Journal of Statistical Software

September 2025, Volume 114, Issue 2. doi: 10.18637/jss.v114.i02

mdendro: An R Package for Extended Agglomerative Hierarchical Clustering

Alberto Fernández
Universitat Rovira i Virgili

Sergio Gómez
Universitat Rovira i Virgili

Abstract

mdendro is an R package that provides a comprehensive collection of linkage methods for agglomerative hierarchical clustering on a matrix of proximity data (distances or similarities), returning a multifurcated dendrogram or multidendrogram. Multidendrograms can group more than two clusters at the same time, solving the nonuniqueness problem that arises when there are ties in the data. This problem causes that different binary dendrograms are possible depending both on the order of the input data and on the criterion used to break ties. Weighted and unweighted versions of the most common linkage methods are included in the package, which also implements two parametric linkage methods. In addition, package **mdendro** provides five descriptive measures to analyze the resulting dendrograms: cophenetic correlation coefficient, space distortion ratio, agglomerative coefficient, chaining coefficient and tree balance.

Keywords: multifurcated dendrogram, parametric linkage, dendrogram descriptor, R.

1. Introduction

Agglomerative hierarchical clustering (AHC) is widely used to classify individuals into a hierarchy of clusters organized in a tree structure called dendrogram (Gordon 1999). There are different types of AHC linkage methods, such as single linkage, complete linkage, average linkage and Ward's method, which only differ in the definition of the distance measure between clusters. All these methods start from a distance matrix between individuals, each one forming a singleton cluster, and gather clusters into groups of clusters, this process being repeated until a complete hierarchy of partitions into clusters is formed.

Except for the single linkage case, all the other AHC linkage methods suffer from a nonuniqueness problem known as the ties in proximity problem. This problem arises whenever there are more than two clusters separated by the same minimum distance during the agglomerative

process of a pair-group AHC algorithm. This type of algorithm breaks ties choosing any pair of clusters, and proceeds in the same way until a binary dendrogram is obtained. However, different binary dendrograms are possible depending both on the order of the input data and on the criterion used to break ties.

The ties in proximity problem is long known (Hart 1983; Morgan and Ray 1995; Backeljau, De Bruyn, De Wolf, Jordaens, Van Dongen, and Winnepennincks 1996), even from studies in different fields, such as biology (Arnau, Mars, and Marín 2005), psychology (Van der Kloot, Spaans, and Heiser 2005) and chemistry (MacCuish, Nicolaou, and MacCuish 2001). The extent of the problem in a particular field has been analyzed for microsatellite markers (Segura-Alabart, Serratosa, Gómez, and Fernández 2022). Nevertheless, this problem is ignored by some software packages: function `hclust()` in package **stats** and function `agnes()` in package **cluster** of R (R Core Team 2025), commands `cluster()` and `clustermat()` of Stata (StataCorp LLC 2021), function `linkage()` in the **Statistics and Machine Learning Toolbox** of MATLAB (The MathWorks Inc. 2022), and function `hclust()` in package **Clustering.jl** of Julia (Bezanson, Edelman, Karpinski, and Shah 2017).

There are some other statistical packages that just warn against the existence of the nonuniqueness problem in AHC. For instance, procedure `Hierarchical Cluster Analysis` of SPSS Statistics (IBM Corporation 2021), procedure `CLUSTER` of SAS (SAS Institute Inc. 2018), function `Agglomerate()` in the **Hierarchical Clustering Package** of Mathematica (Wolfram Language & System Documentation Center 2020), and also function `linkage()` in module `scipy.cluster.hierarchy` of package **SciPy** (Virtanen *et al.* 2020) for Python (Van Rossum *et al.* 2011).

Software packages that do not ignore the nonuniqueness problem fail to adopt a common standard with respect to ties, and they simply break ties in any arbitrary way. Here we introduce **mdendro**, an R package that implements a variable-group AHC algorithm (Fernández and Gómez 2008) to solve the nonuniqueness problem found in any pair-group AHC algorithm.

Package **mdendro** was designed using state-of-the-art methods based on neighbor chains, and its base code was implemented in C++. It was developed using object-oriented programming, where each linkage method constitutes a different class. This eases the extensibility of the package since other linkage methods can be added as new classes. The package is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=mdendro` and on GitHub at `https://github.com/sergio-gomez/mdendro`. The functionality of the R package **mdendro** makes it very similar and compatible with the main ones currently in use, namely the R functions `hclust()` in package **stats** and `agnes()` in package **cluster** (Maechler, Rousseeuw, Struyf, Hubert, and Hornik 2021). The result is a package **mdendro** that includes and extends the functionality of these reference functions.

The rest of the article is structured as follows. In Section 2 we describe the pair-group and the variable-group AHC algorithms, the latter grouping more than two clusters at the same time when ties occur. Section 3 describes the most common AHC linkage methods: single linkage, complete linkage, average linkage, centroid linkage and Ward's method. Package **mdendro** also includes two parametric linkage methods: β-flexible linkage and versatile linkage. In the same section, five descriptive measures for the resulting dendrograms are included: cophenetic correlation coefficient, space distortion ratio, agglomerative coefficient, chaining coefficient and tree balance. Section 4 compares package **mdendro** with other state-of-the-art packages for AHC. Finally, in Section 5, we give some concluding remarks.

2. Agglomerative hierarchical clustering algorithms

2.1. Pair-group algorithm

AHC algorithms build a hierarchical tree in a bottom-up way, from a matrix of pairwise distances between individuals of a set $\Omega = \{x_1, \ldots, x_n\}$. The pair-group algorithm (Sneath and Sokal 1973) has the following steps:

(0) Initialize n singleton clusters with one individual in each one of them: $X_1 = \{x_1\}$, ..., $X_n = \{x_n\}$. Initialize also the distances between clusters, $D(X_i, X_j)$, with the values of the distances between individuals, $d(x_i, x_j)$:

$$D(X_i, X_j) = d(x_i, x_j)\,, \qquad \forall i, j = 1, \ldots, n\,.$$

(1) Find the shortest distance separating two different clusters, D_{shortest}.

(2) Select two clusters X_i and $X_{i'}$ separated by the shortest distance D_{shortest}, and merge them into a new cluster $X_i \cup X_{i'}$.

(3) Compute the distances $D(X_i \cup X_{i'}, X_j)$ between the new cluster $X_i \cup X_{i'}$ and each one of the other clusters X_j.

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

The nonuniqueness problem in the pair-group algorithm arises when two or more shortest distances between different clusters are equal during the agglomerative process (Hart 1983). The standard approach consists in choosing only a single pair to break the tie. However, different hierarchical clusterings are possible depending on the criterion used to break ties (usually a pair is just chosen at random), and the user is unaware of this problem.

For example, let us consider the genetic profiles of 51 grapevine cultivars at six microsatellite loci (Almadanim *et al.* 2007). Microsatellites are a type of molecular markers and, as such, they are useful to characterize genotypes and to study genetic diversity within and between species. The distance between genotypes of two grapevine cultivars is defined, using microsatellite markers, as one minus the fraction of shared alleles, and this definition is used here to calculate a distance matrix `d`. Hierarchical clustering of microsatellites is prone to generate tied distances because the number of shared alleles can only take values between zero and the total number of alleles, which is usually a small number (Segura-Alabart *et al.* 2022). In this example with grapevine cultivars, where there are just six microsatellite loci, the number of pairs of genotypes separated by the same distance is relatively large and, consequently, there are very few different distances (the code for obtaining the distance matrix `d` is provided in the supplementary materials):

```
R> length(unique(d))

[1] 11
```

As a consequence of these 11 unique values out of 1275 pairwise distances in the matrix, it becomes very easy to find tied distances during the agglomeration process. The reach of

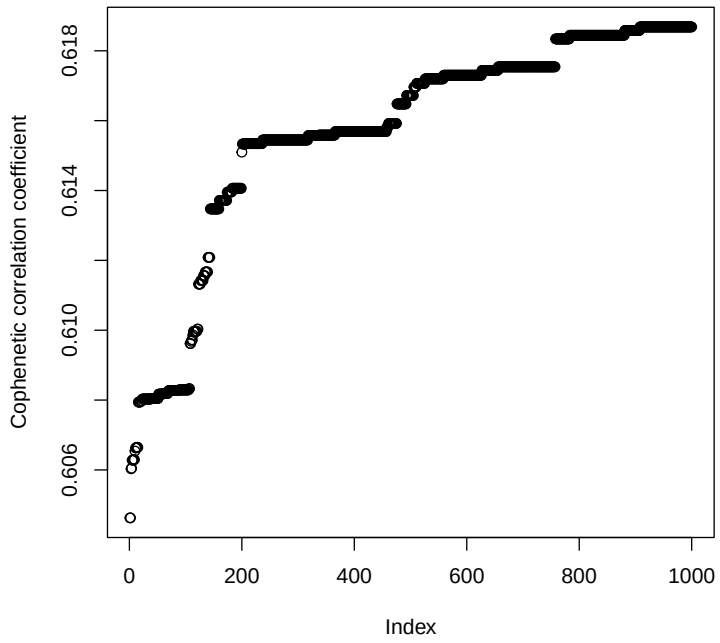

Figure 1: Sorted cophenetic correlation coefficients for the different pair-group dendrograms obtained by using random permutations of the same grapevine cultivars dataset.

the nonuniqueness problem for this example is the existence of 11 160 structurally different binary dendrograms. This number corresponds to the average linkage method and a resolution of 3 decimal digits, and it has been computed using the `Hierarchical_Clustering` tool in **Radatools** (Gómez and Fernández 2021). We can check the diversity of results by just calculating binary dendrograms for random permutations of the data and plotting the range of values of their cophenetic correlation coefficients (see Figure 1), what clearly indicates the existence of many structurally different binary dendrograms. In cases like this, where different dendrograms are possible, the reproducibility of results is compromised because, depending on the input order of data (Podani 1997), or depending on the particular implementation of the pair-group algorithm used, different results may be obtained. The interpretation of these results may be biased towards just one of the different solutions, and any conclusion drawn from a single dendrogram must be considered partial and, therefore, questionable.

2.2. Variable-group algorithm

Fernández and Gómez (2008) introduced a variable-group algorithm to ensure uniqueness in AHC, which differs from the pair-group algorithm in the following steps:

(2) Select all the groups of clusters separated by the shortest distance D_{shortest}, and merge them into several new clusters $X_I = \bigcup_{i \in I} X_i$, each one made up of several subclusters X_i indexed by i in $I = \{i_1, \ldots, i_p\}$.

(3) Compute the distances $D(X_I, X_J)$ between any two clusters $X_I = \bigcup_{i \in I} X_i$ and $X_J = \bigcup_{j \in J} X_j$, each one of them made up of several subclusters X_i and X_j indexed by i in $I = \{i_1, \ldots, i_p\}$ and j in $J = \{j_1, \ldots, j_q\}$, respectively.

When there are tied shortest distances in the agglomerative process, in order to keep track of valuable information regarding the heterogeneity of the clusters that are formed, function `linkage()` in package **mdendro** saves a fusion interval $[D_{\min}(X_I), D_{\max}(X_I)]$ for each cluster X_I made up of more than one subcluster ($|I| > 1$), where:

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

The variable-group algorithm groups more than two clusters at the same time when ties occur, giving rise to a graphical representation called multidendrogram. Its main properties are:

- When there are no ties, the variable-group algorithm gives the same results as the pair-group one.
- It always gives a uniquely-determined solution.
- In the multidendrogram representation for the results, one can explicitly observe the occurrence of ties during the agglomerative process. Furthermore, the range of any fusion interval indicates the degree of heterogeneity inside the corresponding cluster.

For example, let us suppose that we have a set of four individuals $\{x_1, x_2, x_3, x_4\}$, where the initial pairwise distances between them are:

```
R> library("mdendro")
R> d <- as.dist(matrix(c(
+    0, 2, 4, 7,
+    2, 0, 2, 5,
+    4, 2, 0, 3,
+    7, 5, 3, 0), nrow = 4))
```

Notice that there are two pairs of individuals, (x_1, x_2) and (x_2, x_3), separated by the shortest distance in the matrix, which is 2. On the one hand, using the pair-group algorithm, we can obtain three different binary dendrograms depending on the order of rows and columns in the distance matrix (see Figure 2):

```
R> par(mfrow = c(1, 4))
R> lnk1 <- linkage(d, group = "pair")
R> plot(lnk1, main = "dendrogram 1")
R> d2 <- as.dist(as.matrix(d)[c(2, 3, 4, 1), c(2, 3, 4, 1)])
R> lnk2 <- linkage(d2, group = "pair")
R> plot(lnk2, main = "dendrogram 2")
R> d3 <- as.dist(as.matrix(d)[c(4, 1, 2, 3), c(4, 1, 2, 3)])
R> lnk3 <- linkage(d3, group = "pair")
R> plot(lnk3, main = "dendrogram 3")
R> lnk4 <- linkage(d, group = "variable")
R> plot(lnk4, main = "multidendrogram")
```

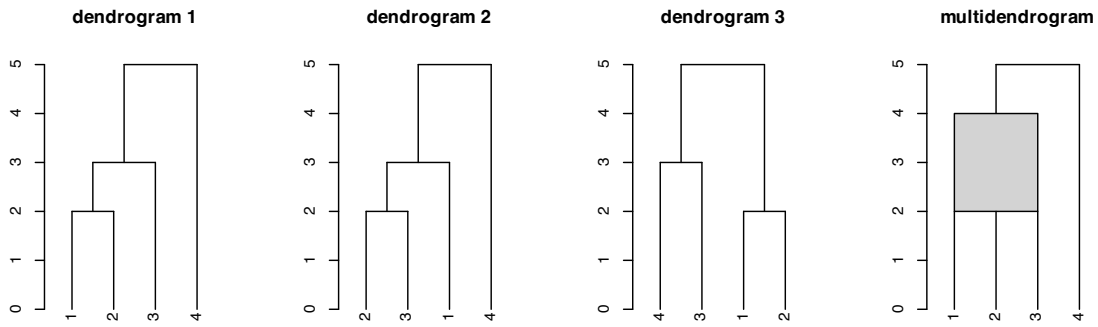

Figure 2: Different pair-group dendrograms and a unique variable-group multidendrogram, all of them obtained using the average linkage method. Observing the third dendrogram, one could get the wrong conclusion that x_3 is closer to x_4 rather than to x_1 or x_2.

On the other hand, using the variable-group algorithm, we obtain a unique multidendrogram where x_1, x_2 and x_3 are grouped in a single cluster at the same time. This new cluster is assigned two height values, corresponding to the minimum and the maximum distances separating any two of the constituent clusters $\{x_1\}$, $\{x_2\}$ and $\{x_3\}$. In this case, the minimum distance is 2 and the maximum distance is 4 (see shadowed rectangle in the multidendrogram of Figure 2). Finally, the distance between the new cluster $\{x_1, x_2, x_3\}$ and the cluster $\{x_4\}$ is calculated. In case of the average linkage method, this distance is equal to 5, that is, the arithmetic mean among the values 7, 5 and 3, corresponding respectively to the distances $d(x_1, x_4)$, $d(x_2, x_4)$ and $d(x_3, x_4)$.

We show now one of the first examples in the literature that presented the ties in proximity problem. This example corresponds to a study of 23 different soils, whose similarity data are given in Table 2 of Morgan and Ray (1995). As in the original work, similarities are transformed into dissimilarities subtracting them from 1, and the complete linkage method is used. The initial data present a tied value for pairs of soils $(3, 15)$ and $(3, 20)$, which are responsible for the existence of two different pair-group dendrograms; they are revealed by just exchanging the positions of soils 15 and 20 in the distance matrix (see Figure 3):

```
R> soils <- 1 - as.dist(read.csv("soils.csv", header = FALSE))
R> labs <- paste(1:23)
R> brown_soils <- c(1, 2, 6, 12, 13)
R> labs[brown_soils] <- paste("B-", labs[brown_soils], sep="")
R> attr(soils, "Labels") <- labs
R> lnk <- linkage(soils, method = "complete", group = "variable")
R> lnk1 <- linkage(soils, method = "complete", group = "pair")
R> s <- 1:23
R> s[c(15, 20)] <- c(20, 15)
R> soils2 <- as.dist(as.matrix(soils)[s, s])
R> lnk2 <- linkage(soils2, method = "complete", group = "pair")
R> par(mfrow = c(3, 1), mar = c(2, 4, 2, 0))
R> plot(lnk, col.rng = "pink", main = "multidendrogram")
R> plot(lnk1, main = "dendrogram 1")
R> plot(lnk2, main = "dendrogram 2")
```

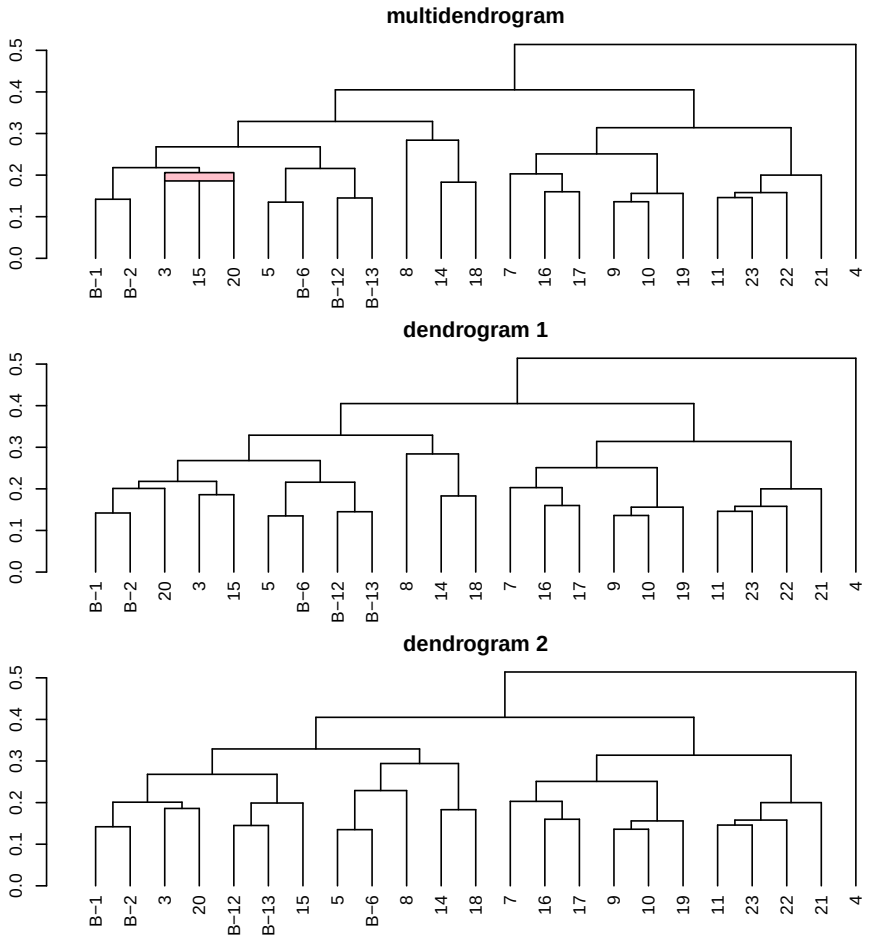

Figure 3: Complete linkage of the soils dataset yields a unique variable-group multidendrogram and two different pair-group dendrograms. The two pair-group dendrograms show important structural differences, despite simply arising from how the tied distances are broken: selecting to join first soils 3 and 15 in dendrogram 1, and soils 3 and 20 in dendrogram 2. The brown soils are marked with a "B-" prefix.

Morgan and Ray (1995) explain that the 23 soils have been categorized into eight soil groups by a surveyor. Focusing on soils 1, 2, 6, 12 and 13, which are the only members of the brown earths soil group, we can see that the second pair-group dendrogram in Figure 3 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 variable-group multidendrogram and the first pair-group dendrogram in Figure 3 are better, since they keep soils 8, 14 and 18 out of the brown earths soil group.

As we have already seen in the previous examples, with function `linkage()` we can use both the pair-group algorithm and the variable-group one by setting the `group` argument of function `linkage()` to either `"pair"` or `"variable"`, respectively.

The identification of ties requires the selection of the number of significant digits in the working dataset. For example, if the original distances are experimentally obtained with a resolution of three decimal digits, two distances that differ in the sixth decimal digit should

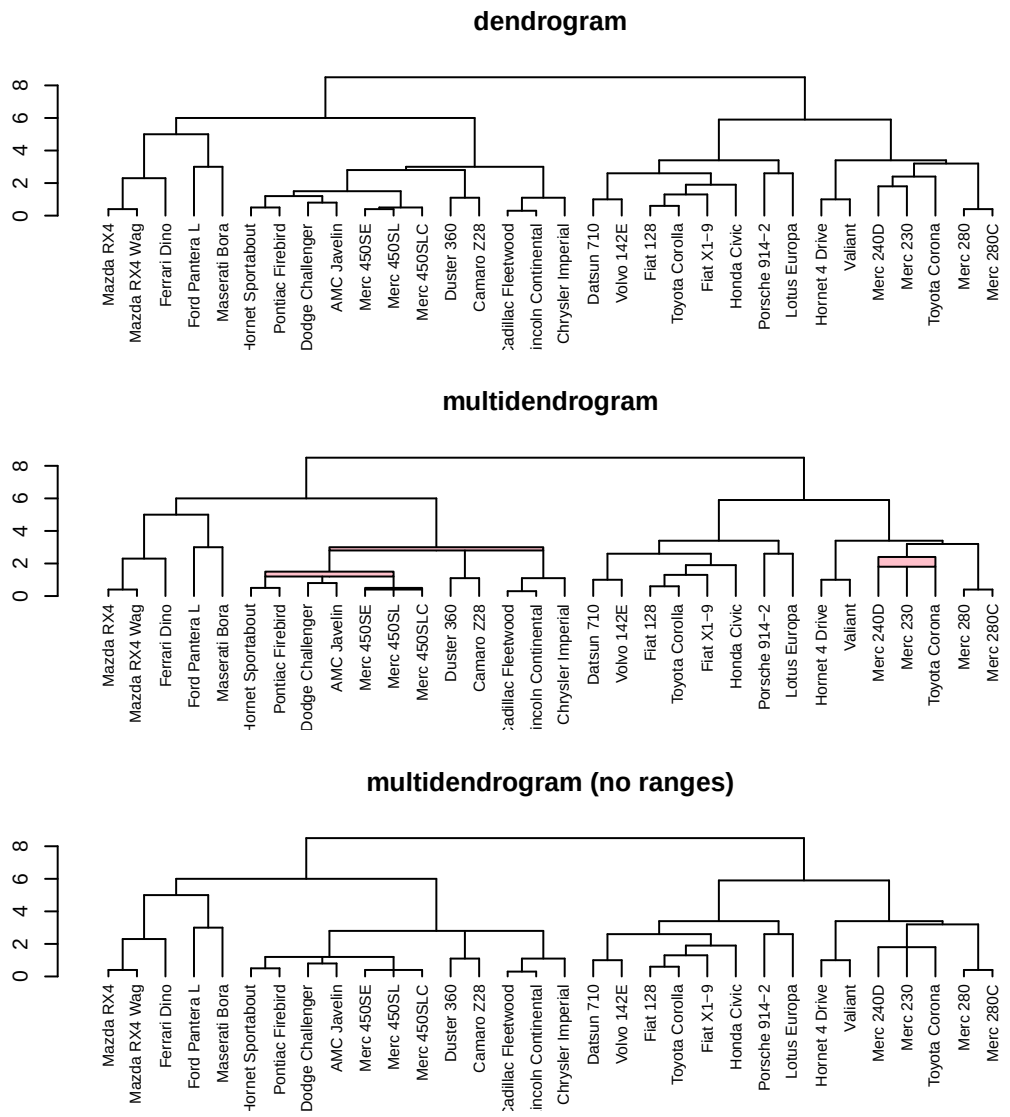

Figure 4: Pair-group dendrogram vs. variable-group multidendrogram. The ranges (rectangles) in the multidendrogram show the heterogeneity of distances within the group, but they are optional in the plots and can be hidden just by setting the `col.rng` argument in the `plot()` function to `NULL`. Distances approximated to one decimal digit both manually (top and center panels) and with the `digits` argument of function `linkage()` (bottom panel).

be considered as equal. In function `linkage()`, the user can control this level of resolution by adjusting its `digits` argument (see Figure 4):

```
R> par(mfrow = c(3, 1))
R> cars <- dist(scale(mtcars))
R> cars1 <- round(cars, digits = 1)
R> nodePar <- list(cex = 0, lab.cex = 0.7)
R> lnk1 <- linkage(cars1, method = "complete", group = "pair")
R> plot(lnk1, main = "dendrogram", nodePar = nodePar)
```

```
R> lnk2 <- linkage(cars1, method = "complete", group = "variable")
R> plot(lnk2, col.rng = "pink", main = "multidendrogram", nodePar = nodePar)
R> lnk3 <- linkage(cars, method = "complete", group = "variable", digits = 1)
R> plot(lnk3, col.rng = NULL, main = "multidendrogram (no ranges)",
+    nodePar = nodePar)
```

3. Linkage methods

3.1. Common linkage methods

During each iteration of the AHC algorithm, the distances $D(X_I, X_J)$ have to be computed between any two clusters $X_I = \bigcup_{i \in I} X_i$ and $X_J = \bigcup_{j \in J} X_j$, each one of them made up of several subclusters X_i and X_j indexed by i in $I = \{i_1, \ldots, i_p\}$ and j in $J = \{j_1, \ldots, j_q\}$, respectively. Lance and Williams (1966) introduced a formula for integrating several AHC linkage methods into a single system, avoiding the need of a separate computer program for each one of them. Similarly, Fernández and Gómez (2008) gave a variable-group generalization of this formula, compatible with the fusion of more than two clusters simultaneously:

$$D(X_I, X_J) = \sum_{i \in I} \sum_{j \in J} \alpha_{ij} D(X_i, X_j) + \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} \beta_{ii'} D(X_i, X_{i'}) + \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} \beta_{jj'} D(X_j, X_{j'}) \,. \quad (1)$$

Function `linkage()` in package **mdendro** uses this recurrence relation to compute the distance $D(X_I, X_J)$ from the distances $D(X_i, X_j)$ obtained during the previous iteration, being unnecessary to look back at the initial distance matrix $d(x_i, x_j)$ at all. The values of the parameters α_{ij}, $\beta_{ii'}$ and $\beta_{jj'}$ determine the nature of the AHC linkage methods (Fernández and Gómez 2008). Some of these methods even have weighted and unweighted forms, which differ in the weights assigned to individuals and clusters during the agglomerative process: weighted methods assign equal weights to clusters, while unweighted methods assign equal weights to individuals. Package **mdendro** implements weighted and unweighted forms of the most commonly used AHC linkage methods, namely:

- `single`: The proximity between clusters equals the minimum distance or the maximum similarity between objects.
- `complete`: The proximity between clusters equals the maximum distance or the minimum similarity between objects.
- `arithmetic`: The proximity between clusters equals the arithmetic mean proximity between objects. Also known as average linkage, WPGMA (weighted version) or UPGMA (unweighted version).
- `ward`: The distance between clusters is a weighted squared Euclidean distance between the centroids of each cluster. This method is available only for distance data.
- `centroid`: The distance between clusters equals the square of the Euclidean distance between the centroids of each cluster. Also known as WPGMC (weighted version) or UPGMC (unweighted version). This method is available only for distance data. Note that both centroid versions, weighted and unweighted, may yield inversions that make dendrograms difficult to interpret.

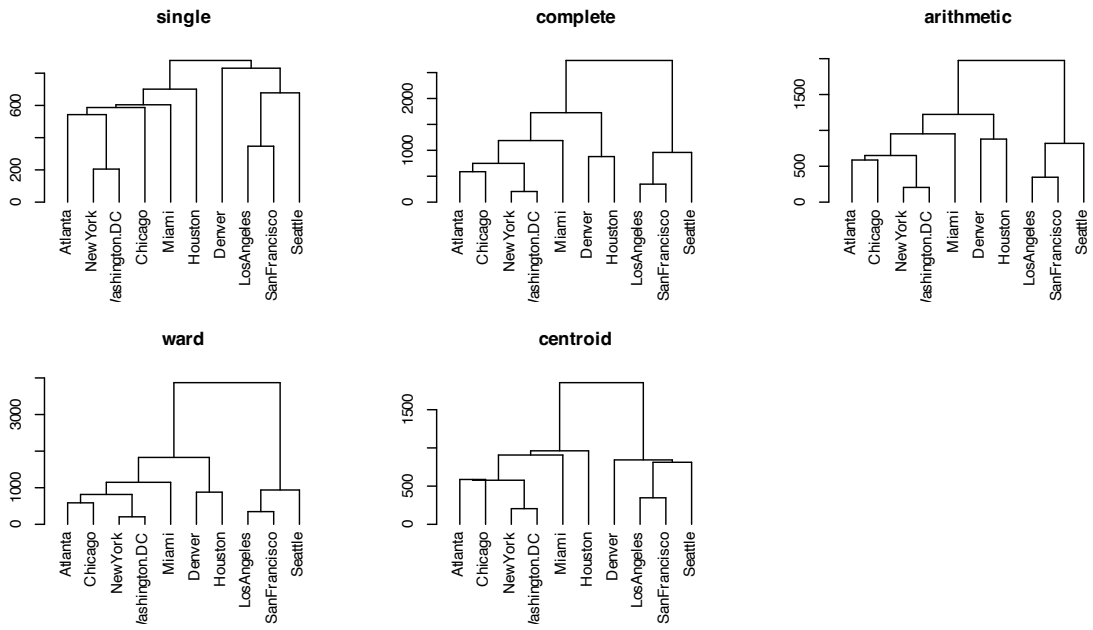

Figure 5: Common linkage methods on the `UScitiesD` dataset.

In Figure 5, we can see the differences between these AHC linkage methods on the `UScitiesD` dataset, a matrix of distances between a few US cities:

```
R> par(mfrow = c(2, 3))
R> methods <- c("single", "complete", "arithmetic", "ward", "centroid")
R> for (m in methods) {
+    lnk <- linkage(UScitiesD, method = m)
+    plot(lnk, cex = 0.4, main = m)
+  }
```

3.2. Descriptive measures

The result of function `linkage()` is an object of class '`linkage`' that describes the resulting dendrogram. In particular, this object contains the following calculated descriptors:

- `cor`: Cophenetic correlation coefficient (Sokal and Rohlf 1962), defined as the Pearson correlation coefficient between the output cophenetic proximity data and the input proximity data. It is a measure of how faithfully the dendrogram preserves the pairwise proximity between objects.

- `sdr`: Space distortion ratio (Fernández and Gómez 2020), calculated as the difference between the maximum and minimum cophenetic proximity data, divided by the difference between the maximum and minimum initial proximity data. Space dilation occurs when the space distortion ratio is greater than 1.

- `ac`: Agglomerative coefficient (Rousseeuw 1986), a number between 0 and 1 measuring the strength of the clustering structure obtained.

- `cc`: Chaining coefficient (Williams, Lambert, and Lance 1966), a number between 0 and 1 measuring the tendency for clusters to grow by the addition of clusters much smaller rather than by fusion with other clusters of comparable size.
- `tb`: Tree balance (Fernández and Gómez 2020), a number between 0 and 1 measuring the equality in the number of leaves in the branches concerned at each fusion in the hierarchical tree.

For instance, when we use function `linkage()` to calculate the complete linkage of the `UScitiesD` dataset, we obtain the following summary for the resulting dendrogram:

```
R> lnk <- linkage(UScitiesD, method = "complete")
R> summary(lnk)

Call:
linkage(prox = UScitiesD,
        type.prox = "distance",
        digits = 0,
        method = "complete",
        group = "variable")

Number of objects: 10

Binary dendrogram: TRUE

Descriptive measures:
      cor       sdr        ac        cc        tb
0.8077859 1.0000000 0.7738478 0.3055556 0.9316262
```

While multidendrograms are unique, users may obtain structurally different pair-group dendrograms by just reordering the data. As a consequence, descriptors are invariant to permutations for multidendrograms, but not for pair-group dendrograms. Let us calculate a variable-group multidendrogram and a pair-group dendrogram for the `mtcars` data:

```
R> cars <- round(dist(scale(mtcars)), digits = 1)
R> lnk1 <- linkage(cars, method = "complete", group = "variable")
R> lnk2 <- linkage(cars, method = "complete", group = "pair")
```

Now, if we apply a random permutation to data:

```
R> set.seed(1234)
R> ord <- sample(attr(cars, "Size"))
R> carsp <- as.dist(as.matrix(cars)[ord, ord])
R> lnk1p <- linkage(carsp, method = "complete", group = "variable")
R> lnk2p <- linkage(carsp, method = "complete", group = "pair")
```

We can check that the original and the permuted cophenetic correlation coefficients are identical for variable-group multidendrograms:

```
R> c(lnk1$cor, lnk1p$cor)

[1] 0.7782257 0.7782257
```

And they are different for pair-group dendrograms:

```
R> c(lnk2$cor, lnk2p$cor)

[1] 0.7780010 0.7776569
```

3.3. Parametric linkage methods

Two of the AHC linkage methods available in package **mdendro**, `flexible` and `versatile`, depend on a parameter that takes values in $[-1, +1]$ for `flexible` linkage, and in $(-\infty, +\infty)$ for `versatile` linkage. In function `linkage()`, the desired value for the parameter is passed through the `par.method` argument. This parameter works as a cluster intensity coefficient, going from space-contracting clustering strategies to space-dilating ones (Fernández and Gómez 2020). On the one hand, using `flexible` linkage with high values of the parameter or `versatile` linkage with low values of the parameter, we obtain space-contracting clusterings where dendrogram heights are shorter because clusters move closer to other clusters as they grow. On the other hand, using `flexible` linkage with low values of the parameter or `versatile` linkage with high values of the parameter, we obtain space-dilating clusterings where dendrogram heights are larger because clusters move further away from other clusters as they grow. Here come some examples on the `UScitiesD` dataset (see Figure 6):

```
R> par(mfrow = c(2, 3))
R> vals <- c(-0.8, 0.0, 0.8)
R> for (v in vals) {
+    lnk <- linkage(UScitiesD, method = "flexible", par.method = v)
+    plot(lnk, cex = 0.6, main = sprintf("flexible (%.1f)", v))
+  }
R> vals <- c(-10.0, 0.0, 10.0)
R> for (v in vals) {
+    lnk <- linkage(UScitiesD, method = "versatile", par.method = v)
+    plot(lnk, cex = 0.6, main = sprintf("versatile (%.1f)", v))
+  }
```

β-flexible linkage

Based on Equation 1, Lance and Williams (1967) proposed an infinite system of AHC strategies defined by the following constraint:

$$\underbrace{\sum_{i \in I} \sum_{j \in J} \alpha_{ij}}_{\alpha} + \underbrace{\sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} \beta_{ii'} + \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} \beta_{jj'}}_{\beta} = 1 \,,$$

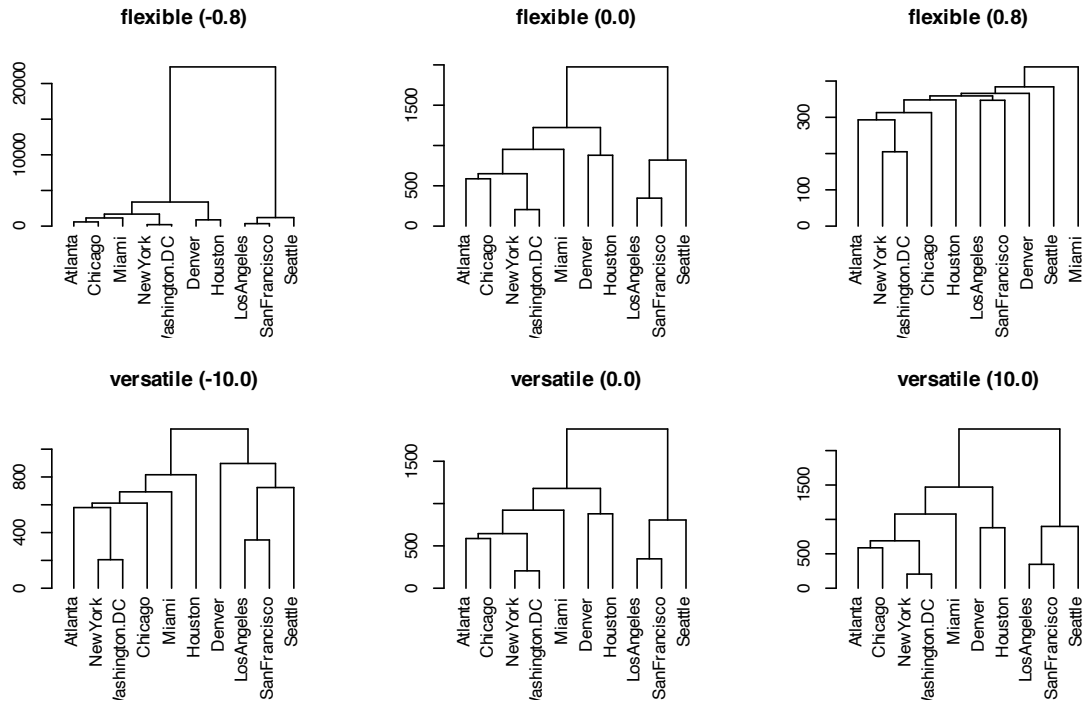

Figure 6: Parametric linkage methods on the `UScitiesD` dataset. Examples `flexible (0.8)` and `versatile (-10.0)` are more space-contracting, while examples `flexible (-0.8)` and `versatile (10.0)` are more space-dilating. Notice the huge heights of the dendrogram `flexible (-0.8)`, taking into account that the maximum value of the original distances is 2734.

where $-1 \leq \beta \leq 1$. Given a value of β, the value for α_{ij} can be assigned following a weighted approach as in the original β-flexible clustering method based on WPGMA and introduced by Lance and Williams (1966), or it can be assigned following an unweighted approach as in the β-flexible clustering method based on UPGMA and introduced by Belbin, Faith, and Milligan (1992). Further details can be consulted in Fernández and Gómez (2020). When β is set equal to 0, `flexible` linkage is equivalent to `arithmetic` linkage.

It is interesting to know how the descriptive measures depend on the parameter of the parametric linkage methods. Package **mdendro** provides the functions `descval()` and `descplot()` for this task. For example, using the `flexible` linkage method on the `UScitiesD` dataset (see Figure 7):

```
R> par(mfrow = c(2, 3))
R> measures <- c("cor", "sdr", "ac", "cc", "tb")
R> vals <- seq(from = -1, to = +1, by = 0.1)
R> for (m in measures) descplot(UScitiesD, method = "flexible",
+    measure = m, par.method = vals, type = "o",  main = m, col = "blue")
```

Versatile linkage

Package **mdendro** also implements another parametric linkage method named versatile linkage (Fernández and Gómez 2020). Substituting the arithmetic means by generalized means, also

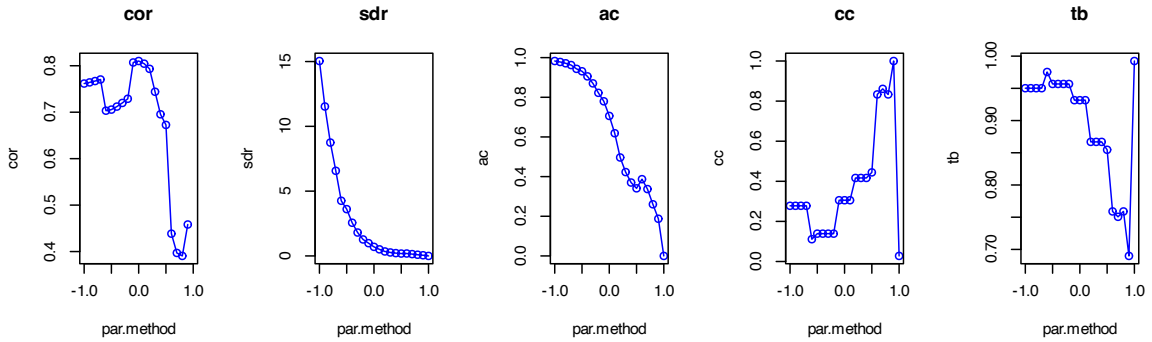

Figure 7: Descriptive measures obtained with the `flexible` linkage method on the `UScitiesD` dataset. The best cophenetic correlation coefficient (`cor`) is obtained when `par.method` is close to 0. Space distortion ratio (`sdr`), agglomerative coefficient (`ac`) and tree balance (`tb`) decrease as `par.method` increases, that is, moving from space-dilating to space-contracting clustering structures. On the contrary, chaining coefficient (`cc`) increases as `par.method` increases. Notice the exceptional behavior observed when `par.method` is 1, where `flexible` linkage yields completely flat dendrograms.

known as power means, we can extend arithmetic linkage to any finite power $p \neq 0$:

$$D_p(X_I, X_J) = \left(\frac{1}{|X_I||X_J|} \sum_{i \in I} \sum_{j \in J} |X_i||X_j|[D_p(X_i, X_j)]^p \right)^{1/p}, \tag{2}$$

where $|X_i|$ and $|X_j|$ are the number of individuals in subclusters X_i and X_j, and $|X_I|$ and $|X_J|$ are the number of individuals in clusters X_I and X_J, i.e., $|X_I| = \sum_{i \in I} |X_i|$ and $|X_J| = \sum_{j \in J} |X_j|$. Equation 2 shows that versatile linkage can be calculated using a combinatorial formula from the distances $D_p(X_i, X_j)$ obtained during the previous iteration, in the same way as the recurrence formula given in Equation 1.

Versatile linkage provides a way of obtaining an infinite number of AHC strategies from a single formula, just changing the value of the power p. The decision of what power p to use can be taken in agreement with the type of distance employed to measure the initial distances between individuals. For instance, if the initial distances were calculated using a generalized distance of order p, then the natural AHC strategy would be versatile linkage with the same power p. However, this procedure does not guarantee that the dendrogram obtained is the best one according to other criteria, e.g., cophenetic correlation coefficient, space distortion ratio or tree balance (see Section 3.2). Another possible approach consists in scanning the whole range of parameters p, calculate the preferred descriptors of the corresponding dendrograms, and decide if it is better to substitute the natural parameter p by another one. This is especially important when only the distances between individuals are available, without coordinates for the individuals, as is common in multidimensional scaling problems, or when the distances have not been calculated using generalized means.

As in the case of `flexible` linkage, the parameter p of `versatile` linkage is introduced using the `par.method` argument of the function `linkage()`. Here, it is also interesting to know how the descriptors depend on the parameter of this method (see Figure 8):

```
R> par(mfrow = c(2, 3))
```

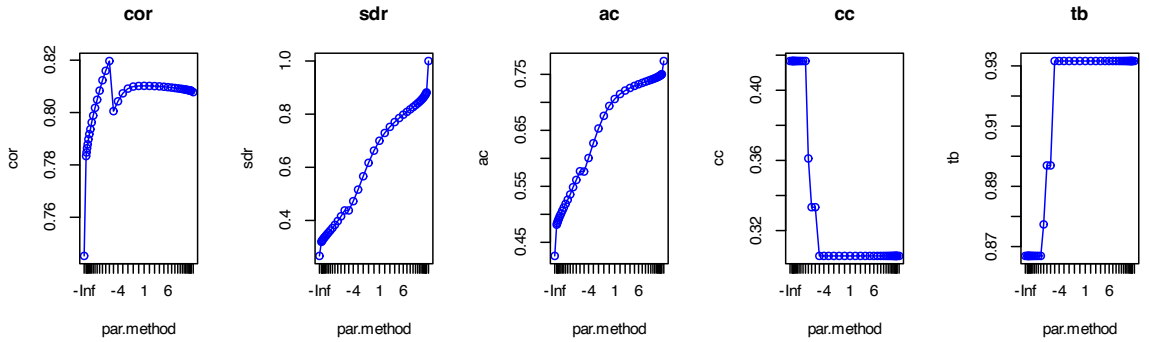

Figure 8: Descriptive measures obtained with the `versatile` linkage method on the `UScitiesD` dataset. The best cophenetic correlation coefficient (`cor`) is obtained when `par.method` is close to -6. Space distortion ratio (`sdr`), agglomerative coefficient (`ac`) and tree balance (`tb`) increase as `par.method` increases, that is, moving from space-contracting to space-dilating clustering structures. On the contrary, chaining coefficient (`cc`) decreases as `par.method` increases. Notice that, unlike `flexible` linkage, `sdr` never exceeds 1 for `versatile` linkage.

```
R> measures <- c("cor", "sdr", "ac", "cc", "tb")
R> vals <- c(-Inf, (-20:+20), +Inf)
R> for (m in measures) descplot(UScitiesD, method = "versatile",
+    measure = m, par.method = vals, type = "o",  main = m, col = "blue")
```

Particular cases. The generalized mean contains several well-known particular cases, depending on the value of the power p. Some of them reduce `versatile` linkage to the most commonly used methods, while others emerge naturally as deserving special attention:

- In the limit when $p \to -\infty$, `versatile` linkage becomes `single` linkage:

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

- In the limit when $p \to +\infty$, `versatile` linkage becomes `complete` linkage:

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

There are also three other particular cases that can be grouped together as Pythagorean linkages. Therefore, in order to emphasize the existence of different types of averages, we have preferred to rename average linkage as `arithmetic` linkage:

- When $p = +1$, the generalized mean is equal to the arithmetic mean and `arithmetic` linkage is recovered.
- When $p = -1$, the generalized mean is equal to the harmonic mean and `harmonic` linkage is obtained.

$$D_{\text{har}}(X_I, X_J) = \left(\frac{1}{|X_I||X_J|} \sum_{i \in I} \sum_{j \in J} |X_i||X_j| [D_{\text{har}}(X_i, X_j)]^{-1} \right)^{-1}.$$

- In the limit when $p \to 0$, the generalized mean tends to the geometric mean and `geometric` linkage is obtained:

$$D_{\text{geo}}(X_I, X_J) = \left(\prod_{i \in I} \prod_{j \in J} [D_{\text{geo}}(X_i, X_j)]^{|X_i||X_j|} \right)^{1/(|X_I||X_J|)} .$$

The correspondence between `versatile` linkage and the above mentioned linkage methods is summarized in Table 1. Let us show a small example in which we plot different dendrograms as we increase the versatile linkage parameter, indicating the corresponding named methods (see Figure 9):

```
R> d <- as.dist(matrix(c(
+     0,  7, 16, 12,
+     7,  0,  9, 19,
+    16,  9,  0, 12,
+    12, 19, 12,  0), nrow = 4))
R> par(mfrow = c(2, 3))
R> vals <- c(-Inf, -1, 0, 1, Inf)
R> names <- c("single", "harmonic", "geometric", "arithmetic", "complete")
R> titles <- sprintf("versatile (%.1f) = %s", vals, names)
R> for (i in 1:length(vals)) {
+    lnk <- linkage(d, method = "versatile", par.method = vals[i], digits = 2)
+    plot(lnk, ylim = c(0, 20), cex = 0.6, main = titles[i])
+  }
```

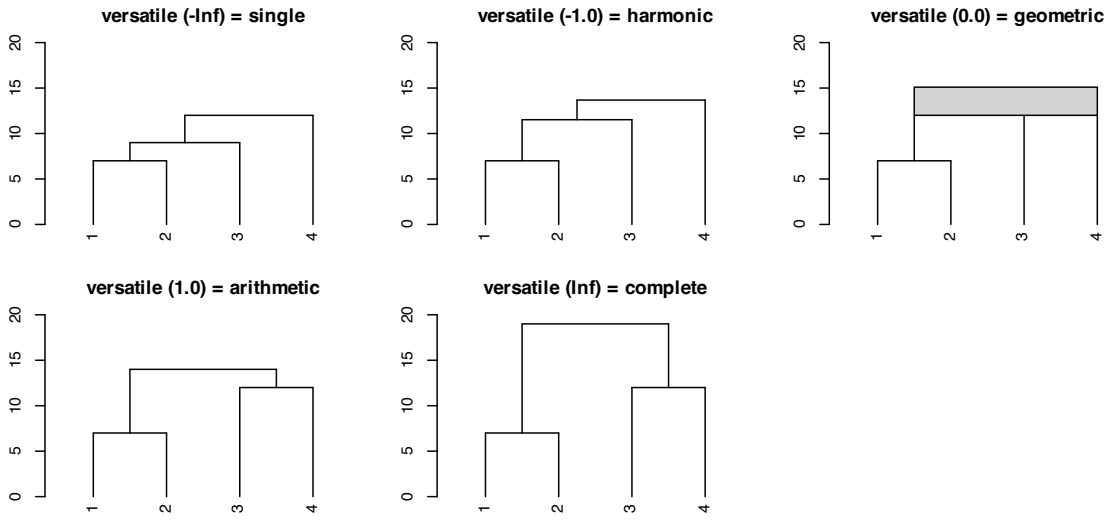

Figure 9: Example of different dendrograms obtained as we increase the versatile linkage parameter.

	versatile (par.method)
complete	+Inf
arithmetic	+1
geometric	0
harmonic	-1
single	-Inf

Table 1: Correspondence between `versatile` linkage and other linkage methods.

4. Comparison with other packages

Except for the cases containing tied distances, the equivalences in Table 2 hold between function `linkage()` in package **mdendro**, function `hclust()` in package **stats** and function `agnes()` in package **cluster**. Special attention must be paid to the equivalence with methods `centroid` and `median` of function `hclust()`, since these methods require the input distances to be squared before calling `hclust()` and, consequently, the square root of its results should be taken afterwards.

For comparison, we can construct the same AHC using the functions `linkage()`, `hclust()` and `agnes()`, where the default plots just show some differences in aesthetics (see Figure 10):

```
R> lnk <- mdendro::linkage(UScitiesD, method = "complete")
R> hcl <- stats::hclust(UScitiesD, method = "complete")
R> agn <- cluster::agnes(UScitiesD, method = "complete")
R> par(mar = c(5, 4, 4, 0), mfrow = c(1, 3))
R> plot(lnk)
R> plot(hcl, main = "")
```

linkage()	hclust()	agnes()
single	single	single
complete	complete	complete
arithmetic, U	average	average
arithmetic, W	mcquitty	weighted
geometric, U/W	—	—
harmonic, U/W	—	—
versatile, U/W, p	—	—
—	ward	—
ward	ward.D2	ward
centroid, U	centroid	—
centroid, W	median	—
flexible, U, β	—	gaverage, β
—	—	gaverage, α_1, α_2, β, γ
flexible, W, β	—	flexible, $(1-\beta)/2$
—	—	flexible, α_1, α_2, β, γ

Table 2: Equivalences between functions `linkage()`, `hclust()` and `agnes()`. When relevant, weighted (W) or unweighted (U) versions of the linkage methods and the value for `par.method` are indicated.

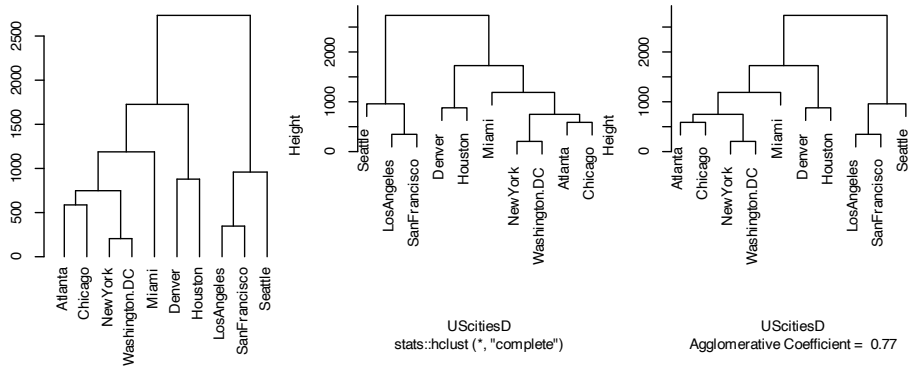

Figure 10: Comparison of `complete` linkage on the `UScitiesD` dataset, using the functions `linkage()`, `hclust()` and `agnes()`.

```
R> plot(agn, which.plots = 2, main = "")
```

The cophenetic or ultrametric matrix is readily available as component `coph` of the returned '`linkage`' object, and coincides with those obtained using the functions `hclust()` and `agnes()`:

```
R> hcl.coph <- cophenetic(hcl)
R> all(lnk$coph == hcl.coph)

[1] TRUE

R> agn.coph <- cophenetic(agn)
R> all(lnk$coph == agn.coph)

[1] TRUE
```

The coincidence also applies to the cophenetic correlation coefficient and the agglomerative coefficient, with the advantage that function `linkage()` has both of them already calculated:

```
R> hcl.cor <- cor(UScitiesD, hcl.coph)
R> all.equal(lnk$cor, hcl.cor)

[1] TRUE

R> all.equal(lnk$ac, agn$ac)

[1] TRUE
```

To enhance usability and interoperability, class '`linkage`' includes method `as.dendrogram()` for class conversion. In the example shown in Figure 10, converting to class '`dendrogram`' the objects returned by the functions `linkage()`, `hclust()` and `agnes()`, we can see that all three

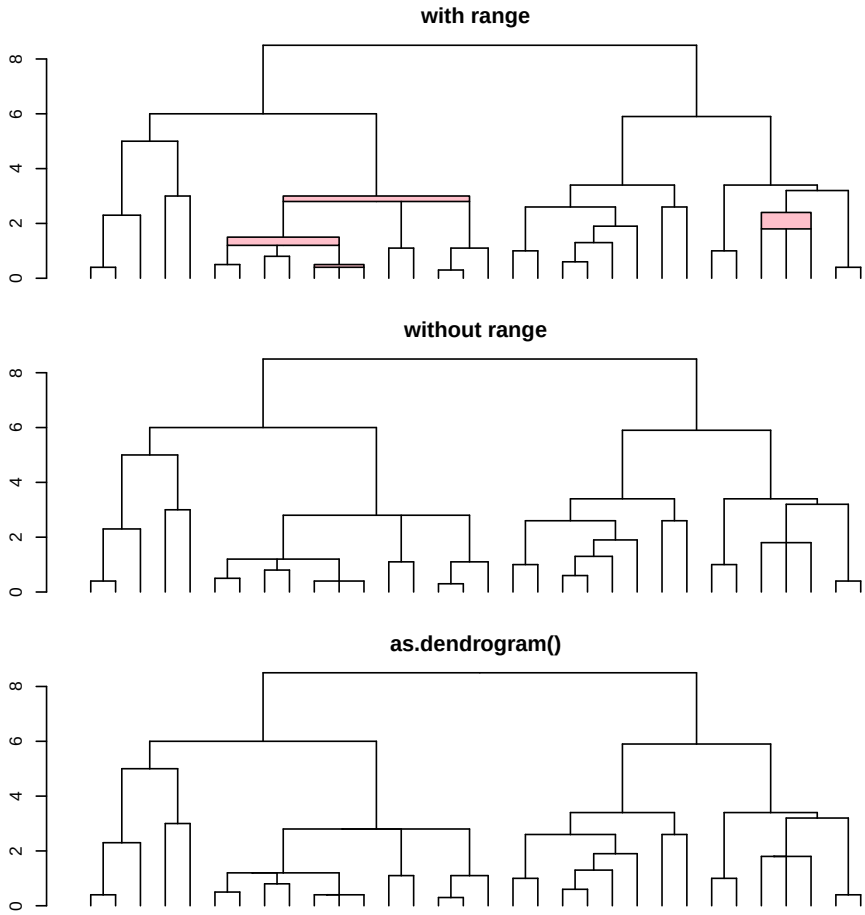

Figure 11: Example of plotting a dendrogram with ties drawing a range rectangle for tied distances, hiding it, and plotting the result returned by function `as.dendrogram()`.

dendrograms are structurally equivalent. Since class '`dendrogram`' can only handle binary edges, function `as.dendrogram()` works by converting tied distances into consecutive binary edges at the same height, thus having the same visual effect as having a single edge with more than two children. Using the same example as in Figure 4, which contains tied distances, we can observe that plotting a dendrogram with ties as returned by function `as.dendrogram()` is the same as plotting it hiding the range rectangles for tied distances (see Figure 11):

```
R> cars <- round(dist(scale(mtcars)), digits = 1)
R> lnk <- linkage(cars, method = "complete", group = "variable")
R> par(mfrow = c(3, 1), mar = c(1, 4, 2, 0))
R> plot(lnk, col.rng = "pink", main = "with range", leaflab = "none")
R> plot(lnk, col.rng = NULL, main = "without range", leaflab = "none")
R> plot(as.dendrogram(lnk), main = "as.dendrogram()", leaflab = "none")
```

The computational efficiency of functions `linkage()`, `hclust()` and `agnes()` is compared in Figure 12, where it can be observed that the time cost of functions `linkage()` and `hclust()` is quadratic (exponents 1.99 and 2.12 respectively), whereas that of function `agnes()` is cubic (exponent 3.12).

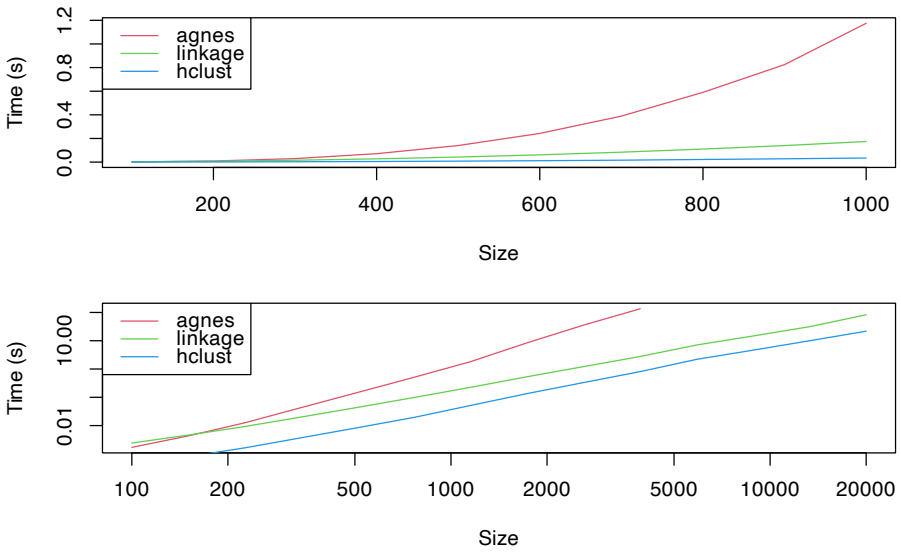

Figure 12: Comparison of the computational efficiency of functions `agnes()`, `linkage()` and `hclust()`, both in linear scale (top) and in log-log scale (bottom). The results are averages of the time taken to compute the dendrogram by each function, calculated over 20 random distance matrices for each size and method (single, complete, arithmetic and ward), and also averaging over the methods. The slope of the lines in the log-log plot indicates the different exponents of the cost.

Plots including ranges are only available if the user directly employs the `plot.linkage()` function from package **mdendro**. Anyway, the user may still take advantage of other dendrogram plotting packages, such as **dendextend** (Galili 2015) and **ape** (Paradis and Schliep 2019) (see Figure 13):

```
R> par(mar = c(5, 0, 4, 0), mfrow = c(1, 2))
R> cars <- round(dist(scale(mtcars)), digits = 1)
R> lnk <- linkage(cars, method = "complete")
R> lnk.dend <- as.dendrogram(lnk)
R> plot(dendextend::set(lnk.dend, "branches_k_color", k = 4),
+    main = "dendextend package", nodePar = list(cex = 0.4, lab.cex = 0.5))
R> lnk.hcl <- as.hclust(lnk)
R> pal4 <- c("red", "forestgreen", "purple", "orange")
R> clu4 <- cutree(lnk.hcl, k = 4)
R> plot(ape::as.phylo(lnk.hcl), type = "fan", main = "ape package",
+    tip.color = pal4[clu4], cex = 0.4)
```

And users can also use function `linkage()` to plot heatmaps containing multidendrograms (see Figure 14):

```
R> heatmap(scale(mtcars), hclustfun = linkage)
```

In addition, it is possible to work directly with similarity data without having to convert them to distances, provided they are in the range [0.0, 1.0]. Otherwise, the original similarities

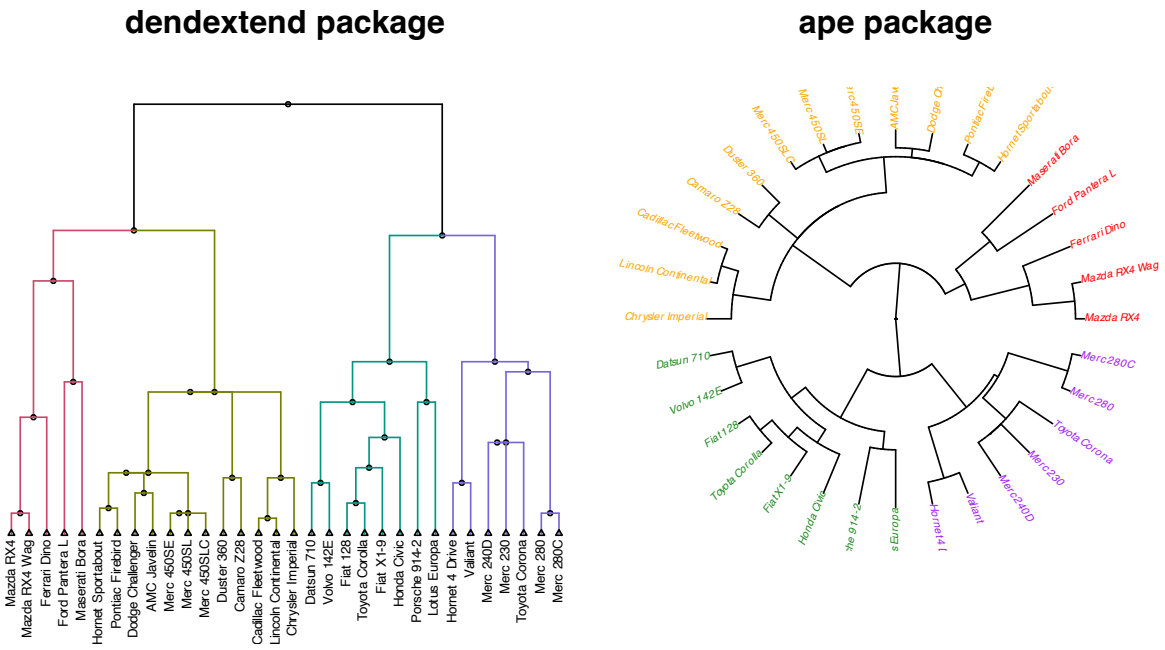

Figure 13: Converting objects of class '`linkage`' using the function `as.dendrogram()`, one can take advantage of other dendrogram plotting packages, such as **dendextend** and **ape**.

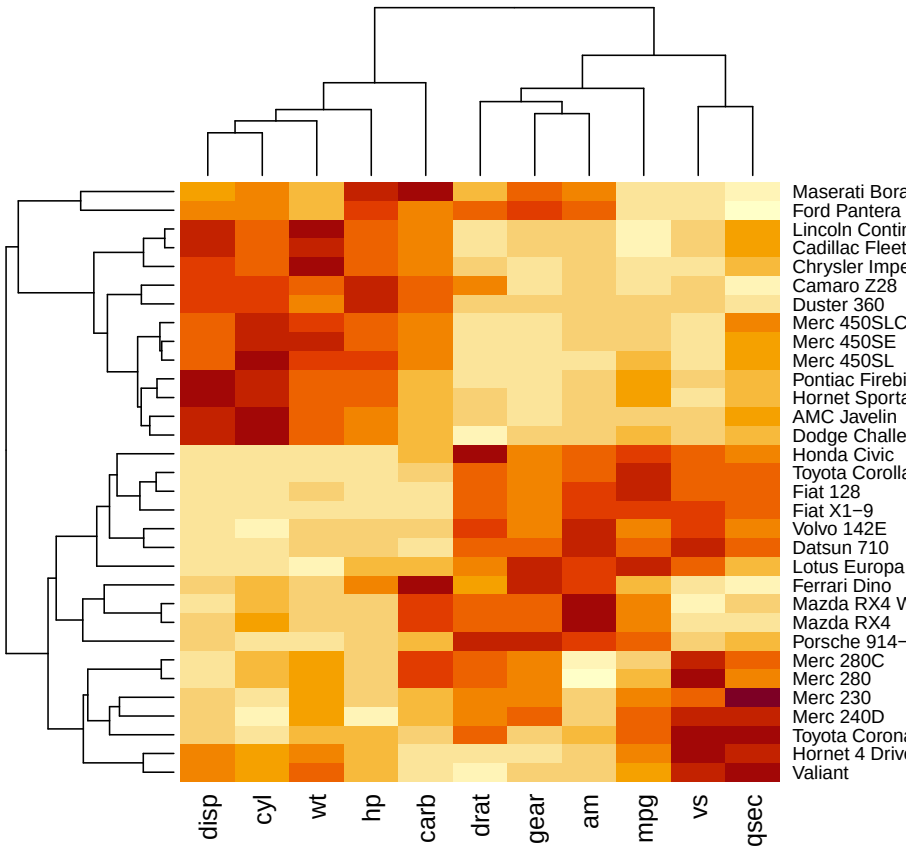

Figure 14: Example of heatmap constructed using the function `linkage()`.

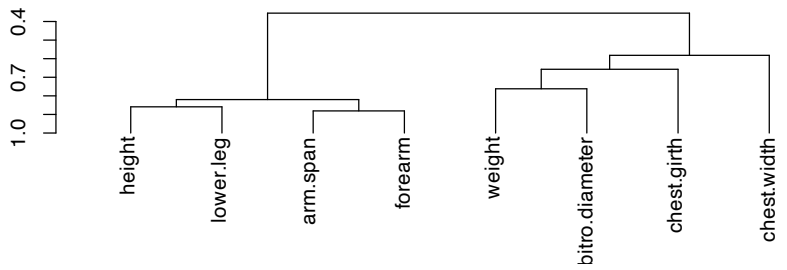

Figure 15: Example of a dendrogram constructed from a matrix of nonnegative correlations, i.e., directly using similarities instead of distances.

could be normalized to fulfill this requirement. The value 1 is selected here as the origin of the numerical axis in similarity dendrograms, just in the same way as the value 0 is the origin of the numerical axis in distance dendrograms. A typical example would be a matrix of nonnegative correlations (see Figure 15):

```
R> sim <- as.dist(Harman23.cor$cov)
R> lnk <- linkage(sim, type.prox = "sim")
R> plot(lnk)
```

It is important to remark that in the literature there exist several types of extensions of AHC for similarity data that are consistent with the standard distance formulation, such as extensions based on kernel methods (Ah-Pine 2018). The extension of function `linkage()` to similarities is based on two straightforward changes: a) in the agglomeration process, join the clusters with highest similarity (instead of those at lowest distance); b) define how to calculate the similarity between clusters. For single linkage, the similarity between clusters is defined as the maximum similarity between their components; for complete linkage, the similarity between clusters is defined as the minimum similarity between their components; and for the rest of the methods, the similarity between clusters is calculated exactly in the same way that the distance between clusters, i.e., Equations 1 and 2 are unchanged.

5. Summary and discussion

Package **mdendro** is a simple yet powerful R package to make hierarchical clusterings of data. It implements a variable-group algorithm for AHC that solves the nonuniqueness problem found in pair-group algorithms. This problem consists in obtaining different hierarchical clusterings from the same matrix of pairwise distances, when two or more shortest distances between different clusters are equal during the agglomeration process. In such cases, selecting a unique clustering 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.

Package **mdendro** computes dendrograms grouping more than two clusters at the same time when ties occur. It includes and extends the functionality of other reference packages in several ways:

- Native handling of both distance and similarity matrices.
- Calculation of variable-group multifurcated dendrograms, which solve the nonuniqueness problem of AHC when there are tied distances.
- Implementation of the most common AHC linkage methods: single linkage, complete linkage, average linkage, centroid linkage and Ward's method.
- Implementation of two parametric linkage methods: β-flexible linkage and versatile linkage. The latter leads naturally to the definition of two new linkage methods: harmonic linkage and geometric linkage.
- Implementation of both weighted and unweighted forms for the previous linkage methods.
- Calculation of the cophenetic (or ultrametric) matrix.
- Calculation of five descriptive measures for the resulting dendrogram: cophenetic correlation coefficient, space distortion ratio, agglomerative coefficient, chaining coefficient and tree balance.
- Plots of the descriptive measures for the parametric linkage methods.

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 package **mdendro**. With a single action one knows whether ties exist or not, and additionally the subsequent hierarchical clustering is obtained.

Computational details

The results in this paper were obtained using R 4.5.0 with the **mdendro** 2.2.3 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

This work was supported by Ministerio de Ciencia e Innovación (PID2021-124139NB-C22, PID2021-128005NB-C21, RED2022-134890-T and TED2021-129851B-I00), Generalitat de Catalunya (2021SGR-633) and Universitat Rovira i Virgili (2021PFR-URV-100 and 2022PFR-URV-56).

References

Ah-Pine J (2018). "An Efficient and Effective Generic Agglomerative Hierarchical Clustering Approach." *Journal of Machine Learning Research*, **19**(42), 1–43. URL http://jmlr.org/papers/v19/18-117.html.

Almadanim MC, Baleiras-Couto MM, Pereira HS, Carneiro LC, Fevereiro P, Eiras-Dias JE, Morais-Cecilio L, Viegas W, Veloso MM (2007). "Genetic Diversity of the Grapevine (Vitis Vinifera L.) Cultivars Most Utilized for Wine Production in Portugal." *Vitis*, **46**(3), 116. doi:10.5073/vitis.2007.46.116-119.

Arnau V, Mars S, Marín I (2005). "Iterative Cluster Analysis of Protein Interaction Data." *Bioinformatics*, **21**(3), 364–378. doi:10.1093/bioinformatics/bti021.

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**(2), 309–313. doi:10.1093/oxfordjournals.molbev.a025590.

Belbin L, Faith DP, Milligan GW (1992). "A Comparison of Two Approaches to Beta-Flexible Clustering." *Multivariate Behavioral Research*, **27**(3), 417–433. doi:10.1207/s15327906mbr2703_6.

Bezanson J, Edelman A, Karpinski S, Shah VB (2017). "Julia: A Fresh Approach to Numerical Computing." *SIAM Review*, **59**(1), 65–98. doi:10.1137/141000671.

Fernández A, Gómez S (2008). "Solving Non-Uniqueness in Agglomerative Hierarchical Clustering Using Multidendrograms." *Journal of Classification*, **25**(1), 43–65. doi:10.1007/s00357-008-9004-x.

Fernández A, Gómez S (2020). "Versatile Linkage: A Family of Space-Conserving Strategies for Agglomerative Hierarchical Clustering." *Journal of Classification*, **37**(3), 584–597. doi:10.1007/s00357-019-09339-z.

Galili T (2015). "**dendextend**: An R Package for Visualizing, Adjusting, and Comparing Trees of Hierarchical Clustering." *Bioinformatics*, **31**(22), 3718–3720. doi:10.1093/bioinformatics/btv428.

Gómez S, Fernández A (2021). "**Radatools** 5.2: Communities Detection in Complex Networks and Other Tools." URL https://webs-deim.urv.cat/~sergio.gomez/radatools.php.

Gordon AD (1999). *Classification*. 2nd edition. Chapman & Hall/CRC.

Hart G (1983). "The Occurrence of Multiple UPGMA Phenograms." In J Felsenstein (ed.), *Numerical Taxonomy*, pp. 254–258. Springer-Verlag. doi:10.1007/978-3-642-69024-2_30.

IBM Corporation (2021). *IBM SPSS Statistics Base 28*. IBM Corporation, Armonk. URL https://www.ibm.com/docs/en/SSLVMB_28.0.0/pdf/IBM_SPSS_Statistics_Base.pdf.

Lance GN, Williams WT (1966). "A Generalized Sorting Strategy for Computer Classifications." *Nature*, **212**, 218. doi:10.1038/212218a0.

Lance GN, Williams WT (1967). "A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems." *The Computer Journal*, **9**(4), 373–380. doi:10.1093/comjnl/9.4.373.

MacCuish J, Nicolaou C, MacCuish NE (2001). "Ties in Proximity and Clustering Compounds." *Journal of Chemical Information and Computer Sciences*, **41**(1), 134–146. doi:10.1021/ci000069q.

Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K (2021). ***cluster**: Cluster Analysis Basics and Extensions*. doi:10.32614/CRAN.package.cluster. R package version 2.1.8.1.

Morgan BJT, Ray APG (1995). "Non-Uniqueness and Inversions in Cluster Analysis." *Applied Statistics*, **44**(1), 117–134. doi:10.2307/2986199.

Paradis E, Schliep K (2019). "**ape** 5.0: An Environment for Modern Phylogenetics and Evolutionary Analyses in R." *Bioinformatics*, **35**(3), 526–528. doi:10.1093/bioinformatics/bty633.

Podani J (1997). "On the Sensitivity of Ordination and Classification Methods to Variation in the Input Order of Data." *Journal of Vegetation Science*, **8**(1), 153–156. doi:10.2307/3237253.

R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. doi:10.32614/R.manuals. URL https://www.R-project.org/.

Rousseeuw PJ (1986). "A Visual Display for Hierarchical Classification." In E Diday, Y Escoufier, L Lebart, J Pagés, Y Schektman, R Tomassone (eds.), *Data Analysis and Informatics, IV*, pp. 743–748. North-Holland, Amsterdam.

SAS Institute Inc (2018). *SAS/STAT 15.1 User's Guide*. SAS Institute Inc., Cary. URL http://documentation.sas.com/api/collections/pgmsascdc/9.4_3.4/docsets/statug/content/statug.pdf.

Segura-Alabart N, Serratosa F, Gómez S, Fernández A (2022). "Nonunique UPGMA Clusterings of Microsatellite Markers." *Briefings in Bioinformatics*, **23**(5), bbac312. doi:10.1093/bib/bbac312.

Sneath PHA, Sokal RR (1973). *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W.H. Freeman and Company.

Sokal RR, Rohlf FJ (1962). "The Comparison of Dendrograms by Objective Methods." *Taxon*, **11**(2), 33–40. doi:10.2307/1217208.

StataCorp LLC (2021). *Stata 17*. StataCorp LLC, College Station. URL http://www.stata.com/.

The MathWorks Inc (2022). *MATLAB — Statistics and Machine Learning Toolbox (R2022a)*. The MathWorks Inc., Natick. URL http://www.mathworks.com/help/stats/linkage.html.

Van der Kloot WA, Spaans AMJ, Heiser WJ (2005). "Instability of Hierarchical Cluster Analysis Due to Input Order of the Data: The **PermuCLUSTER** Solution." *Psychological Methods*, **10**(4), 468–476. doi:10.1037/1082-989X.10.4.468.

Van Rossum G, *et al.* (2011). *Python Programming Language*. URL https://www.python.org/.

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, Van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, **SciPy** 10 Contributors (2020). "**SciPy** 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods*, **17**, 261–272. doi:10.1038/s41592-019-0686-2.

Williams WT, Lambert JM, Lance GN (1966). "Multivariate Methods in Plant Ecology: V. Similarity Analyses and Information-Analysis." *Journal of Ecology*, **54**(2), 427–445. doi:10.2307/2257960.

Wolfram Language & System Documentation Center (2020). *Mathematica 12.1 — Hierarchical Clustering Package Tutorial*. Wolfram Research Inc., Champaign. URL https://reference.wolfram.com/language/HierarchicalClustering/tutorial/HierarchicalClustering.html.

Affiliation:

Alberto Fernández
Universitat Rovira i Virgili
Departament d'Enginyeria Química
Av. Països Catalans 26
43007 Tarragona, Spain
E-mail: alberto.fernandez@urv.cat

Sergio Gómez
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
Av. Països Catalans 26
43007 Tarragona, Spain
E-mail: sergio.gomez@urv.cat
URL: https://webs-deim.urv.cat/~sergio.gomez/

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
September 2025, Volume 114, Issue 2 *Submitted:* 2024-02-03
doi:10.18637/jss.v114.i02 *Accepted:* 2024-09-19