Subject

We show how to induce clustering trees with TANAGRA.

The aim of clustering is to build groups of individuals so that, the examples in the same group are similar, the examples in different groups are dissimilar.

Top down induction of clustering trees adapts the supervised decision/regression trees framework towards clustering. The groups are built by recursive partitioning of the dataset, the internal nodes of the tree are classically split with input attributes. The obtained model, the clustering tree, describes the groups; the learning algorithm selects automatically the relevant attributes.

The clustering trees approach is not very known; we show in this tutorial the interesting properties of this method. Our main references are the papers of Chavent[1] (1998) and Blockeel[2] (1998).

Dataset

We use the ZOO dataset (UCI). We want to group animals using their characteristics such as number of legs, producing milk, …

The expert domain proposes 7 clusters. We want to know (1) if our algorithm can find these clusters; (2) if we find the same clusters as the well-known K-MEANS algorithm.

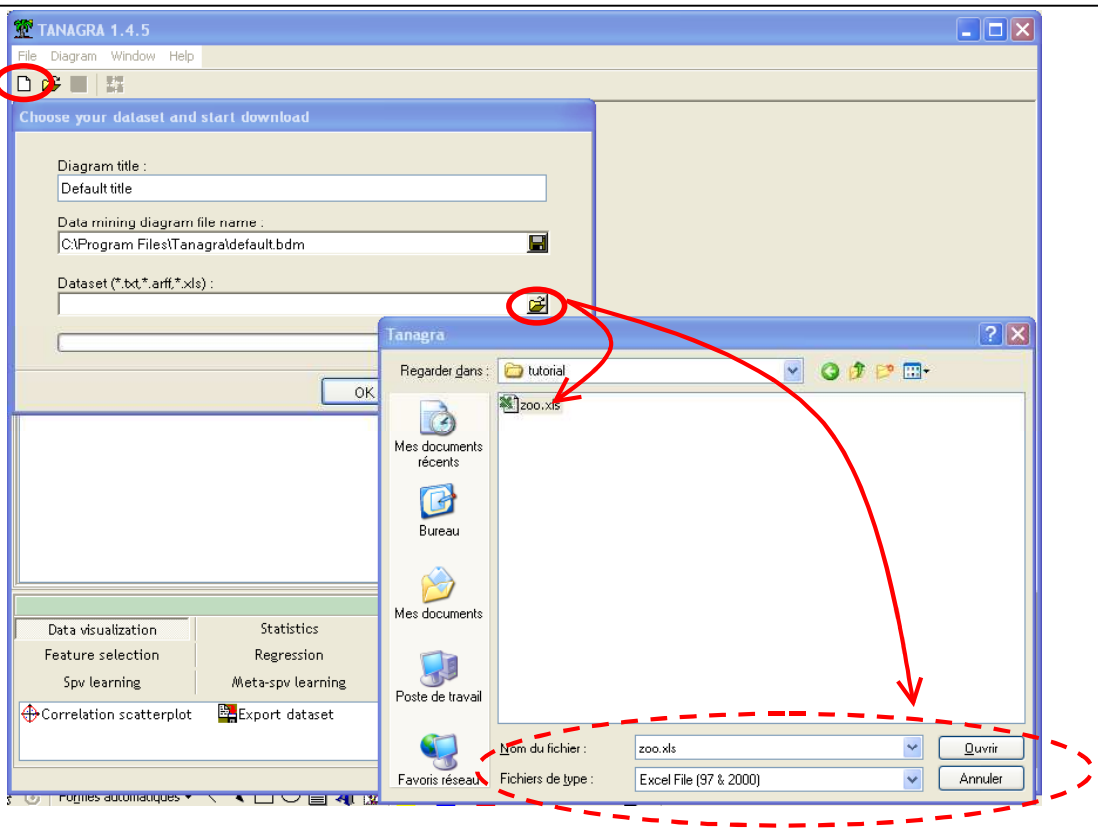
Clustering trees

Downloading the dataset

In the first time, we must create a diagram and import ZOO.XLS. We click on the FILE/NEW menu.

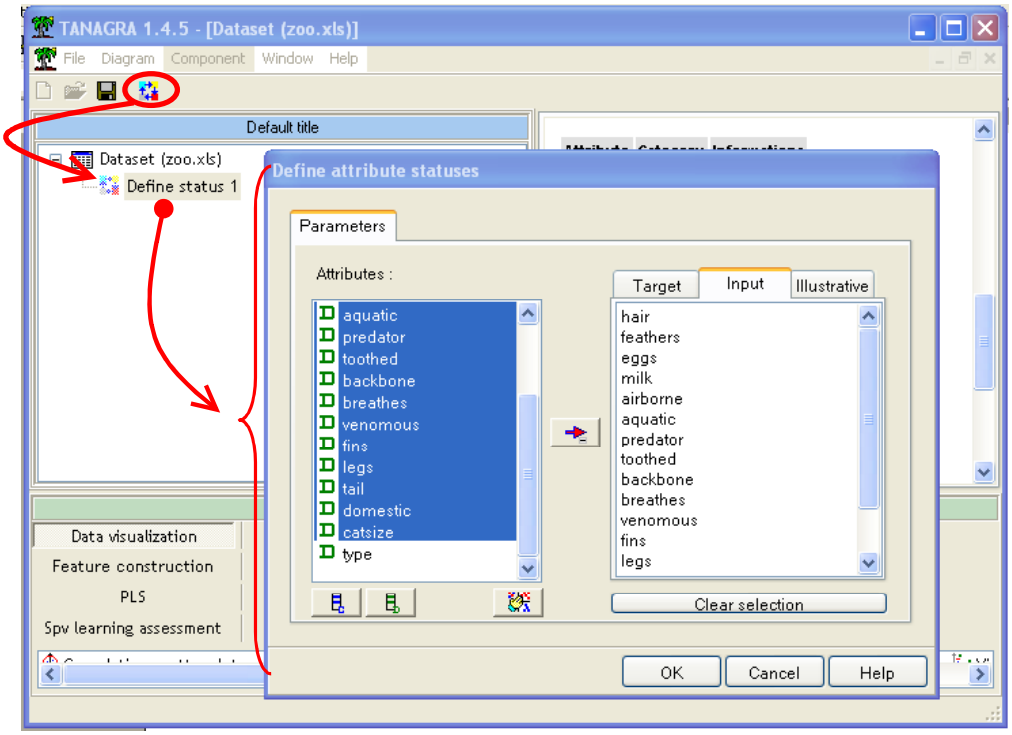
[1] M. Chavent (1998), « A monothetic clustering method », Pattern Recognition Letters, 19, 989—996.

[2] H. Blockeel, L. De Raedt, J. Ramon (1998), « Top-Down Induction of Clustering rees », ICML, 55—63.



Selecting the attributes

In the next step, we select the attributes that we use in order to characterize the homogeneity of groups. We choice all the measured attributes; we do not use the TYPE attribute, which is provided by experts. We use the DEFINE STATUS component.



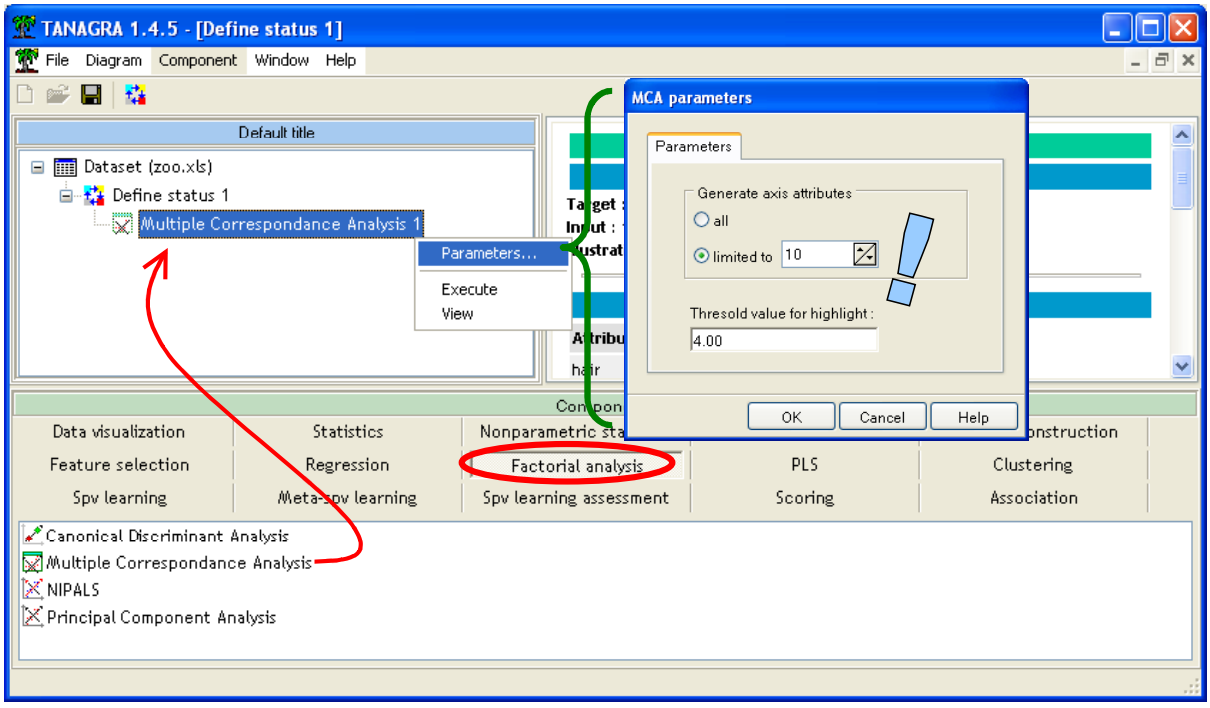
Feature construction

Computing a distance on discrete attributes is possible but not easy. Moreover, some attributes may be redundant. We use factorial analysis in order to build a new representation space where we respect, as much as possible, the proximity between the individuals.

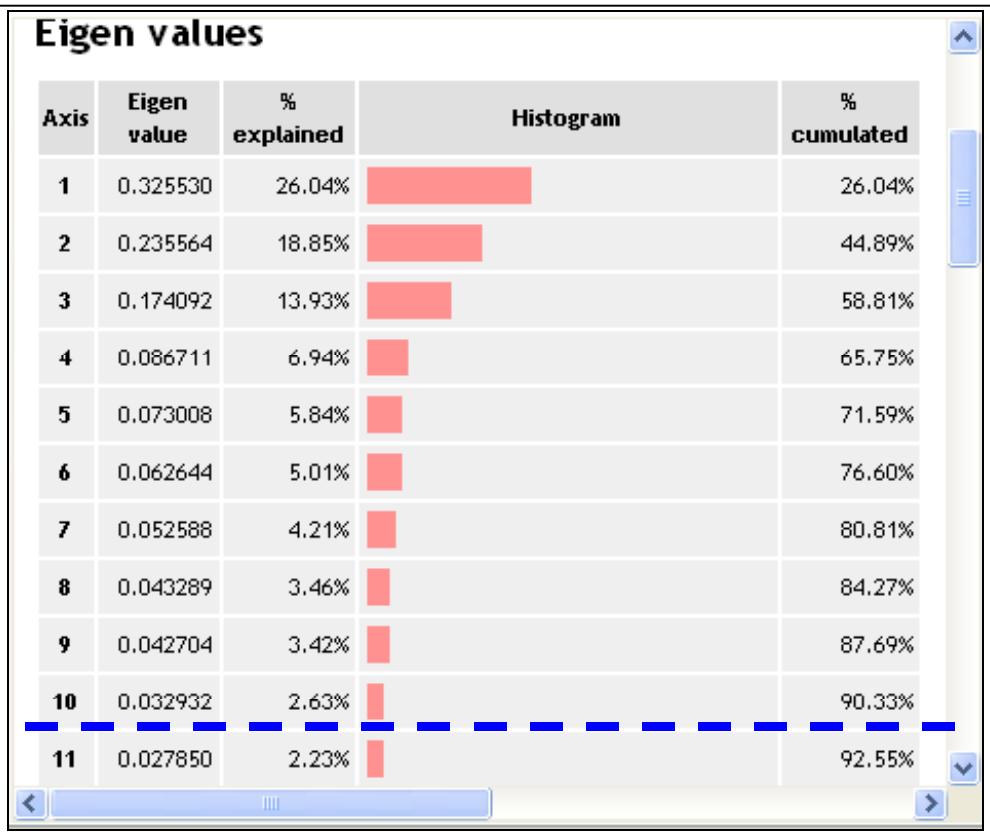
Because we have discrete attributes, we use multiple correspondence analysis (MCA). This data transformation cumulates several advantages: we can use now classical Euclidian distance, more especially as the factorial axes (the latent variables) are independent; by selecting only the first 10 axes, we recover "useful" information and leave side "disturbed" information specific to the file (the artifact information in the dataset).

We add a MCA component in the diagram, we set 10 the number of produced axis (approximately the half of the total number of axis).

Note: In the case of continuous attributes, we follow the same principle and use instead a principal component analysis (PCA). We observe the same advantages.



We click on the VIEW contextual menu. The 10 axis summaries 90% of available information, that is fully suitable.

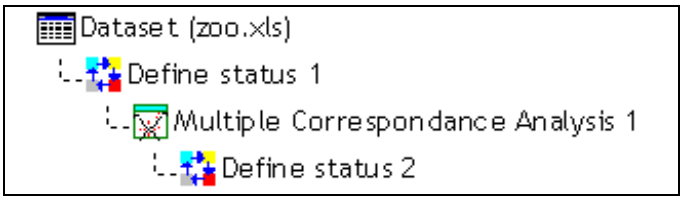


Eigen values

Axis	Eigen value	% explained	Histogram	% cumulated
1	0.325530	26.04%		26.04%
2	0.235564	18.85%		44.89%
3	0.174092	13.93%		58.81%
4	0.086711	6.94%		65.75%
5	0.073008	5.84%		71.59%
6	0.062644	5.01%		76.60%
7	0.052588	4.21%		80.81%
8	0.043289	3.46%		84.27%
9	0.042704	3.42%		87.69%
10	0.032932	2.63%		90.33%
11	0.027850	2.23%		92.55%

Target and input attributes for clustering tree

In order to build groups, we want split the dataset using original attributes (INPUT); the homogeneity of groups is computed on factorial axis (TARGET). We add a DEFINE STATUS component in the diagram and set these parameters.



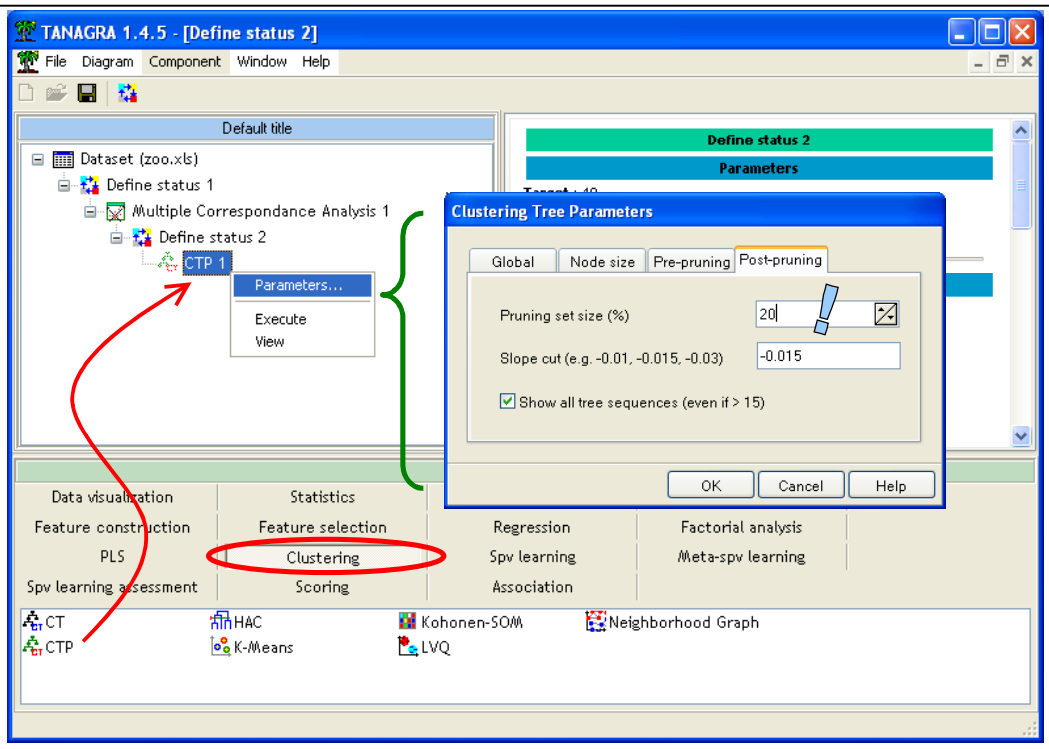
We obtain the following results (VIEW menu).

Attribute	Target	Input	Illustrative
hair	-	yes	-
feathers	-	yes	-
eggs	-	yes	-
milk	-	yes	-
airborne	-	yes	-
aquatic	-	yes	-
predator	-	yes	-
toothed	-	yes	-
backbone	-	yes	-
breathes	-	yes	-
venomous	-	yes	-
fins	-	yes	-
legs	-	yes	-
tail	-	yes	-
domestic	-	yes	-
catsize	-	yes	-
type	-	-	-
MCA_1_Axis_1	yes	-	-
MCA_1_Axis_2	yes	-	-
MCA_1_Axis_3	yes	-	-
MCA_1_Axis_4	yes	-	-
MCA_1_Axis_5	yes	-	-
MCA_1_Axis_6	yes	-	-
MCA_1_Axis_7	yes	-	-
MCA_1_Axis_8	yes	-	-
MCA_1_Axis_9	yes	-	-
MCA_1_Axis_10	yes	-	-

Note: In this tutorial, we use the same attributes for the homogeneity computation and the construction of the tree. But, in fact, we can use two separate sets of attributes. We obtain a generalization of decision/regression trees; some authors call this approach "multi-objective regression/decision trees" or "predictive clustering trees".

Clustering trees

We add the clustering tree component in the diagram (CTP -- CLUSTERING TREE WITH PRUNING).



Roughly speaking, it is a generalization of CART algorithm (Breiman et al, 1984) with two specificities:

1. We compute inertia instead of variance to evaluate homogeneity of groups.
2. Our goal is not to produce an accurate prediction but find "natural" groups. So, we try to detect the "angle" of the within-inertia computed on the pruning set. At the present time, we use a regression on successive 3 points. We select the cut point that corresponds to a slope of the lines near to zero.

In this tutorial, we use 20% of the dataset as pruning set; 80% of examples are used for the growing phase. We obtain the following clustering tree (VIEW menu).

Tree description

Number of nodes	7
Number of leaves	4

Decision tree

- milk in [true] then **cluster n°1**, with 33 examples (41.25%)
- milk in [false]
  - feathers in [false]
    - backbone in [true] then **cluster n°2**, with 18 examples (22.50%)
    - backbone in [false] then **cluster n°3**, with 13 examples (16.25%)
  - feathers in [true] then **cluster n°4**, with 16 examples (20.00%)

Computation time : 78 ms.
Created at 02/05/2006 16:39:16

We obtain 4 groups (the leaves of the tree), each cluster corresponds to the following rule:

If milk = true Then Cluster 1
If milk = false And feathers = false And backbone = true Then Cluster 2
If milk = false And feathers = false And backbone = false Then Cluster 3
If milk = false And feathers = true Then Cluster 4

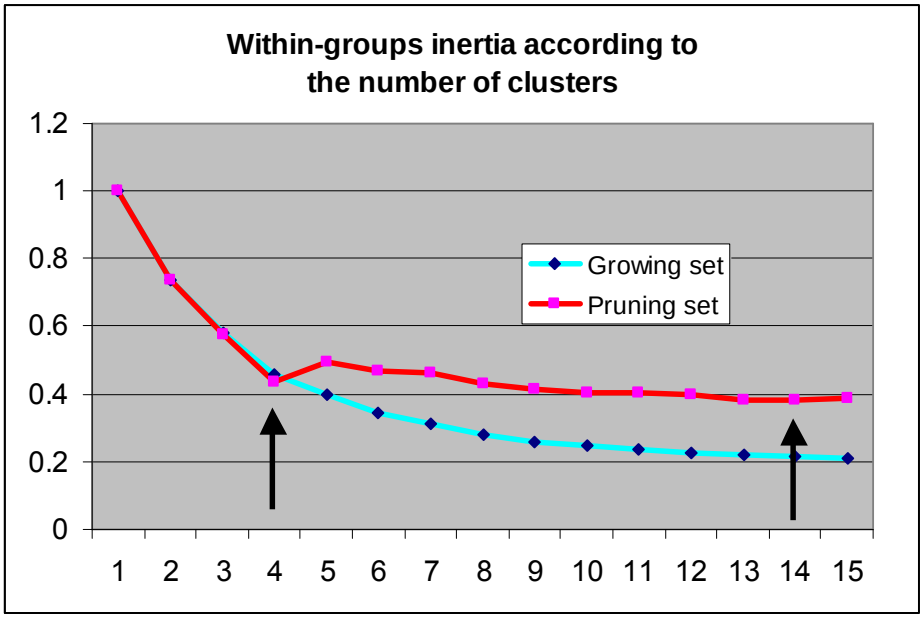
It is very easy to assign a group to a new example with theses rules.

We can see also the decrease of the within-class inertia according to the number of the leaves (groups), on the growing and the pruning set.

Trees sequence (# 15) -- Inertia Within-Groups

N°	# Leaves	Inertia (growing set)	Inertia (pruning set)
15	1	1.0000	1.0000
14	2	0.7389	0.7378
13	3	0.5809	0.5769
12	4	0.4564	0.4337
11	5	0.3992	0.4938
10	6	0.3470	0.4696
9	7	0.3104	0.4625
8	8	0.2791	0.4290
7	9	0.2598	0.4128
6	10	0.2453	0.4032
5	11	0.2348	0.4027
4	12	0.2274	0.3982
3	13	0.2203	0.3826
2	14	0.2134	0.3809
1	15	0.2082	0.3862

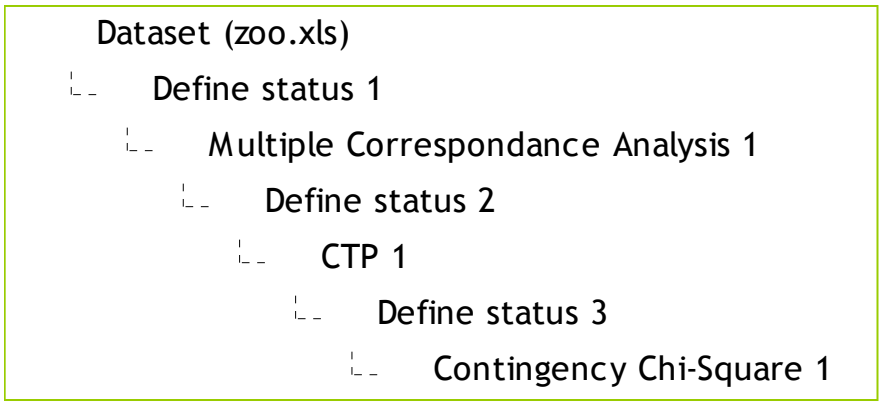
The 14 groups clustering minimizes the within inertia on the pruning set (green mark). But we see an "angle" when we have 4 groups (red mark). The following chart shows the variation of the within inertia.



Comparison with the classification of the domain expert

The experts suggest 7 groups. Our aim is to compare our 4 groups clustering with this classification. It is a good indicator of the relevance of our results.

We add a DEFINE STATUS component in the diagram. We set TYPE as TARGET and our clustering suggestion (CLUSTER_CTP_1) as INPUT. Then we add a CONTINGENCY CHI-SQUARE (NON PARAMETRIC STATISTICS tab) in order to compare the groups.



We note that we have very similar groups.

Contingency Chi-Square 1

Parameters

Cross-tab parameters	
Sort results	non
Input list	Target (Row) and input (Column)
Contribution thresold	2.0

Results

Row (Y)	Column (X)	Statistical indicator		Cross-tab					
		Stat	Value		c_ct_1	c_ct_2	c_ct_3	c_ct_4	Sum
type	Cluster_CTP_1	Tschuprow's t	0.840896	mammal	41 (+0.12)	0	0	0	41
		Cramer's v	1.000000	fish	0	13 (+0.12)	0	0	13
		Phi²	3.000000	bird	0	0	0	20 (+0.21)	20
		Chi²	303.000000	invertebrate	0	0	10 (+0.13)	0	10
		Pr(Chi²)	0.000000	insect	0	0	8 (+0.10)	0	8
				amphibian	0	4	0	0	4
				reptile	0	5	0	0	5
				Sum	41	22	18	20	101

Computation time : 0 ms.

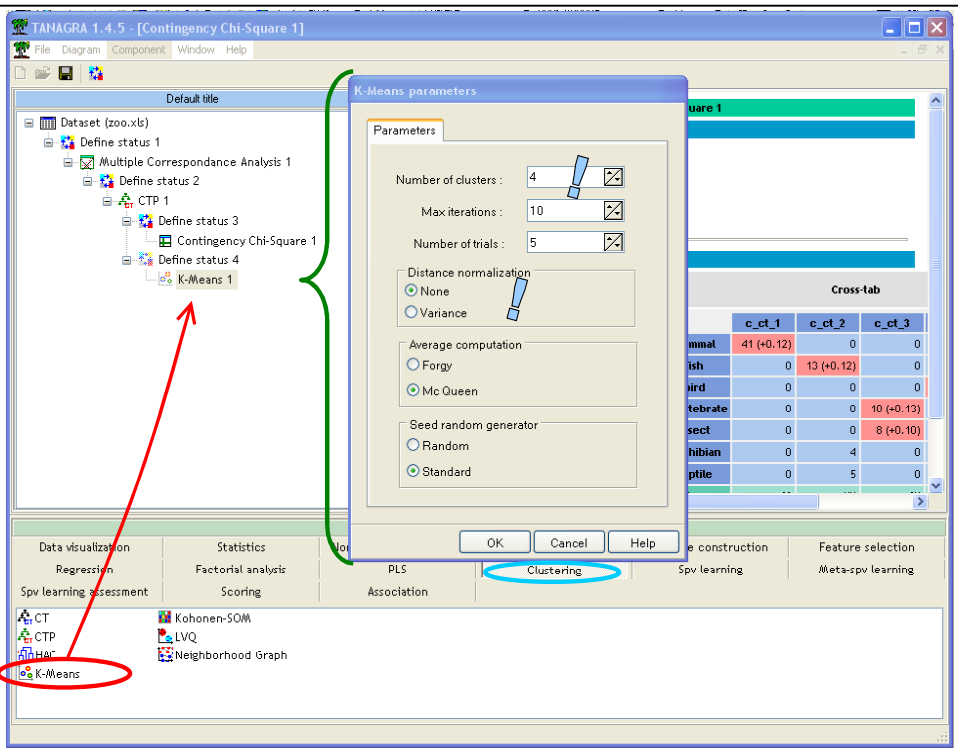
Each expert group is set in one cluster. And each cluster is a pure group (Cluster 1 and Cluster 4) or a mix of similar species (Cluster 2 and Cluster 3)[3].

Comparison with K-MEANS clustering algorithm

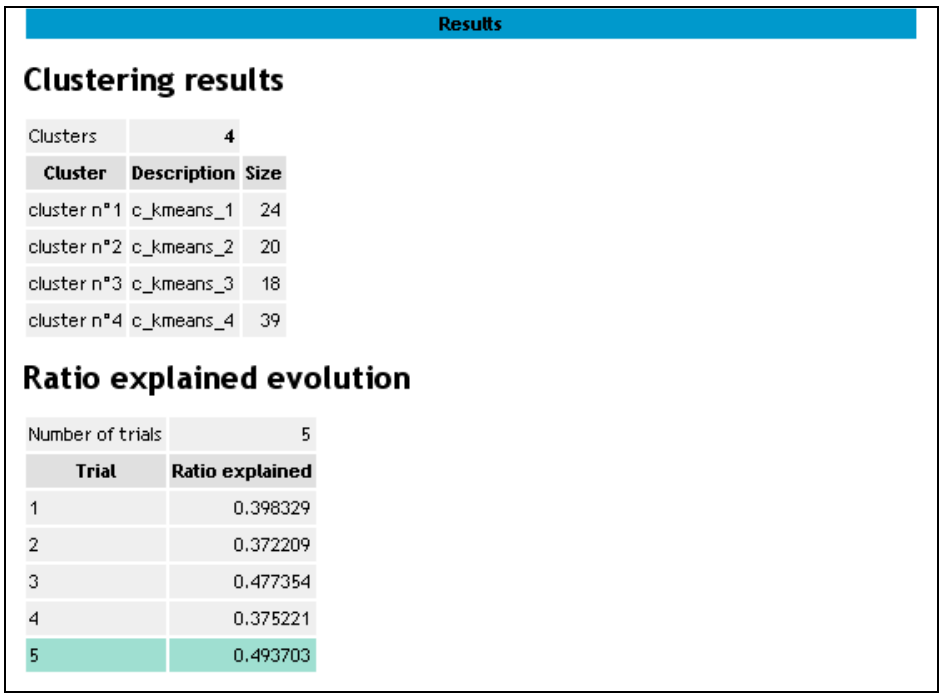
The learning and representation bias of the clustering trees can lead to not very effective solutions compared to well-known methods such as K-MEANS. In this next step, we compare the groups of CTP with the groups produced by K-MEANS.

We insert again a DEFINE STATUS component under the CTP (Clustering Tree) component. We set as INPUT the factorial axis. We add the K-MEANS component that is configured so that the results of the two approaches (tree and k-means) are comparable: we want 4 groups; we must not normalize the factorial axis in the inertia computation.

[3] I am not an expert !



We obtain the following results.



Results

Clustering results

Clusters	4	
Cluster	**Description**	**Size**
cluster n°1	c_kmeans_1	24
cluster n°2	c_kmeans_2	20
cluster n°3	c_kmeans_3	18
cluster n°4	c_kmeans_4	39

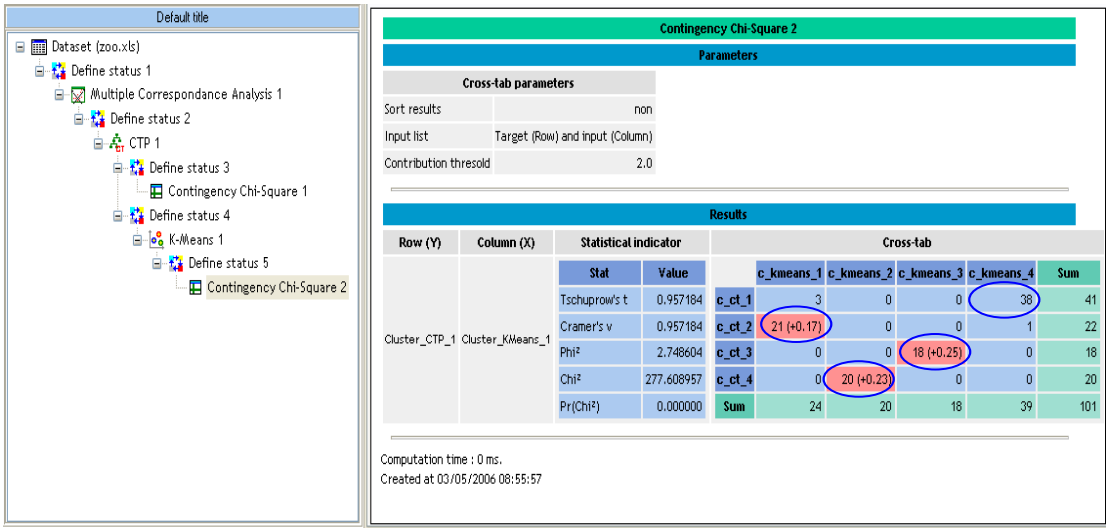
Ratio explained evolution

Number of trials	5
Trial	**Ratio explained**
1	0.398329
2	0.372209
3	0.477354
4	0.375221
5	0.493703

We want to compare these groups with the groups obtained with CTP.

We insert another DEFINE STATUS in the diagram; we set as TARGET the clusters of the tree (CLUSTER_CTP_1), as INPUT the clusters of the K-MEANS (CLUSTER_KMEANS_1).

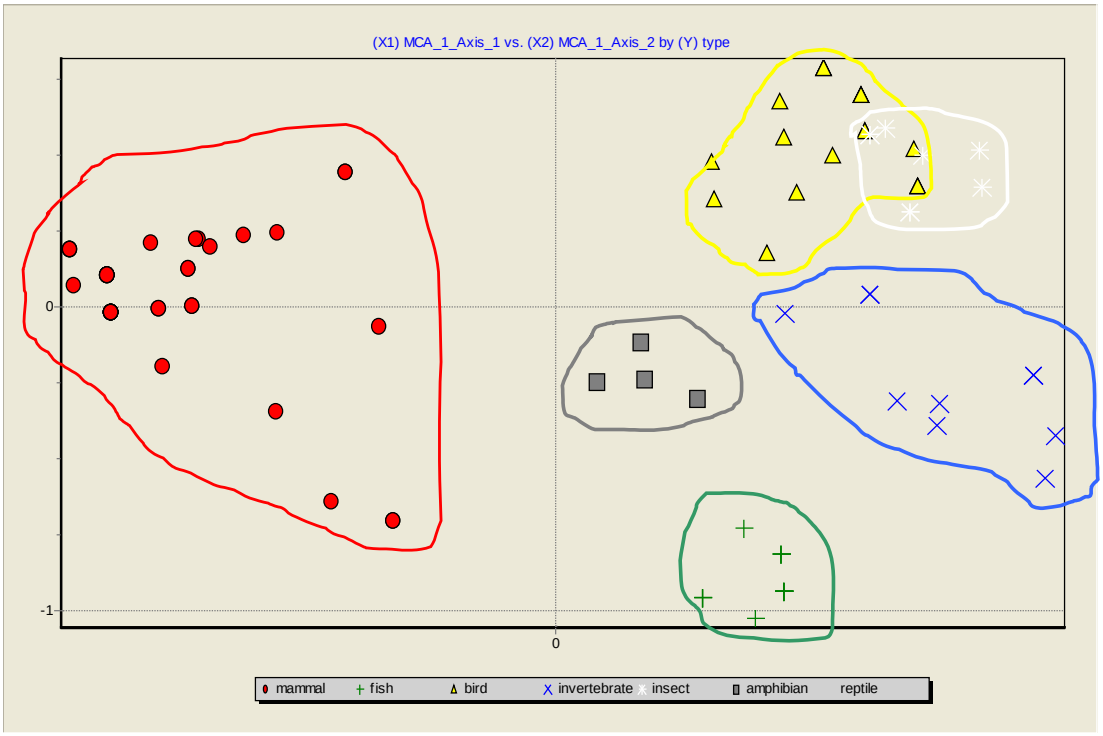
So we add again the contingency table component in order to comparing the two approaches.



The two methods are equivalent; the profit of interpretability of the trees is not counterbalanced by a degradation of the precision of calculations. The other advantage of the tree in this case is that it selected the relevant variables automatically.

Visualization of groups

Factorial analysis allows us to visualize the dataset in a reduced dimension space. We want to see if we can perceive the expert groups in the first two "latent" variables.



This result is edifying. The groups proposed by experts are really distinct even on the first two axes which summaries only about 50% of the available information (see MCA result, 44.89%).

If this example shows well that the visual tools are often very powerful; the main difficulty is to be able to be came back thereafter to the initial space of description and obtain an interpretable results in relation to these descriptors. The reading of the results of the MCA remains obscure for the people who are not accustomed.

The clustering trees approach is a simple method to build automatically clusters and obtain interpretable results.