



A New Hybrid Cuckoo Search Algorithm for Biclustering of Microarray Gene-Expression Data

R. Balamurugan, A.M. Natarajan & K. Premalatha

To cite this article: R. Balamurugan, A.M. Natarajan & K. Premalatha (2018) A New Hybrid Cuckoo Search Algorithm for Biclustering of Microarray Gene-Expression Data, Applied Artificial Intelligence, 32:7-8, 644-659, DOI: [10.1080/08839514.2018.1501918](https://doi.org/10.1080/08839514.2018.1501918)

To link to this article: <https://doi.org/10.1080/08839514.2018.1501918>



Published online: 26 Jul 2018.



Submit your article to this journal [↗](#)



Article views: 249



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 1 View citing articles [↗](#)



A New Hybrid Cuckoo Search Algorithm for Biclustering of Microarray Gene-Expression Data

R. Balamurugan ^a, A.M. Natarajan^b, and K. Premalatha^c

^aSchool of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India;

^bDepartment of Computer Science and Engineering, Bannari Amman Institute of Technology,



Erode, India; ^cDepartment of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, India

ABSTRACT

Biclustering in gene-expression data is a subset of the genes demonstrating consistent patterns over a subset of the conditions. Recently, the most of research in biclustering involving statistical and graph-theoretic approaches by adding or deleting rows and/or columns in the data matrix based on some constraints. This is an exhaustive search of the space, and hence the solutions may not be feasible. The proposed work finds the significant biclusters in large expression data using shuffled cuckoo search with Nelder–Mead (SCS-NM). The diversification and intensification of the search space are obtained through shuffling and simplex NM, respectively. The proposed work is tested on four benchmark datasets, and the results are compared with the swarm intelligence techniques and the various biclustering algorithms. The results show that there is significant improvement in the fitness value of proposed work SCS-NM. In addition, the work determines the biological relevance of the biclusters with Gene Ontology in terms of function, process and component.

Introduction

The DNA microarray analysis is a technology which enables the researchers to analyze the expression level of thousands of genes in a single reaction rapidly and in an efficient manner (Lockhart and Winzeler 2000). A typical DNA microarray analysis involves a multistep procedure which includes fabrication of microarrays by fixing properly designed oligonucleotides representing specific genes, hybridization of complementary DNA (cDNA) populations onto the microarray, scanning hybridization signals, image analysis and normalization of data. After a number of preprocessing steps, the low-level microarray analysis of a microarray can be represented as a numerical matrix. In this matrix, the rows represent different genes and columns represent experimental conditions. Each element of this matrix represents the

CONTACT R. Balamurugan  balacse05@gmail.com; r.balamurugan@vit.ac.in  School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

expression level of a gene under a specific condition, and is represented by a real number. In gene-expression matrix, a common goal is to group the genes and conditions into subsets that convey biological significance. In its most common form, this task translates to the computational problem known as clustering.

However, clustering has some disadvantages (Madeira and Oliveira 2004). To overcome the problems associated with clustering, biclustering was proposed. Biclustering is a powerful analytical tool for the biologist. A bicluster is a submatrix of the gene-expression data matrix. The rows and columns in the submatrix need not be contiguous as in the gene-expression data matrix (Madeira and Oliveira 2004). The computation of biclusters is costly because one will have to consider all the combinations of columns and rows in order to find out all the biclusters. The search space for the biclustering problem is 2^{m+n} where m and n are the number of genes and conditions, respectively. Usually $m + n$ is more than 2000. Therefore, the biclustering problem is NP-hard (Divina and Aguilar-Ruiz 2006; Tanay, Sharan, and Shamir 2009). The problem of finding a coherent bicluster can be formulated as an optimization problem. For stochastic algorithms, in general, there are two types—heuristic and meta-heuristic—though their difference is small. Loosely speaking, heuristic means “to find” or “to discover by trial and error” (Julio and Michael 1997). This is good when we do not necessarily want the best solutions but rather good solutions which are easily reachable. The Nelder–Mead (NM) downhill simplex is an example of heuristic algorithm. Further development over the heuristic algorithms is the so-called meta-heuristic algorithms. All meta-heuristic algorithms use certain trade-off of randomization and local search. Randomization provides a good way to move away from local search to the search on the global scale. Therefore, almost all meta-heuristic algorithms intend to be suitable for global optimization (Christian and Andrea 2003).

This work develops and implements the biclustering based on the most popular and robust bio-inspired strategy cuckoo search (CS) (Yang and Deb 2009). An important advantage of CS algorithm is its simplicity. In fact, comparing with other population- or agent-based meta-heuristic algorithms such as particle swarm optimization and harmony search, CS has fewer parameters that need to be tuned before starting the search compared with other techniques (apart from the population size). Therefore, it is very easy to implement. In the conventional CS, each nest consists of a single egg and cuckoo imitates an egg using Levy flight. In the proposed CS algorithm, a nest contains a clutch (three eggs) instead of single egg. To avoid the premature convergence, the cuckoo imitates an egg using the NM (Nelder and Mead 1965) approach and to obtain near global optimum, the eggs within the nests are shuffled when the stagnation occurs. In this study, shuffled cuckoo search with Nelder–Mead (SCS-NM) is used for biclustering microarray gene-expression data. The remainder of this

paper is organized as follows: Section 2 provides the problem statement. Section 3 gives related works in biclustering. The SCS-NM is illustrated in Section 4. Kennedy and Eberhart proposed a discrete binary version of binary particle swarm optimization (BPSO) for binary problems (Kennedy and Eberhart 1997). The shuffled frog leaping (SFL) algorithm is a memetic meta-heuristic that is designed to seek a global optimal solution by performing a heuristic search. It is based on the evolution of memes carried by individuals and a global exchange of information among the populations (Eusuff, Lansey, and Pasha 2006). Section 5 presents the detailed experimental setup and results for comparing the performance of the SCS-NM with the BPSO, SFL, CS and CS-NM.

Problem Statement

The gene-expression data can be shown as $N \times M$ matrix A of real numbers. Let G be a set of genes, C a set of conditions, and $A(G, C)$ the expression matrix, where $G = \{1, 2, \dots, m\}$ and $C = \{1, 2, \dots, n\}$. The element $GEx_{i,j}$ of $A(G, C)$ represents the expression level of gene ' i ' under condition ' j '. The objective of biclustering is to extract the submatrix $A(G', C')$ of $A(G, C)$, which is identified by gene subset G' of G and condition subset C' of C . In general, the problem can be defined as finding large sets of rows and columns such that the rows show unusual similarities along the dimensions characterized by columns and vice versa. The bicluster cardinality or volume of bicluster is simply the product of the number of genes and number of conditions in the bicluster. The main objective here is to identify the biclusters of maximum size with the minimum mean squared residue (MSR) (homogeneity) and maximum of row variance (nontrivial).

Review of Related Works

As we mentioned in the introduction of this paper, the biclustering problem is NP-hard (Tanay, Sharan, and Shamir 2009). For that reason, heuristic search algorithms are usually used to approximate the problem by finding suboptimal solutions. A number of biclustering techniques have been proposed in the literature for gene-expression data analysis. Table 1 presents the merits and demerits of the various existing biclustering method.

Shuffled Cuckoo Search with Nelder-Mead

In general, the performances of the meta-heuristic algorithms are mainly dependent on two properties of the algorithm: diversification and intensification, also mentioned as exploration and exploitation (Yang et al. 2013). Although the basic CS algorithm demonstrates good global optimal search ability in optimization problems, it has the problem of premature

Table 1. Review of related work in biclustering gene expression data.

S. No.	Algorithm	Merits	Demerits
1.	CC (Cheng and Church 2000)	The algorithm discovers biclusters with coherent values.	CC discovers one bicluster at a time, repeated application of the method on a modified matrix is needed for discovering multiple biclusters. Therefore, it results in highly overlapping gene sets.
2.	SAMBA (Tanay, Sharan, and Shamir 2009)	Significant biclusters were identified using graph theoretic approach simultaneously.	The algorithm is based on exhaustive enumeration of biclusters. Due to its high complexity, the number of rows the bicluster may have is restricted.
3.	xMOTIFs (Murali and Kasif 2003)	In order to prevent the algorithm from finding too small or too large bicluster, some constraints on their size, conservation, and maximality have been added to its formal definition.	The algorithm uses prior knowledge about the sample phenotypes.
4.	ISA (Bergmann, Ihmels, and Barkai 2003)	The method includes data normalization and the use of thresholds that determine the resolutions of the different transcription modules.	There is no evaluation of the statistical significance. Additionally, two threshold parameters should be defined.
5.	OPSM (Ben-Dor et al. 2003)	The algorithm can also be used to discover more than one bicluster in the same dataset, even when they are overlapped.	The model concerns only the order of values and thus makes the model quite restrictive.
6.	FLOC (Yang et al. 2003)	Deals with the Cheng and Church random masking issue.	Minimum overall coverage of bicluster in the dataset.
7.	BIMAX (Prelic et al. 2006)	At each step, the two partitioned matrices may have elements in common or not, allowing thus the possibility of finding overlapped biclusters.	Divide and conquer has the drawback of possibly missing good biclusters by early splits.
8.	MOEA (Mitra and Banka 2006)	The output consists of a large size of bicluster to a given threshold.	Converges slowly and consumes much time to find the best bicluster and no overlapping is carried out.
9.	SEBI (Divina and Aguilar-Ruiz 2006)	Matrix of weights is used for the control of overlapped elements among the different solutions.	Maximum similarity bicluster (MSB) works well for the special case of approximately small biclusters.
10.	MSB (Liu and Wang 2007)	(1) No discretization procedure is required, (2) Performs well for overlapping biclusters and works well for additive biclusters.	MSB works for the special case of approximately squares biclusters.
11.	RWB (Angiulli, Cesario, and Pizzuti 2008)	In order to avoid getting trapped into poor local minima, the algorithm executes random moves according to a probability given by the user.	Randomness reduces the convergence speed for large dataset. So it consumes much time to find the best bicluster.
12.	CMOPSOB (Liu et al. 2009)	Speed up the convergence to the Pareto front and also guarantee diversity of solutions.	Using PSO has the problems of dependency on initial point and parameters, difficulty in finding their optimal design parameters, and the stochastic characteristic of the final outputs.

(Continued)

Table 1. (Continued).

S. No.	Algorithm	Merits	Demerits
13.	BicFinder (Ayadi et al, 2012a)	Do not require fixing a minimum or a maximum number of genes or conditions, enabling a generation of diversified biclusters.	Place the restrictive constraints on the structure of the biclustering solutions.
14.	PDNS (Ayadi et al, 2012b)	It works well for shifting and scaling pattern of expression value.	No overlapping control is carried out among the reported solutions.
15.	CoBi (Roy, Bhattacharyya, and Kalita 2013)	Particularly, it is used for grouping both positively and negatively regulated genes from microarray expression data.	It extracts small volume of biclusters for large MSR value.
16.	MBA (Ayadi, Elloumi, and Hao 2014)	It is used for grouping both positively and negatively regulated genes from microarray expression data.	It extracts small-size biclusters, and the quality of biclusters depends upon the threshold value
17.	EBACross (Maatouk et al. 2014)	Increase in the diversification of solution can be achieved by a mutation operator.	Computational cost is high on large inputs.
18.	BiBin Max (Saber and Elloumi 2015)	Fast retrieval method.	Too many parameter setting may affect the overall performance.
19.	UniBic (Wang, Li, and Robinson 2016)	Biologically meaningful trend-preserving biclusters can be detected.	It returns small volume of biclusters.
20.	SEB (Yin and Liu 2017)	Efficient and scalable in terms of the biological significance and runtime.	SEB works well for the special case of approximately small biclusters.

convergence. Therefore, the CS is improved by balanced intensification and diversification. This paper proposes a variant CS called SCS-NM. The traditional CS considers single egg in a nest and a cuckoo lays one egg at a time by using Levy flight (Yang and Deb 2009). The proposed CS considers a clutch which contains three eggs in each nest. So the population is partitioned into several clutches which are evolved independently. To ensure that the evolution process is competitive, it is required to have higher probabilities that better solutions contribute to the next generation. The use of a triangular probability distribution ensures this fairness. The NM simplex algorithm, a direct search method, is used to generate the new solution. This strategy uses the information contained in the clutches to direct the evolution in an improved direction (Nelder and Mead 1965). Every new solution replaces the worst solution of the current clutch, rather than the worst solution of the entire population. This substitution ensures that every member has at least an opportunity to evolve before being discarded or replaced. Thus, none of the information contained in the nest is ignored. The intensification is caused by while using simplex method.

For high-dimensional data, the local minima has a severe effect on fitness function value so that the global minimum is not well approximated. The CS is said to be converge prematurely when the proposed solution approximates

a local rather than global minimum. The objective of the proposed work is when the solutions have prematurely converged due to stagnation, it shuffles the eggs in a new search space. The purpose of shuffling is to increase the diversity of the population (Yang et al 2013). After the certain number of evolutions, the best solution does not change. The solution has converged to local optimum of the objective function. Therefore, shuffling has a good performance to solve the CS drawbacks. In this regard, all the best solutions (eggs) are sorted in an ascending order according to their fitness. Then, the eggs are partitions or shuffle into the nest, i.e. rank 1 goes to nest 1, rank 2 goes to nest 2, rank 3 goes to nest 3, rank 4 goes to nest 1 and so on. So, the new clutches are formed through this process of shuffling. This strategy helps to improve the solution by sharing the information and properties independently gained by each clutch. Therefore, avoid trapping the local optimal solution. The SCS-NM maintains the balanced intensification and diversification via the process of NM simplex and shuffling in the solution of the search space respectively. The Algorithm 1 for SCS-NM is given as follows:

Algorithm 1. Pseudo code for SCS-NM.

Input: Number of nest n , Discovery rate of alien solutions p_a , Maximum number of iteration $MaxIter$

Output: coherent biclusters

Generate random population with n nests and each nest consists of 3 eggs (clutch).

while ($t < MaxIter$)

Get a cuckoo (say i) randomly and generate a solution using Nelder–Mead

Choose a nest among n (say, j) randomly;

Replace worst egg in j by the new solution i ;

A fraction (p_a) 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;

if stagnation

Sort the eggs by increasing order

Partition or shuffle the egg into the nest, i.e. rank 1 goes to nest 1, rank 2 goes to nest 2, rank 3 goes to nest 3, rank 4 goes to nest 1, and so on.

end while

Arrange the best solution of individual nest in ascending order present the best solution of each nest.

Fitness Function

In order to measure the coherence of bicluster, Cheng and Church (2000) introduced the concept of the MSR. 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 submatrix A_{IJ} equals

$$r(a_{i,j}) = a_{i,j} + a_{I,J} - a_{I,j} - a_{i,J} \quad (1)$$

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 quality of a bicluster can be evaluated by computing the MSR, i.e. the sum of all the squared residues of its elements is as per (2)

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I} \sum_{j \in J} r(a_{i,j})^2 \quad (2)$$

Low MSR value denotes strong coherence in the bicluster. This may include 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. Cheng and Church used row variance as an accompanying score to find out trivial biclusters. The row variance can be represented in Equation (3) as follows:

$$\begin{aligned} \text{Var}_r(I, J) &= \frac{1}{|I|} \sum_{i \in I} v_r(i) \quad (3) \\ v_r(i) &= \frac{1}{|J|} \sum_{j \in J} (a_{i,j} - a_{i,J})^2 \end{aligned}$$

Our goal is to obtain biclusters with the maximum number of genes and conditions and with the minimum value of $f(I, J)$. The fitness function for obtaining bicluster is defined in Equation (4) as follows:

$$f(I, J) = H(I, J) + \frac{1}{\text{Var}(I, J)} \quad (4)$$

Experimental Results and Analysis

The proposed algorithm presented for the bicluster problem is coded in MATLAB R2012a and run on an Intel i3 3.7 GHz. The biclustering algorithm has been applied to four sets in order to study its performance, namely the yeast *Saccharomyces cerevisiae* stress expression data (Gasch et al. 2000), *Arabidopsis thaliana* expression data (Bleuler, Prelic, and Zitzler 2014), yeast *Saccharomyces cerevisiae* cell-cycle expression data (Cho et al. 1998) and rat Central Nervous System (CNS) expression data (Wen et al. 1998) are used. Table 2 shows the description of dataset used in this paper. The parameters p_w , α , and λ are set as 0.25, 1, and 1.5, respectively (Yang and Deb 2009). Through empirical analysis,

Table 2. Dataset description.

Dataset name	Genes	Samples
Yeast <i>Saccharomyces cerevisiae</i> stress data	2993	173
<i>Arabidopsis thaliana</i> expression data	734	69
Yeast <i>Saccharomyces cerevisiae</i> cell-cycle data	2884	17
Rat CNS expression data	112	9

the population size and the number of iterations are set as 20 and 200, respectively.

Figures 1–4 show the fitness value obtained for *Saccharomyces cerevisiae* stress expression data, *Arabidopsis thaliana* expression data, yeast *Saccharomyces cerevisiae* cell-cycle expression data and rat CNS expression data, respectively. Through careful observation, it can be seen that SCS-NM

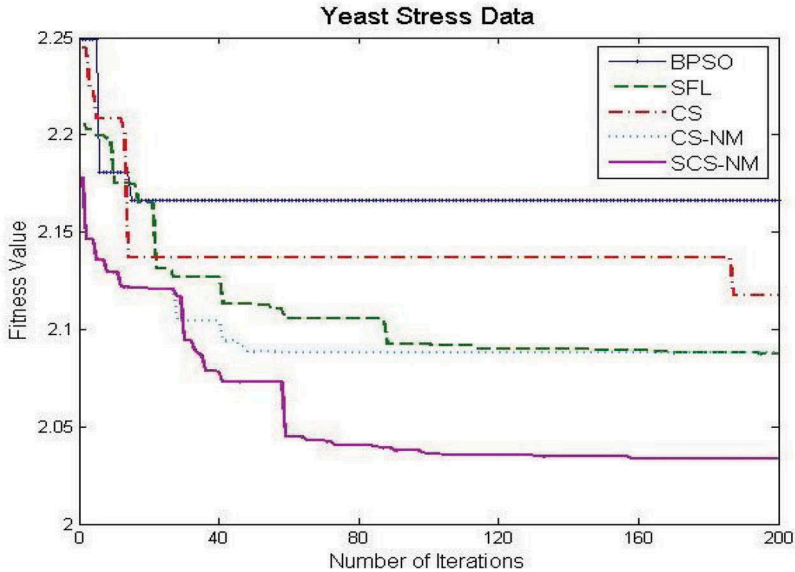


Figure 1. Plot of number of iterations versus fitness value for yeast stress data.

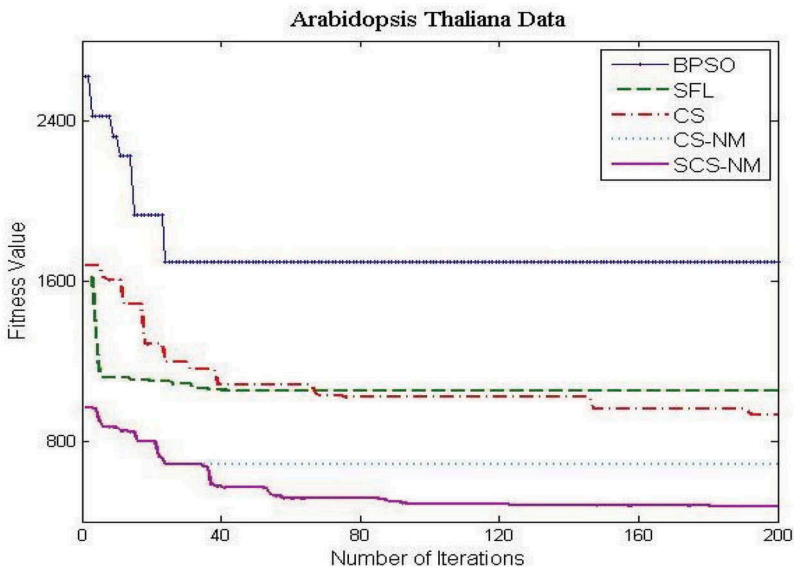


Figure 2. Plot of number of iterations versus fitness value for *Arabidopsis thaliana* data.

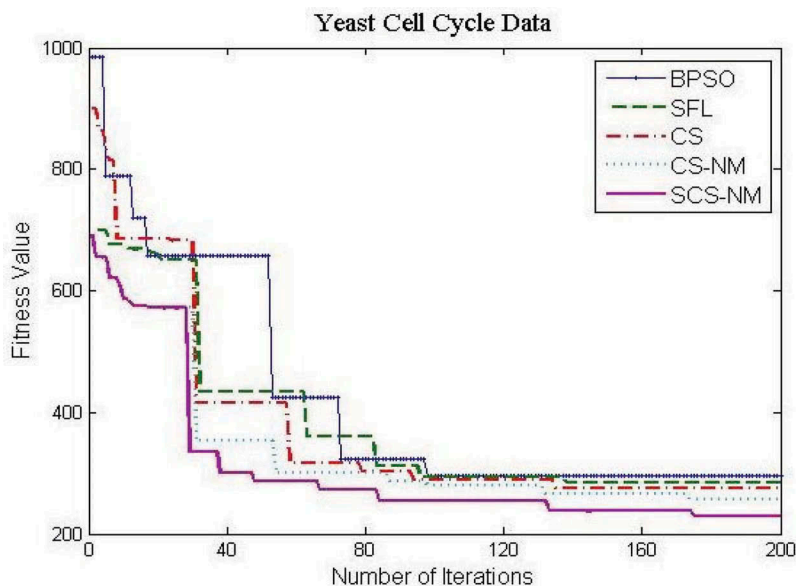


Figure 3. Plot of number of iterations versus fitness value for yeast cell-cycle data.

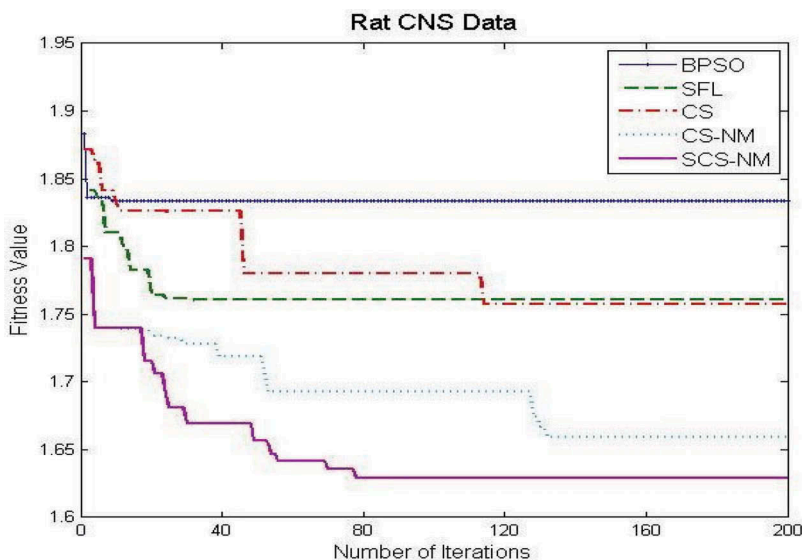


Figure 4. Plot of number of iterations versus fitness value for rat CNS data.

fitness gets down rapidly in the initial stage of the evolution. The BPSO algorithm has premature convergence due to high stagnation. The SFL performs better on yeast stress expression data and the remaining three datasets CS give better performance than SFL. However, CS and SFL have roughly the same convergence speed. In addition, it is obvious to infer that SFL and BPSO get stuck at local optima quickly as can be seen from Figure 1. For all

Table 3. Experiment results for *Saccharomyces cerevisiae* cell expression data.

Bicluster number	Genes	Conditions	Volume	MSR	Gene variance	CI	Fitness
BC ₁	1487	4	5948	181.69	1371.89	0.0305	181.69
BC ₄	1520	5	7600	223.45	1249.52	0.0290	223.45
BC ₈	1451	7	10,157	248.11	1264.74	0.0284	248.11
BC ₁₅	1473	8	11,784	261.75	1290.42	0.0222	261.75
BC ₅	1402	10	14,020	285.24	1364.41	0.0219	285.24

the datasets, the proposed work SCS-NM outperforms all other algorithms because the exploration and exploitation are made with shuffling and simplex method. CS-NM performs better than BPSO, CS, and SFL. Moreover, SCS-NM converges to the global optimum rapidly. It frequently gives significant improvements in the first few iterations and quickly produces quite satisfactory results.

According to the problem formulation, the size of an extracted bicluster should be as large as possible while satisfying a homogeneity criterion. The bicluster should satisfy two requirements simultaneously. The expression levels of each gene within the bicluster should be similar over the range of conditions. 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 and gene variance removes the simple bicluster. To quantify biclusters, homogeneity and size should satisfy the coherence index (CI) which is used as a measure for evaluating their goodness (Mitra and Banka 2006). CI is defined as the ratio of MSR score to the size of the formed bicluster. Table 3 shows the sample experimental results obtained for yeast *Saccharomyces cerevisiae* cell-cycle expression data and the biclusters are chosen randomly from 20 biclusters. In this table, the first column contains the label of each bicluster. The second and third columns report the number of rows (genes) and number of columns (conditions) of the bicluster, respectively. The fourth column reports the volume of the bicluster and the fifth column contains the MSRs of the biclusters. The sixth and seventh columns report the row variance and CI of the bicluster, respectively. The last column contains the fitness of the biclusters. The MSR maximum limit is 300. The largest size bicluster is found at MSR = 285.24, with CI being minimal and indicating the goodness of the discovered partitions. The minimum value of CI is 0.0219, with a corresponding size of 14,020 being the best in the table. As mentioned earlier, a low MSR indicates a high coherence of the discovered biclusters. Figure 5 shows clearly a small bicluster of size 8×5 for *Arabidopsis thaliana* data.

Comparative Analysis

Table 4 shows a comparison summary of results obtained by various biclustering algorithms for the yeast cell-cycle dataset. The MSR value of biclusters obtained

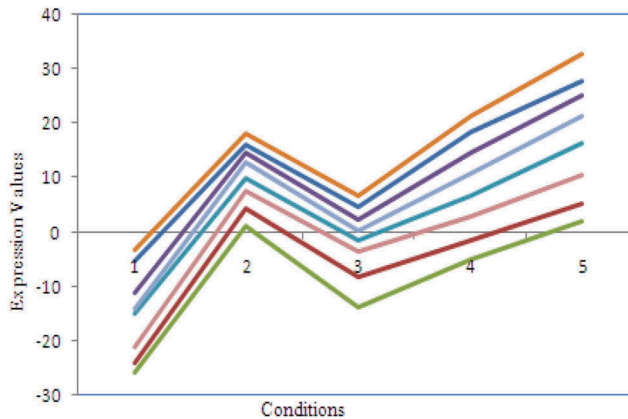


Figure 5. Plot of sample biclusters of size 8×5 for *Arabidopsis thaliana* data.

Table 4. Comparative analysis on yeast cell expression data.

Method	Average MSR	Average volume	Average number of genes	Average number of conditions	Largest volume
FLOC	187.44	1825.78	195.00	12.20	2000
DBF	114.70	1627.20	188.00	11.00	4000
CC	204.29	1576.98	167.00	12.00	4485
Single-objective GA	52.87	570.86	191.12	5.13	1408
SEBI	205.18	209.92	13.61	15.25	1394
MOEA	234.87	10,301.71	1095.43	9.29	14,828
CS	286.24	10,925.10	1281.67	7.63	14,586
CS-NM	254.45	11,963.78	1295.88	7.79	14,996
SCS-NM	229.15	12,387.44	1356.50	8.20	15,012

An italic and bold font represents significance of the proposed method.

by all the algorithms listed in Table 4 and the maximum limit is 300. The performance of SCS-NM is compared with flexible overlapped biclustering (FLOC), deterministic biclustering with frequent pattern mining (DBF), Cheng and Church (CC) and single-objective genetic algorithm (GA) on yeast cell-cycle dataset by Mitra and Banka (2006) and the algorithm sequential evolutionary biclustering (SEBI) by Divina and Aguilar-Ruiz (2006). FLOC uses a probabilistic approach to find biclusters. Even it extracts only half of the average volume of DBF for an average MSR of 187.44. DBF finds 100 biclusters, with half of these lying in the size range 2000–3000 and a maximum size of 4000. Similarly, CC algorithm gives a fractional volume of biclusters. Single-objective GA has also been used with local search to generate considerably overlapped biclusters. It is observed that a population size of 50 leads to the generation of a largest bicluster of size 1408. This is less than the bicluster size generated by all other algorithms. SEBI extracts only an average of 13 genes for average MSR of 205. On the other hand, it could find the biclusters of average genes are less than the set of conditions. Multi-objective evolutionary algorithm (MOEA gives maximum volume with the minimum MSR score. However, there is no

overlapping carried out. Next CS method returns the largest bicluster; however, average MSR of CS is larger than MSR of MOEA. Eventually, the SCS-NM method extracts the largest bicluster of size 15,012 with average MSR of 229.15 as per the objective. In the case of SCS-NM, largest bicluster size as well as average volume is better than that of all other algorithms. Even so, MSR value is not better than that of all other algorithms because it extracts more than 60% average volume of FLOC. It is better than all other methods in all aspects except in the size of samples.

Biological Analysis of Biclusters

The proposed work determines the biological relevance of the biclusters found by SCS-NM on the Gasch yeast dataset in terms of the statistically significant GO annotation database. The degree of enrichment is measured by p values which use a cumulative hypergeometric distribution to compute the probability of observing the number of genes from a particular GO category (function, process, and component) within each bicluster. The p value is the probability that the genes are selected into the cluster by random. A small p value implies that the cluster is highly differed found by chance. The annotations of genes for three ontologies including biological process, cellular component, and molecular function are obtained. With the intention of evaluating the biological relevance of SCS-NM algorithm, the results of the proposed method are compared with CC, ISA, Bimax, OPSM and BiMine on yeast cell-cycle dataset from Ayadi, Elloumi, and Hao (2009) by using web-tool of FuncAssociate (Berriz et al. 2003). The FuncAssociate computes the adjusted significance scores for each bicluster. Indeed, the adjusted significance scores assess genes in each bicluster by computing adjusted p values, which indicates how well they match with the different GO categories. Note that a smaller p value, close to 0, is indicative of a better match. Figure 6 represents the different values of significant p values for each algorithm over the percentage of total extracted biclusters. In fact with SCS-NM, 100% of tested biclusters have a p value = 5%. The same result is obtained with a p value of 1%. Finally, 75% of extracted biclusters with SCS-NM have a p value = 0.001%, while those of CS-NM and CS have 67% and 60%, respectively. We note that SCS-NM performs well for 0.001% p values compared to CC, ISA, Bimax and OPSM and it performs well for all cases of p value (p value = 5%, p value = 1%, p value = 0.5%, p value = 0.1%, and p value = 0.001%).

Biological Annotation for *Saccharomyces cerevisiae* using GOTermFinder Toolbox

In order to identify the biological annotations for the biclusters, we use GOTermFinder which is tool available in the *Saccharomyces* Genome Database (SGD). GOTermFinder is designed to search for the significant

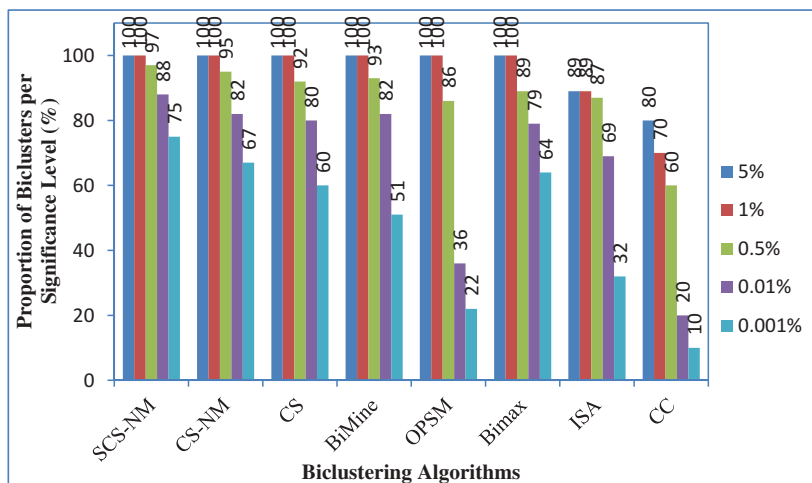


Figure 6. Plot of proportions of biclusters significantly enriched by GO annotations on yeast cell-cycle data.

shared GO terms of the groups of genes and provides users with the means to identify the characteristics that the genes may have in common. Table 5 lists the significant shared GO terms used to describe the set of genes in each bicluster for the process, function, and component ontologies. Only the most significant terms are shown. For example, in the bicluster BC_1 , the genes are mainly involved in binding activity. The tuple ($n = 517$, $p = 2.06 \times 10^{-9}$) represents that out of 1487 genes in bicluster BC_1 , 517 genes belong to binding activity function, and the statistical significance is given by the p value of $p = 2.06 \times 10^{-9}$. Figure 7 shows the biological network of the bicluster with 10 genes; the false discovery rate (FDR) is very low (0.0003) and it is zero in many occasions. Further, the corresponding p value is very small ($p = 0.00042$) which shows that there is a very less probability to obtain the gene cluster in random. These results mean that the proposed SCS-NM biclustering approach can find biologically meaningful biclusters.

Table 5. Significant GO terms for three biclusters on *Saccharomyces cerevisiae* data.

Bicluster number	Number of genes	Process	Function	Component
BC_1	1487	Cellular component organization ($n = 685$, $p = 7.15 \times 10^{-33}$)	Binding activity ($n = 517$, $p = 2.06 \times 10^{-9}$)	Nuclear part ($n = 372$, $p = 7.89 \times 10^{-19}$)
BC_4	1520	Cellular process ($n = 1316$, $p = 3.18 \times 10^{-126}$)	Structural molecule activity ($n = 294$, $p = 6.02 \times 10^{-24}$)	Cell part ($n = 1427$, $p = 7.16 \times 10^{-110}$)
BC_8	1451	Metabolic process ($n = 1124$, $p = 3.22 \times 10^{-101}$)	Hydrolase activity ($n = 299$, $p = 3.53 \times 10^{-29}$)	Intracellular part ($n = 1312$, $p = 1.26 \times 10^{-93}$)

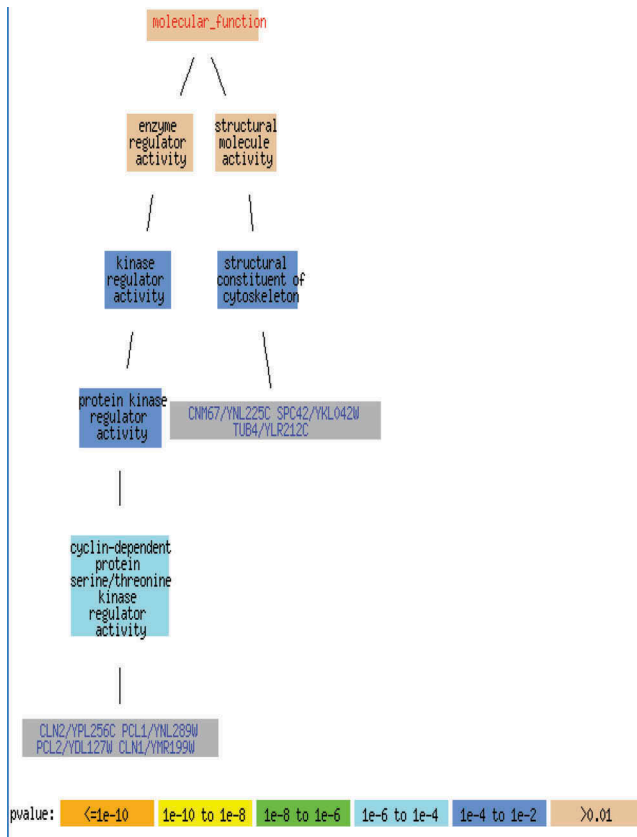


Figure 7. Plot of Gene Ontology biological process of yeast cell-cycle data (10 genes).

Conclusions

In this work, SCS-NM 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 in which the exploration and exploitation of the search space are controlled and balanced through shuffling and simplex local search, respectively. Hence, SCS-NM outperforms the BPSO, SFL, CS-NM, and CS with Levy flight and the different biclustering methods. Moreover, the SCS-NM algorithm maintains its stochastic behavior capacity better than the BPSO and SFL algorithms while searching for the global optimum value. A qualitative measure of the formed biclusters with a comparative assessment of results are provided on four benchmark gene-expression datasets to demonstrate the effectiveness of the proposed method. Biological validation of the selected genes within the biclusters has been provided by publicly available GO consortium. The patterns present a significant biological relevance in terms of related biological processes, components, and molecular functions in a species-independent manner.

ORCID

R. Balamurugan  <http://orcid.org/0000-0003-1348-2291>

References

- Angiulli, F., E. Cesario, and C. Pizzuti. 2008. Random walk biclustering for microarray data. *Information Science* 178 (6):1479–97. doi:10.1016/j.ins.2007.11.007.
- Ayadi, W., M. Elloumi, and J. Hao. 2009. A biclustering algorithm based on a bicluster enumeration tree: Application to DNA microarray data. *BioData Mining* 2:1–9. doi:10.1186/1756-0381-2-1.
- Ayadi, W., M. Elloumi, and J. Hao. 2012a. Pattern-driven neighborhood search for biclustering microarray data. *BMC Bioinformatics* 7:452–66.
- Ayadi, W., M. Elloumi, and J. Hao. 2012b. BicFinder: A biclustering algorithm for microarray data analysis. *Knowledge Information System* 30:341–58. doi:10.1007/s10115-011-0383-7.
- Ayadi, W., M. Elloumi, and J. Hao. 2014. A memetic algorithm for discovering negative correlation biclusters of DNA microarray data. *Neurocomputing* 145:14–22. doi:10.1016/j.neucom.2014.05.074.
- Ben-Dor, A., B. Chor, R. Karp, and Z. Yakhini. 2003. Discovering local structure in gene expression data: The order-preserving submatrix problem. *Journal of Computational Biology* 10:373–84. doi:10.1089/10665270360688075.
- Bergmann, S., J. Ihmels, and N. Barkai. 2003. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical Review E* 67:1–18. doi:10.1103/PhysRevE.67.031902.
- Berriz, G. F., O. D. King, B. Bryant, C. Sander, and P. Frederick. 2003. Characterizing gene sets with FuncAssociate'. *BMC Bioinformatics* 19:2502–04. doi:10.1093/bioinformatics/btg363.
- Bleuler, S., A. Prelic, and E. Zitzler. 2014. An EA framework for biclustering of gene expression data. *Proceeding Congress of IEEE on Evolutionary Computation* 32:166–73.
- Blum, C., and A. Roli. 2003. Metaheuristics in combinatorial optimization: overview and conceptual comparison. *Journal Acm Computing Surveys* 35 (3):268–308. doi:10.1145/937503.
- Cheng, Y., and G. M. Church. 2000. Biclustering of expression data'. Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, Menlo Park, United States, 93–103.
- Cho, R. J., M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, and D. J. Lockhart. 1998. A genome-wide transcriptional analysis of the mitotic cell cycle'. *Molecular Cell* 2:65–73.
- Christian, B., and Andrea, R. 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *Journal ACM Computing Surveys*. 35 (3):268–308.
- Divina, F., and J. S. Aguilar-Ruiz. 2006. Biclustering of expression data with evolutionary computation. *IEEE Transactions on Knowledge Data Engineering* 18:590–602. doi:10.1109/TKDE.2006.74.
- Eusuff, M. M., K. Lansey, and F. Pasha. 2006. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Engineering Optimization* 38:129–54. doi:10.1080/03052150500384759.
- Gasch, A. P., P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. 2000. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell* 11:4241–57. doi:10.1091/mbc.11.12.4241.

- Julio, R., and Michael, W. 1997. An optimization-based econometric framework for the evaluation of monetary policy. *NBER Macroeconomics Annual* 1997, 12:297–361.
- Kennedy, J., and R. C. Eberhart. 1997. A discrete binary version of the particle swarm Algorithm, IEEE international Conference on Systems, Man and Cybernetics, Washington, United States, 5, 4104–8.
- Liu, J., Z. Li, X. Hu, and Y. Chen. 2009. Biclustering of microarray data with mospo based on crowding distance. *Bioinformatics* 10:1–12.
- Liu, X., and L. Wang. 2007. Computing the maximum similarity bi-clusters of gene expression data. *BMC Bioinformatics* 23:50–56. doi:10.1093/bioinformatics/btl560.
- Lockhart, D. J., and E. A. Winzeler. 2000. Genomics, gene expression and DNA arrays. *Nature* 405:827–36. doi:10.1038/35015701.
- Maatouk, O., W. Ayadi, H. Bouziri, and B. Duval. 2014. Evolutionary algorithm based on new crossover for the biclustering of gene expression data. Proceedings of the Ninth International Conference on IAPR Stockholm, Sweden, 48–59.
- Madeira, S. C., and A. L. Oliveira. 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 1:24–45. doi:10.1109/TCBB.2004.2.
- Mitra, S., and H. Banka. 2006. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition* 39:2464–77. doi:10.1016/j.patcog.2006.03.003.
- Murali, T., and S. Kasif. 2003. Extracting conserved gene expression motifs from gene expression data. Pacific Symposium on Biocomputing, Boston University, United States, 77–88.
- Nelder, J. A., and R. Mead. 1965. A simplex method for function minimization. *Computer Journal* 7:308–13. doi:10.1093/comjnl/7.4.308.
- Prelic, A., Bleuler, S., Zimmermann, P., Buhlmann, P., Gruissem, W., and Hennig, L. 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* 22 (9):1122–1129.
- Roy, S., D. K. Bhattacharyya, and J. K. Kalita. 2013. CoBi: Pattern based co-regulated biclustering of gene expression data. *Pattern Recognition* 34:1669–78. doi:10.1016/j.patrec.2013.03.018.
- Saber, H. B., and M. Elloumi. 2015. Efficiently mining gene expression data via novel binary biclustering algorithms. *Journal of Proteomics & Bioinformatics* S9 (8). doi: 10.4172/jpb.S9-008.
- Tanay, A., R. Sharan, and R. Shamir. 2009. Discovering statistically significant biclusters in gene expression data. *BMC Bioinformatics* 18:136–44. doi:10.1093/bioinformatics/18.suppl_1.S136.
- Wang, Z., G. Li, and R. W. Robinson. 2016. UniBic: Sequential row-based biclustering algorithm for analysis of gene expression data'. *Scientific Reports* 6:23466. doi:10.1038/srep23466.
- Wen, X., S. Fuhrman, G. S. Michaels, D. B. Carr, S. Smith, J. L. Barker, and R. Somogyi. 1998. Large-scale temporal gene expression mapping of central nervous system development. *Proceedings of the National Academy of Sciences* 95:334–39. doi:10.1073/pnas.95.1.334.
- Yang, J., Wang, H., Wang, W., and Yu, P. 2003, 'Enhanced biclustering on expression data': proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering, pp. 321–327.
- Yang, X., Deb, S., and Fong, S. 2013. Metaheuristic Algorithms: Optimal Balance of Intensification and Diversification.
- Yang, X. S., and S. Deb. 2009. Cuckoo search via Levy flights. *Proceedings of World Congress on Nature & Biologically Inspired Computing* 210–14.
- Yin, L., and Y. Liu. 2017. Ensemble biclustering gene expression data based on the spectral clustering. *Neural Computing and Applications* 28:1–14.