

A genetic algorithm-based approach for clustering gene expression data

Patrick C.H. Ma^{*}, Keith C.C. Chan

Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, China.

Abstract - The combined interpretation of gene expression data and gene sequences offers a valuable approach to investigate the intricate relationships involving gene transcriptional regulation. The highly interactive expression data produced by microarray hybridization experiments allow us to find coexpressed genes. By analyzing the upstream regions of the identified coexpressed genes, we can discover the regulatory patterns characterized by transcription factor binding sites, which govern the process of transcriptional regulation. This paper presents a generic clustering algorithm that uses a GA approach to discover clusters in gene expression data. The advantage of this method is that large search space can be effectively explored by utilizing the evolutionary algorithm techniques. Moreover, it is able to discover underlying patterns in noisy gene expression data for meaningful data groupings, and also statistically significant patterns hidden in each cluster can be extracted at the same time. Since the proposed method can handle both continuous- and discrete-valued data, it can be used with different microarray and biomedical data. To test its effectiveness, we have used it on real expression data. The experimental results reveal meaningful groupings and uncover many known transcription factor binding sites.

Keywords: Genetic algorithms; Cluster analysis; Data mining; Gene expression data analysis; Regulatory motifs

1. Introduction

One way to prohibit undesirable gene expression (for example, abnormally expressed in cancerous tissue) is to prevent the transcription process from taking place by locating the transcription factor binding sites in the upstream region of a gene. If such binding sites can be identified, we can then bind the corresponding repressors to these sites in order to prevent them from being activated [1]. To do so, we can identify “coexpressed genes” by clustering gene expression data [2]-[6]. Since coexpressed genes have similar transcriptional responses to the external environment, they may be regulated by the same transcription factors and share similar transcription factor binding sites. By applying motif discovery algorithm on their upstream regions, these binding sites may be revealed.

In clustering gene expression data, both “technical” and “biological” noise needs to be overcome [7]. Technical noise can be introduced at a number of different stages, such

^{*} Corresponding author.

E-mail address: cschma@comp.polyu.edu.hk

as production of the DNA array, preparation of the samples, hybridization between cDNA and array, and signal analysis and extraction of the hybridization results. The "biological" noise can come from non-uniform genetic background of the samples being compared, or from the impurity of tissue samples. Due to the unique combination of the two noise sources in gene expression data, the use of a standard clustering algorithm [8]-[13] may not always be effective as hidden patterns in the noisy data may be overlooked. Moreover, clustering is the search for those partitions that reflect the true structure of dataset, but the number of possible ways of sorting n objects into k groups is extremely large. It is impractical for an algorithm to exhaustively search the solution space to find the optimal solution. Therefore, many conventional clustering algorithms are heuristically motivated, and there is no guarantee that the solution found will be optimal. Clearly, we need an algorithm with the potential to search large solution space effectively.

To overcome such problem, we present here a generic clustering algorithm that uses a GA approach to discover clusters in gene expression data. Since genetic algorithms (GAs) are stochastic optimization algorithms based on the mechanism of natural selection and natural genetics. They have been applied to many function optimization problems and are shown to good in finding optimal and near optimal solutions. Their robustness of search in large search space and their domain independent nature motivated their applications in various fields like machine learning, pattern recognition, etc. In our study, we based on the nature of genetic algorithm and developed a novel clustering algorithm to cluster gene expression data. In the following, we present the details of the proposed method.

2. A GA Clustering Algorithm

Given gene expression data generated by N genes, $g_1, \dots, g_i, \dots, g_N$, from a series of M experiments under various conditions. The data can be represented as an $N \times M$ gene expression matrix, G so that:

$$G = \begin{matrix} & \begin{matrix} g_1 & g_2 & \vdots & g_N \end{matrix} \\ \begin{matrix} e_{11} & \vdots & e_{1j} & \vdots & e_{1M} \\ \vdots & e_{ij} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{N1} & \vdots & \vdots & e_{NM} \end{matrix} & \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \end{matrix}$$

A row, g_i , in G is therefore an M -element expression vector for the gene, and the $(i, j)^{th}$ entry in G , which is represented as e_{ij} , represents the expression value of g_i , under the j^{th} experimental condition. Given this representation, the proposed GA clustering algorithm is described below.

GA [14]-[19] is a probabilistic search approach that is founded on the ideas of evolutionary processes. The GA procedure is based on the principle of survival of the fittest. An initial population is created containing a predefined number of individuals (or

solutions), each represented by a genetic string (incorporating the variable information). Each individual has an associated fitness measure, typically representing an objective value. The concept that fittest (or best) individuals in a population will produce fitter offspring is then implemented in order to reproduce the next population. Selected individuals are chosen for reproduction at each generation, with an appropriate mutation factor to randomly modify the genes of an individual, in order to develop the new population. The result is another set of individuals based on the original subjects leading to subsequent populations with better individual fitness. Therefore, the algorithm identifies the individuals with the optimizing fitness values, and those with lower fitness will naturally get discarded from the population. Ultimately this search procedure finds a set of variables that optimizes the fitness of an individual and of the whole population. As a result, the GA technique has advantages over traditional optimization techniques that cannot always achieve an optimal solution.

In the proposed algorithm, we redesign the crossover operator, and also mutation operator for the clustering problem, and the major steps involved in our algorithm are shown in Figure 1, and each component is now described in details.

Begin

1. $t = 0$
2. initialize population $P(t)$
3. compute fitness $F(t)$
4. if termination criteria achieved, goto step 12
5. $t = t + 1$ (next generation)
6. select two individuals (or parents) from $P(t)$
7. randomly select one genetic operator with rate
 - a. 70% crossover, or
 - b. 30% mutation
8. compute the fitness of two offspring
9. offspring compete with their parents and individuals in the population based on their fitness values
10. add two survival back to $P(t)$
11. goto step 4
12. output the best individual and stop

End

Figure 1. Major steps involved in the proposed GA clustering algorithm

2.1. Individual representation and population initialization

We use the group-number encoding [20] scheme that represent a clustering of n objects (individual) as a string of n integers where the i^{th} integer signifies the group number (cluster label) of the i^{th} object. An initial population is created from individuals with each object's group number a random number between 0 and $K-1$ inclusive (where K is a maximal number of group).

2.2. Fitness function

In this study, the data mining technique (a two-step algorithm) described in [21] is used as a fitness function to evaluate the fitness of the individuals within the population. This evaluation method is similar to the internal criterion analysis described in [22], and the details of evaluating clustering results will be presented in section 3.2. Step one attempts to discover statistically significant association patterns in the training set, and in Step two, all objects in the testing set are classified based on the discovered association patterns. The details of these two steps are given below.

1. *Discovering of statistically significant association patterns*: The values of an attribute A_j with class-dependency are those that are statistically associated with certain class labels, allowing some other values of the same attribute to reflect no class information. The statistical dependency between an attribute value and a class label can be described as follows. Let o_{pk} be the total number of objects in the database that belong to class c_p and are

characterized by the attribute value a_{jk} , and let $e_{pk} = \frac{o_{p+}o_{+k}}{N'}$ (where $o_{p+} =$

$\sum_{k=1}^K o_{pk}$, $o_{+k} = \sum_{p=1}^P o_{pk}$, and $N' = \sum_{p,k} o_{pk} \leq N$ due to the missing

values) be the expected total under the assumption that being a member of c_p is independent of whether an object has the characteristic a_{jk} . The statistical significance of the association can be evaluated using the following statistical

technique. Let $z_{pk} = \frac{o_{pk} - e_{pk}}{\sqrt{e_{pk}}}$; the maximum likelihood estimate of its

asymptotic variance, v_{pk} , is then given by $v_{pk} = (1 - \frac{o_{p+}}{N'})(1 - \frac{o_{+k}}{N'})$, then

$d_{pk} = \frac{(o_{pk} - e_{pk}) / \sqrt{e_{pk}}}{\sqrt{v_{pk}}} = \frac{z_{pk}}{\sqrt{v_{pk}}}$. This statistics has an approximate

standard normal distribution and can be evaluated based on a certain confidence level. The attribute value can then be selected based on a statistically significant class-dependency.

2. *Classification and re-classification using a weight-of-evidence information measure*: The information provided by d_{pk} (in step one) with absolute values were greater than 1.96 (table value corresponding to a chosen 95% confidence level) were utilized to construct characteristic descriptions of the various classes. These descriptions are in the form of relational rules with weight such as the following: If A_j of an object is a_{jk} , then it is with certainty $W(\text{Class} =$

$c_p / \text{Class} \neq c_p | a_{jk}$) that the object belongs to c_p , where W , the weight of evidence measure, is defined in terms of the mutual information, $I(c_p : a_{jk})$, between c_p and a_{jk} , and $I(c_p : a_{jk}) = \log \frac{P(c_p | a_{jk})}{P(c_p)}$. Therefore, $W(\text{Class} = c_p / \text{Class} \neq c_p | a_{jk}) = I(c_p : a_{jk}) - I(\neq c_p : a_{jk})$. The weight of evidence measures the amount of positive or negative evidence that is provided by a_{jk} supporting or refuting the labeling of the object as c_p . Given a collection of the selected attribute values, the weight of evidence from all observed attribute values is defined as a summation of the total weights, $W(\text{Class} = c_p / \text{Class} \neq c_p | a_1, \dots, a_m) = \sum_{j=1}^m W(\text{Class} = c_p / \text{Class} \neq c_p | a_j)$. Therefore, the class label c_p is inferred if W is maximized.

2.3. Parent selection technique

Roulette wheel selection is adopted to select parents for the reproduction process. This parent selection technique is a fitness proportionate reproduction, which allocates reproductive opportunities to individuals according to their relative values of the fitness function.

2.4. Replacement

Steady-state without duplicates approach is adopted. In each generation, only two worst individuals are replaced, and the offspring that are duplicates of individuals in the population are immediately discarded. Therefore, parents and offspring may coexist in a population.

2.5. Crossover

We present here a new crossover operator to exchange the genetic information of two parents in an effective way. The idea is as follows: for example, each parent represents a clustering solution of 4 clusters, first parent F_1, F_2, F_3 and F_4 , and second parent S_1, S_2, S_3 and S_4 . Firstly, we randomly borrow one cluster, say F_1 , from the first parent. Then, we use F_1 to replace the cluster of the second parent, say S_3 , which has the largest number of same objects (highly-overlapped objects) in F_1 . After the replacement, those objects in F_1 also appear in other clusters (S_1, S_2 and S_4) are deleted in order to ensure that each object only belongs to one cluster (only keep the objects in F_1). The remaining non-overlapped objects in S_3 (compared to F_1) are reclassified (treat as testing set) into one of the clusters (S_1, S_2, F_1 , and S_4 , treat as training set) by using the two-step algorithm described in section 2.2. Finally, the new clustering solution represented by the first offspring is NS_1, NS_2, NF_1 and NS_4 (N means new, some objects may be deleted

or added in the original cluster). This crossover is repeated once by borrowing a cluster from the second parent to generate the second offspring. The advantage of this crossover operator is that it is more constructive to the entire grouping structure of the clustering than other traditional approaches (e.g. uniform crossover, and two-point crossover).

2.6. Mutation

There are two different mutation operators with equal selective rate used in our study:

- *Reclassify*: randomly select 30% objects from each cluster of an individual to form the testing set, and the remaining 70% objects form the training set. The objects in the testing set are reclassified into one of the clusters by using the two-step algorithm described in section 2.2.
- *Move*: randomly select 30% objects from each cluster of an individual, and randomly assign the cluster labels to these objects.

The advantages of using two different mutation operators are that the convergence rate of the population can be enhanced, and also the diversity of the population can be maintained. Moreover, these operators are more applicable to the clustering problem than uniform mutation.

2.7. Termination

The processes of fitness computation, parent selection, crossover, and mutation are executed until maximum number of iterations reached (for example, 500 or 1000 generations). The best individual seem at the end of execution provides the solution to the clustering problem.

3. Experimental Results

3.1. Sources of experimental data

For experimentation, we used a set of gene expression data that contains a series of gene expression measurements of the transcript (mRNA) levels of *S. cerevisiae* genes [2]. The samples were synchronized by three independent methods: α factor arrest, elutriation, and arrest of a *cdc15* temperature-sensitive mutant (additional dataset included: arrest of a *cdc28* temperature-sensitive mutant was taken from [3]). Using periodicity and correlation algorithms, a total of about 800 genes that meet an objective minimum criterion for cell cycle regulation were identified. In separate experiments, designed to examine the effects of inducing either the G1 cyclin Cln3p or the B-type cyclin Clb2p, the mRNA levels of more than half of these 800 genes were found to respond to one or both of these cyclins. The expression data we used is available at [23] and the corresponding upstream sequences are available at the *Saccharomyces* genome database [24].

According to [2], authors successfully applied hierarchical clustering algorithm [8] to cluster about 250 genes (out of 800) into 8 distinct clusters (see Table 1) based on the similarity of expression profiles and prior biological knowledge. Clearly, the remaining unclassified genes also have their biological meanings, and may have similar regulatory features to the clustered genes. Therefore, in this study, we extracted two sets from this dataset to test the effectiveness of the proposed clustering algorithm. The first set (*dataset 1*) contains 227 genes that were successfully grouped into one of the 8 clusters in [2]. The second set (*dataset 2*) contains all cell cycle-regulated genes (792 genes in total, not 800, since there are 8 genes were deleted in [24]).

Table 1. Summary of the Spellman’s identified clusters [2] (total: 8 clusters, 227 genes, dataset 1)

	Cluster	No. of genes	Peak expression
C0	CLB2	32	M
C1	CLN2	57	G1
C2	Histone	9	S
C3	MAT	13	M/G1
C4	MCM	38	M/G1
C5	MET	20	S
C6	SIC1	27	M/G1
C7	Y’	31	G1
	<i>Total No.</i>	<i>227</i>	

3.2. Method of evaluating the results

To know if the proposed clustering algorithm is effective, the results are carefully evaluated according to the statistical and biological criteria.

To evaluate the results statistically, we try to assess the predictive power of the discovered clusters. We randomly remove 30% of the objects of each of the clusters for testing. Using the data mining technique as described in section 2.2, we try to discover the association patterns in each cluster. Based on these discovered patterns, the cluster membership of the test set is predicted. If the prediction accuracy is high, then the clustering technique proposed can be considered to be a very effective one as it is able to group the expression profiles according to their inherent similarities. If not, then the proposed clustering technique can be considered ineffective as grouping of expression profiles are rather random.

To determine if the patterns discovered in each cluster by the proposed technique is meaningful, the results are also evaluated according to their biological significance. To do so, we use a motif discovery algorithm described in [25] to determine if any transcription factor binding sites located in the upstream regions can be identified. The biological meaning of the discovered binding sites is then validated based on published literature.

3.3. Results

3.3.1. Statistical evaluation

Just like other popular clustering algorithms, for example K-means and SOM, we need to determine a suitable K value (K means the total number of clusters) for the proposed clustering algorithm. Therefore, we tried different value of K (from 5 to 12) and found which K value could give us the best clustering result.

To evaluate the effectiveness of the clustering algorithm in grouping similar expression profiles according to their underlying patterns, we randomly removed 30% objects from each cluster for testing. Using the remaining 70%, we tried to discover hidden association patterns in each cluster using the data mining algorithm described in section 2.2 [21]. Using the discovered patterns, the cluster memberships of the objects in the testing set were predicted. The prediction accuracy was then computed. By repeating this 10 times with different randomly selected training and testing set, the average prediction accuracy was calculated. In addition, we also compared the average prediction accuracy of the proposed clustering algorithm with standard GA (uniform crossover with crossover rate 0.8, and uniform mutation with mutation rate 0.01), K-means, and SOM (K-means and SOM software is available at [26]). The result of comparison is showed in Figure 2 (*dataset 1*) and Figure 3 (*dataset 2*).

(Note: according to [2], authors successfully applied hierarchical clustering algorithm [8] to cluster *dataset 1* into 8 clusters - see Table 1, based on the similarity of expression profiles and prior biological knowledge. Clearly, we can't perform the same comparison (from K = 5 to 12) with hierarchical clustering. Therefore, we only evaluated the solution (K = 8) proposed in [2], and found that the average prediction accuracy is 0.78.)

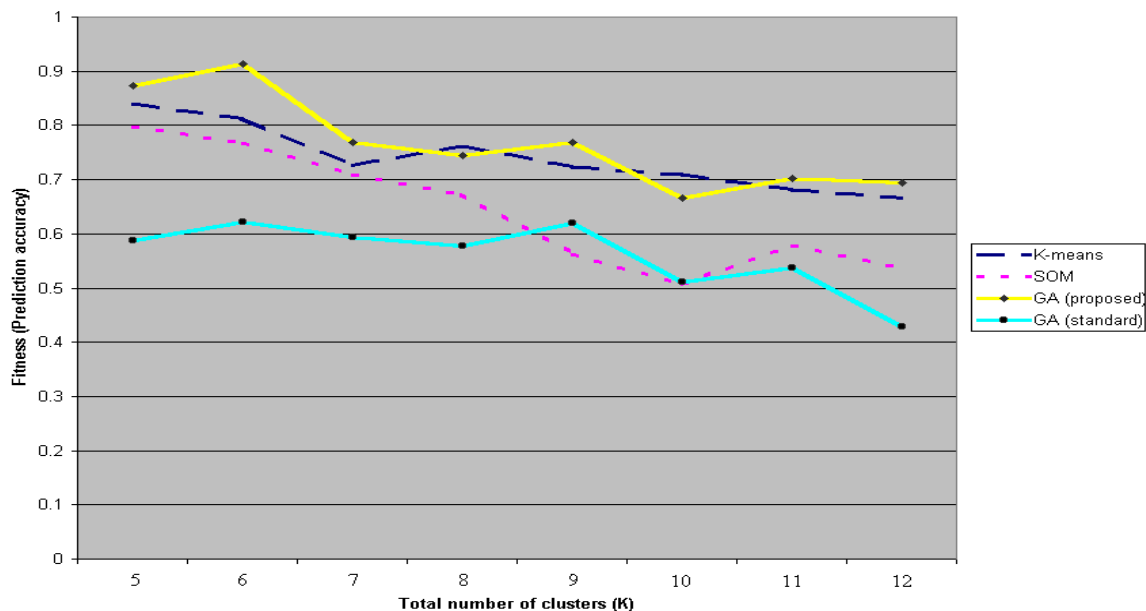


Figure 2. The comparison of the proposed algorithm with K-means, SOM and standard GA (y-axis is the average prediction accuracy, and x-axis is the total no. of clusters. Dataset 1: 227 genes)

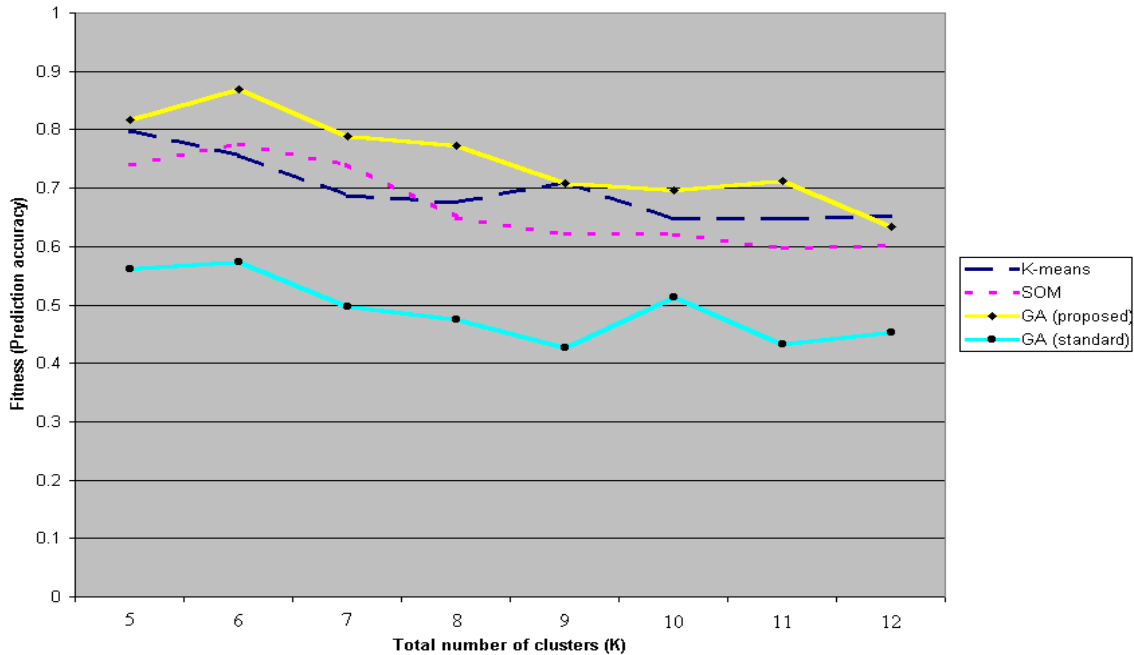


Figure 3. The comparison of the proposed algorithm with K-means, SOM and standard GA (y-axis is the average prediction accuracy, and x-axis is the total no. of clusters. Dataset 2: 792 genes)

According to Figure 2 and Figure 3, we found that the proposed algorithm is not only much better than standard GA, it is also better than the popular clustering techniques of K-means and SOM in most of the K values. This indicates that the proposed algorithm is able to identify the patterns within each cluster. Moreover, we found that the highest average prediction accuracy in both dataset were obtained when K = 6.

In *dataset 1*, we also compared the clusters of the best clustering (K = 6, average prediction accuracy is 0.92) obtained from the proposed algorithm with the Spellman's identified clusters (using hierarchical clustering, average prediction accuracy is 0.78, in Table 1). The summary of clusters comparison is shown in Table 2.

Table 2. Summary of the discovered clusters in dataset 1 (227 genes)

Cluster	Summary (compared to Spellman's clusters)	Peak Expression
NC0	85% genes belong to SIC1 and MAT clusters	M/G1
NC1	94% genes belong to MCM and CLB2 clusters	M and M/G1
NC2	72% genes belong to Y cluster	G1
NC3	95% genes belong to CLN2 cluster	G1
NC4	100% genes belong to Histone and CLN2 clusters	S and G1
NC5	90% genes belong to MET cluster	S

The association patterns that are statistically significant in each cluster are given in Table 3 (*dataset 1*) and Table 4 (*dataset 2*). All patterns were extracted from the best clustering solution (K=6).

Table 3. Association patterns extracted from the new clusters in dataset 1 (227 genes)
(portion, attribute is the experimental condition and value is the range of expression level)

Cluster	Attribute	Value	Cluster	Attribute	Value
NC0	cdc15_80	[-2.56, -0.15]	NC1	cdc15_140	[-2.56, -0.15]
	cdc15_250	[1.01, 3.09]		cdc28_60	[1.01, 3.09]
NC2	cdc28_30	[1.01, 3.09]	NC3	alpha21	[1.01, 3.09]
	cdc28_150	[-2.56, -0.15]		cdc28_20	[1.01, 3.09]
NC4	cdc28_20	[1.01, 3.09]	NC5	cdc15_70	[-2.56, -0.15]
	cdc28_160	[1.01, 3.09]		cdc15_270	[1.01, 3.09]

Table 4. Association patterns extracted from the new clusters in dataset 2 (792 genes)
(portion, attribute is the experimental condition and value is the range of expression level)

Cluster	Attribute	Value	Cluster	Attribute	Value
LC0	cdc15_70	[0.65, 2.84]	LC1	cdc15_100	[0.65, 2.84]
	cdc15_190	[0.65, 2.84]		cdc15_170	[-2.56, -0.3]
LC2	alpha21	[0.65, 2.84]	LC3	alpha42	[-2.56, -0.3]
	cdc28_70	[-2.56, -0.3]		cdc15_130	[0.65, 2.84]
LC4	cdc28_20	[0.65, 2.84]	LC5	alpha49	[-2.56, -0.3]
	cdc28_150	[-2.56, -0.3]		elu360	[0.65, 2.84]

3.3.2. Biological evaluation

In each cluster (the best clustering, K=6), we downloaded the 1000bp upstream sequences of all genes from the SGD [24], and applied the motif discovery algorithm described in [25] to reveal the transcription factor binding site. According to [25], most regulatory sites can be detected with the sample hexanucleotide analysis. In our analysis, we also set the oligonucleotide length to be six, and the pattern significance was set to be equal or greater than zero. All discovered sites were checked against the well-known binding sites listed in Table 5. In Table 6 (*dataset 1*) and Table 7 (*dataset 2*), we list some significant binding sites revealed in the discovered clusters.

Table 5. Known consensus for transcription factors involved in cell cycle and methionine biosynthesis [2],[27]. (Note: R is A or G; M is A or C; S is C or G; D is A, G or T; Y is C or T; W is A or T; N is any base)

Binding Site	Pattern
<i>MCB</i>	ACGCGT
<i>SCB</i>	CRMSAAA = C(A/G)(A/C)(C/G)AAA
<i>Mcm1</i>	DCCYWWNNRG = (A/G/T)CC(C/T)(A/T)(A/T)(A/G/C/T)(A/G/C/T)(A/G)G
<i>Swi5;Ace2</i>	RRCCAGCR = (A/G)(A/G)CCAGC(A/G)
<i>SFF</i>	GTMAACAW = GT(A/C)AACA(A/T)
<i>Met31;Met32</i>	AAAACGTGG
<i>Met4/Met28/Cbf1</i>	TCACGTGA

Table 6. Transcription factor binding sites revealed from the new clusters in dataset 1 (227 genes). Note: R is A or G; M is A or C; S is C or G; D is A, G or T; Y is C or T; W is A or T; N is any base, and rev. means reverse complement, sig. means significance.

Cluster	Sequence revealed	Sig.	Consensus	Binding Site
<i>NC0</i>	CCAGCA	9.18	CCAGCR	Swi5;Ace2
<i>NC1</i>	TCCTAA	2.07	DCCYWW	Mcm1
<i>NC2</i>	ACCCTA	1.50	DCCYWW	Mcm1
<i>NC3</i>	ACGCGT	13.62	ACGCGT	MCB
<i>NC4</i>	CGCGAA	1.18	CRMSAA	SCB
<i>NC5</i>	CACGTG	4.69	CACGTG	Met4/Met28/Cbf1
	CTGTGG (rev.)	4.67	CTGTGG	Met31;Met32

Table 7. Transcription factor binding sites revealed from the new clusters in dataset 2 (792 genes). Note: R is A or G; M is A or C; S is C or G; D is A, G or T; Y is C or T; W is A or T; N is any base, and rev. means reverse complement, sig. means significance.

Cluster	Sequence revealed	Sig.	Consensus	Binding Site
<i>LC0</i>	AAACAA	4.21	MAACAW	SFF
<i>LC1</i>	CCCAA	3.54	CCYWWN	Mcm1
<i>LC2</i>	CTTCAA	1.32	YWWNNR	Mcm1
<i>LC3</i>	ACGCGT	26.53	ACGCGT	MCB
	CGCGAA	13.43	CRMSAA	SCB
<i>LC4</i>	ACCCTA	1.31	DCCYWW	Mcm1
<i>LC5</i>	CCAGCA	5.03	CCAGCR	Swi5;Ace2
	CTGTGG (rev.)	4.67	CTGTGG	Met31;Met32

4. Conclusions

DNA microarray technologies are becoming increasingly important in analysis of bio-molecules. They provide information that eventually may lead to the understanding of the mechanisms that control gene regulation at the transcription level. Since huge amount of gene expression data is produced by the microarray experiments, clustering technique that combines with motif discovery can be highly effective.

In this paper, we present a generic clustering algorithm that uses a GA approach to discover clusters in gene expression data. Experiments with real expression data show that this method is able to search for the possible solutions effectively, and discover underlying patterns in noisy gene expression data for meaningful data groupings with many known transcription factor binding sites. The results also show that the proposed method is able to extract the statistically significant patterns (attribute-value pairs) hidden in each cluster at the same time. Possibly, our study can lead to further understanding of the mechanisms of gene transcriptional regulation.

References

- [1] Hartl, D.L. and Jones, E.W. Genetics: Analysis of Genes and Genomes. 5 ed. Jones & Bartlett, Sudbury, MA. 2001.

- [2] Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, 9, 3273-3297, 1998.
- [3] Cho, R.J., Campbell, M.J., Winzler, 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. *Mol. Cell*, 2, 65-73, 1998.
- [4] Cho, R.J., Huang, M., Campbell, M.J., Dong, H., Steinmetz, L., Sapinoso, L., Hampton, G., Elledge, S.J., Davis, R.W. and Lockhart, D.J. Transcriptional regulation and function during the human cell cycle. *Nature Genet.*, 27, 48-54, 2001.
- [5] DeRisi, J.L., Iyer, V.R. and Brown, P.O. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278, 680-686, 1997.
- [6] Lashkari, D.A., DeRisi, J.L., McCusker, J.H., Namath, A.F., Gentile, C., Hwang, S.Y., Brown, P.O. and Davis, R. Yeast microarrays for genome wide parallel genetic and gene expression analysis. *Proc. Natl Acad. Sci USA*, 94, 13057-13062, 1997.
- [7] Juan, L., Hitoshi, I. and Mitsuru, I. Selecting informative genes with parallel genetic algorithms in tissue classification. *Genome Informatics*, 12, 14-23, 2001.
- [8] Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, 95, 14863-14868, 1998.
- [9] Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J. and Church, G.M. Systematic determination of genetic network architecture. *Nature Genet.*, 22, 281-285, 1999.
- [10] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S. and Golub, T.R. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. of the Natl Acad. Sci. USA*, 96, 2907-2912, 1999.
- [11] Quackenbush, J. Computational analysis of microarray data. *Nat. Rev. Genet.*, 2, 418-427, 2001.
- [12] Ben-Dor, A., Shamir, R. and Yakhini, Z. Clustering gene expression patterns. *J. Comput. Biol.*, 6, 281-297, 1999.
- [13] Herrero, J., Valencia, A. and Dopazo, J. A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, 17, 126-136, 2001.
- [14] D.E. Goldberg, Genetic algorithms in search, optimization and machine learning. Addison-Wesley, New York, 1989.
- [15] L. Davis (Ed.). Handbook of genetic algorithms. Van Nostrand Reinhold, New York, 1991.

- [16] Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer, New York, 1992.
- [17] T Baeck, D Fogel, Z Michalewicz (eds). Evolutionary Computation 1: Basic Algorithms and Operators. Institute of Physics, 2000.
- [18] T Baeck, D Fogel, Z Michalewicz (eds). Evolutionary Computation 2 : Advanced Algorithms and Operators. Institute of Physics, 2000.
- [19] Falkenauer, E. Genetic Algorithms and Grouping Problems. John Wiley, 1998.
- [20] Donald R. Jones and Mark A. Beltramo. Solving partitioning problems with genetic algorithms. In R. K. Belew and L. B. Booker, editors, Proceedings of the Fourth International Conference on Genetic Algorithms, pages 442-9, Morgan Kaufman Publishers, 1991.
- [21] Chan, K.C.C. and Wong, K.C. A statistical technique for extracting classificatory knowledge from databases. In Piatetsky-Shapiro,G. and Frawley, W.J. (eds), Knowledge Discovery in Databases, AAAI Press, Cambridge, pp.107-123, 1991.
- [22] Jain, A.K. and Dubes,R.C. Algorithms for clustering data. Prentice Hall, Englewood Cliffs, NJ, 1988.
- [23] <http://genome-www.stanford.edu/cellcycle>
- [24] Ball, C.A., Jin, H., Sherlock, G., Weng, S., Matese, J.C., Andrada, R., Binkley, G., Dolinski, K., Dwight, S.S., Harris, M.A., Issel-Tarver, L., Schroeder, M., Botstein, D., Cherry, J.M. Saccharomyces genome database provides tools to survey gene expression and functional analysis data. Nucleic Acids Res., 29, 80-81, 2001.
- [25] Helden, J.V., Andre, B. and Collado-Vides, J. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. J. Mol Biol., 281, 827-842, 1998.
- [26] <http://rana.lbl.gov/EisenSoftware.htm>
- [27] Helden, J.V., Rios, A.F. and Collado-Vides, J. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. Nucleic Acids Res., 28, 1808-1818, 2000.