Theoretical Computer Science 699 (2017) 1–20

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Ricci–Ollivier curvature of the rooted phylogenetic subtree–prune–regraft graph ☆,☆☆

Chris Whidden *, Frederick A. Matsen IV

Program in Computational Biology, Fred Hutchinson Cancer Research Center, Seattle, WA 98109, USA

ARTICLE INFO

Article history:
Received 29 April 2016
Received in revised form 20 January 2017
Accepted 3 February 2017
Available online 20 February 2017

Keywords:
Markov chain Monte Carlo
Phylogenetics
Ricci–Ollivier curvature
Subtree–prune–regraft

ABSTRACT

Statistical phylogenetic inference methods use tree rearrangement operations such as subtree–prune–regraft (SPR) to perform Markov chain Monte Carlo (MCMC) across tree topologies. The structure of the graph induced by tree rearrangement operations is an important determinant of the mixing properties of MCMC, motivating the study of the underlying *SPR graph* in greater detail.

In this paper, we investigate the SPR graph of rooted trees (rSPR graph) in a new way: by calculating the Ricci–Ollivier curvature with respect to uniform and Metropolis–Hastings random walks. This value quantifies the degree to which a pair of random walkers from specified points move towards each other; negative curvature means that they move away from one another on average, while positive curvature means that they move towards each other. In order to calculate this curvature, we develop fast new algorithms for rSPR graph computation. We then develop formulas characterizing how the number of rSPR neighbors of a tree changes after an rSPR operation is applied to that tree. These give bounds on the curvature, as well as a flatness-in-the-limit theorem indicating that paths of small topology changes are easy to traverse. However, we find that large topology changes (i.e. moving a large subtree) give pairs of trees with negative curvature. We show using simulation that mean access time distributions depend on distance, degree, and curvature, demonstrating the relevance of these results to stochastic tree search.

1. Introduction

Molecular phylogenetic methods reconstruct evolutionary trees from DNA or RNA data and are of fundamental importance to modern biology. Statistical phylogenetics is the currently most popular means of reconstructing phylogenetic trees, in which the tree is viewed as an unknown parameter in a likelihood-based statistical inference problem. The likelihood function in this setting is the likelihood of generating the observed sequences via a continuous time Markov chain (CTMC) evolving down the tree starting from a sequence assumed to be sampled from the stationary distribution [1]. The lengths of the branches of the phylogenetic tree give the "time" parameter in the CTMC, where the generated sequence accrues muta-

☆ This work was funded by National Science Foundation award 1223057. Chris Whidden is a Simons Foundation Fellow of the Life Sciences Research Foundation. The research of Frederick Matsen was supported in part by a Faculty Scholar grant from the Howard Hughes Medical Institute and the Simons Foundation.

☆☆ A preliminary version of this work appeared in the 2016 Proceedings of the Thirteenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO).

* Corresponding author.
E-mail addresses: cwhidden@fredhutch.org (C. Whidden), matsen@fredhutch.org (F.A. Matsen).

http://dx.doi.org/10.1016/j.tcs.2017.02.006

tions, typically in an IID manner across sites. It is now common for researchers to approximate the posterior distribution of trees and their associated parameters in a Bayesian setting using Markov chain Monte Carlo (MCMC).

In order to estimate these distributions accurately, MCMC samplers must sufficiently explore the set of trees. Phylogenetic search algorithms typically attempt to do so through a combination of modifications to the continuous parameters and tree topology. Topology changes have been identified as the main limiting factor of Bayesian MCMC algorithms [2,3], as other parameters cannot be accurately estimated if the topology distribution is not accurately sampled. Commonly used phylogenetics software packages such as MrBayes [4] and BEAST [5] rearrange subtrees via subtree–prune–regraft (SPR) moves (Fig. 1(d)) or the subset of SPR moves called nearest neighbor interchanges (NNI) [6]. Thus, phylogenetic searches can be viewed as traversing the *SPR graph*: the graph with phylogenetic trees as vertices and SPR adjacencies as edges.

It has become increasingly clear that the structure of the SPR graph plays an important role in determining the accuracy of tree searches. Researchers have previously identified slow mixing in MCMC with pathological data [7–9]. On the other hand, fast mixing has been identified with exceptionally well-behaved data [10] or with a uniform distribution [11]. Probabilists have also approached the problem using related frameworks that are more amenable to proving theorems, both for other sets of moves on trees with a finite number of leaves [12,13] and for SPR and related moves on a continuous tree-like object which formalizes the notion of a tree with infinitely many leaves [14,15]. Studies on real data [16,2], however, have identified posteriors which are difficult to sample using MCMC. Previously, the lack of sufficient computational tools for examining phylogenetic posteriors in terms of SPR operations made it difficult to determine the cause of these difficulties. By developing the first such tools, we recently showed that graph structure has a significant effect on MCMC mixing with MrBayes applied to real data [17], and that multimodal posteriors are common and separated by "bottlenecks" of specific classes of SPR moves.

Although the SPR graph is thus very important in determining the success of phylogenetic inference procedures, little is still known about the rooted or unrooted versions of the SPR graph itself. [18] developed a recursive procedure on a tree to find the degree of the corresponding vertex in the rooted SPR (rSPR) graph, and corresponding bounds on its degree. [19] showed that the diameter Δ_{rSPR} of the rSPR graph is $n - \Theta(\sqrt{n})$, and for the unrooted case they show

$$n - 2\lceil\sqrt{n}\rceil + 1 \leq \Delta_{\text{uSPR}}(n) \leq n - 3 - \left\lfloor \frac{\sqrt{n-2}-1}{2} \right\rfloor. \tag{1}$$

[20] showed that the expected distance between a pair of trees is $n - \Theta(n^{2/3})$. We are not aware of any further work investigating properties of the SPR graph, which may be due to its complexity. Indeed, even computing the distance between topologies in terms of SPR operations (rooted and unrooted) is NP-hard [21,22]. Fortunately, it is fixed-parameter tractable with respect to the distance in the rooted case [21] and unrooted case [23] and efficient fixed-parameter algorithms have recently been developed [17,24,25] that begin to allow such investigations.

Ollivier and colleagues recently pioneered a new approach to calculating the Ricci curvature on a general type of metric space, including graphs [26,27]. In this framework, local information about the metric space is given by a random walk (rather than a Riemann tensor) such that their notion of curvature formalizes the notion of to what extent random walking brings neighborhoods together. Applying the framework to Brownian motion on a manifold returns the classical definition of Ricci curvature. The curvature is determined by the ratio of the earth mover's distance [28] between neighborhoods of a pair of vertices given by a random walk and the distance between the vertices. Here the term *random walk* on a space X simply denotes a family of probability measures parameterized by points of X satisfying reasonable assumptions, which includes biased walks such as MCMC. This approach has been useful for determining properties of a wide variety of graphs including the Internet topology [29] and cancer networks [30].

In this paper, we investigate curvature of the rSPR graph with respect to two random walks and compare those results to access times (i.e. hitting times) for those random walks. Our explicit focus here is to investigate random walks defined only in terms of the graph itself: the uniform random walk and MCMC sampling from the uniform prior on trees. In future work, we will extend these methods to study more complicated distributions with non-uniform topology probabilities.

We required several new computational tools. We present a fast new algorithm for computing rSPR graphs from a set of trees, reducing the time to do so from $O(m^2n)$ to $O(mn^3)$ for a set of m trees each of which has n leaves. As the full rSPR graph on trees with n leaves contains $(2n-3)!! = 3 \cdot 5 \cdot \ldots \cdot (2n-3)$ trees, this is a significant improvement in practice for exploring large subsets of the graph (or, as we do here, the full graph for small numbers of leaves). By exploiting symmetries in the rSPR graph, we were able to calculate all of the curvatures for pairs of trees with up to seven leaves. By carefully examining the overlap in rSPR moves, we present a new method for computing the degree of a tree in the rSPR graph that allows one to select an rSPR neighbor uniformly at random in linear time without explicitly generating the graph. This stands in contrast to the sampling methods used in current software such as MrBayes, which do not propose SPR moves uniformly.

Using our methods to simulate these random walks, we found that the distribution of access times between pairs of trees can be described by the distance between the trees, the degrees of the trees, and the curvature. Our results demonstrated that rSPR graphs for trees with 7 or more leaves have tree pairs with negative curvature, corresponding to direct paths that are difficult to traverse stochastically. By getting a more fine-tuned understanding of the rSPR neighborhood of pairs of vertices, we are able to give bounds on the earth mover's distance in this context and thus curvatures under these random walks. In particular, we present a full characterization of the change in rSPR degree that occurs from a given rSPR

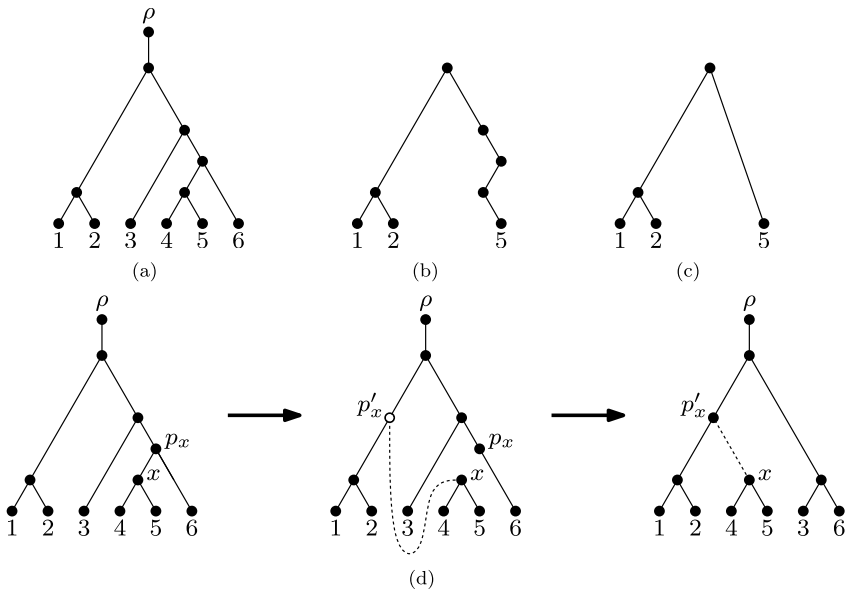


Fig. 1. (a) An X-tree T. (b) $T(V)$, where $V = \{1, 2, 5\}$. (c) $T|V$. (d) An rSPR operation transforms T into a new tree T' by *pruning* a *subtree* and *regrafting* it in another location.

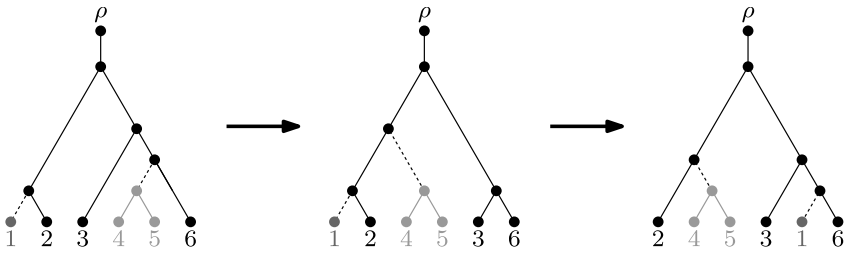


Fig. 2. Two rSPR operations, each of which moves one gray subtree. The leftmost and rightmost trees are rSPR distance two apart.

move and find that even though they each count as one move, rSPR moves which modify large subtrees are less likely to be explored during these random walks. Pairs of trees separated by such moves correspond to the pairs with negative curvature identified in our simulation results. These pairs occur infrequently in these well-connected graphs, however, they may be more problematic in real posterior distributions where the majority of the probability is spread over a relatively small number of trees [17]. In summary, we extend the knowledge about an important graph for phylogenetics, specifically in a way that models phylogenetic MCMC search.

The automated computational analysis code can be found at https://github.com/matsengrp/curvature.

2. Preliminaries

We follow the definitions and notation from [21,24,17]. A (rooted binary phylogenetic) X-tree is a rooted tree T whose nodes have zero or two children such that the leaves of T are bijectively labeled with the members of a label set X. As in [21,24,17], the tree is augmented with a labeled root node ρ and ρ is considered a member of X (Fig. 1(a)). The depth of ρ is considered to be -1, the depth of the original root is 0, and the depth of each other node is given by its distance from the original root. We generally use n to refer to the number of leaves in an X-tree. For a subset V of X, $T(V)$ is the smallest subtree of T that connects all nodes in V (Fig. 1(b)). The V-tree induced by T is the smallest tree $T|V$ that can be obtained from $T(V)$ by suppressing unlabeled nodes with fewer than two children (Fig. 1(c)). For the rest of the paper, we will assume that all phylogenetic trees are binary and rooted, and that tree inclusion is rooted tree inclusion.

A *parent (sub)tree* of a subtree U is the smallest subtree strictly containing U. A *parent edge* of a subtree U is the edge connecting U to the rest of the tree. The *internal edges* of a tree are the edges that do not contact a leaf or ρ. A *ladder tree* (also known as a *caterpillar tree*) is a tree such that every internal node has a leaf as a direct descendant. A *balanced tree* is a tree such that the sum of the depths of internal nodes is minimum over all trees with the same number of leaves. The *least common ancestor* (LCA) of a set R of two or more nodes is the unique node that is an ancestor of each node $r \in R$ and at maximum depth. Similarly, the LCA of two or more subtrees is the LCA of their root nodes.

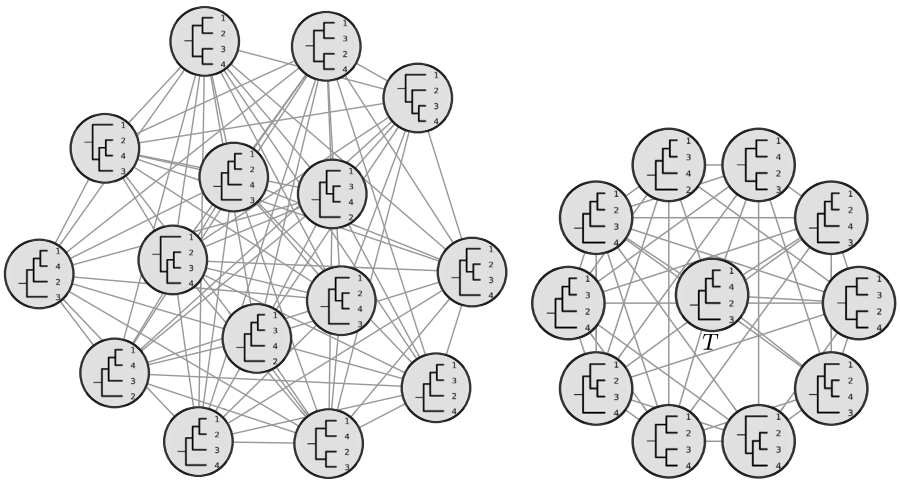


Fig. 3. (a) The rSPR graph of X-trees with 4 leaves. (b) The neighborhood of a particular X-tree T with 4 leaves, showing connections with T and between neighbors.

A (rooted) *subtree–prune–regraft* (rSPR) operation on an X-tree T cuts an edge $e = (x, p_x)$ where p_x denotes the parent of node x. T is divided into two subtrees T_x and T_{p_x} containing x and p_x, respectively. Then the operation adds a new node p'_x to T_{p_x} by subdividing an edge of T_{p_x} and adding a new edge (x, p'_x), making x a child of p'_x. Finally, p_x is suppressed, joining the two edges on either side of that node. See Fig. 1(d) for an example. The inclusion of ρ allows for rSPR moves which move subtrees to the root of the tree, however, this definition does not allow the arbitrary re-rooting of trees by moving ρ.

rSPR operations induce a distance measure between X-trees: $d_{\text{SPR}}(T_1, T_2)$ is the minimum number of rSPR operations required to transform an X-tree T_1 into T_2. For example, the trees in Fig. 2 are separated by two rSPR operations. Moreover, rSPR operations naturally give rise to a graph on the set of X-trees for which this distance is simply the shortest-path graph distance. Let $\mathcal{T}_n$ be the set of trees with n leaves and label set $X = \{1, 2, \ldots n, \rho\}$. Then the rSPR graph G of $\mathcal{T}_n$ is the graph with vertex set $V(G) = \mathcal{T}_n$ and edge set $E(G) = \{(T, S) \mid d_{\text{SPR}}(T, S) = 1, T \in V, S \in V\}$. Fig. 3(a) illustrates the rSPR graph of 4-leaf trees.

To avoid confusion between the two types of graph structures considered here, we refer to vertices of the rSPR graph as *vertices* and vertices of individual trees (i.e. leaves and internal nodes) as *nodes*. Let $N(T)$ be the set of rSPR neighbors of a tree T (this does not include T). For example, the tree T with 4 leaves in Fig. 3(b) has 10 neighbors. The degree of T, i.e. the number of trees which can be obtained from T by a single rSPR operation, will be denoted $|N(T)|$. We assume that all trees are bifurcating, and thus use degree to refer only to the degree of rSPR graph vertices.

Ricci–Ollivier curvature provides a rigorous yet intuitive formalization of the shape of a metric space with respect to a random walk. For the purposes of this paper, we restrict ourselves to graphs equipped with a shortest-path distance. For a more rigorous presentation in the more general setting of a Polish metric space, see [26] or the survey [31].

Let m_x and m_y be the probability densities of the position of a specified random walk after one step of the random walk, starting at points x and y of a graph $G = (V, E)$, respectively. The transportation distance [32] (equivalently Wasserstein distance, or "earth mover's distance" [28]) between m_x and m_y is the minimum amount of "work" required to move m_x to m_y along edges of the graph, that is

$$W_1(m_x, m_y) := \min_{\xi \in \Pi(m_x, m_y)} \sum_{\{z,w\} \subset V} d(z, w)\xi(z, w), \tag{2}$$

where $d(z, w)$ is the graph shortest-path distance ($d_{\text{SPR}}(z, w)$ in our case) and $\Pi(m_x, m_y)$ is the set of densities on $V \times V$ that are m_x after projecting on the first component and m_y after projecting on the second.

The *coarse Ricci–Ollivier curvature* of x and y is then defined as:

$$\kappa(m; x, y) := 1 - \frac{W_1(m_x, m_y)}{d(x, y)}. \tag{3}$$

For the purposes of this paper, "curvature" without further specification will refer to (3). We will use $\kappa(x, y)$ to denote the curvature of the simple (uniform choice of neighbor) random walk, and use $\kappa(\text{MH}; x, y)$ to indicate curvature with respect to the Metropolis–Hastings random walk sampling the uniform distribution (described in detail in Section 3.2). Positive curvature implies that the neighborhoods m_x and m_y are closer in transportation distance than point masses at x and y, zero curvature implies that they are neither closer nor farther, and negative curvature implies that m_x and m_y are more

distant than point masses at x and y. Curvature thus provides an intuitive measure of the difficulty of moving between regions of the graph with a random walk.

Lin et al. [33] defined a variant definition of curvature in terms of lazy random walks which Loisel and Romon [34] dubbed the *asymptotic Ricci–Ollivier curvature*. The lazy random walk only travels according to m_x with probability p and otherwise stays put. Thus the lazy mass assignment m_x^p is the sum of $p\,m_x$ and a point mass of $1-p$ on x. We denote the coarse curvature of the p-lazy random walk between two vertices x and y with respect to a random walk m by $\kappa_p(m; x, y)$. For example, $\kappa_{1/4}(m; x, y)$ describes the curvature of the lazy random walk that follows the given random walk m with probability $1/4$ and remains stationary with probability $3/4$. The asymptotic Ricci–Ollivier curvature of x and y is then:

$$\mathrm{ric}(m; x, y) := \lim_{p\to 0} \frac{\kappa_p(m; x, y)}{p}. \tag{4}$$

As above for κ, we use $\mathrm{ric}(x, y)$ as shorthand for $\mathrm{ric}(m; x, y)$ when m is the uniform lazy random walk, and $\mathrm{ric}(\mathrm{MH}; x, y)$ when m is the Metropolis–Hastings random walk sampling the uniform distribution (Section 3.2). This definition of curvature is invariant of p for small p [34] and can be used to avoid parity problems on graphs where the uniform random walk is periodic without choosing a specific laziness parameter (e.g. Ollivier often considered $\kappa_{\frac{1}{2}}(x, y)$ for this purpose). As we prove in Lemma 6.8, the notions of coarse and asymptotic curvature differ only by a small factor bounded by $\frac{2}{\max(|N(x)|,|N(y)|)}$ between adjacent vertices and are equal for nonadjacent vertices.

3. Efficient algorithms for computing and sampling rSPR graphs

3.1. Computing the rSPR graph of m trees with n leaves in $O(mn^3)$-time

It is necessary to have an efficient method of constructing the full rSPR graph for a fixed number of leaves in order to study it. The previously best algorithm for this problem requires $O(m^2n)$ time, where m is the number of trees in the graph and n the number of leaves [17]. Here we reduce that time to $O(mn^3)$. Note that for the full rSPR graph, the value of m is given by the rapidly growing function $(2n-3)!!$, that is, $3\cdot 5\cdot \ldots \cdot (2n-3)$, and this is therefore a significant improvement in practice, as we demonstrate below.

In previous work [17], we constructed (unrooted) SPR graphs from subsets of high probability trees sampled from phylogenetic posteriors to compare mixing and identify local maxima. Although the SPR distance (rooted and unrooted) is NP-hard to compute [21,22], it is fixed-parameter tractable with respect to the distance in the rooted case [21]. In particular, one can determine in $O(n)$-time whether two rooted phylogenetic trees are adjacent in the rSPR graph ($O(n^2)$-time for unrooted trees) using the algorithms of Whidden et al. [35,36,24,17]. We previously applied this method comparing each of the m trees in a given graph pairwise to identify adjacencies, requiring a total of $O(m^2n)$-time ($O(m^2n^2)$-time in the unrooted case). However, this method is impractical when applied to construct graphs with 7 or more leaves, due to the rapidly growing $O(m^2)$ factor.

The key to our efficient algorithm for quickly computing dense rSPR graphs (those containing a significant portion of the full rSPR graph) lies in avoiding the pairwise comparison of non-adjacent trees and thereby shaving off an $O(m)$ factor. The input to our algorithm is a set $\mathcal{T}$ of phylogenetic trees in the $O(n)$-length Newick [37] representation of each tree as a string of characters (assuming integer labels and bounded reasonable values of n, see [38]). These representations are made unique by ordering each tree so that leftmost subtrees contain the smallest alphanumeric label of descendants. We construct a mapping from each tree T_i to its order index in this ordering i. Begin with an empty graph G. For each tree T_i, we first add a vertex i to the graph and then use Corollary 3.4 below to enumerate the $O(n^2)$ neighbors of T_i in the rSPR graph in $O(n^3)$-time. This efficient enumeration procedure is the key step required to achieve our desired running time of $O(mn^3)$. We use the tree to index mappings to determine whether these trees are already vertices of the graph and, if so, add an edge in the graph from T_i to each such neighbor T_j. The high-level steps are as follows, and we show in Theorem 3.1 that this algorithm is correct and can be implemented to run in the stated time.

Construct-rSPR-Graph($\mathcal{T}$)

1. Let G be an empty graph.
2. Let M be a mapping from trees to integers.
3. Let $i = 0$.
4. For each of the m trees:
   (a) Add a vertex i to G representing the current tree T_i.
   (b) Add $T_i \to i$ to M.
   (c) For each of the $O(n^2)$ neighbors of T_i, enumerated using Enumerate-rSPR-Neighbors(T_i):
      i. If the current neighbor T_j is in M then add an edge $(i, M[T_j])$ to G.
   (d) $i = i + 1$.

Theorem 3.1. *The subgraph of the rSPR graph induced by a set $\mathcal{T}$ of m trees with n leaves can be constructed in $O(mn^3)$-time.*

Proof. The correctness of the procedure follows by induction on the number of trees already processed, i, by observing that given the subgraph of vertices $1, 2, \ldots i$ the procedure will construct the entire subgraph of vertices $1, 2, \ldots i+1$.

We implement the graph with an adjacency list representation with integer-labeled vertices that supports $O(\log n)$ edge insertions and lookups (with e.g. red–black trees [39], as the vertex degrees are $O(n^2)$). As described above, the integer labels are simply the order of the input trees. Adding the vertices to the graph requires $O(m)$-time, as they are added in ascending order to the end of the vertex list, which can be stored as a fixed-size array. Adding the $O(mn^2)$ edges to the graph requires $O(mn^2 \log n)$-time. Enumerating the neighbors of T_i requires $O(n^3)$-time for each T_i, for a total of $O(mn^3)$-time. We discuss below, in Section 3.2 how to do so efficiently without considering duplicate neighbors. We store the tree to index mappings for current vertices of G in a trie [40] using the Newick representation. This requires only $O(n)$-time for each neighbor tree (i.e. a total of $O(mn^3)$-time) using a standard nodes-and-pointers representation of the tree and assuming integer leaf labels (a simple $O(mn \log n)$ leaf preprocessing step could be applied to extend this procedure to phylogenetic trees with string labels). Similarly, it takes $O(n)$-time to determine the index of each of the $O(mn^2)$ considered neighbors. Therefore the graph can be constructed in $O(mn^3)$-time, as claimed. □

We implemented this procedure in the C++ program `dense_spr_graph` of the software package `spr_neighbors` [41], which outputs an edge list format graph suitable for input to other software. The construction procedure reduced the time required to compute the 10,395-vertex 7-leaf rSPR graph from 2,104.68 seconds to 12.71 seconds on an Intel Core 2 Duo E7500 desktop running Ubuntu 14.04. Moreover, although we do not study the 135,135-vertex 8-leaf rSPR graph in this paper, our algorithm required only 303.45 seconds to construct it on the same hardware. Constructing the 8-leaf rSPR graph using the previous method required 377,395 seconds (more than 4 days), and thus that method is infeasible for constructing larger rSPR tree graphs. Thus, we believe our fast graph construction procedure will itself be useful for further studies of rSPR graph subsets induced by MCMC credible sets similar to [17], as the algorithm can quickly construct rSPR graphs for any given subset of trees.

3.2. Simulating random walks on the rSPR graph

The uniform random walk moves from a vertex to one of its neighbors uniformly at random, which makes this walk more likely to sample higher degree vertices. In contrast, the Metropolis Hastings (MH) random walk with constant likelihood function proposes a move from a tree T to a neighbor tree S uniformly at random and then accepts the move according to the Hastings ratio, $\min\left(1, \frac{|N(T)|}{|N(S)|}\right)$. The MH random walk is guaranteed to sample each tree uniformly at random and is therefore representative of a phylogenetic MCMC program sampling trees under a uniform prior.

To efficiently simulate the MH random walk, we developed a linear time algorithm for proposing rSPR moves that does not require the rSPR graph to be explicitly built and stored in memory. A naïve approach would require $O(n^3)$ time: $O(n)$ time to generate each of the $O(n^2)$ neighbors of a given tree so that one could be picked uniformly at random. To eliminate an $O(n^2)$ factor, we developed a deterministic ordering of rSPR moves with a one-to-one correspondence to rSPR neighbors, as described in the next paragraph. Given such an order, a uniform neighbor can be selected by its index in $O(n)$ time. We note that the recursive formula of Song [18] for the degree of a tree does not group rSPR moves that move a particular subtree, and thus would still require $O(n^2)$ time to select a specific rSPR neighbor by index.

We consider the distribution of rSPR moves in terms of the number of nodes contained within the subtree to be moved. Recall that a tree with n leaves has $2n-1$ total nodes (ignoring the artificial ρ node). Given a subtree R with x nodes, observe that there are $2n-1-x$ possible locations to regraft R. However, some of these moves will result in the same neighboring tree as other rSPR moves. In particular, where we call the edge connecting the subtree rooted at that node to the rest of the tree the "node's edge", we have:

i. Moving R to its sibling edge results in the same tree, not a neighboring tree,
ii. Moving R to its parent edge results in the same tree,
iii. Moving R to its grandparent edge is the same as moving its aunt to its sibling edge, and
iv. Moving R to its aunt edge is the same as moving its aunt to R's edge.

We prove in Lemma 3.2 that this list is exhaustive, that is, every other pair of R and destination edge e results in a unique rSPR neighbor. To do so, we assign each of the neighbors to one node of the tree, such that any given neighbor can be obtained by moving the subtree rooted at its assigned node. We assign $(2n-1-x)-2$ neighbors to children of the original non-ρ root (lacking both an aunt and a grandparent), and $(2n-1-x)-4$ neighbors to each other non-root node. Let $N(T,u)$ denote the neighbors of T assigned to node u, obtained by moving the subtree R rooted at u. We thus achieve a new method for computing the neighborhood size:

Lemma 3.2. *For a tree T with n leaves,*

$$|N(T)| = \sum_{u \in T} |N(T,u)|,$$

for nodes u of T, where $N(T,u)$ is as defined above, x is the number of nodes in the subtree rooted at u, and:

$$|N(T,u)| = \begin{cases} 2n-x-5 & \text{if } depth(u) > 1 \\ 2n-x-3 & \text{if } depth(u) = 1 \\ 0 & \text{if } depth(u) \leq 0 \end{cases}.$$

Proof. The statement follows if each of the neighbor assignments are disjoint, that is $N(T,u) \cap N(T,v) = \emptyset$, for all nodes u, v of T. So, suppose, for the purpose of obtaining a contradiction, that there exist two nodes u and v of T such that there exists a tree $S \in (N(T,u) \cap N(T,v))$. Then S can be obtained from T by moving the subtrees rooted at u or v. Call these U and V, respectively. This implies that both $T \setminus U = S \setminus U$ and $T \setminus V = S \setminus V$ by the definition of an rSPR operation, where$\setminus$ here indicates removing the subtree and its parent edge and then replacing its former parent node and both adjacent edges with a single edge. Then the rSPR moves that move U or V to obtain S must be nearest neighbor interchanges (NNIs), that is, rSPR moves which move their subtree to one of four locations: their grandparent edge, aunt edge, sibling's left child edge or sibling's right child edge. This implies that, without loss of generality, U is moved to its grandparent edge and V to U's sibling (move type (iii)) or U is moved to its aunt edge and V to U's edge (move type (iv)), a contradiction. Thus the claim holds. □

This neighbor sequence implies a total ordering of rSPR moves such that every rSPR move which moves the same subtree R forms a contiguous subsequence. We can use this fact to quickly select a neighbor uniformly at random for a tree T. Instead of directly computing each neighbor of T, we count the number of neighbors $|N(T,u)|$ in the sequence that can be reached by moving the subtree rooted at each node u of T. We then use this list of neighbor counts to select the position of a uniformly random neighbor of T in the total ordering and the subtree R corresponding to that position. By generating only the neighbors of T induced by moving the subtree R we then quickly identify the specific neighbor of T. We thus achieve the following algorithm to select a neighbor uniformly at random for a tree T:

SELECT-RSPR-NEIGHBOR(T)

1. Compute the degree of T, $|N(T)|$ using Lemma 3.2.
2. Pick a random integer r in the range $[1, |N(T)|]$.
3. Label each node u of T by its preorder number and compute the number of nodes in the subtree rooted at each u.
4. For each tree node u and while $r > 0$:
   (a) Decrease r by $|N(T,u)|$.
   (b) If $r < 0$, let S be the $|r|$-th member of $N(T,u)$ and terminate the for loop.
5. Return the neighbor S.

We now prove that this algorithm is correct and runs in linear time with respect to the number of leaves in the tree.

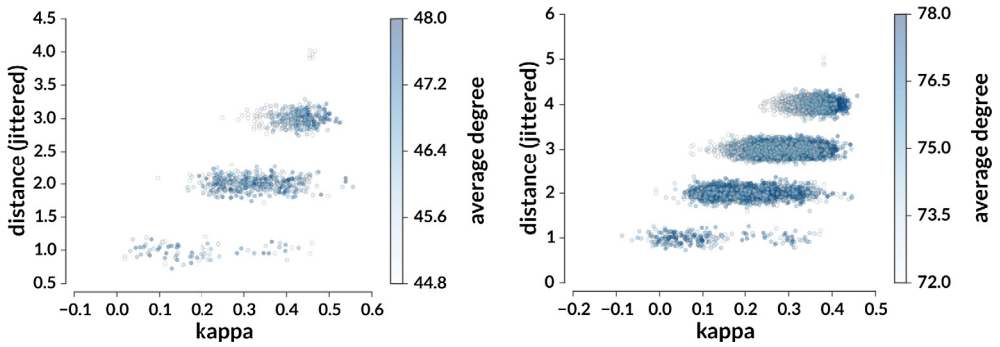
Lemma 3.3. *An rSPR neighbor of a tree T can be chosen uniformly at random in $O(n)$-time using $O(n)$ space (in bits).*

Proof. We apply the above procedure. We use a standard nodes-and-pointers representation of the trees, which can be constructed in $O(n)$-time from a Newick string representation and uses linear space in n. We can compute the degree of T in linear time and space using Lemma 3.2. To efficiently compute $|N(T,u)|$ for each node u of T, we require the number of nodes x in the subtree rooted at u. We pre-compute these by (1) labeling each node with its preorder number in a preorder traversal and (2) summing the number of descendant nodes in a postorder traversal and storing the results in an array indexed by preorder number. Both of these traversals require $O(n)$-time. There are $2n-1 = O(n)$ nodes of T, and $|N(T,u)|$ can be computed in constant time using the subtree sizes. Moreover, the tree S can be found in $O(n)$-time by iterating over the edges of T that are not contained within u's subtree to select the corresponding rSPR destination. Finally, we require linear time to apply the chosen rSPR operation which entails removing a node, adding a node, and updating a constant number of pointers. Thus, the for loop requires linear time. By Lemma 3.2 the chosen tree is an rSPR neighbor of T and is chosen uniformly at random. Therefore, the procedure uses linear time and space and selects an rSPR neighbor of T uniformly at random. □

Observe that this procedure can be easily adapted to explore the full neighborhood of a tree in $O(n^3)$ time, which we use for Theorem 3.1. We call the resulting procedure ENUMERATE-RSPR-NEIGHBORS(T). We thus have the following corollary:

Corollary 3.4. *The rSPR neighbors of a tree T can be enumerated in $O(n^3)$-time.*

We implemented this procedure in the C++ package `random_spr_walk` [42]. We used this procedure to sample random walks on varying size rSPR graphs for our analysis in the next section.



(a) 6 leaves (b) 7 leaves

Fig. 4. Scatter plot of $\kappa(\text{MH}; T_1, T_2)$ values versus $d_{\text{SPR}}(T_1, T_2)$ for the rSPR graph. Color displays the average degree of T_1 and T_2. Distance values randomly perturbed ("jittered") a small amount to avoid superimposed points.

4. Access times of random walks on the rSPR graph can be understood using distance, degree, and curvature

4.1. Computing curvature values

To compute curvature values, we first used `dense_spr_graph` to compute the full rSPR graph for four to seven leaves, as discussed in Section 3.1. We then computed curvatures for pairs of trees directly, by using linear programming [34] to compute the minimal mass transport W_1 using the SAGE [43] front-end to the GLPK [44] solver; code can be found in [45] which grew from the code described in [34]. Specifically, we did this calculation for all representative pairs of trees as described next.

To directly compute curvatures for all $((2n-3)!!)^2$ pairs of trees with n leaves would have required an enormous amount of computation, even for the small values of n we consider here. We instead exploited the fact that pairs of trees which are equivalent modulo label renumbering are symmetric in the rSPR graph and therefore guaranteed to have the same curvature. For example, the pairs $\{((((1,2),3),4),((1,2),(3,4))\}$ and $\{((((1,4),2),3),((1,4),(2,3))\}$ are the same after relabeling, so their curvatures are the same. We thus directly computed curvature values for one representative pair from each such equivalence class, or *tanglegram* [46]; the group-theoretic enumeration methods are described in [47] with a closed formula in [48], and the SAGE [43] and GAP4 [49] code is at [50].

We find a wide variation in curvature among tanglegrams (Fig. 4). Curvature values tended to increase with increasing rSPR distance, and their variance decreased with increasing distance. Neighboring trees achieved minimum curvature values for a given number of leaves, and we found maximum curvature values between trees at maximum distance or one rSPR move closer than the maximum. This suggests that the increased difficulty of moving between trees with a random walk due to distance may be reduced somewhat by larger curvature in the highly connected rSPR graph.

Larger rSPR graphs tended to have pairs of trees with smaller curvature values. Indeed, the 7-leaf rSPR graph contained adjacent pairs of trees with negative curvature. Such pairs indicate difficult paths for phylogenetic searches, which may be exacerbated by likelihood or branch length constraints.

4.2. Access time simulation

The access time for a pair of vertices in a graph is the (random) number of iterations required to go from one of the vertices to the other in a random walk [51]; we were interested in the connection between curvature and access time. In previous work, we computed mean access times (MAT) between pairs of trees in MCMC random walks: the mean number of iterations required to move from one tree to the other. We applied this work to demonstrate the influence of (unrooted) SPR graph structure on real MCMC posteriors sampled with MrBayes [17] using `sprspace` [52].

Here, to gain more insight, we used simulation to approximate the entire access time distribution. Using the procedure of Corollary 3.4, we sampled a 200,000-iteration random walk on the 5-leaf rSPR graph, a 50,000-iteration random walk on the 6-leaf rSPR graph, and a 5,000-iteration random walk on the 7-leaf rSPR graph. Again we use the idea that the access time for a pair of trees with a simple random walk does not depend on the actual labeling of those trees, but rather only on their relative labeling. Thus rather than enumerate access times between trees, which would have required a tremendous amount of memory and computational power to obtain accurate estimates, we enumerate times between pairs of trees in a tanglegram. To calculate the empirical distributions of access times we aggregate all access times for the same tanglegram using our group-theoretic methods [50].

We find that the mean access time between trees T_1 and T_2 is determined by $|N(T_1)|$ and $|N(T_2)|$ (Table 1). Furthermore, plotting the distribution of access times between pairs of trees with respect to their distance and curvature hints that smaller κ slightly shifts the distribution of access times towards larger access times (Fig. 5). We saw a similar effect between pairs of trees with 5 (Fig. 5), 6 (Fig. 6(a)) and 7 (Fig. 6(b)) leaves. We quantify this effect by defining δ_1 to be the difference

Table 1
p-Values for ordinary least squares linear multiple regression of rSPR mean access time against degree and distance (two-tailed t-test of regression coefficient). The p-values for 7 leaves are smaller than the machine precision used to calculate them.

Variable	5 taxa	6 taxa	7 taxa
T_1 degree	2.425e−07	2.726e−55	0
T_2 degree	0.04367	4.302e−21	0
d_{rSPR}	5.026e−09	1.104e−44	0

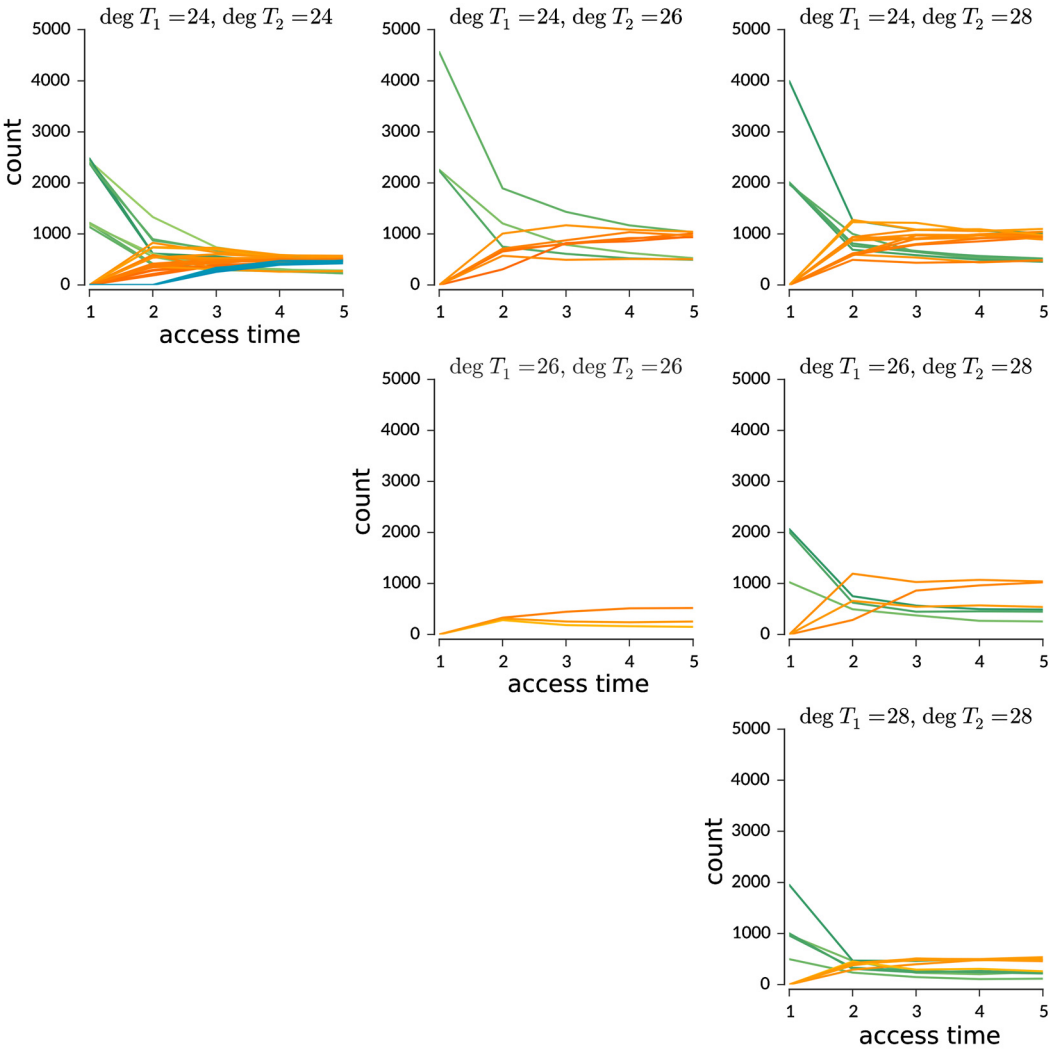


Fig. 5. Distribution of rSPR MH access times (x-axis) for all pairs of 5-leaf trees, grouped by the degree of the trees. Color signifies rSPR distance between the trees, with green, orange, and blue signifying distances of 1, 2, and 3, respectively; the saturation of the color shows coarse curvature $\kappa(\text{MH}; \cdot, \cdot)$, such that increased saturation (i.e. darker color) indicates a smaller κ. Data tables available at https://github.com/matsengrp/curvature/blob/master/prefigs/figure-5-6ab-data.zip. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

between the first pair of access time counts such that the second entry in the pair is nonzero. For example, δ_1 for distance-1 pairs (green lines in Fig. 5) is the count for time 1 minus the count for time 2, while δ_1 for distance-3 pairs (blue lines in Fig. 5) is the count for time 2 minus the count for time 3. Regression finds a clear influence of κ on δ_1 (Table 2). This confirms the intuitive interpretation of $\kappa(T_1, T_2)$ as quantifying the propensity of a random walk to go from T_1 to T_2 relatively directly, certainly before the random walk achieves stationarity. On the other hand, if the random walk starting from T_1 does not quickly arrive at T_2 and instead achieves stationarity, the original position of the random walk is forgotten, and the access time is then a standard exponentially distributed waiting time for an event in a Poisson process (Fig. 6(c) and Fig. 6(d)).

Table 2
p-Values for ordinary least squares linear multiple regression of rSPR δ_1 against degree, distance, and κ (two-tailed t-test of regression coefficient).

Variable	5 taxa	6 taxa	7 taxa
T_1	9.376e−05	2.944e−07	5.51e−09
T_2	0.2366	0.1432	0.1687
d_{rSPR}	5.151e−06	0.0007557	3.276e−23
κ(MH)	4.462e−06	1.436e−22	1.459e−46

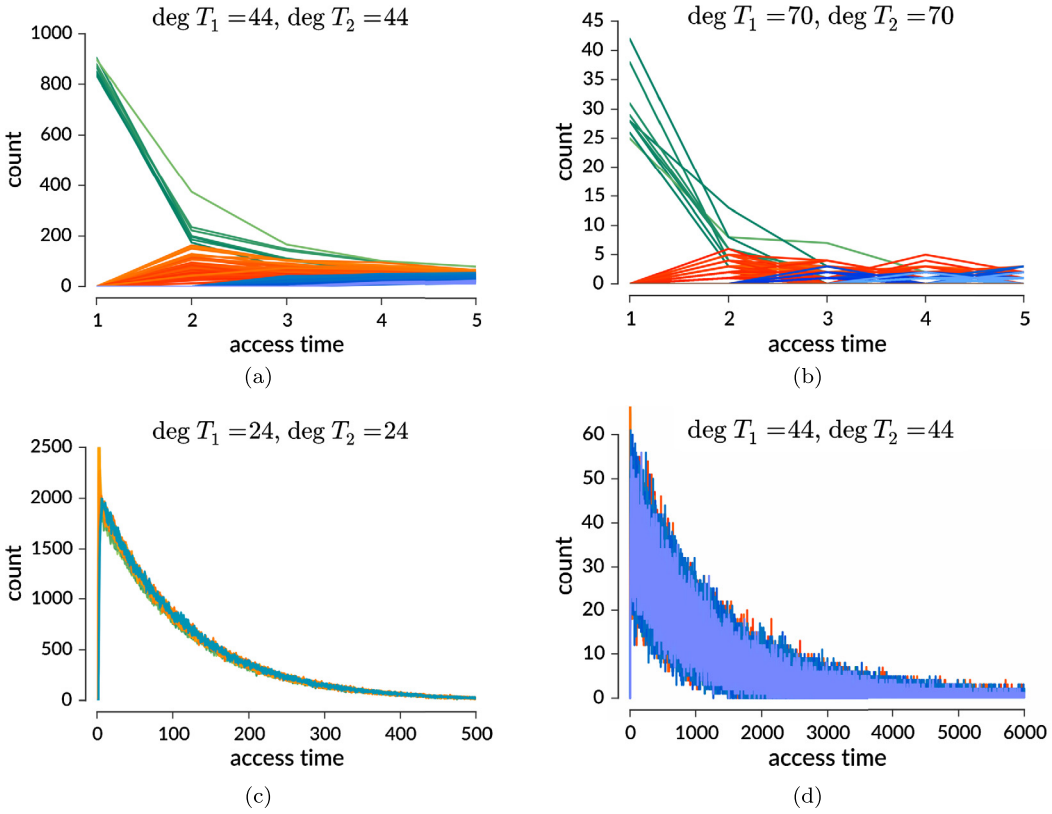


Fig. 6. Distribution of early rSPR MH access times for those pairs of (a) 6-leaf trees with degree 44 and (b) 7-leaf trees with degree 70. Distribution of later rSPR MH access times for those pairs of (c) 5-leaf trees with degree 24 and (d) 6-leaf trees with degree 44. To limit clutter, these plots exclude the pairs that result from modifying pairs of trees with a smaller number of leaves by replacing a given leaf with a given subtree (e.g. we exclude pairs of trees that both contain leaves 1 and 2 as siblings). The additional curves induced by the excluded pairs show a similar influence from curvature. Color signifies rSPR distance between the trees, with green, orange, blue, and light blue signifying distances of 1, 2, 3, and 4, respectively; the saturation of the color shows coarse curvature $\kappa(\text{MH};\cdot,\cdot)$, such that increased saturation (i.e. darker color) indicates a smaller κ. Data tables available at https://github.com/matsengrp/curvature/blob/master/prefigs/figure-5-6ab-data.zip. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The analysis can be reproduced by invoking the SCons (http://scons.org/) build tool and running the cells in an IPython notebook; instructions are in the repository README file.

5. Rooted SPR neighborhoods

Having made the connection between curvature values and access times on rSPR graphs, we now consider curvature theoretically. In this section we consider properties of neighborhoods of trees in rSPR graphs. We will use these properties to bound curvature values in Section 6.

We begin by bounding differences between degrees, and then continue by considering features relevant to the earth mover's distance that we call "squares" and "triangles" in the rSPR graph. Many of our results in this section follow from a characterization of the change in degree and distribution of permissible rSPR moves after an rSPR move is applied.

Our first lemma of this section comes from Song, who characterized the degrees of ladder trees and balanced trees, and showed that the degrees of all other trees fall between these two extremes. Note that every tree degree is quadratic with respect to the number of leaves in the tree.

Lemma 5.1 *(Song [18]). For a tree T with n leaves:*

i. $|N(T)| = 3n^2 - 13n + 14$, *if T is a ladder tree,*
ii. $|N(T)| = 4(n-2)^2 - 2\sum_{m=1}^{n-2} \lfloor \log_2(m+1) \rfloor$, *if T is a balanced tree, and*
iii. $3n^2 - 13n + 14 \leq |N(T)| \leq 4(n-2)^2 - 2\sum_{m=1}^{n-2} \lfloor \log_2(m+1) \rfloor$, *otherwise.*

We can apply Lemma 5.1 to bound the ratio and difference of rSPR degree between any two arbitrary trees with n leaves:

Lemma 5.2. *Let T,S be trees with $n \geq 3$ leaves, and assume w.l.o.g. that $|N(T)| \leq |N(S)|$. Then:*

i. $\frac{|N(T)|}{|N(S)|} \geq 3/4$, *and*
ii. $|N(S)| - |N(T)| \leq n^2 - 5n + 6$.

Proof. To prove (i), we simply note from Lemma 5.1 that the ladder tree achieves the minimum degree, and the balanced tree achieves the maximum degree:

$$\begin{aligned}
\frac{|N(T)|}{|N(S)|} &\geq \frac{3n^2 - 13n + 14}{4(n-2)^2 - 2\sum_{m=1}^{n-2} \lfloor \log_2(m+1) \rfloor} \\
&\geq \frac{3n^2 - 13n + 14}{4(n-2)^2 - 2(n-2)} \\
&= \frac{3n^2 - 13n + 14}{4n^2 - 16n + 16 - 2(n-2)} \\
&= \frac{3n^2 - 13n + 14}{4n^2 - 18n + 20} \\
&\geq \frac{3n^2 - 13n + 14}{4n^2 - 17\frac{1}{3}n + 18\frac{2}{3}} \qquad \forall n \geq 3,
\end{aligned}$$

which is at least 3/4 when $n \geq 3$. Similarly for (ii):

$$\begin{aligned}
\Delta N &= |N(S)| - |N(T)| \\
&\leq (4(n-2)^2 - 2\sum_{m=1}^{n-2} \lfloor \log_2(m+1) \rfloor) \\
&\quad - (3n^2 - 13n + 14) \\
&\leq (4(n-2)^2 - 2(n-2)) - (3n^2 - 13n + 14) \\
&= 4n^2 - 16n + 16 - 2n + 4 - 3n^2 + 13n - 14 \\
&= n^2 - 5n + 6. \quad \square
\end{aligned}$$

The maximum degree difference from Lemma 5.2 may occur between trees that differ by many rSPR operations. In our next lemma, we characterize how the degree of a tree changes after a single rSPR operation. This relation will allow us to improve the degree bounds between adjacent trees. We show that the magnitude of the degree change depends solely on the size of the subtree to be moved, the difference between the depths of its original and new locations, and the size of its original and new neighboring subtrees. See Fig. 7 for an illustration of the lemma.

Lemma 5.3. *Let T and S be trees such that S can be obtained from T by moving a subtree R with k leaves from its position adjacent to subtree U to a location adjacent to subtree V. Let L_1 be the LCA(U, V) in T. Let L_2 be the LCA(U, V) in S (typically $L_1 = L_2$). Let a be the number of intermediate nodes on the path from U to L_2 in S, excluding endpoints. Similarly, let b be the number of intermediate nodes on the path from V to L_1 in T, excluding endpoints. Let i be the number of leaves in U and j be the number of leaves in V, excluding any leaves of R. Then the degrees of T and S differ by:*

$$2\,(k(a-b) + i - j)\,.$$

Proof. The set of permissible rSPR moves changes in four different ways due to the movement of R: (i) subtrees that include nodes on the path from U to L_2 may now be moved into R and its newly introduced parent node, (ii) subtrees that include

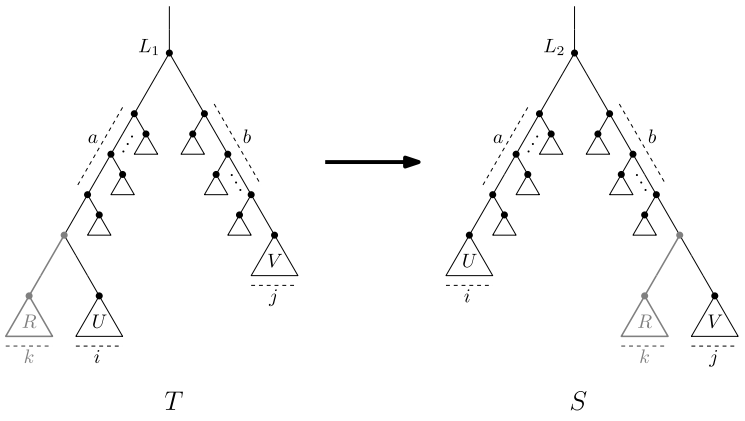


Fig. 7. An rSPR move labeled as in Lemma 5.3. Moving the gray subtree R from its position adjacent to U in tree T to its position adjacent to V in tree S changes the rSPR degree by $2\,(k(a-b)+i-j)$.

nodes on the path from V to L_1 may no longer be moved into R and its parent node, (iii) R's parent subtree may now be moved into U, and (iv) R's parent subtree may no longer be moved into V. No additional moves are introduced or blocked by the original rSPR operation on R.

Recall that a rooted tree with k leaves has $2(k-1)$ edges (recall that we are excluding any "root edge" in these calculations). In the first case there are a subtrees that can now be moved onto the $2k$ edges in R (including its newly introduced parent edge and one of the newly subdivided root edges of V) for a total gain of $2ka$ distinct moves. Similarly, we lose $2kb$ moves in the second case. In the third case, R's parent subtree may now make $2(i-1)$ moves into U. Similarly, we lose $2(j-1)$ moves in the fourth case.

Thus the difference in rSPR degree is $2ka-2kb+2(i-1)-2(j-1)$ as claimed. □

Moreover, we can use these ideas to determine the number of rSPR moves that are, in a sense, independent of a given rSPR move. That is, for two trees S and T differing by a single rSPR move, we wish to know the number of rSPR moves that are applicable to both trees rather than unique to one of the trees. To formalize this concept, consider pairs of trees $T' \in N(T)$ and $S' \in N(S)$ such that $d_{\mathrm{SPR}}(T', S') = 1$. The number of such "squares" involving two adjacent trees will play a key role in our curvature bounds, as they signify pairs of matched paths that help push the curvature of those trees towards 0.

Corollary 5.4. *Continuing with the setting and notation in Lemma 5.3, at least*

$$\gamma := \deg(T) - 2kb - 2(j-1) = \deg(S) - 2ka - 2(i-1)$$

trees in the neighborhood of T can be paired with trees in the neighborhood of S such that $d_{\mathrm{SPR}}(T', S') = 1$ for each (T', S') pair, each neighbor of T is paired with a single neighbor of S, and vice versa. Note that a tree may be a neighbor of both T and S and thus involved in two different pairings.

Proof. By the same arguments as in the proof of Lemma 5.3, γ rSPR moves can be applied to T and S with the same source and target nodes. We consider the parent node of R in both T and S to be the same source node for this purpose. For each such (T', S') pair, there are two cases, depending on whether T' differs from T by R or another subtree. If T' and T differ by a different subtree, then we can move R in T' or S' to obtain the other member of the pair. Thus, $d_{\mathrm{SPR}}(T', S') = 1$. If T' and T differ by R, however, then $T' = S'$. Consider the set of such pairs $(T'_1, S'_1), (T'_2, S'_2), \ldots, (T'_p, S'_p)$. Each tree of the pairs can be obtained from the others by moving R, so we simply pair T'_1 with S'_2, T'_2 with S'_3 and so on, finally pairing T'_p with S'_1. □

We now use Lemma 5.3 to improve the degree change bounds in Lemma 5.2 for two adjacent trees. Surprisingly, we found that the degree change between two adjacent trees may be quadratic in n, that is of the same order as the degree itself. This maximum degree change occurs when a large subtree experiences a large depth change.

Lemma 5.5. *Let T, S be trees with $n \geq 3$ leaves, s.t. $|N(T)| \leq |N(S)|$ and $d_{\mathrm{SPR}}(T, S) = 1$. Then:*

i. $|N(S)| - |N(T)| \leq 2\lfloor \frac{n-2}{2} \rfloor \lceil \frac{n-2}{2} \rceil \leq \frac{1}{2}(n-2)^2$,
ii. $\frac{|N(T)|}{|N(S)|} \geq \frac{5}{6}$, $\forall n \geq 4$, *and*
iii. $\lim_{n\to\infty} \frac{|N(T)|}{|N(S)|} = \frac{6}{7}$.

Proof. We first prove (i). By Lemma 5.3, $|N(S)| - |N(T)| = 2(k(a-b)+i-j)$. Recall that k is the number of leaves in the moved subtree R, i is the number of leaves in R's neighbor U and j is the number of leaves in R's new neighbor V after the SPR move. The length of the path from U's root to the least common ancestor L_2 of U and V in S is a, while b is the length of the path from V's root to the least common ancestor L_1 of U and V in T. Call the former path A.

The degree change is maximized by making L_1 the root of T and thereby minimizing b, namely setting $b=0$. The resulting equation $2(ka+i-j)$ is similarly maximized by including only one leaf in V, setting $i=1$. We then maximally balance the terms in the product ka by adding as many leaves as possible to either R or as children of the path A. Each new leaf of R increases k by 1 and each new child of A lengthens the path and increases a by 1.

There are two cases, depending on whether R is moved to the root (thereby introducing a new level to the tree) or not. If not, then the j leaves of the new neighbor subtree V cannot contribute to increasing k or a. We thus set $j=1$ and split the remaining $n-b-i-j=n-2$ leaves between R and A in as balanced a way as possible, maximizing the product ka and giving (i). Note that this corresponds to moving the bottom subtree of $\lfloor \frac{n-2}{2} \rfloor$ or $\lceil \frac{n-2}{2} \rceil$ leaves in a ladder tree to the root–most leaf of the tree.

If R is moved to the root, then the new neighbor subtree V is the entire tree excluding R itself. The root–most leaf of the tree is on the path from U to L_2 in S and thus a child of A, despite also being a child of L_1 in T. In this case, we have an additional leaf over the previous case that can be added to R or lengthen A, increasing the product ka at the cost of increasing j. This corresponds to moving the bottom subtree of $\lfloor \frac{n}{2} \rfloor$ or $\lceil \frac{n}{2} \rceil$ leaves in a ladder tree to the root. Namely, we have $2(ka+1-j)$, where $j=n-k=a+1$. Let $\Delta N = |N(S)| - |N(T)|$. If we move the additional leaf, we have:

$$\begin{aligned}\Delta N &\leq 2\left(\left\lceil \frac{n}{2} \right\rceil \left\lfloor \frac{n-2}{2} \right\rfloor + 1 - \left(\left\lfloor \frac{n-2}{2} \right\rfloor + 1\right)\right) \\ &= 2\left\lfloor \frac{n-2}{2} \right\rfloor \left\lceil \frac{n-2}{2} \right\rceil,\end{aligned}$$

like before. Similarly, if we do not move the additional leaf, we also have:

$$\begin{aligned}\Delta N &\leq 2\left(\left\lceil \frac{n-2}{2} \right\rceil \left\lfloor \frac{n}{2} \right\rfloor + 1 - \left(\left\lceil \frac{n-2}{2} \right\rceil + 1\right)\right) \\ &= 2\left\lfloor \frac{n-2}{2} \right\rfloor \left\lceil \frac{n-2}{2} \right\rceil,\end{aligned}$$

proving (i).

The relative change in degree, $\frac{|N(T)|}{|N(S)|}$, can also be written as $\frac{|N(T)|}{|N(T)|+(|N(S)|-|N(T)|)}$. By (i), we have that $|N(S)| - |N(T)| \leq \frac{1}{2}(n-2)^2$, so $\frac{|N(T)|}{|N(S)|} \geq \frac{|N(T)|}{|N(T)|+\frac{1}{2}(n-2)^2}$. This bound is minimized when $|N(T)|$ is minimized, and recall by Lemma 5.1 that $|N(T)|$ is bounded below by $3n^2-13n+14$. Thus

$$\begin{aligned}\frac{|N(T)|}{|N(S)|} &\geq \frac{3n^2-13n+14}{3n^2-13n+14+\frac{1}{2}(n-2)^2} \\ &\geq \frac{3n^2-13n+14}{3.5n^2-15n+16}.\end{aligned}$$

Statements (ii) and (iii) follow from this bound. □

In our next lemma of this section, we bound the number of neighbors shared by two adjacent trees. The number of such "triangles" involving two adjacent trees has a key role in determining whether their curvature is positive, as mass assigned to a shared neighbor may not require a mass transport cost. Note that these shared neighbors are exactly the trees from the second case of Corollary 5.4 and an optimal mass transport pairing will pair mass at these trees. We show that the maximum number of shared neighbors is linear in the number of leaves. In other words, only a small proportion of any two neighborhoods are shared and the relative size of shared neighborhoods grows much more slowly than the quadratic scaling of total neighborhood sizes.

Lemma 5.6. *Let T and S be trees such that $d_{\mathrm{SPR}}(T,S)=1$. Then $|N(T)\cap N(S)| \leq 6n-17$.*

Proof. T and S differ by one rSPR move that moves a subtree R. Pick a neighbor $U \in N(T)\cap N(S)$ of both T and S (this intersection is not empty: T and S are different, so R contains at most $n-2$ of the leaves, thus there must be at least one other tree U obtained by moving R in T and S). Then either (i) T and U differ in the location of R, or (ii) T and U differ in the location of another subtree Q. In the latter case, S is reached from U by moving Q in a manner that displaces R from its original position, because T and S differ only in the location of R and $d_{\mathrm{SPR}}(T,U) = d_{\mathrm{SPR}}(S,U) = 1$. Thus, $T|(X \setminus L(Q)) = S|(X \setminus L(Q))$, where $L(Q)$ is the label set of Q. Then leaves $r' \in R$, $q' \in Q$, and $v' \in V$, for some subtree V, form a triple of T and a different triple in S. This incompatible triple can be resolved in at most $6n-17$ ways, the

maximum of which is reached when Q, U, and R are themselves a "triple" of subtrees. By Lemma 3.2, each of the subtrees is assigned to at most $2n-6$ unique moves. Moreover, one additional overlapping move also moves one of the subtrees (that of the aunt of the LCA of the three subtrees). The number of shared neighbors is thus at most $3(2n-6)+1 = 6n-17$. Note that this bound is tight when, for example, T and S are ladders with a different configuration of 3 leaves at maximum depth. □

In our final two lemmas of this section, we consider the distribution of sizes of subtrees moved by rSPR operations from a given tree. First, we bound the number of neighbors of a tree that are induced by rSPR operations involving a subtree with three or more leaves. In particular, we show that the majority of neighbors are induced by moving small subtrees (one or two leaves). Along with our observation from Lemma 5.3 that the magnitude of the degree change induced by an rSPR operation depends on the subtree size and height change, this implies that most rSPR neighbors have a similar degree. We say that a subtree is *small* if it has two leaves or one leaf, and that a *small* rSPR move is one that moves a small subtree.

Lemma 5.7. *Let T be a rooted tree with $n \geq 4$ leaves. Let $N_S(T)$ be the neighbors of T induced by small rSPR moves. Then $\frac{|N_S(T)|}{|N(T)|} > \frac{1}{2}$.*

Proof. Recall our distribution of rSPR moves from a tree in Lemma 3.2. Each subtree at a depth > 1 from the root with x nodes induces $2n-x-5$ rSPR neighbors, while the children of the root induce $2n-x-3$ neighbors. Moreover, every tree has n 1-leaf subtrees. This implies that the number of small rSPR moves is determined by the number of small subtrees, with a possible difference of two moves per small child of the root.

A ladder tree has the smallest proportion of small subtrees, so we first bound the number of small subtree moves that can be applied to a ladder tree. A ladder tree with n leaves has n 1-leaf subtrees and one 2-leaf subtree. One of the 1-leaf subtrees is a child of the root. A 2-leaf subtree contains three nodes. Therefore, a ladder tree induces $n(2n-1-5)+2+(2n-3-5) = 2n^2-4n-6$ small subtree moves.

Now, recall that the maximum degree of any tree is less than $4n^2-16n+16$ by Lemma 5.1. We then have that $2(2n^2-4n-6) = 4n^2-8n-12 \geq 4n^2-8n-12+(28-8n) = 4n^2-16n+16$, for all $n \geq 4$. Therefore more than half of the neighborhood of any tree is induced by small rSPR moves. □

Now, we bound the average size of the moved subtrees (in terms of the number of leaves) from rSPR moves according to our distribution. Note that this average ignores some possible moves that are equivalent to moves of smaller subtrees. This occurs because multiple NNI moves may lead to the same tree, as described previously, and each equivalent move may involve a subtree with a different number of leaves. Our distribution includes only one of these equivalent moves. For example, from the balanced tree with 4 leaves, we do not consider moves of the root's 2-leaf child subtrees because they are equivalent to moves of the leaves, as explained by our distribution.

Lemma 5.8. *Let T be a rooted tree with $n \geq 4$ leaves. Let $N_i(T)$ be the neighbors of T induced by rSPR moves of an i-leaf subtree, where the moved subtree is defined according to our distribution of rSPR moves. Then*

$$\frac{\sum_i i|N_i(T)|}{|N(T)|} \leq \frac{n+3}{4}.$$

Proof. Again recall our distribution of rSPR moves from a tree in Lemma 3.2. Recall from the proof of Lemma 5.7 that a ladder tree has the smallest proportion of small subtrees. This implies that a ladder tree has the greatest average moved subtree size (in terms of the number of leaves) with respect to our distribution. There are n subtrees with one leaf, and one subtree each with $2, 3, \ldots n-1$ leaves. We ignore the root subtree, which does not induce any rSPR moves. Thinking of the ratio as a weighted average of i, we can split the numerator and the denominator of the average into terms for each subtree contributing to the values for $i \leq 2$ and those with $i > 2$. The contribution to $|N_i(T)|$ is $\geq 2n-3-5$ for the first set of terms, and the contribution to $|N_i(T)| < 2n-3-5$ for the second. By replacing the contribution to $|N_i(T)|$ in each case by $2n-3-5$ we decrease the weight for $i \leq 2$ and increase the weight on $i > 2$. Note that this replacement does not modify the $i = 2$ term (which was already exactly $2n-3-5$) and decreases the weight of only the smallest terms (which must already be below the average size of the moved subtrees), so this provides an upper bound on the mean moved subtree size. The bound for large subtrees only is $(3+(n-2))/2 = (n+1)/2$. By Lemma 5.7, at least half of the rSPR moves involve a small subtree. Thus, the average size of a moved subtree over all SPR operations from our distribution on a given tree is at most the weighted average of $1/2 + ((n+1)/2)/2 = (n+1)/4 + 1/2 = (n+3)/4$. □

Further, note that our distribution does not necessarily minimize the moved subtree size among equivalent moves. It may be interesting in future work to improve our bound by explicitly minimizing the moved subtree size in such a manner.

6. Curvature

In this section we apply the rSPR neighborhood properties we proved in Section 5 to derive bounds on curvature. In particular, we consider properties of the uniform (a.k.a. isotropic) random walk on the n-leaf rSPR graph.

Recall that the uniform random walk begins at a tree T and moves to a tree uniformly at random from $N(T)$. Recall that the coarse uniform random walk curvature between two trees T and S is $\kappa(T,S) := 1 - \frac{W_1(m_T,m_S)}{d(T,S)}$, where W_1 is the mass transport term (3). For the uniform random walk, m_T is the probability measure assigning a mass of $\frac{1}{|N(T)|}$ to each of T's neighbors.

We showed in Lemmas 5.7 and 5.8 that most adjacent trees differ by a small subtree, raising the question of what the curvature is between such pairs. In our first theorem of this section, we show that the curvature between two adjacent trees tends towards 0 as the size of the trees increases but the size of the subtree moved to transition between the two trees remains the same size. To do so, we prove that the mass transport term between adjacent trees T and S is dominated by probability moved between neighbors of T and S that are also adjacent. The probability transported between members of these squares (Corollary 5.4) increases more quickly with the number of leaves n than the probability transported between shared neighbor triangles (Lemma 5.6), and outweighs the effect of long distance probability moved between more distant neighbors or due to a difference in the degree of T and S. Thus this "flatness-in-the-limit" theorem helps explain our empirical results in Section 4.1 showing that the distribution of curvature pairs tends more towards 0 as n increases.

Theorem 6.1. *Fix a positive integer k and let R be a tree with k leaves. Let $\{T_n \mid n > k\}$ be a sequence of trees with n leaves all containing R, and let $\{S_n \mid n > k\}$ be the same sequence T_n but with R cut off and attached at a different location. Then $\lim_{n\to\infty} \kappa(T_n, S_n) = 0$ for the uniform random walk on the rSPR graph.*

Proof. Let $W_{1,n}$ be the mass transport term with respect to T_n and S_n, that is $W_1(m_{T_n}, m_{S_n})$. Because $d(T_n, S_n) = 1$, we will prove the theorem by showing that the mass transport term $W_{1,n}$ sits between two bounds, each of which has limit 1 as n goes to infinity.

To start we demonstrate the theorem in the case that T_n and S_n have the same number of neighbors. First we claim that $W_{1,n}$ is bounded above by $(|N(T_n)| + \mathrm{O}(kn))/|N(T_n)|$ by exhibiting a mass transport program satisfying that bound. Let (T'_n, S'_n) be any of the γ pairs of neighbors of (T_n, S_n) which are one rSPR move apart as per Corollary 5.4. We pair these trees in the mass transport. There are $\mathrm{O}(kn)$ trees unmatched by this pairing, because b and a (in the nomenclature of Corollary 5.4) are bounded by n. We can pair each of the unmatched trees arbitrarily with another tree of distance at most 3. Thus, $W_{1,n}$ is bounded above by $(|N(T_n)| + \mathrm{O}(kn))/|N(T_n)|$.

A lower bound is also available because we can't do better than distance 1 for all trees except for shared neighbors, of which there are $\mathrm{O}(n)$ by Lemma 5.6. By ignoring these trees we get a lower bound of $(|N(T_n)| - \mathrm{O}(n))/|N(T_n)|$ for $W_{1,n}$.

The desired control of $W_{1,n}$ is thus obtained because $|N(T_n)|$ is quadratic in n.

Now we prove the theorem when the number of neighbors differ. Assume without loss of generality that $|N(T_n)| < |N(S_n)|$. By Lemma 5.3, $|N(S_n)| - |N(T_n)| = 2(k(a-b) + i - j)$, where each of $\{a, b, i, j\}$ is less than n. Thus, $|N(S_n)| - |N(T_n)| = \mathrm{O}(kn)$. We again pair neighbor T'_n of T with neighbor S'_n of S such that $d_{\mathrm{SPR}}(T'_n, S'_n) = 1$ but, as $|N(T_n)| < |N(S_n)|$ we can only account for at most $|N(T_n)|/|N(S_n)|$ of the mass directly and may have to move the $(|N(S_n)| - |N(T_n)|)/|N(S_n)|$ remainder to trees of distance at most 3. Thus, $W_{1,n}$ is bounded above by $(|N(T_n)| + \mathrm{O}(kn))/|N(S_n)| = (|N(S_n)| + \mathrm{O}(kn))/|N(S_n)|$. We again bound $W_{1,n}$ from below with $(|N(T_n)| - \mathrm{O}(n))/|N(T_n)|$ by ignoring the mass in common neighbors of T_n and S_n. The theorem again follows because $|N(T_n)|$ is quadratic in n. □

Next we note a simple and rough bound on the curvature of two trees with respect to their distance. This bound arises naturally in any metric space with integer distances.

Lemma 6.2. *Let T and S be two trees. Then:*

$$\frac{-2}{d_{\mathrm{SPR}}(T,S)} \le \kappa(T,S) \le \frac{2}{d_{\mathrm{SPR}}(T,S)}.$$

Proof. Observe that the distance between neighbors of T and S is bounded between $d_{\mathrm{SPR}}(T,S) - 2$ and $d_{\mathrm{SPR}}(T,S) + 2$. Then, any mass transported between a probability distribution m_T on neighbors of T and m_S on neighbors of S is transported a distance at least $d_{\mathrm{SPR}}(T,S) - 2$. This implies that the mass transport cost $W_1(m_T, m_S)$ is at least $d_{\mathrm{SPR}}(T,S) - 2$, for any m_T and m_S. For the curvature upper bound, we then have $\kappa(T,S) \le 1 - \frac{d_{\mathrm{SPR}}(T,S)-2}{d_{\mathrm{SPR}}(T,S)} = \frac{2}{d_{\mathrm{SPR}}(T,S)}$. The lower bound follows similarly by using the maximum distance of $d_{\mathrm{SPR}}(T,S) + 2$. □

We now obtain a tighter bound on the maximum curvature between two adjacent trees. We do so by considering trees with a maximum intersecting set of neighbors using Lemma 5.6.

Lemma 6.3. *The maximum curvature between two adjacent trees with n leaves is $\frac{6n-17}{3n^2-13n+14}$.*

Proof. The maximum curvature between adjacent trees T and S occurs when their neighborhoods have maximum overlap and all other tree pairs are at distance 1. By Lemma 5.6 the maximum overlap is $6n - 17$. The amount of overlapping mass

in the shared neighbors of T and S is thus $\frac{6n-17}{\max(|N(T)|,|N(S)|)}$. The minimum mass transfer cost is thus $1-\frac{6n-17}{\max(|N(T)|,|N(S)|)}$. This is minimized when $|N(T)|=|N(S)|$ are as small as possible, that is T, S are ladders and $|N(T)|=3n^2-13n+14$.

The maximum curvature is thus $1-\frac{|N(T)|-(6n-17)}{|N(T)|}=\frac{6n-17}{|N(T)|}=\frac{6n-17}{3n^2-13n+14}$. □

This bound is tight and has been verified computationally for $n \leq 7$.

It is more difficult to obtain a closer bound on the maximum curvature of nonadjacent trees. Lemma 6.2 suggests that more distant pairs of trees should have smaller curvatures than close trees, because neighborhood effects decrease with respect to the increasing distance. However, our experiments with $n \leq 7$ suggest that maximum curvature tends to increase with distance (with respect to a fixed n), as a far greater fraction of the neighbors approach each other as the distance increases. Indeed, for $5 \leq n \leq 7$ the maximum curvature is obtained by pairs of trees at one less than the maximum distance. Moreover, nearly all of the neighbors of these pairs approach each other. We thus conjecture the following:

Conjecture 6.4. *Let k_n be the maximum curvature between two trees with n-leaves. Then:*

i. $k_n \leq \frac{2}{\Delta_{rSPR}(n)-1}$, *and*
ii. $k_n \sim \frac{2}{\Delta_{rSPR}(n)-1}$.

Proving or disproving this conjecture would go a long way toward understanding the effects of relative distance on curvature. However, we suspect that this will require a greater understanding of the distribution of pairwise distances between tree neighborhoods than is currently known.

Next, we bound the minimum curvature of two adjacent trees by bounding the amount of probability mass that cannot be moved between adjacent neighbors.

Lemma 6.5. *The curvature between adjacent trees with n leaves is at least*

$$\frac{-n^2+2n+1}{3.5n^2-15n+16}.$$

Proof. In light of Corollary 5.4, the optimal mass transport cost is maximized (and therefore curvature minimized) across adjacent trees T and S by a combination of two effects: trees that cannot be paired at distance 1 and mass that must be moved between unpaired trees due to differing degrees of T and S. As we will show, these effects can be maximized simultaneously. To bound these effects, let μ be the maximum (across T and S) proportion of mass that cannot be moved between adjacent neighbors of those trees. Pairs of neighbors of adjacent trees are at most distance 3 apart, so μ of the mass is moved a distance at most 3. The remaining $1-\mu$ of the mass is moved between adjacent trees. Thus, we can bound the mass transport cost from above by $3\mu+1(1-\mu)=1+2\mu$. This gives a lower bound of $1-(1+2\mu)/1=-2\mu$ on the curvature.

By Lemmas 5.3 and 5.5, the latter effect is maximized when the relative degree change is maximized. By Corollary 5.4, there are at least $\gamma := |N(T)|-2ka-2(i-1)$ paired trees, bounding the former effect. We now construct a pair of trees that maximizes both effects. Let S be the ladder tree with degree $3n^2-13n+14$ and T be the adjacent tree constructed by moving the lower $\lfloor \frac{n}{2} \rfloor$ leaves of S to the root. T has degree at most $3.5n^2-15n+16$. There are thus $2ka+2(i-1)=2\left(\lfloor \frac{n}{2} \rfloor \lceil \frac{n-2}{2} \rceil+(1-1)\right) \leq \frac{1}{2}n^2-n+\frac{1}{2}$ unpaired neighbors, the maximum possible. Moreover, as shown by Lemma 5.3 this pair of trees obtains the maximum (absolute and relative) degree change. Thus, the maximum μ is:

$$\frac{\frac{1}{2}n^2-n+\frac{1}{2}}{3.5n^2-15n+16}.$$

Note that any pair of trees with a smaller degree change will necessarily reduce the numerator by at least the same amount as the denominator, allowing us to use the degree of T here rather than assuming the minimum degree of S. The claim follows from multiplying this value by -2. □

We further observe that the limit of our curvature lower bound is $-\frac{2}{7}$. Complete enumeration with $n \leq 7$ shows that no pair of trees have curvature less than $-\frac{2}{5}$; our bound meets or exceeds this value for $n > 7$. Moreover, the rSPR distance is a metric, so this bounds the curvature for arbitrary pairs of trees (Proposition 19 of [26]). This directly leads to the following Corollary:

Corollary 6.6. *The curvature between two trees is at least $-\frac{2}{5}$.*

Note that this bound is not tight (at least for small n) as it is rarely necessary to transport mass the maximum distance between unpaired trees. We also note that the lower bounds in this section do not follow from the more general setting

described in [53]. However, the pair of trees used in the proof of Lemma 6.5 will always have negative curvature, for all $n \geq 7$. This follows by assuming that each of the unpaired trees is moved the minimum distance of one, that is, the curvature for this pair of trees falls between $-\mu$ and -2μ.

Corollary 6.7. *For all $n \geq 7$, there exist two adjacent trees T and S with n leaves such that $\kappa(T, S) < 0$.*

We next bound the difference between the coarse and asymptotic curvatures. Recall that $\kappa_p(T, S)$ is the coarse Ricci–Ollivier curvature between trees T and S with respect to the lazy walk that remains at a given tree with probability $1 - p$ and moves with probability p. For the lazy uniform random walk, m_T is now $T \cup N(T)$, with each neighbor assigned mass $\frac{p}{|N(T)|}$ and T assigned the remaining $1 - p$ mass. The asymptotic Ricci–Ollivier curvature $\mathrm{ric}(T, S)$ is $\lim_{p \to 0} \kappa_p(T, S)/p$. As we now prove, these two notions of curvature differ only by a small factor inversely proportional to the maximum degree of T and S.

Lemma 6.8. *Let T and S be trees with n leaves. Then:*

i. $\mathrm{ric}(T, S) = \kappa(T, S)$, *if* $d_{\mathrm{SPR}}(T, S) > 1$,
ii. $\kappa(T, S) \leq \mathrm{ric}(T, S) \leq \kappa(T, S) + \frac{2}{\max(|N(T)|, |N(S)|)}$, *if* $d_{\mathrm{SPR}}(T, S) = 1$.

Proof. We first prove the lower bound in the uniform case, that is $\kappa_{T,S}(\leq) \mathrm{ric}(T, S)$. Let $W_1(T, S)$ be the mass transport cost in the uniform case, and $W_1'(T, S)$ be the same for the lazy uniform case with parameter p. Recall that $\kappa(T, S) = \kappa_1(T, S) = 1 - \frac{W_1(T,S)}{d_{\mathrm{SPR}}(T,S)}$, and $\kappa_p(T, S)/p = \left(1 - \frac{W_1'(T,S)}{d_{\mathrm{SPR}}(T,S)}\right) \Big/ p$. Observe that

$$W_1'(T, S) \leq pW_1(T, S) + (1 - p)\, d_{\mathrm{SPR}}(T, S),$$

by the simple mass transport program obtained by treating the mass at T and S as separate from that of the neighbors. Then:

$$\begin{aligned}
\frac{\kappa_p(T, S)}{p} &= \left(1 - \frac{W_1'(T, S)}{d_{\mathrm{SPR}}(T, S)}\right) \Big/ p \\
&\geq \left(1 - \frac{pW_1(T, S) + (1 - p) d_{\mathrm{SPR}}(T, S)}{d_{\mathrm{SPR}}(T, S)}\right) \Big/ p \\
&= \frac{1}{p} - \frac{W_1(T, S)}{d_{\mathrm{SPR}}(T, S)} - \frac{1 - p}{p} \\
&= 1 - \frac{W_1(T, S)}{d_{\mathrm{SPR}}(T, S)} \\
&= \kappa(T, S).
\end{aligned}$$

For the upper bound, we observe that

$$W_1'(T, S) \geq pW_1(T, S) + (1 - p)\, d_{\mathrm{SPR}}(T, S) - \frac{2p}{\max(|N(T)|, |N(S)|)},$$

as at most $p/\max(|N(T)|, |N(S)|)$ of the mass can remain at each of T and S, paired with the lazy remainder. The upper bound then follows analogously to the lower bound. Moreover, no mass can remain at T or S when $d_{\mathrm{SPR}}(T, S) > 1$, in which case the curvatures are equal. □

Finally, we bound the difference between the curvature of the uniform random walk $\kappa(T, S)$ and that of the Metropolis–Hastings (MH) random walk $\kappa(\mathrm{MH}; T, S)$. Recall that this random walk proposes a move from a tree T to a neighbor tree S uniformly at random and then accepts the move according to the Hastings ratio, which in this case is $\min\left(1, \frac{|N(T)|}{|N(S)|}\right)$. The mass distribution for the MH random walk thus leaves a portion of mass at the origin tree, proportional to the relative degree difference of its higher degree neighbors. In Lemma 6.9 we bound the effect of this mass on curvature and thereby bound the difference in curvatures of the two random walks. Note that the same statement and proof of Lemma 6.8 holds with $\kappa(T, S)$ and $\mathrm{ric}(T, S)$ replaced by the MH curvatures $\kappa(\mathrm{MH}; T, S)$ and $\mathrm{ric}(\mathrm{MH}; T, S)$, respectively.

Lemma 6.9. *Let T and S be trees with $n \geq 4$ leaves. Then:*

$$\kappa(T, S) - \frac{1}{3d_{\mathrm{SPR}}(T, S)} \leq \kappa(\mathrm{MH}; T, S) \leq \kappa(T, S) + \frac{1}{3d_{\mathrm{SPR}}(T, S)}, \text{ and}$$
$$\kappa(T, S) - 1/6 \leq \kappa(\mathrm{MH}; T, S) \leq \kappa(T, S) + 1/6.$$

Proof. We first prove the lower bound. By Lemma 5.5, the quotient of degrees for two adjacent trees $\geq \frac{5}{6}$. Thus, the Hastings ratio is always $\geq \frac{5}{6}$. This implies that at most $\frac{1}{6}$ of the mass remains at tree T in the mass distribution. Let $m_T(z)$ and $m_S(w)$ be the mass assigned for the uniform random walk and $m'_T(z)$ and $m'_S(w)$ be the mass assigned for the MH random walk, for each vertex $z \in N(T)$ and $w \in N(S)$. We construct an upper bound on $W_1(m'_T, m'_S)$ by moving mass according to $W_1(m_T, m_S)$ where possible, and moving the remainder either from T to S, from T to a neighbor of S, or from a neighbor of T to S. That is, for each $W_1(m_T, m_S)$ assignment $\xi(z, w)$, we send $\xi'(z, w) = \xi(z, w) \min\left(\frac{m'_T(z)}{m_T(z)}, \frac{m'_S(w)}{m_S(w)}\right)$ of the mass from z to w. The remaining $\xi(z, w) - \xi'(z, w)$ of the mass is moved from T to S, T to w, and z to S in the respective proportions $\xi(z, w) \max\left(\frac{m'_T(z)}{m_T(z)}, \frac{m'_S(w)}{m_S(w)}\right) - \xi'(z, w)$, $\xi(z, w) \min\left(0, \frac{m'_T(z)}{m_T(z)} - \frac{m'_S(w)}{m_S(w)}\right)$, and, $\xi(z, w) \min\left(0, \frac{m'_S(w)}{m_S(w)} - \frac{m'_T(z)}{m_T(z)}\right)$. The maximum possible mass that is not moved according to $W_1(m_T, m_S)$ is $\frac{1}{6}$. Moreover, the affected mass must be moved through at most two additional trees. Then, $W_1(m'_T, m'_S) \leq W_1(m_T, m_S) + \frac{2}{6}$. We now have:

$$\begin{aligned}\kappa(\mathrm{MH}; T, S) &\geq 1 - \frac{W_1(m_T, m_S) + \frac{1}{3}}{d_{\mathrm{SPR}}(T, S)} \\ &\geq \kappa(T, S) - \frac{1}{3 d_{\mathrm{SPR}}(T, S)}.\end{aligned}$$

In the case that $d_{\mathrm{SPR}}(T, S) = 1$, the affected mass must be moved through only at most one additional tree, as T and S are adjacent. We thus obtain the lower bound of $\kappa(T, S) - \frac{1}{6}$ in this case.

We obtain the upper bounds similarly to the lower bounds, by observing that the affected at most $\frac{1}{6}$ of the mass may move through at most two fewer trees (i.e. directly between T and S rather than a pair of neighbors at distance $d_{\mathrm{SPR}}(T, S) + 2$ from each other). Again, this is at most one fewer tree when $d_{\mathrm{SPR}}(T, S) = 1$. □

7. Conclusion and future work

In summary, we have gone beyond graph diameter and vertex degree to substantially advance the understanding of the phylogenetic rSPR graph. We did so by developing the first theoretical and computational frameworks to bound and compute Ricci–Ollivier curvature of the rSPR graph. We found that curvature, along with degree and distance, determine the early dynamics of hitting times for random walks. Moreover, we proved that rSPR graph degree changes depend on the product of the size of the regrafted subtree with its change in depth. This product is quadratic in the number of leaves in the trees, the same order as the degree. As a result of this dependence, some areas of the rSPR graph are locally negatively curved but the graph tends toward flatness with respect to the majority of rSPR moves which move small subtrees. Finally, we proved that the coarse and asymptotic definitions of Ricci–Ollivier curvature are closely related with respect to uniform and Metropolis–Hastings walks on the rSPR graph.

In this data-free setting the stationary distribution is, unlike with real data, quite evenly spread over all trees. Correspondingly, we found that the influence of curvature is small in this case (Fig. 5) and that the probability of the target node in the stationary distribution predominantly determines access times for pairs of trees (Fig. 6). However, it is well known that MCMC takes a long time to approximate real phylogenetic posterior distributions even when the Bayesian credible set is small, and in fact our previous work showed significant SPR graph influence on the mixing time for phylogenetic MCMC for credible sets that had tens, hundreds or thousands of trees [17]. Thus, our next step will be to investigate curvature of MCMC with nontrivial likelihood functions, which will reduce the posterior distribution to a more realistic effective size, and in certain cases will lead to significant "bottlenecks" like those we have observed in real data. In those cases the curvature between two trees at either end of a bottleneck will describe how difficult it is to traverse the bottleneck. Indeed, in both the setting of challenging real data and in simulation with a nontrivial likelihood function, the stationary distribution takes a long time to be achieved, and thus the curvature will have a substantially greater impact on the overall hitting times rather than being a short prelude to waiting for a Poisson process event as it is here (Fig. 6). One avenue of exploration will be to follow previous work [10] by taking a chain where tree topology likelihoods are maximized across branch length, as the classical phylogenetic likelihood [1] requires both branch length and topology.

One could extend this work in several other directions. Now that we have established the foundations of using curvature to understand graphs relevant for phylogenetic inference, many graph structures remain to be explored. In particular, the case of unrooted SPR, which is also commonly used in phylogenetic search, is an obvious next step. One could also explore random walks on ranked trees [54] and graphical models of tree space relevant for phylogenetic algorithms such as BEAST [55] that infer rooted "time-trees." Random walks on other discrete structures such as partitions [56] that can be expressed as certain types of trees may also form interesting subjects for future work.

Finally, we conjectured that the maximum curvature for an rSPR graph with n leaves is inversely proportional to the diameter of the graph. We suspect that proving or disproving this conjecture will require a new understanding of the effect of multiple rSPR moves in tandem, just as many of our results came from a better understanding of the impact of a single rSPR move. Such an understanding may also lead to improved curvature bounds or a means for computing them more efficiently.

Acknowledgements

The authors would like to thank Alex Gavruskin, Vladimir Minin, and Bianca Viray for helpful discussions. They are also grateful to the authors of the SAGE and GAP4 software, especially Alexander Hulpke. The authors would like to thank the conference organizers, reviewers, and attendees of ANALCO16 for helpful comments and discussions that greatly improved this full version of our work.

References

[1] J. Felsenstein, Evolutionary trees from DNA sequences: a maximum likelihood approach, J. Mol. Evol. 17 (6) (1981) 368–376.
[2] C. Lakner, P. Van Der Mark, J.P. Huelsenbeck, B. Larget, F. Ronquist, Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics, Syst. Biol. 57 (1) (2008) 86–103.
[3] S. Höhna, A.J. Drummond, Guided tree topology proposals for Bayesian phylogenetic inference, Syst. Biol. 61 (1) (2012) 1–11.
[4] F. Ronquist, M. Teslenko, P. van der Mark, D.L. Ayres, A. Darling, S. Höhna, B. Larget, L. Liu, M.A. Suchard, J.P. Huelsenbeck, MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space, Syst. Biol. 61 (3) (2012) 539–542.
[5] R. Bouckaert, J. Heled, D. Kühnert, T. Vaughan, C.-H. Wu, D. Xie, M.A. Suchard, A. Rambaut, A.J. Drummond, BEAST 2: a software platform for Bayesian evolutionary analysis, PLoS Comput. Biol. 10 (4) (2014) e1003537.
[6] D.F. Robinson, Comparison of labeled trees with valency three, J. Combin. Theory Ser. B 11 (2) (1971) 105–119.
[7] E. Mossel, E. Vigoda, Phylogenetic MCMC algorithms are misleading on mixtures of trees, Science 309 (5744) (2005) 2207–2209.
[8] E. Mossel, E. Vigoda, Limitations of Markov chain Monte Carlo algorithms for Bayesian inference of phylogeny, Ann. Appl. Probab. 16 (4) (2006) 2215–2234.
[9] F. Ronquist, B. Larget, J.P. Huelsenbeck, J.B. Kadane, D. Simon, P. van der Mark, Comment on "Phylogenetic MCMC algorithms are misleading on mixtures of trees", Science 312 (5772) (2006) 367a.
[10] D. Štefankovič, E. Vigoda, Fast convergence of Markov chain Monte Carlo algorithms for phylogenetic reconstruction with homogeneous data on closely related species, SIAM J. Discrete Math. 25 (3) (2011) 1194–1211.
[11] D.A. Spade, R. Herbei, L.S. Kubatko, A note on the relaxation time of two Markov chains on rooted phylogenetic tree spaces, Statist. Probab. Lett. 84 (2014) 247–252.
[12] D.J. Aldous, Mixing time for a Markov chain on cladograms, Combin. Probab. Comput. 9 (03) (2000) 191–204.
[13] P. Diaconis, S. Holmes, Random walks on trees and matchings, Electron. J. Probab. 7 (6) (2002) 1–17.
[14] S.N. Evans, A. Winter, Subtree prune and regraft: a reversible real tree-valued Markov process, Ann. Probab. 34 (3) (2006) 918–961.
[15] S. Athreya, W. Löhr, A. Winter, Invariance principle for variable speed random walks on trees, arXiv:1404.6290, http://arxiv.org/abs/1404.6290.
[16] R.G. Beiko, J.M. Keith, T.J. Harlow, M.A. Ragan, Searching for convergence in phylogenetic Markov chain Monte Carlo, Syst. Biol. 55 (4) (2006) 553–565.
[17] C. Whidden, F.A. Matsen, Quantifying MCMC exploration of phylogenetic tree space, Syst. Biol. 64 (3) (2015) 472–491.
[18] Y.S. Song, On the combinatorics of rooted binary phylogenetic trees, Ann. Comb. 7 (3) (2003) 365–379.
[19] Y. Ding, S. Grünewald, P.J. Humphries, On agreement forests, J. Combin. Theory Ser. A 118 (7) (2011) 2059–2065.
[20] R. Atkins, C. McDiarmid, Extremal distances for subtree transfer operations in binary trees, arXiv:1509.00669, http://arxiv.org/abs/1509.00669.
[21] M. Bordewich, C. Semple, On the computational complexity of the rooted subtree prune and regraft distance, Ann. Comb. 8 (4) (2005) 409–423.
[22] G. Hickey, F. Dehne, A. Rau-Chaplin, C. Blouin, SPR distance computation for unrooted trees, Evol. Bioinform. 4 (2008) 17–27.
[23] M.L. Bonet, K. St John, On the complexity of uSPR distance, IEEE/ACM Trans. Comput. Biol. Bioinform. 7 (3) (2010) 572–576.
[24] C. Whidden, R.G. Beiko, N. Zeh, Fixed-parameter algorithms for maximum agreement forests, SIAM J. Comput. 42 (4) (2013) 1431–1466.
[25] C. Whidden, F.A. Matsen IV, Calculating the unrooted subtree prune-and-regraft distance, arXiv:1511.07529, http://arxiv.org/abs/1511.07529.
[26] Y. Ollivier, Ricci curvature of Markov chains on metric spaces, J. Funct. Anal. 256 (3) (2009) 810–864.
[27] A. Joulin, Y. Ollivier, Curvature, concentration and error estimates for Markov chain Monte Carlo, Ann. Probab. 38 (6) (2010) 2418–2442.
[28] Y. Rubner, C. Tomasi, L.J. Guibas, The earth mover's distance as a metric for image retrieval, Int. J. Comput. Vis. 40 (2) (2000) 99–121.
[29] C.-C. Ni, Y.-Y. Lin, J. Gao, D. Gu, E. Saucan, Ricci curvature of the Internet topology, in: Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2015, IEEE Computer Society, 2015.
[30] R. Sandhu, T. Georgiou, E. Reznik, L. Zhu, I. Kolesov, Y. Senbabaoglu, A. Tannenbaum, Graph curvature for differentiating cancer networks, Sci. Rep. 5 (12323).
[31] Y. Ollivier, A survey of Ricci curvature for metric spaces and Markov chains, Probab. Approach Geom. 57 (2010) 343–381.
[32] C. Villani, Topics in Optimal Transportation, Grad. Stud. Math., American Mathematical Society, Providence, 2003.
[33] Y. Lin, L. Lu, S.-T. Yau, Ricci curvature of graphs, Tohoku Math. J. 63 (4) (2011) 605–627.
[34] B. Loisel, P. Romon, Ricci curvature on polyhedral surfaces via optimal transportation, Axioms 3 (1) (2014) 119–139.
[35] C. Whidden, N. Zeh, A unifying view on approximation and FPT of agreement forests, in: Proceedings of the 9th International Workshop, WABI 2009, in: Lect. Notes in Bioinform., vol. 5724, Springer-Verlag, 2009, pp. 390–401.
[36] C. Whidden, R.G. Beiko, N. Zeh, Fast FPT algorithms for computing rooted agreement forests: theory and experiments, in: P. Festa (Ed.), Experimental Algorithms, in: Lecture Notes in Comput. Sci., vol. 6049, Springer, Berlin, Heidelberg, 2010, pp. 141–153.
[37] J. Felsenstein, J. Archie, W. Day, W. Maddison, C. Meacham, F. Rohlf, D. Swofford, The Newick Tree Format, 1990.
[38] C. Whidden, F.A. Matsen IV, Efficiently inferring pairwise subtree prune-and-regraft adjacencies between phylogenetic trees, arXiv:1606.08893.
[39] L.J. Guibas, R. Sedgewick, A dichromatic framework for balanced trees, in: Proceedings of the 19th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1978, pp. 8–21.
[40] E. Fredkin, Trie memory, Commun. ACM 3 (9) (1960) 490–499.
[41] C. Whidden, spr_neighbors, https://github.com/cwhidden/spr_neighbors, http://dx.doi.org/10.5281/zenodo.16543, 2015.
[42] C. Whidden, random_spr_walk, https://github.com/cwhidden/random_spr_walk, http://dx.doi.org/10.5281/zenodo.16541, 2015.
[43] W. Stein, D. Joyner, SAGE: system for algebra and geometry experimentation, ACM SIGSAM Bull. 39 (2) (2005) 61–64, http://sagemath.org/.
[44] GNU linear programming kit, http://www.gnu.org/software/glpk/glpk.html.
[45] F.A. Matsen IV, gricci, https://github.com/matsengrp/gricci, http://dx.doi.org/10.5281/zenodo.16428, 2015.
[46] B. Venkatachalam, J. Apple, K. St John, D. Gusfield, Untangling tanglegrams: comparing trees by their drawings, IEEE/ACM Trans. Comput. Biol. Bioinform. 7 (4) (2010) 588–597, http://dx.doi.org/10.1109/TCBB.2010.57.
[47] F. Matsen, S. Billey, A. Kas, M. Konvalinka, Tanglegrams: a reduction tool for mathematical phylogenetics, IEEE/ACM Trans. Comput. Biol. Bioinform. PP (99) (2016) 1-1, http://dx.doi.org/10.1109/TCBB.2016.2613040.
[48] S.C. Billey, M. Konvalinka, F.A. Matsen IV, On the enumeration of tanglegrams and tangled chains, J. Combin. Theory Ser. A 146 (2017) 239–263, http://dx.doi.org/10.1016/j.jcta.2016.10.003, http://www.sciencedirect.com/science/article/pii/S0097316516301029.
[49] The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.7.7, http://www.gap-system.org, 2015.

[50] F.A. Matsen IV, tangle, https://github.com/matsengrp/tangle, http://dx.doi.org/10.5281/zenodo.16427, 2015.
[51] L. Lovász, Random walks on graphs: a survey, in: Paul Erdös is Eighty, Combinatorics 2 (1) (1993) 1–46.
[52] C. Whidden, sprspace, https://github.com/cwhidden/sprspace, http://dx.doi.org/10.5281/zenodo.16542, 2015.
[53] J. Jost, S. Liu, Ollivier's Ricci curvature, local clustering and curvature-dimension inequalities on graphs, Discrete Comput. Geom. 51 (2) (2013) 300–322.
[54] Y.S. Song, Properties of subtree-prune-and-regraft operations on totally-ordered phylogenetic trees, Ann. Comb. 10 (1) (2006) 147–163.
[55] A.J. Drummond, M.A. Suchard, D. Xie, A. Rambaut, Bayesian phylogenetics with BEAUti and the BEAST 1.7, Mol. Biol. Evol. 29 (8) (2012) 1969–1973.
[56] D. Gusfield, Partition-distance: a problem and class of perfect graphs arising in clustering, Inform. Process. Lett. 82 (3) (2002) 159–164.