Original Paper

A Biclustering Method for Gene Expression Module Discovery Using a Closed Itemset Enumeration Algorithm

Yoshifumi Okada,† Wataru Fujibuchi† and Paul Horton†

A *gene expression module* (*module* for short) is a set of genes with shared expression behavior under certain experimental conditions. Discovering of modules enables us to uncover the function of uncharacterized genes or genetic networks. In recent years, several biclustering methods have been suggested to discover modules from gene expression data matrices, where a bicluster is defined as a subset of genes that exhibit a highly correlated expression pattern over a subset of conditions. Biclustering however involves combinatorial optimization in selecting the rows and columns composing modules. Hence most existing algorithms are based on heuristic or stochastic approaches and produce possibly sub-optimal solutions. In this paper, we propose a novel biclustering method, BiModule, based on a closed itemset enumeration algorithm. By exhaustive enumeration of such biclusters, it is possible to select only biclusters satisfying certain criteria such as a user-specified bicluster size, an enrichment of functional annotation terms, etc. We performed comparative experiments to existing salient biclustering methods to test the validity of biclusters extracted by BiModule using synthetic data and real expression data. We show that BiModule provides high performance compared to the other methods in extracting artificially-embedded modules as well as modules strongly related to GO annotations, protein-protein interactions and metabolic pathways.

1. Introduction

DNA microarray technology 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. Murali and Kasif presented a random algorithm xMotif [6)].

Our goal is to develop a fast biclustering method for enumerating every interesting bicluster within a reasonable time. We conjecture that interesting biclusters (or at least their cores) can be obtained by enumerating maximal biclusters which have identical condition label and discretized expression values; a prob-

† Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST)

lem which can be solved in polynomial time. By exhaustive enumeration of such biclusters, it is possible to select only biclusters satisfying a certain criterion such as a user-specified bicluster size, an enrichment of functional annotation terms, etc. Here, we propose a new biclustering method, BiModule, that enumerates biclusters in polynomial time based on a closed itemset mining algorithm that has been actively studied in data-mining. Comparative experiments with salient biclustering methods are performed to test the validity of biclusters extracted by BiModule using synthetic data and real expression data. We show that BiModule provides high performance compared to the other methods in extracting artificially-embedded modules as well as modules strongly related to GO annotations, protein-protein interactions and metabolic pathways.

2. Closed Itemset Mining and Biclustering

2.1 Bicluster types

We aim to discover three types of biclusters (modules) introduced by Prelic, et al.[7]: *constant*, *additive* and *overlapping*. As shown in **Fig. 1** (a), a constant bicluster is a bicluster with a single constant expression value for all the elements of the matrix. Figure 1 (b) gives an example of an additive bicluster, where expression values vary over the conditions but all genes share the same value for any given condition. If two constant or additive biclusters share cells in d rows and d columns, such biclusters are called an overlapping biclusters with the overlap degree d. Figure 1 (c) is an example of overlapping biclusters with d=2, where 25% of the cells in the smaller bicluster Y are included in the larger bicluster X. We consider extracting these three types of biclusters from discretized expression data matrices. For this formulation, biclustering can be reduced to a data mining problem called closed itemset mining. In next section, we describe the relationship between closed itemset mining and biclustering.

2.2 Closed Itemset and Bicluster

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*. 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.

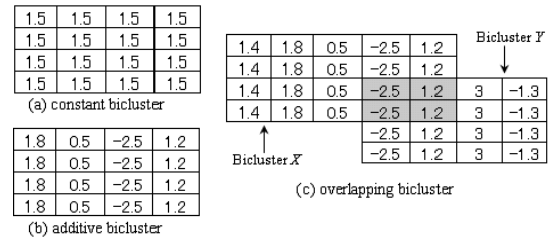

Fig. 1 Examples of the three types of biclusters: *constant*, *additive*, and *overlapping* are shown. In the overlapping bicluster example (c), the overlap degree d=2, and bicluster Y overlaps in 25% of cells with bicluster X.

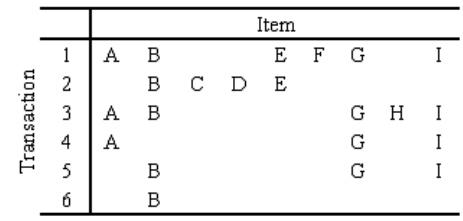

Fig. 2 Transaction database.

The set of occurrences of P is denoted $T(P)$. The size of $T(P)$ is called the *support* of P, denoted by $supp(P)$. Given a constant θ, called a *minimum support*, itemset P is frequent if $supp(P) \geqq \theta$. 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. 2**, 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. 2 can be transformed to a table such as **Fig. 3**. 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 activates 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,

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. 3 Gene expression table.

the 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 binary states, such as up or down regulation. Prelic, et al. and Tanay, et al. proposed a biclustering algorithm based on binary discretization of gene expression matrices [2),3),7)]. 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 (see Section 3.2 Itemization Table and Transaction Data).

2.3 Closed Itemset Enumeration Algorithm

To date, several efficient algorithms have been proposed to enumerate every closed itemsets from a transaction database [8)~11)]. We chose to use LCM (Linear time Closed itemset Miner), which received the best implementation award in the data-mining contest FIMI'04 [11)]. 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 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., 10) for a detailed description of LCM). The LCM program is available from Ref. 12).

3. Methods

Figure 4 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.

3.1 Normalization and Discretization

In our procedure expression data from each microarray condition are linearly normalized to have mean 0 and variance 1, and this normalized data is discretized. Figure 4 (a) 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.

3.2 Itemization Table and Transaction Data

We prepare an itemization table that contains IDs representing each discretization level in each condition. Figure 4 (b) shows the itemization table for the discretized data in Fig. 4 (a). 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. 4 (c) by reference to the itemization table. The transaction data for a gene is represented by a set of IDs, where IDs for missing value are not included. An ID can be regarded as an item that indicates a combination of a condition and a discretized expression value. Thus, closed itemset mining in such a transaction database en-

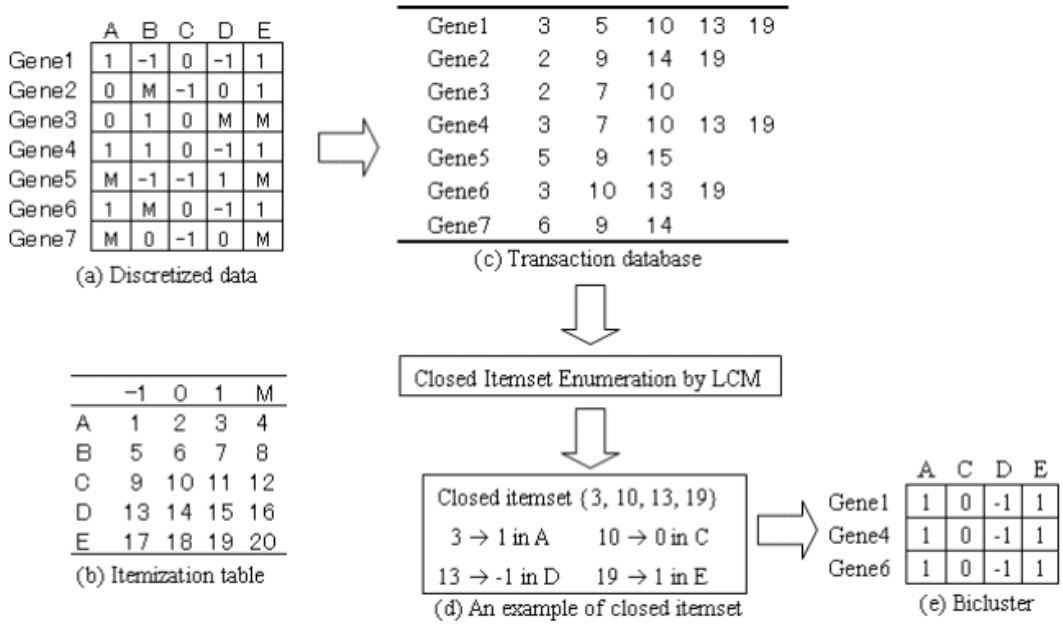

Fig. 4 The procedure of BiModule.

ables us to extract biclusters with multi-valued expression levels.

3.3 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. 4 (d). 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. 4 (d), it is shown that the closed itemset 3, 10, 13, 19 can be converted to the conditions 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.

3.4 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 Section 3.5 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:

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

In this function, B is a bicluster, A represents the average of the absolute values of the discretized values in the conditions included. A gives high priority to biclusters with strongly induced or repressed genes. g and c are the number of genes and the number of conditions, respectively. After sorting, biclusters whose cells overlap by more than 25% with a higher scoring bicluster are filtered out and the remaining biclusters are output to the user.

3.5 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 [12]. 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 extraction accuracy of modules and running time as shown in Section 4.3.

4. Results

We compare the performance of BiModule with those of other prominent biclustering algorithms on synthetic data and a real gene expres-

sion data. The test platform is a desktop PC with Pentium 4, 3 GHz CPU and 2 GB RAM running the Linux operating system.

4.1 Other Biclustering Algorithms

The selected algorithms are Bimax [7], Iterative Signature Algorithm (ISA) [5], Samba [2),3)], Cheng and Church algorithm (CC) [1], Order Preserving Submatrix Algorithm (OPSM) [4], and xMotif [6]. These are all based on greedy search strategies. We downloaded the software, BicAt developed by Barkow, et al. [13] and EXPANDER developed by Shamir, et al. [14]. BicAt implements Bimax, ISA, CC, OPSM and xMotif in Java. In EXPANDER, Samba is available. In our comparative test, the parameters for these algorithms were set to the values recommended in the corresponding publications.

4.2 Datasets

4.2.1 Synthetic Data

A comparative test was performed using a synthetic dataset provided by Prelic, et al. [15]. This dataset includes data matrices with three types of artificially-implanted modules: constant, additive and overlapping. In Prelic's dataset, an up-regulated constant module is considered. For the constant and additive module data, 10 modules, each a 10×5 matrix, are implanted into a 100×50 background matrix without overlap. As for the overlapping module data, 10 modules are implanted into a background matrix. In this study, we consider 11 different overlap degrees (d=0, 1,...,10), where the size of background matrix and modules vary from 100×100 to 110×110 and from 10×10 to 20×20, respectively.

Evaluation Measure

To assess the validity of biclusters extracted by the different programs, we use the following gene match score, proposed by Prelic, et al. Let M_1, M_2 be two sets of biclusters (or modules). The gene match score of M_1 with respect to M_2 is given by

$$S_G(M_1, M_2) = \frac{1}{|M_1|} \sum_{(G_1,C_1)\in M_1} \max_{(G_2,C_2)\in M_2} \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|},$$

where G and C are a set of genes and a set of conditions included in a bicluster, respectively. This measure reflects the average of the maximum match scores for all biclusters in M_1 with respect to the biclusters in M_2. Let M_{opt} be the set of implanted modules (true modules) and let B be the set of biclusters obtained by a biclustering method. $S(B,\ M_{opt})$ represents to what extent the generated biclusters match with true modules in the gene dimension. In contrast, $S(M_{opt},\ B)$ reflects how well each of the true modules are recovered by the biclustering method. Hereafter, we call these performance scores *relevance* and *recovery*, respectively. Both scores have a maximum value of 1, achieved when $M_{opt} = B$.

Evaluation Test

The parameters for BiModule were set to L=7, Mg=8 and Mc=4. We evaluated the performance of BiModule and other programs in the following two settings: 1) sensitivity against noise in non-overlapping modules, and 2) extraction accuracy on overlapping modules without noise. In the first setting, the constant and additive module matrices are used to test the effects of noise on the performance of the respective biclustering methods. Noise was modeled by adding random values derived from a normal distribution to all elements in each matrix. We generated 10 input matrices for each of several noise levels (different magnitude standard deviation used for generating Gaussian noise). The performance for each noise level was obtained by averaging over the 10 matrices. In the second setting, the performance was calculated using an overlapping module matrix for each overlap degree.

Figure 5 (a) and Fig. 5 (b) depict the performance on the constant modules and the additive modules, respectively. From these figs, we can see that ISA shows the best performance for each noise level in both relevance and recovery. BiModule also generally gives superior performance compared to the other methods, although with the additive modules there is a substantial decrease in recovery for high noise levels. Figure 5 (c) shows the results on the overlapping modules. As shown in this figure, BiModule gives the best scores in both the relevance and recovery. In particular, concerning recovery, we can see that BiModule perfectly identifies implanted modules over all tested overlap degrees. In contrast, for the other methods a low accuracy or a substantial decrease with noise is observed for both performance scores. In particular, the performance of ISA, which is most robust against noise, rapidly decreases with increasing overlap degree. From these results, we can see that BiModule has relatively stable performance against noise and can discover overlapping modules with high accuracy.

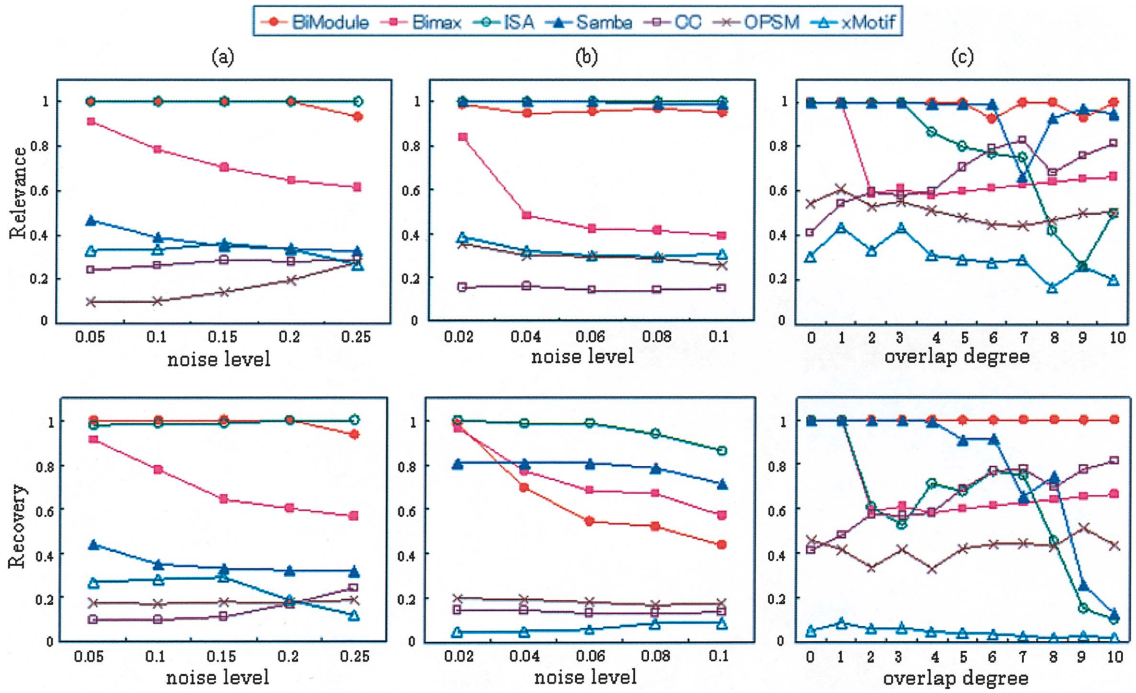

Fig. 5 The relevance (upper) and the recovery (lower) for (a) constant, (b) additive and (c) overlapping biclusters.

4.2.2 Real Data

Following the approach of Prelic, et al., we test the biclustering methods on a real dataset. The real dataset used here is the *S. cerevisiae* dataset containing 2,993 genes and 173 conditions provided by Gasch, et al. [16], which includes gene expression data for several conditions under the 13 different environmental stresses such as heat shock, nitrogen depletion etc. The extracted biclusters are evaluated based on GO annotations, protein-protein interaction networks and metabolic pathway.

Gene Ontology (GO)

Similar to the approach used by Tanay, et al. and Prelic, et al., we investigate whether the set of genes obtained by the biclustering methods shows significant enrichment with respect to a specific Gene Ontology (GO) annotation [17]. We utilize a web tool FuncAssociate [18] to evaluate the discovered biclusters. FuncAssociate computes the hypergeometric functional score of a gene set using Fisher's Exact Test, and then the resulting scores are adjusted for multiple testing based on the Westfall and Young procedure [19]. In this test, we use parameters L=7, Mg=40 and Mc=5. For all of the biclustering methods, we filtered out biclusters overlapped more than 25% with a larger bicluster and output the resultant biclusters up to 100 in descending order of size. The adjusted significant scores (the adjusted p-values) of each discovered bicluster were computed by FuncAssociate. The result of BiModule was compared with those of Bimax, ISA, Samba, CC, OPSM and xMotif obtained from Fig. 3 in Prelic, et al. **Figure 6** is the histogram of proportion of biclusters for one or several over-represent GO categories at different significance levels. In this figure, all biclusters discovered by BiMod-

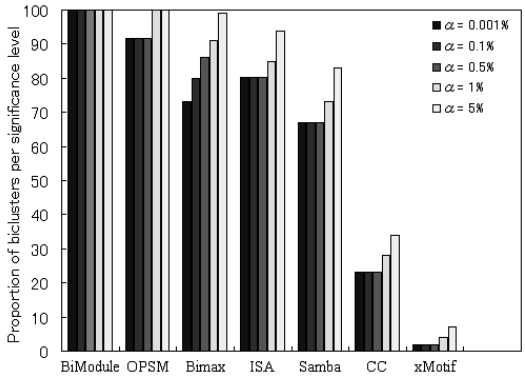

Fig. 6 Proportion of biclusters significantly enriched by any GO biological process category (*S. cerevisiae*). α is the adjusted significant scores of the biclusters.

Table 1 Relevance of biclusters with protein-protein interaction networks.

	BiModule	Bimax	ISA	Samba	CC	OPSM	xModif
Discovered Biclusters	26	100	66	100	100	12	100
Significant Biclusters	24	50	50	30	71	8	0
Proportion (%)	92.3	50	75.8	30	71	33.3	0

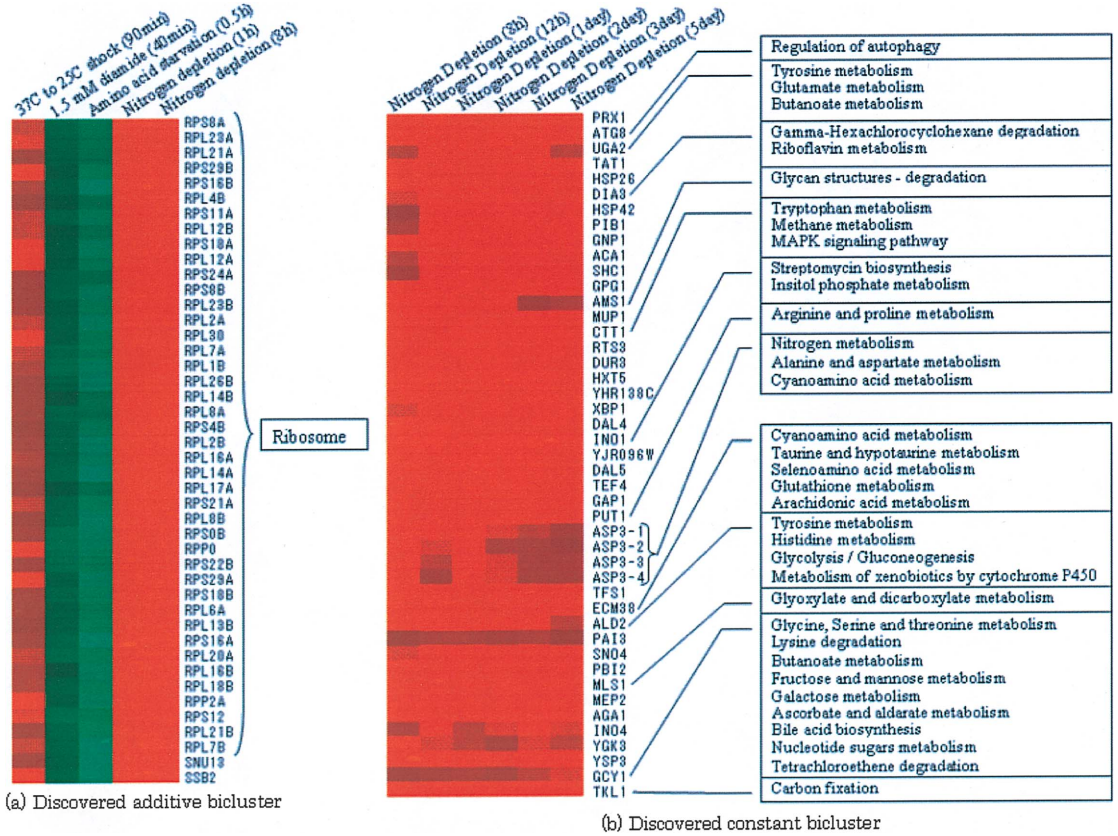

Fig. 7 Biclusters discovered by BiModule are shown. The square(s) to the right of the heat maps show the KEGG metabolic pathways assigned to the indicated genes.

ule contain significantly over-represent GO categories ($\alpha \leq 0.001\%$). This is the best score among the compared methods.

Protein-protein Interaction Network

In addition to the GO enrichment test, we investigate the relationship between the discovered biclusters and protein-protein networks on the *S. cerevisiae* dataset. The protein-protein interaction data was obtained from the DIP database [20]. For each pair of genes, we checked whether the two genes are connected in the protein-protein interaction data. The number of disconnected gene pairs is expected to be significantly smaller for the discovered biclusters than for random gene groups. We generate 1,000 random gene groups of the same size as each bicluster and perform the Z-test to check whether the proportion of disconnected pairs in each bicluster is significantly smaller than the expected values for random gene groups. **Table 1** shows the proportion of biclusters with significantly smaller disconnectedness scores ($\alpha \leq 0.1\%$). In this table, BiModule finds over 90% significant biclusters (24 out of 26) which is the best result among all methods. This suggests that BiModule can work successfully for discovery of potential protein-protein interactions.

Discovered Biclusters and Mapping to Metabolic Pathway

Figure 7 (a) and Fig. 7 (b) are two example of biclusters (one additive and one constant)

discovered by BiModule. We mapped the genes in each bicluster to their KEGG pathways [21]. Almost all of the genes in the additive bicluster (Fig. 7 (a)) produce proteins composing a ribosomal subunit, and the remaining two genes are also related to ribosome synthesis. Figure 7 (b) shows a bicluster for co-expressed genes specific for nitrogen depletion conditions. This bicluster includes some characteristic genes related to nitrogen metabolism. The gene *ASP3* is induced under nitrogen depletion and has 4 copies (*ASP3-1*, *ASP3-2*, *ASP3-3* and *ASP3-4*) in genomic DNA. This bicluster includes all of them. In addition, we can see that it contains genes participating in amino acid metabolism pathways such as "Tyrosine metabolism" and "Arginine and proline metabolism", which are important biological processes in utilizing nitrogen as a nutrient source. *ATG8*, involved in autophagy under nutrient source starvation is assigned to the "Regulation of autophagy". Furthermore, some genes which are not assigned to any pathway also have intriguing biological meaning, such as genes for vacuolar protease inhibitors (*PAI3*, *PBI2*, *TFS1* and *YHR138c*) which have been found to be up-regulated in nitrogen depletion by Unno, et al. [22].

In the same way, we investigated biclusters obtained by the other methods. Samba, OPSM, CC and xMotif discovered only biclusters with many conditions in multiple stresses, that is, these methods generated no biclusters with gene set specific for certain single environmental stress as described above. In contrast, ISA and Bimax generated biclusters for single environmental stress, and biclusters specific for nitrogen depletion stress were contained in both methods. As for Bimax, most genes were not mapped to any pathway and also have no significant GO annotation. The bicluster generated by ISA also had *ASP3* genes as well as genes assigned to amino acid metabolism pathway. Thus, no clear differences for metabolic pathways between BiModule and ISA were observed. As discussed in Prelic, et al. [7] the incompleteness of the metabolic pathway data may be the reason for such unclear result. However, as for GO annotations, two methods show a somewhat clear difference; the bicluster obtained from BiModule contained genes with detailed GO annotations for a nitrogen metabolism such as "Cellular response to nitrogen starvation", in contrast, no genes with such GO annotations were present in the bicluster found by ISA.

4.3 The Accuracy and Running Time of BiModule

We evaluated the influence of the input parameters on the accuracy and the running time of BiModule. We performed the evaluation using synthetic data generated by the following steps: 1) generate a 2000×200 matrix (background) with in all elements set to 0, 2) generate 20 50×10 biclusters which have 1's in all elements, 3) implant the 20 biclusters into the background matrix without overlap, and 4) add the random noise derived from normal distribution $N(0, \sigma)$ to all the elements in the matrix. The noise level σ was set to 0.2. As discussed in section 3.5, BiModule requires the three parameters, L, Mg and Mc. L is relevant to how well the discretized data reflect the structure of real gene expression data. Mg is used as an input data to the LCM algorithm that is the core part of the computation process of BiModule. Mc is used only for filtering out biclusters with too few conditions after enumerating closed itemsets. This parameter has little influence on the performance unless a user specifies a extremely large value. Thus, we focus on the parameters L and Mg that are expected to have critical influence on extraction accuracy of biclusters and running time. **Figures 8** and **9** depict the extraction accuracy (the relevance and recovery) and running time of BiModule, respectively, in the cases of L=3, 5 and 7, versus the value of Mg. As can be seen in these figures, L=7 discovers the implanted modules with the highest accuracy and fastest time over all values of Mg. In contrast, L=3 requires much computational time compared to the other two settings and

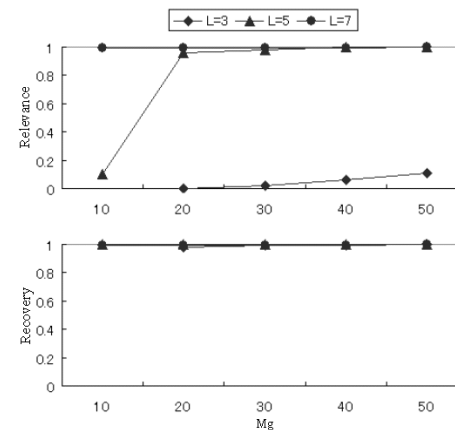

Fig. 8 The relevance (upper) and the recovery (lower) for L=3, 5 and 7.

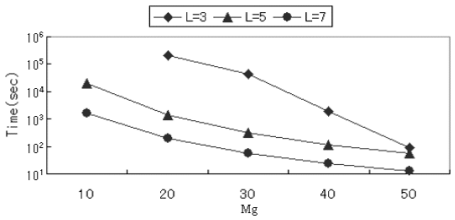

Fig. 9 The running time (logarithmic scale) for L= 3, 5 and 7, is plotted versus the value of Mg.

shows a substantial degradation in relevance. L=5 gives a performance between that of L=3 and 7. Therefore, to obtain good performance in a short time, we recommend setting L to 7. On the other hand, it is generally difficult to define the optimal Mg size. Large values of Mg give fast running time without rapid decrease of extraction accuracy. Therefore, we recommend trying large Mg values first.

5. Conclusions

In this study, we proposed and implemented a new biclustering method, BiModule, for discovering gene expression modules based on a fast closed itemset enumeration algorithm. We performed comparisons with six prominent biclustering methods using both real and synthetic data. For the synthetic dataset, we confirmed that BiModule works successfully for discovering noisy modules and overlapping modules. The robustness against noise seems to be second only to ISA. Moreover, BiModule overwhelmingly outperformed the other methods when extracting overlapping modules. As for the real dataset, BiModule exhibited the most significant enrichment among the methods according to GO annotations and protein-protein interaction data: all the biclusters extracted were functionally enriched and indicated a strong correspondence with the known protein-protein interactions. Some of the discovered biclusters were composed of conditions under a single environmental stress and reflected the metabolic pathways known to be induced by the environmental stress. In addition, we evaluated the influence of the input parameters on the performance. As a result, we determined that L=7 for the number of discretization levels had the best performance in both the extraction accuracy of implanted modules and running time. The running times were less than 1 minute for Mg>30.

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. In order to tackle this problem we are currently developing an itemizing method for interpolating values in a neighborhood of a boundary between two discretization levels.

References

1) Cheng, Y. and Church, G.: Biclustering of expression data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, pp.93–103 (2000).
2) Tanay, A., sharan, R. and Shamir, R.: Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, Vol.18 (Suppl. 1), pp.S136–S144 (2002).
3) Tanay, A., Sharan, R., Kupiec, M. and Shamir, R.: Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc. Natl Acad. Sci. USA*, Vol.101, pp.2981–2986 (2004).
4) Ben-Dor, A., Chor, B., Karp, R. and Yakhini,Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. *Proc. of the 6th Annual Int. Conf. on Computational Biology*, ACM Press, New York, NY, USA, pp.49–57 (2002).
5) Ihmels, J., Bergmann, S. and Brkai, N.: Defining transcription modules using large-scale gene expression data. *Bioinformatics*, Vol.20, No.13, pp.1993–2003 (2004).
6) Murali, T.M. and Kasif, S.: Extracting conserved gene expression motifs from gene expression data. *Pac. Symp. Biocomput.*, Vol.8, pp.77–88 (2003).
7) Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hening, L., Thiele, L. and Zizler, E.: A Systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, Vol.22, No.9, pp.1122–1129 (2006).
8) Pei, J., Han, J. and Mao, R.: CLOSET: An efficient algorithm for mining frequent closed itemsets, *Proc. 2000 ACM-SIGMOD Int. Workshop Data Mining and Knowledge*

Discovery (*DMKD'00*), Dallas, TX, pp.11–20 (2000).
9) Zaki, M.J., Hsiao, C.: An efficient algorithm for mining frequent closed association rule mining, *Proc. 2002 SIAM Data Mining Conf.*, (2002).
10) Uno, T., Asai, T., Uchida, Y. and Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases, *Lecture Notes in Artificial Intelligence*, Vol.3245, pp.16–31 (2004a).
11) Uno, T., Kiyomi, M., and Arimura, H. LCM ver.2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets, *IEEE ICDM'04 Workshop FIMI'04* (2002).
12) LCM ver2: http://research.nii.ac.jp/˜uno/codes-j.html
13) BicAt: http://www.tik.ee.ethz.ch/sop/bicat/
14) EXPANDER: http://www.cs.tau.ac.il/˜rshamir/expander/
15) http://www.tik.ee.ethz.ch/sop/bimax/SupplementMaterials,Biclustering.html
16) Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D. and Brown, P.O.: Genomic expression programs in the response of yeast cells to environmental changes, *Mol. Biol. Cell*. Vol.11, pp.4241–4257 (2000).
17) Gene Ontology Consortium: http://www.geneontology.org/
18) Berriz, G.F., King, O.D., Bryant, B., Sander, C. and Roth, F.P.: Charactering gene sets with FuncAssociate. *Bioinformatics*, Vol.22, No.10, pp.1282–1283 (2003).
19) Westfall, P.H. and Young, S.S.: *Resampling-based multiple testing*, John Wiley & Sons, NewYork, (1993).
20) DIP database: http://dip.doe-mbi.ucla.edu/
21) KEGG: http://www.genome.jp/kegg/
22) Unno, K., Juvvadi, P.R., Nakajima, H., Shirahige, K. and Kitamoto, K.: Identification and characterization of rns4/vps32 mutation in the RNase T1 expression-sensitive strain of Saccharomyces cerevisiae: Evidence for altered ambient response resulting in transportation of the secretory protein to vacuoles, *FEMS Yeast Res.* Vol.5, No.9, pp.801–812 (2005).

(Received December 11, 2006)
(Accepted January 22, 2007)

(Communicated by *Jun Miyazaki*)

Yoshifumi Okada received his Ph.D. degree from Muroran Institute of Technology in 2002. From 2002–2005 he was engaged in a study for Kansei Engineering in Satellite Venture Business Laboratory, Muroran Institute of Technology. Now he is hired as a Research Staff of the Computational Biology Research Center, AIST. Research interests are in microarray data analysis using data-mining techniques.

Wataru Fujibuchi received his Ph.D. degree at the department of biophysics from Kyoto University in 1998. From 1999–2002 he worked as an invited researcher at the NCBI, USA. Now he is hired as a Research Scientist of National Institute of Advanced Industrial Science and has a current position of Visiting Associate Professor, at the Research Institute of IT-Bio, Waseda University. He is an author of Cell Montage database. Research interests: sequence analysis of promoter functions, microarray data analysis, prediction of genetic networks from microarray, integrative analysis of cell features.

Paul Horton received his Ph.D. in computer science from UC Berkeley in 1997, after completing an M.S. degree at Kyoto University. Since 2003 he has been a team leader at the Computational Biology Research Center, AIST. His current research interests are the computational analysis of genomic sequences and other biological data. He serves on the editorial board of the CBI journal and the Mathematical Problem Solving special interest group of IPSJ.