Contribution to Gene Expression Data Analysis by Means of Set Pattern Mining

Ruggero G. Pensa[1], Jérémy Besson[1,2],
Céline Robardet[3], and Jean-François Boulicaut[1]

[1] INSA Lyon, LIRIS CNRS UMR 5205,
F-69621 Villeurbanne cedex, France
{Ruggero.Pensa, Jeremy.Besson, Jean-Francois.Boulicaut}@insa-lyon.fr
[2] UMR INRA/INSERM 1235,
F-69372 Lyon cedex 08, France
[3] INSA Lyon, PRISMA,
F-69621 Villeurbanne cedex, France
Celine.Robardet@insa-lyon.fr

Abstract. One of the exciting scientific challenges in functional genomics concerns the discovery of biologically relevant patterns from gene expression data. For instance, it is extremely useful to provide putative synexpression groups or transcription modules to molecular biologists. We propose a methodology that has been proved useful in real cases. It is described as a prototypical KDD scenario which starts from raw expression data selection until useful patterns are delivered. It has been validated on real data sets. Our conceptual contribution is (a) to emphasize how to take the most from recent progress in constraint-based mining of set patterns, and (b) to propose a generic approach for gene expression data enrichment. Doing so, we survey our algorithmic breakthrough which has been the core of our contribution to the IST FET cInQ project.

1 Introduction

Thanks to a huge research and technological effort, one of the challenges for molecular biologists is to discover knowledge from data generated at very high throughput. Indeed, different techniques (including microarray [1] and SAGE [2]) enable to study the simultaneous expression of (tens of) thousands of genes in various biological situations or experiments. Such data can be seen as expression matrices in which the expression level of genes (the attributes or columns) are recorded in various biological situations (the objects or rows). A toy example of a gene expression matrix is in Fig. 1a. Exploratory data mining techniques are needed that can, roughly speaking, be considered as the search for interesting bi-sets, i.e., sets of biological situations and sets of genes which are associated in some way. Indeed, it is interesting to look for groups of co-regulated genes, also known as synexpression groups [3], for which a reasonable assumption is that they participate in a common function within the cell. The association between

J.-F. Boulicaut et al. (Eds.): Constraint-Based Mining, LNAI 3848, pp. 328–347, 2005.

a set of co-regulated genes and the set of biological situations that gives rise to this co-regulation is called a transcription module and their discovery is a major goal in functional genomics since it paves the way to a better understanding of gene regulation networks.

The use of hierarchical clustering (see, e.g., [4]) is quite popular among practitioners. Genes are grouped together according to similar expression profiles. The same can be done on biological situations. Thanks to the appreciated vizualization component introduced with [4], biologists can identify sets of genes that are co-regulated in some sets of situations. Global patterns like partitions provide an a priori interesting "global picture" of similarity structures in the whole data. The results of most of the clustering algorithms are non overlapping groups of genes. It means that a given gene belongs to one and only one cluster while we already know genes which clearly participate to various biological functions. Furthermore, their heuristic nature can lead to different results. Co-clustering or bi-clustering techniques do not change fundamentally the problem: the benefit comes from an assessment of the association between both partitions, i.e., sets of genes and sets of situations but we still get non overlapping partitions based on a local optimization process [5,6]. In other terms, we get a global pattern which capture some more or less expected phenomena.

A complementary approach is to look for collections of local patterns in the gene expression data. Heuristic statistical methods have been proposed to identify a priori interesting bi-sets from raw numerical data (see, e.g., [7,8]). A promising direction of research is to consider complete constraint-based mining techniques on boolean gene expression data sets. The completeness assumption means that every pattern from the pattern language which satisfies the defined constraints has to be returned (e.g., every frequent set, every closed set) and, in this case, we use non heuristic methods. In these data sets, boolean gene expression properties are encoded, e.g., over-expression, strong variation, co-regulation. We get boolean data sets which are also called in some application domains transactional data sets.

Let $\mathcal{O}$ denotes a set of objects or rows (e.g., biological situations) and $\mathcal{P}$ denotes a set of properties or columns (e.g., genes). For instance, expression properties can be encoded into a boolean matrix $\mathbf{r} \subseteq \mathcal{O} \times \mathcal{P}$. $(o_i, g_j) \in \mathbf{r}$ denotes that gene j has the encoded expression property in situation i. For deriving a boolean context from raw gene expression data, we generally apply discretization operators that, depending of the chosen expression property, compute thresholds from which it is possible to decide between wether the true or the false value must be assigned. On our toy example in Fig. 1, $\mathcal{O} = \{h_1, h_2, h_3, h_4, d_1, d_2, d_3, d_4\}$ and $\mathcal{P} = \{g_1, g_2, \ldots, g_8\}$. A value "1" for a biological situation and a gene means that the gene is up (greater than $|t|$) or down (lower than $-|t|$) regulated in this situation. Using threshold $t = 0.4$ for Fig. 1a leads to the boolean matrix in Fig. 1b.

Local pattern discovery tasks can be performed when searching for putative synexpression groups or transcription modules. To compute synexpression groups, we can extract the so-called frequent itemsets (sets of genes) from the de-

rived boolean contexts. Notice that sets of genes that are frequently co-regulated can be post-processed into association rules [9,10].

In our boolean toy example (Fig. 1), the genes from $\{g_2, g_5\}$ are in relation with $\{h_1, h_2, h_4, d_3\}$.

The relevancy of the extracted patterns can be improved by considering the frequent closed itemsets which are the frequent maximal sets of genes whose encoded expression properties are shared by a same set of biological situations. For instance, $\{g_2, g_4, g_5, g_7\}$ is a closed itemset because g_4 and g_7 are the other genes which are in relation with each element from $\{h_1, h_2, h_4, d_3\}$. Formally these local patterns are the set components of formal concepts [11]. A formal concept is a maximal set of genes associated to a maximal set of situations, e.g., $(\{h_1, h_2, h_4, d_3\}, \{g_2, g_4, g_5, g_7\})$ in the data from Fig. 1b. Such patterns can indeed be considered as putative transcription modules [12,13,14].

	Genes							
Sit.	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
h_1	0.1	-0.5	0.3	0.7	3	0.2	6.1	-0.1
h_2	0.2	-0.6	0.4	0.5	1.2	0.1	4.2	-0.5
h_3	0.2	-0.3	0.9	0.1	0.4	5	0.5	-0.1
h_4	2.1	-0.7	-0.2	0.6	4.1	0.3	5.3	-0.3
d_1	0.2	-0.8	0.2	-0.5	0.4	6.3	0.4	-0.6
d_2	2.3	-0.4	0.1	0.7	-5.1	0.4	5.8	-0.2
d_3	1.2	-0.6	0.1	0.6	3.6	0.3	6.2	-0.1
d_4	1.6	0.1	0.3	0.6	2.8	0.4	4.9	0.1

(a)

	Genes							
Sit.	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
h_1	0	1	0	1	1	0	1	0
h_2	0	1	0	1	1	0	1	1
h_3	0	0	1	0	0	1	1	0
h_4	1	1	0	1	1	0	1	0
d_1	0	1	0	1	0	1	0	1
d_2	1	0	0	1	1	0	1	0
d_3	1	1	0	1	1	0	1	0
d_4	1	0	0	1	1	0	1	0

(b)

Fig. 1. A gene expression matrix (a) and a derived boolean context (b)

This paper is a methodological paper. It abstracts our practice in several real-life gene expression data analysis projects to disseminate a promising practice within the scientific community. Our methodology covers the whole KDD process and not just the mining phase. Starting from raw gene expression data, it supports the analysis and the discovery of transcription modules via a constraint-based bi-set mining approach from computed boolean data sets. The generic process is described within the framework of inductive databases, i.e., each step of the process can be formalized as a query on data and/or patterns that satisfy some constraints [15,16]. It leads us to a formalization of boolean gene expression data enrichment. We already experimented a couple of practical instances of this approach and it has turned to be crucial for increasing the biological relevancy of the extracted patterns.

Details about each step of the method and the algorithms or solvers which have been developed in the context of the CInQ project have been already published. Therefore, we avoid most of the technical details, just emphasizing the main algorithmic principles and the methodological added-value of our "in silico" approach for transcription module discovery.

The main publications which are associated to this method are:

- Preprocessing numerical gene expression data to encode boolean gene expression properties [9,17].
- Using AC-Miner [18] for computing frequent closed sets and interesting association rules between boolean gene expression properties [9];
- Computing putative transcription modules as formal concepts with a AC-Miner-like algorithm [12,13];
- Using D-Miner for computing putative transcription modules as formal concepts under monotonic constraints [19,14];
- Boolean gene expression data enrichment [20,14].
- Post-processing putative transcription modules [21].

2 Classical Approaches in Gene Expression Data Analysis

From a technical point of view, traditional gene expression data analysis is based on similarities between expression profiles. The expression profile of a gene, is the sequence of its expression values in different biological situations. For example, in the drosophila melanogaster data set (see [22]), the expression levels of about 4 000 genes are measured for a number of time points during the drosophila life cycle. Studying the expression profile of each gene, it is possible to observe the behavior of such a gene during the whole life cycle. A typical analysis task, is to compare expression profiles two by two, noticing the principal differences and similarities between two expression profiles. This is clearly not feasible when thousands of genes are involved. An important contribution to gene expression data analysis is due to Eisen et al. (see [4]). They consider a technique based on hierarchical clustering which enables to compare expression profiles of thousands of genes simultaneously. Genes sharing similar expression profiles are grouped together in the same subtree structure of the resulting dendrogram. This supports the analysis for finding putatively cooperating genes. Dually, biological situations can be processed with the same clustering algorithm. The resulting structure enables to associate groups of genes to groups of situations in which these genes are co-expressed. For instance, in Fig. 2, we can observe dendrograms for the data set in Fig. 1a. Such an approach can be used for identifying some patterns like putative transcription modules.

One major problem concerning such a technique is that searching transcription modules is not that simple. For instance, most of the traditional clustering algorithms, including [4], provide non overlapping (bi-)clusters: one gene (resp. one situation) is associated to only one cluster. Moreover, similarities are computed by considering the whole collection of gene or situation vectors. From the biological point of view, we know that a gene can participate in various biological functions, in different cells and environmental conditions, and at the same time, it is not influenced by the whole set of situations. Therefore, traditional unsupervised clustering techniques are not really oriented to the discovery of transcription modules and synexpression groups, even though they remain useful for exploratory analysis of gene expression data sets.

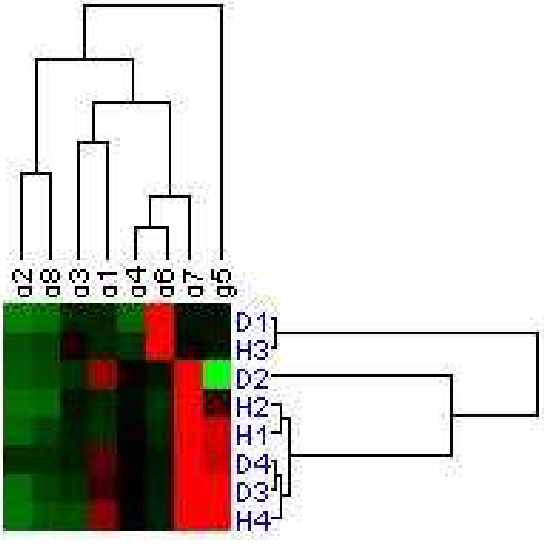


Fig. 2. Dendrograms obtained after a hierarchical clustering on the data from Fig. 1a

A solution can come from local patterns, i.e., patterns which hold in part of the data. For example, the signature algorithm (see [7,8]) enables to find some putative transcription modules starting from a set of known genes. These techniques however heuristically compute some a priori interesting patterns. It makes sense to look at the recent breakthrough concerning complete algorithms for local set pattern mining.

3 A KDD Approach for Gene Expression Analysis

We introduce our KDD-based methodology for gene expression data analysis. It exploits our results in several domains like constraint-based data mining, preprocessing of gene expression raw data, and postprocessing of pattern collections. It has been proved useful for supporting the search of putative transcription modules.

We decided to work on Boolean gene expression data sets instead of numerical data sets. Boolean gene expression data sets encode boolean gene expression properties. The main advantage is that beside encoding techniques based on raw value discretizations, an expert knowledge can be used for assessing the encoding (e.g., checking that computed property is consistent with some available knowledge). A second advantage is that we can add other boolean properties of genes within the same context (e.g., the fact that a gene is or not associated to a given transcription factor). The main drawback is that many different point of views can be considered on a phenomenon like over-expression and the proposed encoding techniques have parameters (i.e., thresholds) that can not be

fixed easily. Of course, if the boolean data do not capture well the chosen property, then most of the patterns extracted from it will be irrelevant. Therefore, we have designed a method for fixing encoding method parameters. Once boolean gene expression data sets are available, we have considered the extraction of set patterns like closed sets, association rules, and formal concepts. The number of discovered patterns can be huge and it happens that the computation turns to be untractable. To increase both the relevancy and the tractability of this task, we have considered user-defined constraints which can be pushed into the extraction phase. The final step consists in post-processing the extracted patterns by deducing new information on data, and exploiting it for further mining tasks. We have also designed a technique to visualize similarities between extracted patterns by means of a user-friendly graphical representation. This post-processing has been proved useful to support pattern interpretation by biologists.

3.1 Pre-processing

We assume that raw expression data, i.e., a function that assigns a real expression value to each couple $(o, g) \in \mathcal{O} \times \mathcal{P}$ is available and that some tasks have been selected by the molecular biologists. A typical example concerns the discovery of putative transcription modules that involve at least a given set of genes that are already known to be co-regulated in a given class of biological situations, e.g., diabetic ones.

Due to the lack of space, we do not consider the typical data manipulation statements that are needed, e.g., for data normalization, data cleaning, gene and/or biological situation selection according to some background knowledge (e.g., removing housekeeping genes from consideration).

Discretization. This step concerns gene expression property encoding and is obviously crucial. The simplest case concerns the computation of a boolean matrix $\mathbf{r} \subset \mathcal{O} \times \mathcal{P}$ which encode a simple expression property for each gene in each situation, e.g., over-expression[1]. Different algorithms can be applied and parameters like thresholds have to be be chosen. For instance, [9] introduces three techniques for encoding gene over-expression:

- "Mid-Ranged". The highest and lowest expression values in a biological situation are identified for each gene and the mid-range value is defined. Then, for a given gene, all expression values that are strictly above the mid-range value give rise to value 1, 0 otherwise.
- "Max - X% Max". The cut off is fixed w.r.t. the maximal expression value observed for each gene. From this value, we deduce a percentage X of this value. All expression values that are greater than the (100 - X)% of the Max value give rise to value 1, 0 otherwise.

[1] Not only it is possible to consider several attributes per gene for one property, e.g., one for "strong overexpression" and one for "suspected strong-expression" but also one can decide to encode various properties per gene like "up-regulation" and "down-regulation".

– "X% Max". For each gene, we consider the biological situations in which its level of expression is in X% of the highest values. These genes are assigned to value 1, 0 for the others.

These techniques give different points of view on the over-expression biological phenomenon and it is unclear which one performs better. The impact of the chosen technique and the used parameters on both the quantity and the relevancy of the extracted patterns is crucial. For instance, the density of the discretized data depends on the discretization parameters and the cardinalities of the resulting sets (collections of itemsets, association rules or formal concepts) can be very different. We clearly need a method to evaluate different boolean encoding (different techniques and/or various parameters) of the same raw data and thus a framework to support user decision about the discretization from which the mining process can start. Our thesis is that a good discretization might preserve some properties that can be already observed from raw data. Let E denote a gene expression matrix. Let $\{Bin_i,\ i = 1..b\}$ denote a set of different discretization operators and $\{\mathbf{r}_i,\ i = 1..b\}$ a set of boolean contexts obtained by applying these operators, i.e. $\forall i = 1..b,\ \mathbf{r}_i = Bin_i(E)$. Let $S : \mathbb{R}^{n,m} \longmapsto \mathbb{R}$ denote an evaluation function that measure the quality of the discretization of a gene expression matrix. We say that a boolean context $\mathbf{r}_i$ is more valid than another context $\mathbf{r}_j$ w.r.t the S measure if $S(\mathbf{r}_i) > S(\mathbf{r}_j)$. In [17], we studied an original method for such an evaluation. We suggest to compare the similarity between the dendrogram generated by a hierarchical clustering algorithm (e.g., [4]) applied to the raw expression data and the dendrograms generated by the same algorithm applied to each derived boolean matrix. Given a gene expression matrix E and two derived boolean contexts $\mathbf{r}_i$ and $\mathbf{r}_j$, we can choose the discretization that leads to the dendrogram which is the most similar to the one built on E. The idea is that a discretization that preserves the expression profile similarities is considered more relevant. A simple measure of similarity between dendrograms has been studied and experimentally validated on various gene expression data sets.

Let $\mathcal{O} = \{o_1, \ldots, o_n\}$ denote the set of n objects. Let T denote a binary tree built on $\mathcal{O}$. Let $\mathcal{L} = \{l_1, \ldots, l_n\}$ denote the set of n leaves of T associated to $\mathcal{O}$ for which, $\forall i \in [1 \ldots n]$, $l_i \equiv o_i$. Let $\mathcal{B} = \{b_1 \ldots b_{n-1}\}$ denote the set of the $n-1$ internal nodes of T generated by a hierarchical clustering algorithm starting from $\mathcal{L}$. By construction, we consider $b_{n-1} = r$, where r denotes the root of T. Let us define the two sets:

$$\delta(b_i) = \{b_j \in \mathcal{B} \mid b_j \textit{ is a descendent of } b_i\}$$
$$\tau(b_i) = \{l_j \in \mathcal{L} \mid l_j \textit{ is a descendent of } b_i\}.$$

We want to measure the similarity between a tree T and a reference tree T_{ref} built on the same set of objects $\mathcal{O}$. For each node b_i of T, we define the following score (denoted S_B and called **BScore**):

$$S_B(b_i, T_{ref}) = \sum_{b_j \in \delta(b_i)} a_j$$

$$a_j = \begin{cases} \frac{1}{|\tau(b_j)|}, & if \quad \exists b_k \in T_{ref} \mid \tau(b_j) = \tau(b_k) \\ 0, & otherwise \end{cases} \quad (1)$$

To obtain the similarity score of T w.r.t. T_{ref} (denoted S_T and called **TScore**), we consider the **BScore** value on the root, i.e.:

$$S_T(T, T_{ref}) = S_B(r, T_{ref}) \quad (2)$$

As usually, it is interesting to normalize the measure to get a score between 0 (for a tree which is totally different from the reference) and 1 (for a tree which is equal to the reference). In the **TScore** measure, since its max value depends on the tree morphology, we can normalize by $S_T(T_{ref}, T_{ref})$:

$$\overline{S_T}(T, T_{ref}) = \frac{S_T(T, T_{ref})}{S_T(T_{ref}, T_{ref})} \quad (3)$$

$\overline{S_T}(T, T_{ref}) = 0$ means that T is totally different from T_{ref}, i.e., there are no matching nodes between T and T_{ref}. Indeed, $\overline{S_T}(T, T_{ref}) = 1$ means that T is totally similar to T_{ref}, i.e., every node in T matches with a node in T_{ref}. Given two trees T_1 and T_2 and a reference T_{ref}, if $\overline{S_T}(T_1, T_{ref}) < \overline{S_T}(T_2, T_{ref})$, then T_2 is said to be more similar to T_{ref} than T_1 according to **TScore**.

We can apply this technique to both the situation and gene trees. Indeed, we obtain two different similarity scores. To consider a unique **TScore**, we can compute the mean between the two scores. However, in order to force the general similarity score to be equal to 0 when at least one of the two scores is equal to 0, we prefer to use the square root of the product of the two similarity scores:

$$\overline{S_{AT}}(T_g, T_s, T_{ref}) = \sqrt{\overline{S_T}(T_g, T_{ref}) \cdot \overline{S_T}(T_s, T_{ref})}$$

where T_g and T_s denote respectively the dendrograms for genes and situations.

Let us apply this technique to the gene expression matrix in Fig. 1a. We decide to evaluate the set of discretization operators Bin_i, where $i = 1..10$, and such that values in the matrix whose absolute value is greater than $i \times 10^{-1}$ are coded with a "1" in the boolean matrix, while the other expression values are coded with a "0" (e.g., for $i = 5$, the threshold is set to $5 \times 10^{-1} = 0,5$). Therefore, we can obtain ten different boolean contexts and we process each of them with the same hierarchical clustering algorithm. Then we compare the resulting gene and situation dendrograms with those obtained by clustering the original real expression matrix from Fig. 1a. The results are presented in Fig. 3. We can observe that for a threshold of 0.4 the square root of the product between the gene similarity score and the situation similarity score is maximal. If we discretize the raw data from Fig. 1a with such a threshold, we obtain the boolean context given in Fig. 1b.

Boolean Gene Expression Data Enrichment. We can mine boolean gene expression matrices for frequent sets of genes and/or situations, association rules between genes and/or situations, formal concepts, etc. In the following, we focus

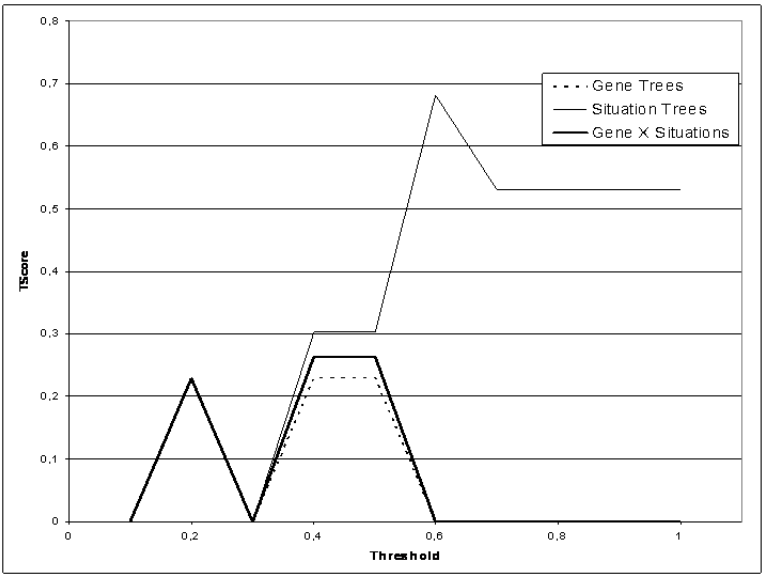


Fig. 3. Similarity scores w.r.t. threshold values

on mining phases that compute formal concepts. When the extractions are feasible, many patterns are discovered (up to several millions) while only a few of them are interesting. It is however extremely hard to decide of the interestingness characteristics a priori. We now propose a powerful approach for improving the relevancy of the extracted formal concepts by boolean data enrichment. It can be done a priori with some complementary information related to genes and/or situations. For instance, we can add information about the known functions of genes as it is recorded in various sources like Gene Ontology [23]. Other information can be considered like the associated transcription factors. A simple way to encode this kind of knowledge consists in adding a row to $\mathbf{r}$ for each gene property. Dually, we can add some properties to the situations vectors. For instance, if we know the class of a group of situations (e.g. diabetic vs. non diabetic individuals) we can add a column to $\mathbf{r}$. We can also add boolean properties about, e.g., cell type or environmental features. Enrichment of boolean data can be performed by more or less trivial data manipulation queries from various bioinformatics databases. $\mathbf{r}' \subset \mathcal{O}' \times \mathcal{P}'$ will denote the relation of the enriched boolean context.

In Fig. 4a, we add three gene properties tf_1, tf_2 and tf_3. A value "1" for a gene and a property means that this gene has the property. For instance, tf_1 could mean that the gene is regulated by a given transcription factor. Dually, in Fig. 4b, we consider two classes of situations, namely c_H and c_D. A value "1" for a situation and a class means that this situation belongs to the class but this could be interpreted in terms of situation properties as well. For instance, c_D (resp. c_H) could mean whether biological situations are diabetic (resp. healthy) ones. In

the data in Fig. 4b, a formal concept like $(\{d_2, d_3, d_4, tf_1, tf_3\}, \{g_1, g_4, g_5, g_7, c_D\})$ informs us about a "maximal rectangle of true values" that involves four genes, regulated by two transcription factors tf_1 and tf_2 in three situations that are of class c_D. This could reveal sets of genes that are co-regulated in diabetic situations but not in healthy ones. We will discuss later how iterative enrichment enables to improve the relevancy of the extracted patterns.

	Genes							
Sit.	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
h_1	0	1	0	1	1	0	1	0
h_2	0	1	0	1	1	0	1	1
h_3	0	0	1	0	0	1	1	0
h_4	1	1	0	1	1	0	1	0
d_1	0	1	0	1	0	1	0	1
d_2	1	0	0	1	1	0	1	0
d_3	1	1	0	1	1	0	1	0
d_4	1	0	0	1	1	0	1	0
tf_1	1	0	0	1	1	0	1	1
tf_2	0	1	0	1	1	0	1	0
tf_3	1	1	0	1	1	0	1	0

(a)

	Genes									
Sit.	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	c_H	c_D
h_1	0	1	0	1	1	0	1	0	1	0
h_2	0	1	0	1	1	0	1	1	1	0
h_3	0	0	1	0	0	1	1	0	1	0
h_4	1	1	0	1	1	0	1	0	1	0
d_1	0	1	0	1	0	1	0	1	0	1
d_2	1	0	0	1	1	0	1	0	0	1
d_3	1	1	0	1	1	0	1	0	0	1
d_4	1	0	0	1	1	0	1	0	0	1
tf_1	1	0	0	1	1	0	1	1	1	1
tf_2	0	1	0	1	1	0	1	0	1	1
tf_3	1	1	0	1	1	0	1	0	1	1

(b)

Fig. 4. Two examples of enriched boolean microarray contexts

3.2 Pattern Extraction

Constraint-Based Extraction of Formal Concepts. We consider here formal concept extraction from eventually enriched boolean contexts.

Definition 1 (Bi-set). *A bi-set (T, G) is a couple of sets such that $T \subseteq \mathcal{O}$ and $G \subseteq \mathcal{P}$. We often use the term rectangle to denote bi-sets: clearly, a bi-set defines a combinatorial rectangle in the boolean matrix, i.e., up to permutations over rows and columns.*

Definition 2 (1-rectangle). *A bi-set (T, G) is a 1-rectangle in $\mathbf{r}$ (constraint $\mathcal{C}_{1R}(T, G)$) iff $\forall t \in T$ and $\forall g \in G$ then $(t, g) \in \mathbf{r}$. When a bi-set (T, G) is not a 1-rectangle, we say that it contains 0 values.*

Definition 3 (Formal concept). *A bi-set (T, G) is a concept in $\mathbf{r}$ iff (T, G) is a 1-rectangle and $\forall T' \subseteq \mathcal{O} \backslash T$, $T' \neq \emptyset$, $(T \cup T', G)$ is not a 1-rectangle and $\forall G' \subseteq \mathcal{P} \backslash G$, $G' \neq \emptyset$, $(T, G \cup G')$ is not a 1-rectangle. A concept (T, G) is thus a maximal 1-rectangle. We denote the associated constraint as $\mathcal{C}_{Concept}(T, G, \mathbf{r})$.*

Thanks to the mathematical properties of formal concepts [11] (e.g., each formal concept is built on closed sets for both dimensions), a first approach to extract the complete collection of formal concepts consists in computing the whole collection of closed itemsets and their associated objectsets. This can be

done by slightly modifying existing algorithms for extracting closed sets (see, e.g., [24] for a survey). Indeed, in some applications, we can use frequent closed set mining with a 0 frequency threshold. In our biological contexts, the number of genes (items) is very large (up to thousands) and it is often impossible to use these algorithms to perform this task. However, in many gene expression data sets, the number of biological situations, i.e., of objects, is quite small. As a result, a simple transposition of the matrix solves the problem [12,13]. When the number of objects increases, this technique is however no more tractable.

To overcome this problem (i.e., working on boolean gene expression matrices whose none of the two dimensions is small enough), we have been considering the definition and the use of constraints which enable to reduce both the search space and the solution space. It is indeed possible to consider formal concepts whose one set component is large enough [25]. We have studied the possibility to enforce constraints on both components.

Definition 4 (Constraints on formal concepts). *Assume that* (T, G) *is a formal concept in* $\mathbf{r}$.
Minimal size constraints:
(T, G) *satisfies the constraint* $\mathcal{C}_t(\mathbf{r}, \sigma_1, T)$ *iff* $|T| \geq \sigma_1$.
(T, G) *satisfies the constraint* $\mathcal{C}_g(\mathbf{r}, \sigma_2, G)$ *iff* $|G| \geq \sigma_2$.
Syntactical constraints:
(T, G) *satisfies the constraint* $\mathcal{C}_{Inclusion}(\mathbf{r}, X, G)$ *iff* $X \subseteq G$.
(T, G) *satisfies the constraint* $\mathcal{C}_{Inclusion}(\mathbf{r}, X, T)$ *iff* $X \subseteq T$.
Minimal area constraint:
(T, G) *satisfies the constraint* $\mathcal{C}_{area}(\mathbf{r}, \sigma, (T, G))$ *iff* $|T| \times |G| \geq \sigma$.

These constraints are quite obvious to interpret for end-users, here molecular biologists. Properties of constraints have been studied extensively and monotonicity properties can lead to major optimizations.

Definition 5 (Monotonic and anti-monotonic constraints). *Let* $\preceq$ *be a partial order on a set* $\mathcal{S}$. *A constraint* $\mathcal{C}$ *on* $\mathcal{S}$ *is said monotonic (resp. anti-monotonic) w.r.t.* $\preceq$ *iff* $\forall\ s_1,\ s_2 \in \mathcal{S}$, *if* $s_1 \preceq s_2$ *and* $\mathcal{C}(s_1)$ *(resp.* $\mathcal{C}(s_2)$*) is satisfied then* $\mathcal{C}(s_2)$ *(resp.* $\mathcal{C}(s_1)$*) is also satisfied.*

Let us now define our partial order on bi-sets.

Definition 6 (Partial order). *The partial order* $\preceq$ *on bi-sets is defined as follows:* $(T_1, G_1) \preceq (T_2, G_2)$ *iff* $T_1 \subseteq T_2$ *and* $G_1 \subseteq G_2$.

Given this partial order, the constraints introduced in Definition 4 are monotonic. We have proposed the D-Miner algorithm for computing every formal concept which satisfies a given monotonic constraint [19]. It generates the formal concept candidates w.r.t. the chosen partial order such that the defined constraints can be pushed deeply into the extraction phase. More precisely, D-Miner first computes a list H of 0-rectangles composed of an object and the items which are not in relation with it. Then, it builds a tree whose root is the bi-set $(\mathcal{O},\mathcal{P})$. Each node (T, G) is recursively split using an element (a, b) of H,

such that $a \cap T \neq \emptyset$ and $b \cap G \neq \emptyset$, until H is empty: the left child is $(T \setminus a, G)$ whereas the right one is $(T, G \setminus b)$. Another constraint denoted $\mathcal{C}_{left}$ has to be pushed to avoid the computation of sub-concepts such that each leaf of the tree is finally a formal concept. Constraint $\mathcal{C}_{left}$ is used to check that all the children of $(T \setminus a, G)$ contain at least one item in b. To illustrate this process, we consider in Fig 5 the extraction of formal concepts (T, G) from $\mathbf{r}_2$ (see Table 1) with an area larger than 4, i.e., satisfying $\mathcal{C}_{area}(\mathbf{r}_2, 4, ((T, G))$. Underlined bi-sets are the leaves which do not satisfy either $\mathcal{C}_{left}$ or $\mathcal{C}_{area}$.

Table 1. Context $\mathbf{r}_2$ (left) and its corresponding H list

	g_1	g_2	g_3
t_1	0	0	1
t_2	1	0	1
t_3	0	0	1
t_4	1	0	1

(t_1, g_1g_2)
(t_2, g_2)
(t_3, g_1g_2)
(t_4, g_2)

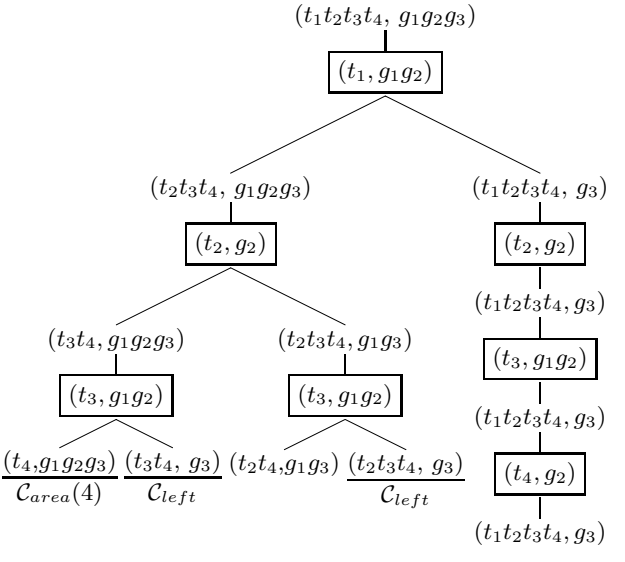


Fig. 5. Formal concept computation on $\mathbf{r}_2$

In $\mathbf{r}_2$, we have only two formal concepts with an area greater or equal to 4: $(\{t_2, t_4\}, \{g_1, g_3)\}$ and $(\{t_1, t_2, t_3, t_4\}, \{g_3\})$.

It is quite useful to use these constraints in enriched contexts. For instance we can search for potentially interesting bi-sets that involve a minimum number of genes (more than γ) to ensure that the extracted formal concepts are not due to noise. They must also be made of enough (say ≥ 3) biological situations from

$\mathcal{D} = \{d_1, .., d_4\}$ and few (say ≤ 1) biological situations of $\mathcal{H} = \{h_1, .., h_4\}$ or vice versa. In other terms, the potentially interesting bi-sets (T, G) are formal concepts that verify the following constraints as well:

$$(|T_H| \geq 3 \ \wedge \ |T_D| \leq 1 \ \wedge \ |G| \geq \gamma) \tag{4}$$
$$\vee (|T_D| \geq 3 \ \wedge \ |T_H| \leq 1 \ \wedge \ |G| \geq \gamma) \tag{5}$$

where T_H and T_D are the subsets of T that concern respectively the biological situations from $\mathcal{H}$ (e.g., healthy individuals) and the ones from $\mathcal{D}$ (e.g., diabetic patients). Notice that this constraint which is the disjunction of Equation 4 and Equation 5 is a disjunction of a conjunction of monotonic and anti-monotonic constraints on $2^{\mathcal{O}}$ and $2^{\mathcal{P}}$. Using D-MINER, we push the monotonic ones, i.e.:

$$q_1 : \mathcal{C}_{Concept}(T, G, \mathbf{r}) \wedge \mathcal{C}_t(\mathbf{r}, 3, T_H) \wedge \mathcal{C}_g(\mathbf{r}, \gamma, G).$$

$$q_2 : \mathcal{C}_{Concept}(T, G, \mathbf{r}) \wedge \mathcal{C}_t(\mathbf{r}, 3, T_D) \wedge \mathcal{C}_g(\mathbf{r}, \gamma, G).$$

Applying the previously defined constraints to the data set in Fig. 4a (using D-MINER, then post-processing the pattern collection to check non monotonic ones), we get the two following formal concepts:

$(\{h_1, h_2, h_4, d_3, tf_2, tf_3\}, \{g_2, g_4, g_5, g_7\})$ for q_1 with $\gamma = 1$
$(\{h_4, d_2, d_3, d_4, tf_1, tf_3\}, \{g_1, g_4, g_5, g_7\})$ for q_2 with $\gamma = 1$

In this example, g_1 and g_2 are putative interesting genes, each of them characterizes only one class of situations represented in the data set. Moreover, all these genes are regulated by the same transcription factor tf_3. This could mean that they are involved in the same biological function of the cell.

Another way to proceed, is to consider the class properties c_H and c_D that we added into the boolean context in Fig. 4b. We can easily perform an extraction of formal concepts under the following constraints:

$$q_3 : \mathcal{C}_{Concept}(T, G, \mathbf{r}) \wedge \mathcal{C}_{Inclusion}(\mathbf{r}, c_H, G) \wedge \mathcal{C}_g(\mathbf{r}, \gamma, G) \wedge \mathcal{C}_t(\mathbf{r}, \gamma\prime, T_H).$$

With $\gamma = 3$ and $\gamma\prime = 3$, two formal concepts satisfy such a constraint:

$(\{h_1, h_2, h_4, tf_2, tf_3\}, \{g_2, g_4, g_5, g_7, c_H\})$
$(\{h_1, h_2, h_4, tf_1, tf_2, tf_3\}, \{g_4, g_5, g_7, c_H\})$

Then, we can ask for a second collection with all the formal concepts (T, G) such that the class attribute c_D is included in G:

$$q_4 : \mathcal{C}_{Concept}(T, G, \mathbf{r}) \wedge \mathcal{C}_{Inclusion}(\mathbf{r}, c_D, G) \wedge \mathcal{C}_g(\mathbf{r}, \gamma, G) \wedge \mathcal{C}_t(\mathbf{r}, \gamma\prime, T_D).$$

The formal concepts resulting from the execution of the second query, with $\gamma = 3$ and $\gamma\prime = 3$, are:

$(\{d_2, d_3, d_4, tf_1, tf_3\}, \{g_1, g_4, g_5, g_7, c_D\})$
$(\{d_2, d_3, d_4, tf_1, tf_2, tf_3\}, \{g_4, g_5, g_7, c_D\})$

Notice that gene g_2 appears only in the first class of patterns, while g_1 appears only in the second class. In other words, using queries q_3 and q_4 we focus on the same putative interesting genes (and the same situations) obtained with queries q_1 and q_2. The difference is that we use here only monotonic constraints that can be efficiently pushed by D-Miner.

Let us compare these results with a classical gene expression data analysis approach. If we observe the dendrogram obtained by applying a hierarchical clustering algorithm to the raw data set (see Fig. 2), we can notice that only gene g_4 and g_7 are grouped together. Other genes belonging to the pattern extracted before are relatively far (w.r.t. the height of the branches), from g_4 and g_7. It is interesting to notice that genes g_2 and g_5 are considered as not belonging to the same cluster of g_4 and g_7, even for a relatively "high" cut.

3.3 Post-processing and Iteration

Formal concept extraction, even constraint-based mining, can produce large numbers of patterns, especially in the first iteration of the KDD process, i.e., when very few information can be used to further constrain the bi-sets to be delivered. Notice also that, from a practical perspective, not all the specified constraints can be pushed into the mining algorithm: some of these constraints have to be checked in a post-processing phase. For instance, we can exploit non monotonic constraints defined in Equation 4 and Equation 5 (i.e., $|T_D| \leq 1$ and $|T_H| \leq 1$) that can not be pushed within D-Miner.

KDD processes are clearly complex iterative processes for which obtained results can give rise to new ideas for more relevant constraint-based mining phases (inductive queries) or data manipulations. When a collection of patterns has been computed, it can be used for deriving new boolean properties. In particular, let us assume that we got two sets of patterns that can characterize two classes of genes and, dually, two classes of situations. Therefore, we can define two new class properties related to genes and their dual class properties related to situations. The boolean context $\mathbf{r}'$ can then be extended towards $\mathbf{r}'' \subset \mathcal{O}'' \times \mathcal{P}''$. Considering our running example, we can associate a new property p_H (resp. p_D) for the genes not belonging to the formal concepts which are returned by q_4 (resp. q_3). It leads to the enriched boolean context given in Fig. 6. New constraints on the classes can be used for the next mining phase. New set size constraints can be defined as well. As a result, a new iteration will provide a new collection of formal concepts which is more relevant according to the user current task. Each time a collection of formal concepts is available, we can decide either to analyze it by hand, e.g., studying each genes separately, or looking for new boolean data enrichment and revisited constraints for the next iteration. Also, genes to which we can associate new functions, are the best candidates to be chosen for iterating the KDD process and take advantage of larger seed sets of genes.

In any cases, at the end of the process, we have a set of putative interesting genes and a set of putative interesting situations. Iterations can be stopped when we have a set of putative interesting genes that can be easily studied by hand. A priori knowledge is very important at this point. In our running example, we did

	Genes									
Sit.	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	c_H	c_D
h_1	0	1	0	1	1	0	1	0	1	0
h_2	0	1	0	1	1	0	1	1	1	0
h_3	0	0	1	0	0	1	1	0	1	0
h_4	1	1	0	1	1	0	1	0	1	0
d_1	0	1	0	1	0	1	0	1	0	1
d_2	1	0	0	1	1	0	1	0	0	1
d_3	1	1	0	1	1	0	1	0	0	1
d_4	1	0	0	1	1	0	1	0	0	1
tf_1	1	0	0	1	1	0	1	1	1	1
tf_2	0	1	0	1	1	0	1	0	1	1
tf_3	1	1	0	1	1	0	1	0	1	1
p_H	0	1	1	0	0	1	0	1	1	1
p_D	1	0	1	0	0	1	0	1	1	1

Fig. 6. A new enriched boolean context

not introduced any additional information about known genes, i.e., genes that are already known as being directly involved in the analyzed problem. However, studying interactions between genes whose functions are already identified, and new putative interesting genes discovered by means of our methodology, can help biologists to suggest putative functions for new genes.

An other important problem concerns the postprocessing of formal concept collections. We need efficient techniques to support the subjective search for interesting patterns. In [21], we introduced an "Eisen-like" visualization technique, that enables to group similar formal concepts by means of a hierarchial clustering algorithm. We defined a distance between two formal concepts and then a distance between two clusters of formal concepts. For the first step, we use the symmetrical set difference Δ between two sets S_i and S_j: $S_i \Delta S_j = S_i \cup S_j \setminus S_i \cap S_j$.

Definition 7. *(Distance between two formal concepts) Assume that* $c_i = (T_i, G_i)$ *and* $c_j = (T_j, G_j)$ *are two formal concepts, the distance* d *between* c_i *and* c_j *is defined as*

$$d(c_i, c_j) = \frac{1}{2}\frac{|T_i \Delta\, T_j|}{|T_i \cup T_j|} + \frac{1}{2}\frac{|G_i \Delta\, G_j|}{|G_i \cup G_j|} \tag{6}$$

where $|S|$ *denotes the cardinality of* S*.*

To compute the distance between two clusters of formal concepts, we associate a pseudo-concept to each cluster. A pseudo-concept is a unique representation for all the formal concepts within a cluster. It is composed of two fuzzy sets, one set of genes and one set of biological situations: a degree of membership α_i (a real number between 0 and 1) is associated to each element e_i of the referential set (i.e., $\mathcal{O}$ or $\mathcal{P}$). Value 0 (resp. value 1) denotes that the element does not belong (resp. belongs) to the set.

Definition 8. *(Pseudo-concept) A pseudo-concept is denoted by* $(T', G', N) \subseteq \mathcal{O}' \times \mathcal{P}' \times \mathbb{N}$ *with* $\mathcal{O}' = \mathcal{O} \times [0;1]$ *and* $\mathcal{P}' = \mathcal{P} \times [0;1]$. *The weight* N *denotes the number of formal concepts represented by the pseudo-concept.*

It is possible to generalize the distance d for measuring the similarity between pseudo-concepts. The classical fuzzy set operators (indexed with f) are used:

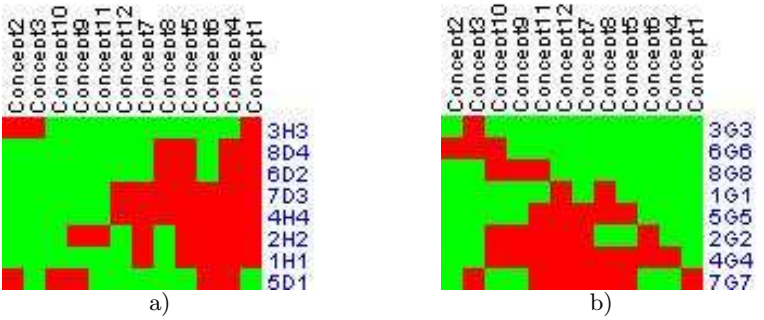
$$S_1 \cup_f S_2 = \{(o, \max\{\alpha_1, \alpha_2\}) \mid o \in \mathcal{O},\ (o, \alpha_1) \in S_1 \text{ and } (o, \alpha_2) \in S_2\}$$
$$S_1 \cap_f S_2 = \{(o, \min\{\alpha_1, \alpha_2\}) \mid o \in \mathcal{O},\ (o, \alpha_1) \in S_1 \text{ and } (o, \alpha_2) \in S_2\}$$
$$S_1 \setminus_f S_2 = \{(o, \alpha_1 - \alpha_2) \mid o \in \mathcal{O},\ (o, \alpha_1) \in S_1 \text{ and } (o, \alpha_2) \in S_2\}$$
$$|S_1|_f = \sum_{o \in \mathcal{O}} \alpha,\ (o, \alpha) \in S_1$$

Thanks to this approach, we can reduce the impact of concept multiplication in noisy boolean data and support the post-processing of tens of thousands of formal concepts.

Example 1. In the boolean context from Fig. 1b, twelve formal concepts (with at least one gene and one situation) can be extracted:

Concept1 : $(\{h_1, h_2, h_3, h_4, d_2, d_3, d_4\}, \{g_7\})$
Concept2 : $(\{h_3, d_1\}, \{g_8\})$
Concept3 : $(\{h_3\}, \{g_3, g_6, g_}7)$
Concept4 : $(\{h_1, h_2, h_4, d_1, d_2, d_3, d_4\}, \{g_4\})$
Concept5 : $(\{h_1, h_2, h_4, d_2, d_3, d_4\}, \{g_4, g_5, g_7\})$
Concept6 : $(\{h_1, h_2, h_4, d_1, d_3\}, \{g_2, g_4\})$
Concept7 : $(\{h_1, h_2, h_4, d_3\}, \{g_2, g_4, g_5, g_7\})$
Concept8 : $(\{h_4, d_2, d_3, d_4\}, \{g_1, g_4, g_5, g_7\})$
Concept9 : $(\{h_2, d_1\}, \{g_2, g_4, g_8\})$
Concept10 : $(\{d_1\}, \{g_2, g_4, g_6, g_8\})$
Concept11 : $(\{h_2\}, \{g_2, g_4, g_5, g_7, g_8\})$
Concept12 : $(\{h_4, d_3\}, \{g_1, g_2, g_4, g_5, g_7\})$

By applying a hierarchical clustering associated to a simple visualization technique (using Treeview from [4]), we provide the pictures (rectangles) in Fig. 7.



a) b)

Fig. 7. Situation (a) and gene (b) rectangles resulting of a hierarchical clustering of concepts

A dark-colored cell in the rectangle means that the related gene (or situation) is present in the related formal concept. Notice that groups of similar formal concepts can be identified by looking for relatively dense red zones either in the situation rectangle or in the gene rectangle.

Thanks to this graphical approach and in contrast to the dendrograms obtained with a simpler approach (see Fig. 2), we can notice a strong correlation between genes involved in previously extracted patterns (g_1,g_2,g_4,g_5,g_7), and a disposition of situations which is more consistent w.r.t. their class values.

4 Biological Validations

The method and the techniques we have considered in the previous sections have been applied with success to different real-life data sets and problems. In some experiments, we have considered well-documented gene expression data sets (i.e., containing accurate biological knowledge) to validate the methods by rediscovery (see, e.g., [17,20]). We have also applied this approach to original gene expression data sets from which new biological knowledge has been extracted. For instance, in [9], the authors have used closed sets (and more precisely some association rules derived from them) to derive biologically relevant knowledge from human SAGE data [26]. The selection on the SAGE data concerns the expression level of 822 genes measured in 74 biological situations (cancerous and not cancerous tissues belonging to various human organs). After an over-expression encoding by means of the "Max - X% Max" method (see Section 3.1), homogeneous closed sets of genes have been studied in detail and, among others, it enabled to suggest a putative function for an EST-encoded protein.

A successful application of constraint-based extraction of formal concepts (see Section 3.2) to an original microarray data set has been described in [14]. Each DNA microarray contains the RNA expression level of about 20 000 genes before and after a perfusion of insulin in human skeletal muscle [27]. It is a nice example of gene expression data enrichment: the considered context encodes information about different gene properties that are biologically relevant (expression level for healthy people and for diabetic patients, regulation by known transcription factors). The set $\mathcal{O}$ of situations was thus partitioned into the set $\mathcal{H}$, the set $\mathcal{D}$ and a set of transcription factors $\mathcal{F}$. After a typical data preprocessing (e.g., removing genes whose none of their transcription factors are known), the final boolean context contained 104 objects (94 transcription factors and 10 biological situations, 5 for healthy individuals and 5 for diabetic patients) and 304 genes. Even though a formal concept discovery from such a boolean context has turned out to be very hard, pushing monotonic constraints has enabled to get significant results. Potentially interesting bi-sets (T, G) were considered as the formal concepts satisfying the following constraints:

$$(|T_H| \geq 4 \wedge |T_D| \leq 2 \wedge |G| \geq \gamma) \tag{7}$$

$$\vee\, (|T_D| \geq 4 \wedge |T_H| \leq 2 \wedge |G| \geq \gamma) \tag{8}$$

The authors have considered in details one of the extracted formal concept which is particularly interesting as it contains genes which are either up-regulated or down-regulated after insulin stimulation, this being based on the homology of their promotor DNA sequences (associated transcription factors) [14]. This is indeed a kind of results we hardly get with classical approaches like [4].

5 Conclusion

We have considered data mining methods and tools which can support knowledge discovery from gene expression data. A prototypical KDD scenario which takes the most from recent progress in constraint-based set pattern mining has been described. Importantly, some of our results on algorithms have been indeed motivated by the gene expression data mining task. For instance, it has motivated the design of D-Miner because of the failure of available algorithms for closed set mining on biological data sets of interest. Concrete instances of this scenario have been considered in several real-life gene expression data analysis problems, including the whole human SAGE [13] data and the microrray data described in [27]. We better understand the crucial issues of boolean gene expression property encoding. Also, boolean gene expression data enrichment appears to be a powerful technique for supporting the iterative search of relevant patterns w.r.t. a given analysis task. The perspectives of this research include the need for fault-tolerant formal concept mining, i.e., strong associations which might however accept some exceptions, but also the multiple uses of the extracted patterns. For instance, local patterns like formal concepts could be used in complementarity with (bi-)clustering techniques, typically to support accurate (bi-)cluster characterization.

Acknowledgements. Most of the results reported in this paper have been obtained during the cInQ IST-2000-26469 European project funded by the European Union. Our research on methodological approaches to gene expression data analysis is also partially funded by CNRS ACI MD46 Bingo. Finally, we would like to thank our colleagues in molecular biology, Olivier Gandrillon and Sophie Rome who have provided such nice challenges for data mining.

References

1. DeRisi, J., Iyer, V., Brown, P.: Exploring the metabolic and genetic control of gene expression on a genomic scale. Science **278** (1997) 680–686
2. Velculescu, V., Zhang, L., Vogelstein, B., Kinzler, K.: Serial analysis of gene expression. Science **270** (1995) 484–487
3. Niehrs, C., Pollet, N.: Synexpression groups in eukaryotes. Nature **402** (1999) 483–487
4. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA **95** (1998) 14863–14868

5. Robardet, C., Feschet, F.: Efficient local search in conceptual clustering. In: Proceedings DS'01. Number 2226 in LNCS, Springer-Verlag (2001) 323–335
6. Dhillon, I., Mallela, S., Modha, D.: Information-theoretic co-clustering. In: Proceedings ACM SIGKDD 2003, ACM (2003) 1–10
7. Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., Barkai, N.: Revealing modular organization in the yeast transcriptional network. Nature Genetics **31** (2002) 370–377
8. Bergmann, S., Ihmels, J., Barkai, N.: Iterative signature algorithm for the analysis of large-scale gene expression data. Physical Review **67** (2003)
9. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.F., Gandrillon, O.: Strong association rule mining for large gene expression data analysis: a case study on human SAGE data. Genome Biology **12** (2002) See http://genomebiology.com/2002/3/12/research/0067.
10. Creighton, C., Hanash, S.: Mining gene expression databases for association rules. Bioinformatics **19** (2003) 79 – 86
11. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In Rival, I., ed.: Ordered sets. Reidel (1982) 445–470
12. Rioult, F., Boulicaut, J.F., Crémilleux, B., Besson, J.: Using transposition for pattern discovery from microarray data. In: Proceedings ACM SIGMOD Workshop DMKD'03, San Diego (USA) (2003) 73–79
13. Rioult, F., Robardet, C., Blachon, S., Crémilleux, B., Gandrillon, O., Boulicaut, J.F.: Mining concepts from large SAGE gene expression matrices. In: Proceedings KDID'03 co-located with ECML-PKDD 2003, Catvat-Dubrovnik (Croatia) (2003) 107–118
14. Besson, J., Robardet, C., Boulicaut, J.F., Rome, S.: Constraint-based concept mining and its application to microarray data analysis. Intelligent Data Analysis journal **9** (2005) 59–82
15. Boulicaut, J.F., Klemettinen, M., Mannila, H.: Modeling KDD processes within the inductive database framework. In: Proceedings DaWaK'99. Volume 1676 of LNCS., Florence, I, Springer-Verlag (1999) 293–302
16. De Raedt, L.: A perspective on inductive databases. SIGKDD Explorations **4** (2003) 69–77
17. Pensa, R., Leschi, C., Besson, J., Boulicaut, J.F.: Assessment of discretization techniques for relevant pattern discovery from gene expression data. In: Proceedings 4th ACM SIGKDD Workshop BIOKDD'04, Seattle (USA), ACM (2004) 24–30
18. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Free-sets: a condensed representation of boolean data for the approximation of frequency queries. Data Mining and Knowledge Discovery journal **7** (2003) 5–22
19. Besson, J., Robardet, C., Boulicaut, J.F.: Constraint-based mining of formal concepts in transactional data. In: Proceedings PAKDD'04. Volume 3056 of LNAI., Sydney (Australia), Springer-Verlag (2004) 615–624
20. Pensa, R., Besson, J., Boulicaut, J.F.: A methodology for biologically relevant pattern discovery from gene expression data. In: Proceedings DS'04. Volume 3245 of LNAI., Padova (Italy), Springer-Verlag (2004) 230–241
21. Robardet, C., Pensa, R., Besson, J., Boulicaut, J.F.: Using classification and visualization on pattern databases for gene expression data analysis. In: Proceedings PaRMa'04 co-located with EDBT 2004. Volume 96 of CEUR Workshop Proceedings., Heraclion - Crete, Greece (2004)
22. Arbeitman, M., Furlong, E., Imam, F., Johnson, E., Null, B., Baker, B., Krasnow, M., Scott, M., Davis, R., White, K.: Gene expression during the life cycle of drosophila melanogaster. Science **297** (2002) 2270–2275

23. Ashburnerand, M., Ball, C., Blake, J., Botstein, D., et al.: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nature Genetics **25** (2000) 25–29
24. Goethals, B., Zaki, M.: Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations FIMI 2003, Melbourne, USA (2003)
25. Stumme, G., Taouil, R., Bastide, Y., Pasqier, N., Lakhal, L.: Computing iceberg concept lattices with TITANIC. Data & Knowledge Engineering **42** (2002) 189–222
26. Lash, A., Tolstoshev, C., Wagner, L., Schuler, G., Strausberg, R., Riggins, G., Altschul, S.: SAGEmap: A public gene expression resource. Genome Research **10** (2000) 1051–1060
27. Rome, S., Clément, K., Rabasa-Lhoret, R., Loizon, E., Poitou, C., Barsh, G.S., Riou, J.P., Laville, M., Vidal, H.: Microarray profiling of human skeletal muscle reveals that insulin regulates 800 genes during an hyperinsulinemic clamp. Journal of Biological Chemistry (2003) 278(20):18063-8.