A Novel Granular-Based Bi-Clustering Method of Deep Mining the Co-Expressed Genes

Kaijie Xu1,2, Witold Pedrycz2,5, Zhiwu Li1,3, Yinghui Quan1, & Weike Nie4

Traditional clustering methods are limited when dealing with huge and heterogeneous groups of gene expression data, which motivates the development of bi-clustering methods. Bi-clustering methods are used to mine bi-clusters whose subsets of samples (genes) are co-regulated under their test conditions. Studies show that mining bi-clusters of consistent trends and trends with similar degrees of fluctuations from the gene expression data is essential in bioinformatics research. Unfortunately, traditional bi-clustering methods are not fully effective in discovering such bi-clusters. Therefore, we propose a novel bi-clustering method by involving here the theory of Granular Computing. In the proposed scheme, the gene data matrix, considered as a group of time series, is transformed into a series of ordered information granules. With the information granules we build a characteristic matrix of the gene data to capture the fluctuation trend of the expression value between consecutive conditions to mine the ideal bi-clusters. The experimental results are in agreement with the theoretical analysis, and show the excellent performance of the proposed method.

Gene expression data analysis1 through DeoxyriboNucleic Acid (DNA) microarray technology helps us solve a variety of problems, such as those encountered in medical diagnosis, molecular biology, and gene expression profiling2. So far, numerous technologies have been proposed for discovering such gene regulations. Among them clustering is popular and useful. After the first gene expression datasets became available, clustering remains widely used nowadays3. Traditional (global) clustering methods only analyze genes under all experimental conditions or only analyze conditions of all the genes. In practice, in numerous cellular processes, many genes are regularly co-expressed (co-regulated) under some special conditions4 but behave differently under different conditions. Consequently, mining local co-expressed valuable patterns becomes a vital objective in discovering genetic pathways that are not very clear when clustered globally5. Especially with the increasing number of conditions most of the objects in a dataset tend to be nearly equidistant from each other, completely masking the clustering structure. Then traditional (distance-based measures) clustering methods are unable to deal with this aspect, which motivates the development of the bi-clustering methods. Bi-clustering is not only able to reveal the global structure in data but it can discover the local information (in other words, it can form clusters in the feature space and the data space simultaneously).

Bi-clustering, first introduced by Hartigan6, has been further developed since Cheng and Church proposed a bi-clustering method (CC method) based on variance and applied it to gene expression data7. Their work remains one of the most important contributions to the bi-clustering field. Bi-clustering methods can uncover co-expressed valuable patterns of gene from plenty of gene expression data, which are more helpful to define genes functioning together than traditional clustering approaches. Popular bi-clustering methods, such as CC7, Flexible Overlapped Clusters (FLOC)8, Plaid9, order-preserving sub-matrix (OPSM)10, Iterative Signature Algorithm (ISA)11, Spectral bi-clustering method12, conserved gene expression MOTIFs (xMOTIFs)13, and BiMax14 have drawn much attention in the literature. Several other methods of bi-clustering using various techniques have also been reported, such as those based on evolutionary methods15, hierarchical bi-clustering16, and Bayesian bi-clustering17. The CC method uses mean squared residue of a bi-cluster as a similarity measure to greedily extract bi-clusters that satisfy a homogeneity constraint. Based on the CC and with the aim of improving the generic CC method, Yang et al. proposed another algorithm called Flexible Overlapped Clusters method8, where an additional function is introduced to deal with the missing data and to discover the overlapping bi-clusters18. Both the CC and the Flexible Overlapped Clusters methods optimizing the mean squared residue have been considered to be the most effective tools for processing gene expression data.

Mean squared residue is the most commonly used index in bi-clustering, however, it cannot always capture the trend consistency of bi-clusters accurately. Patterns with larger mean squared residue scores may present more consistent trends than those with smaller ones19. In contrast, some patterns with smaller mean squared residue scores do not exhibit visible consistent trends. The order-preserving sub-matrix method10, focusing on the relative order of the expression levels of each gene rather than the exact values, is proposed to find the hidden patterns in gene expression data. However, the method only concerns the size of the data, but ignores the actual value. Meanwhile, the order-preserving sub-matrix method requires more computing resources. Several methods based on matrix transformation were proposed to discover the bi-clusters with consistent trends, such as Deterministic Bi-clustering with

1Xidian University, Xi’an, China. 2Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. 3Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, China. 4School of Information Science and Technology, Northwest University, Xi’an, China. 5Faculty of Engineering, King Abdulaziz University, Jeddah, Saudi Arabia.
Frequent pattern mining\textsuperscript{20}, Quick Hierarchical Bi-clustering method\textsuperscript{19}, and Skeleton Bi-clustering\textsuperscript{21}. These methods can produce some more meaningful mining results. However, they still rely only on numerical data, but ignore their semantic aspects (say, linkages among data) and some hidden information is not discovered.

Granular Computing\textsuperscript{22} is a computing paradigm and emerging conceptual framework of information processing and plays an important role in many areas. Especially, in the field of Computational Intelligence, Granular Computing can model human reasoning to help deal with complex problems.

This study intends to introduce the theory of Granular Computing to the field of the DNA microarray technology. In the paper, we focus on building information granules to interpret the gene expression data, which will make the data easier to understand. More specifically, in this study, the gene data matrix is considered as a time series group that is transformed into a series of ordered information granules. Information granularity inherently arises when it comes to the interpretation and perception of time series (gene expression data). Using a vocabulary of information granules to describe the time series makes the interpretation of data easier to comprehend.

To build information granules, we use the time series theory in conjunction with the Fuzzy C-Means (FCM) clustering\textsuperscript{23}, and the principle of justifiable information granularity\textsuperscript{24}. Then we label the conditions of each gene based on the information granules to construct a characteristic matrix of the gene data matrix, which can capture and supervise the changing trend of the gene expression value between consecutive conditions. A collection of initial bi-clusters is generated by traversing the trend matrix and utilizing the D-miner method\textsuperscript{25}. In the final phase, the initial bi-clusters are refined with the supervision of mean squared residue and mean fluctuation degree\textsuperscript{19}. All these features contribute to the originality of the proposed approach. To quantify the performance of the method, a detailed analysis and a comprehensive suite of experiments are provided. The experimental studies demonstrate that the proposed approach achieves better performance compared with that of the two well-known methods used for gene expression. To the best of our knowledge, the idea of the proposed approach has not been exposed in the previous studies.

**BUILDING THE ORDERED INFORMATION GRANULES**

In this section, we build a group of ordered information granules to explain the microarray expression data. In this process, each gene with all conditions is considered as a time series at different time points. Then the series are granulated into several ordered information granules. The conditions of each gene will be labeled according to the information granules.

Let $g_i = \{g_{i,0}, g_{i,1}, \ldots, g_{i,M}\}$ be a gene with $M$ conditions, which is considered to be a multivariable time series $g(t)$ ($t = 1, 2, \ldots, M$). Before proceeding with the granulation process, we sort the numerical data in an ascending order and treat the data with the same values as a single datum. This phase forms an important step in the proposed method; Fig. 1 shows the details of data processing. Assume that the sorted data are as follows:

\begin{equation}
\begin{aligned}
g_{i,0} &= g_{i,1} = \cdots = g_{i,k_0} = g_{i,k_1} = \cdots < g_{i,k_2} = \cdots < g_{i,0} = g_{i,1} = \cdots \end{aligned}
\tag{1}
\end{equation}

where $g_{i,0}, g_{i,1}, \ldots$, and $g_{i,m}$ denote the data with the same value.

Assume that the sorted data sequence $g'(t)$ contains $t(t \equiv M)$ different numerical data, and in the proposed method the building of information granules for $g(t)$ needs the assistance of the $g'(t)$.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{fig.png}
\caption{Example of data processing; see the description in the text.}
\end{figure}

Before building the information granules, some special cases should be considered. In the gene expression matrix such data occasionally occur:

\begin{equation}
\begin{aligned}
\text{case 1:} & \quad \Omega_i = \{g_{i,0}, g_{i,1}\} \\
\text{case 2:} & \quad \Omega_i = \{g_{i,0}, g_{i,1}\} \\
\text{case 3:} & \quad \Omega_i = \{g_{i,0}, g_{i,1}\}
\end{aligned}
\tag{2}
\end{equation}

When encountering such cases as shown above, instead of implementing the following granulation methods, the information granules are presented as interval information granules to cover the data in the following way:

\begin{equation}
\begin{aligned}
\text{case 1:} & \quad \Omega_i = \min[g'(t)]_i, \max[g'(t)]_i + \min[g'(t)]_i/2 \\
\text{case 2:} & \quad \Omega_i = \max[g'(t)]_i + \min[g'(t)]_i/2, \max[g'(t)]_i
\end{aligned}
\tag{3}
\end{equation}

Otherwise, the methods of building information granules will be implemented.

**Building Information Granules Based on Fuzzy C-Means clustering**

With the FCM clustering, the structure of available numerical data is represented by a partition (membership) matrix and prototypes (cluster centers). Considering the time series vector $g(t)$ to be granulated into $C$ information granules, a time series $g(t)$ is ultimately expressed as

\begin{equation}
\begin{aligned}
\left[ Q U^\ast \right] g = V
\end{aligned}
\tag{4}
\end{equation}

where $U = \{\mu_0, \mu_1, \ldots\} \in R^{C \times M}$ is the partition matrix, $\mu_\kappa$ is the degree of membership of sample point $g_i$ belonging to the $\kappa$th cluster, and $\varepsilon(\kappa > 1)$ is a fuzziness coefficient\textsuperscript{23}, $Q$ is such a diagonal matrix

\begin{equation}
\begin{aligned}
Q = \text{diag}\{\ldots, \sum_{\kappa=1}^{M} \mu_\kappa^2, \ldots\}
\end{aligned}
\tag{5}
\end{equation}

and $V \in \mathbb{R}^{k \times t}$ is a collection of the prototypes with $K$ being the number of clusters. Then, we construct an augmented matrix in
the form
\[ Z = [U \ V] \] (6)

To obtain the ordered information granules, we sort all other columns of the matrix according to the ascending order of the last column, and label them also in an ascending order. The sorted augmented matrix is
\[ Z' = [U' \ V'] \] (7)

Thus, the ordered information granules can be determined by \( U' \) and \( V' \). In addition, we set the number of the information granules as
\[ C = \text{ceil} \left[ 0.5 \text{card} \{g'(t)\} \right] \] (8)

where \( \text{ceil} [x] \) denotes the least integer value greater than or equal to \( x \).

Building Information Granules Based on The Principle of Justifiable Information Granularity

Building information granules based on the principle of justifiable information granularity is carried out on the time series \( g'(t) \). Here, the information granules describe the characteristics of the distributions of the gene expression values that are used to build a trend matrix of the gene expression data in the next phase of the proposed method to capture the fluctuation trend of the gene expression values. The specific process is as follows:

Partition the time series \( g'(t) \) into \( P \) sub-series temporal windows \( T_1, T_2, \ldots, T_p \), and denote the corresponding sub-time series as \( S_1, S_2, \ldots, S_p \). For each sub-time series by \( S_p \), we build an optimized information granule \( \Omega_p = [\alpha_p, \beta_p] \) under a given information granularity level \( \tau \), where \( \tau \) is a positive parameter delivering some flexibility\(^4\), which controls the justifiable granularity (coverage) when optimizing the interval information granule \( \Omega \). In the process, two intuitively compelling requirement (criteria) are considered: one is justifiable granularity (coverage) and the other is well-articulated semantics. Since the two criteria are in conflict (the increase of coverage reduces a level of flexibility and vice versa), an aggregate performance index in the form of the product of coverage and specificity is sought. By maximizing it, one produces the lower and upper bound of the interval information granule\(^6\), namely

\[
\max V(\beta) = f_1(\text{card}(g'(t) \in S_p \text{ med}(S_p < g'(t) \leq \beta))) \]
\[ \times f_2(\text{med}(S_p - \beta)) \] (9)

\[
\max V(\alpha) = f_1(\text{card}(g'(t) \in S_p \text{ med}(S_p < \alpha \leq g'(t) < \text{med}(S_p)))) \]
\[ \times f_2(\text{med}(S_p - \alpha)) \] (10)

where \( \text{med}(S_p) \) is the median of \( S_p \). \( f_1 \) and \( f_2 \) are increasing and decreasing functions, respectively. The most frequently used forms of \( f_1 \) and \( f_2 \) are
\[ f_1(u) = u, \quad f_2(u) = \exp(-tu) \] (11)

By maximizing the performance index, a group of optimal lower and upper bounds of an interval can be obtained to generate a number of information granules \( \Omega_p \) (\( p = 1, 2, \ldots, P \)).

Usually, we wish the information granules on the time series are the most “informative” (compact) so that they carry a clearly articulated semantics\(^6\). In other words, we need to find the optimal partition windows \( T_1, T_2, \ldots, T_p \) on the time series \( g'(t) \). An overall measure with this regard is to construct and optimize a multivariable objective function to find the optimal segmentation points. For each sub-time series \( S_p \), we compute the volume of the associated information granule \( \text{Vol}(\Omega_p) \) by

\[
\text{Vol}(\Omega_p) = T_p \sum_{t=1}^{T_p} g(t) \] (12)

Note that the bounds (\( g'_\min \) and \( g'_\max \)) of the integration are the minimal and maximal values of the temporary temporal window \( T_p \). By using the methods of global optimization such as Particle Swarm Optimization\(^2\) we minimize the following multivariable objective function to make the sum of the volume of the information granules reach the minimum:

\[
\min T_1, T_2, \ldots, T_p \sum_{p=1}^{P} \text{Vol}(\Omega_p) \] (13)

the optimized partition windows \( T_1, T_2, \ldots, T_p \) are obtained. The time series \( g'(t) \) is transformed into \( P \) interval information granules to present more efficient interpretation of the data. Assume that the \( P-1 \) segmentation points are \( g'_1, g'_2, \ldots, g'_{p-1} \) arranged in an ascending order. Then we build \( P \) intervals as

\[
\Phi_{p} = \left[ \min(g'(t)), g'_1 \right], \Phi_{p} = \left[ g'_1, g'_2 \right], \Phi_{p} = \left[ g'_2, g'_{p-1} \right], \ldots \]

\[
\Phi_{p} = \left[ g'_{p-1}, \max(g'(t)) \right] \] (14)

So far, the time series \( g'(t) \) has been transformed into interval information granules, labeled in an ascending order. Then we label the time series \( g(t) \) according to the labels of the information granules.

GRANULAR-BASED BI-CLUSTERING

Based on the ordered information granules, in this section, we present the granular-based bi-clustering model.

Construction of a Trend Matrix and a Slope Angle Matrix of the Gene Data Matrix

Consider a gene expression data matrix \( A \in \mathbb{R}^{N \times M} \) whose information granular labeling matrix is expressed as

\[
\Psi_A = [\cdots, \psi_{p,1}, \cdots] \in \mathbb{R}^{N \times M} \] (15)

Based on \( \Psi_A \), we form a trend matrix of the gene expression matrix as follows:

\[
G_A = \begin{bmatrix}
0_{[1 \times 1]} & \cdots & 0_{[1 \times (M-1)]} \\
\vdots & \ddots & \vdots \\
0_{[1 \times (n-1)]} & 0_{[1 \times (n-2)]} & \cdots & 0_{[1 \times (M-1)]} \\
0_{[2 \times 1]} & \cdots & 0_{[2 \times (M-1)]} \\
\vdots & \ddots & \vdots & \vdots \\
0_{[n \times 1]} & \cdots & 0_{[n \times (M-1)]}
\end{bmatrix}
\] (16)

where \( G_A \) contains \( N \times M \) sub-matrices, and the values of elements in each sub-matrix are 0, +1 and -1, which characterize all the rising and falling trends of different conditions of genes. It needs to be pointed out that +1 represents a rising trend, -1 represents a falling trend, and 0 represents both the rising and falling trends. The calculation of the sub-matrix with the \( n \)th row and the \( m \)th column in \( G_A \) is shown as

\[
\begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
\vdots \\
\vdots
\end{bmatrix}_{[n \times M]} = [\psi_{n1}, \cdots, \psi_{nk}, \cdots, \psi_{n(M-1)}], \quad \psi_{nk} = \text{sign}(\psi_{nm} - \psi_{nk})
\] (17)
With the characteristic matrix $\Phi$, the following task is to find the bi-clusters.

**Construction of Initial Bi-Clusters**

Suppose we are starting with the bi-clusters from the $c$th condition (column). The next $(m)$th condition is determined by the following form

$$
\begin{align*}
\sum_{m=1}^{n} \left[ y_{nm} = 1 \right] y_{nm} \in [1,1] + \\
\sum_{m=1}^{n} \left[ y_{nm} = 0 \right] y_{nm} \in [1,1] + \\
\sum_{m=1}^{n} \left[ y_{nm} = -1 \right] y_{nm} \in [1,1] + \\
\sum_{m=1}^{n} \left[ y_{nm} = 0 \right] y_{nm} \in [1,1]
\end{align*}
$$

which means that we are searching for the maximum same trends (MSTs) in the next $L$ conditions rather than all of them each time to make the bi-clusters contain as many co-expressed genes as possible. Here the parameter $L$ is used to control the number of genes and conditions in the initial bi-cluster. We set a unit pulse response function $e$ to adaptively control the value of $L$ during the search process in such a way:

$$
L = L_0(M - L_0 - c) + (M - c)(L_0 + c - M)
$$

where $L_0$ is the initial value of $L$, and the larger the $L_0$ is, the more rows the bi-cluster will contain; the smaller the $L_0$ is, the more columns the bi-cluster will have. Therefore, we obtain different (more) bi-clusters with different values of $L_0$, which is of great significance in gene expression.

When the next condition (the $k$th column) is obtained based on the initial (current, the $c$th) condition, the $k$th column will be set as the current condition to seek the next optimal condition (the $(k+1)$th column). The following search methods are basically the same as the initial one. The difference is that the genes (rows) participating in the following calculation will not be all genes (rows), but the genes (rows) that form the MSTs from the $c$th to the $(c+L)$th, that is

$$
\begin{align*}
\sum_{m=1}^{n} \left[ y_{nm} = 1 \right] y_{nm} \in [1,1] + \\
\sum_{m=1}^{n} \left[ y_{nm} = 0 \right] y_{nm} \in [1,1] + \\
\sum_{m=1}^{n} \left[ y_{nm} = -1 \right] y_{nm} \in [1,1] + \\
\sum_{m=1}^{n} \left[ y_{nm} = 0 \right] y_{nm} \in [1,1]
\end{align*}
$$

Obviously, as the search progresses, the columns (conditions) of the bi-cluster are constantly increasing and rows decreasing at the same time. To make the bi-cluster (sub-matrix) contain more genes (rows), we set a terminal parameter $\min\text{Gene}$, namely, $\min\text{Gene} \geq \min\text{Gene}$. Meanwhile, we also set a parameter $\min\text{Cond}$ to make the bi-cluster contain more conditions. Thus, a number of different initial bi-clusters can be obtained without violating the following requirements

$$
\text{card}[n] \geq \min\text{Gene}, \quad \text{card}[m] \geq \min\text{Cond}
$$

For the different values of the initial condition $c$, we can also obtain more different initial bi-clusters, and to construct much more initial bi-clusters we can set the MSTs as the second (or third etc.) MSTs or their combinations in the search process. In addition, by changing the value of $L_0$, the number of different initial bi-clusters can continue to increase dramatically. With the proposed approach each initial bi-cluster exhibits high consistency.

**Refine the Bi-Clusters**

Although a number of bi-clusters have been obtained, they are not always optimal to some extent. In this section, we refine them with the aid of some indexes.

We build another important characteristic matrix of the gene expression data, namely slope angle matrix, defined as

$$
\Theta = 180\arctan\left(\frac{\text{pivn}(\Phi)}{\pi}\right)
$$

where pivn is the pseudo-inverse of a diagonal matrix $\Phi$ expressed as

$$
\Phi = \text{diag}\left\{\text{max}(a_1), \ldots, \text{max}(a_M), \ldots, \text{max}(a_J)\right\}(M - 1)
$$

and $\Xi$ is an adjacent difference matrix that characterizes the differences between any two adjacent conditions in each gene, say,

$$
\Xi = [\ldots, a_0 - a_{i-1}, \ldots] \quad i = 1, 2, \ldots, N ; \quad j = 2, \ldots, M
$$

With the slope angle matrix, a mean fluctuation degree of a bi-cluster can be defined as

$$
\text{MFD}_{1}(I, J) = \frac{1}{\sqrt{||\Theta||_{\text{trace}}}} \sum_{[a]_{1,1}} \left(\Theta_{a} - \frac{1}{\sqrt{||\Theta||_{\text{trace}}}}\right)^{2}
$$

where $I \subset N$ and $J \subset M$ are subsets of genes and conditions, respectively. Obviously, for a bi-cluster if the genes (rows) have similar changing trends under each condition transition, its mean fluctuation degree score will be relatively smaller. Furthermore, if all genes (rows) in the bi-cluster have the completely similar (or same) changing trends under each condition transition, its mean fluctuation degree score will be zero. Therefore, we can refine the bi-clusters by combining the mean squared residue with the mean fluctuation degree. The refinement process consists of two steps.

**Step 1:** Delete rows and columns. Calculate the mean squared residue and the mean fluctuation degree scores of each initial bi-cluster and the mean squared residue of each row and column. Delete the rows and columns if they satisfy the following two conditions: i. the mean squared residue scores are larger than the given threshold; ii. after deleting them the mean fluctuation degree scores can be decreased; otherwise, this step will not be implemented.

**Step 2:** Add rows and columns. We add the rows and columns that are not in the bi-cluster and will not increase the mean fluctuation degree scores.

In the processes of deletion and addition of the rows and columns, the mean squared residue and mean fluctuation degree are constantly recalculated and improved (reduced) until a new optimal bi-cluster is obtained.

**EXPERIMENTAL RESULTS**

In the following experiments, we compare the performance of the proposed method with CC and the Flexible Overlapped Clusters methods, which are the two well-known bi-clustering methods commonly used for gene expression. The
implementation is completed in MATLAB. In the experiments, a commonly used gene expression dataset (Gordon-2002) is used, for a detailed description refer to\textsuperscript{28}. The methods are used to find 50 bi-clusters with the mean squared residue threshold of 1,200. In the use of the proposed method, the parameters $min\text{Gene}$, $min\text{Cond}$ and $L$, which control the size (number of the co-expressed genes and conditions) of the bi-clusters are set as 15, 10 and 30. For the Fuzzy C-Means clustering, the fuzziness coefficient $\kappa$ is set as 2, which is the most frequently used value in practice. The methods are repeated 10 times; the means and the standard deviations of the experimental results are presented.

The experimental results are plotted in Figs. 2 to 4, where the GBC-I represents the building information granules with Fuzzy C-Means clustering, GBC-II the principle of justifiable information granularity, CC the Cheng and Church’s method and the Flexible Overlapped Clusters method. Fig. 3 plots the heat map and 12 bi-clusters of the dataset obtained by all the methods for visualization comparison (Note that these are not average experiments, but the results of one of the experiments), and in the bi-clusters each color curve represents a single gene. We can see that the trend lines of the bi-clusters based on the proposed methods are very neat and orderly. On the contrary, those obtained by the CC and Flexible Overlapped Clusters methods are disorderly, though they contain more genes and conditions. Accordingly, the mean fluctuation degree scores plotted in Fig. 2 show that the proposed methods exhibit smaller mean values of these degrees. Furthermore, both the initial and the refined bi-clusters have good performance of mean fluctuation degree, in other words, the proposed methods are very effective in discovering the quality of initial bi-clusters by grouping together genes that have trends with more similar fluctuation degrees. Although the mean squared residue of the initial bi-clusters are larger than the given threshold, their mean fluctuation degree scores show that they also have the same changing trends. The CC and the Flexible Overlapped Clusters methods have obtained the bi-clusters without violating the requirements (a given mean squared residue threshold), however, they are not well suited for gene expression (not fully effective in discovering co-expressed genes under some special conditions) to some extent.

In addition, the impact of the random numbers in CC method and the random selection of initial seeds in Flexible Overlapped Clusters method make them very unstable, which are reflected in the large standard deviations as shown in Fig. 2. In Fig. 4, we also present a group of co-expressed genes of the dataset obtained by all the methods.

In other words, compared with the CC and Flexible Overlapped Clusters methods, the proposed methods exhibit visible advantages. The improvement is 30\% (on average) and varies in-between a minimal improvement of 25\% and 35\% in the case of the most visible improvement.

For the computing overhead, the proposed method searches the bi-clusters purposefully and obtains the results as fast as possible, which is outperforms the CC and Flexible Overlapped Clusters methods. Building the information granules and calculating the trend matrix of the gene expression data are also time consuming. Fortunately, both procedures are done off-line, and only needs to be implemented once for a given dataset and then stored.
CONCLUSIONS

In this research, we designed a bi-clustering method to discover the co-expressed genes based on the theory of granular computing. During the design process, information granules are introduced to transform the gene numerical data to capture the changing trend of the gene expression value between consecutive conditions. Bi-clusters are obtained through extracting the features of the information granules. We completed theoretical analysis and offered a comprehensive suite of experiments. Both the theoretical and experimental results are presented to verify the validity of the proposed method. Experimental results show that the proposed method outperforms the existing methods in finding the bi-clusters. To the best of our knowledge, this research approach offers an innovative direction to bi-clustering and comes with tangible improvements (say 30% improvement on average has been achieved). While at this phase sound background promising study has been completed, more experimental work could be pursued in the future. A viable and promising alternative would be to engage other formalisms of information granules, say fuzzy sets or rough sets.
REFERENCES

[1] Butler, A. Hoffman, P. & Snijder, P. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology* **36**, 411–420 (2018).

[2] Sudipta, S. Saha, S. & Sahoo, P. Bi-clustering of microarray data using a symmetry-based multi-objective optimization framework. *Soft Computing* **22**, 1–22 (2018).

[3] Wouter, S. Cannoodt, R. & Saeyts, Y. A comprehensive evaluation of module detection methods for gene expression data. *Nature Commun.* **9**, 5589 (2018), http://hdl.handle.net/1854/LU-8558915.

[4] Mulqueen, R. M. Pokholok, D. & Norberg, S. J. Highly scalable generation of DNA methylation profiles in single cells. *Nature Biotechnology* **36**, 428–431 (2018).

[5] Mishra, D. Shaw, K. & Mishra, S. Gene expression network discovery: a pattern based biclustering approach. In *Proc. 2011 International Conference on Communication, Computing & Security*. ACM, Rourkela, Odisha, India, 307–312 (2011).

[6] Hartigan, J. A. Direct clustering of a data matrix. *Journal of the American Statistical Association* **67**, 123–129 (1972).

[7] Cheng, Y. & Church, G. M. Biclustering of expression data. In *Proc. Conf. Intelligent Systems for Molecular Biology (ISMB)*, 93–103 (2000).

[8] Yang, J. Wang, H. X. & Wang, W. An improved biclustering method for analyzing gene expression profiles. *International Journal on Artificial Intelligence Tools* **14**, 771–789 (2005).

[9] Lazzeroni, L. & Owen, A. Plaid models for gene expression data. *Statistica Sinica* **12**, 61–86 (2000).

[10] Ben-Dor, A. Chor, B. and Karp, R. Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of Computational Biology* **10**, 373–384 (2003).

[11] Bergmann, S. Ihmels, J. & Barkai, N. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics* **67**, 1–18 (2003).

[12] Kluger, Y. Basri, R. & Chang, T. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Research* **13**, 703–716 (2003).

[13] Murali, T. M. & Kasif, S. Extracting conserved gene expression motifs from gene expression data. In *Proc. Pacific Symp. Biocomputing, 77–88 (2003).*

[14] Prelic, A. Bleuler, S. & Zimmermann, P. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* **22**, 1122–1129 (2006).

[15] Mitra, S. & Banka, H. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition* **39**, 2464–2477 (2006).

[16] Eisen, M. Spellman, P. Brown, P. & Botstein, D. “Cluster analysis and display of genome-wide expression patterns.” *National Academy of Sciences* **95**, 14863–14868 (1998).

[17] Gao, C. McDowell, I. C. & Zhao, S. Context specific and differential gene co-expression networks via Bayesian biclustering. *Computational Biology* **12**, 1–39 (2016).

[18] Khalid, B. & Allab, K. Bi-clustering continuous data with self-organizing map. *Neural Computing and Applications* **22**, 1551–1562 (2013).

[19] Ji, L. P. Mock, K. W. L. & Tan, K. L. Quick hierarchical biclustering on microarray gene expression data. In *Proc. Sixth IEEE Symposium on BioInformatics and BioEngineering (BIBE’06)*, Arlington, VA, USA, 110–120 (2006).

[20] Zhang, Z. H. Teo, A. & Tan, K. L. Mining deterministic biclusters in gene expression data. In *Proc. Proceedings. Fourth IEEE Symposium on Bioinformatics and Bioengineering*, Taichung, Taiwan, China, 283–290 (2004).

[21] Chen, J. Ji, L. P. Exploiting domain knowledge to improve biological significance of biclusters with key missing genes. In *Proc. 2009 IEEE 25th International Conference on Data Engineering*, Shanghai, China, 1219–1222 (2009).

[22] Pedrycz, W. Granular computing: analysis and design of intelligent systems. (Boca Raton, Press, CRC, USA, 2016).

[23] Orion, F. & Pedrycz, W. Enhancement of the classification and reconstruction performance of fuzzy c-means with refinements of prototypes. *Fuzzy Sets and Systems* **318**, 80–99 (2017).

[24] Pedrycz, W. & Homenda, W. Building the fundamentals of granular computing: a principle of justifiable granularity. *Applied Soft Computing* **13**, 4209–4218 (2013).

[25] Lawrence, H. Hughes, A. & Tonic, R. D-miner: A framework for mining, searching, visualizing, and alerting on darknet events. In *Proc. 2017 IEEE Conference on Communications and Network Security (CNS)* IEEE, Las Vegas, NV, USA, (2017).

[26] Lu, W. Chen, X. & Pedrycz, W. Using interval information granules to improve forecasting in fuzzy time series. *International Journal of Approximate Reasoning* **57**, 1–18 (2015).

[27] Lee, J. H. Song, J. Y. Kim, D. W. & Kim, J. W. Particle swarm optimization algorithm with intelligent particle number control for optimal design of electric machines. *IEEE Transactions on Industrial Electronics* **65**, 1791–1798 (2018).

[28] Li, X. & Wong, K. C. Evolutionary multiobjective clustering and its applications to patient stratification. *IEEE Transactions on Cybernetics* **99**, 1–14 (2018).