Journal of Classification 2:255-276 (1985)

Journal of
Classification

Obtaining Common Pruned Trees

C. R. Finden
University of St. Andrews

A. D. Gordon
University of St. Andrews

Abstract: Given two or more dendrograms (rooted tree diagrams) based on the same set of objects, ways are presented of defining and obtaining common pruned trees. Bounds on the size of a largest common pruned tree are introduced, as is a categorization of objects according to whether they belong to all, some, or no largest common pruned trees. Also described is a procedure for regrafting pruned branches, yielding trees for which one can assess the reliability of the depicted relationships. The tree obtained by regrafting branches on to a largest common pruned tree is shown to contain all the classes present in the strict consensus tree. The theory is illustrated by application to two classifications of a set of forty-nine stratigraphical pollen spectra.

Keywords: Common pruned trees; Consensus trees; Hierarchical classification; Regrafting.

1. Introduction

In classification studies (Hartigan 1975; Gordon 1981), the resemblances within a set of objects are commonly represented hierarchically in the form of a dendrogram, or rooted tree diagram, such as those shown in Figures 1 and 2. Given two or more trees describing the same N objects, interest has been increasing in ways of defining and obtaining a consensus tree, which in some sense summarizes the information contained in the original tree diagrams.

Several possible reasons for this activity follow. First, it is well known that different clustering criteria applied to the same data set can, and generally do, produce different results; each clustering criterion explicitly or implicitly assumes an underlying model for the data, and distorts the results

This work was supported by the Science and Engineering Research Council. The authors are grateful to the referees for constructive criticisms of an earlier version of the paper, and to Dr. J.T. Henderson for advice on PASCAL.

Authors' Address: Department of Statistics, University of St. Andrews, North Haugh, St. Andrews KY16 9SS, Scotland. Please address correspondence to Dr. Gordon.

to a greater or lesser degree towards this model. To the extent that similar results are obtained using different clustering criteria, one can be more confident that the results are indicating genuine structure in the data and are not purely artifacts of the particular clustering criteria employed. Secondly, it can be relevant to compare classifications of the same set of objects described by different sets of variables, or by the same variables measured at different times. Thirdly, it can be relevant to investigate the stability of a classification when subjected to small changes in the measurements, or to the removal of objects from the data set. In this latter case, one seeks to identify objects which are *influential*, in a similar spirit to that undertaken in regression analysis (e.g., Cook and Weisberg 1982); thus, if the entire data set is denoted by Ω and the subset under investigation is denoted by ω, one could compare a classification of the objects in $\Omega \setminus omega$ with a classification of the objects in Ω from which one had subsequently pruned all branches attached to objects belonging to ω.

Many different measures of the difference between tree diagrams have been proposed, but this paper concentrates on methods which provide as the end result a tree diagram, thus allowing one to assess the relationships between the N objects. Such consensus trees have been presented by Adams (1972), Margush and McMorris (1981), Diday (1982), and Neumann (1983). These approaches have in common the property that the consensus tree contains all N objects. An alternative approach (Gordon 1980) is to prune as few branches as possible from each tree so as to make the reduced trees in some sense equivalent, yielding a common pruned tree; one possible definition of equivalence is described in the next section. The size of this common pruned tree, i.e., the number of base points which it contains, gives a measure of the resemblance between the original trees. In addition, the objects contained in the common pruned tree are indicated as having their relationships more reliably represented than objects which have been excised from the tree.

This paper describes several algorithms, which have been implemented in computer programs, for obtaining common pruned trees. Also presented is a study of bounds on the size of the largest common pruned tree, which indicates objects that cannot belong to a common pruned tree of a specified size. There may be more than one largest common pruned tree, and the final section introduces a categorization of objects according to the number of such trees to which they belong. Also described in that section is a procedure for regrafting branches which have been excised in obtaining a common pruned tree.

The work will be illustrated by application to the data set analyzed in Figures 1 and 2. These data comprise 49 pollen spectra obtained from, and numbered in order down, a core of sediment from Abernethy Forest, near Inverness (Birks and Mathewes 1978). A principal components analysis of the data revealed five well-separated groups, A (objects 1-15), B (16-32),

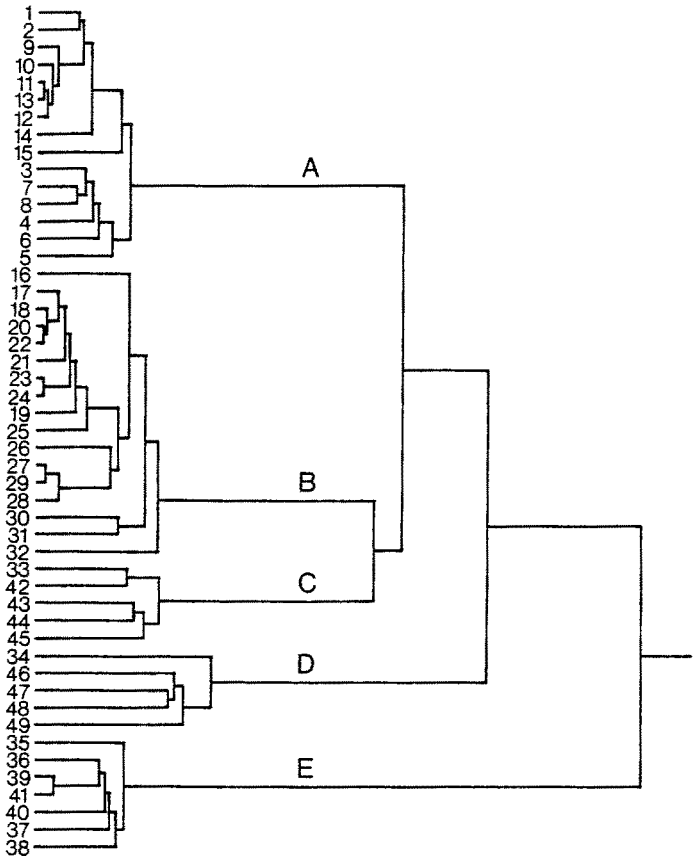

Figure 1. Single link dendrogram for Abernethy Forest data.

C (33,42-45), D (34,46-49), and E (35-41). It can be seen from Figures 1 and 2 that both clustering criteria employed (single link and sum-of-squares) obtain this partition into five groups, but the relationships between these five groups differ in the two dendrograms; the higher level relationships are summarized in Figures 3(a) and 3(b). Figure 3(c) displays the upper levels of the consensus tree obtained from Adams's (1972) second method (relevant for the comparison of trees with unlabeled internal nodes, such as those in Figures 1 and 2). An unsatisfactory feature of the Adams consensus tree is that it contains structure which is not present in either of the original trees.

2. Equivalence of Trees

A rooted tree diagram based on the set of N objects $\Omega \equiv \{ 1, 2, \ldots, N \}$ can be defined as a hierarchically-nested subset of

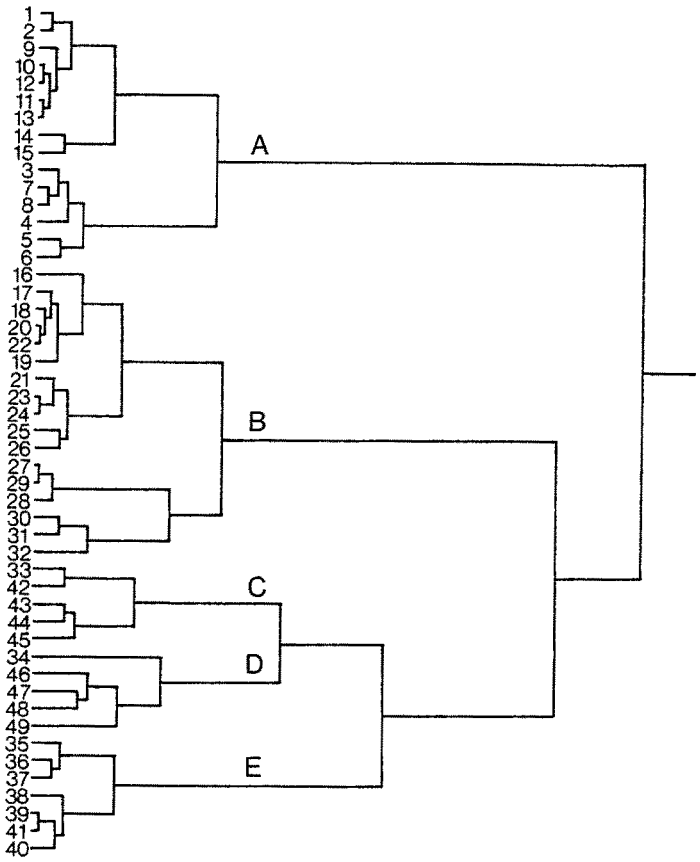

Figure 2. Sum-of-squares dendrogram for Abernethy Forest data.

$P(\Omega)$, the set of all subsets of Ω: it is (Margush and McMorris 1981) a subset T_r of $P(\Omega)$ satisfying the following conditions.

1. $\Omega \in T_r, \varnothing \notin T_r$
2. $\{i\} \in T_r$ for all $i \in \Omega$
3. If $A, B \in T_r$ with $A \cap B \neq \varnothing$, then $A \subseteq B$ or $B \subseteq A$.

Let $\{T_{rj}(j=1, \ldots, n_r)\}$ denote the subsets of objects contained in T_r. These subsets can be placed in a 1:1 relationship with the nodes (vertices) of the tree T_r: there are N terminal nodes, each specifying a singleton subset, and $(n_r - N)$ internal nodes, each of which specifies the subset of objects lying below it in the tree. If T_r is a binary tree, i.e., if each amalgamation is between two subsets, $n_r = 2N - 1$.

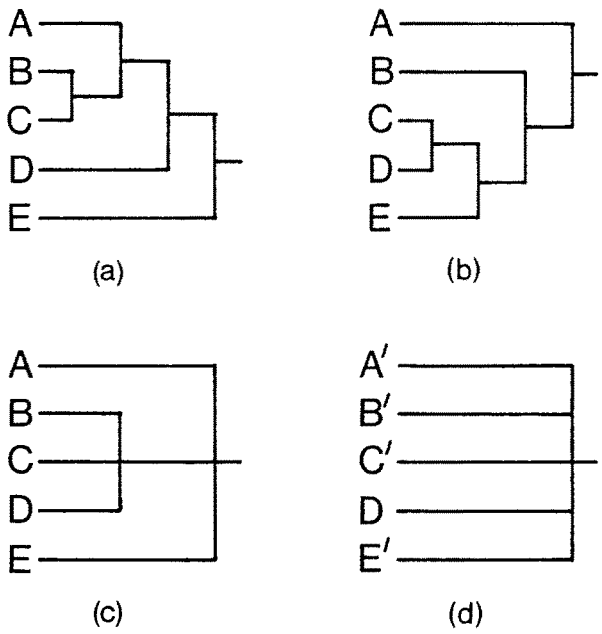

Figure 3. The relationships between the groups in the partition into five groups, as represented in (a) single link dendrogram, (b) sum-of-squares dendrogram, (c) Adams consensus tree, (d) common truncated pruned tree.

In this formulation, no account is taken of the heights of the internal nodes: such a tree has been referred to as local order invariant (Sibson 1972), or as labeled and non-ranked (Murtagh 1984). Two trees comprising the same constituent classes are termed local order equivalent (Sibson 1972).

Given a set of t rooted trees $\{T_1, \ldots, T_t\}$ based on the same set of N objects, consider traversing tree T_r from base point i to the top vertex: let $B_{rk}(i)$ denote the set of objects encountered for the first time at the *k-th* internal vertex that is passed through. For example, the sequence of encounters starting from base point 20 in the tree T_1 depicted in Figure 1 is

$$20,22,18,17,21,(23,24),19,25,(26\text{-}29),16,\ldots,(35\text{-}41),$$

where objects which are bracketed together are first encountered at the same internal vertex; thus,

$$B_{11}(20) = 22,\ B_{12}(20) = 18,\ B_{15}(20) = (23,24),$$
$$B_{1,15}(20) = (35\text{–}41), \text{ and by convention, } B_{10}(20) = 20.$$

The corresponding sequence of encounters in the tree T_2 depicted in Figure 2 is

$$20,22,18,17,19,16,(21,23\text{-}26),(27\text{-}32),...,(1\text{-}15);$$

thus,

$$B_{20}(20) = 20,\ B_{21}(20) = 22,\ B_{25}(20) = 16,\ B_{29}(20) = (1-15).$$

It is clear that trees T_m and T_n are local order equivalent if $B_{mk}(i) \equiv B_{nk}(i)$ for all values of i and k. Restricting attention to the t sequences of encounters starting from the *i-th* base point, let δ_i denote the minimum number of objects which has to be removed from each of the sequences so as to ensure perfect agreement (including the preservation of ties in the ordering) between the reduced sequences; then $\alpha_i \equiv N - \delta_i$ is a measure of the agreement between the sequences starting from the *i-th* base point, and is termed the length of a longest common subsequence starting with the *i-th* base point (Gordon 1979). This problem differs slightly from the standard longest common subsequence problem (see, for example, Wagner and Fischer 1974; Maier 1978) in that attention has to be paid to tied ranks; the longest common subsequence can be obtained using a dynamic programming strategy.

If $\sum_{i=1}^{N} \alpha_i = N^2$, the trees are local order equivalent. If $\sum_{i=1}^{N} \alpha_i < N^2$, in order to obtain local order equivalence one needs to prune the trees. For example, if it is decided to remove object j from the trees, then one prunes the branch attached to base point j at as high a level as possible consistent with not affecting other base points. If one obtains reduced trees based on $M(\leqslant N)$ objects, for which $\sum \alpha_i = M^2$, the reduced trees are local order equivalent; each of them will be referred to as a common pruned tree. The aim is to find common pruned trees with largest value of M. This largest common pruned tree need not be uniquely defined; thus, the two trees depicted in Figures 4(a) and 4(b) have two largest common pruned trees of size 3, these being portrayed in Figures 4(c) and 4(d).

3. Pruning Algorithms

For small data sets, one could envisage examining all $\binom{N}{k}$ ways of pruning the same k objects from each of the t trees and checking the reduced trees for local order equivalence, increasing the value of k until one had identified the largest common pruned tree or trees. For even moderate-sized data sets, however, such a global search is computationally infeasible.

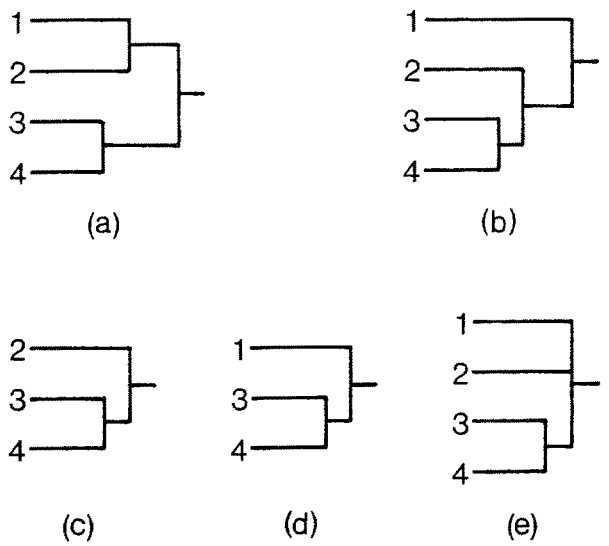

Figure 4. (a), (b) Two trees based on the same set of four objects; (c), (d) the two largest common pruned trees obtained from the trees shown in (a) and (b); (e) strict consensus tree of the trees shown in (a) and (b).

Instead, approximating algorithms which remove objects one at a time have been developed and implemented in two independent programs CMTREE and FSTREE (Finden 1983, 1984a). These programs are restricted to the comparison of two trees, although the trees need not be binary. As is the case with other stepwise optimal classification algorithms, there is no guarantee that the common pruned trees produced by the programs are globally optimal, i.e., largest common pruned trees.

All programs mentioned in this paper run under the VAX/VMS operating system on a DEC VAX-11/780 computer. They are FORTRAN programs written mainly according to the conventions of standard FORTRAN 77 as defined in ANSI(1978); however, there are a few program departures from this specification to take advantage of certain VAX-11 FORTRAN extensions encompassing interactive terminal and space-saving features (DEC 1982).

In programs CMTREE and FSTREE, several different criteria are available for the elimination of objects; thus, the programs will prune a branch and base point j for which σ_j is minimum, where definitions of σ_j include

(i) $\sigma_j = \alpha_j$;
(ii) σ_j = the number of branches for which the *j-th* object occurs in a largest common subsequence;
(iii) a modification of (ii): suppose that the path from the *k-th* base

point to the top vertex has F_k longest common subsequences associated with it, and that the *j-th* object occurs in f_{jk} of these longest common subsequences; then

$$\sigma_j = \sum_k (f_{jk}/ F_k)$$

In options (ii) and (iii), it is not in fact necessary to compute any longest common subsequence, but all longest common subsequences are implicitly considered for object membership; a dynamic programming strategy is used in the computations.

Program CMTREE allows options (i) and (ii), whereas FSTREE allows all three options. FSTREE is an improved and faster version of CMTREE, but CMTREE has been tested over a longer period of time.

The algorithm for program CMTREE is summarized in Table 1 using PASCAL notation and following the conventions and definitions of Day and Edelsbrunner (1984): in that paper, an algorithm's time (resp. space) complexity is described by a function $f(N)$ expressing for each N the largest amount of time (resp. space) the algorithm requires to solve any problem involving N objects; if there exists a positive constant c and a function $g(N)$ such that $f(N) \leqslant cg(N)$ for all large positive N, $f(N)$ is said to be $O(g(N))$.

Using Table 1, the total time complexity of the CMTREE algorithm is seen to be $O(N^5)$. The critical step in resolving the time complexity is step 3: this step is of time complexity $O(N^3)$, and two loops of $O(N)$ encompass it.

The time complexity of step 3 can also be written as $O(K_1\ K_2\ N)$, where K_1 and K_2 are upper bounds to the number of blocks of base points in tracing a sequence from a base point to a top vertex in, respectively, the first and second tree; K_1 and K_2 are each of $O(N)$. The total time complexity of the algorithm is thus $O(K_1\ K_2\ N^3)$. For unbalanced trees, K_1 and K_2 will be close to N, but for well-balanced trees, K_1 and K_2 may lie close to $\log_2 N$. thus, for well-balanced trees, it can be anticipated that the run time will be considerably less than for unbalanced trees of the same size; it has been found in practice that program CMTREE can cope with well-balanced trees having a size N of 128 in 15 minutes dedicated CPU-time on the VAX-11/780. The complexity of the CMTREE algorithm is $O(N^2)$.

Program FSTREE uses virtually the same algorithm and has the same time and space complexities. However, it has been speeded up using experience gained on program CMTREE and uses dynamic storage techniques to save space. Program FSTREE can cope with well-balanced trees of size 128 in 2 minutes dedicated CPU-time on the VAX-11/780.

TABLE 1

Algorithm for Program CMTREE

	Algorithm	Time Complexity
	begin	

1.	Input, check for error, and structure the two trees;
	for k:= 1 to N do
	null[k]:= false;
	repeat
	for k:= 1 to N do
	σ_k:= 0;
	for j:= 1 to N do
	if not null[j] then
	begin
	for i:= 1 to 2 do
	begin
		$O(N^2)$

2.	Calculate sequence of blocks of base points encountered in tracing through the ith reduced tree, T_i', from base point j to top vertex, each block except first representing a branching end;	O(N)
3.	From the sequence of blocks of base points for each tree, calculate, using the dynamic programming equations, the length of a longest common subsequence and the identities of base points which lie in any longest common subsequence;	$O(N^3)$

4.	Update values of σ
	end;
	if $T_1' \neq T_2'$ then
	begin
		O(N)

5.	Find a base point p with smallest σ; null[p]:= true end; until $T_1' = T_2'$;	O(N)
6.	Output the common pruned tree end.	O(N)

When these programs were used to compare the two trees depicted in Figures 1 and 2, the largest common pruned trees obtained contained 26 objects. One of these trees is presented in Figure 5, but this solution is not unique; thus, objects 5, 10, 14, and (23,24) in Figure 5 could be replaced by objects 6, 12, 15, and (19,25) respectively. It is noteworthy that the objects contained in Figure 5 are all derived from only three of the five groups labeled A-E in the original dendrograms. As is shown in Figures 3(a) and 3(b), this is because the relationships between these five groups are different in the two dendrograms. This is an illustration of the fact that hierarchical classifications of the same data using different clustering criteria are particularly likely to differ in the higher levels of the dendrograms.

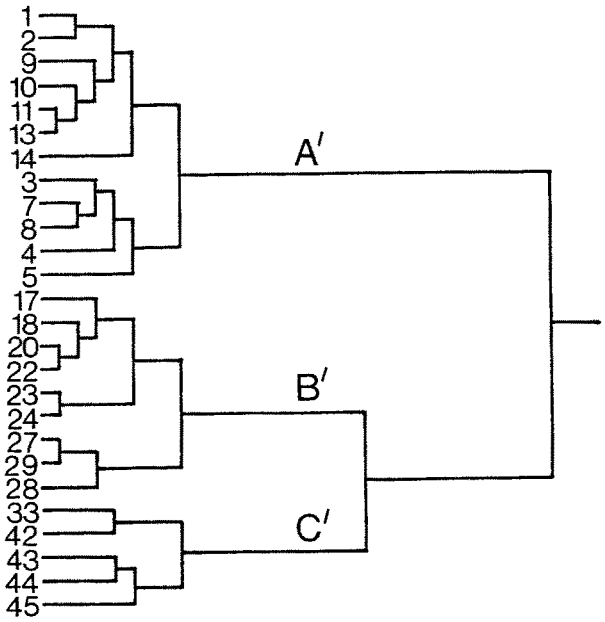

Figure 5. A common pruned tree obtained from the dendrograms depicted in Figures 1 and 2.

Further, it has often been observed (e.g., Rohlf 1970; Kruskal 1977) that hierarchical classifications are more reliable in their representation of low level (small dissimilarity) features than high level features in the data, whereas the reverse is true for geometrical methods of representation such as principal components analysis.

To allow attention to be restricted to the low level relationships, Gordon (1980) suggested comparing truncated sequences, i.e., sequences for which all objects after the first few are bracketed together in a single block and given the same rank. The following methods of truncation of the sequences starting from each base point have been implemented in the programs CMTREE and FSTREE:

(i) each sequence is truncated once it contains (at least) a specified number of objects;
(ii) each sequence is truncated once it has encountered a specified number of internal nodes.

The common pruned structure obtained from applying these truncation criteria need not be a tree, as parts of the structure could be relevant for some but not all of the base points. For example, if sequences are to be truncated once at least four objects have been encountered, the sequences starting

from base points 46, 47, 48 and 49 in T_1 and T_2 would all be truncated at the internal node corresponding to subset (46-49), but the sequences starting from base point 34 would be truncated at the higher internal node corresponding to the subset (34, 46-49). A more stringent requirement leads to a common pruned tree:

(iii) each sequence is truncated once it encounters an internal node below which all sequences passing through the node satisfy criterion (i) (alternatively, (ii)).

Thus, in the above example, sequences from all five base points would be truncated at the internal node corresponding to subset (34, 46-49).

When comparing two trees using truncation criteria, the programs CMTREE and FSTREE operate in the manner already described, deleting objects one at a time until common truncated pruned trees or structures are obtained; an additional feature is that differential weights can be assigned to agreements in the untruncated parts of the sequences and in the final blocks of tied ranks.

To illustrate the methodology, the dendrograms shown in Figures 1 and 2 were compared, with attention being restricted to the first five objects in each sequence, truncated in accordance with criterion (iii). Objects from groups D and E contributed to the largest common truncated pruned tree obtained, which was of size 36. Since the relationships between the objects in group D were in complete agreement in the original trees, all five of them were included; the group E′ contains the objects (35, 38-41). All objects belonging to the final block in a sequence are encountered at the same level, thus the relationships between the final five groups can be represented as shown in Figure 3(d), although the truncation criterion would indicate the two subgroups of A′ as joining together only at the top vertex.

It is worth mentioning that the trees shown in Figures 1 and 2 are fairly similar to one another, having 30 of their 48 non-singleton subsets in common. Experience with comparing other, more divergent, trees indicated that a largest common pruned tree can contain very few objects.

4. Bounds and Further Algorithms

The pruning algorithms described in the previous section cannot be guaranteed to produce a common pruned tree of largest size, and it is pertinent to investigate their efficiency; if the largest common pruned tree is of size N^* and the largest tree provided by a program is of size N', we want $(N^* - N')$ to be small, preferably zero.

One approach is to repeat the analysis of the data many times, using a range of parameter values in the programs: if the largest common pruned tree obtained is of size N', and trees of size N' are obtained a high proportion of the time, one can be more confident that $N' = N^*$.

Alternatively, one can investigate bounds on possible values of N^*; this approach also indicates objects which cannot belong to a largest common pruned tree, and suggests further algorithms for removing object from trees. The bounds and algorithms are based on the matrix $\Lambda \equiv (\lambda_{ij})$, where λ_{ij} is defined to be the length of (i.e., number of objects in) a longest common subsequence starting with base point i and including base point j; and λ_{ii} is α_i, the length of a longest common subsequence starting with base point i. These definitions of length are to be understood as measured in accordance with a truncation criterion if required, although the computer programs introduced in this section are restricted to the investigation of common pruned trees that have been obtained in the absence of any truncation criterion.

The program UBTREE (Finden 1984b) evaluates the elements of the Λ matrix for two given trees, and outputs them to a disc-file for later use. The algorithm for program UBTREE is given in Table 2, from which it can be seen that the total time complexity of the algorithm is $O(N^4)$: the critical step 3, of time complexity $O(N^3)$, is encompassed by a loop of $O(N)$. The space complexity of the algorithm is $O(N^2)$.

A suite of methods of obtaining bounds on possible values of N^* is given in the remainder of this section: in the presentation, it is convenient to distinguish between 1-st order, 2-nd order, and 3-rd order methods, defined in the manner indicated below. Let 1_i denote the *i-th* largest λ_{jj},

$$\text{i.e. } 1_1 \geqslant 1_2 \geqslant \ldots \geqslant 1_N \ .$$

If $1_c \geqslant c$, a common pruned tree of size c might (but need not) exist, but if $1_c < c$ there can be no common pruned tree of size c; hence one can obtain a bound on c^*, the largest value of c for which it is possible that there is a common pruned tree of this size. Further, if $\lambda_{jj} < c$, base point j cannot belong to a common pruned tree of size c. One might, therefore, be able to obtain a sharper bound on c^* by iteratively deleting base points with small values of λ_{jj} and recalculating the other λ_{jj}'s. In that one makes use of only the N diagonal elements of Λ, this approach could be termed a 1-st order method.

The 2-nd order method makes use of the whole of Λ. If there exists a common pruned tree of size c (possibly obtained in accordance with some truncation criterion) with base points $i_1, \ldots, i_c$, then

$$\lambda_{i_j i_k} \geqslant c \quad (j, k = 1, \ldots, c) \ .$$

In other words, there exists a permutation of rows and corresponding columns of Λ which yields a $(c \times c)$ diagonal block in the matrix, all of

TABLE 2

Algorithm for Program UBTREE

Algorithm	Time Complexity
begin	

1. Input, check for error, and structure the two trees;
for j:= 1 to N do
begin
for i:= 1 to 2 do
begin
	$O(N^2)$

2. Calculate sequence of blocks of base points encountered in tracing through the ith tree, T_i, from base point j to top vertex, each block except first representing a branching end;	O(N)
3. From the sequence of blocks of base points for each tree, calculate, using the dynamic programming equations, the jth row of the Λ matrix;	$O(N^3)$
4. Output the jth row of the Λ matrix end end.	O(N)

whose elements are no less than c. If Λ possesses this property, it will be termed block diagonal of order c. If Λ is not block diagonal of order c, there is no common pruned tree of size c; hence one obtains another bound on c^*, which cannot be worse than that provided by the 1-st order method. As before, one could envisage an iterative algorithm which deleted some rows (and corresponding columns) of Λ, and recalculated the remaining elements. Instead of pursuing this idea, the remainder of this section presents a slightly more elaborate deletion scheme, which should provide a sharper bound on c^*.

In this 3-rd order method, one specifies a preferred base point n which cannot be directly deleted in the iterative stage. A necessary condition for base point j to belong to a common tree, which includes base point n, of size c is that it belongs to a diagonalizable block, which includes base point n, of size $(c \times c)$ in Λ all of whose elements are no less than c. If Λ does not contain a $(c \times c)$ block, including base points j and n, all of whose elements are no less than c, base point j cannot belong to a common pruned tree of size c which includes base point n, and so j can be deleted.

The problem of checking for the existence of such $(c \times c)$ blocks (including specified rows) has arisen in many guises; for example, it can be related to the graph theoretical problem of checking for the existence of

complete subgraphs of size c (including specified vertices). This correspondence can be seen by defining the matrix $\Theta \equiv (\theta_{ij})$ by

$$\theta_{ij} = \begin{cases} 1 \text{ if both } \lambda_{ij} \geqslant c \text{ and } \lambda_{ji} \geqslant c \\ 0 \text{ otherwise} \end{cases}$$

with $\theta_{ij} = 1(0)$ denoting the presence (absence) of an edge joining vertices i and j.

This problem is NP-complete (Garey and Johnson 1979). An approximating algorithm has been implemented in program LMTREE (Finden 1984c); this program reads in the Λ matrix calculated by UBTREE, and investigates the feasibility of user-specified values of c', the postulated value of the upper bound c^* to the size of the common pruned tree for the two given trees.

The algorithm for program LMTREE is given in Table 3. The outer loop (step 2) is concerned with the input of various values of c': as the user can zero in on the best estimate for the upper bound c^* by a binary-split method, this loop should be completed $O(\log N)$ times. The total time complexity of the algorithm can thus be seen to be $O(N^3 \log N)$. The space complexity of the algorithm is $O(N^2)$.

Selected explanatory comments on Table 3 follow.

1. The Λ matrix does not possess a $(c \times c)$ diagonalizable block, including base points n and j, all of whose entries are no less than c if either (i) $\lambda_{jj} < c$ or $\lambda_{jn} < c$ or $\lambda_{nj} < c$ (step 3), or (ii) the number of non-excised base points k for which $\lambda_{kj} \geqslant c$ and $\lambda_{jk} \geqslant c$ is less than c (step 4).

2. In steps 7 - 9, an investigation is undertaken of elements of the reduced Λ matrix (containing M rows) which are less than c (alternatively, of the Z (say) zero elements in the reduced Θ matrix). Let ζ_i denote the number of distinct zeroes in the *i-th* row plus the *i-th* column, and let the set $\{\zeta_i (i = 1, \ldots, M)\}$, with the element corresponding to the preferred point n deleted, be ordered in decreasing order of magnitude, denoted by $\{z_j (j = 1, \ldots, M-1)\}$, where

$$z_1 \geqslant z_2 \geqslant \ldots \geqslant z_{M-1} \quad .$$

If the reduced Θ matrix can be made into a matrix whose elements are all 1 by the deletion of L rows and corresponding columns, it follows that

TABLE 3

Algorithm for Program LMTREE

	Algorithm	Time Complexity
	begin	
1.	Input Λ matrix;	$O(N^2)$
	repeat	
2.	Input suspected upper bound, c';	O(1)
	if c'>0 then	
	begin	
	count:= 0;	
	for n:= 1 to N do	
	begin	
	reject:= false;	
	for i:= 1 to N do pres[i]:= true;	
	if λ_{nn}<c' then reject:= true;	
	if not reject then	
3.	begin	O(N)
	for j:= 1 to N do	
	if λ_{jj}<c' or λ_{nj}<c' or λ_{jn}<c' then	
	pres[j]:= false;	
	repeat	
	for j:= 1 to N do	
	if pres[j] and j ≠ n then	
4.	if Number of non-excised base points k for which λ_{kj}≥c' and λ_{jk}≥c' is less than c' then pres[j]:= false;	O(N)
5.	until No base points excised in loop;	O(1)
6.	if Number of non-excised base points k for which λ_{kn}≥c' and λ_{nk}≥c' is less than c' then reject:= true	O(N)
	end;	
	if not reject then	
	begin	
7.	Z:= Number of non-excised elements λ_{ij} of Λ matrix for which λ_{ij}<c' or λ_{ji}<c';	$O(N^2)$
	mm:= 0;	
	for i:= 1 to N do	
	if pres[i] and i≠n then	
	begin	
	mm:= mm + 1; S[mm]:= 0;	
	for j:= 1 to N do	
	if pres[j] and (λ_{ij}<c' or λ_{ji}<c') then	
	if i = j then S[mm]:= S[mm] + 1	
	else S[mm]:= S[mm] + 2	
	end;	
8.	Sort mm entries in vector S into decreasing order;	O(N log N)
	M:= mm + 1; L:= M - c';	
9.	Y:= Sum of first L entries in vector S;	O(N)
	if Z>Y then reject:= true	
	end;	
	if not reject then	
	count:= count + 1	
	end;	
	if count ≥ c' then	
10.	Output that common pruned tree of size c' is possible	O(1)
	else	
11.	Output that common pruned tree of size c' is not possible	O(1)
	end;	
	until c' = 0	
	end.	

$$Z \leqslant \sum_{i=1}^{L} z_i \quad .$$

Thus, if it is the case that

$$Z > \sum_{i=1}^{M-c} z_i \quad ,$$

there can be no common pruned trees of size c.

When the programs UBTREE and LMTREE were used to compare the dendrograms depicted in Figures 1 and 2, the result $c^* = 29$ was obtained, i.e., the largest common pruned tree might contain 29 objects but cannot contain 30 objects. (In this instance, the same bound c^* was provided by the 1-st order method, but for other data sets examined, the 3-rd order bound has been an improvement on the 1-st order bound.) A detailed study of these data indicated that there is in fact no common pruned tree of size greater than 26, and this was the size of tree provided by the programs CMTREE and FSTREE a high proportion of the time. However, the search for common pruned trees of size greater than 26 was facilitated by the marked group structure present in these data; although the bound cannot be guaranteed to be completely sharp, the algorithm should be of assistance in the comparison of tree diagrams. It is also relevant to note that the algorithm indicates that the twelve objects belonging to groups D and E cannot belong to a common pruned tree of size 26. This result is useful, but it will be seen in the next section that a more precise statement can be made about these data.

5. Categorization and Regrafting

The largest common pruned tree need not, and in general will not, be unique. One can thus seek to categorize base points into one of three classes:

class A, comprising base points which belong to all largest common pruned trees;

class S, comprising base points which belong to some but not all largest common pruned trees; and

class N, comprising base points which belong to no largest common pruned tree.

If there are not too many common pruned trees of maximum size, it can be informative to obtain all of them: running the programs CMTREE

and FSTREE with different parameter settings might achieve this; there is also an option allowing the user to specify base points for deletion, thus enabling one to avoid particular optimal solutions. Further, the algorithm described in the previous section allows one to identify (some of the) points belonging to class N. With sufficient effort, therefore, one might be able to categorize all the base points into one of the three classes specified above. For example, a study of the data analyzed in this paper suggested that the class memberships are:

A: 1-4, 7-9, 11, 13, 17, 18, 20, 22, 27-29, 33, 42-45
S: 5, 6, 10, 12, 14, 15, 19, 23-25
N: 16, 21, 26, 30-32, 34-41, 46-49.

If, however, attention is restricted to the first five objects in each sequence, truncated in accordance with criterion (iii), objects (34, 35, 38-41, 46-49) are moved from class N to class A.

It can also be informative to use different symbols to depict the relationships in the common pruned tree of objects belonging to different classes. This is illustrated in Figure 6 for the objects belonging to group B. In this, the two largest common pruned trees of size 9 are depicted by unbroken lines (class A objects) and dashed lines (class S objects); relationships depicted by the other two symbols (- · - and · · ·) should be disregarded at present.

A further elaboration suggested by Gordon (1980) was to regraft branches which had been removed in obtaining the common pruned tree: it was proposed that these be attached at the lowest possible level in the common pruned tree for which their subsequent (higher level) relationships with objects in the common pruned tree are in complete agreement in the original trees. One way of achieving this aim is given by the algorithm summarized in Table 4. Before discussing this algorithm, however, it is necessary to define the concept of a strict consensus tree and introduce further terminology.

Recall that a tree T_r was defined in terms of its constituent hierarchically-nested subsets $\{T_{rj}(j = 1, \ldots, n_r)\}$. Sokal and Rohlf (1981) defined the strict consensus tree SC of the trees $T_1, \ldots, T_t$ to comprise those subsets $\{SC_j(j = 1, \ldots, n_s)\}$ that belong to all of the trees $T_1, \ldots, T_t$. Strict consensus trees are a special case of Margush and McMorris's (1981) majority-rule consensus n-trees, the two types of consensus tree being identical when two original trees are compared.

As an illustration, consider the trees depicted in Figures 4(a) and 4(b): the subsets in these trees are

(a): {1}, {2}, {3}, {4}, {1,2}, {3,4}, {1,2,3,4}; and
(b): {1}, {2}, {3}, {4}, {3,4}, {2,3,4}, {1,2,3,4};

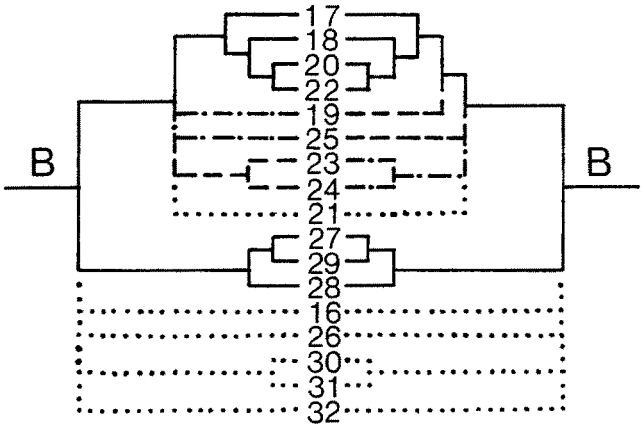

Figure 6. Two common pruned and regrafted trees of the objects in group B. Relationships between base point in class A are denoted by ---- ; class S by - - - or -·- ; class N by · · · .

hence, the strict consensus tree comprises the subsets

$$\{1\}, \{2\}, \{3\}, \{4\}, \{3,4\}, \{1,2,3,4\} .$$

This strict consensus tree is portrayed in Figure 4(e).

Introducing further terminology: let C denote the set of objects belonging to the largest common pruned tree, denoted by CP, which is to be subjected to regrafting; let $P = \Omega \setminus C$ denote the set of objects pruned from the original trees $T_1, \ldots, T_t$; and let $\{CP_j (j = 1, \ldots, n_c)\}$ denote the subsets of objects that are present in CP.

In the algorithm summarized in Table 4, one considers adding an object $k \in P$ to one of the subsets CP_i (say) of CP. The augmented class $CP_i \cup \{k\}$ is defined to be *compatible* with the tree T_r if there exists a class T_{rj} such that

(i) all members of $CP_i \cup \{k\}$ belong to T_{rj}; and
(ii) T_{rj} does not contain any object belonging to $C \setminus CP_i$.

The object k is added to the smallest subset CP_i for which $CP_i \cup \{k\}$ is compatible with all $T_r (r = 1, \ldots, t)$; since $C \cup \{k\}$ is compatible with $T_1, \ldots, T_t$, there does exist a subset to which k can be added. The elements of P are added to CP only at the end of the algorithm: the tree is not sequentially updated.

The time complexity of the algorithm summarized in Table 4 is $O(N^4 t)$. In testing for compatibility, steps 3 and 5 search down trees; if K_1 and K_2 are upper bounds to the numbers of subsets examined in, respectively, steps 3 and 5, then K_1 and K_2 are both $O(N)$. However, if the trees are well-balanced, or the data are such that all objects $k \in P$ are added at a

TABLE 4

Regrafting algorithm

Algorithm	Time Complexity
begin	
1. Identify all classes comprising solely members of P which belong to every T_r (r = 1,...,t). Let the maximal classes be denoted by $P_1,...,P_p$; for each P_m (m = 1,...,p), obtain the strict consensus tree based on the objects in P_m;	$O(N^2t)$
2. for m:= 1 to p do begin Select a representative element k (say) of P_m; C_{old}:= C;	$O(N)$

3. repeat
Determine the set of offspring classes {C[i] (i = 1,...,n)} of C_{old}; i:= 0;
repeat
i:= i+1; r:= 1; matched:= true;
	$O(N)$

4. while r≤t and matched do	$O(t)$

5. begin
if C[i]∪{k} is not compatible with T_r then matched:= false;
r:= r+1;
end;
	$O(N^2)$

until matched or i=n; if matched then C_{old}:= C[i]; until not matched; Print that P_m and its subsets (if any) will be attached to C_{old}; end end.	

NOTE : "A_1 is an offspring class of A_2" means that $A_1 \subset A_2$, and if A_3 is such that $A_1 \subseteq A_3 \subseteq A_2$ then either $A_3 = A_1$ or $A_3 = A_2$.

vertex near the top of *CP*, one can expect the run time to be considerably reduced. The space complexity of the algorithm is $O(N^2t)$.

Figure 6 illustrates the application of the regrafting procedure to the objects contained in group B. The relationships between regrafted objects belonging to class S are depicted by alternate dashes and dots, while those for objects belonging to class N are shown by dotted lines.

It is informative to compare the common pruned and regrafted trees shown in Figure 6 with the strict consensus tree of the objects in group B, depicted in Figure 7. It can be seen that all of the subsets of the strict consensus tree are contained in each of the common pruned and regrafted trees. A referee conjectured that this result held generally. Denoting the tree obtained from applying the regrafting algorithm summarized in Table 4 to a

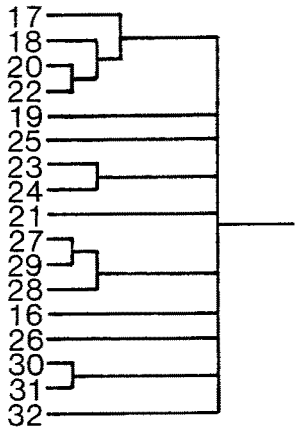

Figure 7. Strict consensus tree of the objects in group B.

largest common pruned tree by *CPG*, and its constituent subsets by $\{CPG_j (j = 1, \ldots, n_g)\}$, the result can be stated formally as:

Theorem $\{SC_j\ (j = 1, \ldots, n_s)\} \subseteq \{CPG_j\ (j = 1, \ldots, n_g)\}$.

Proof The proof is divided into three parts, considering subsets of *SC* which (i) comprise solely members of *P*; (ii) comprise solely members of *C*; (iii) do not belong to (i), (ii) above.

(i) From step 1 of the algorithm in Table 4, all such subsets of *SC* belong to *CPG*.

(iii) All such subsets of *SC* belong to each of $T_1, \ldots, T_t$. Hence, from step 5 of the algorithm, all such subsets of *SC* belong to *CPG*.

(ii) Let $T|_C$ denote the tree obtained from the tree T by pruning from it all objects, with associated branches, which do not belong to C.

Every subset of $SC|_C$ belongs to $T_r|_C$ for $r = 1, \ldots, t$, hence it belongs to *CP* since $T_r|_C = CP (r = 1, \ldots, t)$.

It remains to prove that the regrafting procedure: $CP \rightarrow CPG$ does not destroy subsets of *CP* which belong to *SC* by adding other objects to them.

Suppose SC_j belongs to *CP*. Unless $SC_j = C = \Omega$ (in which case, $CP = CPG$, and the result follows), *CP* must contain at least one other object, or group of objects, C_1 (say), with the property that (SC_j, C_1) is a subset of *CP*.

Consider regrafting an object, or group of objects, P_1 (say) on to $CP.P_1$ will be added to *CP* below the internal node specifying the class (SC_j, C_1) only if group SC_j amalgamates with P_1 before

C_1 in all of the original trees, i.e., only if the class (SC_j, P_1) belongs to $T_r|_{\{C,P_1\}}$ for all r. However, if this is the case, P_1 can be added to CP to obtain a common pruned tree of larger size (contradicting the assumption that CP is a largest common pruned tree).
Hence, SC_j belongs to CPG. •

The common pruned and regrafted trees thus possess the Pareto property of preserving unanimity (Neumann 1983). However, the details of the proof indicate the importance of regrafting objects on to a *largest* common pruned tree.

Figures 6 and 7 illustrate the fact that largest common pruned and regrafted trees can contain more structure than strict consensus trees, this extra structure not being imposed on the data but receiving support from the original trees. This property, and the fact that they present information about which of the relationships in the tree are more reliably depicted, are seen as considerable advantages of such common pruned and regrafted trees. However, it seems likely that detailed categorizations and representations will be helpful only when there are not too many largest common pruned trees.

The manuals listed in the reference section, with associated program listings, are available on request.

References

ADAMS, E.N. (1972), "Consensus Techniques and the Comparison of Taxonomic Trees," *Systematic Zoology, 21*, 390-397.

ANSI (1978), "American National Standard Programming Language FORTRAN, ANSI X3.9 - 1978," American National Standards Institute, New York.

BIRKS, H.H., and MATHEWES, R.W. (1978), "Studies in the Vegetational History of Scotland V," *New Phytologist, 80*, 455-484.

COOK, R.D., and WEISBERG, S. (1982), *Residuals and Influence in Regression*, London: Chapman and Hall.

DAY, W.H.E., and EDELSBRUNNER, H. (1984), "Efficient Algorithms for Agglomerative Hierarchical Clustering Methods," *Journal of Classification, 1*, 7-24.

DEC (1982), "VAX-11 FORTRAN Language Reference Manual," Digital Equipment Corporation, Maynard, Massachusetts.

DIDAY, E. (1982), "Croisements, Ordres et Ultramétriques: Application à la Recherche de Consensus en Classification Automatique," Rapport 144, INRIA, Centre de Rocquencourt, Le Chesnay.

FINDEN, C.R. (1983), "CMTREE Manual," Report, Department of Statistics, University of St. Andrews.

FINDEN, C.R. (1984a), "FSTREE Manual," Report, Department of Statistics, University of St. Andrews.

FINDEN, C.R. (1984b), "UBTREE Manual," Report, Department of Statistics, University of St. Andrews.

FINDEN, C.R. (1984c), "LMTREE Manual," Report, Department of Statistics, University of St. Andrews.

GAREY, M.R., and JOHNSON, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W.H. Freeman and Co.

GORDON, A.D. (1979), "A Measure of the Agreement between Rankings," *Biometrika, 66*, 7-15.

GORDON, A.D. (1980), "On the Assessment and Comparison of Classifications," in *Analyse de Données et Informatique*, ed. R. Tomassone, Le Chesnay: INRIA.

GORDON, A.D. (1981), *Classification: Methods for the Exploratory Analysis of Multivariate Data*, London: Chapman and Hall.

HARTIGAN, J.A. (1975), *Clustering Algorithms*, New York: John Wiley.

KRUSKAL, J. (1977), "The Relationship between Multidimensional Scaling and Clustering," in *Classification and Clustering*, ed. J. Van Ryzin, New York: Academic Press.

MAIER, D. (1978), "The Complexity of Some Problems on Subsequences and Supersequences," *Journal of the Association for Computing Machinery, 25*, 322-336.

MARGUSH, T., and MC MORRIS, F.R. (1981), "Consensus n-Trees," *Bulletin of Mathematical Biology, 43*, 239-244.

MURTAGH, F. (1984), "Counting Dendograms: A Survey," *Discrete Applied Mathematics, 7*, 191-199.

NEUMANN, D.A. (1983), "Faithful Consensus Methods for n-Trees," *Mathematical Biosciences, 63*, 271-287.

ROHLF, F.J. (1970), "Adaptive Hierarchical Clustering Schemes," *Systematic Zoology, 19*, 58-82.

SIBSON, R. (1972), "Order Invariant Methods for Data Analysis," *Journal of the Royal Statistical Society, Series B, 34*, 311-349.

SOKAL, R.R., and ROHLF, F.J. (1981), "Taxonomic Congruence in the Leptopodomorpha Re-examined," *Systematic Zoology, 30*, 309-325.

WAGNER, R.A., and FISCHER, M.J. (1974), "The String-to-String Correction Problem," *Journal of the Association for Computing Machinery, 21*, 168-173.