Two FCA-Based Methods for Mining Gene Expression Data

Mehdi Kaytoue[1], Sébastien Duplessis[2], Sergei O. Kuznetsov[3], and Amedeo Napoli[1]

[1] Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)
Campus Scientifique, B.P. 235 – Vandœuvre-lès-Nancy – France
mehdi.kaytoueuberall@loria.fr, amedeo.napoli@loria.fr
[2] UMR 1136 Institut National de la Recherche Agronomique (INRA)
Nancy Université – Interactions Arbres/Micro-organismes
54280 Champenoux – France
duplessi@nancy.inra.fr
[3] State University Higher School of Economics
Kirpichnaya 33/5 – 125219 Moscow – Russia
skuznetsov@hse.ru

Abstract. Gene expression data are numerical and describe the level of expression of genes in different situations, thus featuring behaviour of the genes. Two methods based on FCA (Formal Concept Analysis) are considered for clustering gene expression data. The first one is based on interordinal scaling and can be realized using standard FCA algorithms. The second method is based on pattern structures and needs adaptations of standard algorithms to computing with interval algebra. The two methods are described in details and discussed. The second method is shown to be more computationally efficient and providing more readable results. Experiments with gene expression data are discussed.

1 Introduction

Gene expression data (GED) consist of numerical tables with thousands of genes in rows and dozens of biological environments or situations (different cells, times,...) in columns (See Table 1). Each table entry is called an *expression value* and reflects the behaviour of the gene in a row in the situation in column. A whole line is a numerical vector called the *expression profile* of the gene. Genes having similar expression profiles are said to be *co-expressed*. GED analysis is of high interest mainly for the identification of groups of co-expressed genes that are known to possibly interact together within a same biological process [1]. GED analysis is an active area of research involving mainly data-mining methods: many clustering [2], biclustering [3,4] and FCA-based [5,6,7] methods have been recently designed and applied in this domain.

Clustering methods group genes into *clusters* w.r.t. a global similarity, e.g. based on Euclidean distance, of their expression profiles. Clustering may fail to detect biological processes common only to some columns of a dataset [1,3].

S. Ferré and S. Rudolph (Eds.): ICFCA 2009, LNAI 5548, pp. 251–266, 2009.

To overcome this difficulty, biclustering algorithms have been suggested [3,8]. *Biclusters* in GED are defined as groups of genes that have similar expression values in a same group of situations, but not necessarily all. However, we know that most of the genes are involved in several processes [1]: extracted biclusters should overlap. Then extracting "homogeneous" biclusters is difficult as the number of possible groups may grow exponentially. Biclustering algorithms generally extract k best biclusters w.r.t. an evaluation function that relies on heuristics: Their complete enumeration is generally not possible, interesting patterns may also be missed [3,5].

This problem is limited when considering binary GED, i.e. *binary relations* between the set of genes and the set of situations [4,9]. A numerical GED is scaled before binary biclusters are extracted. Intuitively, a bicluster is a rectangle in a binary table (modulo line and column permutations) "completely or mostly" filled with crosses, e.g. like in Table 4. Then a complete enumeration respecting some constraints like closure and minimal frequency is possible [4,5,10]. In [4] the authors have proposed the Bi-Max bi-clustering algorithm, which extracts *inclusion-maximal biclusters* defined as follows: Given m genes, n situations and a binary table e such that $e_{ij} = 1$ or $e_{ij} = 0$ for all $i \in [1, m]$ and $j \in [1, n]$, the pair $(G, C) \in 2^{\{1,\dots,n\}} \times 2^{\{1,\dots,m\}}$ is called an inclusion-maximal bicluster if and only if (1) $\forall i \in G, j \in C : e_{ij} = 1$ and (2) $\nexists (G', C') \in 2^{\{1,\dots,n\}} \times 2^{\{1,\dots,m\}}$ with (a) $\forall i' \in G, \forall j' \in C$: $e_{i'j'} = 1$ and (b) $G \subseteq G' \wedge C \subseteq C' \wedge (G', C') \neq (G, C)$. Note that an inclusion-maximal bicluster is nothing else than a formal concept as defined in [11]. Formal Concept Analysis (FCA) can be viewed as a binary biclustering method: It provides means for extracting local patterns from a binary relation, namely *formal concepts*. In application to GED analysis concept extents are maximal sets of genes related to a common maximal set of situations (not necessarily all) [5,6,7].

However to apply either binary biclustering or FCA-based methods, one needs to scale numerical data. Scaling introduces biases and may result in loss of information [4,5,6,7,12]. Our goal here is to try to avoid these problems by designing an FCA-based method that does not need a scaling, but would benefit from formal and computational framework of FCA.

We propose two mathematically equivalent FCA-based methods for extracting groups of co-expressed genes in numerical GED. Co-expression, or similarity is considered by using interval values in initial data. The first approach uses *interordinal scaling*. This scaling is able to reflect all possible value intervals arising from a numerical dataset by a binary relation without loss of information. However it produces large and dense binary data, which are hard to analyse with existing FCA algorithms. This is probably the reason why it has never been used for GED analysis. The second method relies on pattern structures [13] by extending interval algebra in real numbers [13,14] and does not need any scaling.

We have experimented with both methods, trying to compare the quality of their results and their computational efficiency. We show that both methods extract equivalent sets of patterns, but the method based on pattern structures is more efficient than that based on interordinal scaling, and provides with more

readable and interpretable results. Processing pattern structures needs slight adaptations of the FCA framework and well-known efficient algorithms (see [15] for survey).

In Sections 2-3, we present gene expression data and related work. In Sections 4-5, we introduce and discuss both methods and an interestingness measure that allows one to prune uninteresting groups of genes. In Sections 6-7 we discuss computation and experimental results, and conclusion draws further researches.

2 Gene Expression Data

Gene expression is the mechanism that produces a protein from a gene in two steps. First the transcription builds a copy of a gene called an mRNA. Then the mRNA is translated into a protein. This mechanism differs in different biological situations: for each gene the concentration, i.e. the relative quantity, of mRNA and proteins depends on the current cell, time, etc. and reflects the behaviour of the gene. Indeed, biological processes of a living cell are based on chemical reactions and interactions mainly between proteins and mRNA. Thus, it is a major interest to measure and analyse mRNA and protein concentration to understand biological processes activated in a cell.

Microarray biotechnology is able to measure the concentration of mRNA of a gene into a numerical value called *gene expression value*. This value characterizes the behaviour of a gene in a particular cell. Microarray can monitor simultaneously the expression of a large number of genes, possibly the complete coding space of a genome. When several microarrays are considered, the expression value of a gene is measured in multiple situations or environments, e.g. different cells, time points, cells responding to particular environmental stresses, etc. This characterizes the behaviour of the gene w.r.t. all these situations and is represented by a vector of expression values called a *gene expression profile*.

A *gene expression dataset* (GED) consists of a table with n rows corresponding to genes and m columns corresponding to situations. A table entry is called an *expression value*. A table line is called an *expression profile*. For example, in Table 1, the expression value of g_1 in the situation s_1 is 5 and the expression profile for the gene g_1 is $\langle 5, 7, 6\rangle$. In this paper, we consider the NimbleGen Systems Oligonucleotide Arrays technology[1]: expression values are ranged from 0 (not expressed) to 65535 (highly expressed).

3 Related Work

In this paper we discuss methods of extracting co-expressed groups of genes, i.e. sharing similar numerical values in some or all situations. Methods of these kind allow discovering and describing biological processes in living cells [1].

For most of the binary biclustering methods, an *l-cut* scaling is operated by using a single threshold l on expression values determined for each object (see

[1] Details on this biotechnology can be found at http://www.nimblegen.com/

[4,7,16] for threshold definitions). Expression values greater than this threshold are said to be over-expressed and encoded by 1, otherwise by 0. Then strong relations are extracted from the binary table representing genes simultaneously over-expressed. In [6], we proposed a kind of generalization[2] with an interval-based scaling of numerical data, where interval number and size were chosen by experts. Given a set of genes G, a set of situations S and a set of ordered intervals T, $(g,(s,t)) \in I$, where $g \in G$, $s \in S$, $t \in T$ and I and binary relation, means that the expression value of the gene g is the interval of index t for the situation s. Formal concepts of the context $(G, S \times T, I)$ represent groups of genes whose expression values are in same intervals for a subset of situations (may be for all situations), however these intervals are hard to determine *a priori*.

In [17], the authors present an FCA-based method to mine numerical data that does not need any scaling procedure. This is a similar approach of the two equivalent methods presented in this paper as extracted patterns are composed of intervals arising from the data whose size is less than a given parameter (see Section 6). However, in [17], no algorithm for dealing with large data was proposed and no link to interordinal scaling was made. A similar approach for the case of logical formulas was realized in [18] and [19].

4 Mining GED by Means of Interordinal Scaling

4.1 FCA: Main Definitions

Here we use standard definitions from [11]. Let G and M be arbitrary sets and $I \subseteq G \times M$ be an arbitrary binary relation between G and M. The triple (G, M, I) is called a formal context. Each $g \in G$ is interpreted as an object, each $m \in M$ is interpreted as an attribute. The fact $(g, m) \in I$ is interpreted as "g has attribute m". The two following derivation operators $(\cdot)'$ are considered:

$$A' = \{m \in M \mid \forall g \in A : gIm\} \qquad for\ A \subseteq G,$$
$$B' = \{g \in G \mid \forall m \in B : gIm\} \qquad for\ B \subseteq M$$

which define a Galois connection between the powersets of G and M. For $A \subseteq G$, $B \subseteq M$, a pair (A, B) such that $A' = B$ and $B' = A$, is called a *(formal) concept*. Concepts are partially ordered by $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$ ($\Leftrightarrow B_2 \subseteq B_1$). With respect to this partial order, the set of all formal concepts forms a complete lattice called the *concept lattice* of the formal context (G, M, I). For a concept (A, B) the set A is called the *extent* and the set B the *intent* of the concept. Certain data are not given directly by binary relations, they require transformation to contexts, called conceptual scaling. The choice of a scale is done w.r.t. data and goals and directly affects the size and interpretation of resulting concept lattice.

[2] Using a threshold θ is equivalent to considering the interval [θ, maximum_attribute_value].

Table 1. A gene expression data (GED)

	s_1	s_2	s_3
g_1	5	7	6
g_2	6	8	4
g_3	4	8	5
g_4	4	9	8
g_5	5	8	5

Table 2. Scale $\mathbb{I}_N := (N, N, \leq)|(N, N, \geq)$ for s_1, $N = \{4, 5, 6\}$

	$s_1 \leq 4$	$s_1 \leq 5$	$s_1 \leq 6$	$s_1 \geq 4$	$s_1 \geq 5$	$s_1 \geq 6$
4	×	×	×	×		
5		×	×	×	×	
6			×	×	×	×

4.2 Interordinal Scaling for GED

Let G be a set of genes, S a set of situations, $W \subset \mathbb{R}$ a set of expression values and I_1 a ternary relation defined on the Cartesian product $G \times S \times W$, then $\mathbb{K}_1 = (G, S, W, I_1)$ is the many-valued context representing a GED. $(g, s, w) \in I_1$ or simply $g(s) = w$ means that the gene g has an expression value w for the situation s. In the example of Table 1, $G = \{g_1, g_2, g_3, g_4, g_5\}$, $S = \{s_1, s_2, s_3\}$, and I_1 is illustrated, for example, by $g_1(s_1) = 5$, i.e. $(g_1, s_1, 5) \in I_1$. Here the objective is to extract formal concepts (A, B) from $\mathbb{K}_1$, where $A \subseteq G$ is a subset of genes that share "similar values" of W, i.e. lying in a same interval with borders arising from the data in the situations of $B \subseteq S$. To this end, we use an appropriate scale to build the derived formal context $\mathbb{K}_2 = (G, S_2, I_2)$.

A scale is a formal context (cross-table) taking original attributes of $\mathbb{K}_1$ with the derived ones of $\mathbb{K}_2$, i.e. a "plan" to construct $\mathbb{K}_2$. As attributes do not take necessarily same values, each of them is scaled *separately*. $W_s \subseteq W$ is the set of values for the attribute s and is defined for each $s \in S$ as follows: $W_s \subseteq W$ and $(g, s, w) \in I_1 \Longrightarrow w \in W_s, \forall g \in G$.The following interordinal scale (see pp. 42 in [11]) can be used to represent all possible intervals of attribute values:

$$\mathbb{I}_{W_s} = (W_s, W_s, \leq)|(W_s, W_s, \geq)$$

Indeed, the extents of this scale are value intervals. $\mathbb{I}_{W_s}$ is given for the many-valued attributes s_1 in Table 2, where $W_{s_1} = \{4, 5, 6\}$.

Once a scale is chosen, conceptual scaling consists in replacing each many-valued attribute of $\mathbb{K}_1$ by a certain number of attributes to construct $\mathbb{K}_2$ w.r.t. the chosen scale. Here each many-valued attribute s is replaced by $2 \cdot |W_s|$ one-valued attributes with names "$s \leq w$" and "$s \geq w$", for all $w \in W_s$. For example, many-valued attribute s_1 is replaced by the following attributes $\{s_1 \leq 4, s_1 \leq 5, s_1 \leq 6, s_1 \geq 4, s_1 \geq 5, s_1 \geq 6\}$. Derived context $\mathbb{K}_2 = (G, S_2, I_2)$ is given in Table 3 for the attribute s_1 only. Note that this transformation is without loss of information: the many-valued context can easily be reconstructed from the formal context. For example, derived attributes for $(g_1, s_1, 5)$ are $s_1 \leq 5$, $s_1 \leq 6$, $s_1 \geq 4$, $s_1 \geq 5$. The only value in W_{s_1} respecting these predicates is 5 which is the original value.

Density of a formal context (G, M, I) is defined as the proportion of elements of I w.r.t. the size of the Cartesian product $G \times M$, i.e. density $d = |I|/(|G|.|S|)$. In the case of interordinal scaling, density of derived context $\mathbb{K}_2$ is

Table 3. $\mathbb{K}_2 = (G, S, I_2)$ for the attribute s_1

	$s_1 \leq 4$	$s_1 \leq 5$	$s_1 \leq 6$	$s_1 \geq 4$	$s_1 \geq 5$	$s_1 \geq 6$
g_1		×	×	×	×	
g_2			×	×	×	×
g_3	×	×	×	×		
g_4	×	×	×	×		
g_5		×	×	×	×	

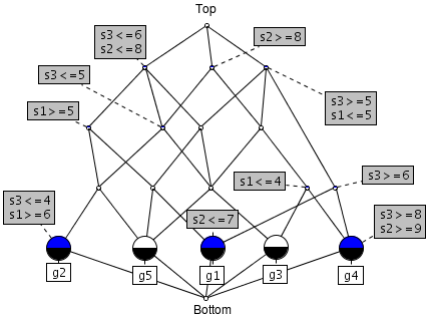

Fig. 1. Concept lattice of formal context $\mathbb{K}_2 = (G, S, I_2)$

$d = \frac{\sum_{i=1}^{i \leq p} |W_i| + 1}{2 \cdot \sum_{i=1}^{i \leq p} |W_i|}$. When $|W|$ grows, d tends towards 50%. Moreover, the number of derived attributes is $2 \cdot \sum_{i=1}^{i \leq p} |W_i|$ and $|g'| = |W| + 1$ for all $g \in G$. This makes the derived contexts dense, large and difficult to process. For comparison, density of binary data in [4] never exceeds 6% and the number of derived attributes remains the same after scaling.

Then the concept lattice of $\mathbb{K}_2$ is given in Figure 1. Concept extents near Bottom concept contain few genes, since the corresponding intents are related to the smallest intervals. Top concept extent is composed of all genes as its intent correspond to intervals of maximal length. The higher a concept lies in the diagram, the larger is the interval corresponding to its intent. Concepts near Top are not interesting: they allow for almost all possible values of attributes. In Section 5.3 we discuss how to select most interesting concepts.

5 Mining a GED with Interval Pattern Structures

Now we suggest how to equivalently mine a gene expression data as a pattern structure avoiding scaling. First we give a general intuition of interval pattern structures that are formally defined and illustrated latter.

For a many-valued context (G, S, W, I), an object $g \in G$ admits a unique description $\langle [a_1, b_1], \ldots, [a_i, b_i], \ldots, [a_p, b_p] \rangle$, where $p = |S|$. Each attribute (or situation in gene expression analysis) value is an interval (may be consisting of one point) given by its left and right limits. In our example, description of the object g_1 is $\langle [5, 5], [7, 7], [6, 6] \rangle$. A set of objects also admits a description of the

form $\langle [a_1, b_1], \ldots, [a_i, b_i], \ldots, [a_p, b_p] \rangle$, where for all objects of the set, the values of all attributes lie within respective intervals. For our example, description of the set $\{g_1, g_2\}$ is $\langle [5, 6], [7, 8], [4, 6] \rangle$. This description is shared by all objects having attribute values in respective intervals, in our example by objects of the set $\{g_1, g_2, g_5\}$. The object g_3 is not contained in this set, because $g_3(s_1) = 4$ and $4 \notin [5, 6]$. The whole set of object sharing a description is closed w.r.t. a closure operator. To formalize this construction one starts from interval algebra on numbers and respective partial order on intervals. Two descriptions are comparable if all intervals of one description are contained in those of the other one, incomparable otherwise.

5.1 General Definition of Pattern Structures

Formally, let G be a set (interpreted as a set of objects), let $(D, \sqcap)$ be a meet-semilattice (of potential object descriptions) and let $\delta : G \longrightarrow D$ be a mapping. Then $(G, \underline{D}, \delta)$ with $\underline{D} = (D, \sqcap)$ is called a *pattern structure*, and the set $\delta(G) := \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$, of $(D, \sqcap)$. Thus each $X \subseteq \delta(G)$ has an infimum $\sqcap X$ in $(D, \sqcap)$ and $(D_\delta, \sqcap)$ is the set of these infima. Each $(D_\delta, \sqcap)$ has both lower and upper bounds, resp. 0 and 1. Elements of D are called *patterns* and are ordered by subsumption relation $\sqsubseteq$: given $c, d \in D$ one has $c \sqsubseteq d \Longleftrightarrow c \sqcap d = c$.

A pattern structure $(G, \underline{D}, \delta)$ gives rise to the following derivation operators $(\cdot)^\square$:

$$A^\square = \prod_{g \in A} \delta(g) \qquad for\ A \subseteq G,$$

$$d^\square = \{g \in G | d \sqsubseteq \delta(g)\} \qquad for\ d \subseteq D.$$

These operators form a Galois connection between the powerset of G and $(D, \sqsubseteq)$. *Pattern concepts* of $(G, \underline{D}, \delta)$ are pairs of the form (A, d), $A \subseteq G$, $d \in D$, such that $A^\square = d$ and $A = d^\square$. For a pattern concept (A, d) the component d is called a *pattern intent* and is a description of all objects in A, called *pattern extent*. Intuitively, (A, d) is a pattern concept if adding any element to A changes d through $(\cdot)^\square$ operator and equivalently taking $e \supset d$ changes A. Like in case of formal contexts, for a pattern structure $(G, \underline{D}, \delta)$ a pattern $d \in D$ is called *closed* if $d^{\square\square} = d$ and a set of objects $A \subseteq G$ is called *closed* if $A^{\square\square} = A$. Obviously, pattern extents and intents are closed.

5.2 Interval Patterns

In a pattern structure, objects have descriptions from a complete semilattice $(D, \sqcap)$, where the operation $\sqcap$ is idempotent, commutative and associative and returns "similarity" of its arguments. Here we consider a similarity operation $\sqcap$ based on an interval algebra on real numbers. For two intervals $[a, b]$ and $[c, d]$, with $a, b, c, d \in \mathbb{R}$ and $a \leq c$, we define their meet $[a, b] \sqcap [c, d]$ as $[min(a, c), max(b, d)]$. Then

$$[a,b] \sqsubseteq [c,d] \iff [a,b] \sqcap [c,d] = [a,b]$$
$$\iff [min(a,c), max(b,d)] = [a,b] \iff a \leq c \quad and \quad b \geq d.$$

Note that in contrast to usual intuition, this definition means that smaller intervals subsume larger intervals.

An interval pattern structure $(G, (D, \sqcap), \delta)$ for a many-valued context (G, M, W, I) with $W \subseteq \mathbb{R}$ is composed of a set of objects G, a meet semilattice $(D, \sqcap)$ and $\delta : G \longrightarrow D$ a mapping that associates a description to a set of genes. The elements of D, called interval patterns, are vectors of p intervals, each interval staying for an attribute or situation. The order on elements of D is given by the natural subsumption order. For interval pattern descriptions $c = \langle [a_i, b_i] \rangle_{i \in [1,p]}$ and $d = \langle [c_i, d_i] \rangle_{i \in [1,p]}$:

$$c \sqsubseteq d \iff c \sqcap d = c$$
$$\iff a_i \leq c_i \quad and \quad b_i \geq d_i, \forall i \in [1,p]$$

The first operator of the Galois connection takes a set of objects (genes in our application) to their common description, which is a vector of intervals (of gene expression values). Consider two objects g_1 and g_2 with $\delta(g_1) = \langle [a_i, b_i] \rangle_{i \in [1,p]}$ and $\delta(g_2) = \langle [c_i, d_i] \rangle_{i \in [1,p]}$, then

$$\{g_1, g_2\}^{\square} = \bigsqcap \delta(\{g_1, g_2\}) = \delta(g_1) \sqcap \delta(g_2)$$
$$\{g_1, g_2\}^{\square} = \langle [min(a_i, c_i), max(b_i, d_i)] \rangle_{i \in [1,p]}$$

The second derivation operator takes a description (vector of intervals of gene expression values) to the set of all objects sharing this description (set of genes whose expression values are within intervals of the description for each attribute). Consider $d \in D$, a pattern such that $d = \langle [a_i, b_i] \rangle_{i \in [1,p]}$, then

$$d^{\square} = \{g \in G \mid d \sqsubseteq \delta(g)\}$$
$$d^{\square} = \{g \in G \mid d \sqcap \delta(g) = d\}$$
$$d^{\square} = \{g \in G \mid \delta(g) = \langle [c_i, d_i] \rangle_{i \in [1,p]}, a_i \leq c_i \; and \; b_i \geq d_i, \; \forall i \in [1,p]\}$$

For a many-valued context (G, M, W, I) with $W \subset \mathbb{R}$ consider the respective pattern structure $(G, (D, \sqcap), \delta)$ on intervals, the interordinal scaling $\mathbb{I}_{W_s} = (W_s, W_s, \leq) \mid (W_s, W_s, \geq)$ from the previous Section, and the context K_I resulting from applying interordinal scaling $\mathbb{I}_{W_s}$ to (G, M, W, I). Consider usual derivation operators $(\cdot)'$ in context K_I. Then the following obvious proposition establishes an isomorphism between the concept lattice of K_I and the pattern concept lattice of $(G, (D, \sqcap), \delta)$.

Proposition. Let $A \subseteq G$. Subset A is an extent of the interval pattern structure $(G, (D, \sqcap), \delta)$ iff $A \subseteq G$ is a concept extent of the context K_I. Moreover, $A^{\square} = \langle [\underline{m}_i, \overline{m}_i] \rangle_{i \in [1,p]}$ iff for all $i \in [1,p]$ $\underline{m}_i$ is the largest number n such that the attribute $\geq n$ is in A' and $\overline{m}_i$ is the smallest number n such that the attribute $\leq n$ is in A'.

Consider an example of pattern concept: $(\{g_1, g_2, g_5\}, \langle[5,6],[7,8],[4,6]\rangle)$, the equivalent concept of the interordinally scaled context is $(\{g_1, g_2, g_5\}, \{s_1 \leq 6, s_1 \geq 4, s_1 \geq 5, s_2 \geq 7, s_2 \leq 8, s_2 \leq 9, s_3 \leq 6, s_3 \leq 8, s_3 \geq 4\})$. The top pattern concept is $(G, \langle[4,6],[7,9],[4,8]\rangle)$. The higher is a concept in the lattice diagram, the larger are the intervals of its pattern intent, in particular, the top pattern concept has maximal intervals. In the next section we consider the problem of selecting most interesting concepts given by small intervals.

5.3 Interestingness of a Pattern Concept

The main goal of GED analysis is extracting homogeneous groups of genes, i.e. groups of genes having similar expression values. Descriptions of homogeneous groups should be composed of intervals with "small" sizes. This can be easily expressed in terms of interval-based patterns. Consider parameter max_{size} that specifies the maximal length of an interval to allow for the whole description represent a homogeneous group of genes. Then in our experiments we may restrict only to pattern concepts with pattern intents $c = \langle[a_i, b_i]\rangle_{i\in[1,p]} \in D$ satisfying the constraint: $\exists i \in [1,p]\ (b_i - a_i) \leq max_{size}$. A stricter constraint would be $\forall i \in [1,p]\ (b_i - a_i) \leq max_{size}$.

Since both constraints are monotone (if an extent does not satisfy it, than a larger intent does not satisfy it too), the subsets of patterns satisfying any of these constraints make an order ideal of the lattice of pattern intents. In terms of computation, using any of these constraints means that only some lower part of the pattern lattice is computed, with patterns satisfying the constraints.

Another possibility is to consider additional $*$-values of interval descriptions replacing intervals, whose lengths exceed threshold max_{size}. So, if we choose $max_{size} = 1$ in our example, then $\{g_1, g_2\}^\square = \langle[5,6],[7,8],*\rangle$ and $\{g_1, g_4\}^\square = \langle[4,5],*,*\rangle$.

6 Computation

Many algorithms for generating formal concepts from a formal context are known, see e.g. a performance comparative [15]. Two families of algorithms are distinguished: incremental and batch ones. At the i^{th} step an incremental algorithm builds the set of concepts for i first objects. Batch algorithms generate sets of concepts from scratch, in a top-down way (resp. bottom up) or from maximal to minimal intents (resp. from minimal to maximal intents). Experimental results of [15] highlight *Norris* (incremental), *CloseByOne* and *NextClosure* (both bottom up) algorithms as best algorithms when the context is dense and large, which is the case of interordinal derived formal contexts.

To compute pattern concepts, one needs generating infima of subsets of D_δ. To this end we have chosen the standard FCA algorithms *Norris*, *CloseByOne*, and *NextClosure*, which need only slight modifications for computing in pattern structures [13]. Computing $A^\square$ for a set $A \subseteq G$ is realized by taking *min* (resp. *max*) of all left (resp. right) limits of the intervals of each object description.

For a pattern $d \in D$, $d^{\square}$ is computed by testing for each object $g \in G$ if each interval of its description is included in the corresponding interval of d.

Worst-case upper bound time complexity of the three highlighted algorithms for computing in a formal context (G, M, I) is $O(|G|^2 \cdot |M| \cdot |L|)$ with G the set of genes, M the set of attributes and L the set of generated concepts [15]. The worst-case time complexity of computing the set of interval pattern structures is $O(|G|^2 \cdot p \cdot |L|)$, where p is the number of components in a description. In both cases, the sets G and L are the same, thus relative efficiency of processing both data representations depends on the number of different attribute values. For a large number of values the time for computing concepts for the interordinally scaled context may be too large. A projection should reduce the number of different attribute values, and also the number of concepts. A simple way is to round real attribute values to n digits after the comma or to a multiple of 10. A direct consequence of this transformation is uncontrolled loss of information which we would like to avoid. However, in this case we just loss precision on attribute values that has limited consequences compared to the binary transformations presented in Related work.

7 Experiments and Results

7.1 Data

Biologists at the UMR IAM (INRA) study interactions between fungi and trees. They recently published the complete genome sequence of the fungus *Laccaria bicolor* [20]. This fungus lives in symbiosis with many trees of boreal, montane and temperate forests. The fungus forms a mixt organ on tree roots and is able to exchange nutrients with its host in a specific symbiotic structure called ectomycorrhiza, contributing to a better tree growth and enhancing forest productivity. On the other hand, the plant retributes its symbiotic partner by providing carbohydrates, allowing the fungus to complete its biological cycle by producing fruit-bodies (e.g. mushrooms). It is thus a major interest to understand how the symbiosis performs at the cellular level. The genome sequence of *Laccaria bicolor* contains more than 20,000 genes [20]. It remains now to study their expression in various biological situations in order to help to understand their roles and functions in the biology of the fungus. Microarray-based gene expression study of different situations is a solution of choice. For example, it enables to compare expression values of all the genes between contrasted situations like free-living cells of the fungus (i.e. mycelium), cells engaged in the symbiotic association (i.e. ectomycorrhiza), and specialized cells forming the fruit-body structure (i.e. mushroom). A *Laccaria bicolor* gene expression dataset is available at the Gene Expression Omnibus of the National Center for Biotechnology Information (NCBI)[3]. It is composed of 22,294 genes in lines and 5 various biological situations in columns, reflecting cells of the organism in various stages of its biological cycle, i.e. free living mycelium (situation FLM), symbiotic tissues (situations MP and MD) or fruiting bodies (situations FBe and FBl).

[3] `http://www.ncbi.nlm.nih.gov/geo/` as series GSE9784

7.2 Biological Experiments

First, a selection from the 22,294 genes is processed. It consists by removing genes having no significant difference of expression across all situations. For each couple of situation, a t-test is performed and a p-value is attributed. If the p-value > 0.05 (cut-off classically applied in biology) for all couples of situations then the current gene is removed from the dataset. Indeed, a gene that shows similar expression values in all situations presents less interest to the biologist than a gene with high differences of expression. Significant changes in gene expression may reflect a role in a biological process and such genes can help the biologist to draw hypotheses. The CyberT tool, available at http://cybert.microarray.ics.uci.edu/, was used to filter the dataset and obtain 10,225 genes.

Another classical processing in GED analysis is to turn values into log_2. Indeed, it allows to capture small expression values into intervals that should be larger for high expression values. Finally, the projection consists in rounding log_2 expression values to one digit after the comma.

We ran the *CloseByOne* algorithm on the obtained interval pattern structure. In this settings, we set $max_{size} = 0.35$. We choose to retain a concept iff its minimal support is greater than 10. Indeed, let us recall that concepts near Bottom, i.e. in the lowest levels of the concept lattice, are composed of a few genes. We also choose to retain a concept iff its description d has a size $|d| \geq 4$: its extent is composed of genes having similar values in at least four situations. Processing lengths 4.2 hours and returns $91,681$ concepts.

Here we present two extracted patterns that group genes with high expression levels in the fruit-bodies situations, whereas their expression remains similar between the mycelium and symbiosis situations (Figure 2). In both patterns, the levels measured are about twice higher in the fruit-body compared to the other situations indicating that these genes correspond to biological functions of importance at this stage.

The first pattern contains 7 genes, of which only 3 possess a putative cellular function assignment based on similarity in international gene databases at NCBI. Interestingly, these genes all encode enzymes involved in distinct metabolic pathways. A gene encodes a 1-pyrroline-5-carboxylate dehydrogenase

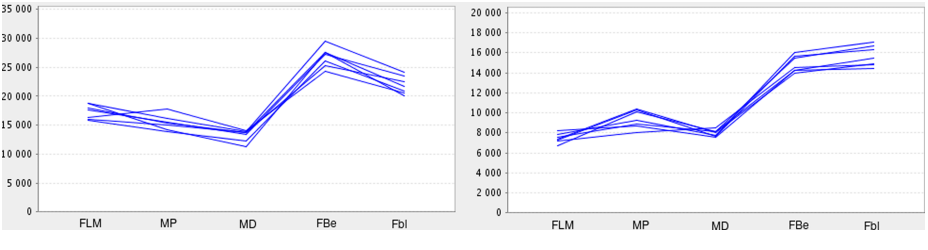

Fig. 2. Graphical visualisation of two extracted concepts. X-axis is composed of situations, Y-axis is the expression values axis. Each line denotes the expression profile of a gene in the concept extent. Values are taken before the logarithmic transformation.

which is involved in amino-acid metabolism, another correspond to an acyl-coA dehydrogenase, involved in fatty acid metabolism and a last gene encodes a transketolase, an enzyme involved in the pentose phosphate pathway of carbohydrate metabolism. All these metabolic functions are essential for the fungus and reflect that the fruit-body is a highly active tissue. The fruit-body is a specific fungal organ that differentiate in order to produce spores and that further ensure spore dispersal in nature [21]. Previous gene expression analyses of the fruit-body development conducted in the ectomycorrhizal fungus *Tuber borchii* also reported the strong induction of several genes involved in carbon and nitrogen metabolisms [22] as well as in lipid metabolism [23]. The present results are consistent with these observations and supports an important mobilization of nutrient sources from the mycelium to the fruit-body. It seems obvious that the primary metabolism requires to be adapted to use these sources in order to properly build spores and provide spore-forming cells with nutrients [21].

The second pattern also contains 7 genes, of which only 3 possess a putative biological function. Interestingly, one of these genes encodes a pseudouridylate synthase, an enzyme involved in nucleotide metabolism that might also be involved in remobilization of fungal components from the mycelium to spore-forming cells and spores. The 2 other genes encode a cytoskeleton protein (actin) and a protein related to autophagy (autophagy-related 10 protein), a process that can contribute to the recycling of cellular material in developing tissues. Both functions participate to reconstruction cellular processes [21], which is consistent with involvement of metabolic enzymes in remobilization of fungal resources towards the new organ in development.

Analysis of these two patterns that present a high expression level in the fruit-body situation is highly informative, comforts existing knowledge in the field and highlights the importance of remobilization in the developing organ. These co-expressed genes share related roles in a particular process. This could indicate that they are under the control of common regulators of gene expression. Interestingly, these patterns also contained a total of 8 genes of unknown functions, i.e. for which no functional assignment was possible in international gene databases. There were 4 genes encoding hypothetical proteins with an homology in databases but no detailed function and 4 genes not previously described in fungi or other organism and which are considered specific to *Laccaria bicolor*. There are about 30% of such genes specific to this fungus and these may play specific roles in the biology of this soil fungus [20]. All these genes show consistent profiles with those encoding metabolic functions. Thus, these genes are interesting investigation leads as they may contain new enzymes not previously described of the pathways or eventual regulator of the cellular process. Altogether, these results contribute to a better understanding of the molecular processes underlying the fruit-body development.

7.3 Computer Experiments

Now we compare time performance and memory usage of three algorithms to equivalently mine interordinal formal contexts (Section 4) and interval pattern

structures (Section 5). We have implemented the *Norris*, *NextClosure*, and *CloseByOne* algorithms, for both processing formal contexts and pattern structures. We have added the Charm algorithm [24] that extracts closed itemsets, i.e. concept intents, in a formal context. FCA algorithms have been implemented in original versions, plus the stack optimization for *NextClosure* and *CloseByOne* as described in [15]. For interval pattern structures, we operate sligth modifications. Charm algorithm is run with the Coron Toolkit [25]. All implementations are in Java: sets of objects and binary attributes are described with the BitSet class and interval descriptions with standard double arrays. Computation was run on Ubuntu 8.10 OS with Intel Core2 Quad CPU 2.40 Ghz and 4 Go RAM.

We tried to compare algorithms on the data presented in biological experiments, i.e. an interval pattern structure from a many-valued context (G, S, W, I_1) where $|G| = 10,225$ and $|S| = 5$. Even with projections, computation is infeasible. Indeed we do not consider here constraints like the maximal interval size: we compute all infima of subsets of D_δ. Then we randomly selected samples of the data, by increasing the number of objects. As attribute values are real numbers with about five digits after the comma, the size of W is large. In worst case, $|W| = |G| \times |S|$, i.e. each attribute value is different in the dataset. This implies very large formal contexts to process and a large number of concepts. We compare time usage for this case, see Table 4. *Norris* algorithm draws best results in formal contexts, which meets conclusions of [15] for large and dense contexts. However, *CloseByOne* in pattern structures is better, and most importantly is the only one that enables computation of very large collection of concepts.

When strongly reducing the size of W by rounding attribute values to the integer, i.e. $|W| << |G| \times |S|$, the *Charm* algorithm outperforms the others. The *Norris* algorithms is still the best FCA-algorithms in formal contexts and *CloseByOne* the best in pattern structures (see Table 5).

Table 4. Generation time in both data representations (no projection)

Datasets							
$\|G\|$	10	20	30	40	50	75	100
$\|W\|$	50	100	150	199	249	374	252
density	51.00%	50.50%	50.33%	50.25%	50.20%	50.13%	50.20%
Generation time in formal contexts (in milliseconds)							
Charm	60	916	16,469	N/A	N/A	N/A	N/A
Next Closure	5	145	1,299	12,569	68,969	N/A	N/A
Norris	**2**	**90**	**609**	**5,180**	**28,831**	N/A	N/A
Close By One	3	106	906	7944	41,238	N/A	N/A
Generation time in pattern structures (in milliseconds)							
Next Closure	6	100	763	5,821	35,197	N/A	N/A
Norris	6	172	1982	15,522	83,837	N/A	N/A
Close By One	**2**	**85**	**585**	**3,094**	**18,320**	**1,004,073**	**2,288,200**
Concept set L							
$\|L\|$	280	9,587	78,173	455,008	1,857,725	40,325,176	64,571,385

Table 5. Generation time in both data representations. Attribute values are rounded.

Datasets							
$\|G\|$	25	50	75	100	125	150	200
$\|W\|$	34	37	44	53	58	62	66
Generation time for formal contexts (in milliseconds)							
density	51.47%	51.35%	51.14%	50.94%	50.86%	50.81%	50.76%
Charm	55	**154**	**184**	**243**	**394**	**936**	**1856**
Next Closure	100	933	3,333	22,973	30,854	78,790	593,416
Norris	**38**	320	861	2,697	5,954	15,359	46,719
Close By One	84	483	2,424	8,452	22,173	59,070	227,432
Generation time for pattern structures (in milliseconds)							
Next Closure	59	372	1,924	6,215	15,417	42,209	143,501
Norris	44	479	2,602	7,243	16,257	40,991	109,814
Close By One	**40**	**220**	**1,084**	**3,832**	**9,289**	**23,989**	**89,804**
Concept set L							
$\|L\|$	1,165	5,928	23,962	48,176	73,463	163,316	252,515

Then, when the number of attribute values w.r.t. $|G| \times |S|$ is low, computing concepts in formal contexts is more efficient. For large datasets with many different attribute values, it is more efficient to compute with pattern structures. The explanation is that for formal concepts the object intent representation is a bit string whose length increases with the growth of $|W|$. Object descriptions in pattern structure are arrays of constant size w.r.t. $|W|$. Therefore, processing them uses less memory for datasets with high number of attribute values.

8 Conclusion

In this paper we discussed FCA-based methods for mining complex data like gene expression data. We compared two mathematically equivalent methods for processing numerical intervals: the first one using interordinal scaling and classical FCA algorithms, and the second one which relies on interval pattern structures. Pattern structures offer more concise representation, better scalability, and better readability of the (pattern) concept lattice. Experiments show that when the number of distinct attribute values is large, adaptation of *ClosebyOne* to pattern structures is most efficient. We also confirmed a general conclusion of [15] for the case of interordinal scaled contexts of our dataset, stating that the *Norris* algorithm is more efficient than *NextClosure* and *CloseByOne* when only the set of concepts needs to be generated, not the covering relation of the lattice.

We have shown how algorithms for processing interval pattern structures can be adapted for particular data and goals. Indeed, the first introduced order of description elements generates all possible descriptions w.r.t. the similarity operation. For GED analysis we have made some propositions to retain "best" concepts. Many other possibilities should be investigated. Another direction of

further research may be models accounting for domain knowledge. The semilattice of descriptions $(D, \sqcap)$ may be viewed as an attribute hierarchy, where domain knowledge (e.g. known functions of genes) may be encoded.

Acknowledgments

The third author was supported by the project of the Russian Foundation for Basic Research, grant no. 08-07-92497-NTsNIL_a. Other authors were supported by the "Contrat de Plan Etat Région – Modélisation des biomolécules et leurs intéractions" (Lorraine, France, 2007–2013).

References

1. Stoughton, R.B.: Applications of DNA microarrays in biology. Annual Review of Biochemistry 74(1), 53–82 (2005)
2. Jiang, D., Tang, C., Zhang, A.: Cluster analysis for gene expression data: a survey. IEEE Trans. on Knowledge and Data Engineering 16(11), 1370–1386 (2004)
3. Madeira, S., Oliveira, A.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1(1), 24–45 (2004)
4. Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)
5. Blachon, S., Pensa, R., Besson, J., Robardet, C., Boulicaut, J.F., Gandrillon, O.: Clustering formal concepts to discover biologically relevant knowledge from gene expression data. In Silico Biology 7(4–5), 467–483 (2007)
6. Kaytoue, M., Duplessis, S., Napoli, A.: Using formal concept analysis for the extraction of groups of co-expressed genes. In: An, L.T.H., Bouvry, P., Tao, P.D. (eds.) Modelling, Computation and Optimization in Information Systems and Management Sciences (2008)
7. Motameny, S., Versmold, B., Schmutzler, R.: Formal concept analysis for the identification of combinatorial biomarkers in breast cancer. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 229–240. Springer, Heidelberg (2008)
8. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. 8th International Conference on Intelligent Systems for Molecular Biology (ISBM), pp. 93–103 (2000)
9. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(suppl. 1), S136–S144 (2002)
10. Robardet, C., Pensa, R.G., Besson, J., Boulicaut, J.-F.: Using classification and visualization on pattern databases for gene expression data analysis. In: Proceedings of the Intl. Workshop on Pattern Representation and Management, Heraklion, Hellas, March 18 (2004)
11. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1999)
12. Choi, V., Huang, Y., Lam, V., Potter, D., Laubenbacher, R., Duca, K.: Using formal concept analysis for microarray data comparison. J. Bioinform. Comput. Biol. 6(1), 65–75 (2008)

13. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) ICCS 2001. LNCS, vol. 2120, pp. 129–142. Springer, Heidelberg (2001)
14. Kuznetsov, S.O.: JSM-method as a machine learning method. Itogi Nauki i Tekhniki, ser. Informatika 15, 17–50 (1991)
15. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. J. Exp. Theor. Artif. Intell. 14(2-3), 189–216 (2002)
16. Pensa, R.G., Besson, J., Boulicaut, J.F.: A methodology for biologically relevant pattern discovery from gene expression data. Discovery Science, 230–241 (2004)
17. Messai, N., Devignes, M.D., Napoli, A., Smaïl-Tabbone, M.: Many-valued concept lattices for conceptual clustering and information retrieval. In: 18th biennial European Conference on Artificial Intelligence (2008)
18. Chaudron, L., Maille, N.: Generalized formal concept analysis. In: International Conference on Conceptual Structures (ICCS), pp. 357–370 (2000)
19. Ferré, S., Ridoux, O.: A logical generalization of formal concept analysis. In: International Conference on Conceptual Structures (ICCS), pp. 371–384 (2000)
20. Martin, F.: The genome of laccaria bicolor provides insights into mycorrhizal symbiosis. Nature 452(7183), 88–92 (2008); 69 co-authors wrote this paper
21. Busch, S., Braus, G.H.: How to build a fungal fruit body: from uniform cells to specialized tissue. Molecular Microbiology 64, 873–876 (2007)
22. Lacourt, I., Duplessis, S., Abba, S., Bonfante, P., Martin, F.: Isolation and characterization of differentially expressed genes in the mycelium and fruit body of tuber borchii. Applied and Environmental Microbiology 68, 4574–4582 (2002)
23. Gabella, S., Abba, S., Duplessis, S., Montanini, B., Martin, F., Bonfante, P.: Transcript profiling reveals novel marker genes involved in fruiting body formation in tuber borchii. Eukaryotic Cell 4, 1599–1602 (2005)
24. Hsiao, C.J., Zaki, M.J.: Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Trans. on Knowl. and Data Eng. 17(4), 462–478 (2005)
25. Szathmary, L.: Symbolic Data Mining Methods with the Coron Platform. PhD Thesis in Computer Science, Univ. Henri Poincaré – Nancy 1, France (November 2006)