Subtree Transfer Operations and their Induced Metrics on Evolutionary Trees

---

A thesis
submitted in partial fulfilment
of the requirements for the degree
of
Master of Science in Mathematics
in the
University of Canterbury
by
Benjamin L. Allen

University of Canterbury
1998

In Memory of Madge Copus Allen

Contents

Abstract

Leaf-labelled trees are widely used to describe evolutionary relationships, particularly in biology. In this setting, extant species label the leaves of the tree, while the internal vertices correspond to ancestral species. Various techniques exist for reconstructing these evolutionary trees from data, and an important problem is to determine how "far apart" two such reconstructed trees are from each other, or indeed from the *true* historical tree. To investigate this question requires tree metrics, and these can be induced by operations that rearrange trees locally. Here we investigate three such operations, *nearest neighbour interchanges* (or NNI), *subtree prune and regrafts* (SPR), and *tree bisection and reconnections* (TBR). The SPR operation is of particular interest as it can be used to model biological processes such as *horizontal gene transfer* and *recombination*. We count the number of unrooted binary trees one SPR from any given unrooted binary tree, as well as providing new upper and lower bounds for the *diameter* of the adjacency graph of trees under SPR and TBR. We also show that the problem of computing the minimum number of TBR operations required to transform one tree to another can be *kernalized* to a problem whose size is a function just of the distance between the trees (and not of the size of the two trees), and thereby establish that the problem is *fixed-parameter tractable*. We conjecture that the SPR equivalent of this problem is also fixed-parameter tractable.

Chapter 1

Introduction

Leaf-labelled trees are widely used to represent evolutionary relationships, particularly in biology, but also in other areas of classification (including linguistics and philology). Typically a set S of extant (present day) species label the leaves and the remaining vertices represent ancestral species.

Tree Metrics

Given data (such as aligned DNA sequences), numerous methods exist for reconstructing a tree (see [21]) that hopefully approximates the true historical tree of descent of the species under study. However, different data sets and different methods often lead to different trees being reconstructed for the same set of species. Thus it becomes imperative to determine how "close" two reconstructed trees are. This requires the introduction of metrics on trees. Several such metrics have been considered (see [15]). A particularly natural choice is to say that two trees are "close together" if one can be obtained from the other by a small number of "local" operations. Typically three types of local rearrangements have been studied and we will consider these in detail in the next chapter. However, little is known about how pairs of trees are distributed according to these metrics, or even how to efficiently calculate them. In this thesis we investigate both questions. In particular, in Chapter 2 we:

- define three tree metrics, the *nearest neighbour interchange* (or NNI), the *subtree prune and regraft* (SPR), and finally the *tree bisection and reconnection* (TBR) (Sections 2.1 – 2.3);
- establish new results on the diameter and density of the adjacency graph of unrooted trees under the subtree prune and regraft and tree bisection and reconnection operations, thereby correcting an oversight in [18], (Sections 2.4 – 2.6);
- establish a relationship between the number of tree bisection and reconnection operations required to transform on tree into another and the size of the maximum agreement forest for the two trees, thereby correcting an error in [10] (Sections 2.7 – 2.8).

In Chapter 3 we turn our attention to computing the distances between evolutionary trees with respect to each tree metric. We:

- investigate the complexity in the conventional sense of the NNI, SPR and TBR Distance Problems, and point out that the TBR-Distance Problem is *NP*-hard, while the complexity of the remaining two is unresolved (Sections 3.1 – 3.3);
- show that, for the tree bisection and reconnection operation, the question of whether a given unrooted binary tree can be transformed to another given unrooted tree by at most k operations is *fixed-parameter tractable* (Sections 3.4);
- conjecture that the Parameterized SPR-Distance Problem is *FPT* as well (Subsection 3.4.4).

Horizontal Gene Transfer and recombination

In the past morphological data has been used to construct trees, such as the presence or absence of wings, number of eyes, and so on. Nowadays trees are constructed using genomic information. Instead of looking for wings, skeletal structure, etc. trees are based on genes that species have in common, the locality and order of genes upon chromosomes, or the sequences that make up genes. However there are circumstances when fitting genomic data to a tree is not appropriate or when genomic data can be fitted to several different trees, with no one tree being "better" than any other. Two causes can be the presence of a horizontal gene transfer or recombination.

In the conventional sense of evolution genes from the father and the mother are merged together in their offspring. This type of evolution, known as *vertical* evolution, can be described using phylogenetic trees. Recombination or horizontal gene transfer causes genes from a different site either within the genome, in the case of a recombination, or from a different species in the case of a horizontal gene transfer, to become part of a organism's genome. This type of genetic event can not be described using conventional phylogenetic methods. However a subtree transfer operation, and in particular, the subtree prune and regraft operation, can be used to model the effect of these types of events on phylogenies. The use of the SPR to model these two events is described further in Section 2.9.

Tree Search Heuristics

A third motivation for studying subtree transfer operations is that they are frequently used in optimization heuristics. In searching for the true evolutionary tree for a set of n species, often a characteristic such as the parsimony score will be taken as a parameter to optimize. Searching all trees on n leaves is a hopelessly intractable task for realistic values of n, and so optimization heuristics are routinely used. One common heuristic, used by software packages such as PAUP and PHYLIP, is to start with an initial approximation to the true tree, that has been constructed in polynomial time. Subtree transfer operations are then applied as part of a hill-climbing search strategy.

1.1 Definitions

Before we continue it will be important to set down some definitions. Much of the language of phylogenetic trees is based on ideas from Graph Theory. For the remainder of this chapter we will review some important results on graphs and, more importantly, on trees.

1.1.1 Graph Theoretic Definitions

Definition 1.1.1 A *graph* $G = (V, E)$ is made up of a set of vertices V and a set of edges $E \subseteq \{\{u, v\} | u, v \in V\}$ that connect the vertices. If the edge $e = \{u, v\} \in E$ for $u, v \in V$ then u and v are said to be *adjacent*, while e is said to be *incident* to u and v. Figure 1.1 provides an example of a graph.

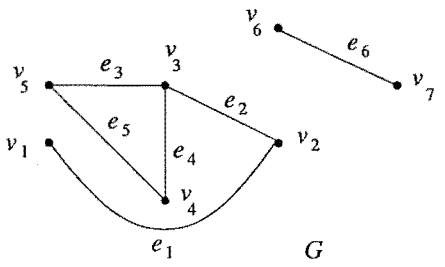


Figure 1.1: Graph G is a graph on seven vertices $\{v_1, \ldots, v_7\}$ and six edges $\{e_1, \ldots, e_6\}$. To illustrate Definition 1.1.1, v_6 and v_7 are adjacent while e_1 is incident to v_1 and v_2. Note also that G has two "disconnected" subgraphs.

Definition 1.1.2 The *degree* of a vertex, $u \in V$, in a graph is the number of edges incident with u.

Definition 1.1.3 A graph is said to be *regular* if all vertices have the same degree.

Definition 1.1.4 For a graph $G = (V, E)$, an edge $e = \{u, v\}$ where $u, v \in V$ is said to be *subdivided*, when a new vertex w is added to V, the edge e is deleted from E, and two new edges $\{u, w\}$ and $\{v, w\}$ are added to E. Note that a new leaf, pendant to w may also be added.

Two important concepts for classification of graphs are that of a *path* and *connectedness*. These two properties allow us to define what we mean by a *tree*.

Definition 1.1.5 A *walk* is a traversal of a subset of vertices along edges connecting the vertices such that any edge may only be traversed once. A *path* is a walk such that each vertex is visited only once.

Definition 1.1.6 A graph is said to be *acyclic* if all walks are paths.

Definition 1.1.7 A graph $G = (V, E)$ is said to be *connected* if, for all $u, v \in V$, there is a path from u to v.

Definition 1.1.8 An acyclic, connected graph is called a *tree.*

1.1.2 Properties of Trees

An important result for trees is that we are able to relate the sum of the degrees of all vertices to the number of vertices in the tree.

Lemma 1.1 *For a tree $T = (V, E)$,*

1. $|E| = |V| - 1$; *and*
2. $\Sigma_{v \in V} \deg(v) = 2|E| = 2|V| - 2$.

Proof (1) is established by induction on the number of vertices. Clearly a tree with one vertex has no edge. Assume (1) is true for a tree with k vertices, and let T be a tree with $k + 1$ vertices. Let T' be a tree obtained by removing a leaf and its pendant edge from T, then T' has $k - 1$ edges and hence T has k edges. Thus the hypothesis is true for all k.

For the first equality in(2), since all edges are incident with two vertices, the sum of the degrees of all vertices is twice the number of edges. This result is true for any graph. The second equality follows immediately from (1). □

Definition 1.1.9 The vertices of degree one in a tree are called the *leaves* or *terminal vertices*, while vertices of degree greater than one are called *internal vertices.* A tree may also have one vertex labelled as the *root.* An edge incident to a leaf is called a *pendant edge*, while edges incident to internal vertices only are *internal edges.* For our purposes only the leaves and the root (if present) will be labelled, such trees are known as *(rooted) leaf-labelled trees.*

Definition 1.1.10 The *topology* or *shape* of a tree is the tree without any labels on the leaves.

The majority of evolutionary relationships can be described by the *binary* trees.

Definition 1.1.11 A tree is said to be *binary* if all internal vertices, with the exception of the root if present, have degree three. If the tree is rooted, then the root must have degree two.

Primarily we will be interested in unrooted binary trees as these are normally the end product of biological data analysis. If data is presented in the form of a rooted tree then we can transform it to an unrooted tree by adding in a leaf corresponding to a species known to have diverged much early

than those being studied. For instance we may unroot a rooted binary tree of monkeys, apes and other advanced primates with an early primate such as a tree shrew, in this case the tree shrew is called an *outgroup.*

Definition 1.1.12 Let $UB(n)$ represent the space of all labelled unrooted binary trees on n leaves, and $|UB(n)|$ be the number of members in this space.

Schröder [20] originally showed, the now well-know result that;

$$|UB(n)| = (2n-5)!! = \frac{(2n-4)!}{2^{n-2}(n-2)!}.$$

Definition 1.1.13 The *distance*, with respect to a specific tree operation Θ, between two trees, T_1, T_2 in $UB(n)$, written $d_\Theta(T_1, T_2)$, is the minimum number of Θ operations required to transform T_1 to T_2.

We wish to investigate the distances between trees for various subtree transfer operations that will be defined in Chapter 2. However, before we can do so we will first define certain parts of a tree.

Definition 1.1.14 A *subtree* is any connected subgraph of a tree.

Definition 1.1.15 A *pendant subtree* is a subtree that can be disconnected from the rest of the tree by removing exactly one edge. Conversely an *internal subtree* is a subtree that is not pendant.

Definition 1.1.16 Let $\mathcal{L}(T)$ denote the *leaf set* of tree T, that is, the set of labels of the leaves of T. Let $|\mathcal{L}(T)|$ be the number of leaves of T.

Occasionally we may wish to delete a leaf (or leaves) or a pendant subtree(s) from a binary tree, along with corresponding incident edges. This will generally introduce vertices of degree two in the resulting trees. Following Hein ([8], [9] and [10]) we introduce the notion of a *forced contraction.*

Definition 1.1.17 When we apply a *forced contraction*, we delete a vertex v of degree two and replace the two edges incident to v with a single edge. This process is continued until all degree two vertices have been eliminated, thereby producing a binary tree. Suppose we have a set $U \subseteq \mathcal{L}(T)$ for some binary tree T, then we let $T(U)$ denote the minimal subtree of T connecting leaves from U, and let $T_{|U}$ denote the tree obtained from $T(U)$ by applying forced contractions.

A common entity used in induction proofs on binary trees is a *cherry.*

Definition 1.1.18 A *cherry* is a subtree t of a tree T consisting of a single internal vertex adjacent to two pendant edges, as well as both pendant edges and leaves.

Lemma 1.2 *For all $T \in UB(n)$, where $n \geq 4$, T has at least two cherries.*

Proof Suppose that $T \in UB(n)$, where $n \geq 4$, and that T contains less than two cherries. Then, all, but at most one, internal vertices are adjacent to at most one leaf. This implies either the presence of a cycle, or internal edges of degree two, both contradicting the definition of a binary tree. □

Lemma 1.3 *Let T be in $UB(n)$, where $n \geq 3$. Then the number of pendant edges in T is n and the number of internal edges is $n - 3$.*

Proof We use induction on the number of leaves. There is only one tree topology for an unrooted binary tree on three leaves. Since there are three leaves, only three pendant edges exist, and furthermore, they all must be incident to a single internal vertex. Therefore no internal edges exist, and so the hypothesis holds. Now suppose that the hypothesis is valid for all trees in $UB(k)$ and let T be in $UB(k+1)$. By Lemma 1.2 T must contain at least two cherries. Distinguish one cherry in T and let T' be the tree obtained from T by removing the two pendant edges and leaves of the distinguished cherry. Hence $T' \in UB(k)$ and hence has k pendant edges and $k-3$ internal edges. Hence T has $k+1$ pendant edges and $k-2$ internal edges. Thus the hypothesis is valid for all $n \geq 3$. See Figure 1.2 for illustration. □

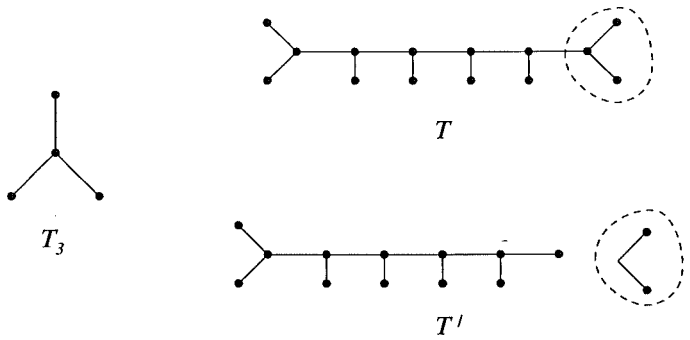


Figure 1.2: T_3 is the only tree topology for an unrooted binary tree on 3 leaves. T is an unrooted binary tree with a distinguished cherry indicated by the dashed region. T' is the tree that results when the cherry is removed as in the proof of Lemma 1.3. This is an example of the induction step in the proof of Lemma 1.3.

Chapter 2

Subtree Transfer Operations

2.1 Nearest Neighbour Interchange

Of branch swapping techniques, the most restrictive is the *nearest neighbour interchange* or *NNI.*

Definition 2.1.1 Any internal edge of a unrooted binary tree has four subtrees attached to it. A *nearest neighbour interchange* occurs when one subtree on one side of an internal edge is swapped with a subtree on the other side of the edge, as illustrated in Figure 2.1.

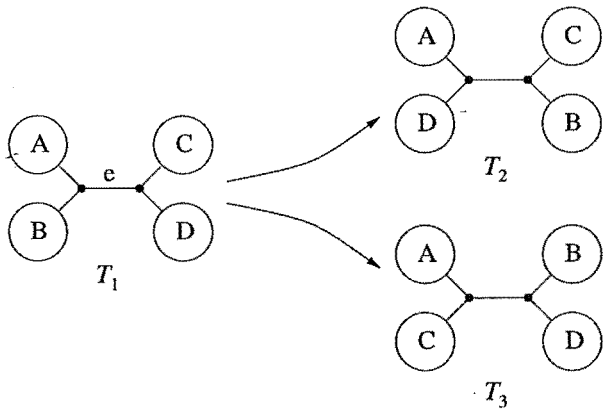


Figure 2.1: Examples of NNI transformations. The two trees T_2 and T_3 result from the two possible NNI's about edge e in T_1

NNI's were independently introduced by Robinson [19] in 1971 and Moore *et al.* [16] in 1973. Problems involving NNI's have received considerable attention since then (Page [18]), however some of the claimed results have contained serious flaws. [13] summarises many of the flaws.

Lemma 2.1 *For any internal edge e in an unrooted binary tree, there are precisely two new distinct trees possible after all NNI's have been carried out about e.*

Proof A NNI swaps two of the subtrees incident to the edge e. Suppose that the four subtrees are labelled A, B, C and D, where A and B form a cherry, as do C and D. Then there are four possibilities.

1. A is swapped with C,
2. A is swapped with D,
3. B is swapped with C, or
4. B is swapped with D.

However these only result in two distinguishable trees, as swapping A with C makes A and D a cherry and B and C a cherry, as does swapping B and D. Similarly swapping A and D is equivalent to swapping B and C. □

From our definition of distance, the NNI-distance, denoted $d_{NNI}(T_1, T_2)$, between two trees $T_1, T_2 \in UB(n)$ is the minimum number of NNI needed to transform T_1 to T_2.

Lemma 2.2 *The number of trees at a distance of one NNI from any given tree $T \in UB(n)$ is $2n - 6$.*

Proof By Lemma 2.1 there are two new distinct trees possible for all NNI's about an internal edge. It thus remains to show that no two NNI's about different edges on $T \in UB(n)$ can result in the same tree. Suppose that T' results from an NNI about edge e or from an NNI about edge e'. Since an NNI swaps subtrees at either end of an edge, it follows that $e = e'$. □

Lemma 2.2 was first given in [19], but without formal proof.

2.2 Subtree Prune and Regraft

Definition 2.2.1 A *subtree prune and regraft* or SPR on an unrooted binary tree T is defined as cutting any edge and thereby pruning a subtree, t, and then regrafting the subtree by the same cut edge to a new vertex obtained by subdividing a pre-existing edge in $T - t$. We also apply a forced contraction to maintain the binary property of the resulting tree.

The SPR operation can also be defined for rooted binary trees, but with the added restriction of not allowing the root to occur in the subtree.

Definition 2.2.2 A *subtree prune and regraft* on a rooted binary tree T is defined as cutting any edge, thereby pruning a subtree t, such that the root of T can not be in t. The subtree is then regrafted by the same cut edge to a new vertex obtained by subdividing a pre-existing edge in $T - t$. We also apply a forced contraction to maintain the rooted binary property of the resulting tree.

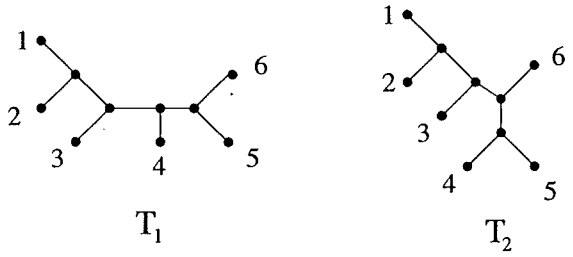


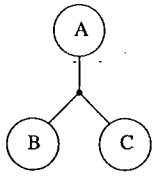
Figure 2.2: Example of a SPR. The subtree with leaf set $\{1, 2, 3\}$ in tree T has been pruned and re-attached to the pendant edge of leaf 6 to give tree T'.

In the Section headed Comparing NNI, SPR and TBR (pp 204 – 208) of [18] the author claims that, the number of trees one SPR from a given tree is dependent on topology. This is false as the next new theorem shows.

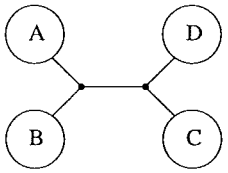
Theorem 2.1 *The number of trees in $UB(n)$ at a distance of one SPR from a given $T \in UB(n)$ is* $2(n-3)(2n-7)$.

Proof When a subtree is pruned and regrafted we cut an edge and then re-attach it to a different edge. The number of edges we can choose to cut is $2n - 3$ and the number we can re-attach to is $2n - 4$. Hence the total number of possible subtree prune and regrafts is $(2n - 3)(2n - 4)$. However not all of these subtree prune and regrafts produce distinct trees, or even different trees to T. We can eliminate over-counts by separating subtree prune and regrafts into three disjoint cases.

(i) The edge to which the subtree will be regrafted is adjacent to the cut edge. This results in no change to the tree's topology. Furthermore we have six such subtree prune and regrafts associated with each internal vertex, hence a total of $6(n - 2)$.

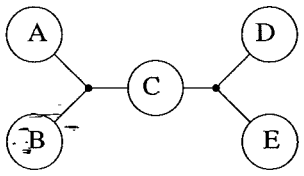


(ii) The edge to which the subtree will be regrafted is separated by exactly one edge from the edge to be cut. –

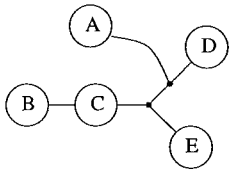


In this case we have 8 possibilities for each internal edge, giving a total of $8(n-3)$. However, only two distinct trees are possible, hence $2(n-3)$ distinct trees result. *c.f.* an NNI.

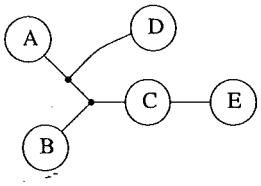
(iii) Finally, we consider the case where the subtree regrafted is separated by more than one edge from the edge which was originally cut.



Subtrees A and D can become adjacent to the same vertex in two ways only. Firstly if A is pruned and regrafted to the D's incident edge.



Or secondly, if D is pruned and regrafted to A's incident edge.



Since both resulting trees are different, we conclude that any such prune and regraft will create a tree that can not be obtained by any other single subtree prune and regraft. The number of such subtree prune and regrafts is the remainder of those not considered in Cases (i) or (ii), that is;

$$(2n-3)(2n-4) - 6(n-2) - 8(n-3) = 4(n-3)(n-4).$$

Hence the total number of trees at a distance of one subtree prune and regraft is;

$$4(n-3)(n-4) + 2(n-3) = 2(n-3)(2n-7). \tag{2.1}$$

□

Theorem 2.1 demonstrates that the number of trees at an SPR-distance of one from any given tree, T say, is independent of the topology of the tree T. Consider the case of rooted binary trees. The placement of the root does affect the number of trees within one SPR of any given tree T, hence the topology of the tree will influence the number of trees at one SPR in distance from a given rooted binary tree. See Figure 2.3 for an example.

2.3 Tree Bisection and Reconnection

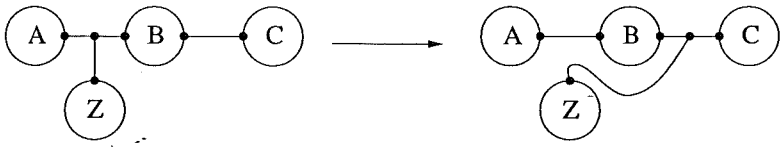
Definition 2.3.1 A *tree bisection and reconnection* (TBR) on an unrooted binary tree T is defined as removing any edge, giving two new subtrees, t_1 and t_2, which are then reconnected by creating a new edge between the midpoint of any edge in t_1 and any edge in t_2. Again forced contractions are applied to ensure the resulting tree is binary. In the case that one of the subtrees is a single leaf, then the edge connecting t_1 and t_2 is incident to the leaf. See Figure 2.4 for an example of a TBR.

When we considered SPR's in Section 2.2 we found that the number of trees in $UB(n)$ one SPR from a tree T was independent of the topology of T. However, the number of unrooted trees one TBR from T does depend on the topology of T. See Figure 2.5 for an example.

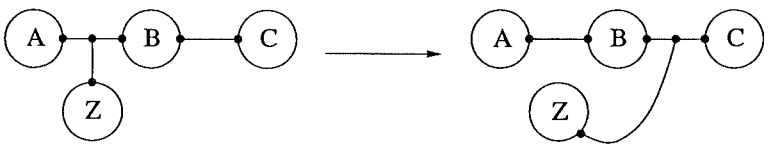
2.4 Tree Distance Results

2.4.1 The Relationship between SPR and TBR

We can draw generalised diagrams of the unrooted SPR and TBR operations. An SPR has the form;



While a TBR has the form;



From the diagrams above it is clear that an SPR is a special case of a TBR. However TBR represents only a limited generalisation over SPR as the following result shows.

Lemma 2.3 *Any $T' \in UB(n)$ obtained from $T \in UB(n)$ by a single TBR operation can also be obtained by at most two SPR operations.*

Proof Consider the TBR of following general form above. We can also obtain the same tree after two SPR's. Firstly the Z component subtree is pruned and regrafted to the correct edge. Then the Z

component is joined to the correct vertex. This is achieved by treating the rest of the tree as a subtree to be pruned and regrafted to the correct vertex in the Z component. See Figure 2.6.

Thus we obtain exactly the same binary tree as that obtained from the tree bisection and reconnection. □

By Definition 1.1.13, $d_{SPR}(T_1, T_2)$ is the minimum number of SPR's required to transform T_1 to T_2. Similarly $d_{TBR}(T_1, T_2)$ is the minimum number of TBR's required. Since an SPR is just a special case of a TBR it follows by Lemma 2.3 that we have the following inequality:

$$d_{TBR}(T_1, T_2) \leq d_{SPR}(T_1, T_2) \leq 2d_{TBR}(T_1, T_2) \tag{2.2}$$

Theorem 2.2 *The number of trees in $UB(n)$ one TBR in distance from $T \in UB(n)$ is bounded above by $(2n-3)(n-3)^2$.*

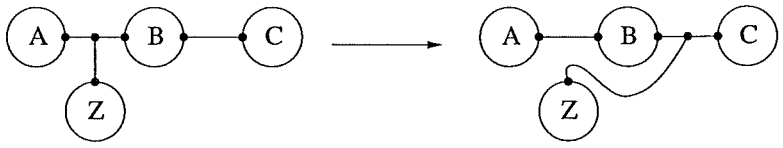
Proof By Definition 2.3.1 there is an injection from the set of TBR's on T to the set of ordered pairs $(e, \{a, b\})$ where e is an edge of T, and where, if $\{A, B\}$ is the bi-partition of the leaf set induced about e, a is edge from subtree $T_{|A}$, or $a = T_{|A}$ if $T_{|A}$ is a single vertex, and b is an edge from subtree $T_{|B}$, or $b = T_{|B}$ if $T_{|B}$ is a single vertex. By Lemma 1.3 there are $2n-3$ choices for e, $|2|A| - 3|$ choices for a and $|2|B| - 3|$ choices for b. Thus, there are at most $(2n-3)|(2|A|-3)(2|B|-3)|$ trees, and furthermore $|A| + |B| = n$. For $x + y = n$, $(2x-3)(2y-3)$ attains its constrained maximum at $x = y = n/2$. Hence the number of trees one TBR from T is at most $(2n-3)(n-3)^2$. □

2.4.2 Induced Subtree Distances

Theorem 2.3 *Suppose we have $T, T' \in UB(n)$. Let $S \subseteq \mathcal{L}(T)$. Then $d_\Theta(T_{|S}, T'_{|S}) \leq d_\Theta(T, T')$ for $\Theta \in \{NNI, SPR, TBR\}$.*

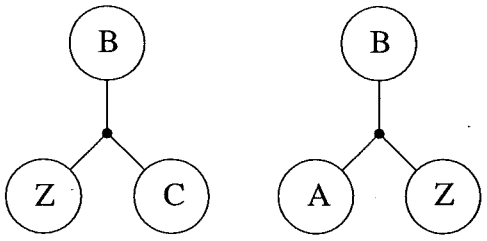
Proof First note that a Θ operation on T induces a Θ operation on $T_{|S}$ (provided we also allow the identity operation which leaves T unchanged to count as a Θ operation).

Next we establish the result in the case $d_\Theta(T, T') = 1$. We will suppose first that $\Theta = SPR$. Assume that two trees one SPR apart have the following form;



(i) If either $S \cap \mathcal{L}(B) = \emptyset$ or $S \cap \mathcal{L}(Z) = \emptyset$ then $d_{SPR}(T_{1|S}, T_{2|S}) = 0$.

(ii) If $S \cap \mathcal{L}(C) = \emptyset$, or $S \cap \mathcal{L}(A) = \emptyset$ and B is a pendant subtree, then there is there is no change in the two trees, since there is one central vertex from which Z is pruned and then reconnected to. Hence $d_{SPR}(T_{1|S}, T_{2|S}) = 0$. T_1 below illustrates this case with T_1 for when $S \cap \mathcal{L}(A) = \emptyset$ and B is pendant, and T_2 for when $S \cap \mathcal{L}(C) = \emptyset$.



T_1 T_2

(iii) Finally if none of the above cases are true then there must be at least one internal vertex that distinguishes the placement of the $Z_{|S}$ subtree. Hence $d_{SPR}(T_{1|S}, T_{2|S}) = 1$, as $Z_{|S}$ can be moved in one SPR.

The NNI and TBR cases are similar. Now, if $d_\Theta(T, T') = k > 1$, there are trees T^0, T^1, ..., T^k such that $T^0 = T, T^k = T'$ and $d_\Theta(T^l, T^{l+1}) = 1$ for all $l \in \{0, 1, \ldots, k-1\}$. Let $t^l = T^l_{|U}$ for all $l \in \{0, 1, \ldots, k-1\}$. Then from the particular case above, $d_\Theta(t^l, t^{l+1}) \leq 1$ for all $l \in \{0, 1, \ldots, k-1\}$. Thus, the trees $t^1, \ldots, t^k$ define a series of at most k Θ operations that transform T to T', as required. □

2.5 Metric Tree Spaces

A space of trees is sometimes referred to as a *Baumraum* (German for tree space). The *Baumraum* that we will examine is $UB(n)$; recall that this is the space of unrooted binary trees on n leaves and that $|UB(n)| = (2n-5)!!$.

Definition 2.5.1 A metric space (X, d) is a set of points X and a metric d.

In our case, the points are unrooted binary trees on n leaves. The next theorem shows three metrics that can defined on $UB(n)$.

Theorem 2.4 *The NNI, SPR and TBR operations all induce metrics on* $UB(n)$

Proof Robinson [19] first established that the $(UB(n), d_{NNI})$ is a metric space. Now let d be either d_{SPR} or d_{TBR} and suppose that $T_1, T_2, T_3 \in UB(n)$. In order for d to be a metric on $UB(n)$ the following three properties must hold;

1. $d(T_1, T_2) \geq 0$ and $d(T_1, T_2) = 0 \iff T_1 = T_2$.

2. $d(T_1, T_2) = d(T_2, T_1)$

3. $d(T_1, T_2) + d(T_2, T_3) \geq d(T_1, T_3)$

Properties 1 and 2 hold trivially in both cases. Property 3, the triangle inequality, is also easily resolved. Let $d(T_1, T_2) = i$ and $d(T_2, T_3) = j$. Hence there are i trees traversed on the path from T_1 to T_2 and, similarly, j trees traversed on the path from T_2 to T_3. Hence there is a path from T_1 to T_3 that traverses $i + j$ trees and as such $d(T_1, T_3) \leq i + j$. □

2.6 Diameters of Metric Tree Spaces

Definition 2.6.1 The Θ-*adjacency graph* $G_\Theta(n) = (V, E)$ is the graph with $V = UB(n)$ and $\{t_u, t_v\} \in E \iff d_\Theta(t_u, t_v) = 1$ for $d_\Theta \in \{NNI, SPR, TBR\}$.

Definition 2.6.2 The *diameter* of a graph $G = (V, E)$, denoted $\Delta(G)$, is

$$\max_{u,v \in V} \{\min\{k : \text{ where } k \text{ is the number of edges in a path from u to v}\}\}.$$

From the definition of distance, we immediately have;

$$\Delta(G_\Theta(n)) = \max_{T_1, T_2} \{d_\Theta(T_1, T_2)\}, \text{ where } \Theta \in \{NNI, SPR, TBR\}. \tag{2.3}$$

2.6.1 Nearest Neighbour Interchange Diameter

Consider the adjacency graph $G_{NNI}(n)$.

Lemma 2.4 *$G_{NNI}(n)$ is connected and regular with degree $2n - 6$.*

This was established by Robinson [19]. Li *et al.* [13] published a tight asymptotic bound on $\Delta(G_{NNI}(n))$:

$$((n-2)/4)\log_2[2(n-2)\sqrt{(2/3e)}] \leq \Delta(G_{NNI}(n)) \leq n \log_2 n + \mathcal{O}(n). \tag{2.4}$$

2.6.2 Subtree Prune and Regraft Diameter

Consider now the adjacency graph $G_{SPR}(n)$.

Lemma 2.5 *If $d_{NNI}(T_1, T_2) = k$, then $d_{SPR}(T_1, T_2) \leq k$.*

Proof An NNI can be regarded as a SPR in which a pruned subtree is regrafted to an edge one edge away. Hence $d_{NNI}(T_1, T_2) \leq d_{SPR}(T_1, T_2)$. □

As mentioned in the proof of Theorem 2.1 NNI's only make up some of the trees at SPR-distance one, hence the inequality of Lemma 2.5 can be strict. We also have similar results for the SPR case as those found for $G_{NNI}(n)$ in Section 2.6.1.

Lemma 2.6 *For the SPR Baumraum:*

1. *$G_{SPR}(n)$ is connected.*
2. $\deg(v) = 2(n-3)(2n-7)\ \forall\ v \in V$
3. $\Delta(G_{SPR}(n)) \leq n \log_2 n + O(n)$

Proof Since an NNI is a special case of an SPR, (1) and (3) follow immediately from Lemma 2.4 and Equation 2.4. (2) follows from Theorem 2.1. □

Lemma 2.6(3) is the upper bound for the NNI instance found in [13] and so can potentially be improved upon, especially if one considers Lemma 2.5. From a more intuitive perspective, the NNI-distance could be much greater than the SPR-distance, suppose a subtree has to be moved from one end of a long, narrow tree to the other. In the worst case a subtree might have to be moved across all $n-3$ internal edges in the NNI instance, while requiring only one SPR. We improve the upper bound by considering the following new theorem.

Theorem 2.5 *For $T_1, T_2 \in UB(n)$, T_1 can be transformed to T_2 by at most $n-3$ SPR.*

Proof We use induction on the number of leaves. There are three binary trees on four leaves, all of which are at distance one SPR from each other. So the hypothesis holds for $n = 4$. Assume now that the hypothesis is true for any pairs of trees in $UB(k)$, and suppose $T_1, T_2 \in UB(k+1)$. Considering the cherries of T_1 and T_2 there are two cases.

(i) There is a cherry that occurs in both T_1 and T_2. Replace this cherry in both trees by a single leaf to get T_1' and T_2', both on k leaves. Hence T_1' can be transformed to T_2' in at most $k-3$ operations and therefore, so too for T_1 and T_2. Hence the hypothesis is valid for $n = k+1$ in this case.

(ii) If there is no cherry that occurs in both trees, then distinguish a cherry in T_2. Let T_1' be the tree obtained from T_1 after one of the leaves of the distinguished cherry in T_2 has been pruned from T_1 and regrafted so that the distinguished cherry occurs in T_1' as well. Now apply case (i) to get that

T_1' can be converted to T_2 in at most $k-3$ SPR. Hence T_1 can be converted to T_2 in at most $k-2$ ways, hence the hypothesis is valid in this case for $n = k+1$ as well.

Since cases (i) and (ii) cover all problem instances, the hypothesis is valid for all n by induction.

□

It immediately follows from Theorem 2.5 that

$$\Delta(G_{SPR}(n)) \leq n - 3. \tag{2.5}$$

The next new theorem provides an asymptotic lower bound for the diameter of $G_{SPR}(n)$.

Theorem 2.6 $\Delta(G_{SPR}(n)) \geq n/2 - o(n)$.

Proof Recalling from Theorem 2.1 that in $UB(n)$ the number of trees one SPR from a given tree is $2(n-3)(2n-7)$, and that the number of unrooted binary trees is $(2n-5)!!$. Thus if $d = \Delta(G_{SPR}(n))$, then

$$\begin{aligned} [2(n-3)(2n-7)]^d &\geq (2n-5)!! \\ &= \frac{(2n-4)!}{2^{(n-2)}(n-2)!}. \end{aligned} \tag{2.6}$$

By Stirling's factorial approximation,

$$k! = \sqrt{2\pi k}\left(\frac{k}{e}\right)^k e^{(\frac{\theta}{12k})},\ 0 < \theta < 1. \tag{2.7}$$

Thus by Equation 2.6 and Equation 2.7,

$$\begin{aligned} [2(n-3)(2n-7)]^d &\geq \frac{\sqrt{2\pi(2n-4)}\,(2n-4)^{(2n-4)}\,e^{-(2n-4)}}{2^{(n-2)}\sqrt{2\pi(n-2)}\,(n-2)^{(n-2)}\,e^{-(n-2)}\,e^{1/(12(n-2))}} \\ &= \sqrt{2}\,2^{(n-2)}\,(n-2)^{(n-2)}\,e^{-(n-2)}\,e^{-1/(12(n-2))}. \end{aligned} \tag{2.8}$$

Taking natural logarithms of both sides gives:

$$d[\log(4) + \log(n-3)(n-7/2)] \geq (n-2)[\log 2 + \log(n-2) - 1] + \frac{1}{2}\log 2 - \frac{1}{12(n-2)}. \tag{2.9}$$

Now for $n \geq 4$, we have that $\frac{1}{12(n-2)} \leq \frac{1}{2}\log 2$, so

$$d[\log(4) + \log(n-3)(n-7/2)] \geq (n-2)[\log 2 + \log(n-2) - 1], \tag{2.10}$$

and if we let $n \to \infty$ we get

$$\frac{d}{n-2} = \frac{\log 2 + \log(n-2) - 1}{\log(4) + \log(n-3)(n-7/2)} \to \frac{1}{2}. \tag{2.11}$$

□

2.6.3 Tree Bisection and Reconnection Diameter

The established link between the SPR and TBR operations, in particular Equation 2.2, allows us to analyse the adjacency graph $G_{TBR}(n)$.

Lemma 2.7 *$G_{TBR}(n)$ is connected.*

Proof This follows immediately as a result of Lemma 2.6(1) and Equation 2.2. □

$G_{TBR}(n)$ is not regular as the number of trees one TBR from any given unrooted binary tree is dependent on topology, as shown in Figure 2.5.

Theorem 2.7 $n/4 - o(n) \leq \Delta(G_{TBR}(n)) \leq n-3$

Proof For the second inequality, Equation 2.2 and Equation 2.5 immediately give that $\Delta(G_{TBR}(n)) \leq n-3$. For the first inequality we will use proof by contradiction. Suppose that $\Delta(G_{TBR}(n)) < n/4 - o(n)$. Then by Lemma 2.3, we are able to construct a path between any two trees from $UB(n)$ in $G_{SPR}(n)$ with length less than $n/2 - o(n)$. This contradicts Theorem 2.6. □

2.7 Maximum Agreement Forests

Definition 2.7.1 Suppose we have two binary trees T_1 and T_2 with $\mathcal{L}(T_1) = \mathcal{L}(T_2) = \mathcal{L}$.

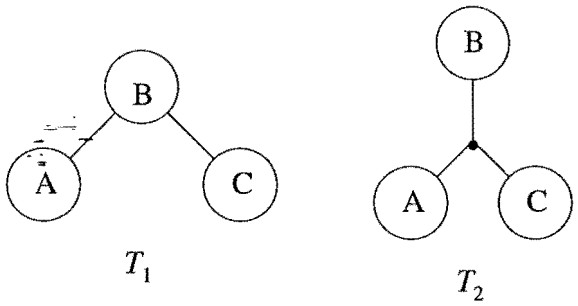
- An *agreement forest* (AF) for T_1, T_2 is a collection $\mathcal{F} = \{t_1, \ldots, t_k\}$ of binary trees such that, if we let $\mathcal{L}_j := \mathcal{L}(t_j)$ for $j \in \{1, \ldots, k\}$, then the following are satisfied:

  1. $t_j = T_{1|\mathcal{L}_j} = T_{2|\mathcal{L}_j}$ for all $j \in \{1, \ldots, k\}$; and
  2. for both $i = 1$ and $i = 2$ the trees $\{T_i(\mathcal{L}_j) \ : \ j = 1, \ldots, k\}$ are vertex-disjoint subtrees of T_i.

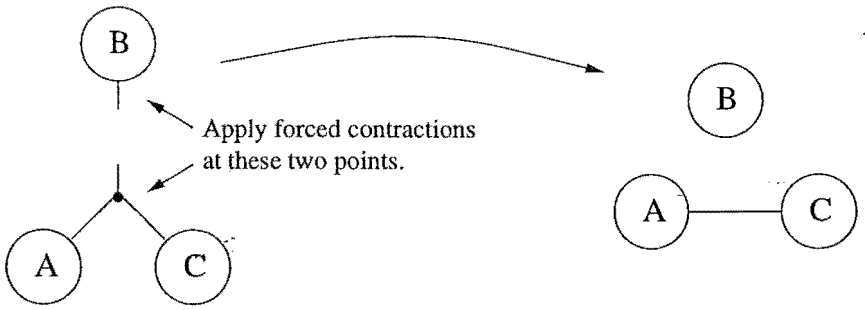
- A *maximum agreement forest* (MAF) for T_1, T_2 is an agreement forest for which k is minimal. Let $m(T_1, T_2)$ denote the value of $(k-1)$ for the minimal k.

Informally, $m(T_1, T_2)$ is the smallest number of edges that need to be cut from each of T_1 and T_2 so that the resulting forests agree, once unlabelled vertices of degree less than three are removed (by forced contraction).

Counting the edges cut, rather than the components, in a MAF may seem intuitive at first. The complicating factor, that may not be apparent upon first glance, is that internal vertices may be present in T_1 or T_2 which do not appear in any component. Consider the following:



Obviously, we can cut two edges in T_1 to construct the forest with trees A, B, and C. We also only need to cut two edges in T_2. If we simply cut the edge incident with tree B and then applying forced contractions to the resulting forest we cause any the internal vertex to be removed. Now we simply have to cut the edge between trees A and C, again applying a forced contraction.



Lemma 2.8 *For $T_1, T_2 \in UB(n)$, the same number of edges must be cut in both T_1 and T_2 to construct their MAF.*

Proof As hinted at, any potential difficulty lies with internal vertices of T_1 or T_2 that are not in any component. Since these vertices are internal, they have degree three. As we prune any component from such a vertex we apply a forced contraction that removes the resulting vertex of degree two and joins the two incident edges. □

This lemma justifies our definition of m. The next lemma shows that there may be several possible MAF's for a given pair of binary trees.

Lemma 2.9 *A MAF for $T_1, T_2 \in UB(n)$ need not be unique.*

Proof Let T_1, T_2 be two different unrooted binary trees on four leaves. Since they are on four leaves, by removing the same leaf from both trees we obtain a MAF, however there are four possible leaves that we can remove. □

2.8 TBR Distance and MAF Size

Theorem 2.8 *Suppose we have two binary trees T, T' with $\mathcal{L}(T) = \mathcal{L}(T') = \mathcal{L}$. Then,*

$$d_{TBR}(T, T') = m(T, T').$$

In particular, m is a metric.

Proof We first show that $m(T, T') \leq d_{TBR}(T, T')$ by using induction on $k = d_{TBR}(T, T')$. If $k = 1$, then only one edge needs to be cut in each of T and T' in order to construct a MAF, hence the hypothesis holds.

Now, suppose that the hypothesis holds for pairs of trees with a TBR-distance of $k \geq 1$ and suppose $d_{TBR}(T, T') = k + 1$. Then there is a tree T'' such that $d_{TBR}(T, T'') = k$ and $d_{TBR}(T'', T') = 1$. Thus, by the inductive hypothesis, there exists a partition $\pi = \{A_1, \ldots, A_k\}$ of $\mathcal{L}$ such that $\{T''_{|A_i} : i = 1, \ldots, k\}$ is a MAF for (T, T''), and a bipartition $\pi' = \{A, B\}$ of $\mathcal{L}$ such that $\{T''_{|A}, T''_{|B}\}$ is a MAF for (T'', T'). Now, by considering the subtrees $\{T''(A_i) : i = 1, \ldots, k\}$ of T'', we see that π' either splits no set in π (case (i)), or π' splits precisely one set in π - say A_j (case (ii)). Thus, if we set π'' equal to π in case (i), or equal to $\{\pi - \{A_j\}\} \cup \{A_j \cap A, A_j \cap B\}$ in case (ii), we have that $\{T''_{|U} : U \in \pi''\}$ forms an agreement forest for (T, T'') and for (T'', T') and thereby for (T, T'). Thus, $m(T, T') \leq k + 1$, which completes the induction step.

To show that $m(T, T') \geq d_{TBR}(T, T')$, we again use induction, this time on $m = m(T, T')$. For $m = 1$, the MAF is obtained by deleting a single edge from each of T and T', hence $d_{TBR}(T, T') = 1$. Now suppose the inductive hypothesis holds for $m \leq k - 1$ and that $m(T, T') = k$. Let $\{t_1, \ldots, t_{k+1}\}$ be a MAF for T, T'. For at least one $i \in \{1, \ldots, k + 1\}$, the subtree $T(\mathcal{L}_i)$ of T can be pruned from the rest of T by deleting one edge only. In T' there exists at least one $j \in \{1, \ldots, k+1\}$ such that $T'(\mathcal{L}_i)$ is joined to $T'(\mathcal{L}_j)$ by a path that does not include any vertices in $\cup_{m \neq i,j} T'(\mathcal{L}_m)$. Note that this last sentence could not also be true with T' replaced by T, else we could construct a smaller MAF for T, T' by amalgamating $\mathcal{L}_i$ and $\mathcal{L}_j$. Now, we can cut the single edge of T incident with $T(\mathcal{L}_i)$ and then re-attach $T(\mathcal{L}_i)$ to $T(\mathcal{L}_j)$ in such a way that $T_{|\mathcal{L}_i \cup \mathcal{L}_j} = T'_{|\mathcal{L}_i \cup \mathcal{L}_j}$ We call this new tree T'' and note that it must differ from T by exactly one TBR. T'' and T' now have an AF of size k, and so $m(T'', T') \leq k - 1$. Thus, by the inductive

hypothesis, $d_{TBR}(T'', T') \leq k - 1$. Thus $d_{TBR}(T, T') \leq d_{TBR}(T, T'') + d_{TBR}(T'', T') \leq k$ as required to establish the induction step.

We conclude that $d_{TBR}(T, T') = m(T, T')$. □

By Equation 2.2 and the first inequality of Theorem 2.8 we have

$$d_{SPR}(T_1, T_2) \geq m(T_1, T_2). \tag{2.12}$$

Thus, the SPR-distance is greater than or equal to the number of edges that are cut to create the maximum agreement forest. This means that the number of components in a MAF for any two unrooted binary trees is at most one more than the SPR-distance between the two trees. The counterexamples of Section 3.3.2 demonstrate that the inequality can be strict.

2.9 Applications to Evolutionary Biology

The basis for most tree reconstruction of the "true" evolutionary tree for a set of n species is a set of n pre-aligned sequences of DNA. DNA sequences can be regarded as long strings made up of the letters A, G, C, and T. These represent the four base nucleotides; adenine, guanine cytosine and thymine. For simplicity most authors simply use their first letter.

Subtree transfer operations are useful in describing certain evolutionary events, in particular SPR's can be used to model two evolutionary events, *horizontal gene transfer* and *recombinations*, both of which cause can cause significant change to DNA.

2.9.1 Horizontal Gene Transfer

Definition 2.9.1 A *horizontal gene transfer* or HGT is the transfer of genetic information from one genome to another, specifically between two species [14].

Horizontal gene transfer differs from normal *vertical gene transfer*, that sees genetic information passed from the parental generation to the progeny. Instead genetic information is passed from one species to another, usually by viruses.

Retroviruses are common mechanisms for HGT, as they are able to transport genetic material and have the molecular machinery for inserting foreign DNA in to a host genome. However, not every HGT will result in changes to a genome. In fact, a gene transferred horizontally is less likely to retain its functionality than a gene transfered from another genomic location with the same species [14].

The frequency of HGT in the biosphere is undetermined. Indeed, there is no actual way to *prove* that it has occurred. Nevertheless, several well known examples make the case for the presence of HGT highly probable. Two such examples, taken from Graur and Li [14], involve cats and monkeys, and fruit-flies.

Cats and Monkeys

A certain type-C virogene is found in Baboons, and homologous sequences have been found in Old World monkeys. This similarity between the sequences and the taxological relationship is consistent with vertical evolution, and the virogene is believed to have been present for approximately 30 million years. The virogene is also found in six species of cat that are closely related to the domestic cat, but it is not present in any other Felidae, such as lions or tigers, nor is it believed to be found in any other carnivores. Thus there is a high probability that the gene has been transferred horizontally from a recent ancestor of the baboon to cats. This is believed to have been about 5–10 million years ago. See Figure 2.7.

Fruit-flies and P Elements

The second exhibit in the case for HGT involves the P elements in the fruit-fly *Drosophila melanogaster*, which has rapidly spread throughout natural populations in the last 100 years. None of the closely related species *D. mauritania*, *D. séchellia* or *D. simulans* have P sequences. However the distantly related *D. saltans* contains P-like sequences very similar to those found in *D. melanogaster*, while *D. willistoni* also has P elements that differ by one base substitution from those of *D. melanogaster*.

The Building Blocks of Life

Some biologists conjecture that life originally evolved through HGT. Early autotrophic prokaryotes were able to photosynthesize and, as they did not depend on organic nutrients, began to proliferate. As they did, the byproduct of their photosynthesis, oxygen, gradually built up in the environment and formed the Ozone layer, enabling life to move on to the land. The eukaryotes were also developing around this time, however they could not photosynthesize, at least until they began engulfing autotrophic prokaryotes in a process known as primary endosymbiosis. The absorbed prokaryote gradually donated a large part of its genome to it eukaryote host, the endosymbiont. Plants and algae have evolved from this process of primary endosymbiosis and more complicated cellular organisms have evolved by a process known as secondary endosymbiosis. This occurs when the endosymbiont becomes absorbed. Gilson and McFadden [7] review secondary endosymbiosis further.

Application of the SPR

An SPR can be used to model the process of HGT. Suppose that a proposed evolutionary tree for a set of species S has been constructed from genetic information known *not* to contain any HGT (or any other factor causing discrepancy between the true tree and the proposed tree). Now suppose that a second

proposed evolutionary tree is produced for S, in which the genetic information is known to contain a single HGT. Then an important question is how far apart can the two proposed trees be. The answer to this is at most one SPR. The two trees may not differ if the HGT causes a subtree to be pruned and then regrafted to an adjacent edge.

If several HGT are present in the data for a species set S, then two different evolutionary trees T_1, T_2 could be constructed from different genetic data sets. The SPR distance between the two trees then gives a lower bound on the number of HGT in the set, assuming that HGT's are the only source of changes in the genetic information. SPR's can also be *masked* if the same gene undergoes HGT twice, effectively hiding the first SPR.

In practise HGT are only one (unlikely) cause of discrepancies between proposed evolutionary trees and the true species tree. We have also assumed that the analysis of the genomic data returns the "correct" tree for that data, this again can not be taken for granted.

2.9.2 Recombination

Definition 2.9.2 A *recombination* occurs when two sub-sequences from two different sequences join to create a new sequence. The point where the two sub-sequences meet in the new sequence is called the *recombination point* [10].

Recombinations are sometimes referred to as *cross overs*, however we will not use this terminology, as an NNI is also referred to as a cross over by some authors. Figure 2.8 provides an example of a recombination.

The conventional model of evolution fails for recombinations because there are two sequences ancestral to the new sequence. In general the sub-sequence to the left of the recombination point has a different evolutionary history to the sub-sequence to the right of the recombination point, and hence we can not use a tree to describe the evolutionary history of the new sequence. Instead we need to use a tree to describe the history of the left sub-sequence and a different tree to describe the history of the right.

Tree reconstruction methods, such as *Maximum Parsimony* or *Neighbour Joining*, do not allow for recombinations, thus when sequences are analysed that contain recombinations erroneous trees are reconstructed, see [8] and [9].

If we were to analyse the left sub-sequences and the right sub-sequences and assuming that the reconstruction method returned the correct trees for both sides, then the question is how different can the two trees be. Assume that the reconstruction has occurred and the two recovered trees that describe the evolutionary history of the left and right sub-sequences are T_l and T_r. If exactly one recombination is present in the data then $d_{SPR}(T_l, T_r) \leq 1$. If several recombinations are present in the data, then we recover several evolutionary trees that are at most one SPR apart, where each one corresponds to a recombination.

In general, we can not guarantee that from a set of sequences, all recombination can be identified. For example, a first recombination may occur at a recombination point and then a second recombination might occur at the same point in the sequences, which will effectively mask the first recombination. Similarly recombination may occur that do not change the topology of the evolutionary tree [9].

Given two evolutionary trees for a set of sequences, say T_1 and T_2, then $k = d_{SPR}(T_1, T_2)$ is a lower bound on the number of recombinations present in the data, again under the assumption that only recombinations are causing the differences in the two trees.

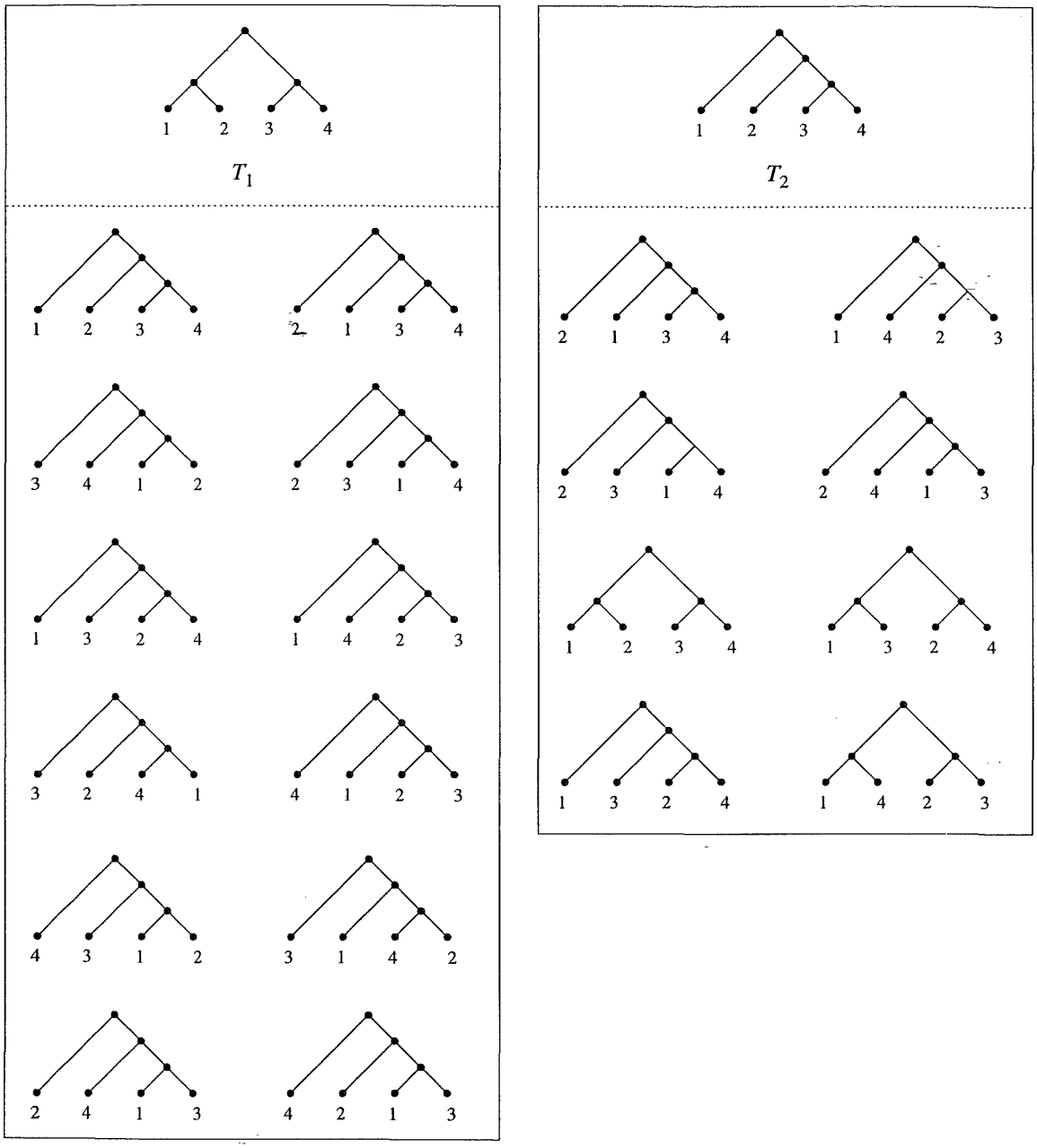


Figure 2.3: An example that tree topology does affect the number of trees within one SPR from a rooted binary tree. The tree T_1 has twelve other trees at distance one SPR, while T_2 has only eight.

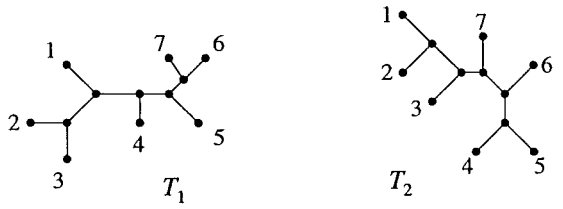


Figure 2.4: Example of a TBR. The tree T_1 has been split into two subtrees, one with leaf set $\{1,2,3\}$ and the other with leaf set $\{4,5,6,7\}$. A new edge has then been created between the two subtrees to reconnect them giving T_2.

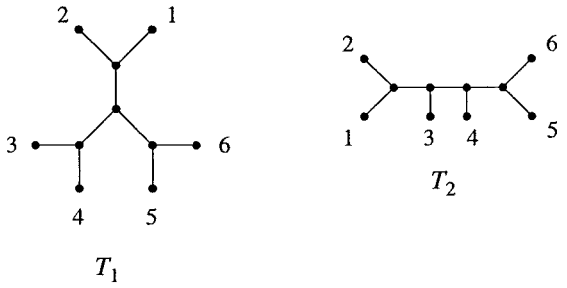


Figure 2.5: An example that tree topology does affect the number of trees within one TBR from an unrooted binary tree. In the tree T_1 no edge can be cut to obtain a subtree with three leaves. Thus any TBR involves subtrees with four and two leaves, or with five and one. In either case a subtree with one or two leaves can only be reconnected in one way, hence any TBR is also an SPR. However we can cut the central edge of T_2 to obtain two subtrees of 3 leaves each. Now we can reconnect these two subtrees in such a way that the resulting tree is at SPR-distance two, but TBR distance one. Hence T_2 will have more trees one TBR from it than T_1.

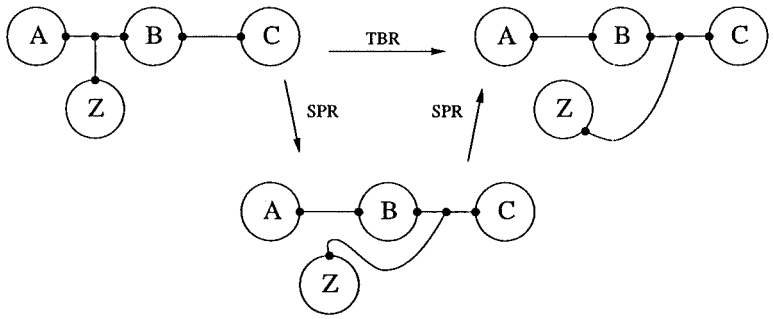


Figure 2.6: Illustration of the Proof of Lemma 2.3.

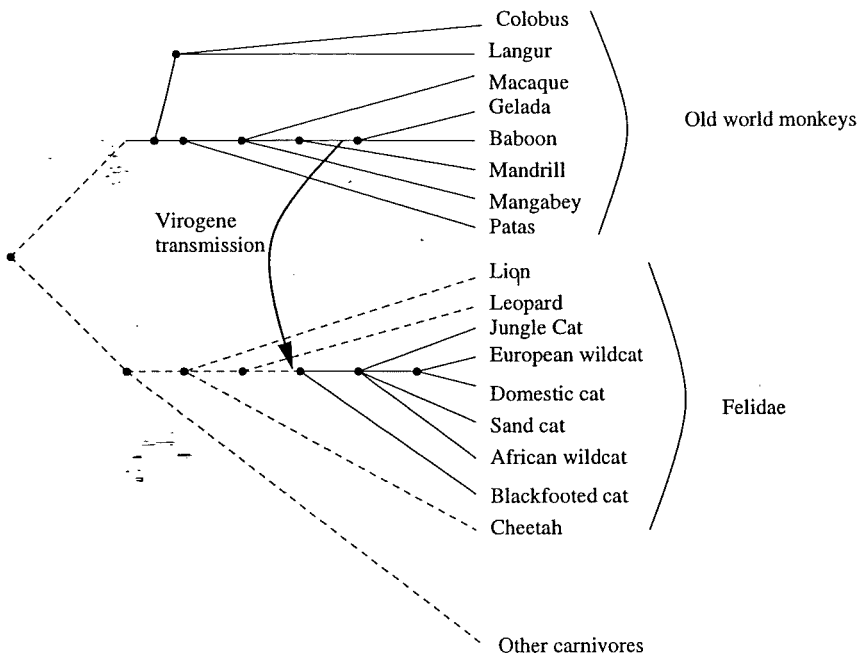


Figure 2.7: A virogene contained in Old World Monkeys is believed to have been horizontally transferred to some members of the Felidae family about 5–10 Million years ago. The dashed lines indicate histories of species not containing the virogene and the solid lines indicate histories of species that do contain the virogene. (Diagram based on that found in [14])

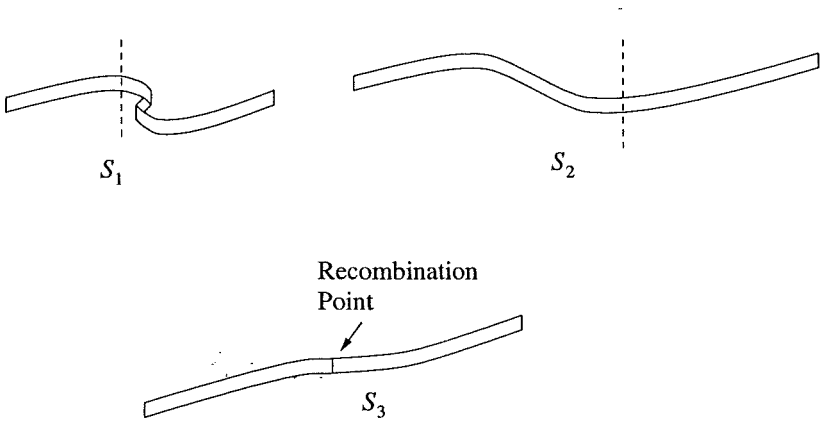


Figure 2.8: Example of a recombination. The left sub-sequence of sequence S_1 and the right sub-sequence of sequence S_2 recombine to form the new sequence S_3.

Chapter 3

Complexity of Computing Distances Between Evolutionary Trees

A fundamental problem is determining the distance between two given trees from $UB(n)$ with respect to some tree metric. Before we dive in the deep end we will formalise our problem nomenclature.

3.1 Tree Distance Problems

In this section, we briefly define the main problems, each whose complexity will later be examined.

The Θ-Distance Problem
Instance: Two trees, T_1 and T_2, from $UB(n)$.
Question: What is $d_\Theta(T_1, T_2)$, where $\Theta \in \{NNI, SPR, TBR\}$?

This "three-in-one" problem is an optimization problem and as such will not fit in with the definition of *NP*-completeness. Thus we also introduce a new "three-in-one" *decision* problem.

The Parameterized Θ-Distance Problem
Instance: Two trees, T_1 and T_2, from $UB(n)$ and a parameter $k \in \mathbb{N}$.
Question: Is $d_\Theta(T_1, T_2)) \leq k$, where $\Theta \in NNI, SPR, TBR\}$?

3.2 Tree Distance Problems and Class *NP*

The first step in analysing the complexity of the Θ-Distance Problem for $\Theta \in \{NNI, SPR, TBR\}$, is to show that the Parameterized Θ-Distance Problem is in the class *NP*. This class is defined formally in Garey and Johnson [6], however for our purposes the following loose definition will suffice.

Definition 3.2.1 *NP* is the class of all decision problems Π that under reasonable encoding schemes can be solved by polynomial time nondeterministic algorithms.

Theorem 3.1 *The Parameterized Θ-Distance Problem is in NP for $\Theta \in \{NNI, SPR, TBR\}$.*

Proof These three problems can all be solved by a non-deterministic algorithm, as they are equivalent to searching for a path between two vertices of $G_\Theta(n)$ of length not more than k. □

Definition 3.2.2 A problem L is in the class *NP-complete* if it is in *NP* and every problem in *NP* can be transformed to L in polynomial time.

This definition informally says that it is at least as difficult to solve a *NP*-complete problem as it is to solve any problem in *NP*. However, optimisation problems, such as trying to find the minimum distance between two trees from $UB(n)$ are not in *NP* and hence cannot be *NP*-complete.

Definition 3.2.3 A problem L is said to be *NP-hard* if there is a problem L' which is *NP*-complete and if L' *Turing reduces* L in polynomial time.

Note see [6] for further details of complexity theory.

If we can show that the problem of determining whether $d_\Theta(T_1, T_2) \leq k$, for $T_1, T_2 \in UB(n)$, is *NP*-complete, then the problem of determining the distance between the two trees is *NP*-hard.

3.3 Conventional Complexity of Tree Metric Problems

In this section, we examine the complexity of the tree metric problems for the NNI, SPR and TBR operations.

3.3.1 The NNI distance Problem

The NNI-distance between two trees from $UB(n)$ can be computed using a brute force method. Start with T_1 and construct all the trees at NNI-distance one and check if any of these are T_2, if so then stop, otherwise construct all trees one NNI from those trees constructed previously and check these trees, stopping if one is T_2, and so on recursively until a match it made. At any one stage we are constructing $2n - 6$ trees, and then another $2n - 6$ trees from those trees. In fact it is not hard to see that if $d_{NNI}(T_1, T_2) = k$ then we have to check $\mathcal{O}((2n)^k)$ trees. This will lead to the problem size quickly growing to computationally intractable levels. In a realistic example with say 100 species and two trees say five NNI apart, then the number of trees to check is of 10^{11} in magnitude. This calculation is intractable and hence of no practical use.

The first to try and reduce the computational complexity of computing d_{NNI} were Waterman and Smith [22] in 1978. They suggested an algorithm based on the decomposability of two trees about any shared split, or bi-partition of the leaf set, occurring in both trees. The authors suggested that this algorithm could be calculated efficiently, and conjectured that this might be an practical method for computing the the NNI distance. However, their algorithm has been shown to be ill-defined for two trees not sharing a split and the distance preserving quality of the decomposition has also been shown to be incorrect, leaving the question of *NP*-completeness still unresolved. Suspicions were raised by Jarvis *et al.* [12] in 1983 who showed Theorem 3 of [22] was incorrect along with Theorem 5. Jarvis *et al.* [11] also concluded that the status of Theorem 4 of [22] was also incorrect. Since then, many authors including Page [18] and Li *et al.* [13], among many, have all come up with counterproofs and counterexamples to the claims in [22] but the complexity still remains undecided, although most would expect the problem to be *NP*-complete — that is intractable.

3.3.2 The SPR Distance Problem

Seemingly the only paper to address the complexity of the SPR-problem is [10] in 1996. However the authors base their treatment on Lemma 7 of [10] that states that "The size of a MAF of T_1 and T_2 is one more than their subtree-transfer distance". The size of a MAF is taken to mean the number of component trees in the forest and the subtree-transfer distance refers to the subtree prune and regraft distance. However the lemma is incorrect as the counterexamples in Figure 3.1 show.

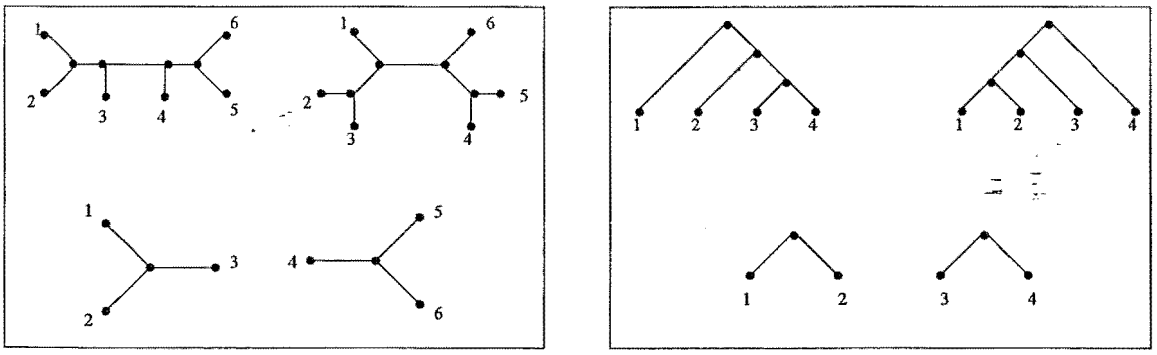


Figure 3.1: Counterexamples to Lemma 7 of [10]. In the first (resp. second) box there are two unrooted (resp. rooted) binary trees that are more than one SPR apart, yet their MAF requires just one edge deletion.

Because Lemma 7 of [10] is necessary in Theorem 8 of [10], the latter stating that it is *NP*-hard to compute the SPR distance between two binary trees, we must conclude that the theorem and hence the complexity of the SPR distance problem remains unresolved.

3.3.3 The TBR Distance Problem

Theorem 3.2 *For any two unrooted binary trees on n leaves, T_1 and T_2, computing $d_{TBR}(T_1, T_2)$ is NP-hard.*

Proof If we replace Lemma 7 of [10] by Theorem 2.8 of this thesis, then the amended proof of Theorem 8 of [10] shows that computing the TBR-distance is *NP*-hard. □

3.4 Fixed Parameter Tractability for Tree Metrics

Often when a problem is shown to be *NP*-complete, further investigation of the complexity is abandoned. Such problems are deemed to grow exponentially, making them intractable for large instances, and thus further investigation is pointless. However, many researchers working with *NP*-complete, and supposedly intractable, problems have come across ways of solving these problems that do not grow exponentially with the size of the problem instance.

The ideas used to reduced the computational complexity of such problems are almost always relevant to a single problem only and solutions for some problems do not seem to exist. One of the observations made of problems with better than expected complexity is that some parameter of the problem is bounded and small. This introduces the idea of *Parameterised computational complexity.*

By investigating problems, and in particular, searching for parameters of a problem with naturally low bounds, classification of problems can be made according to their complexity in a manner much more practical than traditional classifications. For a formal treatment see [3], [4] or [5].

Following [3], the basic framework of parameterised complexity is *fixed-parameter tractability*, a concept we now outline.

Definition 3.4.1 A problem $P = (x, k)$ is *fixed-parameter tractable* if it can be determined in time $f(k)n^{\alpha}$, where x is the problem input, k is a parameter, $|x| = n$, α is a constant independent of both n and k, and f is some arbitrary function. The family of fixed-parameter tractable problems is denoted *FPT.*

It is shown in [2] that the definition of *FPT* is unchanged if $f(k)n^{\alpha}$ is replaced by $f(k) + n^{\beta}$, where α, β are independent of both n and k.

Showing that a problem classed as *NP*-hard is in *FPT* means that it *may* not be intractable for large problem instances. There are many such problems that are able to be solved when the problem becomes huge, for others the improvement is only slight. Nevertheless, once a problem is classed as *FPT* it could potentially have it complexity greatly improved.

Our goal for the remainder of this chapter will be to show that the problem of calculating the TBR-distance between two unrooted binary trees on n leaves in *FPT*. But first we begin with the well examined *Vertex Cover Problem* to illustrate the notion of *FPT*.

The Vertex Cover Problem (Parameterised)

Given a graph $G = (V, E)$, where $|V| = n$, and a natural number k, the Vertex Cover Problem consists of determining whether G has a vertex cover of size k. That is, determining if a subset $V' \subseteq V$ exists with $|V'| \leq k$ such that for every edge $\{u, v\} \in E$, either $u \in V'$ or $v \in V'$. This problem is shown to *NP*-hard by Papadimitriou and Yannakakis [17].

This problem can be parameterized very naturally, by treating k as the parameter. One method of solving this problem would then be to exhaustively search all subsets of size k (k-subsets), requiring $\mathcal{O}(n^{k+1})$ time. However we can do much better when n is large, provided k is small.

To show that the Vertex Cover Problem is in *FPT* we have to show that it can be solved in time $\mathcal{O}(f(k)n^{\alpha})$. Papadimitriou and Yannakakis [17] have shown that this problem can be solved in $\mathcal{O}(3^k n)$, while Balasubramanian *et al.* [1] have improved this to $\mathcal{O}((53/40)^k k^2 + nk)$ (recall this is an equivalent form of the complexity). Hence the tractability of the Vertex Cover Problem is determined by the size of the vertex cover and not the number of vertices. If we are only interested in finding small vertex covers, say of maximum size ten, then we can find them or determine that they do not exist very quickly, even when n is large. Currently the best solution to the vertex problem is that of Downey, Fellows and Stege [5] and is $\mathcal{O}(r^k k^2 + nk)$ where $r = 4^{1/5}$.

Vertex Cover is a very encouraging problem, as its complexity has fallen significantly since it was originally classed as *FPT*. In fact once a problem is shown to be *FPT* it encourages investigation and therefore bounds will often improve.

3.4.1 Tree Reduction Rules

Despite the fact that the TBR-distance problem is *NP*-hard and the suspicion that so too is the SPR-distance problem, our aim is to show that both these problems are not as bad as the "*NP*-hard" tag makes them appear. We show that the Parameterized TBR-distance problem is *FPT*, while we conjecture that so too is the Parameterized SPR-distance problem.

The first step of a typical *FPT* problem is to *kernelize* the problem, that is, the size of the problem is reduced in such a way that the answer to the reduced problem is the same as the answer to the original problem and that the size of the reduced problem is some function of the parameter k, i.e. it does not involve n. In our case we wish to kernelize the problem by reducing the size of the two given trees, while still maintaining the SPR or TBR distance between them. We propose two ways to do this:

- **Rule 1** Replace any pendant subtree that occurs identically in both trees by a single leaf with a new label.

- **Rule 2** Replace any chain of pendant subtrees that occur identically in both trees by three new leaves with new labels correctly oriented to preserve the direction of the chain.

Figure 3.2 and Figure 3.3 illustrate Rule 1 and Rule 2 respectively, whilst Figure 3.4 provides an example of why three leaves are required for Rule 2.

Lemma 3.1 *For $T_1, T_2 \in UB(n)$ Rule 1 and Rule 2 can be repeatedly applied to reduce T_1 and T_2, until they can be reduced no further, in polynomial time.*

Lemma 3.1 is easily demonstrated. We will not attempt to do so here, nor quantify the time required. Useful further work might involve finding a fast implementation.

Preservation of TBR Distance

Definition 3.4.2 An *abc tree* is a binary tree T whose leaf set includes three leaves a, b, c with the following property; if v_a, v_b, v_c are the three vertices of T adjacent to a, b, c (resp.) then $\{v_a, v_b\}$ and $\{v_b, v_c\}$ are edges of T. See Figure 3.4.1.

Lemma 3.2 *(The abc lemma) If $T, T' \in UB(n)$ are two abc trees with $\mathcal{L}(T) = \mathcal{L}(T')$, then there exists a MAF $\mathcal{F}$ for T, T' in which a, b, c are contained in the leaf set of one of the trees in $\mathcal{F}$.*

Proof Suppose $\mathcal{F}$ is a MAF for T, T'. Let L_a (resp. L_c) be the set of leaves connected to a (resp. c) once edge $\{v_a, v_b\}$ (resp. $\{v_b, v_c\}$) is deleted from T. Let $L'_a = L_a - \{a\}; L'_c = L_c - \{c\}$. We now distinguish two cases:

1. There exists a tree $t \in \mathcal{F}$ with leaves from both L'_a and L'_c.
2. No tree in $\mathcal{F}$ contains leaves from both L'_a and L'_c.

In case (1), let $t_a = t_{|L'_a}$ and $t_c = t_{|L'_c}$, and let $I := |\mathcal{L}(t) \cap \{a, b, c\}|$. If $I = 0$ then each of a, b and c must be isolated point in $\mathcal{F}$ (by property (2) in the definition of an AF). Let $\mathcal{F}' := (\mathcal{F} - \{a, b, c, t\}) \cup \{t_a, t_c, t_{abc}\}$ (where t_{abc} is the tree with the three leaves a, b, c). Then $\mathcal{F}'$ is an agreement forest for T, T' with fewer trees than $\mathcal{F}$, contradicting the minimality of $\mathcal{F}$ - thus this case does not arise.

If $I = 1$, let x denote the leaf in $\mathcal{L}(t) \cap \{a, b, c\}$ and y, z denote the other two leaves. Then, y, z must be isolated vertices in $\mathcal{F}$ and so $\mathcal{F}' := (\mathcal{F} - \{y, z, t\}) \cup \{t_a, t_c, t_{abc}\}$ is also an AF for T, T' with the same number of trees as $\mathcal{F}$. Thus we can replace $\mathcal{F}$ by $\mathcal{F}'$ to obtain a MAF in which a, b, c occur in a single component.

If $I = 2$, then one of the leaves, $x \in \{a, b, c\}$ is an isolated vertex in $\mathcal{F}$. Let $t' := T_{|\mathcal{L}(t) \cup \{x\}}$. Then $\mathcal{F}' = (\mathcal{F} - \{x, t\}) \cup \{t'\}$ is also an AF forest for T, T', but with fewer trees than $\mathcal{F}$, a contradiction, so this case does not arise.

If $I = 3$, $\mathcal{F}$ already satisfies the condition we want and we are done.

In case 2, if $\mathcal{F}$ contains all three leaves a, b, c then we are done. Otherwise, we distinguish two subcases:

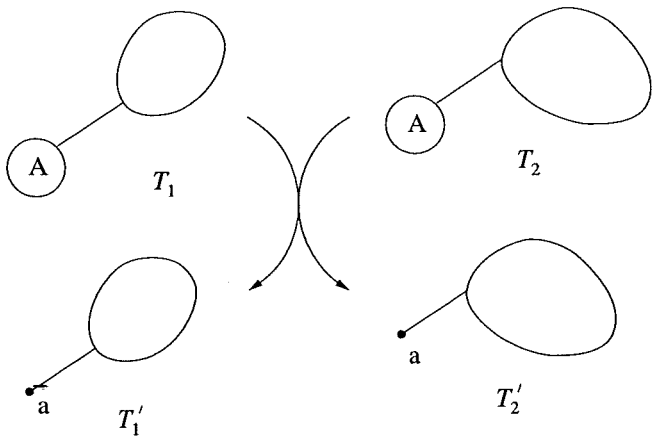


Figure 3.2: Reduction of two trees using Rule 1.

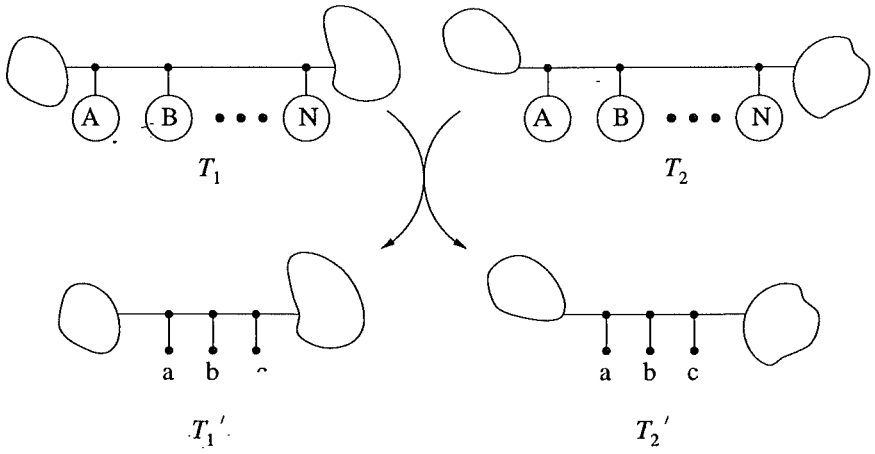


Figure 3.3: Reduction of two trees using Rule 2.

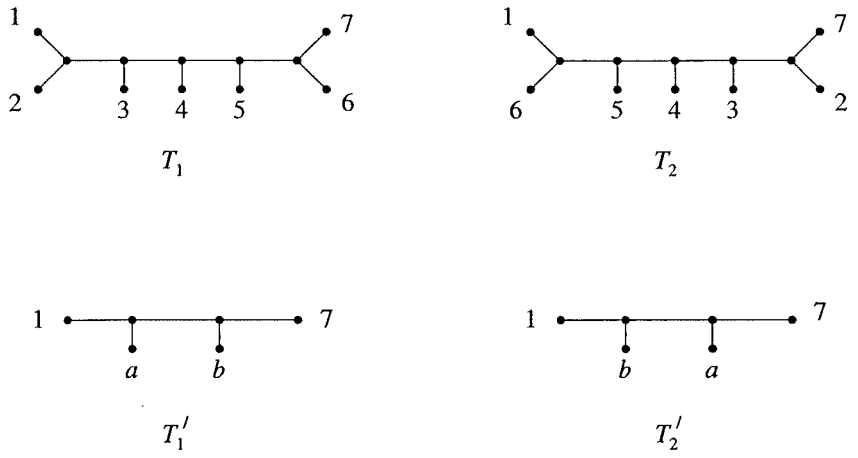


Figure 3.4: This figure gives an example of why at least three leaves are needed in the reduced chain. Initially $d_{TBR}(T_1, T_2) = 2$, however if we reduced the identical chain to only two leaves then $d_{TBR}(T_1', T_2') = 1$ (to achieve this prune off the vertex b in T_1' and regraft it on the other side of vertex a.)

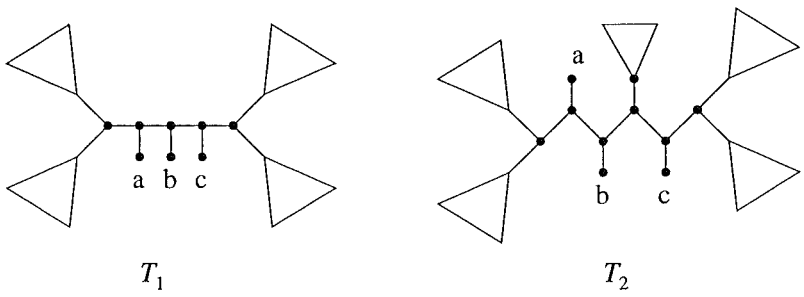


Figure 3.5: T_1 is an example of an abc tree, however T_2 is not, as there are more than three edges between b and c.

(i) at least one leaf $x \in \{a, b, c\}$ occurs as an isolated vertex in $\mathcal{F}$, or

(ii) leaves a, b are in one component $t_1 \in \mathcal{F}$ and leaf c is in another $t_2 \in \mathcal{F}$ (or leaves b, c are in one component, and leaf a is in another).

In subcase (i), delete a, b, c from any trees in $\mathcal{F}$ and replace isolated leaf x by the tree t_{abc} to obtain an AF for T, T' of the same size as $\mathcal{F}$. Since this contains a, b, c in one tree we are done.

In subcase (ii), let $t := T_{|\mathcal{L}(t_1) \cup \mathcal{L}(t_2)}$. Then $\mathcal{F}' := (\mathcal{F} - \{t\}) \cup \{t'\}$ is an AF for T, T' yet smaller than $\mathcal{F}$; a contradiction. □

Theorem 3.3 *Let $T_1, T_2 \in UB(n)$ and let T_1' and T_2' be obtained from T_1 and T_2 respectively by applying Rule 1 or Rule 2. Then $d_{TBR}(T_1, T_2) = d_{TBR}(T_1', T_2')$.*

Proof Rule 2 Label the subtrees in the chain shared by T_1 and T_2 as $t_1, \ldots, t_r$ where $r \geq 3$ (with this order). Suppose these are replaced by new leaves a, b, c under Rule 2. Thus T_1' and T_2' are both abc trees, and so there exists a MAF $\mathcal{F}$ for T_1', T_2' satisfying Lemma 3.2. Now, in these trees let us re-insert the trees $t_1, \ldots, t_r$ in this order in each of T_1', T_2' to new vertices that subdivide the edge $\{v_a, v_b\}$ (where v_a, v_b are the vertices adjacent to a and b). Call the resulting trees T_1'', T_2''. Now, any MAF for T_1', T_2' which has leaves a, b, c in the same component t can be modified to produce an agreement forest for T_1'', T_2'' of the same size, by simply attaching the trees $t_1, \ldots, t_r$ along the edge $\{v_a, v_b\}$ of t (or, in case $v_a = v_b$ in t, along the edge from a to v_a). Thus, by Theorem 2.8, $d_{TBR}(T_1'', T_2'') \leq d_{TBR}(T_1', T_2')$. However, since T_1, T_2 are both induced subtrees of T_1'', T_2'', Theorem 2.3 gives $d_{TBR}(T_1, T_2) \leq d_{TBR}(T_1'', T_2'')$ and thus $d_{TBR}(T_1, T_2) \leq d_{TBR}(T_1', T_2')$.

For the converse inequality, with $t_1, \ldots, t_r$ as before, suppose we select a leaf $a \in \mathcal{L}(t_1), b \in \mathcal{L}(t_2), c \in \mathcal{L}(t_3)$ and replace the chain $t_1, \ldots, t_r$ in T_1, T_2 by leaves a, b, c (correctly oriented) to obtain trees T_1', T_2'. Let U denote the set of leaves of T_1 that do not lie in the chain, together with a, b, c. Then, by Theorem 2.3, $d_{TBR}(T_{1|U}, T_{2|U}) \leq d_{TBR}(T_1, T_2)$, and since $T_{i|U} = T_i'$ for $i = 1, 2$ we obtain $d_{TBR}(T_1', T_2') \leq d_{TBR}(T_1, T_2)$, as required.

Combining both inequalities we get $d_{TBR}(T_1', T_2') = d_{TBR}(T_1, T_2)$.

Rule 1 Similar to, but simpler than Rule 2. □

Preservation of SPR Distance

Currently, we are only able to conjecture that Rule 2 is distance preserving for the SPR transformation.

Conjecture 3.4 *Let $T_1, T_2 \in UB(n)$ and let T_1' and T_2' be obtained from T_1 and T_2 by applying Rule 1 or Rule 2. Then $d_{SPR}(T_1, T_2) = d_{SPR}(T_1', T_2')$.*

The proof that Rule 1 is distance preserving for the SPR operation is straight forward.

Preservation of NNI Distance

Despite the fact that Rule 1 and Rule 2 are distance preserving for the TBR-distance, and conjectured to preserve SPR-distance, Rule 2 does not preserve NNI-distance.

Definition 3.4.3 Given a tree T and leaves $i, j \in \mathcal{L}(T)$, let $\Delta_{i,j}(T)$ be the number of edges between i and j.

Lemma 3.3 *For two binary trees T and T' on n leaves, such that $d_{NNI}(T, T') = 1$, $|\Delta_{i,j}(T) - \Delta_{i,j}(T')| \leq 1$.*

Proof Suppose that we have T and T' as above. Consider the four subtrees A, B, C, D that are rearranged by an NNI operation. The result follows immediately from considering cases where i and j occur in any of these subtrees. □

Lemma 3.4 *Rule 2 does not preserve NNI-distance.*

Proof By the triangle inequality and Lemma 3.3 if $|\Delta_{i,j}(T) - \Delta_{i,j}(T')| > k$, then $d_{NNI}(T, T') > k$. Now consider the four trees in Figure 3.6. Rule 2 reduces T_1 (and T_2 resp.) to T_1' (T_2') and $d_{NNI}(T_1', T_2') = 3$. However $\Delta_{1,2}(T_1) = 1$ and $\Delta_{1,2}(T_2) = n-2$, hence $|\Delta_{1,2}(T_1) - \Delta_{1,2}(T_2)| = n-3$, thus $d_{NNI}(T_1, T_2) \geq n-3$. Choosing $n = 7$ gives $d_{NNI}(T_1, T_2) > d_{NNI}(T_1', T_2')$. Thus, Rule 2 does not preserve NNI-distance. □

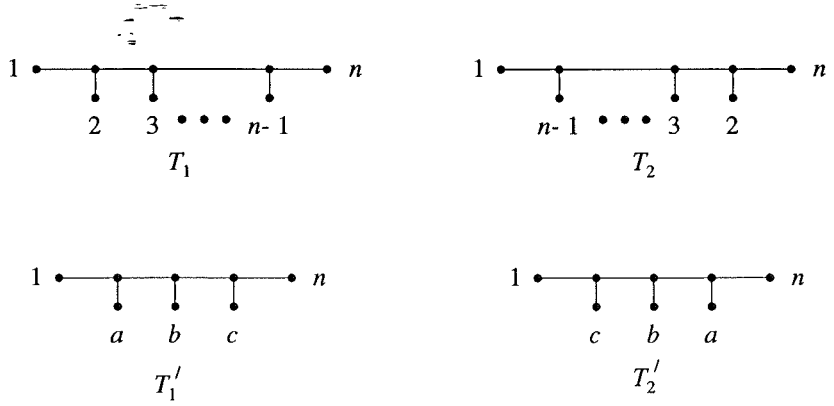


Figure 3.6: Consider the two trees T_1 and T_2. When reduced using Rule 2 we get T_1' and T_2'. By Lemma 3.4 $d_{NNI}(T_1, T_2) > d_{NNI}(T_1, T_2) = 3$ for $n \geq 7$.

3.4.2 Bounded Size of Maximally Reduced Trees

Suppose that we are given $T_1, T_2 \in UB(n)$ such that $d_\Theta(T_1, T_2) = k$ for $\Theta \in \{SPR, TBR\}$, and that T_1 and T_2 can be reduced no further by Rule 1 or Rule 2. In this section, we show that the size of the leaf set of the two trees is bounded by some function f which depends only on k, *ie* $|\mathcal{L}(T_i)| \leq f(k)$, where $i \in \{1, 2\}$. Our goal will be Theorem 3.5, but on the way we will need several new definitions and lemmas.

By Equation 2.12 there is a MAF for T_1 and T_2 with at most to $k+1$ components. Let $t_1, t_2, \ldots, t_r$ be the components of the MAF where $r \leq k+1$. To find an upper bound for the size of T_1 and T_2 we determine a bound on the size of each component.

If the size of the leaf set of a component t_j is one, then it is impossible to reduce the size of the component further, hence the upper bound for the size of the leaf set of this component is always one, and thus we do not need to consider this case. For this reason, all components will be assumed to have a leaf set of size greater than one.

To upper bound the size of the leaf set for the reduced components, we begin by introducing two new definitions.

Definition 3.4.4 Given $T_1, T_2 \in UB(n)$ and their MAF with components $t_1, t_2, \ldots, t_r$, the edges in T_1 or T_2 that connect the components are the *intercomponent edges.* The number of intercomponent edges incident with component t_j in T_i is the *component degree* and shall be denoted $\deg^i(t_j)$. See Figure 3.7.

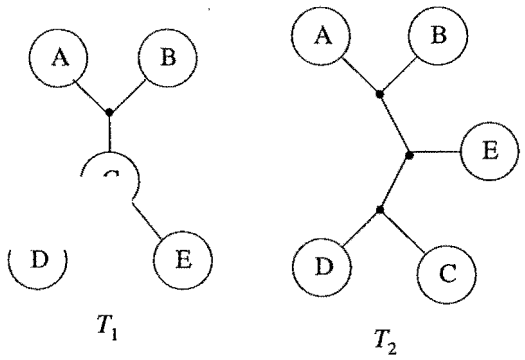


Figure 3.7: T_1 and T_2 are made up of the same components (large circles) and intercomponent edges. In the tree T_1 component C has degree three while in T_2 it has degree one. Note that vertices are present in both trees that are not in any component and that in T_2 intercomponent edges exist that are not incident to any component.

Definition 3.4.5 In the trees T_1 and T_2 there may be vertices between components that disappear when the MAF is constructed. We shall call these vertices *non-component vertices.* Note also that no leaf of T_i can disappear under a forced contraction and so must be in a component. Thus, all non-component vertices are internal.

Definition 3.4.5 allows us to sum the component degrees over all components in much the same way as one can sum degrees over all vertices in a tree.

Lemma 3.5 *Let $t_1, t_2, \ldots, t_r$ be the trees in a MAF for T_1 and T_2 where $r \leq k+1$ and $i \in \{1,2\}$. Then $\Sigma_{j=1}^{r} \deg^i(t_j) \leq 2k$ for $i = 1, 2$.*

Proof Reduce each component to a vertex labelled with the label of that component. We shall call these vertices *component-vertices*, and note that we are now summing over the degree of these vertices instead of the component degrees of each component. However, the notation remains the same, so we continue to write $\Sigma_{j=1}^{r} \deg^i(t_j)$. Figure 3.8 illustrates this transformation.

For T_i reduced in this manner, we use induction on the number of non-component vertices. If T_i contains no non-component vertices, then the total number of vertices is r and hence $\Sigma_{j=1}^{r} \deg^i(t_j) = 2r-2$ by Lemma 1.1. Since $r \leq k+1$, we have $\Sigma_{j=1}^{r} \deg^i(t_j) \leq 2k$. Hence the hypothesis holds.

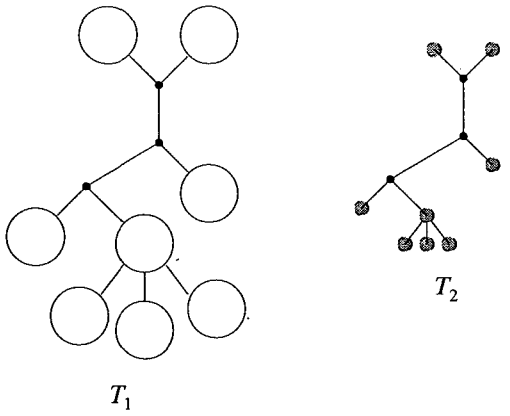


Figure 3.8: T_1 is made up of components (large circles) and three non-component vertices. T_1 is transformed to T_2 by replacing the components by component vertices (grey filled circles). The non-component vertices remain unchanged. Note that T_2 need not be binary.

Assume that if l non-component vertices are present in a tree $T_{i,l}$, then $[\Sigma_{j=1}^{r} \deg^i(t_j)]_{T_{i,l}} \leq 2r-2$ and suppose that we are given a tree $T_{i,l+1}$ with $(l+1)$ non-component vertices. Then there must be at least one non-component vertex w adjacent to two component vertices. Suppose that the two component vertices are labelled t_u and t_v. If we now prune t_v and regraft it to t_u then $\deg(t_v)$ remains unchanged and $\deg(t_u)$ has increased by one. Finally w is removed as it has degree two and its two incident edges are amalgamated into a single edge. Call this new tree $T'_{i,l+1}$, and note that it only contains l non-component vertices, but that its sum over the degrees of component vertices is one more than that of $T_{i,l+1}$. By the induction hypothesis $[\Sigma_{j=1}^{r} \deg^i(t_j)]_{T'_{i,l+1}} \leq 2r-2$, so $[\Sigma_{j=1}^{r} \deg^i(t_j)]_{T_{i,l+1}} < 2r-2 \leq 2k$. Thus $\Sigma_{j=1}^{r} \deg^i(t_j) \leq 2k$ until no more internal non-component vertices can be added (in which case all component vertices are leaves.) Figure 3.9 illustrates the induction step. □

Consider $T_1, T_2 \in UB(n)$ and a MAF with components $t_1, t_2, \ldots, t_r$. The component must be connected differently in each tree otherwise the MAF would not be maximal — this shows the essential differences between T_1 and T_2. The way the components are linked will determine the size of the upper bound for each component, not the number of leaves in the component. If we want to examine the similarities and differences of a component in both trees we must include the intercomponent edges. Our goal will be to reduce the identical sections of t_j^1 and t_j^2 using Rule 1 and Rule 2 and thereby find a suitable reduction for t_j and so on for T_1 and T_2.

We begin by putting $s_j = \deg^1(t_j) + \deg^2(t_j)$. Hence the component t_j has s_j edges incident with it in total. We then proceed to define four new trees, all based on component t_j.

Definition 3.4.6 Let t_j^1 be the tree obtained from t_j by adding $\deg^1(t_j)$ new leaves, such that their pendant edges subdivide the same edges as the intercomponent edges in T_1. Let R_j^1 be the leaf set of the

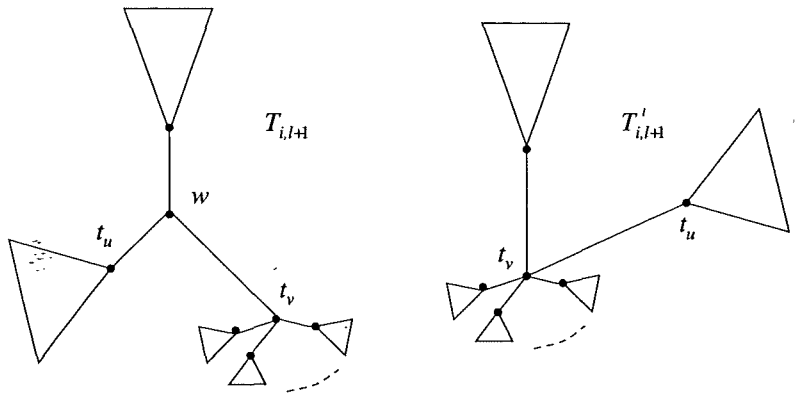


Figure 3.9: $T_{i,l+1}$ and $T'_{i,l+1}$ are unrooted trees on n leaves (neither of which need be binary.) $T_{i,l+1}$ contains a non-component vertex w that shares edges with component vertices t_u and t_v. $T'_{i,l+1}$ is obtained from $T_{i,l+1}$ by pruning t_u from w and regrafting it to t_v.

newly added leaves.

We define t_j^2 similarly, except we use T_2 instead of T_1 and so that that $R_j^1 \cap R_j^2 = \emptyset$. Let τ_j be t_j with s_j leaves added to t_j again so that their pendant edges subdivide the same edges as those subdivided by intercomponent edges in *both* T_1 and T_2. Let $R_j = R_j^1 \cup R_j^2$ and note that $R_j \cap \mathcal{L}(t_j) = \emptyset$. Finally note that $\tau_{j|R_j}$ is τ_j with leaf set restricted to R_j. Figure 3.10 illustrates this definition.

The trees τ_j and $\tau_{j|R_j}$ will be the main tools used to find the upper bound for the size of the component t_j. This will be done in Theorem 3.5, however before we can do so we need several lemmas.

It is quite conceivable that any one edge of t_j may be subdivided several times. In fact an edge may be subdivided by components in both T_1 and T_2. The order in which the edge is subdivided is implicitly defined when constructing t_j^1 and t_j^2, however when constructing τ_j it may be possible to subdivide an edge in several ways. Later we will consider the internal and pendant edges of $\tau_{j|R_j}$, and as the next lemma shows the order in which the edge is subdivided when constructing τ_j will not affect our analysis.

Lemma 3.6 *The order in which an edge is subdivided does not affect whether or not leaves from $\mathcal{L}(t_j)$ are on pendant or internal edges of $\tau_{j|R_j}$ in τ_j.*

Proof Suppose that we are given T_1, T_2 and their MAF made up of components $t_1, t_2, \ldots, t_r$. Assume that component t_j has an edge e, that is subdivided in both T_1 and T_2 by an intercomponent edge. We can regard the edges of t_j as a bipartition of the $\mathcal{L}(t_j)$, and so when a new edge is added by subdividing a pre-existing edge the same partition of the original leaf set is still present. For this reason, the order in which the edge is subdivided will not affect whether or not the leaves from $\mathcal{L}(t_j)$ are on pendant edges or internal edges of $\tau_{j|R_j}$ in the tree τ_j. □

As a consequence of Lemma 3.6 τ_j may not be unique. This is because if there is an edge that is

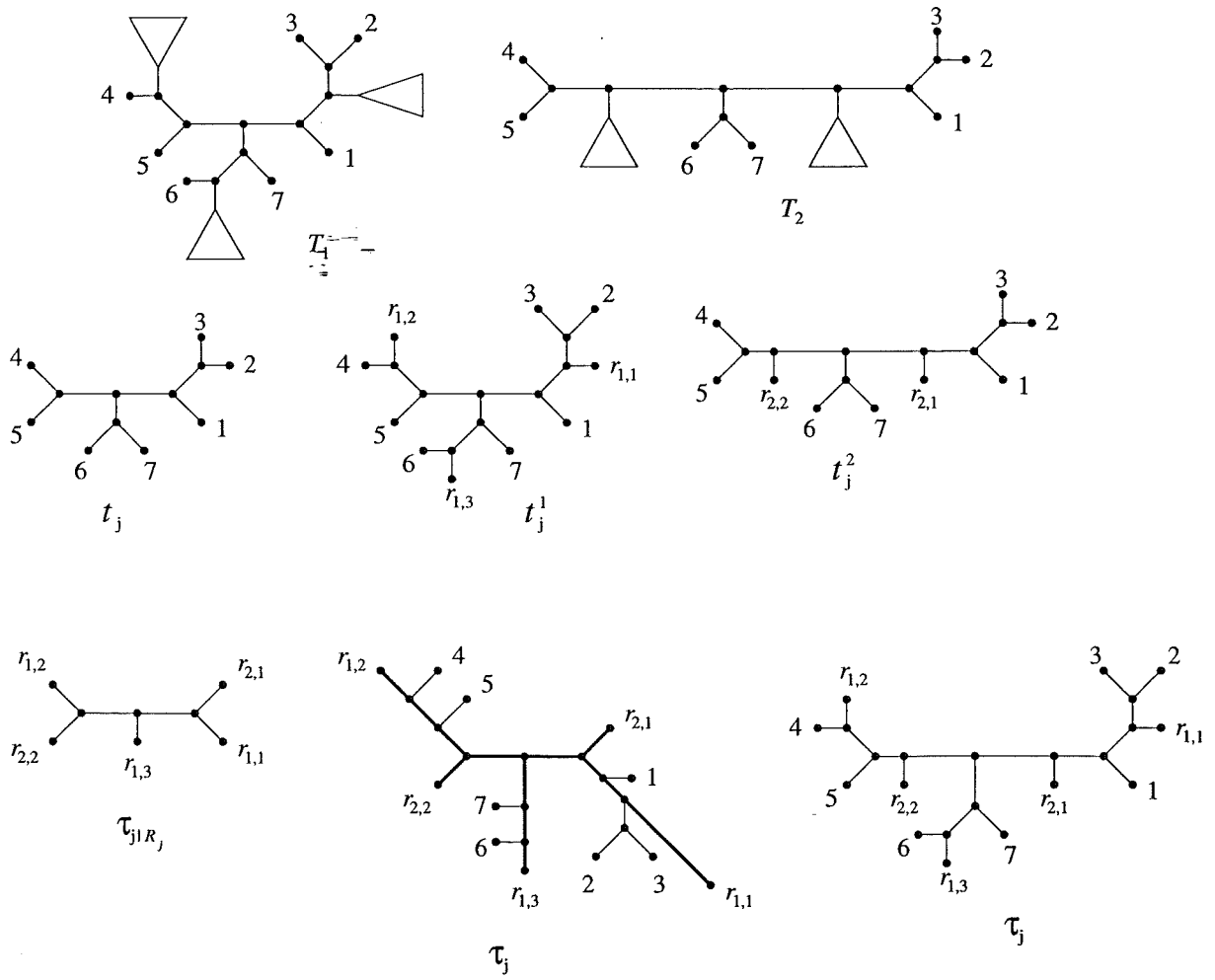


Figure 3.10: Suppose component t_j occurs in a MAF for T_1 and T_2 both in $UB(n)$. The trees t_j^1, t_j^2, τ_j, and $\tau_{j|R_j}$ are all constructed according to Definition 3.4.6. The tree τ_j can also be considered with pendant subtrees along the edges (thick lines) of $\tau_{j|R_j}$.

subdivided by intercomponent edges in both T_1 and T_2, then changing the order in which the edge is subdivided when constructing τ_j will construct a different tree.

Lemma 3.7 *Each of the trees t_j^1, t_j^2, τ_j and $\tau_{j|R_j}$ are binary for all $j \in \{1,\ldots,r\}$.*

Proof Since t_j is binary and all new edges are introduced by subdividing existing edges, all internal vertices of t_j^1, t_j^2, τ_j and $\tau_{j|R_j}$ will have degree three and thus all of these trees will be binary. □

Since $R_j \cap \mathcal{L}(t_j) = \emptyset$, no leaf from t_j appears in the tree $\tau_{j|R_j}$. Furthermore $\tau_{j|R_j}$ is binary with s_j leaves, and therefore will have s_j pendant edges and $s_j - 3$ internal edges. Removing the leaf set restriction can be regarded as adding in pendant subtrees along these edges. We will prove that there can be at most three leaves on an internal edge of $\tau_{j|R_j}$ and at most five leaves on a pendant edge of $\tau_{j|R_j}$. This will enable us to bound the size of components.

Lemma 3.8 *No cherry of $\tau_{j|R_j}$ can be a cherry of τ_j*

Proof Suppose on the contrary, that there is a cherry $\{a, b\}$ of $\tau_{j|R_j}$ that is also a cherry of τ_j. Since the cherry is in $\tau_{j|R_j}$, it must be made up of two intercomponent edges. Therefore there must be a vertex in t_j with at least two intercomponent edges incident to it. However this can not happen as intercomponent edges can not subdivide other intercomponent edges, hence no two can be incident. □

We are almost ready to state our theorem on the upper bound of the size of the leaf set of T_1 and T_2. All that remains is to examine the maximum number of leaves that can be attached to an edge of $\tau_{j|R_j}$ in the tree τ_j. Let us assume for the purposes of Lemmas 3.9 and 3.10, that we have two trees $T_1, T_2 \in UB(n)$ such that $d_\Theta(T_1, T_2) = k$ and a MAF $t_1, \ldots, t_r$, but as yet T_1 and T_2, and hence their components, have not been reduced (if possible) by either Rule 1 or Rule 2.

Lemma 3.9 *For a component t_j in the MAF, there can be at most three leaves in τ_j attached to an internal edge of $\tau_{j|R_j}$ in both t_j^1 and t_j^2, after being reduced by Rule 1 or Rule 2.*

Proof Suppose that for a given component t_j we have constructed τ_j and $\tau_{j|R_j}$. Suppose that $\tau_{j|R_j}$ has an internal edge, e_i to which a chain of subtrees is attached in τ_j. We denote the connected subtree of τ_j between the two vertices of $\tau_{j|R_j}$ adjacent to e_i be Γ. Let P_j be the set over all $u, v \in \mathcal{L}(t_j)$ of the path from u to v. By Lemma 3.8 there is at least one path in P_j that traverses e_i. Thus after the leaves from R_j have been removed at least one pendant leaf will occur at each end of Γ. This ensures that in both t_j^1 and t_j^2 there will always be at least one leaf at each end of Γ. Hence Γ can be reduced to at most three vertices using Rule 2. If Γ contains a single pendant subtree then Rule 1 can reduce it to a single vertex. □

Lemma 3.10 *For a component t_j in the MAF, any pendant subtrees in $\tau_{j|R_j}$ attached to a pendant edge of $\tau_{j|R_j}$ can be reduced to at most five vertices using Rule 1 and Rule 2 in both t_j^1 and t_j^2.*

Proof Suppose that for a given component t_j we have constructed τ_j and $\tau_{j|R_j}$ and that the component has edges that reduce down to a pendant edge, e_p, in $\tau_{j|R_j}$. Since $\tau_{j|R_j}$ is binary by Lemma 3.7 there are three cases to consider.

(i) e_p the only edge of $\tau_{j|R_j}$. This situation occurs if the component only has two intercomponent edges incident to it. In t_j^1 only one intercomponent edge can be incident, and the second intercomponent edge must occur in t_j^2. We can regard all leaves in $\mathcal{L}(t_j)$ to be in a chain, Γ, off e_p, see Figure 3.11. In t_j^1 (or t_j^2 respectively) one end of Γ will not have an intercomponent edge, hence the two pendant subtrees at this end form a cherry that does not exist in the t_j^2 (t_j^1 resp.) This means that we must reduce one pendant subtree at this end of Γ to a leaf using Rule 1. Similarly at the other end of Γ we have a cherry in one tree that is not present in the other which means that one subtree at this end of Γ must be reduced to a leaf. Thus we have one leaf at each end and smaller chain of pendant subtrees that occurs in both t_j^1 and t_j^2, and hence can be replaced by at most three leaves using Rule 2. Hence we have a maximum of five leaves on e_p after reduction by Rules 1 and 2.

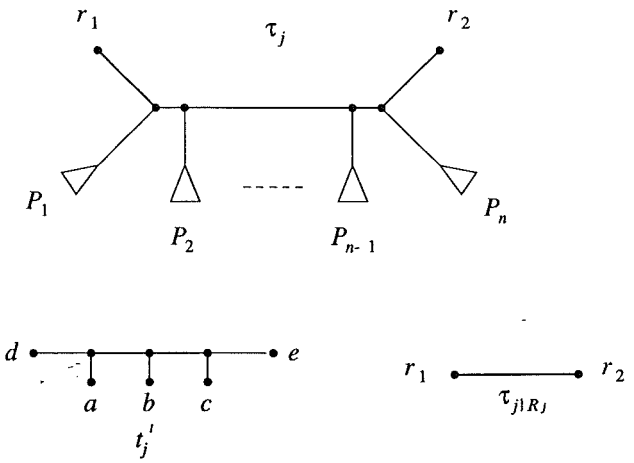


Figure 3.11: Example of case (i). The tree τ_j is constructed from a component t_j, with two incident intercomponent edges consisting of several pendant subtrees $P_1, \ldots, P_n$. $\tau_{j|R_j}$ merely consists of one pendant edge and two leaves, $r_1, r_2 \in R_j$. When reduced, subtree P_1 is reduced to vertex d, while P_n is reduced to vertex e (by Rule 1), the remaining subtrees are reduced to three vertices (Rule 2) to give t_j'

(ii) e_p is only adjacent to two pendant edges in $\tau_{j|R_j}$. Suppose that there are pendant subtrees are on e_p. By Lemma 3.8 no cherry in $\tau_{j|R_j}$ can be a cherry in τ_j also, hence if there is a chain of pendant subtrees along e_p, there must also be a leaf from $\mathcal{L}(t_j)$ on one of the other pendant edges. At the free end of e_p there must be a cherry made up of one pendant subtree from $\mathcal{L}(t_j)$ and a leaf from R_j. Thus in either t_j^1 or t_j^2, this cherry will not appear, and thus the pendant subtree can only be

reduced using Rule 1. All the remaining pendant subtrees can be reduced using Rule 2, hence a maximum of four vertices is needed. See Figure 3.12 for an example of this case.

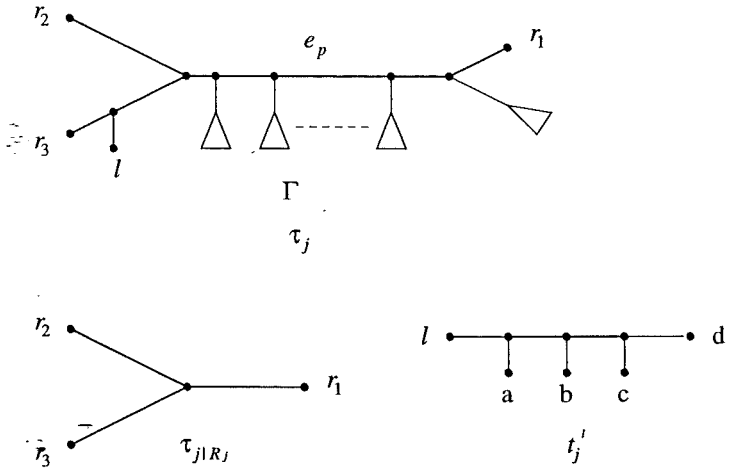


Figure 3.12: Example of case (ii). Suppose we have a component t_j with corresponding τ_j above. The thick lines in τ_j represent edges in $\tau_{j|R_j}$ (also shown) while the thin lines are the remaining edges in τ_j. The existence of leaf $l \in \mathcal{L}(t_j)$ is guaranteed. Leaf $r_1 \in \mathcal{L}(R_j)$ forms a cherry with a pendant subtree. When reduced, and restricted to the leaf set of t_j, the pendant subtree in the cherry becomes a single vertex, d, when the chain of pendant subtrees, Γ is replace by three vertices, a, b and c. The resulting tree is t_j'. Note that edge e_p has contributed four vertices to the reduced tree.

(iii) e_p is adjacent to at least one internal edge in $\tau_{j|R_j}$. If e_p is adjacent to an internal edge then, by Lemma 3.7, there must be at least one leaf that is on the same side of e_p as the internal edge in $\tau_{j|R_j}$. More precisely there must be a path in τ_j from the vertex adjacent to both e_p and the internal edge, to a leaf from the leaf set $\mathcal{L}(t_j)$, which does not include the edge e_p. Thus if we have a chain of subtrees along e_p then when the leaves from leaf set R_j are pruned there will always be at least one leaf at one end of the chain. Hence in t_j^1 and t_j^2 there is at least one leaf at the end of the chain, so we do not have the situation as in case (i) where a cherry is present in one tree but not the other. At the other free end of the chain when the leaf from R_j is pruned from one of either t_j^1 or t_j^2, two pendant subtrees will form a cherry. This means that one pendant subtree can only be reduced to a leaf using Rule 1. The remainder of the chain can be reduced using Rule 1 if a single pendant subtree remains, or Rule 2, otherwise giving a maximum of three leaves and hence a maximum total of four. See Figure 3.13 for an example of this case.

□

Lemmas 3.9 and 3.10 are the driving force behind our result, all that remains to do is state the theorem and tie all the pieces together.

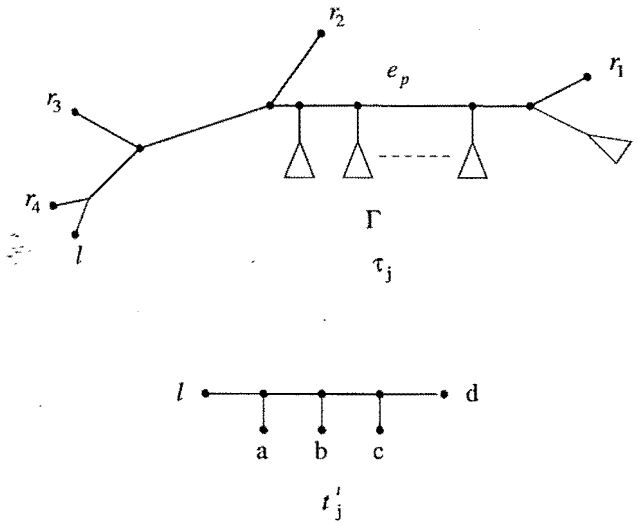


Figure 3.13: Example of case (iii). Suppose we have a component t_j with corresponding τ_j above. The thick lines represent edges in $\tau_{j|R_j}$ while the thin lines are the remaining edges in τ_j. The existence of leaf $l \in \mathcal{L}(t_j)$ is guaranteed. Edge e_p contributes at most four vertices to the reduced tree.

Theorem 3.5 *Let T_1, T_2 be two unrooted binary trees with $d_\Theta(T_1, T_2) = k$ and suppose that T_1, T_2 are reduced as far as possible using Rule 1 and Rule 2. Then $|\mathcal{L}(T_1)| = |\mathcal{L}(T_2)| \leq 23k - 9$.*

Proof By the hypothesis and Equation 2.12 a MAF for T_1 and T_2 has at most $k+1$ components. We assert that if T_1, T_2 have been reduced as far as possible using Rule 1 and Rule 2, then the components must be reduced as far as possible as well. Construct the four trees, t_j^1, t_j^2, τ_j and $\tau_{j|R_j}$ defined earlier. Leaves either find themselves on a pendant edge of $\tau_{j|R_j}$ or on an internal edge of $\tau_{j|R_j}$. Lemma 3.9 ensures that at most three leaves can be on any internal edge of $\tau_{j|R_j}$, while at most five leaves can be on any pendant edge by Lemma 3.10. Hence to establish an upper bound on the number of leaves we can count the number of internal edges and pendant edges of $\tau_{j|R_j}$ for all j. The number of intercomponent edges incident to the component $\tau_{j|R_j}$ is s_j and, by Lemma 3.7, $\tau_{j|R_j}$ is binary for all j, hence by Lemma 1.3 the number of internal edges is $s_j - 3$ and the number of pendant edges is s_j. In fact we need not calculate s_j for each component, we only need to sum over s_j for all j. This is equivalent to summing the number of edges incident to τ_j for all j or the number of edges incident to either t_j^1 or t_j^2. Lemma 3.5 shows that $\Sigma_{j=1}^{k+1} \deg^i(t_j) \leq 2k$ for $i = 1, 2$, which effectively establishes the result. Hence,

$$\begin{aligned} |\mathcal{L}(T_1)| &= |\mathcal{L}(T_2)| \\ &\leq \Sigma_{j=1}^{r}(5(s_j) + 3(s_j - 3)) \\ &= \Sigma_{j=1}^{r}(8s_j - 9) \\ &\leq 8 \times \Sigma_{j=1}^{r} s_j - 9(k+1) \end{aligned}$$

$$\begin{aligned} &= 8 \times \Sigma_{j=1}^{r}(\deg(t_j^1) + \deg(t_j^2)) - 9(k+1) \\ &\leq 16 \times \max_{i=1,2}\{\Sigma_{j=1}^{r} \deg(t_j^i)\} - 9(k+1) \\ &\leq 16(2k) - 9(k+1) \\ &= 23k - 9 \end{aligned}$$

□

3.4.3 Complexity of the Parameterized TBR-distance

Theorem 3.6 *The Parameterized TBR-Distance Problem is fixed-parameter tractable.*

Proof By Lemma 3.1, Rule 1 and Rule 2 can be applied to reduce any two trees from $UB(n)$ in polynomial time, furthermore Theorem 3.3 shows that the reduction preserves the TBR-Distance and Theorem 3.5 shows that the size of leaf set of the reduced trees is bounded by the distance between the trees and not the size of the leaf set of the original two trees. These are sufficient conditions for Parameterized TBR-Distance Problem to be in the class *FPT*. □

The parameter k is the TBR-distance between any two trees from $UB(n)$. Theorem 3.6 shows that, provided the TBR-distance between two trees is sufficiently small we will be able to determine the exact distance in realistic time.

3.4.4 Complexity of the Parameterized SPR-distance

We suspect that the SPR-Distance Problem is *NP*-hard, however this is still unresolved. Furthermore, we can only conjecture that the Parameterized SPR-distance problem is distance preserving. If we could prove Conjecture 3.4, then Theorem 3.5 would give that the Parameterized SPR-Distance Problem is also in *FPT*.

Appendix A

Table of Notation

deg(v) The degree of (number of edges incident to) a vertex v in a graph. Definition 1.1.2.

$UB(n)$ The space of unrooted binary trees on n leaves. Definition 1.1.12.

$\mathcal{L}(T)$ The leaf set of tree T. Definition 1.1.16.

$T(U)$ where $\mathcal{L}(U) \subset \mathcal{L}(T)$. A minimal subtree of T connecting all leaves from U. Definition 1.1.17.

$T_{|U}$ where $\mathcal{L}(U) \subset \mathcal{L}(T)$. The tree obtained from $T(U)$ after forced contractions have been applied. Definition 1.1.17.

NNI Nearest Neighbour Interchange. Definition 2.1.1.

SPR Subtree Prune and Regraft. Definitions 2.2.1 and 2.2.2 .

TBR Tree Bisection and Reconnection. Definition 2.3.1.

$d_\Theta(T_1, T_2)$ where $\Theta \in \{NNI, SPR, TBR\}$. The minimum number of Θ subtree transfer operations required to transform T_1 to T_2. Definition 1.1.13.

$n!!$ (n semi-factorial). Equivalent to $\Pi_{i=0}^{i=\lfloor n/2 \rfloor}(n-2i)$, *ie* $7!! = 1 \times 3 \times 5 \times 7$.

$G_\Theta(n)$ The adjacency graph. Definition 2.6.1

$\Delta(G)$ The diameter of graph G. Definition 2.6.2.

$\mathcal{O}(n^d)$ $f(n)$ is $\mathcal{O}(n^d)$ if $\exists$ constant c such that $|f(n)| \leq c \times n^d \ \forall n$.

$o(n^d)$ $f(n)$ is $o(n^d)$ if

$$\lim_{n\to\infty} f(n)/n^d = 0.$$

AF Agreement Forest. Definition 2.7.1.

MAF Maximum Agreement Forest. Definition 2.7.1.

$m(T_1, T_2)$ The number of edges cut to construct a MAF for T_1 and T_2. Definition 2.7.1.

HGT Horizontal Gene Transfer. Definition 2.9.1.

NP The class of problems which can be solved in non-deterministic polynomial time. Definition 3.2.1

NP-**complete** The class of problems in *NP* that are at least as hard to solve as any other in *NP*. Definition 3.2.2.

NP-**hard** The class of problems that can be Turing reduced to *NP*-complete problems in polynomial time. Definition 3.2.3.

FPT Fixed-Parameter Tractible. Definition 3.4.1.

***abc*-tree** A tree containing leaves a, b and c and respectively adjacent vertices v_a, v_b and v_c such that v_a and v_b are adjacent also, as are v_b and v_c. Definition 3.4.2.

$\deg^i(t_j)$ where t_j is a component in a MAF for $T_1, T_2 \in UB(n)$. This is the number of inter-component edges incident to component t_j in the tree T_i. Definition 3.4.4.

R_j The leaf set of new leaves added to component t_j in a MAF for $T_1, T_2 \in UB(n)$.

$t_j^1, t_j^2, \tau_j, \tau_{j|R_j}$ Trees constructed from a component t_j in a MAF for $T_1, T_2 \in UB(n)$. Definition 3.4.6.

Appendix B

Acknowledgements

I would like to thank Mike Steel for offering me this thesis topic, and for his helpful and enlightening supervision for my thesis. I would also like to acknowledge that this thesis is based on joint work of Mike and mine. I am very grateful to Mike for all the lengths that he has gone to in order to make this a successful (hopefully!) Master's thesis.

Thanks to Mike Fellows of the University of Victoria for help in deciding suitable tree reduction rules, and for introducing and demonstrating the class *FPT*.

Thank you to Charles Semple for useful comments and assistance with editting.

For biological background I thank Hazel Chapman and Jack Heineman of the Plant and Microbial Sciences department at the University of Canterbury.

This research was generously supported by the Marsden Fund (UOC516).

Bibliography

[1] R. Balasubramanian, M. R. Fellows, and V. Raman. An improved fixed-parameter algorithm for vertex cover. *Information Processing Letters*, 65:163–168, 1998.

[2] L. Cai, J. Chen, R. G. Downey, and M. R. Fellows. Advice classes of parameterized tractability. *Annals of Pure and Applied Logic*, 84:119–138, 1997.

[3] R. G. Downey and M. R. Fellows. Concrete complexity analysis: A slice of the future. 1998.

[4] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer Verlag, 1998.

[5] R. G. Downey, M. R. Fellows, and U. Stege. Parameterized complexity: A framework for systematically confronting computational intractibility. 1998.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[7] P. Gilson and G. McFadden. Something borrowed, something green: lateral transfer of chloroplasts by secondary endosymbiosis. *TREE*, 10:12–17, 1995.

[8] J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, 98:185–200, 1990.

[9] J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *Journal of Molecular Evolution*, 36:369–405, 1993.

[10] J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, 71:153–169, 1996.

[11] J. P. Jarvis, J. K. Leudeman, and D. R. Shier. Counterexamples in measuring the distances between binary trees. *Mathematical Social Sciences*, 4:271–274, 1983.

[12] J. P. Jarvis, J. K. Luedeman, and D. R. Shier. Comments on computing the similarity of binary trees. *Journal of Theoretical Biology*, 100:427–433, 1983.

[13] M. Li, J. Tromp, and L. Zhang. On the nearest neighbour interchange distance between evolutionary trees. *Journal of Theoretical Biology*, 182:463–467, 1996.

[14] W. H. Li and D. Graur. *Fundamentals of Molecular Evolution.* Sinauer Associates, Inc., 1991.

[15] D. R. Maddison. The discovery and importance of multiple islands of most-parsimonious trees. *Systematic Zoology*, 43(3):315–328, 1991.

[16] G. W. Moore, M. Goodman, and J. Barnabas. An iterative approach from the standpoint of the additive hypothesis to the dendogram problem posed by molecular data sets. *Journal of Theoretical Biology*, 38:423–457, 1973.

[17] C. H. Padadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[18] R. D. M. Page. On islands of trees and the efficacy of different methods of branch swapping in finding most-parsimonious trees. *Systematic Biology*, 42(2):200–210, 1993.

[19] D. F. Robinson. Comparison of labeled trees with valency three. *Journal of Combinatorial Theory*, 11:105–119, 1971.

[20] E. Schröder. Vier combinatorische probleme. *Z. Math. Phys.*, 15:361–376, 1870.

[21] D. L. Swofford, G. J. Olsen, P. J. Waddell, and D. M. Hillis. *Phylogenetic Inference in Molecular Systematics.* Sinauer Associates, second edition, 1996.

[22] M. S. Waterman and T. F. Smith. On the similarity of dendrograms. *Journal of Theoretical Biology*, 73:789–800, 1978.