3. Cluster analysis

3.1. Overview

Clustering requires the recognition of discontinuous subsets in an environment that is sometimes discrete (as in taxonomy), but most often continuous in ecology. To cluster is to recognise that objects are sufficiently similar to be put in the same group, and also to identify distinctions or separations between groups. The present chapter discusses methods used to decide whether objects are similar enough to be allocated to a group.

Clustering is an operation of multidimensional analysis which consists in partitioning the collection of objects (or descriptors in R mode) in the study. A *partition* is a division of a set (collection) into subsets, such that each object or descriptor belongs to one and only one subset for that partition (for instance, a species cannot belong simultaneously to two genera!). Depending on the clustering model, the result can be a single partition or a series of hierarchically nested partitions.

Note that the large majority of clustering techniques work on association matrices, which stresses the importance of the choice of an appropriate association coefficient.

One can classify the families of clustering methods as follows:

1. *Sequential or simultaneous algorithms*. Most methods are sequential and consist in the repetition of a given procedure until all objects have found their place: progressive division of a collection of objects, or progressive agglomeration of objects into groups. The less frequent simultaneous algorithms, on the contrary, find the solution in a single step.

2. *Agglomerative or divisive.* Among the sequential algorithms, agglomerative procedures begin with the discontinuous collection of objects, that are successively grouped into larger and larger clusters until a single, all-encompassing cluster is obtained. Divisive methods, on the contrary, start with the collection of objects considered as one single group, and divide it into subgroups, and so on until the objects are completely separated. In either

case it is left to the user to decide which of the intermediate partition is to be retained, given the problem under study.

3. *Monothetic versus polythetic.* Divisive methods may be monothetic or polythetic. Monothetic methods use a single descriptor (the one that is considered the best for that level) at each step for partitioning, whereas polythetic methods use several descriptors which, in most cases, are combined into an association matrix.

4. *Hierarchical versus non-hierarchical methods.* In hierarchical methods, the members of inferior-ranking clusters become members of larger, higher-ranking clusters. Most of the time, hierarchical methods produce non-overlapping clusters. Non-hierarchical methods (including the *K*-means method exposed in Section 3.7) produce one single partition, without any hierarchy among the groups. For instance, one can ask for 5 or 10 groups for which the partition optimises the intragroup homogeneity.

5. *Probabilistic versus non-probabilistic methods.* Probabilistic methods define groups in such a way that the within-group association matrices have a given probability of being homogeneous. Sometimes used to define species associations.

3.2. Single-linkage agglomerative clustering

Also called *nearest neighbour clustering*, this method is sequential, agglomerative, polythetic, hierarchical and non-probabilistic, like most of the methods that will be presented here. Based on a matrix of similarities or distances, it proceeds as follows (see example further down):

1. The matrix of association is rewritten in decreasing order of similarities (or increasing order of distances).

2. The clusters are formed hierarchically, starting with the two most similar objects (first row of the rewritten association matrix). Then the second row forms a new group (if it contains two new objects) or aggregates itself to the first group (if one of the objects is a member of the first group formed above), and so on. The objects aggregate and the size of the groups increases as the similarity criterion relaxes.

Table IV (below) is a matrix of Euclidean distances (D_1) among five fictitious objects and will be the base for the clustering examples.

Table IV - D_1 association matrix among five objects

	2	3	4	5
1	0.20	0.25	0.45	0.80
2		0.40	0.35	0.50
3			0.30	0.60
4				0.70

First step of the single linkage clustering: the association matrix (Table IV) is rewritten in order of increasing distances:

D_1	Pairs of objects formed
0.20	1 - 2
0.25	1 - 3
0.30	3 - 4
0.35	2 - 4
0.40	2 - 3
0.45	1 - 4
0.50	2 - 5
0.60	3 - 5
0.70	4 - 5
0.80	1 - 5

Second step: the groups are formed by extending the distance progressively:

a. First group to be formed: pair 1 - 2, distance 0.2.

b. Object 3 rejoins the group above at distance 0.25.

c. Object 4 rejoins the group above at distance 0.30.

d. Object 5 rejoins the group above at distance 0.50.

The qualifier *single* linkage clustering comes from the fact that the fusion of an object (or a group) with a group at a given similarity (or distance) level only needs that *one* object of each of the two groups about to agglomerate be linked to one another at this level. We shall see in Section 3.3 that, at the opposite of the spectrum, complete linkage clustering demands, for the two groups to agglomerate, that *all* objects be related at the given similarity.

The result of a hierarchical clustering is generally presented in the form of a **dendrogram**. The dendrogram resulting from the example above is the following (Figure 13):

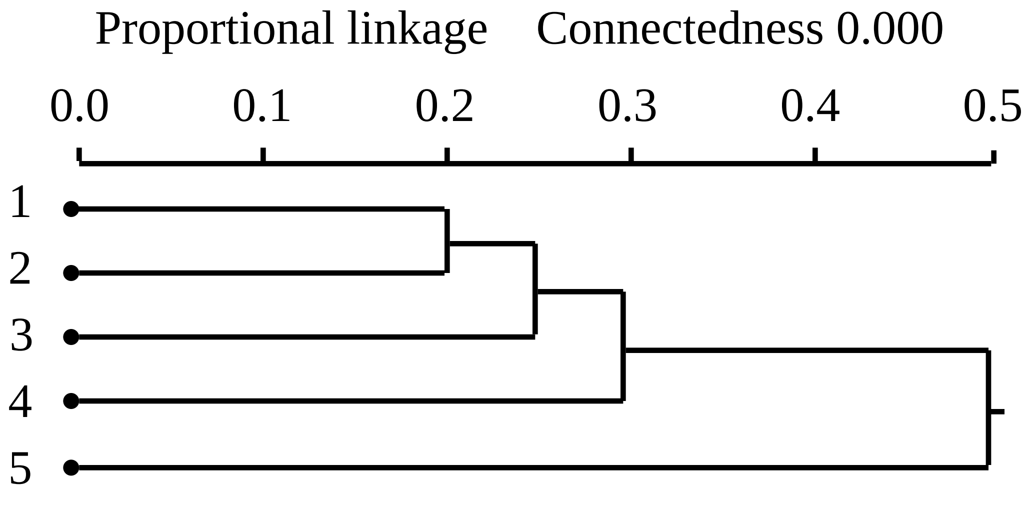

Figure 13: Dendrogram of the single linkage clustering of the data shown in Table IV. The scale represents Euclidean distances (the coefficient used in the association matrix) “Proportional linkage” and “connectedness”: see text.

3.3. Complete linkage agglomerative clustering

Contrary to single linkage clustering, complete linkage clustering (also called *furthest neighbour sorting*) allows an object (or a group) to agglomerate with another group only at a similarity corresponding to that of the most distant pairs of objects (thus, *a fortiori*, all members of both groups are linked).

The procedure and results for Table IV data are as follows:

First step: the association matrix is rewritten in order of increasing distances (same as single linkage).

Second step: agglomeration based on the criteria exposed above:

a. First group to form: pair 1 - 2, distance 0.20.

b. A second group forms, independent of the fist: pair 3 - 4, distance 0.30. Indeed, none of objects 3 or 4 is at a *shorter* distance than 0.30 from the *furthest* member of group 1 - 2 (object 3 is at 0.25 of object 1, but at 0.40 of object 2).

c. Fusion of the two pairs formed above (1-2 and 3-4) can occur only at the distance separating the members that are furthest apart. Here this distance is 0.45 (between objects 1 and 4). The two groups join at this level (since no external object is closer to one of the groups than 0.45).

d. Object 5 can join the group only at the distance of the member that is furthest from it, i.e. 0.80 (distance between object 5 and object 1).

The look of the resulting dendrogram (Figure 14) is quite different from the previous one. The clustering rules have affected not only the distances among objects but also the topology of the dendrogram:

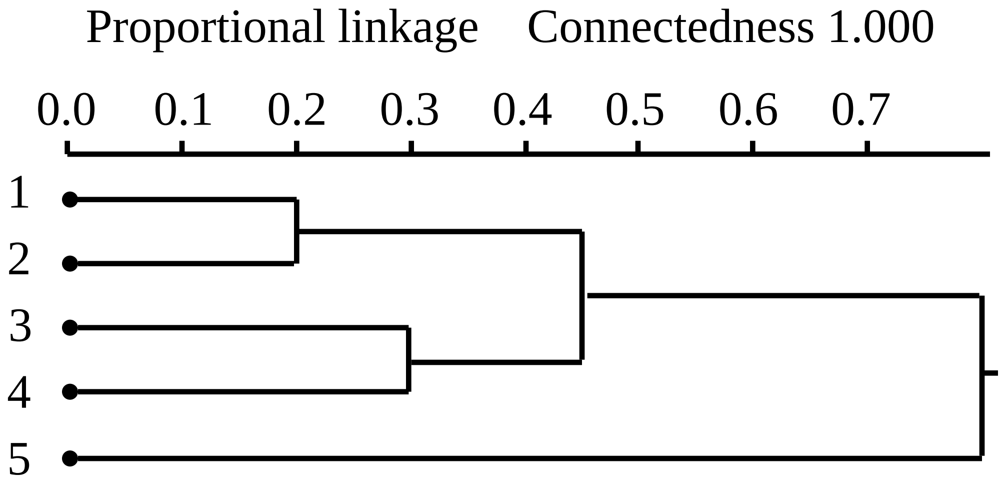

Figure 14: Dendrogram of the complete linkage clustering of the data shown in Table IV.

The comparison between the two dendrograms shows the difference in the philosophy and the results of the two methods: **single linkage** allows an object to agglomerate easily to a group, since a link to one single object of the group suffices to induce the fusion. This is a "closest friend" procedure, so to say. As a result, single linkage clustering has a tendency to produce a **chaining** of objects: a pair forms, then an objects rejoins the pair, and another, and so on. The resulting dendrogram does not show clearly separated groups, but can be used to identify **gradients** in the data.

At the opposite, **complete linkage** clustering is much more constraining (and **contrasting**). A group admits a new member only at a distance corresponding to the furthest object of the group: one could say that the admission requires unanimity of the members of the group! It follows that, the larger a group is, the more difficult it is to agglomerate with it. Complete linkage, therefore, tends to produce **many small groups** separately, that agglomerate at large distances. Therefore, this method is interesting to look for discontinuities in data that are *a priori* quite compact. In other words, single linkage clustering contracts the reference space around a cluster, while complete linkage clustering expands it.

3.4. Intermediate linkage clustering

This expression includes all the intermediates between the above extremes, i.e. algorithms where group fusion occurs when a definite proportion of links is established between the members of the two groups. This proportion is called the **connectedness**. Connectedness varies from 0 (single linkage) to 1 (complete linkage). Often in ecology, appropriate solutions are found in intermediate connectednesses (0.3 to 0.7), where the clustering algorithm approximately conserves the metric properties of the reference space.

The study of this family of linkage techniques shows that it has a great flexibility. This quality could lead the reader to think that one can impose one's preconcieved ideas on the data. In reality you must remember the following points:

1. It is preferable to define what you expect from a clustering *before* running the computation. To show a possible gradient? To reveal faint discontinuities? An intermediate, "neutral" clustering?

2. Whatever the method chosen is, the structures revealed indeed exist in the *association matrix*. Even a complete linkage clustering will not produce small, compact groups from an association matrix describing only a strong gradient, and the converse is true also.

Therefore, it is extremely important that one chooses the appropriate association coefficient and the appropriate clustering method to extract the desired information from the data.

3.5. Average agglomerative clustering

The four methods of this family are commonly used in numerical taxonomy (a bit less in ecology). Their name in this discipline are mentioned in parentheses in Table V below. These methods are not based on the number of links between groups or objects, but rather on average similarities among objects or on centroids of clusters. The difference among them pertains to the way of computing the position of the groups (arithmetic average versus centroids) and to the weighting or non-weighting of the groups according to the number of objects that they contain. Table V summarises these features:

Table V - The four methods of average agglomerative clustering

	Arithmetic average	Centroid clustering
Equal weights	Unweighted arithmetic average clustering (UPGMA)	Unweighted centroid clustering (UPGMC)
Unequal weights	Weighted arithmetic average clustering (WPGMA)	Weighted centroid clustering (WPGMC)

Unweighted arithmetic average clustering (UPGMA)

Also called *group average sorting* and ***U**nweighted **P**air-**G**roup **M**ethod using **A**rithmetic averages)*, this technique must be applied with caution: because it gives equal weights to the original similarities, it assumes that the objects in each group form a representative sample of the corresponding larger groups of objects in the reference population under study. For this reason, UPGMA clustering should only be used in connection with simple random or systematic sampling designs *if the results are to be extrapolated to a larger reference population.*

UPGMA allows an object to join a group at the **average** of the distances between this object and all members of the group. When two groups join, they do it at the average of the distances between all members of one group and all members of the other. This gives, using our example (Table IV):

- objects 1 and 2 join at 0.20;

- object 3 is at distance 0.25 with 1, and 0.40 with 2. The average of these distances is 0.325, i.e., larger than the distance between objects 3 and 4 (0.30). Therefore, the latter join at distance 0.30 as a distinct group;

- object 5 being very far, the two groups 1 - 2 and 3 - 4 join at the average of the inter-group distances, i.e. $[D_1(1\text{-}3) + D_1(1\text{-}4) + D_1(2\text{-}3) + D_1(2\text{-}4)]/4 = (0.25+0.45+0.40+0.35)/4 = 0.3625$;

- similarly, object 5 joins the group at the average of its distances with all the members of the group, i.e. (0.50+0.60+0.70+0.80)/4 = 0.65.

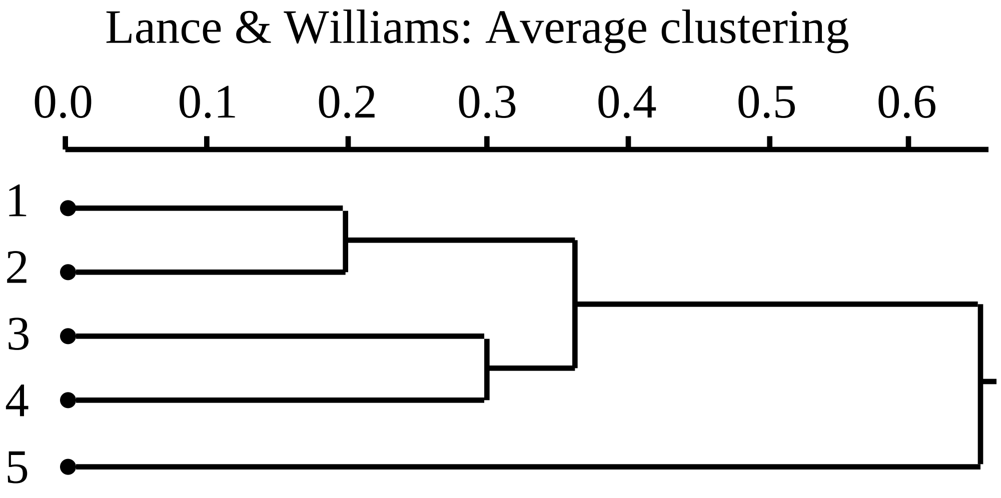

Figure 15: Dendrogram of the UPGMA clustering of the data shown in Table IV.

Unweighted centroid clustering(UPGMC)

The same caveat as in UPGMA, about the representativeness of the sample, applies to UPGMC.

In a cluster of points, the centroid is the point that has the average coordinates of all the objects of the cluster. UPGMC joins the objects or groups that have the highest similarity (or the smallest distance), *by* ***replacing all the objects*** *of the group produced* ***by the centroid*** *of the* *group*. This centroid is considered as a single object at the next clustering step.

A simple manner to achieve this is to replace, in the similarity matrix, the two rows and columns corresponding to the two objects about to join by a single series obtained by computing the averages of the similarities of the two objects with all the others. Presently however, one uses a slightly more complex formula, that is given by Legendre & Legendre (1998) p. 322.

The dendrogram of the UPGMC clustering of our example data has the following aspect (Figure 16):

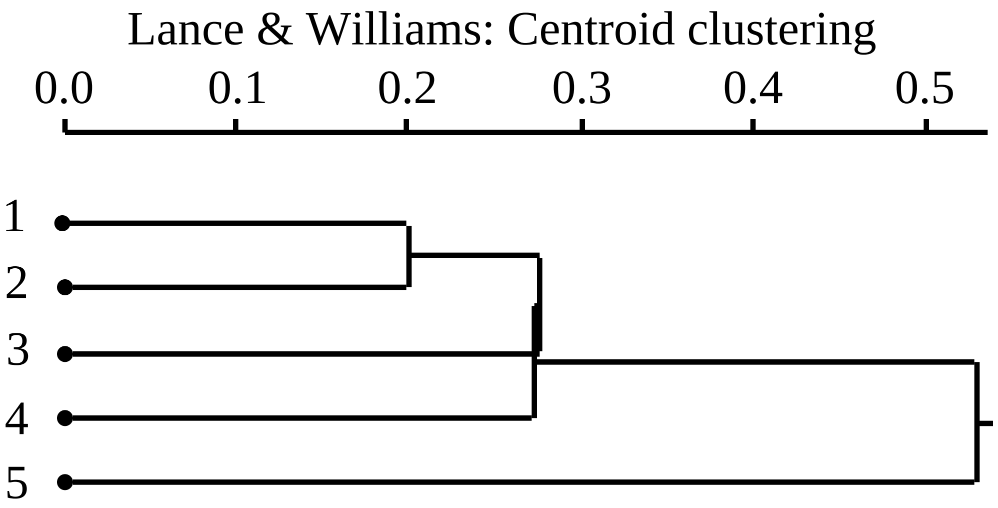

Figure 16: Dendrogram of the UPGMC clustering of the data shown in Table IV, showing a reversal.

UPGMC, as well as WPGMC, can sometimes produce *reversals* in the dendrogram. This situation occurred in our example. This happens when:

1. Two objects about to join (let us call them A and B) are closer to one another than each of them is to a third object C: AB<AC ; AB<BC.

2. After the fusion of A and B, the centroid of the new group A-B is closer to C than A was to B before the fusion: $C_{AB}C<AB$.

This result is due to a violation of the ultrametric property, that states that a distance between two objects A and B must be smaller than or equal to the maximal distance between A and a third object C and B and C: $D_{(A,B)} \leq \text{Max} \mid D_{(A,C)} , D_{(B,C)} \mid$. See Legendre & Legendre (1998) p. 324 for further explanations.

This dendrogram is tricky to interpret. In fact, one cannot consider it as a classification *sensu stricto*.

WPGMA and WPGMC

The weighted counterparts of the two methods above, i.e. WPGMA and WPGMC, are not detailed here. They can be used in cases where groups of objects representing different situations (and thus likely to form different groups) are represented by unequal numbers of objects. In these cases the two unweighted methods above may be distorted when a fusion of a large and a small group of objects occurs. The solution consists in giving equal weights, when computing fusion similarities, to the two *branches* about to fuse.

3.6 Ward's minimum variance clustering method

This method is related to UPGMC and WPGMC: cluster centroids play an important role. The objective is to define groups in such a way that the within-group sum of squares (i.e., the squared error of ANOVA) is minimized.

At the beginning, the *n* objects each form a cluster. Therefore, sum of squared distances between objects and centroids is 0. As clusters form, the centroids move away from actual object coordinates and the **sum of the squared distances from the objects to the centroids** increase.

The sum of squared distances is the same quantity as tho one called "error" in ANOVA. At each clustering step, Ward's method finds the pair of objects or clusters whose fusion increases as little as possible the sum, over all objects, of the squared distances between objects and cluster centroids. The within-cluster sum of squared errors can be computed either from the **raw data**, or as the **mean of the squared distances among cluster members**. Therefore, Ward's method can be applied to **raw data** or to **distance matrices**. In the latter case the Ward method, originally based on the Euclidean distance, can be extended to any distance coefficient.

Dendrograms can be represented using various scales without affecting the topology:

- squared distances;
- square root of the fusion distances (removes the distorsion created by squaring the distances). Used in the R package of Legendre, Casgrain & Vaudor;
- sum of squared errors.

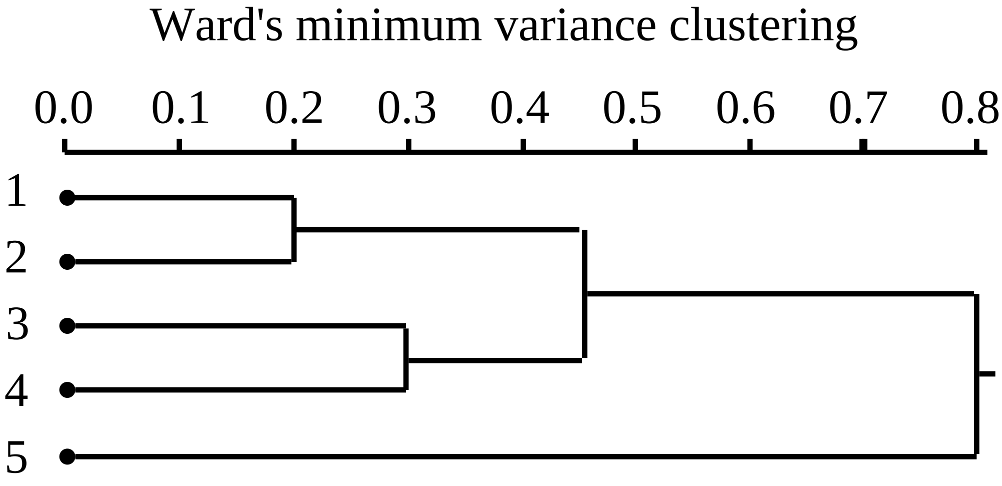

Figure 16b - Result of a Ward clustering on the distance matrix of 5 objects of Table IV. Scale: square root of the fusion distances.

3.7 Partitioning by *K*-means

Partitioning is finding a single partition of a set of objects. The problem has been stated as follows: given *n* objects in a *p*-dimensional space, determine a partition of the objects into *K* groups, or clusters, such as the objects within each cluster are more similar to one another than to objects in the other clusters. The number of groups, *K*, is determined by the user. The *K*-means method uses the *local structure* of the data to delineate clusters: groups are formed by identifying high-density regions in the data. To achieve this, the method iteratively minimises an objective function called the total error sum of squares (E^2_K or TESS). This quantity is the sum, over the *K* groups, of the means of the squared distances among objects in their respective groups. This reminds Ward's clustering, but note that *K*-means partitioning is a *divisive* method while Ward's is agglomerative.

To begin the computation, one has to provide an *initial configuration*, i.e. to assign the objects to the *K* groups as a starting point to the optimisation process. This initial configuration can be random (this is generally offered by the computer programs running *K*-means), or it can be, for instance, a partition derived from a hierarchical clustering computed on the same data, or else it may be provided by an ecological hypothesis.

Some programs ask for "group seeds", i.e. objects around which to start the clustering process. Whenever possible, it is recommended to choose objects that are as close as possible to the expected centroids of the final groups. In a closely related method (partitioning around medoids, PAM) these group seeds are called "medoids".

A problem often encountered in such iterative algorithms, and existing here, is that the final solution somewhat depends on the initial configuration, since the iterative procedure may encounter a local minimum during the process. This is why one often tries several runs, each one based on a different initial configuration, to retain the one that yields the lowest TESS at the end of the process.

K-means partitioning may be computed from either a raw data table or a distance matrix (since TESS can be computed directly from distances among objects). If one wishes to use *K*-means on species abundance data, one has to compute a distance matrix using an appropriate, asymmetrical measure (see Chapter 2). If the computation is run from the raw data table, then the double zeros are counted as resemblances among objects, which is inappropriate. An alternative is to pre-transform the species data as shown in Section 4.3 below before running *K*-means on the matrix of objects by (transformed) species.

This overview does by far not show all the clustering methods available. But it shows that numerous approaches exist, that these methods address different questions, focus on different aspects of the data and therefore do not necessarily yield the same results. The choice depends on the researcher's aims.

3.8 Interpretation of clustering results

By definition, hierarchical clustering methods yield trees (usually represented as dendrograms). Starting from separate objects, these dendrograms show a series of fusions ultimately leading to one single group containing all the objects. Therefore the question arises: where should one "cut the tree", i.e. what is the optimal number of clusters in a tree?

The short answer is: there is no unique solution. Unconstrained clustering is a descriptive, heuristic technique. So the general goal is to retain as many clusters as can be interpreted in the light of the problem under investigation. Note that this implies that not all branches need to be cut at the same level.

In some instances, however, one wishes to apply an objective approach to the determination of the number of clusters. This is usually the case, for instance, when one wants to compare several clusterings. Several possibilities exist. For example:

1. Use a graph of the *fusion levels* to locate the levels where a large distance exists between two consecutive fusions, and cut the whole tree there.

2. At each fusion level, compute *silhouettes* measuring the intensity of the link of the objects to their groups, and choose the level where the within-group mean intensity is highest (largest "silhouette width").

3. Compare the original distance matrix to a binary matrix built from the clustering tree (cut at various levels), and choose the level where the (Mantel) correlation between the two is highest.