Syst. Biol. 0(0):1–10, 2014
© The Author(s) 2014. Published by Oxford University Press, on behalf of the Society of Systematic Biologists. All rights reserved.
For Permissions, please email: journals.permissions@oup.com
DOI:10.1093/sysbio/syu065

Towards Improving Searches for Optimal Phylogenies

Eric Ford[1], Katherine St. John[1,2,*], and Ward C. Wheeler[3]

[1]*Department of Computer Science, Graduate Center, CUNY, New York, NY, 10016, USA;* [2]*Department of Mathematics and Computer Science, Lehman College, CUNY, Bronx, NY, 10468, USA; and* [3]*Division of Invertebrate Zoology, American Museum of Natural History, New York, NY, 10024, USA;*
Correspondence to be sent to: Department of Mathematics and Computer Science, Lehman College, City University of New York, Bronx, NY, 10468, USA; E-mail: stjohn@lehman.cuny.edu.

Received 5 March 2013; reviews returned 19 August 2014; accepted 20 August 2014
Associate Editor: Lars Jermijn

Abstract.—Finding the optimal evolutionary history for a set of taxa is a challenging computational problem, even when restricting possible solutions to be "tree-like" and focusing on the maximum-parsimony optimality criterion. This has led to much work on using heuristic tree searches to find approximate solutions. We present an approach for finding exact optimal solutions that employs and complements the current heuristic methods for finding optimal trees. Given a set of taxa and a set of aligned sequences of characters, there may be subsets of characters that are compatible, and for each such subset there is an associated (possibly partially resolved) phylogeny with edges corresponding to each character state change. These perfect phylogenies serve as anchor trees for our constrained search space. We show that, for sequences with compatible sites, the parsimony score of any tree T is at least the parsimony score of the anchor trees plus the number of inferred changes between T and the anchor trees. As the maximum-parsimony optimality score is additive, the sum of the lower bounds on compatible character partitions provides a lower bound on the complete alignment of characters. This yields a region in the space of trees within which the best tree is guaranteed to be found; limiting the search for the optimal tree to this region can significantly reduce the number of trees that must be examined in a search of the space of trees. We analyze this method empirically using four different biological data sets as well as surveying 400 data sets from the TreeBASE repository, demonstrating the effectiveness of our technique in reducing the number of steps in exact heuristic searches for trees under the maximum-parsimony optimality criterion. [Character compatibility, exact search, maximum-parsimony optimality criterion, phylogenetic islands, tree search]

Phylogenetic trees represent the evolutionary relationships among taxa, one of the primary endeavors of modern biology. Building on the principle of Occam's Razor, Cavalli-Sforza and Edwards (1967), Farris et al. (1970) and Fitch (1971) proposed choosing the "most parsimonious" tree: the one with the least amount of evolutionary change across the edges. Foulds and Graham (1982) showed that the problem of finding the most parsimonious tree for a given set of taxa—that is, determining the topology of the tree as well as the positions of the taxa on the leaves—is a computationally hard problem. When the given characters are compatible (that is, can be fitted onto the same tree witout a need to infer additional substitutions at either site), the resulting tree is optimal (often called "perfect"), and can be quickly computed (Gusfield 1991).

The naive approach to finding the exact optimal tree for a given data set is to enumerate all possible tree topologies, calculate the parsimony score of each of those trees, and choose the tree with the best score. Since the number of tree topologies grows exponentially in the number of taxa ($(2n-5)!!$ possible trees for n taxa—Schröder 1870; Robinson 1971), this is not possible for most data sets. Hendy and Penny (1982) and Penny and Hendy (1987) developed a *branch-and-bound* algorithm (Land and Doig 1960) to find the most parsimonious tree. This method builds and calculates the parsimony score of a tree to determine an upper bound, and then backtracks on the build process, creating new trees by adding in taxa one by one. If the insertion of a new taxon yields a tree with a score greater than the bound, all trees that extend that tree will also have scores greater than the bound and therefore need not be explored. This pruning of a branch of the search can reduce the number of trees that need to be examined by a constant factor (Land and Doig 1960; Hendy and Penny 1982), but may still necessitate much of the search space being explored.

Due to the difficulty of finding the optimal tree(s) under the maximum-parsimony criterion, a great deal of work has been dedicated to tree search, and—because of the size of the search space—specifically to the heuristic search (e.g., Goloboff et al. 2008; Varón et al. 2010). Most of the heuristic search software follows a local search (or hill-climbing) paradigm: At each step, choose a neighboring tree (usually the best scoring) and repeat until an optimum (local or global) has been found or time has been exhausted. These search programs can be affected greatly by the starting point and choice of neighbors (Charleston 1995; Kirkup and Kim 2000, unpublished data). Since the "ruggedness" of the space is not known in general (Bastert et al. 2002), it is possible to have many local optima, stopping the search prematurely. Multiple starting points and sophisticated search techniques have been employed to make the most of the small sampling possible of this incredibly large space (Goloboff et al. 2008; Varón et al. 2010).

Finding a global lower bound on the parsimony score with respect to a given character sequence has been studied. A simple lower bound is the sum of the number of character states witnessed across all characters. Hendy et al. (1980) improved this by grouping characters into small incompatible sets, each of whose lower bound can be calculated quickly. They showed that the sum of

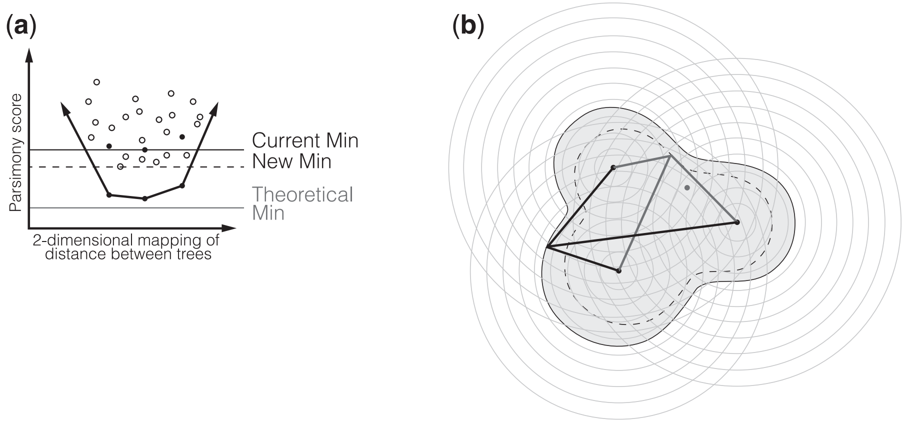

FIGURE 1. In a), the lower bound is graphed. The lower set of solid dots are the parsimony scores on the compatible subsets for each of three anchor trees, and the upper solid dots are the parsimony scores for each of those trees on the complete character set, C. The empty dots are other possible trees on the same set of labeled taxa. Initially, the count of the trees that need to be considered falls below the solid black line. Once a tree with a lower parsimony score is found, the bound can be lowered, as with the dashed line. In b), the black lines are the summed distance from the anchor trees to the original best-scoring tree and the original space that must be exhaustively searched is in grey. Once a tree with a lower parsimony score is found, the summed distance is reduced. The grey lines are the summed distance to the new best-scoring tree, and the decreased search area is bounded by the dashed rule.

these lower bounds gives a lower bound on the overall parsimony score. Holland et al. (2005) used the partition theorem of Hendy et al. (1980) to develop a bound based on pairs of characters. The elegance of their approach is that it avoids the search for incompatible pairs and shows that the average over all pairs gives a lower bound.

In this article, we present an approach that follows the insightful work of Maddison (1991) on "phylogenetic islands," or connected regions of the space of trees that contain optimal or nearly optimal trees (see Fig. 1). We quantify where these regions will occur by exploiting the underlying mathematical structure of the space of trees. In the spirit of the work of Hendy and Penny (1982), we seek to bound the size of the space of trees that needs to be searched. Instead of a global lower bound, we develop a bound on all trees based on their distance to a set of "anchor trees," or perfect phylogenies, that are compatible with subsets of characters. If the input character sequences are compatible, then, as noted above, the exact optimal tree can be found quickly (Gusfield 1991). By carefully defining "neighbors," (see Background section for details) we can show that trees that are neighbors of the optimal tree have a parsimony score that is at least one more than the optimal tree, trees that are second neighbors of the optimal tree have a parsimony score that is at least two more than the optimal tree, and, similarly, trees that are j^{th} neighbors of the optimal tree have a parsimony score that is at least j more than the optimal tree. That is, as we move farther from the optimal tree, we prove mathematically that the parsimony score must increase. As characters from biological data are rarely completely compatible, the above will not work directly for most data, but serves as the first step in the approach. Since the maximum-parsimony criterion is additive, we can calculate the parsimony score over each character separately and sum those parsimony scores to yield the parsimony score for all the characters. We partition the character sequences into subsets of compatible characters, determine the associated anchor tree for each, and apply the lower bound technique to each subset. The sum of the bounds of the compatible subsets yields an overall lower bound for every tree in the space. As in branch-and-bound algorithms, we exclude all trees that have a bound above our current best tree parsimony score. If we move too many steps away, the parsimony score must be worse than our current best tree parsimony score, thus we only need to examine trees that are within a fixed summed distance from the anchor trees (see Fig. 1). We note that the search space, in general, is smaller if the sequence of characters is partitioned into fewer compatible subsets (and thus has a lower number of anchor trees). However, the proof of correctness for our approach does not depend on minimality, only on the fact that the partitioned sets contain compatible characters. With this in mind, we construct the compatible sets using a greedy algorithm. We also note that it is possible to find the exact minimal number of anchor trees via recent work of Gysel and Gusfield (2010), which will limit the search space further, but at the cost of additional time spent in choosing the anchor trees. Although our approach might still yield a large space to examine, we show that it is in practice often much smaller than the original space, and in addition, seems correlated to the consistency index of the best scoring anchor tree. This suggests that the consistency index can be used as a filter to limit the likely starting places for effective heuristic searches. Our approach is implemented using standard software packages, complements heuristic search, and can also be used as a filter to improve heuristic search.

Background

In this section, we briefly introduce the notation and terminology used in this article, beginning with phylogenetic trees and the maximum-parsimony optimality criterion, and formally defining the space of trees—For more thorough treatment of these topics, see Semple and Steel (2003).

Trees and Characters

Following Semple and Steel (2003), a *tree* $T=(V,E)$ is a connected graph with no cycles (i.e., between any two vertices, $v_1,v_2 \in V$, there is exactly one path of edges between them). An *X-tree* is a tree in which some of the vertices (including all *leaves* (vertices of degree one) as well as vertices of degree two) are labeled by disjoint subsets of X. The set X is most often a set of species names with one name assigned to each leaf of the tree. The definition is more general to allow for more complicated evolutionary scenarios to be represented. A *character on* X is a function from a non-empty $X' \subseteq X$ into a set C of character states.

In a *perfect phylogeny*, each interior edge (edge not subtending a leaf) corresponds to one or more binary characters in the character set, and the edges sort the vertices by character (Gusfield 1991). This can be viewed as each binary character *defining* some edge in the tree. The definition can be expanded to the case of non-binary characters, in which case each character corresponds to multiple edges, with each r-state character (for $r>1$) defining $r-1$ edges in the tree and the edges sorting vertices by character *states*, rather than by characters, as in the case of binary characters (this correspondence is not always unique, see §4.2.4 in Semple and Steel (2003)).

On a set of characters and taxa for which no perfect phylogeny exists, there may be subsets of the characters that are compatible. Each of these subsets defines a perfect phylogeny that is an unresolved tree on the taxa. To avoid confusion between perfect phylogenies for the entire set and perfect phylogenies for the compatible subsets, we will call the trees thus derived *anchor trees* and their edges will be referred to as *anchor edges*.

Maximum-Parsimony Optimality Criterion

The maximum-parsimony criterion seeks the tree with the minimal number of changes needed to explain a given character sequence. Formally:

Definition 1 (Semple and Steel 2003): For a graph $G=(V,E)$ and a function f on V, the changing set of f is the subset $\mathrm{Ch}(f)=\{\{u,v\}\in E: f(u)\neq f(v)\}$ of the edges of G. The changing number of f, denoted $\mathrm{ch}(f)$, is the cardinality of $\mathrm{Ch}(f)$.

Definition 2 (Semple and Steel 2003): Let $\chi: X' \to C$ be a character on X and let T be an X-tree. An extension of χ to T is a function $\bar{\chi}: V(T)\to C$, which is identical to χ on X'. The parsimony score, $l(\chi,T)$ of χ on T, is the minimum value of $\mathrm{ch}(\bar{\chi})$ over all extensions $\bar{\chi}$ of χ to T.

Informally, if we have a character assigned to leaves of a tree, we can extend that character to label the internal nodes of the tree. If that extension minimizes the changing number of the initial character, then it is called a minimum extension. Using the above notation, the parsimony score for sequence of character, $C=(\chi_1,\chi_2,\ldots,\chi_k)$ is

$$l(C,T)=\sum_{i=1}^{k} l(\chi_i,T)$$

and is often called the *tree length* of T with respect to the character sequence C. We note that Fitch (1971) gave an efficient algorithm for calculating minimum extension on nonadditive characters, and thus tree lengths. A common measure of how well a tree explains a character sequence is the consistency index:

Definition 3 (Kluge and Farris 1969): The consistency index (CI) of a set of characters and a tree is the ratio m/s, where m is the minimal score of the characters possible on any tree and s is the actual score of the tree.

Space of Phylogenetic Trees

For a given number n of leaves, the number of possible trees is large (as noted above, it is $(2n-5)!!$). A standard way to organize these huge sets of trees, $\mathbb{T}_n$, is via a distance metric that classifies how close any two trees are (see Semple and Steel 2003 for detailed treatment). If the proposed distance, d, is well behaved (i.e., the distance is always nonnegative, it is zero when the trees are equal, it is symmetric, and the triangle inequality holds), then the resulting space $(\mathbb{T}_n,d)$ is a *metric space*. The organization of the space is highly dependent on the choice of the distance. Common metrics for phylogenetic trees include Nearest Neighbor Interchange (NNI) (Robinson 1971), Subtree Prune and Regraft (SPR) (Hein 1990), Tree Bisection and Reconnection (TBR) (Swofford 1990; Farris 1988), and Robinson–Foulds (RF) (Robinson and Foulds 1981). The first three metrics are defined in terms of the minimal number of the specified operation that must be performed to transform one tree into the other, and are hard to compute (Li et al. 1996; Allen and Steel 2001; Bordewich and Semple 2005; Bonet and St. John 2010). The last metric can be computed efficiently (Day 1985). The RF distance is a metric that measures the dissimilarity between two trees by determining the number of edges by which the two trees differ. Formally,

Definition 4 Given two trees $T_1,T_2\in\mathbb{T}_n$, the RF distance, $d_{RF}(T_1,T_2)$, is the minimum number of contractions and resolutions necessary to convert T_1 to T_2.

To find the optimal maximum-parsimony tree, we use a relaxed version of the RF distance, whose neighborhoods have nice properties with respect to parsimony scores.

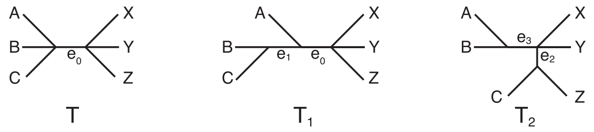

FIGURE 2. Relaxed RF distance, d_{rRF}: Given that T is the original derived tree, a resolution that is still compatible, such as the addition of edge e_1 to get T_1, does not count toward the distance, whereas the addition of a noncompatible edge, such as e_2 does. Thus $d_{rRF}(T,T_1)=0$, whereas $d_{rRF}(T,T_2)=1$. Note here that e_0 in T seems to be the same edge as e_3 in T_2, but because they are splitting T and T_2 differently, they are in fact different edges.

Taxon	Character
A	0
B	0
C	0
X	1
Y	1
Z	1

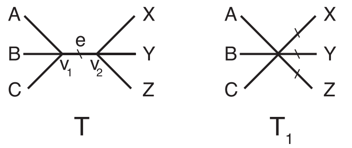

FIGURE 3. Upper bound on d_{rRF}: The score on T is 1. After e is removed, the difference in scores between T and T_1 is 2, which is the lesser of p, the number of parent nodes of v_1, and h, the number of child nodes of v_2, (in this case, both p and h are 3), minus 1.

Definition 5 Given two trees $T_1,T_2 \in \mathbb{T}_n$, the relaxed RF distance, $d_{rRF}(T_1,T_2)$, is the number of edges in T_2 that are incompatible with the tree T_1.

We use this modified version of the RF distance, d_{rRF}, to define neighborhoods around anchor trees. For fully resolved (binary) trees $d_{RF}=d_{rRF}$ and deleted edges are counted identically, but for unresolved trees, only the additions of noncompatible edges are counted toward the distance. As an example, given an initial derived tree, T, with edges E, any further resolutions that are done on T to yield T_1 are not counted in $d(T,T_1)$. This is the case even if T_1 is fully resolved. However, if any edge e_j is added to T, but is incompatible with E, so that an edge e_i must be removed from E before e_j can be added, then e_j is counted to the distance (see Figs. 2 and 3). We note that our relaxed distance is not a metric, as it is not symmetric, and there is no guarantee that the triangle inequality will hold.

MATERIALS AND METHODS

We implemented our overall approach using the off-the-shelf software package POY (Varón et al. 2010) and Python scripts for processing data. Our approach consists of four steps:

1. partition the input sequence of characters into compatible subsets of characters;
2. construct and score the associated anchor tree for each compatible subset found;
3. calculate the starting configuration of the constrained search space; and
4. exhaustively search the constrained space, dynamically updating the remaining space to be searched.

These four steps are describe in detail below.

Partitioning into Sets of Compatible Characters

For each data set, we partitioned the sequence of characters into compatible subsets. Determining whether a set of binary characters is compatible can be accomplished in linear time (Gusfield 1991), while doing so on nonbinary data is NP-Hard (Bodlaender et al. 1992; Steel 1992), although a polynomial-time algorithm has been found for data with a fixed number of states (Agarwala and Fernandez-Baca 1994). As such, there are a number of equally valid heuristic methods to solve this problem. We used two methods to discover sets of compatible characters. On data sets with no initial tree, we used a simple greedy algorithm (see the discussion of vertex cover in Cormen et al. 2001) to discover sets of compatible characters. For data sets that contained a tree for the character sequences, an iterative process was used. At each step, an anchor tree was found. That anchor tree was then collapsed by removing any edges not associated with a character in the subset. This (possibly partially resolved) anchor tree was saved and all characters that were compatible with it were removed from the set of characters to be processed. The initial tree was used as the original anchor tree. Subsequent anchor trees were discovered by successive searches using POY (Varón et al. 2010). We used POY's "`search()`" operation, which builds a set of Wagner trees, runs TBR swaps, perturbs using ratchet, and fuses the resulting trees. At each step the search time was doubled, until a maximum of 4 hours was reached. The searches continued until either there were no characters left in the superset or two successive searches were unable to find a tree with any compatible characters. Once the end case was reached the remaining characters were checked for pairwise compatibility. Additional anchor trees were built using both the pairwise-compatible characters and finally the remaining singletons.

Scoring Anchor Trees

For large sets of characters and small sets of taxa, it is possible that there exists a subset of compatible characters large enough that every interior edge of a binary tree is associated with some character, but in general this is not the case. Instead, each of these sets will define a tree that is not completely resolved, hence a nonbinary tree. Nonetheless, each of these anchor trees will have a minimal parsimony score on that subset.

Recall that the diameter of our eventual search space is the difference between the theoretical minimum score on the data and the current best score. At each step, therefore, we scored the anchor trees on the complete character sequence. For fully resolved trees, it is easy to determine the parsimony score (Fitch 1971). However, the stored anchor trees are unresolved, and for unresolved trees, a tree resolution must be found. Finding this resolution is essentially the search problem on a slightly smaller space of trees, and thus computationally difficult (Bonet et al. 1998; Wu et al. 2009). We therefore used POY to do constrained Wagner builds on the data, to get "good enough" scores for the anchor trees. POY first read in the character set and the anchor tree, which was used as a constraint, using the `build(constraint:"treefile")` command. Then a constrained swap was done and the best trees selected (`swap(constraint:(depth:0, file:"treefile")) select()`). In the case that multiple trees were reported, the first reported score was used.

Determining the Constrained Search Space

The next step in our approach is to constrain the space of all trees to regions that are guaranteed to contain the optimal tree. Intuitively, these regions are the quantification of Maddison's 1991 phylogenetic islands with the constrained search space consisting of the "islands" and the nearly optimal trees near them. To determine the constrained search space, we used the partition of the character sequence into compatible sets and the associated anchor trees as well as the static bound on parsimony to calculate the summed diameter of the constrained space. Any tree whose summed relaxed RF distance to the anchor trees is less than the summed diameter is part of the constrained search space. The static lower bound was calculated as the sum of the number of character state changes across all characters. These simple calculations were performed by a script written in Python. We outline the formulas used for calculating the summed diameter below.

For each compatible subset of characters, S_c, we can bound the parsimony score of trees in the space by their relaxed RF distance to the associated anchor tree T_c for S_c, using the following theorem:

Theorem 1 *Let C be a compatible sequence of characters and let T_C be the associated anchor tree. Then, for any tree T leaf labeled by C,*

$$l(C,T) \geq l(C,T_C) + d_{rRF}(T_C,T)$$

(Proof can be found in the appendix.) Since the parsimony score is additive (i.e. $l(C,T) = \sum_{c=1}^{m} l(S_i,T)$), it follows immediately that:

Corollay 1 Let C be a sequence of characters. Let $C_1, C_2, \ldots, C_m$ be a partitioning of C into compatible subsets of characters. For each $i = 1, \ldots, m$, let T_i be the anchor tree associated with C_i. Then for any tree T leaf labeled by S,

$$l(C,T) \geq \sum_{i=1}^{m} l(C,T_i) + d_{rRF}(T_i,T)$$

To calculate *Diam*, the bound for the sums that determines which trees are part of the constrained search space, we first calculate the parsimony score on each anchor tree, $l(T_i,S)$ as well as the static lower bound for the space, M. The diameter is defined as:

$$Diam = \min_{i=1,\ldots,m} l(T_i,S) - M$$

Searching the Constrained Search Space

As no software currently exists that allows for searching only in spaces constrained by d_{rRF}, we instead built trees using the anchor edges. We generated all possible combinations of anchor edges, where for any given tree T_a, the number of anchor edges is bounded below by the total number of anchor edges less the difference between the current best score and the theoretical minimum score. This set of trees could then be used as constraints (hereafter *constraint trees*), for a subsequent resolution. All resolutions of the constraint trees were discovered and scored.

Materials

To illustrate the merits of our new approach to exploring tree space, we analyzed four data sets: *Metasiro americanus* and three sets from TreeBASE, *Rhexocercosporidium* spp., *Armillaria*, and *Adansonia*. In addition, we surveyed 400 treeBase data sets.

Metasiro.—We ran our initial trials on a phylogeographic data set from *M. americanus* (Opiliones: Neogoveidae) (Clouse and Wheeler 2014), a species of harvestman native to Florida. Harvestmen are found worldwide, and are useful in studies of phylogeography because they are highly localized, with limited dispersal abilities.

The data comprised 62 taxa, with 460 aligned base pairs. Of those 460 aligned characters, 418 were constant, whereas an additional six were autapomorphic, hence compatible with all trees. That left 36 informative characters, so the final tree could not be fully resolved (as there are $n-3=59$ internal edges in the fully resolved tree, and each edge would need to be supported by at least one character). Of the 36 informative characters, 13 were completely compatible—that is, compatible with all other characters, and the remaining 23 characters broke down into two mutually incompatible sets of compatible characters, with respectively eight and fifteen characters, giving us just two perfect phylogenies, and therefore two anchor trees.

TreeBASE.—We also conducted trials on data sets downloaded from TreeBASE (Sanderson et al. 1994).

After a search on TreeBASE for the keyword parsimony, 666 data sets were found and downloaded. We parsed the files, focusing on those where a tree could be automatically associated with a sequence of nucleotides. This was not possible for 266 of the data sets, with the most common issues being multiple or empty taxa blocks, and non-DNA character sequences. For the remaining 400 files, the constant and singleton characters were discarded. Using the trees provided with the data sets, the consistency indices of the sets were found. Finally, for the three data sets with the highest consistency indices, an exhaustive search was run. Those three data sets are:

Armillaria, TreeBASE ID S899 *Armillaria.* is a genus of parasitic fungus of woody plants and causes root rot. The study of Coetzee et al. (2003) compared previously unidentified isolates from South America and Southeast Asia, locations from which it has been little studied. They ran a phylogenetic analysis on 14 isolates of the *Armillaria* using the concatenation of ITS1, including 5.8S, ITS2, and the first intergenic spacer region (IGS-1). The concatenated characters consisted of 559 bases, of which 176 were parsimoniously informative.

Rhexocercosporidium, TreeBASE ID S1786. *Rhexocercosporidium panacis* sp. nov. was a previously undescribed species that causes rusted root of ginseng. Several phylogenetic analyses were done to determine its similarity to the only previously described species of the genus. Reeleder (2007) ran a phylogenetic analysis on nine isolates of the fungus genus *Rhexocercosporidium* and one outgroup, *Phialophora gregata*. The character data consisted of the concatenation of two sequence data sets, the first was an internal transcribed spacer region of ribosomal DNA, ITS1, and ITS2, including the 5.8S gene, the second was a portion of the β-tubulin gene. The concatenated characters consisted of 716 bases, of which 47 were parsimoniously informative.

Adansonia, TreeBASE ID S376. Baum et al. (1998) conducted several phylogenetic analyses, including both parsimony and maximum likelihood analyses, to determine the biogeography and floral evolution of *Adansonia*, which are baobabs. The taxa consisted of multiple specimens of the eight species in *Adansonia*, plus three outgroup taxa from the clade *Adansonieae*, giving a total of 18 taxa. Various character sets were used. Of those sets, the first to appear in the downloaded Nexus file was the chloroplast *rpl*16 intron, consisting of 1350 bases, of which only 17 were parsimoniously informative. This set and its associated tree were used for our analysis.

Results

We present results for our search on the four data sets, as well as our survey of the TreeBASE files.

Metasiro

These data are well suited to our initial investigation because they reduce to binary data. That is, although they are DNA (and therefore 4-state) data, at any given locus only two nucleotides are observed across the data, so they can be treated as binary.

The *Metasiro* data broke down into two sets of compatible characters, of sizes 15 and 8 characters. To each of these subsets was added the subset of completely compatible characters, of size 13. The expected number of anchor edges was therefore $15+8+13=36$, but due to characters in each set that were identical, there were a total of only 15 anchor edges. Each of the two subsets provided an anchor tree, with parsimony scores of 46 and 49, respectively. As there were 36 informative, two-state, characters, the minimum possible score, M_t, for this data set was 36. Therefore, the only trees that needed to be enumerated to find the optima are trees T_x that lay within $\sum_a d_{rRF}(T_x,T_a) \leq 8\ (min(s_C(T_l)) - M_t)$ where T_a is an anchor tree. As there were a total of 15 anchor edges, each tree to be examined needed to have at least seven anchor edges before resolutions, for a total of $\sum_{k=7}^{15}\binom{15}{k}=22818$ unresolved trees. As the characters were binary, the anchor edges were entirely incompatible, so the vast majority of those trees could not be built. In fact, there were a total of 48 unresolved trees, all of whose resolutions needed to be explored in an exhaustive search. In contrast, for fully resolved trees with 15 internal edges there are $(2x18-5)!!=1.92x10^{17}$ (as there are $(2n-5)!!$ trees on n taxa, and $n-3$ internal edges in a fully resolved tree), all of which would need to be visited for an exact solution if our algorithm were not used. In the end, one of the two anchor trees—T_1, which was defined by the larger compatible subset—was one of the 20 optimal trees (see Fig. 4).

Armillaria

Information on this data set, which was downloaded from TreeBASE, is represented in line one of Table 1. There were 2 anchor trees and a total of 12 anchor edges. The total number of unresolved constraint trees that could be built was 473. As many of the constraint trees were highly unresolved, there were a total of 1,905,291 trees to score. Of these, there were only nine optimal trees, including the single tree reported by the authors.

Rhexocercosporidium

Information on this data set, which was downloaded from TreeBASE, is represented in line two of Table 1. There were three anchor trees and a total of six anchor edges. The total number of unresolved constraint trees that could be built was three, with 405 total possible resolutions. These 405 trees were scored, and 45 optimal trees were found. Reeleder (2007) found a single optimal tree for this data, which was given in the downloaded Nexus file, and that tree was among the 45 optimal trees.

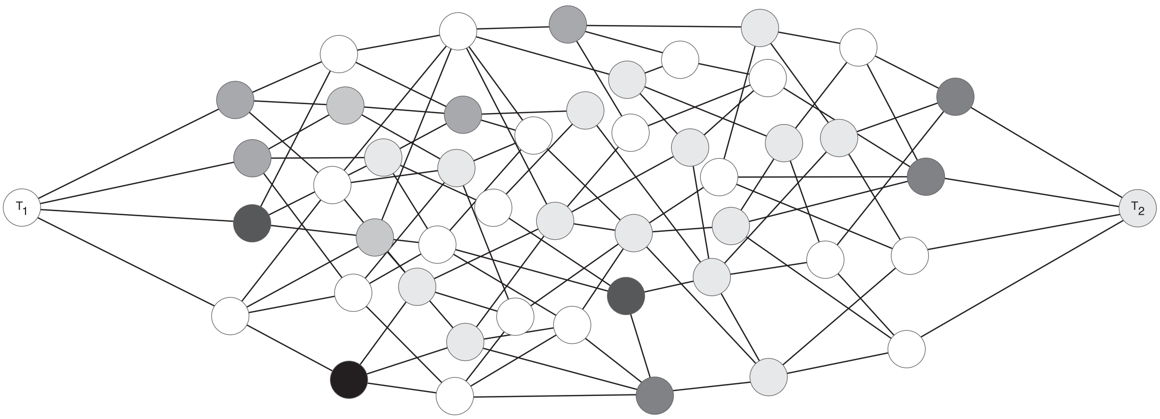

FIGURE 4. Walking between the anchor trees on *Metasiro*. T_1, which is an optimal tree, has 11 internal edges, and T_2, which is not optimal, has 10. These sets of edges intersect, giving a total of 15 anchor edges. $min(s_C(T_l))-M_t=8$. Thus, there are $\sum_{k=7}^{15}\binom{15}{k}=22818$ unresolved trees to visit. However, the anchor edges are highly incompatible, thus supporting conflicting clades that cannot co-exist in a single tree, so only 48 of those trees can be built. The trees were scored on the entire character set, and better scores are lighter.

TABLE 1. The reduction in size of the search space is a function of both $min(s_C(T_l))-M_t$ and the number of anchor edges discovered.

CI	Number of taxa	Actual size of tree space	$min(s_C(T_l))-M_t$	Number of anchor trees found	Size of reduced search space	Log reduction
0.97	14	7.91E+12	6	2	1.69E+06	−6.67
0.96	10	3.44E+07	2	3	4.05E+02	−4.93
0.95	18	6.33E+18	1	2	1.22E+03	−15.72
0.95	6	9.45E+2	5	2	n/a	n/a
0.95	32	1.78E+42	2	2	4.98E+15	−26.55
0.93	39	1.31E+55	4	4	1.35E+39	−15.99
0.93	40	1.01E+57	18	2	4.91E+37	−19.31
0.92	11	6.55E+08	24	2	n/a	n/a
0.92	16	6.19E+15	9	2	1.35E+13	−2.66
0.91	15	2.13E+14	16	2	n/a	n/a
0.91	41	7.98E+58	6	2	2.39E+34	−24.52
0.89	13	3.16E+11	3	2	7.42E+05	−5.63
0.88	29	8.69E+36	60	2	n/a	n/a
0.88	15	2.13E+14	20	2	n/a	n/a
0.87	25	1.19E+30	3	3	9.36E+15	−14.11
0.87	19	2.22E+20	17	2	5.31E+17	−2.62
0.87	19	2.22E+20	8	2	2.09E+09	−11.02
0.86	20	8.20E+21	17	2	n/a	n/a
0.86	55	3.19E+86	28	2	5.79E+81	−4.74
0.86	41	7.98E+58	8	2	9.99E+43	−14.90

Note: Some data sets could not be reduced using our method, as $|E| < \min(s_C(T_l))-M_t$. In that situation, insufficient anchor edges are discovered to constrain the search space.

Adansonia

Information on this data set, which was downloaded from TreeBASE, is represented in line three of Table 1. There were two anchor trees and a total of eleven anchor edges. The total number of unresolved constraint trees that could be built was only one. That tree had 2835 possible resolutions, each of which was scored. Four hundred and five of the resolved trees were optimal. The original paper included five trees, none of which was fully resolved. The most optimal resolutions of these five trees are among our results, but many resolutions are nonoptimal, making comparisons between our results and theirs futile.

TreeBASE Survey

As noted, we computed CIs for 400 TreeBASE data sets. In Table 1, we report the 20 data sets with highest CI scores. Of these, we chose the three sets with the smallest subsequent search spaces and did exhaustive walks of all necessary trees. In two cases the authors reported a single optimal tree, and we found that the reported trees were members of the sets of optimal trees. That is, in these cases the authors had reported one optimal tree among many. For the third data set, no fully resolved—and thus no optimal—trees were given.

We were able to significantly reduce the amount of the tree space necessary to search exhaustively (see

Table 1). In addition, for each of those high-CI data sets, we were able to show that the trees given in TreeBASE were, in fact, optimal trees. This follows, as the CIs were computed using the trees given in TreeBASE, which were the best found for their respective data sets.

Discussion

We provide a new approach for finding optimal phylogenetic tree(s) under the maximum-parsimony criterion. Extending the work of Maddison (1991), we quantify where "phylogenetic islands" of optimal trees can occur by refining the static lower bounds determined by Hendy et al. (1980) to new lower bounds that change across the tree space. As better potential solutions are found, the constrained search space dynamically shrinks, as in the classic branch-and-bound algorithms of Hendy and Penny (1982) and Penny and Hendy (1987). To determine the constrained search space, we first partition the input characters into compatible subsets, which induce "anchor trees" that guide the search.

Our method has the advantage that it reports all optimal trees for a given data set. In contrast to methods that report only a single tree, even if many optimal trees exist, it gives a clearer view of the phylogenetic space being described. Reporting only a single tree out of several possibilities can mislead the reader, or even the researcher, giving false confidence in the results of an analysis. Our method helps to avoid this fate. A researcher presented with 405 optimal trees (as with the *Adansonia* data, above) will have different views than one presented with a single optimal tree and no estimate of the actual number of optimal trees that exist.

As noted, we ran complete searches on the three TreeBASE data sets with the highest CIs. Using our method, we were able to run a complete search with a guaranteed optimal result on all three sets in just a few hours of time on a multicore machine, including an exhaustive search of the areas in which we can guarantee the optimal trees will be found. Prior to this, only an exhaustive search or a branch-and-bound approach would guarantee that an optimal tree would be found. For the smallest of these sets, *Rhexocercosporidium*, TreeBASE ID S1786, with only ten taxa, an exhaustive search of the entire space would certainly have been possible, but exhaustive searches would have been significantly more time consuming for *Armillaria*, TreeBASE ID S899, with 14 taxa, and especially for *Adansonia*, TreeBASE ID S376, with 18 taxa. As one can see from examining Table 1, even for data sets with CIs around 0.85, the size of the space that one would have to search to be sure that the true optimal tree had been found can be significantly reduced, moving some sets of data with many taxa into the realm of guaranteed results.

As each of the edges in the anchor trees is used to constrain the search space, our method is most successful given datasets with high CI, which result in highly structured anchor trees. Likewise, a data set with a high CI implies that a large subset of compatible characters is present. In general, we expect that the method will be more successful when fewer anchor trees are found, as this will be the case when there are large subsets of compatible characters. These large subsets will in turn give large sets of anchor edges whose members are compatible with each other but largely incompatible with members of other sets. The size of the eventual space to be searched is dependent on the number of resolutions of the constraint trees, and as the anchor edges are used to generate constraint trees, large sets of anchor edges will give more resolved constraint trees, and incompatibility between sets will reduce the number of constraint trees. Both of these outcomes will reduce the number of fully resolved trees to search.

In addition to the method of enumerating the trees in the bounded space that we discussed above, the trees could be discovered—using existing software—by employing the SPR distance (because d_{rRF} is not standard). SPR is a superset of the RF distance, thus, if the distance between two trees is k under d_{rRF}, then the distance between the two trees under the SPR distance is $\leq k$:

$$T \in SearchSpace \iff \sum_{i=1}^{m} l(T_i) + d_{rRF}(T_i, T) \leq \mathbb{D}$$

$$\Rightarrow \text{ for some } c,\ d_{SPR}(T, T_c) \leq \mathbb{D}.$$

Our method has some limitations. Specifically, we rely on the fact that, as one moves away from an anchor tree, the scores of subsequent trees must increase. However, if the sizes of the subsets of compatible characters are too small relative to the size of the set of all characters, then that behavior, although still occurring, can be overwhelmed by changes in the combined scores of all the other characters in the superset. That is, each compatible subset has a signal, whose strength relative to the signal of the superset must be large, lest it be lost in the noise of the signal of the superset. Also, it is possible that the number of edges $|E| < \min(s_C(T_I)) - M_t$. In that case, too few anchor edges are discovered, and it is not possible to constrain the search space. This can occur even given anchor edges with high CIs.

There are several directions for future work. The first are algorithmic challenges to improve the efficiency of the approach. For instance, due to limitations in the off-the-shelf software we used, we were not able to take advantage of the tightening bounds during our search of the bounded space to further reduce the size of the space. In addition, we might incorporate the approaches of Hendy et al. (1980) and Holland et al. (2005) for calculating static lower bounds into the computational framework. Both these changes would shrink the number of trees currently explored without requiring improvements to the underlying theoretical results.

Finding a near-perfect phylogeny on binary characters is fixed-parameter tractable, where the parameter is the number of substitutions or transformations (i.e., the distance from the minimum tree). For trees with high CIs, Blelloch et al. (2006) give a polynomial-time algorithm to find the optimal tree. Using our approach to discover trees with high CIs, one could then forego additional steps, and instead use the method of Blelloch et al. (2006) to find the exact optimal tree.

Other possible extensions include using our framework to approach the problem of finding the ML tree for a sequence of characters. The work of Money and Whelan (2011) suggests that islands of trees exists for the SPR metric, but difficulties lie in determining the optimal score for a tree topology given that multiple optima can exist across the choices of branch lengths (Steel 1994). Jermiin et al. (1997) explored regions of the space of tree with nearly-optimum ML scores that could be similarly classified as islands of trees. They pointed out that near optimal trees may contain useful information pertaining to the true, but unknown, tree, and presented a method to generate a weighted consensus tree covering the optimal and near-optimal trees. Their method also allowed them to quantify support for different splits in different trees. Effectively, they focused on generating a weighted average of the trees in phylogenetic islands. Building on Jermiin et al. (1997) and extending the work of others, Wolf et al. (2000) argued for refocussing the search of tree space such that all optimal and near-optimal trees were found for a given data set. In addition, they provided an approximation algorithm to do so. Our method offers advantages over the method of Wolf et al., by providing what they argued for but were unable to accomplish: an algorithm to discover all optimal trees. In doing so, it implicitly provides an exact answer for the number of optimal trees that exist for a given data set. This information is useful. For example, it allows a researcher to illustrate cases of topological model uncertainty.

Funding

Partial funding was provided by the U.S. National Science Foundation [grant number #0920920 to K.S.]. This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number W911NF- 05-1-0271 to WCW.

Acknowledgments

We would like to thank Ronald Clouse for the *Metasiro* data set and Mike Charleston, Sean Cleary, and Barbara Holland for insightful conversations. The article was greatly improved by comments and suggestions of the editors, Andy Anderson and Lars Jermiin, and the referees: Mike Steel and two anonymous referees.

Appendix: Proofs

Theorem 1. Let C be a compatible sequence of characters and let T_C be the associated anchor tree. Then, for any tree T leaf labeled by C,

$$l(C,T) \geq l(C,T_C) + d_{rRF}(T_C,T)$$

Proof: Let $T_C = (V_C, E_C)$ be the minimal tree compatible with the compatible character sequence C and let $T = (V,E)$ be a tree with leaves labeled by C. The proof proceeds by induction on $d = d_{rRF}(T_C,T)$.

If $d=0$, then T is a resolution of T_C (i.e., every edge of T_C also occurs in T). Since T_C is a perfect phylogeny for C, every edge corresponds to a character and further resolution of polytomies will not lower the score. So, $l(C,T) = l(C,T_C)$.

Assume that $d>0$. By definition of the relaxed RF distance, there exists, d_+ and d_- such that:

- $d = d_+ + d_-$,
- d_+ is the number of edges in T that are not compatible with T_C, and
- d_- is the number of edges missing from T that are in T_C.

We note that T could possibly have other edges that are compatible with T_C. These edges make no difference to the score, since, as in the base case, resolutions of perfect phylogenies have the same score as the perfect phylogeny.

If $d_+ > 0$, then there is at least one edge e' that is in T but that is not compatible with T_C. Contract the edge e' and call the resulting tree T'.By construction, $d_{rRF}(T_C,T') = d-1$ and by inductive hypothesis, $l(C,T') \geq l(C,T_C)+d-1$. By definition e' is not compatible with T_C, implying there is at least one edge, e_c, from T_C which is incompatible with e'. Since every edge in T_C is associated with a character from C, there is at least one character $c \in C$ that is incompatible with e' and must take at least one additional change across T'. Combining these two facts, we have:

$$\begin{aligned} l(C,T) &\geq l(C,T')+1 \\ &\geq (l(C,T_C)+d-1)+1 = l(C,T_C)+d \\ &= l(C,T_C)+d_{rRF}(T_C,T) \end{aligned}$$

If $d_+ = 0$, then $d_- > 0$, and there is at least one edge e'' that is in T_C but has been contracted and not found in T. Contract the edge e'' and call the resulting tree T''.By construction, $d_{rRF}(T_C,T'') = d-1$ and by inductive hypothesis, $l(C,T'') \geq l(C,T_C)+d-1$. e'' is an edge of the original tree T_C, and, by definition of T_C, there is at least one character state change across that edge. Thus, contracting e'' raises the parsimony score by at least 1 (possibly more, if there are multiple character state changes across e''. Combining these two facts, we have:

$$\begin{aligned} l(C,T) &\geq l(C,T'')+1\\ &\geq (l(C,T_C)+d-1)+1 = l(C,T_C)+d\\ &= l(C,T_C)+d_{rRF}(T_C,T) \end{aligned}$$

□

Corollary 1. Let C be a sequence of characters. Let $C_1,C_2,\ldots,C_m$ be a partitioning of C into compatible subsets of characters. For each $i=1,\ldots,m$, let T_i be the anchor tree associated with C_i. Then for any tree T leaf labeled by C,

$$l(C,T) \geq \sum_{i=1}^{m} l(C,T_i)+d_{rRF}(T_i,T)$$

Proof: By Theorem 1, we have for each i, $l(C_i,T) \geq l(C_i,T_i)+d_{rRF}(T_i,T)$. Since parsimony scores are additive, we have for any tree T, $l(C,T)=\sum_{i=1}^{m} l(C_i,T) \geq \sum_{i=1}^{m} l(C,T_i)+d_{rRF}(T_i,T)$ □

References

Agarwala R., Fernandez-Baca D. 1994. A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed. SIAM J. Comput. 23:1216–1224.

Allen B.L., Steel M. 2001. Subtree transfer operations and their induced metrics on evolutionary trees. Ann. Comb. 5:1–15.

Bastert O., Rockmore D., Stadler P.F., Tinhofer G. 2002. Landscapes on spaces of trees. Appl. Math. Comput. 131:439–459.

Baum D.A., Small R.L., Wendel J.F. 1998. Biogeography and floral evolution of baobabs adansonia, bombacaceae as inferred from multiple data sets. Syst. Biol. 47:181–207.

Blelloch G.E., Dhamdhere K., Halperin E., Ravi R., Schwartz R., Sridhar S. 2006. Fixed parameter tractability of binary near-perfect phylogenetic tree reconstruction. In: Automata, Languages and Programming. Springer, pp. 667–678.

Bodlaender H.L., Fellows M.R., Warnow T. 1992. Two strikes against perfect phylogeny. In: Kuich W., editor. Proceedings of the 19th International Colloquium on Automata, Languages and Programming ICALP '92. Berlin, Heidelberg: Springer. pp. 273–283.

Bonet M., Steel M., Warnow T., Yooseph S. 1998. Better methods for solving parsimony and compatibility. J. Comput. Biol. 5:391–407.

Bonet M.L., John K.St. 2010. On the complexity of uspr distance. IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB) 7:572–576.

Bordewich M., Semple C. 2005. On the computational complexity of the rooted subtree prune and regraft distance. Ann. Comb. 8:409–423.

Cavalli-Sforza L.L., Edwards A.W. 1967. Phylogenetic analysis. models and estimation procedures. Am. J. Hum. Genet. 19:233.

Charleston M.A., 1995. Toward a characterization of landscapes of combinatorial optimization problems, with special attention to the phylogeny problem. J. Comput. Biol. 2:439–450.

Clouse R.M., Wheeler W.C. 2014. Descriptions of two new, cryptic species of *Metasiro* (Arachnida: Opiliones: Cyphophthalmi: Neogoveidae) from South Carolina, USA, including a discussion of mitochondrial mutation rates. Zootaxa 3814:177–201.

Coetzee M.P., Wingfield B.D., Bloomer P., Ridley G.S., Wingfield M.J. 2003. Molecular identification and phylogeny of Armillaria isolates from South America and Indo-Malaysia. Mycologia 95:285–293.

Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. 2001. Introduction to Algorithms. 2nd ed. Cambridge (MA): MIT Press.

Day W.H. 1985. Optimal algorithms for comparing trees with labeled leaves. J. Classif. 2:7–28.

Farris J. 1988. Hennig86. Published by the author, Port Jefferson Station, NY.

Farris J.S., Kluge A.G., Eckardt M.J. 1970. A numerical approach to phylogenetic systematics. Syst. Biol. 19:172–189.

Fitch W. 1971. Toward defining the course of evolution: minimum change for a specific tree topology. Syst. Biol. 20:406–416.

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

Goloboff P.A., Farris J.S., Nixon K.C. 2008. TNT, a free program for phylogenetic analysis. Cladistics 24:774–786.

Gusfield D. 1991. Efficient algorithms for inferring evolutionary trees. Networks 21:19–28.

Gysel R., Gusfield D. 2010. Extensions and improvements to the chordal graph approach to the multi-state perfect phylogeny problem. In: Borodovsky M., Gogarten J., Przytycka T., Rajasekaran S., editors. Bioinformatics research and applications. Lecture Notes in Computer Science. Vol. 6053. Berlin, Heidelberg: Springer. pp. 52–60.

Hein J. 1990. Reconstructing evolution of sequences subject to recombination using parsimony. Math. Biosci. 98:185–200.

Hendy M., Foulds L., Penny D. 1980. Proving phylogenetic trees minimal with l-clustering and set partitioning. Math. Biosci. 51:71–88.

Hendy M., Penny D. 1982. Branch and bound algorithms to determine minimal evolutionary trees. Math. Biosci. 59:277–290.

Holland B., Huber K., Penny D., Moulton V. 2005. The minmax squeeze: Guaranteeing a minimal tree for population data. Mol. Biol. Evol. 22:235–242.

Jermiin L.S., Olsen G.J., Mengerson K., Easteal S. 1997. Majority-rule consensus of phylogenetic trees obtained by maximum-likelihood analysis. Mol. Biol. Evol. 14:1296.

Kluge A.G., Farris J.S. 1969. Quantitative phyletics and the evolution of Anurans. Syst. Biol. 18:1–32.

Land A.H., Doig A.G. 1960. An automatic method of solving discrete programming problems. Econometrica 28:497–520.

Li M., Tromp J. Zhang L. 1996. Some notes on the nearest neighbour interchange distance. In: Cai Jin-Yi, Wong ChakKuen, editors. Comput. Comb. Lecture Notes in Computer Science. Vol. 1090. Berlin, Heidelberg: Springer. pp. 343–351.

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

Money D., Whelan S. 2011. Characterizing the phylogenetic tree-search problem. Syst. Biol. 61:228–239.

Penny D., Hendy M.D. 1987. Turbo Tree: a fast algorithm for minimal trees. Computer Applications in the Biosciences: Bioinformatics 3:183–187.

Reeleder R. 2007. Rhexocercosporidium panacis sp. nov., a new anamorphic species causing rusted root of ginseng (panax quinquefolius). Mycologia 99:91–98.

Robinson D. Foulds L.R. 1981. Comparison of phylogenetic trees. Math. Biosci. 53:131–147.

Robinson D.F. 1971. Comparison of labeled trees with valency three. J. Comb. Theory B 11:105–119.

Sanderson M., Donoghue M., Piel W. and Eriksson T. 1994. Treebase: a prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life. Am. J. Bot. 81:183.

Schröder E. 1870. Vier combinatorische probleme. Zeitschrift für Mathematik und Physik 15:361–376.

Semple C., Steel M.. 2003. Phylogenetics In: Oxford lecture series in mathematics and its applications. Vol. 24. Oxford: Oxford University Press.

Steel M. 1992. The complexity of reconstructing trees from qualitative characters and subtrees. J. Classif. 9:91–116.

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

Swofford D.L. 1990. PAUP: Phylogenetic Analysis Using Parsimony. Illinois Natural History Survey.

Varón A., Vinh L.S., Wheeler W.C. 2010. POY version 4: phylogenetic analysis using dynamic homologies. Cladistics 26:72–85.

Wolf M.J., Easteal S., Kahn M., McKay B.D., Jermiin L.S. 2000. TrEXML: a maximum-likelihood approach for extensive tree-space exploration. Bioinformatics 16:383–394.

Wu T., Moulton V., Steel M. 2009. Refining phylogenetic trees given additional data: An algorithm based on parsimony. IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB) 6:118–125.