CLUSTERING OF MULTIPLE DNA MICROARRAYS THROUGH COMBINATION OF PARTICLE SWARM INTELLIGENCE AND K-MEANS

Veselka Boeva
Department of Comp. Systems and Technology
Technical University of Sofia-branch Plovdiv
4000 Plovdiv, Bulgaria
email: vboeva@tu-plovdiv.bg

Anna Hristoskova
Department of Information Technology
Ghent University
9050 Ghent, Belgium
email: anna.hristoskova@intec.UGent.be

Elena Tsiporkova
ICT & Software Engineering Group
Sirris
1030 Brussels, Belgium
email: elena.tsiporkova@sirris.be

ABSTRACT
In this article we propose a hybrid approach for clustering of gene expression data across multiple experiments, based on Particle Swarm Optimization and k-means clustering. In the proposed algorithm, each experiment identifies a particle initialized with the result of the k-means algorithm applied over the experiment. The final clustering solution is found by updating the particles using the information about the best clustering solution generated by each experiment and the entire set of experiments. The performance of the proposed cluster 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 hybrid algorithm is able to produce good quality clustering solution, which is representative for the whole test compendium and at the same time adequately reflects the specific characteristics of the individual experiments.

KEY WORDS
data clustering, k-means, particle swarm intelligence, 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 time points. All these measurements contain information about many different aspects of gene regulation and function, ranging from understanding the global cell-cycle control of microorganisms [4], to cancer in humans [3], [8]. Gene clustering is one of most essential analysis methods for gene expression data. Clustering is the process of grouping data objects into sets of disjoint classes called clusters, so that objects in the same cluster are more similar to each other than objects in the other clusters, given a reasonable measure of similarity. In the context of microarray analysis, clustering algorithms have been 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.

In recent years, many diverse clustering algorithms have been proposed and applied for gene expression data analysis. Three major categories may be distinguished: density-based, hierarchical and partitioning clustering methods [11]. Density-based algorithms implement the so-called local principle to group neighboring objects into clusters based on density conditions and thus, they are capable of discovering clusters of arbitrary shapes [2]. Hierarchical clustering methods generate a set of nested clusters by either merging smaller clusters into larger ones, or by splitting larger clusters in a hierarchical manner [7]. In contrast to these approaches, three partitioning algorithms (k-means, k-medians and k-medoids clustering) decompose the data set into a set of k disjoint clusters such that the within-cluster sum of distances between each object in a given cluster and the corresponding cluster center is minimized.

Presently, with the increasing number and complexity of available gene expression data sets the combination of data from multiple microarray studies addressing a similar biological question is gaining high importance [6], [21], [9]. In general, the integration and evaluation of multiple data sets promise to yield 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 weakened. The latter has motivated our research, which is concerned with how to combine Particle Swarm Optimization (PSO) and k-means clustering algorithms in order to derive general and consistent conclusions from a set of independent, but biologically related, microarray data sets.

PSO-based clustering algorithm was first introduced

by Omran *et al.* [16]. They showed that PSO based method outperformed k-means and a few other state-of-the-art clustering algorithms. In their method, each particle represents a possible set of k cluster centroids. Van de Merwe and Engelbrecht hybridized Omran *et al.* approach with the k-means algorithm for clustering general datasets [14]. A single particle of the swarm is initialized with the result of the k-means algorithm while the rest of the swarm is initialized randomly. Xiao *et al.* proposed a new approach based on the combination of PSO and the Self Organizing Maps [22] and applied it for clustering gene expression data. They obtained promising results by applying the combined algorithm over the gene expression data of Yeast and Rat Hepatocytes.

In contrast to conventional clustering algorithms, where a single data set is used to produce a clustering solution, we propose herein a PSO-based approach that can be used to cluster gene expression data across multiple experiments. In this context, each experiment (dataset) defines a particle which is initialized with a set of k cluster centroids obtained after performing k-means clustering algorithm applied over the experiment. The final (optimal) clustering solution is found by updating the particles using the information about the best clustering solution obtained by each experiment and the entire set of datasets.

Section 2 briefly describes the basic principles of PSO and k-means methods and subsequently introduces our hybrid 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.

2 Methods

2.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is an evolutionary computation method introduced in [12]. 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 about 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 [18]. $P_g = (p_{g1}, p_{g2}, \ldots, p_{gn})$ 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 as the *social component*. Acceleration numbers c_1, c_2 and inertia weight w are predefined by the user. For instance, in [12] it is recommended to use 2 for constants c_1 and c_2 since it results in average weights for the *social* and *cognition parts* of 1. It was shown in [18] that when w is in the range $[0.9, 1.2]$ the 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 [15] 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). 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 particles.

2.2 K-means Clustering Algorithm

The k-means algorithm [13] is one of the most widely used techniques for clustering. It starts by initializing the k cluster centers, where k is preliminarily determined. Then, each object (input vector) of the data set 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 no more change in the value of any of the cluster centers.

2.3 Particle Swarm Optimization and K-means clustering

We use here a combination of PSO and k-means for deriving a clustering result from multiple microarray datasets.

Assume that a particular biological phenomenon is monitored in several high-throughput experiments under n different conditions. Each experiment i ($i = 1, 2, \ldots, n$) is supposed to measure the gene expression levels of m genes in n_i different experimental conditions or time points. Thus a list of n different data matrices $M_1, M_2, \ldots, M_n$ will be produced, one per experiment. In this context, each matrix

i is possible 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 about the best clustering solution obtained by each data matrix and the entire set of matrices. The fitness of particles is measured as the quantization error [14]

$$J_e = \frac{\sum_{l=1}^{k} \sum_{\forall g_j \in C_l} d(g_j, C_l)/k_l}{k}, \tag{3}$$

where C_l is the l-th cluster center and k_l is the number of genes belonging to the l-th cluster.

Assume that the i-th particle is initialized with a set of k cluster centers[1] $C_i = \{C_1^i, C_2^i, \ldots, C_k^i\}$ and a set of velocity vectors $V_i = \{V_1^i, V_2^i, \ldots, V_k^i\}$[2] 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 the update equation for the d-th dimension of the j-the velocity vector of the i-th particle is defined as follows

$$\begin{gathered} v_{jd}^i(t+1) = w \cdot v_{jd}^i(t) + c_1 \cdot \varphi_1 \cdot (p_{jd}^i - c_{jd}^i(t)) + \\ c_2 \cdot \varphi_2 \cdot g(t), \end{gathered} \tag{4}$$

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

$$g(t) = \begin{cases} p_{gd} - c_{jd}^i(t), \text{ if } n_g \geq n_i \\ 0, \text{ otherwise} \end{cases}. \tag{5}$$

In addition, $P_g = \{P_{g1}, P_{g2}, \ldots, P_{gk}\}$ 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 the best solution discovered so far by the corresponding matrix. Evidently, 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 (5)) differs from that in equation (1), since each particle matrix has a different number of columns 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 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 distance of gene g_j with all cluster centers
         B. Assign g_j to the cluster that has 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 (4)
4. **end for**

2.4 Computational Complexity

On extremely large datasets the proposed hybrid clustering algorithm is expected to be rather computationally intensive. Initially the particles are initialized by applying k-means clustering on each given expression matrix. This implies computational complexity of

$$O(I(n_1 + n_2 + \ldots + n_n)mk)$$

for n matrices of m rows (genes), k number of clusters and I number of iterations. Then at the second phase, the PSO algorithm is used to find the final clustering solution by updating the particles using the information about the best clustering solution generated by each experiment and the entire set of experiments, *i.e.* its computational complexity will be in the range of

$$O(J(n_1 + n_2 + \ldots + n_n)mk),$$

where J is the bound number of iterations. Thus the total cost of the proposed hybrid algorithm will be approximately

$$O((I + J)(n_1 + n_2 + \ldots + n_n)mk).$$

This can be drastically reduced by first performing some advanced filtering or features selection in order to remove noisy data and preserve lower number (m) of potentially relevant genes for clustering. It is expected that the latter will subsequently lead to lower cluster number (k).

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

[2] The velocity vectors are initialized by zeros.

3 Experimental Setup

3.1 Microarray Datasets

The proposed clustering algorithm has been validated on benchmark datasets where 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* [4]. The study includes eight independent time-course experiments synchronized respectively by:

1. elutriation: three independent biological repeats;
2. cdc25 block-release: two independent biological repeats, of which one in two dye-swapped technical replicates, and one experiment in a sep1 mutant background;
3. a combination of both methods: elutriation and cdc25 block-release as well as elutriation and cdc10 block-release.

Thus, nine different expression test sets are available. In the pre-processing 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 [20]. In this way nine complete matrices have been obtained.

Rustici *et al.* identified 407 genes as cell-cycle regulated [4]. These have been subjected to clustering which resulted in the formation of 4 separate clusters. The genes that are not presented in the intersection of the nine original data sets have been removed. The latter produces a subset of 267 genes. Subsequently, the time expression profiles of these genes have been extracted from the complete data matrices and thus nine new matrices which form our benchmark datasets have been constructed.

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 [1].

3.2 Cluster Validation Measures

One of the most important issues in cluster analysis is the validation of 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 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 clusters:

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

3.2.1 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 [5]. Let us define $m_{i(j)}$ as the jth nearest neighbour 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.

3.2.2 Silhouette Index

Silhouette index reflects the compactness and separation of clusters [17]. 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 *Silhouette Index* 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 Results and Discussion

In this section, the performance of the proposed PSO-based clustering method on the benchmark datasets is presented. The standard k-means and the proposed hybrid (combination of k-means and PSO) clustering algorithm 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 proposed PSO-based clustering algorithm has been implemented in Java. The publicly available open source machine learning software WEKA[3] is used by this implementation for the particle initialization and for the gene assignment to the different clusters.

Initially, the number of cluster centers is identified for the involved experiments. As discussed in [10], [19], this

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

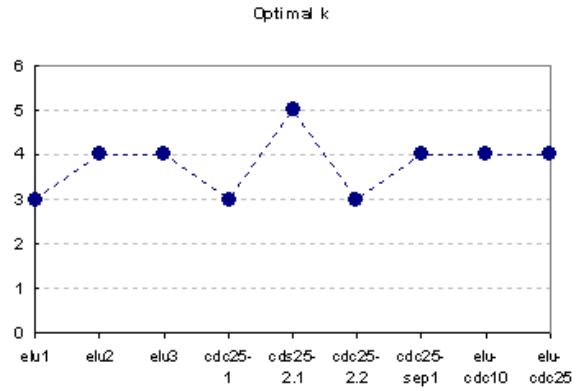

Figure 1. Optimal number of clusters as determined using the Connectivity validation index for the 9 different experiments. For 5 of the 9 datasets $k = 4$ is identified as the optimal cluster number.

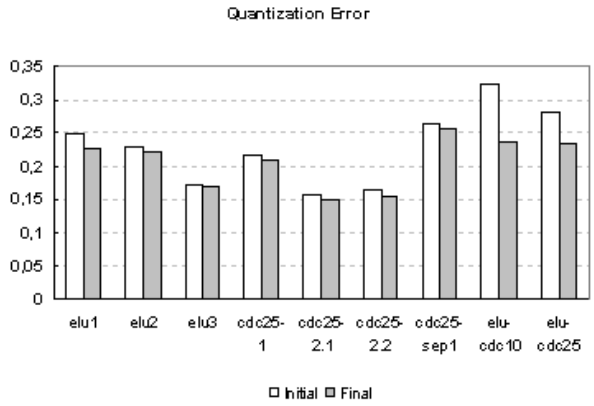

Figure 2. Comparison of the fitness function scores generated by PSO-based clustering algorithm at the initial and the final iteration and based on the average results of 11 separate simulations. The clustering solutions of all the experiments are improved during the algorithm execution. The global best clustering solution is generated in 80% of the simulations by experiment *cdc25-2.1*.

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 validation index. We search for the values of k at which a significant local change in value of the index occurs [10]. The optimal number of clusters for the different experiments range between 3 and 5 as it can be seen in Figure 1. However, $k = 4$ prevails (encountered in five matrices) and therefore it will be used for our experiments.

Next the proposed PSO-based clustering algorithm (see Section 2.3) is executed on the whole test corpus. It is run for 500 iterations and $w = 0.72$ and $c_1 = c_2 = 1.49$. These values have been chosen to ensure good convergence [15].

Figure 2 depicts the calculated fitness function (the quantization error) values (see equation (3)) on the test datasets at the initial time point versus those generated at the last iteration of the PSO-based clustering. 11 separate simulations are performed in total and the figure presents the average values over them. It can be seen that the clustering solutions of all the experiments are improved during the algorithm execution. It is also interesting to note that the global best clustering solution is generated in 80% of the simulations by experiment *cdc25-2.1*.

Figure 3 compares the SI values generated by the k-means and the proposed PSO-based hybrid clustering algorithm on the individual matrices. Note that the SI values for the PSO-based algorithm are obtained by using the global best solution found among the 9 different experimental matrices, while the values for the k-means are produced by using the clustering solutions generated for each of the corresponding individual datasets. It can be observed that the proposed PSO-based algorithm outperforms the k-means algorithm for the datasets with the worst (according to the SI) clustering solution, namely *elu1, cdc25-2.1, elu-cdc10, elu-cdc25*. In addition, as it can be witnessed by the SI results presented in Figure 3, the PSO-based clustering solution is well supported by all the experiments. In other words, the algorithm performs equally well on the different experiments and clearly reduces the high SI fluctuation among the different experiments as exhibited by the k-means algorithm.

Figure 4 depicts the Connectivity values generated by applying the conventional k-means clustering algorithm on each test dataset versus the ones produced by the PSO-based clustering on the whole test corpus. The Connectivity scores generated by PSO-based clustering are significantly better than the values obtained by applying the k-means algorithm for all the experiments with the exception of *elu-cdc10*. However, as it can be seen the k-means result produced on the latter experiment deviates considerably from the Connectivity scores obtained for the rest of the experiments. This may be due to the experiment specific characteristics.

Figures 5 and 6 compare the SI and Connectivity values produced by the standard k-means and the PSO-based clustering algorithm against the ones generated by using the best k-means solution with respect to the both validation indices. The latter one is attained for experiment *cdc25-2.2* which generates the highest SI score and the second best Connectivity result. As it can be noticed the SI values produced by the best k-means clustering solution slightly outperform the ones obtained by applying the PSO-based algorithm for almost all the experiments. However, the corresponding Connectivity results are significantly worse than those produced by the proposed PSO-

Figure 3. Comparison of the SI values generated by the standard k-means and the proposed PSO-based hybrid clustering algorithm on the 9 different experiments. The SI values for the PSO-based algorithm are obtained by using the global best solution found among the 9 different experimental matrices, while the values for the k-means are produced by using the clustering solutions generated for each of the corresponding individual datasets.

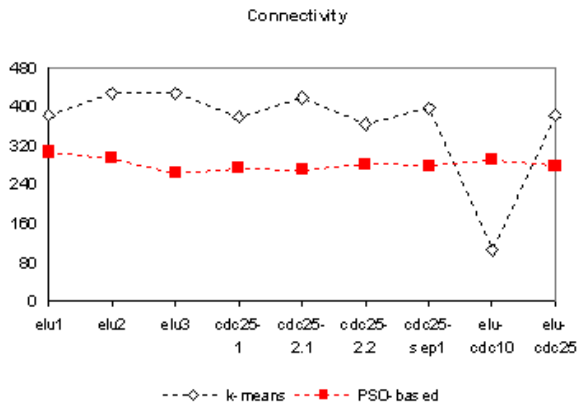

Figure 4. Comparison of the Connectivity values generated by the standard k-means and the proposed PSO-based hybrid clustering algorithm on the 9 different experiments. The Connectivity values for the PSO-based algorithm are obtained by using the global best solution found among the 9 different experimental matrices, while the values for the k-means are produced by using the clustering solutions generated for each of the corresponding individual datasets.

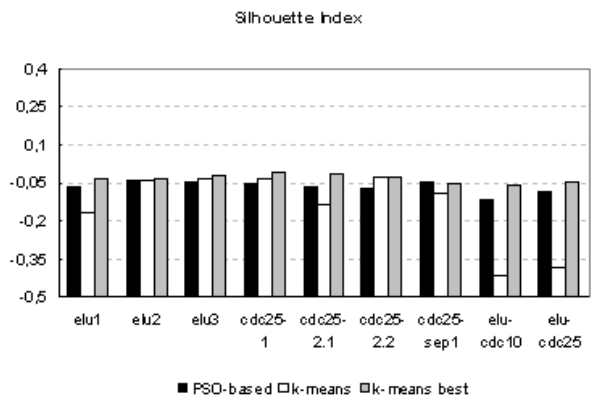

Figure 5. Comparison of the SI values generated by the standard k-means and the PSO-based clustering algorithm and those obtained by using the best k-means clustering result on the 9 different experiments. The SI values for the 'k-means best' are obtained by using the clustering solution among the 9 different experimental matrices which has the best performance with respect to the both, SI and Connectivity, validation measures.

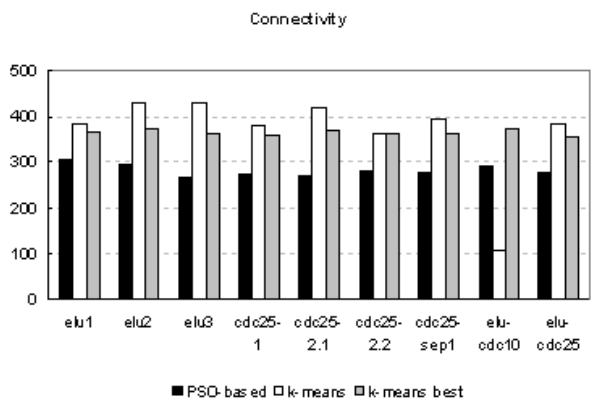

Figure 6. Comparison of the Connectivity values generated by the standard k-means and the PSO-based clustering algorithm and those obtained by using the best k-means clustering result on the 9 different experiments. The Connectivity values for the 'k-means best' are obtained by using the clustering solution among the 9 different experimental matrices which has the best performance with respect to the both, SI and Connectivity, validation measures.

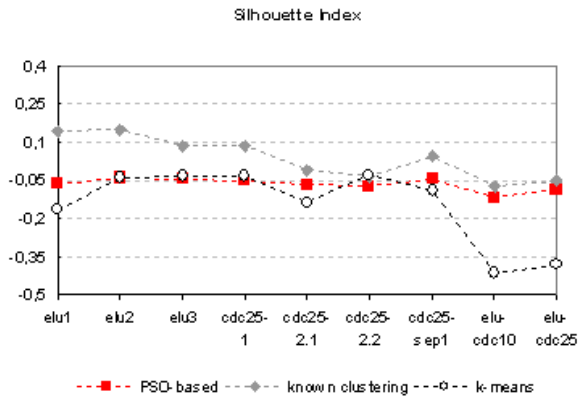

Figure 7. Comparison of the SI values generated by the known clustering solution published in [4], and those obtained by applying the standard k-means and the proposed PSO-based hybrid clustering algorithm on the 9 different experiments.

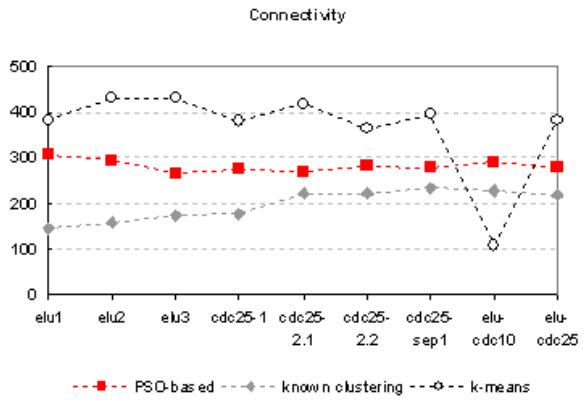

Figure 8. Comparison of the Connectivity values generated by the known clustering solution published in [4], and those obtained by applying the standard k-means and the proposed PSO-based hybrid clustering algorithm on the 9 different experiments.

based clustering algorithm. Evidently, it is difficult to find a k-means clustering solution that presents equally well with respect to all the evaluation criteria. This observation is further confirmed by the fact that experiment *elu-cdc10* produces the best Connectivity result and the worst SI score at the same time. While the proposed PSO-based clustering algorithm finds a partitioning that performs uniformly well under the both validation measures.

Figures 7 and 8 visualize the SI and Connectivity values calculated by using the known clustering solution found (manually) in [4] against those obtained by applying the standard k-means and the proposed PSO-based clustering algorithm on the benchmark matrices. It can be seen that in comparison to the k-means clustering solution the performance of the PSO-based clustering for both indices is much closer to that of the already published partitioning.

5 Conclusion

We have proposed a hybrid clustering method which combines Particle Swarm Optimization and k-means for deriving a global clustering solution for multiple gene expression matrices. The performance of the proposed clustering algorithm has been evaluated on a compendium 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 the article initial experimental results demonstrate that the proposed PSO-based algorithm is able to produce good quality clustering solution, which is representative for the whole test compendium and at the same time adequately reflects the specific characteristics of the individual experiments. In addition, the proposed clustering algorithm outperforms the k-means algorithm for majority of the datasets and in particular for ones with the worst k-means performance.

Our future work will focus on further improvement, parameter optimization and validation of the proposed clustering method on various different in terms of type and size experimental datasets.

References

[1] V. Boeva and E. Tsiporkova. A multi-purpose time series data standardization method. *Intelligent Systems: From Theory to Practice, Springer-Verlag Berlin Heidelberg*, 299:445–460, 2010.

[2] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd ACM SIGKDD, Portland, Oregon*, pages 226–231, 1996.

[3] A. Alizadeh et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[4] G. Rustici et al. Periodic gene expression program of the fission yeast cell cycle. *Nat. Genetics*, 36:809–17, 2004.

[5] J. Handl et al. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21:3201–3212, 2005.

[6] J.K. Choi et al. Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, 19:i84–i90, 2003.

[7] M. Eisen et al. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of Natl. Acad. Science, USA*, volume 95, pages 14863–14868, 1998.

[8] T. Golub et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[9] Zhou et al. Functional annotation and network reconstruction through cross-platform integration of microarray data. *Nature Biotechnology*, 23(2):238–43, 2005.

[10] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.

[11] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[12] 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.

[13] 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.

[14] D. Merwe and A. Engelbrecht. Data clustering using particle swarm optimization. In *Proceedings of IEEE Congress on Evolutionary, Congress on Evolutionary Computation, Piscataway, NJ: IEEE Service Center*, pages 215–220, 2003.

[15] M. Omran, A. Engelbrecht, and A. Salman. Particle swarm optimization method for image clustering. *Pattern Recognition and Artificial Intelligence*, 19(3):297–321, 2005.

[16] M. Omran, A. Salman, and A. Engelbrecht. Image classification using particle swarm optimization. In *Proceedings of Conference on Simulated Evolution and Learning*, volume 1, pages 370–374, 2002.

[17] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational Applied Mathematics*, 20:53–65, 1987.

[18] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *Proceedings of IEEE Int. Conf. on Evolutionary Computation*, pages 69–73, 1998.

[19] S. Theodoridis and K. Koutroubas. Pattern recognition. *Academic Press*, 1999.

[20] 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.

[21] A. Brazma W.R. Gilks, B.D.M. Tom. Fusing microarray experiments with multivariate regression. *Bioinformatics*, 21(2):ii137–ii143, 2005.

[22] X. Xiao, E.R. Dow, R.C. Eberhart, Z. B. Miled, and R. J. Oppelt. Gene clustering using self-organizing maps and particle swarm optimization. In *Proceedings of the 17th International Symposium on Parallel and Distributed Processing, IEEE Computer Society, Washington DC*, 2003.