

# MultiDendrograms: variable-group agglomerative hierarchical clusterings

Sergio Gómez<sup>a,\*</sup>, Justo Montiel<sup>a</sup>, David Torres<sup>a</sup>, Alberto Fernández<sup>b</sup>

<sup>a</sup>*Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Spain*

<sup>b</sup>*Departament d'Enginyeria Química, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Spain*

---

## Abstract

*MultiDendrograms* is a Java-written application that computes agglomerative hierarchical clusterings of data. Starting from a distances (or weights) matrix, *MultiDendrograms* is able to calculate its dendrograms using the most common agglomerative hierarchical clustering methods. The application implements a variable-group algorithm that solves the non-uniqueness problem found in the standard pair-group algorithm. This problem arises when two or more minimum distances between different clusters are equal during the agglomerative process, because then different output clusterings are possible depending on the criterion used to break ties between distances. *MultiDendrograms* solves this problem implementing a variable-group algorithm that groups more than two clusters at the same time when ties occur.

*Keywords:* Hierarchical classification, Agglomerative algorithms, Ties in proximity, Dendrogram

---

## 1. Introduction

Agglomerative hierarchical clustering (Cormack, 1971; Gordon, 1999; Sneath and Sokal, 1973) starts from a proximity matrix between individuals, each one forming a singleton cluster, and gathers clusters into groups of clusters

---

\*Corresponding author. Tel.: +34 977 55 8508; fax: +34 977 55 9710.

*Email addresses:* [sergio.gomez@urv.cat](mailto:sergio.gomez@urv.cat) (Sergio Gómez),  
[alberto.fernandez@urv.cat](mailto:alberto.fernandez@urv.cat) (Alberto Fernández)  
*URL:* <http://deim.urv.cat/~sgomez> (Sergio Gómez)

or *superclusters*, the process being repeated until a complete hierarchy of partitions into clusters is formed. Among the different types of agglomerative methods we find Single Linkage, Complete Linkage, Unweighted Average, Weighted Average, etc., which differ in the definition of the proximity measure between clusters. Except for the Single Linkage case, all the other agglomerative hierarchical clustering techniques suffer from a non-uniqueness problem, sometimes called the *ties in proximity* problem, when the standard pair-group algorithm is used. This problem arises when two or more minimum distances between different clusters are equal during the agglomerative process. The standard approach consists in choosing any pair of clusters, breaking the ties between distances, and proceeds in the same way until a final hierarchical classification is obtained. However, different output clusterings are possible depending on the criterion used to break ties.

From the family of agglomerative hierarchical methods, Single Linkage and Complete Linkage are more susceptible than other methods to encounter ties during the clustering process, since they do not produce new proximity values different from the initial ones. With regard to the presence of ties in the original data, they are more frequent when someone works with binary variables, or even with integer variables comprising just some few distinct values. However, they can also appear using continuous variables, specially if the precision of experimental data is low. Sometimes, on the contrary, the absence of ties might be due to the representation of data with more decimal digits than it should be done. The non-uniqueness problem also depends on the measure used to obtain the proximity values from the initial variables. Moreover, in general, the larger the data set, the more ties arise (MacCuish et al., 2001).

The ties in proximity problem is well-known from several studies in different fields, for example in biology (Arnau et al., 2005; Backeljau et al., 1996; Hart, 1983), in psychology (van der Kloot et al., 2005), or in chemistry (MacCuish et al., 2001). Nevertheless, this problem is frequently ignored by software packages (Backeljau et al., 1996; Morgan and Ray, 1995; van der Kloot et al., 2005). Examples of packages which do not mention the problem are: the `linkage` function in the *Statistics Toolbox* of MATLAB (The MathWorks Inc., 2012); the `hclust` function in the *stats* package and the `agnes` function in the *cluster* package of R (R Development Core Team, 2011); and the `cluster` and `clustermat` commands of Stata (StataCorp LP, 2011).

On the contrary, some other statistical packages warn against the existence of the non-uniqueness problem in agglomerative hierarchical clustering.

For instance, if there are ties, then the `CLUSTER` procedure of SAS (SAS Institute Inc., 2011) reports the presence of ties in the SAS log and in a column of the cluster history. The results of a cluster analysis performed with SAS depend on the order of the observations in the data set, since ties are broken as follows: “Each cluster is identified by the smallest observation number among its members. For each pair of clusters, there is a smaller identification number and a larger identification number. If two or more pairs of clusters are tied for minimum distance between clusters, the pair that has the minimum larger identification number is merged. If there is a tie for minimum larger identification number, the pair that has the minimum smaller identification number is merged.” (SAS Institute Inc., 2011).

Another example can be found in the `Hierarchical Clustering Analysis` procedure of SPSS Statistics (IBM Corp., 2011), where it is explicitly stated that the results of the hierarchical clustering depend on the order of cases in the input file: “If tied distances or similarities exist in the input data or occur among updated clusters during joining, the resulting cluster solution may depend on the order of cases in the file. You may want to obtain several different solutions with cases sorted in different random orders to verify the stability of a given solution.” (IBM Corp., 2011).

Finally, a third example of warning comes from the `Agglomerate` function in the *Hierarchical Clustering Package* of Mathematica (Wolfram Research Inc., 2008), where the user is briefly warned against the presence of ties by the following message: “Ties have been detected; reordering input may produce a different result.” (Wolfram Research Inc., 2008).

Therefore, software packages which do not ignore the non-uniqueness problem fail to adopt a common standard with respect to ties, and they simply break ties in any arbitrary way. Here we introduce *MultiDendrograms*, an application that implements the variable-group algorithm (Fernández and Gómez, 2008) to solve the non-uniqueness problem found in the standard pair-group approach. In Section 2 we describe the variable-group algorithm, which groups more than two clusters at the same time when ties occur. Section 3 contains a basic manual of *MultiDendrograms*. In Section 4 we show a case study performed with *MultiDendrograms* using data from a real example. Finally, in Section 5, we give some concluding remarks.

## 2. Agglomerative hierarchical algorithms

### 2.1. Pair-group algorithm

Agglomerative hierarchical procedures build a hierarchical classification in a bottom-up way, from a proximity matrix containing dissimilarity data between individuals of a set  $\Omega = \{x_1, \dots, x_n\}$ . (Note that the same analysis could be done using similarity data.) The algorithm has the following steps:

- 0) Initialize  $n$  singleton clusters with one individual in each of them:  $\{x_1\}, \dots, \{x_n\}$ . Initialize also the distances between clusters,  $D(\{x_i\}, \{x_j\})$ , with the values of the distances between individuals,  $d(x_i, x_j)$ :

$$D(\{x_i\}, \{x_j\}) = d(x_i, x_j), \quad \forall i, j = 1, \dots, n.$$

- 1) Find the shortest distance separating two different clusters.
- 2) Select two clusters  $X_i$  and  $X_{i'}$ , subsets of  $\Omega$ , separated by such shortest distance and merge them into a new supercluster  $X_i \cup X_{i'}$ .
- 3) Compute the distances  $D(X_i \cup X_{i'}, X_j)$  between the new supercluster  $X_i \cup X_{i'}$  and each of the other clusters  $X_j$ .
- 4) If all individuals are not in the same cluster yet, then go back to step 1.

Following Sneath and Sokal (1973), this type of approach is known as a pair-group method, in opposition to the variable-group method which we will introduce in subsection 2.2. Depending on the criterion used for the calculation of distances in step 3, different agglomerative hierarchical clusterings can be implemented. The most commonly used are: Single Linkage, Complete Linkage, Unweighted Average, Weighted Average, Unweighted Centroid, Weighted Centroid, and Ward's method.

Lance and Williams (1966) put these different hierarchical strategies into a single system, avoiding the need of a separate computer program for each one of them. Assume three clusters  $X_i$ ,  $X_{i'}$  and  $X_j$ , containing  $|X_i|$ ,  $|X_{i'}$  and  $|X_j|$  individuals respectively, and with distances between them already determined as  $D(X_i, X_{i'})$ ,  $D(X_i, X_j)$  and  $D(X_{i'}, X_j)$ . Further assume that the smallest of all distances still to be considered is  $D(X_i, X_{i'})$ , so that  $X_i$  and  $X_{i'}$  are joined to form a new supercluster  $X_i \cup X_{i'}$ , with  $|X_i| + |X_{i'}$  individuals. Lance and Williams (1966) analyzed the distance  $D(X_i \cup X_{i'}, X_j)$  that appears in step 3 of the above algorithm, and they expressed it in terms of the distances already defined, all known at the moment of fusion, using

Method	$\alpha_i$ ( $\alpha_{i'}$ )	$\beta$	$\gamma$
Single Linkage	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete Linkage	$\frac{1}{2}$	0	$+\frac{1}{2}$
Unweighted Average	$\frac{ X_i }{ X_i + X_{i'} }$	0	0
Weighted Average	$\frac{1}{2}$	0	0
Unweighted Centroid	$\frac{ X_i }{ X_i + X_{i'} }$	$-\frac{ X_i  X_{i'} }{( X_i + X_{i'} )^2}$	0
Weighted Centroid	$\frac{1}{2}$	$-\frac{1}{4}$	0
Ward	$\frac{ X_i + X_j }{ X_i + X_{i'} + X_j }$	$-\frac{ X_j }{ X_i + X_{i'} + X_j }$	0

Table 1: Parameter values for Lance and Williams' formula.

the following recurrence relation:

$$\begin{aligned}
D(X_i \cup X_{i'}, X_j) &= \alpha_i D(X_i, X_j) + \alpha_{i'} D(X_{i'}, X_j) \\
&+ \beta D(X_i, X_{i'}) + \gamma |D(X_i, X_j) - D(X_{i'}, X_j)|.
\end{aligned} \tag{1}$$

Using this technique, superclusters can always be computed from previous clusters and it is not necessary to look back at the original data matrix during the agglomerative process. The values of the parameters  $\alpha_i$ ,  $\alpha_{i'}$ ,  $\beta$  and  $\gamma$  determine the nature of the clustering strategy. Table 1 gives the values of the parameters that define the most commonly used agglomerative hierarchical clustering methods.

## 2.2. Variable-group algorithm

The algorithm proposed by Fernández and Gómez (2008) to ensure uniqueness in agglomerative hierarchical clustering has the following steps:

- 0) Initialize  $n$  singleton clusters with one individual in each of them:  $\{x_1\}, \dots, \{x_n\}$ . Initialize also the distances between clusters,  $D(\{x_i\}, \{x_j\})$ , with the values of the distances between individuals,  $d(x_i, x_j)$ :

$$D(\{x_i\}, \{x_j\}) = d(x_i, x_j), \quad \forall i, j = 1, \dots, n.$$

- 1) Find the shortest distance separating two different clusters, and record it as  $D_{lower}$ .

- 2) Select all the groups of clusters separated by shortest distance  $D_{lower}$  and merge them into several new superclusters  $X_I$ . The result of this step can be some superclusters made up of just one single cluster ( $|I| = 1$ ), as well as some superclusters made up of various clusters ( $|I| > 1$ ). Notice that the latter superclusters all must satisfy the condition  $D_{min}(X_I) = D_{lower}$ , where

$$D_{min}(X_I) = \min_{i \in I} \min_{\substack{i' \in I \\ i' \neq i}} D(X_i, X_{i'}).$$

- 3) Update the distances between clusters following the next substeps:
- 3.1) Compute the distances  $D(X_I, X_J)$  between all superclusters, and record the minimum of them as  $D_{next}$  (this will be the shortest distance  $D_{lower}$  in the next iteration of the algorithm).
- 3.2) For each supercluster  $X_I$  made up of various clusters ( $|I| > 1$ ), assign a common agglomeration interval  $[D_{lower}, D_{max}(X_I)]$  for all its constituent clusters  $X_i, i \in I$ , where

$$D_{max}(X_I) = \max_{i \in I} \max_{\substack{i' \in I \\ i' \neq i}} D(X_i, X_{i'}).$$

- 4) If all individuals are not in the same cluster yet, then go back to step 1.

In the same way that several agglomerative hierarchical methods can be computed with the same pair-group algorithm using Lance and Williams' formula, Fernández and Gómez (2008) gave a generalization of Eq. (1), compatible with the agglomeration of more than two clusters simultaneously, that can be used to compute the distances  $D(X_I, X_J)$  in step 3 of the variable-group algorithm. Suppose we want to agglomerate two superclusters  $X_I$  and  $X_J$ , respectively indexed by  $I = \{i_1, i_2, \dots, i_p\}$  and  $J = \{j_1, j_2, \dots, j_q\}$ . Then the distance between them is defined as:

$$\begin{aligned} D(X_I, X_J) &= \sum_{i \in I} \sum_{j \in J} \alpha_{ij} D(X_i, X_j) \\ &+ \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} \beta_{ii'} D(X_i, X_{i'}) + \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} \beta_{jj'} D(X_j, X_{j'}) \\ &+ \delta \sum_{i \in I} \sum_{j \in J} \gamma_{ij} [D_{max}(X_I, X_J) - D(X_i, X_j)] \\ &- (1 - \delta) \sum_{i \in I} \sum_{j \in J} \gamma_{ij} [D(X_i, X_j) - D_{min}(X_I, X_J)], \end{aligned} \quad (2)$$

Method	$\alpha_{ij}$	$\beta_{ii'} (\beta_{jj'})$	$\gamma_{ij}$	$\delta$
Single Linkage	$\frac{1}{ I  J }$	0	$\frac{1}{ I  J }$	0
Complete Linkage	$\frac{1}{ I  J }$	0	$\frac{1}{ I  J }$	1
Unweighted Average	$\frac{ X_i  X_j }{ X_I  X_J }$	0	0	—
Weighted Average	$\frac{1}{ I  J }$	0	0	—
Unweighted Centroid	$\frac{ X_i  X_j }{ X_I  X_J }$	$-\frac{ X_i  X_{i'} }{ X_I ^2}$	0	—
Weighted Centroid	$\frac{1}{ I  J }$	$-\frac{1}{ I ^2}$	0	—
Ward	$\frac{ X_i + X_j }{ X_I + X_J }$	$-\frac{ X_J }{ X_I } \frac{ X_i + X_{i'} }{ X_I + X_J }$	0	—

Table 2: Parameter values for Fernández and Gómez’s formula.

where

$$D_{max}(X_I, X_J) = \max_{i \in I} \max_{j \in J} D(X_i, X_j)$$

and

$$D_{min}(X_I, X_J) = \min_{i \in I} \min_{j \in J} D(X_i, X_j).$$

Table 2 shows the values obtained by Fernández and Gómez (2008) for the parameters  $\alpha_{ij}$ ,  $\beta_{ii'}$ ,  $\beta_{jj'}$ ,  $\gamma_{ij}$  and  $\delta$  which determine the clustering method calculated with Eq. (2).

Using the pair-group algorithm, only the Centroid methods (Weighted and Unweighted) may produce *reversals*. Let us remember that a reversal arises in a dendrogram when it contains at least two clusters  $X$  and  $Y$  for which  $X \subset Y$  but  $h(X) > h(Y)$ , where  $h(X)$  is the height in the dendrogram at which the individuals of cluster  $X$  are merged together (Morgan and Ray, 1995). In the case of the variable-group algorithm, reversals may appear in substep 3.2 when  $D_{max}(X_I) > D_{next}$  for any supercluster  $X_I$ . Although reversals make dendrograms difficult to interpret if they occur during the last stages of the agglomerative process, it can be argued that they are not very disturbing if they occur during the first stages. Thus, as happens with the Centroid methods in the pair-group case, it could be reasonable to use the variable-group algorithm as long as no reversals at all or only unimportant ones were obtained.

In substep 3.2 of the variable-group clustering algorithm, sometimes it will not be enough to adopt a fusion interval, but it will be necessary to obtain an exact fusion value (e.g. in order to calculate a distortion measure). In these

cases, *MultiDendrograms* systematically uses the shortest distance,  $D_{lower}$ , as the fusion value for superclusters made up of several clusters. Such criterion allows the recovering of the pair-group result for the Single Linkage method and, in addition, it avoids the appearance of reversals. However, it must be emphasized that the use of exact fusion values, without considering the fusion intervals at their whole lengths, means that some valuable information regarding the heterogeneity of the clusters is being lost.

### 3. MultiDendrograms manual

This section contains a basic manual for version 3.0 of the application *MultiDendrograms*. To get the latest available version of the software, please visit *MultiDendrograms* web page at

<http://deim.urv.cat/~sgomez/multidendrograms.php>

#### 3.1. Input data

*MultiDendrograms* needs to have input data in a compatible text file format. The data file must represent a distances (or weights) matrix. There are two different arrangements that these data can be stored in such that *MultiDendrograms* may accept them: matrix and list formats.

##### 3.1.1. Matrix-like file

Each line in the text file contains a data matrix row. The characteristics of these files are:

- The matrix must be symmetric, and the diagonal values must be zeros.
- Within each row, the values can be separated by: spaces (' '), tab character, semicolon (';'), comma (','), or vertical bar ('|').
- It is possible to include labels with the names of the nodes in an additional first row or column, but not in both.
- If present, the labels of the nodes cannot contain any of the previous separators.



### 3.1.2. List-like file

Each line in the text file contains three fields, which represent the labels of two nodes and the distance (or weight) between them. The characteristics of these files are:

- The separators between the three fields can be: spaces (' '), tab character, semicolon (';'), comma (','), or vertical bar ('|').
- The labels of the nodes cannot contain any of the previous separators.
- Distances from a node to itself (e.g. “a a 0.0”) must not be included.
- *MultiDendrograms* accepts either the presence or absence of symmetric data lines, i.e. if the distance between nodes **a** and **b** is 2.0, then it is possible to include in the list just the line “a b 2.0”, or both “a b 2.0” and “b a 2.0”. If both are present, the program checks whether they are equal.

### 3.2. Loading data

Once we have our data in a compatible format, we can load them into *MultiDendrograms*:

1. Choose the desired settings, mainly the options for the **Type of measure** and the **Clustering algorithm**. These settings will be explained in detail in subsection 3.4.
2. Click on the **Load** button.
3. Select the file to open and then click on the **Open** button.
4. Now the data is loaded and its multidendrogram representation is shown (see Fig. 1).

### 3.3. Actions

*MultiDendrograms* only has two action buttons: **Load** and **Update**. **Load** is used to read data from a file and create a new window for the corresponding multidendrogram, using the current values of the parameters, while **Update** is needed for the actualization of the active multidendrogram when one or more parameters are changed. Below these buttons it is shown the name of the data file corresponding to the active multidendrogram. It is possible to load the same data file several times, in order to compare the multidendrogram appearance for different parameters settings.

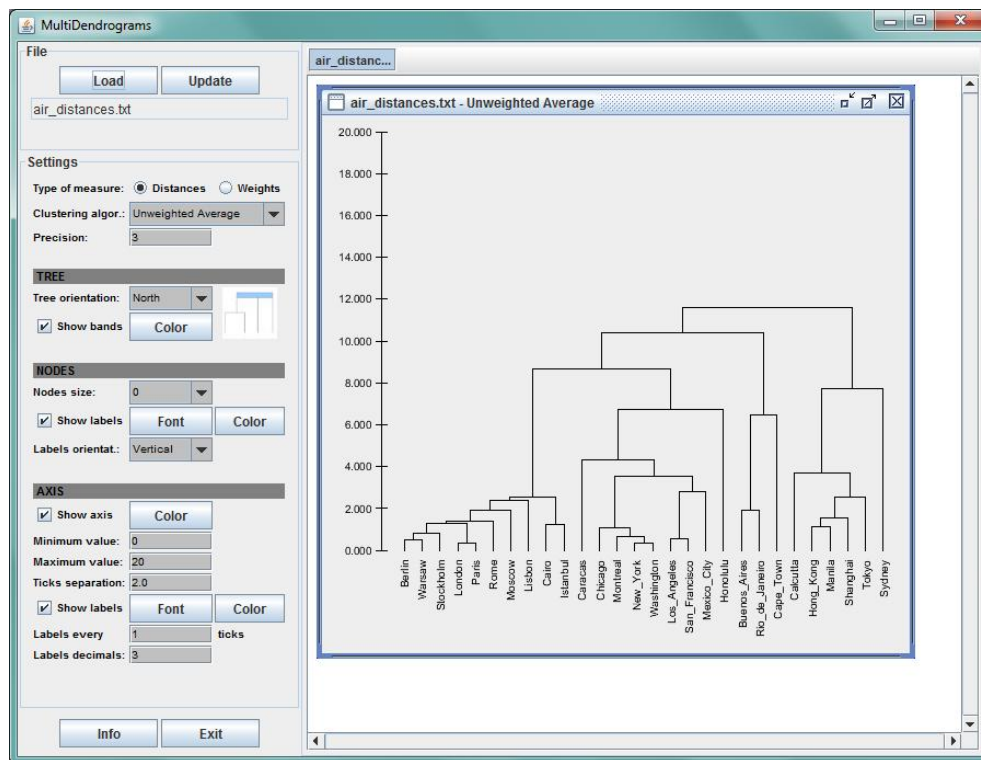


Figure 1: *MultiDendrograms* user interface.

### 3.4. Settings

The program automatically applies default values to the parameters depending on the data loaded, which should be adjusted as desired. Fig. 1 shows the settings panel in the left part of the window, with four different areas corresponding to main data representation, tree, nodes and axis settings, respectively.

Changes in the main data representation parameters affect the structure of the multidendrogram tree, thus it needs to be fully recomputed, operation which may take several seconds, even minutes (depending on the data size and the computer speed). On the other hand, changes in the tree, nodes and axis settings only modify the visual representation of the multidendrogram and are much faster to update.

### 3.4.1. Main data representation settings

**Type of measure:** It allows choosing between two different types of measure, **Distances** and **Weights**. Choose between them according to the meaning of the input data. With **Distances**, the closer the nodes the lower their distance. On the contrary, with **Weights**, the closer the nodes the larger their weight. By default, **Distances** is selected.

**Clustering algorithm:** Seven distinct algorithms are available, **Single Linkage**, **Complete Linkage**, **Unweighted Average**, **Weighted Average**, **Unweighted Centroid**, **Weighted Centroid**, and **Ward**. By default, **Unweighted Average** is selected.

**Precision:** Number of significant digits of the data to be taken into account for the calculations. This is a very important parameter, since equal distances at a certain precision may become different by increasing its value. Thus, it may be responsible of the existence of tied distances. The rule should be not to use a precision larger than the resolution given by the experimental setup that has generated the data. By default, the precision is set to that of the data value with the largest number of significant digits.

### 3.4.2. Tree settings

**Tree orientation:** Four orientations are available, **North**, **South**, **East**, and **West**, which refer to the relative position of the root of the tree. By default, **North** is selected.

**Show bands:** It allows showing a band or not in case of tied minimum distances between three or more elements, and selecting the color of the band. If **Show bands** is selected, the bands show the heterogeneity of distances between the clustered elements. Otherwise, the elements are grouped at their minimum distance (see Fig. 2). By default, **Show bands** is selected and its default color is **light gray**.

Let us explain the meaning of the bands. In MultiDendrograms, if several pairs of elements share the same minimum distance, they are clustered together in one step. For instance, suppose that the minimum distance is 0.4 and it corresponds to the tied pairs (A, B) and (B, C). MultiDendrograms puts them together in the same cluster {A, B, C} at height 0.4. However, if the distance (A, C) is 0.5, it is possible to represent the cluster {A, B, C}

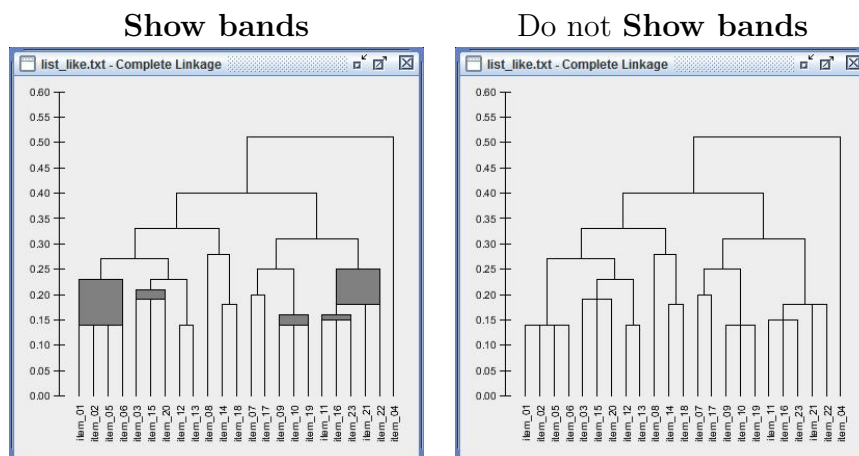


Figure 2: Possibility of showing a heterogeneity band in case of ties.

as a rectangle which spans between heights 0.4 and 0.5, thus showing the heterogeneity of the clustered elements.

### 3.4.3. Nodes settings

**Nodes size:** Six different node sizes are available. By default, 0 is selected (i.e. nodes are not shown).

**Show labels:** It allows showing or not the labels of the nodes and selecting their color and font. By default, **Show labels** is selected, the font is **Arial** and the color is **black**.

**Labels orientation:** Three different orientations are available, **Vertical**, **Horizontal** and **Oblique**. By default, **Vertical** is selected.

### 3.4.4. Axis settings

**Show axis:** It allows showing or not the axis and selecting its color. By default, **Show axis** is selected and the color is **black**.

**Minimum value / Maximum value:** They allow choosing the minimum and maximum values of the axis, respectively. They also affect the view of the multidendrogram. The default values are calculated from the loaded data.

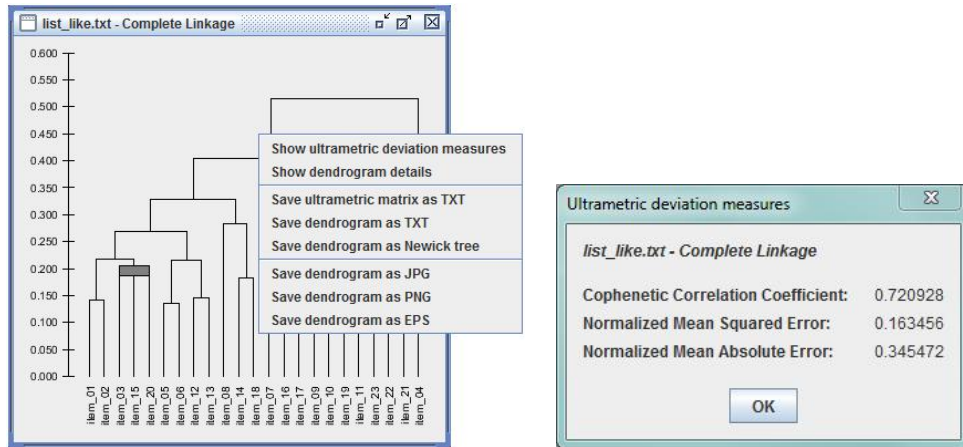


Figure 3: Contextual menu (left) and ultrametric deviation measures (right).

**Ticks separation:** It allows choosing the separation between consecutive ticks of the axis. The default value is calculated from the loaded data.

**Show labels:** It allows showing or not the labels of the axis, and selecting their font and color. By default, **Show labels** is selected, the font is **Arial** and the color is **black**.

**Labels every ... ticks:** Number of consecutive ticks to find the next labeled tick. By default is set to 1.

**Labels decimals:** Number of decimal digits of the tick labels. By default is set equal to the **Precision** parameter.

### 3.5. Analyzing and exporting results

The contextual menu, available by right-clicking on any multidendrogram window, gives access to several options for analyzing and exporting results to file (see Fig. 3).

**Show ultrametric deviation measures:** Computes the ultrametric matrix corresponding to the active multidendrogram and obtains three deviation measures between the original matrix and the ultrametric one, which are the **Cophenetic Correlation Coefficient**, the **Normalized Mean Squared Error**, and the **Normalized Mean Absolute Error** (see Fig. 3).

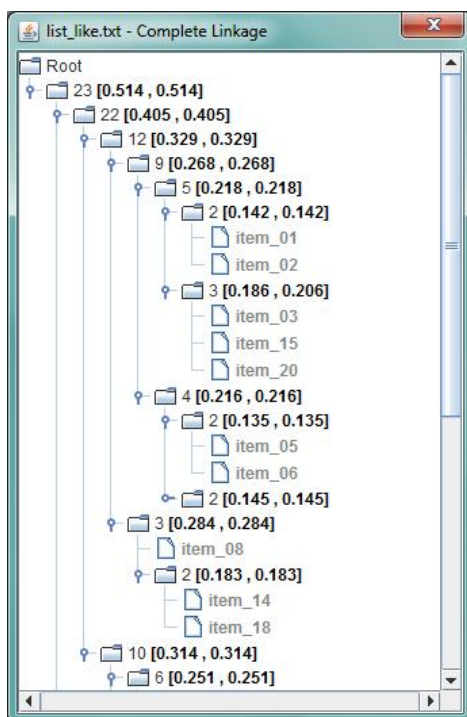


Figure 4: Multidendrogram details window.

**Show dendrogram details:** Opens a window that contains all the information of the multidendrogram in a navigable folder-like structure (see Fig. 4). The available information in the details window is:

- Number of data items (leaves of the tree) under each interior node of the multidendrogram. The interior nodes in the multidendrogram representation correspond to the clusters found during the agglomerative process.
- Minimum and maximum distances at which the children of an interior node are joined to form a new cluster. These values may only be different in case of tied minimum distances, and they become a band in the multidendrogram representation.
- List of children for each interior node, which may be either interior nodes or data items (leaves).

**Save ultrametric matrix as TXT:** Calculates the ultrametric matrix cor-

responding to the loaded data and saves it to a text file represented as a matrix, not as a list, with the labels of the nodes in the first row. This text file can then be easily loaded into any text editor or spreadsheet application.

**Save dendrogram as TXT:** Saves the multidendrogram details to a text file.

**Save dendrogram as Newick tree:** Saves the multidendrogram details in Newick tree format. In this format, the information given by the bands is lost and only the minimum distance is saved. However, it has the advantage that it is a standard format used by many other applications, thus allowing their use to generate other graphical representations.

**Save dendrogram as JPG / PNG / EPS:** It is also possible to save the image of the multidendrogram in three different formats (JPG, PNG and EPS) using the corresponding **Save dendrogram as ...** context menu items.

### 3.6. Command-line direct calculation

It is possible to use *MultiDendrograms* in command-line mode to calculate the dendrogram without the graphical interface. This is useful in several situations:

- To automate the generation of many dendrograms using scripts.
- When there is no need of a plot of the dendrogram.
- When the plot of the dendrogram is to be performed with a different program.
- When the number of elements is too large to allow a graphical representation.
- To be able to call *MultiDendrograms* from a different application.

The input parameters of a command-line call are:

- The name of the input data file, in matrix or list format.
- The type of measure: distances or weights.

- The clustering algorithm: single linkage, complete linkage, unweighted average, weighted average, unweighted centroid, weighted centroid or ward.
- The precision, i.e. the number of significant digits of the data for the calculations. This parameter is optional, and if not given it is calculated from the data. However, the rule should be not to use a precision larger than the resolution given by the experimental setup which has generated the data.

The output results are:

- A file with the dendrogram tree in text format.
- A file with the dendrogram in Newick format.
- A file with the ultrametric matrix.
- The ultrametric deviation measures: the cophenetic correlation coefficient, the normalized mean squared error, and the normalized mean absolute error.

The syntax of a command-line direct calculation is:

```
java -jar multidendrograms.jar -direct FILE TYPE METHOD [PREC]
```

And a concrete example for a distances matrix using the complete linkage method with 3 decimal significant digits is:

```
java -jar multidendrograms.jar -direct data.txt DISTANCES
      Complete_Linkage 3
```

#### 4. Case study

We show here a case study performed with *MultiDendrograms* using data from a real example which had been previously studied by Morgan and Ray (1995), and Fernández and Gómez (2008). It is the *Glamorganshire soils* example, formed by similarity data between 23 different soils. A table with the similarities between soils can be found in Morgan and Ray (1995), where the values are given with an accuracy of three decimal places. In order to work with dissimilarities as in Fernández and Gómez (2008), we have



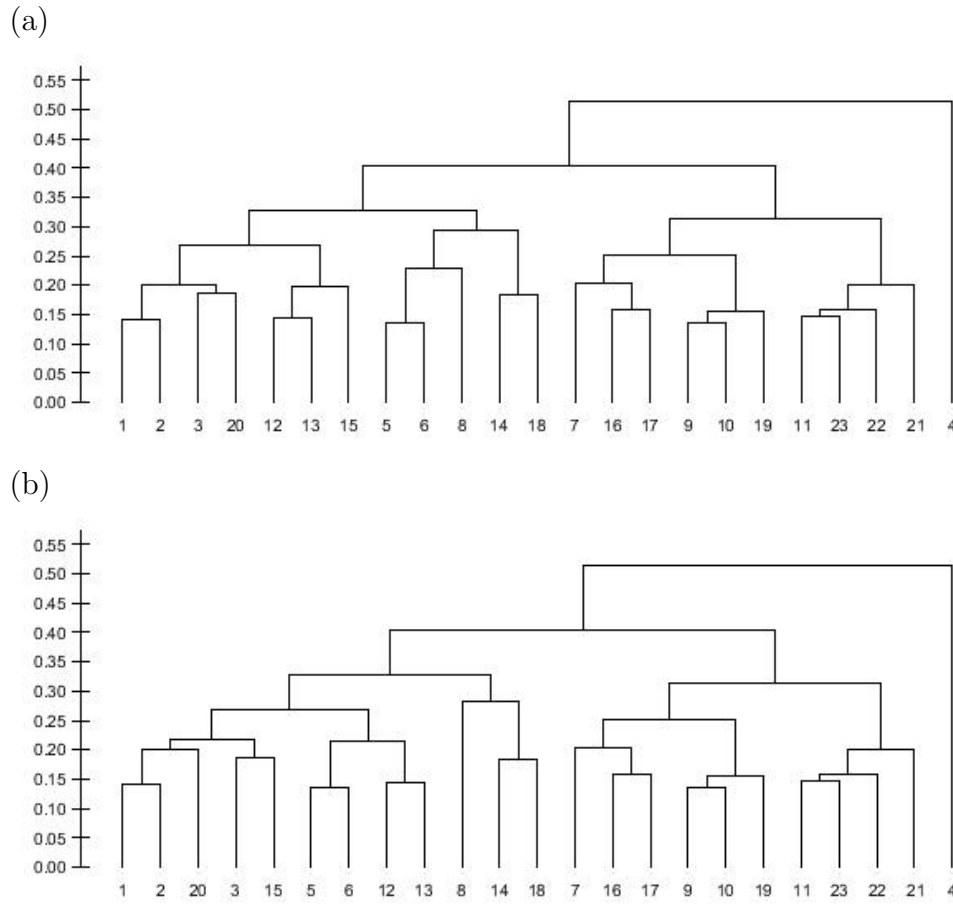


Figure 5: Complete Linkage dendrograms for the soils data. According to the brown earths soil group formed by soils 1, 2, 6, 12 and 13, the dendrogram in (a) is worse than the one in (b) because the former merges these soils at a posterior stage of the agglomerative process.

transformed the similarities  $s(x_i, x_j)$  into the corresponding dissimilarities  $d(x_i, x_j) = 1 - s(x_i, x_j)$ , even though *MultiDendrograms* is also capable of directly working with similarities.

The original data contain a tied value for pairs of soils (3, 15) and (3, 20), which is responsible for two different dendrograms using the Complete Linkage method (see Fig. 5). Morgan and Ray (1995) explain that the 23 soils had been categorized into eight “great soil groups” by a surveyor. Focusing

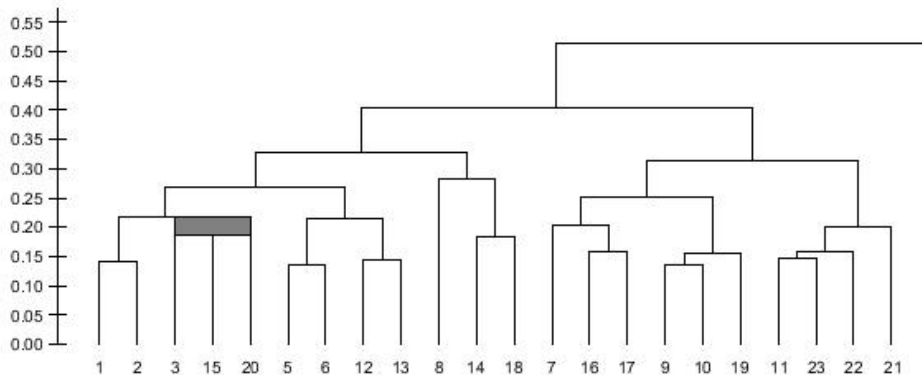


Figure 6: Complete Linkage multidendrogram for the soils data.

on soils 1, 2, 6, 12 and 13, which are the only members of the brown earths soil group, we observe that the dendrogram in Fig. 5(a) does not place them in the same cluster until they join soils from five other soil groups, forming the cluster  $\{1, 2, 3, 20, 12, 13, 15, 5, 6, 8, 14, 18\}$ . From this point of view, the dendrogram in Fig. 5(b) is better, since the corresponding cluster loses soils 8, 14 and 18, each representing a different soil group. Therefore, in this case we have two possible solution dendrograms and the probability of obtaining the “good” one is, hence, 50%.

We now use the *MultiDendrograms* software to obtain the multidendrogram representation corresponding to the *Glamorganshire soils* data. We select **Complete Linkage** as the **Clustering algorithm**, and **Precision = 3**. We obtain a multidendrogram that we have saved in JPG format using the contextual menu. The result, shown in Fig. 6, unravels the existence of a tie comprising soils 3, 15 and 20. Besides, the multidendrogram gives us the good classification, that is, the one with soils 8, 14 and 18 out of the brown earths soil group. Except for the internal structure of the cluster  $\{1, 2, 3, 15, 20\}$ , the rest of the multidendrogram hierarchy coincides with that of the dendrogram shown in Fig. 5(b).

## 5. Conclusions

*MultiDendrograms* is a simple yet powerful software to make hierarchical clusterings of data, distributed under an Open Source license. It implements

the variable-group algorithm for agglomerative hierarchical clustering that solves the non-uniqueness problem found in the standard pair-group algorithm. This problem consists in obtaining different hierarchical classifications from a unique set of proximity data, when two or more minimum distances between different clusters are equal during the agglomerative process. In such cases, selecting a unique classification can be misleading. Software packages that do not ignore this problem fail to adopt a common standard with respect to ties, and many of them simply break ties in any arbitrary way.

Starting from a distances (or weights) matrix, *MultiDendrograms* computes its dendrogram using the variable-group algorithm which groups more than two clusters at the same time when ties occur. Its main properties are:

- When there are no ties, MultiDendrograms gives the same results as any pair-group algorithm.
- It always gives a uniquely determined solution.
- In the multidendrogram representation of the results, the occurrence of ties during the agglomerative process can be explicitly observed. Furthermore, the height of any fusion interval (the *bands* in the program) indicates the degree of heterogeneity inside the corresponding cluster.

*MultiDendrograms* also allows the tuning of many graphical representation parameters, and the results can be easily exported to file. A summary of its characteristics is:

- Multiplatform: developed in Java, runs in all operating systems (e.g. Windows, Linux and MacOS).
- Graphical user interface: data selection, hierarchical clustering options, multidendrogram representation parameters, navigation across the multidendrogram, deviation measures, etc.
- Also command-line direct calculation without graphical user interface.
- Implementation of variable-group algorithms for agglomerative hierarchical clustering.
- Works with distance and weight matrices.
- Many parameters for the customization of the dendrogram layout: size, orientation, labels, axis, etc.

- Navigation through the dendrogram information in a folder-like window.
- Calculation of corresponding ultrametric matrix.
- Calculation of deviation measures: cophenetic correlation coefficient, normalized mean squared error, and normalized mean absolute error.
- Save dendrogram details in text and Newick tree format.
- Save dendrogram image as JPG, PNG and EPS.

Although ties need not be present in the initial proximity data, they may arise during the agglomerative process. For this reason, and given that the results of the variable-group algorithm coincide with those of the pair-group algorithm when there are no ties, we recommend to directly use *MultiDendrograms*. With a single action one knows whether ties exist or not, and additionally the subsequent hierarchical classification is obtained.

## Acknowledgments

The authors thank Josep Maria Mateo for discussion and helpful comments about the practical functionalities of this software. S.G. acknowledges support from the Spanish Ministry of Science and Innovation FIS2009-13730-C02-02, the Generalitat de Catalunya 2009-SGR-838, and the European Union FP7 FET projects PLEXMATH 317614 and MULTIPLEX 317532. A.F. acknowledges support from the Spanish Ministry of Science and Innovation CTQ2009-14627, and the Generalitat de Catalunya 2009-SGR-1529.

## References

- Arnau, V., Mars, S., Marín, I., 2005. Iterative cluster analysis of protein interaction data. *Bioinformatics* 21, 364–378.
- Backeljau, T., De Bruyn, L., De Wolf, H., Jordaens, K., Van Dongen, S., Winnepennincks, B., 1996. Multiple UPGMA and neighbor-joining trees and the performance of some computer packages. *Molecular Biology and Evolution* 13, 309–313.
- Cormack, R., 1971. A review of classification. *Journal of the Royal Statistical Society A* 134, 321–367.

- Fernández, A., Gómez, S., 2008. Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms. *Journal of Classification* 25, 43–65.
- Gordon, A., 1999. *Classification*. Chapman & Hall/CRC. 2nd edition.
- Hart, G., 1983. The occurrence of multiple UPGMA phenograms, in: Felsenstein, J. (Ed.), *Numerical Taxonomy*. Springer-Verlag, pp. 254–258.
- IBM Corp., 2011. *IBM SPSS Statistics Base 20*. IBM Corp.. Armonk, NY, USA.
- van der Kloot, W., Spaans, A., Heiser, W., 2005. Instability of hierarchical cluster analysis due to input order of the data: The PermuCLUSTER solution. *Psychological Methods* 10, 468–476.
- Lance, G., Williams, W., 1966. A generalized sorting strategy for computer classifications. *Nature* 212, 218.
- MacCuish, J., Nicolaou, C., MacCuish, N., 2001. Ties in proximity and clustering compounds. *Journal of Chemical Information and Computer Sciences* 41, 134–146.
- Morgan, B., Ray, A., 1995. Non-uniqueness and inversions in cluster analysis. *Applied Statistics* 44, 117–134.
- R Development Core Team, 2011. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.
- SAS Institute Inc., 2011. *SAS/STAT 9.3 User’s Guide*. SAS Institute Inc.. Cary, NC, USA.
- Sneath, P., Sokal, R., 1973. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W.H. Freeman and Company.
- StataCorp LP, 2011. *Stata Statistical Software: Release 12*. StataCorp LP. College Station, TX, USA.
- The MathWorks Inc., 2012. *MATLAB — Statistics Toolbox User’s Guide R2012a*. The MathWorks Inc.. Natick, MA, USA.

Wolfram Research Inc., 2008. Mathematica, Version 7.0. Wolfram Research Inc.. Champaign, IL, USA.