Geodesics to Characterize the Phylogenetic Landscape

Marzieh Khodaei[1,*], **Megan Owen**[2], **and Peter Beerli**[1]

[1] Department of Scientific Computing, Florida State University, Tallahassee, FL 32306, USA

[2] Department of Mathematics, Lehman College and Graduate Center, CUNY, NY 10468, USA

*Corresponding author: Marzieh Khodaei, email: mk16e@fsu.edu

Abstract

Phylogenetic trees are fundamental for understanding evolutionary history. However, finding maximum likelihood trees is challenging due to the complexity of the likelihood landscape and the size of tree space. Based on the Billera-Holmes-Vogtmann (BHV) distance between trees, we describe a method to generate intermediate trees on the shortest path between two trees, called pathtrees. These pathtrees give a structured way to generate and visualize treespace in an area of interest. They allow investigating intermediate regions between trees of interest, exploring locally optimal trees in topological clusters of treespace, and potentially finding trees of high likelihood unexplored by tree search algorithms. We compared our approach against other tree search tools (Paup*, RAxML, and RevBayes) in terms of generated highest likelihood trees, new topology proportions, and consistency of generated treespace. We assess our method using two datasets. The first consists of 23 primate species (CytB, 1141 bp), leading to well-resolved relationships. The second is a dataset of 182 milksnakes (CytB, 1117 bp), containing many similar sequences and complex relationships among individuals. Our method visualizes the treespace using log likelihood as a fitness function. It finds similarly optimal trees as heuristic methods and presents the likelihood landscape at different scales. It revealed that we could find trees that were not found with MCMC methods. The validation measures indicated that our method performed well mapping treespace into lower dimensions. Our method complements heuristic search analyses, and the visualization allows the inspection of likelihood terraces and exploration of treespace areas not visited by heuristic searches.

Key words: Phylogenetic treespace; Billera-Holmes-Vogtmann treespace; likelihood; convex hull; pathtrees; optimal tree; multidimensional scaling; interpolation

Introduction

Evolutionary trees, or phylogenetic trees, have been used extensively throughout systematic biology and other fields to represent the evolutionary history of species. How to compute the best tree and how to characterize the uncertainty of estimates of the branch lengths and the topology is an on-going challenge. Generally tree search methods seek a globally best tree under some optimization criteria (e.g. parsimony (Fitch 1971), distance methods (Fitch and Margoliash 1967; Saitou and Nei 1987), or maximum likelihood (Felsenstein 1981)), but the number of potential trees grows exponentially relative to the number of leaves (Schröder 1870). Furthermore, it is NP-hard to compute the maximum likelihood tree (Roch 2006; Chor and Tuller 2005) or most parsimonious tree (Foulds and Graham 1982), so except for small numbers of taxa where exhaustive search is possible, heuristic methods must often be used to search the treespace. These methods explore locally best trees in the hope that the best local tree found is equivalent to the global best tree (e.g. maximum likelihood programs RAxML (Stamatakis et al. 2005), PhyML (Guindon and Gascuel 2003), and Paup* (Swofford 2003)). Similarly, Bayesian inference programs use tree rearrangement moves to generate proposals to estimate the posterior probability (e.g. Bayesian inference programs MrBayes (Huelsenbeck and Ronquist 2001), RevBayes (Höhna et al. 2016), and BEAST (Drummond and Rambaut 2007)).

Initially, many optimality criteria to compare trees were developed, the most prominent being parsimony (Fitch 1971), distance methods (Fitch and Margoliash 1967; Saitou and Nei 1987), and maximum likelihood (Felsenstein 1981). Computational power has increased considerably since then, so computationally intense probabilistic methods such as maximum likelihood and Bayesian inference have supplanted faster but less accurate methods. Researchers now commonly use maximum likelihood programs, such as RAxML (Stamatakis et al. 2005), PhyML (Guindon and Gascuel 2003), and Paup* (Swofford 2003), or Bayesian inference programs, such as MrBayes (Huelsenbeck and Ronquist 2001), RevBayes (Höhna et al. 2016), and BEAST (Drummond and Rambaut 2007).

Both Bayesian inference and maximum likelihood methods use random changes in the tree topology to search the treespace. Even when we record all visited trees in a maximum likelihood search or look at all collected trees in a Bayesian Markov chain Monte Carlo run, we can see a "gappy" space, where trees on the space were not evenly spaced: some regions are more concentrated and other regions are sampled sparsely or not at all. For example, an MCMC may not explore the space or posterior distribution efficiently due to revisiting trees topologies (Lakner et al. 2008) or due to low posterior nodes separating peaks (Whidden and Masten 2015). While in an ML tree search, Money and Whelan (2012) show that different rearrangement moves correspond to discretized treespaces with different numbers of local optima.

While the concept of treespace is often used informally to mean the set of all possible phylogenetic

trees meeting some condition, such as having n leaves, a treespace can be formally defined as a discrete or continuous metric space (ie. a geometric space with a distance measure between points meeting certain conditions) where each point corresponds to a tree (see St. John (2017) for a comprehensive review of treespaces). In this paper, we use the Billera-Holmes-Vogtmann (BHV) treespace (Billera et al. 2001), which is a continuous, piece-wise Euclidean space containing all trees with branch lengths and n leaves. This space contains unique shortest paths, or geodesics, between any two points, with the lengths of these paths being the Billera-Holmes-Vogtmann (BHV) distance. Both the BHV distance and geodesics between trees can be computed in polynomial time (Owen and Provan 2011). The weighted Robinson-Foulds distance (wRF) (Robinson and Foulds 1979) corresponds to using an L_1 metric instead of an L_2 metric on the piecewise-Euclidean orthants of BHV treespace. The wRF distance does not have unique geodesics but is faster to compute than the BHV distance and is at most a multiplicative factor of $\sqrt{2}$ larger than the BHV distance (Amenta et al. 2007). The Robinson-Foulds (RF) distance (Robinson and Foulds 1981) is the same as the weighted Robinson-Foulds distance when all edge lengths are set to be 1.

A landscape is a configuration space or a metric space of trees (with or without branch lengths) $\mathcal{T}$ and an associated real-valued cost or fitness function $f : \mathcal{T} \to \mathbb{R}$. Landscapes were first defined by Bastert et al. (2002) on trees without branch lengths, so the metric space $\mathcal{T}$ was a graph. A landscape on trees without branch lengths can be visualized as a colored graph, where the trees are the nodes, colored by the fitness function value, and edges represent a minimal rearrangement move between trees, such as Subtree-Prune and Regraft (SPR) (Whidden and Masten 2015). Alternatively, for trees with or without branch lengths, distances can be computed between the trees, and visualized in 2 or 3 dimensions using Multi-Dimensional Scaling (MDS). MDS approximates the pairwise distances between points by mapping them in a lower-dimensional Euclidean space (Kruskal 1964), and was first applied to sets of phylogenetic trees by Amenta and Klingner (2002) and popularized by Hillis et al. (2005). Most often, the trees in an MDS visualization have been colored to indicate some kind of clustering (e.g. Hillis et al. (2005); Kendall and Colijn (2016); Gori et al. (2016); Jombart et al. (2017)), but they can also be colored by the fitness function to visualize landscapes. For example, Hillis et al. (2005) included a Bayesian MCMC trace with RF distances colored by likelihood; Höhna and Drummond (2012) used the Nearest-Neighbor Interchange (NNI) distance colored by posterior probability with trees separated by a single NNI move joined by an edge; and Wright and Lloyd (2020) using RF distance colored by the minimum implied gap (MIG) score, a measure of congruence with the fossil record, and included an MDS visualization with interpolation of the MIG score across the MDS space.

We use MDS to visualize landscapes where the fitness function is the log likelihood, but approach the visualization in a different way from previous work. For a given area of interest in treespace, we sample trees along geodesics crossing this area to get a representative set of trees ("pathtrees"). We then use MDS to map

these pathtrees into 2 dimensions, color them by their log likelihood, and use interpolation to estimate the likelihood landscape in the treespace area of interest. We also show the landscape function as a 3-D surface over the 2-D MDS plot and call this a likelihood surface. Thus, our visualizations try to illustrate the overall landscape tendency in the part of treespace of interest, instead of only point values of the fitness function, possibly unevenly distributed.

There are several recent packages or programs for visualizing trees using MDS, but they focus on coloring the points by cluster rather than a fitness function. TREESCAPER (Huang et al. 2016) is a standalone GUI that allows different tree distance functions, cost functions for dimensionality reduction, and non-linear dimension reduction algorithms to be used. TREESPACE (Jombart et al. 2017) is an R package that allows a wide variety of tree metrics and methods for clustering trees to be used. Smith (2022) evaluated the performance of multiple aspects of low-dimensional representations of sets of trees, and provides an R package TreeDist for users to do the same. Finally, R We There Yet (RWTY) (Warren et al. 2017), a package for analyzing Bayesian analyses convergence, can produce nonlinear MDS visualization of landscapes using the RF or path difference distance (Steel and Penny 1993) and colored by the likelihood. Some authors (Amenta et al. 2015; Wilgenbusch et al. 2017; Smith 2022) have analyzed how well MDS visualizes treespace, and suggested validation measures. There are other ways to visualize sets of related trees beyond dimensionality reduction, such as super-imposing the trees on each other, as in DensiTree (Bouckaert 2010), or sophisticated tree comparison visualizers, like ADView (Liu et al. 2019).

We will focus on likelihood as the optimality criterion for the rest of this paper. While there is a closed-form expression to compute the likelihood of a given tree, given sequence data, the likelihood function itself is very complex with multiple local optima (Steel 1994; Chor et al. 2000). Finding the best tree in the presence of multiple local and global optima and the presence of regions of trees with similar, high likelihood, such as islands (Maddison 1991; Salter 2001) and terraces (Sanderson et al. 2011; 2015) is difficult. Our method can deliver additional support for other heuristic methods by investigating the relationship among trees in the BHV space and visualizing the landscape at different scales in an area of interest in treespace using MDS. We apply our method to two datasets, and discover novel high likelihood tree topologies. We distribute our approach in the Python package PATHTREES.

1 Materials and Methods

We developed a method to generate and visualize the log-likelihood landscape in an area of interest in a treespace, and potentially find trees of high likelihood unexplored by tree search algorithms. We sample trees ("pathtrees") along the shortest paths in BHV treespace between points on a convex hull enclosing the area

of interest, and also compute the optimized branch lengths for each tree topology found. We then use MDS to map all of these trees into two dimensions and use splines to interpolate the log-likelihood in the spaces between trees to visualize the relationship of the trees and their landscape. For the rest of this paper, we use "treespace" to refer to the Billera-Holmes-Vogtmann (BHV) treespace, unless otherwise specified.

1.1 Billera-Holmes-Vogtmann (BHV) treespace and shortest paths

The Billera-Holmes-Vogtmann (BHV) treespace models all phylogenetic trees with a fixed set of leaves. It is formed from a set of Euclidean regions, called orthants. Each orthant contains only trees with the same topology. Each tree topology consists of a unique set of splits (Buneman 1971), and each of these splits is assigned to one of the dimensions of the orthant. Each branch length in a tree becomes the coordinate in the orthant along the dimension corresponding to the branch's split. Two orthants with corresponding tree topologies that share splits are adjacent in the treespace, and their shared boundary orthant contains all trees containing exactly the shared splits. All orthants contain the origin, which corresponds to the star tree, so the space is connected.

The length of a path between two trees is computed by measuring the Euclidean length of the path in each orthant it passes through, and summing those lengths. There is a unique shortest path, or geodesic, connecting two phylogenetic trees T_1 and T_2 in BHV treespace (Billera et al. 2001), and it can be computed in polynomial time $O(n^4)$, where n is the number of leaves in the trees, by the Geodesic Treepath Problem (GTP) algorithm (Owen and Provan 2011).

If trees T_1 and T_2 have no common splits, then the GTP algorithm starts with a simple initial path, called the cone path, which connects trees T_1 and T_2 to the origin (a star tree), and hence each other, by straight lines. Then, the cone path is transformed into a series of successively shorter paths until the geodesic is obtained. At each step, the algorithm identifies one new orthant that the current path can be modified to pass through to yield a shorter path. When trees T_1 and T_2 have splits in common, the algorithm first subdivides T_1 and T_2 along the common splits, and runs the GTP algorithm described above on each pair of subtrees. The shortest paths between the subtrees are then combined into the overall geodesic between T_1 and T_2. Fig. 1 demonstrates an example of a geodesic between two trees, and the geometric representation of treespace. The top row shows a starting tree (T_1), an ending tree (T_2), and the two trees where the geodesic between T_1 and T_2 crosses orthant boundaries. The bottom row shows the geodesic, the cone path, parts of the three orthants that the geodesic passes through, and an example tree of each orthant. Moving along the geodesic from start tree T_1 to end tree T_2, the intermediate tree branches shrink to zero length at orthant boundaries, and new branches begin to grow.

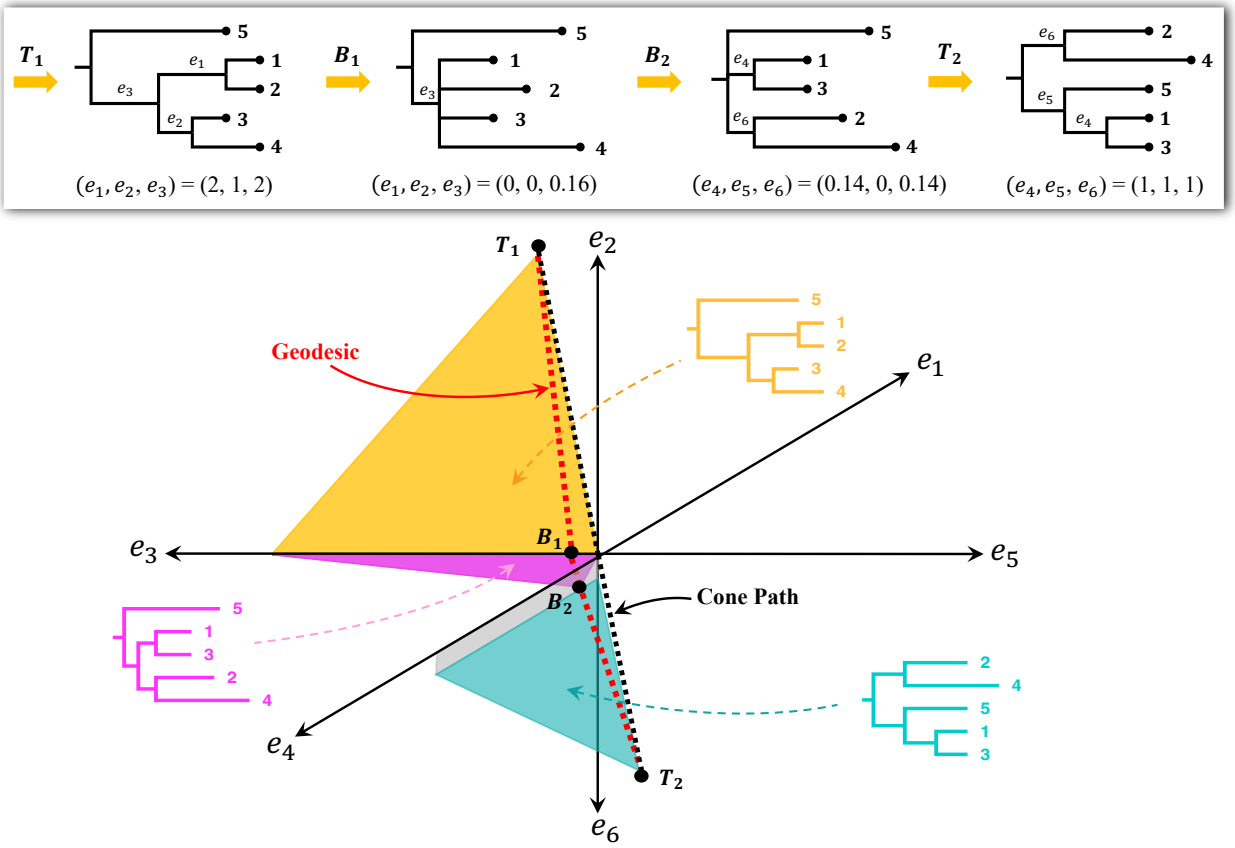

Figure 1. Shortest path (geodesic) between two trees in the BHV treespace. First row: a start tree T_1 and an end tree T_2 with two intermediate trees B_1 and B_2 on the boundaries of orthants, axis e_3 and the quarter-plane formed by axes e_4 and e_6, respectively. Bottom row: Black dots mark the trees in the top row; the cone path is the black dotted line; the geodesic is the red dotted line; each colored region is part of a different orthant containing the geodesic; and an example tree with arbitrary branch lengths is shown for each orthant.

1.2 Sampling trees on the shortest path between tree pairs

Our method samples trees along the shortest paths (geodesics) between points on the boundary of our area of interest. The topologies and edge lengths for these sampled trees on the geodesic between trees T_1 and T_2 are given by Theorem 2.4 of Owen and Provan (2011) and Theorem 1.2 of Miller et al. (2015). First let $(A_1, A_2, ..., A_k)$ be a partition of the splits in T_1 that are not in T_2, where A_i is the set of splits whose branches shrink to zero length at the i-th orthant boundary along the geodesic. Let $(B_1, B_2, ..., B_k)$ be a partition of the splits of T_2 that are not in T_1, where B_i is the set of splits whose branches begin growing from zero length at the i-th orthant boundary. Let C be the set of splits common to T_1 and T_2, and parameterize the geodesic between T_1 and T_2 by $0 \leq \lambda \leq 1$. For split e in tree T, denote its branch length in T by $|e|_T$ and for the set of splits S in tree T, let $||S|| = \sqrt{\sum_{e \in S} |e|_T^2}$. Then by Theorem 2.4 of Owen and Provan (2011) and Theorem 1.2 of Miller et al. (2015), a tree T_i on this geodesic at position λ in the i-th orthant contains exactly the splits

$$\mathcal{S} = C \cup B_1 \cup ... \cup B_i \cup A_{i+1} \cup ... \cup A_k, \tag{1}$$

such that split $e \in \mathcal{S}$ has branch length

$$|e|_{T_i} = \begin{cases} \frac{(1-\lambda)||A_j|| - \lambda||B_j||}{||A_j||}|e|_{T_1} & e \in A_j \\ \frac{\lambda||B_j|| - (1-\lambda)||A_j||}{||B_j||}|e|_{T_2} & e \in B_j \\ (1-\lambda)|e|_{T_1} + \lambda|e|_{T_2} & e \in C \end{cases} \tag{2}$$

Thus, the trees along the geodesic between T_1 and T_2 can only contain splits from T_1 and T_2, and any split common to T_1 and T_2 appears in all trees along the geodesic between them. Additionally, as shown in Fig. 1 where the geodesic passes through the quarter plane formed by the axes e_3 and e_4, a geodesic can non-trivially pass through lower-dimensional orthants, corresponding to trees with 0 length edges. Each tree T_i generated along the path is then stored in the standard Newick format. Some trees on the geodesic are not bifurcating trees.

1.3 Finding the starting trees for PATHTREES

Our method visualizes the landscape in an area of interest in treespace. The boundary of this area is defined by a set of trees. We compute the shortest path between each pair in this set of trees, and sample trees along these paths to get the pathtrees. Our package can generate random starting trees, but with a larger number of taxa, these random trees span a very large section of treespace. Instead of random starting trees, we use a large set of trees generated by another method, for example, by RevBayes using Markov chain Monte Carlo. We calculate the BHV distance between these trees and map them to two dimensions using MDS. The trees that are on the vertices of the convex hull, the smallest convex polygon enclosing all of the trees in the 2-D MDS plane, are then extracted and used as the starting trees for Pathtrees. For example, we collected 50,000 trees using the program RevBayes and the primate dataset (outlined in more detail in Section 1.7) and then extracted 1000 trees from the last 1/10 of the MCMC chain. Fig. 2 shows the space of 1000 trees and the trees on the vertices of the corresponding convex hull. The Pathtrees will only depend on the trees on the convex hull. Ideally, we would want to calculate the hull in treespace and not the 2-D MDS space but currently, there is an algorithm to achieve that.

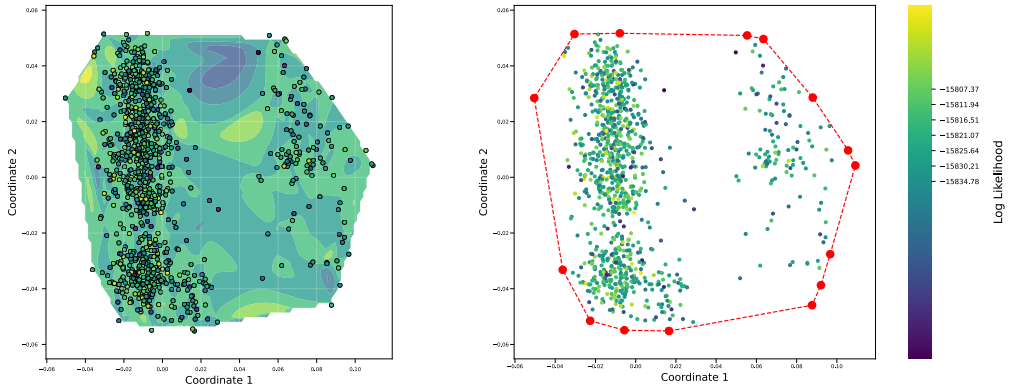

Figure 2. An example of sample treespace and its convex hull. Left: A log-likelihood contour plot of the first two multidimensional scaling (MDS) coordinates of the sampled trees. The log-likelihood contour is a cubic spline interpolation of the log likelihoods of all trees in the MDS plane; the MDS coordinates are computed from the BHV distances between trees. Each dot is a tree; the lighter the dot, the higher the likelihood of the tree. Right: the convex hull of the set of trees. The red dots are the vertices of the convex hull, displaying the sample trees on the boundary of the treespace.

1.4 Visualizing treespace and generating pathtrees

The largest orthants in the BHV treespace for n-leaf rooted trees have dimension $2n-3$, so landscapes on this treespace cannot be visualized directly. Instead, we follow precedent (Amenta and Klingner 2002) and we generate a distance matrix for all N trees we wish to visualize, and use this distance matrix as input into a multidimensional scaling (MDS) algorithm (Cox and Cox 2008) to compress the higher dimensional treespace into two dimensions. The BHV tree space is high dimensional and compressing this space using MDS to 2 dimension may bring unrelated trees close to each other. We evaluated this mapping by comparing the 2-D MDS coordinate distance matrix with the tree distanc matrix using correlation measures, such as Pearson's r. Since we are interested not only in the relationship among the trees but also in how well they fit the data, we calculate the log-likelihood for each tree and add this dimension to the 2-D MDS visualization either as contours or a third coordinate axis. For a smooth representation of the likelihood surface, we interpolate the log-likelihood values between the N trees using an interpolation method. We used two different methods: the cubic spline interpolation method (De Boor 1978) and the radial basis function (RBF) thin-plate spline interpolation (Buhmann 2003). The differences between the two interpolation methods (cubic spline interpolation and the thin-plate spline interpolation) are discussed in Supplementary Section S3.

The MDS procedure is time-consuming for large distance-matrices. We experimented with two different distance metrics for visualization: the BHV distance and the weighted Robinson-Foulds (wRF) distance (Robinson and Foulds 1979). The wRF distance is faster to compute than the BHV distance and is, at most, a multiplicative factor of $\sqrt{2}$ larger than the BHV distance.

The distribution of the sampled trees shown in Fig. 2 highlights that some areas of treespace were sampled less well than others. In contrast, choosing trees along geodesics allow us to visualize trees that are evenly spaced between two arbitrary end-point trees. We demonstrate these pathtrees in Fig. 3 where we selected three trees (colored triangles) from the 1000 sample trees visualized in Fig. 2 and generated 20 trees on the shortest path in the BHV treespace between each pair of them and then visualized the contour and the surface of all 1060 trees in 2-D MDS space. The selected three trees are provided in Supplementary Fig. S1. The pathtrees bridge the gaps between the sampled trees.

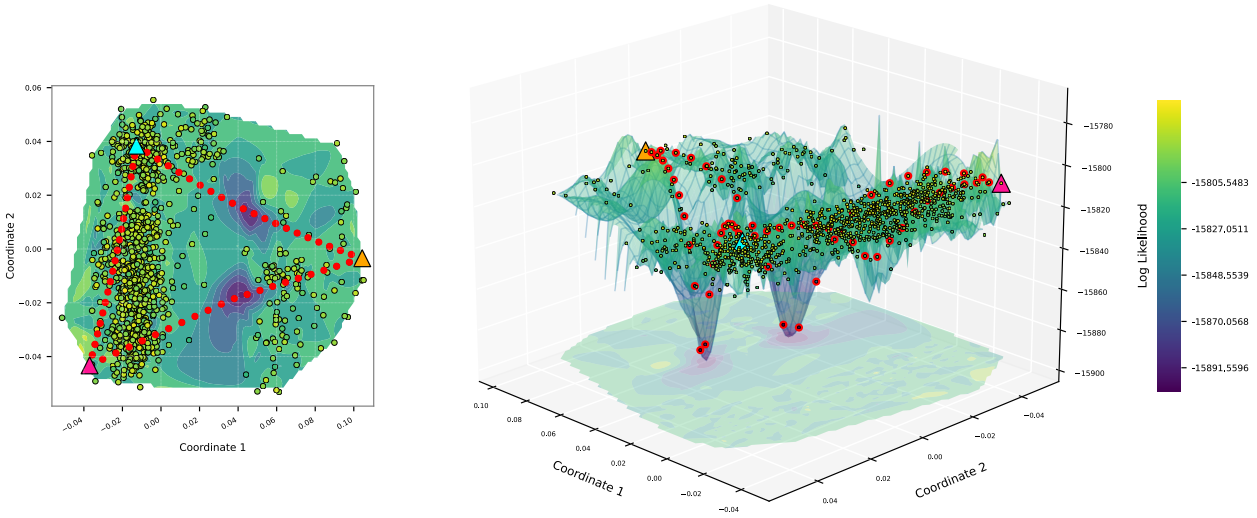

Figure 3. An example of pathtrees between three arbitrary trees in the treespace. Cubic spline interpolation of the log likelihood was used for the contour color (left) and the surface height (right) of the space inside the convex hull. 20 pathtrees (red dots) were generated on the shortest path between each pair of 3 trees (triangles).

1.5 Optimizing the branch-length of pathtrees

The pathtrees lay on the shortest path between a start and end tree (anchor trees). This path is constructed by refinement of the cone path as discussed in Section 1.1, so the resulting pathtrees will more commonly have short branches. To find the best tree and also find potential local likelihood maxima, we need to find pathtrees with the same topology and then optimize the branch lengths for that topology. We apply the unweighted Robinson-Foulds distance (Robinson and Foulds 1981) to detect the different topologies; then we use PAUP* to optimize the branch lengths of a tree in each topology cluster. Fig. 4 shows the locations of the pathtrees and their corresponding optimized trees. If the anchor trees have different topologies, they are located in different orthants. The pathtrees can have different intermediate topologies that, when optimized, will be located away from the shortest path. Unoptimized trees that start in low likelihood areas of treespace may move particularly far away, whereas trees that start on a ridge may not move far.

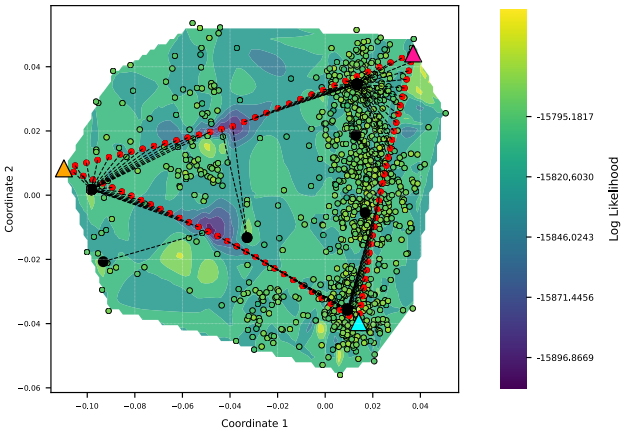

Figure 4. An example of pathtrees and their corresponding optimized trees. Each pathtree is connected to its corresponding optimized tree (black circles).

1.6 Software implementation

Our method is implemented in the Python package PATHTREES. The method uses the Java package GTP (Owen and Provan 2011) to generate the geodesic between pairs of trees, the program PAUP* (Swofford 2003) for likelihood optimization, the Python modules DendroPy (Sukumaran and Holder 2010) for the Robinson-Foulds metric, and several other standard Python modules, such as SCIPY and NUMPY (Virtanen et al. 2020; Harris et al. 2020).

We summarize the tree searching strategy of our package PATHTREES in the algorithm:

Input: Sequence data in PHYLIP format and N rooted, non-ultrametric trees in plain Newick format sampled in connection with the sequence data (e.g. from a MCMC Bayesian analysis chain of the sequence data).

Output: Trees on the shortest paths through treespace between all pairs of starting trees; optimized branch lengths for each different pathtree topology; MDS visualization of treespace and likelihood landscape inside the convex hall of input trees.

Algorithm:

1. Compute all pairwise distances between N sample trees, and compute their MDS coordinates.
2. Extract the trees on the vertices of the convex hull of the N sample trees in the 2-D MDS plane and consider them as starting trees.
3. Generate m equally spaced trees on the geodesic between each pair of starting trees. Put all starting trees and generated pathtrees in the set current sample trees.
4. Calculate the likelihood of all current sample trees.

5. Select n trees with the highest likelihood among current sample trees and classify them with respect to their topologies. Let there be t topologies among them.

6. For each topology cluster, optimize the branch lengths for that topology and add these t optimized trees to the current sample trees.

7. Visualize the contour and surface of the current sample trees by creating a distance matrix using either the weighted Robinson-Foulds or the BHV distance metric, interpolating log-likelihood values, and recomputing the MDS coordinates.

8. Return the current sample trees, the pathtrees, and the visualization; or continue with step 3 to zoom in the area of optimized trees of the current iteration.

1.7 Application to real data

We evaluate our approach PATHTREES using two datasets that were previously published: D_1 is a dataset of primates used in the tutorial for the program REVBAYES (Höhna et al. 2017). The dataset consists of 1141 base pairs of the mitochondrial cytochrome b gene of 23 primate taxa. D_2 is a larger mitochondrial cytochrome b dataset of 182 milksnakes (1117 bp) (Ruane et al. 2014; Chambers and Hillis 2019). We chose the two different datasets because they represent very different situations. D_1 is a relatively small dataset but still too large to consider an exhaustive tree search. The species in the dataset are well defined, and the dataset contains enough variability to establish a phylogeny with branch lengths that are neither zero nor huge. In contrast, D_2 has about eight times more individuals than D_1; these individuals are only from a few species or subspecies, and many individuals share the same DNA sequence with others in the dataset. We deliberately did not attempt to consolidate the dataset into unique site patterns, anticipating that the many zero branch lengths would be a stress test for our method.

For the first dataset D_1, we collected 50,000 trees using the program REVBAYES. We used the instructions from the tutorial of REVBAYES (Höhna et al. 2017), which are shown in Supplementary Section S5. We selected every 38th tree from the last 3/4 of the total 50,000 sample trees (around 1000 sample trees) and then extracted the trees on the vertices of the convex hull of these sample trees (14 trees) as starting trees for PATHTREES. These trees were the starting trees for two experiments: (1) 1 pathtree between each pair of starting trees and (2) a higher number of 15 pathtrees between each pair of starting trees to show a more detailed treespace. We used the BHV distance between trees for MDS for the first experiment and the faster wRF distance for the second experiment to make the computations tractable.

For the second dataset D_2, we collected 10,000 trees using the program REVBAYES. After removing the first 300 trees as burn-in, we selected every 20 trees and extracted 500 trees to be considered in convex hull

analysis. The 14 trees on the vertices of the convex hull were used as starting trees in PATHTREES. For the second dataset, considering that it is a complicated dataset with 182 individuals, we intentionally collected a smaller sample of 10000 trees from REVBAYES, compared to the first dataset, and burned in a small portion of trees just to verify that it is not required to generate a very long chain of REVBAYES trees and large burn-in to get a reasonable treespace to generate starting trees for PATHTREES. We computed 4 pathtrees along the shortest path between each starting tree pair and selected the 100 trees with the highest likelihood to classify by topology for branch length optimization. We then performed a second iteration of our method ("zoomed in") by computing the convex hull of the 100 optimized trees of the first iteration. The vertices of this convex hull became the starting trees for our second iteration, and in this iteration we computed 5 pathtrees along each shortest path between starting tree pairs. For both iterations, we used the wRF distance between trees as input for MDS, and used thin-plate spline interpolation to visualize the likelihood landscape.

1.8 Comparison of PATHTREES with heuristic tree searches

We compared the highest likelihood trees found by PATHTREES with those generated by the maximum likelihood software PAUP* 4.0a (build 168) (Swofford 2003), RAXML 8.2.12 (Stamatakis et al. 2005), and the Bayesian inference software REVBAYES 1.1.1 (Höhna et al. 2016). These programs perform heuristic searches. PAUP* and RAXML search will swap on new tree topologies until a local maximum has been reached and no new tree topologies need further evaluations. These heuristics do not guarantee to recover the global maximum likelihood tree but usually deliver good results (Stamatakis et al. 2005). In contrast, REVBAYES uses Markov chain Monte Carlo to evaluate the posterior probability of a tree while collecting trees along a Markov chain. The run time is user-determined and needs to be long enough to sample good candidate trees. These trees are then used to estimate the maximum a posteriori tree.

We conducted several experiments to evaluate whether the number of intermediate pathtrees for each pair of anchor trees affects the accuracy of the MDS reconstruction of the likelihood surface and how well we can recover the best tree. We use the Jukes-Cantor mutation model for the likelihood calculation throughout all analyses. Using such a simple model reduces potential difficulties introduced by parameter fitting.

We compared PATHTREES with PAUP* and REVBAYES for dataset D_1 because we were confident that the REVBAYES analysis converged. For dataset D_2, we compared PATHTREES with PAUP* and RAXML because even long runs of REVBAYES did not deliver stable results. Both PAUP* and RAXML were run without improving parameters that tune the heuristic search. All generated pathtrees were compared with all the evaluated trees in REVBAYES to investigate whether our approach can find topologies that were not visited by REVBAYES.

2 Results

We use our methods for landscape generation and visualization and finding high likelihood trees to examine the two datasets D_1 and D_2 using our method. Fig. 5 shows the contour and surface plots of the visualized treespace landscape for dataset D_1, generated with one pathtree per pair of starting trees (14 starting trees and 91 total pathtrees), and using the BHV distance for MDS input and thin-plate spline interpolation of the log-likelihood values of all plotted trees. The first row of Fig. 5 shows the 15 trees with the highest likelihood (small pink/purple colored disks) selected from the 91 pathtrees +14 starting trees. These 15 trees contained 4 different topologies, and the branch lengths in each topology cluster were optimized (medium-size circles). The purple spectrum color bar shows the relationship between the likelihood values of the optimized trees for each topology; the lighter the color of the topology (medium-sized dots), the lower the likelihood value of the corresponding optimized tree in that topology. Among locally optimal trees associated with different topologies, we found the best tree with the log likelihood of -15795.1817 (red square). We find the same tree as PAUP* (big white circle) as the optimal tree and also find the same tree as the maximum a posteriori tree (MAP) in REVBAYES (big black circle) as a locally optimal tree. In the plots, we see higher likelihood areas near the outside of the convex hull and a lower likelihood region in the middle. The 15 highest likelihood trees selected for topology analysis and their optimized topology trees are also towards the edges of the convex hull.

The second row in Fig. 5 shows the same 91 pathtrees and 14 starting trees as the first row, but optimized branch lengths have been computed for all their different topologies (30 topologies in total). Among all locally optimized trees (30 local optima corresponding to different topologies), we found again the best tree with the log likelihood of -15795.1817. In the plots, we still see a lower likelihood region in the middle of the convex hull, along with some points of higher likelihood corresponding to optimized trees. The orientation of the landscape is flipped from the first row, but this is an artifact of MDS.

Fig. 6 shows the contour and surface plots of the visualized treespace landscape for dataset D_1, generated with 15 pathtrees per starting tree pair, and using the wRF distance for MDS input and thin-plate spline interpolation of the log-likelihood values of all plotted trees. We generated a total of 1365 pathtrees with 43 different topologies to show a detailed treespace with a high number of trees. After optimizing the branch lengths for all 43 topologies, PATHTREES found the same best tree as PAUP*, as previously found, with the log likelihood of -15795.1817. With a higher number of pathtrees, we have a surface where trees are evenly spread out, and the gaps are filled. The best trees detected by PATHTREES, PAUP*, and REVBAYES were added to Supplementary Fig. S2. We still see a lower likelihood region in the middle of the plots. Pathtrees with the same topology are generally grouped together, and the sampling is dense enough that we can see

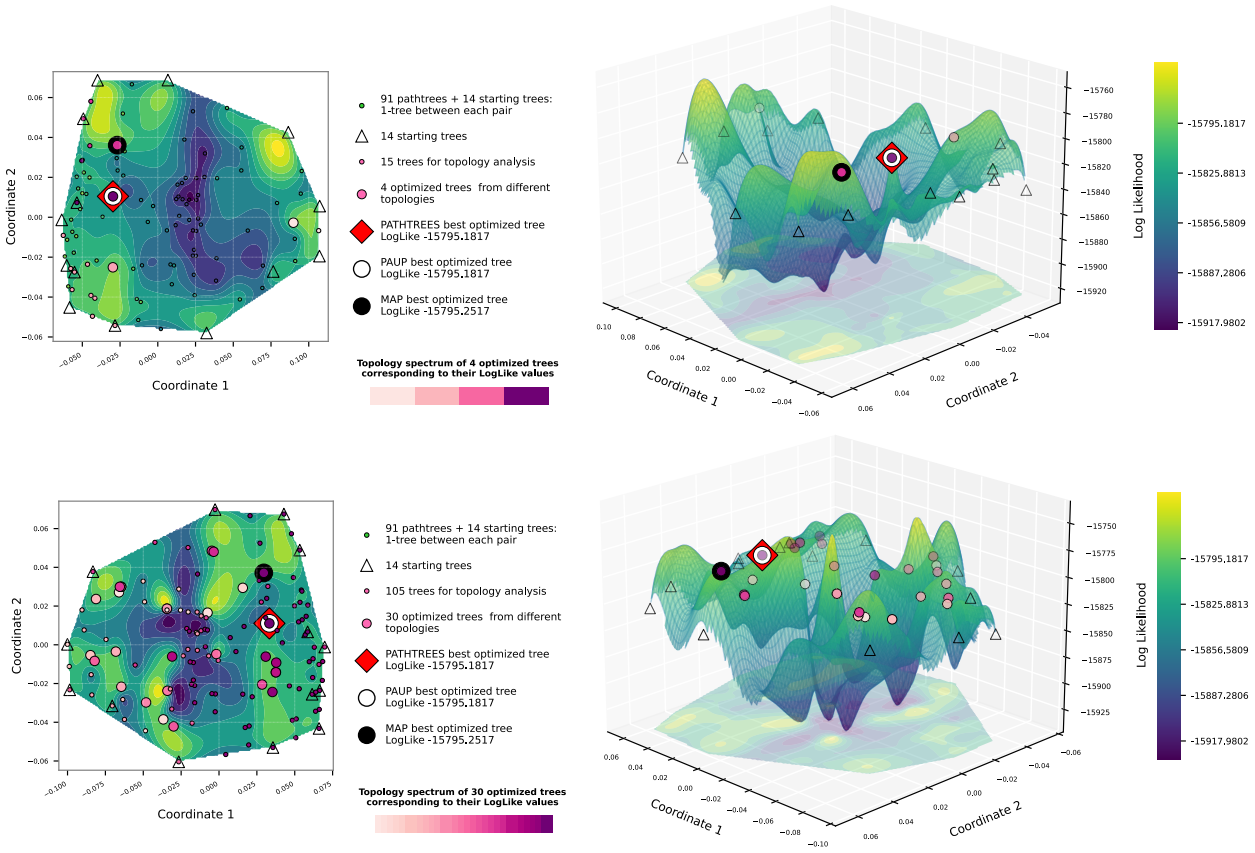

Figure 5. Contour and surface plots of PATHTREES for dataset D_1 when generating one pathtree per starting tree pair (91 pathtrees), using BHV distances for the MDS input, and thin-plate spline interpolation for the landscape. The vertices of the convex hull of the selected 1000 sample trees are the starting trees (14 triangles). First row: 15 trees with the highest likelihood were selected from the $91 + 14$ trees and classified based on their topologies (4 topologies with the relative likelihoods of their branch-length optimized trees given by the purple spectrum). Each medium-size circle with a purple color shows one of these locally optimal trees. The red square shows the best likelihood tree in the treespace found by PATHTREES. The large white disk shows the best tree of PAUP*, which is identical to PATHTREES' optimal tree. One of PATHTREES' locally optimal trees matches the MAP tree (big black circle). Second row: all $91 + 14$ trees were selected and classified based on their topologies (30 topologies). Medium-size disks on a purple spectrum show the 30 locally optimal trees and their relative likelihoods. The big red square shows the best likelihood tree of PATHTREES.

the path of the geodesics (curved lines of pathtrees) in some cases.

For the second dataset D_2, Fig. 7 shows the contour and surface plots of the likelihood landscape from two iterations of PATHTREES, with the second iteration zooming in on the top 100 optimized trees from the first iteration. The first row of Fig. 7 shows the first iteration plots, generated using 4 pathtrees per shortest path between starting trees (14 starting trees) and selecting the 100 trees with the highest likelihood to classify by topology. All selected trees have different topologies, giving 100 optimized trees. PATHTREES found a similar tree to the best tree from PAUP*, both with the log likelihood -5225.5856. The second row of Fig. 7 shows the second iteration plots, zooming in on the convex hull of the 100 optimized trees from the first iteration. 11 trees on the vertices of this convex hull became the starting trees for the second iteration.

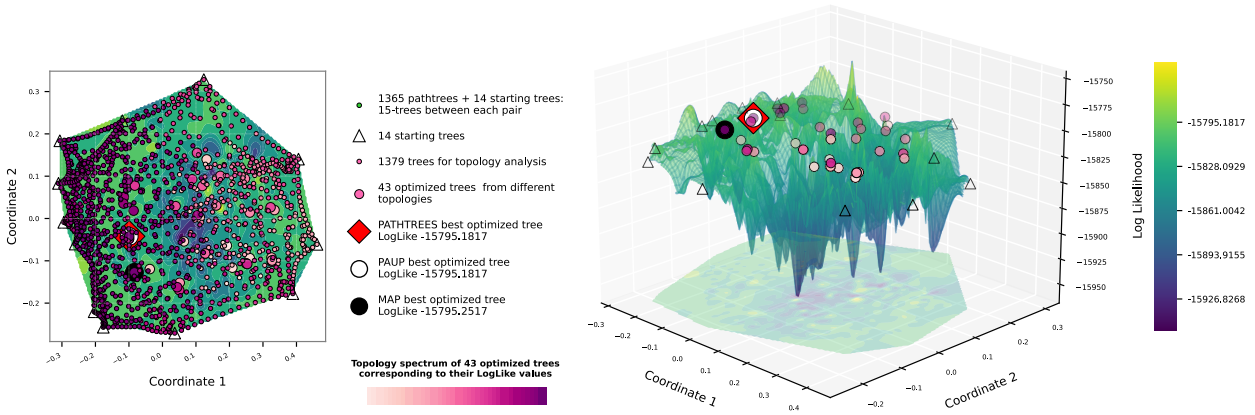

Figure 6. Contour and surface plots of PATHTREES for dataset D_1 when generating 15 pathtree per starting tree pair, using weighted Robinson-Foulds distances for the MDS input, and thin-plate spline interpolation for the landscape. After optimizing branch lengths for the 43 different topologies (colored by relative likelihood of this optimized tree using the purple spectrum), PATHTREES found the best tree (red square) with the log likelihood -15795.1817, which is the same as the optimal tree of PAUP* (white circle).

After generating 5 pathtrees between each starting tree pair and classifying the generated pathtrees based on topology, we found 154 different topologies, which we optimized. Despite additional topologies, we found the same tree with highest likelihood as in the first iteration. In the first row plots, we see only a small area of high likelihood trees, separated into two peaks, near the center. In the second row plots, zoomed in on this high likelihood section, we still see a lower likelihood chasm dividing the convex hull. The optimized topology trees are near the edges of the convex hull.

A comparison of the three optimal trees, namely ours, the best PAUP* tree, and the best RAXML tree, revealed that all trees are different from each other. Table 1 shows the weighted and unweighted Robinson-Foulds distances between the trees and their log likelihoods. Our tree and the PAUP* tree are close and only differ by four splits, whereas the RAXML tree differs from both our tree and the PAUP* tree by nine splits. The wRF distances between these three trees show similar relationships. The RAXML tree is different because its topology was found by applying a JC69 model with site rate variation (RAXML always uses site rate variation), but we then used PAUP* to find the optimal branch lengths for that topology, and the plain JC69 model to compute the likelihoods. The log likelihoods for all these 'best' trees are very similar, and comparing their location on the surface in Fig. 7 also shows that the best PATHTREES and PAUP* trees are close, whereas the RAXML tree seems to be on a different local maximum on the surface. Supplementary Fig. S3 contains all three trees showing their topology differences.

We used MDS to visualize the BHV treespace. This compression of the high dimensional space to 2 dimensions may lead to artifacts. We validated the accuracy of our visualizations using correlation analyses between the distance matrix in BHV space and the MDS distance matrix using the first and second coordinates. For the first dataset D_1 (Fig. 5, bottom row) the Pearson correlation was 0.9237 (additional

Table 1. Comparison of the best tree found by PATHTREES, PAUP*, and RAXML using the data D_2. Above the diagonal is unweighted Robinson-Foulds (uRF) for all pairs and below the diagonal is the weighted Robinson-Foulds distance (wRF). The last column is the log likelihood for each tree.

Tree	uRF / wRF			ln L
	PATHTREES	PAUP*	RAXML	
PATHTREES	-	4	9	-5225.5856
PAUP*	0.0054	-	9	-5225.5856
RAXML	0.0343	0.0325	-	-5225.8638

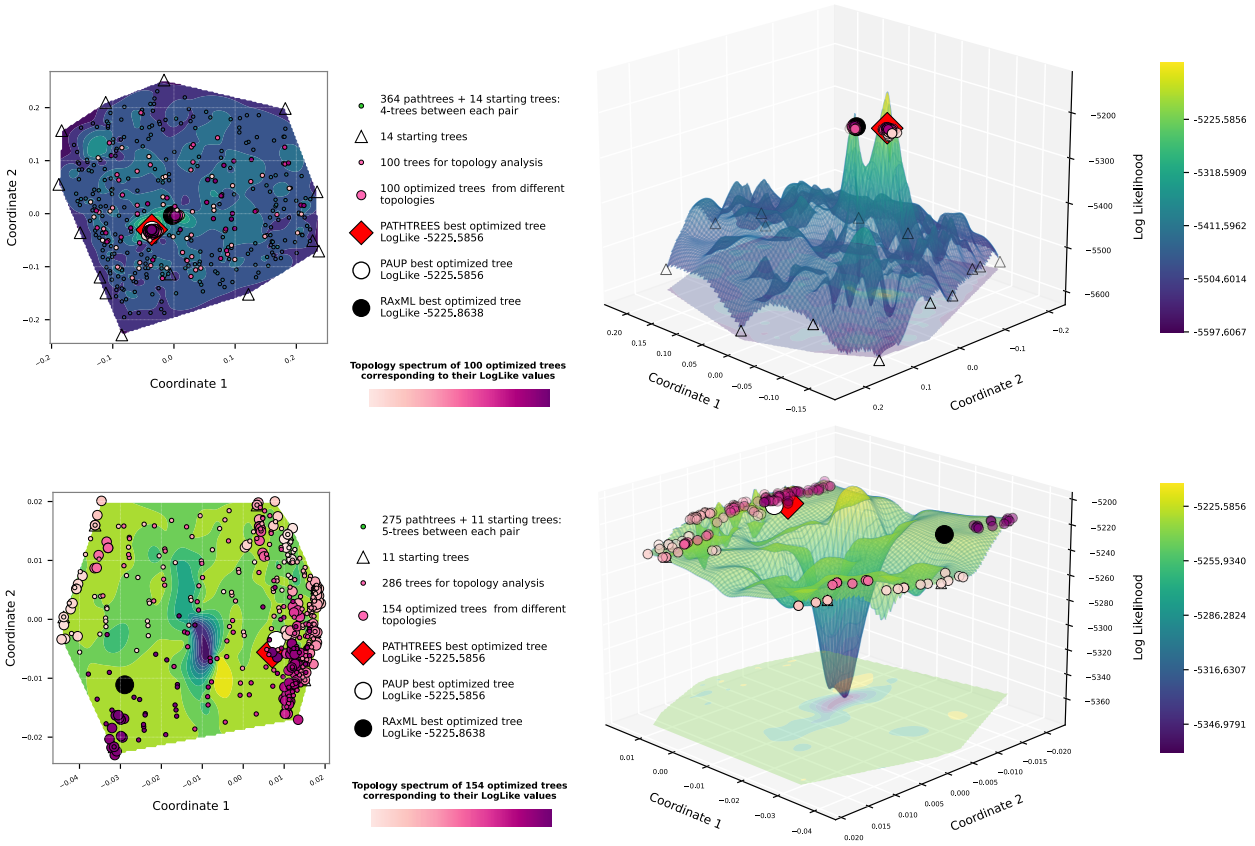

Figure 7. Contour and surface plots from PATHTREES for dataset D_2, using weighted Robinson-Foulds distances for the MDS input, and thin-plate spline interpolation for the landscape. First row: the vertices of the convex hull of the selected 500 sample trees are the starting trees (14 triangles). Four trees were generated on the geodesic of each pair of starting trees (364 pathtrees). Among them, 100 trees with the highest likelihood were selected to be classified based on topology. All 100 selected trees have different topologies (small circles, colored from the purple spectrum by the relative likelihood of their optimized tree). Each medium circle with a color from the purple spectrum shows the optimized tree with the corresponding topology. Among optimized trees, the red square shows the highest likelihood tree in the treespace, which is close but not identical to the best tree of PAUP* (white circle). The RAXML tree (large black circle) is different. Second row: Displays the treespace after zooming in on the optimized trees. The vertices of the convex hull of 100 optimized trees from the first iteration are the starting trees (11 triangles). Five pathtrees were generated between each starting tree pair (275 trees) and then all of them were selected to be classified by topology.

correlation measures and the Shepard diagram are shown in Supplementary Section S4.1). For dataset D_2 the Pearson correlation coefficient for Fig. 7 , top row (first iteration), was lower than for Fig. 7 , bottom

row (second iteration), 0.6856 and 0.9702 respectively, because the first figure covers a much larger area of trees than the second (more correlation measures and the Shepard diagrams are shown in Supplementary Section S4.2).

Our pathtrees are only based on the trees on the convex boundary, and one may wonder whether this reduces the chance to find relevant new trees. For dataset D_1, Fig. 5 (bottom row), PATHTREES starting out with 14 starting trees and generating 91 pathtrees found a total of 28 different topologies among 91 pathtrees, 4 of which were different from all topologies found by 50,000 sampled REVBAYES trees. A comparison with all 10,000 sampled REVBAYES trees for dataset D_2 revealed that our method found all 364 pathtrees with new topologies in Fig. 7 (first iteration), all of which were different from topologies found by REVBAYES. In Fig. 7 (second iteration), PATHTREES found a total of 145 different topologies among 275 pathtrees, again all different from the total topologies found by 10,000 sampled REVBAYES trees.

3 Discussion

Our approach uses the Billera-Holmes-Vogtmann treespace framework to generate and visualize treespace, including the likelihood landscape, in an area of interest, and to augment the search for the maximum likelihood tree by investigating global and local maxima found by our method in this area. While there are other programs and packages for MDS visualizations of treespace under various distance metrics, PATHTREES is the only recent tool to include comprehensive visualization of phylogenetic likelihood landscapes over an area of interest. Our method can be used at different scales (e.g., see Fig. 7) to better understand the spatial relationship between the highest likelihood trees. For example, in Fig. 7, the initial landscape shows the highest likelihood trees are close together within the MCMC searched area of treespace; zooming in allows us to see that these trees form two high likelihood ridges, with a lower-likelihood region between them. Secondly, our method can find trees with the same maximal likelihood as other tree search programs, including one with a different topology for dataset D_2. These results suggest PATHTREES could potentially be used to understand treespace islands and terraces better.

A limitation of our method is the need for starting trees, due to the enormity of treespace. For our two example datasets, treespace has 5.6×10^{26} orthants and 4.7×10^{384} orthants, respectively, so we cannot start with the full treespace and just zoom in. We decided to use a sample of trees from a Bayesian phylogenetic inference program, REVBAYES. In principle, any set of reasonably close trees to the best tree may work as a starting point. Using the convex hull in MDS space to start our approach helped reduce the geodesic distance matrix size used to create the visualizations. In a way, we treat the convex hull in MDS space as an approximation of the convex hull in treespace (Lubiw et al. 2020; Lin et al. 2017). However, we believe that

the hull defined by MDS allows us to investigate good and best trees within its boundary. In our examples, the best tree, found by other procedures and ours, is within this MDS hull.

A second limitation of our method is that tree topologies along geodesics can only contain splits that are in one of the two endpoints trees and contain all splits that are in both of the endpoint trees. However, we argue this constraint is not onerous - there can still be exponentially many topologies fitting this description - but means that the intermediate tree topologies explored are relevant. Additionally, trees on a geodesic between two starting trees are unlikely to be in the set of best trees because this geodesic defines their branch lengths and topology. Thus, our method takes these pathtrees as starting points and optimizes their branch lengths (once per topology). This procedure allows us to describe the tree landscape along the path and describe the maxima for specific topologies. Using many pathtrees that were optimized or not optimized will give a precise picture of the tree landscape (for example, Fig. 5, Fig. 6, and Fig. 7). Our method provides a simple framework for exploring landscapes of phylogenetic trees and visualizing their relationship in a continuous and low-dimensional projection facilitated by multidimensional scaling and an interpolation method, cubic spline or thin-plate spline interpolation, to reveal potential tree islands. The visualization gives a good impression of the likelihood landscape: general patterns can be shown with few trees, but details may need many more trees to create a more smooth surface. However, even with many trees, the visualization may contain artifacts in areas where there are no trees, for example, the spikes in Fig. 7.

A Bayesian inference method evaluates trees according to the posterior probability, which is dominated by the likelihood of the tree when we assume vague prior distributions. It is fair to say that even a long inference run will not explore all possible topologies. Even our small dataset of 23 primate species has too many different topologies to explore all in a Bayesian context in a reasonable time. Of course, most of these topologies will have an inferior fit to the sequence data, but even those trees that fit the data relatively well are many. In contrast to Bayesian inference and heuristic search methods, our method does not depend on an optimality criterium to pick trees that lay on the shortest path between two arbitrarily chosen trees. This allows exploring topologies that were never visited with a good Bayesian run or any other heuristic search as we have shown.

We picked the two datasets because they represent very different situations. The primate dataset D_1 is relatively small. However, it is still too big to be solved exhaustively. It provides many mutational differences allowing good resolution of branch lengths and branching patterns. The second dataset D_2 has eight times more individuals that are closely related. Many sequences are identical, leading to many multifurcations.

Heuristic searches for D_1 and D_2 are fast, and even an MCMC run with RevBayes does not need a long time for D_1. However, we had difficulties estimating a MAP tree for D_2 because we had difficulty running to convergence. Pathtrees generates independent trees, evenly spaced along geodesics, to help

visualize treespace, find optimal trees, and explore the likelihood surface near these optimal trees. PATHTREES optimizes its pathtrees and finds local maxima for the evaluated topologies; these are the same as those found by PAUP* and REVBAYES for D_1. Interestingly, the MAP tree and the PAUP* tree differ by two splits but when the branch lengths are optimized deliver log likelihoods that are very similar. Even ten times longer runs in REVBAYES deliver the same MAP tree. So while it may seem difficult for REVBAYES to explore that particular topology with the highest likelihood, the difference is only 0.07 log units. We certainly would not exclude the MAP tree in a likelihood ratio test. The second dataset D_2 reveals that many trees will be good candidates for the best likelihood tree. The PAUP*, RAXML, and PATHTREES best trees have all different topologies but very similar log likelihoods. These trees are also very similar, with only 4 or 9 different splits between them. PATHTREES helps give insights about the likelihood surface, such that it is rather flat and therefore will have many potential trees with similar likelihoods.

We believe that our method, implemented in PATHTREES, complements heuristic search phylogenetic analyses and allows visualization of the treespace and finding alternative trees with log likelihoods that are potentially better than those of heuristic searchers. For example, a new way to propose topologies for tree search could be by sampling pathtrees along a geodesic between two trees, or in a region, of interest. Alternatively, pathtrees could become starting trees themselves for a maximum likelihood search. The visualization of the likelihood surface also allows the discussion of local likelihood maxima, which we hope will lay the groundwork for improving search algorithms.

Supplementary Material

Supplementary data are available in a single separate PDF file.

Acknowledgements

PB was partially supported by the US National Science Foundation grant DBI 1564822 and DBI 2019989. MO was partially supported by the US National Science Foundation grant DMS 1847271.

Software Availability

We implemented our method as free software named PATHTREES under the MIT open-source licence. The source code and the documentation of PATHTREES are available at https://github.com/TaraKhodaei/pathtrees_project.git

The primate data is available from the Tutorial website of RevBayes: https://revbayes.github.io/tutorials/ctmc/#example-character-evolution-under-the-jukes-cantor-substitution-model (the dataset url is https://revbayes.github.io/tutorials/ctmc/data/primates_and_galeopterus_cytb.nex). The snake dataset is available from the Dryad Digital Repository: http://dx.doi.org/10.5061/dryad.7hs34mj.

References

N. Amenta and J. Klingner. Case study: Visualizing sets of evolutionary trees. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, pages 71–74. IEEE, 2002.

N. Amenta, M. Godwin, N. Postarnakevich, and K. St. John. Approximating geodesic tree distance. *Inform Process Lett*, 103(2):61–65, 2007.

N. Amenta, M. Datar, A. Dirksen, M. d. Bruijne, A. Feragen, X. Ge, J. H. Pedersen, M. Howard, M. Owen, J. Petersen, et al. Quantification and visualization of variation in anatomical trees. In *Research in Shape Modeling*, pages 57–79. Springer, 2015.

O. Bastert, D. Rockmore, P. F. Stadler, and G. Tinhofer. Landscapes on spaces of trees. *Appl Math Comput*, 131(2-3):439–459, 2002.

L. J. Billera, S. P. Holmes, and K. Vogtmann. Geometry of the space of phylogenetic trees. *Adv Appl Math*, 27(4):733–767, 2001.

R. R. Bouckaert. DensiTree: making sense of sets of phylogenetic trees. *Bioinformatics.*, 26(10):1372–1373, 2010.

M. D. Buhmann. Chapter 2 - summary of methods and applications. In *Radial Basis Functions: Theory and Implementations*, volume 12. Cambridge University Press, 2003.

P. Buneman. The recovery of trees from measures of dissimilarity. In *Mathematics in the Archeological and Historical Sciences*, pages 387–395. Edinburgh University Press, 1971.

E. A. Chambers and D. M. Hillis. The multispecies coalescent over-splits species in the case of geographically widespread taxa. *Syst Biol*, 69(1):184–193, 2019.

B. Chor and T. Tuller. Maximum likelihood of evolutionary trees is hard. In *Annual International Conference on Research in Computational Molecular Biology*, pages 296–310. Springer, 2005.

B. Chor, M. D. Hendy, B. R. Holland, and D. Penny. Multiple maxima of likelihood in phylogenetic trees: an analytic approach. *Mol Biol Evol*, 17(10):1529–1541, 2000.

M. A. A. Cox and T. F. Cox. Multidimensional scaling. In *Handbook of Data Visualization*, pages 315–347. Springer, Heidelberg, Berlin, 2008.

C. De Boor. Piecewise cubic interpolation. In *A Practical Guide to Splines*, pages 40–47. Springer-Verlag, New York, 1978.

A. J. Drummond and A. Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol Biol*, 7(1):214, 2007.

J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J Mol Evol*, 17 (6):368–376, 1981.

W. M. Fitch. Toward defining the course of evolution: minimum change for a specified tree topology. *Syst Zool*, 20(4):406–416, 1971.

W. M. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155(3760):279–284, 1967.

L. R. Foulds and R. L. Graham. The Steiner problem in phylogeny is NP-complete. *Adv Appl Math*, 3(1): 43–49, 1982.

K. Gori, T. Suchan, N. Alvarez, N. Goldman, and C. Dessimoz. Clustering genes of common evolutionary history. *Mol Biol Evol*, 33(6):1590–1605, 2016.

S. Guindon and O. Gascuel. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol*, 52(5):696–704, 2003.

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

D. M. Hillis, T. A. Heath, and K. St. John. Analysis and visualization of tree space. *Syst Biol*, 54(3):471–482, 2005.

S. Höhna and A. J. Drummond. Guided tree topology proposals for Bayesian phylogenetic inference. *Syst Biol*, 61(1):1–11, 2012.

S. Höhna, M. J. Landis, T. A. Heath, B. Boussau, N. Lartillot, B. R. Moore, J. P. Huelsenbeck, and F. Ronquist. RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Syst Biol*, 65(4):726–736, 2016.

S. Höhna, M. J. Landis, and T. A. Heath. Phylogenetic inference using RevBayes. *Curr Protocols Bioinformatics*, 57(1):6.16.1–6.16.34, 2017.

W. Huang, G. Zhou, M. Marchand, J. R. Ash, D. Morris, P. Van Dooren, J. M. Brown, K. A. Gallivan, and J. C. Wilgenbusch. Treescaper: visualizing and extracting phylogenetic signal from sets of trees. *Mol Biol Evol*, 33(12):3314–3316, 2016.

J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17 (8):754–755, 2001.

T. Jombart, M. Kendall, A. Almagro, J. Garcia, and C. Colijn. TREESPACE: Statistical exploration of landscapes of phylogenetic trees. *Mol Ecol Resour*, 17(6):1385–1392, 2017.

M. Kendall and C. Colijn. Mapping phylogenetic trees to reveal distinct patterns of evolution. *Mol Biol Evol*, 33(10):2735–2743, 2016.

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

C. Lakner, P. van der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Syst Biol*, 57(1):86–103, 2008.

B. Lin, B. Sturmfels, X. Tang, and R. Yoshida. Convexity in tree spaces. *SIAM J Discrete Math*, 31(3): 2015–2038, 2017.

Z. Liu, S. H. Zhan, and T. Munzner. Aggregated dendrograms for visual comparison between many phylogenetic trees. *IEEE Trans Vis Comput Graph*, 26(9):2732–2747, 2019.

A. Lubiw, D. Maftuleac, and M. Owen. Shortest paths and convex hulls in 2D complexes with non-positive curvature. *Comput Geom*, 89:101626, 2020.

D. R. Maddison. The discovery and importance of multiple islands of most-parsimonious trees. *Syst Biol*, 40 (3):315–328, 1991.

E. Miller, M. Owen, and J. S. Provan. Polyhedral computational geometry for averaging metric phylogenetic trees. *Adv Appl Math*, 68:51–91, 2015.

D. Money and S. Whelan. Characterizing the phylogenetic tree-search problem. *Syst Biol*, 61(2):228, 2012.

M. Owen and J. S. Provan. A fast algorithm for computing geodesic distances in tree space. *IEEE/ACM Trans Comput Biol Bioinf*, 8(1):2–13, 2011.

D. F. Robinson and L. R. Foulds. Comparison of weighted labelled trees. *Lect Notes Math*, 748:119–126, 1979.

D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Math Biosci*, 7(1):1–8, 1981.

S. Roch. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Trans Comput Biol Bioinf*, 3(1):92–94, 2006.

S. Ruane, R. W. J. Bryson, R. A. Pyron, and F. T. Burbrink. Coalescent species delimitation in milksnakes (Genus Lampropeltis) and impacts on phylogenetic comparative analyses. *Syst Biol*, 63(2):231–250, 2014.

N. Saitou and M. Nei. The Neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–425, 1987.

L. A. Salter. Complexity of the likelihood surface for a large DNA dataset. *Syst Biol*, 50(6):970–978, 2001.

M. J. Sanderson, M. M. McMahon, and M. Steel. Terraces in phylogenetic tree space. *Science*, 333(6041): 448–450, 2011.

M. J. Sanderson, M. M. McMahon, A. Stamatakis, D. J. Zwickl, and M. Steel. Impacts of terraces on phylogenetic inference. *Syst Biol*, 64(5):709–726, 2015.

E. Schröder. Vier combinatorische Probleme. *Z Math Phys*, 15:361–376, 1870.

M. R. Smith. Robust analysis of phylogenetic tree space. *Syst Biol*, 71(5):1255–1270, 2022.

K. St. John. The shape of phylogenetic treespace. *Syst Biol*, 66(1):e83–e94, 2017.

A. Stamatakis, T. Ludwig, and H. Meier. RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics*, 21(4):456–463, 2005.

M. Steel. The maximum likelihood point for a phylogenetic tree is not unique. *Syst Biol*, 43(4):560–564, 1994.

M. A. Steel and D. Penny. Distributions of tree comparison metrics—some new results. *Syst Biol*, 42(2): 126–141, 1993.

J. Sukumaran and M. T. Holder. DendroPy: a Python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–1571, 2010.

D. Swofford. PAUP*. phylogenetic analysis using parsimony (*and other methods). version 4., 2003.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*, 17:261–272, 2020.

D. L. Warren, A. J. Geneva, and R. Lanfear. RWTY (R We There Yet): an R package for examining convergence of Bayesian phylogenetic analyses. *Mol Biol Evol*, 34(4):1016–1020, 2017.

C. Whidden and F. A. Masten. Quantifying MCMC exploration of phylogenetic tree space. *Syst Biol*, 64(3): 472–491, 2015.

J. C. Wilgenbusch, W. Huang, and K. A. Gallivan. Visualizing phylogenetic tree landscapes. *BMC bioinformatics*, 18:85, 2017.

A. M. Wright and G. T. Lloyd. Bayesian analyses in phylogenetic palaeontology: interpreting the posterior sample. *Palaeontology*, 63(6):997–1006, 2020.