Evolutionary Fuzzy Clustering for Gene Expression Profile Analysis

Han-Saem Park and Sung-Bae Cho
Dept. of Computer Science, Yonsei University
134 Shinchon-Dong Seodaemun-Gu Seoul 120-749, Korea
email: sammy@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

***Abstract*—Fuzzy clustering, which is one category of clustering method, assigns one sample to multiple clusters according to their degrees of membership. It is more appropriate than hard clustering for analyzing gene expression profiles because single gene might involve multiple genetic functions. Generic clustering methods, however, have inherent problems that they are sensitive to initialization and can be trapped into local optima. In order to solve these problems, we propose an evolutionary fuzzy clustering method which uses genetic algorithm for clustering and Bayesian validation for evaluation. We have conducted thorough experiments to verify the usefulness of the proposed method with well-known gene expression profiles of SRBCT and Saccharomyces.**

I. INTRODUCTION

Analysis of gene expression profiles can be divided into two groups: classification method based on supervised learning which employs data with known classes and clustering method based on unsupervised learning which deals with the data with unknown or partially known classes [1]. Clustering method groups thousands of genes by the similarities of expression levels so that it helps to analyze gene expression profiles [2]. It can be also subdivided into two groups: the hard clustering and fuzzy clustering methods. The fuzzy clustering method, which assigns one sample to multiple clusters at the same time, is appropriate for analyzing genes since a single gene of gene expression profiles may have multiple functions in many cases [3].

Normally, clustering algorithms have common problems that they are very sensitive to initial values and they can be trapped into local optima because the processes are supposed to minimize objective function values [4, 5]. In this paper, we propose an evolutionary fuzzy clustering method. Genetic algorithm is used for the evolutionary fuzzy clustering method. It has been applied to many optimization problems and shown to be good to find optimal solutions [6]. Clustering process gets to be less subject to initial values and closer to the optimal solution using genetic algorithm [7]. There are many publications that are related to evolutionary computation for clustering data. Maulik tried to minimize the distances between the data in the same clusters and their cluster centers [5, 8], and there were studies of genetic algorithm to minimize objective function value of hard and fuzzy c-means algorithms [4, 9]. However, they fixed the number of clusters and used genetic algorithm only for the minimization of objective function, and they did not compare several cluster partitions at the same time.

On the other hand, it is also important for cluster analyses how many clusters are actually in the dataset and how good they are [10]. This evaluation called cluster validity has been used for fitness evaluation of evolutionary clustering. Conventional cluster validity measures cannot fully represent the structure of the dataset since they are based on the distance between the clusters [11, 12].

In order to solve the problems mentioned above, we propose an evolutionary fuzzy clustering method. For evaluation of the evolution process, Bayesian validation method is used. We have conducted experiments with the well-known gene expression datasets (SRBCT and Saccharomyces datasets). Finally, analysis of Saccharomyces cell-cycle expression data follows to show the usefulness of the proposed method.

The rest of the paper is organized as follows. In section 2, description of DNA microarray is introduced. Section 3 describes the proposed method, and experimental results and their analyses are presented in section 4. Section 5 concludes the paper.

II. DNA MICROARRAY

Due to microarray technology, we can measure expression levels of thousands of genes by single experiment. They consist of spatially ordered probes of cDNA or oligonucleotides on a chip. In this paper, two cDNA microarrays, Saccharomyces cell-cycle and SRBCT datasets, are used for experiments.

The first step for cDNA microarray experiment is RNA extraction from a tissue sample and RNA amplification. The RNA is reversely transcribed to cDNA labels using different fluorescent dyes mixed (red-fluorescent dye Cy5 and green-fluorescent dye Cy3). Due to the complementarity of base-pair, the cDNA binds to specific oligonucleotides on the array, and the dye is excited by a laser so that the amount of cDNA can be quantified by measuring the fluorescence intensities [13, 14]. The log ratio of two intensities of each dye is used as the gene expression profiles.

$$gene\_expression = \log_2 \frac{Int(Cy5)}{Int(Cy3)} \qquad (1)$$

Here, Int (Cy5) and Int (Cy3) are the intensities of red and green colors.

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometric Engineering Research Center (BERC) at Yonsei University.

These processes are repeated for every sample. After all processes are finished, the data are incorporated into one table of the gene expression matrix.

III. Evolutionary Fuzzy Clustering Algorithm

We propose an evolutionary fuzzy clustering method, which searches optimal cluster partition using the fuzzy c-means algorithm with GA and evaluates the fitness with Bayesian validation method. Fig. 1 illustrates the overall algorithm of the proposed method.

Evolutionary Fuzzy Clustering Algorithm

Input:

Gene expression data information, *DataInfo*.

Output:

Optimal fuzzy partition information including cluster number and centers, *OptimalClusterPartition*.

Function description:

Initialize (*DataInfo*):

A function that initializes the cluster centers and membership matrix using data information.

FuzzyClustering ():

A function that performs fuzzy clustering using fuzzy c-means algorithm and returns its result.

BayesianValidation (*FuzzyClusterResult*, *DataInfo*):

A function that evaluates fuzzy cluster result with Bayesian validation and returns the validation result, *ValResult*. It requires data information and the results play a role of the fitness for an evolution process.

EvolutionProcess (*ValResult*):

A function that performs genetic operations such as selection, crossover and mutation.

```
{
  Initialize (DataInfo);
  for (counter=0; counter<maxIteration; counter++) {
    FuzzyClusterResult = FuzzyClustering ();
    ValResult =BayesianValidation (FuzzyClusterResult, DataInfo);
    EvolutionProcess (ValidationResult);
  }
  return OptimalClusterPartition;
}
```

Fig. 1. Evolutionary fuzzy clustering algorithm

A. Representation and initialization

Generally, binary representation is used for chromosome representation since it is easy to implement and apply. This paper, however, has used floating point representation to represent a set of cluster centers of cluster partition. One cluster partition consists of K clusters, and a chromosome is represented in a space of $N \times K$ in case that the dimension of each center is N.

This paper evaluates cluster partition with various numbers of clusters, so variable length chromosome has been used. As Fig. 2 illustrates, several chromosomes are in one cluster partition, and each chromosome has different number of clusters and different values of cluster centers.

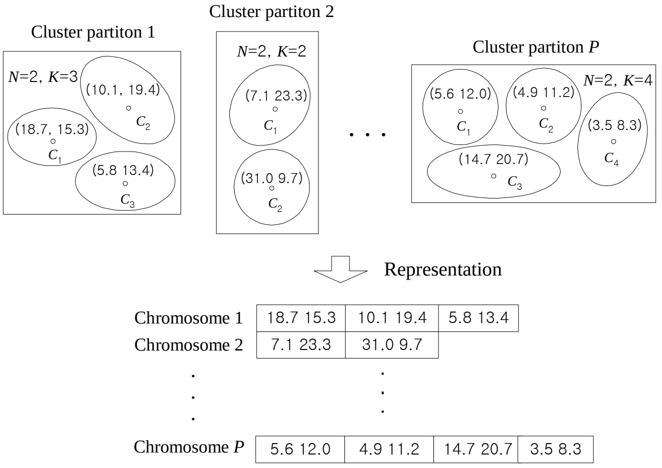


Fig. 2. Variable length chromosome representation

Population is initialized at random. For a chromosome that contains K clusters, K random samples are extracted from data, and they are used for cluster centers. This is repeated as the number of chromosomes. When clustering a specific dataset, numbers less than the square value of the number of samples are used [10]. The minimum number of clusters is set as 2.

B. Fuzzy Clustering

Fuzzy c-means algorithm proposed by Bezdek has been used for fuzzy clustering. It is the most widely used fuzzy clustering method. Given dataset, $X = \{x_1, x_2, ..., x_n\}$, and the central vector of fuzzy clustering, $V = \{v_1, v_2, ..., v_c\}$, an objective function is defined with the membership degree between each data x_j and cluster center v_i.

$$J_m(X,U,V) = \sum_{j=1}^{n}\sum_{i=1}^{c}(\mu_{ij})^m d^2(x_j, v_i) \tag{2}$$

Here, μ_{ij} is the membership degree of x_j and the ith cluster, an element of the membership matrix $U = [\mu_{ij}]$. $d^2(,)$ is the square of the Euclidean distance, and m is the fuzziness parameter, which means the degree of fuzziness of each datum's membership degree that should be bigger than 1.0 [3, 15]. When it is set as 1.0, the algorithm comes to be the same as hard c-means algorithm.

The process below is one of the fuzzy c-means algorithm.

- Step 1: Set c, the number of clusters, and m, the fuzziness parameter.
- Step 2: Initialize μ_{ij} to satisfy Eq. (3).

$$\sum_{i=1}^{c}\mu_{ij} = 1,\ 1 \le j \le n \tag{3}$$

- Step 3: Compute v_i, each center of all clusters. (i=1, 2, ..., c)

$$v_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{m} x_j}{\sum_{j=1}^{n} \mu_{ij}^{m}} \tag{4}$$

- Step 4: Compute the membership matrix *U*.

$$\mu_{ij} = \frac{\left(\frac{1}{d^2(x_j, v_i)}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{c}\left(\frac{1}{d^2(x_j, v_k)}\right)^{\frac{1}{m-1}}} \tag{5}$$

- Step 5: Repeat steps 3 and 4 until Eq. (5) is satisfied. *l* is the iteration step.

$$|\{J_m^{(l)} - J_m^{(l-1)}\}| \le \varepsilon \tag{6}$$

C. Fitness evaluation with Bayesian validation

After performing fuzzy c-means algorithm, the results have been evaluated with Bayesian validation. Bayesian validation method is a probability-based approach. Given dataset, it calculates the posterior probability of cluster partitions and selects the partition of the maximum probability to evaluate them [16].

$$\max P(Cluster \mid Dataset) \tag{7}$$

Applying Bayes' theorem, the posterior probability is calculated as follows.

$$P(Cluster \mid Dataset) = \frac{P(Cluster)P(Dataset \mid Cluster)}{P(Dataset)} \tag{8}$$

When data set *X* satisfies the condition *X*={*x*₁, *x*₂, … , *x*ₙ}, Eq. (8) is represented as Eq. (9) by multiplication rule and independence rule if each x_i is independent on one another.

$$\begin{aligned} P(Cluster \mid Dataset) &= P(Cluster \mid x_1, x_2, ..., x_N) \\ &= P(Cluster \mid x_1) \times P(Cluster \mid x_2) \times ... \times P(Cluster \mid x_N) \end{aligned} \tag{9}$$

By means of this process, Bayesian score (BS) is defined as the sum of all *P*(*Cluster*|*Dataset*) such as Eq. (10). The higher the Bayesian score is, the better the cluster partition is since it means higher posterior probability.

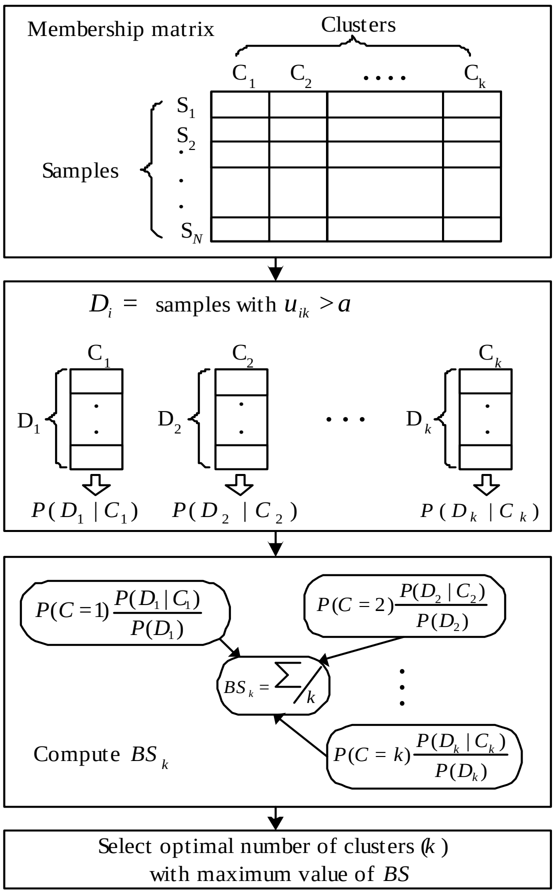
$$\begin{aligned} BS &= \frac{\sum_{i=1}^{c} P(C_i \mid X_i)}{C} = \frac{\sum_{i=1}^{c} P(C_i \mid x_{i1}, x_{i2}, ..., x_{iN})}{C} = \frac{\sum_{i=1}^{c} P(C_i \mid x_{i1}) P(C_i \mid x_{i2}) ... P(C_i \mid x_{iN})}{C} \\ &= \frac{\sum_{i=1}^{c} \prod_{j=1}^{N_i} P(C_i) P(x_{ij} \mid C_i) / P(x_{ij})}{C}, \qquad X_i = \{x_{ij} \mid \mu_{ij} > \alpha,\ 1 \le j \le n\},\ N_i = \mathrm{n}(X_i) \end{aligned} \tag{10}$$

In Eq. (10), n(X_i) is the number of X_i, and we select the samples that have larger degree of membership values than a certain probability value because Bayesian score computation includes multiplication and it produces wrong value if one of those membership degrees is zero. Besides, data of higher membership degrees are more correct and informative. α-cut plays a role of the threshold. Each probability is obtained as follows.

$$P(C_i) = \frac{\sum_{j=1, \mu_{ij} > \alpha}^{n} \mu_{ij}}{\sum_{i=1}^{c} \sum_{j=1}^{n} \mu_{ij}} \tag{11}$$

$$P(x_{ij}) = \sum_{i=1}^{c} P(C_i) P(x_{ij}) = \sum_{i=1}^{c} P(C_i) \mu_{ij} \tag{12}$$

When the membership matrix is produced as the fuzzy cluster result, each degree of membership means the probability that each sample belongs to each cluster. Therefore, the membership degree of each sample, U_{ij}, can be represented as $P(x_{ij}|C_i)$. Fig. 3 provides the overall process of Bayesian validation method where X_1 is the set of samples that belong to c_1 and satisfy the condition of μ_{ij} >α. Cluster results are evaluated using the final Bayesian score (BS) value.



Membership matrix
Clusters
C_1
C_2
....
C_k
Samples
S_1
S_2
S_N
D_i = samples with $u_{ik} > a$
C_1
C_2
C_k
D_1
D_2
D_k
$P(D_1 \mid C_1)$
$P(D_2 \mid C_2)$
$P(D_k \mid C_k)$
$P(C=1)\frac{P(D_1 \mid C_1)}{P(D_1)}$
$P(C=2)\frac{P(D_2 \mid C_2)}{P(D_2)}$
$BS_k = \sum / k$
Compute BS_k
$P(C=k)\frac{P(D_k \mid C_k)}{P(D_k)}$
Select optimal number of clusters (*k*)
with maximum value of *BS*

Fig. 3. Overall process of Bayesian validation

The algorithm of Bayesian validation method is as follow:

- Step 1: Compute the membership matrix U_{ij}.
- Step 2: Construct X_i by selecting samples (μ_{ij} >α) in each cluster.
- Step 3: Compute $P(X_j|C_j)$, $P(X_j)$, and $P(C_j)$ of X_i.
- Step 4: Compute Bayesian Score (BS) using the calculated values in step 2.
- Step 5: Evaluate the fuzzy partition with the maximum value of BS as an optimal partition.

Selection is done with these evaluation results. For selection, we have used a roulette wheel strategy that tries to select many copies of individuals corresponding to its fitness [17].

D. Crossover and mutation

This paper cannot use general crossover operation because the length of chromosome is variable, so crossover operation is performed as shown in Fig. 4. After deciding the crossover point, the length of one part is fixed for crossover, and the other part of the chromosomes is crossed over. In case of the chromosome of length *l*, crossover point is decided randomly in [1, *l*-1] with the fixed crossover rate.

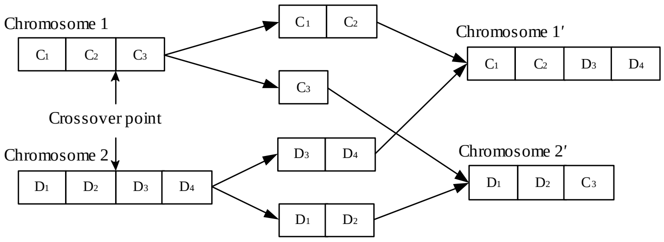


Fig. 4. An example of crossover operation

Mutation is occurred by the fixed mutation rate. Since this paper adopts the floating point representation, mutation is occurred by Eq. (13) and Eq. (14). When δ is a variable of uniform distribution in [0, 1] and v is a value of mutation point, a value of new v is determined as Eq. (13) and Eq. (14) [5].

$$v \pm 2\times\delta\times v,\ v\neq 0 \tag{13}$$

$$v \pm 2\times\delta,\ v = 0 \tag{14}$$

Eq. (14) is used if v is zero, otherwise Eq. (13) is used. The probabilities of sign '+' and '-' are the same.

IV. Experiments

In this paper, thorough experiments have been conducted. First, the experiments for an optimal cluster partition search have been conducted using evolutionary clustering method. After that, the performance comparisons of the original fuzzy c-means and evolutionary clustering algorithms have been performed with SRBCT and Saccharomyces datasets. Finally, the analyses of Saccharomyces cell-cycle gene expression data have been conducted.

A. Experimental environment

1) Experimental data

The description of SRBCT and Saccharomyces cell-cycle expression datasets are as follows.

- SRBCT dataset: This has 63 samples with 6567 genes and consists of 4 classes, NB (neuroblastoma), RMS (rhabdomhosarcoma), NHL (non-Hodgkin lymphoma) and EWS (Ewing family of tumors). They are kinds of cancer, and each of them has different characteristics. This paper has used 96 genes that are known as informative ones [18] to apply the proposed method to 63 samples.
- Saccharomyces cell-cycle dataset: This is a dataset that has expression levels of 6000 genes expressed during 2 cell-cycles. Expression levels are measured on 17 different time points every 10 minutes. This dataset is frequently used for genetic analysis since the genes classified by their biological function have different expression levels according to cycle. 421 genes that show significant change of expression levels are used in this paper [19].

2) Parameters and settings

For Bayesian validation method, the α-cut value of 0.2 and 0.4 are used for SRBCT and Saccharomyces cell-cycle datasets considering prior knowledge of those datasets.

For evolutionary clustering, maximum generation number is set as 1000, and population sizes of 100 and 200 are used for SRBCT and Saccharomyces cell-cycle datasets, respectively. The size of SRBCT dataset is smaller than Saccharomyces cell-cycle dataset. Maximum numbers of clusters are 8 and 20 for SRBCT and Saccharomyces cell-cycle datasets, respectively. Crossover rate of 0.8 and mutation rate of 0.01 are used. The fuzziness parameter of the fuzzy c-means algorithm is set as 1.2 according to Dembele's work [20].

B. Results and analyses

1) Optimal Cluster Partition Search

Fig. 5 and Fig. 6 demonstrate average fitness transition graphs. Fig. 5 illustrates average fitness transition of SRBCT dataset as generation grows. Experiments have been repeated 10 times, and bold line is the average of them. It evolves rapidly until the generation number is close to 20 and converges later. The convergence value of SRBCT dataset is about 0.6. Fig. 6 illustrates the average fitness transition of Saccharomyces cell-cycle dataset. It converges more slowly than SRBCT dataset and average fitness changes slowly until the generation

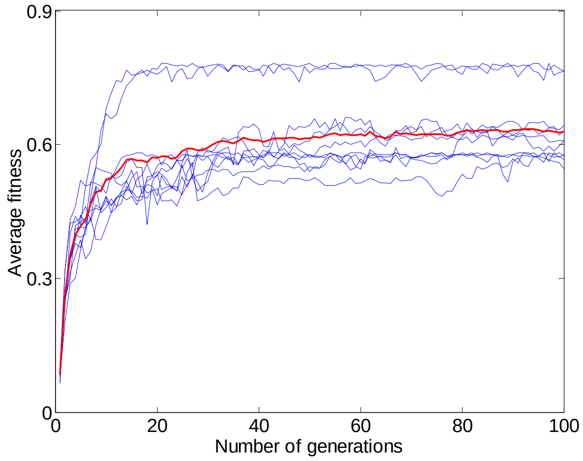


Fig. 5. Average fitness transition of SRBCT dataset

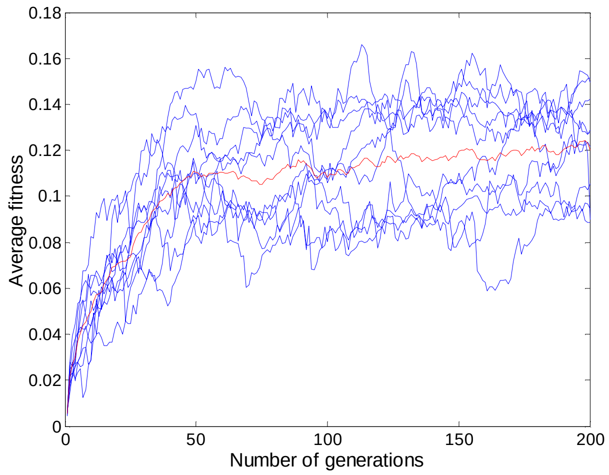


Fig. 6. Average fitness transition of Saccharomyces dataset

number is close to 80. It converges to 0.12 with large oscillation.

Fig. 5 and Fig. 6 show different transition patterns, and this can be thought that different characteristics of the datasets influenced the evolution process. According to the fuzzy cluster result, most genes of SRBCT dataset have the membership degrees that are larger than 0.9 or smaller than 0.1. Saccharomyces cell-cycle dataset, on the other hand, have various range of membership degrees.

2) Comparison with the original fuzzy c-means algorithm

In the previous section, we confirmed that the result of the FCM with GA evolves well. Here, we compare it with the original FCM by means of Bayesian score (BS) and the objective function value (OF value) of the FCM. Table 1 shows 10 experimental results of SRBCT dataset. If BS is high and the objective function value is low, it means that clustering is performed well since the objective function value is based on the distances between cluster centers and samples as mentioned in Eq. (2). The proposed method shows better results than original FCM in both BS and OF value.

Table 1. Comparison of Original FCM and the proposed method (SRBCT dataset)

	Original FCM		Evolutionary FCM	
Count	**BS**	**OF value**	**BS**	**OF value**
1	0.58028	156.9920	0.58042	156.9918
2	0.58034	156.9925	0.58036	156.9920
3	0.58031	156.9921	0.58041	156.9919
4	0.58029	156.9922	0.58036	156.9920
5	0.58041	156.9926	0.58036	156.9920
6	0.58034	156.9921	0.58041	156.9919
7	0.58031	156.9920	0.58042	156.9918
8	0.58028	156.9922	0.58042	156.9918
9	0.58033	156.9922	0.58041	156.9919
10	0.58036	156.9920	0.58042	156.9918
Average	0.58033	156.9922	0.58040	156.9919

Table 2 represents 10 experimental results of Saccharomyces cell-cycle dataset. In case of SRBCT dataset, though the result of the proposed method was better, the difference was not significant. The result of Saccharomyces cell-cycle dataset, however, shows relatively significant difference.

Table 2. Comparison of Original FCM and the proposed method (Saccharomyces cell-cycle dataset)

	Original FCM		Evolutionary FCM	
Count	**BS**	**OF value**	**BS**	**OF value**
1	0.03354	164.472	0.13256	166.883
2	0.00875	163.670	0.11246	161.542
3	0.03238	165.057	0.12661	162.911
4	0.03825	162.653	0.08058	162.073
5	0.02165	163.758	0.10667	162.798
6	0.04096	164.086	0.09778	162.312
7	0.02806	163.052	0.11873	162.042
8	0.04473	164.877	0.13659	162.773
9	0.02478	162.452	0.12898	162.905
10	0.04645	169.216	0.11246	161.542
Average	0.03195	164.329	0.11534	162.778

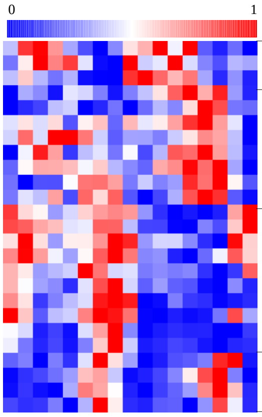
Comparing the proposed method with the original FCM, we have confirmed that the result of the proposed method is closer to the optimal solution than the original FCM

3) Analyses of Saccaromyces cell-cycle dataset

We have compared and analyzed the result of Saccharomyces cell-cycle dataset with the known genes at Cho's work [19]. In particular, we have focused on fuzzy genes, which have membership degrees higher than 0.3 and belong to several clusters simultaneously. Table 3 shows the membership degrees and cluster numbers of fuzzy genes. The number in parenthesis means the cluster number.

Table 3. Fuzzy genes and their degrees of membership (Saccharomyces cell-cycle dataset)

Fuzzy gene	**First cluster**	**Second cluster**
YBL032w	0.35035 (7)	0.33226 (4)
YHR031C	0.40455 (4)	0.38120 (7)
YCL063w	0.40413 (7)	0.39001 (11)
YBR007c	0.52122 (5)	0.39115 (15)
YER019w	0.43167 (5)	0.32937 (15)
YDR297w	0.62344 (5)	0.31825 (13)
YER118c	0.60490 (5)	0.33987 (13)
YHR173C	0.39546 (13)	0.38228 (5)
YLL021w	0.66923 (5)	0.31998 (13)
YBR275c	0.59041 (5)	0.37740 (12)
YJL173C	0.43414 (5)	0.41046 (12)
YBR053c	0.45555 (0)	0.44679 (1)
YKL163W	0.46230 (0)	0.36860 (1)
YLL040c	0.44400 (0)	0.34000 (1)
YML110C	0.59168 (1)	0.32380 (0)
YDL119c	0.48350 (0)	0.38849 (2)
YBR158w	0.58690 (1)	0.40988 (2)
YDL179w	0.55259 (1)	0.43413 (2)
YIL009W	0.60611 (1)	0.35258 (2)
YNL046W	0.55810 (2)	0.43928 (1)
YOR264W	0.69030 (1)	0.30635 (2)
YDL127w	0.52180 (12)	0.44541 (10)
YJL187C	0.56953 (12)	0.33059 (10)
YMR078C	0.58270 (12)	0.41445 (10)
YMR179W	0.56833 (10)	0.40397 (12)



Gene description	**Clusters**
weak similarity to hnRNP complex protein homolog YBR233w	4, 7
similarity to Pif1p	4, 7
weak similarity to yeast translation regulator Gcd6p	7, 11
hypothetical protein	5, 15
hypothetical protein	5, 15
suppressor of rvs161 and rvs167 mutations	5, 13
involved in the HOG1 high-osmolarity signal transduction	5, 13
hypothetical protein	5, 13
forms schmoos poorly	5, 13
Rap1p interacting factor. telomere silencing	5, 12
DNA replication factor A, 13 kDa subunit	5, 12
similarity to rat regucalcin	0, 1
protein with internal repeats	0, 1
involved in regulating membrane traffic	0, 1
ubiquinone biosynthesis, methyltransferase	0, 1
similarity to bovine Graves disease carrier protein	0, 2
hypothetical protein	1, 2
cyclin like protein interacting with Pho85p	1, 2
acyl CoA synthase	1, 2
hypothetical protein	1, 2
hypothetical protein	1, 2
cyclin, G1/S-specific	10, 12
phosporylates Cdc28, SPB separation, nuclear division	10, 12
for chromosome transmission in mitosis and normal telomere	10, 12
required for normal transcription at a number of loci	10, 12

Fig. 7. Expression level, gene description and cluster number of fuzzy genes searched by the proposed method (Saccharomyces cell-cycle dataset)

Fig. 7 illustrates the gene description and cluster number of fuzzy genes in Table 3. We have divided them into 4 groups. First three genes, YBL032w, YHR031C and YCL063w, are members of cluster 4, cluster 7 and cluster 11, respectively, and these are one group. Another group of cluster 5, cluster 12, cluster 13 and cluster 15 is grouped on cluster 5. Group of cluster 0, cluster 1 and cluster 2 and group of cluster 10 and cluster 12 are the remaining two groups. Observing expression levels on the left figure, it is easily confirmed that patterns are distinguished by group.

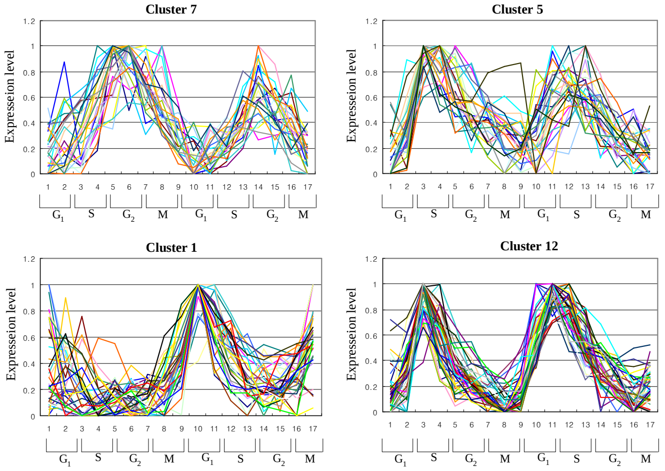


Fig. 8. Change of expression level by the time point (Saccharomyces cell-cycle dataset)

Fig. 8 illustrates the transitions of expression levels of 4 clusters, which are selected by model of each group, according to cluster numbers. Cluster 7 has 26 genes, and they express the most highly at G_2 phase of cell-cycle, so cluster 7 is thought to be related to G_2 phase. In case of cluster 5, most genes express with high level in S phase, and that time point is a little earlier than genes of cluster 7. Cluster 1 of the third group expresses the most highly at G_1 phase, and cluster 12 of the last group expresses the most between G_1 and S phases. Considering that genes of cluster 12 were grouped with cluster 5 in Fig. 8, it is also related to the second group.

V. Conclusion

This paper has proposed an evolutionary clustering method to search optimal cluster partition, and have utilized Bayesian validation method for fitness evaluation effectively. Applying the proposed method to SRBCT and Saccharomyces cell-cycle datasets, the results have evolved well and shown the better performance than the one of conventional methods. Finally, we have analyzed the optimal cluster partition of Saccharomyces cell-cycle expression data searched by the proposed method. Future research will include the experiments on various datasets since we have performed experiments with only two datasets.

References

[1] N. Belacel, P. Hansen, F. Malucelli, “Fuzzy j-means: a new heuristic for fuzzy clustering,” *Pattern Recognition*, vol. 35, pp. 2193-2200, 2002.

[2] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack and A. J. Levine, “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays,” *Proc. Natl. Acad. Sci. USA*, vol. 96, pp. 6745-6750, 1999.

[3] N. Bolshakova and F. Azuaje, “Cluster validation techniques for genome expression data,” *SIGPRO*, vol. 21, no. 82, pp. 1-9, 2002.

[4] L. O. Hall, I. B. Ozyurt, and J. C. Bezdek, “Clustering with a genetically optimized approach,” *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 2, pp. 103-112, 1999.

[5] U. Maulik and S. Bandyopadhyay, “Genetic algorithm-based clustering technique,” *Pattern Recognition*, vol. 33, pp. 1455-1465, 2000.

[6] L. Chamber, *Practical Handbook of Genetic Algorithm*, CRC Press, 1995.

[7] J. N. Bhuyan, V. V. Raghavan, and K. E. Venkatesh, “Genetic algorithm for clustering with an ordered representation,” *in Proc. 4th Int. Conf. Genetic Algorithms*, pp. 408-415, 1991.

[8] S. Bandyopadhyay and U. Maulik, “An evolutionary technique based on k-means algorithm for optimal clustering in R^n,” *Information Sciences*, vol. 146, pp. 221-237, 2002.

[9] A. D. Nola., V. Loia, and A. Staiano, “An evolutionary approach to spatial fuzzy c-means clustering,” *Fuzzy Optimization and Decision Making*, vol. 1, pp. 195-219, 2002.

[10] D. W. Kim, K. H. Lee, and D. Lee, “Fuzzy cluster validation index based on inter-cluster proximity,” *Pattern Recognition Letters*, vol. 24, pp. 2561-2574, 2003.

[11] N. R. Pal and J. C. Bezdek, “On cluster validity for the fuzzy c-means model,” *IEEE Trans. Fuzzy Systems*, vol. 3, no. 3, pp. 370-379, 1995.

[12] R. Rezaee, B. P. F. Lelieveldt, and J. H. C. Reiber, “A new cluster validity index for the fuzzy c-means,” *Pattern Recognition Letters*, vol. 19, pp. 237-246, 1998.

[13] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, “Cluster analysis and display of genome-wide expression patterns,” *Proc. Natl. Acad. Sci. USA*, 95, pp. 14863-14868, 1998.

[14] M. E. Futschik, A. Reeve and N. Kasabov, “Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue,” *Artificial Intelligence in Medicine*, vol. 28, pp. 165-189, 2003.

[15] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis*, Wiley, pp. 43-39, 1999.

[16] S.-B. Cho and S.-H. Yoo, “Fuzzy Bayesian validation for cluster analysis of yeast cell-cycle data,” *Pattern Recognition*, 2006. (In press)

[17] K. Krishna and M. N. Murty, “Genetic k-means algorithm,” *IEEE Trans. on Systems, Man and Cybernetics*, vol. 20, no. 3, June, 1999.

[18] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, “Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks,” *Nature*, vol. 7, no. 6, pp. 673-679, 2001.

[19] R. J. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis, “A genome-wide transcriptional analysis of the mitotic cell-cycle,” *Molecular Cell*, vol. 2, pp. 65-73, 1998.

[20] D. Dembele, and P. Kastner, “Fuzzy c-means method for clustering microarray data,” *Bioinformatics*, vol. 19, no. 8, pp. 973-980, 2003.