An Evolutionary Gene Expression Microarray Clustering Algorithm Based on Optimized Experimental Conditions

Mrinal Sen
Electronics & Communication Engineering Department
Birbhum Institute of Engineering & Technology
Suri, Birbhum, West Bengal 731101, India
mrinal.sen.ahm@gmail.com

Sheli Sinha Chaudhury, Amit Konar
Dept. of Electronics & Telecommunication Engineering
Jadavpur University
Kolkata 700032, West Bengal, India
shelism@rediffmail.com, konaramit@yahoo.co.in

R. Janarthanan
Dept. of Information Technology
Jaya Engineering College
Chennai, India
srmjana_73@yahoo.com

Abstract— **Entities of the real world require partition into groups based on even feature of each entity. Clusters are analyzed to make the groups homologous and well separated. Many algorithms have been developed to tackle clustering problems and are very much needed in our application area of gene expression profile analysis in bioinformatics. It is often difficult to group the data in the real world clearly since there is no clear boundary of clustering. Gene clustering possesses the same problem as they contain multiple functions and can belong to multiple clusters. Hence one sample is assigned to multiple clusters. A variety of clustering techniques have been applied to microarray data in bio-informatics research. We have proposed in this paper an easy to implement evolutionary clustering algorithm based on optimized number of experimental conditions for each individual cluster for which the elements of that group produced similar expression and then compared its performance with some of the previously proposed clustering algorithm for some real life data that proves its superiority compared to the others. The proposed algorithm will produce some overlapping clusters which re-imposes the fact that a gene can participate in multiple biological processes.**

Keywords-Bioinformatics; Clustering; Microarray; Genetic Algorithm; Optimization.

I. INTRODUCTION

In cluster analysis one wishes to partition entities into groups based on given feature of each entity, so that the groups are homologous and well separated. Clustering, or class discovery, is an unsupervised problem, in that there is no response measurement available for the items to be grouped. The records in a cluster behave similarly on the basis of some feature. Many algorithms have been developed to tackle clustering problems in a variety of application domains. Bioinformatics is such a domain that needs clustering with its every footprint. Cluster analysis is the far most used technique in gene expression analysis. It can be performed to identify genes that are regulated in a similar manner under a number of experimental conditions. It is often difficult to group the data in the real world clearly, since often there are no clear boundaries of clusters. Clustering genes, which contain multiple functions and belong to multiple clusters, is a representative example. Since overlapping clustering method assigns one sample (gene) to multiple clusters according to their participation in multiple (biological) processes, it is more appropriate for analyzing gene expression profiles. General clustering algorithms have common problems in that they are very sensitive to initial values and generally the number of clusters often needs to be fixed before analyses are performed. So, it is difficult to analyze the data correctly without previous knowledge. It takes much time and cost to cluster the data, if there is no prior information of the number of clusters. There is also a problem of validating cluster results. Since gene expression profiles are variable depending on experiments and environments from which they were collected, it is not proper to validate them by a single criterion.

A variety of clustering techniques have been applied to microarray data in bioinformatics research. In this paper, we propose an easy to implement overlapping evolutionary clustering algorithm based on optimized experimental conditions (ECoEC) and compare its performance with previously proposed clustering algorithm for three actual data. Our proposed algorithm will produce n overlapping clusters for n genes or record sets. So, the biological fact, that a gene can participate in multiple biological processes, is supported by our algorithm. Though, the microarray data are very noise prone and sometimes absent for some attribute, the proposed technique gives very good measure of similarity as it doesn't consider all the attributes as a whole.

A major use of microarray data is to classify genes with similar expression profiles into groups in order to investigate their biological significance. Generally, the statistical performance of a clustering algorithm degrades as the records under a cluster have a significant possibility to belong in other clusters. Our algorithm is an overlapping clustering algorithm, so that a record may belong to more than one cluster. In that case if we try to evaluate the

978-1-4244-5612-3/09/$26.00 ©2009 IEEE

performance of our algorithm in the similar manner as the non-overlapping algorithms, it may exhibit poor result. So, at the time of performance evaluation, we used another algorithm to select only those clusters that are less correlated and then evaluated the performance of our algorithm using Silhouette Index measure.

II. The Proposed Algorithm

This algorithm considers each attribute/experimental conditions individually, that is, it creates M different microarray data set, from an N gene-M attribute data set, each having n genes and only one attribute/experimental condition. Then it creates clusters for each of the m data set with respect to a particular range of value for every Gene. At the next stage the algorithm considers a subset of attribute set for each individual genes and maximizes the common (intersection) gene count and attribute count product for each individual gene. The result explores n number of overlapping clusters with different attribute set and a range of values for each attribute of those derived attribute set, which can further be used to classify a new gene that wants to participate in the clustering process without actually rerunning the clustering process and that would be a fairly good classification when the total number of previously existing genes (in the microarray data set) is good enough. At last the algorithm selects the most prominent and highly uncorrelated clusters from the resultant N clusters.

While describing the algorithm the text uses 'record' in space of 'gene' as record is a more generalized term and gene is a record of this particular application.

Consider that the Microarray, under process, has N Genes/records and at most M attributes for each Gene/record.

The algorithm uses some user defined constants as follows.

Set U_C as the highest Crossover probability.
Set L_C as the lowest Crossover probability.
Set U_M as the highest Mutation probability.
Set L_M as the lowest Mutation probability.
Set IP as the population size of the Genetic Algorithms.

Assume that RoP is a user defined constant for an application dataset which can be used to increase/decrease element count in a cluster (by decreasing/increasing it) but the resultant cluster will loss/gain precession as it used to accommodate/reject crude similarity.

Let, r_n is the n^{th} record of the record set. i.e., $r_n \in R = \{r_1, r_2, \dots, r_N\}$. $V_{n,m}$ is the value (gene expression value in that case) of the n^{th} record for m^{th} attribute. And L_m is a real value for attribute m, where $\forall n : 1 \le n \le N$, $\left|Max(V_{n,m}) - Min(V_{n,m})\right| \ge L_m$. Then the n^{th} cluster, centering r_n, for an attribute subset AS ($\subseteq \{a_1, a_2, \dots, a_M\}$) can be defined as follows.

$$C_{n,AS} = \bigcap_{\forall\, a:\, a \in AS} \{r_k | \left(\left|V_{k,a} - V_{n,a}\right| \le L_a\right)\} \quad (1)$$

Let $rcc_{n,AS}$ is the element/record count of $C_{n,AS}$ and $asen_{AS}$ is the element/record count of attribute subset AS.

Now, there may be many values of L_a at the time of optimization of L_a for each individual single attribute a. So, the cluster, centering r_n for a single attribute a ($\in \{a_1, a_2, \dots, a_M\}$) and for the i^{th} value of L_a, i.e., $L_{a,i}$, can be redefined as follows.

$$C_{n,a,i} = \{r_k | \left(\left|V_{k,a} - V_{n,a}\right| \le L_{a,i}\right)\} \quad (2)$$

And $rcc_{n,a,i}$ as the element/record count of $C_{n,a,i}$.

Let there is a set of $L_{attr} (= \{L_{attr,1}, L_{attr,2}, \dots, L_{attr,IP}\})$. Then the Standard Deviation of the set σ can be used to terminate the GA of the first stage. The Standard Deviation of the set of L_{attr} is defined as follows.

$$\sigma_A = \sqrt{\frac{\sum_{i=1}^{IP}\left(L_{attr,i} - \left(\sum_{j=1}^{IP}\frac{L_{attr,j}}{IP}\right)\right)^2}{IP}} \le 0.001 \quad (3)$$

A. Attribute Value Range Optimization

FOR each individual attribute $attr = a_1 \; to \; a_M$
BEGIN
/* Use the following Genetic Algorithm to minimize L_{attr}. */

1) *Initial population generation:* Generate a set of randomly chosen IP number of L_{attr} . Such as $\{L_{attr,1}, L_{attr,2}, \dots, L_{attr,IP}\}$.
2) *REPEAT WHILE* $\sigma_A \ge 0.001$ /* Untill the chromosome saturates */

BEGIN

a) *Reproduction:* Reproduction is done by the following two stages.
   - *Crossover:* Use single point crossover with a randomly generated probability within the range $[L_C, U_C]$.
   - *Mutation:* Use single bit mutation with a randomly generated probability within the range$[L_M, U_M]$.

b) *Fitness function:* The fitness of the i^{th} population, which is the most important concept of the algorithm, is defined as follows.

$$F_1(L_{attr,i}) = \frac{\sum_{n=1}^{N}(rec_{n,attr,i} - 1)}{2 \times (L_{attr,i})^{RoP}} \quad (4)$$

/* the equation tries to allocate maximum records in minimum attribute value range. */

c) *Selection procedure:* Use Roulette Wheel selection scheme to select the candidates of the population to participate in the next generation.

END WHILE
END FOR

At this point, the algorithm produces a set of M different optimized range of attribute value $L \in \{L_1, L_2, \dots, L_M\}$ for each M different attributes.

In the following stage, the algorithm maximizes the number of attributes in the subset of attributes for which a

cluster can accommodate the maximum number of records in a cluster. To do so, it again uses Genetic Algorithm.

Set U_F and L_F as the highest and lowest probability attribute removal/addition respectively.

Let FPOWER, CPOWER and MaxFech be three user defined constants, where the first two have been used for defining the fitness function of this GA stage and the last one is specifying the maximum number of attributes that can be allowed to form a chromosome at this stage.

In the following part of the algorithm we used chromosome saturation to terminate the GA.. To do so we need to define one chromosomal distance function as follows.

$$D_{AS_i^{rec},\, AS_j^{rec}} = asen_{\left[\left(AS_i^{rec} - AS_j^{rec}\right) \cup \left(AS_j^{rec} - AS_i^{rec}\right)\right]} \quad (5)$$

The above equation estimates the chromosomal distance of two chromosomes AS_i^{rec} and AS_j^{rec}. Here a chromosome AS_i^{rec} is a subset of attribute set $\{a_1, a_2, \dots, a_M\}$.

Now, if the Standard Deviation σ_B of the above defined chromosomal distance for the population of a generation is less than or equal to 0.001 then the GA terminates.

B. Attribute Subset Optimization

FOR each individual record rec $= r_1$ to r_N
BEGIN

1) *Initial population generation:* Generate IP number of attribute subset $AS_1^{rec}, AS_2^{rec}, \dots, AS_{IP}^{rec}$ from the attribute set $\{a_1, a_2, \dots, a_M\}$, where each subset contains at most MaxFech number of attributes. /* These attribute subsets will act as the chromosomes in that stage of GA. */
   REPEAT WHILE $\sigma_B \geq 0.001$
   BEGIN
   a) *Reproduction:* Reproduction is done by the following two stages.
      - Crossover: Chromosomes are selected for crossover using the randomly generated probability within the range $[L_C, U_C]$.
        At the time of crossover two parents are chosen from the selected list. Then some attributes of one parent chromosome are removed depending on a randomly generated probability within the range $[L_F, U_F]$. Now, some attributes are chosen from the spouse chromosome, using the randomly generated probability within the range $[L_F, U_F]$, and added with the spliced chromosome, maintaining the constraints that the resultant chromosome can not contain more than MaxFech no of attributes and no duplicate attribute should exists, and thus forming one offspring. Considering the spouse chromosome for removal we can generate the other offspring.
      - Mutation: If a chromosome is selected, using randomly generated probability within the range $[L_M, U_M]$, for mutation then any attribute, chosen randomly, is replaced by the rest of the attributes available.

   b) *Fitness function: The fitness of i^{th} chromosome AS_i^{rec} of the population is evaluated by the following equation.*

$$F_2(AS_i^{rec}) = \left(rcc_{rec,AS_i^{rec}}\right)^{CPOWER} \times \left(asen_{AS_i^{rec}}\right)^{FPOWER} \quad (6)$$

   c) *Selection procedure:* The selection of next generation chromosomes is done by the Roulette Wheel selection scheme.

   END WHILE
END FOR

Here the algorithm has n different attribute subset for each individual record which implies n different clusters.

C. Cluster Selection

1) Reject the clusters with single record only.

2) Reject a cluster $C_{j,AS^{r_j}}$, if for any $C_{i,AS^{r_i}}$, and i≠j the following condition holds.

$$\left(C_{j,AS^{r_j}} \subseteq C_{i,AS^{r_i}}\right) \; and \; (AS^{r_j} \subseteq AS^{r_i})$$

3) Assign ranks to the remaining clusters according to their fitness defined as follows.

$$F_3(AS_i^{rec}) = rcc_{rec,AS_i^{rec}} \times asen_{AS_i^{rec}} \quad (7)$$

4) Arrange the clusters in descending order according to their rank.

5) Initialize an empty set SC to maintain the selected cluster's list.

6) FOR each remaining clusters starting from the highest rank
BEGIN
If the median of this cluster is uncorrelated by at least a user defined parameter CF with the median of all the clusters in SC then
BEGIN
Add this cluster with SC.
ELSE
Reject this cluster.
END IF
END FOR

The clusters in SC are the evolved clusters of the whole algorithm. So, finally the algorithm produces some overlapping clusters with some rules for each cluster, which can further guide a record to be a member of some particular clusters depending on its attribute value.

III. Experimental Results

A. Experimental Data

To evaluate, the performance of this overlapping evolutionary clustering algorithm with optimized experimental conditions, we have tested our algorithm with a simulated dataset and some real dataset described below.

1) Ad400_10_10: This is a simulated synthetic dataset generated as in Yeung et al. (2001). This data set has 400

genes each consists of 10 different attributes. The data set has 10 clusters; each contains 40 genes, which represents ten different expression patterns.

2) Yeast Sporulation (Chu et al., 1998): This is a 6118 gene-7 attribute microarray dataset collected during the sporulation process at different time point (0, 0.5, 2, 5, 7, 9 and 11.5 h) of budding yeast. The data are then log-transformed. The Sporulation data set is publicly available at the website http://cmgm.stanford.edu/pbrown/sporulation. Among the 6118 genes, the genes whose expression levels did not change significantly during the harvesting have been ignored from further analysis. This is determined with a threshold level of 1.6 for the root mean squares of the log2-transformed ratios. The resulting set consists of 474 genes.

3) Human Fibroblasts Serum (Iyer et al., 1999): This data set contains the expression levels of 8613 human genes. The data set has 13 dimensions corresponding to 12 time points (0, 0.25, 0.5, 1, 2, 4, 6, 8, 12, 16, 20 and 24 h) and 1 unsynchronized sample. A subset of 517 genes whose expression levels changed substantially across the time points have been chosen (Eisen et al., 1998). This data set can be downloaded from http://www.sciencemag.org/feature/data/984559.shl.

4) Rat CNS (Wen et al., 1998): The Rat CNS data set has been obtained by reverse transcription-coupled PCR to examine the expression levels of a set of 112 genes during rat central nervous system development over 9 time points. This data set is available at http://faculty.washington.edu/kayee/cluster.

Each data set is normalized so that each row has mean 0 and variance 1 (Z normalization).

B. Performance Metrics

For evaluation of the performance criteria we used Silhouette Index measure for the real life data set.

1) . Silhouette index: Silhouette index is a cluster validity index that is used to measure the excellence of any clustering algorithm output. Suppose $d(i, C)$ is the average desimilarity (distance) of i_{th} record (gene) and all (excluding i, if $i \in C$) records of cluster C. Consider that there are K clusters $(C_1, C_2, \dots, C_K)$ and $i \in C_r$. If $a(i) = d(i, C_r)$ and $b(i) = min\left(\{d(i, C_j) \mid 1 \le j \le k \; and \; j \ne r\}\right)$ then silhouette width $s(i)$ of the i_{th} record is defined as:

$$s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))} \tag{7}$$

And silhouette index $Si = mean(\{s(j) \mid 1 \le j \le K\})$. The maximum value of Silhouette Index is +1, which represents best clustering solution and the minimum is -1 that represents the worst.

2) Eisen plot: The Eisen plot is a graphical representation of the normalized gene expression values for each gene. In Eisen plot, each row consists of M number of cells to represent M different attributes, where each is colored depending on the corresponding attribute value. Thus, if the genes of a cluster are assigned in consecutive rows then that part will look similar as the cluster elements are similar. In our representation, the genes of same cluster are placed in consecutive rows, so that the cluster quality can be visualized easily. The cluster boundaries are seperated by white colored blank rows.

3) Cluster profile plot: The cluster profile plot shows the normalized gene expression values of the genes of a cluster with respect to the time points. The median of expression values of the genes of a cluster over different time points are shown as a thick black line.

C. Results

The parameter settings, which have been estimated by the performance metrics, for the Yeast Sporulation Human Fibroblasts Serum and Rat CNS data are listed below (i.e., in Table I, II & III). We have executed the algorithm for 50 times for each different RoP over the Yeast Sporulation dataset and the optimized attribute value range, found in maximum optimization after the first stage (ie II. *A*.) of the proposed Algorithm are listed in Table II.

To test performance of the algorithm, we executed this for consecutive eleven time on Yeast Sporulation, Human Fibroblasts Serum and Rat CNS data with the specified parameter setting (i.e., in Table I, II & III respectively). We have calculated the Silhouette Index of the resultant clusters of this eleven run and the result of the run whose Silhouette Index ranked median is selected to be displayed in this article. We have compared these results with other's in terms of silhouette index and number of clusters (i.e., in Table V).

As stated earlier, changing the parameter RoP can change the optimized attribute values and thus can change a lot in the result. The variation of the optimized attribute value, for the Yeast Sporulation data, has been described by Fig. 1.

Increasing RoP will decrease the optimized attribute value for all the attributes and hence the number of Genes, participating to form a cluster, is going to decrease but the gain is that the precession will increase by only allocating the closely related genes in a cluster. And that is further going to increase the number of clusters also. So, varying RoP, the user has to trade of between number of cluster and number of genes in a cluster.

The cluster profile plots and the Eisen plot for Yeast Sporulation data is given in Fig. 2.

Table VI shows the attribute value rules of the Cluster 1 and Cluster 2 evolved in the selected (stated earlier) run.

TABLE I. PARAMETER SETTING OF ECOEC FOR YEAST SPORULATION DATA

U_C	L_C	U_M	L_M	U_F	L_F	RoP	IP	CF	CPOWER	FPOWER	MaxFech
0.8	0.7	0.15	0.1	0.8	0.6	0.75	20	0.65	1	7	7

TABLE II. PARAMETER SETTING OF ECOEC FOR HUMAN FIBROBLASTS SERUM

U_C	L_C	U_M	L_M	U_F	L_F	RoP	IP	CF	CPOWER	FPOWER	MaxFech
0.8	0.7	0.15	0.1	0.8	0.6	0.75	20	0.65	1	13	13

TABLE III. PARAMETER SETTING OF ECOEC FOR RAT CNS

U_C	L_C	U_M	L_M	U_F	L_F	RoP	IP	CF	CPOWER	FPOWER	MaxFech
0.8	0.7	0.15	0.1	0.8	0.6	0.8	20	0.65	1	9	9

TABLE IV. OPTIMIZED ATTRIBUTE VALUE OF ALL ATTRIBUTE OF YEAST SPORULATION DATA

t0	t0.5	t2	t5	t7	t9	t11.5
0.383	0.515	0.723	0.805	0.267	0.462	0.431

TABLE V. SILHOUTTE INDEX AND NUMBER OF EVOLVED CLUSTERS FOR REAL LIFE DATA SETS AND COMPARISON WITH OTHER ALGORITHM

Algorithm	Yeast Sporulation		Human Fibroblasts Serum		Rat CNS	
	NC[a]	Si[b]	NC[a]	Si[b]	NC[a]	Si[b]
ECoEC	6	0.8379	8	0.5410	7	0.5901
SiMM-TS[2]	6	0.6247	6	0.4289	6	0.5239
IFCM[2]	7	0.4755	8	0.2995	5	0.4050
VGA[2]	6	0.5703	6	0.3443	6	0.4486
Average Linkage[2]	6	0.5007	6	0.3092	6	0.3684
SOM[2]	6	0.5845	6	0.3235	6	0.4122
CRC[2]	8	0.5622	10	0.3174	4	0.4423

a. NC is the number of clusters b. Si is the Silhouette Index

IV. CONCLUSION

Structural genes are controlled by regulatory genes. The proteins encoded by these regulatory genes are capable of binding to the cell's DNA near the promoter sequences of the structural genes to turn on or off the transcription processes depending on the chemical environment of the cell. But the binding site of a gene may also be the binding site of another gene. So, switching off a gene may switch off other genes and thus further may produce another undesirable action. That's why it's very important to identify those genes which behave similarly (the genes which have same binding site, generally have same expression profile in various experimental conditions) and also the number of biological processes any particular gene is involved.

Clustering techniques in microarray data analysis tries to identify the genes which have similarity in expression profile (these genes are called co-expressed genes). The gene expression analysis discovers many new rules to identify similar kind of genes which may further be used to allocate a new gene in the previous group of genes.

The algorithm, proposed in this article, finds the cluster groupings like other clustering algorithm but has a significant difference with many others. That is, it finds out the overlapping clusters which support the biological concept that a gene may be involved in more than one biological process. It also finds the rules to become a member of a particular cluster. Hence, a new gene can be allocated to a cluster without recalculation. Though the time required calculating those cluster grouping is a little high but it provides a good grouping.

The proposed algorithm has many parameters that can be varied to tune the result near to optimum clustering solution. Also, changing the selection criteria CF of the third stage of the algorithm (i.e., *C*) can produce more exiting solutions. The proposed algorithm selects very few no of records in a cluster when ROP is high. Also, the CPOWER and POWER parameters guide the optimization process to emphasis on record count or maximal feature similarity. So, they can also be tuned up to have a desired solution.

As a future work one can optimize the tuning parameters by using some optimization algorithm like GA, PSO, DE etc. This algorithm can be applied to other application domains also.

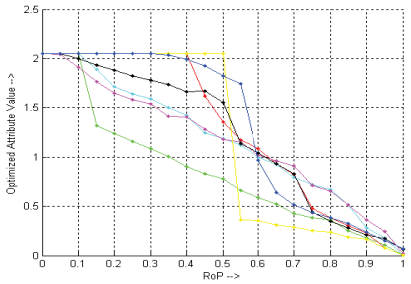

Figure 1. Change of optimized attribute value with with respect to RoP.

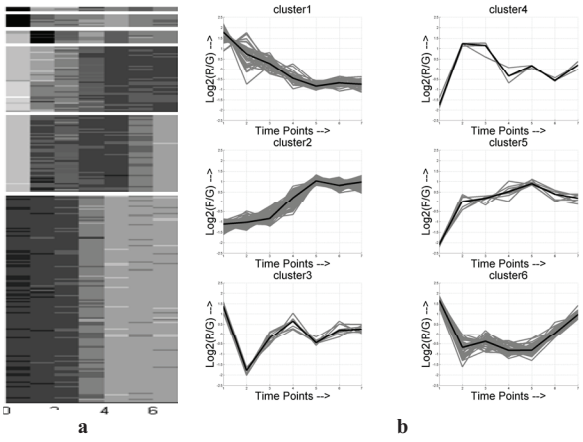

Figure 2. Yeast Sporulation data clustered using ECoEC algorithm: (a)Eisen Plot (b) Clusterprofile plots of the evolved cluster.

TABLE VI. ATTRIBUTE VALUE RULES OF TWO CLUSTERS EVOLVED FROM YEAST SPORULATION DATA

For Cluster	Rule is that the gene expression values of a gene should be within the range	
	Attribute	Value Range
1	t2	0.1246 ± 0.717 and
	t5	-0.3872 ± 0.748 and
	t7	-0.8143 ± 0.255 and
	t9	-0.8658 ± 0.521 and
	t11.5	-0.7167 ±0.431
2	t0	-1.2449 ± 0.384 and
	t0.5	-0.9718 ± 0.52 and
	t2	-0.7872 ±0.717 and
	t5	0.0754 ±0.748 and
	t7	1.0798 ±0.255 and
	t9	0.9756 ±0.521 and
	t11.5	0.8730 ±0.431

V. REFERENCES

[1] IEEE transactions on evolutionary computation, "An Evolutionary Clustering Algorithm for Gene Expression Microarray Data", Analysis Patrick C. H. Ma, Keith C. C. Chan, Xin Yao, *Fellow, IEEE*, and David K. Y. Chiu.

[2] BIOINFORMATICS Vol. 23 no. 21 2007, pages 2859–2865, "An improved algorithm for clustering gene expression data", Sanghamitra Bandyopadhyay, Anirban Mukhopadhyay and Ujjwal Maulik.

[3] "An Information Theory Approach for Validating Clusters in Microarray Data" Sudhakar Jonnalagadda[1] and Rajagopalan Srinivasan[1*] [1]Department of Chemical and Biomolecular Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260

[4] "Evolutionary Fuzzy Clustering Algorithm with Knowledge-Based Evaluation and Applications for Gene Expression Profiling", Han-Saem Park, Si-Ho Yoo, and Sung-Bae Cho, Journal of Computational and Theoretical Nanoscience,Vol.2, 1–10, 2005.

[5] "Computational Intellingence", Amit Konar, ETCE dept. Jadavpur University, Springer.

[6] Y. Wang and A. K. C. Wong, "From association to classification: Inference using weight of evidence," *IEEE Trans. Knowl. Data Eng.*, vol. 15, pp. 764–767, May–Jun. 2003.

[7] Y. Wang and A. K. C. Wong, "Pattern Discovery: A Data Driven Approach to Deission Support",*IEEE Trans. Knowl. Data Eng.*, vol. 15, pp. 764–767, May–Jun. 2003.

[8] H. L. Turner, T. C. Bailey, W. J. Krzanowski and C. A. Hemingway, "Biclustering Models for Structured Microarray Data", IEEE/ACM Trans on Computational Biology and Bioinformatis, Vol. 2 no. 4. Oct-Dec 2005.

[9] "A Genetic Programming Tutorial", John R. Koza1 and Riccardo Poli2 1Stanford University, Stanford, California 2Department of Computer Science, University of Essex, UK

[10] "Unbiased Tournament Selection", Artem Sokolov, Darrell Whitley, GECCO'05, June 25–29, 2005, Washington, DC, USA.

[11] "Genetic Programming ~ An Introduction"- a book by W. Banzhaf, P. Nordin, R. E. Keller and F. D. Francone.

[12] "Analysis of microarray data", Markus Ringn´er, Computational Biology and Biological, Physics Department of Theoretical Physics, Lund University

[13] "FDR adjustments of Microarray Experiments (FDR-AME)", Yoav Benjamini, Ephraim Kenigsberg, Anat Reiner, Daniel Yekutieli, July 18, 2005, Department of Statistics and O.R., Tel Aviv University

[14] "Feature Selection of DNA Microarray Data", Mohammed Liakat Ali, Course: 60-520, Fall 2005, University of Windsor, December 2, 2005

[15] "Analysis of microarray gene expression data" by Wolfgang Huber, German Cancer Research Center, Division of Molecular Genome Analysis, 69120 Heidelberg, Anja von Heydebreck, Max-Planck-Institute for Molecular Genetics, 14195 Berlin, Martin Vingron, Max-Planck-Institute for Molecular Genetics, 14195 Berlin, April 2, 2003

[16] "Introduction to Microarray Technology", Jennifer A. Love, Whitehead Institute Center for Microarray Technology

[17] Journal of Data Science (2003), 103-121, "Analysis of Unbalanced Microarray Data", Mei-Ling Ting Lee1,2,3,4, G.A.Whitmore5, Rus Y. Yukhananov1,2

[18] "Large-scale regulatory network analysis from microarray data: modified Bayesian network learning and association rule mining", Zan Huang , Jiexun Li, Hua Su, George S. Watts, Hsinchun Chen, ScienceDirect, Decision Support Systems 43 (2007) 1207–1225

[19] "Machine Learning in Lowlevel Microarray Analysis", Benjamin I. P. Rubinstein, Jon McAuliffe, Simon Cawley, Marimuthu Palaniswami, Kotagiri Ramamohanarao1, Terence P. Speed

[20] "Minimum information about a microarray experiment (MIAME)—toward standards for microarray data", Alvis Brazma, Pascal Hingamp, John Quackenbush, Gavin Sherlock, Paul Spellman, Chris Stoeckert, John Aach, Wilhelm Ansorge, Catherine A. Ball, Helen C. Causton, Terry Gaasterland, Patrick Glenisson, Frank C.P. Holstege, Irene F. Kim, Victor Markowitz, John C. Matese, Helen Parkinson, Alan Robinson, Ugis Sarkans, Steffen Schulze-Kremer, Jason Stewart, Ronald Taylor, Jaak Vilo & Martin Vingron

[21] "Microarray Data Mining: A Survey", SAMBA/02/01, Kjersti Aas, January 2001

[22] IEEE Transactions On Knowledge And Data Engineering, "Compression, Clustering and Pattern Discovery in Very High Dimensional Discrete-Attribute Datasets", Mehmet Koyut¨urk, Ananth Grama, and Naren Ramakrishnan

[23] "Statistical Significance In Biological Sequence Comparison", William R. Pearson and Todd C. Wood,Department of Biochemistry and Molecular Genetics, University of Virginia, Charlottesville, VA 22908, April 11, 2000 - *final*

[24] "Classifying Microarray Data Using Support Vector Machines", Sayan Mukherjee, *PostDoctoral Fellow: MIT/Whitehead Institute for Genome Research and Center for Biological and Computational Learning at MIT*

[25] "Kernel Analysis for Noisy Microarray Data", Tsuyoshi Kato, Wataru Fujibuchi and Kiyoshi Asai, June 12, 2006

[26] "Comparison of the Empirical Bayes and the Significance Analysis of Microarrays", Holger Schwender, Andreas Krause and Katja Ickstadt

[27] "Visualisation of Reduced-Dimension Microarray Data Using Gaussian Mixture Models", Julien Epps and Eliathamby Ambikairajah

[28] "Feature Selection for High-Dimensional Genomic Microarray Data", Eric P. Xing, Michael I. Jordan and Richard M. Karp

[29] "Speeding Up Evolution through Learning: LEM", Ryszard Michalski*, Guido Cervone and Kenneth Kaufman, Machine Learning and Inference Laboratory George Mason University, Fairfax, Virginia

[30] "New Ways to Calibrate Evolutionary Algorithms", A.E. Eiben, M.C. Schut Department of Computer Science, Faculty of Science, VU University, Amsterdam, The Netherlands.

[31] IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 33, No. 1, February 2003, "Pattern Discovery: A Data Driven Approach to Decision Support", Andrew K. C. Wong, *Senior Member, IEEE,* and Yang Wang, *Member, IEEE*

[32] "Transparent Decision Support Using Statistical Evidence" by Andrew Michael Hamilton-Wright, A thesis presented to the University of Waterloo in fulfillment of the thesis requirement for the degree of Doctor of Philosophy in Systems Design Engineering Waterloo, Ontario, Canada, 2005

[33] "Mining Fuzzy Association Rules", *Keith C.C. Chan Wai-Ho Au,* Department of Computing, The Hong Kong Polytechnic University Hung Hom, Kowloon, Hong Kong

[34] "A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery", Alex A. Freitas Postgraduate Program in Computer Science, Pontificia Universidade Catolica do Parana Rua Imaculada Conceicao, 1155. Curitiba - PR. 80215-901. Brazil

[35] "Clustering with a Genetically Optimized Approach", L.O. Hall, B. Ozyurt, J.C. Bezdek,

[36] "On The Use Of Evolutionary Algorithms In Data Mining", Erick Cantú-Paz and Chandrika Kamath Center for Applied Scientific Computing Lawrence Livermore National Laboratory 7000 East Avenue, Livermore, CA 94550

[37] Intelligent Data Analysis 6 (2002) 531–556 531, IOS Press, "Evolutionary model selection in unsupervised learning", YongSeog Kim, W. Nick Street and Filippo Menczer, Accepted 5 May 2002

[38] "Differential Evolution: The Real-Parameter Genetic Algorithm Applied to Materials and Metallurgy" By N. Chakraborti

[39] "Association Mining in Large Databases: A Re-Examination of Its Measures", Tianyi Wu1, Yuguo Chen2, and Jiawei Han1