Quantitative association rule mining in genomics using apriori knowledge

Filip Karel, Jiří Kléma

Department of cybernetics, Czech Technical University in Prague,
Technická 2, Praha 6, 166 27
karelf1@fel.cvut.cz, klema@labe.felk.cvut.cz

Abstract Regarding association rules, transcriptomic data represent a difficult mining context. First, the data are high-dimensional which asks for an algorithm scalable in the number of variables. Second, expression values are typically quantitative variables. This variable type further increases computational demands and may result in the output with a prohibitive number of redundant rules. Third, the data are often noisy which may also cause a large number of rules of little significance. In this paper we tackle the above-mentioned bottlenecks with an alternative approach to the quantitative association rule mining. The approach is based on simple arithmetic operations with variables and it outputs rules that do not syntactically differentiate from classical association rules. We also demonstrate the way in which apriori genomic knowledge can be used to prune the search space and reduce the amount of derived rules.

Keywords: association rules, quantitative attributes, apriori knowledge, SAGE

1 Introduction

At present, large quantities of gene expression data are generated. Data mining and automated knowledge extraction in this data belong to the major contemporary scientific challenges. For this task clustering is one of the most often used method [2] – the most similar genes are found so that the similarity among genes in one group (cluster) is maximized and similarity among particular groups (clusters) is minimized. Although very good results are gained by this method there are three main drawbacks [3]:

1. One gene has to be clustered in one and only one group, although it functions in numerous physiological pathways.
2. No relationship can be inferred between the different members of a group. That is, a gene and its target genes will be co-clustered, but the type of relationship cannot be rendered explicit by the algorithm.
3. Most clustering algorithms will make comparisons between the gene expression patterns in all the conditions examined. They will therefore miss a gene grouping that only arises in a subset of cells or conditions.

Association rule (AR) mining [1] can overcome these drawbacks. However, when dealing with datasets containing quantitative attributes it is often advisable to adapt the original AR mining algorithm. Mining of quantitative association rules (QARs) is considered as an interesting and important research problem. It was described in several papers such as [5], [6], [18], [19] which proposed various algorithmic solutions. Nevertheless, the proposed algorithms often do not take time consumption into the account.

QAR mining techniques aimed at gene-expression data were proposed for example in [4] or [15]. Half-spaces are used to generate QAR in [4], rules of the form 'if the weighted sum of some variables is greater than a threshold, then, with a high probability, a different weighted sum of variables is greater than second threshold'. An example of such rule can be '0.99 $gene_1$ - 0.11 $gene_2$ > 0.062 → 1.00 $gene_3$ > -0.032'. This approach naturally overcomes the discretization problem, on the other hand it is quite hard to understand the meaning of the rule.

In [15], the authors bring external biological knowledge to the AR mining. They mine rules which directly involve biological knowledge into the antecedent side of the rule. The given method can be applied to mine annotated gene expression datasets in order to extract associations like '$cell\_cycle \rightarrow [+]condition_1$, $[+]condition_2$, $[+]condition_3$, $[-]condition_6$', which means that, in the dataset, a significant number of the genes annotated as 'cell cycle' are over-expressed in condition 1, 2 and 3 and under-expressed in condition 6. This approach works with binary values of gene-expression only.

In this paper, QAR mining algorithm [12] is used and further developed. Despite it is very different from the classical AR algorithms, it outputs association rules in the classical form '$gene_i = <l\_value_{gi}..h\_value_{gi}> \wedge gene_j = <l\_value_{gj}..h\_value_{gj}> \wedge ... \rightarrow cancer = 0/1$. We can read this rule as 'when the value of $gene_i$ is between l_value_{gi} and h_value_{gi} and the value of $gene_j$ is between l_value_{gj} and h_value_{gj} and ... then with a high probability the cancer will (not) occur'. The task can be rephrased as search for the genes and their values that coincide with the appearance of cancer.

The algorithm is by no means limited to the particular right hand side (RHS) of rules. The target variable *cancer* is used here as it represents the most interesting outcome. The invariable RHS also simplifies the evaluation in Section 4. As follows from the structure of the rules, the presented algorithm deals with discretized quantitative attributes. A priori discretization influences resulting rules. One of the main interests of this paper is to compare the discretization into more bins (which prevents information loss) with binarization.

Background knowledge (BK) – the external apriori biological information – can be extracted using various publicly accessible web databases and tools [7], [8], [10]. Possibility of using this source of information to improve the generation of ARs is another aim of this paper. We show that appropriate implementation of BK can improve the quality of generated rules. The simplest utilization of BK is to give the rules their biological sense by straightforward annotation of the set of rules without their pruning. BK also helps to focus on specific rule subsets by early utilization of regular expressions. The most interesting use of BK is to

get the most plausible rules by application of gene similarity. Moreover, BK can significantly reduce the search space.

The paper is organized as follows: Section 2 presents the SAGE data, studies possible ways of its preprocessing and introduces apriori knowledge relevant to the given dataset. Section 3 gives an outline of QAR algorithm and discusses the ways it can employ apriori knowledge. Section 4 summarizes the reached results with the main stress on the effects of discretization and utilization of apriori knowledge. Finally we conclude in Section 5.

2 Character of SAGE data and preprocessing of raw data

The SAGE (Serial Analysis of Gene Expression) technique aims to measure the expression levels of genes in a cell population [20]. In this paper, the raw data matrix described in [11] was used. The expression dataset consists of 11082 tags (i.e., genes or attributes) whose expression was measured in 207 SAGE libraries (i.e. 207 biological situations or experiments). The tags represent the subset of human genome which is currently unambiguously identifiable by Identitag [3], the biological situations embody various tissues (brain, prostate, breast, kidney or heart) stricken by various possible diseases (mainly cancer, but also HIV and healthy tissues).

	$gene_1$	$gene_2$	...	$gene_n$	*cancer*
$situation_1$	0	15	...	0	0
$situation_2$	8	4	...	0	1
⋮	⋮	⋮	⋮	⋮	⋮
$situation_m$	3	0	...	39	1

Table 1. The structure of the raw SAGE data (n=11082, m=207), the gene values correspond to the expression of the particular gene in the particular biological situation, *cancer* stands for a binary class.

The structure of the raw SAGE expression dataset is in Table 1. As the main observed disorder is carcinoma, a target binary attribute *cancer* was introduced by the domain expert. The class value is 0 for all the healthy tissues and also the tissues suffering by other diseases than cancer (77 situations, 37.2%). It is equal to 1 for all the cancerous tissues (130 situations, 62.8%).

SAGE datasets are sparse – a great portion of gene-expression values equal to zero. The distribution of zeroes among genes is very uneven. Housekeeping genes are expressed (nearly) in all the tissues, however there is a reasonable amount of genes having zero values in almost all situations. Such genes are not suitable for further rule mining. Table 2 shows the numbers of frequently expressed genes. We can see that out of the total number of 11082 genes, only 97 have at least 95% non-zero values.

X	number of genes
5%	97
20%	305
50%	1038
80%	2703

Table 2. The number of genes having at the most X% of zero values

2.1 Discretization of expression values

In order to minimize the role of noise in SAGE data, the data are usually discretized first. As the discretization also brings the information loss, it is always disputable which type of discretization to apply. For a thorough discussion upon the impact of discretization see [16].

Binarization is now the most widely used method of discretization of gene expression data, where 0 means that the gene is under expressed and 1 means that the gene is over expressed. There are two disadvantages of data binarization: (1) it results in the biggest information loss, (2) it significantly influences (or rather forms) the output rules.

Table 3 describes the distinction among different types of binarization. 'Max -Y%' binarization means that the Y% of the highest value is the 0/1 threshold (provided the highest value of $gene_i$ is 100 and Y=90%, the threshold is 10, all the values above are encoded as 1). In 'median' binarization the border is the value of median. Logically, the most uniform distribution is obtained through the 'median' binarization. The most similar to 'median' is 'Max -80%' binarization using the gene sets with lower numbers of zeros values and 'Max -90%' using the gene sets with higher numbers of zero values.

	Max -90%	**Max -80%**	**Max -70%**	**Median**
X gene-set	**0/1 ratio**	**0/1 ratio**	**0/1 ratio**	**0/1 ratio**
5%	0.28 / 0.72	0.56 / 0.44	0.74 / 0.26	0.49 / 0.51
20%	0.32 / 0.68	0.59 / 0.41	0.77 / 0.23	0.49 / 0.51
50%	0.45 / 0.55	0.66 / 0.34	0.81 / 0.19	0.49 / 0.51
80%	0.60 / 0.40	0.74 / 0.26	0.84 / 0.16	0.61 / 0.39

Table 3. The results of binarization in terms of the 0/1 ratio. X defines the gene sets shown in Table 2.

Discretization into more bins enables more accurate rules. However, the classical equi-width and equi-depth approaches fail in this case. The former introduces intervals that are nearly empty, the latter keeps the same frequency across the intervals with unnatural bounds. The discretization based on 1-D clustering has to be employed. In short, the discretization steps repeated for each attribute are:

1. Initialize equi-distantly the *centers* of bins.
2. Assign every record value to the nearest center.
3. Recalculate every center position (average value of all records assigned to the center).
4. If the position of all centers did not move then end, else go to 2/.

The results of discretization into four and six bins are in Table 4. 4-bin discretization has approximately the same number of values assigned to the lowest bin as 'Max -80%'. Better resolution is obtained in higher values only. Using 6-bin discretization the resolution is better even in low values. But still low numbers of values are assigned to the higher bins. This is caused by the original distributions of gene expression values, where the majority of values is very close to zero.

	4-bin discretization	**6-bin discretization**
X gene-set	**1/2/3/4 ratio**	**1/2/3/4/5/6 ratio**
5%	0.63 / 0.24 / 0.08 / 0.05	0.45 / 0.27 / 0.13 / 0.06 / 0.06 / 0.03
20%	0.65 / 0.25 / 0.07 / 0.03	0.48 / 0.29 / 0.12 / 0.05 / 0.04 / 0.02
50%	0.69 / 0.23 / 0.06 / 0.02	0.52 / 0.27 / 0.10 / 0.04 / 0.05 / 0.01
80%	0.74 / 0.19 / 0.05 / 0.02	0.59 / 0.20 / 0.08 / 0.04 / 0.08 / 0.01

Table 4. The ratio of the number of values using the clustering discretization.

2.2 Background knowledge

Genomic websites such as NCBI [10] or EBI [9] offer a great amount of heterogeneous background knowledge available for various biological entities. In this paper we focused on Gene Ontology (GO) terms. To access the gene annotation data for every tag considered, RefSeq identifiers were translated into EntrezGene identifiers [8], the mapping approached 1 to 1 relationship. Knowing the gene identifiers, the annotations were automatically accessed through hypertext queries to the EntrezGene database [10] and sequentially parsed by Python scripts.

GO terms A list of related GO terms can be found for each gene (however for a certain portion of genes there are no GO terms available and the list is empty). This list characterizes the given gene and can be used to assume on its molecular function (MF) or the biological processes and the cellular components it participates in. The lists can be searched by regular expressions in order to focus on specific subsets of genes.

Similarity matrices GO terms can straightforwardly be used to compute similarity among genes. The rationale sustaining this method is that the more GO terms the genes share, and the more specific the terms are, the more likely the genes are to be functionally related. Two matrices – for BPs and MFs – created by authors in [11] are used. The structure of the gene similarity matrices is in

Table 5. The similarity values lie in the interval $< 0; 1 >$, where 1 stands for the genes with the identical description for the given category of terms. There are around 85% of missing similarity values (denoted n/a) for the genes with empty lists of related GO terms.

	$gene_1$	$gene_2$	$gene_3$	$gene_4$	...	$gene_n$
$gene_1$	0.15	0.75	n/a	...	n/a
$gene_2$		n/a	0.12	...	0.93
$gene_3$			0.64	...	n/a
$gene_4$				...	n/a
⋮					⋮
$gene_n$					

Table 5. The structure of the gene similarity matrix.

In order to simplify the notion of similarity, both the above-described matrices are combined into one matrix as follows:

$$sim_{ij} = sim(BP)^2_{ij} + sim(MF)^2_{ij}$$

where $sim(BP)_{ij}$ is the similarity value for the genes i and j with respect to their biological process GO terms, $sim(MF)_{ij}$ is the similarity value for the same genes with respect to their molecular function GO terms.

3 QAR algorithm

An innovative QAR algorithm [12] is used for AR generation in this paper. The detailed algorithm description is out of the scope of this paper. The essential principles of the algorithm can be summarized as follows:

1. The input of the algorithm is a set of *atomic attributes*: $a_1, a_2, ...a_n$.
2. All the atomic attributes are discretized into D discretization bins and mapped to the consecutive row of integers beginning with one and ending with D (one represents the lowest value and D the highest value of an atomic attribute).
3. These preprocessed atomic attributes $pa_1, pa_2, ...pa_n$ are used to construct compound attributes – $x_i(pa_1, pa_2, ...pa_n) : N^n \rightarrow N$. *Compound attribute* is $x_i(pa_1, pa_2, ...pa_n) = \sum_{k=1}^{n} c_k a_k$, where c_k = {-1, 0, 1}, where i is number of compound attribute.
4. Each atomic (compound) attribute has a discrete distribution $P_i(t)$, two atomic (compound) attributes have a joint distribution $P_{ij}(t, s)$.
5. O is a set of all compact square or rectangle areas $o \subset <-\infty, \infty>$ x $<-\infty, \infty>$. For each pair $(x_i, x_j) \in P$ the algorithm searches for the best *areas of interest* o, where for each $(\alpha, \beta) \in o$

$$P_i(\alpha)P_j(\beta) - P_{ij}(\alpha, \beta) \geq \epsilon$$

6. From the areas of interest the best rules are extracted.

This algorithm takes an inspiration from earlier proposed algorithms [6], [14] or [19], but it comes with lower time consumption and pruning of redundant rules. On the other hand, the algorithm does not exhaustively enumerate all the relevant rules as it is not based on complete search through the state space. The algorithm works for binary attributes as well, although it loses its main advantages.

3.1 Injection of background knowledge into QAR algorithm

In order to increase noise robustness, focus and speed up the search, it is vital to have a mechanism to exploit background knowledge during AR generation. In the presented algorithm, BK can be taken into the account during the phase that combines atomic attributes into compound attributes.

The first option takes advantage of the lists of terms that describe the individual atomic attributes (genes in the SAGE data). The terms enable to focus on the rules that contain genes with specific characteristics. Provided x denotes a compound attribute, the variable $regexp$(x,'$*ribosom*$') delivers the number of genes that belong to x and whose at least one term matches the regular expression '$*ribosom*$'. The variable can be employed to get a limited set of rules that concern mainly (or only) ribosomal genes.

The second option exploits the gene similarity matrices [11]. This option focuses on plausible ARs, i.e., the rules that contain at least a certain portion of genes having common properties. The properties themselves do not have to be given by the user. An association rule can originate solely from the compound attributes with the value of gene similarity higher than a user defined threshold. Provided x denotes a compound attribute, the variable $svsim(x)$ gives the number of gene pairs belonging to x whose mutual similarity is known (distinct from n/a) and $mvsim(x)$ stands for its counterpart. $Sumsim(x)$ denotes the similarity sum over the set of genes belonging to x, $insim(x, min, max)$ stands for the number of gene pairs whose similarity lies between min and max.

Consequently, the variable $\frac{sumsim(x)}{svsim(x)}$ makes the average similarity of the compound attribute x, while the variable $\frac{insim(x,thres,1)}{svsim(x)}$ gives a proportion of the strong interactions (similarity higher than the threshold) within the compound attribute. The variable $\frac{svsim(x)}{svsim(x)+mvsim(x)}$ can avoid the compound attributes with prevailing genes of an unknown function. Relational and logical operators enable to create the final constraint, e.g., $V_1 \geq thres1$ and $V_2 \neq thres2$ where V_i stands for an arbitrary variable characterizing the compound attribute. Although we consider GO terms only, the framework is obviously general and the constraints can also be simultaneously derived from different external datasets.

The described technique obviously causes early pruning of the search space. Some of the compound attributes are rejected and the algorithm does not further search for the rules which do not satisfy the condition given by BK.

4 Experiments and results

This section presents the achieved experimental results. The influence of selected discretization methods is discussed. ARs in the classical form are generated. Conditions on the gene expression values are conjuncted on their LHS, the number of conditions is limited to three. The rules always have the attribute 'cancer' on their RHS. Confidence, support [1] and lift [17] measures are used to evaluate the quality of rules.

The file with maximum of 5% zero values was used. The input table for AR mining consists of 98 genes (attributes) and 207 situations (transactions). The number of attributes is low as the general scalability of the presented algorithm is not concerned here. It has already been proven in earlier works [12,13], along with its ability to reduce redundancy of the resulting set of rules. The main concern is to demonstrate applicability of BK to further improve understandability and scalability of QAR mining.

4.1 Rules without background knowledge

Table 7 shows the influence of discretization methods on the number of generated rules. This number is several times higher using a multi-bin discretization compared with binarization. There are also distinctions among particular binarization types, although not so significant. More rules are generated using binarizations with a more uniform distribution of zero and one values.

Similarity of rules generated by different discretization techniques was also examined, although it is hard to exactly compare different sets of rules. We considered two rules equal when all the antecedent genes, which occurred in the first rule also occurred in the second rule. For example, if genes with ID numbers 9, 13 and 82 occurr in the $rule_1$ and the same genes also occurr in the $rule_2$, then $rule_1 = rule_2$, no matter what values the genes take in the rules. The results are captured in Table 6, where the value on i-th column and j-th row is gained as

$$r_{ij} = \frac{number\_of\_rules_{i,j}}{number\_of\_rules_j},$$

where $number\_of\_rules_{i,j}$ is the number of rules generated both by the i-th type of discretization and by the j-th type of discretization and $number\_of\_rules_j$ is the total number of rules generated by the j-th type of discretization.

We can see that the ratios are quite low. It means that one can achieve a certain percentage of rules that agree in both types of discretization but quite a high number of rules is different. For example, when using 'Max -70%' and 'Max -80%' we gain approximately the same absolute number of rules from which only one fifth is equal. Also, '6-bin' discretization identifies only from 60% to 70% of rules identified using other types of discretization.

Experimentally it was found that these numbers depend on min_supp threshold. Lowering min_supp the ratios of 'identical' rules increase and higher numbers of similar rules are generated.

	Max -90%	Max -80%	Max -70%	Median	4-bin	6-bin
Max -90%	1	0.37	0.07	0.57	0.30	0.56
Max -80%	0.25	1	0.21	0.41	0.58	0.51
Max -70%	0.05	0.18	1	0.39	0.45	0.74
Median	0.26	0.29	0.23	1	0.48	0.61
4-bin	0.12	0.37	0.25	0.44	1	0.58
6-bin	0.15	0.20	0.25	0.35	0.36	1

Table 6. The number of the equal rules having 3 antecedent attributes generated by different discretization methods.

4.2 Using background knowledge (BK) for rules generation

Syntactically the same rules were generated with using BK, but a pruning condition was added. Using notation from Section 3.1, the applied conditions can be written as: 'generate rules with a compound attribute x only if $insim(x, 0.65, 2) \geq 1$'. It means that x is acceptable only if there is a pair of genes of x whose similarity is higher than the $min\_sim = 0.65$ threshold (at the same time it positively holds $svsim(x) \geq 2$). This condition early prunes the space of compound attributes and it is not only a rule filtering condition as for example min_conf condition.

	Max -90%	**Max -80%**	**Max -70%**	**Median**	**4-bin**	**6-bin**
3-ant (min_conf=0.9)	1 102	1 672	1 453	2 392	2 617	4 210
3-ant (min_conf=1.0)	88	33	15	90	126	65
3-ant (min_conf=0.8)	1 681	3 227	1 977	5 453	4 432	6 966
3-ant (min_conf=0.9)	150	152	117	317	247	360

Table 7. The number of rules created by different types of discretization without using background knowledge (top) and with background knowedge (bottom). Min_supp = 0.1, min_lift = 1.3, min_similarity = 0.65

	Binarization	**4-bin**	**6-bin**
without background knowledge	1.5×10^6	6.5×10^6	1.2×10^7
with background knowledge	1.7×10^5	7.1×10^5	1.3×10^6

Table 8. Number of verifications.

The number of rules (bottom part of table 7) is approximately 10 times lower than without using BK, the same holds for the number of verifications that the algorithm carries out. For $min\_conf = 0.8$ we obtain approximately the same number of rules as for $min\_conf = 0.9$ without BK. Time consumption remains

about ten times lower as the time-consumption of used algorithm does not depend on min_conf.

Further, the similarity of rules generated with and without BK is explored. In Table 9 we can observe the top 5 genes (top) and the top 5 pairs of genes (bottom) according to the number of their occurrences in rules.

without BK				with BK			
Max -80%	Median	4-bin	6-bin	Max -80%	Median	4-bin	6-bin
4	9	2	13	41	58	13	13
75	6	13	97	18	36	97	41
70	58	6	2	43	9	41	97
43	97	3	6	16	43	16	9
72	52	97	3	52	13	58	16
4-44	21-58	25-78	13-97	3-88	16-58	13-75	6-17
4-75	9-55	2-18	2-97	53-75	13-58	13-55	11-97
55-72	9-42	89-97	2-90	42-43	22-51	6-17	11-13
4-71	9-36	2-97	13-46	41-76	43-75	13-40	13-75
4-70	9-52	3-75	13-86	41-63	43-52	11-13	13-95

Table 9. Top 5 genes (top) and top 5 pairs (bottom) according to the number of occurrences in rules.

For '4-bin' and '6-bin' discretizations the top 5 gene lists are almost the same. Without BK, all of the 4-bin discretization top genes are also the top genes for 6-bin discretization. With BK this holds for 4 out of 5 genes. By contrast, for binarizations (both with and without BK) there is no overlap in the top gene lists. If we compare the gene lists of the identical discretizations with and without using BK, we observe that the multi-bin discretization and the 'median' binarization get the identical gene sets with and without BK.

For the top 5 pairs we have very similar observations as for the lists of top 5 genes. Generally, in the categories with and without BK the 4-bin and 6-bin discretizations are giving very similar results. 'Max -80%' and 'median' binarizations differentiate quite a lot. Between the two categories the most similar results are gained for 4-bin and 6-bin discretizations.

A more detailed comparison of particular gene occurrences in generated rules with and without BK is in Figure 1. Some of the genes have almost the same number of occurrences ($gene_{13}$), whereas other genes which have a very high number of occurrences using BK do not appear frequently in runs without application of BK ($gene_{41}$).

In general, the genes with prevalence of 'n/a' values in the similarity matrices are discriminated from the rules when using BK. However, a gene without annotation can still appear in a neighborhood of 'a strong functional cluster' of other genes. This occurrence then signifies its possible functional relationship with the given group of genes and it can initiate its early annotation. On the other hand,

the genes with extensive relationships to the other genes may increase their occurrence in the rules inferred with BK.

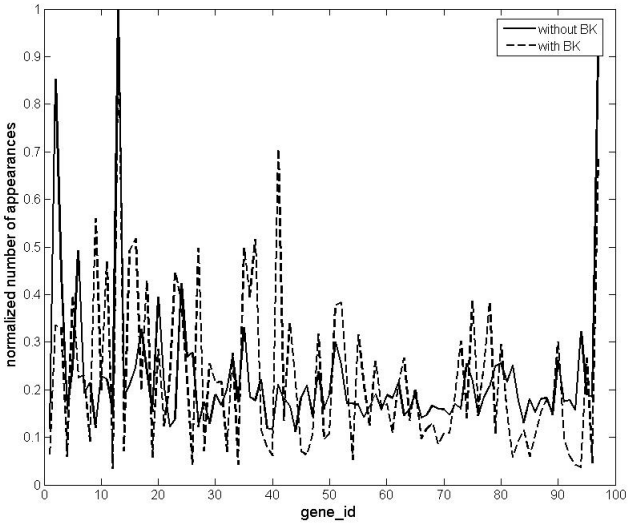

Figure 1. The frequency of particular genes in the generated rules with and without background knowledge for '6-bin' discretization.

5 Conclusions

In this paper, an alternative approach to QAR mining was verified on gene expression data. The paper discussed the influence of discretization methods on the generated rules. It was shown that the output set of rules is significantly influenced by the used discretization both wrt the number of generated rules and their composition. The presented QAR algorithm allowed us to use advantages of discretization into more bins and at the same time to generate rules without combinatoric explosion and without generation of redundant rules. In the light of our findings we think that more attention should be paid to the automatic discretization of gene expression values.

The paper also described and implemented the general framework for exploitation of BK during AR mining. It mainly helps to automatically focus on the most plausible candidate rules. At the same time, pruning conditions based on BK reduce time consumption significantly, while the number of plausible rules remains approximately the same. The conditions used in presented experiments were quite simple. Exploration of other possibilities of this framework and using more complex BK conditions is one of our major future challenges.

Acknowledgement. Filip Karel has been supported by the Ministry of Education, Youth and Sports of the Czech Republic as a part of the specific research

at the CTU in Prague - project nr. CTU0712613. Jiri Klema has been supported by the grant 1ET101210513 "Relational Machine Learning for Analysis of Biomedical Data" funded by the Czech Academy of Sciences.

References

1. R. Agrawal, T. Imelinsky, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Conference on Management of Data*, pages 207–216, Washington, D.C., 1993.
2. Eisen M. B., Spellman P. T., Brown P. O., and Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proceedings of National Academy of Science of the USA 95*, pages 14863–14868,, 1998.
3. Becquet C., Blachon S., Jeudy B., Boulicaut J-F, and Gandril O. Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data. *Genome Biology*, 3:531–537, 2002.
4. Georgii E., Richter L., Ruckert U., and Kramer S. Analyzing Microarray Data Using Quantitative Association Rules. *Bioinformatics*, pages ii123–ii129, 2005.
5. T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining optimized association rules for numeric attributes. In *In Proc. of ACM SIGMOD Conference on Management of Data*, Montreal, Canada, 1996.
6. S. Guillaume. Discovery of ordinal association rules. In *In Proceedings of the Sixth Pacific-Asia Conference PAKDD'02*, Taiwan, 2002.
7. http://crfb.univ mrs.fr/gotoolbox/. GOToolBox website.
8. http://discover.nci.nih.gov/matchminer/. Matchminer website.
9. http://www.ebi.ac.uk/. EBI website.
10. http://www.ncbi.nlm.nih.gov/. NCBI website.
11. Kléma J., Soulet A., Crémilleux B., Blachon S., and Gandrillon O. Mining plausible patterns from genomic data. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pages 183–190, 2006.
12. F. Karel. Quantitative and ordinal association rules mining (QAR mining). In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4251 of *LNAI*, pages 195–202. Springer, 2006.
13. F. Karel and J. Kléma. Ordinální asociační pravidla. In *Konference Znalosti 2005*, pages 226–233. VŠB-TUO, 2005.
14. R.J. Miller and Y. Yang. Association rules over interval data. In *In Proc. of ACM SIGMOD Conference on Management of Data*, Tuscon, AZ, 1997.
15. Carmona-Saez P., Chagoyen M., Rodriguez A., Trelles O., Carazo J.M., and Pascual-Montano A. Integrated analysis of gene expression by association rules discovery). *BMC Bioinformatics*, page 7:54, 2006.
16. Ruggero G. Pensa, Claire Leschi, Jérémy Besson, and Jean-François Boulicaut. Assessment of discretization techniques for relevant pattern discovery from gene expression data. In *BIOKDD*, pages 24–30, 2004.
17. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *in Knowledge Discovery in Databases*, Cambridge, 1991.
18. R. Rastogi and K. Shim. Mining optimized association rules with categorical and numeric attributes. *IEEE Trans. on KD Engineering*, 14(1), 2002.
19. R. Srikant and R. Agrawal. Mining quantitative association rules in large relational databases". In *In Proc. of ACM SIGMOD Montreal*, 1996.
20. Velculescu V., Zhang L., Vogelstein B., and Kinzler K. SAGE (Serial Analysis of Gene Expression). *Science*, page 270:484.7, 1995.