An Association Analysis Approach to Biclustering

Gaurav Pandey, Gowtham Atluri, Michael Steinbach, Chad L. Myers and Vipin Kumar
Department of Computer Science & Engineering, University of Minnesota, Minneapolis, USA
{gaurav,gowtham,steinbac,cmyers,kumar}@cs.umn.edu

ABSTRACT

The discovery of biclusters, which denote groups of items that show coherent values across a subset of all the transactions in a data set, is an important type of analysis performed on real-valued data sets in various domains, such as biology. Several algorithms have been proposed to find different types of biclusters in such data sets. However, these algorithms are unable to search the space of all possible biclusters exhaustively. Pattern mining algorithms in association analysis also essentially produce biclusters as their result, since the patterns consist of items that are supported by a subset of all the transactions. However, a major limitation of the numerous techniques developed in association analysis is that they are only able to analyze data sets with binary and/or categorical variables, and their application to real-valued data sets often involves some lossy transformation such as discretization or binarization of the attributes. In this paper, we propose a novel association analysis framework for exhaustively and efficiently mining "range support" patterns from such a data set. On one hand, this framework reduces the loss of information incurred by the binarization- and discretization-based approaches, and on the other, it enables the exhaustive discovery of coherent biclusters. We compared the performance of our framework with two standard biclustering algorithms through the evaluation of the similarity of the cellular functions of the genes constituting the patterns/biclusters derived by these algorithms from microarray data. These experiments show that the real-valued patterns discovered by our framework are better enriched by small biologically interesting functional classes. Also, through specific examples, we demonstrate the ability of the RAP framework to discover functionally enriched patterns that are not found by the commonly used biclustering algorithm ISA. The source code and data sets used in this paper, as well as the supplementary material, are available at http://www.cs.umn.edu/vk/gaurav/rap.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Data Mining; J.3 [**Computer Applications**]: Life and Medical Sciences

General Terms

Algorithms, Experimentation, Measurement

KDD'09, June 28–July 1, 2009, Paris, France.

Keywords

Real-valued data, biclustering, association analysis, range support, microarray data, functional modules

1. INTRODUCTION

A wide variety of data sets, such as microarray gene expression data, earth science data and stock market measures, are real-valued, and several binary data sets have real-valued versions as well, such as reliability scores attached to edges in protein-protein interaction networks and word frequencies in document data sets. These data sets can be represented as a matrix M, where M_{ij} denotes the value of item j in transaction i. An important type of unsupervised analysis performed on such real-valued data sets in several domains, such as biology, is the discovery of *biclusters*, which are groups of items that show coherent values across a subset of transactions or examples, and thus represent a coherent sub-matrix in M, unlike clustering, where each cluster is discovered using all the transactions. An important example of the utility of biclustering is the discovery of transcription modules from microarray data, which denote groups of genes that show coherent activity only across a subset of all the conditions constituting the data set, and may reveal important information about the regulatory mechanisms operating in a cell [21]. However, since the "coherence" of a bicluster can be defined in several ways, different formulations of the biclustering problem have been proposed in the literature. For instance, Figure 1 illustrates a classification of biclusters proposed by Madeira *et al* [22], particularly in the context of microarray data. They classified different types of biclusters into four categories: (i) constant value biclusters (Figure 1(a)), (ii) constant row (Figure 1(b)) or column biclusters, (iii) biclusters with coherent values, where each row and column is obtained by addition or multiplication of the previous row and column by a constant value (Figure 1(c)), and (iv) biclusters with coherent evolutions, where the direction of change of values is important rather than the coherence of the values (Figure 1(d)). Each of these types of biclusters holds different types of significance for discovering important knowledge from real-valued data sets.

Given the importance of these types of biclusters , a wide variety of algorithms have been developed to find them [22]. Some of the prominent algorithms include ISA [20], SAMBA [37], Cheng and Church's biclustering method [10] (CC), xMotifs [24], CTWC [15], OPSM [5], LCD [29] and co-clustering techniques [13, 32]. Although the principles underlying these algorithms hold for a wide variety of real-valued data, most of them algorithms were developed for or tested using the analysis of microarray data [26], particularly to find transcription modules i.e. the groups of genes that are co-expressed under a subset of conditions. Similar biclusters have been found to be useful for other types of data as well, including genomic data, such as genetic interaction data [29] and

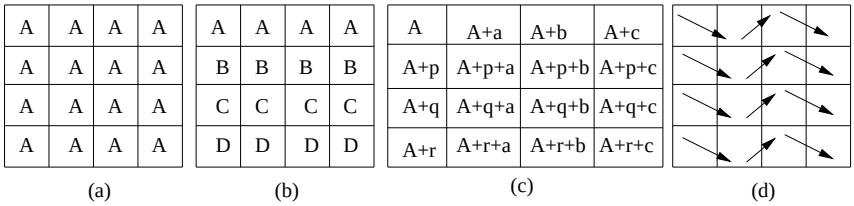


Figure 1: Types of biclusters [22]: (a) Constant value biclusters (b) Constant row biclusters (c) Coherent value biclusters (additive model) (d) Coherent evolution biclusters.

integrated data sets [36]. Interestingly, each of these biclustering algorithms can be viewed from a conceptual perspective according to the classification of biclusters shown in Figure 1. For instance, while SAMBA and co-clustering are designed to find constant value biclusters shown in Figure 1(a), Cheng and Church's method can naturally find both constant value and constant row or column biclusters (Figure 1(b)). Similarly, xMotifs is meant to find biclusters with constant columns in a gene expression data matrix (counterpart of (Figure 1(b)), while OPSM is designed to find coherent trends of up- or down-regulation in biclusters, and thus is suitable for find biclusters like the ones shown in Figure 1(d). However, despite the differences in all these biclustering methods in terms of the type of biclusters they seek, they suffer from some common issues. First, most of the approaches are top-down greedy schemes that start with either all rows and columns, and then iteratively eliminate them to optimize the objective function [13, 32, 10, 15, 29], or they start with a random initial seed and use heuristics to converge to the final bicluster [20, 37, 5]. In either case, the scheme is unable to search the space of all possible biclusters exhaustively. In particular, small patterns tend to get overshadowed by noise and/or by larger biclusters. Another critical issue with at least some of the biclustering methods is with their inability to identify overlapping biclusters. For instance, while ISA, SAMBA and OPSM can find overlapping biclusters, co-clustering (which is designed to only look for disjoint patterns) and Cheng and Church's method (which masks the identified bicluster with random values in each iteration) find it hard to discover such biclusters.

Interestingly, pattern mining algorithms in association analysis [1, 9, 17] also produce biclusters as their result, since the patterns consist of items that are supported by a set of transaction (For this reason, we will use the terms *pattern*, *itemset* and *bicluster* interchangeably in the rest of this paper.). These algorithms enable the exhaustive and efficient discovery of all patterns satisfying the specified thresholds, and these patterns can also overlap with each other. However, traditional association analysis algorithms can only find constant value biclusters (Figure 1(a)) in binary data. Most common efforts to make these techniques usable for real-valued data include discretization [33, 14, 30], binarization [4, 12, 11, 23], and rank-based transformations [8]. However, these data transformation-based approaches face several challenges in addressing the bicluster discovery problem. Most importantly, since all the real values constituting a data set have been transformed to fixed values apriori, these techniques can not distinguish between the different types of biclusters shown in Figure 1, which are defined completely on the basis of these real values, and thus can not ensure the discovery of biclusters of a specific type. Similar challenges are faced by the binarization-based biclustering method BiMax proposed by Prelic *et al* [28]. Some approaches have also been proposed to mine association patterns directly from real-valued data [18, 35, 16]. However, they do not capture some key properties of complex real-valued data sets, such as the distinction between positive and negative values, and the need for values of items in a transaction to be within a range to ensure coherence.

In this paper, we present a novel association pattern discovery framework for data sets where all the attributes are real-valued, i.e. they may take any values in $\mathbb{R}$. This framework is best suited for the type of biclusters shown in Figure 1(b), namely the constant row biclusters, which subsume the constant value biclusters shown in Figure 1(a). The coherence over the rows in these biclusters is ensured using the novel *range support* measure, an integral component of our proposed framework, which ensures that the values of the items constituting a meaningful pattern are coherent for a substantial fraction of transactions in the data set. Since this measure is anti-monotonic, it can be used within an Apriori-like framework [1] to exhaustively discover all the constant row patterns, that satisfy the specified constraints in a given data set (It can also be used to produce constant column biclusters by transposing the original data matrix). Thus, on one hand, this framework reduces the loss of information incurred by discretization- and binarization-based approaches, and on the other, it enables the exhaustive discovery of coherent biclusters, which is currently a limitation of the commonly used biclustering algorithms. We refer to this framework as the RAP (RAnge support Pattern) discovery framework, and the resultant patterns as RAP patterns.

In order to understand the relative effectiveness of our range support-based association analysis methods, we compare RAP's performance with Cheng and Church (2000)'s algorithm [10] (CC) and the Iterative Signature Algorithm (ISA) [20] in the context of microarray data analysis. Comparison with CC is natural, since both RAP and CC find constant row biclusters. ISA, which is one of the most widely used biclustering algorithms, is chosen as a representative of the general class of biclustering algorithms. Thus, comparison against ISA is expected to help indicate the complementarity between biclusters generally found by biclustering techniques and specific constant row patterns found using association analysis. To make this study manageable, other algorithms are excluded from this comparison, either because they are focused on biclusters other than constant row biclusters (eg., OPSM), or are known to have poor performance (eg., xMotifs [28]), or find non-overlapping biclusters (eg., co-clustering). The latter algorithms may not be appropriate for biological data, since genes and proteins are known to be multi-functional [27].

More specifically, in this comparison, we present experiments based on an objective evaluation measure and functional analysis of patterns and biclusters derived from microarray data. Objective evaluation using the mean squared error (MSE) error shows that the patterns derived using our framework indeed capture the constant row model accurately. In functional analysis, we analyze the ability of RAP patterns for extracting co-expression modules (groups of genes) from microarray data that are also functionally coherent, i.e., contain genes that perform the same function in an organism, as indicated by their enrichment by interesting functional classes in

the GO Biological Process ontology [2]. This enrichment is measured as the probability of a group containing the same number of genes as the given pattern having the same or better annotations by a given class by random chance [7], and the lower this probably the more enriched a gene group is with a given functional class. These experiments show that the real-valued patterns discovered by the RAP framework are better enriched by small functional classes, which are considered very interesting for biological analysis [25, 38, 40], than the relatively larger biclusters produced by the CC and ISA algorithms. We also demonstrate the ability of RAP to find novel patterns using specific examples of functionally enriched patterns, as well as functions that are covered by patterns discovered by RAP but not by ISA. These results assert the utility of range support patterns as a potential method for discovering novel coherent biclusters from real-valued data sets in general, and functional modules from microarray data in particular.

The rest of the paper is organized as follows. We provide a brief overview of the CC and ISA algorithms in Section 2. The concepts related to range support patterns are defined and their properties are discussed in Section 3. Section 4 details the experimental methodologies adopted for evaluating the efficacy of range support patterns, and the results obtained. We present a summary of the findings in Section 5, and conclude with the limitations of the proposed framework and possible approaches to address them in Section 6.

2. OVERVIEW OF CC AND ISA

We discuss the basic principles underlying the CC (Cheng and Church's) and ISA algorithm in this section.

2.1 Cheng and Church's algorithm (CC)

Cheng and Church (2000) [10] proposed a greedy heuristic algorithm to find biclusters in a microarray data set that consist of a set of genes having coherent expression values across a set of conditions. They use the *mean squared error* (MSE) measure to capture the coherence of expression levels of a subset of genes across a subset of conditions. If I and J are the set of genes and conditions that define the sub-matrix, and a_{ij} is the expression value of i^{th} gene under j^{th} condition, the MSE score is defined as

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 \qquad (1)$$

where $a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}$ and $a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$ are the means of the values in the i^{th} row and the j^{th} column respectively, while $a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij}$ is the overall mean of the sub-matrix. Note that the minimum value of MSE, i.e. 0, is attained when all the rows and/or columns of the sub-matrix under consideration have constant values, and thus this measure can be used for identifying constant row/column biclusters. Naturally, constant value biclusters are a special case of these biclusters.

However, since the problem of finding all biclusters having the minimum MSE scores in a given data matrix is NP-hard, CC employs a greedy heuristic algorithm for finding bicluster with low MSE scores. This algorithm works in two stages. In the first stage, genes and conditions that provide the maximum reduction in the current MSE score are sequentially removed, and the remaining set of genes and conditions when the MSE score is less than a user-specified threshold δ is reported as a bicluster. The entries in the matrix that correspond to this bicluster are then masked by replacing them with random values to avoid finding duplicate biclusters. This process is iterated several times until desired number of biclusters are found. In the second stage of the algorithm, the genes and conditions deleted earlier are added to each bicluster discovered as long as the MSE score is still within δ. Thus, this row/column addition stage uses the original matrix to add genes or conditions that may belong to another bicluster, to enable the discovery of overlapping biclusters.

Data set				Transaction contributions
i_1	i_2	i_3	i_4	*RS* ($\alpha = 2$)
-0.1	-0.4	1.1	0.8	0
1.6	1.6	1.3	2.7	0
-1.8	-1.9	-1.4	-1.7	1.4
1.4	1.3	0.9	1.5	0.9
-1.5	-1.3	0.3	-1.6	0.0
2.5	2.4	2.1	4.5	0.0
Sum over all transactions				2.3

Table 1: Example table of values for a pattern of four items (columns) $\{i_1, i_2, i_3, i_4\}$ over six transactions (rows), and the contributions of each transaction to *RangeSupport*.

2.2 Iterative Signature Algorithm (ISA)

Ihmels *et al* proposed the Iterative Signature Algorithm (ISA) [20] for finding biclusters that consist of genes that show significant expression individually, and also a high degree of co-expression with each other over a group of conditions. In this algorithm, two versions of the original microarray data matrix E, normalized across conditions (E_C) and genes (E_G) respectively, are maintained. A score for each gene is defined as the average expression (in E_C) over the selected conditions, weighted by the condition score. Analogously, the condition score is defined as the average expression (in E_G) of each selected gene, weighted by the gene score.

The algorithm iterates over two steps. In the first step, a group of genes G^0 is chosen randomly, and a gene score of 1 is assigned to each of them. The condition scores for all the conditions are computed over these genes, and the conditions whose absolute score is greater than a user specified threshold t_c are selected as C^0. In the second step, the gene scores for all genes are computed over these selected conditions and the genes with gene scores greater than a user specified threshold t_g are selected as G^1. These two steps are repeated until the algorithm converges to a group of genes G^n, such that $G^n = G^{n-1}$. Note that the selection of conditions is based on the absolute values of condition scores and the selection of genes is based only on positive gene scores. This ensures that all genes are either significantly positively or negatively expressed for the conditions included in the bicluster.

3. RANGE SUPPORT PATTERNS

In this section, we define a theoretical framework for applying association analysis to real-value data. For this purpose, we introduce the range support measure that capture different semantics of such data, and prove that it is anti-monotonic. Using this anti-monotonicity, we describe an Apriori algorithm-like framework for efficiently extracting range support patterns from data sets. Note that we assume that all the items in the given data set are homogeneous in nature, such as genes in a microarray data set.

3.1 A support measure for real-valued data

Much of the work on the design of efficient algorithms for extracting various types of association patterns from binary data is based on the anti-monotonicity property of various measures, such as the *support* and $confidence$ of an itemset. Of these, the anti-monotonicity of *support* is particularly critical, since it enables the pruning of items that do not have a significant support in the data set, and thus avoids the combinatorial explosion in the number of patterns discovered.

In order to enable association pattern discovery from real-valued

data, we need to define an appropriate anti-monotonic measure, which can be a challenging task. One possible way to formulate a support measure for deriving association patterns from a real-valued data set is as follows. Assuming that a data set only contains positive real values, we define that a transaction supports a pattern if the values of all the items constituting the patterns are within a (user defined) range in the transaction. More formally, given a data set D consisting of a set of transactions T, which contains a value $V_{t,a}$ for each item a in each transaction t, we define the *range support* of a pattern $I = \{i_1, i_2, \ldots, i_k\}$ in this data set as $PositiveRangeSupport(I) = \sum_{t \in T} S(t,I)$, where

$$S(t,I) = \begin{cases} \min\limits_{i \in I} V_{t,i} & \text{if } (\max\limits_{i \in I} V_{t,i} - \min\limits_{i \in I} V_{t,i}) \leq \alpha(\min\limits_{i \in I} |V_{t,i}|)) \\ 0 & otherwise \end{cases} \tag{2}$$

Thus, according to this definition, the contribution of each transaction to an itemset's range support is measured as the minimum of the values taken by any of the items in the itemset in that transaction, if the relative range of these values is within a pre-specified threshold α. Thus, this measure captures the requirement for the values of the items in an itemset to be coherent, or within a range, across several transactions for a pattern to be considered interesting, and can be used to mine interesting patterns from several real-valued data sets, such as document-word $tf-idf$ matrices.

However, in the more general case of real-valued data sets that contain both positive and negative values, such as microarray data, it is important to incorporate the requirement of coherence in the sign or parity of the values also to discover meaningful patterns from such data sets. This requirement can be addressed by enforcing that a transaction can only contribute to the range support of an itemset if the values of all the items in it are of the same sign. This leads us to define the more general support measure *RangeSupport* for real-valued data, that is used in our study.

Formally, given a data set D consisting of a set of transactions T, which contains a value $V_{t,a}$ for each item a in each transaction t, and a range threshold δ, the *RangeSupport* of a real-valued itemset $I = \{i_1, i_2, \ldots, i_k\}$ is defined as $RangeSupport(I) = \sum_{t \in T} RS(t,I)$, where $RS(t,I)$ is defined as

$$RS(t,I) = \begin{cases} \min\limits_{i \in I} |V_{t,i}| & \text{if } [\forall i \in I, V_{t,i} > 0 \text{ or } \forall i \in I, V_{t,i} < 0] \\ & \& \ [(\max\limits_{i \in I} V_{t,i} - \min\limits_{i \in I} V_{t,i}) \leq \alpha(\min\limits_{i \in I} |V_{t,i}|)] \\ 0 & otherwise \end{cases} \tag{3}$$

Thus, *RangeSupport* considers the contribution of a transaction towards the support of an itemset as the minimum absolute value of the constituent items in that transaction, if it satisfies the requirement for *RangeSupport* and all these values are of the same sign. Table 1 demonstrates an example of the computation of *RangeSupport* measure for a simple data set. Also, as Theorem 1 shows, this measure is anti-monotonic.

THEOREM 1. *The RangeSupport measure is anti-monotonic.*

PROOF. See appendix for the proof. The intuition here is that the minimum absolute value of an item in an itemset can only decrease if another item is added to it, if the new itemset still satisfies the requirements for a transaction to have a non-zero contribution to *RangeSupport*. Else the contribution is zero.□

This anti-monotonicity property of the *RangeSupport*, which makes it possible to use it within a standard pattern discovery algorithm, such as Apriori [1], to exhaustively discover meaningful range support patterns, which accurately capture the constant row biclusters (Figure 1) that we are interested in finding, from a given real-valued data set.

In conclusion, the *RangeSupport* measure captures two important characteristics of real-valued data, namely the range and sign of values of the itemsets in a patterns, that are important for the analysis of several such data sets, particularly microarray data, which is the primary data type we have focused on. To the best of our knowledge, none of the current approaches have combined both these factors for defining an anti-monotonic support measure for pattern mining. For instance, Huang *et al*'s measure [18] does not take the range of values explicitly into account, while the generalized support measure proposed by Steinbach *et al* [35] focuses on data sets with only positive values. However, we would like to stress here that for data sets of other types, it may be important to define other variants of the *RangeSupport* measure, but designing those measures is outside the scope of our study.

3.2 Algorithm for finding range support patterns from real-valued data

In the above section, we defined the *RangeSupport* measure for real-valued data, that tries to ensure the coherence and sign of values in a group of items in a pattern, while maintaining the anti-monotonicity property. Due to this property, it is straightforward to employ this measure within an Apriori-like algorithm [1] for finding range support patterns from a data set. In our implementation, we made the pattern search more efficient by representing the set of items and itemsets as a prefix tree [6]. Also, we generated only the closed itemsets [41] as the final output of our pattern discovery algorithm, since they represent a lossless compression of the full set of frequent patterns. We refer this entire framework for mining range support patterns from a real-valued data set as the RAP (RAnge support Pattern) framework.

In summary, this section details a complete framework for efficiently computing coherent patterns from a real-valued data set. The patterns that are eventually extracted are named as RAP patterns. In the next section on experimental evaluation, we describe how RAP can be used for discovering constant row biclusters in microarray data, and examine its efficacy for discovering functionally enriched modules of genes.

4. EXPERIMENTAL RESULTS

In this section, we present the results of our evaluation of the efficacy of our range support pattern mining technique for finding coherent gene groups from microarray data, and compare these results with those obtained from a similar analysis the CC and ISA biclusters. Our evaluation is based on two major methodologies:

- Evaluation using an objective measure of coherence, namely the mean square error (MSE) of the values in a bicluster, as defined by Cheng and Church (2000) [10] and formulated in Equation 1.
- Evaluation of biclusters in terms of functional coherence: Since the result of the different pattern discovery techniques applied to microarray data is groups of genes that co-express with each other strongly, and are expected to perform the same (or similar) functions in an organism, we evaluate the patterns derived in terms of their functional coherence. We selected the Gene Ontology (GO) Biological Process hierarchy [2] as the source for the functions to be studied, and used the principle of the *functional enrichment* of a group of genes by these classes [7]. This generates a $p-value$, which denotes the probability of observing a group of genes of the same size as the one under consideration to be annotated to a certain functional class to the same or greater extent as compared to the original group purely by chance [7, 28, 27]. However, since this probability can be influenced by the size of the functional class, we only considered the classes containing at least 1 and at most 500 genes from *S. cerevisiae* as of 27^{th} January 2009,

Title	Parameter settings*	# biclusters	# selected biclusters	Sizes of patterns (# genes)	# Genes covered	Time taken
RAP patterns						
RAP1	RangeSupport=6,α =0.5	80335	100	2-7	176	2952.56 s
RAP2	RangeSupport=7,α =0.5	36255	100	2-7	176	1068.05 s
RAP3	RangeSupport=8,α =0.5	19281	100	2-7	176	676.03 s
RAP4	RangeSupport=10,α =0.7	28793	100	2-10	185	738.24 s
RAP5	RangeSupport=12,α =1	50359	100	2-10	175	1257.46 s
RAP6	RangeSupport=15,α =1.3	27493	100	2-8	164	415.09 s
CC biclusters						
CC1	$\delta = 0.5$	587	99	21-38	1896	38 hrs
CC2	$\delta = 0.3$	595	99	21-38	1896	38 hrs
ISA biclusters						
ISA1	$t_g = 2, t_c = 2$, |Initial|=100	43	20	40-264	2296	180.07 s
ISA2	$t_g = 2, t_c = 2$, |Initial|=500	165	61	8-264	3192	863.39 s
ISA3	$t_g = 2.5, t_c = 2.5$, |Initial|=100	56	40	10-211	2304	211.3 s
ISA4	$t_g = 2.5, t_c = 2.5$, |Initial|=500	318	100	9-264	3654	1115.66 s
ISA5	$t_g = 3, t_c = 3$, |Initial|=100	6	6	10-50	232	42.22 s
ISA6	$t_g = 3, t_c = 3$, |Initial|=500	47	35	8-140	1108	245.70 s

Table 2: Statistics of biclusters/patterns produced by different algorithms (* Parameters not shown here were set as the default value in BicAT).

when these annotations were downloaded from the GO website (www.geneontology.org). 2652 such classes existed as of this date.

Both these evaluations were carried out on Hughes *et al*'s widely used *S. cerevisiae* (yeast) microarray data set [19]. This dataset has been prepared by treating yeast cells with different chemical compounds and inducing mutations, and is meant to study the functions of yeast genes on a large scale. Its dimensions are 6316 genes$\times$300 conditions, and thus, its large scale nature justifies the use of sophisticated data mining algorithms for extracting useful knowledge about functions of the constituent genes. In our experiments, we applied all the techniques only on the 4684 genes$\times$300 conditions subset of this data set, since the other 1632 genes are not included in any of the functional classes we considered, and will affect the enrichment scores of different algorithms adversely. Also, note that all the values in this data set, which denote the log_{10} of the ratio of the expression of the corresponding gene under the corresponding condition to its expression under a control condition, lie in the range $[-2, 2]$, with a substantial fraction lying close to zero, which denotes the inactivity of the corresponding gene under the corresponding condition.

We used the BicAT tool [3] for implementations of the ISA and CC biclustering algorithms, and our own implementation of the RAP framework for the range support pattern discovery algorithms. Also, since the patterns derived by all the algorithms often have a significant overlap with one or more of the other patterns, which is expected to bias their evaluation, we used Prelic *et al*'s methodology [28], as implemented in BicAT and also used by others [37, 21], for controlling the redundancy between the patterns. This methodology greedily selects up to 100 biggest patterns (size of pattern=|genes|$\times$|conditions| in it) that have an overlap of at most 25% with the current set of selected patterns, starting with the largest pattern output by the algorithm and terminating when all the patterns have been examined, or 100 have been selected. Note that, for this data set, CC generated a selected pattern that included all the genes in the data set, and thus has a poor p-value. To avoid biasing against this pattern, we eliminated this pattern when collecting performance statistics for these algorithms.

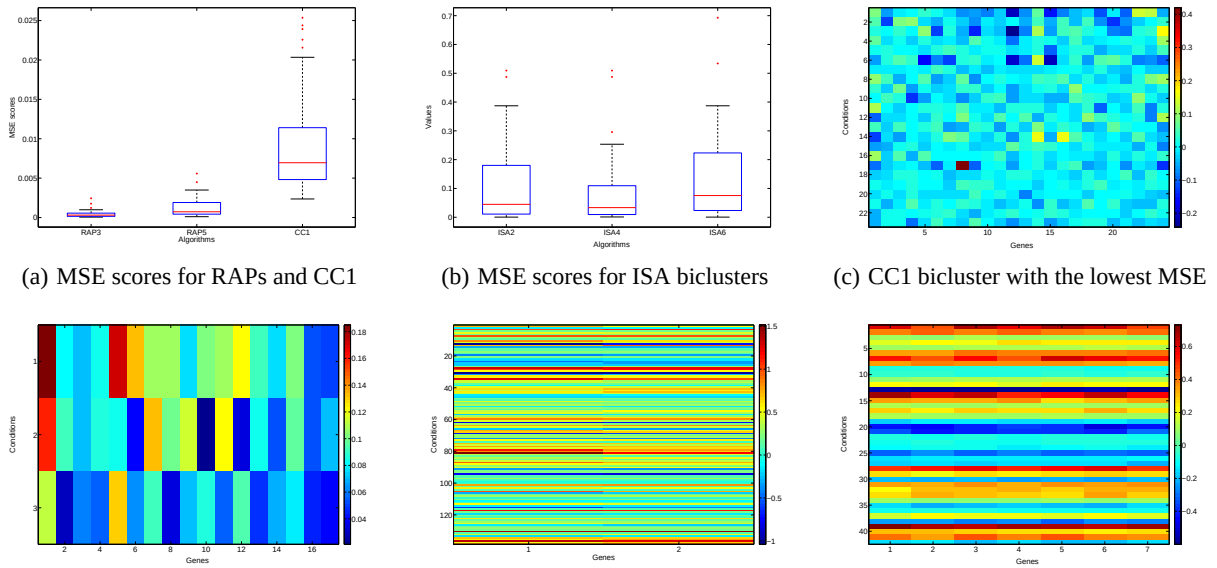
We now discuss the results obtained from our evaluation studies. Note that all the figures shown in this section are best viewed in color, and in a size larger than shown here.

4.1 General statistics about biclusters analyzed

Table 2 details various statistics about the biclusters/patterns discovered using the RAP, CC and ISA algorithm using various parameter settings. The size range and coverage numbers are computed only for the finally selected non-overlapping patterns, since, as mentioned above, only those are used for further analysis. As can be seen, a variety of RAP patterns are produced using different α and RangeSupport thresholds. Since almost all of them produced similar results in experiments presented later in this section, we only present results from RAP3 and RAP5, with the former representing a set of patterns derived using tight range ($\alpha = 0.5$) and low RangeSupport (8) thresholds, while the latter represents patterns from a relatively loose range ($\alpha = 1$) and high RangeSupport (12) configuration. Also, note that CC produced essentially the same set of biclusters at different thresholds, and thus, only one set of biclusters, namely CC1, was used for further analysis. ISA produced a variety of biclusters when its parameters, namely t_g, t_c and the size of the initial random gene set (|Initial|), are varied, as shown by the corresponding rows in Table 2. However, we observed that the results from biclusters derived using |Initial|=500 were generally better that those derived using |Initial|=100, and they also produced comparable number of selected non-overlapping patterns as the other algorithms. Thus, we only used these sets of biclusters, namely ISA2, ISA4 and ISA6, for our comparative evaluation presented in the following subsections.

Several trends can be observed from Table 2. First, it can be seen that the biclusters produced by ISA and CC generally contain larger number of genes than those found by RAP . This is expected, since ISA and CC adopt a top-down approach and achieve the specified thresholds of their objective functions with larger groups of genes and conditions, while RAP searches for patterns that satisfy the specified thresholds exhaustively, starting from single genes, and progressing in a bottom-up fashion. This variation in size has a significant impact on the functional classes that these biclusters represent, as discussed in Section 4.3. Another important observation from Table 2 is that CC and ISA biclusters generally cover many more genes than RAP patterns, which again can be explained on the basis of the exhaustive nature of the RAP algorithm. These differences illustrate the important distinctions between the operation of the traditional biclustering and association analysis algorithms.

Finally, a note about the run time of RAP as compared to other biclustering algorithms, which are provided in the last column of Table 2. For all the parameter settings, some of which produced over $80,000$ patterns, RAP produced the corresponding patterns within an hour, which is comparable to the ISA runs. On the other hand, CC took over a day for computing its biclusters, thus prohibiting its extensive use for large data sets. These numbers show



(a) MSE scores for RAPs and CC1 (b) MSE scores for ISA biclusters (c) CC1 bicluster with the lowest MSE

(d) ISA2 pattern with the lowest MSE (e) RAP3 pattern with highest MSE (f) RAP3 pattern with maximum # genes

Figure 2: MSE distributions for different sets of biclusters, and visualizations of individual biclusters (best seen in color).

that, despite its exhaustive discovery process, RAP is quite efficient due to the anti-monotonicity of the *RangeSupport* measure. However, these results should be treated as preliminary, and we believe that the run times of all the algorithms can be improved by using more efficient data structures and better implementations.

With this understanding of the difference between the nature of the resultant biclusters from the different algorithms, we proceed to their evaluation using objective measures and biological functional enrichment. Note that the results for each set of patterns are referred to by the title assigned to them in Table 2.

4.2 Coherence of Patterns Using MSE

In the first evaluation, we measured the coherence of each bicluster using the MSE score defined in Equation 1, and analyzed the distribution of these scores for all the sets of biclusters discovered by the different algorithms considered. This score and the corresponding distribution is suitable for testing the coherence of the types of patterns we are aiming to find, namely constant row biclusters, since it can be easily verified that the value of this function for such a bicluster having strictly constant rows will be zero. Thus, the closer the distribution of scores for a set of biclusters is to zero, the closer they are expected to capture the constant row model. These distributions for the biclusters produced by the RAP and CC algorithms are shown in Figure 2(a), and another set for the ISA algorithm at various parameter settings is shown in Figure 2(b) (shown separately due to the difference in the scales of the scores).

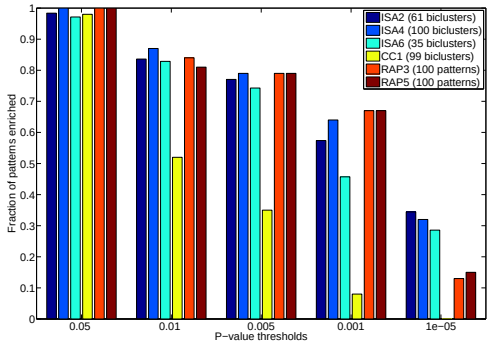
The results in Figure 2(a) show that the scores for the range support patterns in RAP3 and RAP5 are almost all zero, with very few outliers. On the other hand, CC1 patterns have a much wider variability of these scores, which is intriguing, as the CC algorithm attempts to discover biclusters that have the least possible MSE scores. Also, it can be seen from Figure 2(b) that the MSE scores for ISA biclusters are generally quite variable, regardless of the parameters used, with almost all of them having higher MSE scores than both RAP and CC biclusters. However, this is not surprising since the objective function that ISA tries to optimize indirectly is not based on the MSE of the bicluster, but instead the inner product of each included gene's signature row and the module signature of the bicluster. An impact of this is a bias towards higher values in the final bicluster, which can result in a higher MSE even for relatively coherent signatures.

The quantitative results shown in Figures 2(a) and 2(b) can also be qualitatively examined by visualizing individual biclusters produced by these algorithms. For this, we show in Figures 2(c) and 2(d) the sub-matrices of the data set corresponding to the biclusters from CC1 and ISA2 for which the MSE scores (0.0024 and 4.3843×10^{-4} respectively) are the minimum among their corresponding sets of biclusters. It can be seen from these figures that, despite the low MSE scores of these patterns, neither of them show much coherence in their rows, columns or both. In fact, most of the coherence in the CC1 bicluster is contributed by entries having almost neutral (close to zero) values in the range $[-0.05, 0.05]$, which account for 77.9% of all the entries. In contrast, Figure 2(e) shows the sub-matrix of the data set corresponding to the RAP3 pattern that has the highest MSE among all the patterns, and the coherence of the expression of each gene (column) over each condition (row) can be easily observed. This coherence can also be observed for RAP patterns of larger size, such as the pattern of size 7 in RAP3, whose corresponding data sub-matrix is shown in Figure 2(f). Also, observe that because of the wide range of values constituting these patterns, it is difficult to determine binarization or discretization thresholds apriori for applying binary association analysis methods. Thus, using fixed thresholds for these transformations will lead to this pattern or significant subset of it being missed.

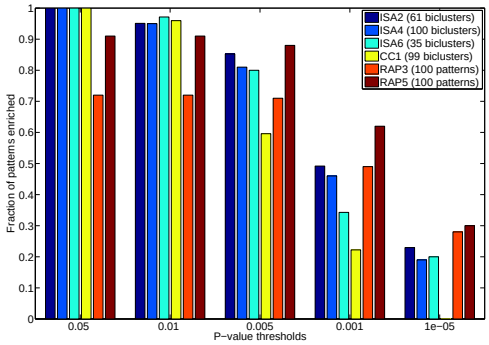
In summary, this quantitative and qualitative evaluation using the MSE score illustrates the ability of RAP patterns to find accurate constant row biclusters.

4.3 Functional Enrichment of Patterns

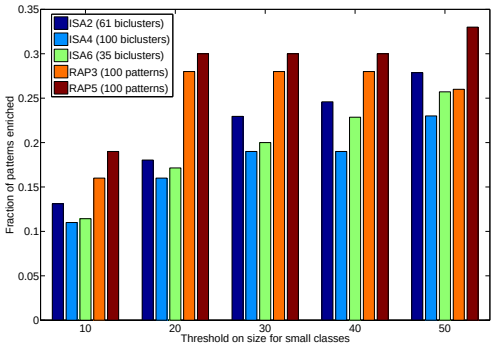
Given the coherence of patterns derived from Hughes *et al*'s microarray data set in terms of their MSE scores, an interesting question to ask is whether the co-expression of the genes constituting a pattern translates into a common function for them. This is a valid hypothesis to be tested for such patterns, since the co-expression of a set of genes over several experimental conditions indicates that they are involved in the same cellular or biological process [27]. An effective and standard way of measuring this *functional coherence*, or *enrichment*, is to compute a $p-value$ for a pattern to be



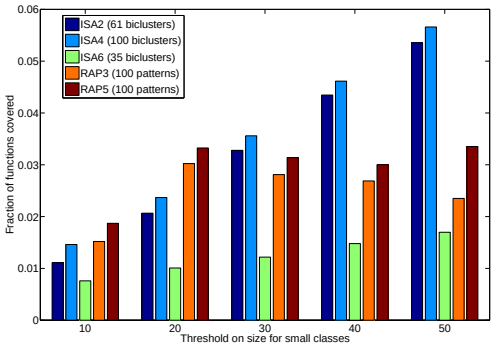
(a) Functional enrichment by large classes (31-500 members)



(b) Functional enrichment by small classes (1-30 members)



(c) Functional enrichment by several groups of small classes



(d) Coverage of small functional classes by biclusters

Figure 3: Statistics about the enrichment of RAP patterns and ISA and CC biclusters with GO BP functional classes.

enriched by a given functional class. This $p-value$ is essentially the probability of observing a group of randomly selected genes of the same size to be co-members of this class to a higher extent than the gene group under consideration (details in [7]). Thus, the lower this $p-value$, the more functionally enriched this gene group is with this class. Now, given a set of functional classes taken from the Biological Process hierarchy of the Gene Ontology [2], a standard methodology for evaluating the quality of a given set of patterns is to determine what fraction of the patterns have a $p-value$ smaller than a specified threshold for at least one of the functional classes in the consideration set [28, 27]. Using this methodology, we illustrate the efficacy of RAP patterns for the task of discovering functionally enriched groups of genes from microarray data, and also compare their performance with biclustering algorithms (CC and ISA), which are the more widely accepted methods for this task. Note that we also performed the analyses described in this section on 100 sets of randomly generated patterns showing the same distribution of sizes and pattern overlap, for each of the sets of biclusters evaluated here, to determine the statistical significance of results obtained. As shown by the results in Figure 1 in the supplementary material, it can be concluded that the results observed at lower $p-value$ thresholds, particularly less than 0.001, were the most statistically significant, and hence the most reliable for this evaluation.

Now, in order to perform a fair analysis, it is important to distinguish between large and small classes in our entire collection of GO BP functional classes, since the size of a class can have an important impact on the $p-values$ computed for the gene group being tested. Using one such definition of small classes as those that have $1-30$ members, and big classes as those that have $31-500$ members, we computed the fraction of selected non-overlapping patterns generated by each biclustering algorithm that were enriched by at least one class in each of these categories at $p-value$ thresholds varying from lenient (0.01) to strict (1×10^{-5}). These results are shown in Figures 3(a) and 3(b) for the large and small classes respectively. It can be seen from Figure 3(a) that biclusters from ISA are better enriched with larger classes, which can be explained by their generally larger sizes, which make then more capable of capturing larger classes accurately. Here, RAP patterns are only able to match the performance of the ISA biclusters at relatively weak $p-value$ thresholds. However, interestingly, most of the biclusters produced by CC do not achieve a significant enrichment with large classes even at moderate $p-value$ thresholds, such as 5×10^{-3}, because of its tendency to find near-zero biclusters, as discussed in the previous subsection. The same analysis on randomly generated patterns also showed the ISA and RAP patterns to be significantly enriched, but not CC patterns.

We also computed the same enrichment statistics for all the sets of biclusters for small functional classes ($1-30$ members). These statistics, shown in Figure 3(b), show that RAP patterns, particularly RAP5, are able to obtain better enrichment than all the ISA and CC biclusters at all the statistically significant $p-value$ thresholds, particularly at the strict threshold of 1×10^{-5}. This result is very interesting, since recent literature in functional analysis of genomic data [25, 38, 40] has suggested that such small and specific classes are often more interesting for further understanding and exploration than larger and more general classes, an underlying design principle for hierarchical functional classification schemes, such as Gene Ontology [2]. Thus, despite their generally small size, RAP patterns are able to cover several small functional classes quite

accurately, which shows the advantages of a bottom-up exhaustive approach for discovering biclusters from genomic data.

The above analysis on enrichment with small classes can be extended to consider other definitions of small classes also, in terms of the maximum number of members in a class for it to be considered small. Figure 3(c) shows the results of the above evaluation at different values of this number in the range of $10-50$, at the strict and most reliable $p-value$ threshold of 1×10^{-5}. CC biclusters are not included in this set of results, since they did not produce any significantly enriched patterns at this threshold. It can be seen that for all these sizes, the patterns included in RAP3 and RAP5 are the most enriched among all the sets of patterns, although the results from RAP3 become comparable to the ISA biclusters at 50 and, beyond this class size, the enrichment statistics are expected to favor biclusters from ISA, due to their larger size, and hence a better opportunity to capture a relatively large class. Also, note that the enrichment statistics can be viewed from a complementary perspective, where instead of the patterns, we compute what fraction of all the functional classes considered are captured by at least one pattern at this strict $p-value$ threshold. These results are shown for the same range of sizes in Figure 3(d). The trends here are similar to the fraction of patterns enriched case, although the advantage of RAP patterns is lost over ISA biclusters much earlier. Still, in summary, range support patterns are quite likely to be useful for exhaustively and efficiently discovering patterns that represent smaller functional classes, and can be used for scientific investigation at a much finer scale than the larger biclusters.

4.4 Complementarity of RAP and ISA

Finally, another benefit of using an approach such as RAP, which adopts a very different pattern discovery algorithm as compared to the more traditional biclustering algorithms such as ISA, is the ability to find finer or completely novel patterns. As an illustration of this, consider the pattern {YAR010C,YBL005W-A,YBR012W-A,YJR026W,YJR028W,YML040W,YMR046C,YMR051C}, which is found among the RAP5 patterns discussed above, but not in ISA4 and ISA6, and is exclusively enriched by the *RNA-mediated transposition* class (GO:0032197) with a low $p-value$ of 4.64×10^{-12}. This pattern is not found among the ISA4 and ISA6 biclusters, and among the ISA2 biclusters, it is embedded within a large bicluster consisting of 138 genes. However, GO:0032197 is ranked eighth in the list of functions this pattern is enriched by. Thus, by adopting a bottom-up algorithm, RAP is able to obtain a pattern of finer granularity, which may otherwise be hidden in a larger biclusters found by other algorithms

Similar observations can be made from the functional coverage viewpoint also. For instance, 48 small classes (having less than 30 members) were over-represented in at least one pattern in the RAP5 set at the $p-value$ threshold of 1×10^{-5}, but none in the ISA2, ISA4 and ISA6 sets. In particular, one pattern {YDR158W,YER052C, YOR130C,YPR145W} was found to be enriched at this threshold for three very small functional classes, namely GO:0009088 (6 members), GO:0009090 (3 members) and GO:0009092 (7 members). However, this pattern was not found in the ISA6 biclusters, and was embedded within large biclusters of sizes 211 and 104 among ISA4 and a bicluster or size 211 in ISA2. Due to their large sizes, these small classes were not found to be significant for these functions. This is another example of RAP patterns being able to capture very small classes that ISA biclusters are not enriched by.

These examples illustrate the complementarity that RAP can provide to standard biclustering algorithm for domain scientists who are trying to find interesting patterns or biclusters, such as groups of functionally related genes, from their real-valued data sets.

5. CONCLUSIONS

In this paper, we presented an efficient framework named RAP (RAnge support Patterns) for directly mining association patterns from real-valued data sets. This algorithm is based on the novel anti-monotonic range-support measure, which ensures that the values of the attributes constituting a meaningful pattern are coherent for a substantial fraction of transactions in the data set. The patterns generated by this algorithm are focused on finding constant row/column biclusters, for which no exhaustive discovery algorithm is currently available. On one hand, this framework reduces the loss of information incurred by discretization- and intervalization-based approaches, and on the other, it enables the exhaustive discovery of coherent biclusters, which is currently a limitation of the commonly used biclustering algorithms.

We compared the efficacy of the range support patterns discovered from microarray data with biclusters produced by Cheng and Church's algorithm for discovering constant row/column biclusters, and ISA, a commonly used biclustering algorithm, using the mean squared error (MSE) coherence measure and their functional enrichment in terms of GO biological process annotations. RAP patterns are found to have the lowest MSE scores among all the sets of biclusters evaluated. In terms of functional enrichment also, the RAP patterns are significantly more enriched by small and specific functional classes as compared to the generally larger in size ISA and CC biclusters, thus indicating their potential for making novel biological discoveries, such as the functions of unannotated genes. We also illustrated that RAP can complement standard biclustering algorithms by finding gene groups not discovered and covering functions not captured by the latter. In summary, these results demonstrate the ability of the RAP framework for discovering previously unavailable knowledge from real-valued data sets.

6. LIMITATIONS AND FUTURE WORK

Although the evidence presented in this paper suggests important utility for range support patterns discovered using the RAP algorithm, there are some limitations that can be addressed using other ideas to enhance this promise significantly. We discuss some of these limitations and possible ideas to address them below:

• **Size of patterns**: Due to the hard range support thresholds imposed on RAP, some larger patterns that do not satisfy these thresholds might be split into smaller ones, thus placing a limit on the size of the patterns produced. Concepts such as colossal patterns [42] for merging the core patterns may be useful for this problem.

• **Coverage of items**: An adverse effect of performing an exhaustive search for patterns is the inability to explore the entire itemset lattice in an acceptable amount of time. This limitation can be addressed using ideas such as length-varying support [31] and support envelopes [34], as well as simple heuristics for including currently uncovered items.

• **Enhancing scalability**: An orthogonal direction for addressing the above issues is to design more efficient data structures to enhance the efficiency of the pattern search process. Ideas from the TAPER algorithm [39] can be explored for improving the efficiency of the computation of item-pairs, which often turns out to be the bottleneck in this process.

In addition to addressing these challenges, an important direction for further research will be to extend the RAP framework to capture constant additive/multiplicative biclusters (Figure 1(c)) and biclusters with constant evolutions (Figure 1(d)). It will also be interesting to examine the impact of pre-processing operations, such as normalization and sparsification or denoising, as well as on the quality of the patterns obtained. In the latter operation, one particularly useful study will be a comparison of the relative strengths

and weaknesses of RAP and pattern discovery from binarized versions of a real-valued data set, and the development of a hybrid of both these approaches. Finally, an interesting extension of this work will be adapting the RAP framework to data sets where the items are heterogeneous in nature.

7. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. VLDB*, pages 487–499, 1994.

[2] M. Ashburner et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[3] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler. BicAT: a biclustering analysis toolbox. *Bioinformatics*, 22(10):1282–1283, 2006.

[4] C. Becquet et al. Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human sage data. *Genome Biology*, 3(12):1–16, 2002.

[5] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering Local Structure in Gene Expression Data: The Order-Preserving Submatrix Problem. *Journal of Computational Biology*, 10(3-4):373–384, 2003.

[6] C. Borgelt. Efficient Implementations of Apriori and Eclat. In *Proc. FIMI*, 2003.

[7] E. I. Boyle et al. Go::termfinder—open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715, 2004.

[8] T. Calders, B. Goethals, and S. Jaroszewicz. Mining rank-correlated sets of numerical attributes. In *Proc. KDD*, pages 96–105, 2006.

[9] A. Ceglar and J. F. Roddick. Association mining. *ACM Comput. Surv.*, 38(2):5, 2006.

[10] Y. Cheng and G. Church. Biclustering of Expression Data. In *Proc. Eighth ISMB Conference*, pages 93–103. AAAI Press, 2000.

[11] G. Cong et al. Mining frequent closed patterns in microarray data. In *Proc. ICDM*, pages 363–366, 2004.

[12] C. Creighton and S. Hanash. Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86, January 2003.

[13] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. In *Proc. ACM SIGKDD*, pages 89–98, 2003.

[14] T. Fukuda et al. Mining optimized association rules for numeric attributes. In *Proc. PODS*, pages 182–191, 1996.

[15] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *PNAS*, 97(22):12079, 2000.

[16] E.-H. Han, G. Karypis, and V. Kumar. Min-apriori: An algorithm for finding association rules in data with continuous attributes. Technical Report 97-068, Dept. of Comp. Sc. and Engg., Univ. of Minnesota.

[17] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *DMKD*, 15:55–86, 2007.

[18] Y. Huang, H. Xiong, W. Wu, and S. Y. Sung. Mining quantitative maximal hyperclique patterns: A summary of results. In *Proc. PAKDD*, pages 552–556, 2006.

[19] T. R. Hughes et al. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.

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

[21] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nat. Genet.*, 31:370–377, 2002.

[22] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM TCBB*, 1(1):24–45, 2004.

[23] T. McIntosh and S. Chawla. High confidence rule mining for microarray analysis. *IEEE/ACM TCBB*, 4(4):611–623, 2007.

[24] T. M. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. In *Proc. Pac Symp Biocomput.*, pages 77–88, 2003.

[25] C. L. Myers et al. Finding function: evaluation methods for functional genomic data. *BMC Genomics*, 7:187, 2006.

[26] D. V. Nguyen, A. B. Arpat, N. Wang, and R. J. Carroll. DNA microarray experiments: biological and technological aspects. *Biometrics*, 58(4):701–717, 2002.

[27] G. Pandey, V. Kumar, and M. Steinbach. Computational approaches for protein function prediction: A survey. Technical Report 06-028, Dept. of Comp. Sc. and Engg., Univ. of Minnesota, 2006.

[28] A. Prelic et al. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.

[29] S. Pu, K. Ronen, J. Vlasblom, J. Greenblatt, and S. J. Wodak. Local coherence in genetic interaction patterns reveals prevalent functional versatility. *Bioinformatics*, 24(20):2376–2383, 2008.

[30] R. Rastogi and K. Shim. Mining optimized association rules with categorical and numeric attributes. *IEEE TKDE*, 14(1):29–50, 2002.

[31] M. Seno and G. Karypis. Finding frequent patterns using length-decreasing support constraints. *Data Min. Knowl. Discov.*, 10(3):197–228, 2005.

[32] H. Shan and A. Banerjee. Bayesian co-clustering. In *Proc. IEEE ICDM*, pages 530–539, 2008.

[33] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *Proc. SIGMOD*, pages 1–12, 1996.

[34] M. Steinbach, P.-N. Tan, and V. Kumar. Support envelopes: A technique for exploring the structure of association patterns. In *Proc. ACM SIGKDD*, pages 296–305, 2004.

[35] M. Steinbach, P.-N. Tan, H. Xiong, and V. Kumar. Generalizing the notion of support. In *Proc. SIGKDD*, pages 689–694, 2004.

[36] A. Tanay et al. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *PNAS*, 101(9):2981–2986, 2004.

[37] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(90001):S136–S144, 2002.

[38] Y. Tao, L. Sam, J. Li, C. Friedman, and Y. A. Lussier. Information theory applied to the sparse gene ontology annotation network to predict novel gene function. *Bioinformatics*, 23(13):i529–i538, 2007.

[39] H. Xiong, S. Shekhar, P.-N. Tan, and V. Kumar. Taper: A two-step approach for all-strong-pairs correlation query in large databases. *IEEE Trans. on Knowl. and Data Eng.*, 18(4):493–508, 2006.

[40] H. Yu, L. Gao, K. Tu, and Z. Guo. Broadly predicting specific gene functions with expression similarity and taxonomy similarity. *Gene*, 352:75 – 81, 2005.

[41] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proc. SDM*, 2002.

[42] F. Zhu, X. Yan, J. Han, P. Yu, and H. Cheng. Mining colossal frequent patterns by core pattern fusion. In *Proc. IEEE ICDE*, pages 706–715, 2007.

Acknowledgement

This work was supported by NSF grant CNS-0551551 and a UMR BICB seed grant. Computing facilities were provided by MSI.

APPENDIX

Proof of Theorem 1 (Using the same notation as Section 3.1): Let I be an itemset, and I' be another itemset, where $I' = I \bigcup x$, where x is an item that is not included in I. Then, for a transaction $t \in T$, the computation of $RS(t,I')$ can be broken down into the following cases:

• The items in I' have different signs in t: In this case, $RS(t,I') = 0$. Also, $RS(t,I) \geq 0$. Thus, $RS(t,I') \leq RS(t,I)$.

• The items in I' have the same sign in t: Two sub-cases may occur here: If $(\max_{i\in I} V_{t,i} - \min_{i\in I} V_{t,i}) > \alpha(\min_{i\in I} |V_{t,i}|)$, then $RS(t,I') = 0$, and as in the previous case, $RS(t,I) \geq RS(t,I')$. Otherwise, $RS(t,I') = \min_{i\in I'} |V_{t,i}| \leq \min_{i\in I} |V_{t,i}| = RS(t,I)$. Combining the above, we get $RS(t,I') \leq RS(t,I)$.

Thus, $\forall t \in T$, $RS(t,I') \leq RS(t,I)$, and thus,

$$\begin{aligned} RangeSupport(I') &= \sum_{t\in T} RS(t,I') \\ &\leq \sum_{t\in T} RS(t,I) = RangeSupport(I) \end{aligned}$$

This proves that *RangeSupport* is anti-monotonic.□