Exact and efficient phylodynamic simulation from arbitrarily large populations

Michael Celentano[1], William S. DeWitt[2], Sebastian Prillo[2], and Yun S. Song[1,2]

[1]Department of Statistics, University of California, Berkeley
[2]Computer Science Division, University of California, Berkeley

Abstract

Many biological studies involve inferring the evolutionary history of a sample of individuals from a large population and interpreting the reconstructed tree. Such an ascertained tree typically represents only a small part of a comprehensive population tree and is distorted by survivorship and sampling biases. Inferring evolutionary parameters from ascertained trees requires modeling both the underlying population dynamics and the ascertainment process. A crucial component of this phylodynamic modeling involves tree simulation, which is used to benchmark probabilistic inference methods. To simulate an ascertained tree, one must first simulate the full population tree and then prune unobserved lineages. Consequently, the computational cost is determined not by the size of the final simulated tree, but by the size of the population tree in which it is embedded. In most biological scenarios, simulations of the entire population are prohibitively expensive due to computational demands placed on lineages without sampled descendants. Here, we address this challenge by proving that, for any partially ascertained process from a general multi-type birth-death-mutation-sampling model, there exists an equivalent process with *complete sampling* and *no death*, a property which we leverage to develop a highly efficient algorithm for simulating trees. Our algorithm scales linearly with the size of the final simulated tree and is independent of the population size, enabling simulations from extremely large populations beyond the reach of current methods but essential for various biological applications. We anticipate that this unprecedented speedup will significantly advance the development of novel inference methods that require extensive training data.

1 Introduction

Phylogenetic trees describe the ancestral relationships within a sample of individuals from a large population and are central objects in studies of evolution. (We make no distinction between phylogenetic and genealogical trees in this work.) These individuals can represent extant or extinct species in macroevolution [16, 18], viral sequences collected over time in epidemiology [29], or individual cells in studies of cancer evolution or affinity maturation [5, 7, 8, 26, 35], for example. The branch lengths and branching rates in these phylogenies have been used to infer shifts in diversification rates across lineages and over time in the tree of life [4, 6, 14], to estimate effective reproduction number or predict future strains in viral evolution [23, 30], and to estimate rates of metastasis, phylodynamics, plasticity, or paths of tumor evolution in cancer [34], among many other applications.

Due to death and incomplete sampling in these applications, the inferred phylogeny typically represents a partial history of the full population. This partial observation process can lead to bias in the estimation of fundamental population parameters. For example, the branching rate in the observed phylogeny is typically substantially smaller than that in the full population phylogeny

[17, 20]. Moreover, certain population-level quantities may not be identified by the distribution of the observed phylogeny without further assumptions [12–14].

Simulations play a central role in tree-based inference. Their uses include benchmarking existing methods [26, 32], training novel methods based on simulated data [31], and, in some approaches, may form a sub-routine of the inferential method itself [33]. In order to faithfully represent the relationship between population-level parameters and the observed data, these simulations must include the partial observation process. Current approaches simulate the partial observation process directly. First, the phylogeny of the full population is simulated; next, some lineages are sampled; and finally, lineages that are not ancestral to a sampled lineage are pruned from the full phylogeny [20].

In many applications, full-population simulations are prohibitively expensive because they must expend substantial computational resources on lineages that, because they have no sampled descendants, do not appear in the observed phylogeny. For example, in a typical year, there are tens of millions of flu infections in the United States alone [27]. Nevertheless, large phylogenetic analyses of flu are often limited to tens of thousands of sampled viral sequences [1], and it is not uncommon to analyze phylogenies with only a few hundred samples per year [21]. In cancer, a cubic centimeter of tumor mass contains roughly one billion cells, yet single-cell resolution studies of cancer with CRISPR-Cas9-based lineage tracing assays sequence fewer than 100,000 cells [10, 25, 28, 34], representing roughly 0.01% of the population. Recent works at best simulate a population of 40,000 cancer cells subsequently sub-sampled down to 1% [10, 24], which is highly unrealistic.

This paper proposes a novel algorithm for exact (up to numerical and time-discretization error) simulation from a general class of multi-type birth-death processes with birth, death, mutation, and incomplete sampling. This class of models agrees with those considered by MacPherson et al. [15], which unified a collection of models used in a wide range of biological contexts, including the applications described above [4,6,11,14,16,18,19,22,30]. This general class of models is referred to as "birth-death-mutation-sampling" (BDMS) models. The computational cost of our algorithm for simulating BDMS models scales not with the size of the full population, as do existing approaches, but rather with the sample size. It thus avoids expending computational resources on unobserved lineages and is essentially optimal in terms of runtime. Based on the typical numbers for flu and cancer evolution studies described above, our approach can reduce computation by a factor of 1,000 to 10,000, and make it feasible to simulate with realistic population-size and sub-sampling parameters. Thus, our novel algorithm will facilitate the assessment of existing methods under more biologically realistic settings and enable the development of simulation-based training and estimation approaches.

From a technical point of view, our algorithm is based on the following insight. The distribution of the observed phylogeny in any BDMS model is equivalent to that generated by an alternative BDMS model with no death and complete sampling. In this alternative BDMS model, which we call the *forward-equivalent model*, the full phylogeny corresponds to the observed phylogeny. Therefore, simulating from the forward-equivalent model allows us to avoid expending computational resources on unobserved lineages while generating observed trees from the same distribution.

Our insight is closely related to recent work on statistical identifiability in sub-sampled phylogenetic birth-death models [12–14]. This work observed that in the single-type setting, different time-dependent population-level parameterizations can give rise to the same observed data distribution. Our algorithm is based on the lack of identifiability in the more general multi-type setting. Whereas lack of identifiability prohibits exact inference (without further assumptions), it also facilitates simulation. Because multiple BMDS models give rise to the same observed data distribution, we can base our simulations on those that can be simulated most efficiently.

Process	Description
Initialization	At time $t_{\max}$, the population begins with one lineage with type $a \sim \text{Cat}(\pi)$.
Birth	For all $a, b \in \{1, \ldots, d\}$ and $\tau \in [0, t_{\max}]$, lineages of type a split into types a and b at rate $\lambda_{a,b}(\tau)$ at time τ.
Death	For all $a \in \{1, \ldots, d\}$ and $\tau \in [0, t_{\max}]$, lineages of type a die at rate $\mu_a(\tau)$ at time τ.
Anagenetic mutation	For all $a, b \in \{1, \ldots, d\}$ and $\tau \in [0, t_{\max}]$, lineages of type a mutate to type b at rate $\gamma_{a,b}(\tau)$ at time τ.
Sampling	For all $a \in \{1, \ldots, d\}$ and $\tau \in [0, t_{\max}]$, lineages of type a are sampled at rate $\psi_a(\tau)$ at time τ. If sampled, they die with probability $r_a(\tau)$ at time τ.
CSEs	For all $l \in \{0, \ldots, L\}$ and $a \in \{1, \ldots, d\}$, lineages of type a at time t_l are sampled with probability $\rho_{a,l}$. If sampled, they die with probability $q_{a,l}$.

Table 1: Summary of processes in the birth-death-mutation-sampling (BDMS) model.

2 The multi-type birth-death-mutation-sampling (BDMS) model

The BDMS model we study is a multi-type birth-death process with mutations and incomplete sampling as presented by MacPherson et al. [15]. It consists of a collection of lineages that give birth, die, mutate, and are sampled over time. The model begins with a single lineage at time $\tau = t_{\max} > 0$ measured as the (positive) distance in time before the present day, $\tau = 0$. The single lineage at time $\tau = t_{\max}$ is initialized with type a drawn from a categorical distribution with parameter $\pi = (\pi_1, \ldots, \pi_d)$ over a finite collection of d types indexed by $1, \ldots, d$. Each lineage progresses independently of all other lineages, with events arriving according to Poisson point processes with potentially time-varying intensity. These processes are as follows. At time τ, a lineage of type a gives birth to a lineage of type b at rate $\lambda_{a,b}(\tau)$, dies at rate $\mu_a(\tau)$, and is sampled at rate $\psi_a(\tau)$. Upon sampling, a lineage dies with probability $r_a(\tau)$. When a lineage gives birth, there are two daughter lineages, one with type a and one with type b. The parameter $\lambda_{a,a}(\tau)$ describes the rate at which births without mutation occur. For $b \neq a$, the parameter $\lambda_{a,b}(\tau)$ describes the rate of cladogenetic mutation. Anagenetic mutations—ones which occur at non-birth events—from type a to type b occur at rate $\gamma_{a,b}(\tau)$, which is non-zero for only $b \neq a$.

In addition to the above processes, whose events arrive according to time-dependent Poisson point processes, we allow for concerted sampling events (CSEs) [15]. These are instances in time at which a fraction of the population gets simultaneously sampled. These occur at fixed times $t_0 = 0 < t_1 < \cdots < t_L \leq t_{\max}$. The lth CSE consists of sampling all extant lineages of type a at time t_l independently with probability $\rho_{a,l}$. During a CSE, each sampled lineage dies with probability $q_{a,l}$. The model is defined by the set of parameters $\Theta = (\pi, \lambda, \mu, \gamma, \psi, r, \rho, q, t, t_{\max})$, whose interpretations are summarized in Table 1. The rate parameters $\lambda, \mu, \gamma, \psi$ and the death probabilities r may depend on time τ.

The reconstructed phylogeny refers to the subset of the full phylogeny containing only those lineages ancestral to a sampled lineage [20]. In Figure 1, we provide a diagram demonstrating the partial observation process on a full phylogeny and the corresponding reconstructed phylogeny. In applications, it is typically the reconstructed phylogeny (or an estimate of it based on sequence data) that is observed. Thus, the distribution of the reconstructed phylogeny and its corresponding
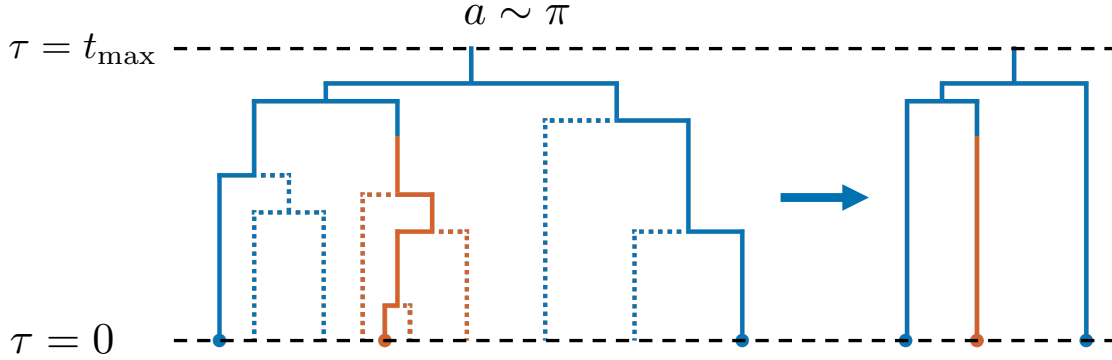
Figure 1: Diagram of partial observation process. The full phylogeny is displayed on the left. Color represents type, with one anagenetic mutation event shown. All sampling events occur at the present day, represented by solid circles at the leaves. Lineages that are not ancestral to a sampled lineage are shown in dashed lines. The observed phylogeny is shown on the right.

likelihood in terms of model parameters, rather than the distribution and likelihood for the full phylogeny, must be used as the basis for inference. A large body of work derives the likelihood for reconstructed phylogenies as a solution to a system of ordinary differential equations which can be solved numerically [11,16,19]. Our simulation method will also rely on solving a system of ordinary differential equations.

3 Simulation algorithms

We describe two methods for simulating reconstructed phylogenies. The first, which we call the *full simulation*, simulates the full phylogeny and then removes (or "prunes") lineages that are not ancestral to a sampling event; this is the current state-of-the-art. The second is our novel method, which we call the *forward-equivalent simulation*; this approach simulates only those lineages that will have sampled descendants, thereby greatly reducing computational resources for most practical applications.

3.1 The full simulation

There are a variety of methods for simulating the full phylogeny which differ according to whether they grow the phylogeny in a depth-first or breadth-first manner and in how they determine the time and type of birth, death, mutation, and sampling events. We used an efficient breadth-first implementation [3] whose time complexity is linear in the number of simulated events, where a single event corresponds to a birth, death, mutation, or sampling event in the simulated population. Pseudocode for the algorithm we used is stated in Algorithm 1. The contribution of this paper is the forward equivalent simulation described in the next section, which can be viewed as a wrapper around the full simulation presented here. Indeed, the forward equivalent simulation can be wrapped around any implementation of a time-varying BDMS model. The implementation we used was selected to optimize time complexity in the number of simulated events. While it is not the contribution of this paper, it is presented for completeness.

In words, Algorithm 1 simulates forward in time starting at $\tau = t_{max}$ and ending at $\tau = 0$ or until all lineages in the simulation die or a user-specified capacity limit n_{max} is exceeded. The nodes in the tree correspond to birth, death, mutation, or sampling events. As we grow the tree, each node contains information about its type, event type, and time.

Algorithm 1 FullSim($\theta, n_{\max}$)

 1: $\tau \leftarrow t_{\max}$;
 2: $S_a \leftarrow \textsc{RandomizedSet}()$ for $a = 1, \ldots, d$;
 3: Draw RootNode with type $a \sim \mathrm{Categorical}(\pi)$;
 4: $\textsc{Insert}(\mathrm{RootNode}, S_a)$;
 5: **while** $\tau \geq 0$ and $1 \leq \sum_{a=1}^{d} |S_a| \leq n_{\max}$ **do**
 6: $\quad$ EventType, $\tau_{\mathrm{event}} \leftarrow \textsc{GetNextEvent}(\{|S_a|\}, \tau)$
 7: $\quad$ **if** $\tau_{\mathrm{event}} < 0$ **then**
 8: $\quad\quad$ **for all** $a = 1, \ldots, d$ and Node $\in S_a$ **do**
 9: $\quad\quad\quad$ Child $\leftarrow \textsc{NewNode}(\mathrm{type} = a, \mathrm{event} = \mathrm{survival}, \mathrm{time} = 0)$;
10: $\quad\quad\quad$ $\textsc{SetChildren}(\mathrm{Node}, [\mathrm{Child}])$;
11: $\quad\quad\quad$ $\textsc{Remove}(\mathrm{Node}, S_a)$; $\textsc{Insert}(\mathrm{Child}, S_a)$;
12: $\quad\quad$ Exit while loop
13: $\quad$ **if** EventType $= \mathrm{Birth}(a \rightarrow a, b)$ **then**
14: $\quad\quad$ $\mathrm{Child}_1 \leftarrow \textsc{NewNode}(\mathrm{type} = a, \mathrm{event} = \mathrm{birth}, \mathrm{time} = \tau_{\mathrm{event}})$;
15: $\quad\quad$ $\mathrm{Child}_2 \leftarrow \textsc{NewNode}(\mathrm{type} = b, \mathrm{event} = \mathrm{birth}, \mathrm{time} = \tau_{\mathrm{event}})$;
16: $\quad\quad$ Parent $\leftarrow \textsc{GetRandom}(S_a)$;
17: $\quad\quad$ $\textsc{SetChildren}(\mathrm{Parent}, [\mathrm{Child}_1, \mathrm{Child}_2])$;
18: $\quad\quad$ $\textsc{Remove}(\mathrm{Parent}, S_a)$;
19: $\quad\quad$ $\textsc{Insert}(\mathrm{Child}_1, S_a)$; $\textsc{Insert}(\mathrm{Child}_2, S_b)$;
20: $\quad$ **if** EventType $= \mathrm{Death}(a)$ **then**
21: $\quad\quad$ Child $\leftarrow \textsc{NewNode}(\mathrm{type} = a, \mathrm{event} = \mathrm{death}, \mathrm{time} = \tau_{\mathrm{event}})$;
22: $\quad\quad$ Parent $\leftarrow \textsc{GetRandom}(S_a)$;
23: $\quad\quad$ $\textsc{SetChildren}(\mathrm{Parent}, [\mathrm{Child}])$;
24: $\quad\quad$ $\textsc{Remove}(\mathrm{Parent}, S_a)$;
25: $\quad$ **if** EventType $= \mathrm{Mutation}(a \rightarrow b)$ **then**
26: $\quad\quad$ Child $\leftarrow \textsc{NewNode}(\mathrm{type} = b, \mathrm{event} = \mathrm{mutation}, \mathrm{time} = \tau_{\mathrm{event}})$;
27: $\quad\quad$ Parent $\leftarrow \textsc{GetRandom}(S_a)$;
28: $\quad\quad$ $\textsc{SetChildren}(\mathrm{Parent}, [\mathrm{Child}])$;
29: $\quad\quad$ $\textsc{Remove}(\mathrm{Parent}, S_a)$; $\textsc{Insert}(\mathrm{Child}, S_b)$;
30: $\quad$ **if** EventType $= \mathrm{Sampling}(a)$ **then**
31: $\quad\quad$ Child $\leftarrow \textsc{NewNode}(\mathrm{type} = a, \mathrm{event} = \mathrm{sampling}, \mathrm{time} = \tau_{\mathrm{event}})$;
32: $\quad\quad$ Parent $\leftarrow \textsc{GetRandom}(S_a)$;
33: $\quad\quad$ $\textsc{SetChildren}(\mathrm{Parent}, [\mathrm{Child}])$;
34: $\quad\quad$ $\textsc{Remove}(\mathrm{Parent}, S_a)$; $\textsc{Insert}(\mathrm{Child}, S_a)$;
35: $\quad$ $\tau \leftarrow \tau_{\mathrm{event}}$;
36: **if** Capacity limit not exceeded **then**
37: $\quad$ **for all** $a = 1, \ldots, d$ and Node $\in S_a$ **do**
38: $\quad\quad$ **if** $\mathrm{Bernoulli}(\rho_{a,0}) = 1$ **then**
39: $\quad\quad\quad$ $\textsc{SetEventType}(\mathrm{Node}, \mathrm{event} = \mathrm{sampling})$;
40: $\quad$ Prune tree
41: **return** Tree $\mathcal{T}$, or "Empty" if no nodes remain after pruning, or "Capacity exceeded" if capacity exceeded.

We generate the tree in a breadth-first manner. At each moment, the algorithm maintains biological time τ and the tree generated up until time τ. Each iteration generates the next event in biological time anywhere in the tree, and advances the tree to the time of that next event. Thus, consecutive iterations generate events that are consecutive in biological time but which need not occur to the same lineage or be close phylogenetically.

In order to achieve this, at each iteration the algorithm maintains, for each type $a = 1, \ldots, d$, a set S_a containing the most recent event nodes for all extant lineages of type a. The next event is then generated as follows:

1. First, the time and type of the next event are determined. The type of the next event specifies i) whether it is a birth, death, mutation, or sampling event, ii) the type of the parent node, and iii) for birth and mutation events, the type of the child or children nodes. It does not specify on which lineage the event occurs. In Algorithm 1, this step is represented by the function GETNEXTEVENT. The time and type of the next event can be determined using the model parameters Θ, the sizes of the extant populations for each type $\{|S_a|\}_{a=1}^d$, and the current biological time τ. In Appendix A, we describe an implementation of GETNEXTEVENT which runs in constant time in the size of the extant population.

2. The parent node of the next event node is determined. If the event type returned by GETNEXTEVENT specifies that the parent is a type a, we pick a parent node uniformly at random from S_a. As we describe below, we use a data structure for S_a that permits uniform random sampling in $O(1)$ time.

3. The phylogeny is advanced to the time of the next event. This involves creating a new event node whose parent is as chosen in the previous step and updating the sets $\{S_a\}$ as needed. See Algorithm 1 for details (which depend on the event type). This step requires possibly inserting or removing nodes from the sets S_a. As we describe below, we use a data structure for S_a that permits insertion and removal in constant time.

These iterations progress until either the time of the next event is after present time $\tau = 0$ or a user-specified population size capacity limit is exceeded. If the time of the next event is after the present time, then this event does not occur. Instead, all extant lineages are advanced to the present, and sampling occurs for each lineage according to the sampling probabilities $\rho_{a,0}$.

Note that we allow for a user-specified capacity limit even though this is not formally a part of the BDMS model. Practically, it is useful to have an implementation that allows termination based on a user-specified capacity limit. Depending on the model parameters, the population size can grow exponentially with time. Thus, if $t_{\max}$ is set too large, the algorithm will fail to terminate in a reasonable amount of time. In practice, one chooses model parameters such that simulation to the present is, except in rare cases, feasible with available computational resources. The capacity limit is chosen large enough so that it rarely is reached but serves as protection against exponential blow-up. In this case, the capacity limit will have a negligible impact on the distribution of simulated trees.

After the full phylogeny has been simulated as described above, the tree is pruned to remove all lineages and nodes that are not ancestral to a sampling event. Moreover, birth events in which only one child lineage survives either (1) are removed if the surviving offspring has the same phenotype as the parent, or (2) appears as anagenetic mutations if the surviving offspring has a different phenotype than the parent. The pruning operation can be carried out in time linear in the number of events in the full phylogeny.

For simplicity and conciseness, we have omitted from this description and from the pseudo-code in Algorithm 1 concerted sampling events (CSEs) except for the one at the present day. These are

straightforward to add and do not affect our discussion on computational complexity. Indeed, when the time of the next event is determined to be after the next concerted sampling event, the event does not occur. Instead, all extant lineages are advanced to the time of the concerted sampling event, at which point the appropriate sampling and death events are carried out.

We implement Algorithm 1 so that each iteration of the while loop (which corresponds to a single event) runs in constant time in the population size. This requires that GETNEXTEVENT has constant time complexity $O(1)$ in the population size, and the sets S_a are implemented using a data structure which supports insertion, removal, and uniform random sampling in constant time in the population size. We describe these implementation details in Appendix A. With this implementation, Algorithm 1 has time complexity linear in the number of events in the full phylogeny. In Section 4 and Supplementary Figure S1, we empirically validate that the computation time scales linearly with the number of events in the full phylogeny.

As we describe in Appendix A, the time complexity of GETNEXTEVENT is quadratic in the size of the type space d. Thus, the overall runtime of Algorithm 1 when taking into account the size of the type space is $O(\#\text{nodes}(\mathcal{T})d^2)$ where $\#\text{nodes}(\mathcal{T})$ is the number of events in the full phylogeny $\mathcal{T}$. The quadratic factor d^2 may be replaced by $d + \log(\#\text{nodes}(\mathcal{T}))$ by using an alternative implementation based on a priority queue. For applications with large type spaces, this implementation may be preferred. We describe this approach in Appendix A.4. In this work, however, we focus on the approach provided in Algorithm 1, since its runtime is strictly linear in the number of events in the full phylogeny.

3.2 The forward-equivalent simulation

The full simulation expends computational time on all lineages, including those that have no sampled descendants and thus do not appear in the reconstructed phylogeny. We propose a method that does not expend computational resources on such unobserved lineages. Avoiding such computation is non-trivial: the appearance of a lineage in the reconstructed phylogeny depends on the occurrence (or not) of a future sampling event. Thus, in Algorithm 1, the appearance of an event in the reconstructed phylogeny cannot be determined at the time the event is simulated.

In this section, we show that the distribution of the reconstructed phylogeny in the full model conditional on it being non-empty is equal to that generated by an alternative BDMS model without death and with complete sampling at the present. We call this alternative BDMS model the *forward-equivalent model*.

Precisely, consider any set of model parameters $\Theta := (\pi, \lambda, \mu, \gamma, \psi, r, \rho, q, t, t_{\max})$ for the full simulation. Our main result is that, for any such Θ, there is an alternative set of parameters $\Theta^{\mathsf{FE}} := (\pi^{\mathsf{FE}}, \lambda^{\mathsf{FE}}, \mu^{\mathsf{FE}}, \gamma^{\mathsf{FE}}, \psi^{\mathsf{FE}}, r^{\mathsf{FE}}, \rho^{\mathsf{FE}}, t, t_{\max})$ satisfying $\mu_a^{\mathsf{FE}}(\tau) = 0$ for all $a \in \{1, \ldots, d\}$, $\tau \in [0, t_{\max}]$ (no death) and $\rho_{a,0}^{\mathsf{FE}} = 1$ for all $a \in \{1, \ldots, d\}$ (complete sampling at the present) such that the distribution of the reconstructed phylogeny $\mathcal{T}$ under Θ conditional on being non-empty, $\mathcal{T} \neq T_\emptyset$, is equal to the distribution of the reconstructed phylogeny under Θ^{FE} (with $T_\emptyset$ denoting the empty tree realization, which corresponds to a full tree in which all lineages are either extinct or unsampled, so that pruning leaves no ascertained subtree). The parameters Θ^{FE} are given precisely in Section 3.3 via a map F on parameter space such that $\Theta^{\mathsf{FE}} = F(\Theta)$. Let random variable $\mathcal{T}_\Theta$ denote the reconstructed phylogeny in the full model parameterized by Θ.

Theorem 1. *The distribution of the reconstructed phylogeny in the full model conditional on its non-emptiness is equivalent to the unconditional distribution of the reconstructed/full phylogeny in the forward-equivalent model:*

$$\mathsf{Law}(\mathcal{T}_\Theta \mid \mathcal{T}_\Theta \neq T_\emptyset) = \mathsf{Law}(\mathcal{T}_{F(\Theta)}). \tag{1}$$

7

We prove Theorem 1 in Appendix B.

Theorem 1 suggests simulating the reconstructed phylogeny in the full model by running the full simulation on the parameters $\Theta^{\mathsf{FE}} = F(\Theta)$. This approach, which we call the forward-equivalent simulation, is given in Algorithm 2. It generates reconstructed phylogenies from the distribution $\mathsf{Law}(\mathcal{T}_\Theta \mid \mathcal{T}_\Theta \neq T_\emptyset)$. If one wants to generate reconstructed phylogenies from $\mathsf{Law}(\mathcal{T}_\Theta)$ allowing for emptiness, one can first generate a Bernoulli random variable B with success probability $\mathbb{P}(\mathcal{T}_\Theta \neq T_\emptyset)$ (which is computed in Section 3.3) and, if $B = 0$, return $T_\emptyset$, whereas if $B = 1$, return $\mathrm{FESim}(\Theta, n_{\max})$. In practice, one often will want to generate non-empty phylogenies, so this extra step will not be necessary.

Algorithm 2 $\mathrm{FESim}(\Theta, n_{\max})$

1: $\Theta^{\mathsf{FE}} \leftarrow F(\Theta)$ ▷ *Precomputated and reused for sampling multiple trees*
2: $\mathcal{T} \leftarrow \mathrm{FullSim}(\Theta^{\mathsf{FE}}, n_{\max})$
3: **return** $\mathcal{T}$

In the forward-equivalent model, all lineages are observed and the full and the reconstructed phylogeny are the same. When used to simulate reconstructed phylogenies in models with death or incomplete sampling, the forward-equivalent simulation effectively avoids expending computational resources on lineages that are not observed. This leads to substantial computational savings when either only a small fraction of the full phylogeny is observed or the full phylogeny only rarely survives until the present. An analysis of computational complexity is carried out in Section 3.4.

Implementing Algorithm 2 requires evaluating the forward-equivalent model parameters Θ^{FE} via the map F on line 1. These can be pre-computed once, and then used to generate an arbitrary number of trees via repeated calls to the full simulation, as on line 2. These precomputations are described in Appendix A. The cost of precomputation is amortized over the number of simulated trees, so contributes negligible computational cost when this number is large. In our experiments, pre-computations were very fast.

In practice, one may choose to set $n_{\max} < \infty$ in Algorithm 2. Because in the forward-equivalent model, reconstructed and full phylogenies are the same, simulating from the forward equivalent model with finite $n_{\max}$ is equivalent to simulating from the full model conditional on its non-emptiness and on the maximal number of simultaneously extant lineages at any point in time in the reconstructed phylogeny being bounded above by $n_{\max}$.

3.3 The forward-equivalent model parameters

In this section, we define the map F giving parameters Θ^{FE} explicitly in terms of the parameters Θ of the full model. A key quantity is the non-observation probability $E_a(\tau)$ for $a = 1, \ldots, d$ and $\tau \in [0, t_{\max}]$ in the full model. It is the probability that, given a lineage of type a in the population at time τ, none of its descendants will be sampled. Note that $\mathbb{P}(\mathcal{T}_\Theta \neq T_\emptyset)$ is given by $1 - \sum_{a=1}^d \pi_a E_a(t_{\max})$. At CSEs, the non-observation probability will be different immediately prior to and after the realization of the CSE's sampling and death events. We denote the non-observation probability prior to the realization of these events by $E_a(t_l^+)$ and after the realization of these events by $E_a(t_l^-)$, where t_l is the time of the CSE. As shown in [15, Equation 15]), the non-observation probabilities are given by the solution to the following ODE system for all $a = 1, \ldots, d$

8

and $\tau \in [0, t_{\max}]$:

$$\textbf{Initialization:} \qquad E_a(0^-) = 1,$$

$$\textbf{Between CSEs:} \quad \frac{\mathrm{d}E_a(\tau)}{\mathrm{d}\tau} = \sum_{b=1}^{d} \lambda_{a,b}(\tau)[E_a(\tau)E_b(\tau) - E_a(\tau)]$$
$$+ \mu_a(\tau)[1 - E_a(\tau)]$$
$$+ \psi_a(\tau)[0 - E_a(\tau)] \tag{2}$$
$$+ \sum_{b=1}^{d} \gamma_{a,b}(\tau)[E_b(\tau) - E_a(\tau)],$$

$$\textbf{At CSEs:} \qquad E_a(t_l^-) = \lim_{\tau \uparrow t_l} E_a(\tau), \quad E_a(t_l^+) = (1 - \rho_{a,l})E_a(t_l^-).$$

For completeness, we derive (2) in Appendix B. The non-observation probability has previously played an important role in likelihood-based and Bayesian methods of inference based on the reconstructed phylogeny [4, 6, 15, 16, 18, 23]. For all $a, b \in \{1, \ldots, d\}$ and $\tau \in [0, t_{\max}]$, the forward-equivalent mapping $\Theta^{\mathsf{FE}} = F(\Theta)$ is defined by

$$\pi_a^{\mathsf{FE}} = \frac{\pi_a[1 - E_a(t_{\max})]}{\sum_b \pi_b[1 - E_b(t_{\max})]}, \qquad \lambda_{a,b}^{\mathsf{FE}}(\tau) = [1 - E_b(\tau)]\lambda_{a,b}(\tau),$$

$$\mu_a^{\mathsf{FE}}(\tau) = 0, \qquad \psi_a^{\mathsf{FE}}(\tau) = \frac{\psi_a(\tau)}{1 - E_a(\tau)},$$

$$r_a^{\mathsf{FE}}(\tau) = r_a(\tau) + [1 - r_a(\tau)]E_a(\tau), \quad \gamma_{a,b}^{\mathsf{FE}}(\tau) = \frac{1 - E_b(\tau)}{1 - E_a(\tau)}\Big[\gamma_{a,b}(\tau) + E_a(\tau)\lambda_{a,b}(\tau)\delta_{a \neq b}\Big], \tag{3}$$

$$\rho_{a,l}^{\mathsf{FE}} = \frac{\rho_{a,l}}{1 - E_a(t_l^+)}, \qquad q_{a,l}^{\mathsf{FE}} = q_{a,l} + (1 - q_{a,l})E_a(t_l^-).$$

By (2), $E_a(0^+) = 1 - \rho_{a,0}$, whence by (3), $\rho_{a,0}^{\mathsf{FE}} = 1$. Thus, in the forward-equivalent simulation, all lineages at $\tau = 0$ are sampled. We note that the ODE (2) is Lipshitz continuous in its solution on each inter-CSE interval, so for $\lambda, \mu, \gamma, \psi, r$ suitably smooth in the time domain, it follows by standard results that there exists a unique solution, i.e., the map F is well-defined. It suffices, for example, to assume that the parameters are continuous or piecewise constant in time.

It is enlightening to interpret some of the equations in (3). The equation for $\lambda_{a,b}^{\mathsf{FE}}(\tau)$ indicates that the birth rate in the reconstructed phylogeny is smaller than that in the population. This effect is stronger for individuals that are more unlikely to have sampled descendants. The equation for $\gamma_{a,b}^{\mathsf{FE}}(\tau)$ indicates that anagenetic mutation rates can be upward or downward biased in the reconstructed phylogeny. In particular, mutations from types that are unlikely to have sampled descendants and to types that are likely to have sampled descendants are higher in the reconstructed phylogeny than in the population; mathematically, $E_a(\tau) > E_b(\tau)$ implies $\frac{1 - E_b(\tau)}{1 - E_a(\tau)} > 1$, so $\gamma_{a,b}^{\mathsf{FE}}(\tau) > \gamma_{a,b}(\tau)$. One factor that increases the probability of having a sampled descendant is having "higher fitness" in terms of a larger birth rate or lower death rate, leading to a higher chance of survival. Thus, these equations indicate that the relative birth rates of "fit" vs. "unfit" individuals in the reconstructed phylogeny are higher than in the population, and mutation rates from "unfit" to "fit" individuals are enhanced. These biases exemplify the importance of incorporating the sampling process into phylogenetic simulations.
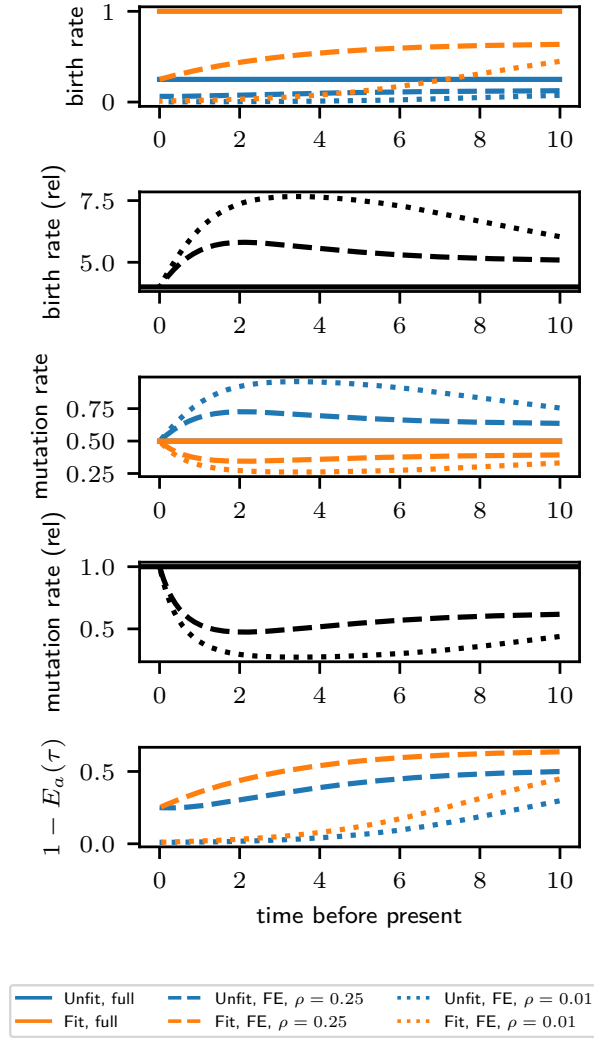
Figure 2: Full and forward equivalent model parameters as a function of time in a two-type model with a Fit and Unfit phenotype, with $\lambda_{\mathsf{Fit}} = 1.0$, $\lambda_{\mathsf{Unfit}} = 0.25$, $\mu_{\mathsf{Fit}} = \mu_{\mathsf{Unfit}} = 0.25$, and anagenetic mutation rates $\gamma_{\mathsf{Fit},\mathsf{Unfit}} = \gamma_{\mathsf{Unfit},\mathsf{Fit}} = 0.5$. The relative birth rate is defined as $\lambda_{\mathsf{Fit}}(\tau)/\lambda_{\mathsf{Unfit}}(\tau)$ and $\lambda_{\mathsf{Fit}}^{\mathsf{FE}}(\tau)/\lambda_{\mathsf{Unfit}}^{\mathsf{FE}}(\tau)$. The relative mutation rate is is defined as $\gamma_{\mathsf{Fit},\mathsf{Unfit}}(\tau)/\gamma_{\mathsf{Unfit},\mathsf{Fit}}(\tau)$ and $\gamma_{\mathsf{Fit},\mathsf{Unfit}}^{\mathsf{FE}}(\tau)/\gamma_{\mathsf{Unfit},\mathsf{Fit}}^{\mathsf{FE}}(\tau)$.

As a concrete example, we plot in Figure 2 the forward equivalent model parameters in a two-type model with a Fit and Unfit phenotype, with $\lambda_{\mathsf{Fit}} = 1.0$, $\lambda_{\mathsf{Unfit}} = 0.25$, $\mu_{\mathsf{Fit}} = \mu_{\mathsf{Unfit}} = 0.25$, and anagenetic mutation rates $\gamma_{\mathsf{Fit},\mathsf{Unfit}} = \gamma_{\mathsf{Unfit},\mathsf{Fit}} = 0.5$. We consider both a low sampling probability $\rho = 0.01$ and moderate sampling probability $\rho = 0.25$, occuring uniformly across phenotypes at the present. We observe that the forward equivalent birth rate is smaller than that in the population for both the Fit and Unfit phenotypes. Moreover, we see that in the forward equivalent model, birth rates are higher in the distant past than in the present. This heightened diversification rate in the observed phylogeny in the distant past has been termed the "push of the past" [2] and results from survivorship bias. We also plot the relative birth rate in the forward equivalent model, defined as $\lambda_{\mathsf{Fit}}^{\mathsf{FE}}(\tau)/\lambda_{\mathsf{Unfit}}^{\mathsf{FE}}(\tau)$. We see that it is larger than 4, indicating that the birth rate of the Unfit phenotype is reduced more relative to its population value than that of the Fit phenotype. This effect is stronger when ρ is smaller. These qualitative phenomena can be easily seen in the

formulas in (3). Indeed, the amount by which the birth rate is depressed is controlled by $1 - E_b(\tau)$, the probability a lineage will end up in the sample, so that the qualitative time dependence of the birth rates, the inflation of the relative birth rate, and the dependence of the relative birth rate on ρ can be read off the plot of $1 - E_a(\tau)$ in Figure 2. We also observe that the mutation rate from the Unfit to Fit phenotype is higher and from the Fit to Unfit phenotype is lower in the forward equivalent model than in the population, with a stronger effect for smaller ρ. This is the result of selective pressures, which are more pronounced when ρ is smaller. Again, (3) indicates the relevant quantity is the ratio $(1 - E_b(\tau))/(1 - E_a(\tau))$, the qualitative behavior of which can also be observed in the plot of $1 - E_a(\tau)$ in Figure 2.

3.4 Efficiency improvements of forward-equivalent simulations

As we justified in Section 3.1, our simulation algorithm has time complexity which is linear in the number of simulated events. The forward-equivalent simulation can be faster than the full simulation because it simulates fewer events per non-empty reconstructed phylogeny.

There are two reasons why the forward-equivalent simulation simulates fewer events per non-empty reconstructed phylogeny. First, the reconstructed phylogeny contains fewer events (i.e., nodes) than the full phylogeny. For example, in a pure birth process that samples present-day lineages with probability ρ, the reconstructed tree will have approximately ρ times as many nodes as the full phylogeny. We thus expect the forward-equivalent simulation to run $1/\rho$ times faster per tree than the full simulation.

Second, in the full simulation, the reconstructed phylogeny may be empty. Thus, the full simulation may require multiple retries to produce a single non-empty reconstructed phylogeny. The events simulated in trees with empty reconstructed phylogenies contribute to the computational cost of simulating a single non-empty phylogeny. In contrast, the forward-equivalent simulation never produces the empty tree $T_\emptyset$. As an artificial extreme case, consider a trivial process without birth or death and which samples present-day lineage with probability ρ. In this case, the full tree is a 'stick'. The reconstructed tree is a 'stick' with probability ρ and is $T_\emptyset$ otherwise. The forward equivalent simulation simulates a stick in its first try, requiring $\Theta(1)$ time per non-empty reconstructed tree. The full simulation requires in expectation $1/\rho$ retries to obtain a non-empty reconstructed tree. In this case, the speedup provided by our method is again $1/\rho$.

In what follows, we study the key quantity R (respectively, R^{FE}), which is the expected number of simulated events required to produce a single non-empty reconstructed tree in the full (respectively, forward-equivalent) simulation. For a single-type birth-death process, we prove the following result:

Proposition 1. *Consider running a birth-death process for 1 unit of time with birth rate λ, death rate μ, and extant sampling probability ρ. Then:*

1. *If $\lambda = \mu$, then the theoretical speedup of the forward-equivalent model is:*

$$\frac{R}{R^{\mathsf{FE}}} = \frac{1}{\rho}(1 + \mu).$$

2. *If $\lambda \neq \mu$, then the theoretical speedup of the forward-equivalent model is:*

$$\frac{R}{R^{\mathsf{FE}}} = \frac{1}{\rho}\left[1 + \frac{\mu(e^{\lambda-\mu} - 1)}{(\lambda - \mu)e^{\lambda-\mu}}\right].$$

We prove Proposition 1 in Appendix C. In the simpler case that the birth and death rates are equal, we see that the theoretical speedup is essentially linear in the (inverse) sampling probability ρ and in the death rate μ. Later, in our empirical benchmarks, we will see that this theoretical estimate is quite accurate.

For the general BDMS model, we have the following result.

Proposition 2. *Let $\mathcal{T}$ denote the full tree in the full model, and $\mathcal{R}(\mathcal{T})$ its corresponding (possibly empty) reconstructed tree. Then*

$$\frac{R}{R^{\mathsf{FE}}} = \frac{\mathbb{E}[\#nodes(\mathcal{T})]}{\mathbb{E}[\#nodes(\mathcal{R}(\mathcal{T}))]}.$$

We emphasize here that nodes in $\mathcal{T}$ include birth events, mutation events, and sampling events and thus need not all be locations of bifurcation in the tree. We prove Proposition 2 in Appendix C. The significance of Proposition 2 is that it unifies the two sources of speed-up described above, avoiding explicit accounting of the number of retries required to observe a non-empty reconstructed phylogeny.

We remark that the expected number of nodes appearing in Proposition 2 can be computed numerically with time discretization and dynamic programming. Thus, Proposition 2 allows for numerical estimates of speed-up factors. Empirically, the forward-equivalent simulation may experience a constant factor computational overhead relative to the full simulation due to its sampling from time-inhomogeneous Poisson processes: see Section 4.2. In our experiments, this overhead was smaller than a factor of 2, so the forward-equivalent simulation was faster than the full simulation for moderate values of ρ and μ.

4 Empirical validation

We carry out several simulations to validate our method. In Section 4.1, we consider a case where both the full and forward-equivalent simulations are feasible, and perform several tests of whether the reconstructed phylogenies produced by the two simulations have the same distribution. In Section 4.2, we perform several simulations to measure the relative computational efficiency of the two approaches and to validate the theory from Section 3.4. In Section 4.3, we use the forward-equivalent simulation to generate phylogenies from a setting in which the full simulation is computationally infeasible on our machines.

4.1 Distribution tests

We performed a simulation study to test that the full and forward-equivalent simulations produce reconstructed phylogenies with the same distribution. We consider a full model consisting of two types, Fit and Unfit. The population is simulated for $t_{\max} = 20$ units of biological time. Both types have a death rate of $\mu_{\mathsf{Fit}} = \mu_{\mathsf{Unfit}} = 0.25$, the Unfit type has birth rate of $\lambda_{\mathsf{Unfit}} = 0.25$, and the Fit type has a birth rate of $\lambda_{\mathsf{Fit}} = 1$. Recall that, without mutation and given infinite time, a population will go extinct with probability 1 if and only if the birth rate does not exceed the death rate: $\lambda \leq \mu$. Thus, we have chosen parameters so that the Unfit population is just barely in the regime where it would, without mutation, be assured eventual extinction. On the other hand, the Fit population would, without mutation, be unlikely to go extinct and would grow exponentially fast. We consider anagenetic mutation rates $\gamma_{\mathsf{Fit},\mathsf{Unfit}} = 0.8$ and $\gamma_{\mathsf{Unfit},\mathsf{Fit}} = 0.1$. Thus, deleterious mutations from Fit to Unfit are much more common than beneficial mutations from Unfit to Fit. Based on our simulations (see below), these mutation rates keep the distribution of

types in the population relatively balanced. We consider a model with sampling at the present $\rho_{\mathsf{Fit},0} = \rho_{\mathsf{Unfit},0} = 0.5$ and no sampling at earlier times or at death events.

We generated 1,000 non-empty phylogenies from both the full and forward-equivalent simulations for the above model. Because we are conditioning on non-emptiness, generating a single non-empty phylogeny in the full simulation may (and sometimes did) require multiple retries.

We tested distributional equivalence of several scalar summary tree statistics. Because phylogenetic trees are high-dimensional objects, distributional agreement on these test statistics does not necessarily imply distributional equivalence of the phylogenetic trees. Moreover, the tree statistics we selected are not intended to be exhaustive, and we do not intend to imply that they provide an optimal summary of tree geometry for any particular application. Rather, we intend to capture a wide enough range of tree features to provide convincing evidence that the distributional equivalence (1) holds. The tree statistics we considered were:

- **Event count.** The total number of nodes in the tree, which includes births, mutations, and sampled leaf nodes.

- **Leaf count.** The total number of leaves in the tree. In this simulation, this is the size of the sample from the present-day population.

- **Branch length.** The cumulative branch length stratified by type (for Fit and Unfit lineages) and in total.

- **Subtree count, stratified by size.** The number of nodes in the tree for which the descendant tree has size nodes (including the root node). For example, the subtree count at $\mathsf{size} = 1$ is the number of leaves. The subtree count at $\mathsf{size} = 3$ contains the sum of the number of cherries and the number of nodes after which one mutation event but no birth events occur before present-day sampling.

- **Lineage count, stratified by type and time.** For a type in $\{\mathsf{Fit}, \mathsf{Unfit}\}$ and time t, the number of lineages in the tree of that type at a cross-section at time t.

We also considered the distribution of "block statistics". In multi-type models, each phylogeny is partitioned into a collection of maximally connected subtrees of uniform type. We call each element of this partition a *block*. A block statistic is a scalar summary of a block; for example, total branch length. We simulated 1,000 non-empty trees and then plot histograms of various block statistics over the list of generated blocks. We emphasize that not all trees contain the same number of blocks, so that trees with more blocks contribute more counts to these histograms than trees with fewer blocks. The histograms represents a distribution which assigns an equal weight to each block, and thus assigns a weight to each tree proportional to its number of blocks. We consider the following block statistics:

- **Events per block, stratified by type.** The number of nodes in the block.

- **Branch length per block, stratified by type.** The cumulative branch length of the block.

In Figure 3, we show histograms for two of these summary statistics—lineage count and branch length per block—stratified by type. For the block-based statistics (in this case, branch length per block), the y-axis counts the number of blocks cumulatively across phylogenies, so that each phylogeny may contribute multiple blocks to the count, and larger phylogenies will typically contribute more. In all cases, the histograms generated by the full and forward-equivalent simulations are visually similar. For each summary statistic, we performed a two-sided Mann-Whitney U-test and
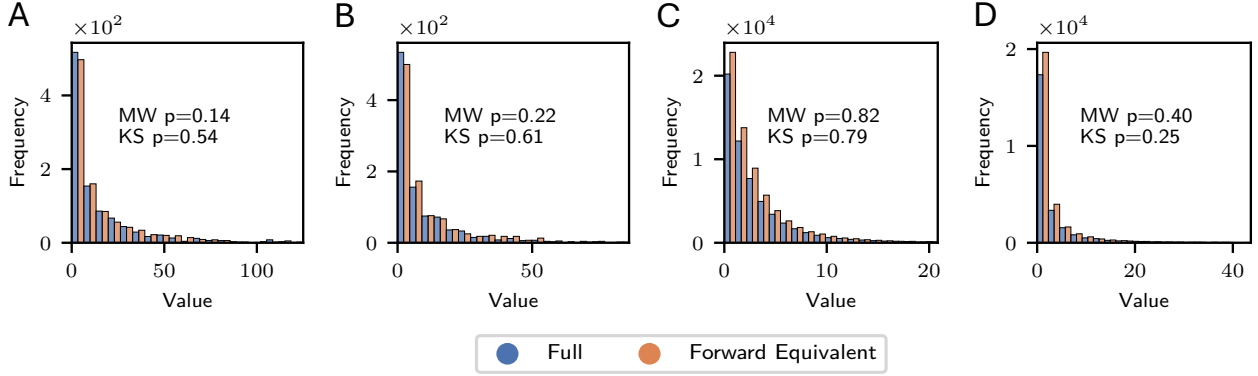
Figure 3: Distributional comparison of summary statistics from two-type model described in the text. The full and forward equivalent simulation are both run until 1,000 non-empty trees are generated. Each plot displays a different summary statistics and the Mann-Whitney (MW) and Kolmogorov-Smirnov (KS) p-value testing the equivalence of the full and forward equivalent simulations. (A) Number of extant lineages at biological time $t = 16$ with phenotype Unfit. (B) Number of extant lineages at biological time $t = 16$ with phenotype Fit. (C) Branch length of blocks with phenotype Unfit. (D) Branch length of blocks with phenotype Fit. The simulation parameters are described in the main text, Section 4.1.

the Kolmogorov-Smirnov test to compare the distributions, with the p-values in each case reported on the corresponding histogram plot. We caution that the theoretical null distribution used to compute Kolmogov-Smirnov test statistic is only asymptotically valid for continuous distributions. Thus, it must be interpreted with caution for the count-based test statistics when the counts are small. Similar plots to Figure 3, reporting results for all the summary statistics listed above, are provided in Supplementary Figure S2.

While not a proof of distributional equivalence, these tests indicate that our implementation successfully produces the distributional equivalence predicted by Theorem 1, providing evidence for its correctness.

4.2 Computational efficiency

We empirically assessed the relative computational cost of the full and the forward-equivalent simulations and their dependence on population size and the size of the sub-sampled phylogeny under three models: single-type without death, single-type with death, and two-type.

The "single-type without death" model consisted of a unique type with constant birth rate given by $\lambda = 1.0$ and no death run for $t_{\max} = 5$ units of biological time. We subsampled the present population with probability ρ, which we set to $\rho = 0.1 \times k$ for $k = 1, \ldots, 10$. At each value of k, we simulated 50 non-empty trees. As in Section 4.1, the full simulation may require multiple retries to generate a single non-empty tree.

All results are displayed in Figure 4. In the left column we report, for each value of ρ, the average total simulation time (including retries) required to produce 50 non-empty trees, normalized by 50. In the right column, we report the ratio of the computation time required by the full simulation to the forward-equivalent simulation on a log-log scale.

The first row shows results for the single-type model without death. For the full simulation, the mean time per tree is approximately constant in ρ. For the forward-equivalent simulation, the mean time per tree appears to be linear in ρ. This is consistent with our simulation scaling linearly in the total number of simulated events, which in the full simulation is independent of ρ and in the forward-equivalent simulation is proportional to ρ on average.
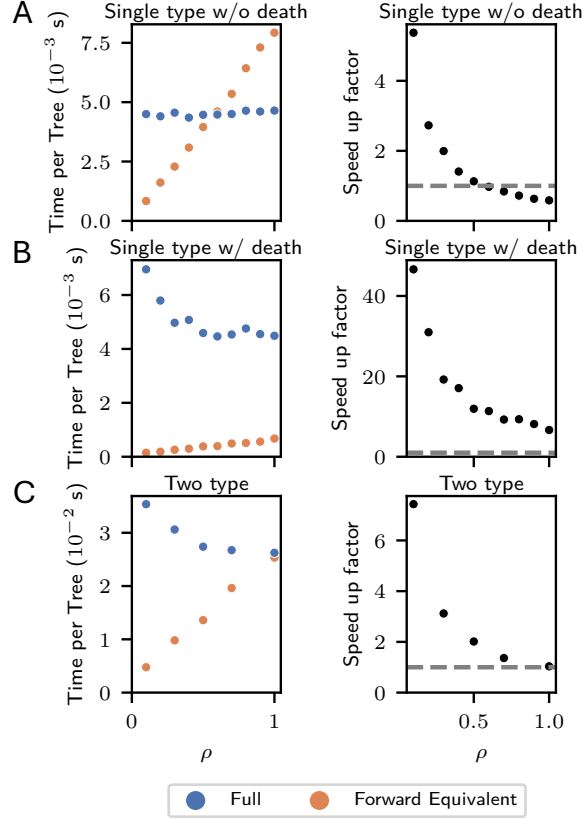
14

Figure 4: Comparison of computational costs. Time per tree (s) is the total amount of simulation time required to produce 50 non-empty trees, divided by 50. Speed up factor is the ratio of the time per tree from the full simulation to the forward equivalent simulation. All sampling occurs at present with probability ρ for each extant lineage identically across types. Horizontal dashed line at speed up factor = 1. A: single-type without death. $\lambda = 1.0$, $\mu = 0$, $t_{\max} = 5$. B: single-type with death. $\lambda = 1.0$, $\mu = 1.0$, $t_{\max} = 10$. C: two-type with death. Two types: Fit and Unfit. $\lambda_{\mathsf{Fit}} = 1.0$, $\lambda_{\mathsf{Unfit}} = 0.25$, $\mu_{\mathsf{Fit}} = \mu_{\mathsf{Unfit}} = 0.25$.

The speed-up factor is decreasing with ρ, and is larger than 1 for $\rho \leq 0.5$, reaching ≈ 5 at $\rho = 0.1$. Regressing the log speed up factor on $\log \rho$ gives the approximation $\mathsf{speed\ up\ factor} \approx 0.58\rho^{-0.99}$, consistent with the inverse dependence on ρ predicted by Proposition 1. The speed-up factor is smaller by approximately a factor of 2 relative to the theory. We believe this is because, in this model, the forward-equivalent incurs some overhead relative to the full simulation because it must sample from time-inhomogenous rather than constant-rate Poisson processes (see Appendix A for implementation details). For $\rho \gtrsim 0.6$, the gain from avoiding simulating unobserved events does not overcome this computational overhead, and the speed-up factor is less than one. We emphasize, however, that this computational overhead increases computation time by a constant factor which does not grow with phylogeny size. Thus, it will be overcome by a small amount of subsampling even for very large phylogenies. For models that do not include death, the forward-equivalent simulation is faster than the full simulation when there is at least a moderate amount of subsampling. When the full model itself includes time-inhomogenous rate parameters, the forward-equivalent simulation will not have this computational overhead relative to the full simulation.

In the second row of Figure 4, we present similar results for a single-type model that includes death. Precisely, we consider a regime with birth rate $\lambda = 1.0$ and death rate $\mu = 1.0$, run for $t_{\max} = 10$ units of biological time. The computational savings are even more dramatic in this case,

15

at $\rho = 0.1$ reaching a speed-up factor of ≈ 45 and at $\rho = 1$ a speed-up factor of ≈ 6. In this case, even at $\rho = 1$, the forward-equivalent simulation is much faster than the full simulation. This is because lineages that die are not sampled. For models that include death, the forward-equivalent simulation can be more efficient for all sub-sampling schemes.

We also observe that the computation time for the full simulation is no longer approximately constant in the sampling probability ρ. In fact, unlike in the forward-equivalent simulation, the computation tends to be larger for smaller values of ρ. We suspect this is a result of the computational cost of pruning the phylogeny, which is more computationally intensive when a smaller fraction of the tree is sampled.

Finally, in the third row of Figure 4, we present similar results for the two-type model described in Section 4.1, and we observe similar phenomena to that appearing in the single-type simulations.

In Supplementary Figure S1, we present further plots demonstrating the computational overhead described above and the linear scaling of computation time in the total number of simulated events.

4.3   Subsampling from massive phylogenies

The previous section focused on situations in which, although the forward-equivalent simulation was much faster than the full simulation, the phylogeny sizes were small enough that both approaches were computationally feasible. The computational cost of the forward-equivalent simulation scales with the size of the sub-sampled rather than the full phylogeny. Thus, in principle, it allows for simulation of phylogenies subsampled from arbitrarily large populations.

As an example, we consider a multi-type model with two types Unfit and Fit with birth rates $\lambda_{\mathsf{Unfit}} = 0.25$ and $\lambda_{\mathsf{Fit}} = 1.0$, death rates $\mu_{\mathsf{Unfit}} = 0.25$ and $\mu_{\mathsf{Fit}} = 0.5$, and mutation rates $\gamma_{\mathsf{Unfit,Fit}} = 0.1$ and $\gamma_{\mathsf{Fit,Unfit}} = 0.25$, respectively. Thus, the Fit type has higher birth rates and death rates but a larger net speciation rate than the Unfit type, i.e., $\lambda_{\mathsf{Fit}} - \mu_{\mathsf{Fit}} > \lambda_{\mathsf{Unfit}} - \mu_{\mathsf{Unfit}}$. Moreover, deleterious mutations occur at a higher rate than beneficial mutations. We simulate the population for biological time $t_{\max} = 47$ and sample each lineage in the present population with probability $\rho = 10^{-9}$. With these simulation parameters, we find we often generate trees with leaf counts in the 10s to 100s. This indicates that the full population size from which the phylogeny is sampled is in the 10s to 100s of *billions*. On a Macbook Pro with an Apple M1 Pro chip, each tree in this simulation required just over a second of computation time on average.

In Figure 5, we display a typical phylogeny drawn from this simulation. From such a tree, we see several features that are relevant to understanding phylogenies generated from experiments involving billions of cells. In particular, we observe that most birth events occur well before the present, and most (but not all) mutations occur after the most recent observed birth event. Most
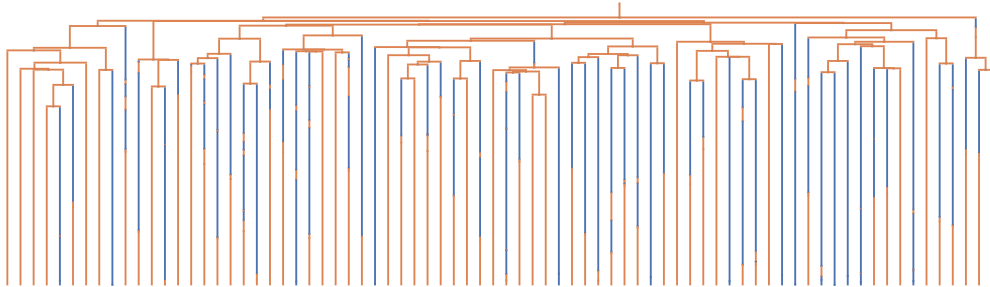

Figure 5: A sub-sampled phylogeny from a multi-type model with population size in the billions. The Fit type is in orange and the Unfit type is in blue.

16

lineages mutated to the Unfit type shortly after the most recent birth event, but mutated again to the Fit type before sampling at the present day. Further information regarding observed mutation rates and birth rates, their time dependence, and the behavior of various approaches to estimating underlying process parameters, can also be investigated empirically using such simulated phylogenies.

5   Discussion

This paper has proposed a novel method for simulating reconstructed phylogenies that leads to substantial computational savings when there is sparse sampling or high levels of death in the population. Many scientific studies rely on extensive phylogenetic simulation. Unfortunately, when the full population size is in the billions or more, as is typical in epidemiological, cancer evolution, and antibody evolution applications, the cost of the full simulation is prohibitive. The forward-equivalent simulation we propose makes simulating the reconstructed phylogeny feasible in these domains when the set of sampled lineages is substantially smaller than the full population, enabling much more realistic simulations and benchmark studies.

   We believe that the forward-equivalent simulation approach will facilitate improved inference of population dynamics. We describe some directions for future work.

- **Training of machine learning models.** Recently, deep learning approaches for estimating epidemiological parameters from reconstructed phylogenies have been proposed [31]. These methods are often trained on large datasets of phylogenies simulated with different model parameters. By enabling faster simulation, our method can enable the training of such models. In particular, the forward-equivalent simulation can allow the simulated phylogenies to be faithful to realistic population sizes common in epidemiological, cancer evolution, and other domains.

- **Simulation at inference-time.** Some inferential methods require simulation at inference time. For example, "Approximate Bayesian Computation" involves computing summary statistics of observed phylogenies, and then stochastically searching for model parameters which, in simulation, generate phylogenies with similar summary statistics [33]. A computational bottleneck in these approaches is the cost of simulation. We anticipate the forward-equivalent simulation may improve or enable simulation-based approaches to inference.

- **Identifiability.** Following the seminal paper of Louca and Pennell [14], there has been substantial interest in issues of identifiability in birth-death models [12, 13]. These papers study single-type models with time-varying birth and death rates, and show that, due to sub-sampling, there are multiple values of the underlying process parameters that can induce the same distribution over observed phylogenies. This calls into question the ability to use phylogenies to reconstruct parameters driving diversification and extinction dynamics in the past. Our simulation approach is related to a lack of identifiability in multi-type models: indeed, the full model and the forward-equivalent model induce the same distribution over reconstructed phylogenies, so cannot be distinguished using the observed data alone. Thus, this paper establishes that the identifiability issues that occur in single-type models also occur in more general multi-type models, and we specify for each full model a different model that belongs to the same equivalence class. The current paper identifies the utility of this lack of identifiability for the purpose of simulation, but it will also be important to investigate the challenges this introduces for the purpose of inference. Moreover, a complete description

of the classes of indistinguishable models and investigation of assumptions that may resolve identifiability issues, as in [12, 13], is left to future work.

6   Materials and methods

Software implementation.   We have made reproducible analysis available on GitHub (Code available at https://github.com/songlab-cal/forward-equivalent-trees). We used the Python package BDMS [3]—which is developed by us and other collaborators—as a reference implementation of the full simulation, and developed wrapper code to implement the forward-equivalent mapping $\Theta \mapsto \Theta^{\mathsf{FE}}$ and perform simulation studies. We use the Python package ETE [9] for tree manipulation and visualization.

Acknowledgments

Appendix

A   Further implementation details

In this section, we describe additional implementation details to enable Algorithm 1 to run in time which scales linearly in the number of events that occur in the full population. First, we must implement GETNEXTEVENT in constant time in the population size, which is described in Appendix A.1. We must also store, at any given time, the active population in a data structure that supports insertion, deletion, and selection of an element uniformly at random, in constant time in the population size, which is described in Appendix A.2. Finally, our ability to implement GETNEXTEVENT requires we perform precomputations before simulation time. These are described in Appendix A.3.

A.1   Implementing GETNEXTEVENT in constant time in the population size

As described in Section 3.1, in order for Algorithm 1 to have time complexity linear in the number of events in the full phylogeny, we must implement the function GETNEXTEVENT in constant time in the population size. Our implementation relies on the following fact. Consider a Poisson point process with time-varying rate given by $\theta(\tau) \geq 0$. Let

$$\Theta(\tau, \tau') := \int_\tau^{\tau'} \theta(\tau)\mathrm{d}\tau, \qquad \Theta^{-1}(\tau, z) := \sup\{\tau' \in \mathbb{R} \mid \Theta(\tau, \tau') \leq z\}. \tag{S.1}$$

Consider n iid copies of this Poisson point process. For a fixed time τ, consider the next time after τ that an event occurs in any one of these n processes. It is routine to show that this arrival time has distribution given by $\Theta^{-1}(\tau, Z/n)$ where $Z \sim \mathrm{Exp}(1)$ has exponential distribution with rate parameter 1 (that is, has density given by e^{-z} for $z \geq 0$). Note that $\Theta(\tau, Z/n)$ can be infinite if $\lim_{\tau' \to \infty} \Theta(\tau, \tau') < \infty$ and $Z/n \geq \lim_{\tau' \to \infty} \Theta(\tau, \tau')$. This corresponds to the event that none of the n Poisson processes have any events occur after time τ, which can occur when the Poisson rate function is integrable.

   The GETNEXTEVENT function takes in the current time τ and the number of extant lineages of each type $|S_a|$. Given the discussion in the previous paragraph, this provides a way of determining when the next event in the population occurs, and what type it is, where the type specifies whether it is a birth, death, mutation, or sampling event, and the types of the parent and child nodes in the phylogeny. For each event type, we generate the time of the next event of that type. The next event type is chosen to be the one that happens the soonest among these generated times. Formally, for $a, b = 1, \ldots, d$, define

$$\Lambda_{a,b}(\tau_1, \tau_2) = \int_{\tau_1}^{\tau_2} \lambda_{a,b}(\tau)\mathrm{d}\tau, \qquad M_a(\tau_1, \tau_2) = \int_{\tau_1}^{\tau_2} \mu_a(\tau)\mathrm{d}\tau,$$
$$\Gamma_{a,b}(\tau_1, \tau_2) = \int_{\tau_1}^{\tau_2} \gamma_{a,b}(\tau)\mathrm{d}\tau, \qquad \Psi_a(\tau_1, \tau_2) = \int_{\tau_1}^{\tau_2} \psi_a(\tau)\mathrm{d}\tau. \tag{S.2}$$

Then define

$$\Lambda_{a,b}^{-1}(\tau, z) = \sup\{\tau' \in \mathbb{R} \mid \Lambda_{a,b}(\tau, \tau') \leq z\}, \qquad M_a^{-1}(\tau_1, z) = \sup\{\tau' \in \mathbb{R} \mid M_a(\tau, \tau') \leq z\}$$
$$\Gamma_{a,b}^{-1}(\tau_1, z) = \sup\{\tau' \in \mathbb{R} \mid \Gamma_{a,b}(\tau, \tau') \leq z\} \qquad \Psi_a^{-1}(\tau_1, z) = \sup\{\tau' \in \mathbb{R} \mid \Psi_a(\tau, \tau') \leq z\}, \tag{S.3}$$

where these can be possibly infinite (for example, if one of the rate parameters is 0). We draw exponential random variables $\{Z_{a,b}^\gamma\}_{a,b=1}^d$, $\{Z_a^\mu\}_{a=1}^d$, $\{Z_{a,b}^\gamma\}_{a,b=1}^d$, $\{Z_a^\psi\}_{a=1}^d$ iid from $\mathrm{Exp}(1)$ so that

we have one exponential random variable for each event type. The occurrence of the next birth event $a \to a, b$ occurs at time $\Lambda_{a,b}^{-1}(\tau, Z_{a,b}^{\lambda}/|S_a|)$; the next death of a lineage with type a occurs at time $M_a^{-1}(\tau, Z_a^{\mu}/|S_a|)$; the next anagenetic mutation $a \to b$ occurs at time $\Gamma_{a,b}^{-1}(\tau, Z_{a,b}^{\gamma}/|S_a|)$; and the next sampling event occurs at $\Psi_a^{-1}(\tau, Z_a^{\psi}/|S_a|)$. The next event of any type occurs at the minimum of these, and the type of the next event is determined by which process achieves the minimum. Thus, GETNEXTEVENT can be implemented by the pseudo-code in Algorithm 3. Provided $\Lambda_{a,b}^{-1}(\tau, Z_{a,b}^{\lambda}/|S_a|)$, $M_a^{-1}(\tau, Z_a^{\mu}/|S_a|)$, $\Gamma_{a,b}^{-1}(\tau, Z_{a,b}^{\gamma}/|S_a|)$, $\Psi_a^{-1}(\tau, Z_a^{\psi}/|S_a|)$ can be evaluated

---

Algorithm 3 GETNEXTEVENT($\{|S_a|\}, \tau$)

Draw $\{Z_{a,b}^{\gamma}\}_{a,b=1}^d$, $\{Z_a^{\mu}\}_{a=1}^d$, $\{Z_{a,b}^{\gamma}\}_{a,b=1}^d$, $\{Z_a^{\psi}\}_{a=1}^d$ iid from $\mathrm{Exp}(1)$;

$\tau_* \leftarrow \min_{a,b} \left\{ \Lambda_{a,b}^{-1}(\tau, Z_{a,b}^{\lambda}/|S_a|), M_a^{-1}(\tau, Z_a^{\mu}/|S_a|), \Gamma_{a,b}^{-1}(\tau, Z_{a,b}^{\gamma}/|S_a|), \Psi_a^{-1}(\tau, Z_a^{\psi}/|S_a|) \right\}$;

EventType $\leftarrow \arg \min \left\{ \Lambda_{a,b}^{-1}(\tau, Z_{a,b}^{\lambda}/|S_a|), M_a^{-1}(\tau, Z_a^{\mu}/|S_a|), \Gamma_{a,b}^{-1}(\tau, Z_{a,b}^{\gamma}/|S_a|), \Psi_a^{-1}(\tau, Z_a^{\psi}/|S_a|) \right\}$;

if $\tau^* < 0$ **then return** $(0, \text{EventType})$;

else return $(\tau^*, \text{EventType})$;

---

in constant time, the time complexity of Algorithm 3 is $O(d^2)$, and in particular, does not grow with population size.

In the case that the time of the next event occurs after the present day, we instead exit the loop and advance all extant lineages to the present day. We then carry out the present day sampling process: each lineage is sampled with probability $\rho_{a,0}$, and upon sampling, goes extinct with probability $r_{a,0}$. If our model includes CSEs prior to the present day, we follow a similar logic when the next event time occurs after the next CSE. We can thus implement GetNextEvent using the pseudo-code in Algorithm 3 complemented by the additional CSE logic. For simplicity, the pseudo-code represents a case without CSEs.

In some cases (for example, for constant rate Poisson processes), $\Gamma_{a,b}^{-1}$, M_a^{-1}, $\Lambda_{a,b}^{-1}$, and Ψ_a^{-1} have analytic forms that permit constant time evaluation. When analytic forms are not available (which is typically the case for the forward-equivalent models), we leverage precomputations. Note that $\Theta^{-1}(\tau, z) = \Theta^{-1}(0, \Theta(0, \tau) + z)$. Thus, to allow for constant time evaluation of Θ^{-1} at simulation time, we can precompute Θ at an equispaced grid of $\tau \in [0, t_{\max}]$ and Θ^{-1} on an equispaced grid of $z \in [0, \Theta(0, t_{\max})]$. We describe the precomputations for $\Lambda_{a,b}^{-1}$, $M_a^{-1}(0, z)$, $\Gamma_{a,b}^{-1}(0, z)$, $\Psi_a^{-1}(0, z)$ in Appendix A.3. Then, at simulation time, we compute $\Theta(0, \tau)$ and $\Theta^{-1}(0, \Theta(0, \tau) + z)$ by linear interpolation with respect to this mesh. This is a constant time operation, because, for example, the interval in which τ occurs has left and right indices (using Python indexing) given by $\lfloor \tau/\Delta\tau \rfloor$ and $\lceil \tau/\Delta\tau \rceil$, where $\Delta\tau$ is the size of the gaps in the discrete grid. Thus, the evaluation requires index-based array lookup.

We remark that in our code,[1] the implementation of the mutation process is slightly different. The event type specifies only the type of the type a experiencing the mutation, not the type b into which it mutates. Conditional on type a mutating, we draw type b from the appropriate distribution. In particular, rather than compute $\Gamma_{a,b}(\tau_1, \tau_2)$, we compute

$$\Gamma_a(\tau_1, \tau_2) = \int_{\tau_1}^{\tau_2} \sum_b \gamma_{a,b}(\tau) \mathrm{d}\tau, \tag{S.4}$$

and define $\Gamma_a^{-1}(\tau, z)$ in the obvious way. Then, conditional on a mutation occurring to type a at time τ^*, we draw b from $\mathbb{P}(b) = \gamma_{a,b}(\tau^*)/\sum_{b'} \gamma_{a,b'}(\tau^*)$. We do this based on the API of the BDMS

---

[1] Code available at https://github.com/songlab-cal/forward-equivalent-trees

simulation engine we use [3], but do not claim there is any intrinsic advantage or disadvantage of either approach relative to the other.

A.2   Randomized Set

Algorithm 1 requires keeping sets of pointers to the active nodes in the population of each type at each point in time τ. This data structure must support insertion and deletion (see, for example, line 11 of Algorithm 1) as well as the ability to drawn an element of uniformly at random (see, for example, line 16 of Algorithm 3). Such a data structure is called a "randomized set," and is the data structure we use in our simulations.

A.3   Precomputation

As described in Appendix A.1, in the absence of analytic forms, we compute $\Theta^{-1}(\tau, z)$ by precomputing $\Theta(0, \tau)$ and $\Theta^{-1}(0, z)$ on a discrete grid, for Θ corresponding to $\Gamma_{a,b}$, M_a, $\Lambda_{a,b}$, or Ψ_a. We here describe this precomputation for the forward-equivalent model.

For each type a, we computed $E_a(\tau)$ according to (2) using Euler's method on $[0, t_{\max}]$ with time discretization $\Delta\tau = .01$. This gives $E_a(\tau)$ on a discrete grid, which we extend to any τ by linear interpolation. The model parameters $(\pi^{\mathsf{FE}}, \lambda_{a,b}^{\mathsf{FE}}, \mu_a^{\mathsf{FE}}, \psi_a, r_a^{\mathsf{FE}}, \gamma_{a,b}^{\mathsf{FE}}, \rho_{a,l}^{\mathsf{FE}}, r_{a,l}^{\mathsf{FE}})$ can then be precomputed on the same discrete grid using (3), consuming space $O(d^2 t_{\max}/\Delta\tau)$ and requiring time $O(d^2 t_{\max}/\Delta\tau)$. We then precompute the integrals $\Gamma_{a,b}(0, \tau)$, $M_a(0, \tau)$, $\Lambda_{a,b}(0, \tau)$, $\Psi_a(0, \tau)$ on the same equispaced grid using numerical integration. For example, this gives us $\Lambda_{a,b}(0, k\Delta\tau)$ for some finite sequence of integers $k = 0, 1, \ldots K$. This allows us to evaluate $\Lambda_{a,b}^{-1}(0, z)$ on the mesh $\{0, \Lambda_{a,b}(0, \Delta\tau), \ldots, \Lambda_{a,b}(0, K\Delta\tau)\}$. This mesh is not equispaced. We get an evaluation of $\Lambda_{a,b}^{-1}(0, z)$ on an equispaced grid of $[0, \Lambda_{a,b}(0, K\Delta\tau)]$ by linear interpolation of the values computed on the mesh $\{0, \Lambda_{a,b}(0, 1\Delta\tau), \ldots, \Lambda_{a,b}(0, K\Delta\tau)\}$. At simulation time, all evaluations are done using linear interpolation on these grids, as described in Appendix A.1.

This approach requires storing precomputations in arrays whose memory footprint is $O(d^2 t_{\max}/\Delta\tau)$. The time-complexity of the precomputation is dominated by the numerical integration of (2), whose time complexity using Euler's method is $O(d^2 t_{\max}/\Delta\tau)$. The precomputation is performed only once, so this cost is amortized over the number of simulations carried out. For the simulations in this paper, the precomputations occurred in a few microseconds. For very large type spaces (i.e., large d), it is possible that both the space and time complexity of these precomputations would become prohibitive.

A.4   Implementation based on a priority queue

An alternative simulation approach to that provided by Algorithm 1 has runtime $O(\#\mathrm{nodes}(\mathcal{T})(d + \log(\#\mathrm{nodes}(\mathcal{T}))))$. Thus, it may be faster when the type space is large. We briefly outline this approach here.

We again simulate forward in time τ and maintain the population of extant individuals at time τ. Additionally, we maintain a priority queue containing the next event for each individual in the extant population. Thus, the next event in the population is given by the event in the priority queue with smallest time, which can be obtained in logarithmic time in the population size by virtue of the priority queue data structure. The algorithm thus proceeds by advancing time to this event, and then performing the event. When this event creates or modifies individuals (such as upon a birth event or a mutation event), the updated or newly created individuals are added to the population of extant individuals, and their next events are immediately sampled and pushed into the priority queue.

Inserting and popping from a priority queue has time complexity which grows logarithmically with the number of elements in the priority queue, which in this case corresponds to the extant population size. For any fixed individual, generating the next event occurring to this individual has time complexity $O(|\mathcal{A}|)$. This is due to generating the time and type of the next mutation event, which involves generating a random type from a state space of $|\mathcal{A}|$ types with non-uniform probabilities stored in some precomputed arrays. Thus, each iteration has time complexity $O(|\mathcal{A}| + \log(\#\text{nodes}(\mathcal{T})))$, and the number of iterations is equivalent to the number of events in the full phylogeny. The total time complexity is thus $O(\#\text{nodes}(\mathcal{T})(|\mathcal{A}| + \log(\#\text{nodes}(\mathcal{T}))))$. This should be compared with the time complexity $O(\#\text{nodes}(\mathcal{T})|\mathcal{A}|^2)$ of the implementation in Algorithm 1. The alternative implementation with a priority queue has worse complexity in terms of the number of events in the full phylogeny but better complexity in terms of the size of the type space. In applications with large type spaces, the priority queue based implementation may be preferred. However, since our original algorithm has runtime that is strictly linear in the size of the full phylogeny, we focus on it for this work.

B   Proof of Theorem 1

Consider a non-empty phylogeny $\mathcal{T} \neq T_\emptyset$. Let $p(\mathcal{T})$ be the probability density of observing $\mathcal{T}$ in the full model, $p(\mathcal{T} \mid \mathcal{T} \neq T_\emptyset)$ be the probability density of observing $\mathcal{T}$ in the full model conditional on the non-emptiness of $\mathcal{T}$, and let $p^{\mathsf{FE}}(\mathcal{T})$ be the probability density of observing $\mathcal{T}$ in the forward-equivalent model. We will show that $p(\mathcal{T} \mid \mathcal{T} \neq T_\emptyset)$ and $p^{\mathsf{FE}}(\mathcal{T})$ can both be computed by solving an ordinary differential equation. We will show $p(\mathcal{T} \mid \mathcal{T} \neq T_\emptyset) = p^{\mathsf{FE}}(\mathcal{T})$ by showing that these differential equations are the same. We draw on a large body of work which computes likelihoods in BDMS models using ordinary differential equations [15].

B.1   The non-observation probability

Eq. (2) agrees with (15) of [15] together with the modifications described in their appendix for handling concerted sampling events. Similar equations have appeared multiple time in earlier works; see [15] for references. For completeness, this section describes the derivation of these equations.

Between CSEs: For τ in an open interval between CSEs, consider a lineage of type a at time $\tau + \mathrm{d}\tau$. There are four ways that this lineage has no sampled descendants. We list this ways, and the probability of each, below.

1. No events occur in the interval $[\tau, \tau + \mathrm{d}\tau]$ and the lineage at time τ has no sampled descendants

$$\text{with probability } \Big(1 - \Big(\sum_{b=1}^{d} \lambda_{a,b}(\tau) + \mu_a(\tau) + \psi_a(\tau) + \sum_{b=1}^{d} \gamma_{a,b}(\tau) \Big)\mathrm{d}\tau \Big) E_a(\tau). \tag{S.1}$$

2. Birth $a \to (a, b)$ occurs in the interval $[\tau, \tau + \mathrm{d}\tau]$, and neither child has a sampled descendant

$$\text{with probability } \lambda_{a,b}(\tau)\mathrm{d}\tau E_a(\tau) E_b(\tau). \tag{S.2}$$

3. Death occurs in the interval $[\tau, \tau + \mathrm{d}\tau]$

$$\text{with probability } \mu_a(\tau)\mathrm{d}\tau. \tag{S.3}$$

22

4. Anagenetic mutation $a \to b$ occurs in the interval $[\tau, \tau + \mathrm{d}\tau]$, and the new lineage has not sampled descendants

$$\text{with probability } \gamma_{a,b}(\tau)\mathrm{d}\tau E_b(\tau). \tag{S.4}$$

Summing these possibilities (and noting that items 2 and 4 can occur for any $b = 1, \ldots, d$) and rearranging gives the expression for $\frac{\mathrm{d}E_a(\tau)}{\mathrm{d}\tau}$ in (2).

Initialization: 0^- refers to the moment immediately after that CSE occurring at the present, after that sampling and death events for that CSE have occurred. There is no possibility for any further sampling, so $E_a(0^-) = 1$.

At CSEs: The equation $E_a(t_l^-) = \lim_{\tau \uparrow t_l} E_a(\tau)$ holds because the time t_l^- is after the sampling and death events for the CSE have occurred, so that $E_a(t_l^-)$ is the continuous extension of the extinction probability on the interval of time following the CSE. The time t_l^+ is immediately before the sampling and death events for the CSE have occurred. At this point in time, there is only one way in which the lineage can have no sampled descendants: the lineage is not sampled during the CSE (which has probability $1 - \rho_{a,l}$), and the lineage following the CSE has no sampled descendants (which has probability $E_a(t_l^-)$). Thus, $E_a(t_l^+) = (1 - \rho_{a,l})E_a(t_l^-)$.

B.2 Unconditional probability density of the reconstructed phylogeny

Next, we describe how to compute $p(\mathcal{T})$ in the full model by solving a system of ordinary differential equations. This argument is provided in [15], which itself unifies arguments appearing many times previously. For completeness, we describe the argument here.

First, let us introduce some notation borrowed from [15]. The phylogeny $\mathcal{T}$ consists of a collection of edges $\mathcal{E}$, where an edge on the tree is defined as the segment spanning two adjacent events, where events are either birth events, anagenetic mutation events, sampling events at a non-CSEs, or CSEs (with the accompany sampling or death events). In the reconstructed phylogeny, death events are simply sampling events after which no further descendant lineages were sampled. Thus, sampling events that were followed by immediate death (whose probability is given by $\psi_a(\tau)$ and $r_a(\tau)$), and sampling events followed by survival but the failure to sample any future descendants, look the same. In the reconstructed phylogeny, sampling events can be internal to the tree or at leaf nodes, but all leaf nodes correspond to sampling events.

Each edge $e \in \mathcal{E}$ spans a time interval $s_e < \tau < t_e$. Letting a_e denote the type of the lineage along edge e, we let $g_e(\tau)$ denote the probability density that a lineage initialized at time τ with type a_e would give rise to the sub-tree of $\mathcal{T}$ "anchored" at time τ on edge e. We use this terminology to refer to the sub-tree whose root is at time τ on edge e and which contains all points on τ which are descendants of this point. The probability density $g_e(\tau)$ *does not* condition on non-emptiness.

The quantities $g_e(\tau)$ are given by the solution to an ODE which is integrated backwards in time along each edge, with the initial condition at s_e determined by the solution along the more recent adjacent edges.

ODE: The ODE is given by

$$\frac{\mathrm{d}g_e(\tau)}{\mathrm{d}\tau} = \lambda_{a,a}(\tau)\big(2E_a(\tau) - 1\big)g_e(\tau) + \sum_{b \neq a}\lambda_{a,b}(\tau)\big(E_b(\tau) - 1\big)g_e(\tau)$$

$$+ \mu_a(\tau)\big(0 - g_e(\tau)\big)$$
$$+ \psi_a(\tau)\big(0 - g_e(\tau)\big) \tag{S.5}$$
$$+ \sum_{b \neq a}\gamma_{a,b}(\tau)(0 - g_e(\tau)).$$

To see this, consider the probability density of the sub-tree anchored at time $\tau + \mathrm{d}\tau$ on edge e. The probability of this tree is given by the product of the probability that a lineage of type a at time $\tau + \mathrm{d}\tau$ does not experience any observed events in the time interval $[\tau, \tau + \mathrm{d}\tau]$ and the probability density of the sub-tree anchored at time τ. There are three ways that no events are observed in the time interval $[\tau, \tau + \mathrm{d}\tau]$. The first way is that no events occur. This has probability $1 - \big(\sum_{b=1}^{d}\lambda_{a,b}(\tau) + \mu_a(\tau) + \psi_a(\tau) + \sum_{b \neq a}\gamma_{a,b}(\tau)\big)\mathrm{d}\tau$. The second is that a birth of type $a \to a, a$ occurs and one of the child lineages has no sampled descendants. This has probability $2\lambda_{a,a}(\tau)\mathrm{d}\tau E_a(\tau)$, where the factor of 2 results from the possibility that either lineage is the one without sampled descendants. The third way is that a birth $a \to (a, b)$ for $b \neq a$ occurs and the child lineage of type b has no sampled descendants. This has probability $\lambda_{a,b}(\tau)\mathrm{d}\tau E_b(\tau)$. Thus,

$$g_e(\tau + \mathrm{d}\tau) = \left\{1 - \left[\sum_{b=1}^{d}\lambda_{a,b}(\tau) + \mu_a(\tau) + \psi_a(\tau) + \sum_{b \neq a}\gamma_{a,b}(\tau)\right]\mathrm{d}\tau \right. \tag{S.6}$$

$$\left. + 2\lambda_{a,a}(\tau)\mathrm{d}\tau E_a(\tau) + \sum_{b \neq a}\lambda_{a,b}(\tau)\mathrm{d}\tau E_b(\tau)\right\}g_e(\tau).$$

This rearranges to (S.5).

Initial conditions: The initial condition at s_e depends on the type of event occurring at the terminal node of an edge.

- **Births:** For birth events resulting in edges (e_1, e_2) with types (a, b) or (b, a),

$$g_e(s_e) = \lambda_{a,b}(s_e)g_{e_1}(s_e)g_{e_2}(s_e). \tag{S.7a}$$

  The factor $\lambda_{a,b}(s_e)$ gives the probability density of a birth occurring at time τ, and $g_{e_1}(s_e)g_{e_2}(s_e)$ the probability of the two children giving rise to their observed sub-trees.

- **Anagenetic mutations:** For anagenetic mutations $a \to b$ with offspring edge e_1,

$$g_e(s_e) = \big(\gamma_{a,b}(s_e) + \lambda_{a,b}(s_e)E_a(s_e)\big)g_{e_1}(s_e). \tag{S.7b}$$

  An anagenetic mutation $a \to b$ in the reconstructed phylogeny can result from two types of events in the full phylogeny. The first type of event is the anagenetic mutation $a \to b$ in the full phylogeny with the child lineage giving rise to the observed sub-tree. This has probability density $\gamma_{a,b}(s_e)g_{e_1}(s_e)$. The second type of event is a birth $a \to (a, b)$, with the child of type a having no observed descendants and the child of type b giving rise to the observed sub-tree. This has probability density $\lambda_{a,b}(s_e)E_a(s_e)g_{e_1}(s_e)$. Adding these two terms gives (S.7b).

24

- **Non-CSE sampling events:** These are initialized by

$$g_e(s_e) = \begin{cases} \psi_a(s_e)(r_a(s_e) + (1 - r_a(s_e))E_a(s_e)) & \text{for leaf nodes,} \\ \psi_a(s_e)(1 - r_a(s_e))g_{e_1}(s_e) & \text{for internal nodes.} \end{cases} \quad \text{(S.7c)}$$

A sampled leaf node at a non-CSE time can result from two types of events in the full phylogeny. The first type is that the lineage was sampled and immediately went extinct. This has probability density $\psi_a(s_e)r_a(s_e)$. The second type is that the lineage was sampled, did not immediately go extinct, and then had no sampled descendants. This has probability density $\psi_a(s_e)(1 - r_a(s_e))E_a(s_e)$. Adding these two terms gives the first line of (S.7c).

A sampled internal node at a non-CSE time can result from only one type of event in the full phylogeny. This event is that the lineage was sampled, did not immediately go extinct, and then gave rise to the observed sub-tree. This has probability density $\psi_a(s_e)(1 - r_a(s_e))g_{e_1}(s_e)$, giving the second line of (S.7c).

- **CSE sampling events:** These are initialized by

$$g_e(t_l^+) = \begin{cases} (1 - \rho_{a,l})g_{e_1}(t_l^-) & \text{for non-sampled nodes,} \\ \rho_{a,l}(r_{a,l} + (1 - r_{a,l})E_a(t_l^-)) & \text{for sampled leaf nodes,} \\ \rho_{a,l}(1 - r_{a,l})g_{e_1}(t_l^-) & \text{for sampled internal nodes.} \end{cases} \quad \text{(S.7d)}$$

If no sampling occurs at the CSE in the reconstructed phylogeny, then no sampling occurred in the full phylogeny and the lineage at time t_l^- gave rise to the observed sub-tree. This has probability density $(1 - \rho_{a,l})g_{e_1}(t_l^-)$, giving the first line of (S.7d). The remaining two lines of (S.7d) follow by exactly the same argument leading to the two lines of (S.7c), with $\psi_a(s_e)$ replaced by $\rho_{a,l}$ and $r_a(s_e)$ replaced by $r_{a,l}$.

One initializes these ODEs at every leaf node in the reconstructed phylogeny, and then integrates backwards in time along all lineages until reaching the root. The value $g_e(t_{\max})$ on the root edge of the tree is the probability density of observing $\mathcal{T}$.

B.3 Probability density conditional on non-emptiness

For an edge e and time τ along edge e in $\mathcal{T}$, let $\bar{g}_e(\tau)$ denote the probability density that a lineage initialized at time τ with type a_e gives rise to the sub-tree of $\mathcal{T}$ anchored at time τ on edge e, conditional on such a lineage having an observed descendant. That is, according to Bayes rule,

$$\bar{g}_e(\tau) = \frac{g_e(\tau)}{1 - E_a(\tau)}. \quad \text{(S.8)}$$

Let $g_e^{\mathsf{FE}}(\tau)$ denote the quantity $g_e(\tau)$ for the forward equivalent model. That is, $g_e^{\mathsf{FE}}(\tau)$ is defined using parameters $(\pi^{\mathsf{FE}}, \lambda_{a,b}^{\mathsf{FE}}, \mu_a^{\mathsf{FE}}, \psi_a^{\mathsf{FE}}, r_a^{\mathsf{FE}}, \gamma_{a,b}^{\mathsf{FE}}, \rho_{a,l}^{\mathsf{FE}}, r_{a,l}^{\mathsf{FE}})$ in (S.5) and (S.7), with $E_a^{\mathsf{FE}}(\tau)$ as determined by these parameters according to Appendix B.1. Denoting by e_* the root lineage of $\mathcal{T}$, Theorem 1 will follow by establishing

$$\bar{g}_{e_*}(t_{\max}) = g_{e_*}^{\mathsf{FE}}(t_{\max}). \quad \text{(S.9)}$$

The remainder of the proof consists of establishing (S.9). This is done by showing that $\bar{g}_e(\tau)$ obeys the same ODEs and initial conditions as $g_e^{\mathsf{FE}}(\tau)$, given by (S.5) and (S.7) with the forward-equivalent model parameters.

First, we collect some facts about the forward-equivalent model parameters. By (3), $r_{a,0}^{\mathsf{FE}} = r_{a,0} + (1 - r_{a,0})E_a(0^-) = 1$ and $\rho_{a,0}^{\mathsf{FE}} = \rho_{a,0}/(1 - E_a(0^+)) = 1$. As explained in Appendix B.1, $E_a^{\mathsf{FE}}(0^-) = 1$. Thus, $E_a^{\mathsf{FE}}(0^+) = (1 - \rho_{a,0}^{\mathsf{FE}})E_a^{\mathsf{FE}}(0^-) = 0$. Then, because $\mu_a^{\mathsf{FE}}(0) = 0$, Eqs. (2) and (2) imply $E_a^{\mathsf{FE}}(\tau) = 0$ for all $\tau \geq 0$ (excluding $E_a^{\mathsf{FE}}(0^-)$). We collect these identities in the next display:

$$\rho_{a,0}^{\mathsf{FE}} = 1, \quad \rho_{a,0}^{\mathsf{FE}} = 1, \quad E_a^{\mathsf{FE}}(0^-) = 1, \quad E_a^{\mathsf{FE}}(\tau) = 0 \text{ for } \tau > 0^-. \tag{S.10}$$

Agreement of ODEs. We differentiate (S.8) to get the ODE satisfied by $\bar{g}_e(\tau)$. In particular,

$$\frac{\mathrm{d}\bar{g}_e(\tau)}{\mathrm{d}\tau} = \frac{\frac{\mathrm{d}}{\mathrm{d}\tau}g_e(\tau)}{1 - E_a(\tau)} + \frac{g_e(\tau)}{(1 - E_a(\tau))^2}\frac{\mathrm{d}E_a(\tau)}{\mathrm{d}\tau}. \tag{S.11}$$

Using (2) and (S.5), this is

$$= \frac{(\lambda_{a,a}(\tau)E_a(\tau) + \sum_b \lambda_{a,b}(\tau)(E_b(\tau) - 1) - \mu_a(\tau) - \psi_a(\tau) - \sum_b \gamma_{a,b}(\tau))g_e(\tau)}{1 - E_a(\tau)}$$

$$+ \frac{g_e(\tau)}{(1 - E_a(\tau))^2}\Big(\sum_b \lambda_{a,b}(\tau)(E_a(\tau)E_b(\tau) - E_a(\tau)) + \mu_a(\tau)(1 - E_a(\tau)) \tag{S.12}$$

$$+ \psi_a(\tau)(0 - E_a(\tau)) + \sum_b \gamma_{a,b}(\tau)(E_b(\tau) - E_a(\tau))\Big).$$

The terms involving $\lambda_{a,b}(\tau)$ are $g_e(\tau)/(1 - E_a(\tau))$ multiplied by

$$\lambda_{a,a}(\tau)E_a(\tau) + \sum_b \lambda_{a,b}(\tau)(E_b(\tau) - 1) + \frac{\sum_b \lambda_{a,b}(\tau)(E_a(\tau)E_b(\tau) - E_a(\tau))}{1 - E_a(\tau)}$$

$$= \lambda_{a,a}(\tau)E_a(\tau) - \sum_b \lambda_{a,b}^{\mathsf{FE}}(\tau) - \sum_b \frac{1 - E_b(\tau)}{1 - E_a(\tau)}E_a(\tau)\lambda_{a,b}(\tau). \tag{S.13}$$

The terms involving $\mu_a(\tau)$ are $g_e(\tau)/(1 - E_a(\tau))$ multiplied by

$$-\mu_a(\tau) + \frac{\mu_a(\tau)(1 - E_a(\tau))}{1 - E_a(\tau)} = 0 = \mu_a^{\mathsf{FE}}(\tau). \tag{S.14}$$

The terms involving $\psi_a(\tau)$ are $g_e(\tau)/(1 - E_a(\tau))$ multiplied by

$$-\psi_a(\tau) - \psi_a(\tau)\frac{E_a(\tau)}{1 - E_a(\tau)} = -\psi_a^{\mathsf{FE}}(\tau). \tag{S.15}$$

The terms involving $\gamma_{a,b}(\tau)$ are $g_e(\tau)/(1 - E_a(\tau))$ multiplied by

$$-\sum_b \gamma_{a,b}(\tau) + \frac{\sum_b \gamma_{a,b}(\tau)(E_b(\tau) - E_a(\tau))}{1 - E_a(\tau)} = -\sum_b \frac{1 - E_b(\tau)}{1 - E_a(\tau)}\gamma_{a,b}(\tau). \tag{S.16}$$

Combining these calculations,

$$\frac{\mathrm{d}\bar{g}_e(\tau)}{\mathrm{d}\tau}$$

$$= \Big(-\sum_b \lambda_{a,b}^{\mathsf{FE}}(\tau) - \mu_a^{\mathsf{FE}}(\tau) - \psi_a^{\mathsf{FE}}(\tau) - \sum_b \frac{1 - E_b(\tau)}{1 - E_a(\tau)}(\gamma_{a,b}(\tau) + E_a(\tau)\lambda_{a,b}(\tau)) + \lambda_{a,a}(\tau)E_a(\tau)\Big)\bar{g}_e(\tau)$$

$$= \Big(-\sum_b \lambda_{a,b}^{\mathsf{FE}}(\tau) - \mu_a^{\mathsf{FE}}(\tau) - \psi_a^{\mathsf{FE}}(\tau) - \sum_{b \neq a} \frac{1 - E_b(\tau)}{1 - E_a(\tau)}(\gamma_{a,b}(\tau) + E_a(\tau)\lambda_{a,b}(\tau))\Big)\bar{g}_e(\tau)$$

$$= \Big(-\sum_b \lambda_{a,b}^{\mathsf{FE}}(\tau) - \mu_a^{\mathsf{FE}}(\tau) - \psi_a^{\mathsf{FE}}(\tau) - \sum_{b \neq a} \gamma_{a,b}^{\mathsf{FE}}(\tau)\Big)\bar{g}_e(\tau).$$

$$\tag{S.17}$$

where the second inequality uses that $(1 - E_a(\tau))/(1 - E_a(\tau)) = 1$ and $\gamma_{a,a}(\tau) = 0$. Because $E_a^{\mathsf{FE}}(\tau) = 0$ for all $\tau \geq 0$, this ODE is (S.5) under the forward-equivalent model parameters.

Agreement of initial conditions. We now check agreement of the initial conditions.

- **Births:** For birth events resulting in edges (e_1, e_2) with types (a, b) or (b, a), Bayes rule and (S.7a) give

$$
\bar{g}_e(s_e) = \frac{g_e(s_e)}{1 - E_a(s_e)} = \frac{\lambda_{a,b}(\tau)g_{e_1}(s_e)g_{e_2}(s_e)}{1 - E_a(s_e)} = \lambda_{a,b}(\tau)(1 - E_b(s_e))\frac{g_{e_1}(s_e)}{1 - E_a(s_e)}\frac{g_{e_2}(s_e)}{1 - E_b(s_e)}
$$
$$
= \lambda_{a,b}^{\mathsf{FE}}(\tau)\bar{g}_{e_1}(\tau)\bar{g}_{e_2}(\tau),
$$

(S.18)

  which is (S.7a) for the forward-equivalent model.

- **Anagenetic mutations:** For anagenetic mutations $a \to b$ with offspring edge e_1, Bayes rule and (S.7b) give

$$
\bar{g}_e(s_e) = \frac{g_e(s_e)}{1 - E_a(s_e)} = \frac{\gamma_{a,b}(s_e) + \lambda_{a,b}(s_e)E_a(s_e)}{1 - E_a(s_e)}g_{e_1}(s_e)
$$
$$
= \frac{(1 - E_b(s_e))(\gamma_{a,b}(s_e) + \lambda_{a,b}(s_e)E_a(s_e))}{1 - E_a(s_e)}\frac{g_{e_1}(s_e)}{1 - E_b(s_e)} = \gamma_{a,b}^{\mathsf{FE}}(s_e)\bar{g}_{e_1}(s_e),
$$

(S.19)

  which is (S.7b) for the forward-equivalent model.

- **Non-CSE sampling events:** At leaf nodes,

$$
\bar{g}_e(s_e) = \frac{g_e(s_e)}{1 - E_a(s_e)} = \frac{\psi_a(s_e)(r_a(s_e) + (1 - r_a(s_e))E_a(s_e))}{1 - E_a(s_e)}
$$
$$
= \psi_a^{\mathsf{FE}}(s_e)(r_a^{\mathsf{FE}}(s_e) + (1 - r_a^{\mathsf{FE}}(s_e))E_a^{\mathsf{FE}}(s_e)),
$$

(S.20)

  which is the first line of (S.7c) for the forward-equivalent model. At internal nodes,

$$
\bar{g}_e(s_e) = \frac{g_e(s_e)}{1 - E_a(s_e)} = \frac{\psi_a(s_e)(1 - r_a(s_e))g_{e_1}(s_e)}{1 - E_a(s_e)}
$$
$$
= \frac{\psi_a(s_e)}{1 - E_a(s_e)}(1 - E_a(s_e))(1 - r_a(s_e))\frac{g_{e_1}(s_e)}{1 - E_a(s_e)}
$$
$$
= \psi_a^{\mathsf{FE}}(s_e)(1 - r_a^{\mathsf{FE}}(s_e))g_{e_1}^{\mathsf{FE}}(s_e),
$$

(S.21)

  which is the second line of (S.7c) for the forward equivalent model.

- **CSE sampling events:** Using $\rho_{a,l}^{\mathsf{FE}} = \rho_{a,l}/(1 - E_a(t_l^+))$ (by (3)) and $E_a(t_l^+) = (1 - r_{a,l})E_a(t_l^-)$ (by (2)), that $1 - \rho_{a,l} = \frac{1 - \rho_{a,l}^{\mathsf{FE}}}{1 - \rho_{a,l}^{\mathsf{FE}}E(t_l^-)}$. Then, for non-sampling at CSE times,

$$
\bar{g}_e(t_l^+) = \frac{g_e(t_l^+)}{1 - E_a(t_l^+)} = \frac{(1 - \rho_{a,l})g_{e_1}(t_l^-)}{1 - (1 - \rho_{a,l})E_1(t_l^-)}
$$
$$
= \frac{\frac{1 - \rho_{a,l}^{\mathsf{FE}}}{1 - \rho_{a,l}^{\mathsf{FE}}E(t_l^-)}g_{e_1}(t_l^-)}{1 - \frac{1 - \rho_{a,l}^{\mathsf{FE}}}{1 - \rho_{a,l}^{\mathsf{FE}}E(t_l^-)}E_1(t_l^-)} = (1 - \rho_{a,l}^{\mathsf{FE}})\frac{g_{e_1}(t_l^-)}{1 - E_1(t_l^-)} = (1 - \rho_{a,l}^{\mathsf{FE}})\bar{g}_{e_1}(t_l^-),
$$

(S.22)

which is the first line of (S.7d) for the forward equivalent model. For sampled leaf nodes at CSE times,

$$\bar{g}_e(t_l^+) = \frac{g_e(t_l^+)}{1 - E_a(t_l^+)} = \frac{\rho_{a,l}(r_{a,l} + (1 - r_{a,l})E_a(t_l^-))}{1 - E_a(t_l^+)} = \rho_{a,l}^{\mathsf{FE}} r_{a,l}^{\mathsf{FE}}, \tag{S.23}$$

which is the second line of (S.7d) for the forward equivalent model (recall $E_a^{\mathsf{FE}}(t_l^+) = 0$). For sampled internal nodes at CSE times,

$$\bar{g}_e(t_l^+) = \frac{g_e(t_l^+)}{1 - E_a(t_l^+)} = \frac{\rho_{a,l}(1 - r_{a,l})g_{e_1}(t_l^-)}{1 - E_a(t_l^+)} = \rho_{a,l}^{\mathsf{FE}}(1 - r_{a,l}^{\mathsf{FE}})\frac{g_{e_1}(t_l^-)}{1 - E_a(t_l^-)} = \rho_{a,l}^{\mathsf{FE}}(1 - r_{a,l}^{\mathsf{FE}})\bar{g}_{e_1}(t_l^-).$$
$$\tag{S.24}$$

where we have used $1 - r_{a,l}^{\mathsf{FE}} = (1 - r_{a,l})(1 - E_a(t_l^-))$ by (3). This is the third line of (S.7d) for the forward equivalent model.

Thus, we have confirmed that $\bar{g}_e(\tau)$ and $g_e^{\mathsf{FE}}(\tau)$ obey the same ODE and initial conditions. We conclude (S.9), which completes the proof of Theorem 1.

C Proof of Proposition 2 and Proposition 1

For a tree T, let $\#\mathrm{nodes}(T)$ be the number of nodes of T. In the context of the BDMS model, nodes correspond to (1) birth events, (2) death (or extinction) events, (3) mutation events, and (4) sampling events. With suitable preprocessing, the runtime required to simulate a full tree T from a given BDMS model is (up to a constant) exactly $\#\mathrm{nodes}(T)$. We mean 'full' tree T, because the 'observed' or 'reconstructed' tree, denoted $r(T)$, is instead the sub-tree induced by the sampled leaves, and usually the desired output from the simulation. Since a full tree might have an empty reconstructed tree, the time required to produce a non-empty reconstructed tree from the original BDMS model via retrying is $\#\mathrm{nodes}(T_1) + \#\mathrm{nodes}(T_2) + \cdots + \#\mathrm{nodes}(T_K)$ where $T_1, T_2, \ldots$ is an infinite sequence of full trees simulated i.i.d. from the BDMS model and T_K is the first full tree with non-empty reconstruction (so that K is a stopping time random variable). As we have shown, given a BDMS model, there exists an equivalent BDMS model—in the sense of providing the same distribution over reconstructed trees conditional on non-emptyness—such that under the new BDMS model there is no death ($\mu_a^{\mathsf{FE}}(\tau) = 0$) and all leaves are sampled. In other words, the new model can simulate exactly from the original BDMS model *without having to perform any retries nor reconstruction*. The runtime of the forward-equivalent BDMS model (under the obvious coupling) is hence just $\#\mathrm{nodes}(r(T_K))$ rather than $\#\mathrm{nodes}(T_1) + \#\mathrm{nodes}(T_2) + \cdots + \#\mathrm{nodes}(T_K)$, which may be orders of magnitudes faster if $r(T_K)$ is significantly smaller than T_K or if a lot of retries are needed (i.e., if K is large). Let $\mathbb{P}, \mathbb{E}$ be probabilities and expectations under the original BDMS model, and let $\mathbb{P}^{\mathsf{FE}}, \mathbb{E}^{\mathsf{FE}}$ be probabilities and expectations under the forward-equivalent BDMS model. The expected runtime of the original BDMS model is:

$$R = \mathbb{E}[\#\mathrm{nodes}(T_1) + \#\mathrm{nodes}(T_2) + \cdots + \#\mathrm{nodes}(T_K)]$$

whereas the expected runtime of the forward-equivalent BDMS model is:

$$R^{\mathsf{FE}} = \mathbb{E}[\#\mathrm{nodes}(r(T_K))]$$

It is immediately clear that $R \geq R^{\mathsf{FE}}$, i.e., it is faster to simulate under the forward-equivalent model. Our Proposition 2 gives a more precise characterization of this speedup, in a way that is easy to compute with time discretization and dynamic programming. We now prove the Lemma.

Proof of Proposition 2. Under our setup, the runtime of the forward-equivalent model is:

$$R^{\mathsf{FE}} = \mathbb{E}[\#\mathrm{nodes}(r(T_K))] = \mathbb{E}[\#\mathrm{nodes}(r(T_1))|r(T_1) \neq T_\emptyset]$$

As for the runtime of the original BDMS model, let $S = \#\mathrm{nodes}(T_1) + \#\mathrm{nodes}(T_2) + \cdots + \#\mathrm{nodes}(T_K)$. By total expectation, we have that:

$$
\begin{aligned}
R =& \mathbb{E}[S] \\
=& \mathbb{P}(r(T_1) = T_\emptyset)\mathbb{E}[S|r(T_1) = T_\emptyset] \\
& + \mathbb{P}(r(T_1) \neq T_\emptyset)\mathbb{E}[S|r(T_1) \neq T_\emptyset] = \\
=& \mathbb{P}(r(T_1) = T_\emptyset)(\mathbb{E}[\#\mathrm{nodes}(T_1)|r(T_1) = T_\emptyset] + R) \\
& + \mathbb{P}(r(T_1) \neq T_\emptyset)\mathbb{E}[\#\mathrm{nodes}(T_1)|r(T_1) \neq T_\emptyset] \\
=& \mathbb{P}(r(T_1) = T_\emptyset)R + \mathbb{E}[\#\mathrm{nodes}(T_1)]
\end{aligned}
$$

so that solving for R we get:

$$R = \frac{\mathbb{E}[\#\mathrm{nodes}(T_1)]}{\mathbb{P}(r(T_1) \neq T_\emptyset)}$$

Another way to show this is by noting that the runtime of tree T_i should be considered if and only if all of $r(T_1), \ldots, r(T_{i-1})$ are empty. Crucially, for fixed i, whether we consider the runtime from tree T_i does not depend on whether T_i is empty or not. (Note that this is not true for T_K; the runtime of T_K is biased by the fact that it is, by definition, non-empty.) Hence,

$$R = \sum_{i=1}^{\infty} \mathbb{P}(r(T_1) = T_\emptyset)^{i-1}\mathbb{E}[\#\mathrm{nodes}(T_i)] = \frac{\mathbb{E}[\#\mathrm{nodes}(T_1)]}{\mathbb{P}(r(T_1) \neq T_\emptyset)}$$

Combining our identities for R and R^{FE} we obtain:

$$\frac{R}{R^{\mathsf{FE}}} = \frac{\mathbb{E}[\#\mathrm{nodes}(T_1)]}{\mathbb{P}(r(T_1) \neq T_\emptyset)\mathbb{E}[\#\mathrm{nodes}(r(T_1))|r(T_1) \neq T_\emptyset]}$$

Since $\mathbb{E}[\#\mathrm{nodes}(r(T_1))|r(T_1) = T_\emptyset] = 0$, we can complete the denominator using the law of total expectation to obtain:

$$\frac{R}{R^{\mathsf{FE}}} = \frac{\mathbb{E}[\#\mathrm{nodes}(T_1)]}{\mathbb{E}[\#\mathrm{nodes}(r(T_1))]}$$

as claimed. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

With this, it is easy to prove Proposition 1:

Proof of Proposition 1. We compute $\mathbb{E}[\#\mathrm{nodes}(T)]$ and $\mathbb{E}[\#\mathrm{nodes}(r(T))]$ and take the ratio, which by the lemma provides the speedup. Note that

$$\mathbb{E}[\#\mathrm{nodes}(r(T))] = 2\rho\mathbb{E}[\text{number of leaf nodes}]$$

and

$$\mathbb{E}[\#\mathrm{nodes}(T)] = 2\mathbb{E}[\text{number of leaf nodes}] + 2\mathbb{E}[\text{number of deaths}]$$

so it suffices to compute $\mathbb{E}[\text{number of leaf nodes}]$ and $\mathbb{E}[\text{number of deaths}]$, which we do separately. Each of these quantities is given by an ODE; let:

$$l(t) = \mathbb{E}[\text{number of leaf nodes}|\text{start with one individual at time } t]$$

and
$$d(t) = \mathbb{E}[\text{number of deaths}|\text{start with one individual at time } t]$$

then:

$$\dot{l}(t) = (\lambda - \mu)l(t); \quad l(0) = 1$$
$$\dot{d}(t) = \mu + (\lambda - \mu)d(t); \quad d(0) = 0$$

If $\lambda = \mu$ then the solutions are $l(t) = 1, d(t) = t\mu$. Otherwise:

$$l(t) = e^{(\lambda - \mu)t}$$
$$d(t) = \frac{\mu(e^{(\lambda - \mu)t} - 1)}{\lambda - \mu}$$

Evaluating at $t = 1$ provides the desired quantities from which we can derive the given ratio.

References

[1] P. Barrat-Charlaix, J. Huddleston, T. Bedford, and R. A. Neher. Limited Predictability of Amino Acid Substitutions in Seasonal Influenza Viruses. *Molecular Biology and Evolution*, 38(7):2767–2777, 03 2021.

[2] G. E. Budd and R. P. Mann. History is written by the victors: The effect of the push of the past on the fossil record. *Evolution*, 72(11):2276–2291, Sept. 2018.

[3] W. S. DeWitt, M. Celentano, T. Bakis, D. K. Ralph, S. Prillo, F. A. Matsen, IV, Y. S. Song, and V. M. Minin. BDMS: A tree simulator for birth-death-mutation-sampling processes, 2024. https://doi.org/10.5281/zenodo.10467425.

[4] R. G. FitzJohn, W. P. Maddison, and S. P. Otto. Estimating Trait-Dependent Speciation and Extinction Rates from Incompletely Resolved Phylogenies. *Systematic Biology*, 58(6):595–611, 10 2009.

[5] K. B. Hoehn and S. H. Kleinstein. B cell phylogenetics in the single cell era. *Trends Immunol.*, 0(0), Dec. 2023.

[6] S. Höhna, T. Stadler, F. Ronquist, and T. Britton. Inferring Speciation and Extinction Rates under Different Sampling Schemes. *Molecular Biology and Evolution*, 28(9):2577–2589, 04 2011.

[7] F. Horns, C. Vollmers, D. Croote, S. F. Mackey, G. E. Swan, C. L. Dekker, M. M. Davis, and S. R. Quake. Lineage tracing of human B cells reveals the in vivo landscape of human antibody class switching. *eLife*, 5:e16578, aug 2016.

[8] M. B. Horton, H. Cheon, K. R. Duffy, D. Brown, S. H. Naik, C. Alvarado, J. R. Groom, S. Heinzel, and P. D. Hodgkin. Lineage tracing reveals b cell antibody class switching is stochastic, cell-autonomous, and tuneable. *Immunity*, 55(10):1843–1855.e6, Oct 2022.

[9] J. Huerta-Cepas, F. Serra, and P. Bork. ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. *Molecular Biology and Evolution*, 33(6):1635–1638, 02 2016.

[10] M. Jones, A. Khodaverdian, J. Quinn, M. Chan, J. Hussmann, R. Wang, C. Xu, J. Weissman, and N. Yosef. Inference of single-cell phylogenies from lineage tracing data using cassiopeia. *Genome Biology*, 21(1), Apr. 2020. Publisher Copyright: © 2020 The Author(s).

[11] G. Laudanno, B. Haegeman, D. L. Rabosky, and R. S. Etienne. Detecting lineage-specific shifts in diversification: A proper likelihood approach. *Systematic Biology*, 70(2):389–407, 07 2020.

[12] B. Legried and J. Terhorst. A class of identifiable phylogenetic birth–death models. *Proceedings of the National Academy of Sciences*, 119(35):e2119513119, Aug. 2022.

[13] B. Legried and J. Terhorst. Identifiability and inference of phylogenetic birth–death models. *Journal of Theoretical Biology*, 568:111520, July 2023.

[14] S. Louca and M. W. Pennell. Extant timetrees are consistent with a myriad of diversification histories. *Nature*, 580(7804):502–505, 2020.

[15] A. MacPherson, S. Louca, A. McLaughlin, J. B. Joy, and M. W. Pennell. Unifying Phylogenetic Birth–Death Models in Epidemiology and Macroevolution. *Systematic Biology*, 71(1):172–189, 06 2021.

[16] W. P. Maddison, P. E. Midford, and S. P. Otto. Estimating a Binary Character's Effect on Speciation and Extinction. *Systematic Biology*, 56(5):701–710, 10 2007.

[17] B. R. Moore, S. Höhna, M. R. May, B. Rannala, and J. P. Huelsenbeck. Critically evaluating the theory and performance of bayesian analysis of macroevolutionary mixtures. *Proceedings of the National Academy of Sciences*, 113(34):9569–9574, 2016.

[18] H. Morlon. Phylogenetic approaches for studying diversification. *Ecology Letters*, 17(4):508–525, 2014.

[19] H. Morlon, T. L. Parsons, and J. B. Plotkin. Reconciling molecular phylogenies with the fossil record. *Proceedings of the National Academy of Sciences*, 108(39):16327–16332, 2011.

[20] S. Nee, R. M. May, and P. H. Harvey. The reconstructed evolutionary process. *Philosophical Transactions: Biological Sciences*, 344(1309):305–311, 1994.

[21] R. A. Neher and T. Bedford. nextflu: real-time tracking of seasonal influenza virus evolution in humans. *Bioinformatics*, 31(21):3546–3548, 06 2015.

[22] R. A. Neher and O. Hallatschek. Genealogies of rapidly adapting populations. *Proceedings of the National Academy of Sciences*, 110(2):437–442, 2013.

[23] R. A. Neher, C. A. Russell, and B. I. Shraiman. Predicting evolution from the shape of genealogical trees. *eLife*, 3:e03568, nov 2014.

[24] S. Prillo, A. Ravoor, N. Yosef, and Y. S. Song. ConvexML: Scalable and accurate inference of single-cell chronograms from CRISPR/Cas9 lineage tracing data, Dec. 2023.

[25] J. Quinn, M. Jones, R. Okimoto, S. Nanjo, M. Chan, N. Yosef, T. Bivona, and J. Weissman. Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts. *Science*, 371:eabc1944, 01 2021.

[26] J. J. Quinn, M. G. Jones, R. A. Okimoto, S. Nanjo, M. M. Chan, N. Yosef, T. G. Bivona, and J. S. Weissman. Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts. *Science*, 371(6532):eabc1944, 2021.

[27] M. A. Rolfes, I. M. Foppa, S. Garg, B. Flannery, L. Brammer, J. A. Singleton, E. Burns, D. Jernigan, S. J. Olsen, J. Bresee, and C. Reed. Annual estimates of the burden of seasonal influenza in the united states: A tool for strengthening influenza surveillance and preparedness. *Influenza Other Respir Viruses*, 12(1):132–137, Feb. 2018.

[28] K. P. Simeonov, C. N. Byrns, M. L. Clark, R. J. Norgard, B. Martin, B. Z. Stanger, J. Shendure, A. McKenna, and C. J. Lengner. Single-cell lineage tracing of metastatic cancer reveals selection of hybrid emt states. *Cancer Cell*, 39(8):1150–1162.e9, 2021.

[29] T. Stadler. Sampling-through-time in birth–death trees. *Journal of Theoretical Biology*, 267(3):396–404, 2010.

[30] T. Stadler, R. Kouyos, V. von Wyl, S. Yerly, J. Böni, P. Bürgisser, T. Klimkait, B. Joos, P. Rieder, D. Xie, H. F. Günthard, A. J. Drummond, S. Bonhoeffer, and the Swiss HIV Cohort Study. Estimating the Basic Reproductive Number from Viral Sequence Data. *Molecular Biology and Evolution*, 29(1):347–357, 09 2011.

[31] A. Thompson, B. Liebeskind, E. J. Scully, and M. Landis. Deep learning and likelihood approaches for viral phylogeography converge on the same answers whether the inference model is right or wrong. *bioRxiv*, 2023.

[32] P. O. Title and D. L. Rabosky. Tip rates, phylogenies and diversification: What are we estimating, and how good are the estimates? *Methods in Ecology and Evolution*, 10(6):821–834, 2019.

[33] S. Xie, L. Valente, and R. S. Etienne. Inferring state-dependent diversification rates using approximate bayesian computation (abc). *bioRxiv*, 2023.

[34] D. Yang, M. G. Jones, S. Naranjo, W. M. Rideout, K. H. J. Min, R. Ho, W. Wu, J. M. Replogle, J. L. Page, J. J. Quinn, F. Horns, X. Qiu, M. Z. Chen, W. A. Freed-Pastor, C. S. McGinnis, D. M. Patterson, Z. J. Gartner, E. D. Chow, T. G. Bivona, M. M. Chan, N. Yosef, T. Jacks, and J. S. Weissman. Lineage tracing reveals the phylodynamics, plasticity, and paths of tumor evolution. *Cell*, 185(11):1905–1923.e25, 2022.

[35] D. Yang, M. G. Jones, S. Naranjo, W. M. Rideout III, K. H. J. Min, R. Ho, W. Wu, J. M. Replogle, J. L. Page, J. J. Quinn, F. Horns, X. Qiu, M. Z. Chen, W. A. Freed-Pastor, C. S. McGinnis, D. M. Patterson, Z. J. Gartner, E. D. Chow, T. G. Bivona, M. M. Chan, N. Yosef, T. Jacks, and J. S. Weissman. Lineage tracing reveals the phylodynamics, plasticity, and paths of tumor evolution. *Cell*, 185(11):1905–1923.e25, May 2022.
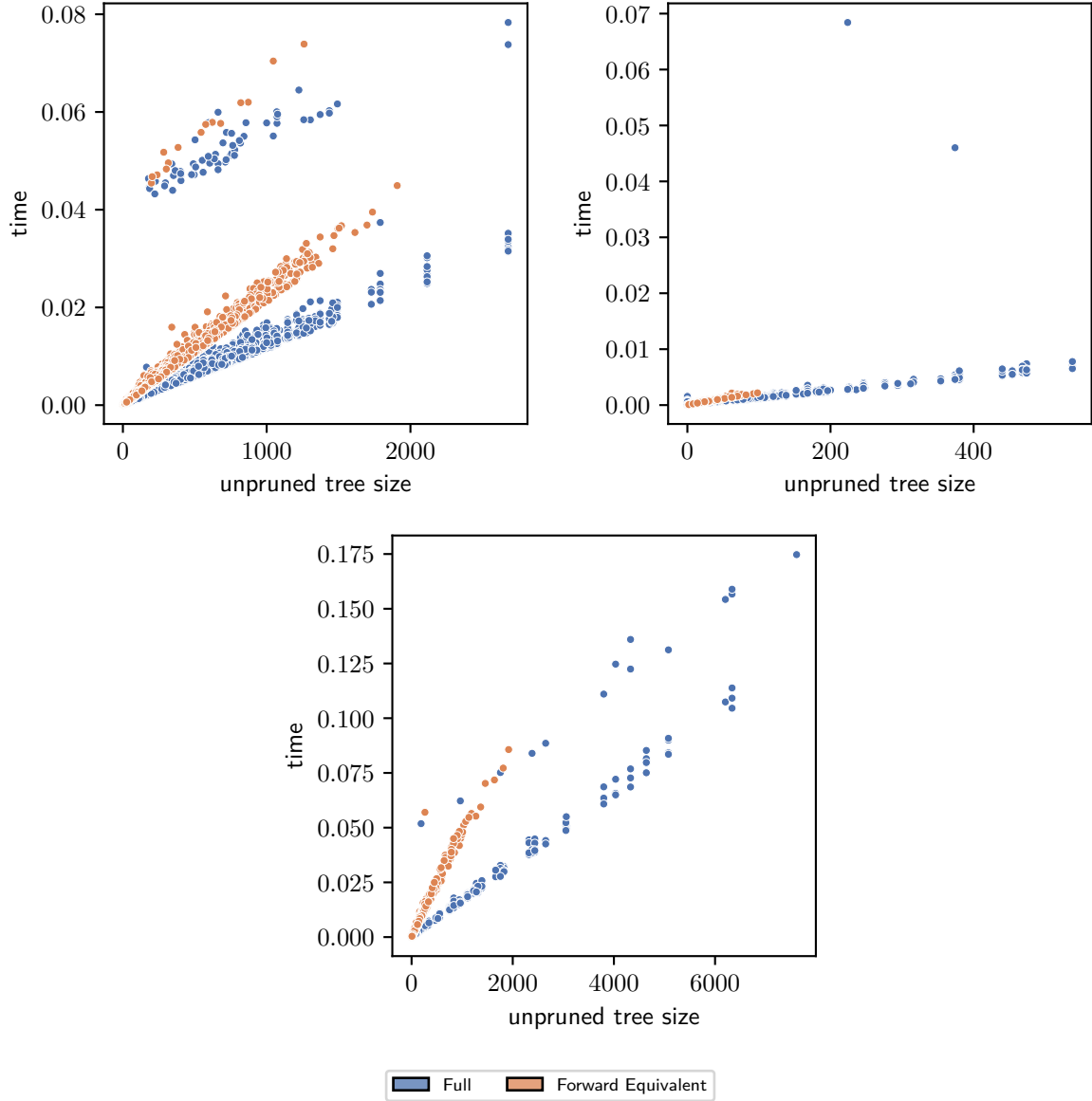
Supplementary Figures

Figure S1: Computation time required for each individual tree against the unpruned tree size across various simulation settings. We see that, in correspondence with the discussion in Appendix A and Section 3.4, the computation time is approximately linear in the size of the unpruned tree. The forward-equivalent model is slower than the full simulation after controlling for the size of the unpruned tree due to the computational overhead of simulating from a BDMS model with time-varying rates. The greater efficiency of the forward-equivalent simulation comes from simulating trees with substantially smaller unpruned tree size. Top left: single-type model without death. Top right: single-type model with death. Bottom: multi-type model. Each point is a single tree.
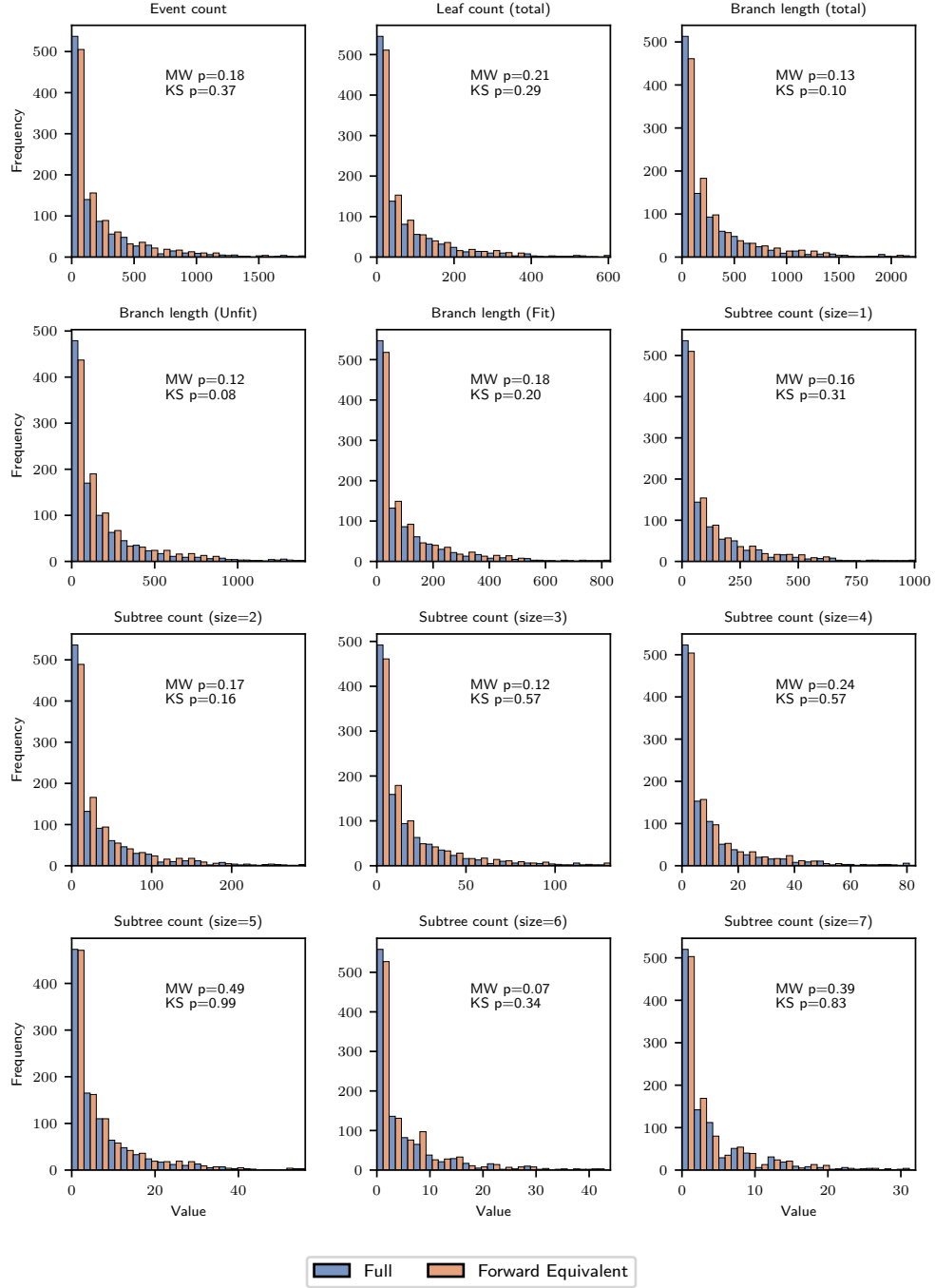
Figure S2: Distributional comparisons analogous to those appearing in Figure 3, for all summary statistics we considered, as described in Section 4.1. We emphasize that because all of these tests were computed using the same sample of phylogenies, the *p*-values for different tests need not be independent.
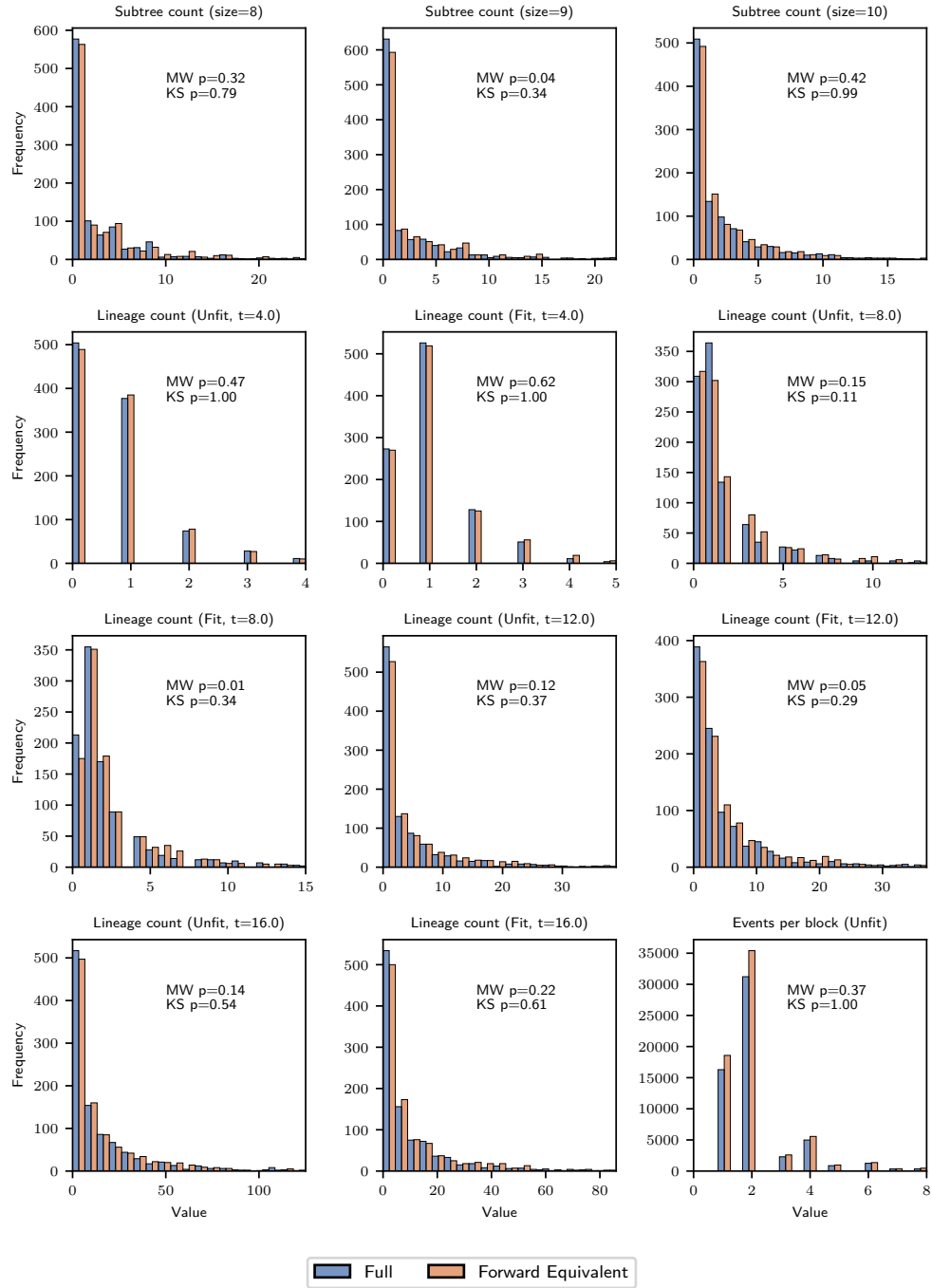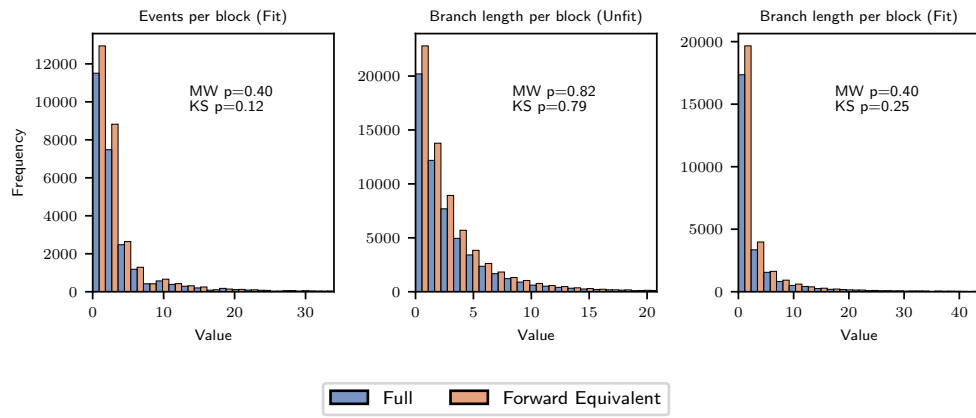
Figure S2: (Continued)

Figure S2: (Continued)