



ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Cladistics 18 (2002) 514–525

Cladistics

www.academicpress.com

Semi-strict supertrees

Pablo A. Goloboff^{a,*} and Diego Pol^b

^a *Instituto Superior de Entomología, Universidad de Tucumán, and Consejo Nacional de Investigaciones Científicas y Técnicas, Miguel Lillo 205, 4000 San Miguel de Tucumán, Argentina*

^b *Department of Vertebrate Paleontology, The American Museum of Natural History, Central Park West at 79th Street, New York, NY 10024, USA*

Accepted 17 April 2002

Abstract

A method to calculate semi-strict supertrees is proposed. The semi-strict supertrees are calculated by creating the matrix that represents all the groups in the source trees (as done in already existing techniques), and then finding the trees determined by the ultra-clique. The ultra-clique is defined as the set of characters where each possible subset is compatible with each possible subset from the entire matrix. Finding the ultra-clique is computationally complex (since in most cases many of the characters have missing entries), but a heuristic method yields reliable results. When the trees have no conflict, or when there are only two trees, the method produces the exact result for any ordering of the input trees and any ordering of the groups within them; when there are more than two trees and they have conflict, a single ordering or sequence can create some spurious groups, but doing multiple sequences eliminates the spurious groups. The method uses only state set operations, and is thus easily implemented in computer programs. Unlike any existing type of supertree, semi-strict supertrees display all the groups, and only those groups, that are implied by at least some combination of the input trees and contradicted by none. The idea that supertrees should take into account the number of occurrences of a given group, so as to retain some groups even in the case of conflict, is discussed; it is argued that a conceptual equivalent of the majority rule consensus is not possible when the sets of taxa differ among trees. Also, when pruning taxa from a set of trees, the supertree can display groups that contradict the consensus for the entire trees, suggesting that supertrees for matrices with very dissimilar sets of taxa should be interpreted with caution. If (for any valid reason) the data cannot be combined in a single matrix, it is advisable that the taxon sets in the matrices be as similar as possible.

© 2002 Elsevier Science (USA). All rights reserved.

Introduction

Many studies or comparisons in phylogenetics require combining results for different sets of taxa, but combining trees with different sets of taxa is more complicated than it appears at first, and none of the existing methods is entirely satisfactory. The methods of Gordon (1986) and Steel (1992) can be used to combine trees only when they are perfectly compatible, and are thus of limited applicability. The most widely used method is MRP (“Matrix Representation with Parsimony”), where a matrix of group membership variables is created for each source tree (scoring taxa not included in the tree as a missing entry, adding an all-zero “root”). The matrix representing the trees is then ana-

lyzed with parsimony (Baum, 1992; Baum and Ragan, 1993; Ragan, 1992). The obvious advantage of MRP is that it can be easily implemented with existing software for parsimony analysis (like NONA, Goloboff, 1996; PAUP*, Swofford, 1998; and TNT, Goloboff et al., 1999). Some of its drawbacks have been pointed out by the very proponents of the method: groups in more pectinate, larger, and more resolved trees are normally given more influence, and it may create groups which are not implied by any (combination of) input trees. Two other methods for building supertrees, much less widely used, are those of Lanyon (1993) and Semple and Steel (2000), which can be used for any set of trees (compatible or not).

In this paper, we point out problems with all pre-existing methods and present a new method which overcomes these problems. Finally, we discuss some general problems with the interpretation of supertrees.

* Corresponding author.

Problems with MRP

MRP is a method with many problems, some of which have been pointed out by the very defendants of MRP (e.g., Bininda-Emonds and Bryant, 1998; Bininda-Emonds et al., 1999). Because the result is actually determined by a parsimony analysis, when source trees conflict MRP may lead to groups that are supported by no (combination of) source tree(s) or even to groups that are contradicted by each of the source trees (“novel clades” of Bininda-Emonds and Bryant, 1998). Consider as example the two trees in Fig. 1. Some authors suggested that differences in tree symmetry or size could cause problems to MRP, but both trees have the same sets of taxa, and both are entirely pectinate. As there are only two completely resolved trees, each group can be either supported in both trees or supported in one tree and contradicted in the other. The correct result then is equivalent to the strict consensus. Using MRP, however, produces four trees, all with *A* and *B* as closest relatives, which is contradicted by each of the source trees. Additionally, the source trees agree in displaying *B* and *F* as closer to each other than to *A*, while *B* is more closely related to *A* than to *F* in each of the four MRP trees (and their consensus). The result of the MRP analysis thus directly contradicts the information in the two input trees.

This result is caused by interactions between the groups in the two trees. For example, in tree 4 of the MRP results, *A* and *B* are grouped because the “0” in characters 2 and 3 of the matrix can then be accounted for with a single step, as a reversal; that is, *A* and *B* are grouped because in source tree 1 they do *not* belong to

groups *CDEF* and *DEF*. The effect of reversals in MRP will in general be grouping terminals that are excluded from groups in the source trees, but this in no way means that they must be grouped.

Some authors (Purvis, 1995; Ronquist, 1996) have proposed ways to alter the coding of the trees into a matrix, to correct what they perceived as problems with MRP, but other defendants of MRP consider this theoretically flawed and showed examples where the corrections produce unreasonable results (Bininda-Emonds and Bryant, 1998; Bininda-Emonds et al., 1999). Also, Bininda-Emonds and Bryant (1998) suggested that using irreversible characters would eliminate the effect of reversals described above. While that is true, reversals are not the only source of problems in MRP; terminal taxa can still be grouped by being part of groups in the source trees that (because of interactions) are broken into parts in the final result. If irreversible characters are used, result tree 1 is the only shortest tree but that tree, too, has group *AB*—in this case, supported by changes to state “1” (the only possibility, since reversals are forbidden) in characters 5–7, which represent membership to groups *CDEABF*, *DEABF*, and *ABF*. None of these groups in itself, however, provides any grounds for placing *A* and *B* in a group that excludes *F*. Irreversible characters, therefore, can ameliorate some of the problems with MRP, but in no way solve them all.

As another example consider the two trees in Fig. 2, where the MRP tree displays three groups (marked) which, by eye inspection, are obviously contradicted by one of the source trees. It is interesting that two of those groups are from the smaller and more symmetrical tree, which contradicts the idea (Purvis, 1995) that groups in

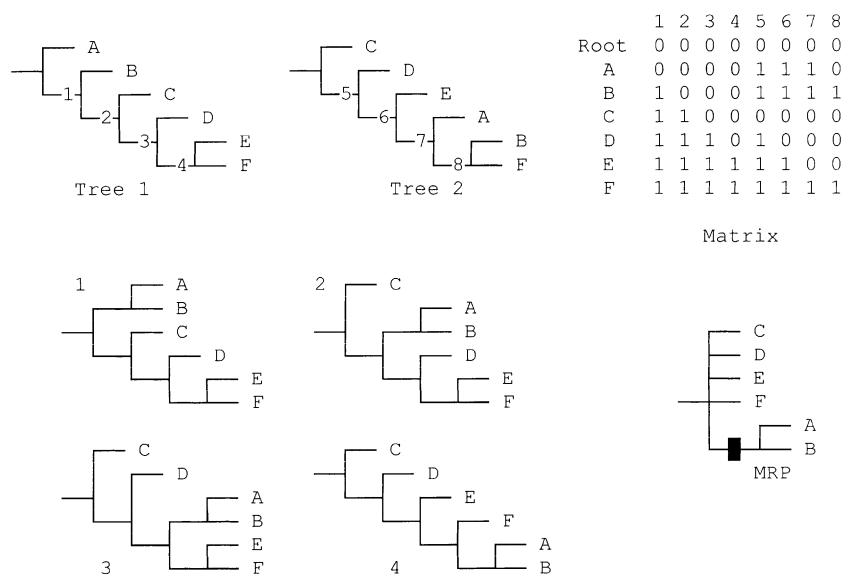


Fig. 1. Two source trees (1–2), the matrix that represents them, the four MRP trees (trees 1–4, produced by analyzing the matrix), and their strict consensus. The group marked, *AB*, is contradicted by both source trees; in both source trees, *B* is closer to *F* than to *A*, while in the MRP tree(s) *B* is closer to *A* than to *F*.

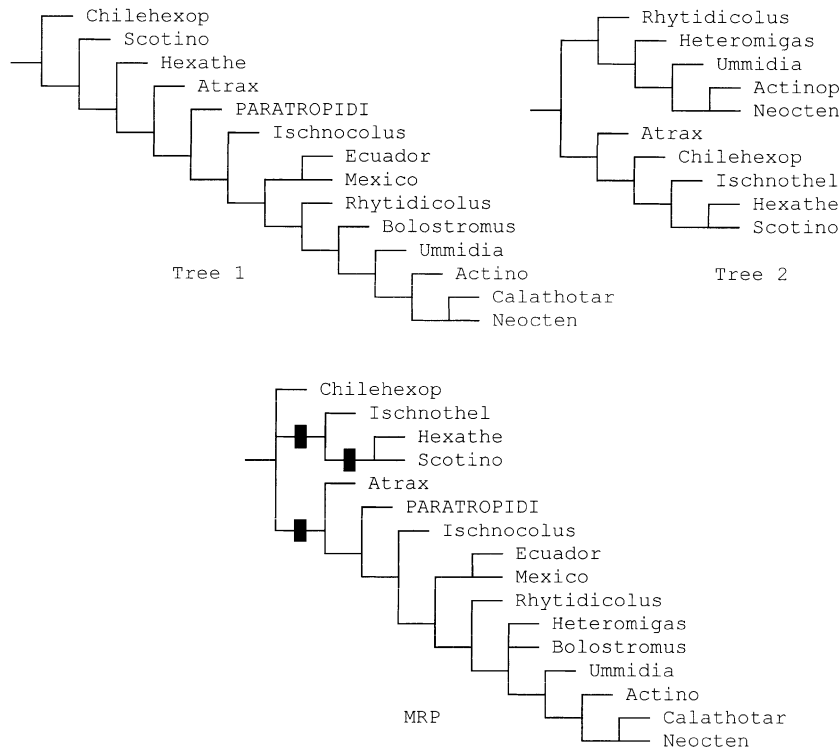


Fig. 2. Two source trees, and their MRP tree. The groups marked in the MRP tree are contradicted by one tree and supported by one tree.

larger and more pectinate trees are given more influence. Bininda-Emonds and Bryant (1998) also provided an example (their Fig. 3) where two trees (of 13 taxa) produce a group contradicted by each of the two source trees.

If MRP can create spurious groups in very simple cases like the ones shown here, it is impossible to predict how it would behave in more complex cases, especially when source trees are numerous. Since MRP is expected to display groups present (or implied) by most of the trees, even if contradicted by a few trees, the results need not be equivalent to a strict consensus, but then it may be impossible to distinguish artifact from groups actually implied by the majority of trees.

It is surprising that MRP has continued being advocated in the face of so many known problems. Perhaps, as suggested by Pisani and Wilkinson (2002), it is because almost no practical alternatives exist. Bininda-Emonds et al. (1999) attempt to defend MRP on the

grounds that it is less biased when there are numerous source trees. However, this is not always true for the simulations of Bininda-Emonds and Sanderson (2001). For real data sets, it would be more correct to say that when the source trees are numerous, the bias cannot be discovered, because it then becomes impossible to tell—by eye—that some results are wrong. That we cannot tell that some results are wrong, however, hardly means that the results are correct.

Problems with MinCut

The MinCut algorithm (Semple and Steel, 2000) is more elaborate than the previous method, and has been proven to have several desirable properties. The method is based on recovering the nestings common to the source trees, as in the Adams consensus (Adams, 1972). While the nestings may be useful under certain very specific circumstances, the MinCut supertree (just like the Adams consensus) will not always be interpretable in terms of monophyly of groups—the interpretation in which taxonomists are normally interested. Thus, the method produces in some cases results which are contrary to expectations.

Semple and Steel’s example 1 (reproduced here as Fig. 3) illustrates this. The result displays only the group *AB*, present in both trees. The group *DE*, however, is present in the first tree and compatible with the second tree.

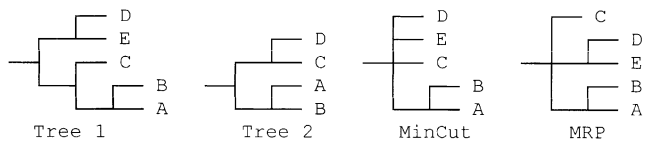


Fig. 3. Two source trees, their MinCut tree, and their MRP tree. The group *DE* is present in one tree and never contradicted, but it is absent from the MinCut tree. The MRP tree displays the group.

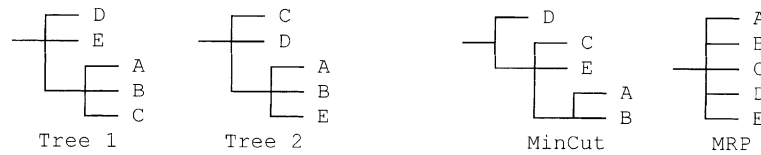


Fig. 4. Two source trees, their MinCut tree, and their MRP tree. The MinCut tree displays two groups which are neither supported nor implied by any of the source trees.

The second tree does not include group *DE* simply because taxon *E* is not included in the tree. Therefore, there is no reason to suppress group *DE* from the output. This illustrates a case in which a group that should be present in the output is not. MRP, despite all its problems, produces a more reasonable result in this case, as it displays the (uncontradicted) group *DE*.

Semple and Steel's example 3 (reproduced here as Fig. 4) shows a case in which the output tree displays a group that is neither supported nor implied by the source trees. Neither the source trees nor their combination imply the groups *ABCE* or *AB*. While the nestings present in the output tree are indeed present in (or compatible with) all the source trees, the groups of the source trees are mutually contradictory. MRP results again are preferable to those of MinCut, because they clearly show that no common or combinable groups are displayed by the input trees.

Problems with Lanyon's (1993) method

Lanyon (1993) proposed a method to construct supertrees which, basically, consists of representing each

tree by a set of "observed" and "possible" groups ("possible" groups result from both alternative resolutions of polytomies and alternative placements of missing taxa). Each of the groups shared by all the trees (i.e., the groups in the intersection of all the sets representing the input trees) will be present in the supertree, as long as (1) it is not contradicted by any other group or (2) it is an "observed" group and is contradicted only by "possible" groups.

The explicit intent of Lanyon was to create a method analogous to the semi-strict or combinable component consensus (Bremer, 1990)—i.e., to create supertrees including groups supported by only some input trees as long as they are not contradicted by other trees. Lanyon's algorithm fails to take into account that a given group may be uncontradicted by the individual input trees but contradicted by some combination of input trees. The algorithm therefore can produce supertrees with groups that are contradicted by some combination of input trees. Fig. 5 is an example, showing three trees and the groups they imply ("observed" groups are marked with an asterisk); the only group shared by the three trees is group *AC*, and thus Lanyon's method produces a supertree with this group. The group *AC* is

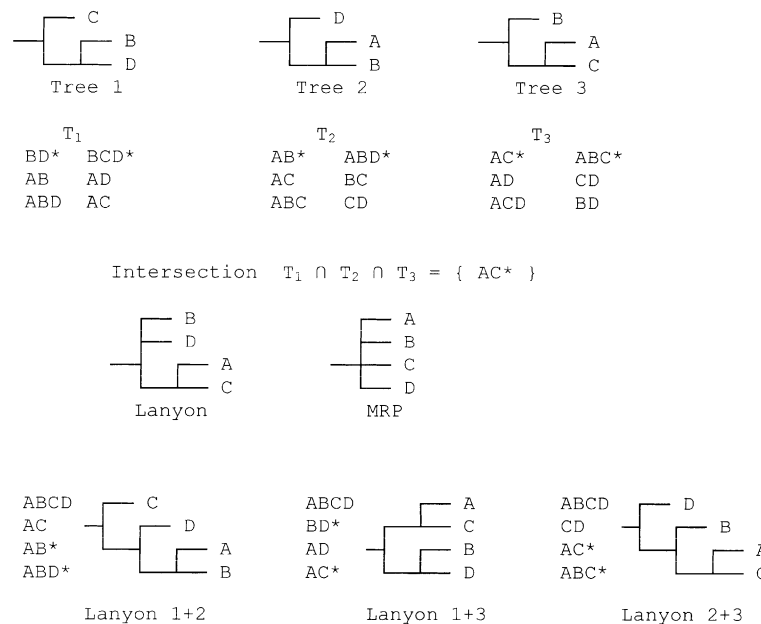


Fig. 5. A case where Lanyon's (1993) method produces a group (*AC*) that is actually contradicted by some input trees. For each input tree, the set of observed/implied groups is indicated (observed groups are marked with an asterisk). For each supertree, the set of groups shared by the input trees is indicated.

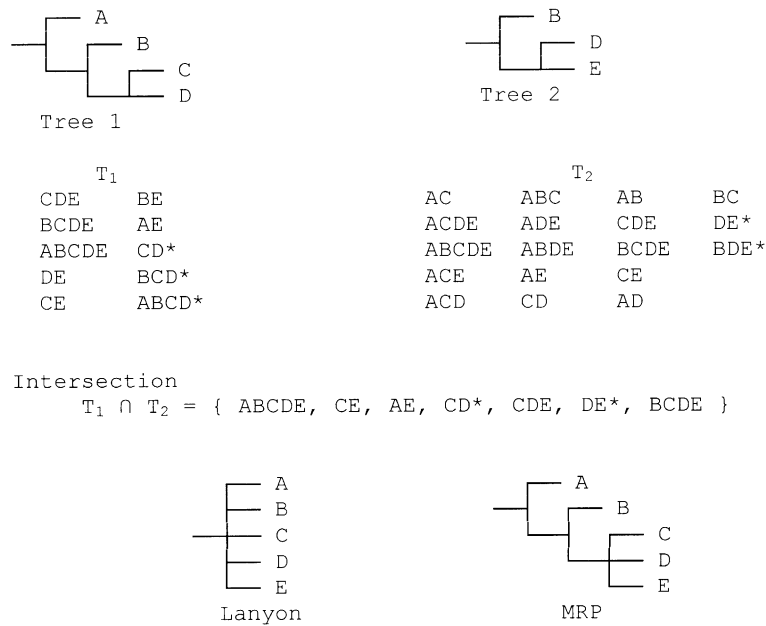


Fig. 6. A case where Lanyon's (1993) method fails to find groups (*BCDE*, *CDE*) that are actually supported by the two input trees combined.

present in some trees, and no tree directly contradicts it, but it is not entirely uncontradicted: combining trees 1 and 2 contradicts group *AC*. In contrast, the MRP tree is a bush, which is accurate for this set of input trees.

Additional problems for Lanyon's algorithm can be created by the distinction between "observed" and "possible" groups, used to decide inclusion or exclusion of a group: when some taxa are missing from a tree, no group is actually "observed," since addition of missing taxa could always make the group non-monophyletic.

Lanyon's algorithm can also miss some groups which are jointly implied by the input trees. Fig. 6 shows two trees, which uniquely determine a supertree with groups *BCDE* and *CDE*: if *E* is closer to *D* than to *B*, *E* can be added to the first tree as a trichotomy between *C* and *D*, and the information in both trees can be combined, without conflict; adding *E* at any other place in the first tree violates implications of the second tree. Lanyon's method, however, produces a bush supertree: the group sets representing the two trees share the group *AE*, which is in conflict with both *BCDE* and *CDE* (none of which is "observed" in the input trees).

Ultra-cliques and semi-strict supertrees

In phylogeny, the strict and semi-strict consensus are normally the preferred means to summarize results, because each group in the strict (or semi-strict) consensus has an unambiguous interpretation: the group must be present in all (or some) of the input trees, and absent (or contradicted) in none. Nixon and Carpenter (1996) have gone so far as to claim that only strict consensus can be

properly called "consensus," and that techniques like Adams or Majority Rule trees must be called "compromise" trees. Terminology aside, it is clear that the strict (and semi-strict) consensus have advantages in the interpretability of the results (we consider pruned trees to be a special case of strict or semi-strict consensus, and not a different technique). As discussed above, none of the supertree methods proposed so far allows for a similarly restrictive interpretation.

An equivalent of the semi-strict consensus can be created from the matrix that represents the source trees (with an all-zero root added), but using only those characters that are not contradicted by any other character (or set of characters) in the entire matrix. We call this set of characters the ultra-clique. The characters in the ultra-clique may not define a unique binary tree; the "semi-strict supertree" or SSS is the consensus of all the binary shortest trees for the ultra-clique.

Note that the ultra-clique is not equivalent to the largest clique in a matrix; the largest clique is the largest number of characters that are not contradicted by each other (Estabrook et al., 1977), while the ultra-clique is instead the set of characters that are not contradicted by any (combination of) character(s) in the matrix.

When some characters have missing entries, different pairs of characters may all be compatible in pairwise comparisons, but collectively non-compatible. Consider the example of Fig. 7, where each of the three characters that represent the input trees is compatible with each of the other characters. Combining each possible pair of trees produces an entirely resolved tree, but the third source tree cannot be accommodated on that resolved tree. When taken together, the characters that represent

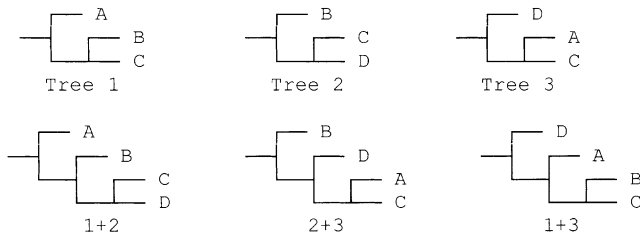


Fig. 7. Three source trees, and the results of combining them in pairs. See text for discussion.

the trees are incompatible, because it is not possible to create a tree where the three characters are free of homoplasy at the same time. The ultra-clique (an empty set, in this case) is thus the largest subset of the characters in a matrix which can be all free of homoplasy on some tree and (singly or collectively) are not in conflict with any other character(s) in the matrix. A matrix may contain several cliques; no matrix can have more than one ultra-clique.

In the example of Fig. 1, every character is contradicted by one (or more) characters. The ultra-clique is an empty set, and therefore the SSS is a bush. In the example of Fig. 2, the characters supporting the marked groups are not compatible with all other characters, and thus the SSS would be like the MRP tree, but lacking those groups. Each of the groups collapsed in the SSS is indeed contradicted by one of the source trees.

The SSS displays groups present in some source trees and unresolved (polytomic) in others, and for the same taxon sets is thus exactly equivalent to the combinable components consensus. In the case of trees with different taxon sets a group may be absent from a source tree either because some of the taxa that make up the group are not included in the tree, or because the group is indeed unresolved. Using the ultra-clique will not distinguish among these two possible situations. A method that collapses a group if unresolved in some input trees (as the strict consensus does) could be called “strict” supertree. Such a method will rarely be needed, however, since supertrees are always used to combine or compare results from different data sets (best done with a semi-strict consensus, not with a strict; see Bremer, 1990, for discussion).

If both are properly calculated, no group from the SSS can be absent from the MRP tree. The SSS will always be compatible with, but possibly less resolved than, the MRP tree. The groups of the MRP absent in the SSS may be either groups that are present in a majority of the source trees (e.g., showing groups that are consistently revealed by many studies), or spurious groups. As with strict or combinable component consensus, there is no direct way to tell. Additionally (as well known for the combinable components consensus; see Nixon and Carpenter, 1996), it may be misleading to combine trees from different matrices if they in turn

represent consensus trees. Consider two matrices, one of which produces two trees, $(A(BC))$ and $(B(AC))$, and the other produces a single tree, $(C(AB))$. The consensus of the first one is (ABC) , which is compatible with $(C(AB))$, and therefore the combinable components consensus (or the SSS) of the two consensus trees is $(C(AB))$ —suggesting that the data sets do not contradict each other. The matrices, however, are incompatible—which would only be detected if the original trees, not their consensus, were used.

Steel et al. (2000) outlined some properties and limitations of possible consensus or supertree methods. Steel et al. show that, given triplets of terminals x , y , and z , no consensus (or supertree) method can satisfy what they call P7, this is, the property of displaying $x(yz)$ if some input tree(s) display $x(yz)$ and no input tree displays $y(xz)$ or $z(xy)$. SSS, however, satisfies the property of displaying $x(yz)$ if it is found in some input tree or implied by some combination of input trees, and no input tree or combination of input trees displays or implies $y(xz)$ or $z(xy)$. This is a refinement of P7 not considered by Steel et al. and, to our knowledge, no other supertree method satisfies this property.

Building semi-strict supertrees

There is a large body of literature on finding cliques of compatible characters, but the evaluation of compatibility if the characters have missing entries remains unsolved. When a (binary) character has no missing entries, it directly defines a group in a tree; when a character has some missing entries, some taxa have their positions as undefined. However, the interaction with other character(s) may define the position of taxa with missing entries. Consider a case with two characters, one with a missing entry for taxon X which defines a group G where taxon X might be excluded or included, and another character which unambiguously places X within a subgroup S of G . Under such a situation, it is obvious that the two characters are compatible, and that a parsimony analysis of those two characters will produce group S as a subgroup of G (and X included within both groups). The missing entry in taxon X could thus be replaced by comparing the two characters. This suggests a way to evaluate compatibility, by replacing the missing entries with pairwise comparisons among characters, preserving compatibility when the character entries could be either compatible or incompatible. If the change makes the character incompatible with a third (as yet unconsidered) character, subsequent pairwise comparisons will detect it. This procedure will always detect the existence of incompatibility (although it may not specify in which characters it occurs).

In the discussion below, we define G_i as the set of terminals with state “1” in the character (variable) that represents group i (i.e., all descendants of node i) and M_i as the set of terminals with missing entry in the character that represents group i (i.e., terminals missing from the tree containing group i). The complement of a set S_i is indicated as $\sim S_i$ (and note that $\sim G_i$ includes all the taxa absent from group i , regardless of whether those taxa were present in the tree or not; therefore, $\sim M_i \cap G_i = G_i$, $G_i \cap M_i = \emptyset$, and $M_i \cap \sim G_i = M_i$). Whether missing entries can be replaced must be checked by making pairwise comparisons for all possible groups i, j . When

$$G_i \cap G_j \neq \emptyset \text{ and } G_i \cap \sim G_j \cap \sim M_j \neq \emptyset$$

$$\text{and } \sim G_i \cap \sim M_i \cap G_j \neq \emptyset$$

the characters have incompatibilities that cannot be solved by replacing missing entries. Otherwise, missing entries can be replaced under four different situations:

- (1) Disjunct groups where one member of the first group is missing in the second group (the missing taxon is placed outside the second group):

if $(G_i \cap G_j = \emptyset)$ and $(G_i \cap \sim G_j \cap \sim M_j \neq \emptyset)$ or $G_j \cap \sim G_i \cap \sim M_i \neq \emptyset$, then:

if $M_i \cap G_j \neq \emptyset$
 $\Rightarrow M_i = M_i \cap \sim G_j$
 if $G_i \cap M_j \neq \emptyset$
 $\Rightarrow M_j = M_j \cap \sim G_i$.

- (2) Identical groups, but with different missing taxa outside the group (the two groups become identical):

if $(G_i = G_j)$
 $\Rightarrow M_i = M_i \cap M_j$
 and
 $\Rightarrow M_j = M_j \cap M_i$.

- (3) One group is contained within the other, and some taxa outside the larger group are missing in the smaller group (the taxa missing from the smaller group are placed outside):

if $G_i \cap \sim G_j \neq \emptyset$ and $\sim G_i \cap G_j = \emptyset$
 $\Rightarrow M_j = M_j \cap \sim (\sim M_i \cap \sim G_i)$
 if $\sim G_i \cap G_j = \emptyset$ and $G_i \cap \sim G_j \neq \emptyset$
 $\Rightarrow M_i = M_i \cap \sim (\sim M_j \cap \sim G_j)$.

- (4) The groups contradict, but the taxa present in the first group and not in the second are missing from the second (the groups are made compatible by including those taxa in the second group):

if $G_i \cap \sim G_j \neq \emptyset$ and $\sim G_i \cap G_j \neq \emptyset$, then:
 if $(G_i \cap M_j \neq \emptyset)$ and $(G_i \cap \sim G_j \cap \sim M_j = \emptyset)$
 and $(\sim G_i \cap \sim M_i \cap G_j \neq \emptyset)$ or
 $(M_i \cap \sim M_j \cap \sim G_j = \emptyset)$ and $(\sim M_i \cap \sim G_i \cap M_j = \emptyset)$
 $\Rightarrow G_j = G_j \cup (G_i \cap M_j)$ and $M_j = M_j \cap \sim G_i$
 if $(G_j \cap M_i \neq \emptyset)$ and $(G_j \cap \sim G_i \cap \sim M_i = \emptyset)$
 and $(\sim G_j \cap \sim M_j \cap G_i \neq \emptyset)$ or
 $(M_j \cap \sim M_i \cap \sim G_i = \emptyset)$ and $(\sim M_j \cap \sim G_j \cap M_i = \emptyset)$
 $\Rightarrow G_i = G_i (G_j \cap M_i)$ and $M_i = M_i \cap \sim G_j$.

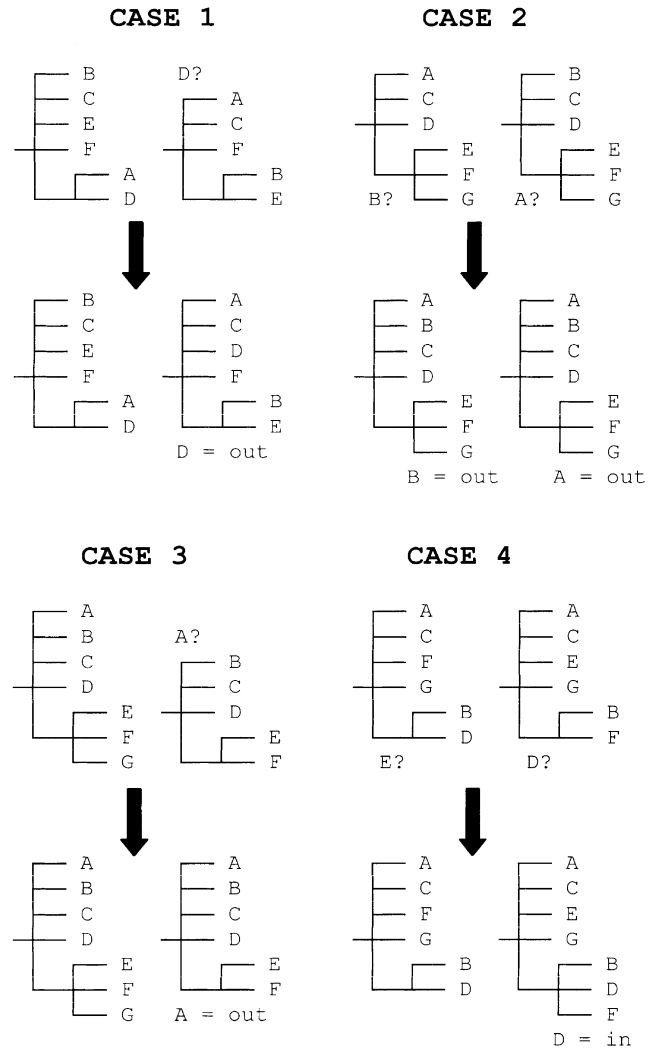


Fig. 8. The possible cases under which missing entries in the characters representing groups can be eliminated. See text for details.

Fig. 8 illustrates the four cases. The cases 1–4 must be iteratively checked for each group i, j . The groups i, j must belong to different trees; the missing entries for a group j in a source tree could not be completed by reference to the other groups in the tree, unless those other groups themselves have been changed by reference to some groups in another source tree (in which case group j as well could be changed by reference to the groups in that other source tree). If missing entries have been changed for some i, j in one cycle, then all the possible comparisons (but only those) where one of the groups has been just changed must be made in the next cycle. The process is repeated until no more changes can be done. Finally, a tree is assembled, using only those characters which: (1) have no incompatibilities, and (2) have no missing entries. Since the characters in the ultra-clique are perfectly compatible, creating a tree from them is automatic. The reason for excluding

characters with missing entries is that for such a character the taxon (or taxa) with missing entry could be either inside or outside the group, yet require a single step for the character. Thus the group—even if not contradicted—will not appear in the consensus of all the trees that are defined by the ultra-clique. Pruning the taxa that retain missing entries by the end of the procedure often improves the resolution of the SSS, and this may help identify taxa which collapse groups because of undefined positions (it will not help identify taxa which collapse groups because of conflicting positions).

By virtue of its design, the algorithm will find all the groups that are actually implied by combining the input trees. When groups do not conflict, or when there are only two source trees, this heuristic approximation never recovers any group that does not actually correspond to the ultra-clique. The method was tested by comparing whether pruning different (random) sets of taxa from a tree recovers the original tree. In a set of 28,000 test cases, where different numbers of taxa were pruned from original trees of 26 taxa, not a single spurious group was detected. As expected, not all the groups present in the original tree were recovered. Table 1 shows the average number of unrecovered groups, for different percentages of taxa pruned, and different numbers of subsets pruned from the original tree. As many cases of unrecovered groups as possible were examined by eye (not the 28,000, of course), to see whether the groups that had not been recovered could be deduced from the source trees. This revealed no cases in which an unrecovered group could be deduced from the source trees.

Although the method works well if all trees are compatible, it can find some wrong groups when conflict exists and there are more than two trees. Fig. 9 is an example. The expected result is a bush (as obvious from comparing the results of combining the three possible pairs of trees in the example) but, depending on which groups are compared first for the replacement of missing entries, the algorithm described above may produce the tree $(BC(AD))$. Comparing first the group $(A(BC))$ (from tree 1) with $(B(AD))$ (from tree 3) the groups $(AD(BC))$ and $(BC(AD))$ are obtained (by rule 1). The only subsequent comparison that can lead to

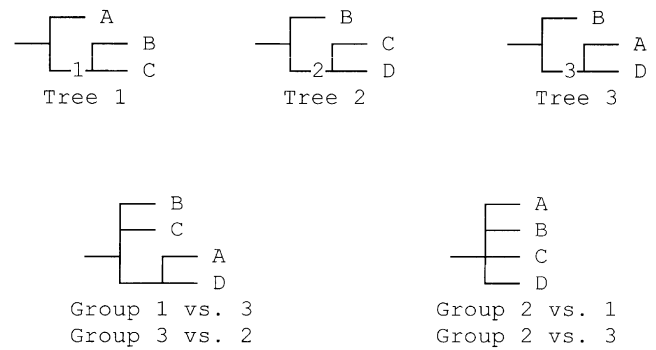


Fig. 9. A case in which different sequences of elimination of missing entries produce different results. See text for details.

replacement of missing entries is the groups from trees 3 (now changed) and 2. Comparing $(BC(AD))$ (from tree 3) with $(B(CD))$ (from tree 2) produces $((AD)BC)$ and $(B(ACD))$ (by rule 4). The group $(BC(AD))$ does not conflict with any other group, and the other two groups conflict with each other. The result is then the (incorrect) tree $(BC(AD))$.

If instead the groups $(B(CD))$ (from tree 2) and $(A(BC))$ (from tree 1) are compared first, they produce (by rule 4) the trees $(B(CD))$ and $(A(BCD))$. Subsequent comparison of groups $(B(CD))$ (from tree 2) and $(B(AD))$ (from tree 3) transforms both groups into $(B(ACD))$ (by rule 4). Each of the final groups now has incompatibilities, so that the result is a bush—as expected.

While applying a single sequence of operations to detect the ultra-clique may lead to errors, it can still detect incompatibilities—if they exist, at least some of the characters that represent the groups will necessarily have incompatibilities. For more than two source trees, any single sequence may yield spurious groups. When incompatibilities are present, the method may consider some unsupported groups as supported, but it can never miss a group that is actually supported—i.e., not contradicted. Thus, when the first sequence detects incompatibilities (and there are more than two trees), further addition sequences can be done, calculating the strict consensus of the results of the different addition sequences, so that the groups incorrectly considered in some cases as supported will (eventually) be

Table 1

Average number of nodes not recovered, for 1000 cases of pruning different sets of taxa from an original tree, to form different numbers of trees

Percentage of taxa pruned	No. of different prunings (No. of source trees)							
	2	3	4	5	6	7	8	
50	5.39	6.47	6.05	5.09	4.40	3.50	2.77	
40	4.18	4.41	3.67	2.78	1.96	1.43	0.98	
33	2.89	2.56	1.78	1.08	0.66	0.43	0.27	
25	1.79	1.24	0.65	0.33	0.14	0.09	0.05	

No case (of the 28,000 cases examined) had any spurious node.

collapsed.¹ As in Goloboff (1999), the best way to do this is to try new sequences for the SSS, calculating the strict consensus of the different SSS until it becomes stable. While this may sound like a lot of work, it is still polynomial time, and (for large numbers of taxa) much less work than finding the shortest trees for the matrix that represents the trees. The method is implemented in the program TNT (Goloboff et al., 1999). For the example 2 of Sanderson et al. (1998) (two trees of 41 and 31 taxa with only 4 taxa in common, and some conflict) TNT takes an average time of 0.006 s ($N = 1000$, with different random seeds, running on an 800 MHz Pentium III) to calculate the same supertree that Sanderson et al. (1998) calculated with MRP. Sanderson et al. do not give timings for the parsimony analysis of the 68-taxon matrix that represents the source trees; since there is little conflict in the data, doing 100 random addition sequences plus TBR (saving a single tree per replication) produces the right consensus (finding the minimum length, 63 steps, in about half of the cases); doing the 100 replications with TNT takes 3.5 s, which may not be much in itself, but is about 500 times slower than the SSS. Another case was provided by five source trees formed by pruning different (random) subsets of 250 taxa from a 500 taxon tree; TNT took an average of 7.5 s to calculate the supertree. These are all cases where a single sequence suffices to guarantee correct results. A more challenging case was in five trees with some incompatibilities (five TBR trees for the “Zilla” data set of Chase et al., 1993, each with a different subset of 50% of the taxa pruned), calculating the SSS until it had stabilized three times (every five different sequences) took 84.2 s to complete (requiring calculation of 32 different sequences).

Are majority rule supertrees possible?

One of the ideas behind MRP is that a group recovered in a larger number of analyses is a better supported group. Proponents of MRP expect the parsimony analysis (and measures of support for the groups

obtained) to automatically take this into account. MRP is then intended as some sort of “majority rule” super-tree technique.

While the SSS is a perfect equivalent of the combinable component consensus, it is dubious that strict equivalents of the majority rule tree are possible when the input trees have different sets of taxa. In many cases it is not possible to count how many trees support (or contradict) a group. A given group may be supported, not by any single tree, but instead only by combinations of trees. One of the trees required for the group to be present in the result may also be a tree that (when combined with a different tree) contradicts the group. Consider the example in Fig. 10, where four trees produce a completely unresolved SSS. Each possible triplet of trees produces a different, completely resolved, tree. Further, there is no conflict in any triplet of trees; conflict only arises when the fourth tree is added. Under such a situation, how many trees support the group *ADE*? The combinations 1 + 3 + 4 and 2 + 3 + 4 support *ADE*, but none of the input trees in themselves support the group. The support for the group is not in trees 1, 2, or 3, because 1 + 2 + 3 contradicts the group, and it is not in 4, because 1 + 2 + 4 also contradicts the group—in both cases, without conflict. This shows that a tree may be required both to support a group, or to contradict it, depending on the trees with which it is to be combined. Therefore, it is not possible to produce a conceptual equivalent of the majority rule consensus tree when the trees have different sets of taxa. A method can check on how many input trees a given partition appears only as long as the taxa involved in the partition—all the ones inside the group and all the ones outside—are present in each and every one of the input trees.

Analyzing the matrix that represents the input trees with maximum compatibility (i.e., finding the trees determined by the largest clique of compatible characters²) will also take into account how frequently a group is recovered, and will often produce results more reasonable than those of MRP. However, more deeply nested groups will still be more influential. In the example of Fig. 2, the compatibility tree has *Atrax* as part of a basal polytomy, but still displays (*Hexathele* + *Scotinoecus*) and (*Ischnothele* + *Hexathele* + *Scotinoecus*)—i.e., it still displays two groups that are supported as often as they are contradicted. Besides these problems, a compatibility analysis would produce the same results as MRP for the example of Fig. 10 (since there is no conflict for any triplet of trees).

¹ It might appear that an exact method would result from creating new groups, instead of changing preexisting ones, when the rules 1–4 above apply. This would seem at first equivalent to trying all possible sequences, but it is not, because rules 1–4 do not do all possible replacements, but instead only those that preserve compatibility; and the method can therefore create spurious groups. An example is the three input trees ($A(B(CD))$), ($B(CE)$), and ($A(D(BE))$), which produce ($A(BD(CE))$), displaying unsupported group *CE*. The method that changes groups (instead of creating new ones) produces ($A(BCDE)$) in about 35% of the cases, and ($A(BD(CE))$) in 65% (thus, for 10 randomly chosen sequences, group *CE* will show up in about 1% of the cases).

² We found the maximum compatibility trees by using a program that searches trees under implied weights, but with the fitting function defined to have value 1 for no homoplasy, and 0 for any number of extra steps.

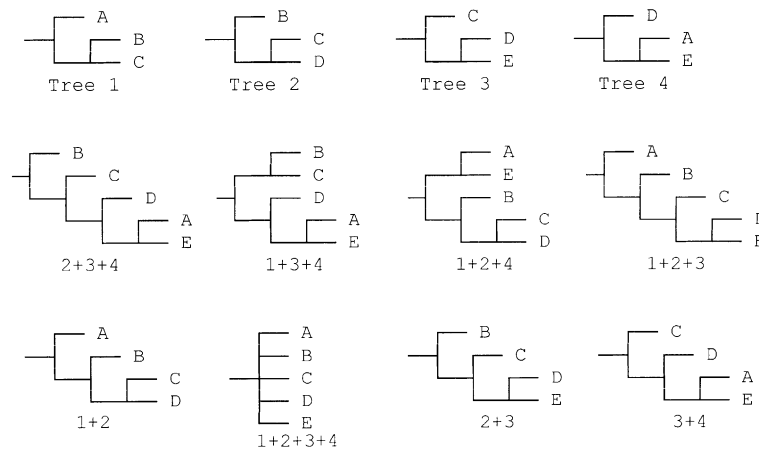


Fig. 10. Results of combining different subsets of four trees (1–4). See text for details.

In conclusion, it is less than obvious how—or whether—it can be determined which groups are “more frequent” when the trees have very different sets of taxa. Designing a method that displays groups that are recovered more often in phylogenetic studies when the taxon subsets for each tree are very different may not be possible.

Even so, no one would dispute the idea that groups recovered more frequently in independent analyses are more likely to be real phenomena. Perhaps this could be taken into account by giving each group in the source trees a “weight” (possibly all unity, although ideally the weight should depend on the character support for the group). Then, use a method such that a given group X of weight W will be collapsed only if contradicted by a group (or groups) for which the sum of the weights is more than W (this could perhaps be accomplished by modifications of the method used to calculate the SSS). While taking into account the problem of the frequency with which a given group is recovered in independent analyses, this is not exactly equivalent to a majority rule tree—i.e., it does not count on how many trees a group is or is not present.

On interpreting the supertree

Supertrees may indeed be valuable exploratory tools, but they have often been defended as more than that. For example, Bininda-Emonds et al. (1999) suggest that the strategy of separately analyzing matrices and combining results has advantages, and that this leads to preferring results from supertrees, rather than from simultaneous analysis of larger data sets. Strictly speaking, supertrees are a summary of the information present in several phylogenetic hypotheses; as such, they must be interpreted as consensus trees. Since a consen-

sus tree is not a phylogenetic hypothesis, it may fit the data more poorly, and characters should therefore not be mapped onto consensus trees (see Nixon and Carpenter, 1996). Supertrees are often used as if they were phylogenetic hypotheses (e.g., Morand and Muller-Graf, 2000), but that practice can lead to unjustified conclusions.³

Even if supertrees are viewed only as a sort of consensus technique, some problems of interpretation remain. The idea behind combining the reduced trees is that one will obtain a tree with a particular relationship to the consensus of the original trees. Should one have a set of complete trees, by pruning some taxa and calculating the tree, it is possible that some of the taxa that create conflict are removed—thus increasing resolution—or that some taxa now have undefined positions—thus decreasing resolution. Either of those cases—increased or decreased resolution when pruning—produces a supertree which is compatible with the consensus for the complete taxon set. In some cases, however, it is possible that by pruning some taxa from a set of (complete) trees, a supertree that contradicts the consensus of the original trees is obtained.

Fig. 11 illustrates such a case. The two input trees produce a consensus with groups BC and EF . If some taxa are pruned from the tree, to produce two reduced trees, their supertree now is completely resolved, but three of the four groups in the supertree contradict the group EF present in each of the complete trees. The reduced trees have no conflict, and thus the MRP tree and the SSS (and possibly the result for any “reason-

³ A reviewer pointed out that supertrees could be safely used as phylogenetic hypotheses if they are fully resolved (to which we would add, having seen the behaviour of methods like MRP: “if they are fully and properly resolved by a strict or semi-strict supertree method”). While that is true in itself, the situation seems (while possible) extraordinarily unlikely in large-scale problems.

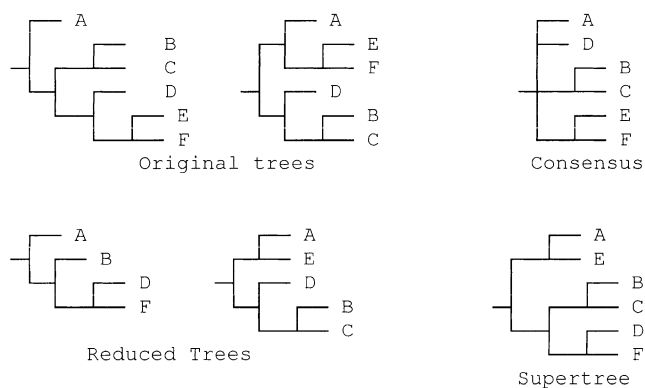


Fig. 11. Two complete trees, and their (strict) consensus, plus two reduced trees (produced by pruning some taxa from the original trees, but otherwise changing nothing) and their (semi-strict) supertree. Some of the groups in the supertree contradict groups in the consensus.

able" supertree method) are identical. Since the consensus for the original trees contradicts groups *AE*, *DF*, and *BCDF* in the supertree, it is difficult to determine what those groups actually mean. It is true that the reduced trees do not contain enough information to determine whether the original trees provided support (or not) for *EF*, but then why should combining the reduced trees lead to the conclusion that those groups are contradicted? It might be argued that the problem is the same as having results changing when removing or adding taxa (or characters) from a real matrix, but removing the taxa from the trees does not alter the evidence used to establish the conclusions—the taxa are not removed from the original matrices, and the relative positions of the remaining taxa are entirely unmodified by the pruning.

Under some specific circumstances problems like this may not occur, as when there are only two trees and the set of taxa for one of them is a subset of the set of taxa for the other. But then this situation—the need for the previous existence of a complete tree—is precisely what supertrees were supposed to avoid. The example suggests that, although building supertrees may be unavoidable when it is impossible to create combined data sets, the results of such an exercise should always be interpreted with caution.

Similar problems may arise when pruning taxa which create conflict from the set of source trees. This is common practice for consensus methods, as pruning the terminals which create conflict (e.g., jumping between different positions in the input trees) may improve the resolution, but may never change the relationships for the rest of the taxa. However, this is true only when the trees have identical taxon sets; when the trees have different sets of terminals, pruning conflicting taxa to create supertrees may change the relationships for the rest of the taxa, and thus may produce results which are difficult to interpret. Consider the four trees of Fig. 10,

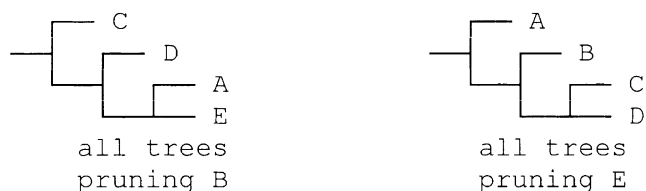


Fig. 12. The results of combining the four trees of Fig. 10, pruning two different terminals, *B* and *E*. When *B* is pruned *D* is closer to *A* than to *C*, but when *E* is pruned *D* is closer to *C* than to *A*. Note that, while the pruning of *B* or *E* makes some trees (1 and 2, or 3 and 4, respectively) irrelevant (as only two taxa are left in the tree), equivalent results are obtained in the presence of additional taxa.

which produced an unresolved SSS. If *B* is pruned from the source trees, the supertree then becomes completely resolved, and the conflict between the (pruned) source trees disappears. But if we prune instead *E* the supertree also becomes completely resolved and the conflict also disappears (as there is no conflict in either case, these results do not depend on the particular supertree method used). However, when pruning *B*, the result has *D* as closer to *A*—that is, $(C(DA))$ —and when pruning *E*, the result has *D* as closer to *C* instead—that is, $(A(CD))$. In other words, if the trees have different sets of taxa, pruning terminals may affect the way in which the *other* terminals are related when trees are combined (see Fig. 12).

The examples, additionally, show that the decision of whether it is preferable to use supertrees (instead of a global analysis) to establish conclusions does not depend on one's position regarding the question of whether or not multiple data sets should be combined or analyzed separately. Even if (for whatever valid reason) the data are to be analyzed separately, they should have a similar taxon composition. Otherwise, the supertree may lead us to conclude that there is unambiguous support for groups which—should the trees have included more taxa—would have been actually contradicted.

Acknowledgments

We thank Jim Carpenter, Jon Coddington, Jan De Laet, Steve Farris, Mike Steel, and two anonymous reviewers for helpful comments and discussion, and FONCyT for financial support (PICT 01-04347 to PAG).

References

- Adams, E., 1972. Consensus techniques and the comparison of taxonomic trees. *Syst. Zool.* 21, 390–397.
- Baum, B., 1992. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. *Taxon* 41, 3–10.

- Baum, B., Ragan, M., 1993. A reply to A.G. Rodrigo's 'A comment on Baum's method for combining phylogenetic trees'. *Taxon* 42, 637–640.
- Bininda-Emonds, O., Bryant, H., 1998. Properties of matrix representation with parsimony analyses. *Syst. Biol.* 47, 497–508.
- Bininda-Emonds, O., Gittleman, J., Purvis, A., 1999. Building large trees by combining phylogenetic information: A complete phylogeny of the extant Carnivora (Mammalia). *Biol. Rev.* 74, 143–175.
- Bininda-Emonds, O., Sanderson, M., 2001. Assessment of the accuracy of matrix representation with parsimony analysis supertree construction. *Syst. Biol.* 50, 565–579.
- Bremer, K., 1990. Combinable component consensus. *Cladistics* 6, 369–372.
- Chase, M., Soltis, D., Olmstead, R., Morgan, D., Les, D., Mishler, B., Duvall, M., Price, R., Hills, H., Qiu, Y.-L., Kron, K., Rettig, J., Conti, E., Palmer, J., Manhart, J., Sytsma, K., Michaels, H., Kress, W., Karol, K., Clark, W., Hedren, M., Gaut, B., Jansen, R., Kim, K.-J., Smith, J., Furnier, G., Strauss, S., Xiang, Q.-Y., Plunkett, G., Soltis, P., Swensen, S., Willimas, S., Gadek, P., Quinn, C., Eguiarte, L., Golenberg, E., Leran, G., Graham, S., Barret, S., Dayanandan, S., Albert, V., 1993. Phylogenetics of seed plants: An analysis of nucleotide sequences from the plastid gene *rbcl*. *Ann. Miss. Bot. Garden* 80, 528–580.
- Estabrook, G., Strauch, J., Fiala, K., 1977. An application of compatibility analysis to the Blackiths data on orthopteroid insects. *Syst. Zool.* 26, 269–276.
- Goloboff, P., 1996. Nona: a tree-searching program. Program and documentation, distributed by the author, San Miguel de Tucumán.
- Goloboff, P., 1999. Analyzing large data sets in reasonable times: solutions for composite optima. *Cladistics* 15, 415–428.
- Goloboff, P., J. Farris, and K. Nixon. 1999. T.N.T.: Tree Analysis Using New Technology. Available at www.cladistics.org.
- Gordon, A., 1986. Consensus supertrees: the synthesis of rooted trees containing overlapping sets of labelled leaves. *J. Classif.* 3, 335–348.
- Lanyon, S., 1993. Phylogenetic frameworks: Towards a firmer foundation for the comparative approach. *Biol. J. Linn. Soc.* 49, 45–61.
- Morand, S., Muller-Graf, C.D., 2000. Muscles or testes? Comparative evidence for sexual competition among dioecious blood parasites (Schistosomatidae) of vertebrates. *Parasitology* 120, 45–56.
- Nixon, K., Carpenter, J., 1996. On consensus, collapsibility, and clade concordance. *Cladistics* 12, 305–321.
- Pisani, D., Wilkinson, M., 2002. Matrix representation with parsimony, taxonomic congruence, and total evidence. *Syst. Biol.* 51, 151–155.
- Purvis, A., 1995. A modification to Baum and Ragan's method for combining phylogenetic trees. *Syst. Biol.* 44, 251–255.
- Ragan, M., 1992. Phylogenetic inference based on matrix representation of trees. *Mol. Phylogenet. Evol.* 1, 51–58.
- Ronquist, F., 1996. Matrix representation of trees, redundancy, and weighting. *Syst. Biol.* 45, 247–253.
- Sanderson, M., Purvis, A., Henze, C., 1998. Phylogenetic supertrees: assembling the trees of life. *Trends Ecol. Evol.* 13, 105–109.
- Semple, C., Steel, M., 2000. A supertree method for rooted trees. *Discrete Appl. Math.* 105, 147–158.
- Steel, M., 1992. The complexity of reconstructing trees from qualitative characters and subtrees. *J. Classif.* 9, 91–116.
- Steel, M., Dress, A., Bocker, S., 2000. Simple but fundamental limitations on supertree and consensus methods. *Syst. Biol.* 49, 363–368.
- Swofford, D., 1998. PAUP*: Phylogenetic Analysis Using Parsimony (* and other methods). Sinauer Associates, Sunderland, Mass.