Biclustering of Gene Expression Data Based on Binary Artificial Fish Swarm Algorithm

Rui Zhang, Huacheng Gao, Yinqiu Liu, Yuanyuan Lu, Yan Cui

Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, PR China

cuiyan@njupt.edu.cn

Abstract: Many existing biclustering algorithms has been used to determine co-expressed genes in gene expression data under subsets of experimental conditions. The Mean Squared Residue (MSR) or the Average Correlation Value (ACV) often be employed as fitness functions. But this measure may not find some relevant genes with shifting and scaling patterns. Here we introduce a new approach - Binary Artificial Fish Swarm Algorithm (BAFSA), which possesses an improved Meta-heuristic search algorithm that combines traditional artificial fish swarm algorithm (AFSA) with binary forms. To find genes with shifting and scaling patterns, we used a fitness function based on the linear correlation. The biclustering algorithm based on BAFSA has been applied to Mice Protein Expression dataset and many biologically significant biclusters are found, which exhibited the superb performance. Then the performance of the proposed method is compared to CC, QUBIC and FLOC.

Keywords: Microarray, Gene Expression Data, Biclustering, Binary Artificial Fish Swarm Algorithm

1 Introduction

Gene expression is the process by which a gene's coded information is converted into the structures present and operating in a cell [1]. DNA microarray technology is one of the most popular ways to mark and monitor gene expression in a cell, it measures the expression level of a large number of genes within a great deal of different experimental conditions [2].

To deal with gene expression data, it is common arranged in a data matrix which rows represent genes, columns represent conditions and each element represents a gene expression level under a specific condition. Also, the implementation of clustering techniques is based on the matrix. Clustering has been used in such applications as the identification of co-expression genes, because it is often useful to gather genes based on conditions or merge conditions depend on genes [3]. However, it is not necessary to cover all the genes or columns in these clusters, and it may be useful to identify a subset of the conditions where a subset of genes act in a coherent manner – this is termed biclustering [4].

Cheng and Church [5] first achieved the biclustering algorithm in gene expression data analysis. So far, a large number of relevant algorithms have emerged and have been applied to address different biclustering problems [6].

In this paper, to further attempt at researching biclustering, we put forward a new biclustering algorithm based on Binary Artificial Fish Swarm Algorithm (BAFSA). Artificial fish swarm algorithm (AFSA) is a kind of swarm intelligence algorithm that imitates the behavior of individuals and information among them interactions among them during feeding process in real environment [7]. BAFSA is an improved AFSA, which each individual is represented in binary form. To find a better bicluster and improve convergence speed, Genetic algorithm (GA) was combined with BAFSA. The fitness function in the BAFSA contains the linear correlation among genes in a bicluster, which improves the localization of shifting and scaling patterns [8].

2 Description of BAFSA for biclustering

The implementation of BAFSA for biclustering is divided into two phases, first is initialization phase of biclustering, second is optimal search biclustering.

2.1 Initial co-clusters generation phase

In the initialization phase, the gene expression data is preprocessed to obtain the input of the BAFSA —— the artificial fishes (AFs). K-Means clustering is a common treatment [9]. First, we apply K-Means clustering to cluster n gene sets under all conditions and m condition sets under all genes respectively. Then the gene sets and condition sets are combined together to form $n \times m$ co-clusters.

2.2 AF — encoding of co-cluster

Each of the co-cluster is encoded in fixed length binary format containing $m + n$ bits, which m represents the

978-1-5386-6005-8/18/$31.00 ©2018 IEEE

number of genes and n stands for the quantity of conditions in the gene expression data. When the gene or condition is in a co-cluster, the corresponding bit is set 1 and otherwise 0. The input of the BAFSA are these encoded co-clusters, which are AFs. The format of AF is shown in Figure 1:

G1	G2	G3	···	Gm	C1	C2	C3	···	Cn

Figure 1 The format of AF

2.3 Bicluster and AF

In BAFSA, each of AFs (AF) represents a bicluster. For example, AF is 101001|101, M is a 6*3 gene expression data matrix, including 6 genes and 3 conditions. As shown in Figure 2, we extract a 3*2 data matrix, which is a corresponding bicluster B.

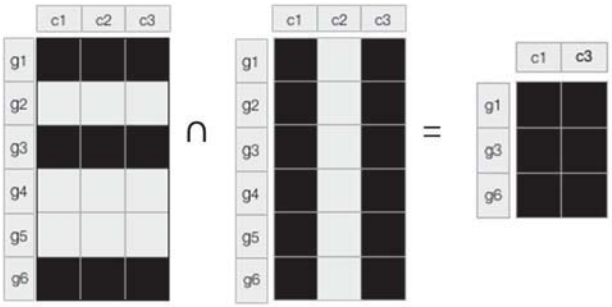


Figure 2 Conversion method between AF and bicluster

2.4 Fitness function in BAFSA

The purpose of the fitness function in BAFSA is to find biclusters presenting shifting and scaling patterns. Therefore, it is defined by:

$$f(B) = \big(1 - \rho(B)\big) + \frac{1}{N_G} + \frac{1}{N_C} \tag{1}$$

where B represents a bicluster, $\rho(B)$ is the average correlation of B, N_G and N_G are the number of genes and conditions of B. Best biclusters are those with the lowest value for the fitness function [8].

2.5 'Visual Range' and 'crowding factor' in BAFSA

In the BAFSA, the Hamming distance, Hd, is used to identify the AFs inside the 'visual range'. The Hamming distance between two bit sequences of equal length is the number of positions at the which the corresponding bits are different [10].

$$m = \max_{i,j\in(1,\dots,N),i\neq j} Hd\big(AF_i, AF_j\big), \tag{2}$$

where N represents the number of AFs.

$$v = \delta \times m,\ \delta \in (0,1) \tag{3}$$

When $Hd\big(AF_i, AF_j\big) < v$, the AF_j inside the 'visual range' of the AF_i.

$$n_i = \begin{cases} n_i + 1, if\ Hd\big(AF_i, AF_j\big) < v \\ n_i, \quad if\ Hd\big(AF_i, AF_j\big) \geq v \end{cases}, i \in (1, \dots, N) \tag{4}$$

where n_i $(n_i \geq 0)$ represents the number of the AFs inside the 'visual range'. The 'crowding factor' Cf_i of AF_i is calculates by the following formula.

$$Cf_i = \frac{n_i}{N} \tag{5}$$

2.6 Evaluation function in BAFSA

In the optimization phase, if the 'visual range' of AF_i is not crowded, we first need to calculate the fitness function value of each AF and $\overline{AF_l}$ inside the 'visual scope', where $\overline{AF_l}$ is the central point inside the 'visual scope' of the AF_i. Then compare their fitness function values. If the fitness function value of a AF is the minimum value, perform the following behavior, otherwise perform aggregating behavior. The pseudocode to compute $\overline{AF_l}$ is shown in Algorithm1 [10].

Algorithm 1 $\overline{AF_i}$

Input: **N: dimensions of each AF;**
n: the number of AF inside the 'visual scope' of the AF_i**;**
$AF_{ij}^l, l \in (1,2,..,N), j \in (1,2,..,n)$**: an AF inside the 'visual scope' of the** AF_i**;**
Output: $\overline{AF_i}$

1: **for** l=1 to N **do**
2: Calculate $\overline{AF_l^l} = \frac{\sum_{j\in(1,2,\dots,n)} AF_{ij}^l}{n}, l \in (1,2,..,N)$;
3: **if** $(1/(1 - e^{-(\overline{AF_l^l}-0.5)})) > rand()$ **then**
4: $\overline{AF_l^l} = 1$
5: **else**
6: $\overline{AF_l^l} = 0$
7: **end if**
8: **end for**
9: **return** Central point $\overline{AF_l} = (\overline{AF_l^1}, \overline{AF_l^2}, ..., \overline{AF_l^N})$

2.7 Following behavior in BAFSA

In the following behavior, each of AFs inside the 'visual range' of AF_i, denoted AF_j, moves to the AF with the lowest value for the fitness function, denoted by AF_{min}. The following method is combined with the uniform crossover present in GA. AF_j', an updated AF_j, each bit of it is created by copying the corresponding bit from AF_j or AF_{min} with equal probability [10].

2.8 Aggregating behavior in BAFSA

In the aggregating behavior, $\overline{AF_l}$ is the central point

inside the 'visual range' of AF_i and its fitness function value is the lowest. So, each of AF inside the 'visual range' of AF_i, which is AF_j, moves to $\overline{AF_i}$ and the way of moving is the same as that in the following behavior [10].

2.9 Preying behavior and Random behavior in BAFSA

In both cases, the following behavior and the aggregating behavior cannot be performed, so we introduce the preying behavior.

(1) the 'visual range' is crowded;

(2) the 'visual range' is not crowded, however, AF_{min} and $\overline{AF_i}$ have not improved in the process of calculating the fitness function value.

In the preying behavior, we first randomly choose a AF_{rand} inside the 'visual range' of AF_i. Then calculating the fitness function value of AF_{rand}. If this value is better, each of AF inside the 'visual range' of AF_i, which is AF_j, moves to AF_{rand} and the way of moving is the same as that in the following behavior. The number of try is $iter\_num$ times, so a random behavior is performed when the fitness value of AF_{rand} don't improve after $iter\_num$ times.

In the random behavior, we can generate AF_j', an updated AF_j, by setting a binary string of 0/1 bits of N, where N represents the length of AF_j'.

2.10 Leaping behavior in BAFSA

BAFSA, the optimization algorithm, is easy to fall into local optimum, which is the lowest value of fitness function remains the same during multiple successive iterations. To solve this problem, the leaping behavior is introduced.

After every L iterations where the fitness function value does not change, the algorithm first selects randomly select a AF in all AFs, denoted AF_{rand}. Then, each bit of AF_{rand} is replaced with opposite value 0/1 in probability $p$and the updated point replaces AF_{rand} to participate in algorithm operations, where $p = 0.01$ [10].

The purpose of the leaping behavior is to make the result closer to the global optimal solution.

2.11 The algorithm

The pseudocode of BAFSA for bicluster is shown in Algorithm2. The BAFSA for biclustering is repeated $iter\_num$ times to find the best bicluster B_{best}. The number of AFs is defined as $fish\_num$ where a AF is a encoded form of bicluster. In every iteration, the algorithm performs different behaviors according to different situations, which is described in detail in Algorithm2. In order to 60 optimum biclusters in the Mice Protein Expression dataset, the Algorithm2 is run 60 times. Therefore, the time complexity of this algorithm is relatively, and it does not have advantage when dealing with larger datasets and it is a common problem for all meta-heuristics. However, the accuracy of this algorithm is very high for certain datasets, indicating that it still has research significance.

Algorithm 2 BAFSA for bicluster

Input: $fish\_num$**: number of AFs;**
$iter\_num$**: number of iterations;**
try_num**: number of try**
Output:Bicluster B_{best}

Initialize co-clusters $C_i, i = 1, 2, ..., fish\_num$;
Encode co-clusters as $AF_i(i = 1, 2, ..., fish\_num)$ in BAFSA;
Calculate the fitness function value of biclusters represented by AFs, recorded $Y_n, n = 1, 2, ..., fish\_num$;
$num \leftarrow 0$
while $num < iter\_num$ **do**
  for i=1 to $fish\_num$ **do**
    Calculate 'visual range' and 'crowing factor' of AF_i;
    if 'visual range' is empty **then**
      Perform random behavior and go to (7);
    else if 'visual range' is not crowded **then**
      if $f(AF_{best}) < f(\overline{AF_i}) < Y_n$ **then**
        Perform the following behavior and update Y_n;
      else if $f(\overline{AF_i}) < f(AF_{best}) < Y_n$ **then**
        Perform the aggregating behavior and update Y_n;
      else
        Perform random behavior and go to (7);
      end if
    else if 'visual range' is crowded **then**
      if $f(AF_{rand}) < Y_n$ **then**
        Perform the preying behavior and update Y_n;
      else
        Perform the random behavior and go to (7);
      end if
    end if
  end for
  $L \leftarrow 0$
  if Y_n changes **then**
    $L \leftarrow 0$
  else if Y_n does not change and $L < m$ **then**
    $L \leftarrow L + 1$
  else if $L = m$ **then**
    Perform the leaping behavior and go to (7);
  end if
  $num \leftarrow num + 1$
end while
Obtain Y_{min} of Y_n and a corresponding B_{best}
return Y_{min} and B_{best}

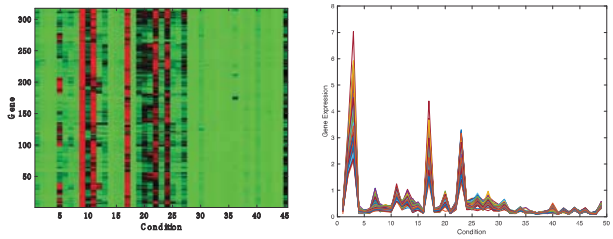
3 Experiments

3.1 Dataset Used

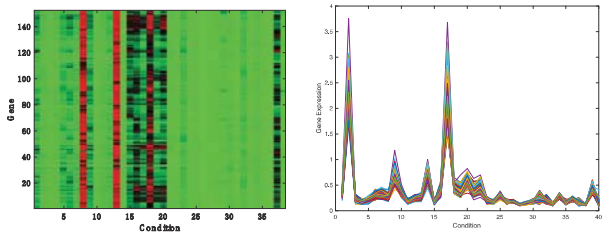
The BAFSA for biclustering has been tested to the Mice Protein Expression dataset, which is composed of 1080 genes and 51 experimental condition. The range of the expression values for this dataset is from -0.0620 to 8.4826. Total missing values are presents in the dataset which are represent by random values from -0.0620 to 8.4826 [11].

3.2 Bicluster results and analysis based on BAFSA

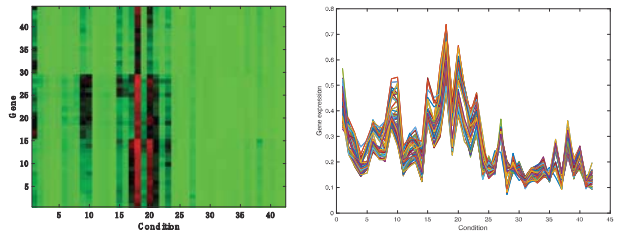
In this algorithm, total 60 optimum biclusters are produced and the most of them have good performance, such smaller MSR values, larger ACV values and size of bicluster. Several biclusters and their heat-mappings are shown in the Figure3 [12]. The specific results are shown in Table1, such as the number of genes and conditions, the size, the correlation coefficient value, MSR and ACV of a bicluster.



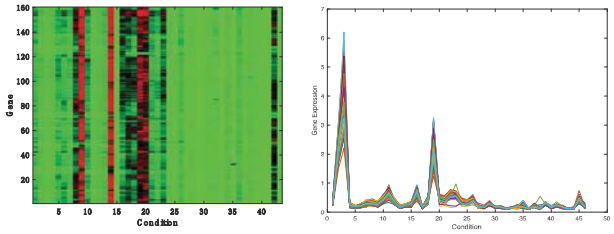
(a) Bicluster 1



(b) Bicluster 2



(c) Bicluster 3



(d) Bicluster 4

Figure 3 Four biclusters and heat-mappings

Table 1 The value of indicators of biclusters

	gene	sample	volume	$\rho(B)$	MSR	ACV
B1	125	49	6125	0.9867	0.0197	0.9867
B2	152	40	6080	0.9920	0.0082	0.9854
B3	44	44	1936	0.9943	0.0013	0.9711
B4	160	43	6923	0.9938	0.0173	0.9875

3.3 Performance evaluation between other algorithms

Performance of existing state-of-the art biclustering techniques, such as CC [5], QUBIC [13] and FLOC [14] are compared with BASFA for bicluster, based on average volume (Ave-volume), average correlation coefficient (Ave-Corr), average MSR value (Ave-MSR) and average ACV (Ave-ACV).

The good performance bicluster with the larger volume, correlation coefficient value, ACV value and the smaller MSR value [15]. The specific comparison is shown in Table 2 and Figure 3. And form the table and figure, BAFSA has better performance than FLOC, QUBIC and CC.

Table 2 Comparison of performance of proposed methods with other algorithms

	Ave-volume	Ave-Corr	Ave-MSR	Ave-ACV
BAFSA	5266	0.9917	0.0011	0.9827
CC	912	0.8369	0.1151	0.7429
QUBIC	27	0.8428	0.1862	0.8425
FLOC	12027	0.9073	0.0426	0.7259

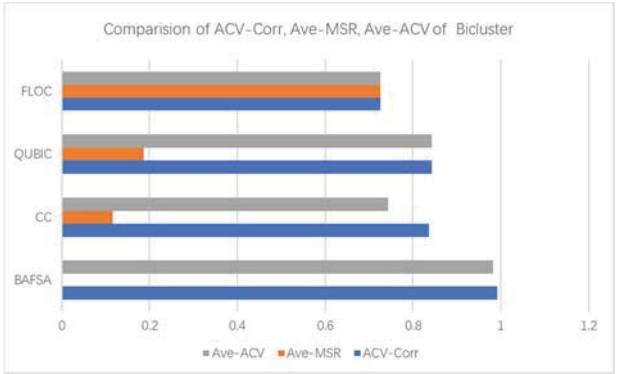


Figure 4 Comparison of ACV-Corr, Ave-MSR, Ave-ACV of bicluster

4 Conclusions

The meaning of biclustering is to reveal lots of new molecular functions and biological of the organism. In this paper, we introduce a new biclustering approach based on BASFA from gene expression data. Most of these similar algorithms are prone to fall into local optimal problem, we may avoid this risk by adopting the Leaping behavior algorithm. Other than that, in order to improve the speed and efficiency of artificial fished movement, GA is combined with BASFA in Following behavior, Aggregating behavior, Preying behavior and Random behavior.

The proposed algorithm has been tested with the real dataset - Mice Protein Expression, and obtained good performance. According to [16], a group biclusters composed by genes with shift and scaling patterns has been discovered some of which cannot be detected using MSR [8], so ACV is introduce to increase credibility.

Acknowledgments

We acknowledge the financial support from the National Natural Science Foundation of China (61402240).

References

[1] Ma, P.C. and K.C. Chan, A novel approach for discovering overlapping clusters in gene expression data. IEEE transactions on bio-medical engineering, 2009, **56**(7): 1803-9.

[2] Madeira, S.C. and A.L. Oliveira, Biclustering Algorithms for Biological Data Analysis:A Survey. IEEE/ACM Transactions on Computational Biology & Bioinformatics, 2004, **1**(1): 24-45.

[3] Berrar, D.P., W. Dubitzky, and M. Granzow, A Practical Approach to Microarray Data Analysis. Briefings in Functional Genomics & Proteomics, 2003, **2**(1): 82-84.

[4] Bryan, K., P. Cunningham, and N. Bolshakova. Biclustering of expression data using simulated annealing. in IEEE Symposium on Computer-Based Medical Systems. 2005.

[5] Cheng, Y. and G.M. Church. Biclustering of expression data. in Eighth International Conference on Intelligent Systems for Molecular Biology. 2000.

[6] Madeira, S.C. and A.L. Oliveira, A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series. Algorithms for Molecular Biology, 2009, **4**(1): 8.

[7] Jiang, M., D. Yuan, and Y. Cheng. Improved Artificial Fish Swarm Algorithm. in International Conference on Natural Computation. 2009.

[8] Nepomuceno, J.A., A. Troncoso, and J.S. Aguilar-Ruiz, Biclustering of gene expression data by correlation-based scatter search. Biodata Mining, 2011, **4**(1): 3.

[9] Thangavel, K., J. Bagyamani, and R. Rathipriya, Novel Hybrid PSO-SA Model for Biclustering of Expression Data. Procedia Engineering, 2012, **30**(4): 1048-1055.

[10] Azad, M.A.K., A.M.A.C. Rocha, and E.M.G.P. Fernandes, Improved binary artificial fish swarm algorithm for the 0–1 multidimensional knapsack problems. Swarm & Evolutionary Computation, 2014, **14**: 66-75.

[11] Saha, S. and P. Das. A Novel SFLA Based Method For Gene Expression Biclustering. in IEEE International Conference on Research in Computational Intelligence and Communication Networks. 2017.

[12] Vengatesan, K., et al. Performance Analysis of Gene Expression data using Biclustering Iterative Signature Algorithm. in IEEE Intl. Conf. on Intelligent Computing, Instrumentation and Control Technologies, Ieee-Icicict'17, Kannur. 2017.

[13] Li, G., et al., QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. Nucleic Acids Research, 2009, **37**(15): e101.

[14] YANG, J., et al., An improved biclustering method for analyzing gene expression profiles. International Journal on Artificial Intelligence Tools, 2005, **14**(05): 771-789.

[15] Tchagang, A.B. and A.H. Tewfik, DNA Microarray Data Analysis: A Novel Biclustering Algorithm Approach. Eurasip Journal on Advances in Signal Processing, 2006, **2006**(1): 059809.

[16] Aguilar-Ruiz, J.S., Shifting and scaling patterns from gene expression data. Bioinformatics, 2005, **21**(20): 3840-5.