# Using transposition for pattern discovery from microarray data

François Rioult
GREYC CNRS UMR 6072
Université de Caen
F-14032 Caen, France
frioult@info.unicaen.fr

Jean-François Boulicaut
LIRIS CNRS FRE 2672
INSA Lyon
F-69621 Villeurbanne, France
jfboulic@lisi.insa-lyon.fr

Bruno Crémilleux
GREYC CNRS UMR 6072
Université de Caen
F-14032 Caen, France
bruno@info.unicaen.fr

Jérémy Besson
LIRIS CNRS FRE 2672
INSA Lyon
F-69621 Villeurbanne, France
jbesson@lisi.insa-lyon.fr

## ABSTRACT

We analyze expression matrices to identify a priori interesting sets of genes, e.g., genes that are frequently co-regulated. Such matrices provide expression values for given biological situations (the lines) and given genes (columns). The frequent itemset (sets of columns) extraction technique enables to process difficult cases (millions of lines, hundreds of columns) provided that data is not too dense. However, expression matrices can be dense and have generally only few lines w.r.t. the number of columns. Known algorithms, including the recent algorithms that compute the so-called condensed representations can fail. Thanks to the properties of Galois connections, we propose an original technique that processes the transposed matrices while computing the sets of genes. We validate the potential of this framework by looking for the closed sets in two microarray data sets.

## 1. INTRODUCTION

We are now entering the post-genome era and it seems obvious that, in a near future, the critical need will not be to generate data, but to derive knowledge from huge data sets generated at very high throughput. Different techniques (including microarrays and SAGE) enable to study the simultaneous expression of (tens of) thousands of genes in various biological situations. The data generated by those experiments can then be seen as expression matrices in which the expression level of genes (the columns) are recorded in various biological situations (the lines). Various knowledge discovery methods can be applied on such data, e.g., the discovery of sets of co-regulated genes, also known as synexpression groups [12]. These sets can be computed from the frequent sets in the boolean matrices coding for the expression data (see Table 1).

One attribute $a_i$ is attributed the value true (1) to represent the over- (or under-) expression of gene $i$ in that particular situation.

Discretization procedures (true is assigned above some threshold value) are used to derive boolean matrices from a raw expression matrix. Discretization can obviously have a large influence on the nature of the extracted sets. It is thus essential that, in exploratory contexts, one can study different threshold values and proceed with a large number of analysis.

What we would like to do is to compute **all** sets of genes that have the true value in a sufficient number (frequency threshold) of biological situations. Extracting frequent sets is one of the most studied data mining techniques since the description of the APRIORI algorithm [1] and tens of algorithms have been published. Nevertheless, the gene expression matrices, obtained through microarrays, raise new difficulties, due to their "pathological" dimensions (i.e. few lines and a huge number of columns). This is a very difficult problem since the overall complexity is exponential in the number of genes. Furthermore the size of the solutions (i.e. collection of extracted sets) is huge whatever the frequency threshold since there is a very limited number of lines.

In Section 2, we present the problems raised by the extraction of frequent sets. Section 3 proposes a solution that combines the power of the closed set extraction with an original use of the properties of the Galois connection [17; 9]. In Section 4 we provide experimental results on two matrices built from microarray data [2; 16]. It establishes the spectacular gains allowed by our approach. Section 5 concludes.

## 2. FREQUENT SET EXTRACTION

### 2.1 Definitions

Let $\mathcal{S}$ denote a set of biological situations and $\mathcal{A}$ denote a set of attributes. In the example from Table 1, $\mathcal{S} = \{s_1, \ldots s_5\}$ and $\mathcal{A} = \{a_1, \ldots a_{10}\}$.

| | Attributes | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Situations | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ |
| $s_1$ | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| $s_2$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| $s_3$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| $s_4$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| $s_5$ | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

Table 1: Example of a boolean matrix $\mathbf{r}_1$

Each attribute denotes a property about the expression of a gene. The encoded expression data is represented by the matrix of the binary relation $R \subset \mathcal{S} \times \mathcal{A}$ defined for each situation and each attribute. $(s_i, a_j) \in R$ denotes that situation $i$ has the property $j$, i.e., that gene $j$ is over-expressed or under-expressed in situation $i$. A database $r$ to be mined is thus a 3-tuple $(\mathcal{S}, \mathcal{A}, R)$. $\mathcal{L}_\mathcal{A} = 2^\mathcal{A}$ is the power set of attributes. For the sake of clarity, sets of attributes are often called sets of genes. $\mathcal{L}_\mathcal{S} = 2^\mathcal{S}$ is the power set of situations.

*Definition 1.* Given $T \subseteq \mathcal{S}$ and $X \subseteq \mathcal{A}$, let $f(T) = \{a \in \mathcal{A} \mid \forall s \in T, (s, a) \in R\}$ and $g(X) = \{s \in \mathcal{S} \mid \forall a \in X, (s, a) \in R\}$. $f$ provides the set of over-expressed or under-expressed genes that are common to a set of situations and $g$ provides the set of situations that share a given set of attributes (expression properties). $(f, g)$ is the so-called Galois connection between $\mathcal{S}$ and $\mathcal{A}$. We use the classical notations $h = f \circ g$ and $h' = g \circ f$ to denote the Galois closure operators.

*Definition 2.* A set of genes $X \subseteq \mathcal{A}$ is closed iff $h(X) = X$. We say that $X$ satisfies the $\mathcal{C}_{Close}$ constraint in $\mathbf{r}$: $\mathcal{C}_{Close}(X, \mathbf{r}) \equiv h(X) = X$. A set of situations $T \subseteq \mathcal{S}$ is closed iff $h'(T) = T$.

*Definition 3.* The *frequency* of a set of genes $X \subseteq \mathcal{A}$ denoted $\mathcal{F}(X, \mathbf{r})$ is the size of $g(X)$. Constraint $\mathcal{C}_{\text{freq}}$ enforces a minimal frequency: $\mathcal{C}_{\text{freq}}(X, \mathbf{r}) \equiv \mathcal{F}(X, \mathbf{r}) \geq \gamma$ where $\gamma$ is the user-defined frequency threshold.

*Example 1.* Given $\mathbf{r}_1$, $\mathcal{F}(\{a_1, a_2\}, \mathbf{r}_1) = 3$ and $\mathcal{F}(\{a_1, a_3, a_5\}, \mathbf{r}_1) = 1$. If $\gamma = 3$, $\{a_9, a_{10}\}$ and $\{a_1, a_2, a_3, a_4\}$ satisfy $\mathcal{C}_{\text{freq}}$ but $\{a_1, a_5\}$ does not. $h(\{a_1, a_2\}) = f(g(\{a_1, a_2\})) = f(\{s_1, s_2, s_3\}) = \{a_1, a_2, a_3, a_4\}$. $\{a_1, a_2, a_3, a_4\}$ satisfies $\mathcal{C}_{Close}$ but $\{a_1, a_2\}$ does not.

**Mining task.** We want to compute the collection of the frequent sets of genes $FS = \{\varphi \in \mathcal{L}_\mathcal{A} \mid \mathcal{C}_{\text{freq}}(\varphi, \mathbf{r}) \text{ satisfied}\}$ where $\mathcal{C}_{\text{freq}}$ is the minimal frequency constraint and $\mathbf{r}$ is a boolean expression matrix. Furthermore, we need the frequencies of each frequent itemset to, e.g., derive interesting association rules from them.
The closure of a set of genes $X$, $h(X)$, is the maximal (w.r.t. set inclusion) superset of $X$ which has the same frequency than $X$. A closed set of genes is thus a maximal set of genes whose expression properties (true values) are shared by a set of situations. E.g., the closed set $\{a_1, a_3\}$ in the data of Table 1, is the largest set of genes that are over-expressed (or under-expressed) simultaneously in situations $s_1$, $s_2$, $s_3$ and $s_5$.
The concept of free set has been introduced in [7] as a special case of the $\delta$-free sets and has been proposed independently in [3] under the name of key pattern. This concept characterizes the closed set generators [13] but is also useful for non redundant association rule computation (see, e.g., [5] for an illustration).

*Definition 4.* A set of genes $X \subseteq \mathcal{A}$ is free iff $X$ is not included in the closure (i.e., $h = f \circ g$) of one of its strict subsets. We say that $X$ satisfies the $\mathcal{C}_{\text{free}}$ constraint in $\mathbf{r}$. An alternative definition is that $X$ is free in $\mathbf{r}$ iff the frequency of $X$ in $\mathbf{r}$ is strictly lower than the frequency of every strict subset of $X$.

*Example 2.* Given Table 1, $\{a_1, a_6\}$ satisfies $\mathcal{C}_{\text{free}}$ but $\{a_1, a_2, a_3\}$ does not.

It is easy to adapt these definitions to sets of situations. An important result is that the closures of the free sets are closed sets. The size of the collection of the free sets is, by construction, greater or equal to the size of the collection of the closed sets (see, e.g., [8]).
It is well known that $\mathcal{L}_\mathcal{A}$ can be represented by a lattice ordered by set inclusion. On top of the lattice, we have the empty set, then the singletons, the pairs, etc. The last level for our example from Table 1 contains the unique set of size 10. A classical framework [11; 10] for an efficient exploration of such a search space is based on the monotonicity of the used constraints w.r.t. the specialization relation, i.e., set inclusion.

*Definition 5.* A constraint $\mathcal{C}$ on sets is said anti-monotonic when $\forall X, X'$: $(X' \subseteq X \wedge X$ satisfies $\mathcal{C}) \Rightarrow X'$ satisfies $\mathcal{C}$. A constraint $\mathcal{C}$ is said monotonic when $\forall X, X'$: $(X \subseteq X' \wedge X$ satisfies $\mathcal{C}) \Rightarrow X'$ satisfies $\mathcal{C}$.

*Example 3.* $\mathcal{C}_{\text{freq}}$, $\mathcal{C}_{\text{free}}$ and $\mathcal{C}_{\text{freq}} \wedge \mathcal{C}_{\text{free}}$ are anti-monotonic. $\mathcal{C}_{size}(X) \equiv |X| > 3$ is monotonic.

Notice that the negation of a monotonic (resp. anti-monotonic) constraint is an anti-monotonic (resp. monotonic) constraint. Anti-monotonic constraints can be pushed efficiently into the extraction process: when a set $X$ does not satisfy an anti-monotonic constraint, we can prune large parts of the lattice since no superset of $X$ can satisfy it. For instance, the Apriori algorithm [1] computes all the frequent sets by a levelwise search on the lattice, starting from the most general sentences (the singletons) until it reaches the most specific sentences that are frequent (the maximal frequent sets w.r.t. set inclusion). Apriori and its variants work well on very large boolean matrices (millions of lines, hundreds or thousands of columns) that are not dense and for lowly correlated data. Notice that such algorithms have to count the frequency of at least every frequent set.

## 2.2 Extraction tractability

The computation of sets that satisfy a given constraint $\mathcal{C}$ is a very hard problem. Indeed, as soon as we have more than a few tens of columns, only a quite small subset of the search space can be explored. Then, the size of the solution, i.e., the collection of the sets that satisfy $\mathcal{C}$ can be so huge that none algorithm can compute them all. When the used constraint is $\mathcal{C}_{\text{freq}}$, it is possible to take a greater frequency threshold to decrease a priori the size of the solution and thus provide the whole collection of the frequent sets. The used threshold can however be disappointing for the biologist: extracted patterns are so frequent that they are already known.

In the expression matrices we have to analyze, the number of the frequent sets can be huge, whatever is the frequency threshold. It comes from the rather low number of lines and thus the small number of possible frequencies. Clearly, Apriori and its variants can not be used here. Since we need for the frequencies of every frequent set, e.g., for deriving valid association rules, algorithms that compute only the maximal frequent itemsets, e.g., [4] do not solve the problem. We decided to investigate the use of the so-called *condensed representations* of the frequent sets by the frequent closed sets, i.e., $CFS = \{\varphi \in \mathcal{L}_{\mathcal{A}} \mid \mathcal{C}_{\text{freq}}(\varphi, \mathbf{r}) \wedge \mathcal{C}_{Close}(\varphi, \mathbf{r}) \text{ satisfied}\}$ because $FS$ can be efficiently derived from $CFS$ [13; 6]. $CFS$ is a compact representation of the information about every frequent set and its frequency. Furthermore, several recent algorithms can compute efficiently the frequent closed sets [13; 7; 8; 14; 18; 3]. To be efficient, these algorithms can not use the properties of $\mathcal{C}_{Close}$ which is neither anti-monotonic nor monotonic. However, we can compute the frequent free sets and provide their closures, i.e., $\{h(\varphi) \in \mathcal{L}_{\mathcal{A}} \mid \mathcal{C}_{\text{freq}}(\varphi, \mathbf{r}) \wedge \mathcal{C}_{\text{free}}(\varphi, \mathbf{r}) \text{ satisfied}\}$ [7; 8; 3]. The lattice is still explored levelwise. At level $k$, the data is accessed to compute the frequency and the closure of each candidate set. The infrequent sets can be pruned. Thanks to pruning at level k-1, the frequent sets are free sets. Candidates for the next level can be generated from two free sets (using an Apriori-like generation procedure [1]) and candidates for which at least one subset is not frequent ($\mathcal{C}_{\text{freq}}$ is violated) or that are included in the closure of one their subsets (i.e., $\mathcal{C}_{\text{free}}$ is violated) are pruned before the next iteration can start. At the end, we compute $h(X)$ for each frequent free set that has been extracted. It turns out that the anti-monotonicity of a constraint like $\mathcal{C}_{\text{freq}} \wedge \mathcal{C}_{\text{free}}$ is used in two phases. First (Criterion 1), we avoid the computation of supersets that do not satisfy the constraint thanks to the Apriori-like generation procedure. Next (Criterion 2), we prune the sets for which some subsets do not satisfy the constraint. The number of pruned candidates in the second phase, i.e., failures for Criterion 2, can be huge for matrices with a number of lines that is small w.r.t. the number of columns and it can lead to intractable extractions. In other terms, even though these approaches have given excellent results on large matrices for transactional data (e.g., correlated and dense data in WWW usage mining applications), they can fail on expression matrices because of their "pathological" dimensions. Furthermore, we want to enable the use of various discretization procedures and thus the analysis of more or less dense matrices. It appears crucial to us that we can achieve a breakthrough w.r.t. extraction intractability and it has lead to the following original method.

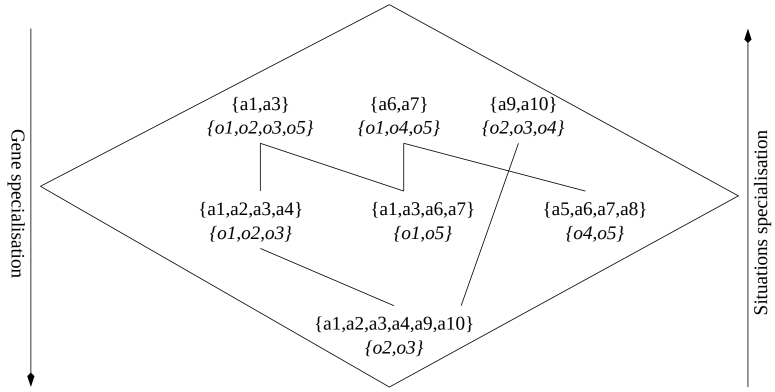
# 3. A NEW METHOD

We have considered the extraction from a transposed matrix using the Galois connection to infer the results that would have been extracted from the initial matrix. Indeed, one can associate to the lattice on genes the lattice on situations. Elements from these lattices are linked by the Galois operators. The Galois connection gives rise to concepts [17] that associate sets of genes with sets of situations, or in the transposed matrix, sets of situations with sets of genes. When we have only few situations and many genes, the transposition enables to reduce the complexity of the search.

*Definition 6.* If $X \in \mathcal{L}_{\mathcal{A}}$ and $T \in \mathcal{L}_{\mathcal{S}}$, we consider the so-called concepts $(X, T)$ where $T = g(X)$ and $X = f(T)$. By construction, concepts are built on closed sets and, each closed set of genes (resp. situations) is linked to a closed set of situations (resp. genes).

*Definition 7.* The concept theory w.r.t. $\mathbf{r}$, $\mathcal{L} = \mathcal{L}_{\mathcal{A}} \times \mathcal{L}_{\mathcal{S}}$, and a constraint $\mathcal{C}$ is denoted $Th_c(\mathcal{L}, \mathbf{r}, \mathcal{C})$.

Figure 1: A Galois lattice




It is the collection of concepts $(X, T)$ such that $X \in \{\varphi \in \mathcal{L}_A \mid \mathcal{C}(\varphi, \mathbf{r}) \text{ satisfied}\}$.

On Figure 1, we provide the so-called Galois lattice for the concepts in the data from Table 1. The specialization relation on the sets of genes which is oriented from the top towards the bottom of the lattice is now associated to a specialization relation on sets of situations which is oriented in the reverse direction. Indeed, if $X \subset Y$ then $g(X) \supseteq g(Y)$.
The collection of the maximally specific sets of genes (e.g., the maximal frequent itemsets) has been called the positive border in [10]. A dual concept is the one of negative border, i.e., the minimally general sets (e.g., the smallest infrequent sets whose every subset is frequent). The lattice is thus split in two parts. On the top, we have the solution which is bordered by the positive border. On the bottom, we have the sets that do not belong to the solution. The minimal elements of this part constitute the negative border. This duality is interesting: borders are related to a specialisation relation and an anti-monotonic constraint. The bottom part of the lattice can be considered as the solution for the negated constraint, the former positive border becomes the negative border and vice versa.
On the Galois lattice, it is possible to perform two types of extraction: one on the gene space (1), starting from the top of the lattice and following the specialisation relation on the genes, and the other one on the biological situations (2), starting from the bottom of the lattice and following the specialisation relation on the situations. We now define the matrix transposition for a matrix $\mathbf{r}$, the constraint transposition for a constraint $\mathcal{C}$ and we state the central result of the complementarity of the extractions.

*Definition 8.* If $\mathbf{r} = (\mathcal{S}, \mathcal{A}, R)$ is an expression matrix, the transposed matrix is ${}^t\mathbf{r} = (\mathcal{A}, \mathcal{S}, {}^tR)$ where $(a, s) \in {}^tR \iff (s, a) \in R$.

Whereas the matrix transposition is quite obvious, it is not the same for the transposition of constraints. In the case of the minimal frequency constraint $\mathcal{C}_{\text{freq}}$, the dual notion of the frequency for the sets of genes is the length of the corresponding sets of situations.

*Definition 9.* Let $\mathcal{C}$ be a constraint on $\mathcal{L}_\mathcal{A}$, its transposed constraint ${}^t\mathcal{C}$ is defined on $\mathcal{L}_\mathcal{S}$ by $\forall T \in \mathcal{L}_\mathcal{S},\ {}^t\mathcal{C}(T, \mathbf{r}) \iff \mathcal{C}(f(T), \mathbf{r})$ where $f$ is the Galois operator. Thus, ${}^t\mathcal{C}_{\text{freq}}(T, \mathbf{r}) \equiv |T| \geq \gamma$ if $\gamma$ is the frequency threshold for $\mathcal{C}_{\text{freq}}$.

With respect to gene specialization, ${}^t\mathcal{C}$ is monotonic (resp. anti-monotonic) if $\mathcal{C}$ is monotonic (resp. anti-monotonic). However, if $\mathcal{C}$ is anti-monotonic (e.g., $\mathcal{C}_{\text{freq}}$) following the gene specialization relation, ${}^t\mathcal{C}$ is monotonic according to the specialization relation on the situations: it has to be negated to get an anti-monotonic constraint that can be use efficiently.

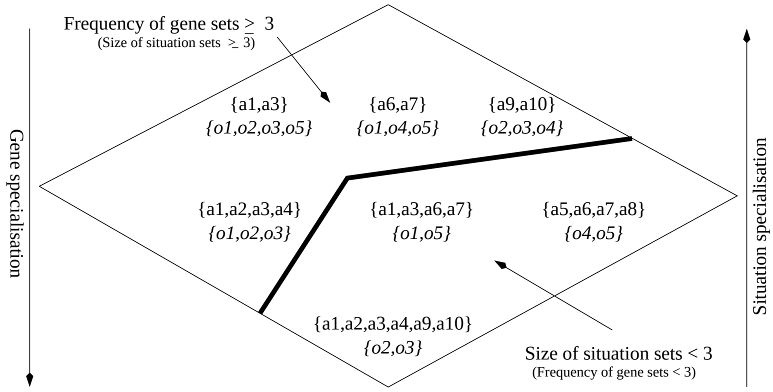
*Property 1.* If $\mathcal{C}$ is anti-monotonic w.r.t. gene specialization, then $\neg{}^t\mathcal{C}$ is anti-monotonic w.r.t. situation specialization.

We have an operation for the transposition of the data and a new anti-monotonic constraint w.r.t. to the specialization relation on the situations. However, to obtain this new anti-monotonic constraint, we had to transpose the original constraint and take its negation: the new extraction turns to be complementary to the collection we would get with the standard extraction.

*Definition 10.* Given ${}^t\mathbf{r}$ and $\neg{}^t\mathcal{C}$, the transposed theory $Th_c(\mathcal{L}, {}^t\mathbf{r}, \neg{}^t\mathcal{C})$ is the transposition of $Th_c(\mathcal{L}, \mathbf{r}, \mathcal{C})$.

*Property 2.* The concept theory $Th_c(\mathcal{L}, \mathbf{r}, \mathcal{C})$ and its transposed theory $Th_c(\mathcal{L}, {}^t\mathbf{r}, \neg{}^t\mathcal{C})$ are complementary w.r.t. the whole collection of concepts.

Figure 2: Complementarity of the extractions




*Example 4.* On the data from Table 1, the sets of genes with a frequency of at least 3 are $\{a_1, a_3\}$, $\{a_6, a_7\}$, $\{a_9, a_{10}\}$, and $\{a_1, a_2, a_3, a_4\}$. A closed set of genes has a frequency greater than 3 if the size of the corresponding situation set is greater than 3. When taking the negation of this constraint, we look for the sets of situations whose size are at most 3 (anti-monotonic constraint w.r.t. the situation specialization). The sets $\{s_1, s_5\}$, $\{s_4, s_5\}$ and $\{s_2, s_3\}$ are extracted. Clearly, the two collections are complementary (see Figure 2).

The correctness of this extraction method for finding the closed sets of genes from the extractions on transposed matrices relies on this complementary property. Due to the lack of space, we consider only a straightforward application of this framework that concerns the computation of concepts.

If we compute the closed sets from the gene space, the Galois connection allows to infer the closed sets of situations. Reciprocally, the extraction on the transposed matrix provides the closed sets on the situations and we can infer the closed sets of genes. Thus, the same collection of closed sets can be extracted from a matrix or its transposed. The choice between one or the other method can be guided by the dimension of the matrix. On the data from Table 1, the smallest dimension concerns the situations (5 elements) and it leads to $2^5 = 32$ possible sets. Among these 32 elements, only 10 are closed. However extracting the closed sets from the original matrix, which contains 10 columns, leads to a search space of $2^{10} = 1024$ sets of genes whereas there is still 10 closed sets. To compute the closed sets, we output the closures of the free sets. This is an efficient solution since $\mathcal{C}_{\text{free}}$ is anti-monotonic. Several free sets can however generate the same closed set. On the data from Table 1, the free set extraction provides 41 sets which generate the 10 closed sets, whereas the transposed matrix extraction provides only 17 free sets. We provide real examples in the next section.

## 4. APPLICATIONS

We have been working on data sets produced with cDNA microarrays at the Stanford Genome Technology Center (Paolo Alto, CA 94306, USA). The first data set is described in [16]. It concerns the study of human insulino-resistance. From 6 cDNA microarrays (around 42 557 spots for 29 308 UniGene clusters), a typical preprocessing for microarray data has given an expression matrix with 6 lines (situations) and 1 065 columns (genes). It is denoted as the `inra` matrix. The second data set concerns gene expression during the development of the drosophila [2]. With the same kind of preprocessing, we got an expression matrix with 162 lines and 1 230 columns denoted `droso`. To derive boolean matrices, we have encoded the over-expression of genes: for each gene $i$, we have computed a threshold $\sigma_i$ under which the attribute boolean value is false, and true otherwise. Different methods can be used for the definition of threshold $\sigma_i$ and we have done as follows [2]: $\sigma_i = Max_i \times (1 - \sigma\_discr)$ where $Max_i$ is the maximal expression value for gene $i$ and $\sigma\_discr$ is a parameter that is common to every genes.

We used the prototype `mv-miner` implemented by F. Rioult and have extracted the closed sets under the frequency threshold 1 to get all of them. These experiments have been performed on a 800MHz processor with RAM 4GB and 3GB for swap (linux operating system). We have used parameter $\sigma\_disc$ to study the extraction complexity w.r.t. the density of the boolean matrices (ratio of the number of true values on the number of values).

First, we compare the extraction in `inra` and $^t$`inra` for a given value of $\sigma\_disc$ (Table 2).

We have 41 closed sets in these boolean matrices. The free set extraction on `inra` provides 667 831 free sets of genes whereas the extraction on $^t$`inra` provides 42 free sets of situations. In

Table 2: Failure/success in pruning for `inra` and ${}^t$`inra`

| | ${}^t$`sain8` | | `sain8` | |
|---|---|---|---|---|
| size | success | failure | success | failure |
| 1 | 6 | 0 | 777 | 0 |
| 2 | 15 | 0 | 172 548 | 128 928 |
| 3 | 16 | 4 | 2 315 383 | 4 713 114 |
| 4 | 6 | 9 | 2 965 726 | 9 371 325 |
| 5 | 0 | 2 | 0 | 1 544 485 |
| Total | 43 | 15 | 5 454 434 | 15 757 852 |
| Nb free sets | 42 | | 667 831 | |
| Nb closed sets | 41 | | | |

Section 2.2, we have seen that algorithms use anti-monotonic constraints in two ways. Criterion 1 avoids to generate some candidates that would have to be pruned. Criterion 2 enables to prune candidates for which the constraint is not satisfied. Checking Criterion 2 is expensive because it needs to store the sets and check the properties of all their subsets. Table 2 provides the number of sets (for each level in the levelwise search) which satisfy these two criteria and the number of sets that have been examined when processing `inra` and ${}^t$`inra`. Extraction in ${}^t$`inra` is clearly more efficient. Not only it provides less candidate sets to test (43 vs. 5 454 434) but also it leads to far less failures: 15 vs. 15 757 852.

We have performed experiments on the two microarray data sets for various discretization thresholds. Considering `droso`, Table 3 shows that extraction becomes feasible for larger densities. It enables that the biologist explore alternatives for discretizations.

Results on `inra` confirm the observations (see Table 4). The difference between standard extraction and transposed matrix extraction is even more spectacular. Indeed, extraction time on the transposed matrix can become negligible w.r.t. the standard extraction time (e.g., 120 ms vs. 368 409 ms). Notice that the number of free sets of genes can be very large w.r.t. the number of closed sets to be found, e.g., 51 881 free sets for only 34 closed sets.

Also, the method has been applied on very large expression matrices derived from human SAGE data (matrix $90 \times 12\,636$) [15]. In these matrices and for different discretization techniques, none of the standard extractions were tractable while extractions on the transposed matrix were easy.

# 5. USING THE CLOSED SETS

An algorithm like `mv-miner` takes a boolean matrix and provides the free sets on the columns and the closed sets on both the lines and the columns with their frequencies in the data. From the closed sets of genes and their frequencies, it is indeed possible to select the frequent closed sets provided a frequency threshold. When using $\mathcal{C}_{\text{freq}}$ with $\gamma > 1$, it is possible to use its transposed constraint and make use of the transposed extractions.

Let us discuss informally how to use the closed sets to derive some knowledge. It is possible to regenerate the whole collection of the frequent sets of genes (resp. situations) from the frequent closed sets of genes (resp. situations). So, the multiples applications of the frequent itemsets are available (e.g., association rule mining, class characterization, some types of clustering, approximation of the joint distribution). Notice also that the extracted collections can be represented as concepts [17] and thus many knowledge discovery tasks based on concept lattices can be considered.

It is quite possible that the number of frequent sets of genes is too huge and that a “blind” regeneration process is not feasible. It is possible to filter at regeneration time, e.g., to take into account some syntactical constraints on the sets of interests. Notice also that the free sets that have been proved useful for non redundant association rule computation are missing. When mining the transposed matrix, we get the free sets on situations but not the free sets on genes.

Let us however sketch typical uses of the extracted patterns. Part of this has been already validated on SAGE data analysis in [5]. It is useful to look at the patterns extracted after several discretizations on the same expression matrix. These extractions can provide different sets of co-regulated genes for the same expression data. So, after such computations, the biologist want to compare different closed set collections, looking at the common patterns, the dissimilarities, etc. Browsing these collections of closed sets can lead to the selection of some of them, e.g., the one that are almost always extracted.

Then, the biologists can select some concepts for an in-depth study of the interactions between the involved genes. Clearly, one objective criterion for selection can be based on the frequency. One important method concerns the use of information sources about gene functions. Quite often, one of the first post-processing is to look

Table 3: Results for the drosophila data

| | $\sigma$_*discr* | density | time (ms) | nb free sets | nb closed sets |
|---|---|---|---|---|---|
| $^t$droso | 0.02 | 0.08 | 160 | 965 | 434 |
| droso | 0.02 | 0.08 | 1 622 | 5 732 | 434 |
| $^t$droso | 0.075 | 0.015 | 420 | 3 667 | 1 508 |
| droso | 0.075 | 0.015 | 35 390 | 60 742 | 1 508 |
| $^t$droso | 0.1 | 0.019 | 721 | 6 890 | 2 569 |
| droso | 0.1 | 0.019 | 146 861 | 162 907 | 2 569 |
| $^t$droso | 0.15 | 0.032 | 4 526 | 36 309 | 10 447 |
| droso | 0.15 | 0.032 | failure | - | - |
| $^t$droso | 0.2 | 0.047 | 36 722 | 410 666 | 4 6751 |
| droso | 0.2 | 0.047 | failure | - | - |
| $^t$droso | 0.25 | 0.067 | 455 575 | 1 330 099 | 259 938 |
| droso | 0.25 | 0.067 | failure | - | - |
| $^t$droso | 0.3 | 0.09 | failure | - | - |
| droso | 0.3 | 0.09 | failure | - | - |

for the homogeneity of the sets (e.g., they all share the same function). It is then quite interesting to focus on almost homogeneous sets of genes and look at the outliers. This approach has been used in the SAGE data analysis described in [5] and has provided a valuable result: one EST (Expressed Sequence Tag) was always co-regulated with a set of around 20 genes that had the same function and it is reasonable to suspect that this EST has that function. For this type of post-processing, it is possible to use the various ontologies that are available, e.g., http://www.geneontology.org/, and, e.g., study the homogeneity of the selected sets of genes at different levels (biological process, molecular function, cellular function).

Last but not the least, the biologist can chose a given closed set of genes $X$ and then project the original expression matrix on $X$. Since the size of a closed set will be generally small w.r.t. the size of whole collection of genes, it is then possible to mine this restricted matrix. For instance, it becomes possible to extract the whole collection of non redundant association rules (free sets of genes in the left-hand side) from this non transposed restricted matrix.

# 6. CONCLUSION

We have been studying the extraction of groups of genes found to be frequently co-regulated in expression matrices. This type of data raises difficult problems due to the huge size of the search space and to the huge size of the solutions. In [5], it has been shown that the use of condensed representations as described in e.g. [6; 7], was useful, at least when the number of biological situations is not too small in light of the number of genes. Unfortunately this situation is rarely observed in most of the available gene expression data. We therefore explored the possibility to process the transposed matrices by making use of properties of the Galois connections. This resulted in a very spectacular improvement of the extraction procedure, allowing to work in context where previous approaches failed. [5] has validated the interest of frequent closed sets in biological terms on a reduced set of genes. We are pretty confident that given the algorithmic breakthrough, biological significant information will be extracted from the expression data we have to mine. We are furthermore exploring the transposition of other constraints.

**Acknowledgements** This work has been partially funded by the EU contract cInQ IST-2000-26469 (FET arm of the IST programme). F. Rioult is supported by the IRM department of the Caen CHU, the comité de la Manche de la Ligue contre le Cancer and the Conseil Régional de Basse-Normandie. The authors thank S. Blachon, O. Gandrillon, C. Robardet, and S. Rome for exciting discussions.

# 8. 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] M. Arbeitman, E. Furlong, F. Imam, E. Johnson, B. Null, B. Baker, M. Krasnow, M. P. Scott, R. Davis, and K. P. White. Gene expression during the life cycle of drosophilia melanogaster. *Science*, 297, 2002.

[3] 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.

Table 4: Results for the human data

| | $\sigma\_discr$ | density | time (ms) | nb free sets | nb closed sets |
|---|---|---|---|---|---|
| $^{t}$`inra` | 0.01 | 0.193 | 80 | 19 | 19 |
| `inra` | 0.01 | 0.193 | 16 183 | 11 039 | 19 |
| $^{t}$`inra` | 0.05 | 0.21 | 90 | 29 | 29 |
| `inra` | 0.05 | 0.21 | 42 991 | 23 377 | 29 |
| $^{t}$`inra` | 0.085 | 0.023 | 110 | 33 | 33 |
| `inra` | 0.085 | 0.023 | 83 570 | 38 659 | 33 |
| $^{t}$`inra` | 0.1 | 0.24 | 120 | 36 | 34 |
| `inra` | 0.1 | 0.24 | 368 409 | 51 881 | 34 |
| $^{t}$`inra` | 0.15 | 0.268 | 120 | 40 | 38 |
| `inra` | 0.15 | 0.268 | failure | - | - |


[4] R. J. Bayardo. Efficiently mining long patterns from databases. In *Proceedings ACM SIGMOD'98*, pages 85–93, Seattle, Washington, USA, 1998. ACM Press.

[5] 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.

[6] J.-F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. In *Proceedings PAKDD'00*, volume 1805 of *LNAI*, pages 62–73, Kyoto, JP, Apr. 2000. Springer-Verlag.

[7] J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by mean of free-sets. In *Proceedings PKDD'00*, volume 1910 of *LNAI*, pages 75–85, Lyon, F, Sept. 2000. Springer-Verlag.

[8] J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets: a condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery journal*, 7(1):5–22, 2003.

[9] R. Godin, R. Missaoui, and H. Alaoui. Incremental algorithms based on galois (concepts) lattices. *Computational Intelligence*, 11(2):246 – 267, 1995.

[10] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery journal*, 1(3):241–258, 1997.

[11] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[12] C. Niehrs and N. Pollet. Synexpression groups in eukaryotes. *Nature*, 402:483–487, 1999.

[13] 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.

[14] J. Pei, J. Han, and R. Mao. CLOSET an efficient algorithm for mining frequent closed itemsets. In *Proceedings ACM SIGMOD Workshop DMKD'00*, Dallas, USA, May 2000.

[15] C. Robardet, F. Rioult, S. Blachon, B. Crémilleux, O. Gandrillon, and J.-F. Boulicaut. Mining closed sets of genes from sage expression matrices: a spectacular improvement. Technical report, LIRIS INSA Lyon, F-69622 Villeurbanne, March 2003. Submitted.

[16] S. Rome, K. Clément, R. Rabasa-Lhoret, E. Loizon, C. Poitou, G. S. Barsh, J.-P. Riou, M. Laville, and H. Vidal. Microarray profiling of human skeletal muscle reveals that insulin regulates 800 genes during an hyperinsulinemic clamp. *Journal of Biological Chemistry*, March 2003. In Press.

[17] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival, editor, *Ordered sets*, pages 445–470. Reidel, 1982.

[18] M. J. Zaki. Generating non-redundant association rules. In *Proceedings ACM SIGKDD'00*, pages 34 – 43, Boston, USA, Aug. 2000. AAAI Press.