An Integrative Clustering Approach Combining Particle Swarm Optimization and Formal Concept Analysis

Anna Hristoskova[1], Veselka Boeva[2], and Elena Tsiporkova[3]

[1] Department of Information Technology,
Ghent University - IBBT, 9050 Ghent, Belgium
anna.hristoskova@intec.UGent.be
[2] Department of Computer Systems and Technology
Technical University of Sofia-branch Plovdiv 4000 Plovdiv, Bulgaria
vboeva@tu-plovdiv.bg
[3] ICT & Software Engineering Group, Sirris, 1030 Brussels, Belgium
elena.tsiporkova@sirris.be

Abstract. In this article we propose an integrative clustering approach for analysis of gene expression data across multiple experiments, based on Particle Swarm Optimization (PSO) and Formal Concept Analysis (FCA). In the proposed algorithm, the available microarray experiments are initially divided into groups of related datasets with respect to a predefined criterion. Subsequently, a hybrid clustering algorithm, based on PSO and k-means clustering, is applied to each group of experiments separately. This produces a list of different clustering solutions, one per each group. These clustering solutions are pooled together and further analyzed by employing FCA which allows to extract valuable insights from the data and generate a gene partition over the whole set of experiments. The performance of the proposed clustering algorithm is evaluated on time series expression data obtained from a study examining the global cell-cycle control of gene expression in fission yeast *Schizosaccharomyces pombe*. The obtained experimental results demonstrate that the proposed integrative algorithm allows to generate a unique and robust gene partition over several different microarray datasets.

Keywords: data clustering, k-means, particle swarm optimization, formal concept analysis, integration analysis, gene expression data

1 Introduction

DNA microarray technology offers the ability to screen the expression levels of thousands of genes in parallel under different experimental conditions or their evolution in discrete time points. All these measurements contain information on many different aspects of gene regulation and function, ranging from understanding the global cell-cycle control of microorganisms [20], to cancer in humans [1, 10]. Gene clustering is one of the most frequently used analysis methods for gene

expression data. Clustering algorithms are used to divide genes into groups according to the degree of their expression similarity. Such a grouping may suggest that the respective genes are correlated and/or co-regulated, and moreover that the genes could possibly share a common biological role.

The combination of data from multiple microarray studies addressing a similar biological question is gaining high importance in the recent years [6, 7, 24] due to the ever increasing number and complexity of the available gene expression datasets. In general, it is expected that the integration and evaluation of multiple datasets yields more reliable and robust results since these results are based on a larger number of samples and the effects of individual study-specific biases are diminished. In [5], we proposed a hybrid algorithm combining k-means and Particle Swarm Optimization (PSO) clustering algorithms in order to derive a gene clustering solution from a set of independent, but biologically related, microarray datasets. It was demonstrated that this hybrid algorithm produces good quality clustering solution, which is representative for the whole experimental compendium and at the same time adequately reflects the specific characteristics of the individual experiments.

In this work, we propose an integrative clustering method that combines PSO and Formal Concept Analysis (FCA) [9] in order to cluster datasets generated in multiple-experiment settings. In contrast to the hybrid clustering algorithm introduced in [5], where PSO-based clustering is applied to the entire set of experiments in order to produce the final clustering solution, the algorithm proposed in this paper initially divides the available microarray experiments into groups of related (similar) datasets with respect to a predefined criterion. The rationale behind this is that if experiments are closely related to one another, then these experiments may produce more accurate and robust clustering solution. Thus PSO-based clustering is applied to each group of experiments separately. This produces a list of different clustering solutions, one per each group. Next these solutions are pooled together and further analyzed by employing FCA which allows to extract valuable insights from the data and further generate a gene partition over the whole experimental compendium. FCA produces a concept lattice where each concept represents a subset of genes that belong to a number of clusters. The concepts compose the final disjoint clustering partition.

A detailed overview of several PSO-based clustering approaches is presented in [5]. The FCA or *concept lattice approach* has been applied for extracting local patterns from microarray data [2, 3] or for performing microarray data comparison [8, 18]. For example, the FCA method proposed in [8] builds a concept lattice from the experimental data together with additional biological information. Each vertex of the lattice corresponds to a subset of genes that are grouped together according to their expression values and some biological information related to the gene function. It is assumed that the lattice structure of the gene sets might reflect biological relationships in the dataset. In [13], a FCA-based method is proposed for extracting groups or classes of co-expressed genes. A concept lattice is constructed where each concept represents a set of co-expressed genes in a number of situations. A serious drawback of the method is the fact that the

expression matrix is transformed into a binary table (the input for the FCA step) which may lead to possible introduction of biases or substantial information loss.

The remainder of this paper is structured as follows: Section 2 briefly describes the basic principles of k-means, PSO, hybrid clustering and FCA, and subsequently introduces our integrative clustering approach. The dataset and the applied experimental setup are outlined in Section 3, followed by analysis and discussion of the clustering results in Section 4. Finally, the main conclusions are drawn in Section 5.

2 Clustering Methods

2.1 K-means Clustering Algorithm

The *k-means algorithm* [15] is one of the most widely used techniques for clustering. It starts by initializing the k cluster centers, where k is preliminarily determined. Subsequently, each object (input vector) of the dataset is assigned to the cluster whose center is the nearest. The mean (centroid) of each cluster is then computed so as to update the cluster center. This update occurs as a result of the change in the membership of each cluster. The processes of re-assigning the objects and the update of the cluster centers is repeated until there is no more change in the value of any of the cluster centers.

2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is an evolutionary computation method introduced in [14]. In order to find an optimal or near-optimal solution to the problem, PSO updates the current generation of particles (each particle is a candidate solution to the problem) using the information on the best solution obtained by each particle and the entire population. Each particle is treated as a point in an n-dimensional space. The i-th particle is initialized with random positions $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and velocities $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ at time point $t = 0$. The performance of each particle is measured according to a predefined fitness function, which uses the particle's positional coordinates as input values. Positions and velocities are adjusted, and the function is evaluated with the new coordinates at each time-step. The basic update equations for the d-th dimension of the i-th particle in PSO may be given as

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot \varphi_1 \cdot (p_{id} - x_{id}(t)) + c_2 \cdot \varphi_2 \cdot (p_{gd} - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1). \quad (2)$$

The variables φ_1 and φ_2 are uniformly generated random numbers in the range $[0, 1]$, c_1 and c_2 are called acceleration constants whereas w is called inertia weight [21]. $P_g = (p_{g1}, p_{g2}, \ldots, p_{gn})$ is the best particle position found so far

within the population and $P_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ is the best position discovered so far by the corresponding particle. The first part of equation (1) represents the *inertia* of the previous velocity, the second part is the *cognition part* and it tells us about the personal experience of the particle, the third part represents the cooperation among particles and is therefore named the *social component*. Acceleration numbers c_1, c_2 and inertia weight w are predefined by the user. It was shown in [21] that when w is in the range $[0.9, 1.2]$, PSO will have the best chance to find the global optimum within a reasonable number of iterations. Furthermore, $w = 0.72$ and $c_1 = c_2 = 1.49$ were found in [17] to ensure good convergence.

Notice that in the multi-experimental context considered in Section 2.3 the *cognition part* representing the personal opinion of the particle is based on its own source of information (dataset), while in the classical one dataset setup the *cognition part* is derived from a common (for all particles) information source. This may also have a reflection on the *social part*, since information contained in different sources may have different representations and may need to be preprocessed before the collaboration of the particles.

2.3 Hybrid PSO-based Approach for Clustering Data Compendiums

We have proposed in [5] a hybrid algorithm combining k-means and PSO for deriving a clustering result from multiple microarray datasets. This algorithm will be used to produce a clustering result from a group of related microarray experiments in Section 2.5. The main idea of the algorithm and its consecutive steps are presented below.

Let us consider a group of n different microarray datasets $M_1, M_2, \ldots, M_n$. Each dataset is supposed to contain the gene expression levels of m genes in n_i different experimental conditions or time points. In this context, each matrix i can be used to generate k cluster centers, which are considered to represent a particle, *i.e.* the particle is treated as a set of points in an n_i-dimensional space. The final (optimal) clustering solution will be found by updating the particles using the information on the best clustering solution obtained by each data matrix and the entire set of matrices.

Assume that the i-th particle is initialized with a set of k cluster centers[4] $C_i = \{C_1^i, C_2^i, \ldots, C_k^i\}$ and a set of velocity vectors $V_i = \{V_1^i, V_2^i, \ldots, V_k^i\}$[5] using gene expression matrix M_i. Thus each cluster center is a vector $C_j^i = (c_{j1}^i, c_{j2}^i, \ldots, c_{jn_i}^i)$ and each velocity vector is a vector $V_j^i = (v_{j1}^i, v_{j2}^i, \ldots, v_{jn_i}^i)$, *i.e.* each particle i is a matrix (or a set of points) in the $k \times n_i$ dimensional space.

Next, assume that $P_g = \{P_{g1}, P_{g2}, \ldots, P_{gk}\}$ is a set of cluster centers in an n_g-dimensional space representing the best clustering solution found so far within the set of matrices and $P_i = \{P_1^i, P_2^i, \ldots, P_k^i\}$ is the set of centroids of

[4] The number of clusters k, is initially identified by analyzing the quality of the obtained clustering solutions generated on the involved datasets for a range of different numbers of clusters.

[5] The velocity vectors are initialized by zeros.

the best solution discovered so far by the corresponding matrix. The update equation for the d-th dimension of the j-the velocity vector of the i-th particle is defined as follows

$$v_{jd}^{i}(t+1) = w \cdot v_{jd}^{i}(t) + c_1 \cdot \varphi_1 \cdot (p_{jd}^{i} - c_{jd}^{i}(t)) + c_2 \cdot \varphi_2 \cdot g(t), \tag{3}$$

where $i = 1, \ldots, n$; $j = 1, \ldots, k$; $d = 1, \ldots, n_i$ and

$$g(t) = \begin{cases} p_{gd} - c_{jd}^{i}(t), & \text{if } n_g \geq n_i \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

Note that the *cognition part* in the above equation has a modified interpretation. Namely, it represents the private 'thinking' (opinion) of the particle based on its own source of information (dataset). Due to this the *social part* (see equation (4)) differs from that in equation (1), since each particle matrix has a different number of columns (n_i) due to different number of experiment points in each dataset.

The clustering algorithm combining PSO and k-means can be summarized as follows:

1. Initialize each particle with k cluster centers obtained as a result of applying the k-means algorithm to the corresponding data matrix.
2. Initialize the personal best clustering solution of each matrix with the corresponding clustering solution found in Step 1.
3. **for** iteration = 1 **to** max-iteration **do**
   (a) **for** $i = 1$ **to** n **do** (i.e. for all datasets)
      i. **for** $j = 1$ **to** m **do** (i.e. for all genes in the current dataset)
         A. Calculate the distance of gene g_j with all cluster centers.
         B. Assign g_j to the cluster that has the nearest center to g_j.
      ii. **end for**
      iii. Calculate the fitness function for the clustering solution C_i.
      iv. Update the personal best clustering solution P_i.
   (b) **end for**
   (c) Find the global best solution P_g.
   (d) Update the cluster centers according to the velocity updating formula proposed in equation (3).
4. **end for**

2.4 Formal Concept Analysis

Formal concept analysis [9] is a mathematical formalism allowing to derive a concept lattice from a formal context constituted of a set of objects O, a set of attributes A, and a binary relation defined as the Cartesian product $O \times A$. The context is described as a table, the rows correspond to objects and the columns to attributes or properties and a cross in a table cell means that "an object possesses a property". Formal Concept Analysis (FCA) can be used for a number of purposes among which knowledge formalization and acquisition, ontology design, and data mining.

The *concept lattice* is composed of formal concepts, or simply concepts, organized into a hierarchy by a partial ordering (a subsumption relation allowing to compare concepts). Intuitively, a concept is a pair *(X, Y)* where $X \subseteq O$, $Y \subseteq A$, and X is the maximal set of objects sharing the whole set of attributes in Y and vice-versa. The set X is called the *extent* and the set Y the *intent* of the concept *(X, Y)*. The subsumption (or subconcept - superconcept) relation between concepts is defined as follows:

$$(X_1, Y_1) \prec (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 \text{ (or } Y_2 \subseteq Y_1). \tag{5}$$

Relying on this subsumption relation $\prec$, the set of all concepts extracted from a context is organized within a complete lattice, that means that for any set of concepts there is a smallest superconcept and a largest subconcept, called the *concept lattice*.

2.5 Integrative Clustering Approach Combining PSO and FCA

We propose herein an integrative clustering method that combines PSO and FCA in order to cluster datasets generated in multiple-experiment settings. It consists of two distinctive steps: *PSO-based clustering* and *FCA-based analysis*. Initially, the available microarray experiments are divided into groups of related (similar) datasets with respect to a predefined criterion. Subsequently, the hybrid (k-means and PSO) clustering algorithm as described in Section 2.3 [5] is applied to each group of experiments. This produces a list of different clustering solutions, one for each group. Next these solutions are pooled together and further analyzed by employing FCA which generates a single clustering solution for the whole data compendium of multiple experiments. FCA produces a concept lattice where each concept represents a subset of genes that belong to a number of clusters. The concepts compose the final disjoint clustering partition.

A detailed explanation of the distinctive phases of the proposed algorithm combining PSO and FCA is given below.

Initialization Phase Assume that a particular biological phenomenon is monitored in several high-throughput experiments under a few different conditions. In this way, a list of different data matrices will be produced, one per experiment. Suppose that N different genes are in total monitored by all the different experimental datasets.

Initially, the available gene expression matrices are divided into r groups of related (similar) datasets with respect to some predefined criterion, *e.g.* the used synchronized method, or the expression similarity between the matrices. Then the number of cluster centers is identified for each group separately. As discussed in [11, 22], this can be performed by running the selected clustering algorithm on each dataset for a range of different numbers of clusters. Subsequently, the quality of the obtained clustering solutions needs to be assessed in some way in order to identify the clustering scheme which best fits the datasets in question. For example, some of the internal validation measures that are presented in Section 3.2

(Silhouette Index or Connectivity) can be used as validity indices to identify the best clustering scheme. Suppose that k_i cluster centers are determined for each group i $(i = 1, 2, \ldots, r)$.

PSO-grouped Clustering The hybrid clustering algorithm explained in Section 2.3 is applied to each group of related experiments i $(i = 1, 2, \ldots, r)$ separately. The latter will generate a list of r different clustering solutions, one per each group, *i.e.* a set of k_i different clusters will be produced for each group i $(i = 1, 2, \ldots, r)$. Suppose that K $(K = k_1 + \ldots + k_r)$ different clusters in total are produced by all the different groups.

FCA Analysis As discussed above, the N studied genes are grouped by the PSO-grouped clustering algorithm into K clusters. As mentioned in Section 2.4, FCA is a principled way of automatically deriving a hierarchical conceptual structure from a collection of objects and their properties. The approach takes as input a matrix (referred as the formal context) specifying a set of objects and the properties thereof, called attributes. In our case, a (formal) **context** consists of the set G of the N studied genes, the set of clusters $C = C_1, C_2, ..., C_K$ produced by the clustering step, and an indication of which genes belong to which clusters. Thus the context is described as a matrix, with the genes corresponding to the rows and the clusters corresponding to the columns of the matrix, and a value 1 in cell *(i, j)* whenever gene *i* belongs to cluster C_j. Subsequently, a (formal) **concept** for this context is defined to be a pair *(X, Y)* such that

- $X \subseteq G$ & $Y \subseteq C$ & every gene in X belongs to every cluster in Y
- for every gene in G that is not in X, there is a cluster in Y that does not contain that gene
- for every cluster in C that is not in Y, there is a gene in X that does not belong to that cluster

The family of these concepts obeys the mathematical axioms defining a **concept lattice**. The constructed lattice consists of concepts where each one represents a subset of genes, all belonging to the same subset of clusters. The set of all concepts partitions the genes into a set of disjoint clusters.

On extremely large datasets the proposed integrative clustering method is expected to be computationally intensive. However, the computational cost can be drastically reduced by first performing some advanced filtering or features selection in order to remove noisy data and preserve lower number of potentially relevant genes for clustering.

3 Experimental Setup

3.1 Microarray Datasets

The proposed clustering algorithm has been validated on benchmark datasets where the true clustering is known. These datasets have been composed by gene

expression time series data obtained from a study examining the global cell-cycle control of gene expression in fission yeast *Schizosaccharomyces pombe* [20]. The study includes eight independent time-course experiments synchronized respectively by:

1. elutriation: three independent biological repeats (*elu1*, *elu2*, *elu3*);
2. cdc25 block-release: two independent biological repeats, of which one in two dye-swapped technical replicates (*cdc25-1*, *cdc25-2.1*, *cdc25-2.2*) and in addition, one experiment in a sep1 mutant background (*cdc25-sep1*);
3. a combination of both methods: elutriation and cdc25 block-release (*elu-cdc10*) as well as elutriation and cdc10 block-release (*elu-cdc25*).

Thus, nine different expression test sets are available. In the preprocessing phase the rows with more than 25% missing entries have been filtered out from each expression matrix and any other missing expression entries have been imputed by the DTWimpute algorithm [23]. In this way nine complete matrices have been obtained.

Rustici *et al.* identified 407 genes as cell-cycle regulated [20]. These have been subjected to clustering which resulted in the formation of 4 separate clusters. Subsequently, the time expression profiles of these genes have been extracted from the complete data matrices and thus nine new matrices have been constructed. Note that some of these 407 genes were removed from the original matrices during the preprocessing phase, *i.e.* each dataset may have a different set of genes. Next, the nine datasets have been divided into three groups with respect to the used synchronization method. The overlapping genes within each group are as follows: a subset of 286 common genes in the elutriation datasets, a subset of 350 common genes in the cdc25 block-release datasets and a subset of 364 common genes in the datasets synchronized by the combination of both methods. Subsequently, the genes that are not presented in the intersection of the datasets of each group have been removed. As a result of this nine new matrices which form our benchmark datasets have been constructed. Notice that the nine different dataset contain 374 different genes in total.

The test datasets have been normalized by applying a data transformation method aiming at multi-purpose data standardization and inspired by gene-centric clustering approaches as proposed in [4].

3.2 Cluster Validation Measures

One of the most important issues in cluster analysis is the validation of the clustering results. Essentially, the cluster validation techniques are designed to find the partitioning that best fits the underlying data, and should therefore be regarded as a key tool in the interpretation of the clustering results. Since none of the clustering algorithms performs uniformly best under all scenarios, it is not reliable to use a single cluster validation measure, but instead to use at least two that reflect different aspects of a partitioning. In this sense, we have implemented two different validation measures for estimating the quality of the clusters:

1. Connectivity: for assessing connectedness;
2. Silhouette Index (SI): for assessing compactness and separation properties of a partitioning.

Connectivity Connectivity captures the degree to which genes are connected within a cluster by keeping track of whether the neighboring genes are put into the same cluster [12]. Let us define $m_{i(j)}$ as the jth nearest neighbor of gene i, and let $\chi_{im_{i(j)}}$ be zero if i and j are in the same cluster and $1/j$ otherwise. Then for a particular clustering solution $C_1, C_2, \ldots, C_k$ of matrix M, which contains the expression values of m genes (rows) in n different experimental conditions or time points (columns), the connectivity is defined as

$$Conn(c) = \sum_{i=1}^{m} \sum_{j=1}^{n} \chi_{im_{i(j)}}.$$

The connectivity has a value between *zero* and *infinity* and should be *minimized*.

Silhouette Index Silhouette index reflects the compactness and separation of clusters [19]. Suppose $C_1, C_2, \ldots, C_k$ is a clustering solution (partition) of matrix M, which contains the expression profiles of m genes. Then the SI is defined as

$$s(k) = \frac{1}{m} \sum_{i=1}^{m} (b_i - a_i) / \max\{a_i, b_i\},$$

where a_i represents the average distance of gene i to the other genes of the cluster to which the gene is assigned, and b_i represents the minimum of the average distances of gene i to genes of the other clusters.

The values of Silhouette Index vary from *-1* to *1* and *higher value* indicates better clustering results.

4 Validation Results and Discussion

In this section, the performance of the proposed integrative clustering method on the benchmark datasets is presented. The standard k-means, the hybrid (combination of k-means and PSO) clustering approach from Section 2.3 and the proposed integrative (combination of PSO and FCA) clustering algorithms are executed in order to generate clustering solutions on each of the considered nine microarray matrices. The quality of these solutions is evaluated using two cluster validation measures: Silhouette Index (SI) and Connectivity. These cluster validation measures have been implemented in C++. The PSO-based clustering algorithm has been implemented in Java. The publicly available open source machine learning software WEKA[6] is used by this implementation for the particle initialization and for the gene assignment to the different clusters.

[6] http://www.cs.waikato.ac.nz/ml/weka/

Initialization Phase Initially, the nine test datasets are divided into three groups with respect to the used synchronized method:

1. elutriation datasets: *elu1*, *elu2*, *elu3*;
2. cdc25 block-release datasets: *cdc25-1*, *cdc25-2.1*, *cdc25-2.2*, *cdc25-sep1*;
3. datasets synchronized by the combination of both methods: *elu-cdc10*, *elu-cdc25*.

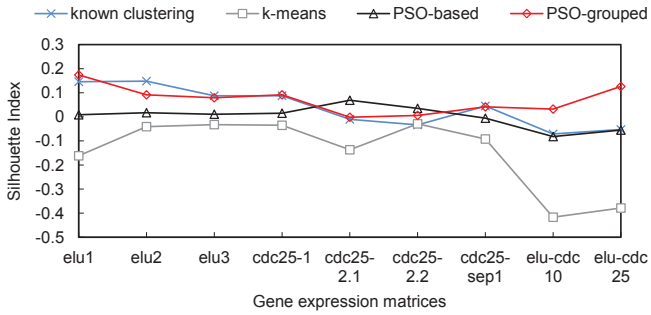
Then the number of cluster centers is identified for each group. As discussed in [11], [22], this can be performed by running the selected clustering algorithm on each dataset for a range of different numbers of clusters. Thus the k-means clustering algorithm is executed for values of k between 2 and 10 on each dataset. Subsequently, the quality of the obtained clustering solutions is assessed by using the Connectivity and SI as validity indices. We search for the values of k for which a significant local change in value of the index occurs [11]. The selected optimal number of clusters for the three groups of experiments is as follows: elutriation datasets: $k = 4$; cdc25 block-release datasets: $k = 6$, and the combined ones: $k = 5$.

PSO-based Clustering Next the PSO-based hybrid clustering algorithm (see Section 2.3) is executed on each group of experiments separately. It is run for 500 iterations with $w = 0.72$ and $c_1 = c_2 = 1.49$. These values have been chosen to ensure good convergence [17]. Notice that 15 different clusters (elutriation: clusters 0-3, cdc25 block-release: clusters 4-9 and combination of both: clusters 10-14) in total are produced by the three groups.

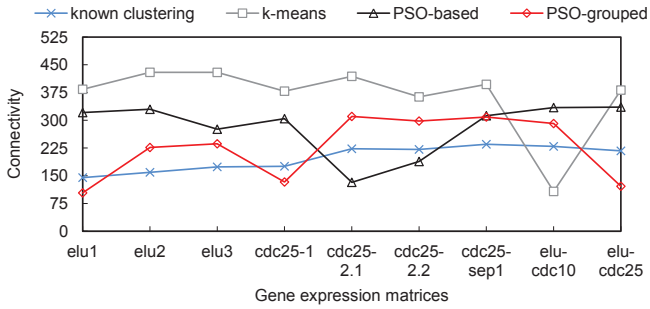
Figure 1 compares the SI and Connectivity values produced by the standard k-means, the known clustering solution published in [20], the hybrid PSO-based clustering algorithm considered in Section 2.3 and the PSO-grouped version of the latter algorithm described in Section 2.5 on the individual matrices. Note that the SI and Connectivity values for the PSO-based algorithm are obtained by using the global best solution found among the 9 different experimental matrices, those for the PSO-grouped algorithm are generated by using the global best solutions found separately for each group of datasets (elutriation, cdc25 block-release and combined), while the values for the k-means are produced by using the clustering solutions generated for each of the corresponding individual datasets.

As it can be seen in Figure 1, the SI values produced by the PSO-based and PSO-grouped algorithms outperform for all the nine experiments the values obtained for the k-means algorithm. Similar superior performance of the PSO-based and PSO-grouped algorithms in comparison to the k-means can be observed for the Connectivity scores with the single exception of *elu-cdc10*. The k-means result produced on the latter experiment appears to be an outlier of the Connectivity scores obtained for the rest of the experiments. There is no any obvious explanation of this phenomenon. It may be due to the experiment specific characteristics.

According to the SI indices, the PSO-grouped algorithm clearly outperforms the known clustering solution. However, the Connectivity index provides less



(a) Silhouette Index



(b) Connectivity Index

Fig. 1. Comparison of the SI (a) and Connectivity (b) values generated by the known clustering solution published in [20], and those obtained by applying the standard k-means, the PSO-based hybrid algorithm and the PSO-grouped version of the latter algorithm on the 9 different experiments. The SI and Connectivity values for the PSO-based algorithm are obtained by using the global best solution found among the 9 different experimental matrices, those for the PSO-grouped algorithm are generated by using the global best solutions found separately for each group of datasets (elutriation, cdc25 block-release and combined), while the values for the k-means are produced by using the clustering solutions generated for each of the corresponding individual datasets.

conclusive results. In general, the PSO-grouped clustering solution is better than the known one in 70% of the experiments under the SI validation index and respectively, in 35% of the test datasets under the Connectivity index. The PSO-grouped version also exhibits better performance than the PSO-based clustering algorithm in 80% of the experiments under both validation measures.

FCA Analysis The gene partitions produced by the clustering step are further analyzed by applying FCA using publicly available tool [7]. We have created a context that consists of the set of 374 studied genes and the set of 15 clusters produced by the clustering step. It is described as a binary matrix, with the genes corresponding to the rows and the clusters corresponding to the columns. Subsequently, a lattice of 109 concepts for this context is generated (see Figure 2). Thus the FCA step partitions the benchmark gene set in 83 disjoint clusters (concepts) in total since the rest of the concepts appear to be empty. However, a number of 27 concepts are singleton sets and only the following seven concepts have cardinality above 10: {1, 6, 10}, {0, 5, 13}, {1, 6, 12}, {2, 4, 10}, {0, 5, 10}, {1, 8, 12}, {2, 6, 10}. It is interesting to notice that all the concepts connecting three clusters (46 such concepts exist) are not empty sets and in addition, they all contain clusters produced by each of the three groups of experiments.

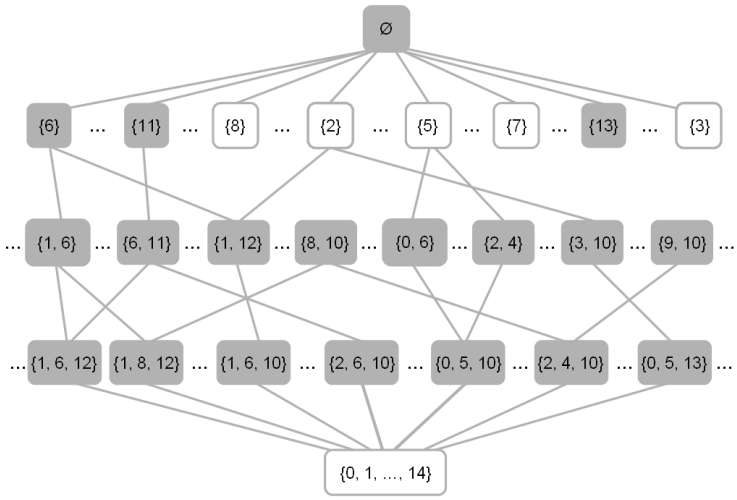


Fig. 2. Part of the generated concept lattice visualizing the seven concepts discussed above and their subconcepts and superconcepts.

Each of the above listed seven concepts was subjected to analysis with the BiNGO tool [16], in order to determine which Gene Ontology categories are statistically overrepresented in each concept. The results are generated for a cutoff

[7] http://www.iro.umontreal.ca/~galicia/features.html

p-value of 0.05 and Benjamini and Hochberg (False Discovery Rate) multiple testing correction. For each gene concept a table is generated consisting of five columns: (1) the GO category identification (GO-id); (2) the multiple testing corrected p-value (p-value); (3) the total number of genes annotated to that GO term divided by total number of genes in the test set (cluster frequency); (4) the number of selected genes versus the total GO number (total frequency); and (5) a detailed description of the selected GO categories (description).

Only 5 of the seven FCA concepts (see above) have been assigned GO categories by the BiNGO tool: {0, 5, 13}, {1, 6, 12}, {2, 4, 10}, {1, 8, 12}, and {2, 6, 10}. Concretely:

- concept {0, 5, 13} contains 23 genes annotated to 4 GO categories (all have cluster frequency 78.2%), which point out to (cellular) response to stress and stimulas;
- concept {1, 6, 12} contains 18 genes connected with 5 GO categories (only 3 have total frequency > 0.0%), all reffering to the regulation of sister chromatid cohesion and segregation;
- concept {2, 4, 10} contains 14 genes associated with about 100 GO categories (25% of these have total frequency = 0.0%), majority of which refer to regulation of different biological processes including cell-cycle;
- concept {1, 8, 12} contains 12 genes annotated to 22 GO categories (16 have total frequency > 0.0%) dominated by RNA metabolic processing related categories;
- concept {2, 6, 10} contains 11 genes connected with 19 GO categories (10 have total frequency > 0.0%), most of which refer to cell-cycle control or regulation of DNA replication.

# Genes	24	14	15	6	7	4	1	1	1	1	1	1	1
Premise	0 13	14	3	2 13	3 14	2 14	0 7	2 5	2 8	2 11	3 6	3 7	9,11
	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
Conclusion	5	4	4	6	4	4	11	10	10	6	12	12	1
# Genes	23	12	12	6	6	4	1	1	1	1	1	1	1
% Genes	96	86	80	100	86	100	100	100	100	100	100	100	100

Table 1. Cluster implications. The top row presents the number of genes contained in the premise clusters (second row). The clusters of the conclusion are listed in the third row and their corresponding number of genes in the forth row. The percentage of premise genes contained in the conclusion is given in the last row.

Beside the generated concepts and the lattice diagram one can examine the implications between attributes (in our case the different gene clusters) valid in a context. Table 1 presents the specific dependencies extracted by the ConExp tool [8] between the clusters of the three groups of experiments. The *premise*

[8] http://conexp.sourceforge.net/

defines a gene lattice and the *conclusion* specifies the dependent lattice that holds for a high percentage of the genes in the premise. This implication describes that if a certain gene is present in the *premise* clusters, it is also found (with some exceptions) in the cluster from the *conclusion*. For instance, 23 out of the 24 genes present in clusters 0 (elutriation dataset) and 13 (combination of elutriation and cdc25 block-release dataset) are present in cluster 5 (cdc25 block-release dataset). Using these implications the genes occurring in the same clusters can be replaced by one representative gene. In addition, the genes that can be obtained as a result of the intersection of some other genes can be removed by ConExp reducing the 374 studied genes to a selection of 50. For example, if we consider the fourth column of Table 1, all the six genes belonging to clusters 2, 6, 13 are replaced by one representative gene. Further gene number 2, belonging to clusters 1, 10, can be obtained as a result of the intersection of genes 3 and 20 respectively, belonging to clusters 1, 6, 10 and 1, 7, 10 and therefore, we can remove gene 2. The so described reduction operation does not change the structure of the constructed lattice as the reduced concept lattice is isomorphic to the original one.

5 Conclusion

We have proposed an integrative clustering method which combines Particle Swarm Optimization and Formal Concept Analysis for deriving a clustering solution for multiple gene expression matrices. The performance of the proposed clustering algorithm has been evaluated on a test set of 9 time series expression datasets obtained from a study examining the global cell-cycle control of gene expression in fission yeast *Schizosaccharomyces pombe*. The presented in this article experimental results demonstrate that the proposed clustering algorithm is a robust data integration technique, which is able to produce good quality clustering solution that is representative for the whole test set. In addition, the employment of the FCA allows to perform a subsequent data analysis, which provides useful insights about the biological role of genes contained in the same FCA concepts. Our future work will focus on further exhaustive analysis of the composition and relationships between the different FCA concepts.

References

1. A. Alizadeh et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.
2. J. Besson, C. Robardet, J-F. Boulicaut. Constraint-Based Mining of Formal Concepts in Transactional Data. *PAKDD*, 615–624, 2004.
3. J. Besson, C. Robardet, J-F. Boulicaut and S. Rome. Constraint-based concept mining and its application to microarray data analysis. *Intell. Data Anal.*, 9(1):59–82, 2005.
4. V. Boeva and E. Tsiporkova. A multi-purpose time series data standardization method. *Intelligent Systems: From Theory to Practice, Springer-Verlag Berlin Heidelberg*, 2010.

5. V. Boeva, A. Hristoskova and E. Tsiporkova. Clustering of Multiple DNA Microarrays through Combination of Particle Swarm Intelligence and K-means. In *Proceedings of the 6th International Conference on Computational Intelligence and Bioinformatics, Pittsburgh, USA*, pages 32–38, 2011.
6. A. Brazma W.R. Gilks, B.D.M. Tom. Fusing microarray experiments with multivariate regression. *Bioinformatics*, 21(2):ii137–ii143, 2005.
7. J.K. Choi et al. Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, 19:i84–i90, 2003.
8. V. Choi, Y. Huang, V. Lam, D. Potter, R. Laubenbacher and K. Duca. Using formal concept analysis for microarray data comparison. *Journal of Bioinformatics and Computational Biology*, 6(1):65–75, 2008
9. B. Ganter, G. Stumme, and R. Wille. Formal Concept Analysis: Foundations and Applications. *Lecture Notes in AI*, no. 3626, 2005, Springer-Verlag.
10. T. Golub et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
11. M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2), 2001.
12. J. Handl et al. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21:3201–3212, 2005.
13. M. Kaytoue-Uberall, S. Duplessis and A. Napoli. Using Formal Concept Analysis for the Extraction of Groups of Co-expressed Genes. *CCIS* no. 14, Springer-Verlag Berlin Heidelberg, pages 445–455, 2008.
14. J.Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948. Piscataway, NJ: IEEE Service Center, 1995.
15. J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceeding of Fifth Berkeley Symp. Math. Stat. Prob*, volume 1, pages 281–297, 1967.
16. S. Maere, K. Heymans, M. Kuiper. BiNGO: a Cytoscape plugin to assess overrepresentation of Gene Ontology categories in biological networks. *Bioinformatics*, 21:3448–9, 2005.
17. M. Omran, A. Engelbrecht, and A. Salman. Particle swarm optimization method for image clustering. *Pattern Recognition and Artificial Intelligence*, 2005.
18. D. P. Potter. A combinatorial approach to scientific exploration of gene expression data: An integrative method using Formal Concept Analysis for the comparative analysis of microarray data. *Thesis dissertation, Department of Mathematics, Virginia Tech*, 2005.
19. P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational Applied Mathematics*, 20:53–65, 1987.
20. G. Rustici et al. Periodic gene expression program of the fission yeast cell cycle. *Nat. Genetics*, 36:809–17, 2004.
21. Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *Proceedings of IEEE Int. Conf. on Evolutionary Computation*, pages 69–73, 1998.
22. S. Theodoridis and K. Koutroubas. Pattern recognition. *Academic Press*, 1999.
23. E. Tsiporkova and V. Boeva. Two-pass imputation algorithm for missing value estimation in gene expression time series. *Journal of Bioinformatics and Computational Biology*, 5(5):1005–1022, 2007.
24. Zhou et al. Functional annotation and network reconstruction through cross-platform integration of microarray data. *Nature Biotechnology*, 23(2):238–43, 2005.