Clustering gene expression data using graph separators

Bangaly Kaba[1], Nicolas Pinet[1],
Gaëlle Lelandais[2], Alain Sigayret[1], Anne Berry[1].

LIMOS/RR-07-02
13/02/2007 - révisé le 11/05/2007

[1] LIMOS, UMR CNRS 6158, Ensemble des Cézeaux, 63173 Aubière cedex, France. Mail: {kaba; pinet; sigayret; berry} @isima.fr
[2] Equipe de Bioinformatique Génomique et Moléculaire, INSERM U726, Université Paris 7, case 7113, 2 Place Jussieu, 75251 Paris cedex 05, France. Mail: lelandais @ebgm.jussieu.fr

Abstract

Recent work has used graphs to modelize expression data from microarray experiments, in view of partitioning the genes into clusters. In this paper, we introduce the use of a decomposition by clique separators. Our aim is to improve the classical clustering methods in two ways: first we want to allow an overlap between clusters, as this seems biologically sound, and second we want to be guided by the structure of the graph to define the number of clusters. We test this approach with a well-known yeast database (*Saccharomyces cerevisiae*). Our results are good, as the expression profiles of the clusters we find are very coherent. Moreover, we are able to organize into another graph the clusters we find, and order them in a fashion which turns out to respect the chronological order defined by the the sporulation process.

Keywords: clustering method, microarray, graph decomposition, threshold family of graphs, expression profile.

Introduction

Cell functioning involves the existence of complex interaction networks between genes. With the advent of high-throughput biotechnologies in the last decade, deciphering the structure and organization of these networks is one of the most significant challenges in systems biology. In particular, transcriptome analysis based on microarray data allows the identification of genes with correlated expression profiles, thus yielding a better understanding of the functional relationships between them.

Many clustering methods have been proposed and are nowadays widely used. These algorithms group genes into clusters of similar expression profiles, in order to suggest possible participation of clustered genes in a common biological process. However, a well-recognized drawback of commonly used clustering algorithms is the fact that they assign each gene to a single cluster. From a biological point of view, it is known that a gene often participates in several functions and therefore should be included in several clusters. Moreover, in standard clustering methods, all the genes are systematically assigned to the "best cluster", even if they are unclassifiable because of an atypical expression behavior.

To cope with these current limitations, we present in this paper a new clustering approach based on graph decomposition methods.

The graphs we use are defined as follows: expression data originating from microarray experiments can easily be transformed into distance matrixes between genes. A distance matrix can in turn be viewed as an undirected graph when a threshold is chosen as maximal: the vertices of the graph are the genes, and there is an edge uniting two genes if the distance between these two genes is not too great, *i.e.* is at most equal to the chosen threshold. In the rest, we will refer to these graphs as *gene graphs*.

Several recent papers have used such a graph for clustering purposes. For example Shamir et al. ([Sha00], [Sha03]) use minimum cut computations to recursively partition the weighted graph into components. Voy et al. ([Voy06]) explore all the maximal cliques of the graph. A *clique* (also called a *complete subgraph*) is a subgraph which is completely connected (there is no missing edge inside this part of the graph), and represents a group of genes which have many interactions between each other. One of the aims of [Voy06] is to define clusters which are not necessarily disjoint. Seno et al. ([Sen04]) define the notion of *p-quasi complete subgraphs* in order to define a partition of the vertices. Both the latter papers work on finding highly connected parts of the graph, and their good results show the importance of these. However, one of the problems they encounter in dealing with cliques or quasi-cliques is that there may be a great number of these in a graph, so they are both expensive to find and require additional heuristics to ensure a 'good' choice.

Our approach here also uses cliques, but it is quite different, because we use special cliques to decompose the graph. The decomposition we use here is called *clique minimal separator decomposition*. (A *separator* is a set of vertices whose removal will disconnect the graph into several parts.) The decomposition process consists in copying a clique minimal separator into the different parts of the graph it defines. One of the interesting attributes of this decomposition is that the groups of objects we define (which are called *atoms*) are not disjoint, but have an overlap which is a clique (and in fact a clique minimal separator). The idea behind this is that these overlaps represent central information which has to be copied into different parts of the graph in order to preserve the structure. In particular, when working with genes, our theory is that this overlap corresponds to genes which are multifunctional regarding gene expression.

Moreover, these clique minimal separators are uniquely defined, there are few of these (less than the number of vertices), and there are efficient algorithms for finding them. Another interesting feature of this decomposition is that for a given graph it is unique: the resulting groups are the same, independently of the algorithm or execution that is used to compute them.

We demonstrate the efficiency and usefulness of our decomposition method by the analysis of a small (40 genes) and well-described microarray dataset in yeast *Saccharomyces cerevisiae* ([Der98]). This dataset was produced in order to study the transcriptional program that drives the yeast developmental process of sporulation,

in which diploid cells undergo meiosis to produce haploid germ cells. In this paper, we chose this microarray data for several reasons. First, the yeast *S. cerevisiae* is a model organism and a large amount of information can be easily retrieved to analyze the atoms obtained in the context of biological knowledge. Second, it is worth noting that previous studies of this sporulation dataset were performed ([Der98]). The sporulation dataset is therefore well described and annotated. In particular, one of its interesting features is that genes whose expression is modified during the sporulation process fall into 4 consecutive temporal classes that coincide with the major biological processes of sporulation: response to nutritional changes ("Nitrogen" class of genes), premeiotic S phase and recombination ("Early" class), meiotic division ("Middle" class) and spore formation ("Late" class). We will illustrate how our decomposition respects this classification.

We show that our decomposition is promising, for the following reasons: the atoms we define have an excellent coherence regarding their expression profiles, and the profiles are roughly the same as those found by classical clustering methods such as k-means. Moreover, we are able to structure these atoms into a linear graph which respects the classification into sporulation phases described above.

Our paper is organized as follows: the next section (Methods) gives a few useful graph definitions and results; we give details on the algorithms used for decomposing the graph into atoms; we then examine our input data, and explain how we transform it into a family of undirected graphs. We go on to discuss how to choose a set of 'good' thresholds, thus enabling us to propose one particular graph to work on. In Section 'Results and Discussion', we decompose the graph we defined, and examine its atoms for the coherence of their expression profiles; we then structure our atoms into a graph and show a strong linear structure. Finally, we compare our profiles to those found by classical clustering methods.

Methods

A few graph notions

All graphs in this work are undirected and finite. Graphs are denoted G=(V,E), with V the set of vertices and E the set of edges; for $X \subset V$, G(X) denotes the subgraph restricted to X. A graph is said to be *connected* if for any pair {x,y} of vertices, there is a path from x to y. A maximal connected part of the graph is called a *connected component*. A *clique* (or complete subgraph) is a set of vertices that are all pairwise adjacent. The *neighborhood* of a vertex x is $N(x)=\{y \neq x \mid xy \in E\}$. The neighborhood of a set of vertices C is $N(C)=\bigcup_{x \in C} (N(x)-C)$. An *isolated vertex* is a vertex with no neighbor.

A set of vertices S is called a *separator* if G(V−S) is not connected, and a *minimal separator* if there are at least two connected components C_1 and C_2 of G(V−S) such that $N(C_1)=N(C_2)=S$. This means that for any pair {a,b} of vertices, with $a \in C_1$ and $b \in C_2$, a and b belong to different connected components of subgraph G(V−S) (S is said to separate a from b); moreover, no proper subset of S will also separate a from b (minimality of S). Naturally enough, a *clique minimal separator* is a minimal separator which is a clique.

A graph is said to be *chordal* (or *triangulated*) if it contains no chordless cycle of length ≥4; in this case, the graph has less than |V| minimal separators, and they are all cliques. A non-chordal graph G=(V,E) can be embedded into a chordal graph H=(V,E+F) by adding a set F of 'fill edges'. H is called a minimal triangulation of G if no subgraph H' of H of the form H'=(V,E + (F−{f})) is chordal ([RTL76]).

The reader is referred to [Gol04] for classical graph definitions, and to [Ber06] for more detailed results on minimal separators.

The *decomposition by clique minimal separators* is obtained by repeatedly choosing a clique minimal separator S, defining the connected components (C_i) of graph G(V−S), and decomposing G into subgraphs $C_i \cup N(C_i)$ ($N(C_i)$ is a clique minimal separator of the graph). When the original graph is thus recursively completely decomposed, the non-separable subgraphs obtained are called *atoms* ([Tar85], [Ber01]). An alternate definition of an atom is that it is a maximal connected subgraph with no clique separator.

Unless the graph is chordal, it does not necessarily have any clique separator, and if it does, it always has less than |V| clique minimal separators. When using clique minimal separators to decompose the graph, the decomposition into atoms which is obtained is unique (see [Lei93], [Ber01]), which is not necessarily the case if clique separators which are not necessarily minimal are used, as in [Tar85].

Algorithmic aspects

The clique minimal separator decomposition of a graph G is usually done by first embedding the graph into a minimal triangulation H (this is because any clique minimal separator S of G is preserved as a minimal separator in H). As a second step, a perfect elimination ordering (peo) of H is computed. A peo is an ordering of the vertices of a chordal graph which is obtained by repeatedly choosing a simplicial vertex x (a vertex whose neighborhood is a clique), giving x the next number from 1 to |V|, and removing x from the graph. It is known that a peo will encounter all minimal separators as transitory neighborhoods ([Ros70]), so the clique minimal separators of G are all detected; the atoms can be computed in a greedy fashion by computing the connected components defined by

each clique minimal separator of G when it is encountered, as defined in [Tar85].

The worst-case complexity analysis for clique minimal separator decomposition is in O(|V||E|) time. In practice, as we will see later in this section, we will tend to choose a low threshold, where the graph is not very dense, so that |E| may be of order |V|. We have tested our decomposition on a larger database (500 genes) and it runs reasonably fast. For our implementations, we used Algorithm MCSM ([Ber04]), which runs fast, is easy to implement, and yields both the minimal triangulation and a corresponding peo directly.

From microarray data to graphs

In a first step, the initial microarray data are stored in a matrix, with rows corresponding to individual genes and columns corresponding to the consecutive intervals during the sporulation program at which gene expression levels were measured after transfer of diploid cells to a nitrogen-deficient medium that induced sporulation. The experiments are done on 7 time points (0, 30 min., 2 hrs., 5 hrs., 7 hrs., 9 hrs., 11 hrs). The reader is referred to [Der98] for more detail. The gene expression vectors are normalized to have 0 as average value and 1 as variance; a distance matrix is then computed, using the Euclidean Distance: the distance between two genes x and y over p experiments will be $d(x,y)=(\Sigma^{p}_{i=1} [\mu(x,i)-\mu(y,i)]^2)^{1/2}$ where μ(x,i) and μ(y,i) represent the measures of genes x and y for experiment i.

Note that in this case, it can be shown that this distance is directly related to the classical Pearson correlation. After normalizing the expression profiles to have a mean value of 0 and a variance of 1, it can be shown that the Euclidean distance (dE) is related to the Pearson correlation (r): dE2 = 2.n.(1 – r), n being the number of point measurements. For more detailed information concerning distance computation between expression profiles the reader can refer to [Ste03].

Once this distance matrix is defined, for each value t of this matrix (each threshold), a corresponding graph $G_t=(V,E_t)$ is defined: V is the set of genes, which are represented as vertices, and there is an edge between vertices x and y in G_t if the distance between x and y in the distance matrix is at most equal to the value t of the threshold, *i.e.* $d(x,y)\leq t$.

The corresponding data (gene expression matrix, distance matrix, binary matrix for threshold 1.25, adjacency list of graph for threshold 1.25) can be downloaded at the following URL: https://www.isima.fr/~kaba/clustering.htm.

Defining a range of good thresholds

In order to define our working graph, we need to choose a threshold in this matrix. We will examine several criteria which will help choose a good threshold.

Let us first take a preliminary look at the family of graphs defined. Figure 1 gives the graphs obtained for a variety of thresholds, to illustrate how the graph obtained varies when the threshold increases.

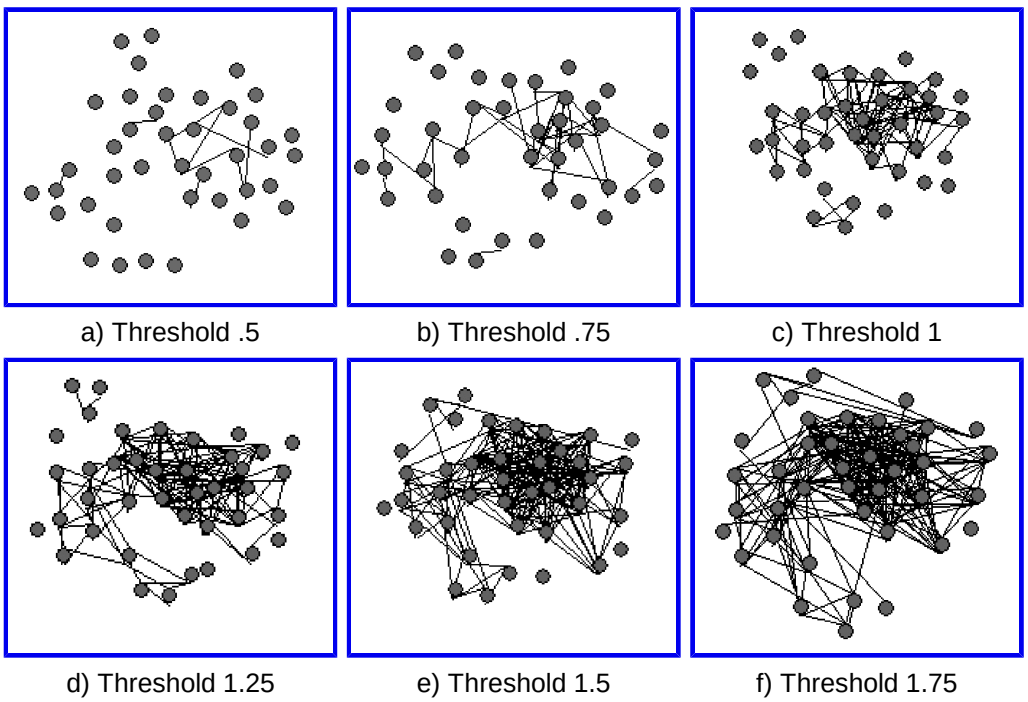
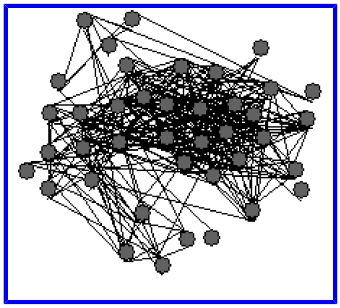
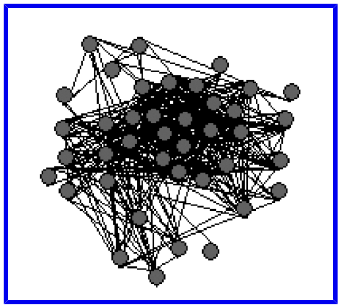


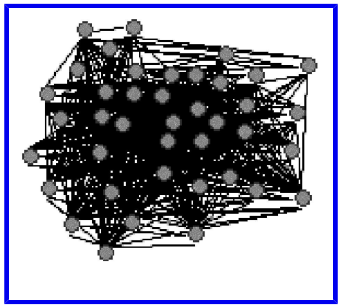
Figure 1: *Gene graphs for different thresholds*.



g) Threshold 2



h) Threshold 2.3



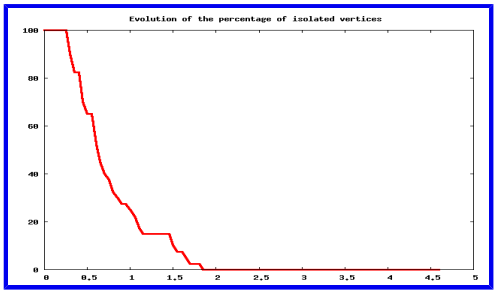
i) Threshold 3.5

If the threshold is very low (as for threshold .5), the graph has almost no edges, and moreover has many isolated vertices. Isolated vertices will each form a trivial atom, so will not yield much information.

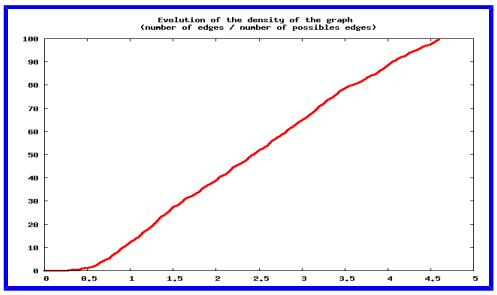
As the threshold value increases, edges are added to the graph, until it is connected (*i.e.* made up of only one component, as in the graph obtained for threshold 2).

When the threshold is high, as for the graph obtained for threshold 3.5, there are many edges.

Naturally enough, we should aim at choosing a graph which is connected or almost connected, because isolated vertices will fail to fall automatically into a group of genes, but with not too many edges, because there will be so much connectivity that the groups will become too large and will not be very informative. We should remark that on this database, most of the graphs which are not connected have one large connected component and some isolated vertices. Figure 2 gives the percentage of isolated vertices and the density of the graph (number of edges divided by maximum possible number of edges, which is 780), according to the threshold.



1. Percentage of isolated vertices



2. Density

Figure 2: *Percentage of isolated vertices and density, with regard to the threshold chosen for decomposition*.

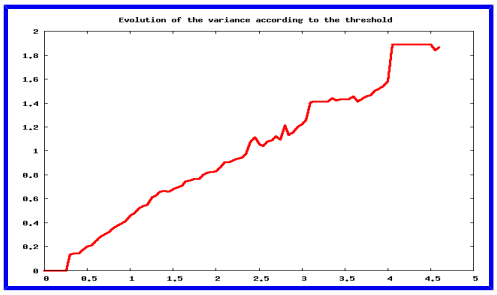
The graph becomes connected at threshold 1.85. The density function is roughly linear. We tentatively propose as a working hypothesis to allow no more than 50% of isolated vertices, and a density between 7% and 35%.

Let us now examine how we can ensure a good quality. We will use variance arguments to determine, for a number of thresholds, the quality of each atom, and then compute the average quality for that threshold.

As we have seen, in the low thresholds, the graph has isolated vertices. Each isolated vertex is an atom in its own right (we call it a *trivial atom*), but it yields no information as to the groupings of genes. We thus removed all the isolated vertices from the graph at each threshold before computing the variance.
Given p experiments on a set of genes resulting into measure μ, given a threshold t that induces a decomposition into k non-trivial atoms, the average measure of an atom X for experiment i will be:

$A(x,i)=\Sigma_{X\in X}\,[\mu(x,i)\,/\,|X|]$
and the variance related to atom X will be:
$var(X)=(\,\Sigma^{p}_{i=1}\,\Sigma_{X\in X}\,\,[(\mu(x,i)-A(X,i))^{2}\,/\,|X|]\,)^{1/2}$.
Finally, we express the global variance for a chosen threshold t as:
$var(t)=\Sigma^{k}_{j=1}\,[var(X_j)\,/\,k]$.

Figure 3 presents the evolution of the variance as the threshold increases. The variance function is roughly linear, so the smaller the threshold is, the more confident we will be that the decomposition into atoms is good. We propose to set the maximum acceptable variance at 1.



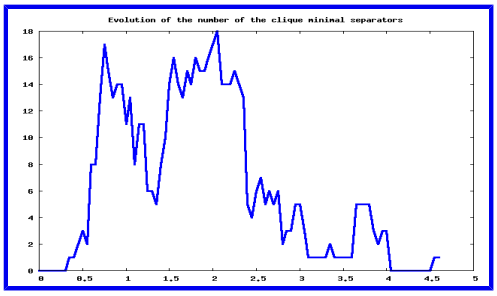
Variance

Figure 3: *Evolution of the variance according to the threshold.*.

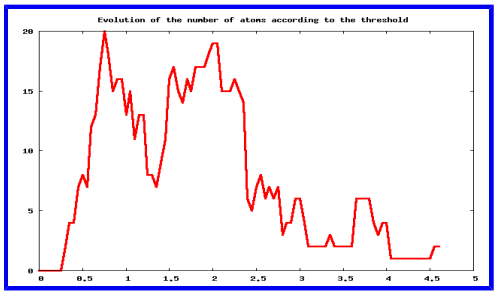
We also want to examine structural aspects which are related to the graphs defined.

To begin with, we need to choose a graph with clique minimal separators. In a given arbitrarily chosen graph, there may not be any clique minimal separator at all, and in this case there is only one atom defined, which is the entire original graph, so no information will be extracted. Our decomposition technique will give results only when the graph has clique minimal separators, and there need to be enough of these, so that enough different atoms will be defined.

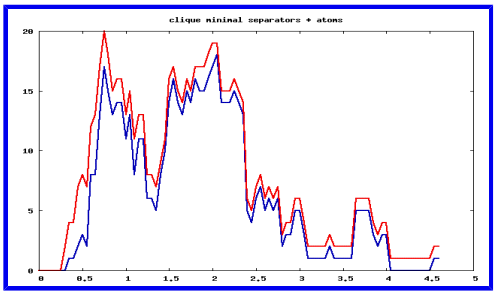
In Figure 4, Graphic 1 gives the evolution of the number of clique minimal separators.



1. Number of clique minimal separators



2. Number of non-trivial atoms



3. Number of clique minimal separators and number of non-trivial atoms

Figure 4: *Number of clique minimal separators (blue), number of atoms (red), and superimposition of both, with regard to the threshold chosen for decomposition*.

There are several remarks on this:

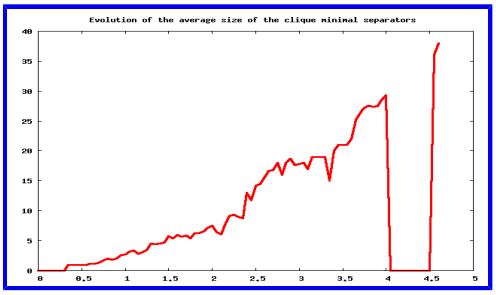
- We see that in the family of graphs defined by this data set, there are almost always some clique minimal separators (except in the threshold interval between 4 and 4.5). This is worth noting, as in general, many graphs do not have any clique minimal separators at all.
- The number of clique minimal separators varies somewhat erratically.

Another concern is the number of atoms obtained: if there are many atoms, each atom tends to be small, and we obtain many groups of two or three genes, which is not very informative.

In Figure 4, Graphic 2 gives the variation of the number of atoms with regard to the threshold. As we see from Graphic 3 of the same figure, the number of atoms is almost in direct correlation with the number of clique minimal separators. This indicates that most of the graphs of this family have separators which each define only two components, *i.e.* their removal will disconnect the graph into 2 parts. In an arbitrary graph, one clique minimal separator may define many connected components and thus there may be considerably more atoms than there are clique minimal separators.

Another concern will be that the size of the clique minimal separators not be too great. As explained in Section 1, there is an overlap between our atoms, and each overlap is a clique minimal separator. When the clique minimal separators become too large, the overlap between the groups is also large, so it will be difficult to distinguish the atoms one from the other, and we will not be able to discriminate between the groups, which will be pairwise too similar.

Figure 5 gives the variation on the average sizes of the clique minimal separators according to the threshold. We see that the size of the clique minimal separators increases almost linearly with the threshold.



Average size of clique minimal separators

Figure 5: *Size of clique minimal separators with regard to the threshold chosen for decomposition*.

The following table summarizes our criteria for choosing a threshold. We partitioned the thresholds into 4 significant intervals:

Threshold interval	[0;.6]	[.6;1.8]	[1.8;2.5]	[2.5;4.6]
Percentage of isolated vertices	100% - 52.5% (bad)	52.5% - 2.5% (good)	2.5% - 0% (good)	0% (good)
Density	0% - 2.3% (bad)	2.3% - 34.1% (good)	34.1% - 51.9% (good)	>51.9% (bad)
Variance	0 - 0.25 (good)	0.25 - 0.77 (good)	0.77 - 1.05 (good)	1.05 - 1.88 (bad)
Average overlap size	0 - 1.125 genes (good)	1.125 - 6.25 genes (good)	6.25 - 14 genes (bad)	>14 or 0 genes (bad)

We see from this table that only the second column is acceptable for every criterion. In the corresponding threshold interval [.6;1.8], the number of atoms is too high except in a very clear sub-interval which corresponds to a local minimum where the number of atoms is less than 10: [1.25;1.4]. We choose 1.25 as having the best variance.

In conclusion, we will study extensively the graph corresponding to threshold 1.25. In the rest of this paper, we will use this graph and proceed to analyze it.

Results and Discussion

We will now examine the graph defined by threshold 1.25, chosen as discussed above. This graph is presented in Figure 7, with the vertices arbitrarily numbered 1 through 40. Figure 6 lists the corresponding genes, and gives the functional class each one belongs to. For better understanding, we will recall the name and class of the vertices in further discussions.

Gene	Class	Vertex number	Gene	Class	Vertex number

YAL062W	Nitrogen	1	YHR139C	Late	21
YBL084C	Middle	2	YHR152W	Middle	22
YBR088C	Early	3	YHR157W	Early	23
YCR002C	Middle	4	YHR166C	Middle	24
YDL155W	Middle	5	YIR028W	Nitrogen	25
YDR402C	Late	6	YIR029W	Nitrogen	26
YDR403W	Late	7	YJL146W	Late	27
YEL061C	Middle	8	YJR137C	Nitrogen	28
YER095W	Early	9	YJR152W	Nitrogen	29
YER096W	Late	10	YKL022C	Middle	30
YFL003C	Early	11	YKR034W	Nitrogen	31
YFR028C	Middle	12	YLR263W	Early	32
YFR030W	Nitrogen	13	YLR329W	Early	33
YFR036W	Middle	14	YLR438W	Nitrogen	34
YGL116W	Middle	15	YOL091W	Middle	35
YGL163C	Early	16	YPL111W	Nitrogen	36
YGL180W	Late	17	YPL121C	Early	37
YGR108W	Middle	18	YPL122C	Early	38
YGR109C	Middle	19	YPL178W	Middle	39
YHL022C	Early	20	YPR120C	Middle	40

Figure 6: *The genes, their sporulation class and their vertex number*.

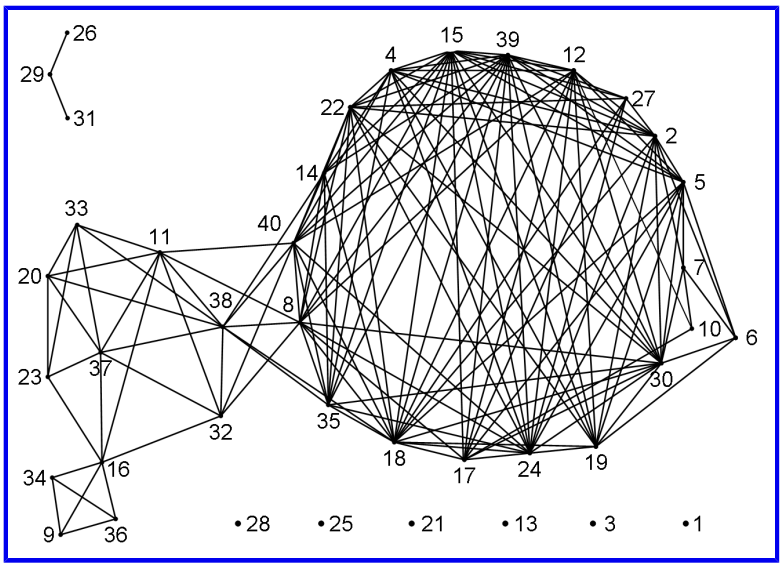


Figure 7: *The gene graph at threshold 1.25*.

Our graph has 8 connected components:

- 6 isolated vertices 1, 3, 13, 21, 25, 28 (i.e. YAL062W[N], YBR088C[E], YFR030W[N], YHR139C[L], YIR028W[N], YJR137C[N]), each forming a trivial atom.
- 1 small component {26,29,31} (YIR029W[N], YJR152W[N], YKR034W[N]), which forms a graph with one clique minimal separator (vertex 29), which defines two atoms each containing 2 genes:
  - Atom B_1 contains vertices 26 and 29 (YIR029W[N], YJR152W[N]),
  - Atom B_2 contains vertices 29 and 31 (YJR152W[N], YKR034W[N]).

- The remaining component is a large one, and contains all the other vertices of the graph.

In the rest, we will concentrate our investigations on this larger component, which has the following 5 clique minimal separators:

- S_1={16}, 1 gene: YGL163C[E];
- S_2={11,32,38}, 3 genes: YFL003C[E], YLR263W[E], YPL122C[E];
- S_3={8,38,40}, 3 genes: YEL061C[M] , YPL122C[E], YPR120C[M];
- S_4={8,14, 18,35,40}, 5 genes: YEL061C[M], YFR036W[M], YGR108W[M], YOL091W[M], YPR120C[M];
- S_5={4,8,15,18,22,35,39,40}, 8 genes: YCR002C[M], YEL061C[M], YGL116W[M], YGR108W[M], YHR152W[M], YOL091W[M], YPL178W[M], YPR120C[M].

These separators decompose the larger component into 6 atoms:

- Atom A_1={9,16,34,36}, 4 genes: YER095W[E], YGL163C[E], YLR438W[N], YPL111W[N];
- Atom A_2={11,16,20,23,32,33,37,38}, 8 genes: YFL003C[E], YGL163C[E], YHL022C[E], YHR157W[E], YLR263W[E], YLR329W[E], YPL121C[E], YPL122C[E];
- Atom A_3={8,11,32,38,40}, 5 genes: YEL061C[M], YFL003C[E], YLR263W[E], YPL122C[E], YPR120C[M];
- Atom A_4={8,14,18,35,38,40}, 6 genes: YEL061C[M], YFR036W[M], YGR108W[M], YOL091W[M], YPL122C[E], YPR120C[M];
- Atom A_5={4,8,14,15,18,22,35,39,40}, 9 genes: YCR002C[M], YEL061C[M], YFR036W[M], YGL116W[M], YGR108W[M], YHR152W[M], YOL091W[M], YPL178W[M], YPR120C[M];
- Atom A_6={2,4,5,6,7,8,10,12,15,17,18,19,22,24,27,30,35,39,40}, 19 genes: YBL084C[M], YCR002C[M], YDL155W[M], YDR402C[L], YDR403W[L], YEL061C[M], YER096W[L], YFR028C[M], YGL116W[M], YGL180W[L], YGR108W[M], YGR109C[M], YHR152W[M], YHR166C[M], YJL146W[L], YKL022C[M], YOL091W[M], YPL178W[M], YPR120C[M].

Figure 8 gives the graph at threshold 1.25 with its decomposition into atoms.

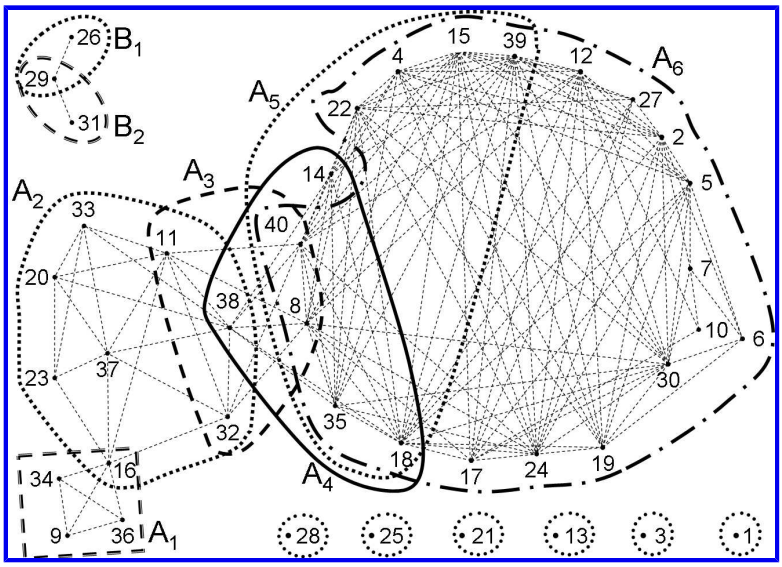


Figure 8: *The gene graph of Figure 7 with its atoms*.

Biological expertise of the atoms

In order to explain the biological relevance of the atoms obtained after graph decomposition, we will start this paragraph with a brief reminder of the developmental program of sporulation in yeast Saccharomyces cerevisiae. We will then go on to discuss the coherence of our atoms, based on biological knowledge and functional annotations coming from Gene Ontology ([AB&00]).

By definition, “sporulation” is the cellular process in which diploid cells undergo meiosis to produce haploid germ cells. It can be assimilated to a “gametogenesis” in yeast. More precisely, sporulation involves two overlapping processes, meiosis (cell division) and spores morphogenesis, and results in four haploid spores. Each spore is finally capable of germinating and merging with a cell of the opposite mating type, analogous to the fusion of egg and sperm. Combining major cellular processes (cell division and spore formation), the sporulation

required the expression of an important number of genes (more than 500) involved in functions as diverse as chromosome pairing and recombination, DNA processing, cell cycle control, spore formation or cell wall maturation. The proper functioning of the entire process needs to precisely coordinate in time the different events required for sporulation. In that respect, it has been shown that, in yeast S. cerevisiae, the sporulation is characterized by sequential changes in expression of at least four sets of genes hereafter referred to as Nitrogen (N), Early (E), Middle (M) and Late (L) ([Chu98]). These temporal classes coincide with the major biological processes of sporulation. Immediately after cell transfer to sporulation medium, the expression of the “Nitrogen” class of genes is activated. It corresponds to genes that have metabolic functions, related to adaptation of nitrogen starvation. Then, expression of the “Early” genes is induced. These genes are involved in meiotic prophase (the first step in meiosis) that consists in pairing of homologous chromosomes and recombination. In a third step, products of the “Middle” genes are required for the concomitant events of meiotic nuclear division and spore formation. Finally, the “Late” genes are involved in the formation of the spore wall, as well as spore maturation.

In this paper, our aim is to test our approach using real biological data. In particular, we want to see if our graph decomposition allows the identification of relevant classes of genes (atoms). From a biological point of view, “relevant atoms” have to be in agreement with the temporal organization “Nitrogen”, “Early”, “Middle” and “Late”. As a result, we can observe that atoms A_2, A_5 are highly significant since only one class of genes is included in each atom (A_2 = “E”; A_5 = “M”). The other atoms contain two different classes, but it is worth noting that these classes are temporally consecutive: A_1 = “N + E ”; A_3 = “E + M” ; A_4 = “E + M” ; A_5 = “M + L”. On the other hand, all the clique minimal separators except one contain exactly one kind of gene, so that we see how they serve as a useful connection between two different groups.

To go further into this investigation, we also analyse our results using functional annotations coming from Gene Ontology (GO). GO is a structural network consisting of defined terms and relationships between them that describe three attributes of gene products: molecular function, biological process and cellular components ([AB&00]). Once again, we obtain promising results. As an illustration, 6/9 genes belonging to the atom A5 are annotated as “M Phase” (GO:0000279), meaning that they are all involved in the progression through M phase, the part of the cell cycle comprising nuclear division and cytokinesis. Moreover, this GO category was found to be statistically significantly over-represented with respect to what would be expected by chance (the calculated p-value is 2.02e-05).

Expression profiles of the atoms

We will now analyze our results by plotting the expression profiles of the groups of genes defined by our atoms. An expression profile of a group of genes is a graphics that represents the evolution of each gene expression of this group with regard to the experiences (7 experiences here, corresponding to the 7 time points), plus as reference, the average expression of the group.

Let us first remark on what we will consider to be a coherent or non-coherent group of genes regarding their expression profiles: if the expressions of all (or almost all) the genes of a group have the `same' variations as their reference expression, we will say that the group is coherent; this may be expressed by a variance value. Figure 9 gives examples for both a coherent group and a non-coherent group.

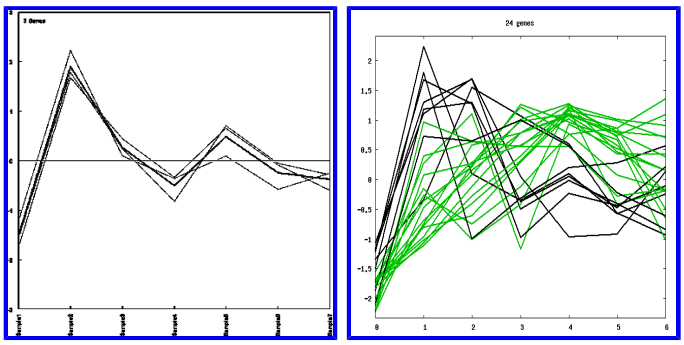


Figure 9: *Example of a coherent profile (left) and a non-coherent profile (right)*.

Figure 10 gives the 6 expression profiles for the 6 atoms of the large component; the profiles are remarkably coherent.

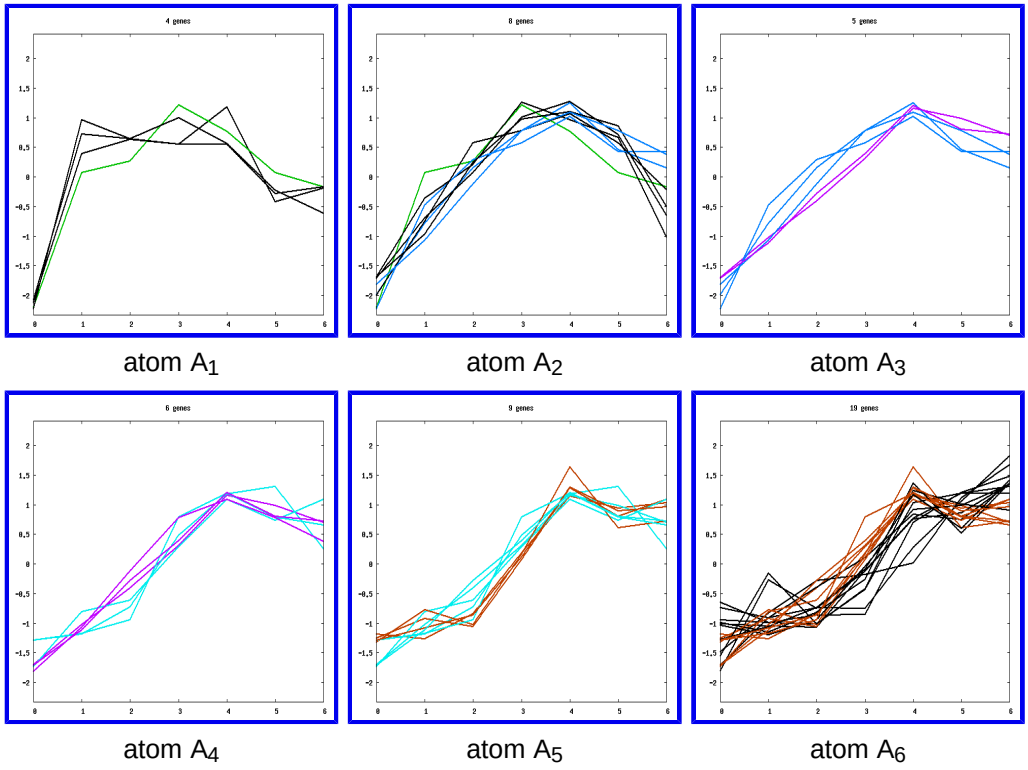


Figure 10: *The expression profiles of the 6 atoms of the large component*.

For information, Figure 11 shows the expression profiles of the 6 isolated vertices of our original graph for threshold 1.25 (*i.e.* vertices 1, 3, 13, 21, 25 and 28), as well as the expression profiles for the 2 atoms of the smaller component (atoms B_1 and B_2), which are similar and could be merged, defining this small component as a single 'atom'.

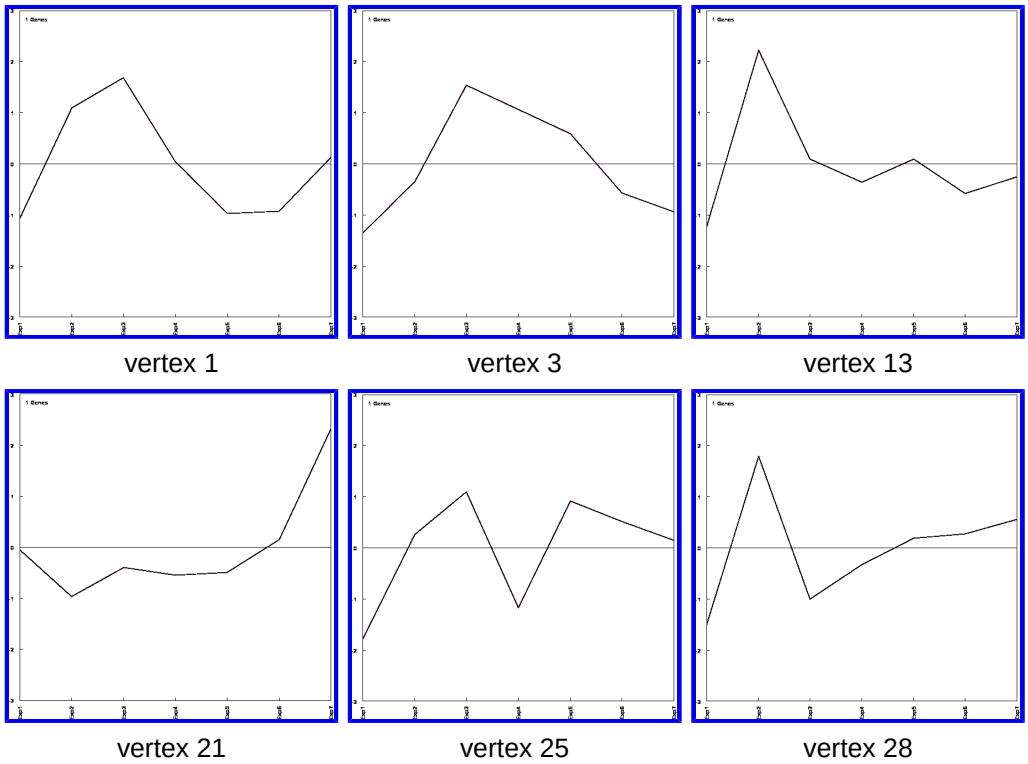
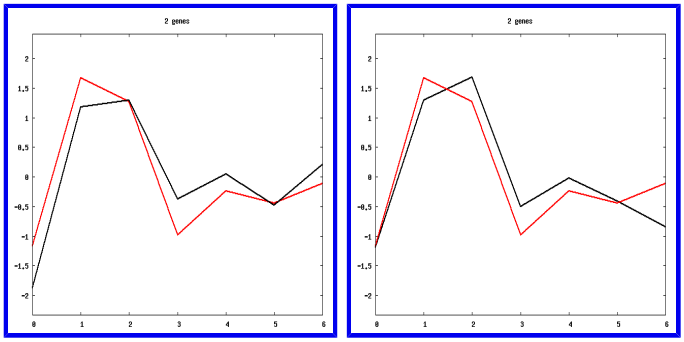


Figure 11: *The expression profiles of the 6 isolated vertices of the graph, and the profiles of the 2 atoms of the smaller component*.



atom B_1 atom B_2

Structuring the atoms into a graph

As discussed in the introduction, one of the strong points of our approach is that the groups we obtain are partially overlapping. This enables us to define an intersection graph, which we call the '*atom graph*': the vertices of that secondary graph are the atoms, and we will draw an edge between two atoms if their intersection is a clique minimal separator. Figure 12 gives this atom graph for the large component, where each atom is represented by its average expression profile, and each edge is labelled with the corresponding minimal separator.

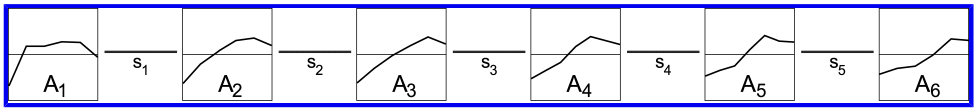


Figure 12: *Atom graph of the large component. Each atom is represented by its average expression profile, and each edge is labelled with the corresponding minimal separator*.

Note the way the profiles vary according to a path in the graph. This graph is very linear in structure; in fact, it belongs to the class of interval graphs, which are the intersection graphs of intervals on a line. This is due to the fact that in this graph, each clique minimal separator defines only two connected components, whereas in the general case there may be many; this corresponds to a particular structure of the input data. Because of this structure, the graph search algorithm we use for our implementation (MCSM) yields the atoms directly in the correct order.

Figure 13 gives an interval representation of our atom graph.

- Each interval (which represents an atom) is labeled with one or two elements from {N,E,M,L}, according to the sporulation phase (“Nitrogen”, “Early”, “Middle” and/or “Late”) that the genes it contains belong to, as discussed above.
- The intersections of atoms, which as we have seen are clique minimal separators, are classically represented by 'scanlines' (vertical lines), also labeled with N,E,M,L.

The chronological succession N,E,M,L appears very clearly, as discussed in the previous subsection. Even more noteworthy is the way the clique minimal separators (scanlines) behave:

- All of them except one (S_2, which has 2 E and 1 M) include only one sporulation class.
- Each one neatly divides 2 atoms which contain 2 different sporulation classes.

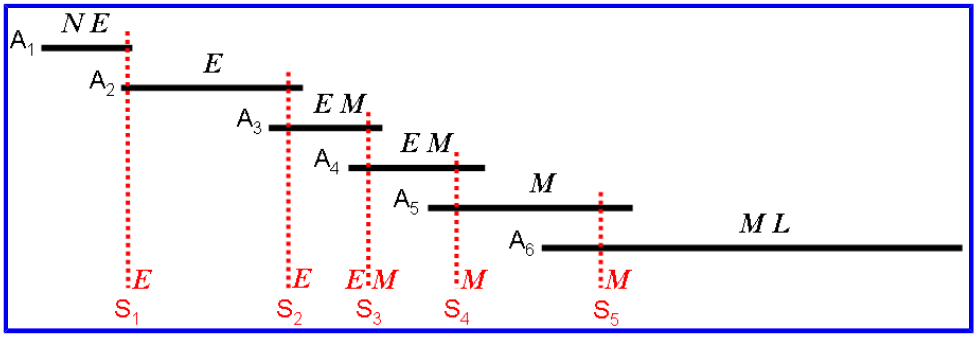


Figure 13: *Interval representation of the atom graph of Figure 12*.

We have tested several other thresholds for this property, and have found that all the atom graphs explored are indeed interval graphs.

Comparing with classical clustering method k-means

We have compared our results with that of k-means. As we have 8 non-trivial atoms in the graph, we put the same number of clusters for the k-means process. The k-means execution we ran results in the following clusters:

- Cluster C_1={11,32,38}, 3 genes: YFL003C[E], YLR263W[E], YPL122C[E];
- Cluster C_2={8,14,35,40}, 4 genes: YEL061C[M], YFR036W[M], YOL091W[M], YPR120C[M];
- Cluster C_3={9,16,34,36}, 4 genes: YER095W[E], YGL163C[E], YLR438W[N], YPL111W[N];
- Cluster C_4={3,20,23,33,37}, 5 genes: YBR088C[E], YHL022C[E], YHR157W[E], YLR329W[E], YPL121C[E];
- Cluster C_5={1,26,29,31}, 4 genes: YAL062W[N], YIR029W[N], YJR152W[N], YKR034W[N];
- Cluster C_6={13,25,28}, 3 genes: YFR030W[N], YIR028W[N], YJR137C[N];
- Cluster C_7={6,7,10,21}, 4 genes: YDR402C[L], YDR403W[L], YER096W[L], YHR139C[L];
- Cluster C_8={2,4,5,12,15,17,18,19,22,24,27,30,39}, 13 genes: YBL084C[M], YCR002C[M], YDL155W[M], YFR028C[M], YGL116W[M], YGL180W[L], YGR108W[M], YGR109C[M], YHR152W[M], YHR166C[M], YJL146W[L], YKL022C[M], YPL178W[M].

Figure 14 gives the expression profiles of these clusters. Note how most of these cluster profiles are similar to some atom profile as given in Figure 10.

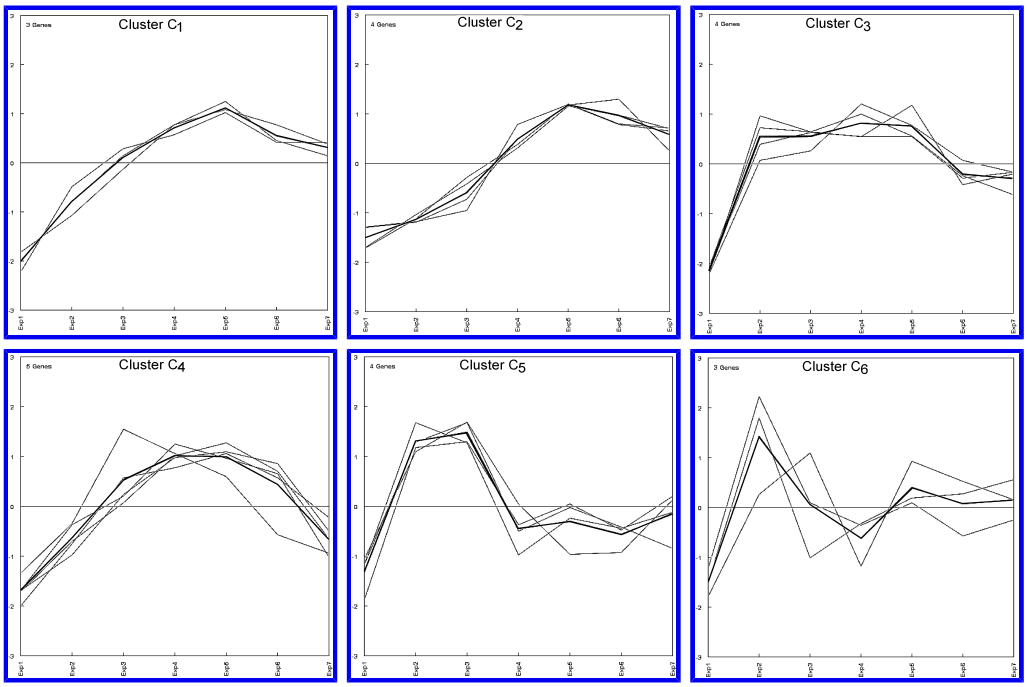
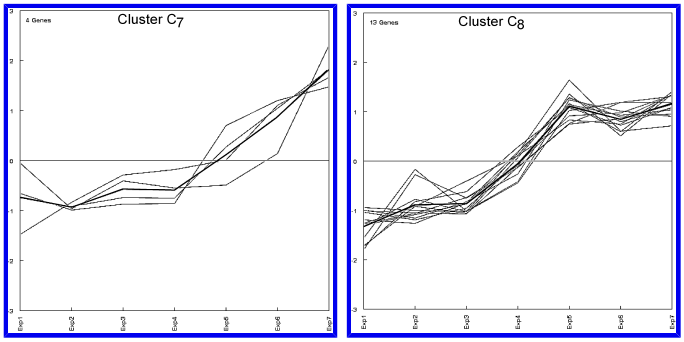


Figure 14: *The 8 expression profiles found by k-means*.



Moreover, these clusters fit with our atoms in a very coherent fashion, as illustrated by Figure 15, where the atoms are circled as in Figure 8, and the vertices are painted according to the k-means cluster they belong to. Note that Cluster C_5 is made out of the small component plus one isolated vertex, and C_6 contains only isolated vertices. The other clusters correspond very nearly to the large component. However, 2 clusters have an isolated vertex: cluster C_4 has isolated vertex 3 and cluster C_7 has isolated vertex 21. From Figure 11 showing the expression profiles of the isolated vertices (which are the trivial atoms), one can see in Figure 14 that, for both of these clusters C_4 and C_7, the one gene which has a deviating profile is the gene corresponding to the isolated vertex, which shows how it may not always be wise for clustering methods to force these genes into a cluster.

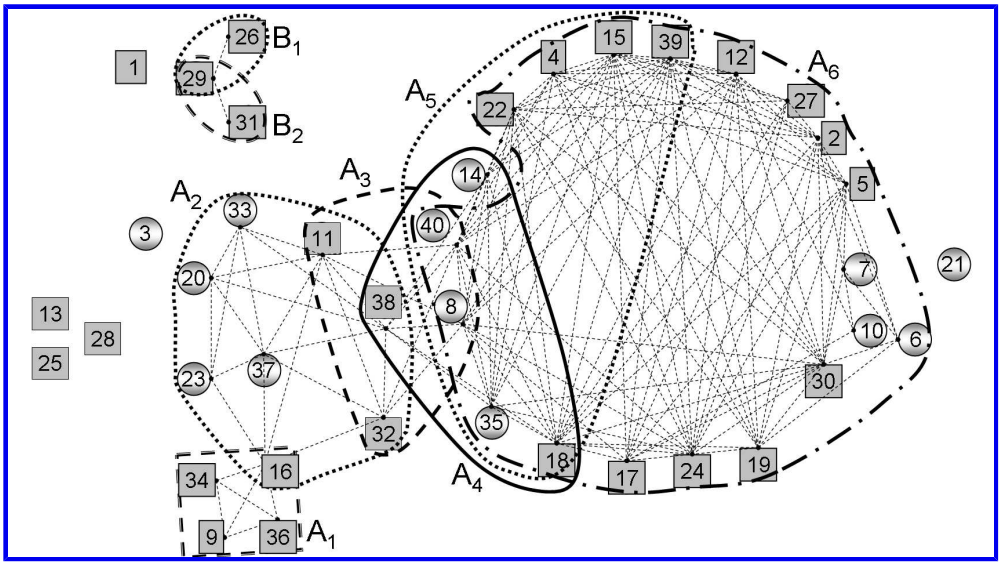


Figure 15: *Graph of Figure 8 with its atoms circled as in Figure 8, and the vertices colored according to the k-means cluster they belong to: alternately white-and-gray circles and gray rectangles. In order to simplify the figure, isolated vertices have been represented next to the cluster they belong to.*

One of our main contributions in this paper is that using the ordering from our atom graph (given in Figure 12), we can define an ordering of the clusters, illustrated by Figure 16. This ordering respects the ordering defined by the sporulation phases, whereas k-means alone does not offer it. Moreover, different executions of K-means do not necessarily yield the same clusters, whereas, once a threshold is chosen, our decomposition into atoms is unique.

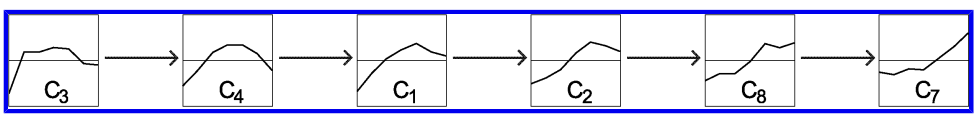


Figure 16: *The cluster graph, showing an ordering of the k-means clusters.*

Conclusion

As a conclusion, we see that not only do we obtain groups of genes which each have inherently coherent expression profiles, but when we compare with clustering methods such as k-means, the corresponding

expression profiles obtained are much the same as ours.

Our method has several advantages over k-means:

- A given gene can belong to only one k-means cluster, but it can belong to more than one atom, while preserving expression profile coherence. This goes to show that the classical classification methods were perhaps deficient in this respect: it is possibly important to express the fact that a gene can belong to several expression groups.
- Classification methods such as k-means require the user to determine in advance the number of classes desired, a task not always easy to perform. With our method, the decomposition is uniquely defined by the structure of the graph (which represents the structure of the data), so that the genes fall naturally into a number of atoms.
- Our method does not require that, for a given threshold, all the genes be classified: we can concentrate on large groups defined by non-trivial or not too small connected components of our graph, and not feel an obligation to classify the genes forming isolated vertices. Methods such as k-means will automatically integrate these isolated vertices into a 'best' class, but may prove disruptive to the global coherence of the class. Our method can deal with these isolated genes independently.
- Our method gives an ordering both of the atoms and of the k-means clusters.

Work in progress includes applying these results to a larger database (500 genes), in order to both further validate the approach proposed in this paper, and suggest new properties of the database.

References

[AB&00] Ashburner M., Ball C. A., Blake J. A., Botstein D., Butler H., Cherry J. M., Davis A. P., Dolinski K., Dwight S. S., Eppig J. T., Harris M.A., Hill D.P., Issel-Tarver L., Kasarskis A., Lewis S., Matese J.C., Richardson J.E., Ringwald M., Rubin G.M., Sherlock G. (2000): *Gene ontology: tool for the unification of biology*. The Gene Ontology Consortium, Nat. Genet. 25(1), pp 25-9.

[Ber01] Berry A. and Bordat J.-P. (2001): *Decomposition by clique minimal separators*, Communication Dagstuhl Seminar No. 01251 Report No. 312, http://www.isima.fr/berry/decomp.ps.

[Ber04] Berry A., Blair J., Heggernes P., Peyton B. (2004): *Maximum Cardinity Search for Computing Triangulations of Graphs*, Algorithmica, 39-4, pp 287-298.

[Ber06] Berry A., Bordat J.-P., Heggernes P., Simonet G. and Villanger Y. (2006): *A wide-range algorithm for minimal triangulation from an arbitrary ordering*. Journal of Algorithms 58.1, pp 33-66.

[Chu98] Chu S., DeRisi J., Eisen M., Mulholland J., Botstein D., Brown PO., Herskowitz I. (1998): *The transcriptional program of sporulation in budding yeast*, Science 23 Oct 1998, 282(5389), pp 699-705.

[Der98] DeRisi J., Chu S. and al. (1998): *The transcriptional program of sporulation in budding yeast* Science 20 Nov 1998, 282(5393):1421.

[Eis98] Eisen M. B, Spellman P. T., Brown P.O. and Botstein D. (1998): *Cluster analysis and display of genome wide expression patterns*, PNAS 95:14863-14868.

[Gol04] Golumbic M. C. (2004): *Algorithmic Graph Theory and Perfect Graphs*, Annals of Discrete Mathematics 57, Elsevier, 2nd edition (2004), Academic Press, New York.

[Lei93] Leimer H.-G. (1993): *Optimal Decomposition by Clique separators*, Discrete Mathematics archive, 113(1-3), pp 99-123.

[Ros70] Rose D. J. (1970): *Triangulated graphs and the elimination process*, J. Math. Anal. Appl., 32(597), pp 597-609.

[RTL76] Rose D. J., Tarjan R. E., Lueker G. S. (1976): *Algorithmic aspects of vertex elimination on graphs*, Siam J. Comput., 5, pp 266-283.

[Sen04] Seno S., Teramoto R., Takenaka Y. and Matsuda H. (2004): *A Method for Clustering Gene Expression Data Based on Graph Structure*, Genome Informatics 2004, 15(2), pp 151-160.

[Sha00] Sharan R. and Shamir R. (2000): *CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis*, Proc. ISMB'00, AAAI Press, Menlo Park (CA, USA), pp 307-316.

[Sha03] Sharan R., Maron-Katz A. and Shamir R. (2003): *CLICK and EXPANDER: a system for clustering and visualizing gene expression data*, Vol. 19 no. 14, DOI: 10.1093/bioinformatics/btg232, 28 Mar. 2003, pp

1787-1799.

[Ste03] Stekel D. (2003): *Microarray Bioinformatics*, Cambridge University Press, UK, Chapter 8, pp 139-182.

[Tar85] Tarjan R. E. (1985): *Decomposition by clique separators*, Discrete Mathematics, 55, pp 221-232.

[Voy06] Voy B. H., Scharff J. A., Perkins A. D., Saxton A.M., Borate B., Chesler E. J., Branstetter L. K. and Langston M. A. (2006): *Extracting Gene Networks for Low-Dose Radiation Using Graph Theoretical Algorithms*, PLOS Computational Biology.