Using classification and visualization on pattern databases for gene expression data analysis

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

1: INSA Lyon, PRISMA
F-69621 Villeurbanne cedex, France
2: INSA Lyon, LIRIS CNRS FRE 2672
F-69621 Villeurbanne cedex, France
Contact: Jean-Francois.Boulicaut@insa-lyon.fr
3: INSERM/INRA U449
F-69372 Lyon cedex 08, France

Abstract. We are designing new data mining techniques on gene expression data, more precisely inductive querying techniques that extract a priori interesting bi-sets, i.e., sets of objects (or biological situations) and associated sets of attributes (or genes). The so-called (formal) concepts are important special cases of a priori interesting bi-sets in derived boolean expression matrices, e.g., matrices that encode over-expression of genes. It has been shown recently that the extraction of every concept is often possible from typical gene expression data because the number of biological situations is generally quite small (a few tens). In specific applications, we have been able to extract every concept and it can lead to millions of concepts. Obviously, post-processing these huge volumes of patterns for the discovery of biologically relevant information is challenging. It is useful since the added-value of transcription module discovery is very high and formal concepts can be seen as putative transcription modules. We describe our ongoing research on concept post-processing by means of classification and visualization. It has been applied to a real-life gene expression data set with a promising feedback from end-users.

1 Introduction

Thanks to a huge research effort and technological breakthroughs, one of the challenges for molecular biologists is to discover knowledge from data generated at very high throughput. This is true for not only for genomic data but also in the domain of transcriptome research, i.e., the analysis of gene expression data. Indeed, different techniques (including microarray [8] and SAGE [21]) enable to study the simultaneous expression of (tens of) thousands of genes in various biological situations. The data generated by those experiments 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 lines). 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 that are associated in some way. Indeed, it is interesting to look for groups of co-regulated genes, also known as synexpression groups [13], which, based on the guilt by association approach, are assumed to participate in a common function, or module, within the cell (and thus a set of biological situations). Such an association between a set of co-regulated genes and a set of biological situations that gives rise to this co-regulation is called a transcription module and the discovery of transcription modules is one of the main goal in functional genomics.

Various techniques can be used to identify a priori interesting bi-sets. Biologists often use clustering techniques to identify sets of genes that have similar expression profiles (see, e.g., [9]). Statistical methods can be used as well (see, e.g., [10, 4]). It is also possible to look for these putative synexpression groups by computing the so-called frequent itemsets from derived boolean contexts [1, 3]. Putative transcription modules can be provided as well by computing the so-called formal concepts in this kind of boolean data [22, 17, 18]. Indeed, it is possible to encode gene expression properties into boolean matrices. Let $\mathcal{O}$ denotes a set of biological situations and $\mathcal{P}$ denotes a set of genes. In Table 1, $\mathcal{O} = \{s_1, \ldots, s_4\}$ and $\mathcal{P} = \{g_1, g_2, \ldots, g_8\}$. The expression properties can be encoded into $\mathbf{r} \subseteq \mathcal{O} \times \mathcal{P}$. $(s_i, g_j) \in \mathbf{r}$ denotes that gene j has the encoded expression property in situation i. Different expression properties might be considered like over-expression or under-expression. It is out of the scope of this paper to discuss how raw numerical gene expression data is processed to get such boolean values (see [3] for a discussion). The basic techniques rely on various discretization operators that, depending of the kind of expression property, compute thresholds from which it is possible to decide between wether the true or the false value must be assigned. For instance, in Table 1, we can say that genes g_1 and g_3 have the same encoded expression property (e.g., over-expression) in situations s_1, s_2 and s_3. This observation might lead us to derive the bi-set $(\{s_1, s_2, s_3\}, \{g_1, g_3\})$ as being potentially interesting.

$\mathcal{O}$ \ $\mathcal{P}$	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
s_1	1	0	1	1	0	0	0	0
s_2	1	1	1	1	0	1	0	1
s_3	1	0	1	1	0	0	0	0
s_4	1	1	0	1	1	1	1	1

Table 1. A boolean expression matrix $\mathbf{r}_1$

Stronger relationships between the components of a bi-set can increase their relevancy. In this paper, we consider formal concepts or concepts following the terminology of [22]. Informally, $(\{s_1, s_2, s_3\}, \{g_1, g_3, g_4\})$ is one of the concepts in $\mathbf{r}_1$. It means that $\{g_1, g_3, g_4\}$ is a maximal set of genes that have the recorded

expression property in every situation from $\{s_1, s_2, s_3\}$ and that $\{s_1, s_2, s_3\}$ is a maximal set of situations which share the true value for every gene from $\{g_1, g_3, g_4\}$. Clearly, discovered concepts can suggest transcription modules.

It has been shown in [17, 18] that concept discovery from typical boolean gene expression data is tractable thanks to efficient algorithms for computing the frequent closed sets [14, 7, 15, 2, 23] and classical properties of Galois operators [22].

The contribution of this paper is twofold. First, we summarize in Section 2 some of our recent work on concept mining using a very simple formalization. Secondly, we describe in Section 3 our ongoing research on pattern post-processing for gene expression data analysis. We focuss on a technique that classifies concepts and provide a visualization that has been designed to be familiar to molecular biologists. Indeed, it relies on the TreeView component of the popular Eisen's Lab software [9]. This paper does not contain a biological validation that is however ongoing. Indeed, some biologically relevant information has been discovered on human SAGE data using the described post-processing technique. [16] contains a preliminary report of this biological validation.

2 From gene expression data to concept databases

We do not want to provide here a formalization in terms of inductive databases for bi-set mining. A simple formalization has been proposed in [18] and bi-set mining is a straightforward extension of the set pattern domain [6]. We just recall the most important notions for concept discovery and, based on them, refer to interesting recent work and motivate our ongoing research in this area.

Definition 1. *(Language of bi-sets) The language of bi-sets is the collection of couples from $\mathcal{L}_\mathcal{O} \times \mathcal{L}_\mathcal{P}$ where $\mathcal{L}_\mathcal{O} = 2^\mathcal{O}$ (sets of situations) and $\mathcal{L}_\mathcal{P} = 2^\mathcal{P}$ (sets of genes).*

Let us now consider evaluation functions for such patterns. We only consider Galois operators (denoted as ϕ and ψ) that have been proved extremely useful.

Definition 2. *(Galois connection [22]) If $T \subseteq \mathcal{O}$ and $G \subseteq \mathcal{P}$, assume $\phi(T, \mathbf{r}) = \{g \in \mathcal{P} \mid \forall t \in T,\ (t, g) \in \mathbf{r}\}$ and $\psi(G, \mathbf{r}) = \{t \in \mathcal{O} \mid \forall g \in G,\ (t, g) \in \mathbf{r}\}$. ϕ provides the set of genes that share the expression property in a given set of situations. ψ provides the set of situations in which a given set of genes share the expression property. (ϕ, ψ) is the so-called Galois connection between $\mathcal{O}$ and $\mathcal{P}$. We use the classical notations $h = \phi \circ \psi$ and $h' = \psi \circ \phi$ to denote the Galois closure operators.*

Notice that the classical frequency measures are defined easily in terms of $|\psi(G, \mathbf{r})|$ for $G \subseteq \mathcal{P}$ and $|\phi(T, \mathbf{r})|$ for $T \subseteq \mathcal{O}$.

Definition 3. *(Closed set and $\mathcal{C}_{Close}$ constraint) A set of genes $G \subseteq \mathcal{P}$ is closed when it satisfies $\mathcal{C}_{Close}(G, \mathbf{r}) \equiv h(G, \mathbf{r}) = G$. Dually, for sets of situations $T \subseteq \mathcal{O}$, $\mathcal{C}_{Close}(T, \mathbf{r}) \equiv h'(T, \mathbf{r}) = T$.*

Definition 4. *(1-rectangle and concept) A bi-set (T, G) is a 1-rectangle iff $\forall g \in G$ and $\forall t \in T$, $(t, g) \in \mathbf{r}$. (T, G) is called a formal concept or concept in $\mathbf{r}$ when $T = \psi(G, \mathbf{r})$ and $G = \phi(T, \mathbf{r})$.*

Example 1. Let us consider the bi-set $(\{s_1, s_2, s_3\}, \{g_1, g_3\})$ and the data from Table 1. We have $\psi(\{g_1, g_3\}, \mathbf{r}_1) = \{s_1, s_2, s_3\}$ and this bi-set is a 1-rectangle. The set $\{g_1, g_3\}$ is not closed since $h(\{g_1, g_3\}, \mathbf{r}_1) = \phi(\psi(\{g_1, g_3\}, \mathbf{r}_1), \mathbf{r}_1) = \{g_1, g_3, g_4\}$. $\{g_1, g_2, g_3, g_4\}$ is a closed set of genes ($h(\{g_1, g_4\}, \mathbf{r}_1) = \{g_1, g_4\}$). $\{s_1, s_2, s_3, s_4\}$ is a closed set on situations ($h'(\{s_1, s_2, s_3, s_4\}, \mathbf{r}_1) = \{s_1, s_2, s_3, s_4\}$). Furthermore, $\{s_1, s_2, s_3, s_4\} = \psi(\{g_1, g_4\}, \mathbf{r}_1)$ and $\{g_1, g_4\} = \phi(\{s_1, s_2, s_3, s_4\}, \mathbf{r}_1)$. $(\{s_1, s_2, s_3, s_4\}, \{g_1, g_4\})$ is indeed one of the six concepts in $\mathbf{r}_1$ (see Table 2 for the complete list).

By construction, concepts are built on closed sets and each closed set of genes (resp. situations) is linked to a closed set of situations using ψ (resp. genes using ϕ) [22]. In other terms, for a concept (T, G) in $\mathbf{r}$ we have $\mathcal{C}_{Close}(T, \mathbf{r}) \wedge \mathcal{C}_{Close}(G, \mathbf{r})$ and the collection of concepts is included in the collection of 1-rectangles.

c_1	$(\{s_1, s_2, s_3, s_4\}, \{g_1, g_4\})$
c_2	$(\{s_1, s_2, s_3\}, \{g_1, g_3, g_4\})$
c_3	$(\{s_2, s_4\}, \{g_1, g_2, g_4, g_6, g_8\})$
c_4	$(\{s_2\}), \{g_1, g_2, g_3, g_4, g_6, g_8\})$
c_5	$(\{s_4\}, \{g_1, g_2, g_4, g_5, g_6, g_7, g_8\})$
c_6	$(\{\emptyset\}, \{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8\})$

Table 2. Concepts in $\mathbf{r}_1$

Many data mining processes on boolean data can be formalized as the computation of bi-sets whose set components satisfy some constraints. For instance, computing frequent sets of genes (frequent w.r.t. a threshold γ) and the situations in which they are co-regulated means that we compute bi-sets (T, G) such that $|\psi(G, \mathbf{r})| \geq \gamma$ and $T = \psi(G, \mathbf{r})$.

Closed set mining for genes is specified as the computation of $\{G \in \mathcal{L}_{\mathcal{P}} \mid \mathcal{C}_{Close}(G, \mathbf{r})$ satisfied$\}$. The collection $\Theta = \{(T, G) \in \mathcal{L}_{\mathcal{O}} \times \mathcal{L}_{\mathcal{P}} \mid \mathcal{C}_{Close}(G, \mathbf{r}) \wedge T = \psi(G, \mathbf{r})\}$ is the collection of concepts. Interestingly, we have also $\Theta = \{(T, G) \in \mathcal{L}_{\mathcal{O}} \times \mathcal{L}_{\mathcal{P}} \mid \mathcal{C}_{Close}(T, \mathbf{r}) \wedge G = \phi(T, \mathbf{r})\}$. It provides a strategy for computing concepts [17].

Several efficient algorithms have been designed for computing the frequent closed sets of columns in boolean matrices [14, 7, 15, 2, 23]. These algorithms can work in dense boolean data. When the number of columns is small enough, it is however possible to compute the closed sets for the frequency threshold 1 such that every closed set is computed. We can compute the closed sets on the smaller dimension by a simple transposition and the Galois operators enable to provide the associated closed sets on the other dimension [17]. In practice, we have been

	Density	Number of concepts
Human SAGE 74×822	12.2	80 068
Human SAGE 74×822	3.8	1 386
Human SAGE 74×822	4.8	1 808
Human SAGE $90 \times 12\,636$	4.8	196 130
Human SAGE $90 \times 12\,636$	2.2	9 150
Human SAGE $90 \times 12\,636$	4.7	31 766
`droso` $162 \times 1\,230$	1.5	1 508
`droso` $162 \times 1\,230$	3.2	10 447
`droso` $162 \times 1\,230$	6.7	259 938

Table 3. Concept extractions in human SAGE data [18] and Drosophila [17]

able to use a frequent closed set mining algorithm designed in our group ([7]) with a frequency threshold of 1 such that every closed set and thus concept is extracted. Table 3 contains the number of extracted concepts as reported in [17, 18]. Density is the percentage of true values among the matrix cells. The various densities are obtained by using different discretization techniques. The largest collection reported here is of size 259 938. It is however common to find applications where millions of patterns are extracted from gene expression data [20, 5].

The important message is thus that databases that already contain raw expression data and multiple derived boolean contexts can now be associated to huge collections of patterns that hold in these data. These both components (data and patterns) constitute the source data for pattern post-processing that must be done in cooperation with the end-users. This post-processing is fundamentally needed since data mining algorithms only provide a priori interesting patterns. We have to support queries on such huge pattern databases. Not only we need query languages for that but also efficient query evaluation techniques in order to preserve the interactivity between end-users and the databases. We are far from solutions on these two topics. Many open problems are still to be addressed like the efficient storage of huge collections of set patterns within databases. It is important that post-processing techniques can take the most from the available database technologies. For instance, [12] studies set pattern querying optimization within relational database management systems. Also, the smart integration of the many data sources that are needed for concept post-processing is challenging. An obvious example concerns the use of additional biological information, e.g., Gene Ontology[1] in order to support the interpretation of concepts and, e.g., study the homogeneity of the sets of genes at different levels (biological process, molecular function, cellular function). The definition of query languages for pattern post-processing, and more generally inductive databases is also challenging.

[1] http://www.geneontology.org/

Furthermore, it is important to design new post-processing techniques in cooperation with the end-users, molecular biologists in our case. Indeed, one of our motivations has been to reuse the visualization approach of the popular Eisen's lab clustering software [9] in the new context of concept post-processing.

3 Concept classification and visualization

The variability of the measurement process and the noise can dramatically increase the number of extracted concepts. When an error of measurement or an intrinsic variability of the observed phenomenon lead to a 0 value whereas the value 1 should be obtained, we have to face with a multiplication of the number of concepts. To illustrate this fact, we represent on the left of Figure 1 a data matrix made of "1" values except one symbolized by a white square. The center and the right of Figure 1 represent the two extracted concepts (the dark area) from the data. Assume that nz denotes the number of 0 values that are present by error in the data and that all these 0 values are on different lines and different columns of matrix. In such a special case, the additional number of concepts is equal to 2^{nz}.

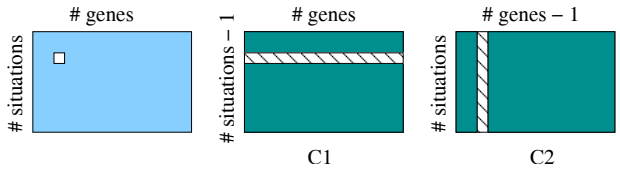


Fig. 1. Multiplication of concepts due to noise/variability

We consider that one of the main goals of concept post-processing is to group concepts that are similar enough. It can be achieved by a hierarchical classification method to cluster concepts and then a visualization of the obtained hierarchy of concepts.

3.1 Hierarchical clustering on formal concepts

We have been using an ascendant hierarchical clustering method to group concepts. At each iteration, the principle is to merge the two closest clusters into a single one until there is only one cluster. The difficulty in using such an algorithm to cluster concepts is to take into account simultaneously the genes and the biological situations instead of clustering separately the genes and the biological situations as it is done usually, e.g., in [9].

We propose to use a distance measure to compare two concept clusters which takes into account the genes and the biological situations that are overlapping between the two clusters. We have to define a distance between two concepts and

then a distance between two clusters. 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 5. *(Distance between two concepts) Assume that $c_i = (T_i, G_i)$ and $c_j = (T_j, G_j)$ are two concepts, the distance d between c_i and c_j is defined as*

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

where $|S|$ denotes the cardinal of S.

Classically, in hierarchical clustering, the distance between two clusters is defined by the shortest distance between a concept of the first cluster and one of the second cluster, or the largest distance between a concept of the first cluster and one of the second cluster, or the average distance between all the couples of concepts from the two clusters. We have used the average distance which is more robust to data variability.

The time complexity for computing the average distance between two clusters of concepts is in $O(n \times m)$ where n and m are the size of the clusters. Given the huge size of the collections of concepts we want to post-process, we have considered an optimization based on a fuzzy measure.

3.2 Fuzzy hierarchical clustering

One idea for reducing the computation complexity is to associate a pseudo-concept to each cluster. A pseudo-concept is a unique representation for all the concepts in a cluster. It is composed of two fuzzy sets, one set of genes and one set of biological situations. As usually, a fuzzy set is a set whose elements belong more or less to that set, i.e., 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 $\mathcal{P}$). The exact value 0 means that the element does not belong to the fuzzy set and the exact value 1 means that the element belongs to the fuzzy set.

Definition 6. *(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 concepts represented by the pseudo-concept. A pseudo-concept (T', G', N) of a single concept (T, G) is defined by:*

$$\begin{cases} T' = \{(s, \alpha) \mid s \in \mathcal{O},\ \alpha = 1 \text{ if } s \in T,\ \alpha = 0 \text{ otherwise}\} \\ G' = \{(g, \alpha) \mid g \in \mathcal{P},\ \alpha = 1 \text{ if } g \in G,\ \alpha = 0 \text{ otherwise}\} \end{cases}$$

The pseudo-concept (T', G', N) for two pseudo-concepts (T'_1, G'_1, N_1) and (T'_2, G'_2, N_2) is defined as follows:

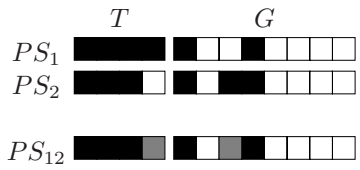
$$\begin{cases} T' = \left\{ \left(s, \frac{N_1 \times \alpha_1 + N_2 \times \alpha_2}{N_1 + N_2}\right) \mid \forall s \in \mathcal{O} \text{ and } (s, \alpha_1) \in T'_1 \text{ and } (s, \alpha_2) \in T'_2 \right\} \\ G' = \left\{ \left(g, \frac{N_1 \times \alpha_1 + N_2 \times \alpha_2}{N_1 + N_2}\right) \mid \forall g \in \mathcal{P} \text{ and } (g, \alpha_1) \in G'_1 \text{ and } (g, \alpha_2) \in G'_2 \right\} \\ N = N_1 + N_2 \end{cases}$$

The pseudo-concept corresponding to $\{c_1, c_2, c_3\}$ (see Table 2) is built as follows. First, we compute the pseudo-concepts for c_1 (PS_1), c_2 (PS_2) and c_3 (PS_3). Then, we merge PS_1 and PS_2 into PS_{12} and finally we merge PS_{12} and PS_3 into PS_{123}.

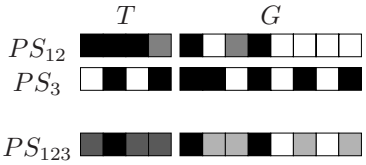
$$PS_1 = (\{(s_1;1),\ (s_2;1)\ ,(s_3;1),\ (s_4;1)\}, \{(g_1;1),\ \ (g_4;1)\}, 1)$$
$$PS_2 = (\{(s_1;1)\ ,(s_2;1),\ (s_3;1)\}, \{(g_1;1),\ (g_3;1),\ (g_4;1)\}, 1)$$
$$PS_3 = (\{(s_2;1)\ (s_4;1)\}, \{(g_1;1),\ (g_2;1),\ (g_4;1),\ (g_6;1),\ (g_8;1)\}, 1)$$

$$PS_{12} = (\{(s_1;1),\ (s_2;1)\ ,(s_3;1),\ (s_4;0.5)\}, \{(g_1;1),\ (g_3;0.5),\ (g_4;1)\}, 2)$$

Graphically, this can be represented by two arrays, the first one (on the left) represents the degree of membership of the eight genes. The second one (on the right) represents the degree of membership of the four biological situations. The darkness of a cell is correlated with the degree of membership, i.e., the more it is black, the more the element belongs to the set.



$$PS_{123} = (\{(s_1;2/3),\ (s_2;1)\ ,(s_3;2/3),\ (s_4;2/3)\}$$
$$\{(g_1;1),\ (g_2;1/3),\ (g_3;1/3),\ (g_4;1),\ (g_6;1/3),\ (g_8;1/3)\}, 3)$$



Pseudo-concepts can be computed incrementally. The order in which the concepts are considered does not influence the result. The pseudo-concept which comes from the merge of two clusters is computed using only the two pseudo-concepts of these clusters and not every underlying concept.

Definition 7. *(Fuzzy distance) It is possible to generalize distance d for measuring the similarity between pseudo-concepts. The classical fuzzy set operators*

(indexed with f) are used:

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

Let us illustrate the computation of this fuzzy distance between PS_1 and PS_2, and between PS_{12} and PS_3:

$$d(PS_1, PS_2) = \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{3} = 0.292$$
$$d(PS_{12}, PS_3) = \frac{1}{2} \times \frac{2.5}{4} + \frac{1}{2} \times \frac{3.5}{5.5} = 0.6306$$

	PS_1	PS_2	PS_3	PS_4	PS_5	PS_6
PS_1		0.292	0.45	0.625	0.6607	0.8125
PS_2			0.625	0.5	0.8125	0.75
PS_3				0.333	0.393	0.688
PS_4					0.688	0.625
PS_5						0.5625

Step 1

	PS_{12}	PS_3	PS_4	PS_5	PS_6
PS_{12}		0.6306	0.6488	0.8041	0.8437
PS_3			0.333	0.393	0.688
PS_4				0.688	0.625
PS_5					0.5625

Step 2

	PS_{12}	PS_{34}	PS_5	PS_6
PS_{12}		0.5773	0.8041	0.8437
PS_{34}			0.5083	0.625
PS_5				0.5625

Step 3

	PS_{12}	PS_{345}	PS_6
PS_{12}		0.6614	0.8437
PS_{345}			0.6041

Step 4

Fig. 2. Evolution of the distance table during tree construction

Figure 2 provides the distance tables computed at each step of the hierarchical clustering process. At Step 1, all concepts are in a single cluster and the closest clusters PS_1 and PS_2 are merged. At Step 2 PS_3 and PS_4 are merged, then PS_{34} and PS_5 are merged. Finally, PS_{345} and PS_6 are merged into a single concept.

When using pseudo-concepts, the time complexity to compute the distance between two clusters is in $O(|\mathcal{O}| + |\mathcal{P}|)$. When two clusters are merged, the corresponding pseudo-concept is computed in $O(|\mathcal{O}| + |\mathcal{P}|)$. Thus, time depends linearly on the size of the gene and situation sets whereas it was related to the number of concepts in the first approach.

Example of a hierarchy of concepts The hierarchical clustering result, i.e., a hierarchy of concepts, can be represented as in Figure 3. The difficulty is that we have three dimensions: the concepts, the situations and the genes. In order to draw the result in a plan, we decided to represent the hierarchy of concepts either w.r.t. the genes or w.r.t. the situations. When a gene or a situation belongs to a concept, the corresponding cell is black.

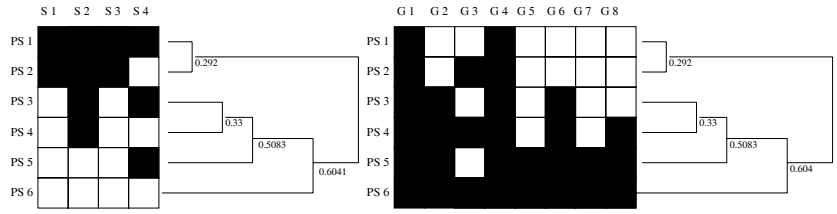


Fig. 3. First visualization concepts × situations (left) and concepts × genes (right)

We observe in Figure 3 that concepts PS_1 and PS_2 have been merged. Indeed, we can see that these concepts are very similar and it makes sense to see them as a single one.

The fuzzy distance can be used to merge very close concepts. For example, Figure 4 presents the hierarchy obtained when concepts with distance lower than $\frac{1}{3}$ are merged.

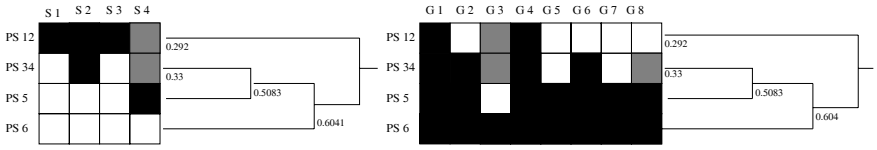


Fig. 4. Second visualization concepts × situations (left) and concepts × genes (right)

3.3 An application to real gene expression data

As a simple illustration, let us apply this technique on the gene expression data from [11]. It concerns the expression of 190 genes over 11 biological situations. After discretization, we have extracted 191 concepts representing sets of genes which are up-regulated over associated sets of situations. Figure 5 presents the hierarchical clustering of these concepts (columns) and the expression level of the genes (rows) over them. We used HCE [19] to visualize the dendrogram.

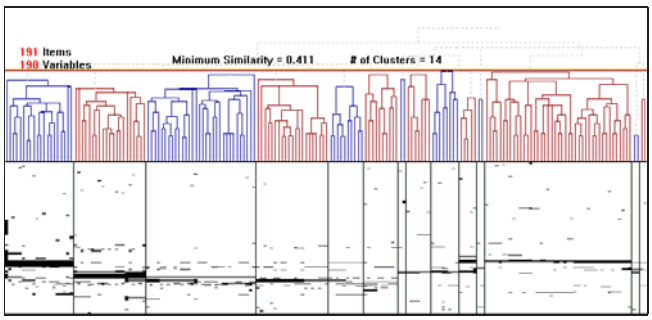


Fig. 5. Hierarchical clustering of concepts visualized w.r.t. genes

Figure 5 shows that concepts of a single cluster share a small set of genes up-expressed (in dark on the figure) which characterize the cluster. Some other genes are punctually up-regulated in some of these concepts: one can suspect that these variations are due to noise or errors of measurement. The impact of these unwilling variations can be tackled thanks to the classification of concepts.

4 Conclusion

Starting from the possibility to extract huge collections of concepts in boolean gene expression data, we are considering post-processing techniques that can support molecular biology discovery, more precisely the discovery of transcription modules. Using the same approach than a popular software for gene or biological situation clustering, the Eisen's clustering software, we have applied the same principles to concept classification. The validation of the biological relevancy of these techniques on real data sets is ongoing. It concerns both the human SAGE data for which we have been able to extract every concept [18] but also microarray data that has been processed for concept discovery [5].

Acknowledgment The authors thank Olivier Gandrillon and Sylvain Blachon for exciting discussions and their participation to the evaluation of the biological relevancy of this research. Jérémy Besson is funded by INRA. This work has been partially funded by the EU contract cInQ IST-2000-26469 (FET arm of the IST programme).

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In Advances in Knowledge Discovery and Data Mining, pages 307–328. AAAI Press, 1996.
2. Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining frequent patterns with counting inference. SIGKDD Explorations, 2(2):66 – 75, Dec. 2000.

3. C. Becquet, S. Blachon, B. Jeudy, J.-F. Boulicaut, and O. Gandrillon. Strong association rule mining for large gene expression data analysis: a case study on human SAGE data. Genome Biology, 12, 2002.
4. S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. Physical Review, 67, March 2003.
5. J. Besson, C. Robardet, J.-F. Boulicaut, and S. Rome. Constraint-based concept mining and its application in microarray data analysis. Intelligent Data Analysis, 2004. Accepted for publication.
6. J.-F. Boulicaut. Inductive databases and multiple uses of frequent itemsets: the cInQ approach. In R. Meo, P. L. Lanzi, and M. Klemettinen, editors, Database Support for Data Mining Applications, number 2682 in LNCS. springer, 2004.
7. J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by mean of free-sets. In PKDD'00, volume 1910 of LNAI, pages 75–85, 2000.
8. J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science, 278, 1997.
9. M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. National Academy of Science USA, 95:14863–14868, 1998.
10. J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. Nature Genetics, 31:370–377, august 2002.
11. A. Lash, C. Tolstoshev, L. Wagner, G. Schuler, R. Strausberg, G. Riggins, and S. Altschul. Sagemap: a public gene expression resource. Genome Research, 10:1051–1060, 2000.
12. C. Masson, C. Robardet, and J.-F. Boulicaut. Optimizing subset queries: a step towards SQL-based inductive databases for itemsets. In ACM SAC 2003 Data Mining Track, Nicosia, Cyprus, March 2004. To appear.
13. C. Niehrs and N. Pollet. Synexpression groups in eukaryotes. Nature, 402:483–487, 1999.
14. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. Information Systems, 24(1):25–46, Jan. 1999.
15. J. Pei, J. Han, and R. Mao. CLOSET an efficient algorithm for mining frequent closed itemsets. In ACM SIGMOD Workshop DMKD'00, pages 21–30, 2000.
16. R. Pensa. Algorithmes de clustering et de caractérisation de clusters: application à des données d'expression de gènes. Master thesis, Politecnico di Torino, 2003.
17. F. Rioult, J.-F. Boulicaut, B. Crémilleux, and J. Besson. Using transposition for pattern discovery from microarray data. In ACM SIGMOD Workshop DMKD'03, pages 73–79, San Diego, USA, June 2003.
18. F. Rioult, C. Robardet, S. Blachon, B. Crémilleux, O. Gandrillon, and J.-F. Boulicaut. Mining concepts from large SAGE gene expression matrices. In KDID'03 co-located with ECML-PKDD'03, pages 107–118, 2003.
19. J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. IEEE Computer, 35(7):80–86, 2002.
20. A. Tuzhilin and G. Adomavicius. Handling very large numbers of association rules in the analysis of microarray data. In ACM SIGKDD'02. AAAI Press, 2002.
21. V. Velculescu, L. Zhang, B. Vogelstein, and K. Kinzler. Serial analysis of gene expression. Science, 270:484–487, 1995.
22. R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival, editor, Ordered sets, pages 445–470. Reidel, 1982.
23. M. J. Zaki and C.-J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. In SIAM DM'02, Arlington, USA, April 2002.