Mining Association Rule Bases from Integrated Genomic Data and Annotations

Ricardo Martinez[1], Nicolas Pasquier[1], and Claude Pasquier[2]

[1] Laboratoire I3S, UNS/CNRS UMR-6070, 06903 Sophia Antipolis, France.
[2] IDBC, UNS/CNRS UMR-6543, Parc Valrose, 06108 Nice, France.

Abstract. During the last decade, several clustering and association rule mining techniques have been applied to highlight groups of co-regulated genes in gene expression data. Nowadays, integrating these data and biological knowledge into a single framework has become a major challenge to improve the relevance of mined patterns and simplify their interpretation by biologists. GenMiner was developed for mining association rules from such integrated datasets. It combines a new nomalized discretization method, called NorDi, and the Close algorithm to extract minimal non-redundant association rules only. Experimental results show that GenMiner requires less memory than Apriori based approaches and that it improves the relevance of extracted rules. Moreover, association rules obtained revealed significant co-annotated and co-expressed gene patterns showing important biological relationships supported by recent biological literature.

1 Introduction

Gene expression technologies are powerful methods for studying biological processes through a transcriptional viewpoint. Since many years these technologies have produced vast amounts of data by measuring simultaneously expression levels of thousands of genes under hundreds of biological conditions. The analysis of these numerical datasets consists in giving meaning to changes in gene expression to increase our knowledge about cell behavior. In other words, we want to interpret gene expression data via integration of gene expression profiles with corresponding biological knowledge (gene annotations, literature, etc.) extracted from biological databases. Consequently, the key task in the interpretation step is to detect the present co-expressed (sharing similar expression profiles) and co-annotated (sharing the same properties such as function, regulatory mechanism, etc.) gene groups.

Several approaches dealing with the interpretation problem have recently been reported. These approaches can be classified in three axes [15]: *expression-based approaches*, *knowledge-based approaches* and *co-clustering approaches*. The most currently used interpretation axis is the *expression-based* axis that gives more weight to gene expression profiles. However, it presents many well-known drawbacks. First, this approach cluster genes by similarity in expression profiles across all biological conditions. However, gene groups involved in a biological process might be only co-expressed in a small subset of conditions [2].

Second, many genes have different biological roles in the cell, they may be conditionally co-expressed with different groups of genes. Since almost all clustering methods used place each gene in a single cluster, that is a single group of genes, his relationships with different groups of conditionally regulated genes may remain undiscovered. Third, discovering biological relationships among co-expressed genes is not a trivial task and requires a lot of additional work, even when similar gene expression profiles are related to similar biological roles [20].

The use of association rule mining (ARM), that is another unsupervised data mining technique, was proposed to overcome these drawbacks. ARM aims at discovering relationships between sets of variable values, such as gene expression levels or annotations, from very large datasets. Association rules identify groups of variable values that frequently co-occur in data lines, establishing relationships with the form: $A \Rightarrow B$ between them. This rule means that when a data line contains variable values in A it is also likely to contain variable values in B. It has been shown in several research reports that ARM has several advantages. First, ARs can contain genes that are co-expressed in a subset of the biological conditions only. From this viewpoint, it and can be considered as a *bi-clustering* technique. Second, a gene can appear in several AR, if its expression profile fulfills the assignation criteria. That means, if a gene is involved in several co-expressed gene groups, it will appear in each and every one of these groups. Third, association rules are orientated knowledge patterns with the form *if condition then consequent* that describe directed relationships. This enables the discovery of any type of relationships between gene expression measures and annotations as they can be premisses or consequents of association rules. Fourth, since all types of data are considered in the same manner with ARM, several heterogeneous biological sources of information can be easily integrated in the dataset. These features make ARM a technique that is complementary to clustering for gene expression data analysis.

The GenMiner principle was introduced, with preliminary experimental results, in [16]. In this paper, we present a new Java implementation of GenMiner and new experimental results on the biological significance of extracted rules, the applicability and scalability of the algorithm and performance comparisons with other ARM approaches. This paper is organized as follows. Section 2 and 3 present ARM basics and related works respectively. The GenMiner approach is described in section 4 and the integrated dataset constituted for the experiments is presented in section 5. Experimental results are presented in section 6 and the paper ends with a discussion and conclusion in section 7.

2 Association rule mining

Association rules (ARs) express correlations between occurrences of variable values in the dataset as directed relationships between sets of variable values. In the data mining literature, variable values are called *items* and sets of items are called *itemsets*. For each AR, statistical measures assess the scope, or frequency, and the precision of the rule in the dataset. The classical statistics for this are respectively the *support* and the *confidence* measures. For instance, an AR *Event(A)*, *Event(B)* $\Rightarrow$ *Event(C)*, *support=20%*, *confidence=70%* states that when events A and B occur, event C also occurs in 70% of cases, and that all three events occur together in 20% of all situations. This AR is extracted from a dataset containing *Event(A)*, *Event(B)* and *Event(C)* as items and data lines of the dataset describe co-occurred events, that is known situations. Since all ARs are not useful or relevant, depending on their frequency and precision, only ARs with support and confidence exceeding some user defined minimum support (*minsupp*) and minimum confidence (*minconf*) thresholds are extracted.

Extracting ARs is a challenging problem since the search space, i.e. the number of potential ARs, is exponential in the size of the set of items and several dataset scans, that are time expensive, are required. Several studies have shown that ARM is a NP-complete problem and that a trivial approach, considering all potential ARs, is unfeasible for large datasets. The first efficient approach proposed to extract ARs is the Apriori algorithm [1]. Several optimisations of this approach have been proposed since, but all these algorithms give response times of the same order of magnitude and have similar scalability properties. Indeed, this approach was conceived for the analysis of sales data and is thus efficient when data is weakly correlated and sparse but performances drastically decrease when data are correlated or dense [5]. Moreover, with such data, a huge number of ARs are extracted, even for high *minsupp* and *minconf* values, and a majority of these rules are redundant, that is they cover the same information. For instance, consider the following five rules that all have the same support and confidence and the item *annotation* in the antecedent:

1. *annotation* $\Rightarrow$ *gene1*↑
2. *annotation* $\Rightarrow$ *gene2*↑
3. *annotation* $\Rightarrow$ *gene1*↑, *gene2*↑
4. *annotation*, *gene1*↑ $\Rightarrow$ *gene2*↑
5. *annotation*, *gene2*↑ $\Rightarrow$ *gene1*↑

The most relevant rule from the user's viewpoint is rule 3 since all other rules can be deduced by inference from this one, including support and confidence (but the reverse does not hold). Information brougth by all other rules are summed up in rule 3, that is a *non-redundant association rule with minimal antecedent and maximal consequent*, or *minimal non-redundant ARs* for short. This situation is frequent when mining correlated or dense data, such as genomic data, and to address this problem the GenMiner ARM approach uses the Close algorithm to extract minimal non-redundant ARs only.

3 Related works

Several applications of ARM to the analysis of gene expression data have been recently reported [7, 21, 11]. These applications aimed at discovering frequent gene patterns among a subset of biological conditions. These patterns were represented as ARs such as: *gene1*↓ $\Rightarrow$ *gene2*↑, *gene3*↓. This rule states that, in a significant number of biological conditions, when *gene1* is under-expresssed, we also observe an over-expression of *gene2* and an under-expression of *gene3*. These applications successfully highlighted correlations between gene expression profiles, avoiding some drawbacks of classical clustering techniques [11]. However, in these applications, biological knowledge was not taken into account and the task of discovering and interpreting biological similarities hidden within gene groups was left to the expert.

Recently, an approach to integrate gene expression profiles and gene annotations to extract rule with the form *annotations* ⇒ *expression profiles* was proposed in [6]. However, this approach presents several weaknesses. First, it uses the Apriori ARM algorithm [1] that is time and memory expensive in the case of correlated data. Moreover, it generates a huge number of rules among which many are redundant thus complexifying the interpretation of results. This is a well-known major limitation of the Apriori algorithm for correlated data [6, 21]. Second, extracted rules are restricted to a single form: Annotations in the left-hand-side and expression profiles in the right-hand-side. However, all rules containing annotations and/or expression profiles, regardless of the side, bring important information for the biologist. Third, it uses the two-fold change cut-off method for discretizing expression measures in three intervals, a dangerous simplification that presents several drawbacks [18].

Discretization, which is needed for most of ARM implementations, is a delicate issue. According to the criteria used, there may be drastic changes on the rules generated. A recent paper proposed a way around this problem by running a biclustering algorithm over the gene expression matrix and then, by associating genes with the groups to which they belong [14]. The authors claim that the main advantage of this approach is that it reduces drastically the number of columns in the matrix and thus, that it simplify both the processing of the data and the interpretation of the rules. However, this depends mainly on the number of biclusters generated. In order to obtain very specific rules with low support, one needs to generate a huge number of small biclusters. Thus, the use of an efficient ARM algorithm is still needed and the interpretation of the resultant rules will still be very difficult.

GenMiner was developed to address these weaknesses and fully exploit ARM capabilities. It enables the integration of gene annotations and gene expression profile data to discover intrinsic associations between them. We chose to keep every colum from gene expression data but we use the novel NorDi method for discretizing gene expression measures. GenMiner takes advantage of the Close [19] algorithm that can efficiently generate low support and high confidence non-redundant association rules, thus reducing the number of ARs and facilitating their interpretation by the biologist. With these features, GenMiner is an ARM approach that is adequate to biologists requirements for genomic data analysis.

4 GenMiner approach

GenMiner follows the classical three steps of ARM approaches: (1) data selection and preparation, (2) ARs extraction and (3) ARs interpretation. It uses the NorDi algorithm for discretizing gene expression data during phase (1) and the Close algorithm for extracting minimal non-redundant ARs during phase (2). It is a co-clustering approach that discovers co-expressed and co-annotated gene groups at the same time according to co-ocurrences of gene expression profiles and annotations. It is a bi-clustering approach that finds co-annotated and co-expressed gene groups even in a small subset of biological conditions.

The whole process of GenMiner is deterministic and extracted ARs are not constrained in their form and their size in order to ensure that all kinds of relationships between gene expression profiles and annotations are discovered. The actual implementation of GenMiner does not integrate graphical visualization tools and complementary programs must be used to manipulate the results.

4.1 NorDi algorithm

The *Normal Discretization* (NorDi) algorithm was developed to improve gene expression measures discretization into items. This algorithm is based on statistical detection of outliers and the continuous application of normality tests for transforming the initial sample distribution "almost normal" to a "more normal" one. The term "almost" means that the sample distribution can be normally distributed without the outlier's presence.

Let us assume that the expression data measures are presented as an nXm matrix: $\boldsymbol{E}$ with n genes (rows) and m samples or biological conditions (columns). Each matrix entry, $e_{i,j}$ represents the gene expression measure of gene i in sample j where $e_{i,j}$ is continuous in all real numbers. Let's suppose that the gene expression matrix $\boldsymbol{E}$ accomplishes the following assumptions:

1. All data is well cleaned (minimal noise).
2. Number of genes is largely enough.
3. The samples of the matrix S_j for every $j = 1, 2, ..., m$ are independent from each other and they are "almost" normally distributed $S_j \sim N(\mu_j, \sigma_j)$.
4. Missing values are no significant regarding the number of genes.

The NorDi algorithm is based on the observation that every sample of the expression matrix S_j can be "more" normally distributed $S_j^k \sim N(\mu_j, \sigma_j)$ if all outliers of each sample are momentarily removed (that is keeping a list of the k removed outliers for each sample, i.e. L_j^k) by Grubbs outliers method [12]. Each time an outlier k is removed, a Jaque-Bera normality test [3] has to be accomplished for the remaining sample S_j^k, where k is the number of removed outliers at each step in sample S_j and $k = 0, 1, 2, \ldots, clean$ ($k = clean$ means that there are no more outliers in the sample according to the Grubbs criterium). So, for every sample, we obtain the remaining sample S_j^{clean} that is "more normally" distributed than the original sample S_j. To verify this assertion we compare S_j^{clean} against S_j using the QQ-plot [17] and Lilliefors [13] normality tests. Then, we calculate the over-expressed, Ot, and under-expressed, Ut, cutoff thresholds using the $z - score$ methodology [22] over the cleaned sample S_j^{clean}.

Supposing the four precedent assumptions with $S_j^{clean} \sim N(\mu_j, \sigma_j)$ normal distributed and a $1 - \alpha$ predetermined confidence degree, the $z - score$ threshold cutoffs for three intervals are defined as:

- $Z_j = \frac{e_{i,j}-\mu_j}{\sigma_j} \geq z_{\alpha/2} = Ot \Rightarrow e_{i,j}$: over-expressed (↑),
- $Z_j = \frac{e_{i,j}-\mu_j}{\sigma_j} \leq z_{\alpha/2} = Ut \Rightarrow e_{i,j}$: under-expressed (↓),
- $Ut < e_{i,j} > Ot \Rightarrow e_{i,j}$: unexpressed,

where $z_{\alpha/2} = \Phi^{-1}(1-\alpha/2)$, if the cumulative distribution function is $\Phi(z_{\alpha/2}) = P(S_j^{clean} \leq z_{\alpha/2}) = 1- \alpha/2$.

It is important to notice that this procedure for computing the threshold cutoffs is done over all the m cleaned samples S_j^{clean} contained in the expression matrix $\boldsymbol{E}$. Once the computation of threshold cutoffs is done, the k elements in each sample's outliers list L_j^k are integrated to the original sample S_j and the discretization procedure is calculated for all values in S_j. The main reason is that outliers values cannot be removed from the analysis because they may contain relevant information of the biological experiment.

4.2 Close algorithm

Close is a *frequent closed itemsets* based approach [19] for extracting minimal non-redundant AR defined as follows. An AR is *redundant* if it brings the same or less general information than is brought by another rule with identical support and confidence [8]. Then, an AR *R* is a minimal non-redundant AR if there is no AR *R*' with same support and confidence, which antecedent is a subset of the antecedent of *R* and which consequent is a superset of the consequent of *R*. Close first extracts equivalence classes of itemsets, defined by *generators* and *frequent closed itemsets*, and generates from them the *Informative Basis* containing only minimal non-redundant ARs. This basis (minimal set) is a generating set for all ARs that captures all information brought by the set of all rules in a minimal number of rules, without information loss [8]. Experiments conducted on benchmark datasets show that the rule number reduction factor varies from 5 to 400 according to data density and correlation [19]. Moreover, when data is dense or correlated, Close reduces extraction time and memory usage since the search space of frequent closed itemsets based approaches is a subset of the search space of Apriori based approaches. Several algorithms for extracting frequent closed itemsets, using complex data structures to improve efficiency, have been proposed since Close. However, they do not extract generators, precluding the Informative Basis generation, and their response times, that depends mainly on data density and correlation, are of the same order of magnitude.

5 Annotations enriched Eisen *et al.* dataset

To validate the GenMiner approach we applied it to the well-known genomic dataset used by Eisen *et al.* [10]. This dataset contains expression measures of 2 465 *Saccharomyces cerevisiae* genes under 79 biological conditions extracted from a collection of four independent microarray studies during several biological processes: Cell cycle, Sporulation, Temperature shock and Diauxic shift experiments. Gene expression measures were discretized using NorDi algorithm at a 95% confidence level.

Each yeast gene was annotated with its associated terms in *Yeast GO Slim* (a yeast-specific cut-down version of Gene Ontology), its associations with research papers, the KEGG pathways in which it is involved, its phenotypes and the transcriptional regulators that bind its promoter regions.

The resulting dataset is a matrix of 2 465 lines representing yeast genes and 737 columns representing expression levels (discretized gene expression measures) over the 79 biological conditions and at most 658 gene annotations (24 GO annotations, 14 KEGG annotations, 25 transcriptional regulators, 14 phenotypes and 581 pubmed keywords). On the whole, the dataset contains 9 839 items (variable values). This dataset and the GenMiner implementation are available on the GenMiner web site[3].

6 Experimental results

We conducted several experiments to evaluate the biological significance of extracted ARs, to compare the applicability of GenMiner and Apriori based approaches and to evaluate the scalability of GenMiner when mining very large dense biological datasets. For these experiments, the Java implementation of GenMiner was applied to the annotations enriched Eisen *et al.* dataset. All types of rules, containing gene annotations or gene expression levels either or both in the antecedent and the consequent, were extracted for *minsupp*=0.003 (at least 7 lines) and *minconf*=30%.

6.1 Biological interpretation of extracted association rules

Table 1 to 3 show some examples of the different form of rules extracted by GenMiner. In these tables, supports are given in number of transactions and confidences are given in percentages; the prefixes *go:*, *path:*, *pmid:*, *pr:*, *phenot:* are used to identify GO terms, KEGG pathways, Pubmed identifiers, promoters and phenotypes respectively; the labels *heat*, *diau* and *spo* refer to the different time points of the Heat shock, Diauxic shift and Sporulation experiments respectively; $\uparrow$ denotes an over-expression while $\downarrow$ denotes an under-expression.

ARs with the form *annotations* $\Rightarrow$ *expression levels* (Table 1) show groups of genes associated with the same annotations that are over-expressed or under-expressed in a set of biological conditions. Rules 1 and 2 highlight a general reduction of transcription and protein synthesis following a heat shock, leading to cellular damages. This is confirmed by rule 3 which shows that genes regulated by RAP1 and FHL1, which are two key regulators of ribosomal protein genes, are under-expressed in this experiment. This last rule reflects the known fact that RAP1 recruits FHL1 to activate transcription [23]. A reduction of protein synthesis in the last time point of the Diauxic shif experiment is highlighted by rule 4. Additionally, rules 5 to 7 show that the genes involved in *oxidative phosphorylation*, *citrate cycle* and *glyoxylate and dicarboxylate metabolism* were

[3] `http://bioinfo.unice.fr/publications/genminer_article`.

also mainly over-expressed at the last time points. These rules reflect the main metabolic changes associated to the diauxic shift in yeast, manually identified in [9]

Table 1. Associations *annotations* ⇒ *expression levels.*

Rule	Antecedent	Consequent	supp.	conf.
1	go:0006412 (translation) go:0005840 (ribosome)	heat3↓	103	51
2	go:0005840 (ribosome) go:0003723 (RNA binding)	heat3↓	12	57
3	pr:RAP1 pr:FHL1	heat3↓	71	62
4	path:sce03010 (ribosome)	diau7↓	121	92
5	path:sce00190 (oxidative phosphorylation)	diau7↑	18	33
6	path:sce00020 (citrate cycle)	diau6↑ diau7↑	18	60
7	path:sce00630 (glyoxylate/dicarboxylate metabolism)	diau7↑	8	53

ARs with the form *expression levels* ⇒ *annotations* (Table 2) show groups of genes that are over-expressed or under-expressed in a set of biological conditions and have the corresponding gene annotations. Selected rules show information related to the Sporulation experiment (rules 1 and 2), the Heat shock process (rules 3 and 4) and the Diauxic shift process (rule 5) reported in the corresponding biological literature.

Table 2. Associations *expression levels* ⇒ *annotations.*

Rule	Antecedent	Consequent	supp.	conf.
1	spo4↓ spo5↓ spo6↓	go:0005975 (carbohydrate metabolism)	12	52
2	spo3↓ spo4↓ spo5↓	path:sce00010 (Glycolysis)	13	52
3	heat3↓ heat4↓ heat5↓	go:0006412 (translation)	35	88
4	heat2↓	go:0042254 (ribosome biogenesis)	39	66
5	diauxic6↓ diauxic7↓	go:0006412 (translation)	21	66

ARs with the form *annotations* ⇒ *annotations* (Table 3) contain gene annotations both in the antecedent and consequent. They highlight existent relationships among gene annotations, independently from gene expression levelsRules 1 and 2 identify associations between annotations from different sources like the relationship between the KEGG term *cell cycle* and the Gene Ontology term *cell cycle*, or the less obvious one between the KEGG term *purine metabolism* and the GO term *cytoplasm.* Rules 3 and 4 confirm the strong relationship between promoters $FHL1$ and $RAP1$. Rule 5 highlight a relationship between genes cited in a scientific article (which presents a review of the essential yeast genes) with the phenotype *inviable.* Rules 6 and 7 are two examples of a special group of rules that simply reflect the hierarchical structure of the bio-ontologies used. They represent an important proportion of rules that either depict the hierarchical links or represent identical relationships at different levels of abstraction corresponding to the hierarchically linked annotations. Such kind of rules can be filtered during a post-processing phase without information loss.

Table 3. Associations *annotations* ⇒ *annotations.*

Rule	Antecedent	Consequent	supp.	conf.
1	path:sce04111 (cell cycle)	go:0007049 (cell cycle)	67	78
2	path:sce00190 (purine metabolism)	go:0005737 (cytoplasm)	49	91
3	pr:FHL1	pr:RAP1	114	86
4	pr:RAP1	pr:FHL1	114	61
5	pmid:16155567	phenot:inviable	168	93
6	go:0016192 (vesicle transport)	go:0006810 (transport)	171	100
7	go:0005739 (mitochondrion)	go:0005737 (cytoplasm)	532	100

6.2 Execution times and memory usage

These experiments were conducted to assess the applicability of GenMiner to very large dense biological datasets and to compare its results with Apriori based approaches. They were performed on a PC with one Pentium IV processor running at 2 GHz and 1 GB of RAM was allocated for the execution of GenMiner and implementations of Apriori based approaches. We tested several implementations of Apriori based approaches (Apriori, FP-Growth, Eclat, LCM, DCI, etc.). Execution times presented in Table 4 are these of Borgelt's implementation[4] described in [4] that is globally the most efficient for mining ARs (and not only frequent itemsets). We can see in this table that execution times of GenMiner and the Apriori implementation are similar for *minsupp* between 0.02 (2%) and 0.007 (0.7%). However, executions of Apriori based approaches for lower *minsupp* values were interrupted as they required more than 1 GB of RAM. GenMiner could be run for *minsupp* = 0.003, i.e. rules supported by at least 7 data lines (genes), but the execution for *minsupp* = 0.002 was interrupted as more than 1 GB of RAM was required.

Table 4. Execution times and number of rules (*minconf*=0.3).

	GenMiner		Apriori	
minsupp (#)	Time (s)	Number of rules	Time (s)	Number of rules
0.020 (50)	10	10 028	5	65 312
0.015 (37)	21	28 492	16	325 482
0.010 (25)	72	110 989	76	3 605 486
0.009 (22)	101	147 966	110	6 115 366
0.008 (19)	187	230 255	182	12 138 561
0.007 (17)	289	315 090	264	21 507 415
0.006 (14)	673	542 746	Out of Memory	-
0.005 (12)	1 415	824 518	Out of Memory	-
0.004 (9)	5 353	1 675 811	Out of Memory	-
0.003 (7)	18 424	2 883 710	Out of Memory	-
0.002 (4)	Out of Memory	-	Out of Memory	-

[4] Available at `http://fimi.cs.helsinki.fi/`.

We can see that for *minsupp* between 0.02 (2%) and 0.007 (0.7%), the Informative Basis is from 6 to 68 times smaller than the set of all ARs, that contains up to more than 21 millions of rules. However, the number of ARs in the Informative Basis is important for low *minsupp* values and it cannot be manually explored without tools to select subsets of ARs.

6.3 GenMiner scalability

Experimental results presented in Table 5 were conducted to evaluate execution times and memory usage of GenMiner when the *minsupp* and *minconf* thresholds vary. Three series of executions were run for *minconf* equals to 0.9 (90%), 0.5 (50%) and 0.3 (30%). For each serie, *minsupp* was varied between 0.02 (2%) and 0.002 (0.2%). As in the previous experiment, GenMiner could not be run for *minsupp* lower than 0.003, independently from the *minconf* value. We can also see that the longest executions, for *minsupp* equals to 0.003, took from 4 to 5 hours depending on the *minconf* value.

Table 5. Execution times of GenMiner (in seconds).

minsupp (#)	minconf = 0.9	minconf = 0.5	minconf = 0.3
0.020 (50)	9.18	10.40	10.88
0.015 (37)	16.47	19.58	21.21
0.010 (25)	47.50	63.47	72.63
0.009 (22)	65.10	87.68	101.49
0.008 (19)	118.78	162.17	187.33
0.007 (17)	182.27	249.60	289.41
0.006 (14)	435.41	595.23	673.27
0.005 (12)	974.14	1 274.57	1 415.38
0.004 (9)	4 065.05	4 937.74	5 353.63
0.003 (7)	14 163.02	17 412.65	18 424.72
0.002 (4)	Out of Memory	Out of Memory	Out of Memory

7 Discussion and conclusion

GenMiner was developed for mining association rules from very large dense datasets containing both gene expression data and annotations. Contrarily to most approaches for gene expression interpretation, as well *expression-based* as *knowledge-based*, in which biological information and gene expression profiles are incorporated in an independent manner, with GenMiner both data sources are integrated in a single framework.

GenMiner implements a new discretization algorithm, called NorDi, that was designed for processing data generated by gene expression technologies in the case of independent biological conditions. Experiments conducted on the Eisen *et al.* dataset show that its results are relevant. However, the discretization issue is delicate when using data mining methods such as ARM. We thus propose to use several discretization scenarios, analyzing the pertinence of obtained results against expected results, to validate the discretization method. As pointed out in [18]: "The robustness of biological conclusions made by using microarray analysis should be routinely assessed by examining the validity of the conclusions by using a range of threshold parameters issued from different discretization algorithms". Unfortunately, to our knowledge no discretization algorithm, specially designed for time process data, can integrate the time variable without an important loss of temporal information.

GenMiner also integrates the Close algorithm [19] developed to extract ARs from dense and correlated data. Close is based on the frequent closed itemsets framework that allows to reduces both the search space and the number of dataset accesses, and thus the memory usage, for dense and correlated data. It extracts a minimal set of non-redundant ARs called Informative Basis [19] in order to reduce the number of extracted ARs and improve the result's relevance. In this basis, all information is summarized in a minimal number of ARs, each rule bringing as much information as possible, without information loss.

To evaluate the efficiency and scalability of GenMiner, it was run on a dataset combining the Eisen *et al.* gene expression data [10] and annotations of these genes. Experimental results show that GenMiner can deal with such large datasets and that its memory usage, as well as the number of ARs generated, are significantly smaller than these of Apriori based approaches. Moreover, ARs extracted by GenMiner are not constrained in their form and can contain both gene annotations and gene expression profiles in the antecedent and the consequent. The analyze of these ARs has shown important relationships supported by recent biological literature. These results show that GenMiner is a promising tool for finding meaningful relationships between gene expression patterns and gene annotations. Furthermore, it enables the integration of thousands of gene annotations from heterogenous sources of information with related gene expression data. This is an essential feature as the integration of different types of biological information is indispensable to fully understand the underlying biological processes. In addition, qualitative variables such as gender, tissue and age could easily be integrated in order to extract ARs among these features and gene expression patterns. In the future, we plan to integrate in GenMiner tools to filter, select, compare and visualize ARs during the interpretation phase to simplify these manipulations.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In *Proc. VLDB conf.*, pp. 478–499 (1994)
2. Altman, R., Raychaudhuri, S.: Whole-genome expression analysis: challenges beyond clustering. *Current Opinion Structural Biology*, **11**, pp. 340–347 (2001)
3. Bera, A., Jarque, C.: Efficient tests for normality, homoscedasticity and serial independence of regression residuals: Monte carlo evidence. *Economics Letters*, **7**, pp. 313–318 (1981)

4. Borgelt, C.: Recursion Pruning for the Apriori Algorithm. In *Proc. FIMI workshop* (2004)
5. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. In *Proc. ACM SIGMOD conf.*, pp. 255–264 (1997)
6. Carmona-Saez, P., Chagoyen, M., Rodriguez, A., Trelles, O., Carazo, J., Pascual-Montano, A.: Integrated analyis of gene expression by association rules discovery. *BMC Bioinformatics*, **7**:54 (2006)
7. Creighton, C. Hanansh, S.: Mining gene expression databases for association rules. *Bioinformatics*, **19**, pp. 79–86 (2003)
8. Cristofor, L., Simovici, D.A. Generating an informative cover for association rules. In *Proc. ICDM conf.*, pp. 597–600 (2002)
9. DeRisi, J., Iyer, L., Brown, V.: Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **278**, pp. 680–686 (1997)
10. Eisen, M., Spellman, P., Brown, P., Botsein, D.: Cluster analysis and display of genome wide expression patterns. In *Proc. National Academy of Sciences USA*, vol. 95, pp. 14863–8 (1998)
11. Georgi, E., Richter, L., Ruckert, U., Kramer, S.: Analyzing microarray data using quantitative association rules. *Bioinformatics*, **21**, pp. 123–129 (2005)
12. Grubbs, F.: Procedures for detecting outlying observations in samples. *Technometrics*, **11**, pp. 1–21 (1969)
13. Lilliefors, H., On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, **62** (1967)
14. Lopez, F.J., Blanco, A., Garcia, F., Cano, C., Marin, A.: Fuzzy association rules for biological data analysis: a case study on yeast. *BMC Bioinformatics*, **9**:107 (2008)
15. Martinez, R., Collard, M.: Extracted knowledge: Interpretation in mining biological data, a survey. *Int. J. of Computer Science and Applications*, **1**, pp. 1–21 (2007)
16. Martinez, R., Pasquier, N., Pasquier, C.: GenMiner: Mining informative association rules from genomic data. In *Proc. IEEE BIBM conf.*, pp. 15–22 (2007)
17. NIST: *e-Handbook of Statistical Methods*. SEMATECH. http://www.itl.nist.gov/-div898/handbook/ (2007)
18. Pan, K., Lih, C., Cohen, N.: Effects of threshold choice on biological conclusions reched during analysis of gene expression by dna microarrays. *National Academy of Sciences PNAS*, **102**, pp. 8961–8965 (2005)
19. Pasquier, N., Taouil, R., Bastide, Y., Stumme, G., Lakhal, L.: Generating a condensed representation for association rules. *Journal of Intelligent Information Systems*, **24**(1), pp. 29–60 (2005)
20. Shatkay, H., Edwards, S., Wilbur, W., Boguski, M. Genes, themes, microarrays: using information retrieval for large-scale gene analysis. In *Proc. ISMB conf.*, pp. 340–347 (2000)
21. Tuzhilin, A., Adomavicius, G.: Handling very large numbers of association rules in the analysis of microarray data. In *Proc. SIGKDD conf.*, pp. 396–404 (2002)
22. Yang, I., Chen, E., Hasseman, J., Liang, W., Frank, B., Sharov, V., Quackenbush, J.: Within the fold: assesing differential expression measures and reproducibility in microarray assays. *Genome Biology*, **3**:11 (2002)
23. Zhao, Y., McIntosh, K., Rudra, D., Schawalder, S., Shore, D., Warner, J.: Fine-structure analysis of ribosomal protein gene transcription. *Molecular Cellular Biology*, **26**(13), pp. 4853–62 (2006)