On Contract-and-Refine Transformations Between Phylogenetic Trees

Ganeshkumar Ganapathy* Vijaya Ramachandran*† Tandy Warnow*

Abstract

The inference of evolutionary trees using approaches which attempt to solve the maximum parsimony (MP) and maximum likelihood (ML) optimization problems is a standard part of much of biological data analysis. However, both problems are hard to solve: MP provably NP-hard, and ML even harder in practice. Consequently, hill-climbing heuristics are used to analyze datasets for phylogeny reconstruction. Two primary topological transformations have been used in the most popular heuristics: TBR (tree-bisection-and-reconnection) and ECR (edge-contractions-and-refinements). While most of the popular heuristics exclusively use TBR moves to explore tree space, some recent methods have used ECR in conjunction with TBR and found significant improvements in the speed and accuracy with which they can analyze datasets. In this paper we analyze ECR moves in detail, and provide results on the diameter of the tree space, the neigborhood intersection with TBR, structural analysis of the ECR operation, and an efficient method for sampling uniformly from the 2-ECR neighborhood of a tree. Our results should lead to a better understanding of the impact of ECR moves on the performance of heuristic searches.

1 Introduction

Most, if not all, of the favored approaches in biology for inferring phylogenetic (i.e., evolutionary) trees are based upon attempts to solve certain NP-hard optimization problems; of these, perhaps Maximum Parsimony [7] is the most popular. Maximum Likelihood [6] is also favored, but considerably harder in practice to solve than Maximum Parsimony (though not established to be NP-hard). Approximation algorithms for Maximum Parsimony exist, but the approximation ratios are not good enough for use in molecular systematics where errors as small as 1% are unacceptable. Consequently, heuristics, largely based upon hill-climbing (also called local-search), are used to search for optimal trees.

Two topological transformations on trees, TBR (for Tree-Bisection-and-Reconnection) and p-ECR, short for p-Edge Contract and Refine [8], are the basis for the most popular heuristics in use for phylogenetic analysis under Maximum Parsimony. Of these two, the TBR transformation has been traditionally more popular, and is better understood in terms of the properties of the "landscape" of trees it induces [23, 18, 14, 16, 1].

In a p-ECR move p of the edges in the given tree are contracted and the resulting tree is refined to give back a new tree. Sankoff et.al [22] define a version of the ECR move where the contracted edges are restricted to form a subtree (henceforth, we will call this move the p-sECR or *p-subtreeECR* move). In [22] an experimental comparison of local searches based on p-sECR moves for different values of p is presented, and evaluated with regard to the quality of local optima generated. Subsequently the p-ECR move has appeared implicitly rather then explicitly in the local-search heuristic *sectorial-search* [9]. In sectorial-search, a tree is transformed through contractions of edges subsequent refinements, but the edges to be contracted are chosen using some specific heuristic, and so the number of edges contracted can vary during the search. The p-ECR move as used in this paper was defined recently in [8], where the neighborhoods of trees induced by the 2-ECR move and by the TBR move were compared and were shown to have a small intersection.

In this paper, we present several results about the properties of the general p-ECR operation and the search space induced by it. In particular, we present

- asymptotically tight bounds for the diameter of tree-space under p-ECR moves as a function of p, showing that the diameter of the search space is in $\Theta(\frac{n\log n}{p\log p})$ (where n is the number of leaves in the trees). This result could be potentially useful in selecting a suitable range of values of p for performing local searches based on p-ECR operations.
- a comparison of the neighborhoods of a tree induced by TBR and p-ECR moves, showing that their intersection is of size $O(min\{n2^p, n^2p\})$. The neighborhoods themselves are much larger: there could be $\Theta(n^3)$ trees in the TBR neighborhood of a tree, while the p-ECR neighborhood contains $\Omega(n^p2^p)$ trees. These results may help explain why the combination of the two moves improves upon the use of just one, as reported in [9]. This work generalizes the result in [8] for 2-ECR.

*Address: Dept. of Computer Sciences, The Univ. of Texas at Austin, Austin, TX 78712. Email: {gsgk,vlr,tandy}@cs.utexas.edu

†Supported in part by NSF grant CCR-9988160

- an $O(n)$ pre-processing-time, $O(1)$ update-time algorithm for sampling a tree uniformly at random from the set of 2-ECR neighbors of a phylogenetic tree. This potentially has applications in Markov Chain Monte Carlo methods for inferring evolutionary histories through Bayesian analysis [15, 12, 13].
- a structural analysis of the p-ECR operation, motivated by its application in our algorithm for uniformly sampling from the 2-ECR neighborhood of a tree. We define the properties of *irreducibility* and *commutativity* of p-ECR operations, and observe a surprising connection between irreducible p-ECR operations and *elementary* bipartite graphs. We exploit this connection to develop an $O(n+p^2)$ algorithm to reduce a p-ECR operation into an equivalent sequence of irreducible ECR operations.

The rest of the paper is organized as follows: In Section 2 we introduce some basic concepts necessary for the remaining sections. In Section 3 we present upper and lower bounds on the diameter of the search space induced by the p-ECR operation. In Section 4 we compare the neighborhoods of a tree induced by the p-ECR and TBR operations. In Section 5 we present our algorithm for sampling uniformly from the set of 2-ECR neighbors of a tree, and in Section 6 we carry out structural analyses of the p-ECR operation vis-a-vis the properties of irreducibility and commutativity.

2 Basics

A *phylogeny* is an unrooted tree (rooted, if the evolutionary origin is known) whose leaves are labeled and represent extant species, and all of whose internal nodes have degree at least three. A *binary phylogeny* is one where all internal nodes are of degree three. Edges that are *not* incident on leaves are called *internal edges*. Non-binary phylogenies are referred to as being *unresolved* at the nodes of degree greater than three. Any isomorphism between phylogenies must preserve the leaf labels.

2.1 Bipartitions

A notion crucial to the study of phylogenies is that of a *bipartition*: removing an edge e from a leaf-labeled tree T induces a bipartition π_e on its set of leaves. We denote by $C(T)$ the set $\{\pi_e : e \in E(T)\}$, which represents the set of bipartitions induced by T. The set $C(T)$ is known as the *character encoding* of the tree T. Buneman proved [2] that two phylogenies are isomorphic if and only if they have the same character encoding.

2.2 Tree Transformations

Contractions and Refinements A contraction collapses an edge in the tree and identifies its two end points, while a refinement expands an unresolved node into two nodes connected by an edge (see Figure 1). The *p-ECR* tree rearrangement operation on a binary phylogeny is defined to be p edge-contractions, which are then followed by refinements that give back a binary phylogeny. The trees $T1$ and $T5$ in Figure 1 are separated by one 2-ECR operation.

The Robinson-Foulds Metric The Robinson-Foulds distance between two unrooted leaf-labeled (not necessarily binary) trees T and T', denoted $RF(T,T')$ is defined to be the length of a shortest sequences of contractions and refinements that transforms T to T' [21]. It was also shown in [21] that $RF(T,T') = |C(T)-C(T')| + |C(T')-C(T)|$

Based on the above definitions we can deduce the following simple fact.

OBSERVATION 1. *Let T and T' be two unrooted binary leaf-labeled trees on n leaves, and let p be any integer between 1 and $n-3$. Then $RF(T,T') \leq 2p$ if and only if T and T' are one p-ECR move apart.*

Tree Bisection and Reconnection (TBR) In a TBR move, an edge is removed from T, creating subtrees t and $T-t$, and then a new edge is added between the midpoints of any two edges in t and $T-t$, creating a new tree. Throughout the operation any internal node of degree two is suppressed. The TBR operation is illustrated in Figure 2.

Nearest Neighbor Interchange (NNI) The NNI move swaps one rooted subtree on one side of an internal edge e with another on the other side; note that this is equivalent to contracting the edge e, and then resolving the resultant tree into a new binary tree. The NNI operation is thus the same as a 1-ECR operation, and is also a special case of the TBR operation. Every sequence of p NNI moves on a tree is a p-ECR move on that tree; however there are p-ECR moves that cannot be performed by a sequence of p NNI moves (see, e.g., Figure 1).

Neighborhoods, distances, and diameters We define the neighborhood of an unrooted binary leaf-labeled tree T under a tree-rearrangement move to be the set of all trees that can be obtained from T by one move. For each of the different tree rearrangement operations (TBR, NNI, and p-ECR), we define the edit distance between two trees on the same set of leaves as the minimum number of moves needed to move from one tree to the other. All these distances are known to be finite (follows from [20]), but tend to be hard to compute [1, 11, 4]. We denote the edit distance under the p-ECR move by $\delta_{p-ECR}(T,T')$, and the others are similarly defined. Given a specific move (such as TBR, p-ECR, etc.), we can define the *diameter* of tree space to be the maximum edit distance between any two trees. For convenience, we will phrase the diameter of the search space as the diameter of a graph, in which the trees on a given set of leaves constitute the vertices, and an edge exists between two trees if they are related to each other by one move. Thus, the graph defined by the p-ECR move is $G_{p-ECR} = (U,E)$, where U is the set of unrooted leaf-labeled binary trees on

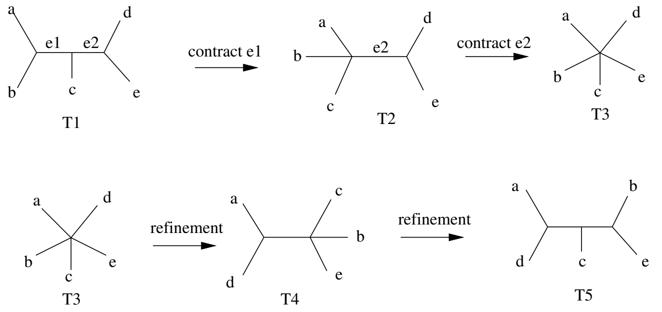

Figure 1: Two edges are contracted in T1 to produce T3, which is then refined to produce T5; T3 and T5 are thus separated by one 2-ECR operation.

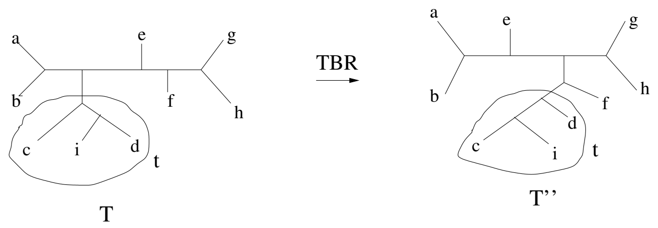

Figure 2: Tree T'' is one TBR move away from T.

n leaves, and $(u,v) \in E$ if and only if u and v are separated by one p-ECR move. We denote the diameter of G_{p-ECR} by $\Delta(G_{p-ECR})$.

3 Bounds on the diameter of G_{p-ECR}

In this section we derive asymptotically tight bounds for the diameter of the tree-space induced by the p-ECR operation (as a function of p). It was shown in [16] that $\Delta(G_{NNI}) \in \Theta(n \log n)$, and it was shown in [1] that $\Delta(G_{TBR}) \in \Theta(n)$. As mentioned earlier, the NNI operation is just the 1-ECR operation. Hence the diameter of the 1-ECR operation is in $\Theta(n \log n)$. The diameter of the $(n-3)$-ECR operation is, of course, 1. Obtaining the diameter as a function of p might give us a way to pick the right range of values of p to use in a search, based on the diameter.

3.1 Upper Bound

THEOREM 3.1. $\Delta(G_{(2p-2)-ECR}) \in O(\frac{n \log n}{p \log p}) + \frac{2n-10}{p-1}$.

Proof. We show that $\Delta(G_{(2p-2)-ECR}) \leq \frac{n \log n}{p \log p} + \frac{2n-10}{2p-2}$, for p a power of two greater than one, and $n > 5$. It can then be shown that $\Delta(G_{(2p-2)-ECR}) \leq \frac{2n \log_2 n}{p(\log_2 p - 1)} + \frac{2n-10}{p-1}$, for all values of $p \geq 2$ and $n > 5$, thus proving out theorem.

Let C be the sorted "caterpillar" tree for the set of leaf labels $[n]$ (Figure 3). We will show that for any unrooted binary leaf-labeled tree T, $\delta_{(2p-2)-ECR}(T,C) \leq \frac{n}{p} \log_p n + \frac{2n-10}{2p-2}$.

We will first show that the number of $(2p-2)$-ECR steps needed to convert a complete binary tree on n leaves to a caterpillar C is at most $\frac{n}{p} \log_p n$. We will then apply an idea from [3] to show how any tree can be converted to a complete binary tree in $(2n-10)$ NNI moves. This implies that the number of $(2p-2)$-ECR steps needed to convert any tree to a complete binary tree is at most $\frac{2n-10}{2p-2}$. The above two results would then imply that any tree can be converted to C in at most $\frac{n}{p} \log_p n + \frac{2n-10}{2p-2}$ steps. A complete binary tree [1] and the sorted caterpillar tree for the set of leaves labeled from 1 through 7 are given in Figure 3.

Converting an arbitrary complete binary tree to a sorted caterpillar tree The procedure is recursive, and is illustrated in Figures **??** and **??**. Let B be the complete binary tree on n leaves. Let B_1, B_2, B_3, $\ldots B_p$ be the subtrees of B at a depth of $\log_2 p$. Since B is a complete binary tree, so are subtrees B_1 through B_p. Recursively convert each of the p subtrees to a sorted caterpillar tree, producing the *bc-tree* (binary-cum-caterpillar tree) B'. The subtrees of B' at a depth $\log_2 p$ are now sorted caterpillar trees C_1 through C_p (see Figure **??**).

The following process is illustrated in Figure **??**: consider the first p leaves in the sorted order. These p leaves can be "pulled" up to the root by contracting only $(2p-2)$ edges (this is because the caterpillar trees C_1 through C_p are sorted). The contraction of these $(2p-2)$ edges make the root of the tree unresolved, with each of the p leaves now a descendant of the root. To complete the $(2p-2)$-ECR

[1] A complete binary tree is shown rooted at the edge that divides the set of leaves most evenly. The tree being rooted makes little difference to our analysis.

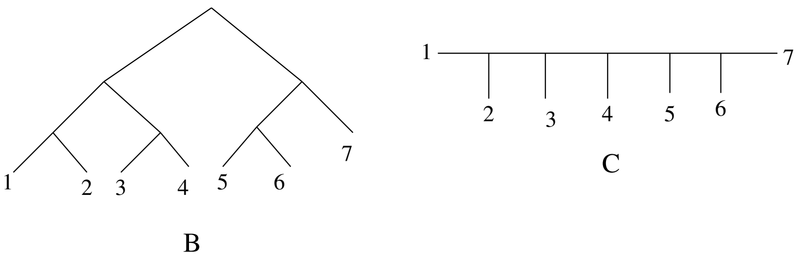

Figure 3: B is a complete rooted binary tree on seven leaves. C is the sorted caterpillar tree for the same set of leaves.

operation, we have to refine the root, and when we refine the root the p leaves can be transferred to "above" (see Figure **??**) the root in sorted order. The next $(2p-2)$-ECR move will transfer the next p leaves in the sorted order to above the root. In this manner we can obtain a C from B' in $\frac{n}{p}$ $(2p-2)$-ECR moves. This gives us the following recursive equation for the number of moves required to convert B to C. Let $S(n)$ denote the number of $(2p-2)$-ECR steps required to convert an n-leaf complete binary tree to the corresponding sorted caterpillar tree. Then,

$$S(n) \leq pS(\frac{n}{p}) + \frac{n}{p}$$

Solving the recurrence yields us $S(n) \leq \frac{n}{p}\log_p n$.

Converting any tree to a complete binary tree in $O(n/p)$ p-sECR moves

In a caterpillar we define the *end leaves* as the two pairs of leaves at each end of the caterpillar; the remaining leaves will be called *internal leaves*. The path connecting the two pairs of end leaves is the *spine* of the caterpillar. We define a q-caterpillar as a caterpillar in which each internal leaf is replaced by a *q-spoke*, which is a caterpillar with $q-2$ internal leaves and one pair of end leaves (see figure). The very last spoke (one that is adjacent to the parent of one of the two end pairs of leaves) is a q'-spoke for $q' = n-4-\lfloor n/q \rfloor$.

Let T be an unrooted binary tree. Any binary tree contains two pairs of leaves where each pair has a comment parent. We fix two such pairs of leaves in T and we call the unique path connecting their parents p_1 and p_2 as its spine. We fix one of the parents, say p_1, and relative to p_1 we define the *potential* $\phi(T) = 2 \cdot n_1 + n_2$, where n_1 is the number of succ essive q-spokes in T starting with the vertex adjacent to p_1 in T, and n_2 is the number of successive subtrees of size at least q rooted at vertices following the initial n_1 q-spokes.

Consider the transformation of an arbitrary binary tree T into a q-caterpillar using p-sECR moves, where $q = \lfloor p/2 \rfloor$. The initial potential of T is non-negative and final potential of the transformed tree is $2 \cdot \lfloor n/q \rfloor$. We now describe a method that transforms T into a q-caterpillar using p-ECR moves, which increases the potential of the transformed tree in each step by at least one. Thus this method transforms T into a q-caterpilla r in $O(n/p)$ moves.

Our method will apply a p-sECR move by contracting edges starting with the edges in the subtree rooted at the vertex v on the spine that is adjacent to the last vertex that is a root of one of the n_1 q-spokes already constructed (if $n_1 = 0$ then this is the first vertex on the spine). The resulting contracted subtree S on p internal edges will have $p+3$ external edges incident on it, of which two are on the spine and $p+1$ are within subtrees rooted at one or more vertices on the spine. If S can be refined to form a new successive q-spoke, then the potential increases by at least one. Otherwise, at least q external edges end in subtrees, each of which contain at least two leaves, which implies that S can be refined into two subtrees, each containing at least q leaves. Further, since S could not be refined into a q-spoke the subtree rooted at v was not of size between q and $2q$, so one of the two subtrees formed is a new one, resulting in an increase in potential of at least one. Hence T can be transformed into a q-caterpillar in $O(n/p)$ p-sECR moves.

By reversing the above strategy the q-caterpillar can be transformed into any binary tree in the same number of moves, hence any binary tree can be transformed into a complete binary tree in $O(n/p)$ p-sECR moves. □

THEOREM 3.2. $\Delta(G_{p-ECR}) \geq \frac{n\log_2 n - o(n\log n)}{8p\log_2 p + O(p)}$, *for all* $p > 1$.

Proof. Let T be an unrooted binary tree on n labeled leaves, and let T' be a tree such that $\delta_{p-ECR}(T,T') = 1$. We first show that, $\delta_{NNI}(T,T') \leq 2p\log_2 p + O(p)$, for p

Without loss of generality, assume that $|C(T) - C(T')| = p$, and let X be the p-ECR operation that transforms T to T'. If the edges corresponding to bipartitions in $C(T) - C(T')$ form a subtree of T, say S, let the corresponding subtree in T' be S'. Then S and S' may be considered to form a tree with $p+3$ leaves each, and S can be transformed into S' using at most $2p\log_2 p + O(p)$ NNI moves, as was shown in [16].

If the edges corresponding to bipartitions in $C(T) - C(T')$ form a forest of k trees, then it can be shown that there exist some k ECR operations X_1 through X_k that act on disjoint set of bipartitions such that X is equivalent to performing the k operations X_1 through X_k one after the other (see Corollary 6.3). Let X_i be a p_i-ECR operation. We have $\sum_{i=1}^{k} p_i = p$. Let T^i be the tree obtained by the application of X_i on T^{i-1} (thus, $T = T^0$ and $T' = T^k$). Then, $\delta_{NNI}(T^{i-1}, T^i) \leq 2p_i \log_2 p_i + O(p_i)$. Hence, $\delta_{NNI}(T, T') \leq 2\sum_{i=1}^{k} p_i \log p_i + O(p)$. The summation on the right-hand side is less than $p \log_2 p$ for all $p > 1$ and $k > 1$, and hence $\delta_{NNI}(T, T') \leq 2p \log_2 p + O(p)$, for $p > 1$.

From the results in [16, 1], $\Delta(G_{NNI}) \geq \frac{n \log_2 n - o(n \log n)}{4}$. This, together with the fact that any p-ECR move can be emulated by at most $2p \log_2 p + O(p)$ NNI moves, gives us the desired result.

□

4 Comparison of p-ECR and TBR neighborhoods

Recall that the neighborhood of a tree T under a tree rearrangement operation is the set of all trees that can be obtained by performing one such operation on T. In this section we first establish bounds on the size of the p-ECR neighborhood of a tree on n leaves, and then show that size of the intersection of the p-ECR neighborhood and the TBR neighborhood of a tree is small. We will denote the neighborhood of a tree T under, say, the TBR operation, as $\Gamma_{TBR}(T)$. It is known that $|\Gamma_{TBR}(T)| = \Theta(n^3)$ [1].

LEMMA 4.1. *Let T be an unrooted binary leaf-labeled tree on n leaves. Then, $\sum_{k=1}^{p} \binom{n-3}{k} 2^k \leq |\Gamma_{p-ECR}(T)| \leq \binom{n}{p}(p+3)!!$, where $(p+3)!!$ is the product of all odd numbers up to $p+3$.*

Proof. For any tree T' in $\Gamma_{p-ECR}(T)$, $RF(T, T') \in \{2, 4, 6, \ldots, 2p\}$. We will show that the number of trees T' in $\Gamma_{p-ECR}(T)$ such that $RF(T, T') = 2k$ is at least $\binom{n-3}{k} 2^k$, and that will give us the result that we desire.

Let k be such that $1 \leq k \leq (n-3)$. For every way of choosing k edges in T, there are at least 2^k different k-ECR moves that can be performed on T: for each chosen edge, contract the edge and refine the resulting unresolved node one of at least two ways that results in the alteration of the bipartition corresponding to the edge. Thus, there are at least $\binom{n-3}{k} 2^k$ trees T' such that $RF(T, T') = 2k$. This completes our proof of the lower bound. For the upper bound, observe that for each of the $\binom{n}{p}$ ways of selecting p edges to contract, there are at most $(p+3)!!$ neighbors ($(p+3)!!$ is the number of unrooted leaf-labeled binary trees on $p+3$ leaves). This completes our proof. □

THEOREM 4.1. *Let T be an unrooted binary leaf-labeled tree on n leaves. Then, for any p, $|\Gamma_{p-ECR}(T) \cap \Gamma_{TBR}(T)| \leq min\{(2n-3)(p+1)2^{p+3}, (2n-3)^2(p+1)\}$).*

Proof. Let $S = \Gamma_{p-ECR}(T) \cap \Gamma_{TBR}(T)$, and let T' be in S. Then, $|C(T) - C(T')| \leq p$, since $T' \in \Gamma_{p-ECR}(T)$. Moreover, since $T' \in \Gamma_{TBR}(T)$, the edges in T corresponding to bipartitions in $C(T) - C(T')$ all must lie on a path, and the bipartitions corresponding to all edges on the path except three (the first edge, the last edge and the edge that is broken for the TBR move) must be in $C(T) - C(T')$. Hence, each such T' can be specified by three edges that lie on a path of length at most $(p+3)$. Now, the number of paths of length at most $(p+3)$ is at most $(2n-3)2^{p+3}$. This is because the number of paths of length exactly $p+3$ is at most $(2n-3)2^{p+2}$: fix one of the terminal edges of the path, there are at most 2^{p+3} paths with a given terminal edge, since the tree is binary. But in this manner, each path will be counted at least twice, and hence there are at most $(2n-3)2^{p+2}$ paths of length exactly $(p+3)$. Summing over all p we get that the number of paths of length at most $(p+3)$ is at most $(2n-3)2^{p+3}$.

Also, each path of length at most $(p+3)$ corresponds to at most $(p+1)$ trees that are in S, since there are $(p+1)$ ways of choosing the edge that is broken for the TBR move. Hence we have that $|S| \leq (2n-3)2^{p+3}(p+1)$.

Moreover, the total number of paths in T is $(2n-3)^2$. For every tree in S, there is a path in T, and each path contributes at most $p+1$ trees to S. Hence $|S| \leq (2n-3)^2(p+1)$.

Thus, we have that $|\Gamma_{p-ECR}(T) \cap \Gamma_{TBR}(T)| \leq min\{(2n-3)(p+1)2^{p+3}, (2n-3)^2(p+1)\}$). □

5 Uniform Sampling from the set of 2-ECR Neighbors

The use of MCMC (Markov Chain Monte Carlo) algorithms in phylogeny reconstruction is of increasing interest in the research and user community [13, 12, 15]. In this section, therefore, we address the problem of selecting a tree uniformly at random from the set of 2-ECR neighbors of a tree. Our algorithm takes $O(1)$ time, after a one-time pre-processing step that costs $O(n)$ time.

We partition the set of 2-ECR neighbors of T into two subsets: $\Gamma_{NNI}(T)$ and $S = \Gamma_{2-ECR}(T) - \Gamma_{NNI}(T)$. The size of the former set is $2n-6$, and the size of the latter set depends on the structure of T. The outline of our algorithm is as follows:

1. Compute, in $O(n)$ time, $s = |S|$.
2. Generate q at random from a uniform distribution on $[0, 1]$.
3. If $q \leq \frac{2n-6}{2n-6+s}$, generate a tree uniformly at random from $\Gamma_{NNI}(T)$.
4. If $q > \frac{2n-6}{2n-6+s}$, generate a tree uniformly at random from S.

Sampling from $\Gamma_{NNI}(T)$: Step (3) is easy and can be performed in $O(1)$ time, given the set of internal edges of T. We choose an internal edge e uniformly at random, and pick each of the two trees that can be obtained by contracting and refining e with probability $1/2$. It can be verified that in this manner we do sample uniformly at random from $\Gamma_{NNI}(T)$.

Sampling from S: This is a complicated by the fact that sampling two edges e_1 and e_2 one after the other without replacement, and then sampling uniformly at random from the set of neighbors obtained by performing a 2-ECR move involving edges e_1 and e_2 does not induce a uniform distribution on S. This is due to the following reason: when e_1 and e_2 are adjacent, there are 14 neighbors, while there are only 8 when e_1 and e_2 are not adjacent.

Hence, we adopt the following strategy: we arbitrarily order the internal edges in T, and let $index(e)$ denote the position of the edge e in such an order.

- We let Y be the set of neighbors that can be obtained from T through a sequence of two 1-ECR moves, the first one involving edge e_1 and the next involving e_2, and such that $index(e_1) < index(e_2)$. Every pair of internal edges (whether adjacent or non-adjacent) contributes four trees to Y. We let $|Y| = y$.
- Let $X = S - Y$, and let $|X| = x$. The set of neighbors X contains the following two classes of trees:
  - Trees that cannot be obtained by a sequence of two 1-ECR moves. There are two such trees for every pair of adjacent internal edges.
  - Trees that are obtained by two 1-ECR moves involving two *adjacent* internal edges, e_1 first and e_2 next, such that $index(e_1) > index(e_2)$. Every pair of adjacent internal edges contributes four such trees to X. Note that two 1-ECR operations performed in the reverse order on two non-adjacent edges do not generate any new trees, since the order does not matter when the edges are non-adjacent.

Note that $|\Gamma_{2-ECR}(T)| = 2n - 6 + x + y$. We are now in a position to describe our algorithm.

Algorithm to sample uniformly from S

1. Calculate x and y. This can be done in $O(n)$ time since x depends only on the number of pairs of adjacent internal edges in T, and y depends only on n.

2. Generate q at random from a uniform distribution on $[0,1]$.

3. if $q \leq \frac{x}{x+y}$, then sample a pair of adjacent internal edges e_1 and e_2, and then sample a tree uniformly at random from the set of neighbors contributed to X by a 2-ECR move involving e_1 and e_2 (this involves sampling one tree from a set of six trees).

4. if $q > \frac{x}{x+y}$, then sample two internal edges e_1 and e_2 one after the other without replacement from the set of internal edges. Then sample a tree uniformly at random from the set of neighbors contributed to Y by a 2-ECR move involving e_1 and e_2 (this involves sampling one tree from a set of four trees).

Every 2-ECR neighbor is generated with a probability of $\frac{1}{2n-6+x+y}$ by our algorithm. The running time is $O(n)$, the time taken to to calculate the number of pairs of adjacent internal edges in T. However, note that once a new a tree is generated, this number can be calculated for the new tree in $O(1)$ time, since a 2-ECR move makes only local changes to the tree structure. Hence, we have the following:

THEOREM 5.1. *We can generate a tree uniformly at random from the set of 2-ECR neighbors of an unrooted leaf-labeled binary tree on n leaves in $O(1)$ time, after an $O(n)$ preprocessing step.*

At first sight, our algorithm seems to be a series of case analyses. However, the analyses reveals some interesting properties of structure of 2-ECR moves: there are some 2-ECR operations are not *reducible* to two successive NNI moves, and among those that *are* thus reducible, some involve successive NNI moves that are *commutable* (i.e, those that can be reordered), and the rest involve successive moves that are not commutable. We believe that these concepts (and generalizations of them) will be essential in designing an algorithm that samples efficiently from the set of p-ECR neighbors of a tree for $p > 2$. In the next section we study reducibility and commutability of p-ECR moves, and show that these concepts generalize to general p through a surprising connection to elementary bipartite graphs.

6 Structural Analyses of the p-ECR Operation

In this section we will show how to construct, for any two given trees, a sequence of *elementary* or *irreducible* ECR operations that transforms one tree to another, where an ECR operation is a p-ECR operation for some (unspecified) p.

We first introduce some terminology and notation. Let T be an unrooted leaf-labeled tree. Let X and Y be two ECR operations on T. We will say X equals Y if performing X on T results in the same tree as the one obtained by performing Y on T. For two *ECR* sequences X and Y, we will let $Y \circ X$ be the following sequence of two *ECR* operations: X on T, followed by Y on the tree that results from performing X on T.

DEFINITION 1. *Reducible p-ECR operation*

Let T be an unrooted leaf-labeled tree. Let X be a p-ECR operation on T. X is said to be reducible if there exists a p_1-ECR operation X_1 and a p_2-ECR operation X_2 such that $X = X_2 \circ X_1$ and $p = p_1 + p_2$.

The concepts of reducibility and irreducibility of ECR operations are illustrated in Figure 4.

The problem that we address in this section is this: given two binary trees T and T' such that $RF(T,T') = 2p$, decompose the p-ECR operation X that separates T and T' such that,

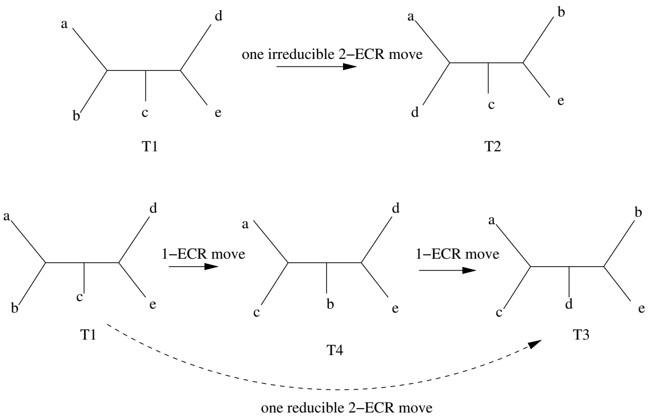

Figure 4: Trees T1 and T2 are one irreducible 2-ECR move away, and trees T1 and T3 are one reducible 2-ECR move away.

- $X = X_k \circ X_{k-1} \circ \ldots \circ X_1$, with X_i being an irreducible p_i-ECR operation, and
- $\sum_{i=1}^{k} p_i = p$

6.1 Irreducibility and Elementary Bipartite Graphs

We begin with a definition:

DEFINITION 2. *Bipartition (or edge) compatibility:*

A set of bipartitions B is said to be compatible *if and only if $B \subseteq C(T)$ for some tree T.*

LEMMA 6.1. (FROM BUNEMAN [2]) *A set of bipartitions is compatible iff any two bipartitions in the set are are pairwise compatible. Furthermore, two bipartitions $A = A_1 : A_2$ and $B = B_1 : B_2$ are compatible iff at least one of the four sets $A_1 \cap B_1$, $A_1 \cap B_2$, $A_2 \cap B_1$ and $A_2 \cap B_2$ is empty.*

Observe that there can not be more than $2n-3$ edges in a phylogenetic tree with n leaves, since there are no internal nodes of degree two (through out the rest of the paper we use n to denote the number of leaves). This gives us the following:

COROLLARY 6.1. *The maximum cardinality of any set of compatible bipartitions of a set of n elements is $2n-3$.*

We now define a graph, which we call the *incompatibility* graph, defined by two leaf-labeled trees.

DEFINITION 3. *Incompatibility Graph*

Let T and T' be two unrooted leaf-labeled trees. The incompatibility graph G between T and T', G, is defined thus: G is a bipartite graph, and $G = (U,V,E)$ where $U = C(T) - C(T')$, $V = C(T') - C(T)$ [2]*, and $(u,v) \in E$ if and only if u and v are incompatible.*

[2] Note that the definition here is almost the same as the definition of the incompatibility graph appearing in [19], where U and V were $C(T)$ and $C(T')$ respectively. Our definition has the effect of removing isolated vertices from the incompatibility graph.

An elementary bipartite graph is one where every edge is in some perfect matching [17]. Suppose T and T' are two trees such that $\delta_{p-ECR}(T,T') = 1$, then, we show that the p-ECR move that separates them is irreducible if and only if the incompatibility graph induced by the two trees is elementary. We start with the following lemma.

LEMMA 6.2. *Let G be the incompatibility graph between two unrooted binary leaf-labeled trees. Then G has a perfect matching.*

Proof. We will show that the incompatibility graph G satisfies the following two properties: (1) For every subset S of V, $|\Gamma(S)| \geq |S|$, and (2) For every subset R of U, $|\Gamma(R)| \geq |R|$. Our result will then follow from Hall's matching theorem [10]. Let $q = |C(T) - C(T')| = |C(T') - C(T)|$. We first show that (1) holds. Let S be a subset of V. Note then that the set of bipartitions $A = (U - \Gamma(S)) \cup S \cup (C(T) \cap C(T'))$ is compatible. Now if $|\Gamma(S)| < |S|$, then the set A contains more than $2n-3$ bipartitions that are pairwise compatible, since $|(U - \Gamma(S)) \cup S| > q$ and $|C(T) \cap C(T')| = 2n-3-q$. But this is a contradiction by Corollary 6.1. Similarly, we can show that (2) holds. □

THEOREM 6.1. *Let X be a p-ECR move that can be carried out on an unrooted leaf-labeled binary tree T. Let T' be the result of carrying out X on T. Let $G = (U,V,E)$ be the incompatibility graph between T and T'. Then, X is reducible if and only if there is a proper subset S of V such that $|\Gamma(S)| = |S|$.*

Proof. Suppose that X is reducible and is equivalent to $X_2 \circ X_1$. Then the set of bipartitions S that results from the refinement phase of X_1 satisfies the condition that $|\Gamma(S)| = |S|$. If not there is at least one bipartition in S that is incompatible with a bipartition in $U - \Gamma(S)$. But this makes carrying out X_1 on T impossible, and hence $|S| = \Gamma(S)$.

Conversely, if there is a set $S \subset V$ that satisfies $|S| = |\Gamma(S)|$, then the contraction of the bipartitions in $\Gamma(S)$ and the creation of bipartitions in S can be "scheduled" before the other contractions and refinements in X, and that makes X reducible. $\square$

COROLLARY 6.2. *A p-ECR move is irreducible if and only if the corresponding incompatibility graph is elementary.*

Using the above characterization, we now show that we can check efficiently if a p-ECR move is irreducible. It also means that we can compute, for any given p-ECR move, an equivalent sequence of irreducible ECR operations.

THEOREM 6.2. *Let X be a p-ECR move that can be performed on an unrooted binary leaf-labeled tree on n leaves. Then, in $O(n+p^2)$ time, we can determine if X is reducible, and we can compute a sequence of ECR moves X_1 through X_k (for some k), with each X_i being an irreducible p_i-ECR move for $1 \leq i \leq k$, such that $X = X_k \circ \ldots \circ X_1$ and $\sum_{i=1}^{k} p_i = p$.*

Proof. Let $G = (U,V,E)$ be the incompatibility graph corresponding to the p-ECR move X. The graph G can be constructed in $O(n+p^2)$ time as follows: The sets U and V can be computed in $O(n)$ time, while calculating the RF distance between T and T' [5]. Once U and V have been determined, E can be calculated in $O(n+p^2)$ time, since for each bipartition in U, we can identify all bipartitions in V incompatible with it in $O(p)$ time.

Once we have G, we use the method in [17] (Section 4.3) to decompose G into maximal vertex-disjoint components such that the subgraph of G induced by each component is elementary, as follows: we compute a perfect matching M in G (which is guaranteed to exist by Lemma 6.2) and then compute an associated directed graph, say H. The graph H is computed from G by first orienting all the edges uniformly towards either U or V, and then identifying the vertices matched by M.

The directed graph H is strongly connected if and only if G is elementary. This is due to the following reason: each edge of G not in M is in some perfect matching if and only if the corresponding directed edge in H is in a directed cycle. Hence, G is elementary if and only if every edge not in M is also in a perfect matching, be definition. Every edge in H is in a cycle if and only if H is strongly connected. This proves our claim.

If G is not elementary, then H can be decomposed into strongly connected components, say C_1 through C_k, with component C_i representing an elementary subgraph of G induced by the sets of vertices (S_i,T_i), with $S_i \subset U$ and $T_i \subset V$. Without loss of generality, let $C_1, C_2, \ldots, C_k$ be the topologically sorted order of the strongly connected components. Then, this represents an ordering of the corresponding elementary subgraphs of G, as follows: if $(u,v) \in E$ such that $u \in S_i$ and $v \in T_j$, then $i \leq j$ (assuming without loss of generality that all edges were oriented towards V while creating H from G). Hence, if we let the induced subgraph (S_i,T_i) stand for the i^{th} ECR operation X_i, it is assured that X_i can be performed once operations X_1 through X_{i-1} have been performed. This is because the incompatibilities of the bipartitions created by X_i (i.e, those in T_i) are with bipartitions in components S_j with $j < i$, and these bipartitions would have been eliminated by the ECR moves from X_1 through X_{i-1}. The outcome of the sequence of operations X_1 through X_k is X. $\square$

6.2 Commutable p-ECR moves

DEFINITION 4. *A p-ECR operation X is separable if and only if there are two ECR moves X_1 and X_2 such that $X = X_2 \circ X_1 = X_1 \circ X_2$. The ECR moves X_1 and X_2 are then said to be commutable.*

Suppose for an ECR move X, the corresponding incompatibility graph is not connected. That X is reducible is immediate. However, in the following lemma (proof omitted) we observe that X is in fact separable.

LEMMA 6.3. *Let X be an ECR move executable on an unrooted leaf-labeled binary tree. The incompatibility graph induced by X is not connected if and only if X is separable.*

We now present a necessary and sufficient condition for separability of X that can be verified without computing the incompatibility graph.

COROLLARY 6.3. *Let X be a p-ECR move that can be carried out on an unrooted leaf-labeled tree T. Let T' be the result of carrying out X on T. Then X is separable if and only if the edges corresponding to the bipartitions in $C(T) - C(T')$ do not form a connected subtree.*

Proof. Let $U = C(T) - C(T')$ and $V = C(T') - C(T)$. Let subsets S_1 and S_2 partition U, such that the edges in S_1 and S_2 form vertex-disjoint components in T. Let T_1 and T_2 form the corresponding partition of V. Now let $G = (U,V,E)$ be the incompatibility graph between T and T'. In G, it can be seen that there can not be any edges between either S_1 and T_2 or S_2 and T_1.

To prove the other direction, we prove the following: let u_1, u_2 be two bipartitions in U, corresponding to two edges e_1 and e_2 adjacent in T. Let v_1 and v_2 be two bipartitions in V such that,

- $(u_1, v_1) \in E$,
- $(u_2, v_2) \in E$,

- $(u_1, v_2) \notin E$, and
- $(u_2, v_1) \notin E$.

We show that u_1, u_2, v_1 and v_2 can be reached from one another in G. This would imply that if the edges in U form a single subtree in T, then G is connected.

We now prove the above claim. Let u_1 be the bipartition $P : P'$ and let u_2 be $P \cup Y : P' - Y$, for some $Y \subset P'$ (this entails no loss of generality since u_1 and u_2 are compatible). Similarly, let v_1 and v_2 be $Q : Q'$ and $Q \cup Z : Q' - Z$ respectively, for some $Z \subset Q'$. Since u_1 and v_2 are incompatible, we have $P \cap (Q' - Z) = \emptyset$ (it can be verified that the other three pairwise intersection cannot be empty) from Lemma 6.1. Similarly, we have $Q \cap (P' - Y) = \emptyset$.

Now, since the tree T is binary, and the edges e_1 and e_2 are adjacent, we have that $Y : Y'$ (where Y' is the complement of Y) is a bipartition in T, and the corresponding edge is adjacent to both e_1 and e_2. We show that $Y : Y'$ is incompatible with both v_1 and v_2, thus showing that u_1, u_2, v_1, v_2 are reachable from each other.

Now, $Q \cap (P' - Y) = \emptyset$ and $Q \cap P' \neq \emptyset$ implies that $Q \cap P' \subseteq Y$. Now, we show that $Y \not\subseteq Q \cap P'$. Suppose, to the contrary, that $Y \subseteq Q \cap P'$. Then, $Y \subseteq Q$. This, combined with the fact that $(Q' - Z) \subseteq P$, means that $(Q' - Z) \subseteq (P' - Y)$. However, this contradicts $(Q' - Z) \cap (P \cup Y) \neq \emptyset$, and hence $Y \not\subseteq Q \cap P'$. Thus, we have $(Q \cap P') \subset Y$. We now show that $Y : Y'$ is incompatible with both v_1 and v_2, thus completing our proof.

1. $Y \cap Q \neq \emptyset$, since as we already saw, $Q \cap P' \subset Y$. Also, $Y \not\subseteq Q$. Hence, $Y \cap Q' \neq \emptyset$. Moreover, $Q \not\subseteq Y$ (since $Q \not\subseteq P'$), and hence $Q \cap Y' \neq \emptyset$. Similarly, $Q' \cap Y' \neq \emptyset$. Thus, we have that $Y : Y'$ is incompatible with v_1.

2. $Y \cap (Q \cup Z) \neq \emptyset$, since $Y \cap Q \neq \emptyset$. Now, since $(Q' - Z) \cap P = \emptyset$ and $(Q' - Z) \cap (P \cup Y) \neq \emptyset$, we have $Y \cap (Q' - Z) \neq \emptyset$. This means that $Y' \cap (Q \cup Z) \neq \emptyset$ and $Y' \cap (Q' - Z) \neq \emptyset$ as well. Thus, we have that $Y : Y'$ is incompatible with v_2.

   This completes our proof. □

Corollary 6.3 can be generalized as follows:

COROLLARY 6.4. *Let T be an unrooted leaf-labeled tree. Let X be a p-ECR operation on T that would result in a tree T'. Suppose the edges corresponding to bipartitions in $C(T) - C(T')$ constitute a forest with k trees. There exist k ECR operations X_1, X_2, $\ldots X_k$ such that $X = X_k \circ X_{k-1} \circ \ldots \circ X_1$.*

Proof. By repeated application of Corollary 6.3. □

6.3 Existence of Irreducible p-ECR moves In this section we establish that one can construct an irreducible p-ECR move for every set of p connected edges in any tree. Our construction relies on the concept of an *edge-minimal elementary bipartite graph*, i.e., an elementary bipartite graph in which the removal of any single edge makes the graph non-elementary.

THEOREM 6.3. *Let T be any unrooted binary tree with at least p internal edges. Then there is a tree T' such that $RF(T, T') = 2p$, and the incompatibility graph between T and T' is elementary.*

Proof.

The proof is by explicit construction. Given any T, we show how to construct such a T'.

We will call a pair of leaves that are siblings as forming a *cherry*. Let T have k cherries, $(x_1, y_1), \ldots, (x_k, y_k)$. We create T' thus: T' is identical to T as far as tree topology (neglecting leaf-labels) is concerned. Hence, T' also contains k cherries. In T' we let (x_1, y_k), (x_2, y_1), $(x_3, y_2), \ldots, (x_k, y_{k-1})$ be the cherries. For an illustration of this construction, see Figure 5.

We claim that T and T' are separated by one irreducible p-ECR move. We will prove this by showing that the incompatibility graph of T and T' is elementary.

We omit detailed proof and just present an outline here. Let $G = (U, V, E)$ be the incompatibility graph where $U = C(T) - C(T')$ and $V = C(T) - C(T')$. Then,

1. It can be shown that $|U| = |V| = p$.

2. Observe that our transformation just permutes the leaves. For each bipartition π in U there is a unique bipartition π' in V that is obtained by just permuting the leaves in π according to our transformation. Then, π is incompatible with π'.

3. There is a permutation of the vertices in U, π_1, $\pi_2, \ldots, \pi_p$, such that
   π_1, π'_1, π_2, π'_2, $\ldots, \pi_p$, π'_p, π_1 is a (Hamiltonian) cycle in G. See Figure 5 for an example on how to obtain this permutation.

The presence of a Hamiltonian cycle in G implies that G is elementary, since the cycle by itself is a minimally elementary bipartite graph on U and V. This completes our proof. □

Acknowledgement. We thank the referee for pointing us to reference [22].

References

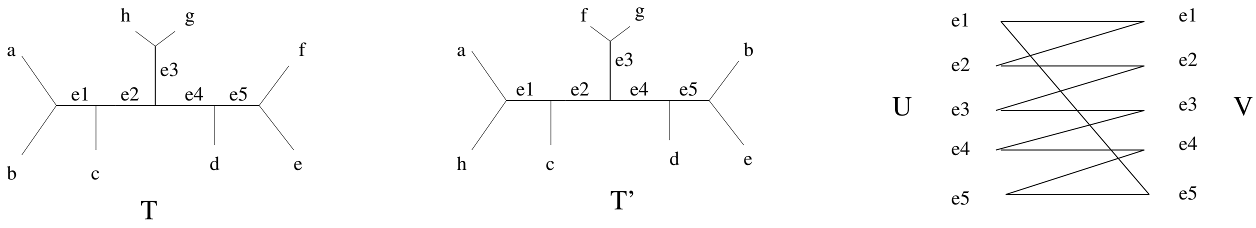

Figure 5: Trees T and T' are separated by one irreducible 5-ECR move. A minimally elementary subgraph (a Hamiltonian cycle) of the incompatibility graph of T and T' is shown alongside. Other edges in the graph have been omitted for the sake of clarity.

[1] B. Allen and M. Steel. Subtree Transfer Operations and Their Induced metrics on Evolutionary Trees. *Annals of Combinatorics*, 5:1–15, 2001.

[2] P. Buneman. The Recovery of Trees from Measures of Dissimilarity. *Mathematics in the Archaelogical and Historical Sciences*, pages 387–395, 1971.

[3] Karol Culik-II and Derick Wood. A note on Some Tree Similarity Measures. *Information Processing Letters*, 15:39–42, 1982.

[4] B. Dasgupta, X. He, T. Jiang, M. Li, J. Tromp, and L. Zhang. On the Distances Between Phylogenetic Trees. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 427–436. ACM-SIAM, 1997.

[5] W. H. E. Day. Optimal Algorithms for Comparing Trees with Labeled Leaves. *Journal of Classification*, 2:7–28, 1985.

[6] J. Felsenstein. Evolutionary Trees for DNA Sequences: A Maximum Likelihood Approach. *Journal of Molecular Evolution*, 17:368–376, 1981.

[7] W. Fitch. Toward Defining Course of Evolution: Minimum Change for a Specified Tree Topology. *Systematic Zoology*, 20:406–416, 1971.

[8] G. Ganapathy, V. Ramachandran, and T. Warnow. Better Hill-Climbing Seaches for Parsimony. In *Proceedings of the Third International Workshop on Algorithms in Bioinformatics (WABI)*, 2003. To appear.

[9] P. A. Goloboff. Analyzing Large Datasets in Reasonable Times: Solutions for Composite Optima. *Cladistics*, 15:415–428, 1999.

[10] P. Hall. On Representatives of Subsets. *Journal of the London Mathematical Society*, 10:26–30, 1935.

[11] J. Hein, T. Jiang, L. Wang, and K. Zhang. On the Complexity of Comparing Evolutionary trees. *Discrete Applied Mathematics*, 71:153–169, 1996.

[12] J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian Inference of Phylogenetic Trees. *Bioinformatics*, 17(8):754–755, 2001.

[13] J. P. Huelsenbeck, F. Ronquist, R. Nielsen, and J. P. Bollback. Bayesian Inference of Phylogeny and its Impact on Evolutionary Biology. *Science*, 294:2310–2314, 2001.

[14] B. Kirkup and J. Kim. From Rolling Hills to Jagged Mountains: Scaling of Heuristic Searches For Phylogenetic Estimation. *Molecular Biology and Evolution*, 2000. In Revision.

[15] B. Larget and D.L.Simon. Markov Chain Monte Carlo Algorithms for the Bayesian Analysis of Phylogenetic Trees. *Molecular Biology and Evolution*, 16:277–283, 1999.

[16] M. Li, J. Tromp, and L. Zhang. On the Nearest Neighbour Interchange Distance Between Evolutionary Trees. *Journal of Theoretical Biology*, 182:463–467, 1996.

[17] L. Lovasz and M. D. Plummer. *Matching Theory*. North-Holland Publishing Company, 1986.

[18] D. R. Maddison. The Discovery and Importance of Multiple Islands of Most Parsimonious Trees. *Systematic Zoology*, 43(3):315–328, 1991.

[19] C. A. Phillips and T. Warnow. The Asymmetric Median Tree: A New Model for Building Consensus Trees. *Discrete Applied Mathematics*, 71:311–335, 1996.

[20] D. F. Robinson. Comparison of Labeled Trees with Valency Three. *Journal of Combinatorial Theory*, 11:105–119, 1971.

[21] D. F. Robinson and L. R. Foulds. Comparison of Phylogenetic Trees. *Mathematical Bio-sciences*, 53:131–147, 1981.

[22] D. Sankoff, Y. Abel, and J. Hein. A Tree, A Window, A Hill, Generalization of Nearest Neighbor Interchange in Phylogenetic Optimization. *Journal of Classification*, 11:209–232, 1994.

[23] D. Swofford, G. J. Olson, P. J. Waddell, and D. M. Hillis. *Molecular Systematics*, chapter Phylogenetic Inference, pages 407–425. Sinauer Associates, Sunderland, Massachusetts, second edition, 1996.