MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **54** (2005) 561-582

ISSN 0340 - 6253

Techniques of Rearrangements in Binary Trees (Dendrograms) and Applications

Hans-Joachim Mucha[1], Hans-Georg Bartel[2] and Jens Dolata[3]

1 Weierstraß-Institut für Angewandte Analysis und Stochastik, Mohrenstraße 39, D-10117 Berlin, Germany, E-Mail: mucha@wias-berlin.de
2 Humboldt-Universität zu Berlin, Institut für Chemie, Brook-Taylor-Straße 2, D-12489 Berlin, Germany.
3 Landesamt für Denkmalpflege Rheinland-Pfalz, Abt. „Archäologische Denkmalpflege“, Amt Mainz, Große Langgasse 29, D-55116 Mainz, Germany, E-Mail: dolata@em.uni-frankfurt.de

Abstract

Hierarchical cluster analysis is a well-known method of stepwise data compression. As a result one gets a dendrogram, that is, a special binary tree with a distinguished root and with all the data points (objects) at its leaves. Unfortunately both the real or potential order of the objects and the potential quantitative locations of the objects are not reflected in the dendrogram. Often, neighbouring objects in the dendrogram are quite distinct from one to each other in the reality of a heterogeneous, high-dimensional setting. Therefore the reading of conventional dendrograms as well as their interpretation becomes difficult and it is often confusing. Here some dendrogram drawings and reordering techniques are proposed that reflect the total order in the one-dimensional case (univariate case) and, in the multivariate case, an order that corresponds approximately to a total order in some degree. The result, a so-called ordered dendrogram, is recommended because it makes the interpretation of hierarchical structures much easier. The proposed dendrogram reordering and drawing techniques are applied on high-dimensional data points of chemical compounds.

1. Introduction

Most generally, cluster analysis aims at finding interesting structures or clusters directly from the datasets without using any background knowledge about structures. There are model-based as well as heuristic clustering techniques. At most, one will set up new hypotheses about the data. At least, clustering should result in practically useful partitions or hierarchies. Here the family of hierarchical clustering techniques will be considered only. They start with pairwise proximities (distances, similarities) between objects. Then, in a stepwise manner, they form clusters by amalgamations of pairs of similar objects and/or clusters. A hierarchical clustering method gives a unique solution. Usually the steps of the algorithm of hierarchical clustering are presented in a dendrogram, that is, a special binary tree with a distinguished root and with all the data points (objects) at its leaves. Some original methods of cluster analysis as well as modified ones are based on graph theory [1]. For instance, the algorithm of *minimum spanning tree* (usual synonyms are *Single Linkage method* or *Nearest Neighbour*) is such a well-known technique of hierarchical cluster analysis (HCA) with roots in graph theory [2, 9, 11, 12]. As already mentioned above, dendrograms are the graphical output of HCA. Alternatively, they often are called binary trees in graph theory. However, dendrograms record both the steps of agglomerative or divisive clustering and the quantitative increment of distances between the emerging clusters (see at the top of Figure 1: the axis shows the distance levels of merging clusters). One of the most difficult tasks of hierarchical cluster analysis remains: finding an appropriate number of clusters by cutting the dendrogram at a certain distance value. An automatic validation technique for hierarchical cluster analysis is recommended that can be considered as a so-called built-in validation of the number of clusters and of each cluster itself, respectively [35, 36]. Improved dendrograms, that will be proposed here, can support these decisions.

Concerning an extensive consideration about dendrograms and trees in the framework of graph theory the reader is referred to [3]. Another reference concerning dendrograms is [8] where the authors propose a new interactive interface to help the user to interpret dendrograms. Here we recommend also improvements in the graphical presentations of dendrograms, but we will reach this aim in a quite different way.

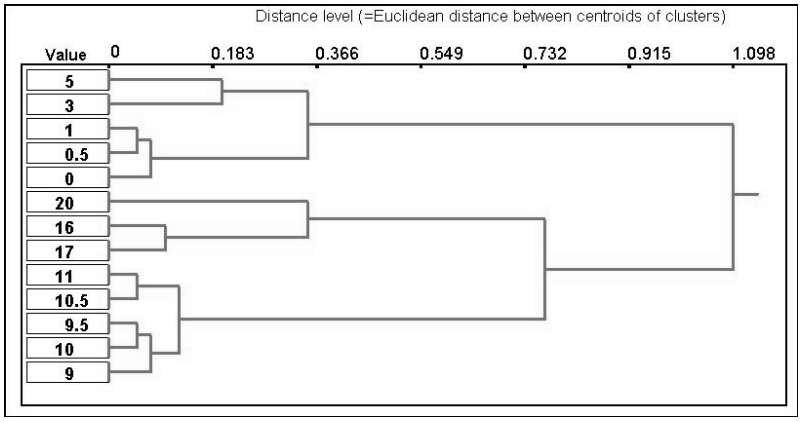


Figure 1: Conventional dendrogram of the result of the *unweighted pair-group method using centroids (UPGMC)*. Instead of names here the values of the objects (leaves) are presented at the left hand side.

Unfortunately both the real or potential order of the objects and potential quantitative locations of the objects are not reflected in the dendrogram. Conventionally, the ordering of the objects (leaves) is arbitrary in certain degree and the leaves are drawn equidistant (see Figure 1). Therefore often, neighbouring objects in the dendrogram are quite distinct from one to each other in the reality of distance in a high-dimensional setting. As a consequence, the reading of conventional dendrograms as well as their interpretation becomes difficult and it is often confusing. Here some dendrogram drawing and reordering techniques are proposed that reflect the total order in the one-dimensional case (univariate case) and, in the multivariate case, an order that corresponds approximately to a total order in some degree. To illustrate this idea, let us go through a binary tree from the root to the leaves. In order to guarantee the total order one has to stop at a certain level (i.e. a certain number of clusters). The result is a truncated tree (called ordered dendrogram) that satisfies a given order of the nodes. By the way, in hierarchical clustering the main question arises: What is the right number of clusters (groups, subsets), i.e. when should the aggregation stop without losing the essential character of the data? The ordered dendrogram can a little bit better support this decision than the conventional dendrograms.

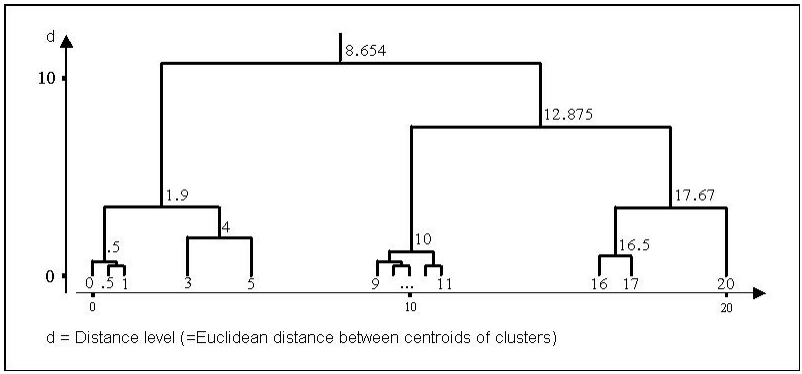


Figure 2: The "rearranged" dendrogram of Figure 1 reflecting both the total order and the native quantitative position of the objects (leaves). Here an example of reading at the right hand side: The leaves with values 16 and 17 will be merged together to the new node with value 16.5. Some steps of the algorithm later on, this node will be merged together with the leave with value 20 to the bigger node with value 17.67. This value is the average over the values of the corresponding three leaves.

The focus of this paper is on ordering of objects in dendrograms only. A dendrogram can be seen as a binary tree with a distinguished root (right hand side of Figure 1) and with all the objects at its leaves (left hand side of Figure 1). Additionally the distance levels of merging the clusters are reported in a dendrogram. In the following monotone non-decreasing distance levels are assumed during the merging process. It should be mentioned that some of the pairwise agglomerative clustering methods can lead to decreasing distance levels. Furthermore, the problem of violation of the uniqueness of the merging process will not be discussed here, because equal distance levels (i.e., non-increasing levels) do not really affect the drawing of dendrograms. In the framework of graph theory, distinct edge weights $w(e)$, $e \in \mathbf{E}$, of a connected, undirected, weighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ guarantee that the corresponding minimum spanning tree is unique. Here $\mathbf{V}$ and $\mathbf{E}$ are the set of vertices (nodes) and edges, respectively. For more details on graph theory see [18, 19].

As an appetizer let us consider the dendrogram of the tiny dataset of $I = 13$ objects in $\mathbb{R}^1$ in more detail. Without any doubt there is a total order of the objects in $\mathbb{R}^1$ given by their values. Figure 1 shows both the data values of the objects at the left hand side and the result of hierarchical clustering. Unfortunately for an easy reading of dendrograms,

the ordering of the objects is arbitrary and the leaves are drawn equidistantly by conventional statistical software. Really, the order of leaves in conventional dendrograms is arbitrary because usually it depends on two factors at least. First it depends on how the objects are stored in the dataset, i.e., in which succession they occur. Second it depends on the algorithms that are used for drawing dendrograms. Generally, these algorithms rearrange the given order of the leaves (in the dataset) in order to avoid the crossing of the branches (lines) of the tree. Some examples of well-known statistical software, often with a long history, are: *CLUSTAN* [15], SPSS [16], SAS [17], S and S-PLUS [20]. Figure 1 shows such a conventional dendrogram. Here the reading as well as the interpretation becomes difficult and it is simply confusing.

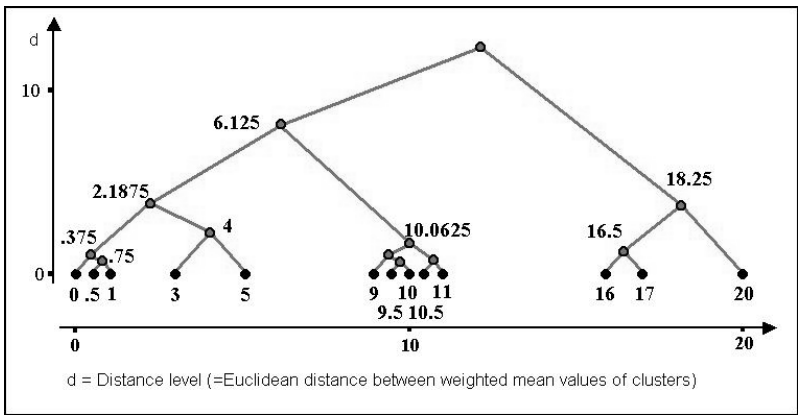
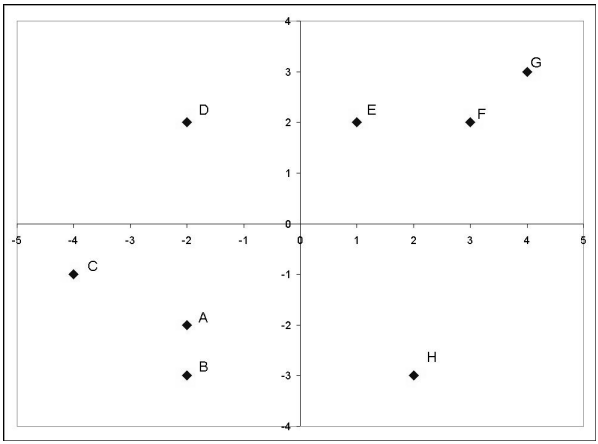


Figure 3**:** "Triangle shape" dendrogram of the result of the *weighted pair-group method using arithmetic averages (WPGMA)* reflecting both the total order and the native quantitative position of the objects (leaves).

The well-known *Centroid* method is applied (see for more details [4, 11, 13, 14]). Usual synonyms of the *Centroid* method are *unweighted pair-group method using centroids* (*UPGMC*) or *centroid sorting*. This method is based on the (squared) Euclidean distance between the objects. In Figure 1 one can see that some leaves with quite distinct values are drawn aside in the tree. For example, the most distinct leaves with the values 0 and 20 are located side by side in the dendrogram.

Figure 2 shows the result of the *Centroid* method in a more informative manner (see Figure 1 for a comparison). Here the leaves are drawn non-equidistantly and they are rearranged in their total order, i.e. they occur in their native order. Additionally most of the terminal nodes (at the bottom of the figure) and non-terminal nodes (branching points) are marked by their value. Therefore let's illustrate briefly the algorithm of the pairwise agglomerative clustering method that is applied here, namely the *unweighted pair-group method using centroids*. The values of 13 terminal nodes (= 13 separate trivial clusters of one object apiece) at the bottom of Figure 2 are the starting point. For each step in the algorithm the closest two clusters in terms of the Euclidean distance between their centroids are successively merged to a bigger one, i.e., the two most similar clusters are replaced by a new one. The value of the new cluster becomes equal to the unweighted average (centroid) of the values of the two corresponding small clusters. The next steps repeat the same procedure of finding the closest two clusters (nodes), calculating the average and merging these two nodes to a new one. At the end the cluster at the left hand side consisting of five objects (average = 1.9) and the cluster at the right hand side consisting of all remaining objects (average = 12.875) are merged together at a distance level of 10.975 (= 12.875 – 1.9). At the left hand side of Figure 2 (see also at the top in Figure 1) the axis of the (Euclidean) distance levels between clusters is drawn.

An alternative representation of binary trees is shown in Figure 3. Here the same dataset is used as in Figures 1 and 2, respectively. However, another simple automatic pairwise agglomerative clustering method is applied, namely the *weighted average linkage*. Usual synonyms are *weighted pair-group method using arithmetic averages* (*WPGMA*) or *simple average linkage*. This stepwise algorithm is similar to the above one. It begins with 13 terminal clusters of one leave apiece. For each step the closest two clusters in terms of the Euclidean distance between their values are merged successively to a bigger one. The value of the new cluster becomes equal to the weighted average of the values of the two corresponding small clusters. At the end of the stepwise procedure the cluster at the right hand side consisting of three objects (weighted average = 18.25) and the cluster at the left hand side consisting of all remaining objects (weighted average = 6.125) are merged together at a distance level of 12.125 (= 18.25 – 6.125). For further details on these and other hierarchical methods and on dendrograms see [4,9,21,25].

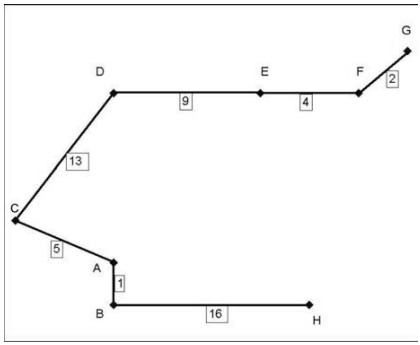


Name	X	Y
A	-2	-2
B	-2	-3
C	-4	-1
D	-2	2
E	1	2
F	3	2
G	4	3
H	2	-3

Figure 4**:** Plot of an example of eight points in a tiny dataset in $\mathbb{R}^2$ (left hand side), and the corresponding data table at the right hand side.

Unfortunately, the property of total order of objects is usually limited to the univariate case. Figure 4 shows a tiny two-dimensional artificial data set. Obviously it is a hard problem to find a proper rearrangement of the points in the subspace $\mathbb{R}^1$. The minimum spanning tree of these eight points can be obtained by the *Single Linkage* method. It is shown in Figure 5. Here the squared Euclidean distance (4) (see below) is used to weight the edges $e \in \mathbf{E}$ of the connected, undirected, weighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. By cutting some edges in the minimum spanning tree subgraphs (clusters, subtrees) are obtained. The conventional dendrogram of the *Single Linkage* cluster analysis is shown in Figure 6.

Generally in the multivariate case, an approximate order can be reached by using projection methods like principal component analysis or discriminate analysis. This order is approximate in some sense by taking into account an appropriate number of clusters K (equivalence classes). The latter are the result of hierarchical clustering and they are in total order for at most K clusters. The greater K the better the used projection method is in view of rearranging the leaves in the dendrogram.



Squared Euclidean Distances

Name	A	B	C	D	E	F	G	H
A	0	1	5	16	25	41	61	17
B	1	0	8	25	34	50	72	16
C	5	8	0	13	34	58	80	40
D	16	25	13	0	9	25	37	41
E	25	34	34	9	0	4	10	26
F	41	50	58	25	4	0	2	26
G	61	72	80	37	10	2	0	40
H	17	16	40	41	26	26	40	0

Figure 5**:** Minimum spanning tree with $I = 8$ vertices (nodes, leaves) and 7 edges (left hand side) and the corresponding distance matrix at he right hand side. The total (minimum) weight of the MST is equal to 50. The MST of this tiny dataset is a unique one.

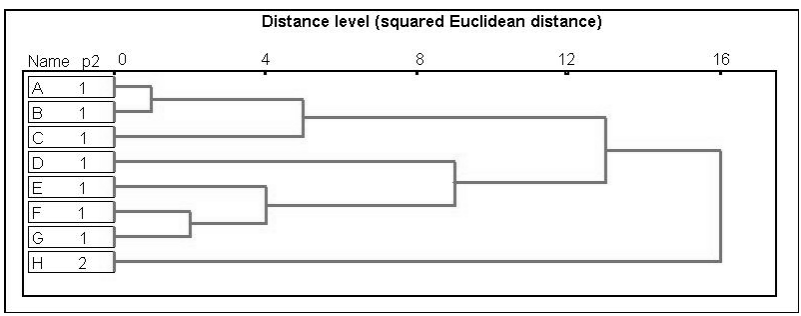


Figure 6**:** Dendrogram of *Single Linkage* cluster analysis of the set of eight points of Figure 4.

An underlying hypothesis for application of many distance measures like the Euclidean one is that the variables are measured in the same scale. If this is not the case (as in many statistical applications, see section 5 below) a standardization of the data should be first applied. Otherwise, the statistical analysis of ranks is an interesting special case of order statistics, i.e. transforming the original variables into quantiles. (This case will be not traced here, for an impression on this see, for instance [4].) In that way, both cluster analysis and principal component analysis become independent of the scales of variables. The latter can be used for getting an approximate order of objects.

2. Simple model-based Gaussian hierarchical cluster analysis

Often (weighted) Euclidean distances are the basis of both the methods of cluster analysis and projection methods. The ideal case would be the use of the same distance measure in both families of multivariate methods. Concerning model-based clustering the paper [5] give a comprehensive insight into the topic. Some relevant relations between simple model-based Gaussian clustering and graph theory are considered in [1].

Let a sample of I observations in $\mathbb{R}^J$ be given. Let the matrix $\mathbf{X} = (x_{ij})$ denote this sample. A partition $\boldsymbol{P}(I, K)$ is an exhaustive subdivision of the set of I objects into K non-empty clusters C_k which are pairwise disjoint. Let us focus on simple covariance structures. When the covariance matrix is constrained to be diagonal and uniform across all groups, the sum-of-squares criterion

$$V_K = tr(\sum_{k=1}^{K} \mathbf{W}_k) \qquad \text{(Eq. 1)}$$

has to be minimized for fixed K. Here

$$\mathbf{W}_k = \sum_{i \in C_k} (\mathbf{x}_i - \overline{\mathbf{x}}_k)(\mathbf{x}_i - \overline{\mathbf{x}}_k)^T \qquad \text{(Eq. 2)}$$

is the sample cross-product matrix for the kth cluster with the usual maximum ressehood estimate $\overline{\mathbf{x}}_k$ of its expectation value. An equivalent formulation of (1) without explicit specification of cluster centres $\overline{\mathbf{x}}_k$ is:

$$V_K = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} \sum_{\substack{l \in C_k \\ l>i}} d_{il}\,, \qquad \text{(Eq. 3)}$$

where n_k is the number of observations in kth cluster and d_{il} are the pairwise squared Euclidean distances

$$d_{il} = \sum_{j=1}^{J} \left(x_{ij} - x_{lj}\right)^2 = \left\|\mathbf{x}_i - \mathbf{x}_l\right\|^2 \qquad \text{(Eq. 4)}$$

between the two observations I and l. The well-known *Ward*'s method (synonym: *minimum variance method*) starts with pairwise squared Euclidean distances between terminal clusters and minimises the criterion (1) by agglomerative hierarchical clustering [6]. Terminal clusters consist of a single object only, i.e. each object is in a cluster by itself. Figures 7 and 8 show the same results of *Ward*'s method applied to the tiny

dataset, but in different fashions. The linear multivariate projection method *principal components analysis* [2, 23] is used in Figure 8 in order to find an appropriate order of objects. Obviously this attempt fails here. With respect to the first principal axis the order of the objects is simply denoted by the running numbers. Figure 9 shows the quality of □resservation of distances when using the first principal component.

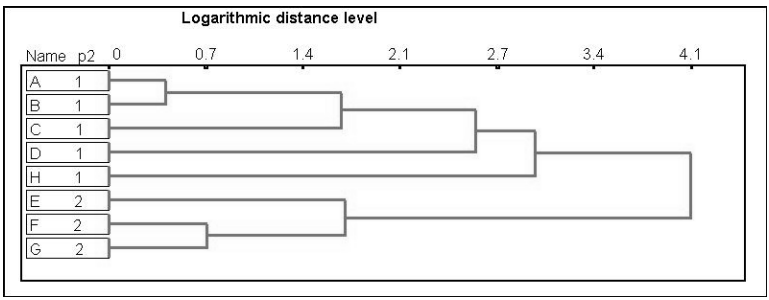


Figure 7 **:** Dendrogram of *Ward*'s hierarchical clustering of the set of eight point of Figure 4. The distance axis points up the increment of within-clusters variances during the process of fusions.

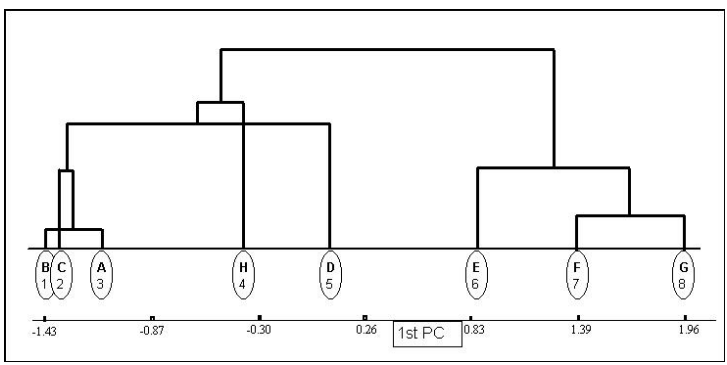


Figure 8: Non-equidistant rearranged dendrogram of *Ward*'s hierarchical clustering of the tiny dataset of the eight points of Figure 4. The set of objects {A, B, …, H} is rearranged by the 1st principal component. Below the names of the objects and their order is marked by the running numbers 1, 2, …, 8. There are some intersections of the branches of the tree.

When the covariance matrix of each cluster is constrained to be diagonal, but otherwise allowed to vary between groups, the logarithmic sum-of-squares criterion

$$U_K = \sum_{k=1}^{K} n_k \log tr(\mathbf{W}_k / n_k) \qquad \text{(Eq. 5)}$$

has to be minimized. Once again an equivalent formulation holds:

$$U_K = \sum_{k=1}^{K} n_k \log \left(\sum_{i \in C_k} \sum_{\substack{j \in C_k \\ j>i}} \frac{1}{n_k^2} d_{ij}\right) \qquad \text{(Eq. 6)}$$

According to (5) or (6) an agglomerative hierarchical method like *Ward*'s method was proposed by [7]. It is named here *logarithmic Ward*.

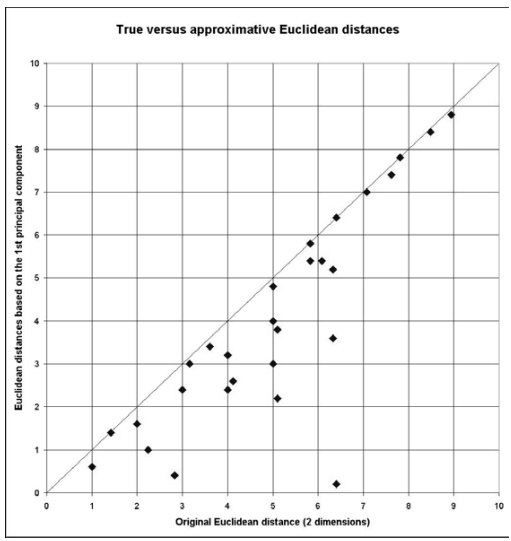


Figure 9: Graphic display of the quality of preservation of distances after projection of the eight points from $\mathbb{R}^2$ in $\mathbb{R}^1$. For instance, the true distance d(D, H)= $\sqrt{41}$ (≈ 6.4) between object D and object H is broken down to about 0.2 by the projection (see the well isolated symbol at the bottom).

A three-dimensional dendrogram is one way out when the rearrangement of objects in $\mathbb{R}^1$ fails. Here the tree is drawn on a plane. Figure 10 shows the same result of *Ward*'s method as Figure 7 and 8. In comparison with Figure 7 the number of crossing branches of the tree can be slightly reduced.

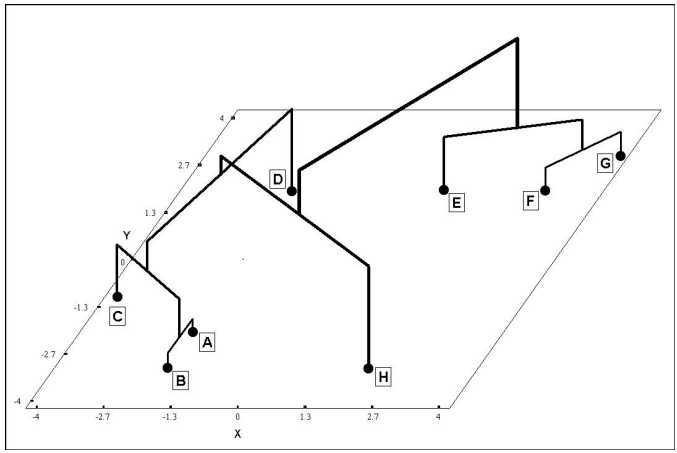


Figure 10**:** A three-dimensional dendrogram of *Ward*'s hierarchical clustering. Usually, such a "plot-dendrogram" is drawn on a plane that is the result of projection methods like principal components analysis. Here the original co-ordinates of the points are used.

3. Approximate order of clusters

Let's assume, an order of I objects is given. Furthermore, let's assume that a hierarchical cluster analysis results in a set of partitions. The aim of the following algorithm is to determine the number of clusters q so that all the clusters can be drawn without crossing the branches in an "ordered" dendrogram. The order of objects is assumed to be labelled by the running numbers 1, 2, …, I. Below the following notation is used:

- I : number of objects
- $c_i = I - i + 1$: number of classes on step i ($i = 1, 2, \ldots, I$)
- $\boldsymbol{P} = \{\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \ldots, \boldsymbol{P}^{(I)}\}$: set of partitions
  - with $\boldsymbol{P}^{(i)} = \{P_1^{(i)}, P_2^{(i)}, \ldots, P_{c_i}^{(i)}\} \in \boldsymbol{P}$
    - with $P_j^{(i)} \in \boldsymbol{P}^{(i)}$: j^{th} partition on step i ($i = 1, 2, \ldots, I; j = 1, 2, \ldots, c_i$)
- $\boldsymbol{M} \equiv \boldsymbol{P}^{(I)} = \{1, 2, \ldots, I\}$: set of objects
  - with $\{1, 2, \ldots, I\} = \mathbb{N}^{(I)} \subset \mathbb{N}$
    - with $\mathbb{N}$: set of natural numbers
- q : number of clusters that is available
  due to the ordering $(1, 2, \ldots, I) \in (\mathbb{N}^{(I)})^I$

Here $\boldsymbol{P}^{(1)}$ and $\boldsymbol{P}^{(I)}$ are the trivial partitions into the I clusters {1}, {2},…,{I} and into one cluster {1, 2,…,*I*}, respectively. The following symbolic algorithm *ClusterOrder* finds the number of clusters q that corresponds to a given order of objects in the sense of maximum number of non-crossing branches by starting at the root of the dendrogram and going down to the leaves (objects).

INPUT $\boldsymbol{P}$
FOR $i = I - 1$ TO 1
  IF $i = 1$ THEN $q = I$: END
  FOR $j = 1$ TO $I - i + 1$
    IF $\max(P_j^{(i)}) - \min(P_j^{(i)}) + 1 > \mathrm{card}(P_j^{(i)})$ THEN $q = I\text{-}i$: END
  NEXT j
NEXT i

Figure 11 The algorithm *ClusterOrder*.

The greater q the better the agreement is between the given order and the distances between clusters that built up the partitions. Each cluster analysis method computes the distances between clusters in a specific way based on the true distances between objects.

Examples:

1) Eight points in R^2

For a better understanding of the algorithm let's look at the tiny dataset of Figure 8. The rearranged objects in Figure 8 are denoted simply by the usual running numbers 1, 2,…, 8. These identifiers are used in the partitions and corresponding subsets (classes). With respect to the given order of objects by the first principal component the set of partitions is investigated starting from the root of the tree. The first split gives the partition into the two clusters {1, 2, 3, 4, 5} and {6, 7, 8}. This partition corresponds with the given order. In the next split however, the condition of the algorithm is not fulfilled for the cluster {1, 2, 3, 5} and the algorithm stops and returns q = 2. As a consequence, for more than two clusters the dendrogram contains crossing lines.

2) Artificial dataset (I = 3 observations)

Let us look at the following partitions:

P(3)={1, 2, 3} (trivial partition: one cluster only),

P(2)={1, 3},{2} (nontrivial partition, alternative partitions are {1}, {2, 3}, …)

P(1)={1}, {2}, {3} (trivial partition).

Object	$\mathbf{P}^{(1)}$	$\mathbf{P}^{(2)}$	$\mathbf{P}^{(3)}$
A	1	1	1
B	2	2	1
C	3	1	1

The algorithm ClusterOrder finds the number of clusters q = 1 for the above given set of partitions. As usual, the given order of these three objects is assumed to be 1 (= A), 2 (= B), and 3 (= C).

4 Techniques of rearrangements of objects

The most sophisticated task remains: to find an appropriate order of objects. Figure 12 shows the case of three objects only. The underlying pairwise distances between the three objects are:

$$\mathbf{D} = \begin{pmatrix} d_{AA} & d_{AB} & d_{AC} \\ d_{BA} & d_{BB} & d_{BC} \\ d_{CA} & d_{CB} & d_{CC} \end{pmatrix} = \begin{pmatrix} 0 & \sqrt{10} & 2 \\ \sqrt{10} & 0 & \sqrt{10} \\ 2 & \sqrt{10} & 0 \end{pmatrix}$$

It should be mentioned that there are an infinite number of set of points that all fit the given distance matrix above.

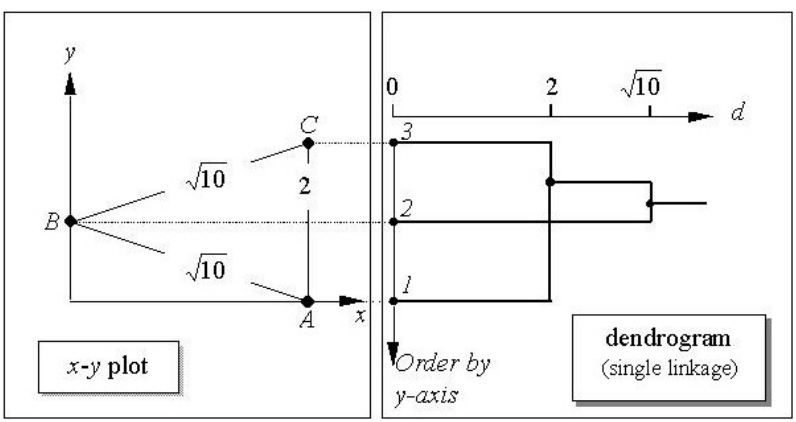


Figure 12**:** The starting point: three artificial two-dimensional objects (*x*-*y* plot at the left hand side), and the dendrogram (method: *Single Linkage*) regarding the given order of objects (A→1, B→2, and C→3) at the right hand side. In order to avoid unclear graphical representations the node of the fusion of {1} and {3} is not drawn in the middle of the two nodes as usual.

The ordering regarding the *y*-axis results in crossing lines in the dendrogram. Here the algorithm of the previous section stops at the first split and returns $q = 1$. The corresponding partition {1, 3}, {2} does not agree with the given order.

By means of the correspondence analysis a rearrangement of objects can be obtained as shown in Figure 13. The correspondence analysis is a special non-linear version of the principal components analysis (for details, see for instance [4, 23, 24]). Here the algorithm *ClusterOrder* of the previous section stops at $i = 1$ and returns $q = 3$. That means that the dendrogram reflects the given order totally without crossing of lines.

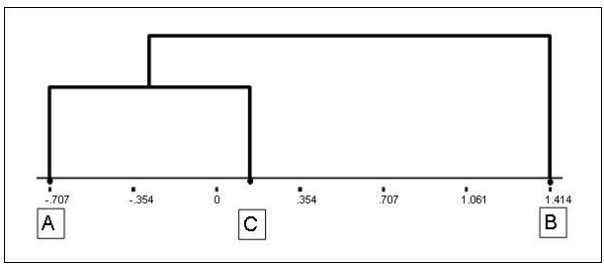


Figure 13 **:** The "rearranged" dendrogram of Figure 12 based on the scores of the correspondence analysis.

There are other projection techniques for finding an appropriate order like discriminant analysis or multidimensional scaling (see [34]). The choice of the technique should depend on both the data under investigation and the cluster analysis model that is used.

5 Application in Archaeometry

Theory and successful applications of mathematical methods in chemistry are reported by [26]. Especially several successful applications of cluster analysis methods are given by [32, 33, 37]. Here the statistical cluster analysis of bricks and tiles is presented that is based on measurements of chemical elements. 613 Roman bricks and tiles (=observations, objects) from the northern part of the former Roman Empires's province *Germania Superior* are described by 19 chemical elements. Some of the main references on this topic are [27-30]. The main aims of the statistical cluster analysis are

- to verify supposed locations of brickyards by their chemical attributes only, and
- to find brickyards which are unknown yet.

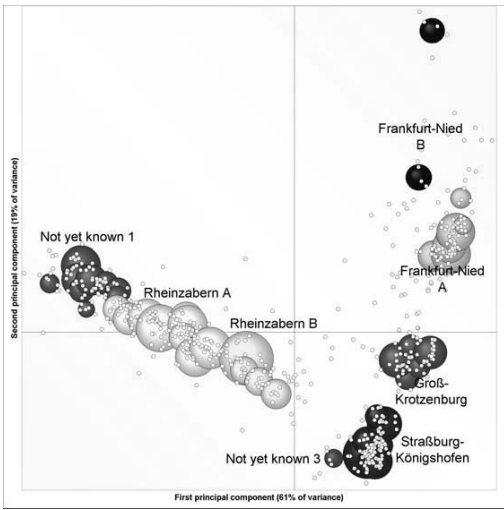


Figure 14: Principal components plot of the final eight clusters (same colour of the large bubbles) found by core-based clustering (for details see [31]). The 613 original observations are projected additionally.

Figure 14 shows the cluster analysis result in a principal components plot. Here both all 613 objects and 44 cores ("mini-clusters") are shown. The latter suggest visually an order of the bubbles mainly at the left hand side (for further details see [31]). The size of a bubble is proportional to the logarithmic sum-of-squares (5) of the corresponding cluster. Cores are very small and homogeneous clusters. The final eight clusters (large bubbles of the same colour) are denoted by the supposed locations of brickyards.

Figure 15 shows the plot-dendrogram of the eight clusters from Figure 14. Here the coordinates of each cluster are the corresponding centroids in the two principal components. Using only the first principal component a "rearranged" dendrogram can reflect the given order of the centroids without crossing branches. For this purpose the algorithm *ClusterOrder* of Section 3 is applied to the eight centroids only and not to all 613 objects.

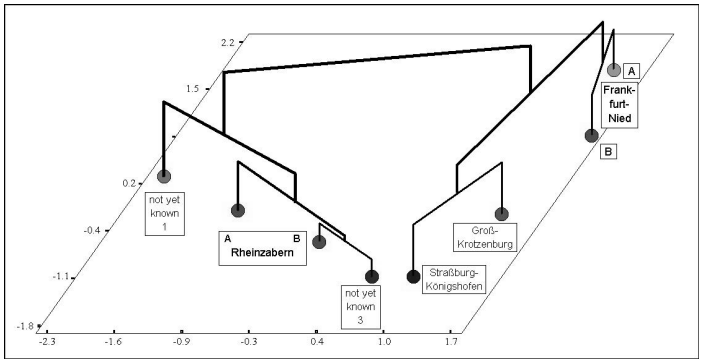


Figure 15 Plot-dendrogram of the eight clusters from Figure 14. There is no crossing of branches of the tree.

A quantitative graphical display of the 613 observations of Figure 14 is given in Figure 16. Here the non-parametric density estimation is applied, and afterwards the resulting density surface was cut at several levels.

Now the aims are to find an appropriate order of the 44 cores in Figure 14 first and then to draw an ordered dendrogram without crossing of branches. To reach these aims one can search for a rearrangement of the cores. For this purpose, for instance, two local principal component analyses (PCA) have to be performed: one PCA for the observations of the three clusters at the left hand side of Figure 14 and another PCA for the remaining observations. Then the resulting scores of the first components of each PCA are arranged into one score axis in a consecutive manner. By doing so the first aim is reached and a dendrogram can be drawn that reflects the order of the score axis (Figure 17). This dendrogram is quite informative. However, there occur some crossings of branches of the tree. Therefore, to reach the second aim the appropriate number of clusters has to find that allows drawing a dendrogram without crossing of lines.

Based on the rearrangement of the 44 cores in Figure 17 the set of partitions is investigated using the stepwise procedure *ClusterOrder* starting from the root of the tree. The condition of the algorithm is fulfilled for numbers of cluster 2,3, … until 10. The algorithm stops at the partition into 11 clusters and returns $q = 10$ clusters. As a consequence, for more than ten clusters the dendrogram contains crossing lines. Figure 18 shows the corresponding result: an ordered dendrogram.

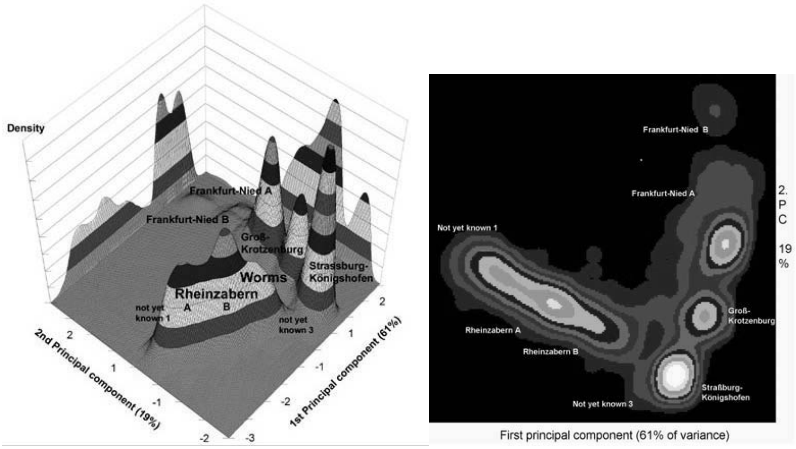


Figure 16: Univariate and bivariate non-parametric density estimations of the 613 observations based on the first two principal components (at the left hand side) and several cuts of the bivariate density at different levels at the right hand side.

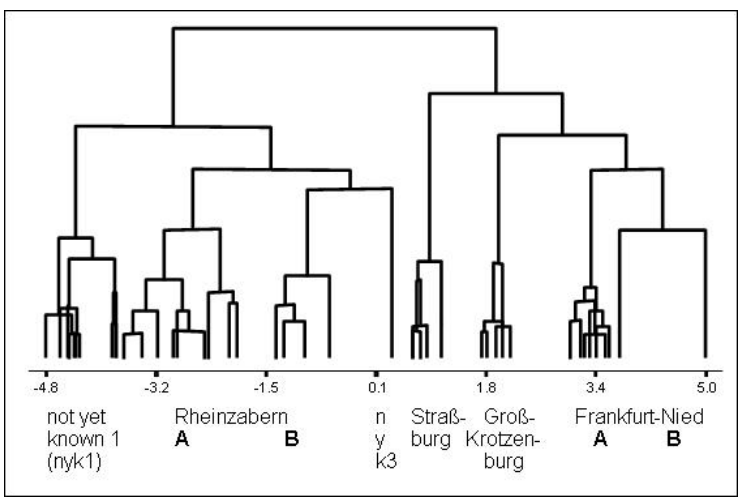


Figure 17: Non-equidistantly rearranged dendrogram of *logarithmic Ward*'s hierarchical clustering (6) of the 44 cores (bubbles) in Figure 14. In this local rearrangement by two PCA some intersections of lines are observed.

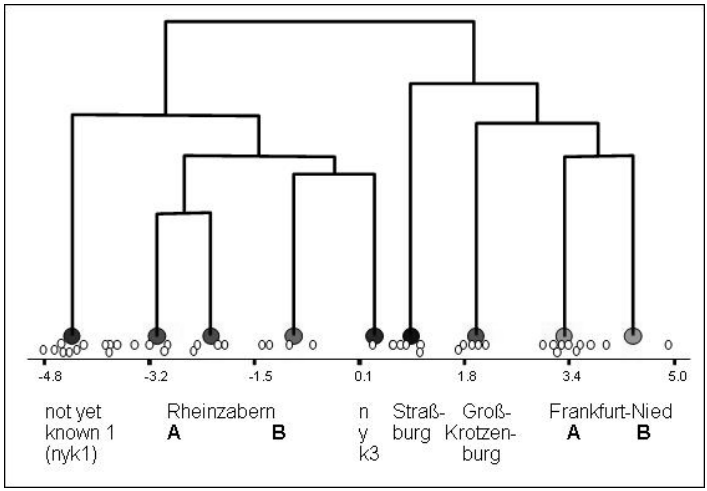


Figure 18: Part of the dendrogram of Figure 17 that is determined by the algorithm *ClusterOrder* in order to avoid intersections of lines.

6 Summary

Often, easy-to-understand graphical output of statistical analysis would be appreciated in the reality of a high-dimensional setting. Especially the reading of conventional dendrograms as well as their interpretation becomes difficult and it is often confusing. Some dendrogram drawing and reordering techniques are recommended that reflect a given order until a maximum degree. The resulting ordered dendrograms are much more informative. These dendrograms are drawn without crossing of lines. However, to find an appropriate order remains as the most difficult task. This task is the connecting point to the on-going research on both the *Order Theory* and the *Hasse diagram methodology*.

***

All the cluster analysis algorithms and multivariate visualisation techniques used here are part of the statistical software ClusCorr98 [37]. It uses the spreadsheet environment of Excel for displaying data, numerical results and graphics. The programming language is Visual Basic for Application (VBA). This prototype-software is under development at the Weierstrass Institute for Applied Analysis and Stochastics, Berlin.

References

[1] Hans-Georg Bartel, Hans-Joachim Mucha, and Jens Dolata: Über eine Modifikation eines graphentheoretisch basierten partitionierenden Verfahrens der Clusteranalyse. *MATCH Commun. Math. Comput. Chem.* **48** (2003), 209–223.

[2] Ludovic Lebart, Alain Morineau, and Jean-Pierre Fénelon: *Statistische Datenanalyse Methoden und Programme*. Berlin: Akademie Verlag 1984.

[3] Jean-Pierre Barthélemy and Alain Guénoche: *Trees and Proximity Representations.* Chichester/New York: Wiley 1991.

[4] Hans-Joachim Mucha: *Clusteranalyse mit Mikrocomputern.* Berlin: Akademie Verlag 1992.

[5] Jeffrey D. Banfield and Adrian E. Raftery: Model-based Gaussian and non-Gaussian clustering. *Biometrics* **49** (1993), 803–821.

[6] J.H. Ward: Hierarchical Grouping to Optimize an Objective Function. *Journal Amer. Stat. Ass.* **58** (1963), 236–244.

[7] Chris Fraley: Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.* **20** (1) (1998), 270–281.

[8] Eric Boudaillier and Georges Hebrail: Interactive Interpretation of Hierarchical Clustering. *Intelligent Data Analysis* **2**, Issue 1–4 (1998), 229–244.

[9] G.N. Lance and W.T. Williams: A General Theory of Classification Sorting Strategies. *Computer J.* **9** (1967), 373–380.

[10] K. Florek: Sur la Liaison et la Division des Points d'un Ensemble Fini. *Colloq. Math.* **2** (1951), 282–285.

[11] Anil K. Jain and Richard C. Dubes: *Algorithms for Clustering Data.* Englewood Cliffs: Prentice-Hall 1988.

[12] Allan D. Gordon: *Classification.* Boca Raton: Chapman & Hall/CRC, 2nd Edition 1999.

[13] Jean-Paul Benzecri: *L'Analyse des Données. Tom 1: La Taxinomie.* Paris: Dunod 1973, ^{2}1976.

[14] Jean-Paul Benzecri: Histoire et Préhistoire de l'Analyse des Données. *Les Cahiers de l'Analyse des Données* **1** (1976), n° 1–4.

[15] David Wishart: *CLUSTAN. Benutzerhandbuch*. Stuttgart: Gustav Fischer Verlag 1984.

[16] Klaus Backhaus, Bernd Erichson, and Wulff Plinke, Christiane Schuchard-ficher and Rolf Weiber: *Multivariate Analysemethoden.* Berlin: Springer-Verlag 1989.

[17] SAS Institute Inc.: *SAS/Stat User's Guide. Volume 1. Version 6*. Cary: SAS Institute Inc. Fourth Edition 1990.

[18] Horst Sachs: *Einführung in die Theorie der endlichen Graphen, Teil I.* (Mathematisch-Naturwissenschaftliche Bibliothek, Bd. 43). Leipzig: B.G. Teubner Verlagsgesellschaft 1970.

[19] Robin J. Wilson: *Introduction to Graph Theory.* Edinburgh: Oliver and Boyd 1972.

[20] John M. Chambers and Trevor J. Hastie (Eds.): *Statistical Models in S.* Pacific Grove: Wadsworth & Brooks 1992.

[21] Michel Jambu and Marie-Olile Lebeaux: *Cluster analysis and data analysis.* Amsterdam: North-Holland Publishing Company 1983.

[22] Rainer Bodendiek and R. Lang: *Lehrbuch der Graphentheorie, Bd. 1 und 2.* Heidelberg: Spektrum Akademischer Verlag 1995.

[23] Michael J. Greenacre: *Theory and Application of Correspondence Analysis.* London: Academic Press 1984.

[24] Michael J. Greenacre: *Correspondence Analysis in Practice.* London: Academic Press 1993.

[25] Hans Hermann Bock: *Automatische Klassifikation. Theoretische und praktische Methoden zur Gruppierung und Strukturierung von Daten (Cluster-Analyse)* Göttingen: Vandenhoeck & Ruprecht 1974.

[26] Hans-Georg Bartel: *Mathematische Methoden in der Chemie.* Heidelberg/Berlin/Oxford: Spektrum Akademischer Verlag 1996.

[27] Jens Dolata: *Römische Ziegelstempel aus Mainz und dem nördlichen Obergermanien - Archäologische und archäometrische Untersuchungen zu chronologischem und baugeschichtlichem Quellenmaterial*. (Inauguraldissertation, Johann Wolfgang Goethe-Universität. Frankfurt/M. 2000).

[28] Hans-Georg Bartel, Hans-Joachim Mucha, and Jens Dolata: Automatische Klassifikation in der Archäometrie: Berliner und Mainzer Arbeiten zu oberrheinischen Ziegeleien in römischer Zeit. *Berliner Beiträge zur Archäometrie* **19** (2002), 31–62.

[29] Hans-Joachim Mucha, Hans-Georg Bartel, and Jens Dolata: Exploring Roman Brick and Tile by Cluster Analysis with Validation of Results. In: Wolfgang Gaul, Gunter Ritter (Eds.): *Classification, Automation, and New Media.* (Proceedings of the 24th Annual Conference of the Gesellschaft für Klassifikation, University of Passau, March 2000). Berlin: Springer-Verlag 2002, 471–478.

[30] Jens Dolata, Hans-Joachim Mucha, and Hans-Georg Bartel: Archäologische und mathematisch-statistische Neuordnung der Orte römischer Baukeramikherstellung im nördlichen Obergermanien, *Xantener Berichte* **13** (2003), 381–409.

[31] Hans-Joachim Mucha, Hans-Georg Bartel, and Jens Dolata: Core-based clustering techniques. In: Martin Schader, Wolfgang Gaul, and Maurizio Vichi (Eds.): *Between Data Science and Applied Data Analysis*. Berlin: Springer Verlag, (2003), 74–82.

[32] Helmuth Späth: *Cluster Analysis Algorithms*. Chichester: Ellis Horwood Limited Publisher 1980.

[33] Helmuth Späth: *Cluster Dissection and Analysis. Theory, FORTRAN programs, Examples.* Chichester: Ellis Horwood Limited Publisher 1985.

[34] Brian D. Ripley: *Pattern Recognition and Neural Networks.* Cambridge: Cambridge University Press 1996.

[35] Hans-Joachim Mucha: *Automatic Validation of Hierarchical Clustering*. In: Jaromir Antoch (Ed.): *Proceedings in Computational Statistics*, *COMPSTAT 2004*, *16th Symposium*. Heidelberg: Physica-Verlag, (2004), 1535–1542.

[36] Hans-Joachim Mucha and Hans-Georg Bartel: *ClusCorr98 - Adaptive Clustering, Multivariate Visualization, and Validation of Results*. In: Daniel Baier and Klaus-

Dieter Wernecke (Eds.): *Innovations in Classification, Data Science, and Information Systems*. Berlin: Springer Verlag, (2004), 46–53.

[37] Ute Simon, Hans-Joachim Mucha, and Rainer Brüggemann: *Model-Based Cluster Analysis Applied to Flow Cytometry Data*. In: Daniel Baier and Klaus-Dieter Wernecke (Eds.): *Innovations in Classification, Data Science, and Information Systems*. Berlin: Springer Verlag, (2004), 69–76.