Exhaustive Search Method of Gene Expression Modules and Its Application to Human Tissue Data

Yoshifumi Okada, Kosaku Okubo, Paul Horton, and Wataru Fujibuchi

Abstract— **Recently, several biclustering methods have been suggested to discover modules in gene expression data matrices. A module, namely a bicluster, is defined as a subset of genes that exhibit a highly correlated expression pattern over a subset of conditions. Most existing methods produce sub-optimal solutions by approximation approaches since biclustering requires combinational searches for pairs of genes and conditions in a large search space. In this paper, we propose a fast biclustering method, *BiModule*, that exhaustively searches modules in real time based on a closed itemset mining algorithm. We show that BiModule can discover functionally-enriched biclusters better than the approximation approaches, while maintaining a comparably fast running time. In addition, we apply BiModule to a gene expression data matrix obtained from various human tissues/cells and demonstrate that genes found in each bicluster well reflect the functions and morphology of specific tissues/cells.**

***Index Terms*—biclustering, closed itemset, gene expression module, LCM.**

I. INTRODUCTION

The advent of high-throughput gene expression profiling techniques such as cDNA microarray has made it possible to simultaneously analyze expression levels for thousands of genes under a number of different conditions. Gene expression data is usually arranged in the form of a matrix, in which each row corresponds to a gene, each column corresponds to a condition and each element represents an expression level of a gene under a condition. The typical approach to analyze gene expression data is clustering such as hierarchical clustering and *k*-means clustering. Clustering divides genes into mutually exclusive groups with similar expression patterns across all conditions. However, one would expect that many gene groups might exhibit similar expression patterns only under a specific set of conditions. We refer to such a group as a *gene expression module*, or simply *module*.

Recent studies have focused on the problem of discovering hidden module structures in large expression matrices. This involves simultaneous clustering of genes and conditions and is thus an instance of *biclustering*. Using that terminology, the modules we seek can be referred to as *biclusters*. The aim of biclustering is to identify subset pairs (each pair consisting of a subset of genes and a subset of conditions) by clustering both the rows and the columns of an expression matrix. This is a combinatorial search problem in an exponentially large search space. Hence most existing biclustering algorithms are based on greedy or stochastic heuristic approaches and produce possibly sub-optimal solutions. Cheng and Church [1] gave a greedy algorithm that searches biclusters with a mean squared difference less than δ. Tanay *et al.* [2], [3] identified biclusters based on a bipartite graph-based model and using a greedy approach to add/remove vertices to find maximum weight sub-graphs. Ben-Dor *et al.* [4] proposed a randomized algorithm to find the order-preserving sub-matrix (OPSM) in which all genes have same linear ordering. Ihmels *et al.* [5] proposed a random Iterative Signature Algorithm (ISA) which uses gene signatures and condition signatures to find biclusters with both up and down-regulated expression values. Such approximation approaches may produce many similar biclusters since the explored regions of the search space may be limited. We expect that interesting biclusters (or at least their cores) can be obtained by exhaustively enumerating every maximal bicluster.

In this paper, we propose a fast biclustering method, *BiModule*, that allows fast exhaustive search of maximal biclusters from discretized expression data matrices. This is based on a closed itemset mining algorithm that has been actively studied in data-mining and knowledge discovery. A well known application of closed itemset mining is pattern discovery from large point of sale (POS) data. The aim here is to find the maximal sets of items purchased by customers at the same time, namely closed itemsets. In the same manner, we can obtain maximal biclusters by finding closed itemsets for conditions over which genes have identical discretized expression values. BiModule achieves exhaustive enumeration of maximal biclusters in polynomial time by a fast and efficient algorithm called LCM (Linear time Closed itemset Miner) [8], [9]. Prelic *et al*. [10] developed an exhaustive biclustering

Manuscript received May 31, 2007.

Y. Okada is with Computational Biology Research Center (CBRC), Advanced Industrial Science and Technology (AIST), AIST Tokyo Waterfront Bio-IT Research Building, 2-42 Aomi, Koto-ku, Tokyo, 135-0064, Japan (e-mail: okada-yoshifumi@aist.go.jp).

K. Okubo is with National Institute of Genetics, Research Organization of Information and Systems, Yata 1111, Mishima, Shizuoka, 411-8540, Japan (e-mail:kokubo@genes.nig.ac.jp).

P. Horton is with Computational Biology Research Center (CBRC), Advanced Industrial Science and Technology (AIST), AIST Tokyo Waterfront Bio-IT Research Building, 2-42 Aomi, Koto-ku, Tokyo, 135-0064, Japan (e-mail: horton-p@aist.go.jp).

W. Fujibuchi is with Computational Biology Research Center (CBRC), Advanced Industrial Science and Technology (AIST), AIST Tokyo Waterfront Bio-IT Research Building, 2-42 Aomi, Koto-ku, Tokyo, 135-0064, Japan (corresponding author to provide phone: +81-3-3599-8619; fax: +81-3-3599-8081; e-mail: w.fujibuchi@aist.go.jp).

Transaction	Item								
1	A	B			E	F	G		I
2		B	C	D	E				
3	A	B					G	H	I
4	A						G		I
5		B					G		I
6		B							

Fig.1: Transaction database.

Gene	Condition								
	A	B	C	D	E	F	G	H	I
1	1	1	0	0	1	1	1	0	1
2	0	1	1	1	1	0	0	0	0
3	1	1	0	0	0	0	1	1	1
4	1	0	0	0	0	0	1	0	1
5	0	1	0	0	0	0	1	0	1
6	0	1	0	0	0	0	0	0	0

Fig.2: Gene expression table.

method for binary expression data based on a divide-and-conquer algorithm. In contrast, BiModule can address multi-valued expression data as well as binary data. In this study, we conducted benchmark tests using *S. cerevisiae* expression data to compare the performances of salient methods with that of BiModule. In addition, we applied BiModule to an expression data from various human tissues/cells and investigated the biological meaning of the generated biclusters. The rest of the paper is organized as follows. In the next section, we give the definition of "closed itemset" and explain its application to biclustering. In section III, we describe the procedure of BiModule. In section IV, we compare the performance of BiModule with other prominent methods by conducting enrichment analysis on four different kinds of functional information: Gene Ontology terms, protein-protein interaction pairs, functional motifs and metabolic pathways. In section V, we show the results of enrichment analysis and biological interpretations on biclusters discovered from human tissue/cell expression. In section VI, we summarize and close with our conclusions from this study and some ideas for future work.

II. CLOSED ITEMSET MINING AND BICLUSTERING

A. Closed Itemsets

A closed itemset mining searches co-occurring *items* from a *transaction database* as shown in Fig.1. First, we define the closed itemset more formally. Let I be a set of items. A transaction database is a subset of the power set of I. In other words, it is a set of sets $T_i = \{t_1, t_2, ..., t_m\}$ of items from I. Each T_i is called a *transaction*. Fig. 1 is an example of a transaction database that consists of six transactions and nine items. A subset of I is called an *itemset*. For an itemset P, a transaction which contains (*i.e.* is a superset of) P is called an *occurrence* of P. The set of occurrences of P is denoted $S(P)$. The size of $S(P)$ is called the *support* of P, denoted by $supp(P)$. Given a constant θ, called a *minimum support*, itemset P is *frequent* if $supp(P) \geq \theta$. A *closed itemset* is maximal for its set of occurrences. In other words, an itemset P is a closed itemset if there exists no itemset P' such that $P \subset P'$ and $supp(P)=supp(P')$. For example, in the transaction database in Fig.1, the itemset $\{A, G, I\}$ is a closed itemset because this is the maximum set of items shared by transactions $\{1, 3, 4\}$. For a minimum support of 2, the itemset $\{A, G, I\}$ is a frequent closed itemset because $supp(A, G, I)>2$. $\{A, G\}$ is not a closed itemset since all of the transactions including items A and G also include the item I.

Next we describe how we apply the closed itemset problem to biclustering gene expression matrices. For simplicity, suppose each gene expression value is represented by 0 or 1 (up or down regulation). In this context, Fig.1 can be transformed to a table such as Fig.2. This is the same form as a typical gene expression matrix, where a gene (row) corresponds to a transaction and a condition (column) corresponds to an item. If a condition is activated by a specific gene, the corresponding element takes a value of 1. A set of conditions in a bicluster is a maximal set of conditions in which a certain set of genes exhibit common expression values. For example, a condition set $\{A, G, I\}$ is a set of conditions composing a bicluster because this is a maximal set with a value of 1 for genes $\{1, 3, 4\}$. In this way, closed itemset mining corresponds to extracting condition sets composing biclusters under the restriction of using discretized expression values. However, the above formulation can deal with only the binary states, such as up or down regulation. Prelic *et al*. [10] proposed a biclustering algorithm based on binary discretization of gene expression matrices. Such a rough discretization may blur the original structure of gene expression matrices and consequently obscure biologically meaningful modules. In contrast, our method can deal with multi-valued discretization levels and thus can discover not only biclusters with constant values but also biclusters with expression patterns changed over conditions. The former bicluster is called a constant bicluster and the latter is referred to as an additive bicluster.

B. Closed Itemset Enumeration Algorithm

To date, several efficient algorithms have been proposed to enumerate every closed itemset from a transaction database [6]-[9]. We chose to use LCM (Linear time Closed itemset Miner), which received the best implementation award in the data-mining contest FIMI'04 [9]. LCM achieves a fast enumeration of closed itemsets using a unique technique called prefix preserving closure extension (ppc extension for short), which is an extension from a closed itemset to another closed itemset. The extension induces a search tree on the set of frequent closed itemsets, thereby enabling the completely enumeration of closed itemsets without duplication. Because of this efficient traversal of itemsets LCM can avoid redundant calculation without keeping a list of previously obtained closed

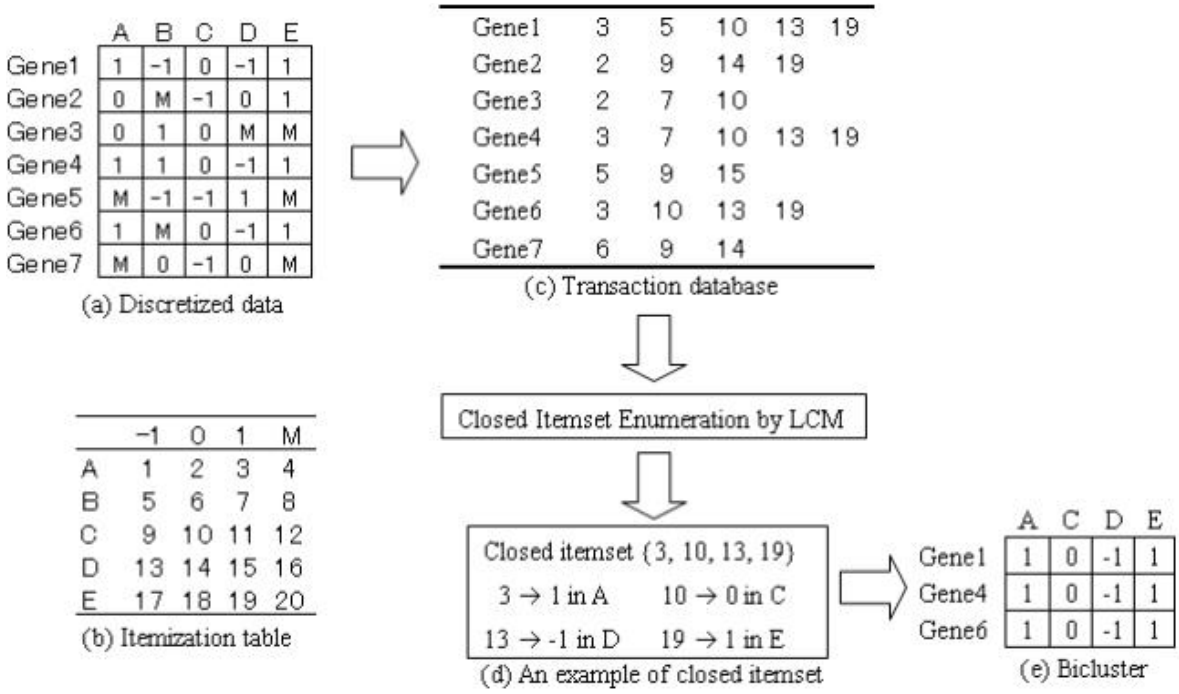

Fig.3: The procedure of BiModule.

itemset. Hence, the memory use of LCM does not depend on the number of frequent closed itemsets. The computational time of LCM is theoretically linear in the number of frequent closed itemsets. (*cf.* [8] for a detailed description of LCM). The LCM program is available from [11].

III. METHODS

Fig.3 is the procedure of BiModule. BiModule consists of the four parts: 1) normalize and discretize gene expression data, 2) generate a transaction database, 3) enumerate biclusters (closed itemsets) and 4) filter out unnecessary biclusters.

A. Normalization and Discretization

In our procedure expression data from each microarray sample are linearly normalized to have mean 0 and variance 1, and this normalized data is discretized. Fig.3a illustrates an example of a discretized data matrix, where the number of levels is set to 3, namely (-1, 0, 1), for simplicity. 'M' in this matrix denotes a missing value. The interval for each expression level is given by uniformly dividing the difference between the maximum and the minimum in the normalized data. However, if the maximum or the minimum takes an extreme value (outlier), most of the data will be unevenly assigned to a few levels because unduly large intervals are needed to include the outlier. Hence, we perform the following processing for outliers before discretization. Data farther than a threshold (3 standard deviations in this work) are regarded as outliers and are temporarily removed. The rest of data are renormalized and if the renormalized data contains new outliers the procedure is repeated until no outliers remain. At this point the temporarily removed outliers are given values equal to the corresponding extreme value of the final normalized data (minimum for outliers below the mean, maximum for outliers above the mean). The discretization is performed on this data.

B. Transaction Data

We prepare an itemization table that contains IDs representing each discretization level in each condition. Fig.3b shows the itemization table for the discretized data in Fig.3a. In this figure, for example, discretization level '1' in condition 'B' is specified by ID '7'. Subsequently, the discretized data are converted to a transaction database as shown in Fig.3c by reference to the itemization table. The transaction data for a gene is represented by a set of IDs (corresponding to items), where IDs for missing values are not included. In this manner, an item in the multi-valued discretization indicates a combination of a condition and a discretized value. Thus, itemization enables us to extract additive biclusters as well as constant biclusters.

C. Enumeration of Biclusters

We use LCM to enumerate closed itemsets and their corresponding biclusters. The input to LCM is a transaction database and a minimum support value, *i.e.*, the minimum number of genes in extracted biclusters. The output is closed itemsets with IDs as shown in Fig.3d. In this figure, an example of a closed itemset enumerated by LCM is shown. We can convert the IDs to the condition names and discretized values by reference to the itemization table. In Fig.3d, it is shown that the closed itemset {3, 10, 13, 19} can be converted to the conditions

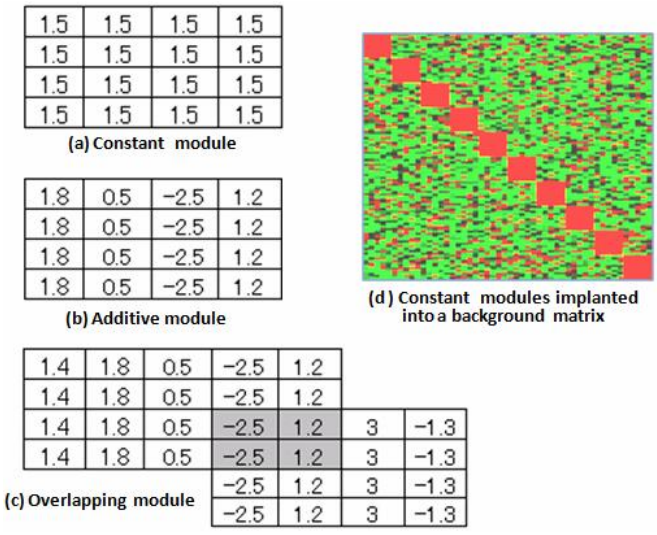

Fig.4: Example of the three types of modules, (a) *constant*, (b) *additive*, and (c) *overlapping*. (d) Example of constant modules implanted into a background matrix

A, C, D and E taking discretized values 1, 0, -1 and 1, respectively. Corresponding biclusters can be completed by selecting the genes which match the required discretized value for each condition.

D. Selection of Biclusters

In most cases, a large number of biclusters are enumerated, *e.g.*, 115,737 for a 2000×200 matrix with the parameters L=7, Mg=40 and Mc=5 (see *E. Implementation*). However, most of them are small biclusters and most of their elements overlap with larger biclusters. We filter out such small biclusters by the following procedure. First, the enumerated biclusters are sorted using the following score *F*:

$$F(B) = A \times \log_2(g) \times \log_2(c). \qquad (1)$$

In (1), *B* is a bicluster, *A* represents the average of the absolute values of the discretized values in the conditions included. *g* and *c* are the number of genes and the number of conditions, respectively. Subsequently, biclusters overlapping more than 25% with a bicluster having higher score are filtered out and the remaining biclusters are output to the user.

E. Implementation

We implemented the procedure above in Java except for the closed itemset enumeration by LCM. The LCM program is implemented in the C language [11]. The input to BiModule is a pre-normalized gene expression matrix and three parameters: L, Mg and Mc, where L is the number of discretization levels, Mg is a minimum number of genes and Mc is a minimum number of conditions. As for the number of discretization levels, users can choose from L=3, 5 and 7. In this study, we use L=7 because BiModule shows the best performance with this setting, both in terms of the extraction accuracy of modules and running time [14].

IV. Performance with Benchmark Datasets

A. Compared Biclustering methods

We compare the performance of BiModule with that of other prominent biclustering methods using benchmark datasets. The test platform is a desktop PC with Pentium 4, 2.4GHz CPU and 1GB RAM running the Linux operating system. The methods selected here are: Order Preserving Submatrix Algorithm (OPSM) [4], Iterative Signature Algorithm (ISA) [5], Samba [2], [3], the Cheng and Church algorithm (CC) [1] and Bimax [10]. These are all based on greedy search strategies. We downloaded the software, BicAt developed by Barkow *et al.* [12] and EXPANDER developed by Shamir *et al.* [13]. BicAt implements Bimax, ISA, CC and OPSM in Java. Samba is available in EXPANDER. In our comparative test, the parameters for these algorithms were set to the values recommended in the corresponding publications.

B. Experimental Results

Synthetic Data

Prelic *et al*. provides synthetic datasets that contain the sets of data matrices with artificially-implanted modules. In the previous report [14], we compared the performances of the selected methods with that of BiModule using the synthetic datasets, where extraction accuracies of constant, additive and overlapping modules (illustrated in Fig.4) were evaluated according to two performance scores, *relevance* and *recovery*. As a result, it was shown that BiModule can discover modules with higher accuracy than competing methods, for any of the three module types [14].

S. cerevisiae Data

Prelic *et al*. also provides *S. cerevisiae* expression data containing 2,993 genes and 173 conditions [15], derived from Gasch's dataset [16]. This includes expression data for several conditions under 13 different environmental stresses such as heat shock, nitrogen depletion etc. In this section, we use this dataset as a benchmark and compare the functional enrichment of biclusters discovered by each method. The input parameters used for BiModule are L=7, Mg=40 and Mc=5. For all of the biclustering methods, we filtered out biclusters overlapped by more than 25% with a larger bicluster and output the resultant biclusters up to 100 in descending order of size.

In [14], we presented the proportions of biclusters containing significantly over-represented Gene Ontology (GO) terms [17] and protein-protein interaction (PPI) pairs. The GO enrichment test was performed using a web tool FuncAssociate [18], while for PPI pairs, we used interaction data obtained from the DIP database [19]. The significance of PPI enrichment was calculated by using the z-test to check whether the proportion of non-interacted pairs in each bicluster is significantly smaller than the expected values for random gene groups under normal distribution model. In this work, we also test for the enrichment InterPro motifs [20] and KEGG pathways [21]. The experiments were performed using a web-based tool GENECODIS [22] that allows integrative extraction of frequently co-occurring annotations in a given gene list across different sources such as GO terms, InterPro motifs, KEGG

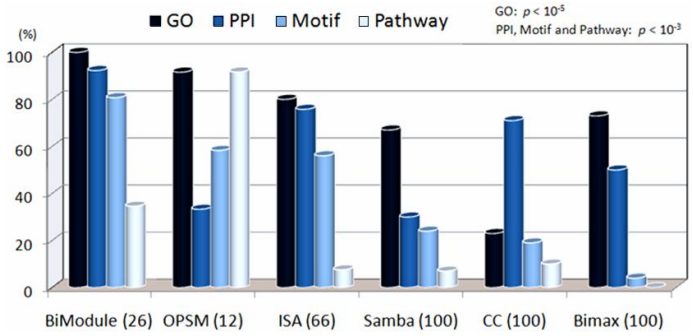

Fig.5: Proportion of biclusters significantly enriched by GO terms (GO), protein-protein interactions (PPI), InterPro motifs (Motif) and KEGG metabolic pathway (Pathway).

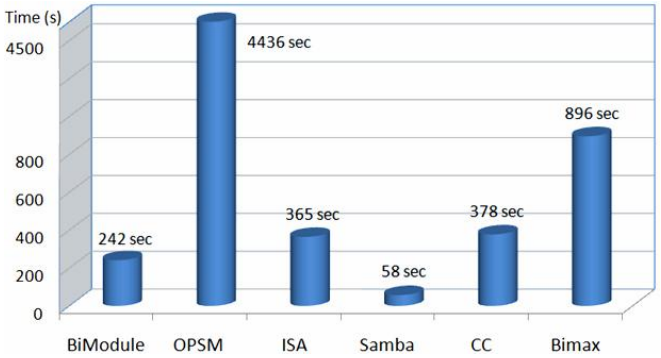

Fig.6: Running time (in second) of the methods.

pathways and Swiss-Prot keywords. GENECODIS uses the *apriori* algorithm [23] to mine frequent itemsets, *i.e.*, co-annotations satisfying a minimum support value (see section II) and then a statistical test (the hypergeometric distribution or the χ^2-test) is applied to identify significant combinations of annotations. The *p* values can then be adjusted for multiple tests using a simulation-based correlation approach [18] or the false discovery rate (FDR) method [24]. In this paper, we use the hypergeometric distribution for statistical tests and the FDR method for adjustments of multiple tests.

Fig.5 summarizes the proportion of biclusters with one or several over-represented GO terms ($p<10^{-5}$), PPI pairs ($p<10^{-3}$), InterPro motifs ($p<10^{-3}$) or KEGG pathways ($p<10^{-3}$) in the selected methods, which are hereafter referred to as GO, PPI, Motif and Pathway. As can be seen in this figure, all biclusters discovered by BiModule contain significantly over-represented GO terms. This is the best score among the compared methods. BiModule also attains the highest scores for PPI and Motif, 92.3% and 80.8% respectively. As for Pathway, although BiModule is second behind OPSM, an obvious difference in the variations of the pathway names was observed; all of the significant biclusters by OPSM were associated with ribosome synthesis-related pathways and furthermore were similar to each other, namely overlapped in most of their genes and conditions. In contrast, BiModule generated not only ribosome synthesis-related biclusters but also biclusters corresponding to

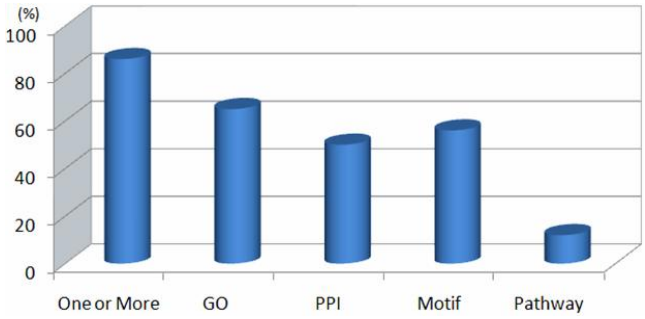

Fig.7: Results of enrichment analysis on biclusters from human tissue/cell data. In addition to GO, PPI, Motif and Pathway, proportion of biclusters significantly enriched in one or more themes (One or More) are shown.

distinct metabolic pathways such as the sulfur metabolism or the nitrogen metabolism. These results suggest that BiModule can discover diverse modules in actual problem.

Fig.6 shows the running times (in seconds) of the respective methods for this dataset. Among them, Samba is the fastest and BiModule is second only to this. It is noteworthy that the running time of BiModule by the exhaustive approach is comparable to that of Samba using the probabilistic approach and furthermore is faster than the other approximation methods.

V. Application to the Human Tissue Data

Currently, we have been carrying out comprehensive analysis of the human transcriptome toward elucidation of the functional and morphological diversity in various tissues/cells. As a part of this study, we have applied BiModule to gene expression data of human tissues/cells measured by iAFLP method (introduced Amplified Fragment Length Polymorphism) [25]. In this section, the results are evaluated by functional enrichment analysis and an investigation of the function of genes found in the generated biclusters.

A. Dataset and Preprocessing

The iAFLP has high specificity and sensitivity for transcript detection and a throughput level comparable to that of DNA microarray hybridization [25]. The iAFLP expression data used here is a matrix of 20,703 gene (probe) profiles measured under 83 tissues/cells of adult humans. Since iAFLP quantitates expression levels of multiple samples based on a gene by gene approach; as a preprocessing step, the expression values of each gene are normalized to a normal distribution with mean 0 and variance 1. The input parameters used for BiModule were L=7, Mg=50 and Mc=5. As a result, 166 biclusters were obtained and the top 100 scoring biclusters were used for the analysis.

B. Statistical Annotation Enrichment Analysis

First, functional enrichment analyses for generated biclusters were conducted on four different types of functional information (hereafter called *themes*): GO terms, PPI pairs, InterPro motifs and KEGG pathways. In this experiment, GENECODIS was used to find co-occurring annotations for all themes except PPI. PPI was analyzed in the same manner as

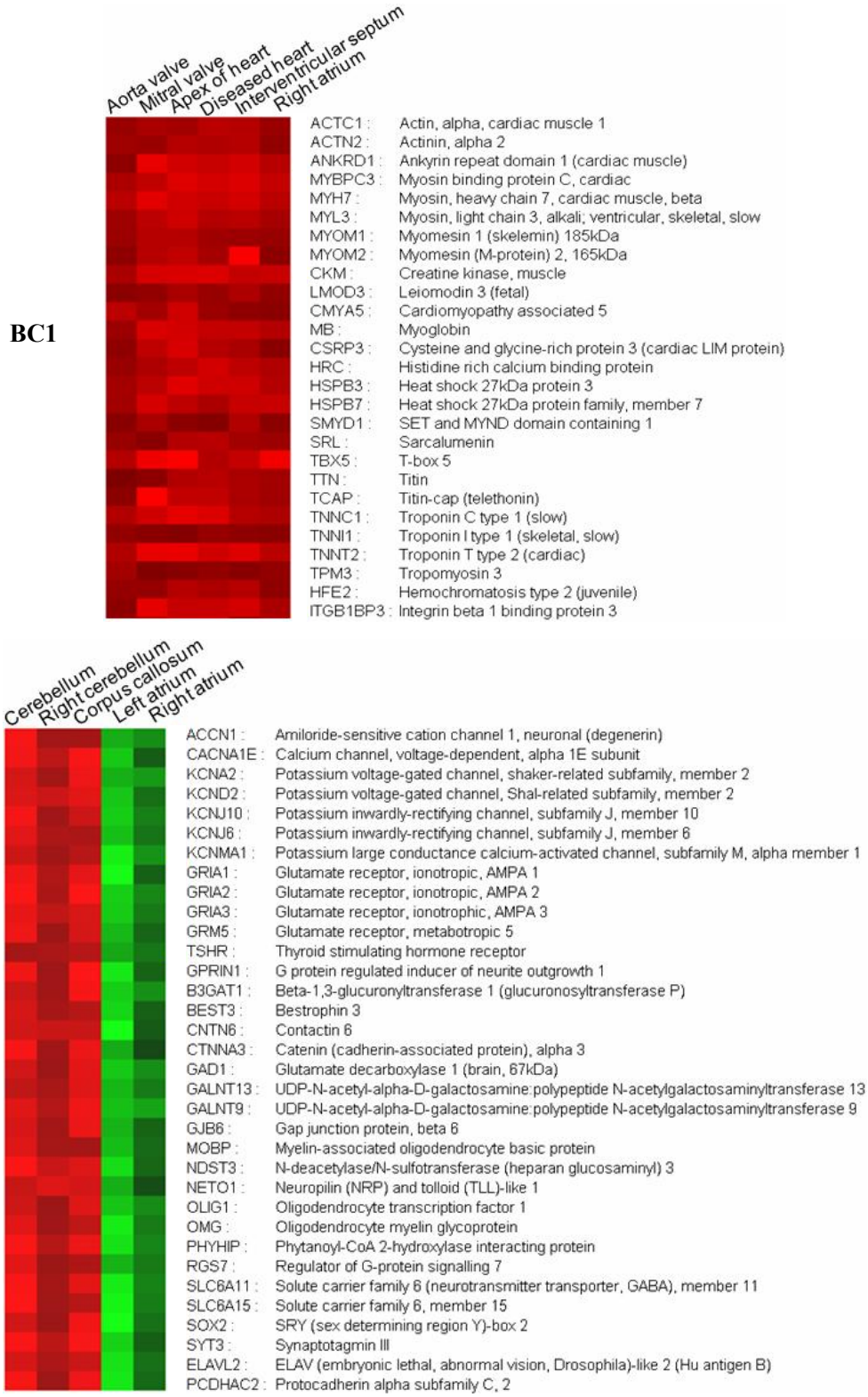

Fig.8: Examples of biclusters discovered by BiModule. The upper half shows a bicluster specific for cardiac muscles, and the lower shows a bicluster consisting of genes that are up-regulated in brain tissues and repressed/down-regulated in cardiac muscles

shown in section IV (see also [14]), using interaction data obtained from the HPRD database [26]. The results are shown in Fig.7, where "One or More" indicates the number of biclusters judged to be significant for one or more themes. As

seen in this figure, 86 biclusters have significant annotations ($p < 0.05$) in one or more theme. The number of significant biclusters in each theme is 65 for GO, 50 for PPI, 56 for Motif and 12 for Pathway, respectively. These results are substantially smaller in number compared to those obtained with the *S. cerevisiae* data. This may be due to the incompleteness of annotation for human biological data. For example, about 60% of the GO term annotations for human gene products are inferred by computer programs, in contrast all annotations for *S. cerevisiae* are based on biological evidences.

C. Biological Interpretations

Subsequently, we investigate the functions of genes in the significant biclusters obtained by the above enrichment analysis. In these biclusters, the same types of tissues/cells tend to be assigned into identical biclusters and the functions of the tissues/cells are well characterized by the genes contained in each bicluster. Below, we discuss the biological meaning of the two biclusters shown in Fig.8. These biclusters are hereafter referred to as BC1 (upper) and BC2 (lower), respectively. Although these biclusters actually contained 43 genes for BC1 and 69 genes for BC2, in Fig. 8 we show the genes that seem to be associated with the tissues/cells of each bicluster.

BC1 consists of 27 genes expressed in cardiac muscle cells. In this bicluster, besides genes involved in myofibril formation such as actin, myosin and troponin genes, several novel genes discovered in recent years are also contained. For example, Papanikolaou *et al*. [27] reported that the expression of HFE2 is restricted to liver, heart and skeletal muscle, similar to that of hepcidin, a key protein implicated in iron metabolism. Nojiri *et al*. [28] showed that ITGB1BP3, a modulator of muscle proliferation and differentiation, is specifically up-regulated for cardiac oxidative stress.

The bicluster BC2 is composed of genes that are up-regulated in brain tissues (cerebellum, right cerebellum and callosum) but are repressed/down-regulated in cardiac tissues (left atrium and right atrium). Potassium channels are important in shaping the action potential, and in neuronal excitability and plasticity. This bicluster has several potassium channel-related genes, KCNA2, KCND2, KCNJ6, KCNJ10 and KCNMA1. AMPA glutamate receptors are composed of four subunits, GRIA1-GRIA4, that are believed to play critical roles in synaptic transmission. BC2 contains three of them, GRIA1, 2 and 3. Besides the above genes, BC2 includes several brain-specific genes such as GALNT9, 13[29] and MOBP [30].

As demonstrated above, BiModule enables us to find genes co-activated in certain tissue/cell types (such as BC1) as well as genes regulated exclusively between different tissue groups (such as BC2). Besides these biclusters, we obtained several interesting biclusters such as biclusters specific for blood cells, intestine cells or lymphoid cells. In future work, the functions of unannotated genes in generated biclusters could be investigated according to sequence homology/orthology or literature surveys, and so on.

VI. CONCLUSIONS

We proposed a new biclustering method, BiModule, that allows exhaustive search of gene expression modules based on a fast closed itemset enumeration algorithm. In this study, we use the *S. cerevisiae* expression data as a benchmark for performance comparisons with six prominent methods. The performance was evaluated based on functional enrichment analysis in the generated biclusters. As a result, BiModule exhibited the best performance in the enrichment analyses on GO terms, protein-protein interaction pairs and functional motifs. Moreover, the running time of BiModule was comparable to those of approximation algorithms. We also applied BiModule to human tissue data. As a result, we obtained intriguing biclusters derived from not only single organs but also multiple organs, and we were able to confirm that those biclusters include important genes characterizing the functions and roles of the respective tissues/cells.

BiModule does have some limitations. BiModule searches for biclusters in which the rows in each bicluster are completely identical. Therefore, if a large amount of noise is included in some elements of a true module, the observed expression value may not fall into the desired interval during the discretization process. In such case, true modules will be subdivided into some smaller biclusters. Furthermore, since BiModule cannot extract biclusters with a gene size smaller than Mg, such small biclusters are ignored by the process of the closed itemset enumeration. Consequently, with Mg set to an excessively large value, BiModule may not be able to properly detect small biologically meaningful biclusters.

As future work, in order to reduce noise in gene expression data, we will incorporate a de-noising method such as [31] to BiModule. We are currently implementing a parallel computing version of BiModule toward the development of a web tool.

REFERENCES

[1] Y. Cheng and G. Church, "Biclustering of expression data", *Proc. Int. Conf. Intell. Syst. Mol. Biol*., 2000, pp. 93–103.

[2] A. Tanay, R. sharan, and R. Shamir, "Discovering statistically significant biclusters in gene expression data", *Bioinformatics*, **18** (Suppl. 1), 2002, pp. S136–S144.

[3] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir, "Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data", *Proc. Natl Acad. Sci. USA*, 101, 2004, pp. 2981–2986.

[4] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: the order-preserving sub-matrix problem", *Proc. of the 6th Annual Int. Conf. on Computational Biology*, ACM Press, New York, NY, USA, 2002, pp. 49–57.

[5] J. Ihmels, S. Bergmann, and N. Brkai, "Defining transcription modules using large-scale gene expression data", *Bioinformatics*, **20** (13), 2004, pp. 1993–2003.

[6] J. Pei, J. Han, and R. Mao, "CLOSET: An efficient algorithm for mining frequent closed itemsets", *Proc. of 2000 ACM-SIGMOD Int. Workshop Data Mining and Knowledge Discovery (DMKD'00)*, Dallas, TX, 2000, pp. 11-20.

[7] M. J. Zaki, C. Hsiao, "An efficient algorithm for mining frequent closed association rule mining", *Proc. of 2002 SIAM Data Mining Conf,* 2002.

[8] T. Uno, T. Asai, Y. Uchida, and H. Arimura, "An efficient algorithm for enumerating closed patterns in transaction databases", *Lecture Notes in Artificial Intelligence*, **3245**, 2004, pp. 16-31.

[9] T. Uno, M. Kiyomi, and H. Arimura, "LCM ver.2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets", *IEEE ICDM'04 Workshop FIMI'04*, 2004.

[10] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hening, L. Thiele, and E. Zizler, "A Systematic comparison and evaluation of biclustering methods for gene expression data", *Bioinformatics*, **22** (9), 2006, pp. 1122-1129.

[11] LCM. ver2: http://research.nii.ac.jp/~uno/codes-j.html

[12] BicAt: http://www.tik.ee.ethz.ch/sop/bicat/

[13] EXPANDER: http://www.cs.tau.ac.il/~rshamir/expander/

[14] Y. Okada, W. Fujibuchi and P. Horton, "Exhaustive search of maximal biclusters in gene expression data", *Proc . of Int. MultiConf. Of Eng. and Compt Sci. 2007*, **2**(1), Hong Kong, China, 2007, pp. 307-312

[15] http://www.tik.ee.ethz.ch/sop/bimax/SupplementMaterials,Biclustering.html

[16] A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel-Harel, M.B. Eisen, G. Storz, D. Botstein, and P.O. Brown, "Genomic expression programs in the response of yeast cells to environmental changes", *Mol. Biol. Cell.* **11**, 2000, pp. 4241-4257.

[17] Gene Ontology Consortium: http://www.geneontology.org/

[18] G.F. Berriz, O.D. King, B. Bryant, C. Sander, and F.P. Roth, "Charactering gene sets with FuncAssociate", *Bioinformatics*, **22** (10), 2003, pp. 1282-1283

[19] DIP database: http://dip.doe-mbi.ucla.edu/

[20] http://www.ebi.ac.uk/interpro/

[21] http://www.genome.jp/kegg/

[22] C.S. Pedro, C. Monica, T. Francisco, M.C. Jose and P.M. Pascal, "GENECODIS: a web-based tool for finding significant concurrent annotations in gene lists", *Genome Biology*, **8**:R3, 2007.

[23] R. Agrawal, T. Imielinski, A. Swami, "Mining association rules between sets of items in large database, *Proc. Of the ACM SIGMOD Int. Conf. on Management of Data,* New York, USA, 1993, pp. 207-216.

[24] Y. Benjamini, Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing", *J. Roy. Statist. Soc. Ser.*, **B 57**, 1995, pp. 289-300.

[25] S. Kawamoto, T. Ohnishi, H. Kita, O. Chisaka and K. Okubo, "Expression profiling by iAFLP: A PCR-based method for genome-wide gene expression profiling", *Genome Res.*, **9**, 1999, pp. 1305-1312.

[26] S. Mathivanan *et al*., "An evaluation of human protein-protein interaction data in the public domain", *BMC Bioinformatics*, **7**(Suppl 5):S19, 2006

[27] G. Papanikolaou *et al*., "Mutations in HFE2 gene iron overload in chromosome 1q-linked juvenile hemochromatosis", *Nature Genet.*, **36**(1), 2004, pp. 77-82.

[28] Nojiri *et al*., "Oxidative Stress Causes Heart Failure with Impaired Mitochondrial Respiration", *J. Biol. Chem.*, **281**(44), 33789-33801

[29] Y. Zhang *et al*., "Cloning and characterization of a new human UDP-N-acetyl-alpha-D-galactosamine:polypeptide N-acetylgalactosaminyltransferase, designated pp-GalNAc-T13, that is specifically expressed in neurons and synthesizes GalNAc alpha-serine/threonine antigen. J. Biol. Chem., **278**, 2003, pp. 573-584.

[30] Y. Yamamoto *et al*., "Cloning and expression of myelin-associated oligodendrocytic basic protein. A novel basic protein constituting the central nervous system myelin. *J. Biol. Chem.,* **269**(50), 1994, pp. 31725-31730.

[31] T. Kato, Y Murata, K. Miura, K. Asai, P. Horton, K. Tsuda and W. Fujibuchi, "Network-based de-noising improves prediction from microarray data", *BMC Bioinformatics*, **7**(Suppl 1): S4, 2006