VINE: Variational inference for scalable Bayesian reconstruction of species and cell-lineage phylogenies

Adam Siepel[1,*], Rebecca Hassett[1], and Stephen J. Staklinski[1]

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY
[*]Corresponding author: asiepel@cshl.edu

Abstract

Bayesian methods are now widely used in reconstructing both species and cell-lineage phylogenies, but they remain heavily reliant on computationally intensive Markov chain Monte Carlo sampling. Phylogenetic variational inference (VI) circumvents this dependency but so far has been limited in speed and scalability. Here we introduce Variational Inference with Node Embeddings (VINE), a computational method that combines an embedding of taxa in a high-dimensional space and a distance-based "decoder" with several algorithmic innovations to dramatically improve phylogenetic VI. VINE supports both standard DNA substitution models and CRISPR barcode-mutation models for inference of cell-lineage trees and tissue-migration histories. In extensive simulation experiments, we show that VINE is comparable in accuracy to the best available Bayesian methods with speeds orders of magnitude faster. We then apply VINE to ∼1,000 complete SARS-CoV-2 genomes and ∼900 lung-cancer cell barcodes, showing reductions in compute time from days to hours or minutes.

Introduction

Phylogenetic trees are now ubiquitous across the life sciences, with applications ranging from large-scale comparative and population genomics [1, 2] to charting and combating the spread of infectious diseases [3]. A particularly exciting new frontier for phylogenetics is the reconstruction of cell-lineage trees from CRISPR-based lineage-tracing data [4–7], which can shed important light on tumor evolution [8, 9], developmental biology [10–12], and neurobiology [13, 14]. Datasets for species and cell-lineage trees alike are rapidly growing in size and now often include thousands of taxa, leading to steadily increasing demand for fast and accurate phylogenetic methods.

For many applications, Bayesian inference has become the preferred approach for phylogenetic reconstruction. Bayesian methods naturally quantify the uncertainty about the phylogenetic tree given the data, by inferring a full posterior distribution for the tree topology, the branch lengths, and the parameters governing the substitution process. Since their introduction in the mid 1990s [15–17], these methods have steadily improved, and mature implementations such as MrBayes [18, 19], BEAST/BEAST 2 [20–22],

and RevBayes [23] are now extraordinarily widely used. Bayesian approaches are particularly valuable in CRISPR-based lineage tracing, where current datasets often contain limited phylogenetic information, leading to considerable uncertainty in tree reconstruction [12, 24–27].

Despite many recent advances [28], however, Bayesian phylogeny inference methods still lag considerably behind the leading maximum-likelihood tools [29–31] in speed and scalability. They are primarily limited by a reliance on Markov chain Monte Carlo (MCMC) sampling, which explores the full space of tree topologies through a series of discrete rearrangement operations, and inevitably stalls as the number of taxa becomes large. In addition, MCMC-based methods often require tuning of proposal distributions and careful monitoring of convergence, sometimes making them challenging to use for non-experts. Bayesian methods for cell-lineage reconstruction also depend on MCMC [24–27] and therefore face the same bottleneck.

Recognizing the limitations of MCMC, investigators have recently explored a variety of alternative methods for characterizing phylogenetic posterior distributions. Most of these efforts have made use of some form of variational inference (VI), which has been widely employed for approximate inference in other fields [32, 33]. The core idea of VI is, instead of sampling from a high-dimensional posterior distribution, to compel a flexible alternative distribution to approximate the true posterior by minimizing the divergence between distributions (reviewed in [34]). In phylogenetics, the first method of this kind—Variational Bayesian Phylogenetic Inference (VBPI)—was proposed seven years ago by Zhang and Matsen [35] (see also [36, 37]) and subsequently extended to make use of normalizing flows and graph neural networks [38–41], among other features. VBPI has now been followed by several other phylogenetic VI methods including VaiPhy [42], VBPI-Mixtures [43], GeoPhy [44], and Dodonaphy [45] (see also [46–48]). In general, however, research in this area is still at the exploratory stage, and the available methods have yet to achieve the scalability and accuracy needed for applied phylogenetics. To our knowledge, VI has yet to be used in any published phylogenetic analysis beyond benchmarking for methods development.

In this article, we revisit the problem of VI for phylogenetics, with an eye toward applications to both DNA alignments and CRISPR-based lineage-tracing data. We begin with a recently proposed idea [44,45] to embed taxa in a continuous space, decode phylogenies by standard distance-based reconstruction methods, and optimize model parameters by stochastic gradient ascent (SGA). We redesign this method from the ground up, introducing numerous simplifications, extensions, and algorithmic innovations that both improve its accuracy and boost its speed by several orders of magnitude. Our methods are implemented in a freely available computer program, called VINE (Variational Inference with Node Embeddings), that supports a variety of nucleotide substitution models, a recently developed model for CRISPR-barcode editing, and an extension to tissue migration-graph inference, among other features. We show that VINE is competitive in accuracy with the best available Bayesian phylogenetic methods for both species and cell-lineage phylogeny inference, but it is considerably faster. For the first time, we demonstrate that variational phylogenetic inference can out-perform mature MCMC-based implementations such as BEAST 2 and MrBayes, enabling Bayesian phylogenetic inference for datasets beyond the reach of current methods.

Results

Variational inference using continuous embeddings

The central problem in phylogenic VI is to approximate the Bayesian posterior distribution of phylogenetic trees given a set of observed genotypes $\mathbf{X}$ using a flexible variational distribution $q(\tau, \mathbf{b}; \boldsymbol{\theta})$, where τ denotes the *topology* of a phylogenetic tree, $\mathbf{b}$ is a corresponding vector of *branch lengths*, and $\boldsymbol{\theta}$ denotes the free parameters of the variational distribution. Our particular starting point is a method recently introduced by Mimori and Hamada [44] (see also [45]) that has three essential components. First, the tips of the tree (corresponding to the observed data) are represented by an embedding $\mathbf{x}$ in a d-dimensional continuous space, with an induced matrix of pairwise distances $\mathbf{D}$. A simple, flexible multivariate Gaussian sampling distribution, $q(\mathbf{x}) = \mathrm{MVN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, is assumed for the embedded taxa. Second, classical distance-based methods for tree reconstruction, such as the neighbor-joining method [49], are leveraged to convert the distance matrix $\mathbf{D}$ to a tree with branch lengths $(\tau, \mathbf{b})$. In this way, sampled embeddings $\mathbf{x}$ can be deterministically converted to fully defined phylogenies, permitting calculation of the likelihood of the data by standard methods. Third, the free parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of the variational distribution q are optimized by minimizing the Kullback-Leibler (KL) divergence from the true posterior by stochastic gradient ascent (SGA). As the authors noted, the resulting model can be thought of as a variational autoencoder [50, 51], where the decoder takes the form of the deterministic mapping from embedded taxa to trees.

Despite several appealing properties of this general modeling strategy, initial implementations have not been competitive with well-established MCMC-based methods for Bayesian phylogenetics. We therefore sought to devise new algorithms that would substantially improve performance while maintaining the core ideas of a continuous embedding and distance-based phylogeny decoder. Briefly, our new computational method, VINE, encodes tree topologies and branch lengths together, in one unified continuous embedding, rather than separately; operates in substantially higher-dimensional spaces ($d = 5$ or higher) than previous methods (typically $d = 2$ or $d = 3$); makes use of a new algorithm for efficient backpropagation of gradients through distance-based phylogeny reconstruction algorithms; and permits calculation of the objective function for VI using a fast Taylor approximation instead of the original Monte Carlo approach (**Fig. 1**; full details in **Methods**). We also introduce optional normalizing flows to accommodate nonlinearities in the approximate posterior distribution as well as richer parameterizations of the variational covariance matrix. With these innovations, we obtain large improvements in efficiency without compromising accuracy in Bayesian phylogenetic inference, allowing applications to datasets with 1,000 taxa or more. Finally, we extend VINE to support not only a variety of richly parameterized DNA substitution models but also mutation models for CRISPR-based barcode editing, making it the first phylogenetic VI method to apply to both species and cell-lineage phylogeny reconstruction problems. As we will show, an additional extension allows VINE to be used for inference of tissue-migration graphs based on CRISPR-barcode data (see [27]).

VINE is comparable to MCMC in model fit and offers substantially improved speed

We first evaluated VINE's performance against simulated DNA sequence data, where the true evolutionary history is known and model misspecification can be controlled. We simulated phylogenies under a birth-death model, followed by subsampling and rescaling, to achieve branching patterns and overall tree scales roughly similar to those of real phylogenies for mammals, and then simulated DNA alignments by allowing nucleotides to evolve along these trees under the HKY substitution model [52] (see **Methods**). We generated trees for $n = 10$ to $n = 1000$ taxa, with ten replicates each, and we produced both short alignments of $L = 300$ bp and longer ones of $L = 10,000$ bp, with the shorter alignments designed to permit faster phylogenetic reconstruction but with more statistical uncertainty.

In our benchmarking experiments, we compared VINE to three recently developed VI methods—VaiPhy [42], GeoPhy [44] and Dodonaphy [45]—as well as to two of the leading MCMC-based Bayesian phylogenetic inference packages, MrBayes [19] and BEAST 2 [21, 22]. Notably, GeoPhy and Dodonaphy are the two previous methods to use a similar embedding scheme to the one we adopted, although unlike VINE, both rely heavily on hyperbolic geometries (see **Discussion**). We also tried to include VBPI-GNN [39] in our experiments but found that it was not sufficiently fast for practical consideration. Because the previously published VI methods all assume the simple Jukes-Cantor (JC) substitution model [53], we started by assuming that model for inference with all methods. As a baseline, we also reconstructed trees using a straightforward neighbor-joining implementation with no further optimization.

Across all simulated datasets, we found that VINE was able to obtain reconstructed phylogenetic models that fit the data well. As would be expected for a likelihood-based method, its maximized log likelihoods were reliably higher than the log likelihoods of the true (generating) models at all dataset sizes (**Fig. 2A**). Inspection of individual trees showed that the reconstructions were generally close to the ground truth, with occasional errors that tended to correspond to difficult-to-resolve features of trees (**Fig. 2C&D**). VINE's performance by these measures was very close to that of MrBayes and BEAST 2 (**Fig. 2A**). The previous VI methods had slightly lower log likelihoods, with GeoPhy and VaiPhy obtaining values close to those of the true models, and Dodonaphy showing somewhat poorer performance (**Fig. 2A**).

Because the main goal of VI is speed, we also kept track of the CPU time required for each experiment. Comparisons of running times are complicated, however, by questions of how long to sample with MCMC-based methods as well as differences in multithreading schemes and GPU utilization. To keep our comparisons as straightforward as possible, we ran all methods using a single CPU core and we used measures based on split chains and effective sample sizes to calibrate the number of MCMC iterations in a dataset-dependent manner (see **Methods**). Nevertheless, we found quite stark differences in running times across methods. The previous VI methods, in particular, were highly compute-intensive, requiring from many minutes to several hours for even small alignments. The fastest of these methods, VaiPhy, took more than five minutes per replicate for our smallest (10-taxon, 300-site) simulated alignments, and seventeen minutes per replicate for similar alignments with 20-taxa. By contrast, in all cases with $n \leq 20$ taxa and 300 sites, VINE obtained good-quality phylogenies in less than two seconds per replicate, showing speedups of roughly 500-fold relative to VaiPhy and as much as $\sim$20,000-fold relative to Dodonaphy.

The MCMC-based methods were much faster than the experimental VI methods but not as fast as VINE. MrBayes required between about 20 seconds per replicate for the 10-taxon trees and 27 seconds per replicate for the 20-taxon trees. In these experiments, BEAST 2 was slower than MrBayes by about an order of magnitude, largely owing to slower convergence by our criteria (see **Methods**). Even relative to MrBayes, however, VINE was able to obtain comparable reconstructions with speeds 15–30 times greater. Because running times began to reach many hours per replicate for some of the VI methods, we did not extend these experiments beyond 20 taxa.

To consider datasets of more realistic size, we ran VINE and the two MCMC-based methods on the simulated alignments with up to 1000 taxa. In this case, we assumed the more realistic HKY model for inference, and we evaluated the quality of all reconstructed phylogenies by both likelihood-based measures of model fit and measures of discordance from the true (simulated) trees. We found, again, that the maximized log likelihoods were similar across methods and generally better than those of the true model (**Fig. 2E**, **Supplementary Fig. S1A**). To control for overfitting in model estimation, we also evaluated the average log likelihoods of sampled trees on held-out sequence alignments generated by the same models, and found—again as expected—that both VINE and the MCMC-based methods performed only slightly worse than the true models, with deviations of $\sim$1% (**Fig. 2F**, **Supplementary Fig. S1B**). By the Robinson-Foulds measure of topological discordance [54], VINE and the MCMC-based methods also performed fairly similarly, although VINE did show some increased discordance from the true trees at larger values of n (**Supplementary Fig. S2**). This difference appears to be a consequence of its reduced posterior variance, as discussed further below; VINE tends to converge on a small set of tree topologies and therefore is less effective at "hedging its bets" relative to the truth in measures of topological discordance.

The trend with running times remained similar to what we observed with smaller alignments, with VINE showing speedups of about 30–80 times relative to MrBayes, which, in turn, improved on BEAST 2 by about another 3–8 times (**Fig. 2G**, **Supplementary Fig. S3**). VINE was again able to obtain trees for 10 taxa in less than a second per replicate, on average, increasing to about 7 seconds for 50 taxa and 19 seconds for 100 taxa. It required about 15 minutes per replicate for 500 taxa and 70 minutes for 1000 taxa. By comparison, MrBayes exhibited running times of 20 seconds for 10 taxa up to 9.6 hours for 1000 taxa (**Fig. 2G**). With longer alignments, interestingly, the difference between MrBayes and BEAST 2 was less pronounced, but VINE still clearly improved on both of them (**Supplementary Fig. S3**). Overall, these experiments demonstrate that VINE is the first variational phylogenetic inference method to offer both comparable model-fitting performance to MCMC-based methods and significant improvements in speed (see **Discussion**).

Performance improves with the dimensionality of the embedding space

We noticed that model fit seemed to be quite sensitive to the dimensionality d of the embedding space, more than to the use of a hyperbolic geometry rather than a Euclidean one. We therefore carried out a series of experiments where we evaluated the impact of the dimensionality d on both goodness of fit and running time, again using simulated data. We focused on datasets with $n = 25$ and $n = 50$ taxa and considered

values of d between 2 and 8 under both the Euclidean and hyperbolic embedding schemes.

We found, under the Euclidean embedding, that the maximized log likelihood was poor with $d = 2$ and $d = 3$ (e.g., $\sim$200 units lower than the value reported by BEAST for $n = 25$ and $\sim$800 units lower for $n = 50$ taxa; see **Fig. 3A**, **Supplementary Fig. S4**). As d increased, however, the log likelihood rapidly improved from $d = 2$ to $d = 3$, tapering off at about $d = 4$. We expected that this improvement in model fit would come with a time penalty, since the number of free parameters in the model is roughly equal to nd (see **Methods**). Interestingly, however, we found that the running time slightly *decreased* with d rather than increasing, owing to more efficient convergence of the optimization algorithm. For example, with $n = 25$ taxa, the average time to convergence decreased from about 1.7 seconds for $d = 2$ to about 1.2 seconds at $d = 4$ and then reached a minimum of 1.0 seconds by $d = 6$. Similarly, with $n = 50$ taxa, the time to convergence was cut nearly in half from $d = 2$ to $d = 6$.

Under the hyperbolic embedding scheme, the log likelihood was similarly sensitive to the embedding dimension, but the running time behaved in a less predictable manner (**Supplementary Fig. S4**). In addition, the running time was generally greater, and with higher variance across replicates, owing to delays in convergence of the optimizer.

Overall we found that, once SGA is suitably tuned (see **Methods**), it is capable of optimizing hundreds of parameters in a highly efficient manner, and there seems to be little cost, and substantial benefit, to operating at values of $d = 8$ or higher. At the same time, we found no reason to prefer the more complex hyperbolic geometry over a simple Euclidean scheme for embedding. At least in our hands, any advantages in flexibility from the hyperbolic geometry are offset by difficulties in fitting the model (see **Discussion**).

Capturing the full posterior variance remains challenging

We observed in our experiments with simulated data that, while VINE achieved high likelihoods, the variance of its approximate posterior distributions tended to collapse during optimization. For example, under our initial version of the model (CONST), which assumed a simple diagonal covariance structure with a single free parameter (with $\mathbf{\Sigma} = e^{\eta}\mathbf{I}$), the η parameter was driven toward strongly negative values and approached the floor we had set for it (such that $e^{\eta} = 1 \times 10^{-3}$; see **Supplementary Fig. S5**).

We therefore experimented with various alternative parameterizations of the covariance matrix, with the goal of capturing more of the structure of the true posterior. We introduced a fully parameterized diagonal covariance matrix (DIAG), a covariance matrix proportional to a double-centered version of the initial distance matrix (DIST), and a general low-rank parameterization of the covariance matrix (LOWR) (see **Methods** for details). We found even under these richer parameterizations, however, that the covariance still tended to collapse. This behavior is known to occur in VI when the approximate posterior distribution has insufficient flexibility to capture the structure of the true posterior (see **Discussion**).

To counter this problem, we introduced two additional extensions to our model. First, we regularized the variance in a manner dependent on the choice of parameterization. For example, in the CONST and

DIST parameterizations, we applied an ℓ_2 penalty to the free parameter η, pushing it toward zero (see **Supplementary Fig. S5**); and in the DIAG parameterization, we applied a similar ℓ_2 penalty to all free log-variance terms (see **Methods**). Second, as outlined above, we introduced optional normalizing flows to accommodate nonlinearities between the MVN-distributed points $\mathbf{x}$ and the embedding $\mathbf{y} = f_{\text{NF}}(\mathbf{x})$ from which the distance matrix $\mathbf{D}$ is computed. We allowed for two types of normalizing flows: a *radial flow*, which allows for radial contraction or expansion of points around a designated center (itself a free parameter estimated from the data); and a *planar flow*, which moves points based on their location relative to a hyperplane that is estimated from the data (see **Methods** for complete details). Together with our four parameterizations, these two strategies gave us a variety of means for tuning the representation of the approximate posterior to better reflect the true distribution.

We evaluated the effectiveness of these strategies by carrying out experiments with simulated DNA alignments of various sizes, across a grid of combinations of our four parameterizations, strengths of regularization penalties, the two normalizing flows, and with the Euclidean vs. the hyperbolic geometries. In each case, we allowed VINE and BEAST 2 to estimate approximate posterior distributions from the same ten simulated alignments, and we considered not only the quality of the model fit, but also the variance of the posterior. In assessing these posterior distributions, we focused on two measures: (1) the fraction of 95% credible intervals (CIs) for all $\binom{n}{2}$ pairwise distances between taxa along the reconstructed trees that contained the true value used in simulation; and (2) the *topological entropy* of the posterior distribution, defined as the Shannon entropy of the distinct tree topologies sampled (see **Methods**).

We found, indeed, that the posterior variance was poorly characterized with our simplest models (CONST parameterization, no regularization, no normalizing flows, and Euclidean geometry). In this case, while the likelihoods were excellent (as shown in **Fig. 2**), the 95% CIs were quite narrow and fewer than 20% of true values were contained within them (**Fig. 3B**). By contrast, BEAST 2 performed exceptionally well by this measure, with 92–97% inclusion of true values across all values of n. Accordingly, the topological entropy was considerably lower (sometimes by 50% or more) under VINE's approximate posterior distribution than under the MCMC-based distribution from BEAM 2 (**Fig. 3C**). The extensions of our baseline model did substantially improve VINE's ability to capture the posterior variance by both measures. The best model used the DIST parameterization, a moderate variance regularization, both normalizing flows, and the Euclidean geometry. In this case, the 95% CI inclusion reached about 35–55% and the topological entropy was on the order of that observed with BEAST 2, although in some cases it appeared to be somewhat too high (**Fig. 3B&C**). Thus, it appears to be possible to adapt our VI methods to capture more of the true posterior variance but even our best models still fall short of MCMC-based methods in this respect (see **Discussion**).

VINE shows top performance in cell-lineage phylogeny inference

As noted, a major area of interest in phylogeny inference today is the reconstruction of cell lineage trees from CRISPR-based barcoding data. This problem is closely related to standard phylogeny inference, but certain unusual features of the CRISPR-based editing process—which tends to produce insertions and deletions (indels) rather than point mutations, and which can be "silenced" in a site-specific manner if target sites

are disrupted—have led to the development of customized mutation models [24, 31, 55–57]. It has been shown that incorporation of these models into likelihood-based phylogenetic inference methods can lead to substantial improvements in reconstructed trees [31].

We therefore extended VINE to support the mutation model recently used in the LAML (Lineage Analysis via Maximum Likelihood) program [31] (see also [24])—a continuous-time Markov model that captures the irreversibility of CRISPR-mediated indels, site-specific mutation rates, barcode silencing, and missing data (available via the `-i CRISPR` option in VINE; see **Methods**). We ensured that VINE produces time-resolved ultrametric trees (with all tips equidistant from the root) similar to those inferred by LAML, by using the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) algorithm [58] in place of neighbor-joining in CRISPR mode (see **Methods**). We then benchmarked VINE against LAML on simulated CRISPR barcode mutation matrices. We also evaluated our recently developed MCMC-based method BEAM (Bayesian Evolutionary Analysis of Metastasis), which is primarily designed for inference of tissue migration graphs in metastasis but can also be used for cell lineage-tree reconstruction [27]. Several other tree-reconstruction programs are available for this problem, including Cassiopeia [55], TiDeTree [24], and Startle [59], but LAML and BEAM appear to be the best-performing methods at present [27, 31], so we focused on them for our comparisons.

As with our DNA alignments, we began by simulating trees and CRISPR barcode mutation matrices, taking advantage of the barcode-editing simulation tools in the Cassiopeia package and approximately matching mutation patterns observed in real data (see **Methods**). We assumed a barcode cassette with ten arrays of three target sites for a total of 30 editing sites per cell, and we generated 10 simulation replicates for trees with 10, 25, 50, 100, 250, 500, and 1000 taxa. We then applied VINE, LAML, and BEAM to each simulated alignment and recorded measures of model fit and tree discordance. Before comparing the reported likelihoods, we verified that the three programs returned exactly the same values (up to numerical precision) for the same mutation matrices and trees.

We found that VINE was able to obtain trees of similar log likelihood to those reported by LAML and BEAM across all simulated datasets (**Fig. 4A** and **Supplementary Fig. S6**). The maximized log likelihoods differed by less than 1% on average at small n and were nearly identical (within $\sim$0.1%) at $n = 1000$. These average differences were small relative to the standard deviation across replicates and were not statistically significant. The topological distances of the inferred trees from the true ones were also similar under all three methods, although, as noted for the DNA simulations, VINE did show slightly elevated Robinson-Foulds distances at large n, possibly owing to its reduced posterior variance (**Supplementary Fig. S7**).

As with the DNA models, VINE was able to achieve these high levels of accuracy at considerably greater speed than the other methods (**Fig. 4B**). Running times per replicate ranged from a fraction of a second for $n = 10$ to about 14 minutes for $n = 1000$. VINE was roughly 50–100 times faster than LAML for $n \leq 250$ taxa (**Fig. 4C**). As the number of taxa increased, this speed advantage declined somewhat but VINE was still more than 35 times faster than LAML for $n \geq 500$. Interestingly, the MCMC-based BEAM method was also considerably faster than LAML in these benchmarks, particularly for smaller numbers of taxa (e.g., by 15x for $n = 25$ and 9x for $n = 50$). Nevertheless, VINE still improved on BEAM by roughly an order of magnitude or more across benchmarks. Notably, the other likelihood-based methods available for this

problem, such as TiDeTree [24], are considerably slower than LAML, so to our knowledge, VINE is now the fastest such method available. Overall, VINE is able to fit CRISPR barcoding data at least as well as competing methods, at considerably greater speed, and (like BEAM) it does so by approximating a full posterior distribution rather than a single point estimate—a feature that can be particularly important in cell-lineage reconstruction, where there is often a great deal of uncertainty in tree inference (see **Discussion**).

Applications to real DNA data

To demonstrate the applicability of VINE to real DNA data, we first applied it to a collection of eight alignments originally assembled by Lakner et al. [60], which contain various types of nucleic-acid sequence data (DNA, rRNA, rDNA, and mtDNA) for between 27 and 59 taxa (median 42) and range in length from 378 to 2520 (median 1736) sites (**Supplementary Table S1**). These alignments are typical of modest-sized datasets frequently analyzed in applied phylogenetics and have been widely used in recent benchmarking studies (e.g., [44, 45, 61]). We again compared VINE with BEAST 2 and MrBayes using the HKY substitution model, omitting the other VI methods owing to their long running times. As expected from our simulation experiments, all methods were comparable in model fit, with VINE achieving maximized log likelihoods within 0.4% of those of BEAST 2 and MrBayes on average (**Supplementary Fig. S8A**). On visual inspection, the reconstructed trees were generally similar under all reconstruction methods, with some minor differences at difficult-to-resolve branchings (**Supplementary Fig. S9**). As with the simulated data, however, the running times for VINE were substantially reduced, by average factors of 5.5 relative to MrBayes and 10.3 relative to BEAST 2 (**Supplementary Fig. S8B**). Running times for VINE ranged from 8–60 seconds, in comparison to 1–4 minutes for MrBayes and 2–7 minutes for BEAST 2.

We then examined a larger data set more representative of modern applications in Bayesian phylogenetics. Inspired by recent widespread interest in the use of phylogenetic methods to study the SARS-CoV-2 pandemic [3], we obtained the latest SARS-CoV-2 whole genome sequences from Nextstrain [62], consisting of ∼74,000 genomes after data-quality filtering, and extracted two subsets by stratified random sampling: a large subset of 1060 genomes and a smaller subset of 364 genomes, each comprising ∼30k nucleotide sites (see **Methods**). We first ran VINE and BEAST 2 on the 364-taxon subset using the general time reversible (GTR) DNA substitution model and the discrete gamma model for rate variation among sites. The two methods produced broadly similar trees (**Fig. 5A**) in which SARS-CoV-2 genomes clearly grouped by collection time, reflecting their evolution as the pandemic progressed. These two trees exhibited highly correlated pairwise distances (**Fig. 5B**) and only minor differences in overall branching patterns (**Fig. 5C**). Notably, however, BEAST 2 required over 22 hours for this data set, whereas VINE finished in about 30 minutes (in this case, both programs were permitted eight threads on our server). We did observe somewhat higher posterior variance from BEAST 2 than from VINE (**Fig. 5A**), consistent with our findings from simulated data.

Next we applied both methods to the larger 1030-taxon data set. VINE was able to complete this analysis in about five hours, producing a tree with a similar overall structure to the smaller one but with about three times as many tips (**Fig. 5D**). We inspected the high-dimensional embedding learned by VINE and found,

interestingly, that the overall correlation structure of these SARS-CoV-2 genomes was directly apparent from its first two principal components (**Fig. 5E**). By contrast, BEAST 2 struggled to converge on this larger data set, and we terminated the run after about three days of processing. Overall, this example demonstrates that VINE is capable of carrying out state-of-the-art Bayesian phylogenetic analyses of large modern datasets, with similar results to MCMC-based methods but with considerably better scaling properties.

Applications to real CRISPR data

To demonstrate applicability to real data for cell-lineage tree reconstruction, we focused on a dataset based on a lung-cancer xenograft mouse model that contains 83 clonal populations (CPs) ranging in size from >11,000 to ∼30 cells [63]. Larger datasets now exist, but this one includes particularly rich mutational data for a broad range of CPs and it has been widely analyzed. For our purposes, we excluded CPs 1–3, which are prohibitively large (>5000 cells), leaving 80 CPs that ranged in size (after removing duplicates) from 26 to 899 cells (median 63 cells). As above, we compared VINE with LAML on these data, assuming a uniform prior distribution for mutation rates and allowing for a free silencing-rate parameter (see **Methods**).

On this dataset, VINE obtained somewhat higher log likelihood values than LAML on average (by 21.2%), with some variability across clonal populations (**Supplementary Fig. S10A**). As with the simulated data, VINE and LAML behaved similarly for the smaller trees but VINE often significantly outperformed LAML on the larger ones. Nevertheless, VINE was faster than LAML by orders of magnitude, with an average speed-up of 406-fold (**Supplementary Fig. S10B**). The average running time for these CPs was under two minutes for VINE in comparison to 12.7 hours for LAML. The largest clone took over five days with LAML and only 28 minutes with VINE.

Beyond inference of cell-lineage trees, phylogenetic methods have recently been adapted to reconstruct the spread of cancer cells across tissues [64–67] to reveal the rates, routes, and molecular changes associated with metastasis [63, 68–70]. Our recent method BEAM [27] is the first to simultaneously reconstruct both a cell-lineage phylogeny and a tissue-migration graph in a fully Bayesian manner. BEAM models the barcode mutation and tissue migration processes using conditionally independent continuous-time Markov chains, and samples from the joint posterior distribution of lineage trees and tissue labels. The method shows excellent performance in migration-graph reconstruction but requires MCMC for inference (using BEAST 2), and is limited in scalability to a few hundred cells. To address these limits in scalability, we extended VINE to accept tissue labels for cells and support BEAM's tissue-migration model during inference. This extension required only changes to the likelihood and gradient calculations, as well as support for estimation of migration rates as nuisance parameters in SGA (see **Methods**).

To validate this approach, we benchmarked VINE in migration mode against BEAM, Metient [66], and MACH2 [67], using our recently described simulation framework [27]. We found, indeed, that VINE's migration model produced reconstructions of simulated migration graphs more accurate than those from Metient and MACH2, and nearly as accurate as those from BEAM (**Fig. 6A**, **Supplementary Fig. S11A**), but with speeds orders of magnitude faster (**Fig. 6B**, **Supplementary Fig. S11B**). The decrease in accuracy

relative to BEAM appears to be at least partly driven by over-fitting of migration rates, which might be improved by regularization (see **Discussion**).

We then applied VINE in migration mode to the lung-cancer xenograft data from ref. [63], and found that it was able to reconstruct tissue-labeled trees similar to those from BEAM and generally better than those from other methods. Like BEAM, VINE tends to produce cell-lineage trees that group together cells having the same tissue label, resulting in more parsimonious migration histories than Cassiopeia-Greedy [55] or LAML [31], which do not have access to tissue labels (**Fig. 6C**). These improvements are evident in the numbers of both mutations and migrations required to explain the observed data, as well as the tissue homogeneity index, a measure of clustering by tissue label (**Fig. 6C**, **Supplementary Figs. S12&S13**; see **Methods**). Interestingly, in comparison to BEAM, VINE sometimes appears to trade additional mutations for fewer migrations, possibly owing to its more aggressive optimization of migration rates. Nevertheless, most differences between VINE and BEAM trees appear in the detailed branching patterns within single-tissue clades, for which the signal in the data is weak.

The improved efficiency of VINE allowed us to analyze some of the largest CPs from ref. [63], including ones prohibitively large for BEAM or LAML. For example, VINE took only about 25 minutes to produce a tissue-labeled tree and migration graph for CP 4, which comprises 904 distinct barcode/tissue combinations (**Fig. 6D&E**), whereas the LAML tree inference step alone for this CP (not including migration inference) required several days. For comparison with VINE, we followed ref. [66] in building a cell-lineage tree for this clone using the heuristic Cassiopeia-Greedy method, and then reconstructing migration graphs using MACH2 and Metient (which took 44 min and $\sim$9 hrs, respectively, with multithreading). We found that VINE obtained considerably simpler tissue-migration graphs than the other methods, requiring $\sim$40% fewer migrations to explain the observed data (**Fig. 6E–G**, **Supplementary Fig. S14**). The number of inferred migrations from mediastinum 1 (M1) to mediastinum 2 (M2) was particularly diminished. In other respects, the tissue-migration graphs were broadly similar. Together, these analyses indicate that VINE's variational strategy for migration inference maintains many of the strengths of the fully Bayesian BEAM method, including the ability to characterize the joint posterior distribution of lineage trees and migration graphs, but with considerably improved scalability.

Discussion

Bayesian methods are now widely used in phylogenetic inference, but as datasets steadily grow in size and complexity, the computational cost of MCMC sampling becomes increasingly burdensome. In this article, we have introduced a new computational method, called VINE, that demonstrates for the first time that variational phylogenetic inference can be competitive with state-of-the-art MCMC methods in terms of model fit while offering substantially shorter running times. In addition to several common DNA substitution models, VINE also supports inference of cell-lineage phylogenies based on CRISPR barcode-editing mutation matrices. On this task, VINE fits the data as well as the recently published LAML [31] and BEAM [27] methods, but is significantly faster across a broad range of simulated and real datasets.

We additionally extended VINE to implement the tissue-migration model in BEAM and found, again, that it offered similar performance at much greater speeds. We did observe some reduction in accuracy on this problem in comparison to BEAM, which might be related VINE's heuristic approach of estimating migration rates as nuisance variables, rather than treating them in a fully Bayesian manner. (Many of these rates were driven to zero in VINE.) We plan to experiment with regularization strategies to improve this behavior. Nevertheless, the generally strong performance of VINE on this task suggests that it might be worth extending the model to address other inference tasks associated with phylogeny inference. One possibility would be to replace our model of discrete tissue labels with a more general model for latent cell states (e.g., [71]). Other possibilities are to adapt VINE to address the problems of phylogeographic reconstruction of the movements of ancestral species [72] or ancestral recombination graph (ARG) inference [2].

Development of VI for Bayesian phylogenetic inference has been an active area of recent methodological development in phylogenetics, with numerous innovative new programs appearing over the last few years [35, 38–45]. In our design of VINE, we drew heavily from this emerging literature, focusing in particular on the use of a continous embedding of taxa, distance-based phylogenetic reconstruction, and stochastic gradient ascent for optimization of the evidence lower bound (ELBO) [44, 45]. This approach has several important advantages over earlier attempts at phylogenetic VI, by avoiding the need to enumerate discrete topological features [35] and aligning naturally with powerful tools for stochastic optimization. It effectively leverages the long history of distance-based methods for phylogenetic reconstruction (e.g., [49, 58]), but marries them with classical likelihood-based strategies within the framework of variational inference. In our experiments, however, we found that previously published phylogenetic VI methods are not yet viable alternatives to MCMC for applied phylogenetics, and typically require orders of magnitude longer run times on datasets of even modest size. By introducing several new algorithmic innovations and optimizing our code, we were able to improve speeds by factors of hundreds to thousands, making it practical for the first time to use VI in large-scale phylogenetics.

One key difference of VINE from similar VI programs such as GeoPhy [44] and Dodonaphy [45] is that, by default, it makes use of a higher dimensional embedding space, typically with $d \geq 5$ rather than $d = 2$ or $d = 3$. To our surprise, we found that stochastic optimization was more, rather than less, effective at these higher dimensions, despite the larger number of free parameters. In this way, our approach is less like conventional phylogenetics and more like modern strategies for training deep neural networks, where investigators typically rely on the remarkable effectivess of SGA to optimize models that are intentionally overparameterized. Notably, however, the use of a hyperbolic embedding geometry offered no advantage in our hands, even after considerable effort in tuning the dimensionality, curvature, and scale of the space—despite the potential value of non-Euclidean embeddings in phylogenetics [44, 45, 48, 61].

Another key innovation in VINE is its strategy for differentiation through the neighbor-joining or UPGMA algorithms. Because gradients are approximate anyway in the setting of SGA, we chose not to work with formally differentiable relaxations of these algorithms (see [45]) and instead derived our own recursive procedure for propagating derivatives through the standard algorithms conditional on a choice of nearest neighbors (see **Methods**). Notably, once the neighbors are fixed, both algorithms can be shown to perform linear transformations on a vector of pairwise distances. This strategy allowed us to keep the procedures

for backpropagation lightweight and simple, and avoid the heavy computational machinery of automatic differentiation. Our approach effectively factors out the tree topology from gradient calculations, but as we show, the fundamental stochasticity of the optimization algorithm is nonetheless sufficient to ensure that the space of topologies is explored.

In our benchmarking experiments for DNA alignments, we compared the running times of VINE with those of two of the most widely used packages for MCMC-based Bayesian phylogenetics, BEAST 2 and MrBayes. There are many challenges in ensuring that such a comparison is fair, however, including the perennial issue of how to assess convergence of the Markov chain (e.g., [73]), as well as more technical concerns such as whether or not to run multiple coupled chains, how to tune proposal distributions, and how to manage parallelization. (Both BEAST 2 and MrBayes have the option of using the BEAGLE library for parallelization [74], but it was not included in our simulation experiments.) We sought to place the MCMC and VI methods on an even field by using rigorous convergence criteria and standard models with default proposals, and running all programs on a single CPU core, but these criteria are inevitably somewhat subjective. Still, even if alternative benchmarking strategies were to diminish the speed advantages of VINE, the VI paradigm has the advantage of requiring less tuning of sampling strategies and monitoring of convergence. In addition, once the variational model has been fitted to the data, any number of independent samples can be drawn from the approximate posterior at low computational cost.

At the same time, we observed a persistent tendency in VINE to underestimate the variance of the posterior distribution—a known problem in VI when the approximating distribution does not have sufficient flexibility to accommodate the structure of the true posterior [34, 75]. We attempted to address this problem with several modeling extensions, including richer parameterizations of the covariance matrix, regularization of covariance parameters, and the use of radial and planar normalizing flows to accommodate nonlinearities in the relationship between the multivariate normal sampling distribution and the space of phylogenetic trees. These extensions improved the posterior variance somewhat, but it remained underdispersed. By contrast, current MCMC-based methods appear to explore the posterior distribution effectively, at least at the scales we considered. At present, they should be preferred to VI in applications that require a complete representation of posterior uncertainty. It might be possible to improve the quality of VINE's approximate posterior further by using techniques such as mixture models (e.g., [43]).

In our comparisons of DNA alignments, we chose to focus on Bayesian methods to the exclusion of methods based on maximum likelihood (ML). It bears mentioning, however, that programs such as RAxML [29, 76] and IQ-TREE [30, 77] have recently made it possible to perform ML phylogenetic inference at scale with astonishing speeds. These programs remain substantially faster than both VINE and MCMC-based methods and for the foreseeable future will remain better choices for trees with many thousands of taxa. Still, these ML methods generally report a single tree with branch lengths—a point estimate of the phylogeny—rather than attempting to characterize the full posterior distribution, and in many applications it will be worthwhile to expend additional CPU cycles to characterize the uncertainty of the phylogeny.

The full scaling potential of phylogenetic VI based on continuous embeddings of taxa remains unclear. Our current implementation supports multithreading of likelihood calculations, which considerably accelerates model fitting, and GPU acceleration (perhaps via BEAGLE) could potentially also be added. At the

same time, the use of an explicit distance matrix for taxa and distance-based phylogeny reconstruction on each iteration of the algorithm imposes a fundamental lower bound of $O(n^2)$ on its computational complexity. We speculate that, with aggressive optimizations to the current design, VINE could perhaps be scaled up one to two orders of magnitude more, allowing practical application to alignments with >10,000 taxa. To scale the method further would likely require more heuristic approximations, such as divide-and-conquer approaches. In any case, the work presented here points a path forward to increasing the range of datasets to which approximate Bayesian methods can be applied.

Methods

The core variational inference problem

The core problem is to approximate the posterior distribution of phylogenetic trees given a set of observed genotypes using an approximate variational distribution $q(\tau, \mathbf{b})$, where τ is a rooted binary tree with n tips and $\mathbf{b} \in \mathbb{R}^{2n-2}$ is a corresponding vector of nonnegative branch lengths. We denote the data by an $n \times L$ matrix $\mathbf{X}$, which may be either a standard multiple alignment of DNA sequences or a mutation matrix representing CRISPR-edited barcodes (with L observed sites in either case).

In the standard manner for variational inference (reviewed in [34]), we fit the model by minimizing the Kullback-Leibler (KL) divergence between $q(\tau, \mathbf{b})$ and the true posterior distribution, $p(\tau, \mathbf{b} \,|\, \mathbf{X})$. The KL divergence can be expressed as,

$$\begin{aligned}
\mathrm{KL}(q(\tau, \mathbf{b}) \,\|\, p(\tau, \mathbf{b} \,|\, \mathbf{X})) &= \mathbb{E}_q \left[\log \frac{q(\tau, \mathbf{b})}{p(\tau, \mathbf{b} \,|\, \mathbf{X})} \right] \\
&= \mathbb{E}_q \left[\log \frac{q(\tau, \mathbf{b})}{p(\mathbf{X} \,|\, \tau, \mathbf{b})\, p(\tau, \mathbf{b})} + \log p(\mathbf{X}) \right] \qquad \text{(Bayes' rule)} \\
&= \mathbb{E}_q \left[\log q(\tau, \mathbf{b}) - \log p(\mathbf{X} \,|\, \tau, \mathbf{b}) - \log p(\tau, \mathbf{b}) \right] + \log p(\mathbf{X}) \\
&= -\mathbb{E}_q \left[\log p(\mathbf{X} \,|\, \tau, \mathbf{b}) \right] + \mathrm{KL}(q(\tau, \mathbf{b}) \,\|\, p(\tau, \mathbf{b})) + \log p(\mathbf{X}) \\
&= \log p(\mathbf{X}) - \mathcal{L}_q, \qquad (1)
\end{aligned}$$

where the expectations are evaluated with respect to $q(\tau, \mathbf{b})$ and the evidence lower bound (ELBO) $\mathcal{L}_q$ is defined as,

$$\begin{aligned}
\mathcal{L}_q &= \mathbb{E}_q \left[\log p(\mathbf{X} \,|\, \tau, \mathbf{b}) \right] - \mathrm{KL}(q(\tau, \mathbf{b}) \,\|\, p(\tau, \mathbf{b})) \\
&= \mathbb{E}_q \left[\ell(\tau, \mathbf{b}; \mathbf{X}) \right] - \mathrm{KL}(q(\tau, \mathbf{b}) \,\|\, p(\tau, \mathbf{b})). \qquad (2)
\end{aligned}$$

Here we introduce the notation $\ell(\tau, \mathbf{b}; \mathbf{X}) = \log p(\mathbf{X} \,|\, \tau, \mathbf{b})$ for the standard phylogenetic log likelihood, that is, the log probability of the genotype data given the tree. Notice that, because the KL divergence must be nonnegative, the ELBO $\mathcal{L}_q$ is a strict lower bound on the marginal log likelihood, $\log p(\mathbf{X})$. The essential idea of variational inference, therefore, is to choose the free parameters of q to maximize $\mathcal{L}_q$, forcing the KL divergence to shrink toward zero and making $q(\tau, \mathbf{b})$ approximate the true posterior distribution as closely as possible given its functional form.

Continuous embeddings and differentiable transformations

Our general strategy (**Fig. 1**) is to sample embeddings $\mathbf{x} \in \mathbb{R}^{nd}$ of n taxa in d-dimensional space from a multivariate normal (MVN) distribution $q = \mathrm{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} \in \mathbb{R}^{nd}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{nd \times nd}$; to optionally convert $\mathbf{x}$ to $\mathbf{y}$ using normalizing flows; to compute an induced distance matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ from $\mathbf{y}$; to obtain a rooted phylogenetic tree τ with branch lengths $\mathbf{b} \in \mathbb{R}^{2n-2}$ from $\mathbf{D}$ using a deterministic, distance-based tree reconstruction algorithm; to compute the phylogenetic log likelihood $\ell(\tau, \mathbf{b}; \mathbf{X})$ based on that tree and the specified mutation model; and, finally, to estimate the ELBO using an expectation of the phylogenetic log likelihood. We then compute the gradient of the ELBO with respect to the free parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by backpropagation through all steps of this transformation, and optimize the parameters by SGA. Notably, all nd components of $\boldsymbol{\mu}$ are maintained as free parameters in the optimization problem, but various reduced parameterizations can be assumed for $\boldsymbol{\Sigma}$ (see below).

To allow for pathwise differentiation with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we use the standard technique of reparameterizing the MVN variational distribution (the "reparameterization trick"). We introduce a standard multivariate normal random variate $\mathbf{z} \in \mathbb{R}^{nd}$, with $\mathbf{z} \sim \mathrm{MVN}(\mathbf{0}, \mathbf{I})$, and redefine $\mathbf{x}$ as,

$$\mathbf{x} \;=\; f_{\mathrm{MVN}}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;=\; \boldsymbol{\mu} + \mathbf{L}\mathbf{z},$$

where $\mathbf{L}$ is a (lower-triangular) Cholesky factor such that $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$. In this way, the expressiveness of the MVN is maintained but the randomness in samples from q now derives from a parameter-free density, and all subsequent transformations are deterministic and differentiable.

Following this linear map, $\mathbf{x} = f_{\mathrm{MVN}}(\mathbf{z})$, a series of differentiable normalizing flows can optionally be applied to accommodate nonlinear distortions to the MVN without altering the dimensionality of $\mathbf{x}$ (detailed in the **Supplementary Material**). We can summarize the cumulative effect of these normalizing flows as,

$$\mathbf{y} \;=\; f_{\mathrm{NF}}(\mathbf{x}; \theta_{\mathrm{NF}}) \in \mathbb{R}^{nd},$$

where θ_{NF} denotes a set of relevant free parameters. (If the normalizing flows are omitted, f_{NF} can simply be assumed to be the identity function, so that $\mathbf{y} = \mathbf{x}$.)

A third transformation produces a symmetric matrix of pairwise distances $\mathbf{D} \in \mathbb{R}^{n \times n}$ from $\mathbf{y}$:

$$\mathbf{D} \;=\; f_{\mathrm{D}}(\mathbf{y}),$$

where f_{D} is such that $D_{ij} = \mathscr{d}(\mathbf{y}_i, \mathbf{y}_j)$ where $\mathscr{d}$ is a suitable distance function between points in the embedding space (detailed below).

Finally, a fourth transformation produces a tree with branch lengths, $(\tau, \mathbf{b})$, from $\mathbf{D}$ using a deterministic distance-based phylogeny reconstruction algorithm:

$$(\tau, \mathbf{b}) \;=\; f_{\mathrm{phy}}(\mathbf{D}),$$

where τ is a rooted binary tree on n leaves and $\mathbf{b} \in \mathbb{R}^{2n-2}$ contains the corresponding branch lengths. In the case of DNA substitution models, we use the neighbor-joining algorithm [49], and in the case of CRISPR

barcode editing models, where an ultrametric tree is customary, we use the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) algorithm [58]. In order to allow informative priors to be applied, we impose a rooting on neighbor-joining trees using the midpoint method (root at midpoint of longest span between taxa). In the CRISPR case, we introduce an additional free parameter for a leading branch to the root of the tree, as is typical in this literature.

Taking advantage of the reparameterization trick, let g represent the full transformation from $\mathbf{z}$ to $(\tau, \mathbf{b})$:

$$(\tau, \mathbf{b}) = g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \theta_{\text{NF}}) = \left(f_{\text{phy}} \circ f_{\text{D}} \circ f_{\text{NF}} \circ f_{\text{MVN}}\right)(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \theta_{\text{NF}}).$$

Notice that g depends only on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, as well as any free parameters θ_{NF} of the normalizing flows, because f_{phy} and f_{D} are parameter-free. For simplicity, we henceforth ignore θ_{NF} and focus on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

The ELBO $\mathcal{L}_q$ can therefore be re-expressed explicitly in terms of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as,

$$\begin{aligned}\mathcal{L}_q &= \mathbb{E}_{q(\tau,\mathbf{b})}\left[\ell(\tau, \mathbf{b}; \mathbf{X})\right] - \text{KL}\left(q(\tau, \mathbf{b}) \,\|\, p(\tau, \mathbf{b})\right) \\ &= \mathbb{E}_{q(\mathbf{z})}[\ell(g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X})] - \text{KL}\left(q(\tau, \mathbf{b}) \,\|\, p(\tau, \mathbf{b})\right), \end{aligned} \tag{3}$$

where we take care to specify the distribution used for each expectation. Here, $q(\mathbf{z})$ denotes the standard MVN for $\mathbf{z}$ and $q(\tau, \mathbf{b})$ denotes the induced distribution over phylogenetic trees.

To proceed further, we must specify the prior distribution, $p(\tau, \mathbf{b})$. We consider two cases. In case (1), the default in VINE, we allow the prior to be defined implicitly by assuming that the MVN random variate $\mathbf{x} \sim \text{MVN}(\mathbf{0}, \mathbf{I})$, as is often done with variational autoencoders. In case (2), discussed below, we allow for a general prior distribution over phylogenetic trees. Details on phylogenetic priors supported by VINE are provided in the **Supplementary Materials**.

In case (1), the variational distribution is defined entirely with respect to $\mathbf{x}$; the tree $(\tau, \mathbf{b})$ is simply a deterministic function of this random variable that enables us to evaluate the phylogenetic likelihood. Therefore, we can write,

$$\begin{aligned}\mathcal{L}_q &= \mathbb{E}_{q(\mathbf{z})}[\ell(g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X})] - \text{KL}\left(q(\mathbf{x}) \,\|\, p(\mathbf{x})\right) \\ &= \mathbb{E}_{q(\mathbf{z})}[\ell(g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X})] - \frac{1}{2}\left[\text{tr}(\boldsymbol{\Sigma}) + \boldsymbol{\mu}^\top \boldsymbol{\mu} - nd - \log\det \boldsymbol{\Sigma}\right], \end{aligned} \tag{4}$$

where the last term represents the KLD for $q(\mathbf{x}) = \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $p(\mathbf{x}) = \text{MVN}(\mathbf{0}, \mathbf{I})$.

The first term of equation 4 is simply the expectation of the standard phylogenetic log likelihood under the variational distribution. Because of the complex nonlinear nature of the transformation g, there is no straightforward way to calculate this quantity exactly, but it can easily be estimated by Monte Carlo sampling (an alternative estimation method based on a Taylor approximation is described below). As a result, the approximate ELBO becomes,

$$\mathcal{L}_q \approx \frac{1}{M}\sum_{i=1}^{M}\left[\ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X})\right] - \frac{1}{2}\left[\text{tr}(\boldsymbol{\Sigma}) + \boldsymbol{\mu}^\top \boldsymbol{\mu} - nd - \log\det \boldsymbol{\Sigma}\right], \tag{5}$$

where $\mathbf{z}_i$ is the ith sample drawn from q and M is the number of samples.

In case (2), allowing for a general phylogenetic prior, $p(\tau, \mathbf{b})$, the KLD is no longer available in closed form, so we re-express the ELBO in a form more typical for standard VI. Making the transformation g explicit, we have,

$$\begin{aligned}\mathcal{L}_q &= \mathbb{E}_{q(\mathbf{z})}[\ell(g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X}) + \log p(g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))] + \mathrm{H}\left(q(\mathbf{x})\right) \\ &= \mathbb{E}_{q(\mathbf{z})}[\ell(g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X}) + \log p(g(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))] + \frac{1}{2}\left[nd(1 + \log 2\pi) + \log\det \boldsymbol{\Sigma}\right] \end{aligned} \tag{6}$$

where $\mathrm{H}(q(\mathbf{x}))$ denotes the entropy of $q(\mathbf{x})$, which is also available in closed form. The expectation in equation 6 now represents the expected complete-data log likelihood, including both the prior $p(\tau, \mathbf{b})$ and the conditional likelihood, $p(\mathbf{X}; \tau, \mathbf{b})$. These terms can be evaluated together by Monte Carlo sampling, in an analogous manner to case 1,

$$\mathcal{L}_q \approx \frac{1}{M}\sum_{i=1}^{M}\left[\ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X}) + \log p(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}))\right] + \frac{1}{2}\left[nd(1 + \log 2\pi) + \log\det \boldsymbol{\Sigma}\right]. \tag{7}$$

Notice that, for each sample $\mathbf{z}_i$ and induced tree $(\tau_i, \mathbf{b}_i) = g(\mathbf{z}_i)$, the log likelihood can be evaluated efficiently in the standard manner, using Felsenstein's pruning algorithm and either an appropriate DNA substitution model or a mutation model for CRISPR barcodes. Thus, the calculation of the ELBO reduces to elementary operations and well-known algorithms for phylogenetics (neighbor joining/UPGMA and the pruning algorithm).

Chain rule for gradients

The gradient of the Monte-Carlo approximated ELBO (equation 5) with respect to the free parameters of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can easily be re-expressed in terms of gradients of the phylogenetic log likelihoods for the sampled points. In case (1) above (equation 5),

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\mu}}\mathcal{L}_q &\approx \frac{1}{M}\sum_{i=1}^{M}\left[\frac{\partial}{\partial \boldsymbol{\mu}}\ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X})\right] - \frac{1}{2}\left[\frac{\partial}{\partial \boldsymbol{\mu}}\boldsymbol{\mu}^\top \boldsymbol{\mu})\right] \\ &= \frac{1}{M}\sum_{i=1}^{M}\left[\frac{\partial}{\partial \boldsymbol{\mu}}\ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X})\right] - \boldsymbol{\mu} \end{aligned} \tag{8}$$

and

$$\frac{\partial}{\partial \boldsymbol{\Sigma}}\mathcal{L}_q \approx \frac{1}{M}\sum_{i=1}^{M}\left[\frac{\partial}{\partial \boldsymbol{\Sigma}}\ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X})\right] - \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\Sigma}}\left[\mathrm{tr}(\boldsymbol{\Sigma}) - \log\det \boldsymbol{\Sigma}\right]. \tag{9}$$

Similarly, in case (2) (equation 7),

$$\frac{\partial}{\partial \boldsymbol{\mu}}\mathcal{L}_q \approx \frac{1}{M}\sum_{i=1}^{M}\left[\frac{\partial}{\partial \boldsymbol{\mu}}\ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X}) + \frac{\partial}{\partial \boldsymbol{\mu}}\log p(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}))\right], \tag{10}$$

because the entropy of $q(\mathbf{x})$ does not depend on $\boldsymbol{\mu}$, and,

$$\frac{\partial}{\partial \boldsymbol{\Sigma}}\mathcal{L}_q \approx \frac{1}{M}\sum_{i=1}^{M}\left[\frac{\partial}{\partial \boldsymbol{\Sigma}}\ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X}) + \frac{\partial}{\partial \boldsymbol{\Sigma}}\log p(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}))\right] + \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\Sigma}}\log\det \boldsymbol{\Sigma}. \tag{11}$$

The last terms in equations 9 and 11 can be computed in closed-form in a manner that depends on the choice of parameterization for $\boldsymbol{\Sigma}$, as shown later. The key challenge is therefore to compute gradients for the phylogenetic log likelihood, $\ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X})$, and optionally, the corresponding log prior term. We will focus on the log likelihood here; the prior follows by analogy.

We start by observing that, in principle, we can propagate derivatives forward along the sequence of component functions f_{MVN}, f_{NF}, $f_{\mathbf{D}}$, and f_{phy} by computing the corresponding Jacobian matrices. In particular, for a vector of free parameters $\boldsymbol{\phi} \in \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ of dimension k, we can write,

$$
\begin{aligned}
J_{\text{MVN}} \in \mathbb{R}^{nd \times k} &= \frac{\partial \mathbf{x}}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} f_{\text{MVN}}(\mathbf{z}_i) \\
J_{\text{NF}} \in \mathbb{R}^{nd \times nd} &= \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} f_{\text{NF}}(\mathbf{x}) \\
J_{\mathbf{D}} \in \mathbb{R}^{\binom{n}{2} \times nd} &= \frac{\partial \mathbf{D}}{\partial \mathbf{y}} = \frac{\partial}{\partial \mathbf{y}} f_{\mathbf{D}}(\mathbf{y}) \\
J_{\text{phy}} \in \mathbb{R}^{2n-2 \times \binom{n}{2}} &= \frac{\partial \mathbf{b}}{\partial \mathbf{D}} = \frac{\partial}{\partial \mathbf{D}} f_{\text{phy}}(\mathbf{D}),
\end{aligned}
$$

where in the last step we consider the branch lengths $\mathbf{b}$ only and ignore the tree topology τ, as explained in the **Supplementary Material**. To complete the process, we must also consider the gradient of the phylogenetic log likelihood with respect to the $2n - 2$ branch lengths:

$$\nabla_{\mathbf{b}} \ell(\tau, \mathbf{b}; \mathbf{X}).$$

This is a familiar quantity in phylogenetic analysis, which can be efficiently calculated using an inside-outside algorithm on the tree (see **Supplementary Material**).

Assuming for the moment that these objects can all be obtained, the gradient of interest can be computed by a straightforward application of the chain rule:

$$\nabla_{\boldsymbol{\phi}} \ell(g(\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}); \mathbf{X}) = J_{\text{MVN}}^{\top} \; J_{\text{NF}}^{\top} \; J_{\mathbf{D}}^{\top} \; J_{\text{phy}}^{\top} \; \nabla_{\mathbf{b}} \ell(\tau, \mathbf{b}; \mathbf{X}).$$

The product of four Jacobian matrices thus converts the standard $(2n - 2)$-dimensional branch-length gradient to a k-dimensional parameter gradient, as required. In this way, $\nabla_{\mathbf{b}} \ell(\tau, \mathbf{b}; \mathbf{X})$ can be propagated backwards from the branch lengths through the distance matrix, flows, and embedded points, to the MVN parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

In practice, we avoid instantiating the full Jacobian matrices and instead implicitly propagate gradients through them using an efficient reverse-mode algorithm. Moreover, the first three Jacobians reflect elementary differentiable functions and can be accommodated analytically. The main difficulty lies with the fourth Jacobian, J_{phy}, which reflects the entire tranformation of a distance matrix $\mathbf{D}$ to a phylogenetic tree with branch lengths. An algorithm for efficiently computing this Jacobian implicitly and additional details are provided in the **Supplementary Material**.

Implementations of Neighbor-Joining and UPGMA

VINE uses custom implementations in C of the NJ and UPGMA algorithms that take a distance matrix as input and return a tree with branch lengths. In the case of NJ, the sequence of merged neighbors and associated meta-data are recorded for later use in backpropagation (see **Supplementary Material**).

Because these tree-building routines are rate-limiting for variational inference, we adapted the standard algorithms to use a min-heap for efficient identification of the pair of nodes to join on each step (e.g., the minimum entry of $\mathbf{Q}$ in NJ). This approach reduces the asymptotic running time from $O(n^3)$ for naive implementations to $O(n^2 \log n)$ (see, e.g., [78], who use further optimizations to achieve $O(n^2)$ for NJ).

Taylor approximation of the ELBO

Estimating the ELBO by Monte Carlo sampling (equations 5 & 7) is effective for optimization but computationally expensive. We found that the efficiency of the algorithm can be substantially improved by making use of a Taylor approximation for the ELBO. In particular, we approximate the expectation of the log likelihood (see equation 4) using a second-order Taylor approximation around the mean (corresponding to $\mathbf{z} = \mathbf{0}$),

$$\begin{aligned}\mathbb{E}_{q(\mathbf{z})}[\ell(g(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma});\mathbf{X})] &\approx \ell(g(\mathbf{z}=\mathbf{0};\boldsymbol{\mu},\boldsymbol{\Sigma});\mathbf{X}) + \nabla_{\boldsymbol{\mu},\boldsymbol{\Sigma}}\,\ell(g(\mathbf{z}=\mathbf{0};\boldsymbol{\mu},\boldsymbol{\Sigma});\mathbf{X})^\top \mathbf{0} + \frac{1}{2}\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Sigma}\right) \\ &= \ell(g(\mathbf{z}=\mathbf{0};\boldsymbol{\mu});\mathbf{X}) + \frac{1}{2}\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Sigma}\right), \end{aligned} \tag{12}$$

where $\mathbf{H}$ is the Hessian matrix for the entire transformation, $\ell(g(\mathbf{z}=\mathbf{0};\boldsymbol{\mu},\boldsymbol{\Sigma});\mathbf{X})$, as evaluated at $\mathbf{z}=\mathbf{0}$. (Here we focus on case (1) for the prior, $\mathbf{x} \sim \mathrm{MVN}(\mathbf{0},\mathbf{I})$; case (2) follows by analogy; see equation 6.)

If we ignore the dependency of $\mathbf{H}$ on $\boldsymbol{\mu}$, which we expect to be weak, then the gradient of the approximate ELBO with respect to $\boldsymbol{\mu}$ depends only on the first term of equation 12 and the gradient with respect to the covariance $\boldsymbol{\Sigma}$ depends only on the second term. Thus, this formulation allows approximate decomposition of the ELBO into mean- and variance-related components. In practice, we find that mean-related component is strongly dominant when the model is fitted to data.

On its face, the second term in equation 12 still poses a problem, because the $nd \times nd$ Hessian matrix is impractical to instantiate explicitly. It is possible to estimate this term efficiently by way of matrix-vector products, without realizing the Hessian, using Hutchinson's method [79]. In practice, however, we find that we obtain a better approximation in comparable time by periodically estimating the LHS of equation 12 by our standard Monte Carlo method and then estimating the second term on the RHS by the difference,

$$\frac{1}{2}\mathrm{tr}\left(\mathbf{H}\boldsymbol{\Sigma}\right) \approx \mathbb{E}_{q(\mathbf{z})}[\ell(g(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma});\mathbf{X})] - \ell(g(\mathbf{z}=\mathbf{0};\boldsymbol{\mu},\boldsymbol{\Sigma});\mathbf{X}). \tag{13}$$

Because this quantity is not highly sensitive to $\boldsymbol{\mu}$, it need not be updated on every iteration of SGA. We delay computing it during a "warmup" period (50 iterations), and then update it only every 30 iterations, using an exponential moving average for both the trace quantity and its gradient. This strategy requires less

than one additional evaluation per iteration of the full $O(n^2 \log n)$ transformation g, decreasing the overall cost of SGA by nearly a factor of M in comparison to full Monte Carlo sampling.

Parameterizations of the covariance matrix $\boldsymbol{\Sigma}$

VINE supports four parameterizations of the MVN covariance matrix $\boldsymbol{\Sigma}$, which are labeled CONST, DIAG, DIST, and LOWR in order of increasing complexity (accessible via the `--covar` option). The CONST parameterization simply assumes $\boldsymbol{\Sigma} = \lambda \mathbf{I}$ and ensures nonnegativity of λ by defining it as $\lambda = e^{\eta}$, with η as a free parameter. The DIAG parameterization allows for a general diagonal covariance matrix, with free variance along each of the nd dimensions but no covariance between them: $\boldsymbol{\Sigma} = \text{diag}\{\lambda_1, \ldots, \lambda_{nd}\}$, with each $\lambda_i = e^{\eta_i}$ for nd free parameters.

The DIST and LOWR parameterizations attempt to capture more of the covariance structure across taxa while keeping the parameterization as sparse as possible. In these cases, the same covariance structure is assumed for each of the d dimensions in the embedding space, with no covariance across these dimensions. The full covariance matrix can therefore be described as $\boldsymbol{\Sigma} = \mathbf{I}_d \otimes \boldsymbol{\Sigma}_0$, where $\mathbf{I}_d$ is a $d \times d$ identity matrix, $\otimes$ is the Kronecker product, and $\boldsymbol{\Sigma}_0$ is the $n \times n$ covariance matrix shared for each embedding dimension. In these cases, VINE gains further efficiency by performing all MVN-related calculations on d independent n-dimensional MVNs with a shared covariance structure, rather than on one nd-dimensional MVN.

The DIST parameterization further assumes $\boldsymbol{\Sigma}_0 = \lambda \mathbf{S}$ and $\lambda = e^{\eta}$, resulting in a single free variance parameter η for all taxa. In this case, $\mathbf{S}$ is kept fixed at a double-centered version of the initial distance matrix $\mathbf{D}_0$,

$$\mathbf{S} = -\tfrac{1}{2}\mathbf{H}\mathbf{D}_0\mathbf{H}, \qquad \mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top, \tag{14}$$

where $\mathbf{1}$ indicates an n-dimensional vector of all 1s and $\mathbf{1}\mathbf{1}^\top$ is therefore an $n \times n$ matrix of all 1s. This matrix $\mathbf{S}$ captures the expected covariance structure of a random variable evolving by Brownian motion along the branches of an unrooted tree, and therefore, the DIST parameterization is a simple way to consider the distances between taxa without restricting the tree topology.

The most flexible parameterization, LOWR, allows for a general low-rank representation of $\boldsymbol{\Sigma}_0$, with $\boldsymbol{\Sigma}_0 = \mathbf{R}\mathbf{R}^\top$ for a general matrix $\mathbf{R}$ of dimension $n \times w$ (with nw free parameters). By default, $w = 3$ (see `--rank`). In this case, the full covariance of the embedding is $\Sigma = (RR^\top) \otimes I_d$, which has rank at most wd. Consequently, VINE can reparameterize sampling from $\text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ using only wd latent dimensions, instead of the full nd dimensions required by an unrestricted covariance matrix.

All of these parameterizations permit closed-form expressions for the gradient of the KLD (equation 9) or entropy (equation 11) and for the Jacobian J_{MVN} (see **Supplementary Material**).

Euclidean and hyperbolic geometries

By default, VINE embeds taxa in a d-dimensional Euclidean space, where d can be selected by the user via the `--dimensionality` option. (A general-purpose default is determined as a linear function of $\log n$, where n is the number of taxa.) Let $\mathbf{y}_i \in \mathbb{R}^d$ denote the embedded point for taxon i (after normalizing flows are applied). The Euclidean pairwise distance is given by,

$$D_{ij} \;=\; \frac{\|\mathbf{y}_i - \mathbf{y}_j\|}{s}, \qquad 1 \le i < j \le n,$$

where $s > 1$ is a global scale factor that allows the embedding space to be expanded relative to the distance matrix $\mathbf{D}$.

The initial mean $\boldsymbol{\mu}$ of the MVN variational distribution is estimated by classical multidimensional scaling (MDS) applied to the starting distance matrix $\mathbf{D}_0$. Let $\mathbf{D}_0^2$ be the (symmetric) matrix of squared distances, double-centered to obtain a Gram matrix $\mathbf{G}$. An eigendecomposition $\mathbf{G} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$ is computed, and the initial embedding is given by the top d scaled principal components, $\boldsymbol{\mu}_i = s(\sqrt{\lambda_1}v_{i1}, \ldots, \sqrt{\lambda_d}v_{id})$, where s is the scale factor.

In the hyperbolic setting, VINE follows the general approach outlined by Macaulay et al. [61] and embeds points on the upper sheet of the $(d{+}1)$-dimensional hyperboloid model with negative curvature $-\kappa$ (set by `--negcurvature`, default $-\kappa = 1$). Each taxon is represented by a spatial coordinate $\mathbf{y}_i \in \mathbb{R}^d$, which is lifted to a point on the hyperboloid via,

$$x_{0,i} = \sqrt{1 + \|\mathbf{y}_i\|^2}, \qquad \tilde{\mathbf{y}}_i = (x_{0,i}, \mathbf{y}_i) \in \mathbb{R}^{d+1}.$$

The hyperbolic distance between taxa i and j is,

$$D_{ij} \;=\; \frac{\alpha}{s}\,\operatorname{arcosh}\!\big(-\langle \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j\rangle_L\big), \qquad 1 \le i < j \le n,$$

where $\alpha = 1/\sqrt{-\kappa}$ is the curvature radius and $\langle \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j\rangle_L$ is the Lorentz inner product,

$$\langle \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j\rangle_L = -x_{0,i}x_{0,j} + \mathbf{y}_i^\top \mathbf{y}_j.$$

For hyperbolic embeddings, the MVN mean is initialized using a spectral variant of the *hydra* algorithm [80]. Given an initial distance matrix $\mathbf{D}^{(0)}$, we form a symmetric matrix $\mathbf{A}$ such that,

$$A_{ij} = \begin{cases} 1, & i = j, \\ \cosh\!\big(\sqrt{-\kappa}\, D_{ij}^{(0)}\, s\big), & i \ne j, \end{cases}$$

diagonalize $\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$, and set the initial spatial coordinates from the leading d eigenmodes, followed by the same global scaling by s.

The scale factor s is chosen so that the median pairwise distance between points is 25 in the Euclidean case and 4 in the hyperbolic case. The Jacobians of these mappings from embedded points to pairwise distances ($J_{\mathbf{D}}$) are provided in the **Supplementary Material**.

Stochastic gradient ascent

For stochastic gradient ascent (SGA), VINE uses a customized implementation in C of the Adam algorithm [81]. In the Monte-Carlo case, the algorithm draws a minibatch of M sets of embedded points on each iteration, $\mathbf{x}_1, \ldots, \mathbf{x}_M$ (transformed from $\mathbf{z}_1, \ldots, \mathbf{z}_M$), and uses them to obtain estimates of the ELBO and its gradient. For each minibatch, it precomputes $\mathrm{KL}(q(\mathbf{x}) \,\|\, p(\mathbf{x}))$ and its gradient, which do not depend on the sampled points. It then updates all parameters based on the gradient in the standard manner. Command-line options allow the user to control the minibatch size (`--batchsize`; default 10) and the learning rate α (`--learnrate`; default 0.05), as well as convergence criteria for the algorithm (e.g., minimum number of iterations, `--miniter`, default 200, and number of iterations over which to average when assessing convergence, `--niterconv`, default 50). In the case of the Taylor approximation, the procedure is similar but no minibatch sampling is required; the ELBO is estimated directly from the value of the log likelihood at the current mean $\boldsymbol{\mu}$ and the estimate of the trace term (equation 12). In both cases, the Adam parameters β_1 and β_2 are both fixed at 0.9. A simple scheduler is employed to manage subsampling of sites in the alignment in early iterations of the algorithm, adaptive clipping of unusually large gradients, and gradual increases of the learning rate to its target value. The algorithm also allows for a separate set of "nuisance parameters" (including parameters for substitution rates, tree priors, and the normalizing flows), which are also optimized by SGA but are not encoded in the continuous embedding.

The SGA algorithm was carefully tuned to optimize the speed and accuracy of convergence over datasets of various sizes. During this process, we found that it was quite sensitive to the scale of the embedding space, sometimes struggling to converge when distances between points grew close to zero. Therefore, we introduced the scaling factor s that allows for a difference in scale between the embedding space and the distances used for tree reconstruction (as detailed above). In particular, VINE rescales the MVN distribution for embedded points so that median distances between points are 25 and 4, respectively, for the Euclidean and hyperbolic geometries.

Mutation models

VINE includes implementations of the Jukes-Cantor (JC) [53], HKY [52], and general time reversible (GTR) [82] DNA substitution models, as well as the CRISPR-barcode mutation model developed by Seidel et al. [24] and extended by Chu et al. [31]. It also supports the discrete gamma model for rate variation among sites [83] with DNA substitution models. For the DNA models, the substitution rate matrix $\mathbf{Q}$ is normalized in the standard manner, such that $\sum_{i \neq j} \pi_i q_{ij} = 1$, where π_i is the equilibrium frequency of nucleotide i under the model, so that branch lengths can be interpreted as having units of expected substitutions per site. For the HKY and GTR models, equilibrium frequencies for the four nucleotides are simply estimated using the relative frequencies in the alignment, whereas under the JC model they are assumed to be uniform. As a result, the JC model has no free substitution rate parameters, the HKY model has a single free parameter—the transition/transversion rate ratio κ—and the GTR model has five free parameters (after accounting for rescaling). These are considered as "nuisance" parameters that are optimized in SGA but not considered for full variational Bayes characterization. The CRISPR model has a single free parame-

ter, corresponding to the silencing rate for barcodes, which is also treated as a nuisance parameter. VINE supports two parameterizations for the barcode-editing rate matrix: a single rate matrix shared by all sites (as in TiDeTree [24]; `--crispr-modtype GLOBAL`) or a separate rate matrix per site (as in LAML [31]; `--crispr-modtype SITEWISE`). It also allows for either a uniform distribution for all mutation-rate priors (`--crispr-mutprior UNIF`) or a prior that reflects the relative frequencies of mutations in the input matrix (`--crispr-mutprior EMPIRICAL`). The CRISPR mutation matrix is subjected to the same scaling constraint as the DNA substitution matrix, so the estimated branch lengths can be interpreted in expected mutations per site. For comparison, LAML trees must be scaled by the separately estimated mutation rate.

Simulations

For our simulations of DNA alignments, we used a script based on DendroPy [84] to simulate trees under a birth–death process (birth rate 1.0, death rate 0.5), with three-fold oversampling of tips followed by pruning and rescaling, to produce tree heights of 1 substitution per site and minimum branch lengths of 0.02. Branch-specific rates were drawn independently under an uncorrelated log-normal relaxed clock with mean of 1 and standard deviation of 0.6. Nucleotide alignments were then generated for each tree using `base_evolve` from PHAST under an HKY model with $\kappa = 4$ and equilibrium frequencies of $\pi_A = \pi_T = 0.3$ and $\pi_C = \pi_G = 0.2$. As noted in the text, we generated alignments of 300 bp and 10,000 bp, for various numbers of taxa n between 10 and 1000. For held-out data, we ran `base_evolve` a second time on the same simulated trees used to generate the training data.

For CRISPR lineage-tracing simulations, we used Cassiopeia's [55] BirthDeathFitnessSimulator with birth rate 0.075 and death rate 0.005, drawing branch waiting times from the absolute value of a normal distribution (μ =1/rate, $\sigma = \mu/5$) to obtain approximately balanced ultrametric trees, which were post-scaled to a fixed height of 54 days to match the data set of ref. [63]. We then applied Cas9LineageTracingDataSimulator using a cassette of 3 CRISPR target sites repeated 10 times (30 total sites), a mutation rate of 0.01, a heritable silencing rate of 1×10^{-4}, and no stochastic silencing. Edit outcomes were restricted to a predefined library of 100 possible states with fixed probabilities, ensuring reproducible mutation signatures across simulations.

Convergence criteria for MCMC

For convergence monitoring of MrBayes and BEAST 2, we applied a uniform procedure across all analyses for both simulated and real data. First, for each combination of taxa number, sequence length, substitution model, and program (BEAST 2 or MrBayes), we ran three pilot MCMC replicates with two chains per replicate and intentionally long chain lengths, recording all samples (no thinning). For each replicate, we then retrospectively computed the rank-normalized split $\hat{R}$ [85] and the effective sample size (ESS) for each of several scalar summaries in each chain in intervals of at least 100,000 iterations. For BEAST 2, the selected statistics were the log posterior, log likelihood, tree height, and tree length, and for MrBayes they were the log likelihood and tree length. For each replicate, we then identified the earliest MCMC iteration

calculation at which all parameters met ESS ≥ 400 and $\hat{R} \leq 1.01$. (For the large alignments—with 10,000 columns—we had to relax the $\hat{R}$ criterion in a few cases because not all pilot replicates converged.) ESS calculations were based on the implementation in Tracer [86], reimplemented in our own scripts. Finally, we averaged these convergence points across the three replicates to determine the target chain length. These target chain lengths were then applied uniformly across all ten replicates for each simulated data set size, model, and program. Notably, the chain lengths required for our convergence and sampling criteria were generally substantially shorter for MrBayes than for BEAST 2, and this was the dominant factor in the differences in running times we observed. Note also that we experimented with a more stringent requirement of ESS ≥ 625 but found it to require excessively long chains in some cases, so we relaxed the threshold to 400, as recommended in ref. [73].

Evaluating posterior distributions

We evaluated posterior distributions from VINE and BEAST 2 using two complementary methods: 95% credible-interval (CI) inclusion and topological entropy (**Fig. 3**). Both measures were computed from the posterior samples of trees output by each program after running on simulated data. To assess 95% CI inclusion, we extracted the empirical distribution of values induced by the posterior samples for each pairwise distance between taxa, then excluded the top and bottom 2.5% of values to obtain an empirical 95% CI. We then compared this range with the true value used in simulation. The reported values are the fractions of true pairwise distances that fall within the corresponding 95% CI. For topological entropy, we calculated the Shannon entropy of distinct topologies from the posterior samples,

$$H_\tau = -\sum_{\tau \in \mathcal{T}} p(\tau) \log p(\tau),$$

where $\mathcal{T}$ is the set of distinct tree topologies (ignoring branch lengths) and $p(\tau)$ is the sampled frequency of topology τ. Both of these calculations are supported by `evalTrees`, a utility distributed with VINE.

Models selected for analysis of simulated DNA

We selected models in BEAST 2 (v2.7.7) and MrBayes (3.2.7a) that were as close as possible to the ones in VINE given choices available for each program, generally keeping the prior distributions diffuse so that the log likelihood dominated in inference. In particular, in BEAST 2, we used a Yule prior for the phylogeny with a single free birth-rate parameter (uniform prior) and an uncorrelated log-normal relaxed clock (exponential prior with mean 2 on ucldStdev, the standard deviation in log-space). For the κ parameter in the HKY model, we used a log-normal prior with a log mean of 1.386 (corresponding to $\kappa = 4$, the value used in our simulations) and a standard deviation of 0.1. Similarly in MrBayes, we used a uniform prior over tree topologies and the default Gamma-Dirichlet prior for branch lengths. For the HKY model, we used a Beta(4,1) prior distribution for the rate of transitions relative to the rate of all substitutions. In both programs, we did not allow for variation across sites in rates, we used random starting trees, and we used empirical frequencies for the four nucleotides.

In VINE we used an implicit prior on trees corresponding to a standard multivariate normal distribution for $p(\mathbf{x})$ (the default, as discussed above). The HKY parameter κ was optimized as a nuisance parameter in SGA (`--hky85` option). Empirical nucleotide frequencies were used. Analyses that required sampled trees (such as the comparison of Robinson-Foulds distances) were based on 1000 samples from the approximate posterior following convergence of the ELBO (`-s 1000`). In most cases, default parameters were used for the minimum number of iterations before convergence (`--miniter 200`) and the number of iterations over which to average when assessing convergence (`--niteconv 50`), although for some larger trees we used larger values for the minimum number of iterations.

The same choices of models and parameters for BEAST 2, MrBayes, and VINE were used in the benchmarking analysis of real nucleic-acid data from ref. [60], as shown in **Supplementary Figs. S8 & S9**.

SARS-CoV-2 Analysis

We downloaded raw sequences (`sequences.fasta.xz`) and accompanying metadata (`metadata.tsv.xz`) from https://data.nextstrain.org/ (in `files/ncov/open/100k`), as well as the associated alignment (`aligned.fasta.xz`). All source data was downloaded on February 25, 2026. We used `augur` to exclude sequences less than 29,000 bp in length (`--min-length 29000`) or lacking complete date information (`--exclude-ambiguous-dates-by any`). We then performed stratified subsampling by region and date (`--group-by region year month --sequences-per-group 3 --subsample-seed 1`) to reduce the data set to 1030 aligned sequences. A more stringent stratified sampling step produced the smaller set of 364 sequences (`--group-by region year month --sequences-per-group 1 --subsample-seed 2`). We analyzed the resulting alignments using the same settings as for simulated data except that we used the GTR substitution model (`--gtr` in VINE), the discrete gamma model for rate variation (`--dgamma 4`), and multithreading (`--parallel 8`). Equivalent options were used in BEAST 2, with BEAGLE employed for multithreading in this case. For this analysis, we simplified our MCMC convergence criteria for BEAST 2 to require only ESS > 400, avoiding the use of pilot replicates and the $\hat{R}$ criterion.

Migration mode in VINE

VINE was extended to support the joint model for mutation and migration implemented in BEAM [27]. This version of the model is activated when tissue labels for cells are specified using the `--migration` option. In migration mode, the phylogenetic log likelihood is redefined as a sum of the standard mutation-based log likelihood and a log likelihood based on a general time reversible model for migration transitions, both computed for the same tree and branch lengths. The migration rate parameters are treated as nuisance parameters and estimated by SGA, and the calculation of gradients is modified accordingly. The embedding strategy, conversion to trees by UPGMA, and distance-based initialization required no change. An option (`--primary`, used here for both simulated and real data) allows a particular tissue label to be designated as "primary" and enforced at the root of the tree. After convergence, the program produces samples of tissue-labeled trees under the joint model, by first sampling trees from the approximate posterior and then sampling

tissue-labels conditional on each tree using an inside-outside algorithm. Output can be tissue-labeled trees in nexus format (`--labeled-trees`) and/or collapsed migration graphs in dot format (`--sample-graphs`).

To compare tissue-labeled trees across inference methods (**Fig. 6C**), we calculated four phylogenetic statistics: the minimum numbers of (1) mutations and (2) migration events required to explain the data at the tips of the tree according to Fitch-Hartigan parsimony [87, 88]; (3) the cophenetic correlation, defined as Pearson's correlation (r) between the Hamming distances between barcodes and patristic distances along the branches of the tree; and (4) a tissue homogeneity index, defined as the mean fraction of same-tissue neighbors within each clade, normalized by the expected frequency under random tip shuffling. The cophenetic correlation is a measure of how well the tree reflects the raw mutation data, approaching a value of 1 for a perfect correlation [89]. The tissue homogeneity index is a measure of clustering by tissue type similar to measures of cluster purity [90] or phylogenetic trait clustering [91].

Running BEAM

BEAM was run in two modes: tree-only inference using the sitewise model with uniform fixed edit rates, and joint tree-migration inference that additionally incorporated a GTR migration rate matrix. In both modes, the starting tree was a Cassiopeia-Greedy tree uniformly rescaled to match the appropriate origin time. Markov Chain Monte Carlo (MCMC) was used to sample from the posterior distribution of the tree and other phylogenetic parameters using MCMC proposals that included Wilson-Balding subtree prune and regraft [92] and nearest-neighbor interchange tree moves, internal-node and total-tree-height rescaling, and standard continuous scaling operators on all other continuous parameters. Bactrian proposal moves [93–95] were used where possible, and the frequency and scale of different proposals were tuned adaptively as is standard in BEAST 2 adaptive Metropolis-Hastings MCMC. Priors were shared across both modes where applicable. As in ref. [27], the tree prior was a birth-death process. Other priors included exponential priors (mean 0.1) on the strict clock and silencing rates, an exponential prior (mean 1.0) on the birth-death rate difference, and a Uniform(0,1) prior on the relative death rate. For joint tree-migration inference, migration rates additionally had exponential priors (mean 1.0) and the migration clock rate had an exponential prior (mean 0.1). Parameter initializations were based on test runs for birth/death and clock rates to ensure a stable starting state with sufficient prior density, while migration rates were initialized uniformly. MCMC was run until parameter ESS values exceeded 400 and traces were visually stationary.

Running other VI programs and LAML

Details on how all other programs were run are provided in the **Supplementary Material**.

Software implementation

VINE is written in C (C99) and available from github (https://github.com/CshlSiepelLab/vine) under a standard BSD 3-Clause License. It requires a recent installation of PHAST (https://github.com/CshlSiepelLab/phast) [96], from which it borrows DNA substitution models and alignment-handling routines. VINE is easily installable via bioconda (use `conda install -c conda-forge -c bioconda vine-phylo`) and homebrew (use `brew tap CshlSiepelLab/tools` and `brew install vine`). In both cases, PHAST will be installed as a dependency if needed.

Funding

This work was supported by US National Institutes of Health (NIH) National Institute of General Medical Sciences Grant R35-GM127070 and National Cancer Institute (NCI) Grants R01-CA272466 and 5P30-CA045508, as well as Starr Cancer Consortium Grant I16-0060, a National Science Foundation Graduate Research Fellowship (to S.J.S.), a Starr Centennial Scholarship (to S.J.S.), and the Simons Center for Quantitative Biology at CSHL. The content is solely the responsibility of the authors and does not necessarily represent the official views of the US National Institutes of Health.

Conflict of Interest

The authors declare no competing interests.

Acknowledgments

We thank other members of the community at Cold Spring Harbor Laboratory as well as Dawid Nowak and his team at Weill Cornell Medicine for helpful feedback.

References

[1] Dewar, A. E., Belcher, L. J. & West, S. A. A phylogenetic approach to comparative genomics. *Nat Rev Genet* **26**, 395–405 (2025). URL http://dx.doi.org/10.1038/s41576-024-00803-0.

[2] Nielsen, R., Vaughn, A. H. & Deng, Y. Inference and applications of ancestral recombination graphs. *Nat Rev Genet* **26**, 47–58 (2024). URL http://dx.doi.org/10.1038/s41576-024-00772-4.

[3] Attwood, S. W., Hill, S. C., Aanensen, D. M., Connor, T. R. & Pybus, O. G. Phylogenetic and phylodynamic approaches to understanding and combating the early SARS-CoV-2 pandemic. *Nat Rev Genet* **23**, 547–562 (2022). URL http://dx.doi.org/10.1038/s41576-022-00483-8.

[4] McKenna, A. *et al.* Whole-organism lineage tracing by combinatorial and cumulative genome editing. *Science* **353**, aaf7907 (2016).

[5] Frieda, K. L. *et al.* Synthetic recording and in situ readout of lineage information in single cells. *Nature* **541**, 107–111 (2017).

[6] Jiang, J. *et al.* scLTdb: a comprehensive single-cell lineage tracing database. *Nuc Acids Res* **53**, D1173–D1185 (2024).

[7] Askary, A. *et al.* The lives of cells, recorded. *Nat Rev Genet* 1–20 (2024).

[8] Kuipers, J., Jahn, K. & Beerenwinkel, N. Advances in understanding tumour evolution through single-cell sequencing. *Biochimica et Biophysica Acta* **1867**, 127–138 (2017).

[9] Li, L. *et al.* Resolving tumor evolution: a phylogenetic approach. *J Nat Cancer Cent* **4**, 97–106 (2024).

[10] Alemany, A., Florescu, M., Baron, C. S., Peterson-Maduro, J. & van Oudenaarden, A. Whole-organism clone tracing using single-cell sequencing. *Nature* **556**, 108–112 (2018).

[11] Chan, M. M. *et al.* Molecular recording of mammalian embryogenesis. *Nature* **570**, 77–82 (2019).

[12] Gong, W. *et al.* Benchmarked approaches for reconstruction of in vitro cell lineages and *in silico* models of *C. elegans* and *M. musculus* developmental trees. *Cell Syst* **12**, 810–826 (2021).

[13] Raj, B. *et al.* Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. *Nat Biotech* **36**, 442–450 (2018).

[14] Xie, L. *et al.* Comprehensive spatiotemporal mapping of single-cell lineages in developing mouse brain by CRISPR-based barcoding. *Nature Methods* **20**, 1244–1255 (2023).

[15] Rannala, B. & Yang, Z. Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference. *J Mol Evol* **43**, 304–311 (1996).

[16] Mau, B., Newton, M. A. & Larget, B. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. *Biometrics* **55**, 1–12 (1999).

[17] Larget, B. & Simon, D. L. Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. *Mol Biol Evol* **16**, 750–759 (1999).

[18] Huelsenbeck, J. P. & Ronquist, F. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics* **17**, 754–755 (2001).

[19] Ronquist, F. *et al.* MrBayes 3.2: Efficient Bayesian phylogenetic inference and model choice across a large model space. *Syst Biol* **61**, 539–542 (2012). URL http://dx.doi.org/10.1093/sysbio/sys029.

[20] Drummond, A. J. & Rambaut, A. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol Biol* **7**, 214 (2007). URL http://dx.doi.org/10.1186/1471-2148-7-214.

[21] Bouckaert, R. *et al.* BEAST 2: a software platform for Bayesian evolutionary analysis. *PLoS Comput Biol* **10**, e1003537 (2014). URL http://dx.doi.org/10.1371/journal.pcbi.1003537.

[22] Bouckaert, R. *et al.* BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. *PLoS Comput Biol* **15**, e1006650 (2019).

[23] Höhna, S. *et al.* RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Syst Biol* **65**, 726–736 (2016). URL http://dx.doi.org/10.1093/sysbio/syw021.

[24] Seidel, S. & Stadler, T. TiDeTree: a Bayesian phylogenetic framework to estimate single-cell trees and population dynamic parameters from genetic lineage tracing data. *Proc Biol Sci* **289**, 20221844 (2022).

[25] Seidel, S. *et al.* SciPhy: A Bayesian phylogenetic framework using sequential genetic lineage tracing data. *bioRxiv* (2024). URL http://dx.doi.org/10.1101/2024.10.01.615771.

[26] Zwaans, A., Seidel, S., Manceau, M. & Stadler, T. A Bayesian phylodynamic inference framework for single-cell CRISPR/Cas9 lineage tracing barcode data with dependent target sites. *Phil Trans R Soc B* **380**, 20230318 (2025). URL http://dx.doi.org/10.1098/rstb.2023.0318.

[27] Staklinski, S. J. *et al.* Bayesian inference of tissue-migration histories in metastatic cancer from cell-lineage tracing data. *Cell Genomics (in press)* (2026).

[28] Fisher, A. A. *et al.* Scalable Bayesian phylogenetics. *Phil Trans R Soc B* **377**, 20210242 (2022). URL http://dx.doi.org/10.1098/rstb.2021.0242.

[29] Kozlov, A. M., Darriba, D., Flouri, T., Morel, B. & Stamatakis, A. RAxML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics* **35**, 4453–4455 (2019). URL http://dx.doi.org/10.1093/bioinformatics/btz305.

[30] Minh, B. Q. *et al.* IQ-TREE 2: New models and efficient methods for phylogenetic inference in the genomic era. *Mol Biol Evol* **37**, 1530–1534 (2020). URL http://dx.doi.org/10.1093/molbev/msaa015.

[31] Chu, G., Mai, U., Schmidt, H. *et al.* Maximum likelihood inference of time-scaled cell lineage trees with mixed-type missing data using LAML. *Genome Biol* **26**, 189 (2025).

[32] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. & Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning* **37**, 183–233 (1999).

[33] Wainwright, M. J., Jordan, M. I. *et al.* Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* **1**, 1–305 (2008).

[34] Blei, D. M., Kucukelbir, A. & McAuliffe, J. D. Variational inference: a review for statisticians. *J Am Stat Assoc* **112**, 859–877 (2017). URL http://dx.doi.org/10.1080/01621459.2017.1285773.

[35] Zhang, C. & Matsen IV, F. A. Variational Bayesian phylogenetic inference. In *International Conference on Learning Representations* (2019). URL https://openreview.net/forum?id=SJVmjjR9FX.

[36] Dang, T. & Kishino, H. Stochastic variational inference for Bayesian phylogenetics: a case of CAT model. *Mol Biol Evol* **36**, 825–833 (2019).

[37] Fourment, M. & Darling, A. E. Evaluating probabilistic programming and fast variational Bayesian inference in phylogenetics. *PeerJ* **7**, e8272 (2019).

[38] Zhang, C. Improved variational Bayesian phylogenetic inference with normalizing flows. *arXiv* (2020). URL https://arxiv.org/abs/2012.00459.

[39] Zhang, C. & Matsen, F. A. A variational approach to Bayesian phylogenetic inference. *arXiv* (2022). URL https://arxiv.org/abs/2204.07747.

[40] Zhang, C. Learnable topological features for phylogenetic inference via graph neural networks. *arXiv* (2023). URL https://arxiv.org/abs/2302.08840.

[41] Xie, T., Matsen, F. A., Suchard, M. A. & Zhang, C. Variational Bayesian phylogenetic inference with semi-implicit branch length distributions. *arXiv* (2024). URL https://arxiv.org/abs/2408.05058.

[42] Koptagel, H., Kviman, O., Melin, H., Safinianaini, N. & Lagergren, J. VaiPhy: a variational inference based algorithm for phylogeny. *arXiv* (2022). URL https://arxiv.org/abs/2203.01121.

[43] Kviman, O., Molén, R. & Lagergren, J. Improved variational Bayesian phylogenetic inference using mixtures. *arXiv* (2023). URL https://arxiv.org/abs/2310.00941.

[44] Mimori, T. & Hamada, M. GeoPhy: Differentiable phylogenetic inference via geometric gradients of tree topologies. *arXiv* (2023). URL https://arxiv.org/abs/2307.03675.

[45] Macaulay, M. & Fourment, M. Differentiable phylogenetics via hyperbolic embeddings with Dodonaphy. *Bioinf Adv* **4**, vbae082 (2024).

[46] Bouckaert, R. R. Variational Bayesian phylogenies through matrix representation of tree space. *PeerJ* **12**, e17276 (2024). URL http://dx.doi.org/10.7717/peerj.17276.

[47] Hotti, A., Kviman, O., Molén, R., Elvira, V. & Lagergren, J. Efficient mixture learning in black-box variational inference. *arXiv* (2024). URL https://arxiv.org/abs/2406.07083.

[48] Chen, A. *et al.* Variational combinatorial sequential Monte Carlo for Bayesian phylogenetics in hyperbolic space. *arXiv* (2025). URL https://arxiv.org/abs/2501.17965.

[49] Saitou, N. & Nei, M. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *J Mol Biol* **4**, 406–425 (1987).

[50] Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. *arXiv* (2013). URL https://arxiv.org/abs/1312.6114.

[51] Doersch, C. Tutorial on variational autoencoders. *arXiv* (2016). URL https://arxiv.org/abs/1606.05908.

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

[53] Jukes, T. H. & Cantor, C. R. Evolution of protein molecules. In Munro, H. (ed.) *Mammalian Protein Metabolism*, 21–132 (Academic Press, 1969).

[54] Robinson, D. & Foulds, L. Comparison of phylogenetic trees. *Mathematical Biosciences* **53**, 131–147 (1981). URL http://dx.doi.org/10.1016/0025-5564(81)90043-2.

[55] Jones, M. G. *et al.* Inference of single-cell phylogenies from lineage tracing data using Cassiopeia. *Genome Biol* **21**, 92 (2020).

[56] Feng, J. *et al.* Estimation of cell lineage trees by maximum-likelihood phylogenetics. *Ann Appl Stat* **15**, 343–362 (2021).

[57] Prillo, S., Ravoor, A., Yosef, N. & Song, Y. S. ConvexML: Fast and accurate branch length estimation under irreversible mutation models, illustrated through applications to CRISPR/Cas9-based lineage tracing. *Syst Biol* syaf054 (2025). URL http://dx.doi.org/10.1093/sysbio/syaf054.

[58] Sokal, R. R. & Michener, C. D. A statistical method for evaluating systematic relationships. *Univ Kans Sci Bull* **38**, 1409–1438 (1958).

[59] Sashittal, P., Schmidt, H., Chan, M. & Raphael, B. J. Startle: A star homoplasy approach for CRISPR-Cas9 lineage tracing. *Cell Syst* **14**, 1113–1121 (2023).

[60] Lakner, C., van der Mark, P., Huelsenbeck, J. P., Larget, B. & Ronquist, F. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Syst Biol* **57**, 86–103 (2008). URL http://dx.doi.org/10.1080/10635150801886156.

[61] Macaulay, M., Darling, A. & Fourment, M. Fidelity of hyperbolic space for Bayesian phylogenetic inference. *PLoS Comput Biol* **19**, e1011084 (2023). URL http://dx.doi.org/10.1371/journal.pcbi.1011084.

[62] Hadfield, J. *et al.* Nextstrain: Real-time tracking of pathogen evolution. *Bioinformatics* **34**, 4121–4123 (2018). URL http://dx.doi.org/10.1093/bioinformatics/bty407.

[63] Quinn, J. J. *et al.* Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts. *Science* **371** (2021).

[64] El-Kebir, M., Satas, G. & Raphael, B. J. Inferring parsimonious migration histories for metastatic cancers. *Nat Genet* **50**, 718–726 (2018).

[65] Kumar, S. *et al.* Pathfinder: Bayesian inference of clone migration histories in cancer. *Bioinformatics* **36**, i675–i683 (2020).

[66] Koyyalagunta, D., Ganesh, K. & Morris, Q. Inferring cancer type-specific patterns of metastatic spread using metient. *Nature Methods* (2025).

[67] Roddur, M. S. *et al.* Characterizing the solution space of migration histories of metastatic cancers with MACH2. *bioRxiv* (2024).

[68] Simeonov, K. P. *et al.* Single-cell lineage tracing of metastatic cancer reveals selection of hybrid EMT states. *Cancer Cell* **39**, 1150–1162 (2021).

[69] Yang, D. *et al.* Lineage tracing reveals the phylodynamics, plasticity, and paths of tumor evolution. *Cell* **185**, 1905–1923 (2022).

[70] Serio, R. N. *et al.* Clonal lineage tracing with somatic delivery of recordable barcodes reveals migration histories of metastatic prostate cancer. *Cancer Discov* **14**, 1990–2009 (2024).

[71] Wang, K. *et al.* Phylovelo enhances transcriptomic velocity field mapping using monotonically expressed genes. *Nature Biotechnology* **42**, 778–789 (2023). URL http://dx.doi.org/10.1038/s41587-023-01887-5.

[72] Chroni, A. & Kumar, S. Tumors are evolutionary island-like ecosystems. *Genome Biol Evol* **13** (2021).

[73] Fabreti, L. G. & Höhna, S. Convergence assessment for Bayesian phylogenetic analysis using MCMC simulation. *Methods in Ecology and Evolution* **13**, 77–90 (2021). URL http://dx.doi.org/10.1111/2041-210X.13727.

[74] Ayres, D. L. *et al.* BEAGLE 3: Improved performance, scaling, and usability for a high-performance computing library for statistical phylogenetics. *Syst Biol* **68**, 1052–1061 (2019). URL http://dx.doi.org/10.1093/sysbio/syz020.

[75] Zhang, C., Bütepage, J., Kjellström, H. & Mandt, S. Advances in variational inference. *IEEE Trans Pattern Anal Mach Intell* **41**, 2008–2026 (2018).

[76] Stamatakis, A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* **30**, 1312–1313 (2014). URL http://dx.doi.org/10.1093/bioinformatics/btu033.

[77] Nguyen, L.-T., Schmidt, H. A., von Haeseler, A. & Minh, B. Q. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol Biol Evol* **32**, 268–274 (2014). URL http://dx.doi.org/10.1093/molbev/msu300.

[78] Elias, I. & Lagergren, J. Fast neighbor joining. *Theor Comput Sci* **410**, 1993–2000 (2009). URL http://dx.doi.org/10.1016/j.tcs.2008.12.040.

[79] Hutchinson, M. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Commun Stats—Simulation and Computation* **19**, 433–450 (1990). URL http://dx.doi.org/10.1080/03610919008812866.

[80] Keller-Ressel, M. & Nargang, S. Hydra: a method for strain-minimizing hyperbolic embedding of network- and distance-based data. *Journal of Complex Networks* **8**, cnaa002 (2020). URL http://dx.doi.org/10.1093/comnet/cnaa002.

[81] Kingma, D. & Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2015).

[82] Tavaré, S. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences* **17**, 57–86 (1986).

[83] Yang, Z. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *J Mol Evol* **39**, 306–314 (1994). URL http://dx.doi.org/10.1007/BF00160154.

[84] Moreno, M. A., Holder, M. T. & Sukumaran, J. DendroPy 5: a mature python library for phylogenetic computing. *Journal of Open Source Software* **9**, 6943 (2024). URL http://dx.doi.org/10.21105/joss.06943.

[85] Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. & Bürkner, P.-C. Rank-normalization, folding, and localization: an improved $\hat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis* **16** (2021). URL http://dx.doi.org/10.1214/20-BA1221.

[86] Rambaut, A., Drummond, A. J., Xie, D., Baele, G. & Suchard, M. A. Posterior summarization in Bayesian phylogenetics using Tracer 1.7. *Syst Biol* **67**, 901–904 (2018). URL http://dx.doi.org/10.1093/sysbio/syy032.

[87] Fitch, W. M. Toward defining the course of evolution: Minimum change for a specific tree topology. *Syst Zool* **20**, 406–416 (1971).

[88] Hartigan, J. A. Minimum mutation fits to a given tree. *Biometrics* **29**, 53–65 (1973).

[89] Felsenstein, J. *Inferring Phylogenies* (Sinauer Associates, 2004).

[90] Manning, C. D., Raghavan, P. & Schütze, H. *Introduction to Information Retrieval* (Cambridge University Press, 2008).

[91] Blomberg, S. P., Garland, T. & Ives, A. R. Testing for phylogenetic signal in comparative data: Behavioral traits are more labile. *Evolution* **57**, 717–745 (2003). URL http://dx.doi.org/10.1111/j.0014-3820.2003.tb00285.x.

[92] Wilson, I. J. & Balding, D. J. Genealogical inference from microsatellite data. *Genetics* **150**, 499–510 (1998). URL http://dx.doi.org/10.1093/genetics/150.1.499.

[93] Yang, Z. & Rodríguez, C. E. Searching for efficient Markov chain Monte Carlo proposal kernels. *Proc Natl Acad Sci USA* **110**, 19307–19312 (2013). URL http://dx.doi.org/10.1073/pnas.1311790110.

[94] Thawornwattana, Y., Dalquen, D. & Yang, Z. Designing simple and efficient Markov chain Monte Carlo proposal kernels. *Bayesian Analysis* **13**, nil (2018). URL http://dx.doi.org/10.1214/17-BA1084.

[95] Douglas, J., Zhang, R. & Bouckaert, R. Adaptive dating and fast proposals: Revisiting the phylogenetic relaxed clock model. *PLoS Comput Biol* **17**, e1008322 (2021). URL http://dx.doi.org/10.1371/journal.pcbi.1008322.

[96] Hubisz, M. J., Pollard, K. S. & Siepel, A. PHAST and RPHAST: Phylogenetic analysis with space/time models. *Brief Bioinf* **12**, 41–51 (2011).

Figures

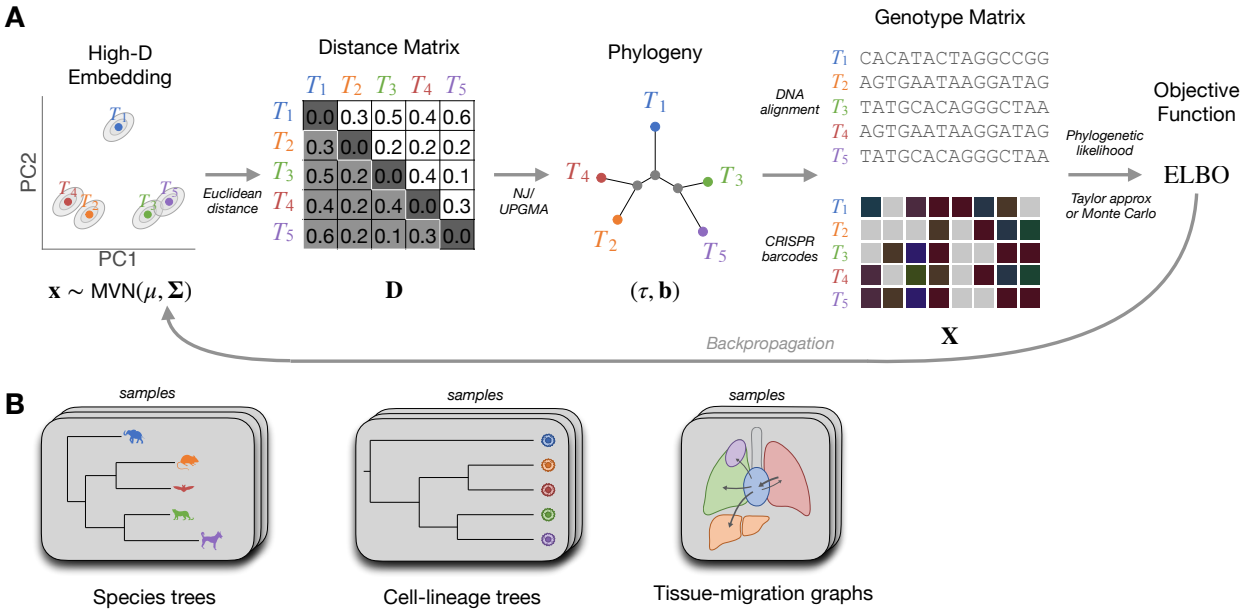

Figure 1: **(A)** Flow of information in VINE from high-dimensional embedding of taxa to Evidence Lower Bound (ELBO) for variational inference, followed by flow in the reverse direction via backpropagation. Each of n taxa is represented as a point $\mathbf{x}$ in a d-dimensional space, which is sampled from a multivariate normal distribution, $\mathrm{MVN}(\boldsymbol{\mu}, \mathbf{\Sigma})$. A distance matrix $\mathbf{D}$ is computed from $\mathbf{x}$ and then converted by neighbor-joining (NJ) or UPGMA to a tree τ with branch lengths $\mathbf{b}$, allowing calculation of a phylogenetic likelihood from a genotype matrix $\mathbf{X}$, which may be a DNA alignment or a CRISPR-barcode mutation matrix. Finally, the ELBO is computed using either a Taylor approximation or Monte Carlo sampling. The gradient of the ELBO with respect to $(\boldsymbol{\mu}, \mathbf{\Sigma})$ is computed by backpropagation through the component transformations, allowing for efficient optimization by stochastic gradient ascent (SGA). **(B)** Three applications of interest: inference of species trees (*left*), cell-lineage trees (*middle*), or tissue-migration graphs (*right*). In each case, samples from the approximate posterior distribution are obtained by sampling values of $\mathbf{x}$ from the optimized MVN distribution and transforming them as in (A).

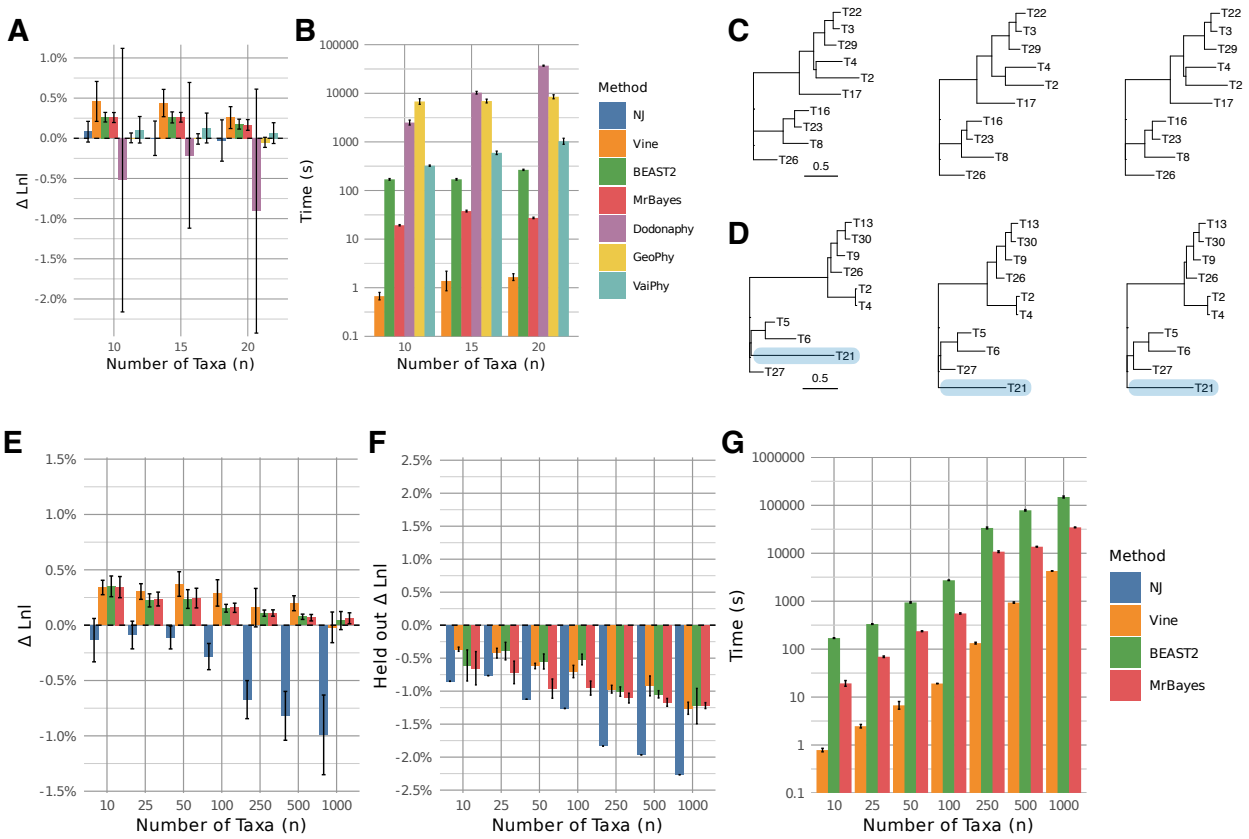

Figure 2: Performance of VI and MCMC-based methods on simulated DNA datasets. **(A)** Maximized log likelihood during model fitting, relative to the log likelihood of the true (generating) model (zero line), for small numbers of taxa ($n \leq 20$) under the Jukes-Cantor substitution model. **(B)** Compute time required when running without parallelization or GPU acceleration on an HPE ProLiant DL380 Gen10 server (see **Methods**), in seconds per replicate (note log scale). **(C)** Reconstruction of a simulated 10-taxon tree (*left*) by VINE (*center*) and BEAST 2 (*right*). **(D)** A second example with a reconstruction error. Horizontal branch lengths are drawn to scale in substitutions per site. For VINE and BEAST 2, samples from the posterior distributions are summarized by maximum-clade-credibility (MCC) trees (using TreeAnnotator [22]). **(E)** Maximized log likelihood relative to the true model for larger numbers of taxa (up to $n = 1000$) under the HKY substitution model. **(F)** Average log likelihood across posterior samples for held-out data, also relative to the true model. **(G)** Compute time required. Results shown are for 300 bp alignments, with ten replicates per bar. Error bars represent one standard deviation. VINE (*orange*) was applied with a Euclidean geometry, the Taylor approximation, and the CONST variance parameterization without normalizing flows (see **Methods**). Dimensionality varied from $d = 5$ for $n = 10$ to $d = 10$ for $n = 1000$. See additional results in **Supplementary Figs. S1 & S3**).

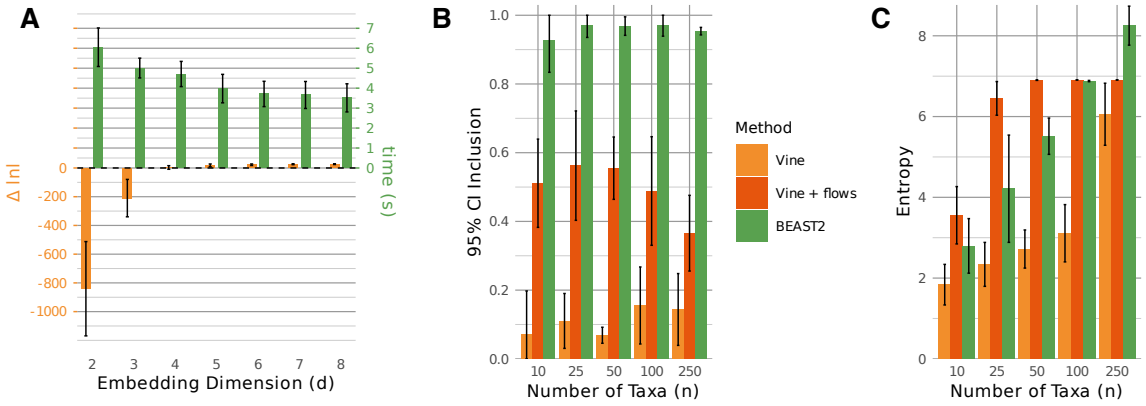

Figure 3: **(A)** Performance improvement of VINE with increasing dimensionality d of the embedding space. Shown are running times per replicate (*green*) and deviations of the maximized log likelihood from that of the true model (*orange*). Results are for $n = 25$ taxa, alignments of 300 bp, and estimation under the HKY model with the CONST variance parameterization, a Euclidean geometry, and the Taylor approximation without normalizing flows (see also **Supplementary Fig. S4**). **(B)** Accuracy of posterior distributions, as measured by the fraction of all pairwise distances between taxa that fall within the estimated 95% credible interval (95% CI Inclusion). **(C)** Topological entropy of approximate posteriors, defined as the Shannon entropy of distinct topologies (see **Methods**). In (B) and (C), results are shown for simulated trees of $n \in \{10, 25, 50, 100, 250\}$ taxa for the baseline version of VINE (VINE), the best version of VINE (VINE + flows), and BEAST 2. Bars represent averages over ten replicates. The baseline version of VINE uses the CONST variance parameterization, no variance regularization, and no normalizing flows. The best version uses the DIST parameterization, a variance regularization multiplier of three (`--var-reg 3`), and both the radial and planar flows. A Euclidean geometry and the Taylor approximation were used in both cases.

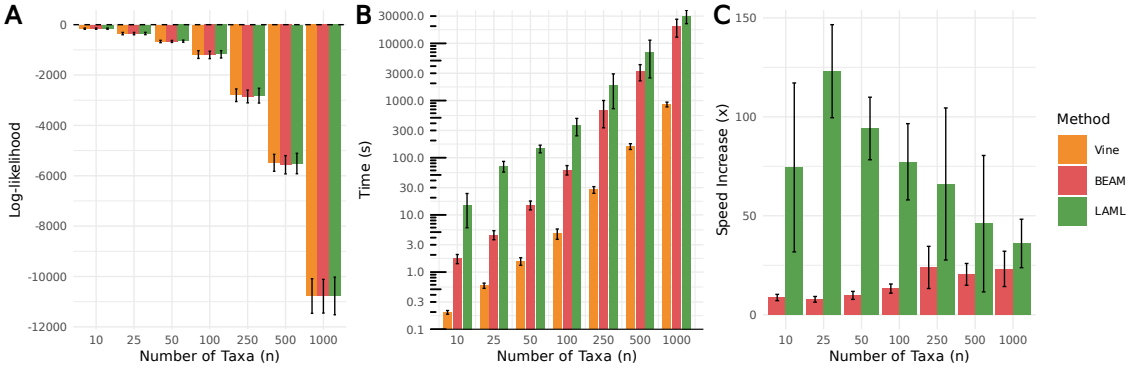

Figure 4: Comparison of VINE with LAML [31] and BEAM [27] on simulated CRISPR-barcoding data, for various numbers of taxa n. **(A)** Maximized log likelihood during model fitting. **(B)** Compute time per replicate. **(C)** Speed increase of VINE relative to LAML and BEAM. Results are for 10 simulated datasets for each value of n with 30 barcode sites and editing parameters based on real data [63] (see **Methods**). Error bars represent one standard deviation.

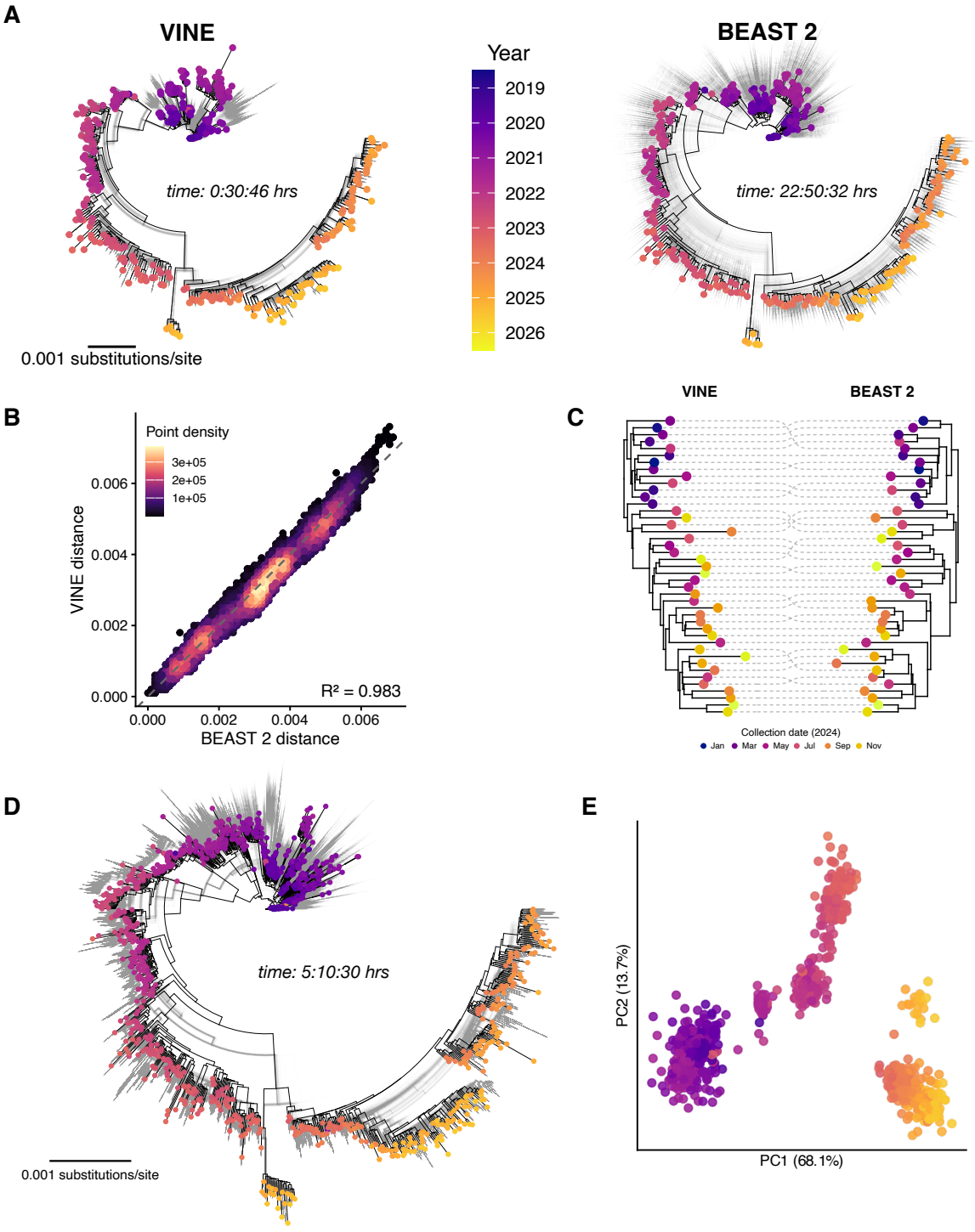

Figure 5: **(A)** Maximum-clade-credibility (MCC) trees inferred by VINE and BEAST 2 for 364 randomly selected SARS-CoV-2 genomes from Nextstrain with tips colored by collection date. Clouds of trees in gray represent posterior samples. **(B)** BEAST 2 vs. VINE posterior mean estimates of pairwise distances in substitutions per site for all pairs of taxa in (A). **(C)** Aligned MCC subtrees for samples from 2024, showing broad topological agreement with minor differences. **(D)** MCC tree and posterior cloud inferred by VINE for a larger 1030-taxon SARS-CoV-2 data set. **(E)** First two principal components for the embedding learned by VINE for the tree in (D).

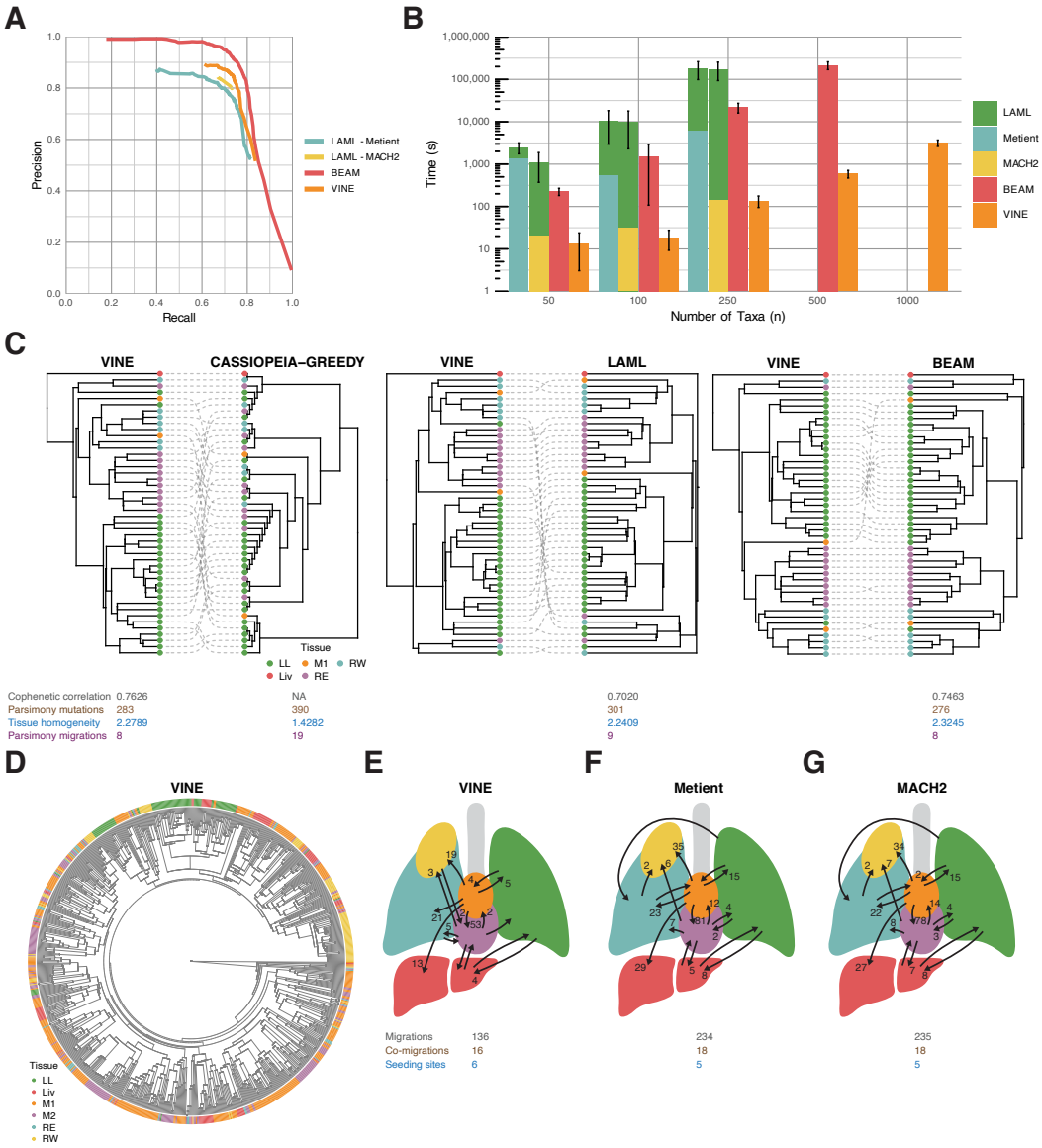

Figure 6: Results from running VINE in migration mode on CRISPR-based lineage-tracing data. **(A)** Precision vs. recall for individual edges of simulated tissue-migration graphs (as detailed in [27]) relative to Metient [66], MACH2 [67], and BEAM [27]. Metient and MACH2 used input trees from LAML. **(B)** Compute time per replicate on simulated data (log scale). Additional time to run LAML [31] is shown (*green*) for Metient and MACH2. Some methods were omitted for $n \geq 500$ owing to time constraints. **(C)** Comparison of tissue-labeled tree inferred by VINE for CP70 from ref. [63] with trees inferred by Cassiopeia-Greedy [55], LAML, and BEAM. Shown for each tree are the cophenetic correlation (*gray*), number of mutations by parsimony (*brown*), tissue homogeneity (*blue*), and number of migrations by parsimony (*violet*) (see **Methods** and **Supplementary Fig. S12**). **(D)** Tissue-labeled MCC tree inferred by VINE for CP4 from ref. [63], comprising 904 distinct barcode sequences across six tissues. **(E–G)** Tissue-migration graphs for CP4 inferred by **(E)** VINE (at >0.9 posterior probability), **(F)** Metient (the best scoring of three solutions), and **(G)** MACH2 (the first of three solutions). Shown for each graph are corresponding numbers of migrations (*gray*), co-migrations (*brown*), and seeding sites (*blue*) (see **Methods**, **Supplementary Figs. S13&S14**). Tissue colors match (D). LL, left lung; Liv, liver; M1, mediastinum 1; M2, mediastinum 2; RE, right lung E; RW, right lung W (see [63]).