RN-Cluster: Discovering coherent biclusters which is Robust to Noise

Jaegyoon Ahn1, Youngmi Yoon1,2, and Sanghyun Park1
1. Computer Science Department, Yonsei University, South Korea
2. Information Technology Department, Gachon University of Medicine and Science, South Korea
{ajk, amyyoon, sanghyun} @cs.yonsei.ac.kr

Abstract

A bicluster is a subset of genes that show similar behavior within a subset of conditions. Biclustering algorithm is a useful tool to uncover groups of genes involved in the same cellular process and groups of conditions which take place in this process. We are proposing a polynomial time algorithm to identify functionally highly correlated biclusters. Our algorithm identifies 1) the gene set that follows additive, multiplicative, and combined patterns simultaneously that allow high level of noise, 2) the multiple, possibly overlapped, and diverse gene sets, 3) biclusters with negatively correlated as well as positively correlated gene set simultaneously, and 4) gene sets whose functional association is strongly high. We validated the level of functional association of our method, and compared with current methods using GO.

1. Introduction

Not all the genes in microarray dataset participate in a particular cellular process, and not all samples can be observed in a particular cellular process. We can expect subsets of genes to be co-regulated under certain experimental conditions, but to behave almost independently under other conditions [1]. Finding the set of co-regulated genes can lead to identify the functionality of the group of genes and eventually find the genetic pathways. Cheng [2] named the data mining technique that finds a submatrix of coherent gene set and sample set in a microarray as biclustering.

Many biclustering algorithms have been introduced, and biclustering is proven to be a NP-hard problem [2], so all these algorithms used heuristic methods or probabilistic approximation. Accordingly strengths and weaknesses of each algorithm are various and the patterns that each biclustering algorithm identified are also various. Biclustering algorithms could be divided into two groups largely by the patterns they find.

- Algorithms that find additive or multiplicative patterns:
  - δ-biclustering [2] uses mean squared residue of a submatrix to find biclusters. As a result, it finds additive or multiplicative co-regulation patterns. One weakness of δ-biclustering is that it allows only a small degree of noise. Thus it can identify strict patterns only. Also it can easily miss overlapping clusters due to the random value substitutions once a bicluster is identified.
  - p-Cluster [3] first scans the dataset to find all column-pair and row-pair maximal clusters called MDS. Then it does the pruning in turn using the row-pair MDS and the column-pair MDS. It then mines the final clusters based on a prefix tree. However, p-Cluster is not robust to noise, either.
  - Tri-Cluster [4] is the first algorithm that mines 3 dimensional microarray dataset. It makes a DFS(Depth First Search) tree whose node is the genes which show same range of fluctuation within user specified threshold ε. If ε is too big, DFS tree could grow too deep to complete the mining. However, Tri-Cluster with small ε does not allow high degree of noise. Moreover, its time complexity is exponential to the number of samples.
  - reg-Cluster [5] mines additive and multiplicative co-regulation patterns. It defines dij as a difference of the gene expression value between conditions ci and cj. Then it finds the gene set whose ratio of d01 and dij is within ε. Although the idea of mining additive and multiplicative patterns together is novel, it has a few problems. First, finding a proper ε is very difficult job. If ε is too big, the gene set in a bicluster would have many false positive genes. And if ε is too small, the gene set in a bicluster would have many false negative genes. Second, it constructs a DFS tree like Tri-Cluster. Thus it has same problems with Tri-Cluster.
The number of nodes of DFS tree becomes exponential as the level grows. Therefore, the algorithms that construct DFS tree have common problem that the number of resulting biclusters are extremely many. That is because all the biclusters which are slightly different are accounted into result. It is a difficult problem to examine all these biclusters and to choose the one among the similar biclusters. The noise level that the previously mentioned algorithms allow is not enough to find all the meaningful patterns. Thus we could say that they commonly find strictly additive or multiplicative patterns.

- **Algorithms that find pattern by keeping ordered sequence:**

  OPSM [1] defines a cluster as a submatrix of the original microarray matrix after performing column permutation separately for each row, whose gene expression values are in a non-decreasing pattern. Although OPSM shows good GO validation result [6], it is problematic that OPSM can find only one bicluster at a time. It could miss many meaningful biclusters hidden in the microarray data. OP-Cluster [7] and KiWi [8] use different algorithm with OPSM, but basically use same definition of OPSM. Commonly, OPSM based algorithms have possibility that the biologically significant patterns which do not preserve the order could be missed [9].

  In this paper, we propose a new model for biclusters with functionally highly correlated gene expression data called RN-cluster. 1) RN-cluster identifies the gene set that follows additive, multiplicative, and combined patterns within user specified threshold simultaneously. Combined pattern of additive and multiplicative shape allows highly flexible patterns with specified level of noise tolerance and allowing high level of noise does not require exponential time or space complexity in RN-cluster, which means that RN-Cluster is robust to experimental noise. 2) RN-cluster identifies the multiple, possibly overlapped gene sets while guaranteeing gene-diversity in a bicluster by complying with user specified similarity threshold. 3) RN-cluster identifies biclusters with negatively correlated as well as positively correlated gene set simultaneously. 4) RN-cluster identifies biclusters whose functional association is strongly high. This functional association is validated using GO database.

2. PRELIMINARIES

This section describes the notations and preliminary concepts before we detail the algorithm.

### 2.1 Notations

- **$G$** A set of genes
- **$S$** A set of samples
- **$(O, T)$** A submatrix of the data set, where $O \subseteq S$, $T \subseteq G$
- **$g_0, g_1, ...$** Genes in $O$
- **$s_0, s_1, ...$** Samples in $T$
- **$c_{ij}$** Expression value of gene $g_i$ on sample $s_j$
- **$d_{ab}^c$** $c_{ab}^c$ - difference of expression values of sample $s_b$ and sample $s_a$ on gene $g_c$
- **$f_{cd}$** $d_{cd}/d_{ab}$ ratio of gene $g_c$, where $s_b$ and $s_a$ are the first and second samples in RN cluster, respectively
- **$\delta$** User-specified maximum ratio threshold (> 1)
- **$mg$** User-specified minimum # of genes of a RN cluster
- **$ms$** User-specified minimum # of samples of a RN cluster

### 2.2 Preliminary concepts

Let $O = \{g_0, g_1, ..., g_m\}$ and $T = \{s_0, s_1, ..., s_n\}$, then $c_{ij}$ would be the expression level of gene $g_i$ in sample $s_j$. Let $C$ be a $m \times n$ submatrix of $(O, T)$ of the dataset $(G, S)$, then we can write $C = (O, T) = \{c_{ij}\}, i [0, m-1] \text{ and } j [0, n-1]$.

**Definition 1 (RN-cluster).** Let $C = (O, T)$ be a bicluster, where $g_0, g_1, ... \in O$ and $T = \{s_0, s_1, ..., s_n\}$. $C$ is a RN -cluster if $C$ satisfies the following properties:

1. $d_{ab}^c \neq 0, d_{ab}^c \neq 0, d_{cd}^a \neq 0$ and $d_{cd}^a \neq 0$
2. $\text{sign}(f_{cd}) = \text{sign}(f_{cd})$, where sign(x) returns -1 if x is negative and +1 if x is positive for all a > 1
3. $|f_{cd}| / \delta \leq |f_{cd}| \leq |f_{cd}| x \delta$, where $|f_{cd}|$ and $|f_{cd}|$ are maximum and minimum $|f_{cd}|$ values for $g_b$
4. $|O| \geq mg \geq 2$ and $|T| \geq ms \geq 3$

**Definition 2 (p-RNC).** When RN-Cluster $C = (O, T)$ and $|T| = p$, then we refer to $C$ as p-RNC. p refers to the number of samples involved in the bicluster.

For example, suppose there are 10 x 6 microarray dataset $(G, S)$ as shown in Table 1. Let 2-RNC be $G \times \{s_0, s_2\}$ submatrix, $ms = 3$, $mg = 3$ and $\delta = 2$. If we examine sample $s_3$, then $T = \{s_0, s_2, s_3\}$. The $d_{02}$ and $d_{23}$ for gene $g_k$ are shown in Table 2.

<table>
<thead>
<tr>
<th>Table 1. 10 x 6 microarray dataset</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>gene/sample</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_0$</td>
<td>0.15</td>
<td>-0.07</td>
<td>-0.25</td>
<td>-0.3</td>
<td>-1.12</td>
<td>-0.67</td>
</tr>
<tr>
<td>$g_1$</td>
<td>0.21</td>
<td>0.03</td>
<td>0.18</td>
<td>-0.27</td>
<td>-0.32</td>
<td>0.62</td>
</tr>
<tr>
<td>$g_2$</td>
<td>-0.03</td>
<td>-0.07</td>
<td>0.28</td>
<td>0.32</td>
<td>-0.27</td>
<td>-0.36</td>
</tr>
<tr>
<td>$g_3$</td>
<td>-0.25</td>
<td>0.58</td>
<td>0.77</td>
<td>0.28</td>
<td>0.32</td>
<td>0.65</td>
</tr>
<tr>
<td>$g_4$</td>
<td>0.11</td>
<td>0.04</td>
<td>0.75</td>
<td>0.82</td>
<td>0.21</td>
<td>-0.2</td>
</tr>
<tr>
<td>$g_5$</td>
<td>0.24</td>
<td>0.31</td>
<td>0.95</td>
<td>0.12</td>
<td>0.18</td>
<td>0.69</td>
</tr>
<tr>
<td>$g_6$</td>
<td>-0.3</td>
<td>0.32</td>
<td>0.02</td>
<td>-0.64</td>
<td>0.06</td>
<td>-0.04</td>
</tr>
<tr>
<td>$g_7$</td>
<td>-0.15</td>
<td>-0.25</td>
<td>0.18</td>
<td>0.06</td>
<td>-0.15</td>
<td>-0.17</td>
</tr>
<tr>
<td>$g_8$</td>
<td>0</td>
<td>-0.74</td>
<td>-0.38</td>
<td>0.87</td>
<td>-0.34</td>
<td>0.12</td>
</tr>
<tr>
<td>$g_9$</td>
<td>-0.15</td>
<td>0.2</td>
<td>0.31</td>
<td>0.15</td>
<td>0.04</td>
<td>-0.22</td>
</tr>
</tbody>
</table>
Table 2. Difference values and ratio of difference values for gene set O

<table>
<thead>
<tr>
<th>d_{21}</th>
<th>( \delta_g )</th>
<th>( \delta_t )</th>
<th>( \delta_{g_0} )</th>
<th>( \delta_{g_1} )</th>
<th>( \delta_{g_2} )</th>
<th>( \delta_{g_3} )</th>
<th>( \delta_{g_4} )</th>
<th>( \delta_{g_5} )</th>
<th>( \delta_{g_6} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.067</td>
<td>7.75</td>
<td>2.08</td>
<td>9.14</td>
<td>0.86</td>
<td>0.48</td>
<td>2.75</td>
<td>0.30</td>
<td>2.88</td>
</tr>
</tbody>
</table>

We can see that bicluster \( B_3 \) with gene set \( O = \{g_0, g_2, g_3\} \) and sample set \( T = \{s_0, s_2, s_1\} \) satisfies all the properties: 1) \( d_{21} \) and \( d_{32} \) for \( k = 0, 2, 4 \) are not zero, 2) \( d_{21} \) and \( d_{32} \) for \( k = 0, 2, 4 \) have same sign, 3) the values \( \max(||d_{21}||) = 9.14 \) when \( k = 4 \), \( \min(||d_{32}||) = 7.75 \) when \( k = 2 \) and \( ||d_{21}|| = 8 \) when \( k = 0 \). Thus, \( 9.14 / 2 = 4.58 \) < \( 8 < 7.75 \times 2 = 15.5 \), 4) \( |\delta_{g_2}| = 3 \geq 3 \) and \( |\delta_{g_1}| = 3 \geq 3 \). Thus \( B_3 \) can be said to be a 3-RNC. Similarly, we can see that bicluster \( B_3 \) with gene set \( O = \{g_0, g_1, g_0\} \) and \( B_1 \) with gene set \( O = \{g_0, g_4, g_4\} \) satisfies all the properties, thus \( B_3 \) and \( B_1 \) can be said to be a 3-RNC.

3. ALGORITHM

RN-Cluster mines a set of genes that behave similarly through a set of samples. RN-Cluster has two main steps: 1) get the initial 2-RNC set whose samples are all possible sample pairs, 2) for each p-RNC, get the (p+1)-RNC. We describe the details of each step.

3.1 Get the initial 2-RNC set

The set of samples \( \{s_i, s_j\} \) where \( i < j \) and the set of genes \( \{g_{i1}, g_{i2}, \ldots, g_{ni}\} \) form a 2-RNC. The number of all the 2-RNCS is \( n(n-1) / 2 \), which are all possible sample pairs. For example, in Table 1, possible sample sets of 2-RNC are \( \{s_0, s_1\}, \{s_0, s_2\}, \ldots, \{s_0, s_(n-1)\} \). Note that if \( ms = 3 \), then \( \{s_0, s_3\}, \{s_0, s_4\}, \{s_2, s_3\}, \{s_2, s_4\} \) and \( \{s_3, s_4\} \) cannot form a 2-RNC, because it cannot grow up to 3 or more-RNC (e.g., \( G \times \{s_0, s_3, s_4\}\). Similarly, \( \{s_0, s_1\}, \{s_1, s_2\}, \{s_2, s_4\} \) and \( \{s_3, s_4\} \) cannot form a 2-RNC if \( ms = 4 \).

3.2 Get the (p+1)-RNCs from p-RNCs

For all 2-RNCS \( C = (O, T) \), we make 3-RNC by examining the sample \( s_i \) such that \( l < i \), where \( s_i \) is the last sample in sample set \( T \). We can get 4-RNCS from 3-RNCs, 5-RNCs from 4-RNCs, and so on. In other words, we take the iterative method [10] by doing breadth first search to get (p+1)-RNCs from p-RNCs.

The examining process is composed of running and queuing process. Details of these parts are described in the following sub sections.

3.3 Ranging

The examining process first calculates \( \delta_{g_i} \) between \( s_i \) and \( s_j \) for all the genes \( g_i \). Then genes in the gene set \( O \) are clustered into two sets: set of positive values, \( OP \) and set of negative values, \( ON \) according to the sign of \( \delta_{g_i} \). If the sign of \( \delta_{g_i} \) is positive, \( g_i \) is included into \( OP \), else \( g_i \) is included into \( ON \). Keeping same signed sets in separate sets makes negatively correlated genes to be included in a bicluster.

First table of Figure 1 shows genes in \( OP \) and their \( \delta_{g_i} \) values. \( OP \) is sorted according to \( \delta_{g_i} \) in ascending order. Let \( q = (|OP| / 2) – 1 \). Then \( g_q \) is gene that is positioned in the middle of ordered \( OP \). Second table of Figure 1 shows ordered \( OP \) of first table, and \( g_q \) is positioned middle of ordered \( OP \).

Then we heuristically get 5 sub-gene sets \( OP_0, OP_1, OP_2, OP_3 \) and \( OP_4 \) from \( OP \) with Definition 1 being satisfied. Each of sub-gene set \( OP_1 \) contains genes whose \( t \) values are within range, where \( i = 0, 1, 2, 3, 4 \). Each range, are shown in Figure 1.

Once the ranges are derived, the sorted sequence of gene \( OP \) is linearly examined once. During examining, for each \( g_i \) in \( OP \), if \( |\delta_{g_i}| \subseteq \text{range}_{g_i} \), then \( g_i \) is clustered into \( OP_i \). All those \( OP_i \) except the ones whose size is less than \( mg \) become the gene set for new (p+1)-RNC whose new sample set is \( T = T \cup \{s_i\} \).
From the second table of Figure 1, we can see that if \( \delta = 2 \), \( range_0 = [1 \times 2^3, 1 \times 2^1] = [0.125, 0.5] \). Likewise, \( range_1 \), \( range_2 \), and \( range_3 \) can be calculated as in Figure 1. Because \( OP_0 \) is composed of \( g_i \)'s whose \( f'_i \) is in \( range_0 \), \( OP_0 = \{g_6, g_2, g_17\} \). \( OP_1 \), \( OP_2 \), \( OP_3 \), and \( OP_4 \) can be derived in a similar way. These processes are illustrated in Figure 1. We can apply the same process on \( ON \). While generating \((p+1)\)-RNCs, if we cannot get any \( OP, s \) or \( ON, s \) whose size is greater than or equal to minimum number of genes \( mg \), for all the \( p\)-RNCs and all the samples \( s_i \) where \( s_i \in S - T \), then there are no valid \((p+1)\)-RNC, and the entire process ends.

### 3.4 Queuing

Let the number of the \((p-1)\)-RNC be \( r \). For each \((p-1)\)-RNC, there are at most \( n \) samples to examine, and for each examination, at most \( p\)-RNCs are generated (5 ranges from \( OP \) and \( ON \)). Thus there are \( O(nr) \) \( p\)-RNCs. We cannot keep all these \( p\)-RNCs due to limitation of memory, and examining all these \( p\)-RNCs is exponentially time consuming process. However we observe that 1) we are only interested in distinguished \( p\)-RNCs whose gene set is bigger than others, and 2) we only need to keep \( p\)-RNCs which have higher possibility to grow up to \( p'\)-RNC where \( p' > p \).

To satisfy condition 1), we keep set of priority queues whose priority measure is the size of the gene set, \( |O| \), to keep the \( p\)-RNCs. The \( p\)-RNCs in these queues form the output biclusters. Figure 2 compares two queuing strategies: single priority queue, and multiple priority queues. Ours is implemented using multiple priority queues. Each node in BFS tree of Figure 2 denotes the name of RNC and the size of its gene set, \( |O| \).

Former strategy prunes the \((p+1)\)-RNCs from \( p\)-RNCs B and C, that means it does not guarantee variety of genes. Thus we keep multiple priority queues to guarantee the variety of the \( p\)-RNCs. Every \((p+1)\)-RNCs in each priority queues can be a result.

To satisfy condition 2), we need to keep another set of priority queues to keep the \( p\)-RNCs for next \((p+1)\)-RNCs. The queuing strategy is same as the case above, but we heuristically set the priority measure as \( |O| \times (n - last) \), where \( last \) is the index of last sample of \( T \) (for example, when \( T = \{s_0, s_2, s_3\} \), \( last = 3 \)). The strongholds for the priority measure \(|O| \times (n - last) \) are followings: 1) it is generally true that \( p\)-RNC with bigger gene set grows up to a \((p+1)\)-RNC with bigger gene set, and 2) as \( last \) gets bigger, the possibility that the \( p\)-RNC grows up to larger-RNC gets smaller. For example, suppose that \( S = \{s_0, s_1, s_2, s_3, s_4, s_5\} \) and there are two 3-RNCs \( SB_1 \) and \( SB_2 \) whose \( T \) is \( \{s_0, s_1, s_2\} \) and \( \{s_0, s_1, s_3\} \), respectively. \( SB_1 \) has more samples \( \{s_3, s_4, s_5\} \) to examine than \( SB_2 \) \( \{s_4, s_5\} \), which means \( SB_1 \) can grow up to three 4-RNCs with \( T = \{s_0, s_1, s_2, s_3\} \), \( T = \{s_0, s_1, s_3, s_4\} \) and \( T = \{s_0, s_1, s_2, s_3\} \), while \( SB_2 \) can grow up to only two 4-RNCs with \( T = \{s_0, s_1, s_3, s_4\} \) and \( T = \{s_0, s_1, s_2, s_3\} \). Furthermore, \( SB_1 \) can grow up to 6-RNC with \( T = \{s_0, s_1, s_2, s_3, s_4, s_5\} \), but \( SB_2 \) cannot grow up to 6-RNC. So we can say that, as \( last \) gets bigger, the possibility of the \( p\)-RNC growing gets smaller. Accordingly, priority queues have \( p\)-RNCs which have bigger possibility to grow up to larger-RNC. After we examine all the \( p\)-RNCs, we can get two sets of \((p+1)\)-RNCs, one is for output and another is the candidates for next \((p+1)\)-RNCs.

Let the size of each priority queue and the total number of priority queues be \( qsize \) and \( qnum \), respectively. Both \( qsize \) and \( qnum \) can be input as a user-specified parameter. \( qsize \) affects the diversity of results. We internally set the \( qsize \) as 100, which is big enough, since \( qsize \) is not directly proportional to the degree of diversity. Let \( k = qnum \times qsize \), then \( k \) is the...
total number of p-RNCs before eliminating duplicated RNCs. Accordingly, bigger \( qnum \) leads to bigger \( k \). Generally, the bigger \( k \) leads to less pruning, thus prevents local optima. However, experimental results (Table 5) revealed that \( qnum \) does not affect the quality of biclusters once the value is bigger than 100.

4. EXPERIMENTAL RESULTS

The experimental environment is Windows XP operating system on AMD Athlon 64 X2 Dual, 2.81GHz, 1.93GB RAM machine. We used only real microarray datasets to evaluate RN-Cluster which is the Gasch [11] yeast dataset. The yeast dataset consists of 2944 genes over 173 samples. In all experiments, we used \( ms = 5 \), \( mg = 10 \), \( \delta = 1.7 \), \( rt = 0.4 \) and \( qnum = 100 \) unless otherwise specified.

To show the quality of the RN-Cluster, we validated the results through GO, using FuncAssociate [12]. We compared GO validation result on OPSM, Bimax [6], ISA [13] and \( \delta \)-biclustering [2]. According to Prelic [6], the proportion of biclusters enriched with significance level \( \alpha = 0.001\% \) is about 88% in case of OPSM, when the same real data was used. OPSM shows best GO validation result among Bimax, ISA and \( \delta \)-biclustering. Meanwhile, the proportion of RN-clusters enriched with significant level \( \alpha = 0.001\% \) is 100%. That means every RN-Cluster is biologically meaningful.

To show more detailed result, we compared GO validation results of our bicluster with those of OPSM bicluster, which has approximately same number of genes with ours. We used BicAT v2.22 [14] to execute OPSM with iteration parameter = 10, on the same environment. Totally 14 biclusters were generated by OPSM, and we chose top 2 biclusters which show good GO validation result among them. Then we also chose biclusters with similar size of gene set from resulting RN-Clusters. Table 3-a), b), c) and d) show the top 5 p-values of biclusters from RN Cluster and OPSM and Table 3-e) is for the legends for other tables. We can see that p-values with same GO Attribute is generally much lower in RN Cluster than OPSM.

That means the significance of association is much bigger in the case of RN Cluster than OPSM. The graphs of Figure 3 show the RN-Clusters used in Table 3-a). We can easily find the negative correlation and the additive and multiplicative patterns in Figure 3.

Next we made experiments which guide the selection of default values for \( \delta \), and \( qnum \). Firstly we made measurements of run-time while varying \( \delta \) in seconds. Bigger \( \delta \) means accommodation of higher level of noise. Figure 4 shows RN-Clustering’s time complexity on varying \( \delta \). We can see that time complexity does not increase exponentially as \( \delta \) increases and lowest p-

![Figure 3. Gene curve graphs of 12-RNC](attachment:12-RNC.png)

<table>
<thead>
<tr>
<th>Rank</th>
<th>N</th>
<th>X</th>
<th>F</th>
<th>GO Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>114</td>
<td>186</td>
<td>4.9e-90</td>
<td>GO:0005820: cytosolic ribosome</td>
</tr>
<tr>
<td>2</td>
<td>94</td>
<td>186</td>
<td>2.4e-84</td>
<td>GO:0044445: cytosolic part</td>
</tr>
<tr>
<td>3</td>
<td>106</td>
<td>276</td>
<td>6.8e-80</td>
<td>GO:0005840: ribosome</td>
</tr>
<tr>
<td>4</td>
<td>108</td>
<td>948</td>
<td>7.1e-76</td>
<td>GO:0003228: non-membrane-bound organelle</td>
</tr>
<tr>
<td>5</td>
<td>108</td>
<td>948</td>
<td>7.1e-76</td>
<td>GO:0043232: intracellular non-membrane-bound organelle</td>
</tr>
</tbody>
</table>

(a) 271 x 12 Bicluster from RN-Cluster (12-RNC)

<table>
<thead>
<tr>
<th>Rank</th>
<th>N</th>
<th>X</th>
<th>F</th>
<th>GO Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>78</td>
<td>168</td>
<td>2.66E-120</td>
<td>GO:0005830: cytosolic ribosome</td>
</tr>
<tr>
<td>2</td>
<td>76</td>
<td>168</td>
<td>6.1e-116</td>
<td>GO:0044442: cytosolic part</td>
</tr>
<tr>
<td>3</td>
<td>81</td>
<td>276</td>
<td>1.1e-113</td>
<td>GO:0005840: ribosome</td>
</tr>
<tr>
<td>4</td>
<td>78</td>
<td>210</td>
<td>3e-107</td>
<td>GO:0003273: constituent of ribosome</td>
</tr>
<tr>
<td>5</td>
<td>76</td>
<td>210</td>
<td>3.2e-106</td>
<td>GO:00032279: ribosomal subunit</td>
</tr>
</tbody>
</table>

(c) 83 x 25 Bicluster from RN-Cluster (25-RNC)

<table>
<thead>
<tr>
<th>Rank</th>
<th>N</th>
<th>X</th>
<th>F</th>
<th>GO Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62</td>
<td>168</td>
<td>1.40E-82</td>
<td>GO:0005830: cytosolic ribosome</td>
</tr>
<tr>
<td>2</td>
<td>62</td>
<td>168</td>
<td>3.00E-79</td>
<td>GO:0044442: cytosolic part</td>
</tr>
<tr>
<td>3</td>
<td>66</td>
<td>276</td>
<td>1.30E-74</td>
<td>GO:0005840: ribosome</td>
</tr>
<tr>
<td>4</td>
<td>62</td>
<td>256</td>
<td>9.50E-72</td>
<td>GO:0003279: ribosomal subunit</td>
</tr>
<tr>
<td>5</td>
<td>61</td>
<td>250</td>
<td>1.30E-70</td>
<td>GO:0003273: constituent of ribosome</td>
</tr>
</tbody>
</table>

(d) 88 x 10 Bicluster from OPSM

- Table 3-a)
- Table 3-b)
- Table 3-c)
- Table 3-d)
value of GO from randomly chosen 25-RNC does not decrease substantially as \( \delta \) increases as in Table 4.

Thus we can say that our algorithm is robust to the noise. Also we can see that allowing high level of \( \delta \) is not necessary and 1.7 is sufficient as \( \delta \) value. Secondly we measured the p-values while increasing \( qnum \) p-value does not decrease further, which means that \( qnum \) doesn’t have to be bigger than 100. We recommend that \( qnum < 200 \) if the number of samples of microarray test dataset is less than 170.

5. CONCLUSION

RN-Clustering is robust to experimental noise by unique ranging, tree forming and queuing algorithm. These also guarantee the diversity of the results. RN-Clusters are proven to have significant level of functional association by GO validation.

The rapid increase in large-scale gene expression data provides us for tremendous chances to integrate many microarray datasets and identify a set of biclusters which are functional modules. For future works, we’ll extend and apply RN-Clustering to integrated microarray datasets, and identify genetic regulation of specific biological pathways under a variety of conditions.

6. REFERENCES


