A New Approach for Visualizing Fuzzy-clustered Microarray Data

LUIS RUEDA YUANQUAN ZHANG
School of Computer Science
University of Windsor
401 Sunset Ave., Windsor, ON N9B 3P4
CANADA

Abstratct: Microarrays are relatively new techniques that allow scientists to measure the expression level of thousands of genes simultaneously. Many clustering methods are currently applied to analyze microarray data, including fuzzy k-means, which allows an object to be assigned to multi-clusters with different degree of membership. However, the memberships which result from fuzzy k-means, are rarely analyzed and visualized properly, but converted to 0-1 memberships. In this paper, we propose a new approach to visualize fuzzy-clustered data. The scheme provides a geometric view by grouping the objects with similar cluster memberships, and shows clear advantages over existing methods, demonstrating its capabilities for viewing and navigating inter-cluster relationships in a spatial manner. The advantages are analyzed on microarray data for experiments on the cell cycle of the budding yeast.

Key-Words: yeast expression analysis, clustering visualization, microarray analysis, fuzzy clustering.

1 Introduction

Microarray constitute quite useful tools that allow scientists to analyze the expression levels of thousands of genes on snapshots made in one or several experiments. There are two approaches based on different microarray fabrications: cDNA microarray and *In situ* synthesis. The relative information about cDNA microarray data processing and data analysis is described in [1, 2]. In order to study the gene hybridization during the biological experiments, the gene expression values are measured along time series. The result of gene expression values is stored in an $n \times p$ matrix, where n is the number of genes, and p is the number of measurement at different points in time. An important step in analyzing such a massive amount of data is to group the genes that show a certain *degree of similarity*. In this regard, clustering is a well-known technique that is commonly applied.

Many clustering algorithms have been proposed so far, including k-means [3], fuzzy k-means [4], expectation maximization (EM) [5], hierarchical clustering [6], and self-organizing maps [7]. Fuzzy clustering is an approach that consists of grouping objects which share a certain degree of similarity. This method assigns a certain probability of cluster membership to each object corresponding to the distance between the object and the centroid of each cluster. Applying fuzzy clustering to microarray data brings the advantage that the clustering result allows a gene to be assigned to more than one cluster [4, 8]. The problem, however, is how to assign the genes (objects) to one of the clusters. A common technique to deal with this is to use a "cutoff" value and assign an object to a cluster if its membership value is above the cutoff. On the other hand, visualizing the "fuzzy" membership of an object belonging to different clusters constitutes in itself an interesting and challenging problem.

In the past, two methods addressed the topic of visualizing fuzzy-clustered data. Berthold *et al.* proposed a method to visualize fuzzy points in parallel coordinates [9]. They point out the disadvantage of the existing fuzzy point techniques, which only show centroids or use shaded areas to represent the general variance of cluster centers. An extended version of parallel coordinates visualization is to apply the color shading for representing the degree of membership. Gasch *et al.* proposed a modified hierarchical visualization to observe the fuzzy-clustered genes utilizing different membership cutoff values [10].

The main drawback of these two approaches is that they cannot provide a visualization scheme,

which represents the position of clustered points corresponding to each other. In this paper, we present a visualization approach that solves this problem. Our approach projects the fuzzy membership data points onto a *multi-dimensional tetrahedron*, which allows to observe the inter-cluster relationships in a spatial manner.

2 Fuzzy clustering

Fuzzy clustering, (also called "soft" clustering) assigns each sample to multiple clusters using a membership value, which is the probability that the sample belongs to the corresponding cluster. Various methods use this idea, being the most widely used ones, fuzzy k-means and expectation maximization [5].

Consider a dataset $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ where $\mathbf{x}_i = [x_1, x_2, x_3, \ldots, x_p]^t$ is a p-dimensional feature vector that represents a sample (gene), where x_r is the r^{th} feature.

The aim is to partition D into k clusters (classes), $\omega_1, \ldots, \omega_k$, in such a way that samples that belong to the same cluster are similar to each other. In fuzzy clustering, the membership is given by the probability that $\mathbf{x}_i$ belongs to cluster ω_j, $\hat{P}(\omega_j|\mathbf{x}_i)$.

Fuzzy k-means seeks a minimum of a heuristic global cost function:

$$J_{fuz} = \sum_{j=1}^{k}\sum_{j=1}^{n}[\hat{P}(\omega_j|\mathbf{x}_i)]^b d(\mathbf{x}_i, \mu_j) \quad , \qquad (1)$$

where b is a parameter, which is greater than unity,

$$\mu_j = \frac{\sum_{i=1}^{n}[\hat{P}(\omega_j|\mathbf{x}_i)]^b \mathbf{x}_i}{\sum_{i=1}^{n}[\hat{P}(\omega_j|\mathbf{x}_i)]^b} \quad , \qquad (2)$$

is the cluster center of ω_j

$$\hat{P}(\omega_j|\mathbf{x}_i) = \frac{(1/d_{ji})^{1/(b-1)}}{\sum_{r=1}^{k}(1/d_{ri})^{1/(b-1)}} \quad , \qquad (3)$$

and $d(\mathbf{x}_i, \mu_j)$ (d_{ij} for short) is a distance function that states the "dissimilarity" between $\mathbf{x}_i$ and μ_j.

Fuzzy k-means proceeds in an iteratively manner. It receives k and b as parameters, and initializes $\mu_1, \ldots, \mu_k$, and $\hat{P}(\omega_j|\mathbf{x}_i)$, $i = 1, \ldots, n, j = 1, \ldots, k$. It then iteratively recomputes μ_j and $\hat{P}(\omega_j|\mathbf{x}_i)$ using (2) and (3) until a small change in μ_j and $\hat{P}(\omega_j|\mathbf{x}_i)$ is observed.

Another criteria that can be optimized, instead of (1), is to maximize the likelihood that a sample belongs to ω_j with a probability given by a *mixture of densities*. In such a case, the aim is to estimate the parameters, θ , while maximizing the likelihood. A common approach used in optimizing this criterion is the EM algorithm.

The result of both fuzzy k-means and EM is then a $k \times n$ membership matrix, and our aim is to provide an efficient scheme to visualize these data.

3 The Visualization Method

The visualization method that we introduce in this paper takes the membership matrix, and performs three different steps. It, first of all, finds the k vertices of a regular hyper-tetrahedron, onto which all the n data points are projected. It finally transform the regular tetrahedron into an irregular one that reflects the inter-cluster center distances.

3.1 Obtaining Vertices of the Regular Space

As pointed out earlier, fuzzy clustering results in a $k \times n$ membership matrix, $\mathbf{M}$. The columns of $\mathbf{M}$ are n vectors, $\mathbf{m}_1, \ldots, \mathbf{m}_n$, where $\mathbf{m}_i = [m_{1i}, m_{2i}, \ldots, m_{ki}]^t$ is a k-dimensional vector and m_{ji} contains the probability that $\mathbf{m}_i$ belongs to ω_j. The important feature of the membership matrix is that the sum of each column is unity, i.e. $\sum_{j=1}^{k} m_{ji} = 1$. The visualization method described in this paper is based on this observation.

We see the vectors $\mathbf{m}_1, \ldots, \mathbf{m}_n$ lying in the k-dimensional space, where the j^{th} cluster ω_j is represented by a k-dimensional vector $\mathbf{y}_j = [y_{j1}, \ldots, y_{jk}]^t$ with $y_{jr} = 0$ for $r = 1, \ldots, k$, $r \neq j$ and $y_{jj} = 1$. The first step is to project these vectors onto the $(k-1)$-dimensional space, where the k clusters are represented by k points, $\mathbf{y}'_1, \ldots, \mathbf{y}'_k$, in the $(k-1)$ dimensional space. These points compose a metric $\mathbf{Y}'_k = [\mathbf{y}'_1, \ldots, \mathbf{y}'_k]^t$, which is computed as per the algorithm below.

Algorithm 1 Regular_Tetrahedra_Vertices.

Step 1. Initialize $\mathbf{Y}'_2 \leftarrow \begin{bmatrix} 0 & 0 \\ \sqrt{2} & 0 \end{bmatrix}$.

Step 2. Let

$$\mathbf{Y}'_j = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ y'_{21} & 0 & 0 & \ldots & 0 \\ y'_{31} & y'_{32} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y'_{j1} & y'_{j2} & \ldots & y'_{j(j-1)} & 0 \end{bmatrix}, \qquad (4)$$

$$\mathbf{D}'_j = \begin{bmatrix} 0 & d'_{12} & d'_{13} & \dots & d'_{1j} \\ 0 & 0 & d'_{23} & \dots & d'_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d'_{(j-1)j} \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad (5)$$

where $\mathbf{y}_j = [y_{j1}, y_{j2}, \dots, y_{jk}]^t$.

Step 3.

Assume that the distance from $\mathbf{y}'_{j+1}$ to $\mathbf{y}'_1, \mathbf{y}'_2, \dots, \mathbf{y}'_j$ is given by a j-dimensional vector $\mathbf{d}'_{j+1} = [d'_{(j+1)1}, d'_{(j+1)2}, \dots, d'_{(j+1)j}]^t$.

We set $\mathbf{d}'_{j+1} \leftarrow [\sqrt{2}, \sqrt{2}, \dots, \sqrt{2}]^t$, since those vertices compose a hyper-tetrahedron whose edge length is $\sqrt{2}$ in the j-dimensional space. Then, $\mathbf{y}'_{j+1}$ is computed as follows:

$$[y'_{(j+1)1}, \dots, y'_{(j+1)(j-1)}]^t \leftarrow \frac{1}{2}\mathbf{Y}'_c(\hat{\mathbf{d}} + \mathbf{y}'_s) \quad , \quad (6)$$

where

$$\mathbf{y}'_s = \begin{bmatrix} y'^2_{21} \\ y'^2_{31} + y'^2_{32} \\ \vdots \\ y'^2_{j1} + y'^2_{j2} + \dots + y'^2_{j(j-1)} \end{bmatrix}, \quad (7)$$

$$\mathbf{Y}'_c = \begin{bmatrix} y'_{21} & 0 & \dots & 0 \\ y'_{31} & y_{32} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \dots & y_{n(n-1)} \end{bmatrix}, \quad \text{and} \quad (8)$$

$$\hat{\mathbf{d}} = [d^2_{v1} - d^2_{v2}, d^2_{v1} - d^2_{v3}, \dots, d^2_{v1} - d^2_{vn}]^t. \quad (9)$$

Step 4. The last component of $\mathbf{y}'_{j+1}$ is computed as:

$$y'_{(j+1)j} = \sqrt{d'^2_{(j+1)1} - y'^2_{(j+1)1} \cdots - y'^2_{(j+1)(j-1)}} \quad . \quad (10)$$

As a result, we obtain a j-dimensional vector, $\mathbf{y}'_{j+1}$.

Step 5. Transform $\mathbf{y}'_1, \dots, \mathbf{y}'_{j+1}$ into $(j+1)$-dimensional vectors as follows: $\mathbf{y}'_r \leftarrow [(\mathbf{y}'_r)^t, 0]^t$, where $r = 1, \dots, j+1$, and update $\mathbf{Y}'$ by $\mathbf{Y}' \leftarrow [\mathbf{y}'_1, \dots, \mathbf{y}'_r]^t$, where $r = 1, \dots, j+1$.

Step 6. If $j + 1 = k$, stop. Otherwise, go to **Step 2.**

3.2 Transferring Objects to a Regular Tetrahedron

The vectors in $\mathbf{M}$ can be seen as points lying in the k-dimensional space, and the aim now is to transform the points $\mathbf{m}_i$ into new points, $\mathbf{m}'_i$, which are enclosed in a regular hyper-tetrahedron in the $(k-1)$-dimensional space, where the vertices of the hyper-tetrahedron are given by $\mathbf{Y}'$. In particular, for $k = 3$, the points will lie on an equilateral triangle and $\mathbf{Y}' = \begin{bmatrix} 0 & 0 & 0 \\ \sqrt{2} & 0 & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{2} & 0 \end{bmatrix}$.

The procedure for transferring $\mathbf{m}'_i$ to a regular hyper-tetrahedron is similar to Algorithm 1 except the following two steps. First, $\mathbf{d}_i$ is a distance vector that contains the k distances from $\mathbf{m}_i$ to each $\mathbf{y}_j$, where $j = 1, \dots, k$. Second, since $\mathbf{m}'_i$ represents a point in the $(k-1)$-dimensional space, it is not necessary to perform Steps 4, 5 and 6.

As a result, we obtain all points $\mathbf{m}'_1, \dots, \mathbf{m}'_n$, which are $(k-1)$-dimensional points enclosed in a hyper-tetrahedron given by vertices $\mathbf{y}'_1, \dots, \mathbf{y}'_k$.

3.3 Creating an Irregular Tetrahedron

It is important to note that, at this point, each cluster is represented by a center (vertex) and a "fixed radius", which is unity for every cluster. However, this model does not reflect the actual clusters produced by the fuzzy clustering algorithm, which also depend on the distance function being used. Then, to enhance the visualization we show the points in an irregular hyper-tetrahedron that uses the corresponding edges to reflect the inter-cluster distances. Let $\mathbf{D}''$ be a $k \times k$ matrix that contains the distances between each pair of vertices, where d_{ij}, $i, j = 1, \dots, k$ and $i < j$, represents the distance from the centroid of the i^{th} cluster to the centroid of the j^{th} cluster. The value of d_{ij} depends on the specific distance function used in fuzzy k-means. To avoid the extremely large distance value that may result from certain real-life problems, we let $d''_{12} = \sqrt{2}$ which coincides with the distance in the regular hyper-tetrahedron. Therefore, $d''_{ij} = \sqrt{2}\frac{d_{ij}}{d''_{12}}$, for $1 \le i \le k, 1 \le j \le k$, results in a relative distance based on d''_{12}. Thus, $\mathbf{Y}''_j$ and $\mathbf{D}''_j$ have the same format as $\mathbf{Y}'_j$ and $\mathbf{D}'_j$.

We now apply a procedure similar to Algorithm 1, except the vectors $\mathbf{y}'_1, \dots, \mathbf{y}'_n$ are transformed into vectors $\mathbf{y}''_1, \dots, \mathbf{y}''_n$ lying in an irregular tetrahedron. In addition, the distance vector, $\mathbf{d}''_{j+1}$ is computed as above, i.e. $\mathbf{d}''_{j+1} = [d''_{(j+1)1}, d''_{(j+1)2}, \dots, d''_{(j+1)j}]^t$, where $d''_{ij} = \sqrt{2}\frac{d_{ij}}{d''_{12}}$, for $1 \le i \le k, 1 \le j \le k$.

To reflect the distances between vertices in the visualization, the points in the regular hyper-tetrahedron are stretched along a series of steps, which depend on the distance values. For each pair of vertices, the distance between these vertices in

the tetrahedron is equivalent to the distance between the corresponding clusters in the clustering algorithm. All points inside the tetrahedron that includes all vertices are shifted together.

The transformed coordinates of the n vertices of the hyper-tetrahedron are contained in $\mathbf{Y}''_k$, which has the same format as $\mathbf{Y}'_k$.

Note that $\mathbf{y}''_1 = [0, \ldots, 0]^t$, which implies that the vector is located in the origin of the coordinate system. The coordinates for point $\mathbf{m}''_i$ are computed as follows:

$$\mathbf{m}''_i = \mathbf{m}'_i + [\rho^i_r]^t \mathbf{F} \quad , \tag{11}$$

where

$$\rho^i_r = \frac{\mathbf{m}'_{ir}}{y_{(r+1)r}} \ , \quad 1 \le r \le j, \tag{12}$$

$$f_{ij} = y''_{i(j+1)} - y'_{i(j+1)} \ , \qquad 1 \le i \le k, 1 \le j \le k-1 \ , \tag{13}$$

represents the difference between the point on the irregular tetrahedron and that of the regular one.

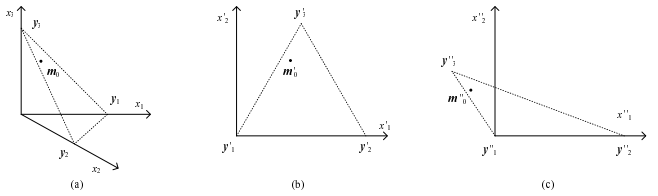


Figure 1: A two-dimensional visualization for the 3-cluster problem.

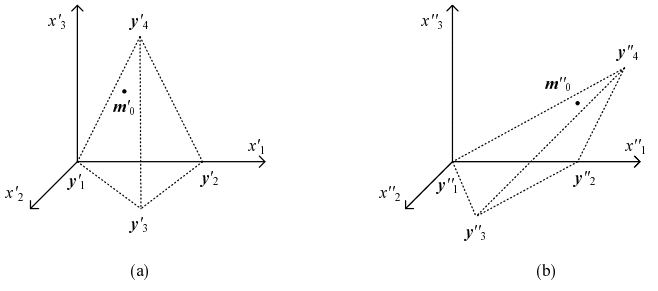


Figure 2: Visualization of fuzzy-clustered data in the three-dimensional case, where $k = 4$.

We now discuss two examples which help understand the transformations involved in the proposed visualization method.

Figure 1 (a) represents the visualization of three clusters in the three-dimensional original space, which can be transformed into a regular hyper-tetrahedron (an equilateral triangle) in the two-dimensional space, as shown in Figure 1 (b). The corresponding irregular triangle and the transformed point, $\mathbf{m}''_0$, are shown in Figure 1 (c).

The visualization of four clusters in the four-dimensional original space can not be plotted, and hence the corresponding figure is omitted. However, the fuzzy-clustered data can be transformed into a tetrahedron in the three-dimensional space, as shown in Figure 2 (a). The irregular tetrahedron and the transformed point, $\mathbf{m}''_0$, which is contained in it, are displayed in Figure 2 (b).

4 Simulations on Real-life Data

The visualization scheme presented in this paper was applied to the result of performing fuzzy k-means on the yeast dataset [11]. The visualization is presented in the three-dimensional space and shows the distribution of the fuzzy-clustered data which were clustered into four classes. In our simulations, we use fuzzy k-means and two distance functions: the Euclidean distance, which is computed as follows:

$$d(\mathbf{x}_i, \mu_j) = ||\mathbf{x}_i - \mu_j||^2 \quad , \tag{14}$$

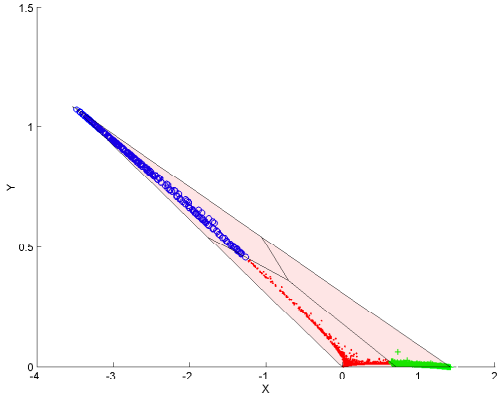
and the Pearson correlation, obtained as:

$$d(\mathbf{x}_i, \mu_j) = 1 - \frac{\sum_{r=1}^{p}(x_{ik} - \bar{x}_i)(\mu_{jr} - \bar{\mu}_j)}{\sqrt{\sum_{k=1}^{k-1}(x_{ir} - \bar{x}_i)^2}\sqrt{\sum_{r=1}^{p}(\mu_{jr} - \bar{\mu}_j)^2}} \tag{15}$$

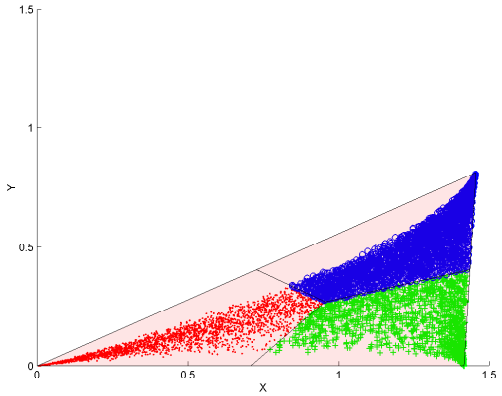
where $\bar{x}_i$ is the mean of $x_{i1}, \ldots, x_{ip}$ and $\bar{\mu}_i$ is the mean of $\mu_{i1}, \ldots, \mu_{ip}$.

The Yeast dataset contains the expression profiles of 6,200 yeast genes whose expression was measured along a time series. We applied two different distance functions, the Euclidean distance and the Pearson correlation distance, to "fuzzy cluster" the dataset into three and four classes. In addition, we applied each fuzzy clustering algorithm to the original yeast data set and the normalized yeast dataset [11]. In our case, a point in the visualization scheme represents a gene in the yeast dataset.

The two-dimensional visualization shown in Figure 3 (a), and the three-dimensional visualization shown in Figure 4 (a) come from the membership table obtained from fuzzy 3-means and fuzzy 4-means clustering, which were applied to the yeast dataset using the Euclidean distance. Figures 3 (b) and 4 (b) depict the visualization of fuzzy 3-means and fuzzy 4-means clustering using the correlation distance. In these figures, only the points lying in the irregular tetrahedron are plotted. This means that the figures are "stretched" depending on the corresponding distance between each pair of cluster cen-

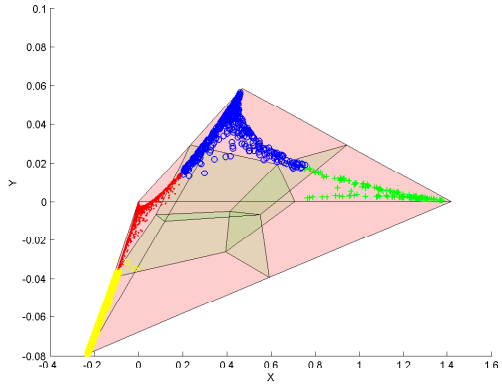


(a) Euclidean distance

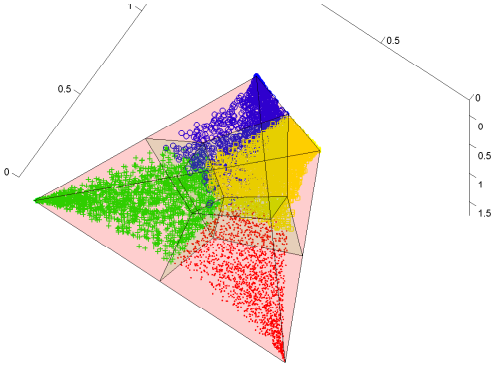


(b) Correlation distance

Figure 3: Visualization of fuzzy 3-means clustering results using the original yeast data set.



(a) Euclidean distance



(b) Correlation distance

Figure 4: Visualization of fuzzy 4-means clustering results using the original yeast data set.

troids. Different point patterns were used to assign the clusters to which the point most likely belongs.

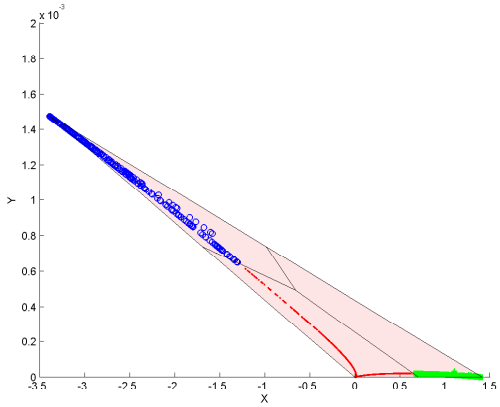
In the normalized yeast dataset, each original value of the time series is divided by the mean. We show four figures for the visualization of fuzzy-clustered data, which are similar to Figures 3 and 4. Figure 5 represents the visualization of 3-means fuzzy-clustered memberships, while Figure 6 shows the visualization of fuzzy 4-means memberships. There are four different patterns representing four areas of the tetrahedra. A point inside one area has a higher membership to the corresponding cluster than its membership to other clusters.

5 Discussion and Conclusion

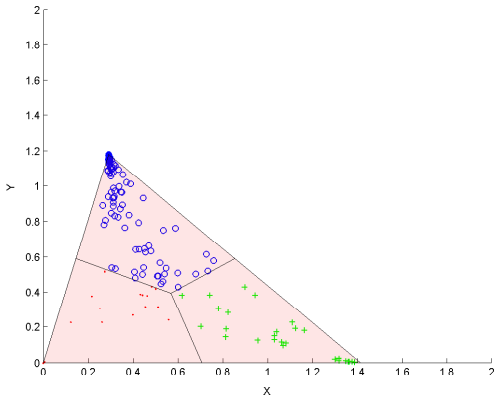
Compared to the visualization of the original yeast data set, the genes in the visualization of the normalized data set are more concentrated near the vertices than the genes of original data set. From this point of view, we conclude that the clustering result of the normalized data set has more reliable clustering than the result of the original dataset.

Regarding the comparison between the Euclidean distance and the correlation distance visualization, the latter provides a more reliable clustering result. The distribution of the points in the visualization of the clustering result applying the Euclidean distance has the form of a triangle which "squeezes" to nearly a line. Note that we have reduced the viewing range in order to avoid this situation. The distribution of the points in the visualization of the clustering result using the correlation distance sparsely appear inside the tetrahedra, thus, enhancing their visualization.

The proposed scheme not only provides a precise visualization of the probability of a point belonging to each cluster, but also represents the geometric distribution of the points in the two or three-dimensional spaces. The future work focuses on how to deal with the restriction of visualizing fuzzy membership data which contains more than four classes. We plan to extract a subspace of the clustered data, which allows the user to visualize

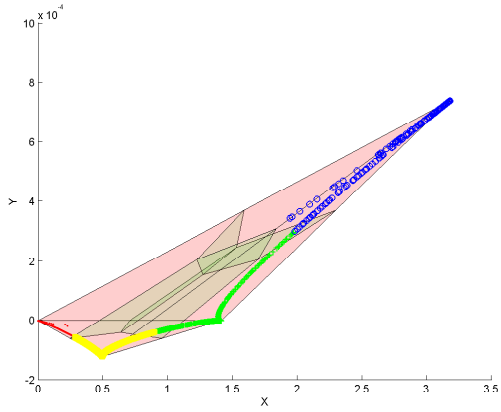


(a) Euclidean distance

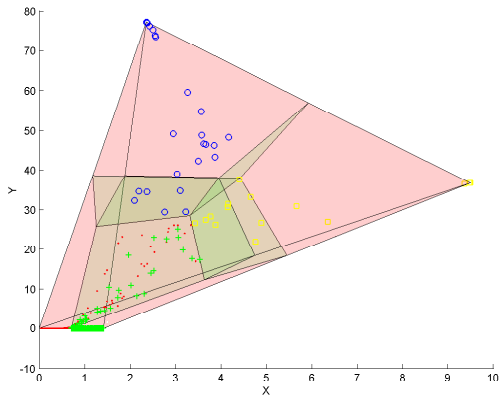


(b) Correlation distance

Figure 5: Visualization of fuzzy 3-means clustering results using the normalized yeast dataset.



(a) Euclidean distance



(b) Correlation distance

Figure 6: Visualization of fuzzy 4-means clustering results using the normalized yeast dataset.

sub-sets of classes and project them onto the three-dimensional space.

Acknowledgments: This research works has been partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada, CFI, the Canadian Foundation for Innovation, and OIT, the Ontario Innovation Trust.

References

[1] Draghici S. *Data Analysis Tools for DNA Microarrays.* Prentice-Hall, Inc., 2003.

[2] Schena M. *Microarray Analysis.* Wiley-Liss, 2002.

[3] Hartigan J. *Clustering Algorithms.* John Wiley and Sons, Inc., New York, 1975.

[4] Dembele D. and Kastner P. Fuzzy C-means Method for Clustering Microarray Data. *Bioinformatics*, vol. 19(8), May 2003, pp. 973–80.

[5] Duda R., Hart P., and Stork D. *Pattern Classification, 2nd Edition.* John Wiley and Sons, Inc., New York, NY, 2000.

[6] Eisen M., Spellman P., Brown P., and Botstein D. Cluster Analysis and Display of Genome-Wide Expression Patterns. *Proceedings of the National Academy of Sciences, USA*, vol. 95, 1998, pp. 14,863–14,868.

[7] Kohonen T. *The Self-Organizing Map, 3rd Edition.* Springer, 2001.

[8] Futschik M.E. and Kasabov N.K. Fuzzy Clustering of Gene Expression Data. *World Congress of Computational Intelligence WCCI*, 2002.

[9] Berthold M.R. and Hall L.O. Visualizing Fuzzy Points in Parallel Coordinates. *Fuzzy Systems*, vol. 11, June 2003, pp. 369–374.

[10] Gasch A.P. and Eisen M.B. Exploring the Conditional Coregulation of Yeast Gene Expression through Fuzzy k-means Clustering. *Genome Biology*, vol. 3(11), 2002, pp. 1–22.

[11] Cho R.J., J.Campbell M., Winzeler E.A., Steinmetz L., Conway A., Wodicka L., Wolfsberg T.G., Gabrielian A.E., Landsman D., Lockhart D.J., and Davis R.W. A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell*, vol. 2, July 1998, pp. 65–73.