Solvents Classification using a Multivariate Approach: Cluster Analysis.

1. INTRODUCTION

This work is intended to complete the study reported in the previous post. That study showed the intercorrelation existing among the physicochemical descriptors used to characterize the solvents and the possibility of capturing 70% of the initial data variability just using two principal components (PCs). The first PC was related to "polarity/polarizability" and "lipophilicity" of molecules while the second was related to "strength of intermolecular forces". 2d-contour plots designed in the space of the first three principal components, or dimensions, suggested the possibility of grouping solvents into aggregates (or *clusters*) of similar individuals and this aspect is covered by this study. *Data Clustering*, or *Cluster Analysis*, is an *unsupervised method* of creating groups of objects, or *clusters*, in such a way that objects in one cluster are very similar and objects in different clusters are quite distinct.

Cluster Analysis has a wide variety of applications such as data exploration, data reduction, hypothesis generation and prediction based on groups. Therefore, it represents a powerful "tool of discovery" [1].

Data Clustering must not be confused with *Data Classification* in which objects are assigned to predefined classes. In *Data Clustering*, the classes are also to be defined [2].

Since, as widely reported in the technical literature [3, 4], there is no one clustering method that can be judged "best" in all circumstances and since there is no methodical guidance for clustering tool-selection for a given clustering task [4], this study will cover several clustering procedures and it will compare their results.

R/RStudio are used for both cluster analysis and data visualization.

2. EXPERIMENTAL SECTION

This study is based on those solvents listed in the previous post and fully characterized by the eight physicochemical descriptors (or *active variables*) considered: *i.e.*, *melting point* (mp), *boiling point* (bp), *dielectric constant* (dc), *dipole moment* (dm), *refractive index* (ri), E_T (ET), *density* (d) and *log P* (logP).

Since here, like in the previous post, a number identifies each solvent in the diagrams, Table 1 (which corresponds to Table 2 of previous post) allows to quickly trace from that number to the corresponding chemical entity:

Table 1

Solvents actually considered					
No.	**solvent**	**No.**	**solvent**	**No.**	**solvent**
1	Water	23	N,N-Dimethylformamide	45	Iodobenzene
2	Formamide	24	N,N-Dimethylacetamide	46	Chlorobenzene
3	Methanol	25	Propionitrile	47	Bromobenzene
4	2-Methoxyethanol	26	Acetone	48	Tetrahydrofuran
5	N-Methylacetamide	27	Nitrobenzene	49	Anisole
6	Ethanol	28	Benzonitrile	50	Ethyl-phenyl-ether
7	2-Aminoethanol	29	1,2-Dichloroethane	51	1,1,1-Trichloroethane
8	Acetic acid	30	2-Methyl-2-butanol	52	1,4-Dioxane
9	Benzyl alcohol	31	2-Butanone	53	Trichloroethylene
10	1-Propanol	32	Acetophenone	54	Piperidine
11	1-Butanol	33	Dichloromethane	55	Diphenyl ether
12	2-Methyl-1-propanol	34	Hexamethylphosphoric triamide	56	Diethyl ether
13	2-Propanol	35	Cyclohexanone	57	Benzene
14	2-Butanol	36	Pyridine	58	Diisopropyl ether
15	3-Methyl-1-butanol	37	Methyl acetate	59	Toluene
16	Cyclohexanol	38	1,1-Dichloroethane	60	Triethylamine
17	Nitromethane	39	Quinoline	61	1,3,5-Trimethylbenzene
18	Acetonitrile	40	3-Pentanone	62	Carbon tetrachloride
19	3-Pentanol	41	Chloroform	63	Tetrachloroethylene
20	Dimethylsulfoxide	42	1,2-Dichlorobenzene	64	Cyclohexane
21	Aniline	43	Ethyl acetate		
22	2-Methyl-2-propanol	44	Fluorobenzene		

Data analysis and visualization have been performed using RStudio version 1.1.414 and R version 3.5.0 (The R Foundation for Statistical Computing). In particular, the following specific R packages have been used:

- *tidyverse* (H. Wickham, RStudio Inc., Boston, USA)[5]
- *FactoMineR* (F. Husson, J. Josse, S. Le, J. Mazet, Rennes University, France) [6, 7]
- *factoextra* (A. Kassambara, F. Mundt, HalioDx, Marseille, France and Pädagogische Hochschule, Karlsruhe, Germany) [8, 9, 10]

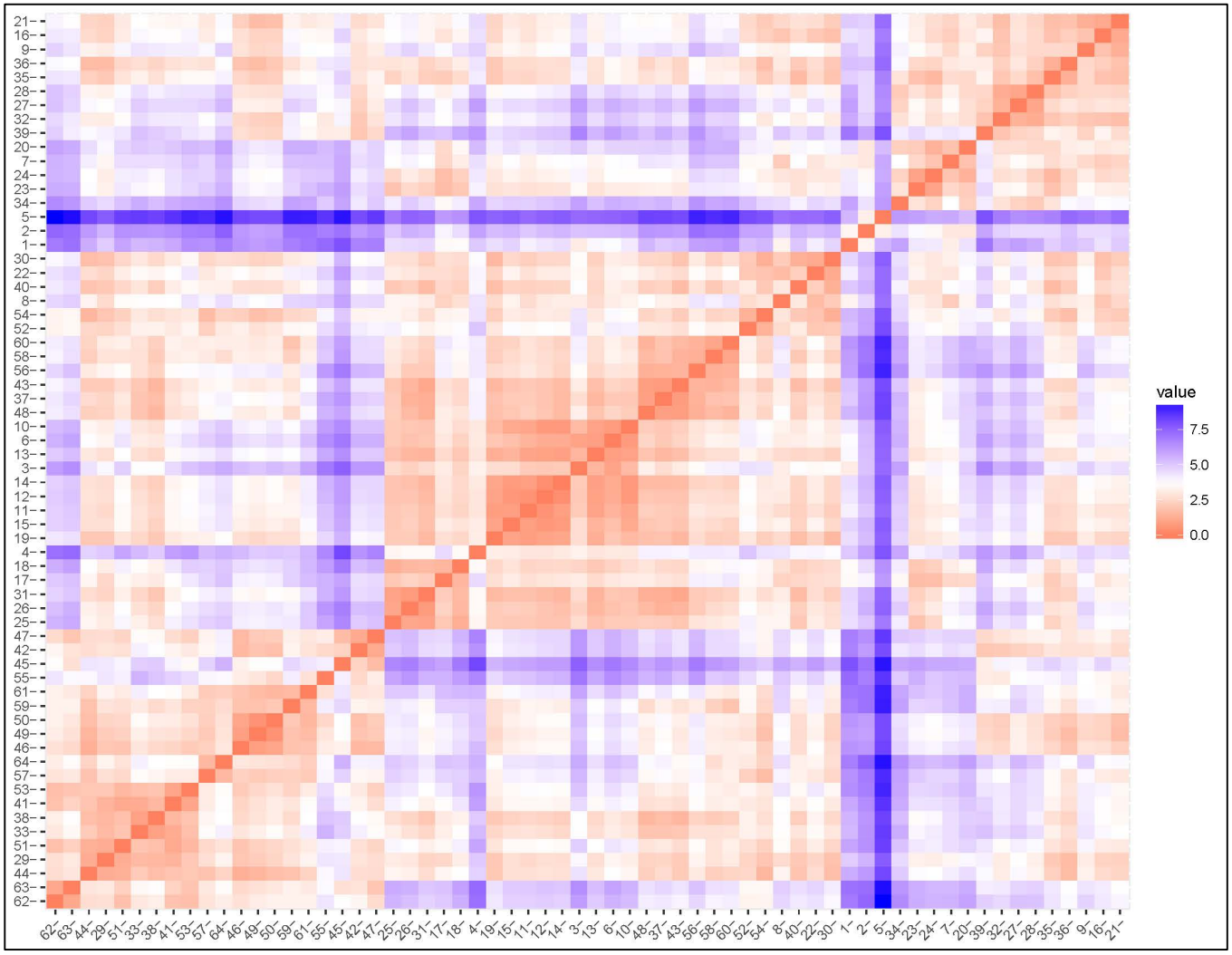
- *cluster* (M. Mächler, ETH Zürich, Switzerland) [11]
- *NbClust* (M. Charrad, N. Ghazzali, V. Boiteau and A. Niknafs, Université de Gabes, Université du Québec and Université Laval, Canada) [12]

3. RESULTS AND DISCUSSION

A very good overview of the cluster analysis procedure is that provided by Professors Dillon and Goldstein in their excellent book on Multivariate Analysis [13] in which they state that:
… The process typically begins by taking, say, p measurements on each of the n objects. The n x p matrix of raw data is then transformed into an n x n matrix of similarity or, alternatively, distance measures, where the similarities or distances are computed between pairs of objects across the p variables. Next, a clustering algorithm is selected, which defines the rules concerning how to cluster the objects into subgroups on the basis of the inter-object similarities. As we indicated, the goal in many cluster applications is to arrive at clusters of objects that display small within-cluster variation relative to the between-cluster variation…

Preliminary to the application of clustering procedures is to decide on a measure of inter-object similarity. Every clustering algorithm is in fact based on the index of similarity between data points[14]. If there is no measure of similarity or dissimilarity between pairs of data points, then no meaningful cluster analysis is possible [2].
With data having metric properties, like in this case, a *distance-type measure* is appropriate to estimate *indices of proximity*, or *closeness*, or *similarity* of a data set.
Since there are many methods to calculate the distance between each pair of observations and the chosen method influence the shape of the clusters [10], for the purposes of this study it has been selected the classical Euclidean approach. This distance measure has the property that the distances between two individuals can be interpreted as physical distances between two *n*-dimensional points (with $n = 8$ in this case) in the Euclidean space. As the Euclidean distance is not scale invariant, raw data need, first, to be standardized (or *scaled*) before computing the Euclidean distance. In this case data have been *autoscaled, i.e.,* transformed so that variables have mean equal to zero and standard deviation equal to one.
The function *fviz_dist()* of the R package *factoextra* allows beautiful visualizations of *distance matrices* (or *proximity matrices*) such as that shown in Figure 1.

Figure 1



This type of diagram, often used in the analysis of gene expression data, provides an immediate visual assessment of clustering tendency plotting the distances between pairs of individuals using different colors and intensities that account for the degree of similarity (or dissimilarity) existing between the observations. The "red" color indicates high similarity (or, alternatively, low dissimilarity) while the "blue" color indicates low similarity. A "pure red" color corresponds to $dist(x_i, x_j) = 0$ such as on matrix diagonal (the distance of each individual from itself is zero) while "pure blue" corresponds to $dist(x_i, x_j) = 1$ such as, for instance for pairs like: 5, 56 = N-Methylacetamide, Diethyl ether or 5, 64 = N-Methylacetamide, Cyclohexane.

Figure 1 displays, set along the diagonal, three main red colored blocks respectively delimited by the following pairs of individuals: 62 - 47 (1st block), 25 - 30 (2nd block) and 34 - 21 (3rd block).

However, a more careful inspection shows that these three main blocks are not uniformly red-colored, but they display a fragmented patchwork with areas intensely and uniformly red colored and other areas of a pale or very pale red color. The areas uniformly red colored correspond to aggregates of individuals highly similar among them.

In Table 2, here below, is provided a detailed analysis of those areas uniformly red-colored that can be visually identified within each main block of Figure 1:

Table 2

Block No.	Pairs of Individuals delimiting the block	Cluster No.	Pairs of Individuals delimiting the cluster	Number of individuals	Individuals' names
1	62 - 47	1	62 - 63	2	• 62: Carbon tetrachloride • 63:Tetrachloroethylene
		2	44 - 53	7	• 44: 1,2-Dichlorobenzene • 29: 1,2-Dichloroethane • 51: 1,1,1-Trichloethane • 33: Dichloromethane • 38: 1,1-Dichloroethane • 41: Chloroform • 53: Trichloroethylene
		3	57 - 64	2	• 57: Benzene • 64: Cyclohexane
		4	46 - 61	5	• 46: Chlorobenzene • 49: Anisole • 50: Ethyl-phenyl-ether • 59: Toluene • 61: 1,3,5-Trimethylbenzene
		5	55 - 47	4	• 55: Diphenyl ether • 45: Iodobenzene • 42: 1,2-Dichlorobenzene • 47: Bromobenzene

Table 2 (*cont.*)

Block No.	Pairs of Individuals delimiting the block	Cluster No.	Pairs of Individuals delimiting the cluster	Number of individuals	Individuals' names
2	25 - 30*	6	25 - 18	5	• 25: Proprionitrile • 26: Acetone • 31: 2-Butanone • 17: Nitromethane • 18: Acetonitrile
		7	19 - 10	9	• 19: 3-Pentanol • 15: 3-Methyl-1-butanol • 11: 1-Butanol • 12: 2-Methyl-1-propanol • 14: 2-Butanol • 3: Methanol • 13: 2-Propanol • 6: Ethanol • 10: 1-Propanol
		8	48 - 60	6	• 48: Tetrahydrofuran • 37: Methyl acetate • 43: Ethyl acetate • 56: Diethyl ether • 58: Diisopropyl ether • 60: Triethylamine
		9	52 - 54	2	• 52: 1,4-Dioxane • 54: Piperidine
		10	8 - 30	4	• 8: Acetic acid • 40: 3-Pentanone • 22: 2-Methyl-2-propanol • 30: 2-Methyl-2-butanol
3	34 - 21	11	34 - 20	5	• 34: Hexamethylphosphoric triamide • 23: N, N-Dimethylformamide • 24: N, N-Dimethylacetamide • 7: 2-Aminoethanol • 20: Dimethylsulfoxide
		12	39 - 28	4	• 39: Quinoline • 32: Acetophenone • 27: Nitrobenzene • 28: Benzonitrile
		13	35 - 36	2	• 35: Cyclohexanone • 36: Pyridine
		14	9 - 21	3	• 9: Benzyl alcohol • 16: Cyclohexanol • 21: Aniline

The fourteen areas that look practically uniformly red colored and that can be visualized in the *distance matrix* (Figure 1) correspond to as many groups, or *clusters*, of individuals highly similar among them. This similarity is substantiated by the fact that several clusters consist of members of a given chemical family (*e.g.*, alcohols, chlorinated hydrocarbons, etc.) or of chemical entities sharing common characteristics (*e.g.*, aprotic dipolar solvents, *etc.*).

Beside this, Figure 1 shows two blue stripes of substances unrelated or scarcely related with the others and in particular:

Table 3

Pairs of Individuals delimiting the blue stripe	Numbers of individuals	Individuals' names
1 - 5	1	Water
	2	Formamide
	5	N-Methylacetamide
4	4	2- Methoxyethanol

Figure 1 shows that each chemical entity listed in Table 3 practically relates just with itself as *per* the red square just on the main diagonal. N-Methylacetamide, in particular, is the chemical entity that shows the lowest correlation level with the rest of solvents listed in Table 1. It should be noted that data points 1, 2, 4 and 5 already occurred as points separated from the others in the 2d-contour plots shown in Figures 7-9 of the previous post.

From a simplified standpoint (*i.e.*, that does not take into consideration *fuzzy* or *soft* clustering, but that just considers *hard clustering*) the goal of *Clustering Analysis* is that of assigning data points with similar properties to the same group and dissimilar data points to different groups. The conventional approach to this problem can essentially be based on two categories of algorithms: *hierarchical* and *partitional*.

One of the primary features distinguishing *hierarchical techniques* from other clustering algorithms is that the allocation of an object to a cluster is irrevocable; that is, once an object joins a cluster it is never removed and fused with other objects belonging to some other clusters[13].

Hierarchical clustering techniques may be subdivided into *agglomerative methods*, which proceed by a series of successive fusions of the *n* individuals into groups, and *divisive methods*, which separate the *n* individuals successively into finer groupings [3].

The result of hierarchical classifications is a two-dimensional diagram known as *dendrogram* which is a special type of tree structure that visualizes a hierarchical clustering.

Between the two techniques above mentioned *agglomerative methods* probably represent the most widely used type of hierarchical procedure. According to different distance measures between groups, *agglomerative hierarchical methods* can be subdivided into *single-linkage*, *complete-linkage*, *average linkage* and *Ward's* methods.

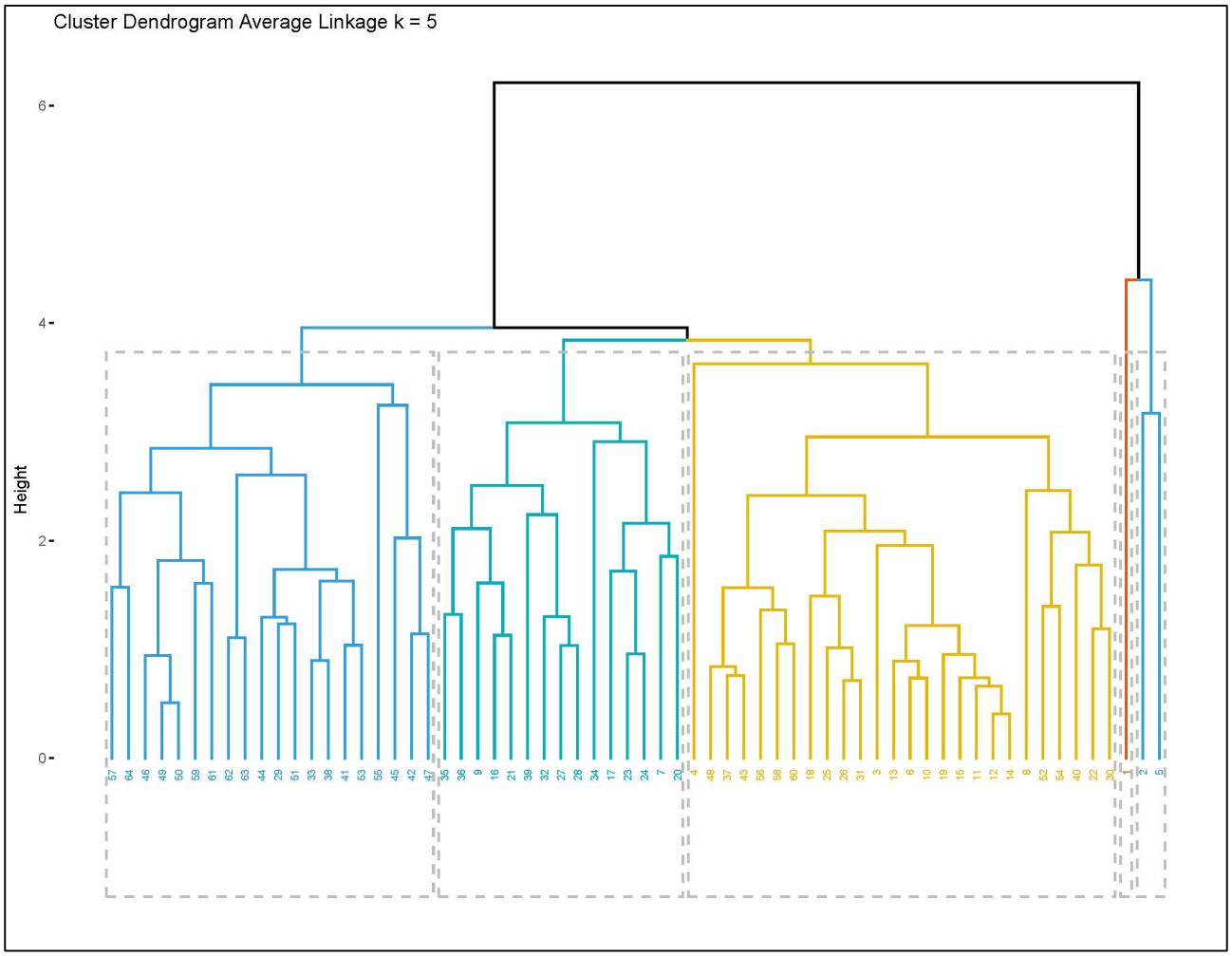
The application of the R base function *hclust()* to a distance, or dissimilarity, matrix containing the Euclidean distances between objects, leads to different dendrograms each corresponding to the different linkage method chosen. As these dendrograms are complex graphical representations, their visual comparison cannot provide any information regarding which linkage better reflects the initial data. Because of this it has been calculated the *cophenetic distances* and they have been correlated with the original distance data generated by the R function *dist()*.
In Table 4 are summarized the obtained results:

Table 4

Type of algorithm	**Correlation between cophenetic distance and Euclidean distance**
Single linkage	0.68
Complete linkage	0.48
Average linkage	0.72
Ward's method	0.53

As the value of the *cophenetic correlation coefficient* lies in the range [-1, +1] and a value close to 1 indicates a good fit of the hierarchy to the data, from Table 4 it follows that the *average linkage* is the hierarchical clustering approach that better reflects the data.
In Figure 2 is displayed the corresponding dendrogram obtained "chopping the tree" (*i.e.*, partitioning the hierarchical tree by drawing a horizontal line through the tree at an appropriate point [1]) into five groups.

Figure2



It is worth observing that the three *blue – turquoise - yellow* sections of the dendrogram shown in Figure 2 correspond, apart from two minor differences, to blocks *1 – 3 – 2* detailed in Table 2 and visualized in Figure 1.

The following data points represent the differences:

- 17 (Nitromethane): it belongs to block 2 (and not 3) of Figure 1 and Table 2
- 4 (2-Methoxyethanol): it belongs to those few solvents that are practically not correlated with the others and that are listed in Table 3.

Moreover, the partitions within each section of the dendrogram display the same structure detailed in Table 2. Therefore, the *average linkage* approach to hierarchical clustering reflects well the initial data.

A probably more intuitive visualization of these results can be obtained using the *fviz_cluster()* function of the R package *factoextra* and it is displayed in Figure 3.

Figure 3

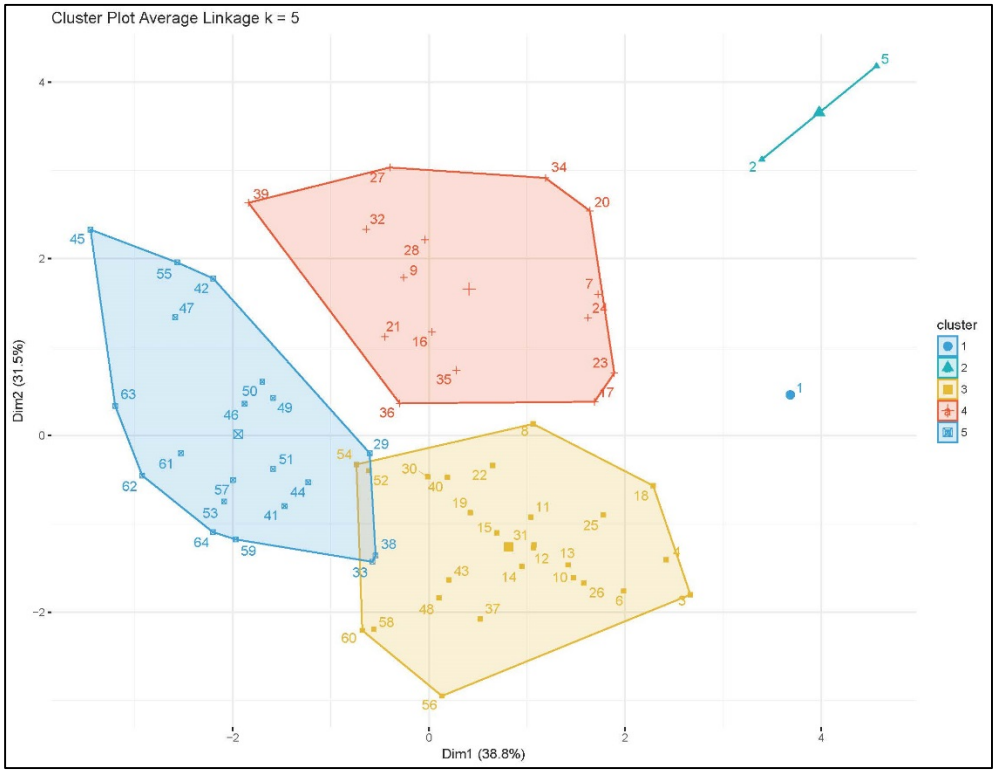


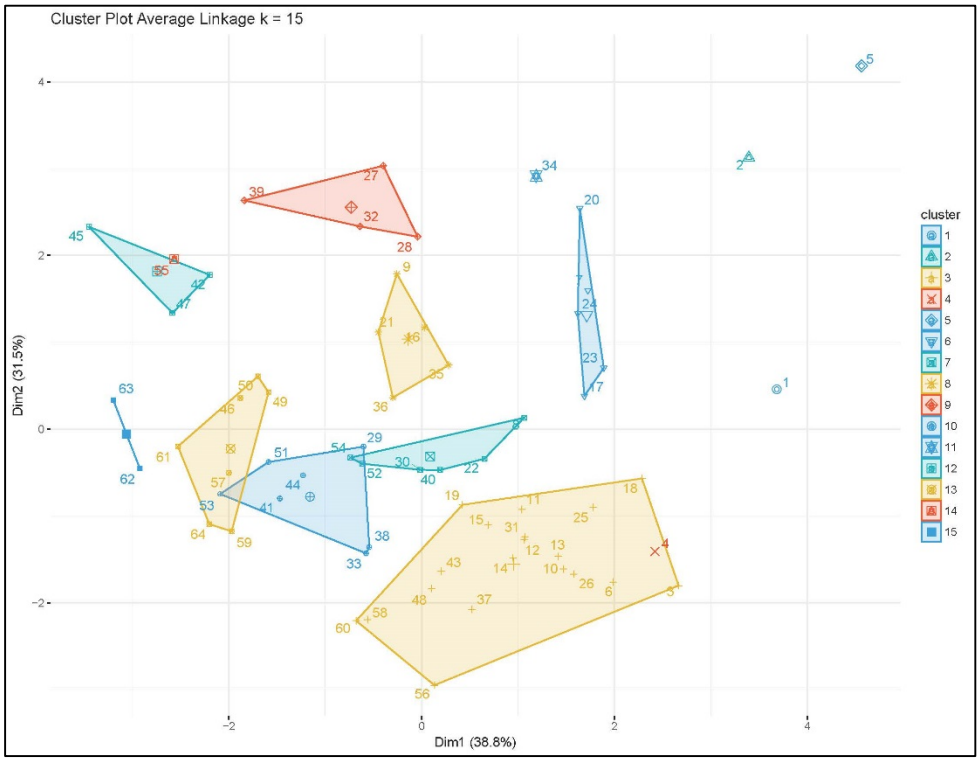
Figure 3 is a graphical representation obtained using the first two principal components (see Figure 4 and Table 2 of previous post) in which:

- Cluster 5 corresponds to the blue section of the dendrogram displayed in Figure 2 and to Block 1 of Table 2
- Cluster 3 corresponds to the yellow section of the dendrogram displayed in Figure 2 and to Block 2 of Table 2
- Cluster 4 corresponds to the turquoise section of the dendrogram displayed in Figure 2 and to Block 3 of Table 2

Cluster 2 and Cluster 1 are formed, respectively, just by two and one element, *i.e.*, data points 2, 5 and 1, that form the small blue and red blocks on the right of Figure 2

Cutting the dendrogram of Figure 2 lower down (*i.e.*, in fifteen groups) and displaying the cluster plot using the first two principal components, it can be obtained the partitioning of Figure 4.

Figure 4



Apart from a few clusters that combine two sets separately listed in Table 2 (*i.e.*, 3 + 4, 9 + 10, 13+14), the partition visualized in Figure 4 reflects that described in Table 2.

To summarize, the hierarchical classification analysis carried out until now was based on:

- pairs' distances calculated on the initial data autoscaled
- agglomeration based on the *average linkage* method for computing the distance between clusters.

The *average linkage* method was chosen as it was that characterized by the highest value of the *cophenetic correlation coefficient*.

For the visualization of both, dendrograms and cluster plots, were respectively used the *fviz_dend()* and *fviz_cluster()* functions of the R package *factoextra*.

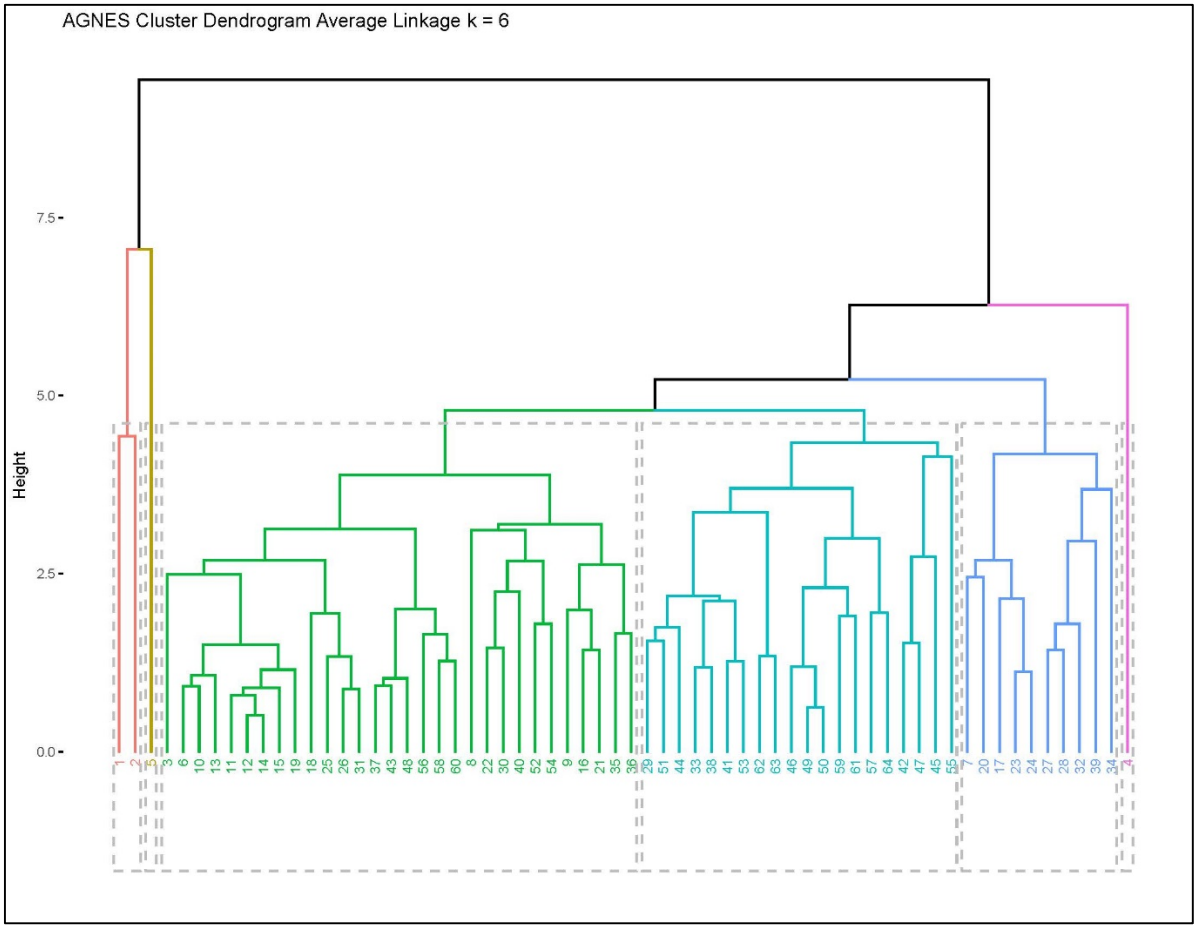
Beside *hclust()*, that is the built-in function of the R package *stats* for computing *hierarchical clustering*, other different functions are also available in R for computing hierarchical clustering. That most commonly used is probably *agnes()* that can be obtained from the R package *cluster*[11].

Using *agnes()* (***AG***glomerative ***NES***ting) on standardized data and:

- applying an Euclidean metric and an average linkage method
- cutting the tree in six groups (k = 6)

it can be obtained the dendrogram shown in Figure 5.

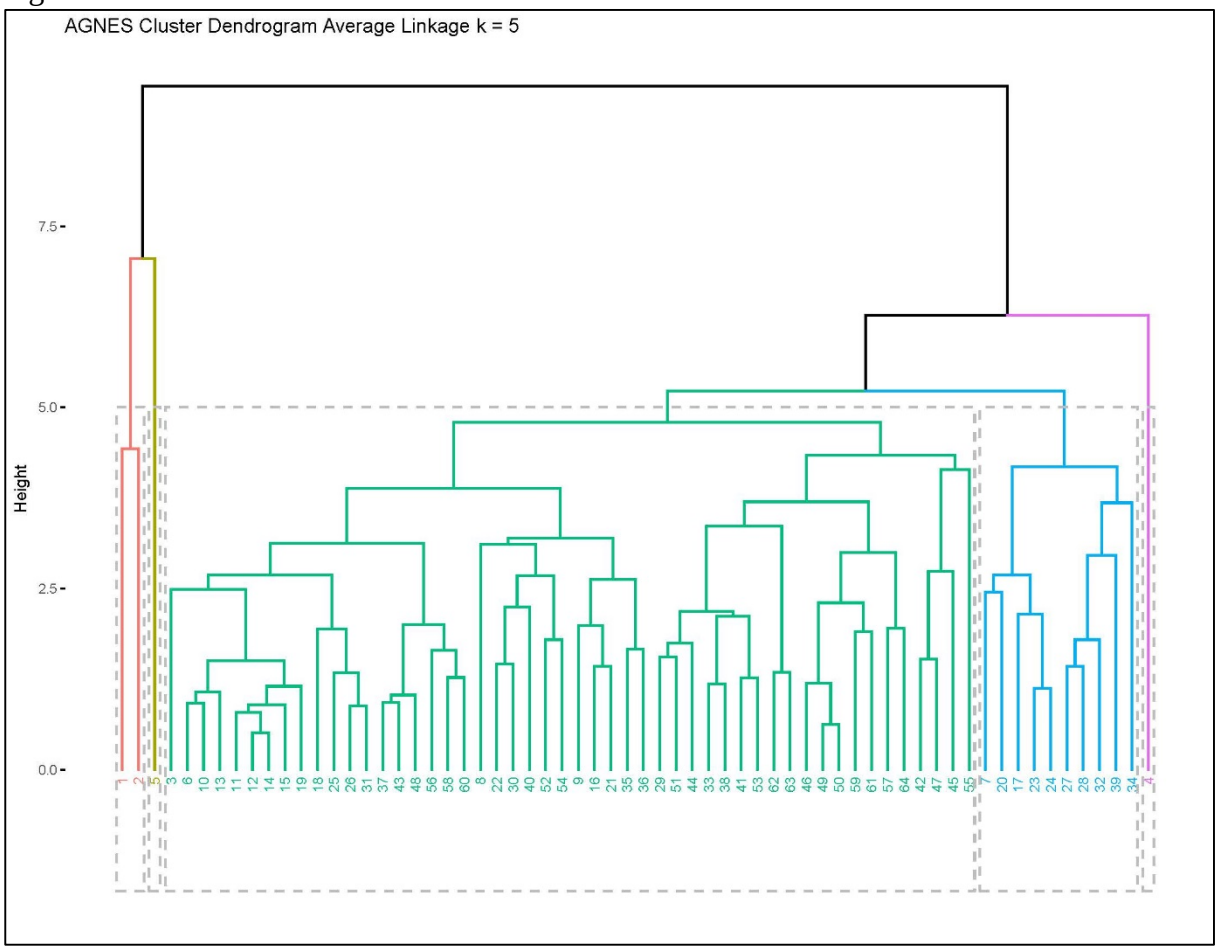
Figure 5



The two dendrograms in Figures 5 and 2 look rather similar at a glance, but a more careful examination shows several differences between them. For example, all *leaves* in the yellow section of Figure 2, with the exception of that corresponding to data point 4, can be found in the green section of Figure 5. However, this last one contains five leaves more that in the partitioning shown in Figure 2 are assigned to the turquoise section (*i.e.*, 9, 16, 21, 35, and 36).

Moreover, cutting the tree obtained using *agnes()* in just five groups (k = 5), as it was previously done for that obtained using *hclust()*, leads to a partition displaying just two main groups (see Figure 6) and not three as in both cases of Figures 2 and 5.

Figure 6

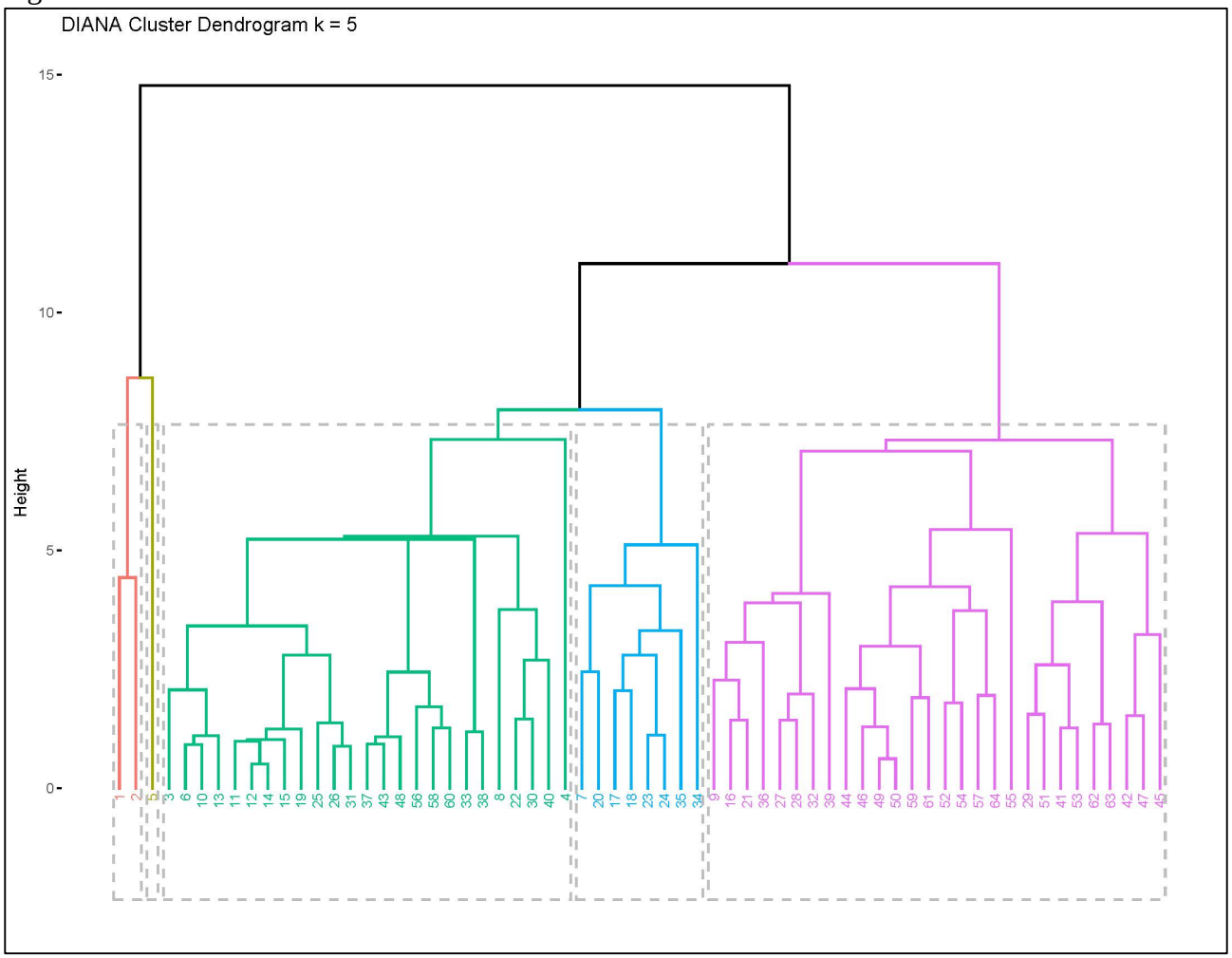


It is evident from what above that even using the same autoscaled data, and the same Euclidean metric and linkage method (average), the two agglomerative functions *hclust()* and *agnes()* lead to different dendrograms.

As already mentioned, beside *agglomerative methods*, hierarchical clustering techniques also include *divisive methods*, which operates by successive splitting of groups, starting with one group of *n* individuals and finishing with *n* groups of one individual. In this respect, the R package cluster provides the function *diana()* to perform divisive clustering.

Using *diana()* (***DI***visive ***ANA***lysis) on standardized data, applying an Euclidean metric and cutting the tree in five groups (*i.e.*, k = 5) it can be obtained the dendrogram shown in Figure 7. At a first glance this dendrogram looks similar to that obtained using *agnes()* and shown in Figure 5, however, a close comparison between the two reveals differences in both the absolute number and the type of *leaves* constituting the several groupings.

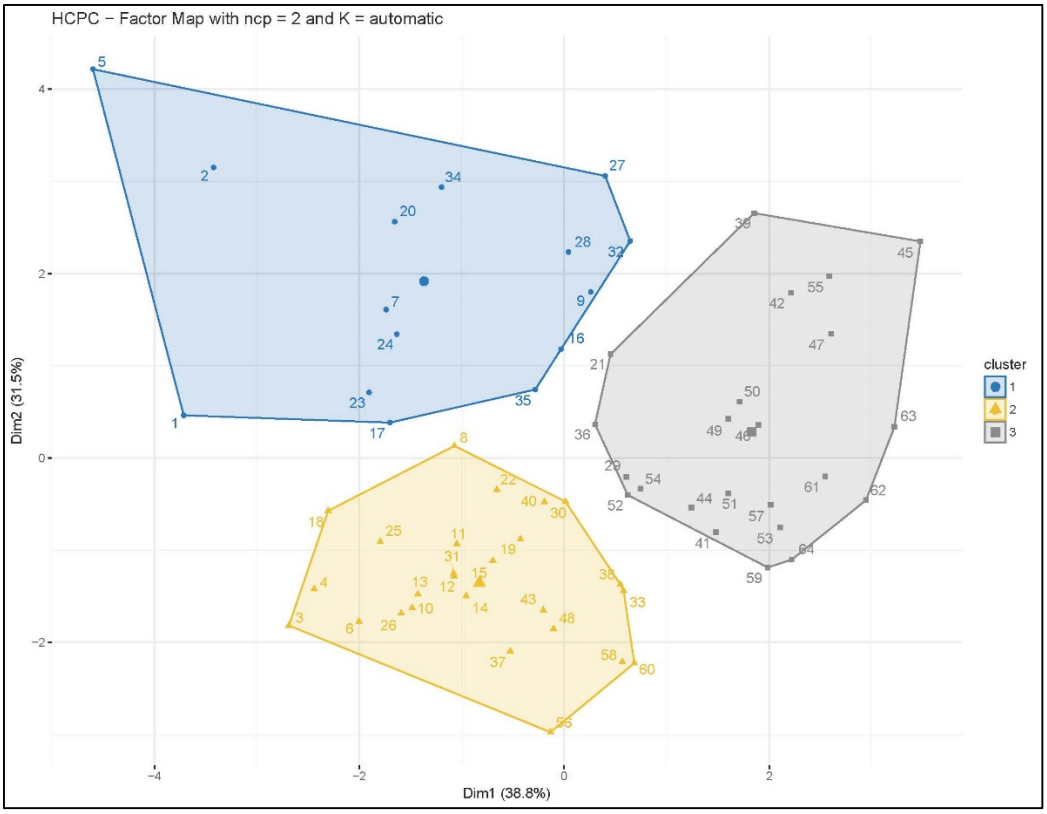
Figure 7



The *Cluster Analysis* conducted until now was always based on initial data suitably scaled, but it has led to different groupings depending on the specific function used (*i.e.*, *hclust()* or *agnes()* or *diana()*). As in Table 2 of previous post it was shown that principal components are capable of well capturing variance in the original data, it has been attempted a *Hierarchical Clustering based on Principal Components*, or *HCPC*.

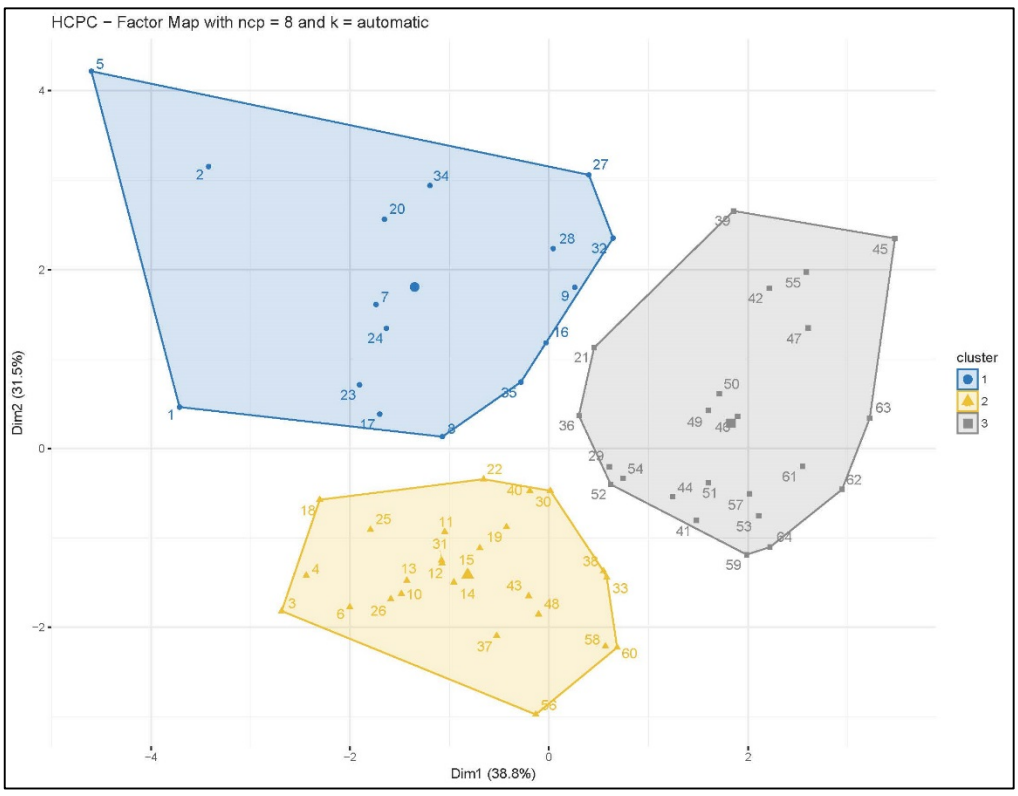
For the sake of investigation, HCPC has been carried out considering a progressively increasing number of principal components to see if and with how many principal components this type of Cluster Analysis would have led to results comparable to those obtained using the whole database. This type of analysis has been performed using two R packages: *FactoMineR* [6, 7] for computing HCPC and *factoextra* to visualize the obtained results [9]. As first attempt, cluster analysis was conducted using just the first two principal components (~ 70% of explained variance) and letting the function *HCPC()* of *FactoMineR* to suggest the best level to cut the tree. The obtained plot is shown in Figure 8 and it displays three clusters

Figure 8



In Figure 9 is the same type of analysis repeated using eight (8) principal components (*i.e.*, 100% explained variance) and leaving to the software cutting the tree at the suggested level.

Figure 9



Comparing Figures 8 and 9 is evident that even quadruplicating the number of principal components considered, but leaving to the software cutting the tree, this does not affect the final clusters structure apart from just one data point. In fact, data point 8 (Acetic Acid) is assigned to *Cluster 2* when using two principal components and to *Cluster 1* when using eight components.

This partition in three clusters is not at all unexpected, in fact, already the *distance* (or *proximity*) *matrix* shown in Figure 1 displayed three main red blocks set along the diagonal and therefore this classification in three clusters basically reproduces that finding. In fact, a careful comparison of the data points belonging to the three clusters with those forming the three red blocks of Figure 1 shows that:

- *Cluster 3* (the grey one) approximately corresponds to the first block down on the left in Figure 1
- *Cluster 2* (the yellow one) approximately corresponds to the main central red block in Figure 1 and
- *Cluster 1* (the blue one) approximately corresponds to the red block upwards (right) in Figure 1.

It is interesting to observe how "limit data points" such as:

- 1, 2 and 5 (*i.e.*, those belonging to the stripe most intensely blue colored in Figure 1) and
- 34, 20, 7, 24, 23 and 17 (*i.e.*, also belonging to blue colored areas in Figure 1)

are all assigned to cluster 1 within which they are arranged along "lines" moving top down.

At a glance, in fact, *Cluster 1* displays three series of data points respectively located on the left, on the middle and on the right of its area. A similar provision cannot be observed, for instance, in *Cluster 2* where data points look more uniformly distributed within the defined area. *Cluster 3*, on the contrary, shows groupings of data points inside.

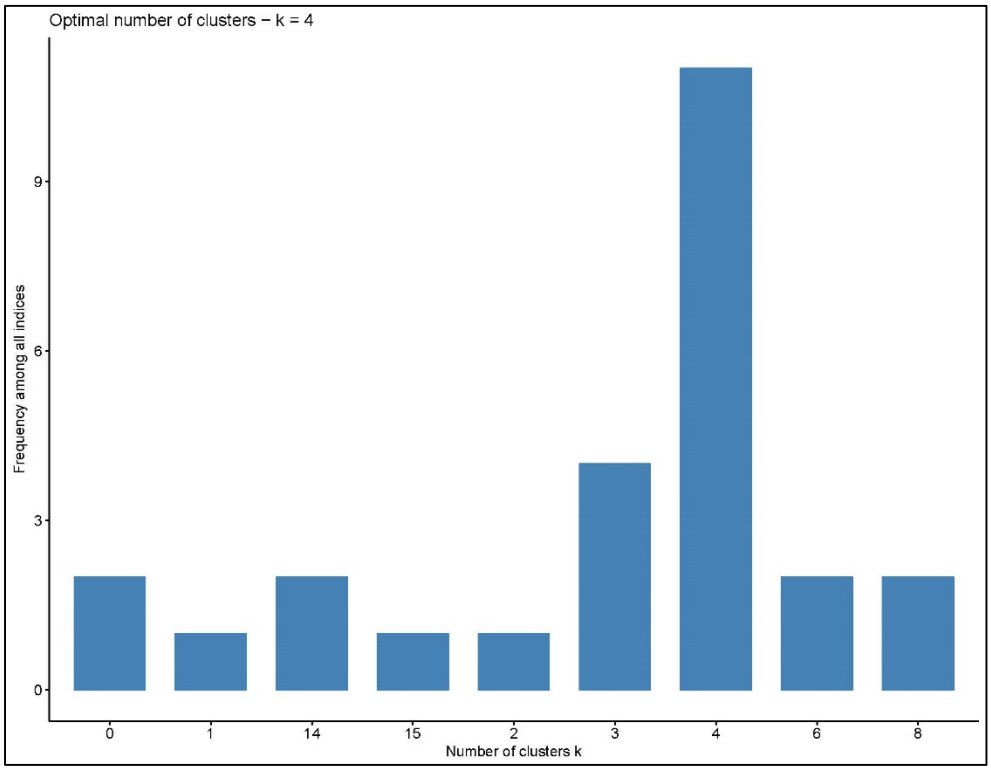
These visual evidences further confirm the existence of a finer structure within the three clusters just as it was observed in the *distance matrix* of Figure 1.

In cases like this, to better define the number of clusters there are two possibilities:

- perform a detailed visual analysis such as that done for Figure 1 or
- rely on predictive tests such as that expressed by the R package *NbClust* [12].

The result provided by the R package function *NbClust()* applied to initial data using an Euclidean metric and the Ward's method, as recommended for hierarchical clustering, is shown in Figure 10.

Figure 10



In light of this, the cluster analysis summarized in Figures 8 and 9 has been repeated setting a value of four for the function *HCPC()* of *FactoMineR* to cut the tree. The plots obtained using two and eighth principal components are respectively shown in Figures 11 and 12.

Figure 11

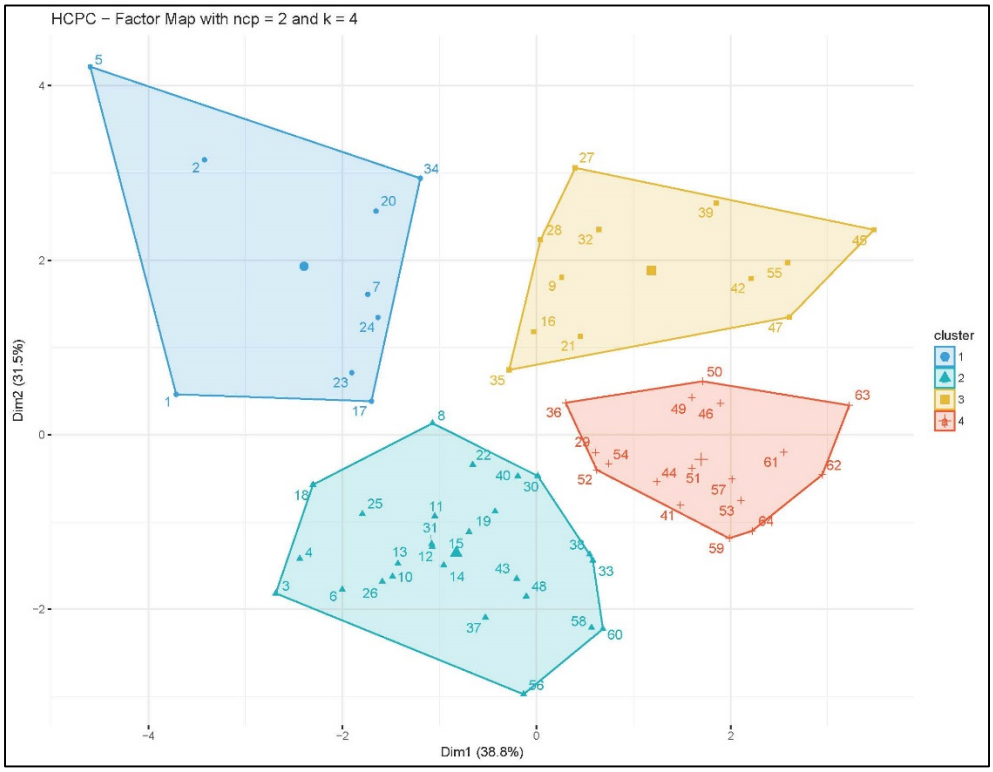
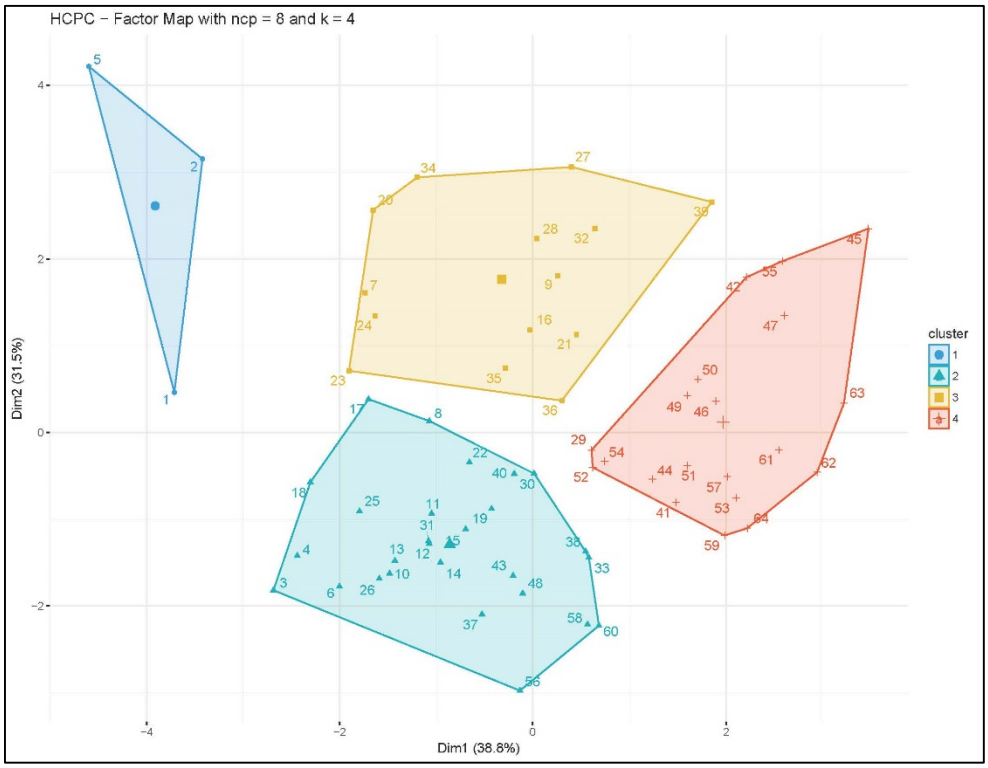


Figure 12



It is evident, in this case, that increasing the number of principal components, and therefore approaching the initial situation in which we used all data, the clusters partitioning changes. In this respect, Figures 11 and 12 should be compared with the corresponding Figures 8 and 9 (both obtained letting the algorithm to cut the tree) and with Figure 1.

As for *partitioning clustering* (the other type of clustering techniques that together with *hierarchical clustering* form the *hard clustering methods*), the number of clusters to be generated is required in advance, there are methods (*direct* and *statistical*) that have been developed for this purpose such as, for instance:

- *Statistical methods*: Gap-statistic
- *Direct methods*: Elbow, Silhouette

These methods have been applied to different *partitional* clustering algorithms (*e.g.*, K-means, PAM or ***P****artitioning* ***A****round* ***M****edoids* and CLARA or ***C****lustering* ***LAR****ge* ***A****pplications*) obtaining the plots shown in Figures 13 – 22. All these plots have been obtained using the *fviz_nbclust()* function of the R package *factoextra* [8,9,10].

Figure 13

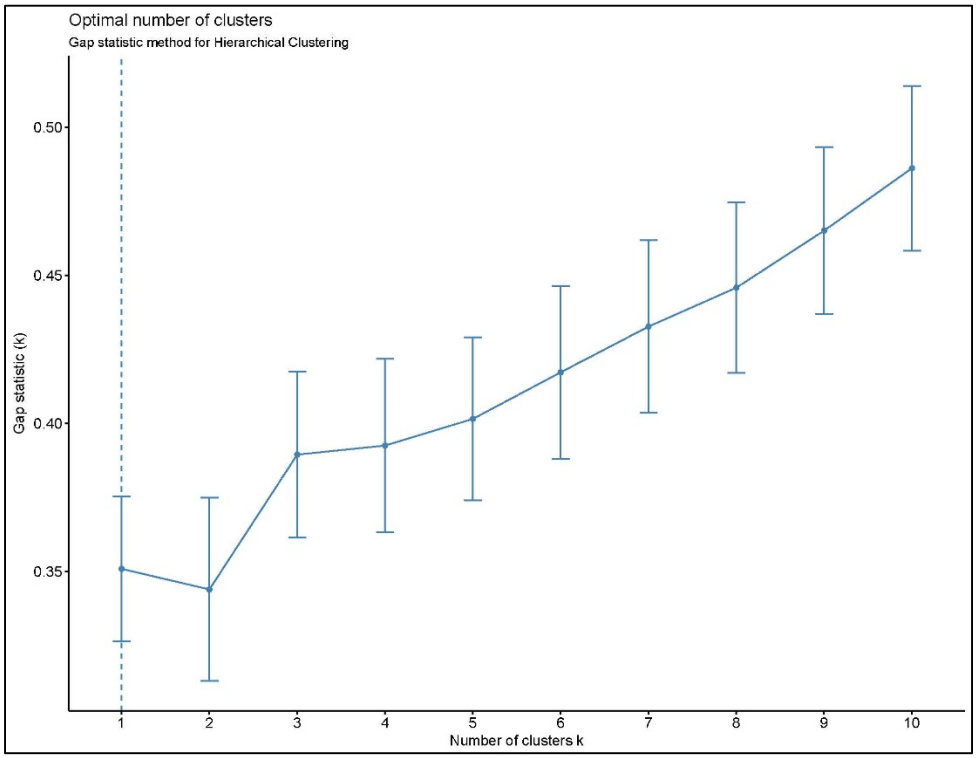


Figure 14

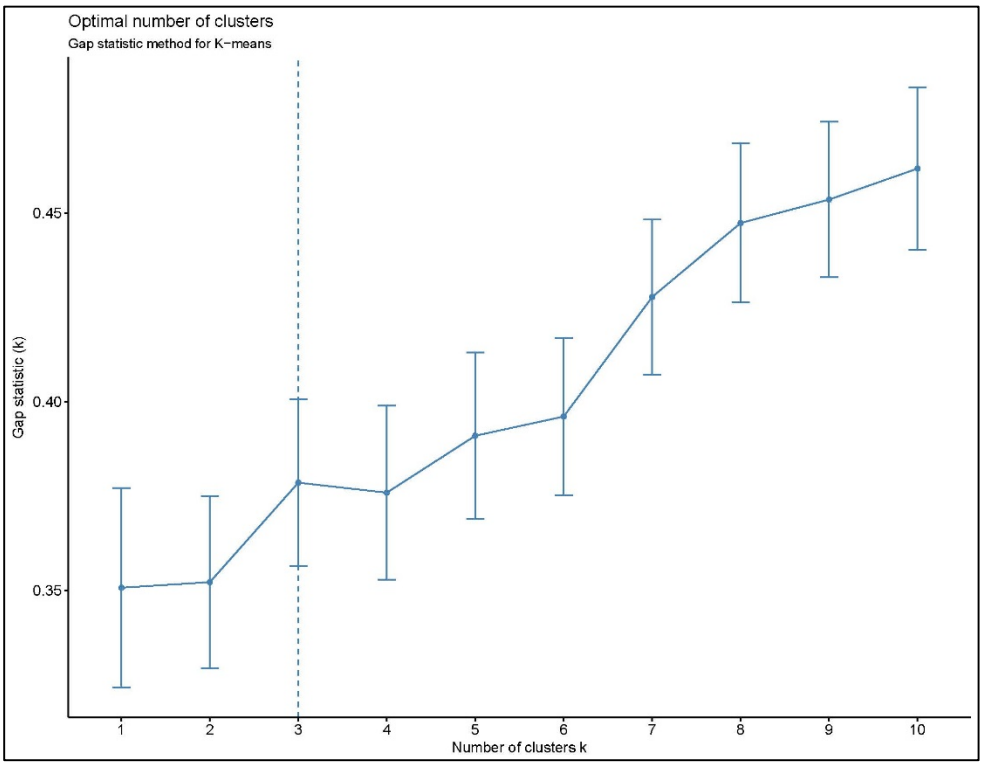


Figure 15

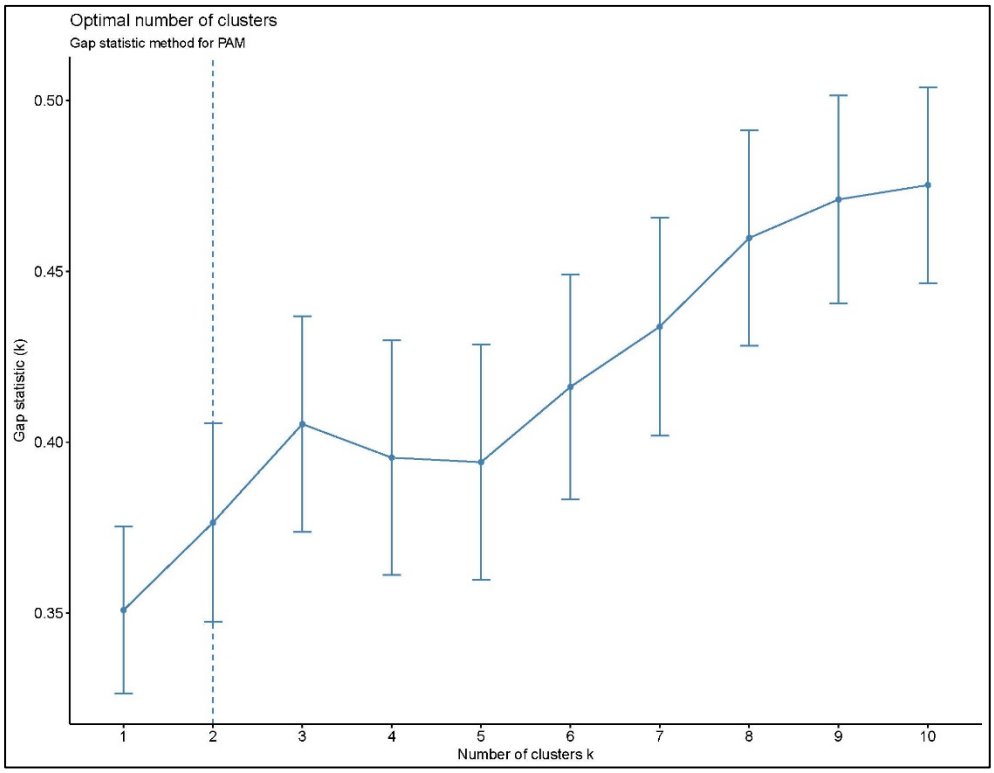


Figure 16

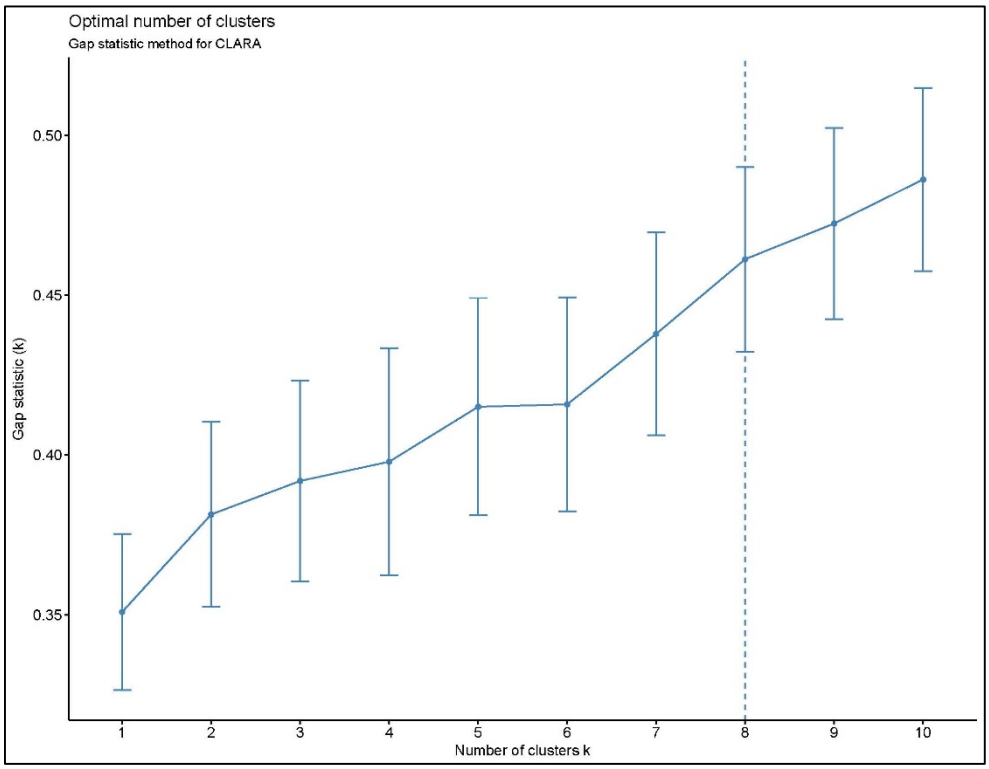


Figure 17

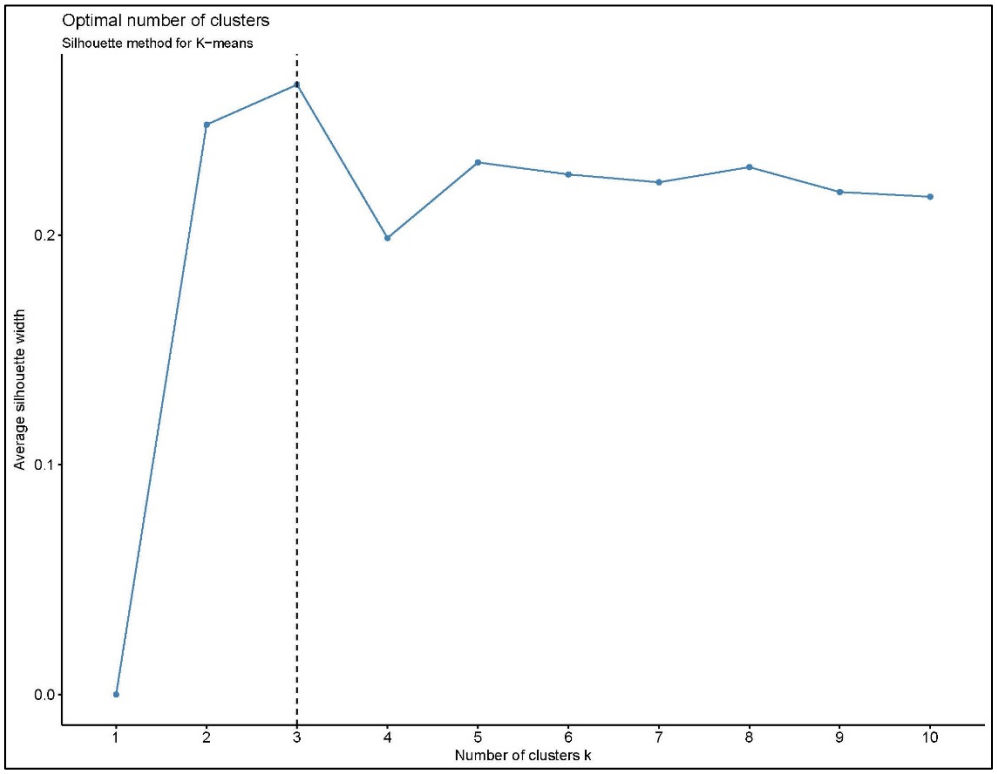


Figure 18

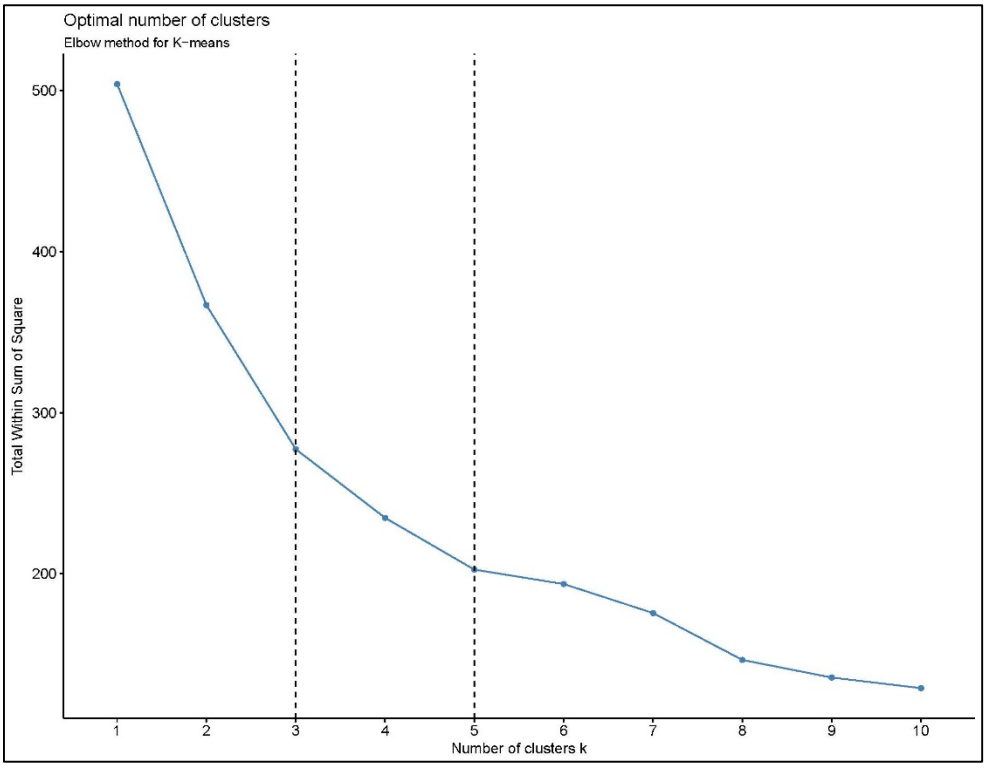


Figure 19

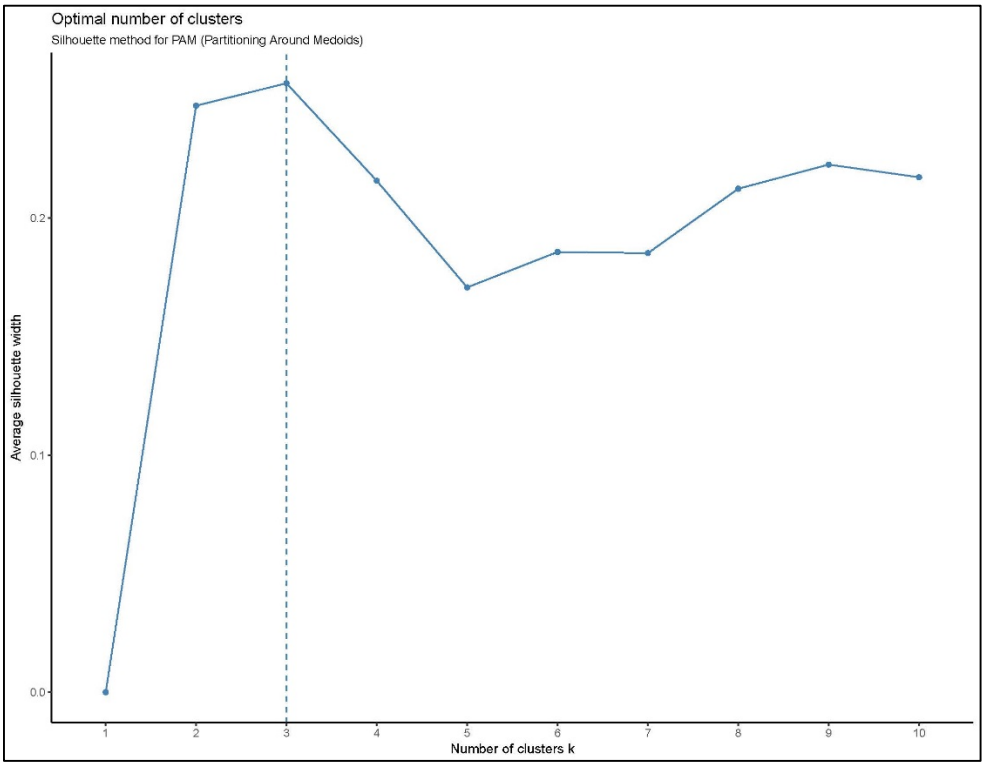


Figure 20

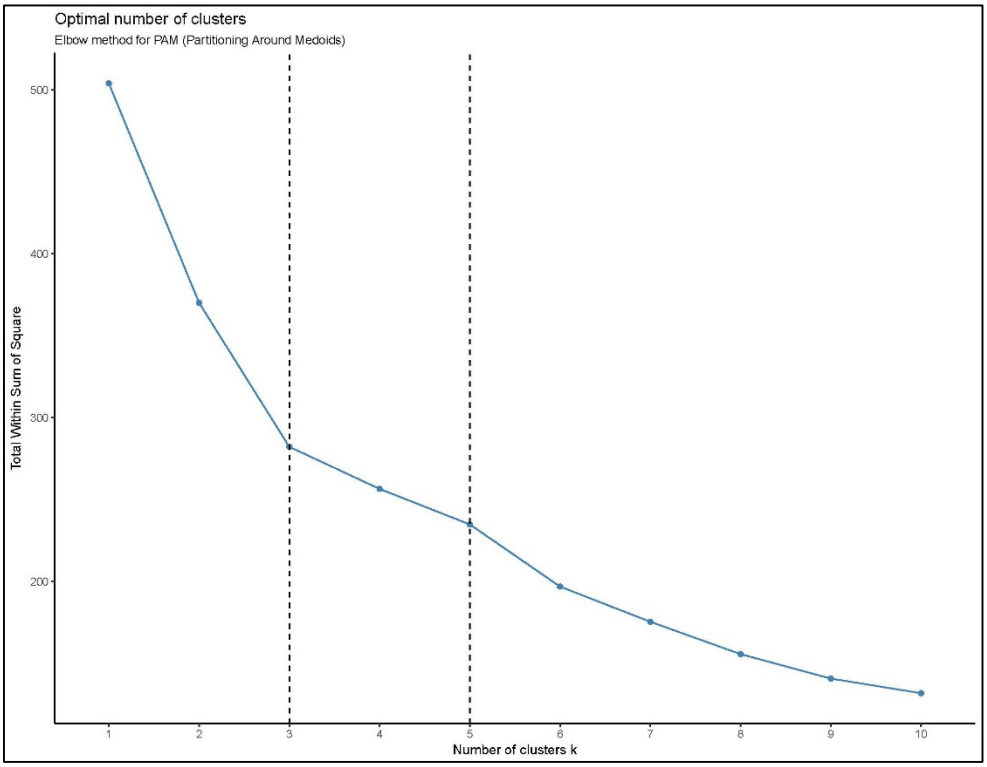


Figure 21

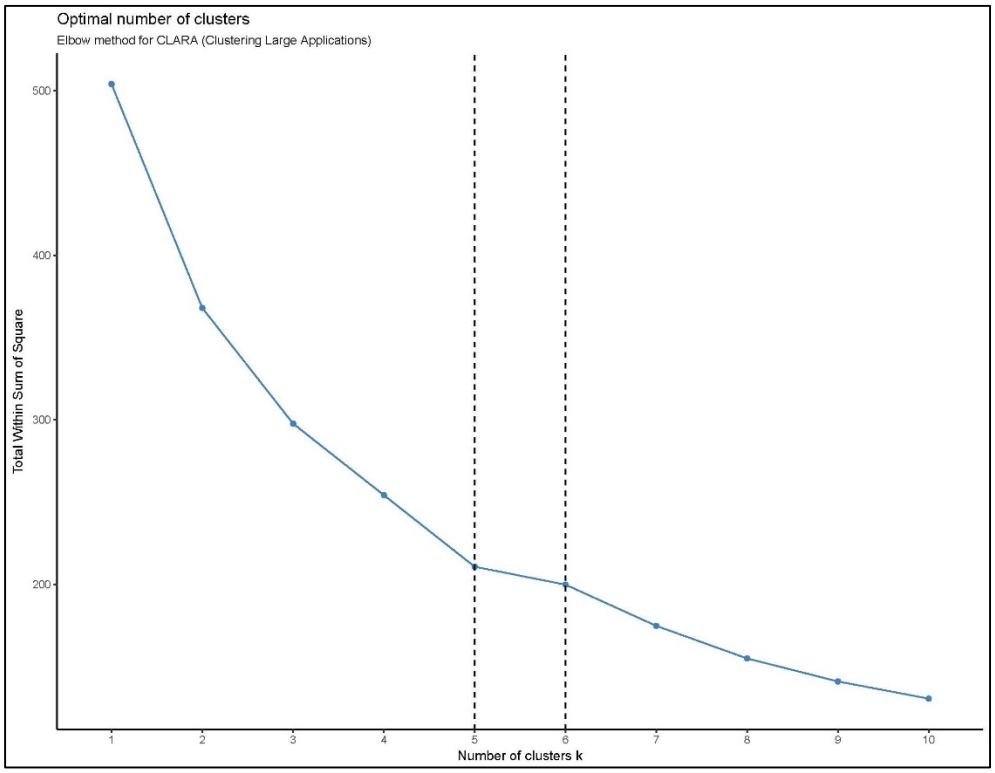
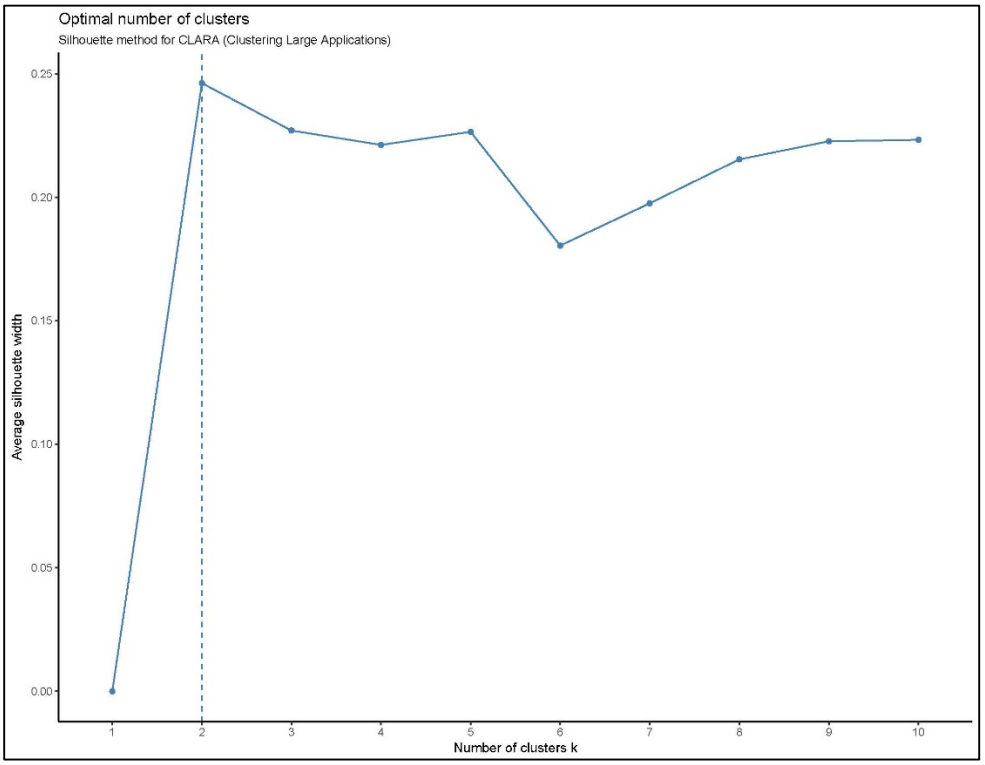


Figure 22



The findings displayed in Figures 13 – 20 are summarized, grouped *per* method, in Table 5 here below.

Table 5

Method	**Clustering Algorithm**	**Optimal number of clusters calculated**
Gap statistic	Hierarchical Clustering	1
	K-means	3
	PAM	2
	CLARA	8
Elbow	K-means	3 / 5
	PAM	3 / 5
	CLARA	5 / 6
Silhouette	K-means	3
	PAM	3
	CLARA	2

Even if several methods provide results rather different among them (*e.g.*, *gap-statistic*) or somehow ambiguous (*e.g.*, the *elbow method* recognizes the possibility of two bends in all tested cases), the majority of them converge towards a common value of three (3). This result is not unexpected as it is consistent with the three main red blocks identified in the *distance* (or *proximity*) *matrix* shown in Figure 1. Moreover, three was also the number of clusters obtained letting the function *HCPC()* of *FactoMineR* to suggest the best level to cut the dendrogram (see Figures 8 and 9).

In light of this, it has been investigated the application of *partitioning clustering* methods to the database under study setting three (3) as optimal number of clusters in the data. The obtained results are as follows:

- *K-means Clustering* [15]: it is probably the most commonly used algorithm for partitioning a dataset into a set of k pre-specified clusters. The basis for this method are high *intra-class similarity* and low *inter-class similarity* [10]. In *K-means Clustering*, each cluster is represented by its center, or *centroid*, which corresponds to the mean of points assigned to the cluster.

Figure 21

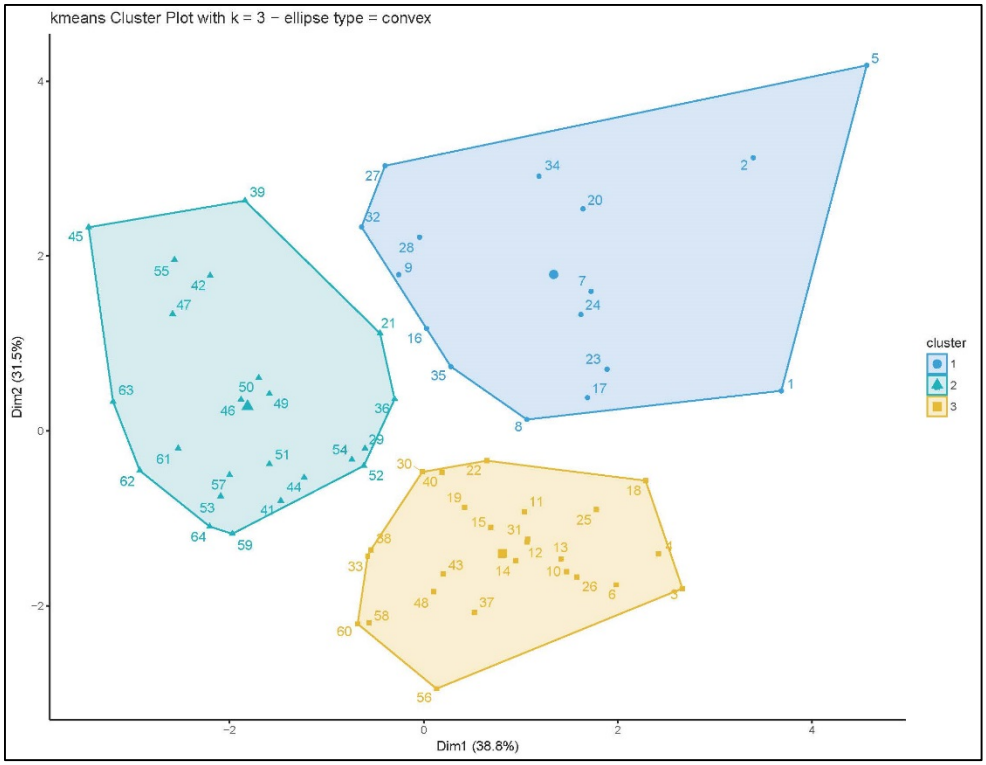


Figure 22

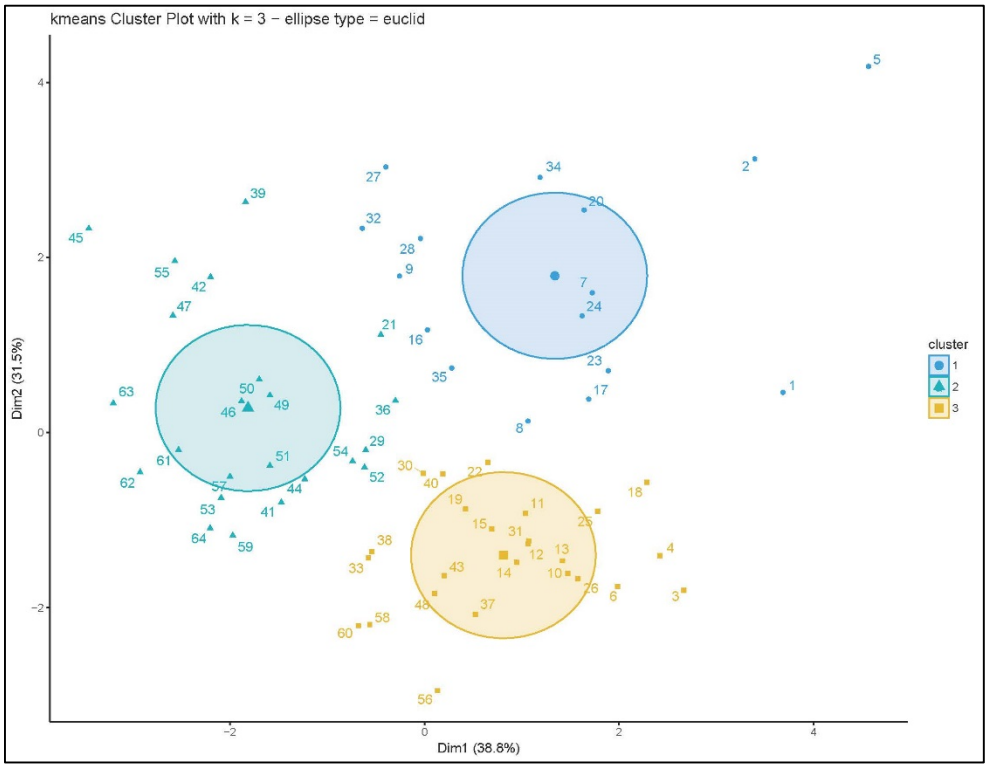


Figure 21 shows the observations (*i.e.*, the solvents), each represented by a point, with a frame drawn around each cluster while Figure 22 shows the same observations, but with a concentration ellipse around each cluster centroid. In particular, the size of the concentration ellipse is in normal probability with a radius equal to level (*i.e.*, ellipse.level = 0.95), representing the Euclidean distance from the center.

Both plots displayed in Figures 21 and 22 have been obtained using the *fviz_cluster()* function of the R package *factoextra* [8,9,10] using PCA to reduce the dimensionality of the initial dataset that was equal to eight.

This second type of visualization is more effective than the first as it gives a much better view of the real state of things. Figure 22, for instance, highlights at a glance the degree of separation existing between data points 1, 2, 5 and the rest of Cluster 1 members. Likewise, for instance, Figure 22 shows the progressive degree of separation existing among the members of a chemical family (*i.e.*, alcohols) moving from cluster's center outwards. In fact, Cluster 3 centroid is close to data point 14 (2- Butanol) and not far from data point 12 (2-Methyl-1- propanol), but it is definitely far from data points 6 (Ethanol) or even more from 3 (Methanol).

The foregoing considerations can also be made examining Figure 21 but in that case the visual impact is lower.

It is anyway interesting to note that Figure 21, which is based on a plot obtained using two principal components, corresponds to the mirror image of Figure 9 that displays a hierarchical clustering based on eight principal components. Interestingly, a *hierarchical technique* (*HCPC*) and a *partitional* one (*K-means*) lead to the same data points assignment.

- *PAM Clustering* [16]: is another algorithm for partitioning a dataset into a set of k pre-specified clusters that is alternative to *K-means*. *PAM* is usually considered more robust than *K-means* as less sensitive to outliers.

Figure 23

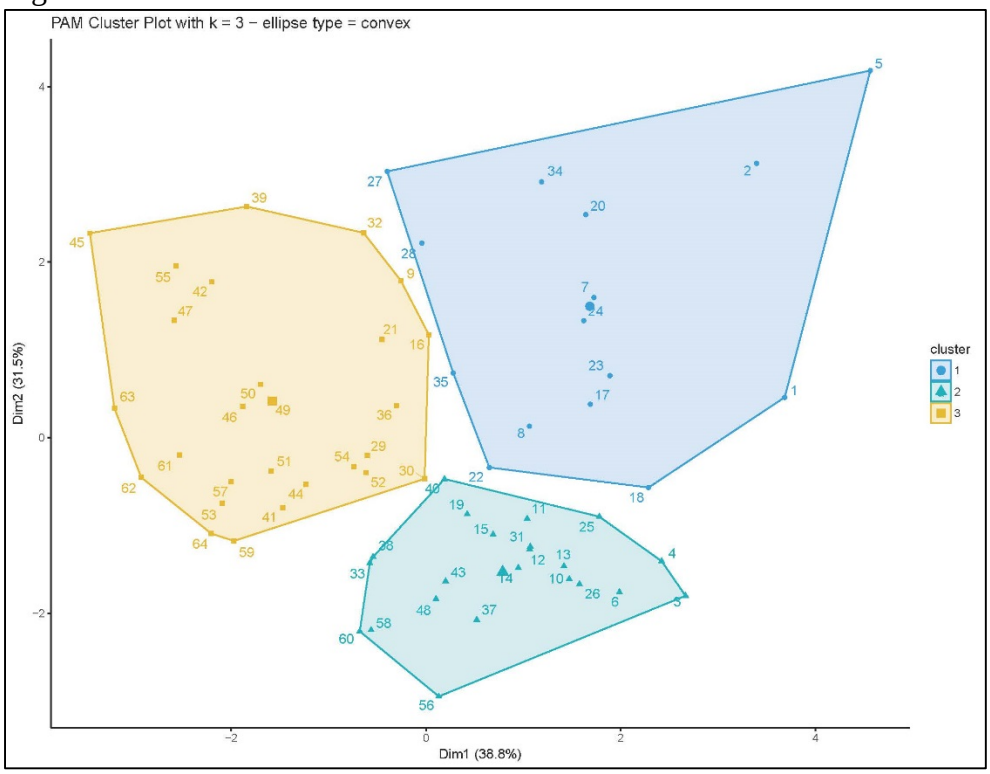
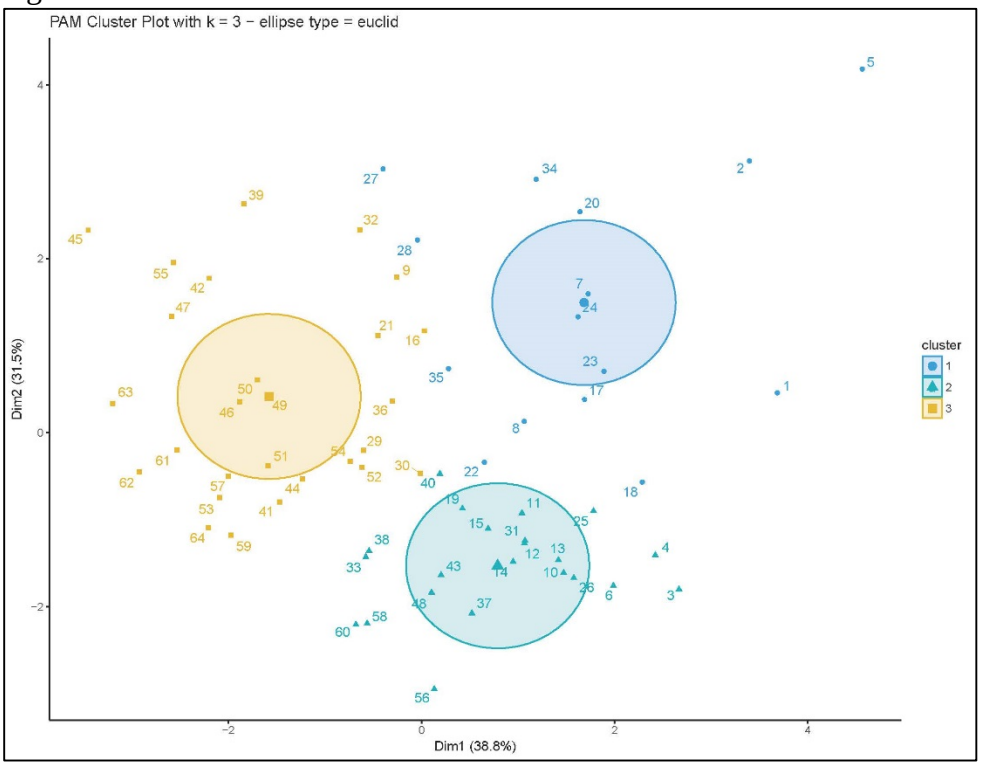


Figure 24



Even if, at a glance, Figures 23 and 24 could look similar to the correspondent Figures 21 and 22, a careful comparison between Figure 23 and 21 reveals several difference. In this case, this type of clustering visualization provides more help than that resulting from concentration ellipses.

The first difference that could be perceived, as of instant visual impact, is the reduction in the separation area between the clusters. In other words, the three clusters of Figure 23 look much closer each other than those displayed in Figure 21. Moreover, many data points in the boundary areas of Figure 23 had been differently assigned in comparison to Figure 21. Benzonitrile (28) and Benzyl alcohol (9), for instance, in the partitioning displayed in Figure 21 have been both assigned to the same cluster (1) while in that of Figure 23 they belong to two different clusters (1 and 3). Clearly, different clustering algorithms may not collect the same set of individuals into the groups.

- *CLARA Clustering* [16]: is a clustering algorithm that extends the PAM approach to deal with large datasets. It is considered just for the sake of comparison, as the database under study certainly does not require it.

Figure 25

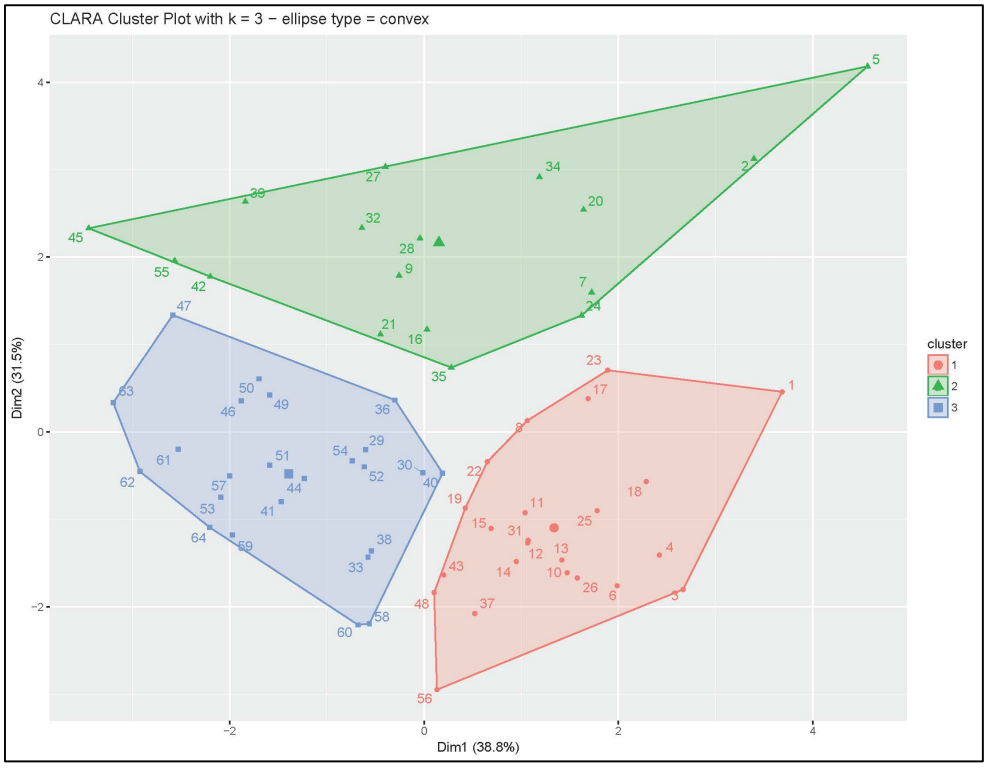
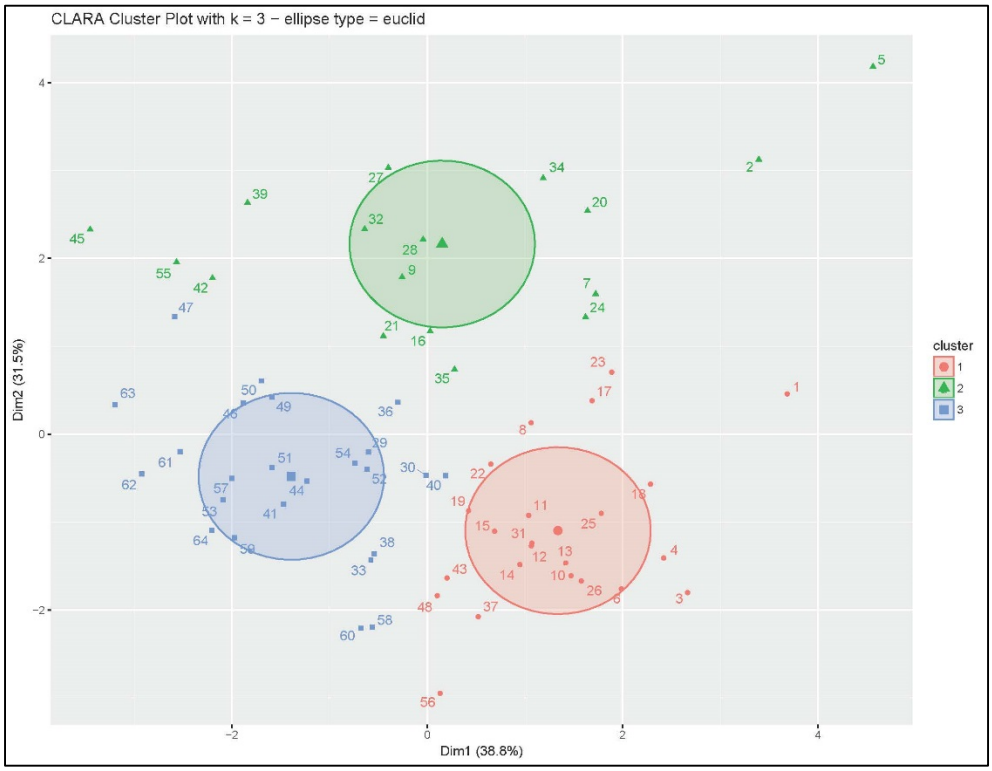


Figure 26



The cluster plots displayed in Figures 25 and 26, as well as those in Figures 21 – 24, have been obtained using the *fviz_cluster()* function of the R package *factoextra* [8,9,10] using PCA to reduce the dimensionality of the initial dataset and, in all cases, the data points position on the scatter plot is the same . However, comparing Figure 25 with Figures 23 and 21, it evident how the partitioning obtained using *CLARA* differs from those obtained using the other *partitional clustering* techniques (*i.e.*, *K-means* and *PAM*). The three clusters are always centered around the same data points (e.g., 8, 28 – 44, 51 – 11, 12), but they are differently aggregated with respect to what obtained using the *hierarchical clustering on principal components* or the *partitional clustering* using *K-means* and *PAM* algorithms. This finding is probably because *CLARA* is usually indicated for large dataset and this is not the case of that under study that is, on the contrary, rather small.

4. CONCLUSIONS

Cluster Analysis, like *Principal Components Analysis* (or PCA) and *Correlation* that were considered in the previous post, is a basic step in data analysis and it "should be used routinely in early description of data, playing the same role for multivariate data that histograms play for univariate data" [18].

In this post, to limit the field of investigation and for the sake of simplicity, the analysis was restricted just to *hard clustering methods*, *i.e.*, to those assigning data points with similar properties to the same group and dissimilar data points to different groups. Two categories of algorithms have been considered: *i.e.*, *hierarchical* and *partitional*.

The main points emerging from the documented experimental evidence can be summarized as follows:

- the *distance* (or *proximity*) *matrix* (Figure 1) reveals the presence of three (3) main blocks, each categorized as consisting of individuals *similar* among them, and of a few isolated individuals (data points: 1,2,4, and 5) that were also detected in the previous post using 2d-contour plots,

- a closer examination of the *distance matrix* reveals, within the three main blocks, a finer structure, detailed in Table 1, consisting of smaller groups of individuals *highly similar* among them (*e.g.*, members of a given chemical family),

- adopting an *agglomerative approach* (*i.e.*, using *hclust()* function) for *hierarchical clustering* it has been obtained a dendrogram (Figure 2) and a cluster plot (Figure 3) reflecting the partitioning provided by the *distance* (or *proximity*) *matrix* of Figure 1,

- the use of an *agglomerative approach* for *hierarchical clustering* using a different R function (*i.e., agnes(),* ***AG***glomerative ***NES***ting) leads to a different partitioning in comparison to those based on the *distance* (or *proximity*) *matrix* or on *agglomerative hierarchical clustering* using *hclust(),*

- the adoption of a *divisive approach* (*e.g.*, using *diana(),* ***DI***visive ***ANA***lysis) for *hierarchical clustering* leads to a dendrogram (Figure 7) that differs from that obtained using an *agglomerative approach* (*e.g., hclust()* or *agnes()*),

- the use of a *hierarchical clustering approach based on principal components* combined with an automatic cut of the tree, leads to a final partitioning in three clusters no matter how many components are considered (Figures 8 and 9). Each cluster approximately corresponds to a block of the *distance* (or *proximity*) *matrix* (Figure 1). In this case cutting the hierarchical tree upon suggestion of *NbClust()* function and using eight principal components leads to a better definition (Figure 12),

- *K-means* and *PAM partitioning clustering* methods lead to similar partitioning, apart from a few differences. Interestingly, a *hierarchical technique* (*HCPC*, Figure 9) and a *partitional* one (*K-means*, Figure 21) lead to the same data points assignment,

- *CLARA (**C**lustering **LAR**ge **A**pplications)*, even if is a *partitioning clustering* method, it leads to a pattern completely different from that obtained using *K-means* and *PAM* methods and this probably because the database under study is rather small.

To conclude, the cluster analysis provides evidence that the 64 solvents considered in this study and characterized by the eight physico-chemical descriptors reported in the previous post, may be roughly grouped into three (3) major classes (or clusters). This finding follows from both *hierarchical* and *non-hierarchical* partitioning methods. Cluster analysis also reveals an underlying finer structure consisting of smaller clusters each formed by individuals highly similar among them (*e.g.*, members of a given chemical family (*e.g.*, alcohols, chlorinated hydrocarbons) or chemical entities sharing common characteristics (*e.g.*, aprotic dipolar solvents)).

5. ACKNOWLEDGMENTS

I wish to express my deepest gratitude and appreciation to the R Foundation, RStudio and to all Authors of the R packages that I have used in this post.

6. BIBLIOGRAPHY

1. C. Chatfield, A.J. Collins, *Introduction to Multivariate Analysis*, Chapman and Hall, London, 1980
2. G. Gan, C. Ma, J. Wu, *Data Clustering. Theory, Algorithms and Applications*, ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA, 2007
3. B.S. Everitt, S. Landau, M. Leese, D. Stahl, *Cluster Analysis*, 5th Edition, Wiley (2011) 16-24
4. S. Ben-David, *Clustering - What Both Theoreticians and Practitioners are Doing Wrong*, arXiv: 1805.08838 [cs.LG], 22 May 2018
5. H. Wickham, G. Grolemund, *R for Data Science*, 2017, O'Reilly
6. F. Husson, S. Lê, J. Pagès, *Exploratory Multivariate Analysis by Example using R*, 2011, CRC Press
7. F. Husson, J. Josse, J. Pagès, *Principal component methods – hierarchical clustering – partitional clustering: why would we need to choose for visualizing data?*, September 2010, Technical Report - Agrocampus
8. A. Kassambara, *R Graphics Essentials for Great Data Visualization*, STHDA, 2017
9. A. Kassambara, *Practical Guide to Principal Component Methods in R*, STHDA, 2017
10. A. Kassambara, *Practical Guide to Cluster Analysis in R*, STHDA, 2017
11. M. Friendly, *Corrgrams: Exploratory displays for correlation matrices*, The American Statistician, 56 (2002) 316-324
12. M. Charrad, N. Ghazzali, V. Boiteau and A. Niknafs, *NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set* , J. Stat. Soft., 61(6), (2014) 1-36

13. W.R. Dillon, M. Goldstein, *Multivariate Analysis. Methods and Applications*, J. Wiley and Sons, New York, 1984, Chapter 5

14. A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988

15. J. MacQueen, *Some methods for classification and analysis of multivariate observations*, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, eds. L.M. LeCam & J. Neyman, Berkeley, CA, University of California Press, 1 (1967) 281-297

16. L. Kaufman, P.J. Rousseeuw, *Finding Groups in Data: An introduction to Cluster Analysis*, Wiley, New York (1990)

17. R. Carlson, T. Lundstedt, C. Albano, *Screening of Suitable Solvents in Organic Synthesis. Strategies for Solvents Selection*, Acta Chemica Scandinavica B, 39 (1985) 79-91

18. J.A. Hartigan, Clustering Algorithms, John Wiley and Sons, New York (1975) vii