Cuckoo Search with Mutation for Biclustering of Microarray Gene Expression Data

Balamurugan Rengeswaran, Natarajan Mathaiyan, and Premalatha Kandasamy
Department of Computer Science and Engineering, Bannari Amman Institute of Technology, India

Abstract: *DNA microarrays have been applied successfully in diverse research fields such as gene discovery, disease diagnosis and drug discovery. The roles of the genes and the mechanisms of the underlying diseases can be identified using microarrays. Biclustering is a two dimensional clustering problem, where we group the genes and samples simultaneously. It has a great potential in detecting marker genes that are associated with certain tissues or diseases. The proposed work finds the significant biclusters in large expression data using the Cuckoo Search with Mutation (CSM). The cuckoo imitates its egg similar to host bird's egg using a mutation operator. Mutation is used for exploration of search space, more precisely to allow candidates to escape from local minima. It focuses on finding maximum biclusters with lower Mean Squared Residue (MSR) and higher gene variance. A qualitative measurement of the formed biclusters with a comparative assessment of results is provided on four benchmark gene expression dataset. To demonstrate the effectiveness of the proposed method, the results are compared with the swarm intelligence techniques Binary Particle Swarm Optimization (BPSO), Shuffled Frog Leaping (SFL), and Cuckoo Search with Levy flight (CS) algorithm. The results show that there is significant improvement in the fitness value.*

Keywords: *Biclustering, CS, BPSO, SFL, levy flight, gene expression data, mutation.*

Received January 1, 2014, accepted July 22, 2014

1. Introduction

DNA microarray technology is attracting wonderful interest both among the scientific community and in industry, with its ability to measure simultaneously the activities and interactions of thousands of genes [16]. Gene expression data are typically analyzed in matrix form with each row representing a gene and each column representing a condition or sample. The conditions may belong to different time points or different environmental conditions. The row vector of a gene is called the expression pattern of the gene and a column vector is called the expression profile of the condition. Each element of this matrix represents the expression level of a gene under a specific condition, and is represented by a real number. It is usually the logarithm of the relative profusion of the mRNA under the specific condition. Figure 1 shows the gene expression matrix.

	Con 1	*Con 2*	...	...	*Con M*
Gene 1	$GEx_{1,1}$	$GEx_{1,2}$	...	...	$GEx_{1,M}$
Gene 2	$GEx_{2,1}$	$GEx_{2,2}$	...	...	$GEx_{2,M}$
...	...	...	...	...	...
...	...	...	...	...	...
Gene N	$GEx_{N,1}$	$GEx_{N,2}$	...	...	$GEx_{N,M}$

Figure 1. Gene expression matrix.

Given a gene expression matrix a common analysis goal is to group genes and conditions into subsets that convey biological significance. In its most common form, this task translates to the computational problem known as clustering. Formally, for a given set of objects and its vector of attributes, the clustering aims to partition the object into disjoint classes. So that the objects within a cluster are similar and the objects of disjoint clusters are dissimilar. For example, when analyzing a gene expression matrix clustering may be applied to the genes for identifying groups of co-regulated genes or cluster the conditions for discovering groups of similar conditions.

Analysis via clustering makes several assumptions that may not be completely adequate in all situations. First the clustering can be applied to either genes or conditions; it implicitly directs the analysis of a particular aspect of the system. Second, clustering algorithms usually seek a disjoint cover of the set of elements, requiring that no gene or sample belongs to more than one cluster. The concept of a bicluster rises to a more flexible computational framework. For example if two genes are related they can have similar expression patterns under certain conditions; similarly, for two related conditions, some genes may exhibit different expression patterns. As a result, each cluster may involve only a subset of genes and a subset of conditions. Biclustering is a simultaneous clustering of both rows and columns of a gene expression data.

The problem of partitioning a set of objects into k groups, which optimizes a stated condition of partition adequacy, is not straightforward. Given n objects, the number of ways in which these objects can be partitioned into k non-empty subsets is [13] given in Equation 1.

$$P(n,k)=\frac{1}{k!}\sum_{j=0}^{k}\binom{k}{j}(-1)^{j}(k-j)^{n} \quad (1)$$

Equation 2 approximates Equation 1:

$$P(n,k)\approx\frac{k^{n}}{k!}\approx k^{n-k}e^{k}\sqrt{2\pi k} \quad (2)$$

Therefore, when the number of clusters *k* is not known in advance then the total number of valuations is given in Equation 3.

$$T(n)=\sum_{k=1}^{n}P(n,k) \quad (3)$$

Finding significant biclusters in a microarray is a much more complex problem than clustering [7] and it is a NP-hard problem [19]. The problem of finding a consistent biclustering can be formulated as an optimization problem. An optimization problem is a problem which determines the set of potential solutions to the problem and defines one or more criteria which measures the quality of an individual solution. The solution is obtained by identifying the best solution from the set or an adequately high quality solution among the set.

This work develops and implements the biclustering based on the most popular and robust bio inspired strategy Cuckoo Search (CS). In the conventional CS, each nest consists of a single egg and cuckoo imitates the egg using Levy flight. In the proposed CS algorithm Levy flight is replaced by mutation operator. The remainder of this paper is organized as follows: section 2 provides the related works in biclustering. Section 3 gives a general overview of the CS. The Cuckoo Search with Mutation (CSM) is illustrated in section 4. Kennedy and Eberhart proposed a discrete binary version of Binary Particle Swarm Optimization (BPSO) for binary problems [12]. The Shuffled Frog Leaping (SFL) algorithm is a memetic metaheuristic that is designed to seek a global optimal solution by performing a heuristic search [8]. It is based on the evolution of memes carried by individuals and a global exchange of information among the population. Section 5 presents the detailed experimental setup and results for comparing the performance of the CSM with BPSO, SFL and CS.

2. Review of Related Works

As we mentioned in the introduction of this paper, the biclustering problem is a NP-hard [19]. For that reason, heuristic search algorithms are usually used to approximate the problem by finding suboptimal solutions. The biclustering algorithms are classified into two different approaches: systematic search and metaheuristic algorithms. Cheng and Church [4] presented a first biclustering approach for gene expression data. Their algorithm adopts a sequential covering strategy in order to return a list of *n* biclusters from an expression data matrix. Statistical-Algorithmic Method for Bicluster Analysis (SAMBA), a biclustering algorithm that performs simultaneous bicluster identification by using exhaustive enumeration [19]. CoBi: pattern based co-regulated biclustering of gene expression data [18]. It is mainly used for grouping both positively and negatively regulated genes from microarray expression data.

Order-Preserving Sub-Matrix (OPSM) is a submatrix where there is a permutation of its columns under which the sequence of values in every row is strictly increasing [1]. An Iterative Signature Algorithm (ISA) defines biclusters as transcription modules to be retrieved from the expression data [2]. Divina and Aguilar-Ruiz [7] presented a Sequential Evolutionary BIclustering (SEBI) approach. The term sequential refers the way in which bicluster are discovered, only one bicluster obtained per each run of the evolutionary algorithm.

Maximum Similarity Bicluster (MSB) algorithm [15] is based on greedy iterative search. A greedy strategy of removing rows/columns iteratively is employed to provide the MSB in polynomial time. Liu *et al*. [14] proposed their biclustering approach based on the use of a PSO together with crowding distance as the nearest neighbour search strategy. A novel biclustering algorithm is based on the use of an Evolutionary Approach (EA) together with hierarchical clustering [10]. It merges both the neighbourhood search and the evolutionary approaches.

3. CS with Levy Flight

CS is an optimization technique developed by Yang and Deb [21] based on the brood parasitism of the cuckoo species by laying their eggs in the nests of other host birds. Based on the selfish gene theory [6] this parasitic behaviour increases the chance of survival of the cuckoo's genes. Since, the cuckoo need not spend any energy rearing its young one. The CS algorithm utilizes these behaviours in order to traverse the search space and find optimal solutions. A set of nests with one egg are placed in random locations in the search space where the each egg represent a candidate solution. The number of cuckoos is assigned to traverse the search space, recording the highest objective values for different encountered candidate solutions. The cuckoos utilize a search pattern called levy flight which is encountered in real insects, fish and birds. When generating new solutions $x(t+1)$ for a cuckoo *i*, a Levy flight is performed using the following Equation 4.

$$x_{i}(t+1)=x_{i}(t)+\alpha\oplus Levy(\lambda) \quad (4)$$

The symbol $\oplus$ is an entry-wise multiplication. Basically Levy flights provide a random walk while their random steps are drawn from a Levy distribution for large steps given in Equation 5, which has an infinite variance with an infinite mean. Here the

consecutive jumps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail. The rules for CS are described as follows:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
- The best nests with high quality of eggs will carry over to the next generations.
- The number of available host nests is fixed, and a host can discover a foreign egg with a probability $p_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

$$Levy \sim u = t^{-\lambda} \quad (5)$$

4. CS with Mutation

The traditional CS [21] considers single egg in a nest and a cuckoo lays one egg at a time by using Levy flight. Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state in genetic algorithm [17]. This can result in entirely new gene values being added to the gene pool. Mutation is an important part of the genetic search as it helps to prevent the population from stagnating at any local optima. Mutation occurs during evolution according to a user-definable mutation probability. In case of a large mutation rate the population has difficulties to converge to a (global) minimum. This probability should usually be set fairly low (0.01 is a good first choice). If it is set to high, the search will turn into a primitive random search. The proposed CS uses the mutation operator to generate a new solution. The cuckoo imitates the host bird's egg by using mutation.

4.1. Biclutering Representation

Each cuckoo is represented as candidate solution for the problem. Solutions are encoded by means of binary strings of length N+M, where N and M are the number of rows (genes) and of columns (conditions) of the expression. A bit is set to one if the corresponding gene and/or condition are present in the bicluster, and reset to zero otherwise. The CS works well for continuous optimization problem. So the individual dimension of an egg is represented by a real number. The mapping function for an egg into a binary string representation of a bicluster is given in Equation 6 as follows:

$$y_{ij} = \begin{cases} x_{ij} < 0.5 & 0 \\ otherwise & 1 \end{cases} \quad (6)$$

Where x_{ij}: Random value generated for j^{th} gene/condition of i^{th} egg, and y_{ij}: Binary string representation of bicluster of x_{ij} in y_{ij}, if a bit is set to 1 then the corresponding gene or condition belongs to the encoded bicluster; otherwise it is not. Figure 2 shows the representation of an egg and its mapped bicluster representation.

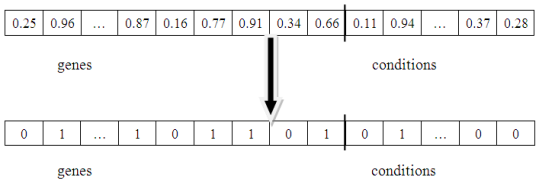

Figure 2. Representation of an egg and its mapping to bicluster.

4.2. Fitness Function

Mean Squared Residue (MSR) problem has been proposed by Cheng and Church [4] for identifying biclusters. Let gene expression data matrix A has N rows and M columns, where a cell a_{ij} is a real value that represents the expression level of gene i under condition j. Matrix A is defined by its set of rows $R=\{r_1, r_2, ..., r_N\}$ and its set of columns $C=\{c_1, c_2, ..., c_M\}$. Given a matrix, biclustering finds sub-matrices, which are subgroups of genes and subgroups of conditions, where the genes exhibit highly correlated behaviour for every condition. Given a data matrix A, the goal is to find a set of biclusters such that each bicluster exhibits some similar characteristics.

- *Definition 1:* Let $A_{IJ}=(I, J)$ be a submatrix of A where $I \in R$ and $J \in C$. A_{IJ} contains only the elements a_{ij} belonging to the submatrix with set of rows I and set of columns J. The residue of an element a_{ij} in a sub matrix A_{IJ} equals, $r_{i,j}=a_{i,j}+a_{I,J}-a_{I,j}-a_{i,J}$ where a_{iJ} is the mean of the i^{th} row in the bicluster, a_{Ij} the mean of the j^{th} column in the bicluster, and a_{IJ} is the mean of all the elements within the bicluster.

The difference between the actual value of a_{ij} and its expected value, predicted from its row, column and bicluster mean, are given by the residue of an element. It also reveals its degree of coherence with the other entries of the bicluster it belongs to. The quality of a bicluster can be evaluated by computing the MSR f_1, i.e., the sum of all the squared residues of its elements is given in Equation 7.

- *Definition 2:* The sum of all the squared residues of its elements of bicluster (I, J) is defined:

$$f_1(I, J) = \frac{1}{|I||J|} \sum_{i \in I} \sum_{j \in J} r_{i,j}^2 \quad (7)$$

The lowest score of $f_1(I, J)$ is 0, which indicates that the gene expression levels vary in harmony. This includes the trivial or constant biclusters where there is no fluctuation. These trivial biclusters may not be interesting but need to be revealed and masked so more interesting ones can be found. The gene variance may be a complementary score to reject trivial biclusters. The gene variance can be represented in Equation 8 as follows:

- *Definition 3:* The gene variance of bicluster (*I*, *J*) is defined:

$$f_2(I,J) = \frac{1}{|I|}\sum_{i\in I} v_r(i)$$
$$v_r(i) = \frac{1}{|J|}\sum_{j\in J}(a_{i,j} - a_{i,J})^2 \quad (8)$$

The optimization task is finding one or more biclusters by maintaining the two competing constraints, viz., homogeneity and gene variance. Our goal is to obtain biclusters with the maximum number of genes and conditions, with the minimum value of *f*(*I*, *J*). The fitness function for obtaining bicluster is defined in Equation 9 as follows:

- *Definition 4:* The fitness function of bicluster (*I*, *J*) is defined:

$$f(I, J) = f_1(I, J) + \frac{1}{f_2(I, J)} \quad (9)$$

The final objective of Algorithm 1 is to minimize the fitness.

Algorithm 1: CS with Mutation (CSM) algorithm.
for k= 1 *to n do*
Generate random population with n nests and each nest consists of an egg.
While (t<MaxGeneration)
Get a cuckoo(say i) randomly and generate a solution using mutation
Choose a nest among n (*say, j*) *randomly*;
Replace worst egg in j by the new solution i;
A fraction (*pa*) *of worse nests are abandoned and new ones/solutions are built/generated*
Keep best solutions (*or nests with quality solutions*)
Rank the solutions/nests and find the current best;
Pass the current best to the next generation;
end while
Arrange the best solution of individual nest in ascending order
BC(k)=the best solution
endfor
for each solution i from to 2 *to n*-1
Find the Jaccard index with previous (*i*-1) *solutions* (*Equation* 10)
Get the maximum rate (*mr*)
If mr exceeds the given threshold
Reject the solution i
Otherwise
Present the solution i
End

4.3. Identifying the Overlapping between the Biclusters

The gene expression data may consist of number of biclusters. The best egg of each nest is taken as a solution for bicluster. The most important part for bicluster validation is the comparison of a current bicluster to a already found bicluster. The proposed work adapts Jaccard index [11] for identifying similarity/overlapping between biclusters. To compare the biclusters it calculates the fraction of row-column combinations in both (intersection) bicluster with respect to row-column combination in at least one (union) bicluster. The Jaccard index for two biclusters is given in Equation 10.

$$jac\left(BC_i, BC_j\right) = jac_{ij} = \frac{\left|BC_i \cap BC_j\right|}{\left|BC_i \cup BC_j\right|} \quad (10)$$

Where BC_i: Ith bicluster, BC_j: Jth biclusters, $|BC_i \cap BC_j|$: The size of intersection of two biclusters BC_i and BC_j, $|BC_i \cup BC_j|$: The size of union of two biclusters BC_i and BC_j, The maximum Jaccard index is considered as overlapping rate of the bicluster. In general $m_r(i)=max(jac_{ij})$ where $1 \leq j < i$.

5. Experimental Results and Analysis

5.1. Data Sets

The biclustering algorithm has been applied to four datasets in order to study its performance, namely the yeast saccharomyces cerevisiae stress expression data [9], arabidopsis thaliana expression data [3], yeast saccharomyces cerevisiae cell cycle expression data [5] and rat CNS expression data [20] are used. The first Gasch yeast is the saccharomyces cerevisiae with 2993 genes and 173 conditions. The second one arabidopsis thaliana expression data contain 734 genes and 69 conditions. The third dataset yeast cell cycle data contains 2884 genes and 17 experimental conditions. The rat CNS dataset has set of 112 genes under 9 conditions. Table 1 shows the parameter and its value used in this paper. The parameters p_a, α and λ are set as 0.25, 1 and 1.5 respectively [21]. Through empirical analysis the population size and the number of iterations are set as 20 and 100 respectively.

Table 1. Parameter and its value.

Parameter	Value
Mutation Probability	0.1
p_a	0.25
A	1
λ	1.5
Number of Nests	20
Iteration	100
Overlapping Rate	50%

Figures 3, 4, 5, and 6 shows the fitness value obtained for Gasch yeast expression data, Arabidopsis thaliana expression data, yeast cell cycle expression data and rat CNS data respectively. The BPSO has premature converge due to stagnation. The SFL gives better performance than BPSO and CS. For all the data sets the proposed CSM outperforms all other algorithms because the mutation allows CS to escape from local optimum and successfully continue to the global optimum.

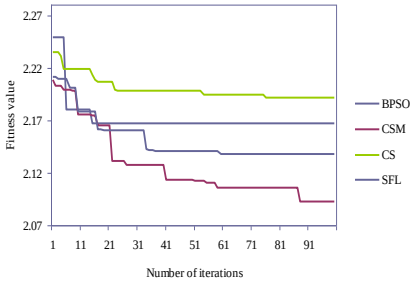

Figure 3. Fitness value obtained for saccharomyces cerevisiae expression data.

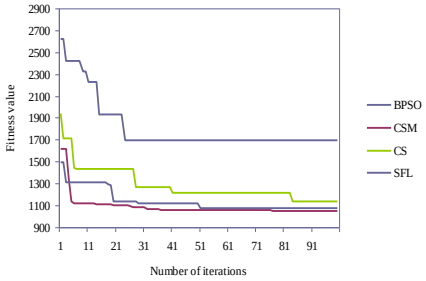

Figure 4. Fitness value obtained for arabidopsis thaliana expression data.

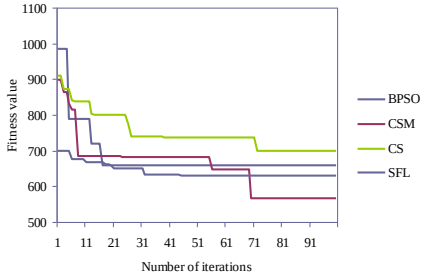

Figure 5. Fitness value obtained for yeast cell cycle expression data.

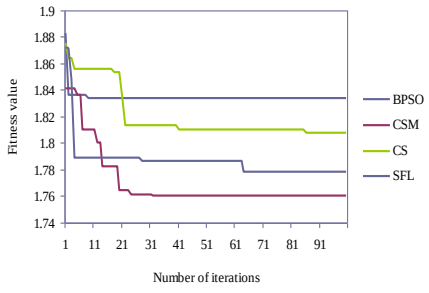

Figure 6. Fitness value obtained for rat CNS expression data.

According to the problem formulation the size of an extracted bicluster should be as large as possible while satisfying a homogeneity criterion. The expression levels of each gene within the bicluster should be similar over the range of conditions. The bicluster should satisfy two requirements simultaneously. That is it should have a low MSR score. On the other hand, the bicluster gene variance should be high. The MSR represents the variance of the selected genes and conditions with respect to the homogeneity of the bicluster. Gene variance removes the simple or trivial bicluster. Coherence Index (CI) is used as a measure of evaluating bicluster's goodness. CI is defined as the ratio of MSR score to the size of the formed bicluster.

Tables 2, 3, 4, and 5 show sample experimental results obtained for saccharomyces cerevisiae expression data, arabidopsis thaliana expression data, yeast saccharomyces cerevisiae cell cycle expression data and rat CNS expression data respectively. After removing the overlapping biclusters among the 20 biclusters, 5 biclusters are chosen randomly. Clearly Figure 7 shows the small bicluster of size 8×5 for rat CNS expression data.

Table 2. Experiment results for saccharomyces cerevisiae stress expression data.

Bicluster No.	Genes	Conditions	Volume	MSR	Gene Variance	CI	Fitness
BC_1	1533	94	144102	0.6727	0.6888	4.7×10^{-6}	2.1247
BC_5	1477	87	128499	0.6448	0.6671	5.0×10^{-6}	2.1437
BC_8	1482	77	114114	0.6390	0.6631	5.6×10^{-6}	2.1470
BC_{10}	1486	92	136712	0.6611	0.6773	4.8×10^{-6}	2.1376
BC_{13}	1508	88	132704	0.6615	0.6823	4.9×10^{-6}	2.1272

Table 3. Experiment results for arabidopsis thaliana expression data.

Bicluster No.	Genes	Conditions	Volume	MSR	Gene Variance	CI	Fitness
BC_1	373	34	12682	1299.2	1304.0	0.1025	1299.20
BC_3	381	29	11049	1253.0	1257.8	0.1134	1253.00
BC_6	374	36	13464	1211.5	1215.3	0.0899	1211.50
BC_8	361	37	13357	1242.5	1246.6	0.0930	1242.50
BC_{17}	375	29	10875	1228.2	1231.0	0.1129	1228.20

Table 4. Experiment results for saccharomyces cerevisiae cell expression data.

Bicluster No.	Genes	Conditions	Volume	MSR	Gene Variance	CI	Fitness
BC_1	1372	6	8232	579.08	614.79	0.0703	579.08
BC_3	1461	6	8766	620.13	650.41	0.0707	620.13
BC_5	1409	7	9863	630.33	660.96	0.0639	630.34
BC_7	1455	4	5820	622.74	677.65	0.1070	622.74
BC_{10}	1474	5	7370	566.06	606.68	0.0768	566.06

Table 5. Experiment results for rat CNS expression data.

Bicluster No.	Genes	Conditions	Volume	MSR	Gene Variance	CI	Fitness
BC_1	56	4	224	0.8874	1.0853	0.0039	1.8088
BC_2	60	5	300	0.8680	1.0838	0.0028	1.7907
BC_8	64	3	192	0.7946	1.0342	0.0041	1.7616
BC9	63	6	378	0.9615	1.2201	0.0025	1.7811
BC_{15}	49	5	245	0.7917	0.9347	0.0032	1.8616

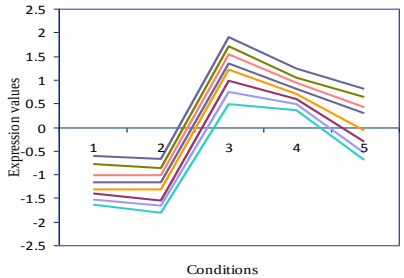

Figure 7. Small biclusters of size 8×5 for rat CNS expression data.

6. Conclusions

Through this work, CSM algorithm for biclustering microarray gene expression data is proposed. It focuses on finding maximum biclusters with lower MSR and higher gene variance. CS strategy is applied to find the optimal bicluster using mutation operator. Mutation operator avoids premature convergence. The CSM

outperforms the CS with Levy flight, BPSO and SFL. The overlapping rate of the bicluster is finding through Jaccard index. A qualitative measurement of the formed biclusters with a comparative assessment of results are provided on four benchmark gene expression datasets, namely yeast saccharomyces cerevisiae stress expression data, Arabidopsis thaliana expression data, yeast saccharomyces cerevisiae cell cycle expression data and rat CNS data to demonstrate the effectiveness of the proposed method.

Acknowledgment

This work is supported by the research grant from Department of Biotechnology (DBT), Ministry of Science and Technology, Government of India, New Delhi, Grant No. BT/PR3741/BID/382/2011.

References

[1] Ben-Dor A., Chor B., Karp R., and Yakhini Z., "Discovering Local Structure in Gene Expression Data: The Order-Preserving Submatrix Problem," *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 373-384, 2003.

[2] Bergmann S., Ihmels J., and Barkai N., "Iterative Signature Algorithm for the Analysis of Large-Scale Gene Expression Data," *Physical Review E*, vol. 67, no. 3, pp. 1-18, 2003.

[3] Bleuler S., Prelic A., and Zitzler E., "An EA Framework for Biclustering of Gene Expression Data," *in Proceeding of IEEE on Evolutionary Computation*, Portland, pp. 166-173, 2004.

[4] Cheng Y. and Church G., "Biclustering of Expression Data," *in Proceeding of the 8th International Conference on Intelligent Systems for Molecular Biology*, Menlo Park, pp. 93 -103, 2000.

[5] Cho J., Campbell J., Winzeler A., Steinmetz L., Conway A., Wodicka L., Wolfsberg G., Gabrielian E., Landsman D., and Lockhart J., "A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle," *Molecular Cell*, vol. 2, no. 1, pp. 65-73, 1998.

[6] Dawkins R., *The Selfish Gene*, Oxford University Press, 2006.

[7] Divina F. and Aguilar-Ruiz S., "Biclustering of Expression Data with Evolutionary Computation," *IEEE Transactions on Knowledge Data Engineering*, vol. 18, no. 5, pp. 590-602, 2006.

[8] Eusuff M., Lansey K., and Pasha F., "Shuffled Frog-Leaping Algorithm: A Memetic Meta-Heuristic for Discrete Optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129-154, 2007.

[9] Gasch A., Spellman P., Kao C., Carmel-Harel O., Eisen M., Storz G., Botstein D., and Brown P., "Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes," *Molecular Biology of the Cell*, vol. 11, no. 12, pp. 4241-4257, 2000.

[10] Huang Q., Tao D., Li X., and Liew A., "Parallelized Evolutionary Learning for Detection of Biclusters in Gene Expression Data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 2, pp. 560-570, 2012.

[11] Jaccard P., "The Distribution of the Flora in the Alpine Zone," *New Phytologist*, vol. 11, no. 2, pp. 37-50, 1912.

[12] Kennedy J. and Eberhart R., "A Discrete Binary Version of the Particle Swarm Algorithm," *in Proceeding of IEEE International Conference on Systems, Man and Cybernetics*, Orlando, pp. 4104-4108, 1997.

[13] Liu C., *Introduction to Combinatorial Mathematics*, McGraw-Hill Publication, 1968.

[14] Liu J., Li Z., Hu X., and Chen Y., "Biclustering of Microarray Data With Mospo Based on Crowding Distance," *Bioinformatics*, vol. 10, no. 4, 2009.

[15] Liu X. and Wang L., "Computing the Maximum Similarity Bi-Clusters of Gene Expression Data," *BMC Bioinformatics*, vol. 23, no. 1, pp. 50-56, 2007.

[16] Lockhart D. and Winzeler E., "Genomics, Gene Expression and DNA Arrays," *Nature*, vol. 405, no. 6788, pp. 827-836, 2000.

[17] Mansour N., Awad M., and El-Fakih K., "Incremental Genetic Algorithm," *The International Arab Journal of Information Technology*, vol. 3, no. 1, pp. 42-47, 2006.

[18] Roy S., Bhattacharyya D., and Kalita J., "CoBi: Pattern Based Co-Regulated Biclustering of Gene Expression Data," *Pattern Recognition Letters*, vol. 34, no. 14, pp. 1669-1678, 2013.

[19] Tanay A., Sharan R., and Shamir R., "Discovering Statistically Significant Biclusters in Gene Expression Data," *BMC Bioinformatics*, vol. 18, no. 1, pp. 136-144, 2002.

[20] Wen X., Fuhrman S., Michaels G., Carr D., Smith S., Barker J., and Somogyi R., "Large-Scale Temporal Gene Expression Mapping of Central Nervous System Development," *Proceeding of the National Academy of Sciences*, vol. 95, no. 1, pp. 334-339, 1998.

[21] Yang X. and Deb S., "Cuckoo Search via Levy Flights," *in Proceeding of the World Congress on Nature and Biologically Inspired Computing*, Coimbatore, pp. 210-214, 2009.

Balamurugan Rengeswaran is currently, working as a Senior Research Fellow for the DBT sponsored project at Bannari Amman Institute of Technology, Erode, Tamil Nadu, India. He received his BE and ME degrees in Computer Science and Engineering (CSE) from Anna University, Chennai. His areas of interest include data mining and optimization techniques.

Natarajan Mathaiyan is currently, working as Chief Executive at Bannari Amman Institute of Technology, Erode, Tamil Nadu, India. He received BE, MSc and PhD degrees from the PSG College of Technology, Coimbatore, India. He has more than 40 years of experience in Academic-Teaching, Research and Administration. He had published more than 110 papers in National and International Journals and He authored and published 10 Books. His research areas of interest include data mining, image processing and soft computing.

Premalatha Kandasamy is currently, working as a Professor in the Department of Computer Science and Engineering at Bannari Amman Institute of Technology, Erode, Tamil Nadu, India. She completed her PhD in Computer Science and Engineering (CSE) at Anna University, Chennai, India. She did her ME and BE degrees in CSE at Bharathiar University, Coimbatore, Tamil Nadu, India. Her research interests include data mining, image processing, information retrieval and soft computing.