1 Subject

Implementing the Kohonen's SOM (Self Organizing Map) algorithm with Tanagra.

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a kind of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. Self-organizing maps are different than other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space (http://en.wikipedia.org/wiki/Self-organizing_map).

SOM is a clustering method. Indeed, it organizes the data in clusters (cells of map) such as the instances in the same cell are similar, and the instances in different cells are different. In this point of view, SOM gives comparable results to state-of-the-art clustering algorithm such as K-Means (Forgy, 1965; McQueen, 1967).

SOM can be viewed also as a visualization technique. It allows us to visualize in a low dimensional representation space (2D) the original dataset. Indeed, the individuals located in adjacent cells are more similar than individuals located in distant cells. In this point of view, it is comparable to visualization techniques such as Multidimensional scaling (Kruskal, 1978) or PCA (Principal Component Analysis, http://en.wikipedia.org/wiki/Principal_component_analysis).

In this tutorial, we show how to implement the Kohonen's SOM algorithm with Tanagra. We try to assess the properties of this approach by comparing the results with those of the PCA algorithm. Then, we compare the results to those of K-Means, which is a clustering algorithm. Finally, we implement the Two-step Clustering process by combining the SOM algorithm with the HAC process (Hierarchical Agglomerative Clustering). It is a variant of the Two-Step Clustering where we combine K-Means and HAC (http://data-mining-tutorials.blogspot.com/2009/06/two-step-clustering-for-handling-large.html). We observe that the HAC primarily merges the adjacent cells.

2 Dataset

We analyze the WAVEFORM dataset (Breiman and al., 1984)[1]. This is an artificial dataset[2].

There are 21 descriptors. We have 5000 instances. We do not use the CLASS attribute which classify the instances into 3 pre-defined classes in this tutorial.

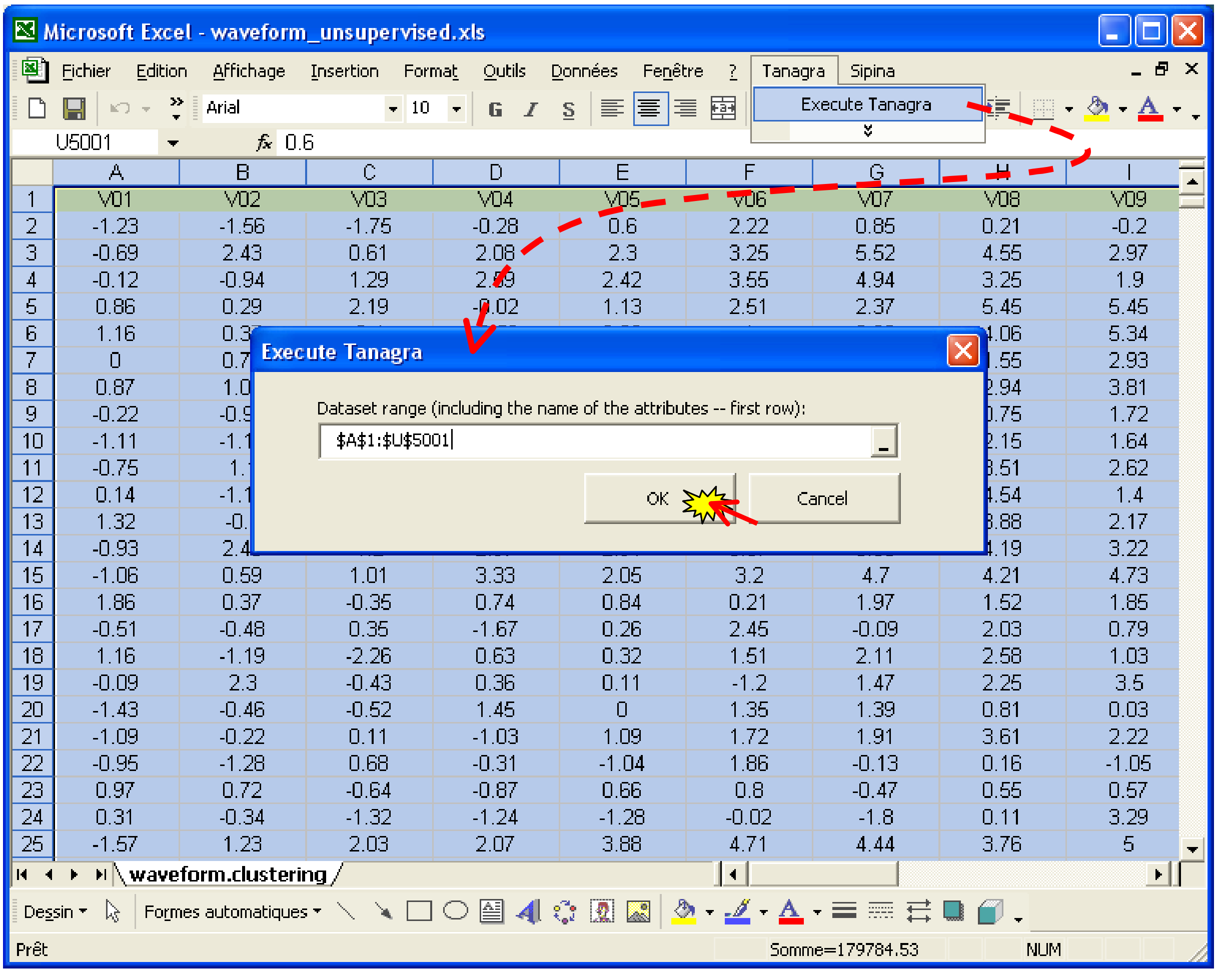
[1] http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/waveform_unsupervised.xls

[2] http://archive.ics.uci.edu/ml/datasets/Waveform+Database+Generator+%28Version+1%29

3 Kohonen's SOM approach with Tanagra

3.1 Importing the data file

The easiest way to import a XLS file is to open the data file into Excel spreadsheet. Then, using the add-in TANAGRA.XLA[3], we can send the dataset to Tanagra which is launched automatically. We can check the range of selected cells in the worksheet.


Microsoft Excel - waveform_unsupervised.xls
Fichier Edition Affichage Insertion Format Outils Données Fenêtre ? Tanagra Sipina
Execute Tanagra
Arial 10
U5001
0.6
waveform.clustering
Dessin Formes automatiques
Prêt
Somme=179784.53
NUM

	A	B	C	D	E	F	G	H	I
1	V01	V02	V03	V04	V05	V06	V07	V08	V09
2	-1.23	-1.56	-1.75	-0.28	0.6	2.22	0.85	0.21	-0.2
3	-0.69	2.43	0.61	2.08	2.3	3.25	5.52	4.55	2.97
4	-0.12	-0.94	1.29	2.59	2.42	3.55	4.94	3.25	1.9
5	0.86	0.29	2.19	-0.02	1.13	2.51	2.37	5.45	5.45
6	1.16	0.3						4.06	5.34
7	0	0.7						1.55	2.93
8	0.87	1.0						2.94	3.81
9	-0.22	-0.9						0.75	1.72
10	-1.11	-1.1						2.15	1.64
11	-0.75	1.1						3.51	2.62
12	0.14	-1.1						4.54	1.4
13	1.32	-0.						3.88	2.17
14	-0.93	2.4						4.19	3.22
15	-1.06	0.59	1.01	3.33	2.05	3.2	4.7	4.21	4.73
16	1.86	0.37	-0.35	0.74	0.84	0.21	1.97	1.52	1.85
17	-0.51	-0.48	0.35	-1.67	0.26	2.45	-0.09	2.03	0.79
18	1.16	-1.19	-2.26	0.63	0.32	1.51	2.11	2.58	1.03
19	-0.09	2.3	-0.43	0.36	0.11	-1.2	1.47	2.25	3.5
20	-1.43	-0.46	-0.52	1.45	0	1.35	1.39	0.81	0.03
21	-1.09	-0.22	0.11	-1.03	1.09	1.72	1.91	3.61	2.22
22	-0.95	-1.28	0.68	-0.31	-1.04	1.86	-0.13	0.16	-1.05
23	0.97	0.72	-0.64	-0.87	0.66	0.8	-0.47	0.55	0.57
24	0.31	-0.34	-1.32	-1.24	-1.28	-0.02	-1.8	0.11	3.29
25	-1.57	1.23	2.03	2.07	3.88	4.71	4.44	3.76	5

Execute Tanagra

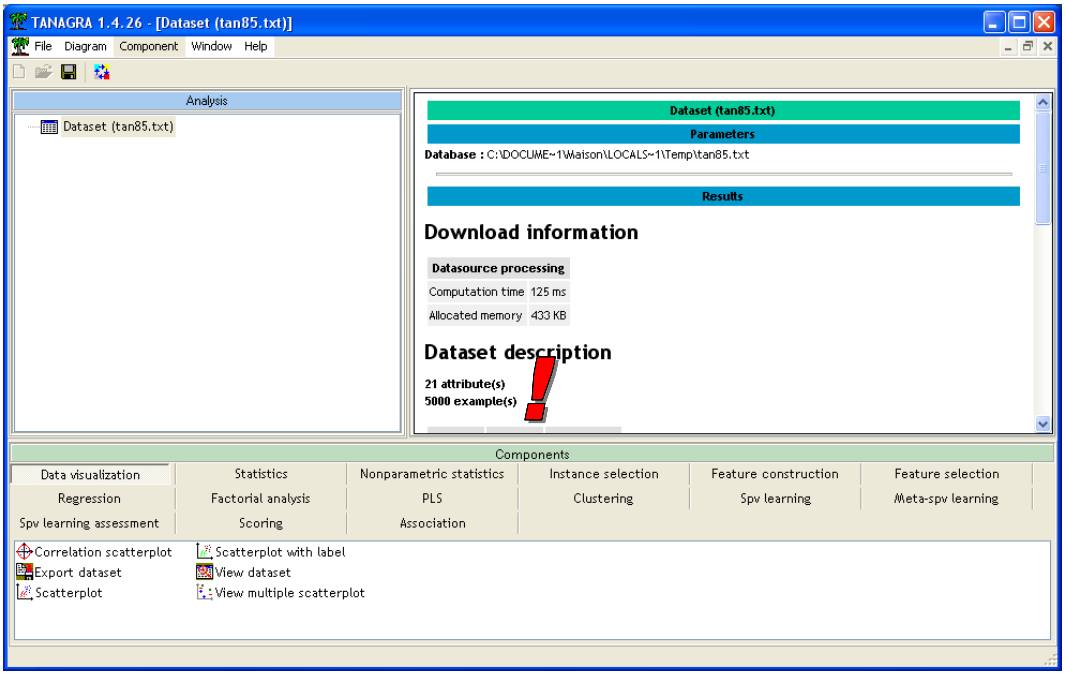
Dataset range (including the name of the attributes -- first row):

$A$1:$U$5001

Tanagra is launched, a new diagram is created and the dataset is loaded. We have 5000 instances and 21 attributes.

[3] See http://data-mining-tutorials.blogspot.com/2008/10/excel-file-handling-using-add-in.html; Tanagra can handle also the XLS file even if Excel is not installed on our computer http://data-mining-tutorials.blogspot.com/2008/10/excel-file-format-direct-importation.html

In addition, Tanagra can be incorporated also in Open Office Calc spreadsheet, under Windows http://data-mining-tutorials.blogspot.com/2008/10/ooocalc-file-handling-using-add-in.html or under Linux http://data-mining-tutorials.blogspot.com/2009/04/launching-tanagra-from-oocalc-under.html.



Dataset (tan85.txt)

Parameters

Database : C:\DOCUME~1\Maison\LOCALS~1\Temp\tan85.txt

Results

Download information

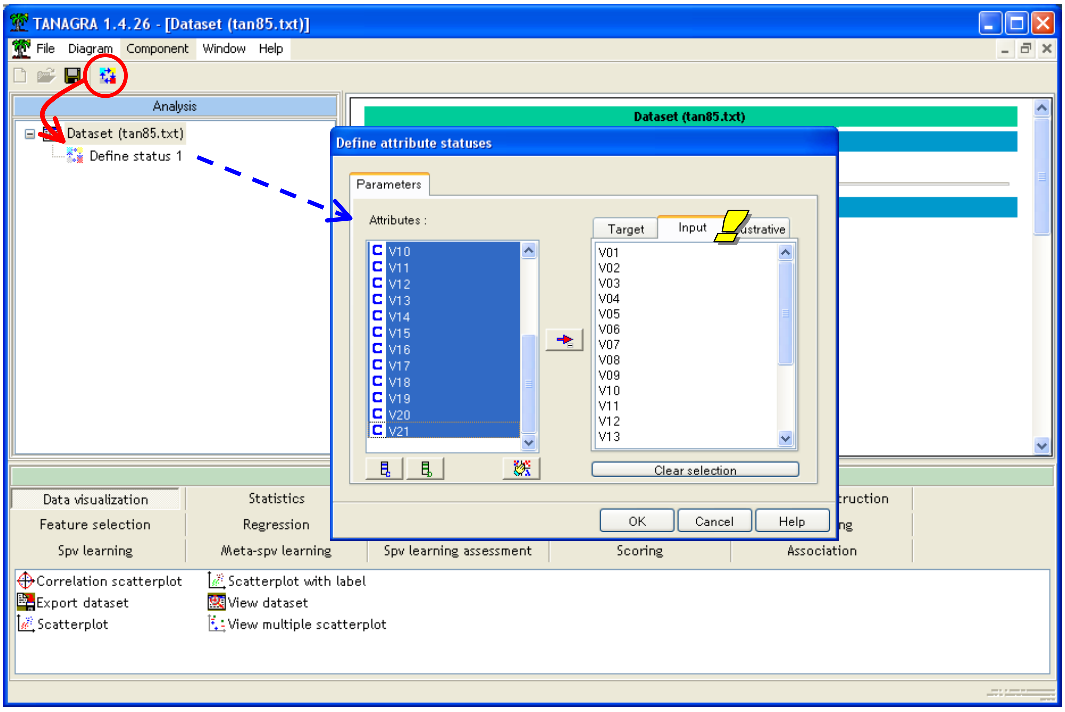
Datasource processing	
Computation time	125 ms
Allocated memory	433 KB

Dataset description

21 attribute(s)
5000 example(s)

3.2 Descriptive statistics and outliers detection

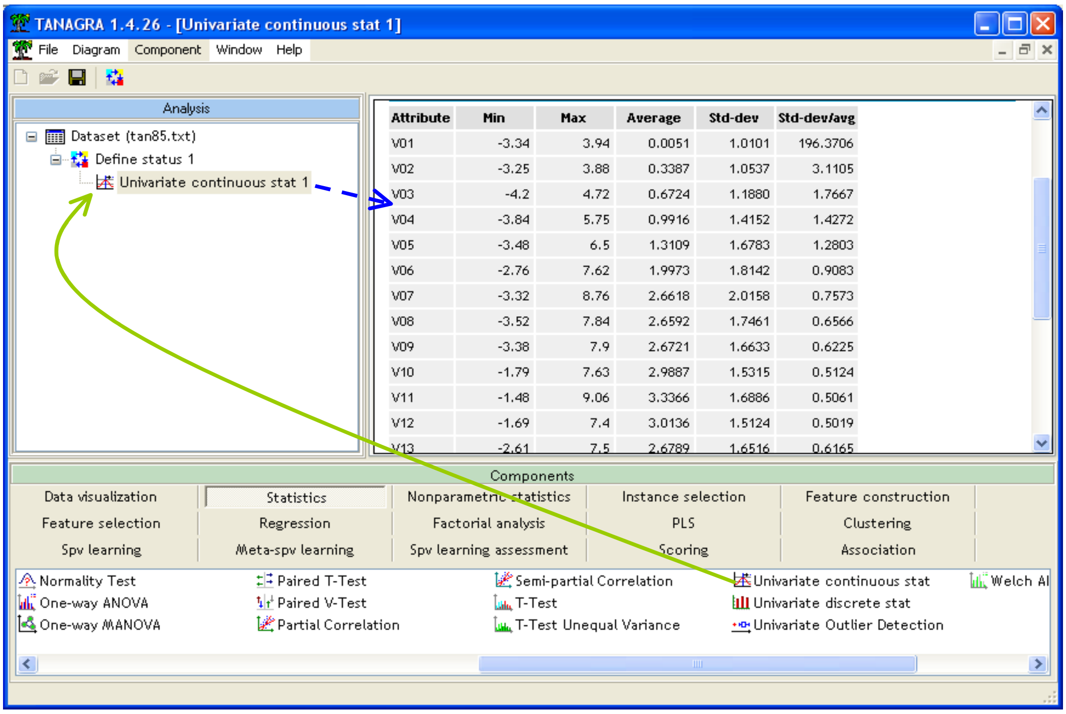
First step, we check the integrity of the dataset by computing some descriptive statistics indicators. We insert the DEFINE STATUS component into the diagram using the shortcut into the toolbar. Then we set all the variables as INPUT. **Caution: the variables must be continuous if we want to perform a Kohonen'SOM processing below**.



Attributes : C V10, C V11, C V12, C V13, C V14, C V15, C V16, C V17, C V18, C V19, C V20, C V21

Input : V01, V02, V03, V04, V05, V06, V07, V08, V09, V10, V11, V12, V13

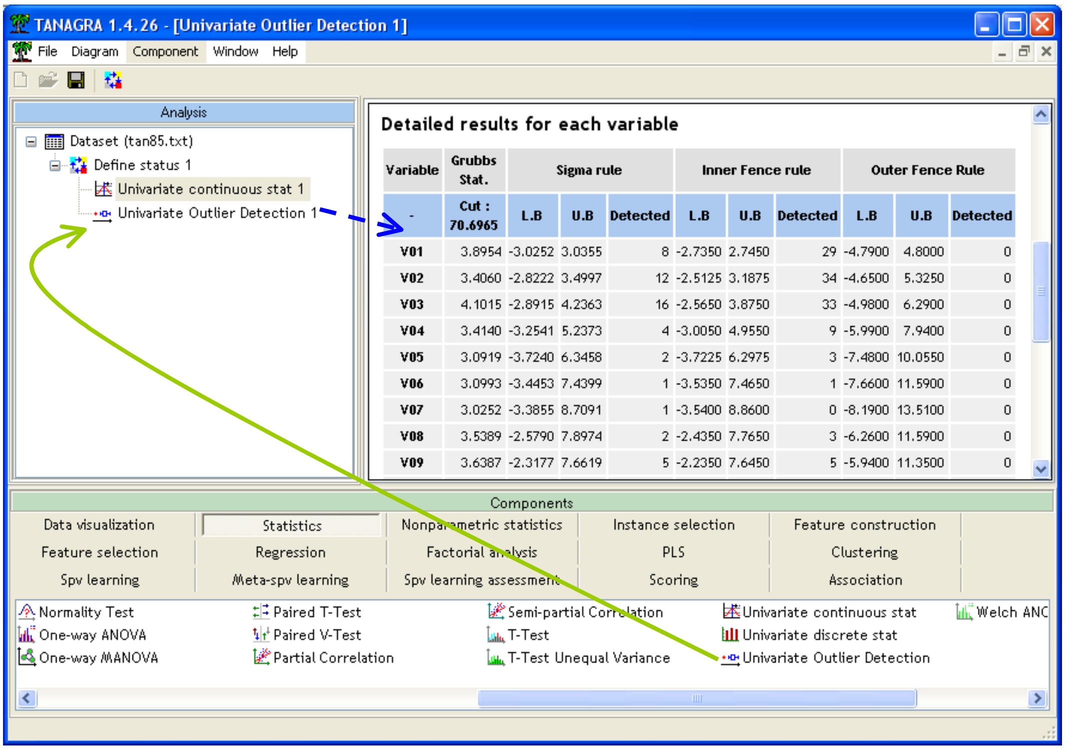
We add the UNIVARIATE CONTINUOUS STAT component (STATISTICS tab). We click on the VIEW menu. We obtain the following report.


TANAGRA 1.4.26 - [Univariate continuous stat 1]
File Diagram Component Window Help
Analysis
Dataset (tan85.txt)
Define status 1
Univariate continuous stat 1
Components
Data visualization Statistics Nonparametric statistics Instance selection Feature construction
Feature selection Regression Factorial analysis PLS Clustering
Spv learning Meta-spv learning Spv learning assessment Scoring Association
Normality Test Paired T-Test Semi-partial Correlation Univariate continuous stat Welch AI
One-way ANOVA Paired V-Test T-Test Univariate discrete stat
One-way MANOVA Partial Correlation T-Test Unequal Variance Univariate Outlier Detection

Attribute	Min	Max	Average	Std-dev	Std-dev/avg
V01	-3.34	3.94	0.0051	1.0101	196.3706
V02	-3.25	3.88	0.3387	1.0537	3.1105
V03	-4.2	4.72	0.6724	1.1880	1.7667
V04	-3.84	5.75	0.9916	1.4152	1.4272
V05	-3.48	6.5	1.3109	1.6783	1.2803
V06	-2.76	7.62	1.9973	1.8142	0.9083
V07	-3.32	8.76	2.6618	2.0158	0.7573
V08	-3.52	7.84	2.6592	1.7461	0.6566
V09	-3.38	7.9	2.6721	1.6633	0.6225
V10	-1.79	7.63	2.9887	1.5315	0.5124
V11	-1.48	9.06	3.3366	1.6886	0.5061
V12	-1.69	7.4	3.0136	1.5124	0.5019
V13	-2.61	7.5	2.6789	1.6516	0.6165

We note that there is no constant in our dataset i.e. standard deviation = 0. We note also that all the variables seem defined in the same scale.

We can complete this dataset checkout by detecting eventual outliers. We add the UNIVARIATE OUTLIER DETECTION (STATISTICS tab) component[4].


TANAGRA 1.4.26 - [Univariate Outlier Detection 1]
File Diagram Component Window Help
Analysis
Dataset (tan85.txt)
Define status 1
Univariate continuous stat 1
Univariate Outlier Detection 1
Components
Data visualization Statistics Nonparametric statistics Instance selection Feature construction
Feature selection Regression Factorial analysis PLS Clustering
Spv learning Meta-spv learning Spv learning assessment Scoring Association
Normality Test Paired T-Test Semi-partial Correlation Univariate continuous stat Welch ANC
One-way ANOVA Paired V-Test T-Test Univariate discrete stat
One-way MANOVA Partial Correlation T-Test Unequal Variance Univariate Outlier Detection

Detailed results for each variable

Variable	Grubbs Stat.	Sigma rule			Inner Fence rule			Outer Fence Rule		
-	Cut : 70.6965	L.B	U.B	Detected	L.B	U.B	Detected	L.B	U.B	Detected
V01	3.8954	-3.0252	3.0355	8	-2.7350	2.7450	29	-4.7900	4.8000	0
V02	3.4060	-2.8222	3.4997	12	-2.5125	3.1875	34	-4.6500	5.3250	0
V03	4.1015	-2.8915	4.2363	16	-2.5650	3.8750	33	-4.9800	6.2900	0
V04	3.4140	-3.2541	5.2373	4	-3.0050	4.9550	9	-5.9900	7.9400	0
V05	3.0919	-3.7240	6.3458	2	-3.7225	6.2975	3	-7.4800	10.0550	0
V06	3.0993	-3.4453	7.4399	1	-3.5350	7.4650	1	-7.6600	11.5900	0
V07	3.0252	-3.3855	8.7091	1	-3.5400	8.8600	0	-8.1900	13.5100	0
V08	3.5389	-2.5790	7.8974	2	-2.4350	7.7650	3	-6.2600	11.5900	0
V09	3.6387	-2.3177	7.6619	5	-2.2350	7.6450	5	-5.9400	11.3500	0

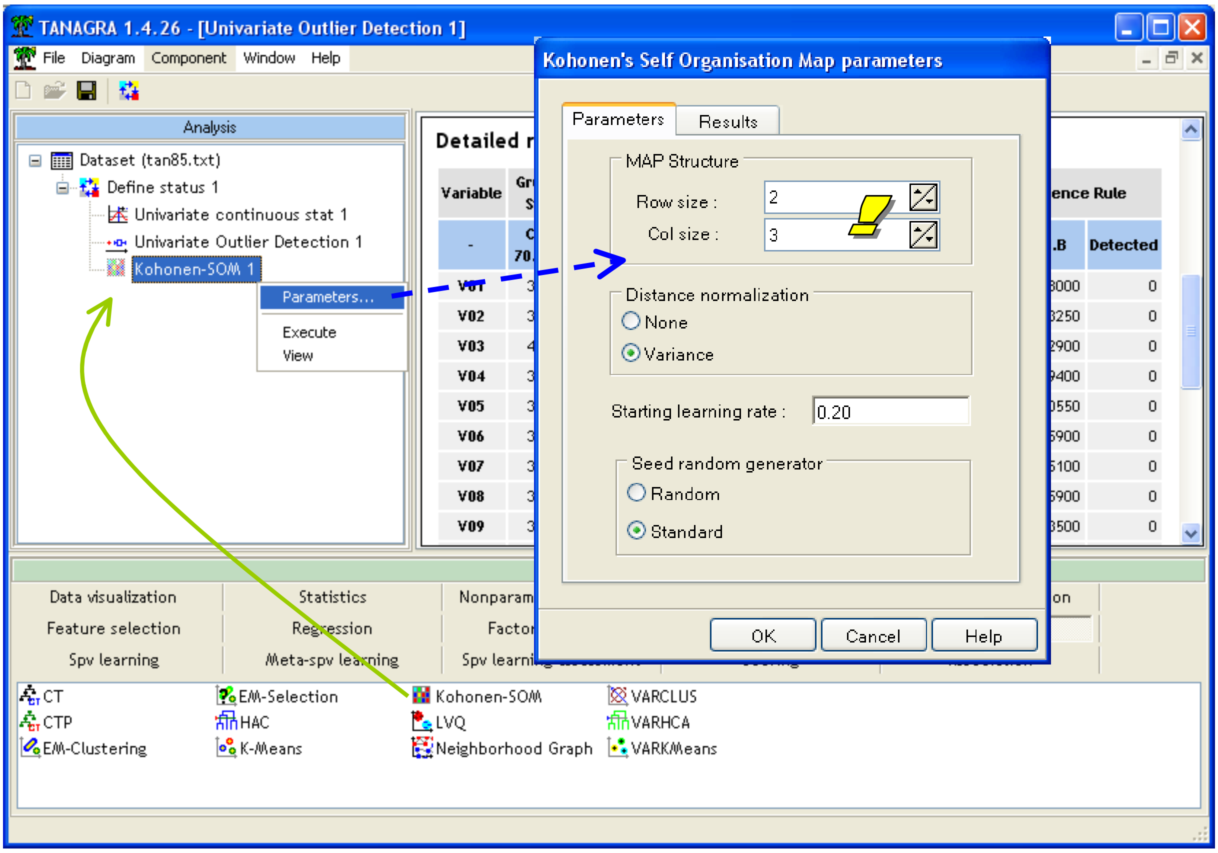
[4] See http://data-mining-tutorials.blogspot.com/2009/06/univariate-outlier-detection-methods.html

Whatever the variable, there is no unusual value. It is not surprising. We know that this dataset is generated artificially.

3.3 The KOHONEN-SOM component

We want to launch the analysis now. We create a grid with 2 rows and 3 columns i.e. a classification of the instances into 6 groups (2 x 3 = 6 clusters).

We add the KOHONEN-SOM component (CLUSTERING tab) into the diagram. We click on the PARAMETERS menu. We set the following settings.



TANAGRA 1.4.26 - [Univariate Outlier Detection 1]

- Dataset (tan85.txt)
  - Define status 1
    - Univariate continuous stat 1
    - Univariate Outlier Detection 1
    - Kohonen-SOM 1

Kohonen's Self Organisation Map parameters

MAP Structure
Row size : 2
Col size : 3

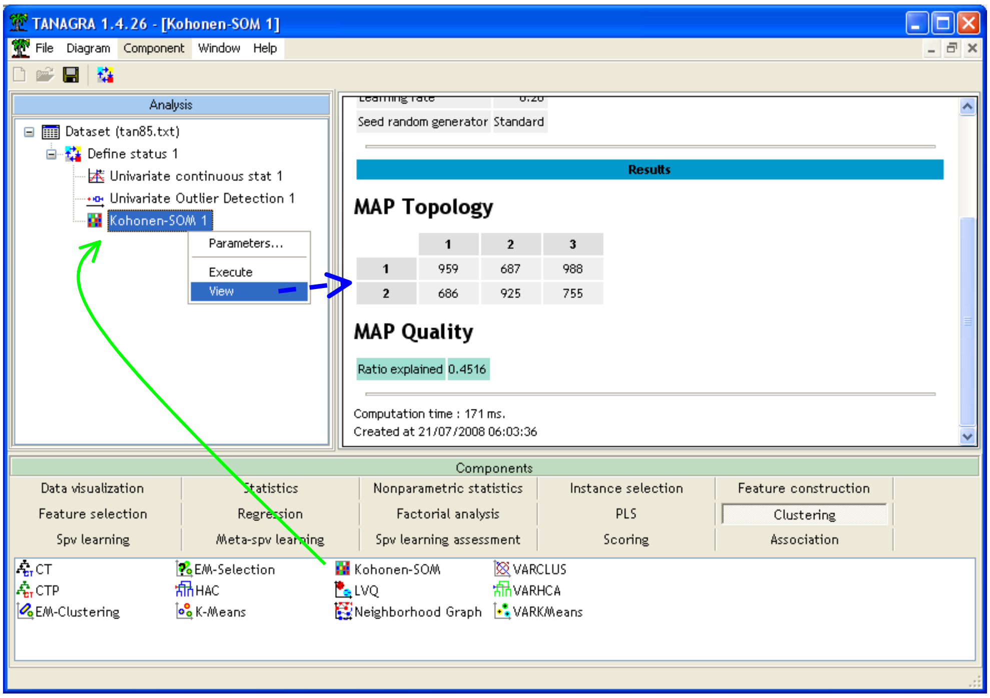
Distance normalization
○ None
● Variance

Starting learning rate : 0.20

Seed random generator
○ Random
● Standard

The number of rows of the map is 2 (Row Size), the number of columns is 3 (Col Size). We standardize the data i.e. we divide each variable by their standard deviation. It is recommended when the variables are not in the same scale. It is not necessary if they are defined in the same scale or when we want to take into consideration explicitly the differences in scale.

We do not modify the other settings (see http://en.wikipedia.org/wiki/Self-organizing_map about SOM approach and its settings). We validate and we click on the VIEW menu. We obtain the following report.



Learning rate	0.20
Seed random generator	Standard

Results

MAP Topology

	1	2	3
1	959	687	988
2	686	925	755

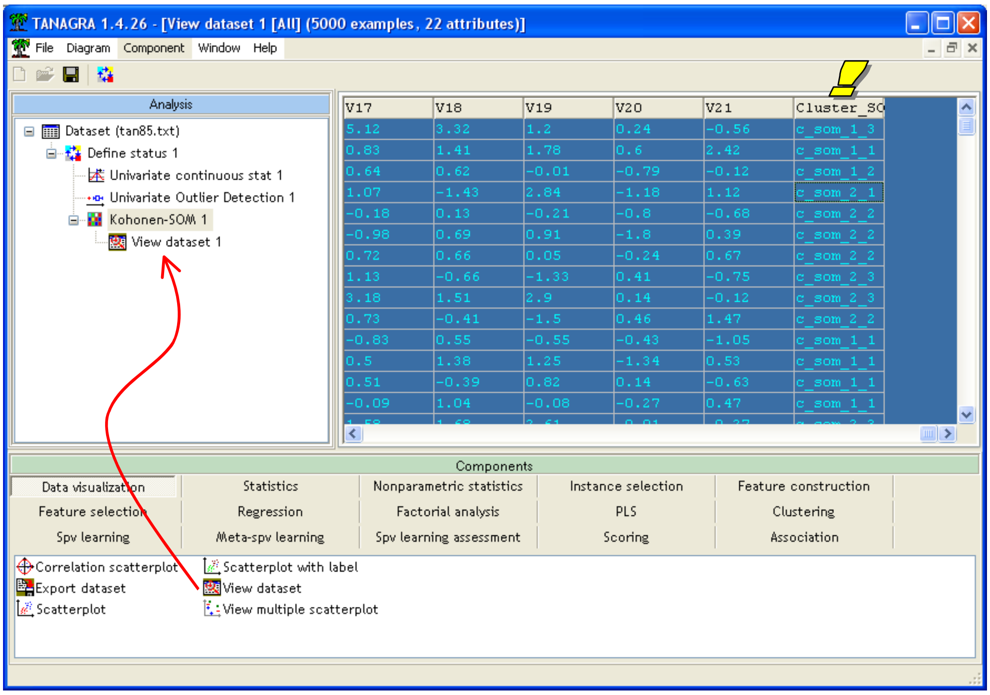
MAP Quality

Ratio explained	0.4516

Computation time : 171 ms.
Created at 21/07/2008 06:03:36

The number of instances into each cell is displayed. We see that **45.16 %** of the TSS (Total Sum of Squares) is explained by the partitioning into 6 classes. We can compare this result to those of the other methods below.

3.4 Classification of the instances



V17	V18	V19	V20	V21	Cluster_SO
5.12	3.32	1.2	0.24	-0.56	c_som_1_3
0.83	1.41	1.78	0.6	2.42	c_som_1_1
0.64	0.62	-0.01	-0.79	-0.12	c_som_1_2
1.07	-1.43	2.84	-1.18	1.12	c_som_2_1
-0.18	0.13	-0.21	-0.8	-0.68	c_som_2_2
-0.98	0.69	0.91	-1.8	0.39	c_som_2_2
0.72	0.66	0.05	-0.24	0.67	c_som_2_2
1.13	-0.66	-1.33	0.41	-0.75	c_som_2_3
3.18	1.51	2.9	0.14	-0.12	c_som_2_3
0.73	-0.41	-1.5	0.46	1.47	c_som_2_2
-0.83	0.55	-0.55	-0.43	-1.05	c_som_1_1
0.5	1.38	1.25	-1.34	0.53	c_som_1_1
0.51	-0.39	0.82	0.14	-0.63	c_som_1_1
-0.09	1.04	-0.08	-0.27	0.47	c_som_1_1

The KOHONEN-SOM component adds a new column to the current dataset. It states the group membership of each instance. This new attribute is available in the subsequent part of the diagram. We can visualize the current dataset with the VIEW DATASET component. We see for instance that the first example belongs to the cell (1, 3) i.e. first row and third column.

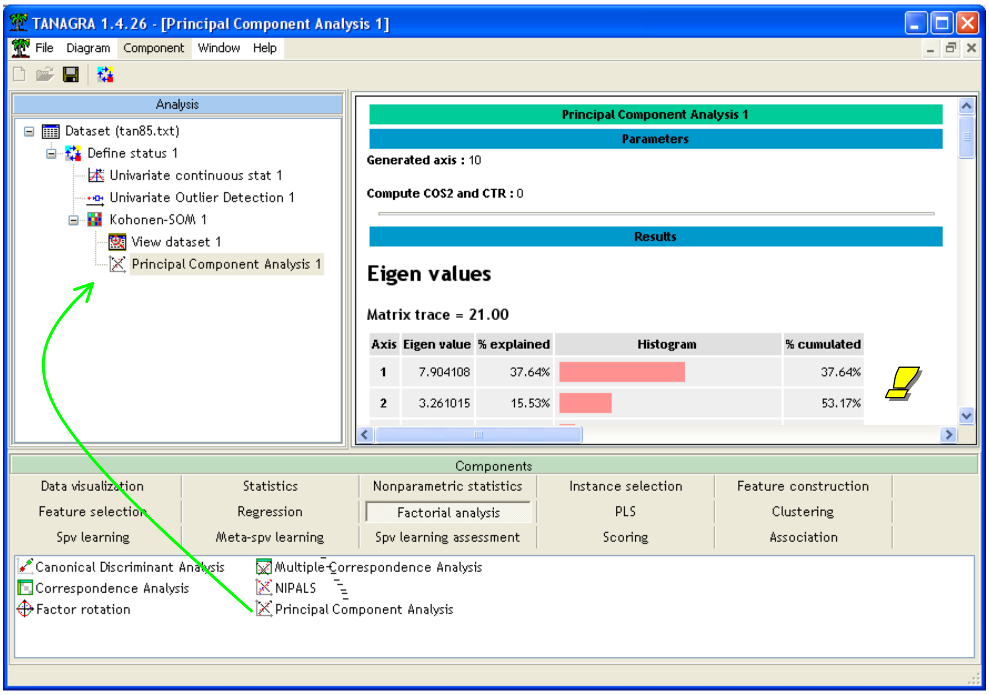
Note: we can classify an additional instance with the same framework i.e. an example which is not involved in the learning process. This deployment phase is one of the most important steps of the Data Mining process[5].

4 Proximity of nodes into the grid

Individuals who are in adjacent cells are also close in the original representation space. This is one of the main interests of this method. Let us check this assertion on the WAVEFORM dataset.

We cannot visualize the dataset into the original space. So we use a PCA in order to obtain a 2D representation. We try to visualize the relative positions of groups (clusters) in the scatter plot.

We add the PRINCIPAL COMPONENT ANALYSIS component (FACTORIAL ANALYSIS tab) after the KOHONEN-SOM 1 component. We click on the VIEW menu.



Principal Component Analysis 1

Parameters

Generated axis : 10

Compute COS2 and CTR : 0

Results

Eigen values

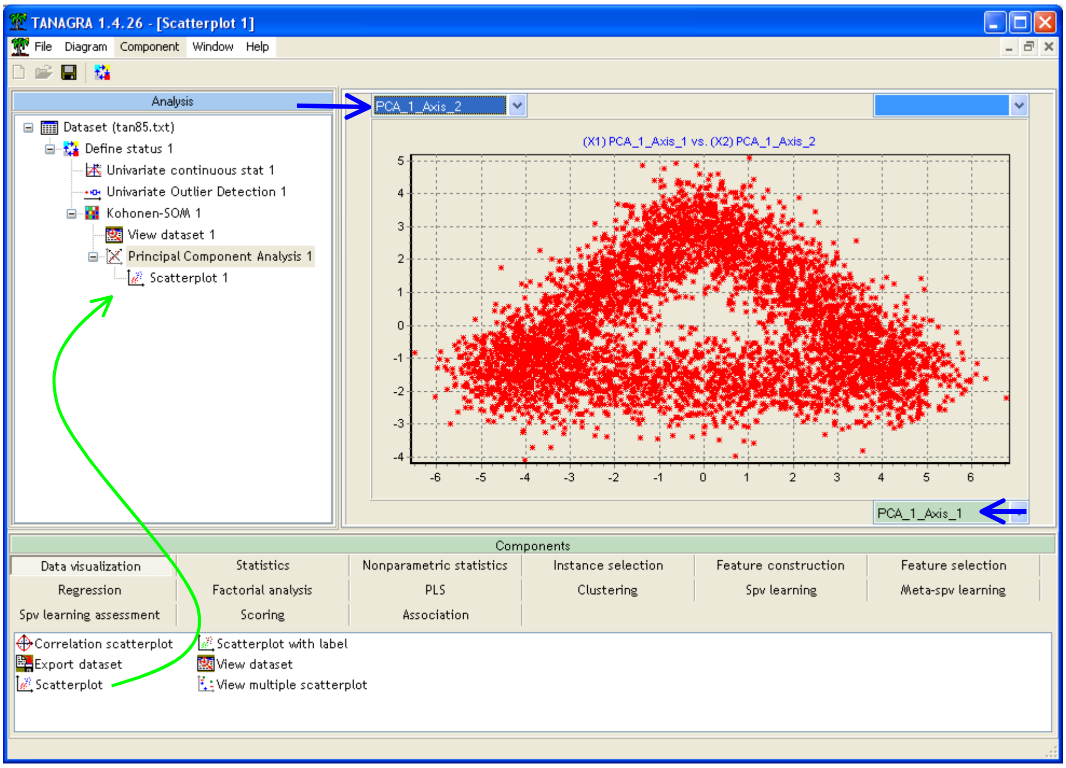
Matrix trace = 21.00

Axis	Eigen value	% explained	Histogram	% cumulated
1	7.904108	37.64%		37.64%
2	3.261015	15.53%		53.17%

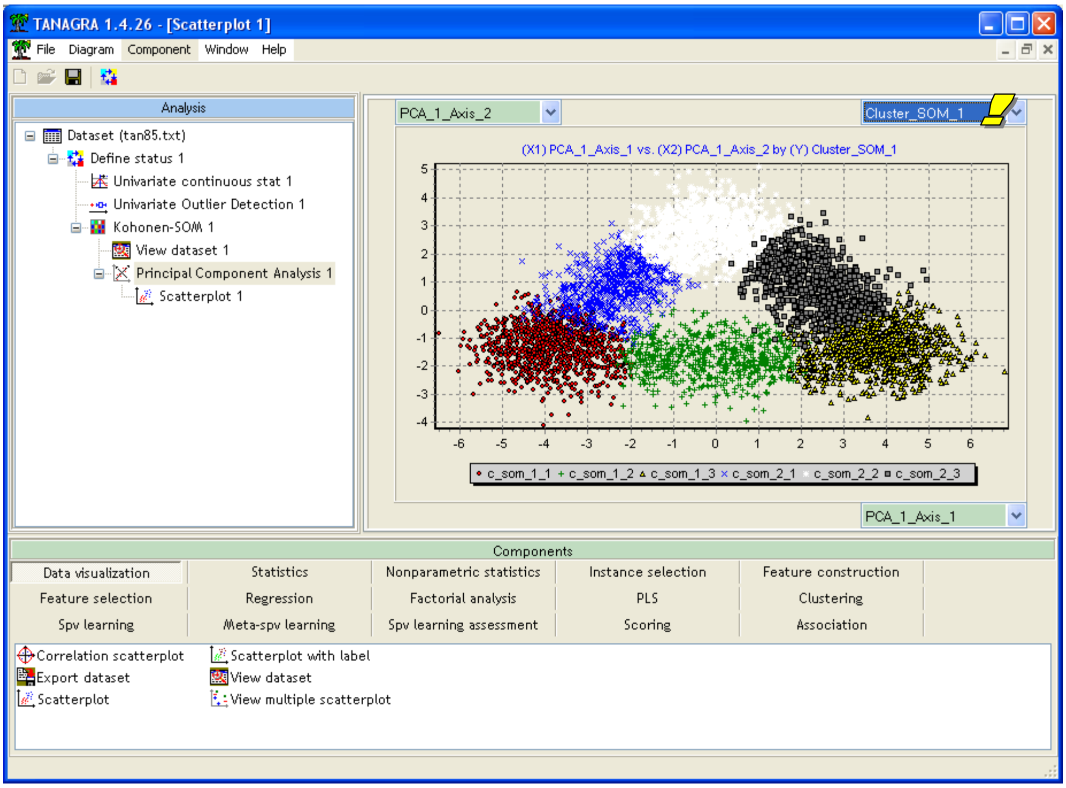
The first two factors representation space accounts for the 53.17% of the total variability. It seems weak, but on this dataset, it is enough to represent properly the instances.

We add the SCATTERPLOT component (DATA VISUALIZATION tab) into the diagram. We set the first factor as the horizontal axis, the second one as vertical axis.

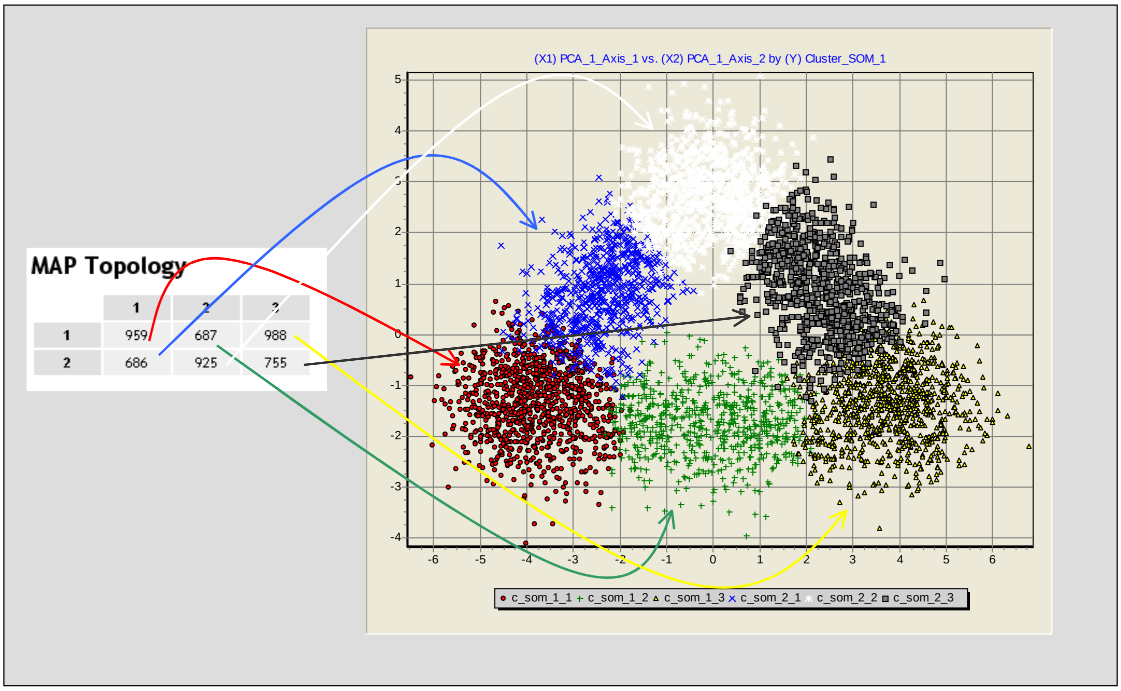
[5] See http://data-mining-tutorials.blogspot.com/2008/12/k-means-classification-of-new-instance.html for the unsupervised learning framework; http://data-mining-tutorials.blogspot.com/2008/11/apply-classifier-on-new-dataset.html for the supervised learning framework.



A crucial step of this tutorial, we colorize the points with the cluster membership supplied by the SOM algorithm (CLUSTER_SOM_1).



We note the correspondence between the proximities into the SOM map and the proximities into the 2 first factors of PCA. It means also that the instances into adjacent cells are close into the original representation space (with 21 attributes).



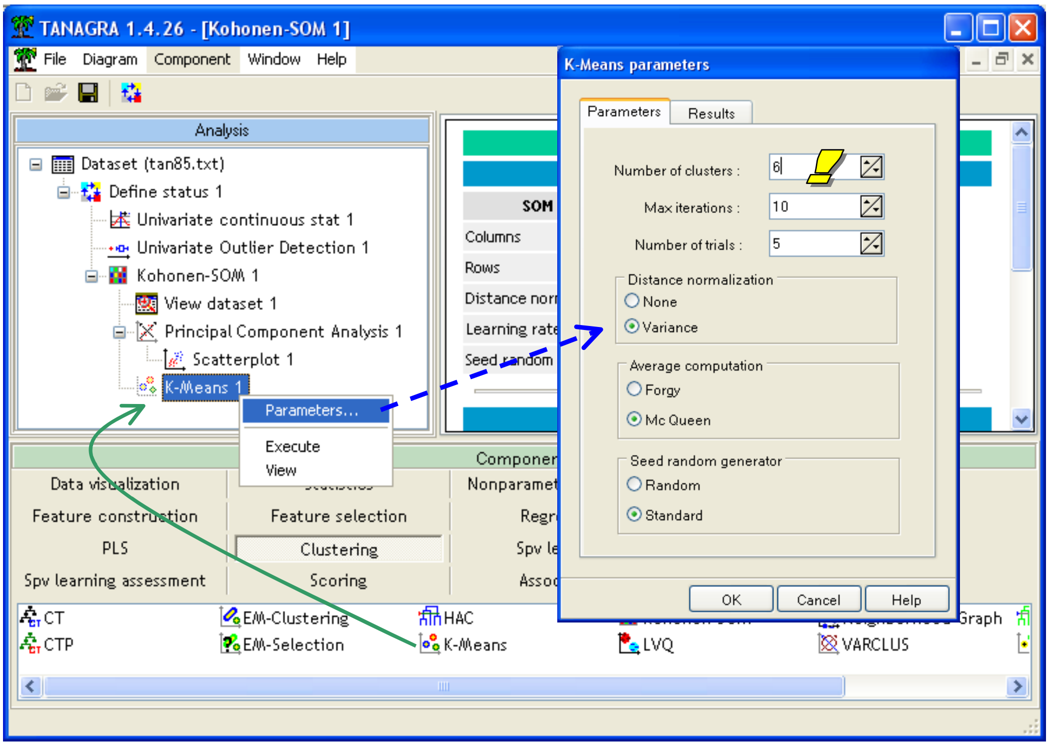
MAP Topology

	1	2	3
1	959	687	988
2	686	925	755

5 A comparison with K-MEANS

5.1 Clustering process with K-Means

K-Means is a state-of-the-art approach for clustering process (http://en.wikipedia.org/wiki/K-means_clustering). We add the component into the diagram and we ask 6 clusters. There is no constraint about the relative position of the clusters here.



TANAGRA 1.4.26 - [Kohonen-SOM 1]

Analysis

- Dataset (tan85.txt)
  - Define status 1
    - Univariate continuous stat 1
    - Univariate Outlier Detection 1
    - Kohonen-SOM 1
      - View dataset 1
      - Principal Component Analysis 1
        - Scatterplot 1
      - K-Means 1

K-Means parameters

Parameters | Results

Number of clusters : 6

Max iterations : 10

Number of trials : 5

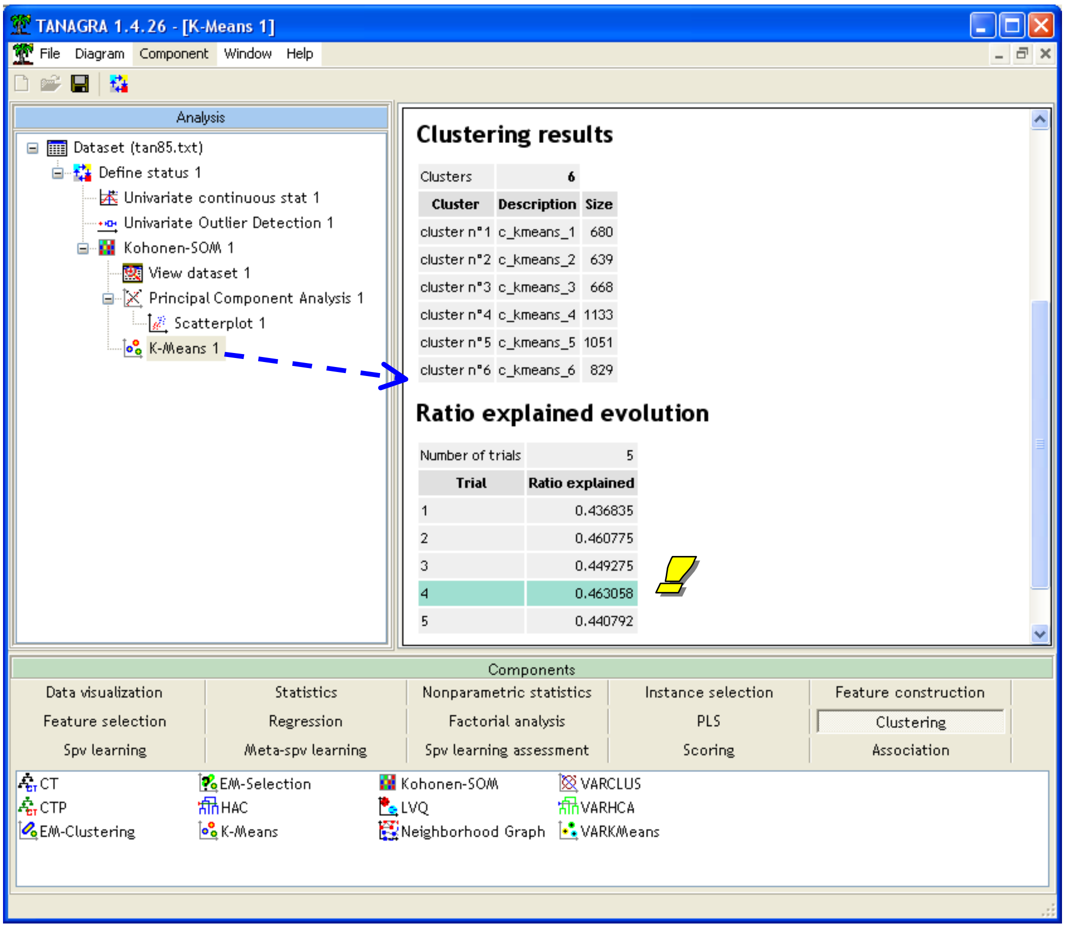
Distance normalization
○ None
● Variance

Average computation
○ Forgy
● Mc Queen

Seed random generator
○ Random
● Standard

OK | Cancel | Help

We click on the VIEW menu in order to launch the calculations.



Clustering results

Clusters		6
Cluster	**Description**	**Size**
cluster n°1	c_kmeans_1	680
cluster n°2	c_kmeans_2	639
cluster n°3	c_kmeans_3	668
cluster n°4	c_kmeans_4	1133
cluster n°5	c_kmeans_5	1051
cluster n°6	c_kmeans_6	829

Ratio explained evolution

Number of trials	5
Trial	**Ratio explained**
1	0.436835
2	0.460775
3	0.449275
4	0.463058
5	0.440792

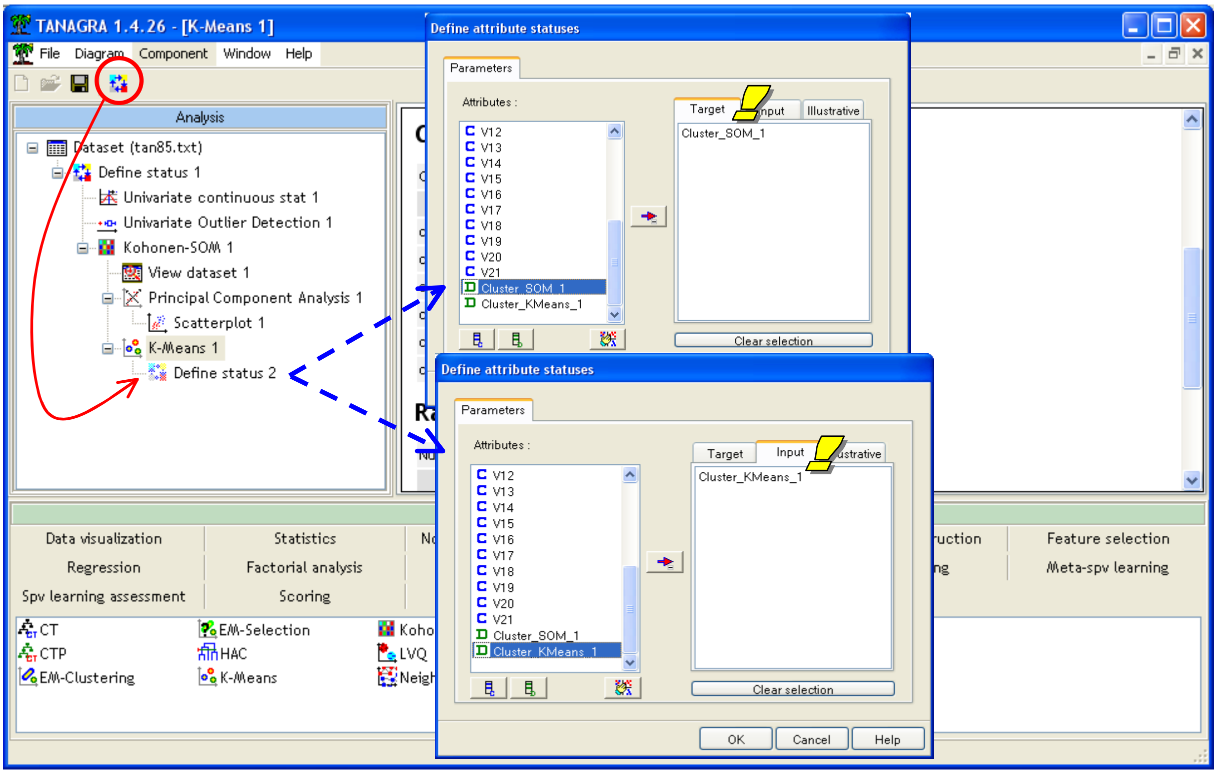
The relative part of the total sum of squares explained by the partitioning is **46.31%.** It is rather comparable to the one obtained with SOM (45.16%). But we remind that there is no constraint about the relative position of the clusters for the K-Means algorithm.

5.2 Agreement between the clusters

If the performances of these approaches seem similar, are the clusters comparable?

To check the correspondence, we create a cross-tabulation between the column memberships supplied by the two approaches.

We insert a new DEFINE STATUS component into the diagram. We set CLUSTER_SOM_1, the cluster membership column supplied by the SOM algorithm, as TARGET; we set CLUSTER_KMEANS_1, supplied by the K-MEANS algorithm, as INPUT.



Define attribute statuses

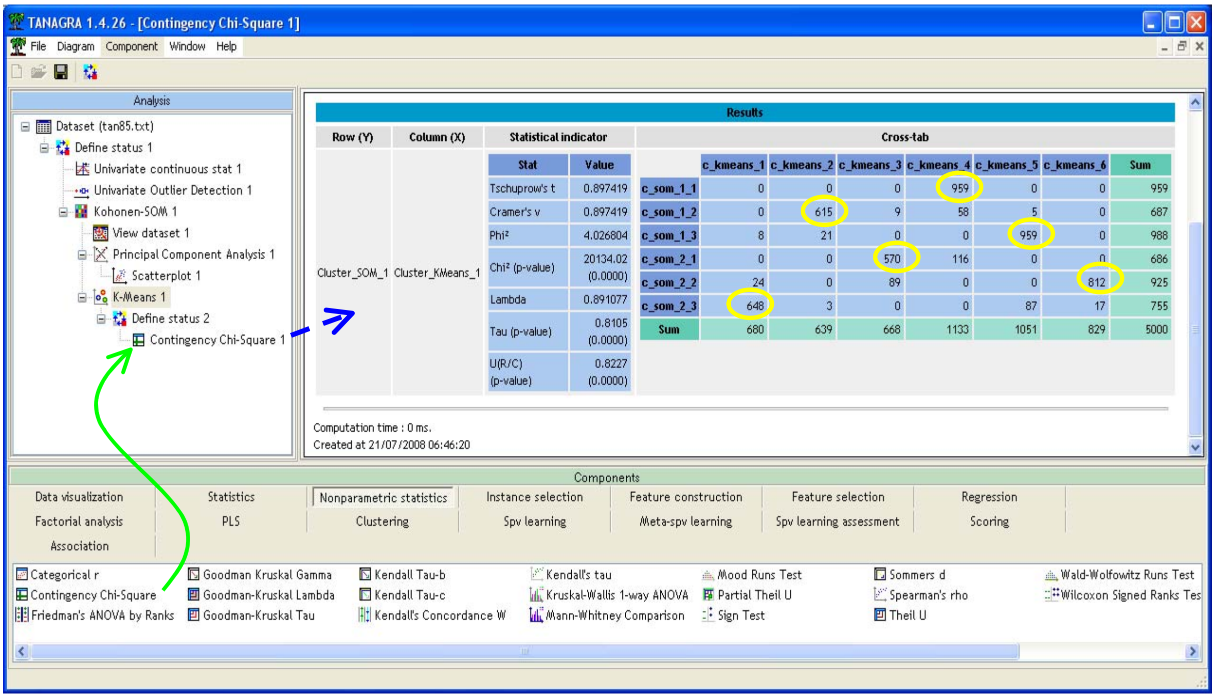
Target: Cluster_SOM_1

Define attribute statuses

Input: Cluster_KMeans_1

TANAGRA 1.4.26 - [K-Means 1]
Dataset (tan85.txt)
Define status 1
Univariate continuous stat 1
Univariate Outlier Detection 1
Kohonen-SOM 1
View dataset 1
Principal Component Analysis 1
Scatterplot 1
K-Means 1
Define status 2
OK
Cancel
Help

Then we add the cross-tabulation component (CONTINGENCY CHI-SQUARE, NONPARAMETRIC STATISTICS tab). We click on the VIEW menu.



Results

Row (Y)	Column (X)
Cluster_SOM_1	Cluster_KMeans_1

Stat	Value
Tschuprow's t	0.897419
Cramer's v	0.897419
Phi²	4.026804
Chi² (p-value)	20134.02 (0.0000)
Lambda	0.891077
Tau (p-value)	0.8105 (0.0000)
U(R/C) (p-value)	0.8227 (0.0000)

	c_kmeans_1	c_kmeans_2	c_kmeans_3	c_kmeans_4	c_kmeans_5	c_kmeans_6	Sum
c_som_1_1	0	0	0	959	0	0	959
c_som_1_2	0	615	9	58	5	0	687
c_som_1_3	8	21	0	0	959	0	988
c_som_2_1	0	0	570	116	0	0	686
c_som_2_2	24	0	89	0	0	812	925
c_som_2_3	648	3	0	0	87	17	755
Sum	680	639	668	1133	1051	829	5000

Computation time : 0 ms.
Created at 21/07/2008 06:46:20

We note that these approaches build a very similar partitioning of the instances. The correspondence is almost exact: each cluster of K-Means corresponds to one cluster of SOM.

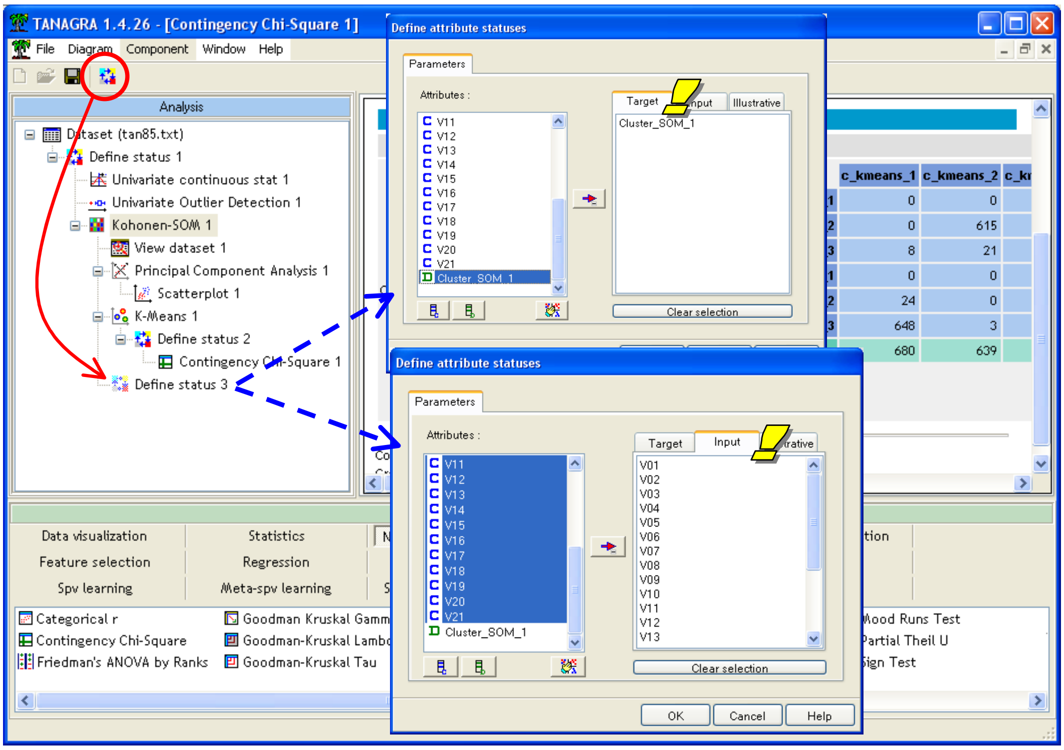
In the table below, we show the correspondence between clusters.

Cluster	Cluster SOM	Cluster K-MEANS
A	(1 ; 1)	4
B	(1 ; 2)	2
C	(1 ; 3)	5
D	(2 ; 1)	3
E	(2 ; 2)	6
F	(2 ; 3)	1

6 Two-step clustering

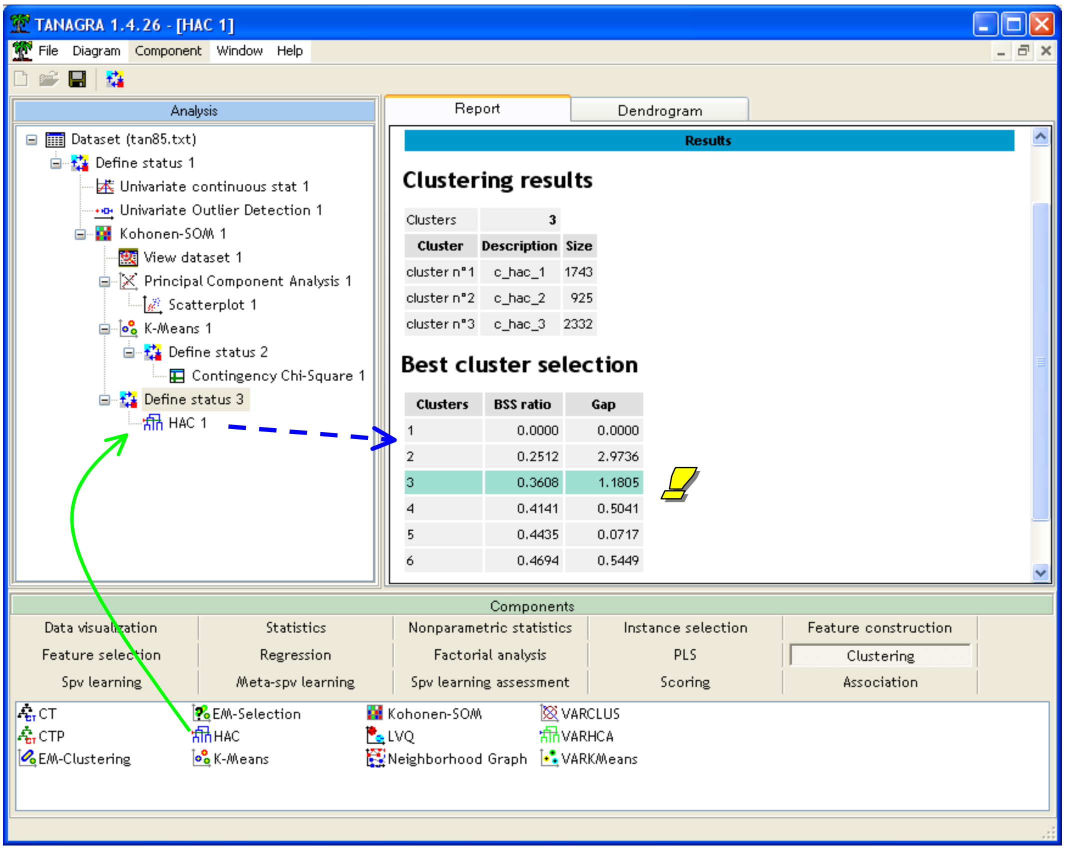
Two-step clustering creates pre-clusters, and then it clusters the pre-clusters using hierarchical methods (HCA). Two step clustering handles very large datasets[6]. The K-Means is usually used in the first phase where the pre-clusters are created. In this tutorial, instead of K-Means, we use the SOM results for this first phase. This variant involves a very interesting property: the adjacent pre-clusters correspond to nearby areas in the original representation space. This strengthens the interpretation of the dendrogram created with the subsequent HCA algorithm.

We add the DEFINE STATUS component into the diagram. We set CLUSTER_SOM_1 as TARGET, the descriptive variables (V1...V21) as INPUT.



We add the HAC component (CLUSTERING tab).

[6] http://faculty.chass.ncsu.edu/garson/PA765/cluster.htm ; the implementation of this approach with Tanagra is described in various tutorials e.g. http://data-mining-tutorials.blogspot.com/2009/06/two-step-clustering-for-handling-large.html



Clustering results

Clusters	3

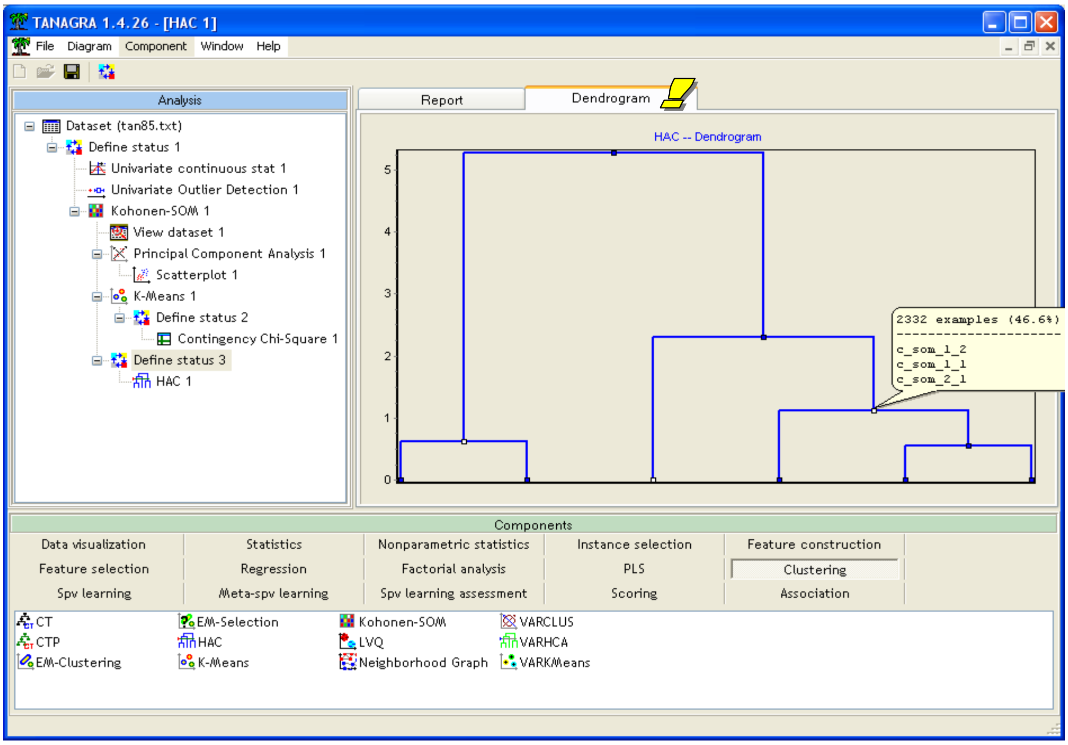
Cluster	Description	Size
cluster n°1	c_hac_1	1743
cluster n°2	c_hac_2	925
cluster n°3	c_hac_3	2332

Best cluster selection

Clusters	BSS ratio	Gap
1	0.0000	0.0000
2	0.2512	2.9736
3	0.3608	1.1805
4	0.4141	0.5041
5	0.4435	0.0717
6	0.4694	0.5449

The component automatically detects 3 groups. This choice relies on the height between each merging. There is no theoretical justification here.

The DENDROGRAM tab of the visualization window is very important. By clicking on each node of the tree, we obtain the ID of the pre-clusters supplied by the SOM algorithm.



```
2332 examples (46.6%)
-----------------------
c_som_1_2
c_som_1_1
c_som_2_1
```

The white nodes of the tree states the groups computed with the HAC algorithm. If we select the white node at right, we obtain the SOM's pre-clusters ID i.e. the individuals in this group come from the pre-clusters (1 ; 1), (1 ; 2) et (2 ; 1).

In the table, we see the correspondence between SOM pre-clusters ID and the HAC clusters ID.

Cluster CAH	Cluster SOM
1	(1 ; 3) + (2 ; 3)
2	(2 ; 2)
3	(1 ; 1) + (1 ; 2) + (2 ; 1)

The results are strikingly consistent with the theoretical consideration underlying the SOM approach: the HAC above all merges the adjacent cells of the Kohonen's map.