated at the branching times via interpolation if necessary. An accurate interpolation is ensured by using sufficiently fine time grids.

Second, the differential equation (20) tends to generate matrices $\mathbb{G}(t)$ that become increasingly difficult to invert, or equivalently, the linear system in Eq. (22) becomes increasingly difficult to solve accurately. Intuitively, trajectories of the differential equation $d\mathbf{X}/dt = \mathbb{A} \cdot \mathbf{X}$ starting at distinct initial conditions tend to gradually converge to similarly shaped (and thus linearly dependent) likelihood vectors $\mathbf{X}$ for increasing t. For illustration, suppose $\mathbb{A}$ is roughly constant over time and has eigenvalues $\sigma_1, \sigma_2, ..$, with σ_1 having the largest (i.e., most positive) real part. Then $\mathbb{G}(t)$ will have eigenvalues $\exp(t\sigma_1), \exp(t\sigma_2), ..$, with the largest modulus of any eigenvalue being $\exp(t\Re\sigma_1)$, where $\Re$ denotes the real part. Since the eigenvalues of $\mathbb{G}(t)$ will increasingly diverge from one another at an exponential rate, for almost any initial condition $\mathbf{X}(0)$ the shape of $\mathbf{X}(t) = \mathbb{G}(t)\mathbf{X}(0)$ will increasingly resemble the eigenvector corresponding to the dominant eigenvalue $\exp(t\sigma_1)$. A standard measure for how close the matrix $\mathbb{G}(t)$ is to singularity for numerical purposes is the "condition number", denoted $\kappa(t)$ (Cline *et al.*, 1979, Turing, 1948); a greater $\kappa(t)$ generally means that $\mathbb{G}(t)$ is harder to invert numerically. The condition number is given by the ratio of the largest over the smallest singular value, s_1/s_n, where $s_1, .., s_n$ are the singular values in decreasing size (Watkins, 2010). Estimating the singular values of $\mathbb{G}(t)$ without explicitly solving the differential equation (20) is hard, however an upper bound can be found. Assuming as before that $\mathbb{A}$ is constant, then the singular values of $\mathbb{G}(t)$ are related to the singular values of $\mathbb{A}$ according to the following inequality (So and Thompson, 2000, Theorem 2.1):

$$s_1(\mathbb{G}(t)) = s_1(e^{t\mathbb{A}}) \leq e^{ts_1(\mathbb{A})}, \tag{1}$$

and:

$$s_n(\mathbb{G}(t)) = \frac{1}{s_1(\mathbb{G}^{-1}(t))} = \frac{1}{s_1(e^{s_1(-t\mathbb{A})})} = \frac{1}{s_1(e^{ts_1(\mathbb{A})})} \geq \frac{1}{e^{ts_1(\mathbb{A})}} \geq e^{-ts_1(\mathbb{A})}. \tag{2}$$

Consequently, the condition number $\kappa(\mathbb{G}(t))$ can be bounded as follows:

$$\kappa(\mathbb{G}(t)) = \frac{s_1(\mathbb{G}(t))}{s_n(\mathbb{G}(t))} \leq e^{2ts_1(\mathbb{A})}. \tag{3}$$

In our implementation, we thus check whether the condition number $\kappa(t)$ could become dangerously high as t approaches the root age, based on the given $\mathbb{A}$ and using the upper bound in Eq. (3). In those scenarios, we split the tree's age interval into sufficiently short subintervals, and solve the differential equation (20) separately for each subinterval with the initial condition $\mathbb{G}(t_i) = \mathbb{I}$ applied to the beginning (t_i) of each subinterval. Eq. (21) is then adjusted by replacing $\mathbb{G}(t_P)$ with a product of matrices corresponding to subintervals between ages t_N and t_p. Further, when solving for $\mathbf{X}_N^o$ in Eq. (22), we explicitly verify that the matrix $\mathbb{G}(t_N)$ has full rank within the bounds of numerical accuracy; in the rare occasion where $\mathbb{G}(t_N)$ is rank-deficient or close to rank-deficient, we instead calculate the "best" approximate solution to (22) in the least-squares sense.

Third, the linear structure of the differential equation (20) permits the use of specialized efficient numerical differential equation integrators. Despite this useful property, existing dSSE implementations use generic integrators (mostly in the Runge-Kutta family; Butcher, 1987) that do not exploit this structure and often necessitate impractically small time steps to achieve satisfactory accuracy. Here we use a variant of exponential integrators (Friedli, 1978, Lawson, 1967), which are particularly suited for problems where time-step requirements are largely determined by the linear component, nested into an explicit 2-stage Runge-Kutta scheme. Within this scheme, each iteration $\mathbb{G}^{(n)} \to \mathbb{G}^{(n+1)}$ proceeding from age t_n to t_{n+1} is computed as:

$$\mathbb{G}^{(n+1)} = e^{\varepsilon\cdot\overline{A}(t_n)} \cdot \mathbb{G}^{(n)}(t), \tag{4}$$

where $\varepsilon = t_{n+1} - t_n$ is the time step and $\overline{A}(t_n) = 0.5 \cdot (\mathbb{A}(t_n) + \mathbb{A}(t_{n+1}))$ corresponds to the averaged dynamics at times t_n and t_{n+1}. In our implementation, the matrix exponential is approximated numerically using a finite sum of matrix polynomials, and the time step ε is adjusted adaptively at each iteration to ensure sufficient accuracy. For linear differential equations such as here, this scheme tends to be more robust than standard explicit Runge-Kutta schemes, because the exponential form strongly reduces the risk of overshooting into negative values. In our simulations, we also observed that this scheme yielded more accurate solutions than classical 2-stage Runge-Kutta schemes, for any given time step ε. This is not surprising, since the iteration in Eq. (4) yields exact solutions if $\mathbb{A}$ is constant, whereas this is not the case for classical Runge-Kutta. As the age t increases, the extinction probabilities $E_i(t)$ typically converge towards an asymptotic value and consequently $\mathbb{A}(t)$ indeed converges towards a constant matrix.

Fourth, for large trees the solution to the differential equation (20) can reach quite extreme scales, eventually leading to numerical underflow or overflow. This problem is especially common for models where speciation, extinction and/or transition rates are fast compared to the time scales covered by the tree, for example when net diversification rates were low but species turnover rates were high, or when state transitions occurred much more frequently than speciations, or when the likelihood is computed for an unlikely parameter set. Typically, the eigenvalues of $\mathbb{A}(t)$ will have negative real part, and hence $\mathbb{G}(t)$ decays exponentially fast towards zero as $t \to \infty$. In our tests we encountered cases where entries in $\mathbb{G}(t)$ become as small as $10^{-10,000}$ — such values currently cannot be represented by standard computer floating point variables. In existing dSSE implementations, similar underflow problems are encountered in the computation of $\mathbf{D}$ as one approaches the root, and are mitigated by normalizing $\mathbf{D}$ at each node and adjusting the model's log-likelihood to correct for this normalization. Here we use a similar rescaling approach when calculating $\mathbb{G}(t)$, whereby we normalize entries in $\mathbb{G}(t)$ at each iteration, keeping track of the logarithm of the cumulative rescaling applied and correcting the model's log-likelihood for this rescaling. Since the differential equation (20) is linear, this is permissible, i.e. the obtained solution is mathematically equivalent (but numerically easier to represent) to the original integration scheme in Eq. (4).

S.2 Pseudocode description of the flow algorithm

The following pseudocode outlines the flow algorithm, as implemented for MuSSE models in the package `castor`. Let bold characters (e.g. $\boldsymbol{\rho}$ or $\mathbf{X}$) denote vectors of size S, where S is the number of modeled states, and let double-lined symbols (e.g. $\mathbb{Q}$ or $\mathbb{G}$) denote matrices of size $S \times S$. For any two vectors $\mathbf{x}, \mathbf{y}$ of the same length denote $\mathbf{x} \odot \mathbf{y}$ the element-wise product, also known as Hadamard product. For any vector $\mathbf{x}$, let $\|\mathbf{x}\|$ denote the sum of the modulus of its components.

```
Input: Model parameters $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, $\mathbb{Q}$, $\boldsymbol{\rho}$, $\mathbf{r}$
Input: Ultrametric tree and tip states (whenever known)
Input: Maximum allowed condition number $\kappa_o$ (typically $\sim 10^4$–$10^8$)
Output: Log-likelihood $L$
    # Pre-calculate $\mathbf{E}$ and $\mathbb{G}$
Calculate the root age $t_r$
Calculate $\mathbf{E}(t)$ for all $t \in [0, t_r]$, by solving the differential equation (8) (main article) in backward time
    with initial condition $\mathbf{E}(0) = 1 - \boldsymbol{\rho}$. Store the result at discrete times.
Using $\mathbf{E}$ and the model parameters, construct interpolation function for obtaining $\mathbb{A}(t)$ at any $t \in [0, t_r]$
Calculate the maximum condition number ($s_{\max}$) of $\mathbb{A}(t)$ at various $t \in [0, t_r]$
$\Delta_{\text{int}} \leftarrow \ln(\kappa_o)/(2 \cdot s_{\max})$; $N_{\text{int}} \leftarrow \lceil t_r/\Delta \rceil$; $\Delta_{\text{int}} \leftarrow t_r/N_{\text{int}}$   # Define age intervals for calculating $\mathbb{G}$
for $i = 1 : N_{\text{int}}$ do
    $t_o \leftarrow (i-1) \cdot \Delta_{\text{int}}$
    Calculate $\mathbb{G}_i(t)$ for $t \in [t_o, t_o + \Delta_{\text{int}}]$, by solving $d\mathbb{G}_i/dt = \mathbb{A}(t) \cdot \mathbb{G}_i$ with initial condition $\mathbb{G}_i(t_o) = \mathbb{I}$
end for

    # Define function for getting $\Psi(t_{\text{c}}, t_{\text{p}}) \cdot \mathbf{X}_{\text{c}}$ for any $t_{\text{c}} \leq t_{\text{p}}$ and any $\mathbf{X}_{\text{c}}$
function MAP($\mathbf{X}_{\text{c}}, t_{\text{c}}, t_{\text{p}}$)
    $i_{\text{c}} \leftarrow 1 + \lfloor t_{\text{c}}/\Delta_{\text{int}} \rfloor$
    $i_{\text{p}} \leftarrow 1 + \lfloor t_{\text{p}}/\Delta_{\text{int}} \rfloor$
    Solve the linear system $\mathbb{G}_{i_{\text{c}}}(t_{\text{c}}) \cdot \mathbf{X}_o = \mathbf{X}_{\text{c}}$ to obtain $\mathbf{X}_o$, using least-squares
    Initialize $\mathbf{X}_{\text{p}} \leftarrow \mathbf{X}_o$
    for $i = i_{\text{c}} : (i_{\text{p}} - 1)$ do
        $\mathbf{X}_{\text{p}} \leftarrow \mathbb{G}_i(i \cdot \Delta_{\text{int}}) \cdot \mathbf{X}_{\text{p}}$
    end for
    $\mathbf{X}_{\text{p}} \leftarrow \mathbb{G}_{i_{\text{p}}}(t_{\text{p}}) \cdot \mathbf{X}_{\text{p}}$
    return $\mathbf{X}_{\text{p}}$
end function

    # Postorder traversal
For any node $k$, let $\mathbf{X}_k$ be the log-likelihoods up until that node, $t_k$ the age of the node and $n_k$ the number
    of children at the node (if not a tip)
```

```
for each tip $k$ do
    Initialize $\mathbf{X}_k$ according to Eqs. (24) and (25)
end for
Initialize $L \leftarrow 0$
for each internal node $k$ in postorder do
    Initialize $\mathbf{X}_k \leftarrow \boldsymbol{\lambda}(t)^{n_k-1}$
    for each child $c$ of the node do
        $\mathbf{Y} \leftarrow \textsc{map}(\mathbf{X}_c, t_c, t_k)$    # map child $\mathbf{X}_c$ to parent node
        $\mathbf{X}_k \leftarrow \mathbf{X}_k \odot \mathbf{Y}$
    end for
    # Avoid numeric overflow
    $\alpha \leftarrow \|\mathbf{X}_k\|$
    $\mathbf{X}_k \leftarrow \mathbf{X}_k/\alpha$
    $L \leftarrow L + \ln(\alpha)$
end for

# Use root likelihoods, $\mathbf{X}_r$, to calculate overall model likelihood
$\boldsymbol{\pi} \leftarrow \mathbf{X_r}/\|\mathbf{X_r}\|$    # calculate root prior from likelihoods
$\alpha \leftarrow \|\boldsymbol{\pi} \odot \boldsymbol{\lambda} \odot (1 - \mathbf{E}(t_\mathbf{r})) \odot (1 - \mathbf{E}(t_\mathbf{r}))\|$    # 'madfitz' conditioning
$\mathbf{X_r} \leftarrow \mathbf{X_r}/\alpha$
$L \leftarrow L + \ln\|\mathbf{X_r} \odot \boldsymbol{\pi}\|$
```

S.3 Evaluating dSSE likelihood accuracy

To verify that our algorithm is correct and that our numeric implementation yields accurate likelihoods, we performed simulations of BiSSE models with random parameters and calculated the likelihoods of the simulated data using `castor` (Louca and Doebeli, 2017) and `diversitree` (FitzJohn, 2012). Transition rates between states were chosen randomly and uniformly within the interval $[0.1, 0.5]$, birth rates were chosen randomly and uniformly within the interval $[5, 10]$, and death rates were chosen randomly and uniformly within the interval $[0, 5]$. The root state was chosen randomly and uniformly among all possible states. The sampling fraction of tips was set to 10% (i.e. $\rho_i = 0.1$) in order to emulate the common scenario where only a small fraction of species is included in the phylogeny. All tip states were assumed to be known during the evaluation of the likelihood (i.e. $r_i = 1$). Simulated trees comprised either 500 or 10,000 tips. The likelihoods were calculated using the actual model parameters (i.e., as used in the simulations), and are shown in Supplemental Fig. S1.

To further evaluate the accuracy of our implementation, we also performed maximum-likelihood estimates of model parameters for multiple simulated BiSSE and HiSSE trees (each comprising 500 tips). We considered the following R packages: `castor` v1.4.0 (Louca and Doebeli, 2017) for BiSSE and HiSSE, `diversitree` v0.9-10 (FitzJohn, 2012) for BiSSE, and `hisse` v1.8.9 (Beaulieu and O'Meara, 2016) for BiSSE and HiSSE. The package `secsse` (van Els *et al.*, 2018) was omitted due to impractically long computation times. We first simulated 50 BiSSE or HiSSE models with randomly chosen parameters, and used each package to re-estimate model parameters from the simulated data via maximum-likelihood. Model parameters, root state, sampling fractions and reveal fractions were chosen in the same way as described above. No parameters were held fixed during maximum-likelihood, and all parameters (λ_i, μ_i and Q_{ij}) were assumed to be independent. Parameter start values were chosen by first fitting a simple birth-death model (Louca *et al.*, 2018) as well as an Mk-model to the simulated data (Yang *et al.*, 1995) using `castor`; the same start values were used for all packages. The “subplex” optimization algorithm (Rowan, 1990) was used with all packages, with a relative tolerance of 10^{-6}. The maximum number of iterations was set to 10^5, which was sufficiently high to always achieve convergence. The likelihoods at the root were conditioned on the survival of the two child lineages and the speciation event joining them; this is option “`root_conditioning=‘madfitz’`” in `castor`, option “`condition.surv=TRUE`” in `diversitree` and option “`root.type=‘madfitz’`” in `hisse`. The root prior (i.e., the weights for averaging the likelihoods) was set to the likelihoods themselves (after normalizing), as described by FitzJohn *et al.* (2009, Appendix 1); this is option “`root_prior=‘likelihoods’`” in `castor`, option “`root=ROOT.OBS`” in `diversitree` and option “`root.p=NULL`” in `hisse`. For each tree and each package, we calculated the relative estimation errors as $|\hat{x} - x| \, / \, |x|$, where x is the true value of a parameter (i.e., as used in the simulation) and $\hat{x}$ is the estimated value of that parameter. The box plots in Supplemental Figs. S2 and S3 show the distribution of relative errors for selected parameters and for each package. Note that the higher estimation errors by the `hisse` package (compared to `castor`) seen in Figs. S2E–H and S3E–H are probably due to approximations done by the package when solving the differential equations for the likelihood, for purposes of computational efficiency and at the cost of reduced accuracy.

S.4 Evaluating dSSE likelihood run times

To evaluate the computational efficiency of our algorithm, we assessed the time needed for computing dSSE likelihoods and compared them to those of other R packages, using simulated BiSSE and HiSSE trees. Throughout this article, all benchmarks were performed on a MacBook Pro (Retina, 13 inch, early 2015), with 8 GB RAM and 2.9 GHz Intel Core i5 processor, running MacOS 10.13.6 and R 3.6.0. For computation time benchmarks, only a single core was used. The following R packages were considered: `castor` v1.4.0 (Louca and Doebeli, 2017) for BiSSE and HiSSE, `diversitree` v0.9-10 (FitzJohn, 2012) for BiSSE, `hisse` v1.8.9 (Beaulieu and O’Meara, 2016) for BiSSE and HiSSE, `secsse` v1.0.0 (van Els *et al.*, 2018) for HiSSE. For any given tree size, we simulated 10 BiSSE or HiSSE models with randomly chosen parameters, and used each package to calculate the likelihood of the model for the same parameters and given the simulated data. For any given tree size, the run times needed by each package were averaged across simulations. Model parameters, root states, sampling fractions and reveal fractions were chosen in the same way as described above (Supplement S.3). BiSSE likelihoods were calculated using the `castor` function `fit_musse` (options “`sampling_fractions=0.1, root_prior=‘likelihoods’, root_conditioning=‘madfitz’, check_input=TRUE`”), the `diversitree` function `make.bisse` (with option “`sampling.f=0.1`”) and the `hisse` function `makeHiSSELikelihood` (with options “`f=0.1, condition.on.survival=TRUE, root.type=‘madfitz’`”). For purposes of benchmarking, we disabled the automatic calculation of start parameters in `hisse` so as not to inflate `hisse`’s evaluation time with tasks not performed by the other packages. HiSSE likelihoods were calculated using the `castor` function `fit_musse` (with similar options as for BiSSE), the `hisse` function `makeHiSSELikelihood` (with similar options as for BiSSE) and the `secsse` function `secsse_loglik` (with options “`num_concealed_states=2, cond=‘maddison_cond’, root_state_weight=‘maddison_weights’, sampling_fraction=0.1`”).

S.5 Scaling of dSSE maximum-likelihood accuracy on large trees

Our revised algorithm for computing dSSE likelihoods, combined with rapid dSSE simulation methods, allows an assessment of the behavior of dSSE maximum-likelihood estimation for large phylogenies. So far the statistical performance of dSSE models has only been tested for trees with at most a few hundred tips (Davis *et al.*, 2013, Gamisch, 2016, Pyron and Burbrink, 2013, Rabosky and Goldberg, 2015), and hence it is unclear how much better dSSE models perform on larger trees. Here we used `castor` to simulate multiple BiSSE models with up to hundreds of thousands of tips, and to evaluate the accuracy of BiSSE parameter estimates via maximum-likelihood for those trees (Supplemental Fig. S6). Model parameters, root states, sampling fractions and reveal fractions were chosen as described above (Supplement S.3). We found that estimation accuracy, measured in terms of relative error, clearly improves when comparing trees with hundreds of thousands of tips versus only hundreds of tips, although the extent of improvement differs substantially between model parameters. For trees with >10,000 tips, speciation rates (λ_i) and net diversification rates (δ_i) in particular could be estimated to great accuracy, with median relative errors below 3%. This underscores the great statistical potential of modern massive phylogenetic datasets.

S.6 Pseudocode description of the MuSSE simulation algorithm

The following pseudocode outlines the algorithm for simulating MuSSE trees, as implemented in the package `castor`. Let S denote the number of states. Let bold characters (e.g. $\boldsymbol{\lambda}$ or $\mathbf{q}$) denote vectors of size S, and double-lined characters (e.g. $\mathbb{Q}$) denote matrices of size $S \times S$. At any time, every node or tip (henceforth, "clade") ever created has a unique index, called its "clade index". We maintain the following auxiliary data structures:

- EXT[][] keeps track of extant tips at each state, i.e. EXT[s][] lists clade indices of extant tips in state s
- C2PARENT[] maps each clade to its parent, i.e. C2PARENT[k] is the clade index of the clade with index k.
- C2STATE[] maps each clade to latest state, i.e. C2STATE[k] is one of 1,..,S
- C2END[] maps each clade to its time of extinction or branching, i.e. C2END[k] is the time when clade k ceased to be an extant tip. Will be negative if the clade is an extant tip.

```
Input: Model parameters $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, $\mathbb{Q}$
Input: Final number of tips, $N_{\max}$
Input: Initial state of stem lineage, $s_o$
Output: Ultrametric timetree with $N_{\max}$ extant tips
Output: Final state at each extant tip
Let $q_s \leftarrow \sum_{x \neq s} Q_{sx}$ # Sum of transition rates from each state
Initialize EXT, C2PARENT, C2STATE, C2END with a single extant tip at state $s_o$
$\Lambda \leftarrow \lambda_{s_o}$ # Total speciation event rate
$M \leftarrow \mu_{s_o}$ # Total extinction event rate
$T \leftarrow q_{s_o}$ # Total transition rate
$t \leftarrow 0$ # Current time
$N \leftarrow 1$ # Number of extant tips
while $N < N_{\max}$ do
    $R \leftarrow \Lambda + M + T$ # Total event rate
    Choose time until next event, $dt$, according to rate $R$
    $t \leftarrow t + dt$
    Decide if speciation, extinction or anagenetic transition event according to $\Lambda, M, T$
    if speciation then
        Choose random tip $k$ for speciation from within EXT, according to $\boldsymbol{\lambda}$
        $s \leftarrow$C2STATE[$k$]
        # Create $2$ new extant tips as children of $k$
        for $j$ in 1:2 do
            Choose child state $\sigma \leftarrow s$ # Note that cladogenetic transitions can be incorporated here
            Add child to EXT, C2PARENT, C2STATE, C2END
            Update $\Lambda \leftarrow \Lambda + \lambda_\sigma$, $M \leftarrow M + \mu_\sigma$, $T \leftarrow T + q_\sigma$
        end for
        Remove $k$ from EXT[][], and set C2END[$k$]$\leftarrow t$
        Update $\Lambda \leftarrow \Lambda - \lambda_s$, $M \leftarrow M - \mu_s$, $T \leftarrow T - q_s$
        $N \leftarrow N + 1$
    else if extinction then
        Choose random tip $k$ for extinction from within EXT, according to $\boldsymbol{\mu}$
        $s \leftarrow$C2STATE[$k$]
        Remove $k$ from EXT[][], and set C2END[$k$]$\leftarrow t$
        Update $\Lambda \leftarrow \Lambda - \lambda_s$, $M \leftarrow M - \mu_s$, $T \leftarrow T - q_s$
        $N \leftarrow N - 1$
        if $N = 0$ then
            return failure code
        end if
    else if transition then
        # Perform an anagenetic transition
        Choose random tip $k$ for transition from within EXT, according to $\mathbf{q}$
        $s \leftarrow$C2STATE[$k$]
        Choose random new state $\sigma \neq s$ according to probabilities $Q_{s1}/q_s, .., Q_{sS}/q_s$
        Update $\Lambda \leftarrow \Lambda + \lambda_\sigma - \lambda_s$, $M \leftarrow M + \mu_\sigma - \mu_s$, $T \leftarrow T + q_\sigma - q_s$
        C2STATE[$k$]$\leftarrow \sigma$
    end if
end while
Built timetree according to $t$, C2PARENT, C2END and EXT
Assign states to tips according to C2STATE
Remove extinct tips if needed, and remove any resulting monofurcations
```

S.7 Evaluating BiSSE simulation run times

The following R packages were included in the benchmarks: `castor` v1.4.0 (Louca and Doebeli, 2017), `diversitree` v0.9-10 (FitzJohn, 2012), `hisse` v1.8.9 (Beaulieu and O'Meara, 2016), `geiger` v2.0.6.1 (Pennell *et al.*, 2014), `TESS` v2.1.0 (Höhna *et al.*, 2015), `phytools` v0.6-60 (Revell, 2012), `TreeSim` v2.3 (Stadler, 2011) and `ape` v5.2 (Paradis *et al.*, 2004). For any given tree size, each package was used to simulate multiple BiSSE models with randomly chosen parameters; the same model parameters were used across packages. Transition rates between states were chosen randomly and uniformly within the interval $[0.1, 0.5]$, birth rates were chosen randomly and uniformly within the interval $[5, 10]$, and death rates were chosen randomly and uniformly within the interval $[0, 5]$. All packages except `TreeSim` and `TESS` simulate trees in forward time according to the model until a stopping criterion is met ("simple sampling approach" or SSA; Stadler, 2011), whereas `TreeSim` and `TESS` sample the branching times of a reconstructed timetree conditional upon the time span of the tree and/or the number of extant species. For all SSA simulators except `ape`, the targeted number of tips was provided as the sole stopping criterion for the simulation. Because `ape` did not support such a stopping criterion, a maximum simulation time interval of $\log(N/2)/(\lambda - \mu)$ was used as sole stopping criterion for ape, where N is the targeted number of tips; whenever the resulting tree had fewer than $0.9 \cdot N$ tips, the test was repeated. For TreeSim and TESS, trees were conditioned upon a fixed number of tips. BiSSE trees were simulated using the castor function simulate_musse (with options "max_tips=N, coalescent=TRUE, all_Mk_transitions=TRUE, no_full_extinction=TRUE"), the diversitree function tree.bisse (with options "max.taxa=N, include.extinct=FALSE, max.t=Inf") and the hisse function SimulateHisse (with options "max.taxa=N, nstart=1"). Simple birth-death cladogenic models (i.e., without trait evolution) were simulated using the geiger function sim.bdtree (with options "stop='taxa', n=N, extinct=TRUE"), the TESS function tess.sim.taxa (with options "n=1, nTaxa=N, max=log(N)/(lambda-mu), samplingProbability=1"), the phytools function pbtree (with options "n=N, nsim=1, type='continuous', extant.only=TRUE"), the TreeSim function sim.bd.taxa (with options "n=N, numbsim=1, frac=1, complete=TRUE, stochsampling=TRUE"), and the ape function rbdtree (with option "Tmax=log(N/2)/(lambda-mu)"). Whenever a simulation failed due to a tree going extinct, the test was repeated. For any given tree size, 10 models were simulated and the run times of each package were averaged.

S.8 Evaluation of BiSSE simulation accuracy

As mentioned in the main article, our MuSSE simulator is essentially a Gillespie algorithm that simulates the exact stochastic process in forward time until a halting criterion is met — in our case, until a specific number of extant species is reached (Stadler, 2011 calls this the "simple sampling approach", or SSA). To confirm the accuracy of our implementation and consistency with existing software, we performed multiple simulations of models using castor, and compared the generated trees to those generated by another SSA simulator implemented in the package diversitree. We considered two different models: The first model was a BiSSE model, with parameters $\lambda_1 = 1$, $\lambda_2 = 2$, $\mu_1 = 0.5$, $\mu_2 = 1$, $Q_{12} = 1$ and $Q_{21} = 2$ (all rates are in Myr^{-1}), and with the initial state chosen randomly according to the stationary distribution of $\mathbb{Q}$. The second model was essentially a birth-death model, which however was simulated as a BiSSE model with equal speciation rates, equal extinction rates and zero transition rates, i.e. $\lambda_1 = \lambda_2 = 1$, $\mu_1 = \mu_2 = 0.5$ and $Q_{ij} = 0$. For the simulations we used the castor function simulate_musse and the diversitree function tree.bisse. Each simulation was halted when the generated tree reached 500 extant tips. Each model was simulated 1000 times. Trees generated by castor were compared to trees generated by diversitree in terms of their lineages-through-time curves (LTT) as well as the distribution of pairwise phylogenetic distances between nodes ("patristic distance"). LTTs were calculated using the castor function count_lineages_through_time and averaged over all simulations of a given model and simulator. For each of the two models, averaged LTTs were then visually compared between castor and diversitree (Figs. S5A,C). Pairwise node distances were calculated using the castor function get_all_pairwise_distances, and their distribution density was calculated using the R function density, using a Gaussian kernel and a smoothing bandwidth equal to the inverse mean speciation rate. Distance distribution densities were averaged over all simulations of a given model and simulator; averaged distributions were then visually compared between castor and diversitree (Figs. S5B,D). As can be seen in Figs. S5A–D, for the models examined, trees generated by castor are almost statistically identical to those generated by diversitree, increasing confidence in the correctness of our code.

S.9 Assessing the effects of reveal biases

If the identification of tip states is biased towards certain states, i.e., the reveal fractions r_i differ between states, then maximum-likelihood parameter estimates can become substantially distorted if these biases are not properly accounted for (as is the case in all existing tools). To demonstrate this effect, we proceeded as follows. We simulated 50 BiSSE trees, whereby model parameters were randomly chosen as in the other benchmarks described above (Supplement S.3). The sampling fraction was set to 10% as before (i.e., $\rho_i = 0.1$). Only a randomly chosen subset of tips was considered to have a known tip state, with reveal fractions (r_i) being 0.2 and 0.8 for states 1 and 2, respectively. We then used these simulated data as input to maximum-likelihood estimation with the `castor` function `fit_musse`, while either providing the true reveal fractions or while wrongly assuming that all reveal fractions were equal. Optimization options were as described above. Default parameter start values were chosen by first fitting a simple birth-death model (Louca *et al.*, 2018) as well as an Mk-model to the simulated data (Yang *et al.*, 1995), however multiple alternative start values were also considered to avoid non-global local optima in the likelihood function (option "`Ntrials=10`"). For each tree and for each method (i.e., while providing or omitting information on the reveal fractions) we calculated the relative estimation errors of parameters as described above (Supplement S.3). The box plots in Supplemental Fig. S7 show the distribution of relative errors for both methods and for a selection of model parameters, for BiSSE trees with 500 or 10,000 tips.

We found that the mean and median relative errors of estimated parameters were substantially higher when not correcting for reveal biases than when reveal biases were corrected for. The greatest reduction in accuracy was observed for the transition rates Q_{ij}. Indeed, when reveal biases were not corrected for, the transition rate $Q_{1,2}$ was substantially overestimated and Q_{21} was substantially underestimated. This effect persisted even for large trees with thousands of tips (Supplemental Fig. S7). Hence, not correcting for the fact that state 1 is much harder to reveal than state 2, leads to biases in the estimated transition rates between states 1 and 2, in addition to higher estimation errors in all model parameters.

Note that if there are no biases in identifying tip states (i.e., the r_i are the same across states), then the mere fact that a tip state is known or unknown contains no useful information for parameter estimation. Mathematically, this means that the factors r_i and $(1 - r_i)$ in the initial conditions can be omitted, since their inclusion only rescales the model's likelihood function by a constant factor that is independent of model parameters. In that case, one retrieves the dSSE models formulated in previous studies (FitzJohn, 2012, FitzJohn *et al.*, 2009, Maddison *et al.*, 2007).

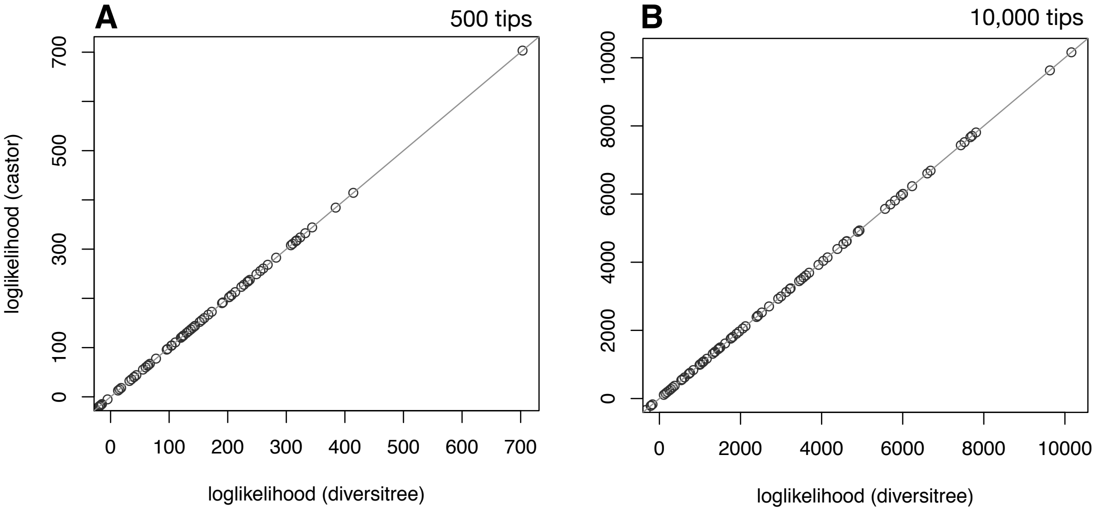

Figure S1: **Comparison of calculated BiSSE likelihoods.** Comparison of BiSSE log-likelihoods calculated using `diversitree` (horizontal axis; FitzJohn, 2012) and `castor` (vertical axis; Louca and Doebeli, 2017), for multiple simulated BiSSE models with randomly chosen parameters (one point per simulation). Trees either comprised 500 tips (A) or 10,000 tips (B). Detailed functions and options used are explained in the Methods.

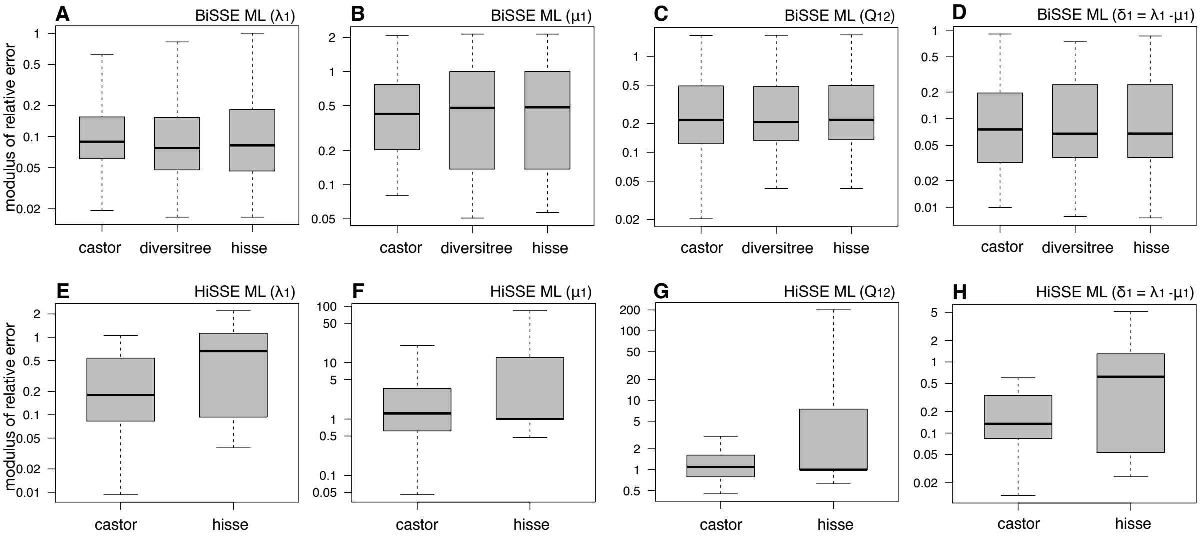

Figure S2: **Comparison of maximum-likelihood accuracies across tools (modulus).** Comparison of maximum-likelihood estimation accuracy for BiSSE models (first row) and HiSSE models (second row), using `castor` and other software packages (one sub-figure per model parameter, one box per package). Box plots show the distribution of the modulus of relative errors of estimated parameters compared to their true values, over multiple random trees; boxes span the 2nd and 3rd quartiles, whiskers span 90% percentiles, outliers are not shown. Note the logarithmic axes in all plots. Trees and tip states used as input were simulated under the BiSSE or HiSSE model (top and bottom row, respectively), with randomly chosen parameters. Trees comprised 500 tips. Compared packages include `diversitree` (FitzJohn, 2012) and `hisse` (Beaulieu and O'Meara, 2016). The package `secsse` (van Els *et al.*, 2018) was omitted due to impractically long computation times. Detailed functions and options used are explained in the Methods. See Fig. S3 for plots of the relative errors.

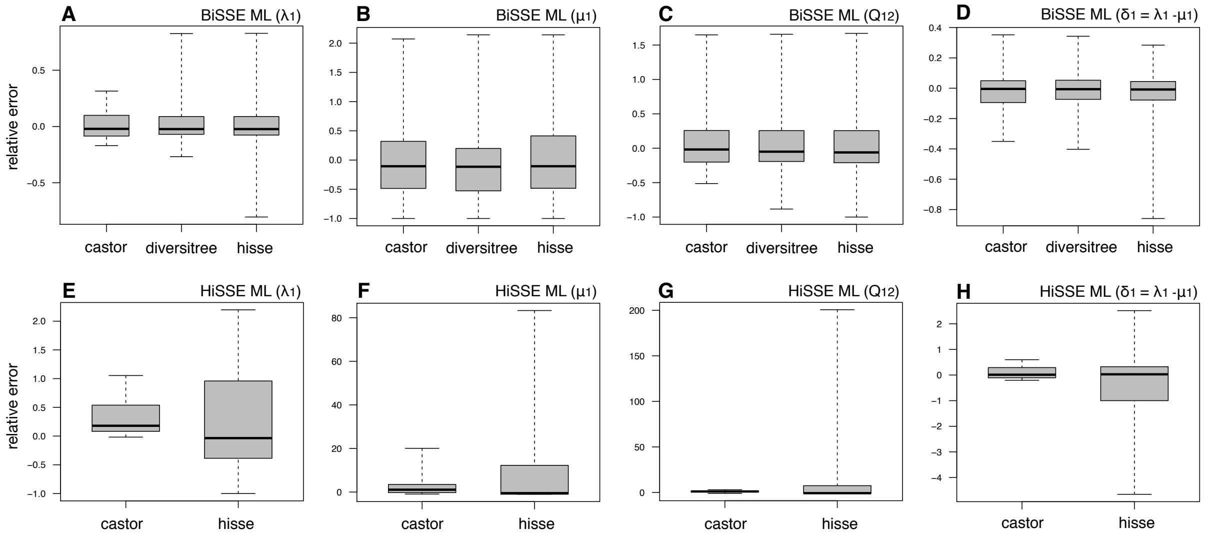

Figure S3: **Comparison of maximum-likelihood accuracies across tools.** Comparison of maximum-likelihood estimation accuracy for BiSSE models (first row) and HiSSE models (second row), using `castor` and other software packages (one sub-figure per model parameter, one box per package). Box plots show the distribution of the relative errors of estimated parameters compared to their true values, over multiple random trees; boxes span the 2nd and 3rd quartiles, whiskers span 90% percentiles, outliers are not shown. Based on the same simulated trees as Fig. S2. Compared packages include `diversitree` (FitzJohn, 2012) and `hisse` (Beaulieu and O'Meara, 2016). Detailed functions and options used are explained in the Methods. See Fig. S2 for the modulus of relative errors.

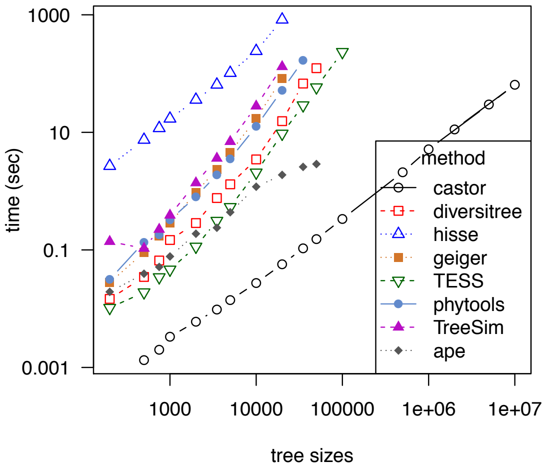

Figure S4: **Comparison of run times between simulators.** Comparison of computation times needed for the simulation of a single BiSSE model (using `castor`, `diversitree` or `hisse`) or a single bird-death cladogenic model (using `geiger`, `TESS`, `phytools`, `TreeSim` or `ape`) (time T over tree size S, one curve per package). Note the logarithmic axes. Except for `castor` and `ape`, the power-law exponents for all other packages were ≥ 1.5. The package `ape` (Paradis *et al.*, 2004) failed to generate any trees with more than 50,000 tips in our trials, even for the simple case $\lambda = 1$ and $\mu = 0$. Detailed functions and options used are explained in the Methods.

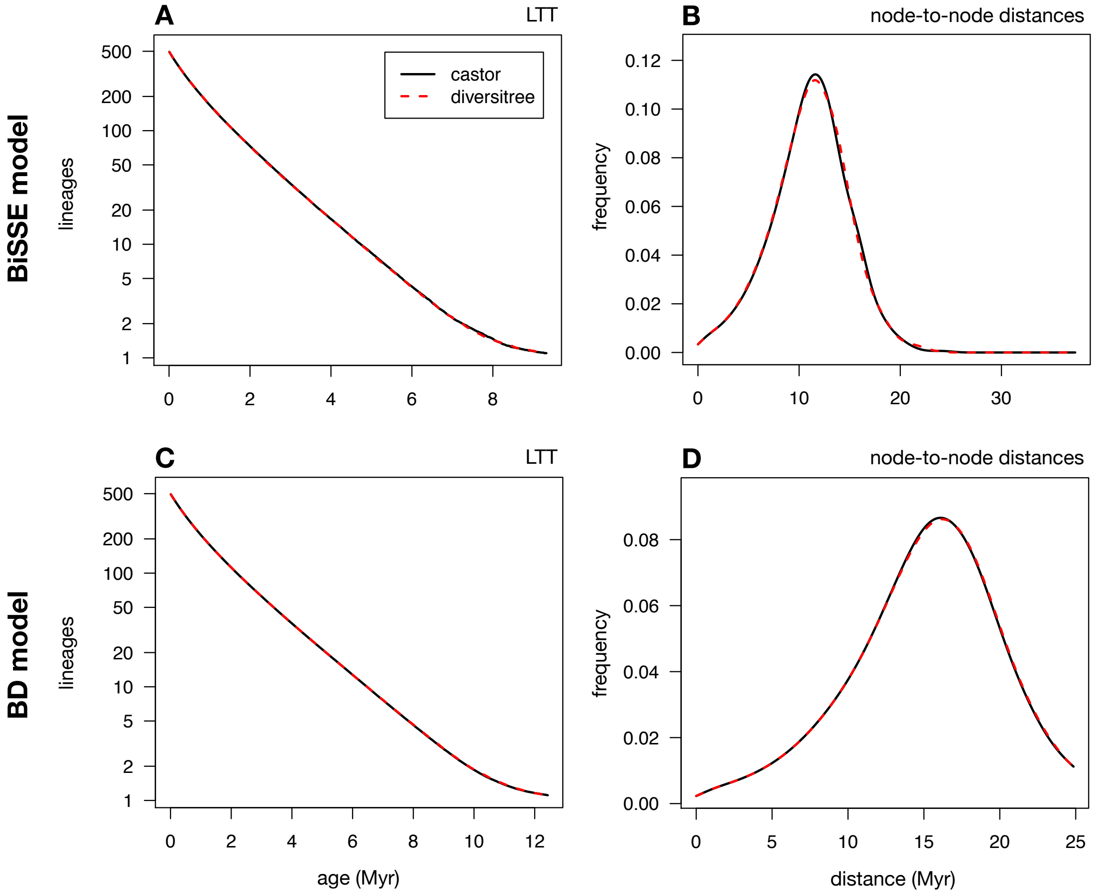

Figure S5: **Accuracy of simulated BiSSE models.** Comparison of trees simulated under the BiSSE model (top row) or the birth-death model (bottom row), in terms of their lineages-through-time curves (LTT, left column) or the distribution of pairwise node distance (right column), between `castor` (continuous curves) and `diversitree` (dashed curves). Each curve represents the average LTT or the average distribution of pairwise node distances, obtained across 1000 independent simulations of the same model. All trees had 500 tips. Simulations of the birth-death model were performed using the BiSSE simulators, after setting all transition rates to zero ($Q_{ij} = 0$) and setting $\lambda_1 = \lambda_2$ and $\mu_1 = \mu_2$. Detailed functions and options used are explained in the Methods.

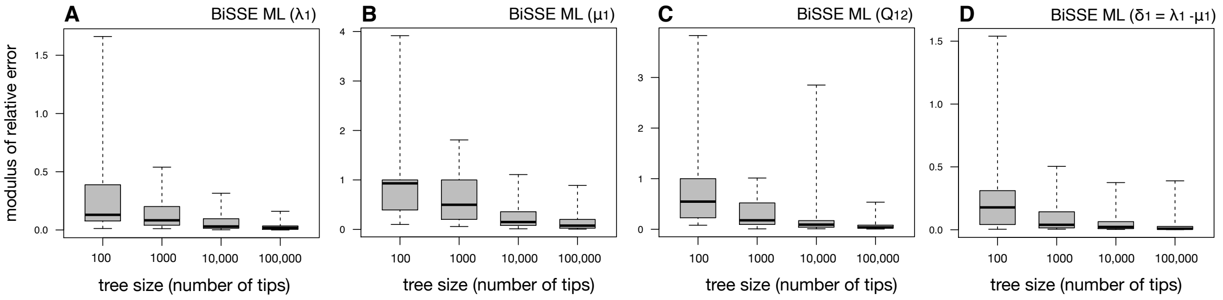

Figure S6: **Comparison of maximum-likelihood accuracies across tree sizes.** Comparison of maximum-likelihood estimation accuracy for BiSSE models using `castor`, estimated using random trees of various sizes (one sub-figure per model parameter, one box per tree size). Box plots show the distribution of the relative errors of estimated parameters compared to their true values, over multiple random trees; boxes span the 2nd and 3rd quartiles, whiskers span 90% percentiles, outliers are not shown. Trees and tip states used as input were simulated under the BiSSE model with randomly chosen parameters.

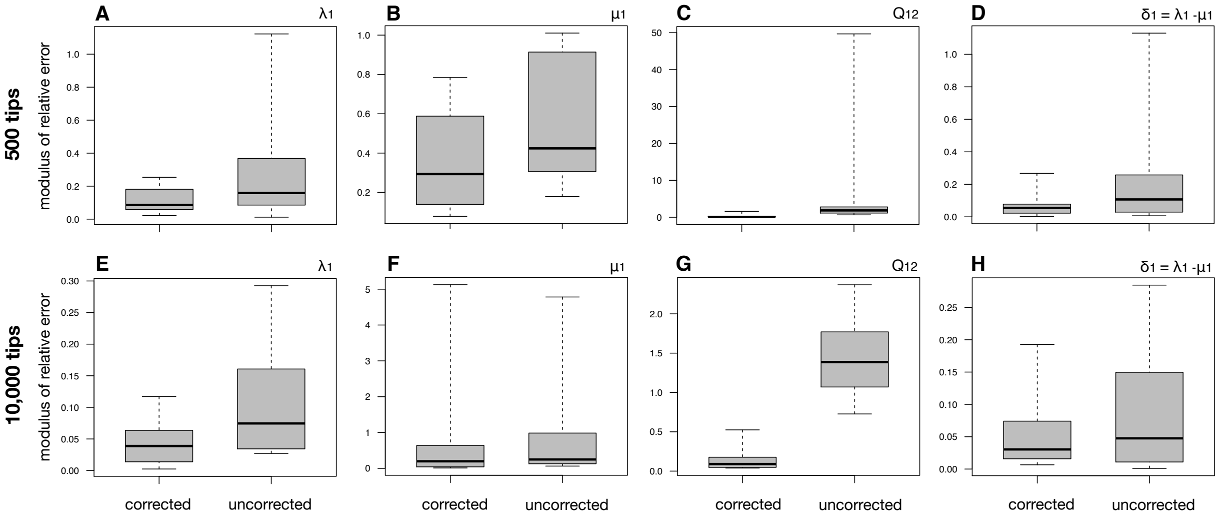

Figure S7: **Correcting for state identification biases.** Comparison of maximum-likelihood estimation accuracy for BiSSE models using `castor`, while correcting or without correcting for biases in the identification of different tip states (one sub-figure per model parameter, left box with correction, right box without correction). Box plots show the distribution of the relative errors of estimated parameters compared to their true values, over multiple random trees; boxes span the 2nd and 3rd quartiles, whiskers span 90% percentiles, outliers are not shown. Trees and tip states used as input were simulated under the BiSSE model with randomly chosen parameters. All trees comprised 500 or 10,000 tips. Probabilities of state identification were 0.2 and 0.8 for states 1 and 2, respectively.

References

Beaulieu, J.M. and O'Meara, B.C. (2016) Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. *Systematic biology* **65**: 583–601.

Butcher, J.C. (1987) The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods. Wiley-Interscience publication. John Wiley & Sons.

Cline, A., Moler, C., Stewart, G., and Wilkinson, J. (1979) An estimate for the condition number of a matrix. *SIAM Journal on Numerical Analysis* **16**: 368–375.

Davis, M.P., Midford, P.E., and Maddison, W. (2013) Exploring power and parameter estimation of the bisse method for analyzing species diversification. *BMC Evolutionary Biology* **13**: 38.

FitzJohn, R.G. (2012) Diversitree: comparative phylogenetic analyses of diversification in R. *Methods in Ecology and Evolution* **3**: 1084–1092.

FitzJohn, R.G., Maddison, W.P., and Otto, S.P. (2009) Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. *Systematic Biology* **58**: 595–611.

Friedli, A. (1978) Verallgemeinerte Runge-Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme. In R. Bulirsch, R. Grigorieff, and J. Schröder (eds.) Numerical treatment of differential equations, volume 631, pp. 35–50. Berlin: Springer.

Gamisch, A. (2016) Notes on the statistical power of the Binary State Speciation and Extinction (BiSSE) model. *Evolutionary Bioinformatics* **12**.

Höhna, S., May, M.R., and Moore, B.R. (2015) TESS: an R package for efficiently simulating phylogenetic trees and performing Bayesian inference of lineage diversification rates. *Bioinformatics* **32**: 789–791.

Lawson, J.D. (1967) Generalized Runge-Kutta processes for stable systems with large Lipschitz constants. *SIAM Journal on Numerical Analysis* **4**: 372–380.

Louca, S. and Doebeli, M. (2017) Efficient comparative phylogenetics on large trees. *Bioinformatics* **34**: 1053–1055.

Louca, S., Shih, P.M., Pennell, M.W., Fischer, W.W., Parfrey, L.W., and Doebeli, M. (2018) Bacterial diversification through geological time. *Nature Ecology & Evolution* **2**: 1458–1467.

Maddison, W.P., Midford, P.E., Otto, S.P., and Oakley, T. (2007) Estimating a binary character's effect on speciation and extinction. *Systematic Biology* **56**: 701–710.

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

Pennell, M.W., Eastman, J.M., Slater, G.J., Brown, J.W., Uyeda, J.C., FitzJohn, R.G., *et al.* (2014) geiger v2. 0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees. *Bioinformatics* **30**: 2216–2218.

Pyron, R.A. and Burbrink, F.T. (2013) Phylogenetic estimates of speciation and extinction rates for testing ecological and evolutionary hypotheses. *Trends in Ecology & Evolution* **28**: 729–736.

Rabosky, D.L. and Goldberg, E.E. (2015) Model inadequacy and mistaken inferences of trait-dependent speciation. *Systematic Biology* **64**: 340–355.

Revell, L.J. (2012) phytools: an r package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution* **3**: 217–223.

Rowan, T. (1990) Functional stability analysis of numerical algorithms. Ph.D. thesis, Department of Computer Sciences, University of Texas at Austin.

So, W. and Thompson, R.C. (2000) Singular values of matrix exponentials. *Linear and Multilinear Algebra* **47**: 249–258.

Stadler, T. (2011) Simulating trees with a fixed number of extant species. *Systematic Biology* **60**: 676–684.

Turing, A.M. (1948) Rounding-off errors in matrix processes. *The Quarterly Journal of Mechanics and Applied Mathematics* **1**: 287–308.

van Els, P., Etienne, R.S., and Herrera-Alsina, L. (2018) Detecting the dependence of diversification on multiple traits from phylogenetic trees and trait data. *Systematic Biology* .

Watkins, D. (2010) Fundamentals of matrix computations. Wiley international editions, 3rd edition. New York, USA: Wiley.

Yang, Z., Kumar, S., and Nei, M. (1995) A new method of inference of ancestral nucleotide and amino acid sequences. *Genetics* **141**: 1641–1650.