Mining Association Rule Bases from Integrated Genomic Data and Annotations

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

(1) I3S Laboratory
UNSA/CNRS UMR-6070, 2000 route des Lucioles, 06903 Valbonne, France. {rmartine, pasquier}@i3s.unice.fr

(2) Institute of Developmental Biology and Cancer
UNSA/CNRS UMR-6543, Parc Valrose, 06108 Nice, France. claude.pasquier@unice.fr

Keywords: data mining, gene expression data, gene annotations, minimal non-redundant association rules.

Abstract. During the last decade, several clustering and association rule mining techniques have been applied to identify groups of co-regulated genes in gene expression data. Nowadays, integrating biological knowledge and gene expression data into a single framework has become a major challenge to improve the relevance of mined patterns and simplify their interpretation by the biologists. The GenMiner approach was developed for mining association rules showing gene groups that are both co-expressed (sharing similar expression profiles) and co-annotated (sharing the same annotations such as function, regulatory mechanism, etc.) 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. Compared with classical Apriori based approaches, GenMiner improves the extraction applicability for these datasets and reduces the number of association rules by suppressing redundant rules that are uninformative and useless. We present a new Java implementation of GenMiner and experimental results obtained from microarray datasets with integrated biological knowledge (bio-ontologies, descriptions of regulation pathways and literature). These 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. One of the great potentials of these technologies is that generated data contain hidden information about the biological processes that govern cell behavior. Nowadays, one of the main goals of these technologies is to discover this hidden information to achieve biological knowledge. In other words, we want to interpret gene expression technology results 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.

In order to process the interpretation step in an automatic or semi-automatic way, the bioinformatics community faces an ever-increasing volume of sources of biological information that are: Information on microarray experiments (spotted probes, experimental design, data processing protocols, etc.); Molecular databases (GenBank, Embl, Unigene, etc.); Semantic sources as thesaurus, ontologies, taxonomies or semantic networks (UMLS, GO, etc.); Gene expression databases (GEO, Arrayexpress, etc.); Bibliographic databases (Medline, Biosis, etc.); Gene/protein related specific sources (KEGG, OMIM, etc.).

Several approaches dealing with the interpretation problem have recently been reported. These approaches can be classified in three axes [19]: *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, these approaches 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 [12]. 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 [25].

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 [20]. In this paper, we present a new Java implementation of the GenMiner approach and new experimental results on the biological significance of extracted rules, the applicability and scalability of GenMiner 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 brief 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* ⇒ *gene1*[↑]
2. *annotation* ⇒ *gene2*[↑]
3. *annotation* ⇒ *gene1*[↑], *gene2*[↑]
4. *annotation*, *gene1*[↑] ⇒ *gene2*[↑]
5. *annotation*, *gene2*[↑] ⇒ *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 [8, 26, 13]. These applications aimed at discovering frequent gene patterns among a subset of biological conditions. These patterns were represented as ARs such as: *gene1*[↓] ⇒ *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 [13]. 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, 26]. 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 [22].

The GenMiner approach 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. Gene annotations can be integrated from any source of biological information (semantic sources, bibiographic databases, gene expression databases, etc.). It uses the novel NorDi method for discretizing gene expression measures and generate gene expression profiles. It takes advantage of the Close [23] algorithm that can efficiently generate low support and high confidence non-redundant association rules. When data is dense or correlated, such as genomic data, Close reduces both execution times and memory space usage compared with Apriori, thus enabling the analysis of large datasets. Furthermore, it improves the result's relevance by extracting a minimal set of rules containing only non-redundant ARs, 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 The 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 anno-

tations are discovered. The actual implementation of GenMiner does not integrate graphical visualization tools and complementary programs must be used to manipulate the resulting file.

4.1 *NorDi algorithm*

The *Normal Discretization* (NorDi) algorithm was developed to improve gene expression measures discretization into items. This phase is essential to extract relevant ARs. 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 [14]. 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 [21] and Lilliefors [17] normality tests. Then, we calculate the over-expressed, Ot, and under-expressed, Ut, cutoff thresholds using the $z - score$ methodology [27] 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 [23] 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 [9]. 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 [9]. Experiments conducted on benchmark datasets show that the rule number reduction

factor varies from 5 to 400 according to data density and correlation [23]. 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.* [11]. This dataset contains expression measures of 2 465 yeast genes under 79 biological conditions extracted from a collection of four independent microarray studies about the *Saccharomyces cerevisiae* during several biological processes:

- Cell cycle experiments [24] (variables alpha1 to alpha18, elu1 to elu14 and cdc15-1 to cdc15-15).
- Sporulation experiments [7] (variables spo1 to spo6, spo5-1 to spo5-3 and ndt80-1 to ndt80-2).
- Temperature shock experiments [11] (variables heat1 to heat6, dtt1 to dtt4 and cold1 to cold4).
- Diauxic shift [10] (variables diauxic1 to diauxic7).

The resulting dataset[1] is a matrix of 2 465 lines representing yeast genes and 737 columns representing expression levels (discretized gene expression measures) and gene annotations. Each line contains expression levels 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).

5.1 *Gene expression measures*

The microarray technology used is spotted cDNA chips obtained by two color fluorochromes with distinct emission spectra Cy3 and Cy5. The Eisen *et al.* dataset contains the expression levels of 2 465 open reading frames of the yeast for 79 biological conditions. This dataset was pretreated by taking the log_2 ratios (to consider cellular inductions and repressions in a numerically equal way) and applying the imputation algorithm of k-nearest neighbors [18] in order to treat the missing values (1.9% of the total).

The studied biological processes of the yeast, that are independent from each other, are supposed to be normally distributed. Furthermore, each sample condition is supposed to be "almost" normally distributed, i.e. S_j for every $j = 1, 2, ..., 79$. In this manner, the Eisen dataset accomplished the four NorDi assumptions and discretized gene expression values were calculated using NorDi algorithm at a 95% confidence level.

5.2 *Gene annotations*

We used the *Saccharomyces cerevisiae database* (SGD) nomenclature for naming the yeast genes. All yeast genes were annotated using five sources of biological information:

- the Yeast-specific cut-down version of Gene Ontology (GO) semantic source of information (known as GOSlim), containing annotations from biological processes, molecular functions and cellular annotations,
- the bibliographic source of information from SGD's manually curated PubMed/Medline papers,
- the gene/protein related specific database KEGG [15] containing the metabolic pathways in which each gene is involved,
- the phenotype information of given yeast genes extracted from SGD's file,
- the information of transcriptional regulators that bind to promoter regions, these data were reported in [16]. This information was used to annotate yeast genes whose promoter regions were bound by at least one transcriptor regulator (with a *p-value* threshold of 0.0005).

All gene annotations were taken as boolean variables, i.e. $i \in \{0, 1\}$, indicating if an annotation pertains, $i = 1$, or not, $i = 0$, to a given gene. The prefixes *go:*, *path:*, *pmid:*, *pr:*, *phenot:* are used to identify Gene Ontology terms, KEGG pathways, Pubmed identifiers, promoters and phenotypes respectively.

[1] Available at `http://bioinfo.unice.fr/publications/genminer_article`.

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 [2] 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.

6.1 *Biological interpretation of extracted association rules*

In the following, we describe selected meaningful biological rules, grouped according to their form, to show the potential of the GenMiner approach. ARs with the form *annotations* $\Rightarrow$ *expression levels* show groups of genes associated with the same annotations that are over-expressed or under-expressed in a set of biological conditions. Selected ARs with this form extracted with GenMiner for *minsupp*=0.003 (at least 7 lines) and *minconf*=30% are presented in Tab. 1. Supports are given in number of transactions and confidences in percentages. Rules 1 to 11 are relative to the shock by high temperature experiment and show known relationships described in [28]. Rules 12 to 15 reflect the main metabolic changes associated to the diauxic shift, manually identified in [10], and are similar to ARs presented in [6].

Table 1: Associations *annotations* $\Rightarrow$ *expression levels*.

Rule	Antecedent	Consequent	Supp. (#)	Conf. (%)
1	go:0006412 go:0005840	heat3$\downarrow$	103	51
2	go:0005840 go:0005198	heat3$\downarrow$	96	56
3	go:0006412 go:0042254	heat3$\downarrow$	22	61
4	go:0005840 go:0003723	heat3$\downarrow$	12	57
5	go:0005737 go:0042254 go:0005198	heat3$\downarrow$	20	67
6	go:0042254 go:0005840 go:0005198	heat4$\downarrow$	15	52
7	go:0006412 go:0006996 go:0005198	heat3$\downarrow$	30	65
8	path:sce03010	heat3$\downarrow$	97	74
9	path:sce03010	heat4$\downarrow$	69	53
10	pr:RAP1 pr:FHL1	heat3$\downarrow$	71	62
11	pmid:5542014 pmid:9649613 pmid:3533916	heat3$\downarrow$	12	100
12	path:sce00190	diauxic6$\uparrow$	17	31
13	path:sce00190	diauxic7$\uparrow$	18	33
14	path:sce00020	diauxic5$\uparrow$ diauxic6$\uparrow$ diauxic7$\uparrow$	8	32
15	path:sce00630	diauxic7$\uparrow$	7	55

ARs with the form *expression levels* $\Rightarrow$ *annotations* show groups of genes that are over-expressed or under-expressed in a set of biological conditions and have the corresponding gene annotations. Selected ARs with this form, extracted with GenMiner are presented in Tab. 2. These rules show information related to the elutriation process (rules 1 to 5), the sporulation experiment (rules 6 to 11), the heat shock process (rules 12 to 16), the cold shock experiment (rules 17 and 18) and the diauxic shift process (rules 19 and 20) reported in the corresponding biological literature.

ARs with the form *annotations* $\Rightarrow$ *annotations* contain gene annotations both in the antecedent and consequent. They highlight existent relationships among gene annotations, independently from gene expression levels. Selected ARs with this form extracted with GenMiner are presented in Tab. 3. These ARs show relationships between KEGG pathways and GO terms (rules 1 and 2), between promoters (rules 3 and 4), between promoters and GO terms (rules 5), between scientific articles and phenotypes (rule 6) and between GO terms (rules 7 to 10).

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

[2] Available at `http://bioinfo.unice.fr/publications/genminer_article`.

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

Rule	Antecedent	Consequent	Supp (#)	Conf (%)
1	elu5↑ elu6↑ elu7↑	go:0006412	26	87
2	elu4↑	go:0006412	39	52
3	elu4↑ elu5↑ elu6↑	go:0006412	17	81
4	elu6↑ elu7↑	go:0006412	33	69
5	elu2↓ elu3↓	go:0006996	12	55
6	spo4↓ spo5↓ spo6↓	go:0005975	12	52
7	spo3↓ spo4↓ spo5↓	go:0005975	12	48
8	spo3↓	go:0006412	42	52
9	spo2↓ spo3↓	go:0006412	27	57
10	spo4↑ spo5↑ spo6↑	go:0006996	26	43
11	spo3↓ spo4↓ spo5↓	path:sce00010	13	52
12	heat3↓ heat5↓ heat6↓	go:0006412	16	76
13	heat3↓ heat4↓ heat5↓	go:0006412	35	88
14	heat2↓	go:0006996	41	69
15	heat2↓	go:0042254	39	66
16	heat3↑ heat4↑ heat5↑	go:0006950	10	45
17	cold3↓ cold4↓	go:0006412	15	79
18	cold4↓	go:0006412	71	73
19	diauxic6↑ diauxic7↑	go:0006091	23	47
20	diauxic6↓ diauxic7↓	go:0006412	21	66

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

Rule	Antecedent	Consequent	Supp. (#)	Conf. (%)
1	path:sce04111	go:0007049	67	78
2	path:sce00190	go:0005737	49	91
3	pr:FHL1	pr:RAP1	114	86
4	pr:RAP1	pr:FHL1	114	61
5	pr:RAP1, pr:FHL1	go:0005737 go:0006412 go:0005840	93	82
6	pmid:16155567	phenot:inviable	168	93
7	go:0005737, go:0045333	go:0006091	56	100
8	go:0016192	go:0006810	171	100
9	go:0005739	go:0005737	532	100
10	go:0005740	go:0005737 go:0005739	165	100

with one Pentium IV processor running at 2 GHz and 1 GO 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 Tab. 4 are these of Borgelt's implementation[3] 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 when *minsupp* varies 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 GO 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 GO of RAM was required.

Experimental results presented in Tab. 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.

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

Table 4: Execution times and memory usage (*minconf*=0.3).

minsupp (#)	GenMiner (s)	Apriori (s)
0.020 (50)	10	5
0.015 (37)	21	16
0.010 (25)	72	76
0.009 (22)	101	110
0.008 (19)	187	182
0.007 (17)	289	264
0.006 (14)	673	Out of memory
0.005 (12)	1 415	Out of memory
0.004 (9)	5 353	Out of memory
0.003 (7)	18 424	Out of memory
0.002 (4)	Out of memory	Out of memory

Table 5: Scalability of GenMiner.

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

6.3 *Number of association rules*

The number of ARs in the Informative Basis extracted by GenMiner and the total number of ARs extracted by Apriori based approaches are presented in Tab. 6. For this experiment, *minconf* was fixed to 0.3 (30%) and *minsupp* was varied between 0.02 (2%) and 0.003 (0.3%). 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. Examining the basis, we note that an important proportion of rules contain similar information at different levels of precision. These rules either contain annotations linked in the bio-ontology hierarchies or are identical except that they contain different annotations that are hierarchically related in the bio-ontology. This is related to the presence of very general annotations, that are common to numerous genes and are thus present in an important proportion of rules, among GO terms for exemple. In order to improve the relevance of extracted ARs, only ARs with the most specific of these annotations should be conserved as they represent the most precise knowledge. This problem can also be addressed during the data selection and preparation phase by suppressing the most general annotations.

7 **Discussion and conclusion**

The GenMiner approach was developed for mining association rules from very large dense datasets containing both gene expression data and annotations. It is a co-clustering technique that extracts intrinsic associations among gene expression levels and annotations. It is a bi-clustering technique that discovers patterns describing genes co-expressed in a subset of biological conditions. 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

Table 6: Number of association rules (*minconf*=0.3).

minsupp (#)	Informative Basis	All association rules
0.020 (50)	10 028	65 312
0.015 (37)	28 492	325 482
0.010 (25)	110 989	3 605 486
0.009 (22)	147 966	6 115 366
0.008 (19)	230 255	12 138 561
0.007 (17)	315 090	21 507 415
0.006 (14)	542 746	Out of memory
0.005 (12)	824 518	Out of memory
0.004 (9)	1 675 811	Out of memory
0.003 (7)	2 883 710	Out of memory

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 [22]: "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 [23] developed to extract ARs from dense and correlated data. With such data, classical ARM algorithms, based on the Apriori approach [1], have high execution times and memory usage [5]. They can thus only extract ARs with high support and confidence values, that is concerning large groups of genes. Moreover, the number of ARs extracted by these algorithms from such data is most often very important and many of these rules are redundant as they bring the same information [9]. This is an important drawback for ARs interpretation by the analysts as redundant rules sometimes represent the majority of extracted ARs. 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 [23] 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.

Gene expression data are highly correlated, due to the numerous groups of genes that are co-expressed in different biological conditions, and when gene annotations are integrated the average number of items per gene becomes important. As GenMiner integrates the Close algorithm, it can efficiently extract meaningful associations between gene expression profiles and gene annotations, even for small groups of genes, from such data. To evaluate its efficiency and scalability, it was run on a dataset combining the Eisen *et al.* gene expression data [11] and annotations of these genes (GO, KEGG, phenotype information, transcriptional regulators information and information of selected articles). 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. and Srikant, R. (1994). Fast algorithms for mining association rules. In *Proc. VLDB conf.*, pages 478–499.

[2] Altman, R. and Raychaudhuri, S. (2001). Whole-genome expression analysis: challenges beyond clustering. *Current Opinion Structural Biology*, **11**, 340–347.

[3] Bera, A. and Jarque, C. (1981). Efficient tests for normality, homoscedasticity and serial independence of regression residuals: Monte carlo evidence. *Economics Letters*, **7**, 313–318.

[4] Borgelt, C. (2004). Recursion Pruning for the Apriori Algorithm. In *FIMI workshop*.

[5] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In *Proc. ACM SIGMOD conf.*, pages 255–264.

[6] Carmona-Saez, P., Chagoyen, M., Rodriguez, A., Trelles, O., Carazo, J., and Pascual-Montano, A. (2006). Integrated analyis of gene expression by association rules discovery. *BMC Bioinformatics*, **7**(54).

[7] Chu, S., DeRisi, J., Eisen, M., Mullholland, J., Botstein, D., and Brown, P. O. e. a. (1998). The transcriptional program of sporulation in budding yeast. *Science*, **282**, 699–705.

[8] Creighton, C. and Hanansh, S. (2003). Mining gene expression databases for association rules. *Bioinformatics*, **19**, 79–86.

[9] Cristofor, L. and Simovici, D. A. (2002). Generating an informative cover for association rules. In *Proc. ICDM conf.*, pages 597–600.

[10] DeRisi, J., Iyer, L., and Brown, V. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **278**, 680–686.

[11] Eisen, M., Spellman, P., Brown, P., and Botsein, D. (1998). Cluster analysis and display of genome wide expression patterns. In *Proc. National Academy of Sciences USA*, volume 95, pages 14863–8.

[12] Gasch, A. and Eisen, M. (2002). Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, **3**, 1–22.

[13] Georgi, E., Richter, L., Ruckert, U., and Kramer, S. (2005). Analyzing microarray data using quantitative association rules. *Bioinformatics*, **21**, 123–129.

[14] Grubbs, F. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, **11**, 1–21.

[15] Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K., Itoh, M., Kawashima, S., Katayama, T., Araki, M., , and Hirakawa, M. (2006). From genomics to chemical genomics: new developments in kegg. *Nucleic Acids Res.*, **34**, D354–357.

[16] Lee, T., Rinaldi, N., Robert, F., Odom, D., Bar-Joseph, Z., Gerber, G., Hannett, N., Harbison, C., Thompson, C., Simon, I., Zeitlinger, J., Jennings, G., Murray, H., Gordon, B., and Young, R. (2002). Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, **298**(5594), 799–804.

[17] Lilliefors, H. (1967). On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, **62**.

[18] Little, R. and Rubin, D. (2002). *Statistical Analysis with Missing Data*. Wiley and Sons, 2 edition.

[19] Martinez, R. and Collard, M. (2007). Extracted knowledge: Interpretation in mining biological data, a survey. *International Journal of Computer Science and Applications*, **1**, 1–21.

[20] Martinez, R., Pasquier, N., Pasquier, C. (2007). GenMiner: Mining informative association rules from genomic data. In *Proc. of the IEEE BIBM conf.*, pages 15–22.

[21] NIST (2007). *e-Handbook of Statistical Methods*. SEMATECH. http://www.itl.nist.gov/div898/handbook/.

[22] Pan, K., Lih, C., and Cohen, N. (2005). Effects of threshold choice on biological conclusions reched during analysis of gene expression by dna microarrays. *National Academy of Sciences PNAS*, **102**, 8961–8965.

[23] Pasquier, N., Taouil, R., Bastide, Y., Stumme, G., and Lakhal, L. (2005). Generating a condensed representation for association rules. *Journal of Intelligent Information Systems*, **24**(1), 29–60.

[24] Paul, T. S., Sherlock, G., Michael, D., Zhang, Q., Iyer, V., Anders, K., Eisen, M., O. Brown, P., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, **9**, 3273–97.

[25] Shatkay, H., Edwards, S., W., W., and M., B. (2000). Genes, themes, microarrays: using information retrieval for large-scale gene analysis. In *Proc. ISMB conf.*, pages 340–347.

[26] Tuzhilin, A. and Adomavicius, G. (2002). Handling very large numbers of association rules in the analysis of microarray data. In *Proc. SIGKDD conf.*, pages 396–404.

[27] Yang, I., Chen, E., Hasseman, J., Liang, W., Frank, B., Sharov, V., and Quackenbush, J. (2002). Within the fold: assesing differential expression measures and reproducibility in microarray assays. *Genome Biology*, **3**, 11.

[28] Zhao, Y., McIntosh, K., Rudra, D., Schawalder, S., Shore, D., and Warner, J. (2006). Fine-structure analysis of ribosomal protein gene transcription. *Molecular Cellular Biology*, **26**(13), 4853–62.