



a Bitlab software

Association Rules collaborative tool

Integrated suite for association rule discovering in medical and molecular data



Version v1: 8th November 2007.
On-line updated information available at:
<http://chirimoyo.ac.uma.es/arco>

Developed by:
Jesús Jiménez Espada
Javier Ríos
Andrés Rodríguez
Oswaldo Trelles

Report incidences to:
ots@ac.uma.es



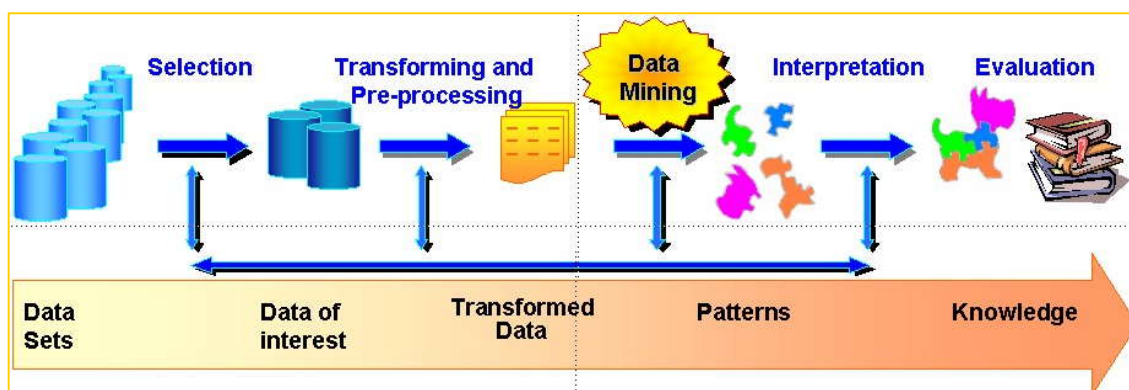
a Bitlab software

Association Rules collaborative tool

Integrated suite for association rule discovering in medical and molecular data

Welcome to **Arco**, a versatile, complete and powerful suite for the production of association rules with particular focus on gene-expression data combining both gene related descriptors and sample metadata. This document concerns with some general but important terminology that must be mastered to fully understand the **ARco** application and follow-up technical and training documents.

ARco (stands for **A**ssociation **R**ules **c**ollaborative tool) integrates the typical steps on KDD procedures. The KDD process involves different steps, from the selection of appropriated data, coding the data and identifying patterns. The association rule is an expression of the discovered knowledge. Next Picture depicts the procedure.



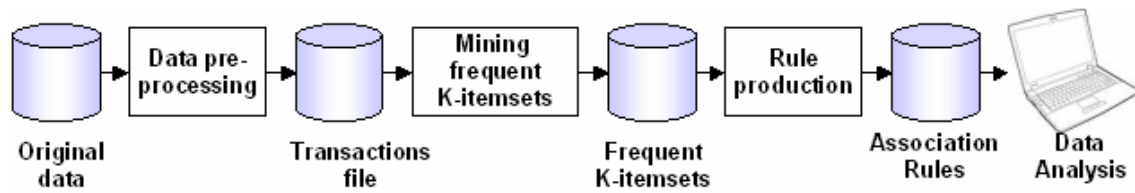
First step of KDD process deals with original data that must be appropriated selected, filtered and transformed into a transaction file. Each transaction on this file is an experimental observation consisting of a set of items (e.g., items that appears together in the same shopping; or in the same cash-register ticket, values in a row of a gene-expression file; or a set of keywords belonging to particular proteins, etc). In this step prior knowledge of the application domain allows cleaning and pre-processing the data set by removing or filling incomplete data, or by data reduction and transformation using the main item features, or applying dimensionality or variable reduction, invariant representation; etc

A second step is focused on finding frequent itemsets, this is to say, item collections that appear together in different transactions more frequent than a given minimal support threshold. This step fits well with a typical pattern finding procedure.

Then in a third step frequent itemsets are combined to produce a set of association rules whose antecedents imply the consequent with a given probability (confidence)

Rules are analysed in the last step by the expert by using browsing and filtering procedures. The new knowledge is normally used to fine-tuning parameters and selection criterions in the iterative KDD process.

ARco architecture has been designed to resemble the KDD process. In fact, the main modules of ARco are depicted in the next picture:



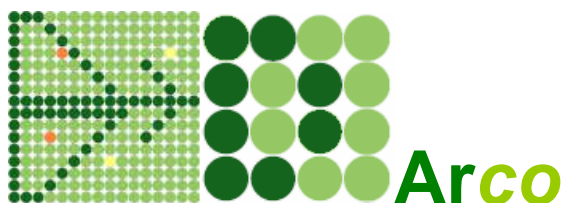
- Data pre-processing: including data transformation, selection and filtering. Visual tools accomplish the tasks to facilitate identification of main features of data. This step ends with the production of a transaction dataset, including dictionaries and complementary information internally used by ARco to interpret the data.
- Mining frequent itemsets. This step aims to identify items that frequently appear together in the same transaction. New settings such as specific support thresholds by item-type are available in ARco.
- Rules production. From the frequent item-sets ARco produce rules. User can specify where the items should be placed (antecedent, consequent, both, or none), and minimal confidence thresholds.
- Data analysis consists in rules exploration, browsing and visualization (rule profile, transactions that hold the rule, statistical information, etc). Diverse filters have been implemented: improvement, confidence, information gain, redundant rules, trivial non-informative rules, etc.

ARco has been prepared especially for gene-expression data in which the following information can be used for ruling:

- Gene expression-ratios (in several experiments or samples)
- Gene metadata: such as function, pathways, GO-terms, etc
- Sample metadata such as (depending on the experiment type) clinical information on the patient

Not only traditional and exhaustively tested algorithms have been implemented in ARco, but also novel ideas in the field such as specific item-type thresholds, L-transformation, etc. that enable the analysis –in particular- of gene expression data in an easy and transparent manner. Free access to this tool is available upon request.

ARco provides an integrated environment to perform data transformations to produce a set of transactions over which apply mining procedures for finding frequent itemsets and produce rules to be explored.



a Bitlab software

Association Rules collaborative tool

Integrated suite for association rule discovering in medical and molecular data

Introduction to association rule discovering

What an association rule is?



Knowledge Discovery from Databases (KDD) is a discipline at the borders of computer science, artificial intelligence and statistics. Roughly speaking, its goal is to find something interesting in given data. One part of data mining concentrates on finding potentially useful knowledge in the form of association rules.

An association rule is a mathematical formula expressing some relationship that -very probably- holds in data. The result of association rules mining process serves mostly as a tool for understanding the data character.

The following is a formal statement of the problem [*]: Let $I = \{i_1; i_2; \dots; i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subset I$. associated with each transaction is a unique identifier, called its TID.

We say that a transaction T contains X , a set of some items in I , if $X \subset T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$.

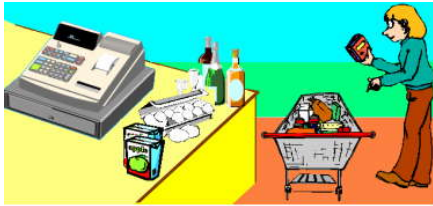
The rule $X \Rightarrow Y$ holds in the transaction set D with confidence c if $c\%$ of transactions in D that contain X also contain Y . The rule $X \Rightarrow Y$ has support s in the transaction set D if $s\%$ of transactions in D contains $X \cup Y$. Our rules allow a consequent to have more than one item.

Given a set of transactions D , the problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence respectively.

[*] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., May 1993.

My first example:

Let's see a basic example of association rules in its original market-basket context.



Code	item
B	Bread
C	Coffee
K	Cookies
M	Milk
S	Sugar

Transactions DB	
Transaction	items
1	BCMS
2	KMS
3	BCKM
4	BCK
5	CKS
6	BCKS





Let's imagine a typical retail store in which each client goes through the cash-register point where the set of items bought together by this customer are stored in a transactional database (see the cartoon on the left)

As it is illustrated in the picture each customer produces one transaction as is represented in the "Transactions DB" where each letter in {BCMS} represents Bread, Coffee, Milk and Sugar respectively (see the Code-Item table).

In general, supermarkets use different strategies such as saving prices to get additional information about the customer such as address, economical level, professional skills, family details, etc). This information regarding the customer is known as the "customer metadata".

Normally there also exists some information regarding the "products" or items. Also depending on the application domain the metadata can refer to:

Market application: supplier; content of given supplements; item classification; etc.
Tissue samples: provenance; gender; familiar record; age; etc.

#	Bread (B)	Coffee (C)	Cookies (K)	Milk (M)	Sugar (S)
					
1	4	2		1	2
2			3	2	2
3	3	1	2	3	
4	3	2	4		
5		5	4		2
6	3	2	1		2

On the left there is another representation of the transaction database. As it has been mentioned, the set of transactions describing the set of items bought together by each customer can also contain metadata information about both clients (rows) and products or items (columns).

1-itemset		3-itemset	
B	4	BCK	3
C	5	BCM	2
K	5	BCS	2
M	3	BKM	1
S	4	BKS	1
2-itemset		4-itemset	
BC	4	BMS	1
BK	3	CKM	1
BM	2	CKS	2
BS	2	CMS	1
CK	4	KMS	1
CM	2	BCKM	
CS	3	BCKS	1
KM	2	BCMS	1
KS	3	BKMS	
MS	2	CKMS	
		5-itemset	
		BCKMS	

All this information can be processed to determine which products are frequently bought together. A counting procedure is needed to obtain 'the number of times a given combination of items is located in the same transaction'.

For this short number of items (and short number of transactions) it is still possible to count manually as is shown in the picture. In this case, for N = 5 items, there are $2^N - 1 = 31$ different combinations.

From this information we can deduce that the items B, C and K (BCK) are bought together for the 50% of the customers (frequency 3 on 6 transactions). Even more, we can compute how many times, when a customer bought BC also bought K. Since BC=4 times; BCK=3 times, we deduce a confidence of 75%.

These are the basic concepts on mining frequent itemsets (items that appear together more frequently than a given cut-off). Of course, for high number of items the number of possible combinations grows exponentially (i.e. for N=32 items there are more than 4 millions combinations); and mining is especially interesting for N high (i.e. for 20000 genes).

To reduce the computational space Agrawal (et.al, 1993) introduced main improvements in the process of mining frequent k-itemsets. The key idea of Agrawal approach is that "any k-frequent itemset can contain any infrequent sub-itemset"

In general, all mining procedures starts for counting all items with cardinality k=1 and determining the frequent "1_itemsets" (individual items with support greater than a given threshold). From the second iteration (k=2) the subset L[k-1] is used to produce the *a priori* candidate subset C[k] and the database is scanned again to explore each transaction T_i , and accounting the frequency of each candidate. The strength of the approach lies on the generation of the candidate set C[k]. For the k-th iteration the L[k-1] frequent itemsets are combined in the following way:

$$C[k] = \{ X = \{x_1, x_2, \dots, x_{k-2}, x_{k-1}, x_k\} \mid \exists A, B \in L[k-1], \\ A = \{x_1, x_2, \dots, x_{k-2}, x_{k-1}\}, B = \{x_1, x_2, \dots, x_{k-2}, x_k\}, x_{k-1} < x_k \}$$

For example if we set 4 minimum support, in the first step the support of M (milk) is 3, so, it will be no used to produce a combination in the next step. In other words, if the itemset BCK has a support of 4 transactions, any subset (BC, BK, CK, B, C, K) must hold the minimum support. This strategy allows pruning several putative combinations, reducing the computational space.

Once the frequent itemsets have been identified, in a second step, these itemsets are broken into association rules that hold the minimum confidence threshold.

Note: in this example transactions database is formed by binary data for five items mean they where bought or not, in a transaction (not information about quantities is processed).

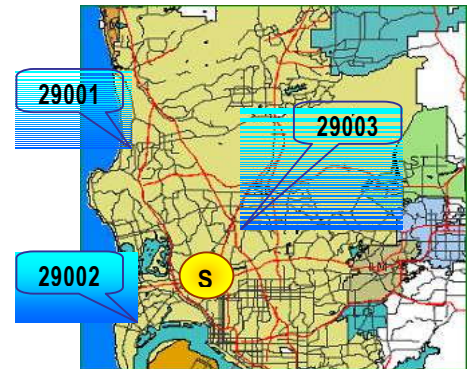
My first example using ARco

Now, let's repeat and extend the example using ARco. To this end we are going to use the he marketDemo100.xls file available at <http://chirimoyo.ac.uma.es/arco/marketDemo100.xls>. See "Annexe 1" for more information about file formats supported by ARco.

As it has been presented in the previous section this exercise simulates the case of a short basket data from an imaginary retail store. This intuitive collection of data will be used to introduce the tool and some concepts. Let's describe the content of the data file and some characteristics of the data.

Each record in this file contains information about the number of each of the 5 different items {Beer, Juice, Champagne, HighQ-Wine, Common-Wine} purchased by a customer, together with metadata information about the customer:

The customer metadata is formed by (see picture on the right):
 Zip code: [29001 high; 29002 medium and 29003 low quality]
 Familiar incoming level: A (low) –D (high)
 Marital status: {Single, Married}



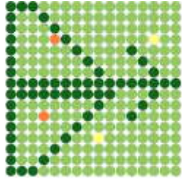
Some metadata about the items have been also coded for each column: supplier, price range and restrictions

Note that other metadata could be included both, at client (i.e. gender) or item level (article price, category: {fruits, vegetables, meat, etc.}) and a few others).

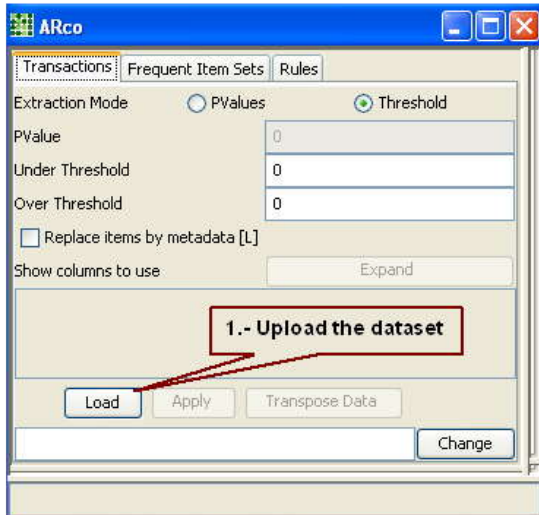
The marketData100.xls file looks as follow:

	A	B	C	D	E	F	G	H	I
3				Restrictions	Yes	Not	Yes	Yes	Yes
4	TID	ZIP code	Level(ABCD)	Civilstage	beer	juice	champagne	High Q wine	wine X
5	1	29003	A	Single	4	1			3
6	2	29003	A	Single	5	1			2
7	3	29003	A	Single	7	5			4
8	4	29003	A	Single	3	4			1
9	5	29003	B	Single	2	3			5
10	6	29003	B	Single	1	6	2	1	
11	7	29003	B	Single	1	1			
12	8	29003	B	Single	2	1			3
13	9	29003	B	Single	3	6			2
14	10	29003	A	Single	4	1			4
15	11	29003	A	Single	1	1			5
16	12	29003	A	Single	5	6			1
17	13	29003	A	Single	1	6			7
18	14	29003	A	Married		3			1
19	15	29003	C	Married		5			2
20	16	29003	C	Married		2			2
21	17	29003	C	Married		1			1
...									
99	95	29001	D	Married			1	1	
100	96	29001	D	Married			1	2	
101	97	29001	D	Married			3	1	
102	98	29001	D	Married			1	7	
103	99	29001	D	Married			6	1	
104	100	29001	D	Married			2	9	

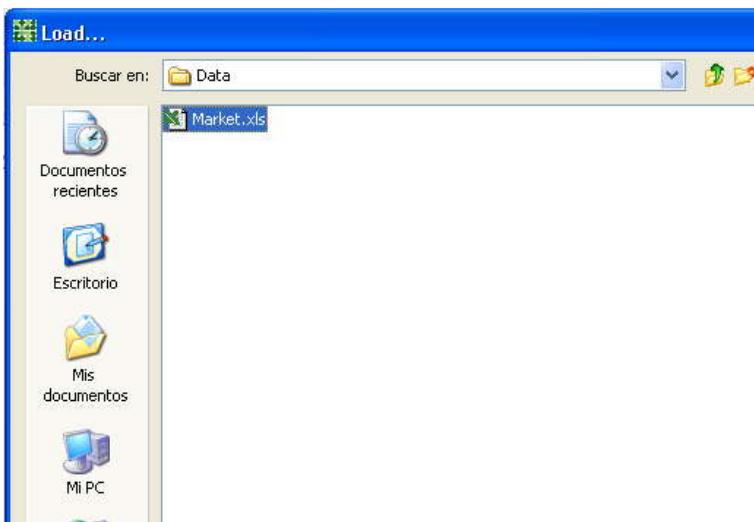
1.1- Load Step



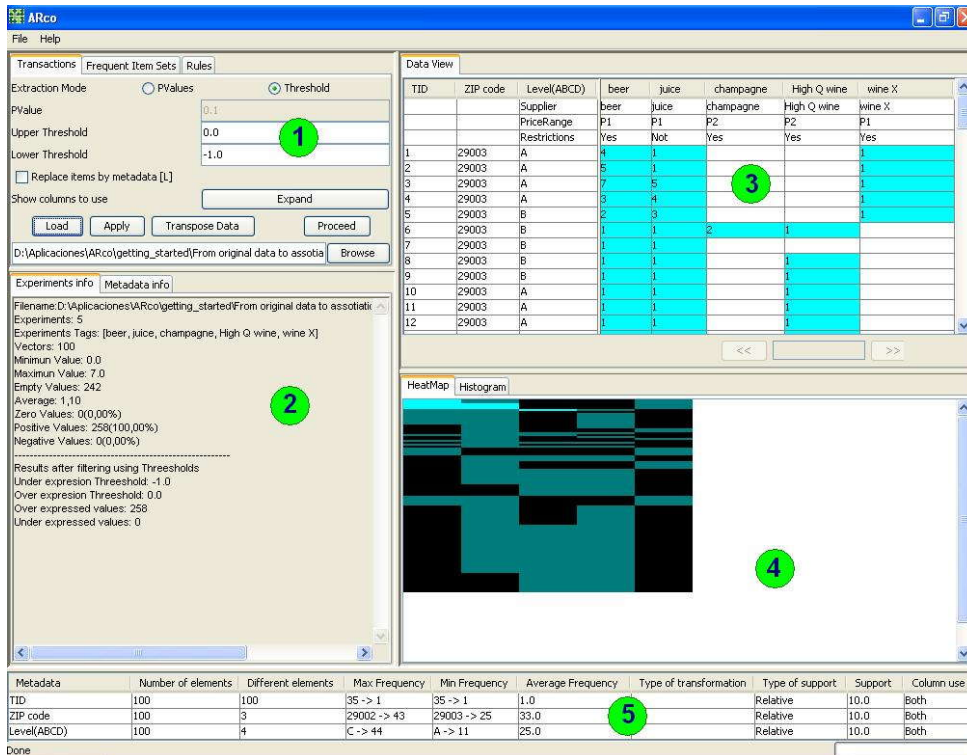
Double click in ARco icon to launch the program



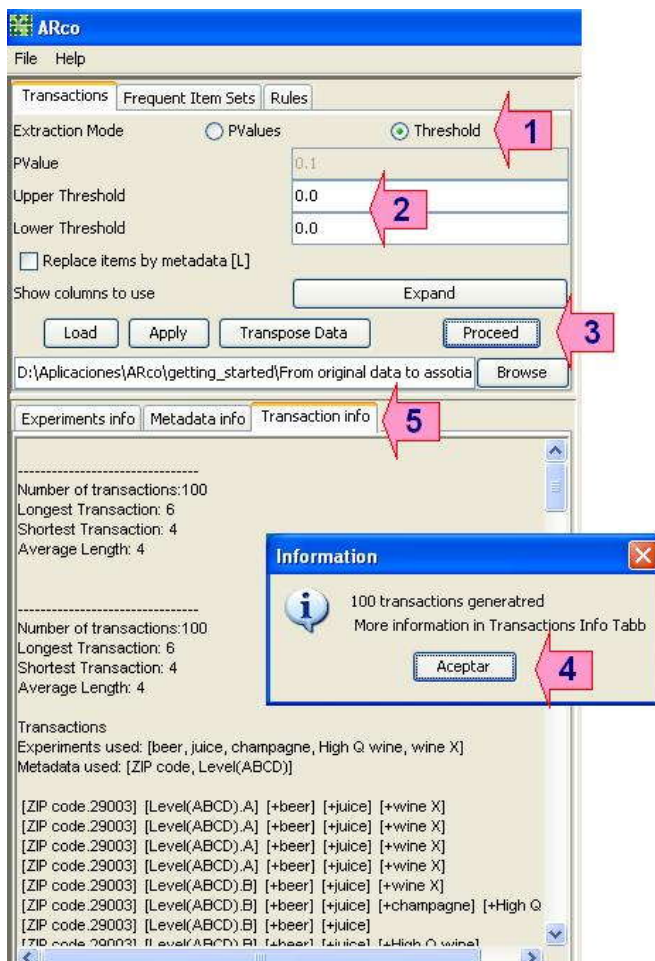
From the initial screen, in the Transactions Tab, use the "Load" button to start the load procedure.



Browsing the file.- A new data file will be uploaded. A "file dialog box" is used to allow browse and specify the file. Try with "marketDemo100.xls".



Full ARco screenshot: the main panels are: (1) Input parameter area; (2) text information about processes, (3) table display area; (4) image zone; (5) results information



Create transactions.

Data binarization can be performed by :

(1) using "threshold" extraction mode with

(2) value 0

(3) Press the Proceed button to produce transactions

(4) The text box will inform about the number of transactions produced;

(5) and additional information will become available on the transactions tab .

The transactions file will –by default- be created in the same directory as the original data with the same name and extension "*.TR" (user can modify the target directory and name).

Mining frequent itemsets from transaction file

ARco

File Help

Transactions Frequent Item Sets Run

Algorithm: Borglet Ard

Support Type: Absolute Relative

Support Mode: Unique Multiple

Minimal number of items: 1

Maximal number of items: 5

Minimal support: 10.0

Maximal support: 100.0

Multiple supports: Expand

Load Run

D:\Aplicaciones\ARco\getting_started\From original data to assotia Browse

Experiments info Metadata info Transaction info frequent sets info

Number of frequent item sets: 79
Longest frequent item set: 5
Shortest frequent item set: 1
Average Length: 2

Information

78 frequents items sets generated
More information in frequents sets info tabb

Aceptar

- (1) Select in the Frequent Item Sets" tab
- (2) set-on Borglet algorithm
- (3) Unique support
- (4) and relative support .

Keep the default parameter for maximum and minimal values.

- (5) Launch the procedure;

- (6) check the results

- (7) visualize the summary

Rules production from frequent itemsets

The screenshot shows the ARco software interface with the following components and annotations:

- Rules Configuration Panel:**
 - Confidence: 50.0 (Arrow 1)
 - Improvement: 1.0 (Arrow 2)
 - Minimal consequent size: 1
 - Buttons: Load, Run (Arrow 3), Expand, Browse
- Information Dialog:**
 - Message: 109 rules generated
 - Button: Aceptar (Arrow 4)
- Rules Table:**

Antecedent	Consequent	Confidence	Support	ABS Support
[Level(ABC...	[ZIP code.2...	66,67	16,00	16,00
[ZIP code.2...	[Level(ABC...	50,00	16,00	16,00
[Level(ABC...	[ZIP code.2...	61,36	27,00	27,00
[ZIP code.2...	[Level(ABC...	62,79	27,00	27,00
[Level(ABC...	[ZIP code.2...	61,90	13,00	13,00
[ZIP code.2...	[Level(ABC...	52,00	13,00	13,00
[+beer]	[ZIP code.2...	64,00	16,00	16,00
[ZIP code.2...	[+beer]	64,00	16,00	16,00
[ZIP code.2...	[+juice]	100,00	25,00	25,00
[+champag...	[ZIP code.2...	56,14	32,00	32,00
[ZIP code.2...	[+champag...	100,00	32,00	32,00
[ZIP code.2...	[+High Q wi...	100,00	32,00	32,00
[Level(ABC...	[+juice]	88,64	39,00	39,00
[Level(ABC...	[Level(ABC...	59,09	13,00	13,00
[Level(ABC...	[+juice]	92,86	13,00	13,00
[Level(ABC...	[ZIP code.2...	58,97	23,00	23,00
- Visualization Panel:** (Arrow 5)
- Original Data View Table:**

TID	ZIP code	Level(AB...	beer	juic
		Supplier	beer	juic
		PriceRange	P1	P1
		Restrictions	Yes	Not
1	29003	A	1.0	1.0
2	29003	A	5.0	1.0
3	29003	A	7.0	5.0
4	29003	A	3.0	4.0
5	29003	B	2.0	3.0
6	29003	B	1.0	1.0
7	29003	B	1.0	1.0
8	29003	B	1.0	1.0
9	29003	B	1.0	1.0
10	29003	A	1.0	1.0
11	29003	A	1.0	1.0
12	29003	A	1.0	1.0
13	29003	A	1.0	1.0

- (1) Select the “Rules” tab;
- (2) set the Confidence values to 50;
- (3) launch the algorithm;
- (4) check the global result;
- (5) prepare the visualization tab and
- (6) choose a rule