MSR-based algorithms for biclustering of microarray gene expression data

R. Balamurugan* and S. P. Raja

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632 014, India

Biclustering plays a vital role in the analysis of gene expression data. The biclustering technique was proposed in the year 2000. For the past two decades, several biclustering methods and applications have been used to improve the quality to make sense of large microarray datasets. To find a highly correlated set of genes under specific conditions, usually one uses a measure or cost function. In such cases, it does not indicate that biclustering methods base their search on evaluation measures to identify the coherent biclusters. However, there is a substantial deviation between exploration in biclustering techniques and qualitative measure. Here, we present a review of different biclustering methods with the use of the most efficient measure called mean square residue within the search method. This review will guide researchers to fruitfully investigate their large microarray gene expression data and give meaningful, novel insights with greater efficiency.

Keywords: Biclustering, machine learning, mean square residue algorithm, microarray, optimization.

TECHNOLOGICAL improvement in the field of bioinformatics offers a complete opportunity to the researchers for the genome analysis of the living species[1]. DNA microarray technologies have made it feasible to observe the transcription levels of more than 10,000 genes in a single investigation. Figure 1 depicts the gene expression matrix. Usually, the outcome of the microarray technology is represented in a numerical matrix known as the two-dimensional data matrix. Rows and columns represent genes and samples respectively[2]. The column vector of a matrix is known as the expression pattern of the gene and the row vector as the expression profile of the sample. Each entry of this two-dimensional matrix refers to the expression level of a gene under a specific condition, and is denoted by an integer.

Machine learning techniques such as frequent pattern mining, classification and clustering play vital part to detect the set of similar gene expression profiles from microarray data. Clustering is the process of segmenting data points that have many dimensions (multi-dimensional data) into finite and novel disjoint groups[3]. In microarray gene expression data analysis, the process of unsupervised learning is one of the most utilized machine learning techniques for mining significant biological patterns[4]. Clustering of gene expression data helps us to find similar patterns underlying the genes over a set of samples, such as biological conditions. The ultimate aim is to detect the hidden pattern that shows the change in expression levels under specific conditions which include co-expressed gene groups. If a couple of rows depict correlated expression profiles across the columns, probably this reflects some kind of interaction and recommends a universal pattern of regulation. In microarray data analysis the cluster process can be organized into (i) gene-based clustering; (ii) condition-based clustering and (iii) biclustering (Figure 2).

The process of grouping a set of co-regulated genes is referred as gene-based clustering[5]. Here, genes and conditions are mapped into objects and features. Condition-based clustering is the clustering of the substructure of the condition under all the rows; it regards the conditions as objects and genes as the features. Conversely, most of the genes must be relevant only under a subset of samples. This is needed for several bioinformatics use cases; for example, the cellular processes are active only under a subset of conditions. Hence, a process needs to be grouped with a set of genes under a set of conditions concurrently. Thus, biclustering is a two-dimensional clustering problem where the

	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.

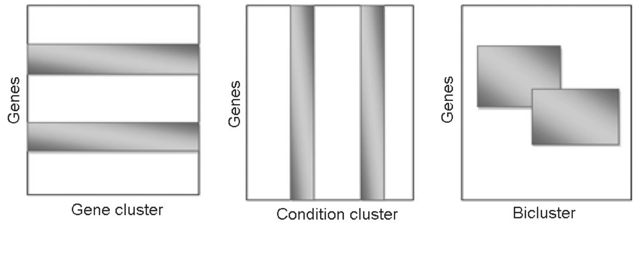


Figure 2. Types of microarray data clusters.

*For correspondence. (e-mail: balacse05@gmail.com) genes and conditions are grouped simultaneously. It has an excessive capacity of finding marker genes that are associated with certain tissues or diseases[6]. In 1970, Hartigan[7] introduced the term 'biclustering'; however, it was first applied to gene expression data analysis by Cheng and Church[8] in 2000. Biclustering is also called subspace clustering, two-way clustering, co-clustering or bi-dimensional clustering.

Extracting co-regulated genes under a specific group of samples from the large microarray data is a computationally intensive problem compared to clustering[9]. Moreover, it has been proven as an non-deterministic polynomial-time (NP)-hard problem[8]. Therefore, most of the biclustering methods for bicluster analysis are based on nature-inspired techniques such as swarm intelligence, evolutionary and multi-objective evolutionary frameworks. To do a global search in the solution space, the development of both a good fitness measure and a suitable heuristic method is needed for determining quality biclusters in an expression matrix. All of them use mean square residue (MSR) as the cost function. It is the most widely used measure for detecting coherent biclusters from the microarray expression data, and is the only metric used in more than 50% of approaches by different researchers since 2000. More than 40 techniques used MSR to find highly coherent biclusters. As per the biological myth, attentiveness are available in discovering two-dimensional clusters[10]. Therefore, to evaluate the subset of rows and columns simultaneously within the matrix as a correlated bicluster, MSR is being adopted.

Expression pattern-based bicluster structure

Different kinds of bicluster patterns have been described by Madeira and Oliveira[11] based on which genes are similar under the experimental conditions. They have identified four well-known bicluster patterns that in the gene expression matrix $m \times n$ (ref. 11). These are listed below.

Constant pattern: Consider a matrix which is referred as MIJ. This matrix is having the subsets of rows (genes) and subsets of columns (conditions)[12].

$$a_{ij} = \mu. \tag{1}$$

Constant pattern concerning rows (genes): Let a matrix M_{IJ} have a constant value on every row. It indicates that the expression levels vary from gene to gene. This pattern can be represented using additive and multiplicative expressions

$$a_{ij} = \mu + \beta_i \text{ or } a_{ij} = \mu \times \alpha_i. \tag{2}$$

Constant pattern concerning columns (conditions): Let a matrix M_{IJ} have a constant value on every column. It indicates that the expression levels vary from sample to sample. This pattern can be represented using additive and multiplicative expressions

$$a_{ij} = \mu + \beta_j \text{ or } a_{ij} = \mu \times \alpha_j, \tag{3}$$

Coherent pattern: In this type of bicluster pattern (either additive or multiplicative model), each row or column can be obtained by adding or multiplying a constant to another row or column. The following expression is used to obtain the resultant matrix

$$a_{ij} = \mu + \beta_i + \beta_j \text{ or } a_{ij} = \mu \times \alpha_i \times \alpha_j. \tag{4}$$

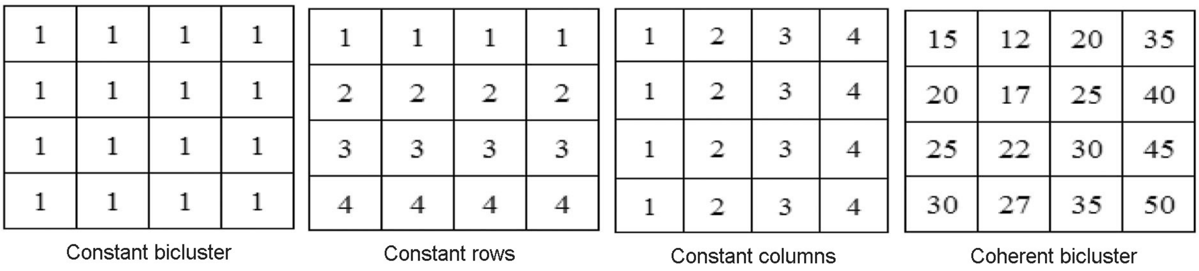
In eq. (4), $(1 \le j \le J)$ and $(1 \le i \le I)$ are denoted as a constant which is available in additive models for each row i and column j; similarly $(1 \le i \le I)$ and $(1 \le j \le J)$ are also a constant which used in multiplicative models. Moreover, if the rows of the matrix are upregulated or down-regulated under the columns irrespective of considering their actual expression values, then it is called a coherent evolution-based bicluster. Mathematically, this kind of bicluster model is difficult to express[12]. Figure 3 depicts the perfect bicluster patterns for the additive model.

Biclustering approaches based on evaluation measure – MSR

Mean squared residue (MSR) is termed as a subgroup of rows with a novel and hidden pattern across subsets of samples and it was implemented on the microarray data by Chenga and Church in 2000. Their objective was to extract submatrices from the large gene expression data with an MSR value lower than a given minimum constraint. So, they derived a naive greedy algorithm for finding the biclusters. It recursively deletes the row or column when the residual value is greater than a threshold. This approach finds a bicluster during the entire process.

To accelerate the biclustering process and address the random inference of the values in the data matrix, Yang *et al.*[13] were presented a probabilistic move-based algorithm named FLexible Overlapped biClustering (FLOC). This approach begins with some of the seeds which are called as initial biclusters and to find the coherent bicluster, this process continues until it meets the specific threshold. The algorithm performs two-dimensional clustering (row and column) iteratively, then it adopts the divide and conquer methodology. The row count in the bicluster is fixed during the execution. Hence larger biclusters cannot be considered.

Zhang *et al.*[14] proposed a scheme of deterministic biclustering with frequent pattern mining (DBF)[14] which is similar to FLOC. It is probabilistic algorithm that gives better accuracy than the method one introduced by Cheng and Church[8]. DBF is implemented in two phases. In the first phase, frequent pattern mining is used to find a group of highly correlated patterns. The inconsistent pattern between two consecutive conditions is represented as an item



1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Constant bicluster

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

Constant rows

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

Constant columns

15	12	20	35
20	17	25	40
25	22	30	45
30	27	35	50

Coherent bicluster

Figure 3. Types of bicluster patterns.

and each gene is considered as a transaction. Next, iteratively by adding one or more rows or columns, the size of the bicluster is enlarged without compromising the quality. The DBF algorithm finds the final bicluster in the deterministic time, whereas FLOC biclusters are nondeterministic.

Cheng and Church[8] have mentioned that finding a coherent bicluster from microarray data is a kind of NP-hard problem. Bleuler *et al.*[15] adopted the meta-heuristic natural evolutionary process in the biclustering algorithm. They first applied genetic algorithm (GA) to biclustering, whereby initialization of random solutions and the encoding of binary value for the chromosome representation are done. A general GA operator called environment selection is used to avoid any redundancy of the resulting biclusters. Reproduction operators such as uniform crossover and bit mutation are adopted and MSR is applied as an objective measure.

To overcome the random interference issue associated with the technique of Cheng and Church[8], Chakraborty[16] proposed an approach is called 'biclustering of gene expression data by simulated annealing'. This method differs from the Cheng and Church (CC) technique where rows and columns were excluded from the microarray matrix to extract a bicluster using simulated annealing. It adds rows and columns until their residual score attains a given specific minimum MSR threshold value. This approach is promising in terms of controlling the required computational cost to define coherent bicluster. There is no significance in the resultant bicluster because a small size of the pattern is extracted for a high residual score.

Most of the heuristic-based biclustering procedures consider MSR value as one of the important parameters to tune the quality of the pattern. This constant value has an impact on the size as well as quality of a bicluster. It is complicated to decide the priority. Chakraborty[17] has introduced a biclustering technique which mimics the behaviour of GA, but differs with respect to population initialization. *K*-means clustering is used to define the initial chromosomes. A GA search most specifically finds a maximal set of biclusters. Since to only the best chromosome persists during genetic selection, it could be difficult to get a collection of different, non-redundant biclusters.

Divina and Aguilar-Ruiz[18] proposed a variant of GA-based biclustering, the sequential evolutionary BI clustering approach (SEBI). The term 'sequential of the evolutionary algorithm' refers to how only one bicluster is obtained per run. A sequential principle is carried out to get several biclusters from the evolutionary process. Additionally, to minimize overlapping among the different solutions, a weight matrix is used. Initially, the weight matrix is zero and it will be changed every time a bicluster is returned. The ultimate aim of this method is to obtain maximal biclusters with a residual value lower than the specifically mentioned constraint. Even so, SEBI works well for the distinctive place to detect a tiny pattern.

Mitra and Banka[19] proposed a method based on pareto dominance, which is called a multi-objective evolutionary algorithm (MOEA). This technique differs from the single-objective optimization problems in terms of considering more than conflicting objectives such as volume and coherence index (CI) of the bicluster in addition to the residual value. A local search strategy-based CC algorithm is used for the entire population at the commencement of every iteration. A measure named crowding distance is used to maintain diversity in the population. This approach has the advantage of being able to detect a bicluster with maximum size for a given constraint. Yet, this approach fails to converge and computational complexity is high in order to find the best solutions.

Liu *et al.*[20] introduced a biclustering algorithm based on the use of an estimation of distribution algorithms (EDAs) together with an evolutionary approach (GA) to avoid slow convergence rate and reduce the computation time. It is working like flow mechanisms which are the part of natural selection algorithms. The populations for the new generation are formed as a logical hierarchal structure. However, dependency on the solution is depicted clearly via the multimodel search space. Finally, this method converges into the local optimum solution.

Divina and Aguilar-Ruiz[21] presented sequential multi-objective biclustering (SMOB), which adopts a sequential strategy. The algorithm mimics the behaviour of MOEA. The weighted sum of the residue score, row variance and size are the parameters to decide the objective function. Moreover, the fitness solution is based on the Pareto front method[19]. So, it simultaneously reduces the number of parameters of the algorithm. Nevertheless, compared with MOEA, this method returns a limited size of the bicluster.

To find biclusters using spectral clustering principles and also find overlapping structures, Cano *et al.*[22] developed a procedure that follows the steps of one-dimensional clustering with singular value decomposition (SVD) named possibilistic spectral biclustering algorithm (PSB). Theoretically, this method obtains overlapping biclusters based on fuzzy technology and spectral clustering. In the following way, excessive overlapping among the biclusters is minimized: initially, check whether a set of existing efficient biclusters is more overlapped than a quantum of another set of a bicluster. The worst bicluster can be replaced with a previously generated bicluster. Choose the two overlapped biclusters which are available in the set, if any overlapping occurs. MSR value of the biclusters of PSB is better than those of the FLOC and CC algorithms. However, the overall quality of the bicluster is based on the number of eigenvectors.

A fuzzy set-based biclustering method named multi-objective fuzzy biclustering algorithm (MOFB) was proposed by Maulik *et al.*[23]. In two-dimensional microarray data, many of the biclusters may not disjoint; typically the boundaries of the biclusters overlap as rows and columns may belong to various co-clusters with unique membership degrees. Therefore, incorporating the fuzzy concepts is useful for detecting such overlapping biclusters. The objective of MOFB is that it concurrently minimizes the MSR value and maximizes the volume of bicluster and gene variance. To encode a group of biclusters in a string, the author a new variable string length encoding mechanism has been proposed. The fuzzy *K*-medoids algorithm is used to cluster the dataset into *K* partitions. The accuracy of MOFB for the extracted bicluster is better than the other algorithms in terms of MSR value and overall coverage. However, its computational cost is relatively high compared to the Cheng and Church algorithm.

A greedy technique based on a local search strategy biclustering method, i.e. random walk biclustering (RWB) was introduced by Angiulli *et al.*[24] to avoid premature convergence. This approach finds one bicluster at a time. Initially, the process begins with a random solution, then instantly it adopts successive transformations to obtain a locally optimal solution that improves the overall performance of the bicluster in terms of MSR, gene variance and volume. The ultimate aim of the transformation is to minimize MSR or maximize either the volume of the bicluster or the row variance. The algorithm walks randomly based on the probability value given by the user to get rid of local minima. Moreover, two distinct frequency thresholds were used to control the degree of overlapping rate of the extracted biclusters.

Gremalschi *et al.*[25] presented a different greedy approach to tackle control bicluster overlapping of the extracted bicluster. To handle this pitfall and accelerate the quality of bicluster with minimum MSR, the authors proposed a pair of novel MSR-based biclustering methods. Initially, this technique finds ($m \times n$)-bicluster with minimum MSR and is known as a dual biclustering algorithm. Next, the dual biclustering algorithm is combined with quadratic programming (QP) which will detect an optimal co-cluster sensibly because the size of the matrix is reduced by the dual biclustering method. Changing the threshold frequently can help reduce overlapping among the biclusters.

Based on the strategy of immune response and the local optimum strategy, a simple multi-objective immune biclustering (MOIB) method proposed by Liu *et al.*[26]. The objective of this algorithm was to find more than one meaningful bicluster with maximum size for low MSR value in the microarray matrix. The solution diversification is improved by combining the crowding distance strategy and ε-dominance. The crowding distance value of a particular solution can be computed by the average distance of its two neighbouring solutions. The mutation parameter is used in the solution to add either a row or a column and exclude one element, either column or row. The whole number of clones is referred as six multiply with the size of the antibody population. Moreover, there is no overlapping control mechanism adopted among the reported solutions.

The multi-objective ant colony optimization algorithm (MOACO) is used more often than other heuristic algorithms while solving the discrete path multi-objective optimization problem[27]. Liu *et al.*[28] proposed an application of ant colony optimization to the microarray data. The multi objective any colony optimization based biclustering incorporates local search strategies to find the average maximum biclusters with lower MSR and higher row variance. Here, a separate pheromone table is available for every ant because biclustering is not a unimodal problem. It requires that a few diverse solutions be given at the same time. The main drawback of this approach is poor convergence due to a decentralized processor to guide the ant system towards good solutions.

Liu *et al.*[29] presented a biclustering method on the basis of particle swarm optimization (PSO) as the neighbour search strategy along with the crowding distance. This kind of logic can enhance the convergence speed to the Pareto front and is also promising for a diversity of solutions. The authors focused on three objectives, namely homogeneity, row variance and size of the biclusters, which were satisfied concurrently by applying these fitness functions in the optimization framework. Many complicated optimization and search problems use PSO to reveal its speed in providing solutions. Even so, there is difficulty in selecting the probable value of inertia weight and constant acceleration coefficients.

A biclustering technique that is based on the multi-objective multi-population artificial immune network (MOM-aiNet) was proposed by Coelho *et al.*[30]. This method is inspired by the logic of clonal selection and the theory of immune network is incorporated into the original aiNet algorithm. After an initialization step, it consists of individuals randomly generated with just one column and row. All the solutions are grown by mutating and cloning the individuals. This approach includes three types of mutations namely insert one row and column, remove one element, from the row and remove one element from the column. After a certain number of iterations, within each population, this method gives all the non-dominated individuals. Nevertheless, there is no biological proof for the extracted biclusters and return trivial biclusters.

Gallo *et al.*[31] presented a novel memetic approach with a local search for microarray data biclustering that uses the multi-objective based evolutionary algorithm (MOEA), and it was employed in the PISA platform. In this approach, the optimization process has two modules[31]. The first phase has the apparatus required for the optimization problem. The next module holds the components of an optimization process that are independent of the optimization problem. The performance evaluation is done with MOEA on a couple of widely used benchmark datasets. The proposed method was competent to acquire maximum size biclusters with a high receptivity to the independent parameter.

In the population-based meta-heuristic algorithm, a fixed size of the population has the greatest impact on computational cost. Therefore, to mine coherent patterns from microarray data, Liu *et al.*[32] have proposed a novel dynamic multi-objective immune optimization biclustering (DMOIOB) algorithm, which adapts dynamically to adjusting the population size strategy. This method is inspired by the behaviour of the MOIB, and the sigma method is adopted to find the global best solutions. Moreover, the population declining strategy is used to restrict the population size so that it does not to grow excessively. However, it has higher computational cost compared to greedy approaches.

Joung *et al.*[33] presented a probabilistic coevolutionary biclustering algorithm (PCOBA) that generates clusters of rows and columns in a two-dimensional matrix simultaneously, based on coevolutionary searching and probabilistic learning. This strategy is most appropriate since it can perform clustering without specific constraints. Additionally, probabilistic learning is used to get statistical information on two populations. In this way, it improves the ability to search for the optimum value. The quality of a bicluster is examined through a suitable objective function. The low fitness value depicts the highly correlated bicluster and it should have a low residual score with the maximal size bicluster. Nevertheless, the performance of the proposed PCOBA is fully dependent on the control parameters. Similarly, a biclustering method that adopts the evolutionary strategies with tree-based search called condition-based evolutionary biclustering (CBEB) reported by Huang *et al.*[34]. The drawback of this method is that it fails to generate the multiplicative model biclusters.

Ayadi *et al.*[35] proposed an iterative local search approach for the biclustering problem is known as pattern driven neighbourhood search (PDNS). Initially, to transform the raw input data matrix into a behaviour matrix, normalization is used. Next, consecutive local search processes are considered to detect patterns of information. Since this approach utilizes a divide and conquer strategy to make exploited initial biclusters with high accuracy, it returns one bicluster for the entire run. So, to find more than one bicluster, the algorithm must be executed more times with different initial populations. Moreover, the authors considered initial biclusters from the outcome of two familiar methods. However, there are no mechanisms were adopted to control overlapping among the final deliverables.

To extract more than one large-sized correlated biclusters from the complex microarray dataset, Liu *et al.*[36] proposed a multi-objective dynamic population shuffled frog-leaping biclustering (MODPSFLB) approach. This algorithm incorporates a dynamic population and ε-dominance strategy. It uses crowding distance on the shuffled frog-leaping algorithm. Frogs are represented as a feasible solution in the search space. To preserve Pareto optimal solutions, it adopts computation of crowding distance and the ε-dominance relation. Hence, each new frog is generated as a population based on the strategy of the dynamic population at every next iteration. However, it has a higher computational cost compared to the algorithm of Cheng and Church[8].

Many researchers have implemented biclustering algorithms based on evolutionary techniques; the use of a general crossover concept does not extract highly correlated genes. Therefore, the biclustering algorithm is based on a new crossover method called EBACross, proposed by Maatouk *et al.*[37] for the specific biclustering of gene expression data. Standard deviation is applied to check whether the conditions belong to the same cluster or not. However, it possibly takes a long time to discover the coherent bicluster on large inputs.

Biclustering-based binary particle swarm optimization (BPSO) was proposed by Li *et al.*[38] and was found to be similar to the crowding distance-based multi-objective PSO biclustering[28]. However, they differ in the objective function, and BPSO focuses only on computing MSR. To improve the search efficiency of BPSO, it incorporates a pattern-driven local search operator. Initial bicluster is generated from the particle positions using a fixed-size binary string with a part for genes and the other for conditions. However, the MSR threshold plays an important role in deciding the size of the bicluster.

Inspired by the properties of the black hole, stellar-mass blackhole optimization (SBO) has emerged as a computational paradigm that applies the mass of the black hole principles to problem-solving in a wide range of areas. Balamurugan *et al.*[39] presented a nature-inspired algorithm for biclustering based on the concepts of absorption and emission. It is constituted by sequences of absorption, emission, coalescing and vanishing steps. The individuals who got success in new characteristics are getting the survival. It is based on the concept of 'strong survive and the weak perish'. The Jaccard coefficient distance measure is used to control the overlapping between the biclusters. The biological significance of the cluster genes can be verified using the gene ontology (GO) database.

Balamurugan *et al.*[40] derived their subspace clustering method based on the contribution of Nelder–Mead (NM) together with differential evolution (DE) as the neighborhood search strategy[41], which increases the convergence speed and also guarantees diversity of solutions. The NM procedure for high-dimensional data may reach premature convergence due to its poor ability to control coordinate moves in the solution space; also, it works well only for the unimodal problem. Hence, the modified Nelder–Mead (MNM) takes the median rather than the mean of the coordinates, and the evolutionary principle is adopted before performing the shrinking operation. Bicluster volume and MSR requirements are often conflicting here; for instance, the larger bicluster is more probable and has a higher MSR value. However, larger biclusters that have low MSR values are preferred.

In recent years many binary versions of the biclustering algorithm have failed to deal with the large size of data; during the clustering process, occurrence of more irrelevant rows or columns may lead to poor performance in clustering. Therefore, to improve the performance of the biclustering algorithm, Zhu *et al.*[42] have proposed an algorithm, which combines the features of fuzzy member matrix and comprehensive evaluation in fuzzy mathematics with a multi-objective optimization algorithm (MOFM). An important generalization principle applied in the fuzzification of algebraic operations is the closure property. To minimize the MSR value, a single point delete method is used, which deletes the rows or columns with maximum MSR.

Shuffled cuckoo search with the Nelder–Mead (SCS-NM) technique has been implemented[43] and it is similar to the Cuckoo Search with Mutation biclustering algorithm[44]. Both perform an exploration search based on the cuckoo search strategy and yield a set of eggs in the last population as output. However, they differ in generating a cuckoo egg. This clutch contains three eggs in each nest instead of a single egg and also to initiate diversification in the search space, it shuffles the eggs into a new search space after a certain number of epochs if the solution does not change.

Huang *et al.*[45] have recently introduced a technique using GA together with hierarchical clustering. To detect biclusters more proficiently in such a large search space, this bi-phase evolutionary architecture is used. It has two populations, i.e. a population of biclusters and a population of columns and rows grow in two phases which interacts with each other. On the other hand, traditional evolutionary biclustering algorithm uses single population structure. Cui *et al.*[46] presented a nature-inspired hybrid biclustering algorithm that is inclusive of a binary artificial fish swarm[47] with a binary simulated annealing algorithm (BAFS–BSA–BIC). In BAFS, every fish population is denoted as a boolean string instead of traditional values. But in simulated annealing the solution is represented in a binary form to process the large gene expression data matrix.

Discussion and conclusion

This article is the outcome of a comprehensive analysis of various available strategies for subspace clustering of microarray data. Table 1 summarizes the objective of the most widely used biclustering methods based on the evaluation measure MSR, together with the used datasets and the corresponding references. This present study considers a list of 41 subspace clustering methods. It paves a way for researchers to understand the evolution hierarchy and facilitates new investigators to start with the right initial point in the domain. From Table 1, it can be consequent that modern exploration on biclustering is being engrossed more based on the quality measure MSR. This tendency is created because of the managerial metaheuristics through a residual resource. However, the search policy does not create any impact on the method validation. Bio-inspired approaches for subspace clustering creates the utmost reconnoitered domain within stochastic schemes.

The dimension and complexity of raw clinical samples are the ultimate objectives for researchers to develop biclustering algorithms. Several use-cases of the biclustering algorithm are done on microarray expression data for bioinformatics research such as protein network analysis, accurate diagnosis, treatment planning, prognosis and drug design. Cheng and Church[8] implemented 2D clustering to a couple of microarray data matrices, namely yeast cell data and the lymphoma microarray data. Yeast data have 2884 rows and 17 samples, while 4026 genes and 96 samples are available with the human B-cells data. Later, most of the researchers have done biclustering on the yeast data. The dataset, namely *Arabidopsis thaliana* has 1000 selected genes under 153 samples.

Angiulli *et al.*[24] developed a technique for the most widely used couple of matrices covering cancer data: the dataset contains 181 tissue samples, defined by 12,533 genes which are known as lung cancer; the dataset leukemia collected 7129 genes from 72 acute leukemia patients. The yeast *Saccharomyces cerevisiae* (yeast stress) dataset consists of 2993 genes and includes 173 different samples such as amino acid starvation, shock and nitrogen source deletion used by several approaches such as PDNS and EBACross. The rat CNS dataset has nine tissue samples defined by 102 genes. Zhu *et al.*[42] applied MOFM to a mice protein expression dataset with 1080 measurements and 51 proteins. BAFS-BSA also considered the Mice Protein expression dataset used by MOFM. Recently, hybrid swarm intelligence BAFS-BSA method for co-clustering has been implemented on four datasets, namely cdc_15, complete_DTT, Mice Protein expression dataset and elutriation.

The biclustering of microarray data use-case is NP-hard[8]. So, the biclustering methods must balance the quality of the extracted bicluster and the cost of computation. Some researchers have mentioned the computational cost of their approaches. However, this information is not useful

Table 1. Mean square residue (MSR)-based biclustering algorithms

Method	Dataset	Pros	Cons
CC[8]	Yeast (2884 × 17), lymphoma (4026 × 96)	Automatic finding of similarities in a subset of attributes Grouping of genes and conditions. Overlapped grouping for representing the genes in a better manner.	High time complexity Biclustering does not need the computation of overall similarity between genes.
FLOC[13]	Yeast (2884 × 17)	Proposed a new algorithm named probabilistic algorithm which is used to incorporate null values in the bicluster model. It is able to identify a set of *k* possible overlapping biclusters simultaneously. Low cost	Temporary blocking of certain actions which are violating the biclustering process. Larger biclusters may not be considered.
DBF[14]	Yeast (2884 × 17)	Used to generate good quality biclusters which is based on frequent pattern mining. Refining the biclusters by adding more genes.	Quality of the created biclusters is not better than the method proposed here.
Bleuler-B[15]	Yeast (2884 × 17), *Arabidopsis thaliana* (1000 × 153)	Reduces the requirement of additional run-time resources. The quality of the biclusters is comparatively improved when compared to the methods which use the greedy strategy alone.	Quality of the created biclusters is not better than the method proposed here.
SA-B[16]	Yeast (2884 × 17), lymphoma (4026 × 96)	Presented the method to find out the high-quality bicluster seeds After finding the quality bicluster seeds, more genes are added to it.	Computational cost is bit high.
GA-B[17]	Yeast (2884 × 17), lymphoma (4026×96)	Used greedy algorithm which is embedded as a local search procedure to find the best biclusters. Yields good results when compared to the lymphoma and yeast datasets.	It is difficult to get a variety of non-redundant biclusters.
SEBI[18]	Yeast (2884 × 17), lymphoma (4026 × 96)	Uses multi-objective-based evolutionary algorithms for finding the best biclusters. Used to find quality biclusters with large variations.	The biclusters are found by a higher row variance. The size of the biclusters is limited.
MOEA[19]	Yeast (2884 × 17), lymphoma (4026 × 96)	It uses simple local search algorithms. Detects a bicluster with maximum size for a given constraint.	Computational complexity is high.
EDA-B[20]	Simulated matrix (200 × 60)	Introduced a biclustering algorithm based on the use of estimation of distribution algorithms together with an evolutionary approach genetic algorithm to escape from slow convergence rate and reduce the computation time.	Dependency on the solution is depicted clearly via the multimodel search space.
PSB[22]	Yeast (2884 × 17), lymphoma (4026 × 96)	Identified the potentially overlapping biclusters.	The overall quality of the bicluster is based on the number of eigenvectors.
MFOB[23]	Yeast (2884 × 17), lymphoma (4026 × 96)	The residual size is minimized. Cluster size and expression profile variance are maximized. Multiple biclusters are encoded into a single string. Generates a set of biclusters in a single run.	Time complexity is very high.
RWB[24]	Lung cancer (12533 × 181), colon cancer (2000 × 62)	Identified the overlapped biclusters. Poor local minima is disabled using a local search strategy.	The algorithm walks randomly based on the probability value given by the user to get rid of local minima.
DB-QP[25]	Yeast (2884 × 17)	The size of the matrix is reduced in order to find the optimal bicluster. Time complexity is low	
MOIB[26]	Yeast (2884 × 17), lymphoma (4026 × 96)	Dynamically adjust the population size strategy.	Computational cost is high.
MOACOB[28]	Yeast (2884 × 17), lymphoma (4026 × 96)	It is used to find more than one meaningful biclusters with maximum size for low MSR in the microarray matrix.	
CMOPSOB[29]	Yeast (2884 × 17), lymphoma (4026 × 96)	Proposed an application of ant colony optimization to the microarray data. The MOACOB incorporated local search strategies to find the average maximum biclusters with lower MSR and higher row variance.	Poor convergence due to a decentralized processor to guide the ant system towards good solutions.

(*Contd*)

Table 1. (*Contd*)

Method	Dataset	Pros	Cons
MODPSFLB[36]	Yeast (2884 × 17), lymphoma (4026 × 96)	Focuses on three objectives, namely homogeneity, row variance and the size of biclusters.	There is difficulty in selecting the probable value of inertia weight and constant acceleration coefficients.
CBEB34	Yeast (2884 × 17), lymphoma (4026 × 96)	Identifies all the non-dominated individuals.	There is no biological proof for extracted biclusters and return trivial biclusters.
CoBi48	Rat CNS (112 × 9), yeast (2884 × 17)	Requires a single pass over the database to generate all biclusters.	Extracts small biclusters for large MSR values.

Table 2. An empirical analysis of various methods for the yeast dataset

Method	Average MSR	Average volume	Average genes	Average samples
CC[8]	204.29	1557.0	167.0	12.0
FLOC[13]	187.84	1825.8	195.0	12.0
DBF[14]	114.70	1627.2	188	11
SA-B[16]	166.0	2605.5	268.6	10.8
GA-B[17]	161.87	3492.54	351.7	10.97
SEBI[18]	205.18	209.9	13.6	15.3
MOEA[19]	234.87	10301.68	1095.4	9.29
SMOB[21]	206.17	453.48	27.28	15.46
PSB[22]	169.03	1725.4	274.42	7.42
DB-QP[25]	171.19	–	–	–
MOIB[26]	202.32	2638.74	–	–
MOACOB[28]	203.12	2745.12	–	–
CMOPSOB[29]	218.54	1510.78	1102.8	9.31
MOM-aiNet[30]	178.28	1831.80	–	–
PISA-B[31]	261.61	13116.33	1047.63	12.52
DMOIOB[32]	201.86	2841.08	–	–
PCOBA[33]	219.15	1321.30	92.40	14.30
MODPSFLB[36]	215.98	11220.7	1154.21	9.81
CBEB[34]	233.59	–	–	–
EBACross[37]	167.62	495.3	38.08	3.78
BPSO-B[38]	301	1089	121	9
SBO-B[39]	160.75	3227.14	332.28	9.15
MNM-B[40]	180.56	2903.32	307.67	8.04
MOFM[42]	211.7	12082.85	1180.0	10.24
SCSNM-B[43]	167.43	3086.18	315.63	8.59
CSM-B[44]	176.12	2912.37	291.27	8.11
CoBi[53]	652.45	4992	347	15
MHS-B[54]	165.05	3049.54	334.21	8.73

because it is affected by various factors such as the size of the input, machine environmental settings, the programming platform used, number of iterations, etc. Tables 2 and 3 give a summary based on the qualitative performance measure for researchers who have revealed these statistics in their articles. As can be seen in Tables 2 and 3, with regard to MSR value, the results obtained by majority of methods are analysed on the yeast and lymphoma datasets. The second column reports the mean MSR value found by each method, the third column, the size of the cluster; while fourth and the fifth columns report the mean values of genes and conditions contained in the biclusters respectively. The symbol '-' indicates that the present authors do not consider the value for evaluation. Generally, the number of genes is multiplied by the number of samples and this refers to the 'volume' of the bicluster.

The process of extracting biclusters from a given dataset can be seen as a multi-objective optimization problem. For instance, in Table 2, biclusters found by SBO-B have a larger average bicluster size than those by MHS-B, though with the same MSR. However, when comparing MHS-B with EBACross and PSB, the biclusters found by the former are larger than those found by EBACross and PSB. Most of the algorithm is repeated until it reaches a pre-defined threshold. This threshold used in the fitness function is set to 300 for the yeast dataset and 1200 for the lymphoma dataset. It is a variational parameter with different datasets. Table 3 shows the performance of different state-of-the-art methods on the lymphoma data. Various approaches reveal that the multi-objective biclustering techniques can determine biclusters with maximum rows of samples, which indicates that the detected biclusters have many genes and

Table 3. An empirical analysis of various methods for the lymphoma dataset

Method	Average MSR	Average volume	Average genes	Average samples
CC[8]	850.04	4595.98	269.22	24.50
SA-B[16]	792.05	7711.98	730.6	16.8
GA-B[17]	592.28	3492.54	795.43	17.44
SEBI[18]	1028.24	615.84	14.07	43.57
SMOB[21]	1019.16	709.13	11.60	78.47
PSB[22]	361.4	4725.4	965.1	49.5
MOIB[26]	839.74	6918.29	–	–
MOACOB[28]	841.87	7274.19	–	–
CMOPSOB[29]	927.47	34012.24	902.41	40.12
MOM-aiNet[30]	759.37	2953.00	–	–
PISA-B[31]	1089.61	39821.51	655.93	60.71
DMOIOB[32]	832.79	7106.51	–	–
MODPSFLB[36]	913.53	35601.8	933.9	43.29
SBO-B[39]	780.45	9562.23	289.45	32.29
MNM-B[40]	832.09	8226.55	284.20	30.11
MOFM[42]	934.4	40604.32	976.3	41.59
SCSNM-B[43]	810.75	8876.46	292.61	31.51
CSM-B[44]	822.36	8387.42	281.52	30.93
MHS-B[54]	798.49	8882.51	295.92	31.94

samples with low MSR values. Here, a detailed review has been carried out for the most significant biclustering approaches, pointing out their merits and demerits, both denoting the implemented strategy and the quality of the obtained bicluster.

Wang *et al.*[48] developed a computer program which is used to combine both biclustering and divide and conquer approach[48]. This computer program is mainly applicable for local MSA and BlockMSA. The main objective of studying about single-cell RNA sequencing is to make new cell subtypes with the help of clustering. Ming Chu *et al.*[49] introduced a new bicluster method named JCB (joint CC and BIMAX). The proposed method is based on the algorithm of Cheng and church[8] and binary inclusion–maximal biclustering algorithm (Bimax). It merges the MSR introduced by Cheng and Church with the BIMAX algorithm. The merit of single-cell RNA sequencing is that is used to study about the cell based changes in the transcriptomic data. The main drawback of scRNA-seq data is that they contain noise and sometimes are sparse due to sampling deficiencies. Fang *et al.*[50] proposed a biclusering framework named DivBiclust, which is used to identify cell subpopulations[50]. It identifies subpopulations with good accuacy. Xie *et al.*[51] proposed a biclustering alogirthm which is named qualitatic biclustering algorithm[51,52]. The proposed model has a new mixture gaussian model[53] to test the importance of all the identified biclusters.

Comparative analysis with biological validation

Recently, scientists have understood the need for two-dimensional clustering in the bioinformatics domain and made significant efforts in this direction. Therefore, this study analytically reviews the fundamental need for biclustering techniques in biological data. The *p*-value is required to select meaningfully overrepresented functions. The *p*-value speaks indicates the proportion of genes added into the cluster randomly. If the *p*-value is small, then the cluster is framed without approximation. There is a major bioinformatics initiative to compute the probability of observing the number of genes from a particular GO category (function, process and component) within each bicluster. One of the most widely adopted gene-based benchmarks for biclustering methods is GO-based significance. It depicts how significantly a group of genes identified by a biclustering method is enriched with a similar GO category in terms of the statistically significant GO annotation database.

Recently, to identify the biological relevance of the biclusters from the Gasch yeast dataset, the nature-inspired SBO technique was used. The interpretations of genes for three ontologies, namely cellular component, molecular function, and biological process are acquired. To evaluate the biological significance, the results of the recent technique were compared with traditional approaches such as Bimax, BiMine, CC, ISA and OPSM for yeast expression data[35]. For this, we used the FuncAssociate 2.0 web tool[52]. The adjusted significance scores for each bicluster were computed using this web tool. Indeed, these scores were computed as adjusted *p*-values. These values specify how they match with the different GO categories. When the *p*-value is close to 0, it indicates a good match. Figure 4 shows a comparative analysis of different values of the significant score (*p*-value) of yeast cell-cycle expression data. For instance, 100% of the tested biclusters under all the mentioned methods have a *p*-value of 5% and 1%. At the *p*-value of 0.5% and 0.1%, only SBO shows a higher percentage of the tested bicluster which are 100% and 98% respectively. Lastly, 87% of the detected biclusters of SBO are statistically significant with *p*-value = 0.001%,

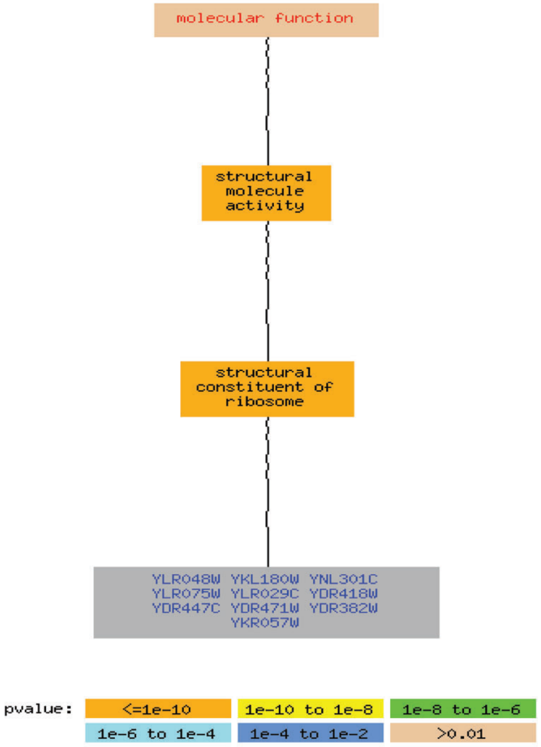


Figure 4. The GO functional activity of yeast expression data (ten genes).

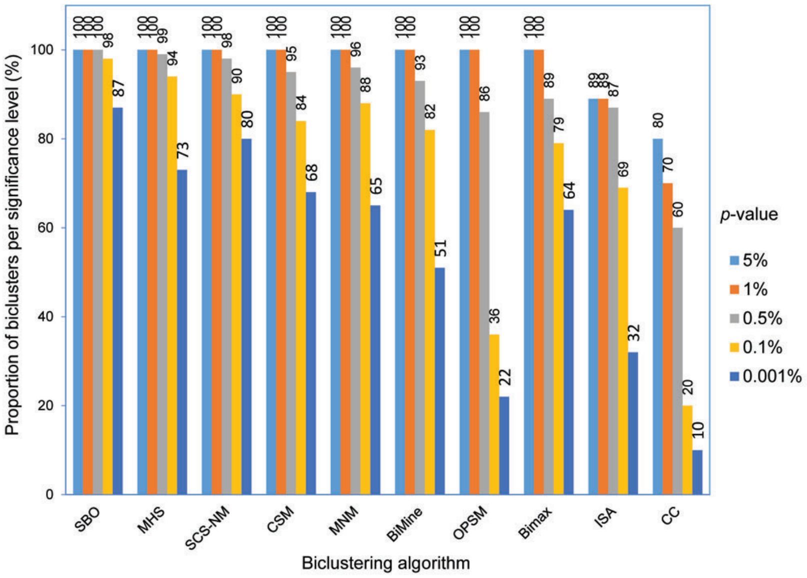


Figure 5. Proportion of biclusters significantly enriched by gene ontology (GO) annotations on yeast cell-cycle data.

while in the case of MHS, SCS-NM, CSM and MNM it is 73%, 80%, 68% and 65% respectively. Comparatively, we can conclude the SBO is better than the other proposed methods on this dataset for all *p*-values. We also note that SBO performs well for 0.001% *p*-values compared to CC, ISA, Bimax and OPSM. It performs well for all *p*-values (5%, 1%, 0.5%, 0.1% and 0.001%).

Functional activity analysis

The molecular function vocabulary is three-structured. It represents basic activities such as catalysis or binding. GOTermFinder is a functional analysis tool available in the *Saccharomyces* genome database. It supports much extracting the major shared GO terms of the cluster of genes

Table 4. Remarkable gene ontology terms for three biclusters on *Saccharomyces cerevisiae* data

Bicluster #	#Genes	Process	Function	Component
BC_4	1475	Catalytic process ($n = 712, p = 2.17 \times 10^{-29}$)	Structural molecule activity ($n = 594, p = 3.08 \times 10^{-8}$)	Extracellular ($n = 1287, p = 8.43 \times 10^{-21}$)
BC_5	1510	Hydrolase ($n = 658, p = 4.17 \times 10^{-16}$)	Orgonic cyclic activity ($n = 294, p = 7.21 \times 10^{-27}$)	Nuclear part ($n = 1347, p = 7.16 \times 10^{-19}$)
BC_8	1492	Transferase ($n = 881, p = 3.16 \times 10^{-28}$)	Hydrolase activity ($n = 299, p = 1.29 \times 10^{-27}$)	Intracellular part ($n = 1354, p = 2.76 \times 10^{-23}$)

and offers users to obtain the characteristics that the genes have. Figure 5 shows the proportions of biclusters significantly enriched by GO annotations on yeast cell-cycle data. Table 4 shows the major common GO terms available to define the group of genes in each bicluster for the ontologies of function, component and process. The supreme terms are depicted here. Most of the genes are predominantly involved only in structural molecule activity. For instance, in bicluster BC_1, the record (n =594, $p = 3.08 \times 10^{-8}$) depicts that 594 genes out of 1475 genes belong to structural molecule activity function with statistical significance (p-value = 3.08×10^{-8}). Moreover, for the clusters with ten genes, the biological network false discovery rate (FDR) is 0.00000. This is very low value which indicates that the subspace clustering methods can return biologically significant co-clusters and in most case, it is only zero. The analogous which has much less p-value ($p = 8.34 \times 10^{-15}$) is used to obtain the gene cluster that is random and more biased on it.

1. Achuthsankar, S. N., Computational biology and bioinformatics: a gentle overview. *Commun. Comput. Soc. India*, 2003, 1–12.
2. Liew, A. W. C., Yan, H. and Yang, M., Data mining for bioinformatics. In *Bioinformatics Technologies* (eds Chen, P. and Yi-Ping), Springer, Heidelberg, 2005, chapter 4, pp. 63–116.
3. Pérez-Suárez, A., Martínez-Trinidad, J. F. and Carrasco-Ochoa, J. A., A review of conceptual clustering algorithms. *Artif. Intell. Rev.*, 2019, **52**, 1267–1296.
4. Shannon, W., Culverhouse, R. and Duncan, J., Analyzing microarray data using cluster analyses. *Pharmacogenomics*, 2003, **4**(1), 41–52.
5. Domany, E., Cluster analysis of gene expression data. *J. Stat. Phys.*, 2003, **110**(3–6), 1–18.
6. Risch, N. and Merikangas, K., The future of genetic studies of complex human diseases. *Science*, 1996, **273**, 1516–1517.
7. Hartigan, J., Direct clustering of a data matrix. *J. Am. Stat. Assoc.*, 1972, **67**(337), 123–129.
8. Cheng, Y. and Church, G. M., Biclustering of expression data. In Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, Menlo Park, USA, 2000, pp. 93–103.
9. Yan, D. and Wang, J., Biclustering of gene expression data based on related genes and conditions extraction. *Pattern Recogn.*, 2013, **46**, 1170–1182.
10. Xie, J., Ma, A., Fennell, A., Ma, Q. and Zhao, J., It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data. *Brief. Bioinform.*, 2013, **20**(4), 1449–1464.
11. Madeira, S. C. and Oliveira, A. L., Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 2004, **1**(1), 24–45.
12. Pontes, B., Girldez, R. and Aguilar-Ruiz, J. S., Biclustering on expression data: a review. *J. Biomed. Inform.*, 2015, **57**, 163–180.
13. Yang, J., Wang, H., Wang, W. and Yu, P., Enhanced biclustering on expression data. In Proceedings of the Third IEEE Symposium on Bioinformatics and Bioengineering, Bethesda, USA, 2003, pp. 321–327.
14. Zhang, Z., Teo, A., Ooi, B. C. and Tan, L., Mining deterministic biclusters in gene expression data. In Proceedings of International Conference on Fourth IEEE Symposium on Bioinformatics and Bioengineering, Taichung, Taiwan, 2004, pp. 157–169.
15. Bleuler, S., Prelic, A. and Zitzler, E., An EA framework for biclustering of gene expression data. In Proceedings of Congress on Sixth Evolutionary Computation, Portland, OR, USA, 2004, pp. 166–173.
16. Chakraborty, A., Biclustering of gene expression data by simulated annealing. In Proceedings of Eighth International Conference on High-Performance Computing in Asia-Pacific Region, Beijing, 2005, pp. 78–90.
17. Chakraborty, H. M., Biclustering of gene expression data using genetic algorithm. In Proceedings of Computational Intelligence in Bioinformatics and Computational Biology, Toronto, Canada, 2006.
18. Divina, F. and Aguilar-Ruiz, J. S., Biclustering of expression data with evolutionary computation. *IEEE Trans. Knowl. Data Eng.*, 2006, **18**(5), 590–602.
19. Mitra, S. and Banka, H., Multi-objective evolutionary biclustering of gene expression data. *Pattern Recogn.*, 2006, **39**(12), 2464–2477.
20. Liu, F., Zhou, H. and Liu, J., Biclustering of gene expression data using EDA-GA hybrid. In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 2006, pp. 1598–1602.
21. Divina, F. and Aguilar-Ruiz, J. S., A multi-objective approach to discover biclusters in microarray data. In Proceedings of the Ninth Annual Conference on Genetic and Evolutionary Computation, London, UK. 2007, pp. 385–392.
22. Cano, L., Adarve, J., López, A. and Blanco, Possibilistic approach for biclustering microarray data. *Comput. Biol. Med.*, 2007, **37**(10), 1426–1436.
23. Maulik, U. A., Mukhopadhyay, S., Bandyopadhyay, M. Q. and Zhang, X. Z., Multi objective fuzzy biclustering in microarray data: method and a new performance measure. In Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, China, 2008, pp. 1536–1543.
24. Angiulli, F., Cesario, E. and Pizzuti, C., Random walk biclustering for microarray data. *J. Inform. Sci.*, 2008, **178**(6), 1479–1497.
25. Gremalschi, S. and Altun, G., Mean squared residue based biclustering algorithms. In *Bioinformatics Research and Applications* (eds Măndoiu, I., Sunderraman, R. and Zelikovsky, A.), ISBRA, Lecture Notes in Computer Science, Springer, Berlin, Germany, ISBRA, 2008, vol. 4983.
26. Liu, J., Li, Z. and Chen, Y., Microarray data biclustering with multi-objective immune optimization algorithm. In Proceedings of the Fifth International Conference on Natural Computation, Tianjin, China, 2009, pp. 564–580.
27. Dorigo, M. and Stützle T., *Ant Colony Optimization*, MIT Press, Cambridge, USA, 2004.

28. Liu, J., Li, Z. and Hu, X., Multi-objective ant colony optimization biclustering of microarray data. In Proceedings of the IEEE International Conference on Granular Computing, Nanchang, China, 2009, pp. 424–429.
29. Liu, J., Li, Z., Hu, X. and Chen, Y., Biclustering of microarray data with MOSPO based on crowding distance. *Bioinformatics*, 2009, **10**(4), 1–9.
30. Coelho, G. P., De Franca, F. O. and Zuben, F. J. V., Multi-objective biclustering: when non-dominated solutions are not enough. *J. Math. Modell. Algorithms*, 2009, **8**(2), 175–202.
31. Gallo, C. A., Carballido, J. A. and Ponzoni, I., Microarray biclustering: a novel memetic approach based on the PISA platform. In Proceedings of the Seventh European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Tübingen, Germany, 2009, pp. 44–55.
32. Liu, J., Li, Z. and Hu, X., Dynamic biclustering of microarray data by multi-objective immune optimization. *BMC Genomics*, 2011, **12**(1), S11.
33. Joung, J. G., Kim, S. J., Shin, S. Y. and Zhang, B. T., A probabilistic coevolutionary biclustering algorithm for discovering coherent patterns in gene expression dataset. *BMC Bioinformatics*, 2012, **13**(1), S12.
34. Huang, Q., Tao, D., Li, X. and Liew, A. W. C., Parallelized evolutionary learning for detection of biclusters in gene expression data. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 2012, **9**, 560–570.
35. Ayadi, W., Elloumi, M. and Hao, J. K., Pattern-driven neighborhood search for biclustering of microarray data. *BMC Bioinformatics*, 2012, **13**(7), 1–15.
36. Liu, J., Li, Z., Hu, X. and Chen, Y., Multi-objective dynamic population shuffled frog leaping biclustering of microarray data. *BMC Genomics*, 2012, **13**(3), 25–36.
37. Maatouk, O., Ayadi, W., Bouziri, H. and Duval, B., Evolutionary algorithm based on new crossover for the biclustering of gene expression data. In Proceedings of the Pattern Recognition in Bioinformatics, Stockholm, Sweden, 2014, pp. 48–59.
38. Li, Y., Tian, X., Jiao, L. and Zhang, X., Biclustering of gene expression data using particle swarm optimization integrated with pattern-driven local search. *IEEE Congress Evolut. Comput.*, 2014, **29**, 1367–1373.
39. Balamurugan, R., Natarajan, A. M. and Premalatha, K., Stellar-mass black hole optimization for biclustering microarray gene expression data. *App. Artif. Intell. Int. J.*, 2015, **29**(4), 353–381.
40. Balamurugan, R., Natarajan, A. M. and Premalatha, K., Biclustering microarray gene expression data using modified Nelder–Mead method. *Int. J. Inf. Commun. Technol.*, 2016, **9**(1), 43–63.
41. Lagarias, J. C., Reeds, J. A., Wright, M. H. and Wright, P., Convergence properties of the Nelder–Mead simplex algorithm in low dimensions. *SIAM J. Optimiz.*, 1998, **9**(1), 112–147.
42. Zhu, X., Qiub, J. and Jianxin, M., A multi-objective biclustering algorithm based on fuzzy mathematics. *Neurocomputing*, 2017, **253**, 177–182.
43. Balamurugan, R., Natarajan, A. M. and Premalatha, K., A new hybrid cuckoo search algorithm for biclustering of microarray gene-expression data. *Appl. Artif. Intell.*, 2018, **32**(7–8), 644–659.
44. Balamurugan, R., Natarajan, A. M. and Premalatha, K., Cuckoo search with mutation for biclustering of microarray gene expression data. *Int. Arab J. Infor. Technol.*, 2017, **14**(3), 300–306.
45. Huang, Q., Huang, X., Kong, Z., Li, X. and Tao, D., Bi-phase evolutionary searching for biclusters in gene expression data. *IEEE Trans. Evolut. Comput.*, 2019, **23**(5), 803–814.
46. Cui, Y., Zhang, R. and Gao, H., A novel biclustering of gene expression data based on hybrid BAFS–BSA algorithm. *Multimedia Tools Appl.*, 2019; doi:org/10.1007/s11042-019-7656-7.
47. Azad, M. A. K., Rocha, A. M. A. C. and Fernandes, E. M. G. P., A simplified binary artificial fish swarm algorithm for 0–1 quadratic knapsack problems. *J. Comput. Appl. Math.*, 2014, **259**, 897–904.
48. Wang, S., Gutell, R. R. and Miranker, D. P., Biclustering as a method for RNA local multiple sequence alignment. *Bioinformatics*, 2007, **15**(23), 3289–3296.
49. Chu, H.-M., Kong, X.-Z., Liu, J.-X., Wang, J., Yuan, S.-S. and Dai, L.-Y., Joint CC and Bimax: a biclustering method for single-cell RNA-seq data analysis. *Bioinfor. Res. Appl.*, 2021, **18**, 499–510.
50. Fang, Q., Su, D., Ng, W. and Feng, J., An effective biclustering-based framework for identifying cell subpopulations from scRNA-seq data. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 2021, **18**(6), 2249–2260.
51. Xie, J. *et al.*, QUBIC2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale RNA-Seq data. *Bioinformatics*, 2020, **36**(4), 1143–1149.
52. Berriz, G. F., Beaver, J. E., Cenik, C., Tasan, M. and Roth, F. P., Next generation software for functional trend analysis. *Bioinformatics*, 2009, **25**(22), 3043–3044.
53. Roy, S., Bhattacharyya, D. K. and Kalita, J. K., CoBi: pattern based co-regulated biclustering of gene expression data. *Pattern Recog. Lett.*, 2013, **34**(14), 1669–1678.
54. Balamurugan, R., Natarajan, A. M. and Premalatha, K., A modified harmony search method for biclustering microarray gene expression data. *Int. J. Data Min. Bioinform.*, 2016, **16**(4), 269–289.

Received 4 October 2021; revised accepted 10 June 2022

doi: 10.18520/cs/v123/i4/530-541