Journal of Classification 8:177-200 (1991)

The Generation of Random Ultrametric Matrices Representing Dendrograms

François-Joseph Lapointe
Université de Montréal

Pierre Legendre
Université de Montréal

Abstract: Many methods and algorithms to generate random trees of many kinds have been proposed in the literature. No procedure exists however for the generation of dendrograms with randomized fusion levels. Randomized dendrograms can be obtained by randomizing the associated cophenetic matrix. Two algorithms are described. The first one generates completely random dendrograms, i.e., trees with a random topology, random fusion level values, and random assignment of the labels. The second algorithm uses a double-permutation procedure to randomize a given dendrogram; it proceeds by randomization of the fixed fusion levels, instead of using random fusion level values. A proof is presented that the double-permutation procedure is a Uniform Random Generation Algorithm *sensu* Furnas (1984), and a complete example is given.

This work was supported by NSERC Grant No. A7738 to P. Legendre and by a NSERC scholarship to F.-J. Lapointe. The authors would like to thank W. H. E. Day and anonymous referees for their helpful comments.

Authors' address: Département de Sciences biologiques, Université de Montréal, C.P. 6128, Succursale A, Montréal, Québec, Canada H3C 3J7.

Résumé: On retrouve dans la littérature plusieurs méthodes et algorithmes destinés à générer des arbres aléatoires de toutes sortes. Il n'existe cependant aucune procédure permettant la génération de dendrogrammes comportant des niveaux de fusion aléatoires. De tels dendrogrammes peuvent être obtenus à partir des matrices cophénétiques associées. Nous décrivons deux algorithmes pour ce faire. Le premier permet de générer des dendrogrammes complètement aléatoires, c'est-à-dire des arbres possédant une topologie aléatoire, des niveaux de fusion aléatoires ainsi que des feuilles étiquetées de façon aléatoire. Le deuxième algo- rithme utilise une procédure à double permutation afin de randomiser un dendrogramme donné; on proc de dans ce cas à la permutation des véritables niveaux de fusion au lieu de génèrer des niveaux aléatoires. Nous présentons la preuve démontrant que la procédure à double permutation représente un Algorithme de Génération Aléatoire Uniforme *sensu* Furnas (1984). Un exemple complet est également fourni.

Keywords: Random dendrograms; Random matrices; Uniform sampling; Tree algorithm; Monte Carlo studies; Clustering methodology.

1. Introduction

One recent trend in numerical taxonomy is to compare phylogenetic trees on the basis of a random distribution of such trees (Shao and Rohlf 1983; Shao and Sokal 1986; Lapointe and Legendre 1990). In order to facilitate the simulation procedure, simple methods have been developed to enumerate (Knott 1977; Göbel 1980; Solomon and Finkel 1980; Rohlf 1983), generate (Harding 1971; Nijenhuis and Wilf 1978; Rotem and Varol 1978; Proskurowski 1980; Guénoche 1983; Furnas 1984; Oden and Shao 1984; Quiroz 1989), or count (Phipps 1975; Felsenstein 1978; Frank and Svensson 1981; Murtagh 1984) random trees of different kinds. Most of the methods proposed so far have dealt with binary trees, although some have been suggested to enumerate (Rohlf 1983) or generate (Murtagh 1983) random dendrograms (definitions below). In a previous paper (Lapointe and Legendre 1990), we have proposed a new method to compare hierarchical trees; that paper also includes a procedure to generate random cophenetic matrices, which are associated to dendrograms. The present paper presents the algorithm we used to generate these random dendrograms and provides a justification of the method, a complete example, and a proof that our procedure is a Uniform Random Generation Algorithm (*sensu* Furnas 1984), or in other words an algorithm that generates each random element equiprobably. A detailed description is deemed necessary to allow other workers to duplicate and eventually to modify the method.

2. Definitions of Trees

Several types of trees are found in the literature, with sometimes different terminologies. All types of trees are covered by the following definition: *a tree is a connected graph without cycles*. This definition means that for a given set of objects represented by the *nodes* (= *vertices*) of the tree, there should be exactly one path from any one node to another along the *branches* (= *edges*) connecting them (Figure 1).

The *degree* of a node of the tree is the number of edges connected to it. *Terminal nodes* or *pendant vertices* are always of degree one whereas internal nodes are of higher degrees. Terminal nodes are also said to be the *leaves* of the tree (except for degree-one roots). We define a tree as *binary* if none of its internal nodes are of a degree greater than three and as *fully binary* when all its internal nodes are of degree three exactly (Figure 2a). Non-binary alternatives are trees with nodes of degree greater than three (Figure 2b). When a tree contains one and only one internal node, it is said to be a *star tree* or a *bush* (Figure 2c).

A tree is said to be *rooted* (= *directed*) when one of its nodes is labeled as the "*root*" to induce a direction on the edges of the tree. The presence of a root implies that there exist ancestry relations among the nodes, and that the branches are directed. The root is associated to the node representing the ancestor of all the others vertices, whereas the terminal nodes are seen as the offsprings of the internal vertices located closer to the root (Figure 3).

Labeled trees differ from *unlabeled* ones in that labels are assigned (or not) to the nodes of the tree to refer to a given set of objects. These objects usually represent taxa, called Evolutionary Units (EU) or Operational Taxonomic Units (OTU) in numerical taxonomy. They may also represent areas in historical biogeography (Rosen 1978), and so on. Labeled trees are *fully labeled* when all nodes are labeled, or *terminally labeled* when object names are associated to the terminal nodes only (Figure 4).

When values are assigned to the branches of a tree, representing a given function between two nodes, the tree is said to be *weighted*. *Additive trees*, also called *path length trees*, (Figure 5) are weighted trees in which the length of the path connecting two nodes is equal to the sum of all the weighted edges along that path.

Now, let a *dendrogram* be defined as a rooted weighted tree where all terminal nodes are at the same distance (path length) from the root (Figure 6a). Dendrograms can be said to represent *spherical trees* with all terminal nodes placed on the circumference of a sphere with a given radius. The center of the sphere is the root of the dendrogram. Dendrograms will also be referred to as *ultrametric trees*, later on. In this kind of tree, the internal nodes are *ranked* on the basis of their relative distance to the root. *Fusion*

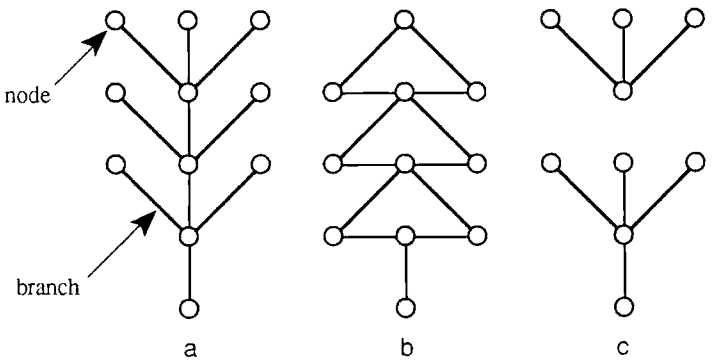


Figure 1. Three graphs: a is a tree, but b and c are not: b contains cycles, while c is not connected.

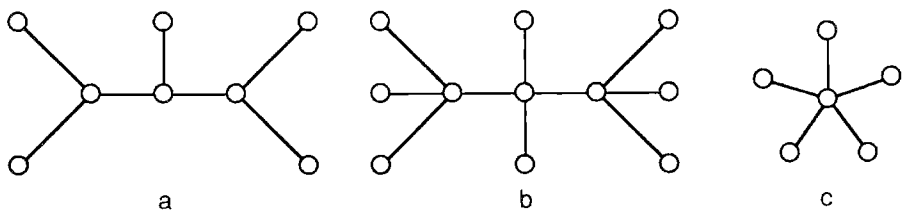


Figure 2. Tree a is a fully binary tree. Tree b is not binary since some internal nodes are of degree greater than three. Tree c is a star tree since it contains only one internal node.

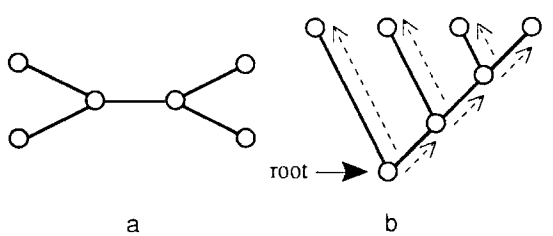


Figure 3. Tree a is unrooted. Tree b is presented in its rooted form.

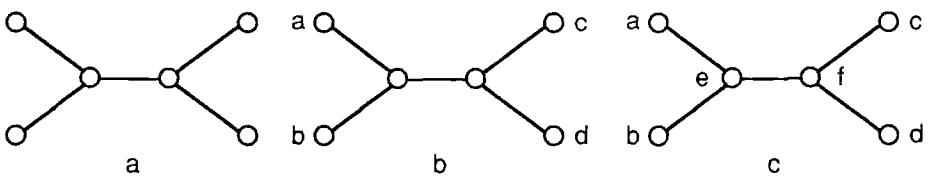
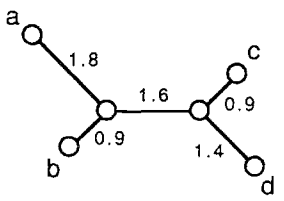


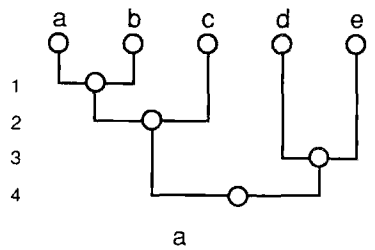
Figure 4. Tree a is unlabeled. Tree b is terminally labeled. Tree c is fully labeled.



	a	b	c
b	2.7		
c	4.3	3.4	
d	4.8	3.9	2.3

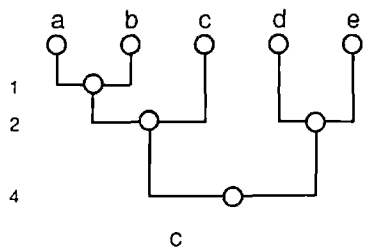
b

Figure 5. A full-binary weighted and terminally labeled additive tree (a) and the corresponding path length matrix (b).



	a	b	c	d
b	1			
c	2	2		
d	4	4	4	
e	4	4	4	3

b



	a	b	c	d
b	1			
c	2	2		
d	4	4	4	
e	4	4	4	2

d

Figure 6. a: Representation of a binary terminally labeled dendrogram with all fusion levels distinct. b: The cophenetic matrix associated to that dendrogram. c: A binary terminally labeled dendrogram with two equal fusion levels (tied values). d: Its cophenetic matrix.

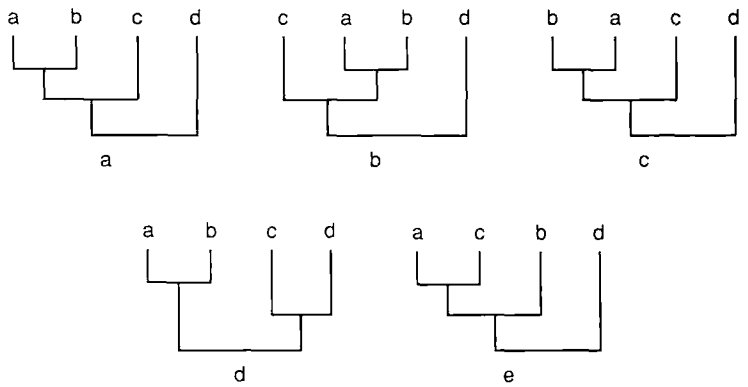


Figure 7. Dendrograms a , b and c are isomorphic, while d and e are distinguishable.

level values instead of ranks can be assigned to the internal nodes of a dendrogram at a level corresponding to the path length between that internal node and its offspring terminal nodes. The root always has the highest fusion value (or highest rank) since it represents the internal node farthest to all pendant vertices. Reversals, therefore, are not legal in dendrograms; that is that if vertex y lies on the path between vertex x and the root, then the height (the "fusion level") of vertex y must be greater than that of vertex x. Like other weighted trees, dendrograms can be binary or not, labeled or unlabeled, but they are always rooted. They are binary when all fusion level values are distinct; this is a sufficient but not necessary condition for a dendrogram to be binary (Figure 6c).

A given dendrogram can be represented in different ways. Not all these representations are seen as different when considering the left-right flipping or pivoting of the vertical branches (Figure 7). We define as *isomorphic* or *symmetrical* a pair of dendrograms that differ only in the pivoting order of their branches (a and b) or their labels (a and c). Likewise, *distinguishable* dendrograms must have distinct topologies (a and d) or different label positions (a and e).

Sibson (1972) has defined two families of dendrograms that he calls *Global-Order Invariant (GOI)* and *Local-Order Invariant (LOI)*. The rooted unranked binary trees studied in the present paper pertain to the *LOI* type (additive trees are thus weighted *LOI* trees) whereas rooted ranked binary trees are of the *GOI* type. Global-order dendrograms with actual fusion level values are *fully-weighted* dendrograms. The term *dendrogram* alone will be used either for *GOI* or for fully-weighted trees, with no distinction.

TABLE 1

The Number of Different Unweighted Binary Trees (BT_n) and Dendrograms (D_n) for a Given Number of Objects n.

n	BT_n	D_n	$D_n - BT_n$
1	1	1	0
2	1	1	0
3	3	3	0
4	15	18	3
5	105	180	75
6	945	2 700	1 755
7	10 395	56 700	46 305
8	135 135	1 587 600	1 452 465
9	2 027 025	57 153 600	55 126 575
10	34 459 425	2 571 912 000	2 537 452 575

3. Dendrograms Versus Binary Trees

The only difference between a rooted binary tree and a dendrogram lies in the fusion level information. Dendrograms have ranked internal nodes whereas binary trees are unranked. Considering that, one easily understands that the number of possible dendrograms differs from the number of binary trees for the same number n of objects. Phipps (1975) and Felsenstein (1978) have shown that the number (BT_n) of unweighted rooted binary trees of order n can be obtained from the formula:

$$BT_n = (2n-3)! / 2^{n-2}(n-2)! \tag{1}$$

On the other hand, Frank and Svensson (1981) have shown that when there are no ties in the fusion levels, the number (D_n) of topologically distinguishable binary dendrograms (i.e., with all fusion levels distinct) of order n is obtained as:

$$Dn = n!\,(n-1)! / 2^{n-1} \tag{2}$$

For a given number of objects, there are many more distinguishable dendrograms, for a given set of fusion levels without ties, than there are non-weighted rooted binary trees (Table 1). The gap becomes more important as the number n of objects increases. With four objects, there are 15 possible rooted binary trees and 18 distinguishable dendrograms (Figure 8). In this

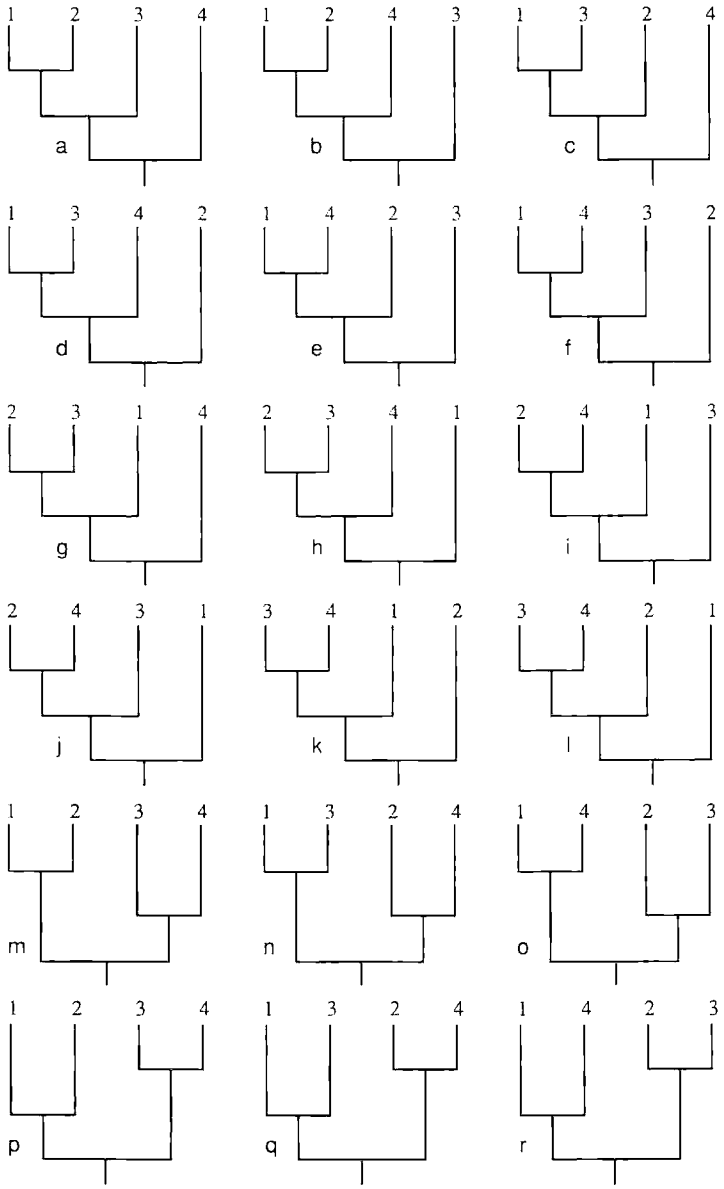


Figure 8. All 18 possible distinguishable dendrograms for 4 objects. Topological Type I is represented by dendrograms a to l. Topological Type II is represented by dendrograms m to r.

example, binary trees corresponding to dendrograms m and p, n and q, and o and r, are not distinct because fusion level heights (or ranks) are not considered in binary trees. That difference has to be taken into account when generating random trees.

Furnas (1984) has published a two-step procedure for the generation of additive trees. His method consists of assigning random lengths to the branches of a previously generated binary tree. This approach has been extended to the generation of dendrograms by assigning random fusion level heights to the vertices, instead of random lengths to the edges (De Soete 1984). Apart from these methods, most of the enumeration and generation procedures proposed so far do not take into account the values of the fusion levels but only consider the rank order of the nodes (Murtagh 1983; Rohlf 1983). We have developed a simple generation procedure for randomizing a fixed set of fusion levels. This is a new way of generating random fully-weighted dendrograms, which are widely used for Monte Carlo simulations in classification studies.

4. Monte Carlo Simulations Involving Random Trees

As we have seen, the number of trees increases rapidly as a function of the number of objects, leading to some problems when the time comes to evaluate the distribution of some type of tree for a large number of objects. Random sampling must be used instead of complete enumeration in these large problems, thus accounting for the large literature on the generation of random trees. One important aspect of the problem deals with the development of algorithms capable of generating random trees equiprobably, drawing them at random from the set of all possible trees, for a given number of objects n. The trees generated must also be relevant to the hypothesis one wishes to test.

The particular problem one often has to deal with in numerical taxonomy is to compare two actual trees on the basis of a consensus index. The null hypothesis may state for instance that the two trees under comparison are as similar as random trees sampled uniformly from the correct distribution (Shao and Rohlf 1983; Shao and Sokal 1986; Page 1988; Lapointe and Legendre 1990). What is the correct distribution? Different questions, related to different kinds of tree, call for different distributions. One may wish to compare labeled or unlabeled, rooted or unrooted trees; one has to decide whether only binary trees, or trees of any other specific type, must be allowed in the generation; is one to consider also the lengths of the branches, and if so, should the trees be additive or ultrametric? Many combinations involving different options are possible. In any case, the comparison scheme is not simple.

For the dendrograms considered in this paper, the derived tree is the result of previous computations and transformations and can be symbolically described as follows:

DerivedDendrogram = ClusteringMethod (Distances (TrueTree) + noise) (3)

Randomization of the derived dendrogram is but one of many other possible ways of looking at the problem. Instead, one might approach the tree comparison by randomizing the raw data with bootstrapping methods (Felsenstein 1985; Nemec and Brinkhurst 1988). The emphasis could also be placed on the distance measure or the clustering method used. There is more than one way to deal with this problem. Each method is based on a specific hypothesis. However, a direct comparison of the actual derived trees may represent the only alternative when no other information is available besides the dendrograms published in the literature. This paper deals with that case.

When comparing actual dendrograms, three levels of information may be of interest. (a) The user may only wish to take the bifurcation pattern into account, which leads to the generation of binary trees by randomization. Furnas (1984) considered at length the generation of such local order invariant trees. (b) One may wish to consider more information, taking into account the ranks of fusion levels in the generation of trees and thus implying the randomization of global order invariant trees, as described by Murtagh (1983). (c) Last, all the information available may be preserved by considering the actual values of the fusion levels instead of their ranks. This third approach is the subject of this paper, which introduces a "metric dimension" not found in either of the other two approaches dealing with local order invariant or global order invariant dendrograms (Sibson 1972). Figure 9 presents some dendrograms that are considered identical, or not, depending on which comparison criterion is used. A binary tree comparison, for instance, would capture no differences among these dendrograms. A global order approach would distinguish a from b but not from c. The metric approach views all three dendrograms as different since identical ranked trees can still have distinct fusion values (e.g., a *versus* c). That this extra information is important seems obvious to us. The final part of this paper proposes a new algorithm designed to generate random dendrograms considering the metric information embedded in the fusion levels.

5. Correspondence Between Dendrograms, Packed Representations, and Cophenetic Matrices

Every dendrogram is composed of a topology — a "shape" *sensu* Harding (1971), which can be labeled in different ways. Shapes representing

unlabeled dendrograms can be defined by a vector of fusion values (or ranks) associated with the internal nodes of a given dendrogram. Only $(n-1)$ values are sufficient and necessary to define this vector, which then represents a "packed representation" of the corresponding topology (Murtagh 1984). Every permutation order of this fusion level vector corresponds to a shape. Shapes, however, are invariant under left-right pivoting of the branches (Harding 1971). There exists indeed a surjective mapping of the set of permutation orders onto the set of shapes. That is that every shape is the image of at least one permutation order but more than one order can represent the same shape. Still, one can generate every random shape by a simple uniform permutation of the fusion level vector (Figure 10). We will also demonstrate below that the precise distribution pattern of these shapes allows the uniform generation of dendrograms when labeling the topologies uniformly.

Labeled dendrograms can be uniquely represented by an ultrametric matrix (Figures 6b and 6d) containing the $n(n-1)/2$ fusions among all pairs of objects of the corresponding tree (Hartigan 1967). There is a one-to-one correspondence between this so-called cophenetic matrix (Sokal and Rohlf 1962) and its associated dendrogram. A matrix is ultrametric when it satisfies the following axioms for all triplets of objects a, b and c (see for example Sneath and Sokal 1973):

Identity	if $a = b$ then $D(a,b) = 0$	(4)
Definiteness	if $a \neq b$ then $D(a,b) > 0$	(5)
Symmetry	$D(a,b) = D(b,a)$	(6)
Ultrametricity	$D(a,b) \leq \max[D(a,c), D(b,c)]$	(7)

where D is some appropriate measure of dissimilarity or distance. With real data, the definiteness condition (expression 5) could be violated when two objects have a zero distance between them, but let us ignore that situation for the sake of the demonstration.

Using the ultrametric property (inequation 7), one can easily build the cophenetic matrix corresponding to a given vector of fusion levels. Or, one can instead generate other random matrices using random fusion values, thus reducing the problem of randomizing dendrograms to the production of random cophenetic matrices, with the constraint that the resulting randomized matrix must still be ultrametric. Both the packed representation and the cophenetic matrix are crucial in the algorithms that follow.

6. Generating Random Dendrograms

Generating random *GOI* dendrograms is like generating ranked binary trees and requires the generation of a tree followed by random assignment of ranks onto the nodes. Our method proceeds in the reverse order, using the

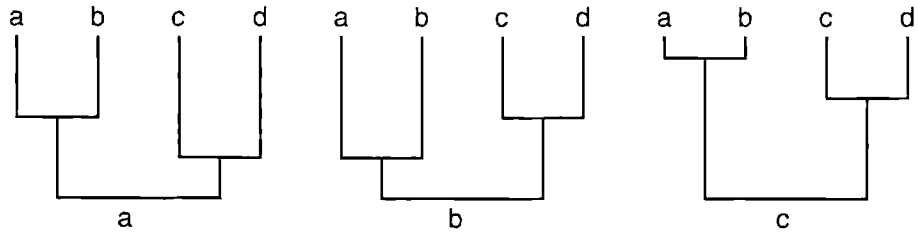


Figure 9. Three dendrograms differing in various aspects.

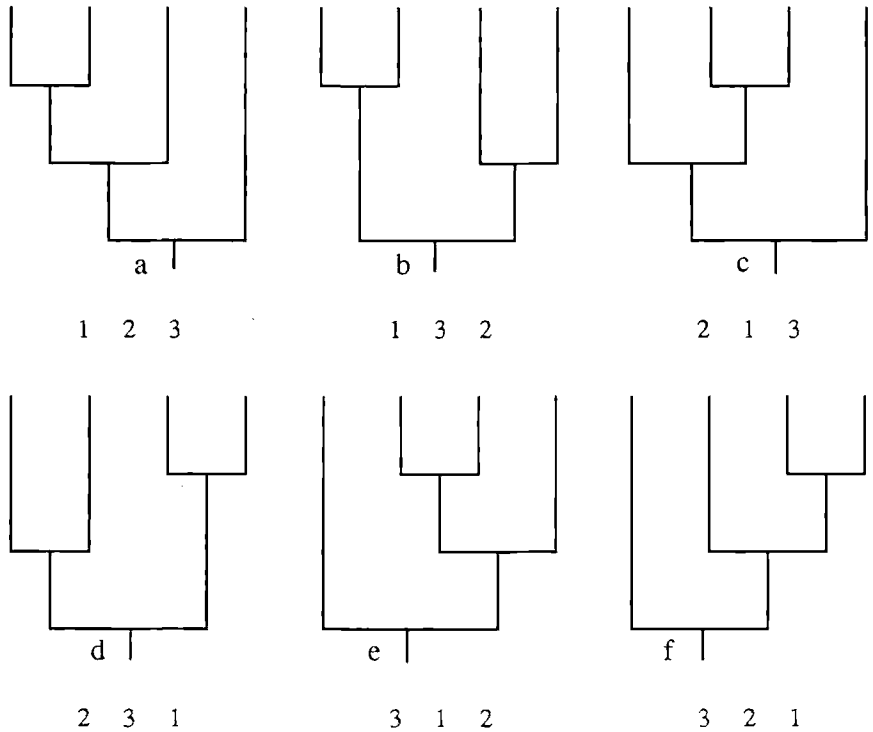


Figure 10. The six topologies resulting from the permutations of the fusion level vector. Topological Type I is represented by the isomorphic dendrograms a, c, e and f. Topological Type II is represented by the isomorphic dendrograms b and d. Packed representations are presented under each tree.

ranks to construct the tree. The procedure implies the generation of a random "shape" followed by random labeling of this shape:

(1) Given a vector containing $n - 1$ ranks, one can construct an unlabeled dendrogram corresponding to that rank order, as explained in Section 7. The ranks specify the merging order of the clusters of the corresponding "shape".
(2) Given a vector containing n labels, the next operation is to assign those labels at random onto the leaves of the unlabeled tree.

This method produces *GOI* dendrograms equiprobably (proof in Section 9). To generate fully-weighted dendrograms, one simply has to assign random fusion values to the nodes of a *GOI* dendrogram constructed using the general algorithm described above, substituting random metric fusion levels for the ranks. A possible alternative is to generate random cophenetic matrices instead of dendrograms.

7. Generating Completely Random Cophenetic Matrices

A "completely random" cophenetic matrix represents a random fully-weighted dendrogram, which is a tree with random fusion level values, random topology, and random position of the labels. To generate such matrices, one has to proceed in three steps: (a) create a random vector of fusion level values (i.e., a packed representation of a random weighted shape); (b) fill the random cophenetic matrix; and finally (c) relabel at random the leaves of that matrix. These steps are easily incorporated into a straightforward algorithm to generate matrices of order n (see Table 2).

Generate random fusion levels — This first operation consists of generating a random vector of fusion levels. Procedure RANDVECT returns a vector containing $n - 1$ random values drawn at random from a uniform distribution.

Filling the cophenetic matrix — Using only the fusion values, one can fill the entire random cophenetic matrix. To do so, we proceed in two steps. First, the random vector of fusion levels is written in the off-diagonal of an empty triangular matrix. From this off-diagonal, the rest of the matrix is filled using the ultrametric property (inequation 7). The FILLMAT procedure executes this operation by repeatedly applying the ultrametric axiom to specific triplets of objects. Both its time and space complexity are $O(n^2)$.

If the user prefers to obtain random fully-weighted dendrograms instead of random cophenetic matrices, it is unnecessary to reconstruct the entire half-matrix. The fusion level vector can lead directly to the dendrogram by specifying the order of the merges. It is essentially like performing a

TABLE 2

Procedures of the Complete and Constrained Randomization Algorithms

```
procedure RANDVECT;
    begin
      for i := 1 to n-1 do
          Vect[i] := Random[1];
    end;

procedure PERMVECT;
    begin
      for i := 1 to n-1 do
          Vectperm[i] := Vect[Address(i)][2]
    end;

procedure FILLMAT;
    begin
    {initialization of the off-diagonal}
      for i := 1 to n-1 do
          Matrix[i,i+1] := Vect[i];
    {fill the matrix}
      for i := 1 to n-2 do
          for j := i+2 to n do
              if Matrix[i,j-1] > Matrix[j-1,j] then
                  Matrix[i,j] := Matrix[i,j-1]
              else
                  Matrix[i,j] := Matrix[j-1,j]
    end;

procedure PERMMAT;
    begin
      for i := 1 to n do
          for j := i+1 to n do
              Matperm[i,j] := Matrix[Address(i),Address(j)][3];
    end;
```

[1] *Random* is a uniform pseudo-random generating function.

[2] *Address* is a randomly permuted vector of the integers 1 to n-1 while *Vect* contains the actual fusion levels of the reference dendrogram. Notice that the pseudo-random number generator is re-seeded after each use, so that the values in PERMVECT have no influence on those in PERMMAT.

[3] *Address* contains a randomly permuted vector of the integers 1 to n (as in the PERMMAT procedure) corresponding to the new positions of the objects in the matrix. In real programs, it is preferable to address the randomly permuted *Address* vector by indirection, instead of actually permuting the matrix.

clustering operation on the matrix, only simpler; it is simpler because one never needs to look outside the off-diagonal to find what to merge next, nor to re-estimate distances to clusters, since the ultrametric inequality is assumed to hold. In that situation, FILLMAT must be replaced by a tree reconstruction procedure. The random generation process is then reduced to $O(n)$ complexity.

Labeling the cophenetic matrix (or the dendrogram) — This operation simply consists of permuting at random the object labels, corresponding to rows and columns of the matrix, to change the positions of the objects in the corresponding dendrogram. The PERMMAT procedure returns a randomly permuted matrix called *Matperm[i,j]*. Alternatively, one can design a PERMLABELS procedure that would execute the same task for dendrograms.

This three-step method can produce "completely random" cophenetic matrices sampled from an infinite population, each representing a random ranked tree in which random heights are assigned to the internal nodes. The procedure therefore might be regarded as an algorithm for generating global-order invariant dendrograms with random fusion level values.

8. Generating Constrained Random Cophenetic Matrices

The algorithm described above can generate a random cophenetic matrix corresponding to a random dendrogram of any order n. This "completely randomized" approach is, however, not suitable when one wishes to build reference distributions of the association between random matrices for the comparison of real dendrograms. The RANDVECT procedure (Table 2) allows all possible fusion values to occur in the generation process. In the case of real dendrograms, this property is not desirable because the various dissimilarity coefficients that are used to construct dendrograms differ in the distributions of their values (Hajdu 1981; Gower and Legendre 1986). When generating fusion values from a uniform pseudo-random number generator, the resulting vector is not comparable to the real set of fusion values in the actual dendrogram.

To overcome the problem of the randomization of fusion values, one might think of generating the fusion levels from a distribution which is relevant to the dissimilarity coefficient that was used to compute the reference dendrogram. This task may be very difficult without any *a priori* information about the underlying distributions of all possible coefficients. An easy way to approach this problem is to constrain the fusion levels of the random vector, forcing them to take the same values as in the actual dendrogram. In that simpler case, a randomization of a cophenetic matrix can be obtained by

randomizing the fusion level *positions*, instead of generating random fusion level *values*. This constrained randomization will insure that the random dendrograms remain comparable to the real one.

The algorithm we propose proceeds by double permutation. First, the actual vector of fusion levels is permuted. This vector is written in the off-diagonal of the random matrix, and the matrix is filled using the FILLMAT procedure. Then, the objects are relabeled using the permuted *Address* vector in the PERMMAT procedure, as above.

The only difference between the complete and the constrained randomization is the replacement of RANDVECT by the PERMVECT procedure (see Table 2), that returns new addresses for the fusion values by permuting the values of the real vector of fusion levels, as read from the cophenetic matrix associated with the actual dendrogram. PERMVECT therefore may be seen as a procedure that permutes the elements of a packed representation, whereas RANDVECT is generating random packed representations.

This constrained randomization algorithm generates random dendrograms from the set of all possible non-isomorphic dendrograms given a fixed vector of fusion levels. When the number of objects n is small, the complete set of dendrograms can be generated instead of a random subset. To do so, one has to change the PERMVECT procedure to a complete enumeration loop that generates all possible permutations of the given vector of fusion levels.

The remainder of this "constrained algorithm" is identical to the "complete approach". We still generate fully-weighted dendrograms. The distinction lies in the fusion values that are not generated randomly but constrained to take fixed values. We are in fact simply permuting the fusion values of a dendrogram. If only ranks were to be considered instead of fusion levels, no difference would exist between the two procedures since the generation of a random rank vector is identical to the permutation of a given set of ranks. The distinction between the two algorithms becomes important here because we are dealing with metric values instead of ranked fusion values (see Figure 9).

9. Example and Justification of the Constrained Randomization

We have seen that in the case of four objects, 18 dendrograms can be distinguished (Figure 8). Let us now verify that the double-permutation procedure allows each of these dendrograms to occur equiprobably given a fixed set of fusion levels.

We already know that the algorithm proceeds in two major steps designed (a) to generate a random topology ("shape") and (b) to relabel that topology randomly. The first step encompasses the PERMVECT and

FILLMAT procedures, and the second step is performed by the PERMMAT procedure.

We will work out in detail the four-object, three-different-fusion-levels case. Suppose that we are dealing with a real vector of fusion levels containing dissimilarity values 1, 2 and 3. The PERMVECT procedure allows 3! = 6 different ways to order these values, corresponding to six unlabeled trees (Figure 10). In reality, only two non-isomorphic topologies are distinguishable among those six because of symmetry of shapes. Type I is represented by dendrograms a, c, e, and f while Type II is represented by dendrograms b and d. The probability of each topological type to occur is 4/6 = 2/3 for Type I and 2/6 = 1/3 for Type II. These values are in agreement with the frequencies of Type I and Type II topologies represented in the 18 dendrograms (Figure 8) that are distinguishable for four objects (Type I = 12/18 = 2/3; Type II = 6/18 =1/3).

Now that we have two different possible unlabeled topologies, we have to label them, that is, to address the positions of the objects on these topologies. This operation is performed by the PERMMAT procedure that allows every permutation of the object order to occur equiprobably. For 4 objects, there are 4! = 24 such different orders. Let us now label both topological types in each of the 24 possible ways. For Type I (Figure 11), we see that 12 dendrograms are distinguishable after the identification of symmetrical trees. Each dendrogram was obtained twice by the labeling operation (a is isomorphic to g; b to h; etc.). All trees have the same probability of occurrence: 2/24 = 1/12. For Type II (Figure 12), fewer dendrograms are possible because each tree has four symmetrical forms (a, b, g, h; etc.). The probability of occurrence of each dendrogram of Type II is then 4/24 = 1/6.

Now, if we combine Figures 11 and 12 to obtain all possible dendrograms, we see that 18 distinguishable forms are produced by the double-permutation procedure; twelve are of Type I and six of Type II. In other words, the probability of each topology to occur being 2/3 and 1/3, we have for Type I, 2/3 x 1/12 = 1/18, and for Type II, 1/3 x 1/6 = 1/18 chances of obtaining one distinguishable dendrogram, the probability being the same for members of both topological types. This example shows that the double-permutation procedure allows every non-isomorphic dendrogram of Figure 8 to occur equiprobably. Higher numbers of objects n could be analyzed in the same way.

Let us now demonstrate that this example can be extended to a general theorem.

Theorem: *Given any number of objects* n, *the probability of each distinguishable dendrogram of order* n *to occur is the same for every topological type.*

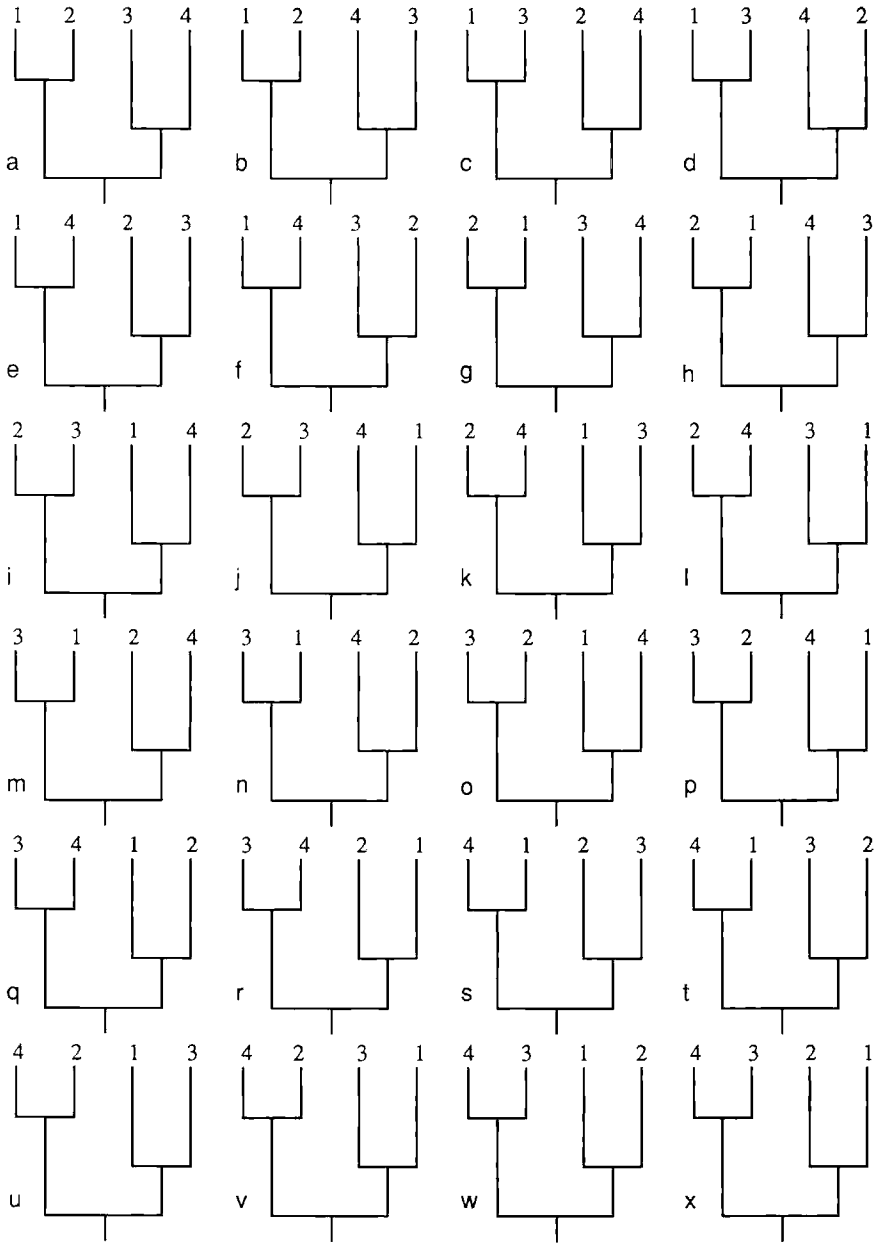


Figure 11. All possible orders of the labels for topological Type I.

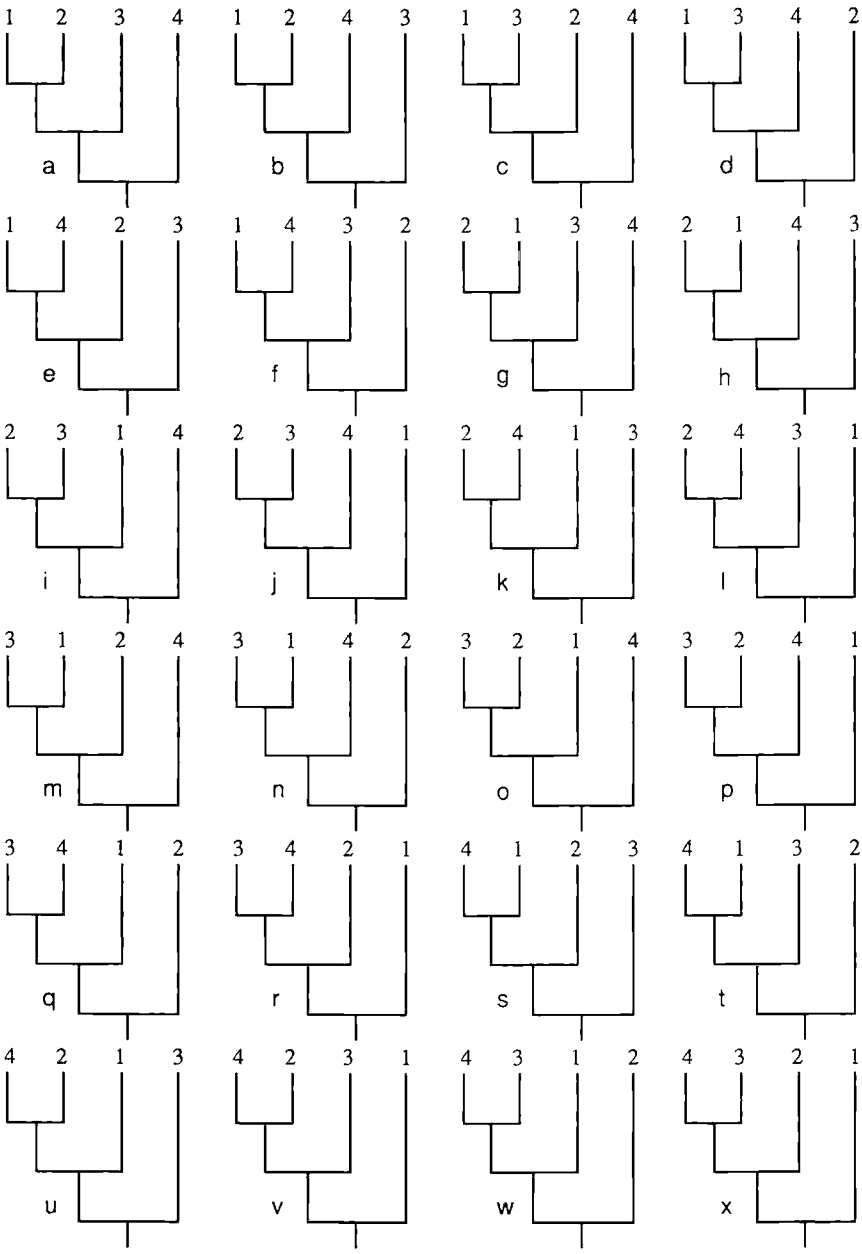


Figure 12. All possible orders of the labels for topological Type II.

Corollary: *Our algorithm is a Uniform Random Generation Algorithm,* sensu *Furnas (1984).*

Proof: If this is true, each dendrogram should be generated with a probability corresponding to $1 / D_n$ where D_n is given by equation (2). Thus,

$$1 / D_n = 1 / [n! (n-1)! / 2^{n-1}] = 2^{n-1} / (n! (n-1)!) \tag{8}$$

The double-permutation algorithm proceeds in two steps. The first permutation is applied to the vector of fusion level values (or ranks) containing $n-1$ elements. This permutation allows the $(n-1)!$ distinct orders of the vector to occur equiprobably. Some permutations of these packed representations represent identical topological types or shapes, however. Each distinguishable unlabeled tree may correspond to different permutation orders. Murtagh (1984) has defined a recurrence function to calculate the number of isomorphic unlabeled dendrograms of order n containing m internal nodes ($m \leq n / 2$) with exactly two offspring terminal nodes. The number of distinct topological types resulting from the permutation of the fusion level vector is a function of the value m. One can show that for any topology (T_1) of order n with m internal nodes with exactly two offspring terminal nodes, the number of isomorphic permutations is equal to:

$$T_1(n,m) = 2^{n-m-1} \tag{9}$$

In this formula, $(n-m-1)$ is the number of fusion levels that can be permuted, once those levels (m) that have exactly two terminal nodes have been excluded. Since each of these $(n-m-1)$ fusion levels is binary, only two alternatives are possible by permuting its two derived vertical branches; this is why 2^{n-m-1} gives the total number of isomorphic permuted packed representations of the corresponding shapes. Therefore, the probability (P) of each non-isomorphic unlabeled dendrogram is given by:

$$P\,[T_1(n,m)] = 2^{n-m-1} / (n-1)! \tag{10}$$

which represents the first step of the double-permutation procedure.

Once the first permutation is completed, a second permutation is performed, this time on the labels for the leaves of the tree. This operation allows $n!$ different orderings of the labels to occur equiprobably. Similar to the packed representations, some label permutations represent isomorphic labeling of the dendrograms. The number of such isomorphic orders (T_2) for a given dendrogram with n leaves is also a function of m:

$$T_2(n,m) = 2^m \tag{11}$$

The probability (P) of each distinguishable ordering of the labels for a given shape is:

$$P\,[T_2(n,m)] = 2^m / n! \tag{12}$$

That corresponds to the final step of the constrained randomization algorithm.

The combination of equations 10 and 12 represents the probability of each dendrogram T of order n with m nodes, each with exactly two terminal descendant nodes:

$$\begin{aligned} P\,[T(n,m)] &= P\,[T_1(n,m)] \times P\,[T_2(n,m)] \\ &= 2^{n-m-1} / (n-1)! \times 2^m / n! \\ &= 2^{n-1} / ((n-1)!\, n!) \end{aligned} \tag{13}$$

which is equal to the value for $1 / D_n$ obtained from Equation 8. Thus, the double-permutation algorithm generates each distinguishable labeled dendrogram with a probability equal to $1 / D_n$ and the procedure is a Uniform Random Generation Algorithm. •

10. Conclusion

This paper presented two algorithms that are useful for comparing dendrograms. Both methods are generating global order dendrograms equiprobably. The first algorithm generates dendrograms with random fusion values, whereas the second procedure generates dendrograms with fixed fusion levels.

The "completely random" algorithm should be used in Monte Carlo studies where a distribution of some statistic based on completely randomized trees is needed. Studies of that type include the papers by Shao and Rohlf (1983) and Shao and Sokal (1986) who generated reference distributions for testing the significance of consensus indices between trees. These authors used trees without specified fusion levels. Using our first algorithm, similar distributions could now be worked out for trees with specified fusion levels (i.e., fully-weighted dendrograms).

Our second method, the "constrained algorithm", is intended to answer a different type of problem, that is, the comparison of two dendrograms obtained *independently*. Such problems are found in all fields where classifications are used: evolutionary biology, biogeography, ecology,

sociometry, psychometrics, econometrics, and so on. One of these dendrograms is usually obtained from clustering data, while the second one may either be known *a priori*, or, like the first dendrogram, be the result of a cluster analysis. Such comparisons usually imply the computation of some form of consensus or agreement measure. When a test is needed of the statistical significance of the agreement, the reference distribution can be generated after repeated random permutations of the dendrograms, followed by recomputation of the agreement measure. Examples include Lapointe and Legendre (1990), based on a modified form of the Faith and Belbin (1986) consensus index, and Page (1990). When the agreement statistic chosen varies as a function of the fusion levels, and not only as a function of the tree topology and position of the leaves, then one might choose to limit the possible fusion levels to those found in the real dendrogram being permuted, in order to eliminate that effect from the reference distribution, hence from the statistical test. If the chosen agreement measure varies only as a function of the tree topology and leaf positions, as it is the case for instance with the consensus-fork index (Colless 1980) and Mickevich's (1978) index, then either one of the algorithms presented in this paper would be adequate for generating the random trees.

Both methods described in this paper can be modified to allow the generation of random additive trees (Lapointe and Legendre, submitted). Completely random path length matrices can be generated by the combination of random ultrametric (Figure 6) and star (Figure 2c) components. One could also use a constrained approach in that case to obtain relevant additive trees for comparison purposes. Further developments in that area are badly needed.

References

COLLESS, D. H. (1980), "Congruence Between Morphometric and Allozyme Data for *Menidia* Species: a Reappraisal," *Systematic Zoology, 29*, 288-299.

DE SOETE, G. (1984), "Ultrametric Tree Representations of Incomplete Dissimilarity Data," *Journal of Classification, 1*, 235-242.

FAITH, D. P., and BELBIN, L. (1986), "Comparison of Classifications Using Measures Intermediate Between Metric Dissimilarity and Consensus Similarity," *Journal of Classification, 3*, 257-280.

FELSENSTEIN, J. (1978), "The Number of Evolutionary Trees," *Systematic Zoology, 27*, 27-33.

FELSENSTEIN, J. (1985), "Confidence Limits on Phylogenies: An Approach Using the Bootstrap," *Evolution, 39*, 783-791.

FRANK, O., and SVENSSON, K. (1981), "On Probability Distributions of Single-Linkage Dendrograms," *Journal of Statistics and Computer Simulation, 12*, 121-131.

FURNAS, G. W. (1984), "The Generation of Random, Binary Unordered Trees," *Journal of Classification, 1*, 187-233.

GÖBEL, F. (1980), "On a 1-1 Correspondence Between Rooted Trees and Natural Numbers," *Journal of Combinatorial Theory, Series B, 29*, 141-143.

GOWER, J. C., and LEGENDRE, P. (1986), "Metric and Euclidean Properties of Dissimilarity Coefficients," *Journal of Classification, 3*, 5-48.

GUENOCHE, A. (1983), "Random Spanning Trees," *Journal of Algorithms, 4*, 214-220.

HAJDU, L. J. (1981), "Graphical Comparison of Resemblance Measures in Phytosociology," *Vegetatio, 48*, 47-59.

HARDING, E. F. (1971), "The Probabilities of Rooted Tree-Shapes Generated by Random Bifurcation," *Advances in Applied Probability, 3*, 44-77.

HARTIGAN, J. A. (1967), "Representation of Similarity Matrices by Trees," *Journal of the American Statistical Association, 62*, 1140-1158.

KNOTT, G. D. (1977), "A Numbering System for Binary Trees," *Communication of the Association for Computing Machinery, 20(2)*.

LAPOINTE, F.-J., and LEGENDRE, P. (1990), "A Statistical Framework to Test the Consensus of Two Nested Classifications," *Systematic Zoology, 39*, 1-14.

LAPOINTE, F.-J., and LEGENDRE, P. (submitted), "A Statistical Framework to Test the Consensus among Additive Trees (Cladograms)."

MICKEVICH, M. F. (1978), "Taxonomic Congruence," *Systematic Zoology, 27*, 143-158.

MURTAGH, F. (1983), "A Probability Theory of Hierarchic Clustering Using Random Dendrograms," *Journal of Statistics and Computer Simulation, 18*, 145-157.

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

NEMEC, A. F. L., and BRINKHURST, R. O. (1988), "Using the Bootstrap to Assess Statistical Significance in the Cluster Analysis of Species Abundance Data," *Canadian Journal of Fisheries and Aquatic Sciences, 45*, 965-970.

NIJENHUIS, A., and WILF, H. S. (Eds.) (1978), *Combinatorial Algorithms for Computers and Calculators*, Second Edition, New York: Academic Press.

ODEN, N. L., and SHAO, K. T. (1984), "An Algorithm to Equiprobably Generate All Directed Trees With k Labeled Terminal Nodes and Unlabeled Interior Nodes," *Bulletin of Mathematical Biology, 46*, 379-387.

PAGE, R. D. M. (1988), "Quantitative Cladistic Biogeography: Constructing and Comparing Area Cladograms," *Systematic Zoology, 37*, 254-270.

PAGE, R. D. M. (1990), "Temporal Congruence and Cladistic Analysis of Biogeography and Cospeciation," *Systematic Zoology, 39*, 205-226.

PHIPPS, J. B. (1975), "The Numbers of Classifications," *Canadian Journal of Botany, 54*, 686-688.

PROSKUROWSKI, A. (1980), "On the Generation of Binary Trees," *Journal of the Association for Computing Machinery, 27*, 1-2.

QUIROZ, A. J. (1989), "Fast Random Generation of Binary, t-ary and Other Types of Trees," *Journal of Classification, 6*, 223-231.

ROHLF, F. J. (1983), "Numbering Binary Trees With Labeled Terminal Vertices," *Bulletin of Mathematical Biology, 45*, 33-40.

ROSEN, D. E. (1978), "Vicariant Patterns and Historical Explanation in Biogeography," *Systematic Zoology, 27*, 159-188.

ROTEM, D., and VAROL, Y. L. (1978), "Generation of Binary Trees from Ballot Sequences," *Journal of the Association for Computing Machinery, 25*, 396-404.

SHAO, K., and ROHLF, F. J. (1983), "Sampling Distributions of Consensus Indices when all Bifurcating Trees are Equally Likely" in *Numerical Taxonomy*, Ed., J. Felsenstein, NATO Advanced Studies Institute, Ser. G. (Ecological Sciences) 1, Berlin: Springer Verlag, 132-137.

SHAO, K., and SOKAL, R. R. (1986), "Significance Tests of Consensus Indices," *Systematic Zoology, 35*, 582-590.

SIBSON, R. (1972), "Order Invariant Methods for Data Analysis," *Journal of the Royal Statistical Society, B34*, 311-349.

SNEATH, P. H. A., and SOKAL, R. R. (1973), *Numerical Taxonomy*, San Francisco: Freeman.

SOKAL, R. R. and ROHLF, F. J. (1962), "The Comparison of Dendrograms by Objective Methods," *Taxon, 11*, 33-40.

SOLOMON, M., and FINKEL, R. A. (1980), "A Note on Enumerating Binary Trees," *Journal of the Association for Computing Machinery, 27*, 3-5.