High-Confidence Rule Mining for Microarray Analysis

Tara McIntosh and Sanjay Chawla

Abstract—We present an association rule mining method for mining high-confidence rules, which describe interesting gene relationships from microarray data sets. Microarray data sets typically contain an order of magnitude more genes than experiments, rendering many data mining methods impractical as they are optimized for sparse data sets. A new family of row-enumeration rule mining algorithms has emerged to facilitate mining in dense data sets. These algorithms rely on pruning infrequent relationships to reduce the search space by using the support measure. This major shortcoming results in the pruning of many potentially interesting rules with low support but high confidence. We propose a new row-enumeration rule mining method, MAXCONF, to mine high-confidence rules from microarray data. MAXCONF is a support-free algorithm that directly uses the confidence measure to effectively prune the search space. Experiments on three microarray data sets show that MAXCONF outperforms support-based rule mining with respect to scalability and rule extraction. Furthermore, detailed biological analyses demonstrate the effectiveness of our approach—the rules discovered by MAXCONF are substantially more interesting and meaningful compared with support-based methods.

Index Terms—Data mining, association rules, high-confidence rule mining, microarray analysis.

1 INTRODUCTION

The increasing volume of biological data collected in recent years has prompted considerable interest in developing efficient bioinformatics tools for genomic and proteomic data analysis. One main objective of molecular biology is to develop a deeper understanding of how genes are functionally related and, more specifically, to explain how cells control and regulate the expression of their genes and other cellular functions. Deciphering gene relationships has the potential to assist biomedical research in identifying the underlying cause of a disease and developing specific gene-targeting treatments.

Microarrays have revolutionized the way in which biological research is carried out. They allow biologists to analyze the behavior of an organism’s genome globally by measuring the expression levels of thousands of genes within a cell in a single experiment. Despite these global genome studies, research in gene relationships is hindered by the large volumes of data produced by microarray experiments. Microarray data presents new challenges that render many traditional data mining techniques infeasible for extracting and exploring the hidden gene relationships. The main challenge is its high density—a large number of attributes (columns) and a considerably smaller number of expression experiments (rows).

To use current data mining algorithms, biologists are forced to simplify the complexity of their data by restricting the analysis to a small proportion of attributes. For example, Boolean Networks [1], [2] and support-based Association Rule (AR) mining [3], [4] often restrict the search space to as few as 5 percent of the entire genes studied. As a result, many potentially interesting gene relationships (low support and high confidence) are not retrieved.

AR mining is a foundational technique that allows for the simultaneous discovery of relationships between attributes. AR algorithms can extract associations among genes from microarray data sets, where the expression of one gene is related to the expression of others. For example,

\[ \text{GENE1} \Rightarrow \text{GENE2} \text{ (support 10\%, confidence 90\%)} \]

states that, when GENE1 is expressed, 90 percent of the time GENE2 is also expressed and that GENE1 and GENE2 are expressed together in 10 percent of the microarray experiments.

In comparison with Boolean Networks, where a small group of genes are selected prior to data analysis [5], traditional AR mining algorithms can include all genes, allowing for the global analysis of microarrays. The number of genes is then iteratively reduced by pruning sets of genes that are considered uncommon/infrequent. As a result, AR algorithms search for common gene relationships within experiments. These AR algorithms, however, were developed for sparse data sets, where there are few columns and many more rows and, thus, are not appropriate for microarray data. They work by enumerating the relationships among columns (genes) and thus must consider an enormous number of gene associations. This often results in itemset explosion, where the number of associations that must be considered exceeds the available memory space. Recently, support-based rowenumeration AR mining algorithms have been introduced to prevent itemset explosion, allowing the mining of dense data sets [4], [6].
In this paper, we will show that mining common relationships between attributes using support-based pruning is not suitable for all types of microarray experiments. Motivated by this concern, we developed a new row-enumeration algorithm, MAXCONF, which successfully mines ARs without support pruning. We incorporate new confidence pruning methods allowing us to reduce the row-enumeration space and, in turn, mine not only common relationships but rare interesting relationships as well.

We compare MAXCONF with the recently introduced support-based row-enumeration algorithm RERII [4]. Our evaluation on three microarray data sets demonstrates how MAXCONF outperforms RERII with respect to efficiency and the number of rules identified. To investigate the biological relevance of the gene relationships MAXCONF identifies, we evaluated them using the BIND [7] and the Gene Ontology [8]. Our experimental results indicate that MAXCONF is much more effective in discovering gene relationships from microarrays than support-based approaches.

This paper is organized as follows: In Section 2, we introduce microarrays and their characteristics that make analysis difficult. In Section 3, we present the relevant work in the literature, which motivated the development of our MAXCONF algorithm, described in Section 4. The experimental results of our evaluation are outlined in Section 5. In Section 6, we conclude this paper with a summary.

2 MICROARRAYS

The DNA microarray allows parallel genome-wide gene expression measurements of thousands of genes at a given time, under a given set of conditions, for a cell/tissue of interest. The presence of a gene’s mRNA transcript in a cell indicates that the gene is expressed and there is a strong correlation between the degree of a gene’s expression and the amount of mRNA. Furthermore, the expression level of a single gene is highly dependent on the presence and/or absence of various proteins and, thus, the expression levels of the genes encoding those proteins.

The generation of microarray data introduces a variety of data analysis issues not encountered in traditional molecular biology or medicine. The data from a series of microarray experiments is commonly in the form of an $N \times M$ matrix of expression levels, where the $N$ rows correspond to the various experimental conditions (generally < 500) and the $M$ columns correspond to the genes studies (generally $\geq$ 6,000). This data form renders many data analysis issues not encountered in traditional data mining algorithms ineffective as these algorithms are designed to mine sparse data, where the number of nonzero columns is a small fraction of the number of rows. This aspect will be further detailed in Section 3.

There are three main designs of microarray experiments: temporal, duplicate, and perturbation. In temporal experiments, each row corresponds to a different time point to monitor the expression changes of the genes over time. For example, the Spellman et al. [9] data set measures the changes in expression of S. cerevisiae genes during the cell cycle. Duplicate experiments are often used to identify common characteristics within a population for classification purposes. For example, the prostate cancer data set [10] contains the expression values of 12,600 genes from 52 cancerous and 50 healthy prostate cells.

In this paper, we concentrate on analyzing perturbation microarrays as they are specifically designed to understand the relationships between genes. Perturbation experiments are based on the rationale that, if a gene or cell is no longer able to function normally, the expression levels of other genes that are functionally related may be altered. In perturbation data, each column corresponds to a cell that may be genetically altered to prevent the expression of a selected gene or stress induced [11] to infer its affect.

Perturbation microarrays exhibit data characteristics not observed in duplicate microarray experiments. When analyzing duplicate experiments, identifying common gene relationships across the majority of experiments is appropriate and, thus, clustering [12] and support-based row-enumeration AR mining are suitable. Perturbation data, on the other hand, will contain not only common relationships but also rare relationships describing the effects of the perturbations. Therefore, methods designed to analyze duplicate microarrays are not particularly effective for analyzing perturbation data. The main related literature on AR mining to date has focused on improving support-based classification tasks and, thus, the mining of duplicate data [4], [6], [13]. To our knowledge, there is no reported work in the literature on algorithms that can mine perturbation data effectively.

3 RELATED WORK

3.1 AR Mining

AR mining was originally designed to examine the behavior of customers in terms of the products (items) they purchase together in one visit (transaction) [14]. ARs from this data provide valuable information that can be used for marketing and product placement. A formal statement of the AR mining problem is as follows: Let the data set $D = \{I_1, I_2, \ldots, I_n\}$ be a set of $n$ transactions and let $I = \{i_1, i_2, \ldots, i_m\}$ be the set of all possible items ($m$). Each transaction $I_t$ consists of a set of items $I$ from $I$. The aim is to mine all ARs (implications) of the form $I_1 \Rightarrow I_2$ which describe strong relationships between the items based on the transactions in $D$. In the previous AR, $I_1$ is referred to as the antecedent itemset and $I_2$ as the consequent itemset. The strength of an AR is predominately measured by support and confidence and the goal is to identify rules that have a support and confidence greater than the user-specified thresholds minimum support (minsup) and minimum confidence (minconf), respectively. For brevity, we refer to an itemset with $k$ different items as a $k$-itemset.

Definition 1 (Support). Let $I \subseteq I$ be a set of items from $D$. The support of an itemset $I$ in $D$, denoted by $\sigma(I)$, is the proportion of transactions that contain $I$:

$$\sigma(I) = \frac{\# \text{ of transactions containing } I}{\# \text{ of transactions}}.$$ (1)
The support of an AR \( I_1 \rightarrow I_2 \) is \( \sigma(I_1 \cup I_2) \). If \( \sigma(I) \geq \minsup \), then \( I \) is a frequent itemset.

**Definition 2 (Confidence).** The confidence of an AR \( I_1 \rightarrow I_2 \), denoted by \( \text{conf}(I_1 \rightarrow I_2) \), refers to the strength of the association and is given by

\[
\frac{\sigma(I_1 \cup I_2)}{\sigma(I_1)}.
\]

For example, the support and confidence of the rule \( A \rightarrow CD \) in Table 1a are 3 and 3/5, respectively.

The first stage of standard AR mining algorithms like Apriori [14] is to identify all frequent itemsets. Following this, the confidence of all rules that can be formed from the frequent itemsets is calculated and the confident rules are retained. This final phase is not computationally expensive; hence, the majority of research has been devoted to the first.

The main concern during the first phase is that the search space for frequent itemsets is exponential with respect to the number of different single items within a data set. We refer to any itemset that is generated and whose support is counted during this process as a *candidate itemset*. To naively identify all frequent itemsets, all possible candidate itemsets must be tested. This is not necessary however. The *support monotonicity* property states that if an itemset is infrequent, then all of its supersets will also be infrequent [14]. Based on this, if an infrequent itemset is found, we can reduce the search space of candidates by not considering any of its supersets.

**Definition 3 (Support monotonicity [14]).** Given a transaction data set with items \( \mathcal{I} \), let \( I_1 \) and \( I_2 \) be two itemsets such that \( I_1 \subseteq I_2 \subseteq \mathcal{I} \), then

\[
I_1 \subseteq I_2 \Rightarrow \sigma(I_1) \geq \sigma(I_2).
\]

The Apriori algorithm [14] employs this property, systematically generating and counting the support of all candidate itemsets in a bottom-up procedure. That is, Apriori begins with all frequent itemsets of size 1 (1-itemsets) and systematically extends these to 2-itemsets by merging them with other frequent 1-itemsets. For example, if \( \minsup = 3 \), the frequent 1-itemsets in the transactions in Table 1a are \( A, C, D, \) and \( E \). Each of these frequent 1-itemsets is then combined with another to form candidate 2-itemsets. These include \( AC, AD, AE, CD, CE, \) and \( DE \). The 2-itemsets that are infrequent are pruned, in this case, \( AE \), and the remainder are iteratively extended to form larger itemsets until no new candidate itemsets can be formed. This process is referred to as *item enumeration*.

The Apriori algorithm is generally effective for mining sparse data sets. As data density increases, \( \minsup \) will need to be increased and less interesting rules will be mined. This is because Apriori works well on the assumption that the number of frequent itemsets is low and, thus, the number of candidate itemsets will also be low. Microarray data is considered dense, however, where there are many more items than transactions and there are many large candidate and frequent itemsets. As a result, Apriori suffers from *itemset explosion*, which occurs when the space required to store the candidate itemsets exceeds the space available. For example, to identify a frequent 5-itemset, at least 30 smaller candidate itemsets (including 1, 2, 3, and 4-itemsets) will need to be generated.

When applying AR mining to microarray data, each microarray experiment is considered to be a single transaction. In our experiments, genes that are considered to have an up-regulated or down-regulated expression level in at least one transaction form the items. This is detailed in Section 5.

**TABLE 1**
Example Transaction Data Set and Rules (\( \minsup \geq 3 \) and \( \minconf \geq 4/5 \)):
(a) Transaction Set, (b) Rules Found by MAXCONF and RERII

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
<th>MAXCONF Rule</th>
<th>Confidence</th>
<th>Support</th>
<th>RERII Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A B C D E G</td>
<td>C ( \Rightarrow ) D E G</td>
<td>4/6</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>A C D E G</td>
<td>E ( \Rightarrow ) CD G</td>
<td>4/5</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>C D E F G H I</td>
<td>G ( \Rightarrow ) C D E</td>
<td>4/4</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>B C D E G</td>
<td>A ( \Rightarrow ) C G</td>
<td>4/5</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>A C E G I</td>
<td>C ( \Rightarrow ) A G</td>
<td>4/6</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>A D I</td>
<td>G ( \Rightarrow ) A C</td>
<td>4/6</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>D I J</td>
<td>A ( \Rightarrow ) D</td>
<td>4/5</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>A B C D G</td>
<td>B ( \Rightarrow ) C D E G</td>
<td>2/3</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>1</td>
<td>A D I</td>
<td>B ( \Rightarrow ) C D G</td>
<td>3/3</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>D I J</td>
<td>I ( \Rightarrow ) D</td>
<td>3/4</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>D I J</td>
<td>J ( \Rightarrow ) D</td>
<td>1/1</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>D I J</td>
<td>F ( \Rightarrow ) C D E G H I</td>
<td>1/1</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>D I J</td>
<td>H ( \Rightarrow ) C D E G F I</td>
<td>1/1</td>
<td>1</td>
<td>No</td>
</tr>
</tbody>
</table>
3.2 Row Enumeration

Recently, support-based row-enumeration methods have emerged to facilitate the mining of microarray data. These include FARMER [15], TOPKRGS [13], CARPENTER [6], and RERII [4]. These algorithms effectively prevent itemset explosion by only expanding closed itemsets and enumerating the rows (transactions) rather than the items.

**Definition 4 (Closed itemset).** The candidate itemset $I_1$ is a closed itemset if there does not exist an itemset $I_2$ such that $I_1 \subseteq I_2$ and $\sigma(I_1) = \sigma(I_2)$.

FARMER and TOPKRGS were specifically designed to generate classification ARs of the form $X \Rightarrow C$, where $C$ is a class label [13], [15]. These two algorithms require duplicate microarray data, where each microarray experiment is classified into one of two classes prior to mining.

The algorithm CARPENTER and its extension RERII are designed to mine frequent closed itemsets (FCI) from microarray data that may not be classified, that is, these algorithms simply do not consider any classes [4], [6]. Our MAXCONF algorithm is closely related to RERII in that they are both row-enumeration algorithms, specifically designed to mine unclassified microarray data. Therefore, for the remainder of this section, we will concentrate on introducing RERII to provide a strong foundation and motivation for our algorithm.

RERII extracts all FCI by searching the row-enumeration space depth first. It begins by removing all infrequent 1-itemsets from the transactions. These transactions are then considered as individual itemsets, each assigned a support of 1. These itemsets are then intersected with one another, iteratively generating subitemsets of greater support. This continues recursively until no smaller itemsets can be formed [4].

The search space (without support pruning) for the transactions in Table 1a is represented as a row-enumeration tree in Fig. 1a. Here, each node $n$ corresponds to an itemset whose child nodes $c(n)$ correspond to subitemsets with greater support. We use the phrase sibling nodes of $n$, denoted by $s(n)$, to refer to the nodes to its right with the same parent.

Child nodes are generated by taking the intersection of the parent itemset with one or more of its sibling nodes. For example, in Fig. 1a, the node \{12\} (indicated by the edge label) corresponds to the intersection between nodes \{1\} and \{2\} and its support is simply the number of transactions that were intersected during formation; in this case, it is two. There are two situations when an intersection does not form a child node:
MCINTOSH AND CHAWLA: HIGH-CONFIDENCE RULE MINING FOR MICROARRAY ANALYSIS

1. If the intersection is a 1-itemset, the child node is not formed as this simply cannot form an AR. This occurs between nodes \{1\} and \{7\}.

2. If the current parent node \(n\) is completely contained within a sibling node, a child node is not constructed and the support of \(n\) and all \(c(n)\) is incremented by one.

After all child nodes of a node are generated, the algorithm continues recursively depth first by forming the next set of child nodes.

Itemset support pruning is included to reduce any unnecessary node expansion. With respect to the support monotonicity property, row-enumeration algorithms can only apply this pruning during initialization, where the infrequent 1-itemsets are removed from the initial transaction nodes. The main support pruning that RERII employs is based on predicting the maximum support a node \(n\) may exhibit.

**Definition 5 (Maximum support [4]).** Given a node \(n\) with \(k\) sibling nodes, the maximum support of the itemset at \(n\), represented as \(\sigma_{\text{max}}(n)\), or any of \(n\)’s potential child nodes is

\[
\sigma_{\text{max}}(n) = n\.\text{initial support} + k.
\]

The maximum support increase of \(n\) is the cardinality of \(s(n)\) [4]. More specifically, a node’s support is only increased if it is completely contained within at least one of its sibling nodes. Furthermore, the maximum support increase of all \(c(n)\) is also the cardinality of \(s(n)\) [4]. For example, the node \{16\} with initial support of two will only be intersected with one node (\(|s(\{16\})| = 1\)). Thus, the support of \{16\} and all \(c(\{16\})\) will be at most three. If a node’s maximum support is less than \(\text{minsup}\), the node can be pruned.

Cong et al. [4] applied RERII to microarray data; however, their analysis only involved performance studies with respect to time and space requirements compared with state-of-the-art Apriori style methods CHARM [16] and CLOSET [17]. As \(\text{minsup}\) was decreased, CHARM failed due to using all available memory and CLOSET was found to be too slow. RERII, on the other hand, performed superiorly to both [4], without memory issues, indicating the appropriateness of applying row-enumeration over item-enumeration methods.

Unlike Boolean networks and item-enumeration AR algorithms, row-enumeration algorithms can identify more gene relationships by including many more genes in the mining process. However, there is a fundamental issue related to the limitation of support-based pruning that these algorithms do not address—many rules that a biologist would consider of high interest are pruned (based on support), leaving them undiscovered. This is particularly the case with perturbation microarrays.

**3.3 Maximum Participation Index**

The support-based techniques deem infrequent itemsets uninteresting, resulting in them being pruned during frequent itemset generation. Therefore, in the final phase of rule mining, only a subset of the confident rules may be identified.

The Maximal Participation Index (MAXPI) was introduced in [18] to mine collocation patterns from spatial data sets. It excludes the support threshold from the search, allowing all confident rules to be identified. The MAXPI of an itemset \(I\) is the maximum confidence of all generated rules from \(I\). Therefore, if the MAXPI of an itemset is below the confidence threshold, it cannot generate any confident rules. Unlike support, MAXPI is not monotonic with respect to itemset containment relations: Given two itemsets, \(I_1\) and \(I_2\), such that \(I_1 \subseteq I_2\), we are not guaranteed that \(\text{MAXPI}(I_1) \geq \text{MAXPI}(I_2)\). MAXPI does, however, exhibit a weak monotonic property, which states that if a \(k\)-itemset is MAXPI frequent, then at most one of its \((k-1)\)-subsets is not confident. By incorporating this weak monotonic property, an Apriori style algorithm to mine confident rules without a support threshold is possible.

**Definition 6 (MAXPI).** Given an itemset \(I\), the MAXPI of \(I\) is defined as the maximal participation ratio (\(pr\)) of all items \(i \in I\):

\[
\text{MAXPI}(I) = \max_{i \in I} (pr(I,i)), \text{ where}
\]

\[
pr(I,i) = \frac{\text{conf}(i \Rightarrow (I/i))}{\text{conf}(i \Rightarrow (I/i))}.
\]

One drawback of using MAXPI is that no 1-itemsets can be pruned in the first phase as they all have a confidence of 100 percent. Therefore, Apriori-MAXPI algorithms must deal with all the candidate 1-itemsets and the \(|I|^2\) 2-itemsets before any pruning can take place. Another downfall of MAXPI is that itemset pruning is not as stringent as that of support and thus works against Apriori, which is efficient on the assumption that the number of frequent itemsets is low. Furthermore, with a large number of items, like in microarray data, Apriori-MAXPI approaches significantly suffer from itemset explosion. Unfortunately, there is no property of MAXPI that can be exploited by a row-enumeration approach. Motivated by the possibility of mining high-confidence rules without support pruning from microarray data, we investigated and identified confidence pruning techniques that can be exploited by our row-enumeration algorithm, MAXCONF, which is described in the following section.

4 HIGH-CONFIDENCE RULE MINING

In this section, we introduce our row-enumeration approach to mining high-confidence rules efficiently. MAXCONF (Algorithm 1) addresses the two main shortcomings of AR mining: support pruning and itemset explosion. The main challenge is that no support pruning can take place to reduce the search space. A naive approach would be to grow the entire enumeration tree with no support pruning until no more itemsets can be formed. This would be equivalent to generating all closed itemsets, including those that cannot produce confident rules, and, thus, for large and dense data sets, it will require unnecessary expensive computations and memory. We applied this naive approach on the Hughes et al. [19] perturbation microarray data set and an error was reported after using up all available
memory (4 Gbyte RAM) when only 30 percent of the transactions had been processed.

**Algorithm 1.** MAXCONF—high-confidence rule mining

**Input:** Transaction database $D$, minimum confidence $\text{minconf}$

**Output:** High-confidence spanning rules satisfying $\text{minconf}$

Initialization:
Let $N$ = set of parent nodes corresponding to each transaction in $D$. Let $n.iitems = \text{itemset represented by node } n$ with support $\sigma(n)$. For each transaction node, $\sigma(n) = 1$ initially. Let $R := \emptyset$ be the set of maximal confident rules.

**Procedure:** MAXCONF_depthfirst($N$)

foreach node $n_i \in N$ do
1. if $n_i$ has been discovered then delete $n_i$ and return;
2. **Level 1 Confidence Pruning;**
   if $n_i$ cannot form a confident spanning rule then delete $n_i$ and continue;
3. **Expand subtree:**
   Calculate $\sigma(n_i)$ and form children of $n_i$;
4. **Maximal Rule Generation:**
   $M := \text{getMaxFeatures}(n_i)$;
   foreach $m \in M$ do
     if $m \not\subseteq n_i.iitems - m$ then add rule $m \Rightarrow \{n_i.iitems - m\}$ into $R$
5. **Level 2 Confidence Pruning:**
   foreach child $c \in n_i.children$ do
     if $c.iitems \cap M \neq \emptyset$ then delete $c$
6. if $n_i.children \neq \emptyset$ then MAXCONF_depthfirst($n_i.children$)

**Procedure getMaxFeatures($n$)**

maxFeatures := $\emptyset$;
foreach item $i \in n.iitems$ do
  if $\sigma(n_i)/\sigma(i) \geq \text{minconf}$ then maxFeatures.insert($i$)
return maxFeatures

MAXCONF exploits two confidence pruning methods, Level 1 and Level 2, allowing us to effectively prune the search space. **Level 1 pruning** will remove nodes that cannot generate confident $I$-spanning rules. **Level 2 pruning** removes nodes that can only generate confident $I$-spanning rules that can be derived from their parent node. MAXCONF is further enhanced to mine all maximal confident rules. These methods are detailed in Sections 4.1, 4.2, and 4.3 and we continue with our running example data set in Table 1a for detailed explanations. The rules generated by MAXCONF on this example data set are shown in Table 1b. Rules that are not identified using RERII when $\text{minsup} = 3$ are indicated in the last column. As can be seen in Table 1b, if support pruning takes place on this small data set, almost 50 percent of the rules are not identified.

**4.1 Level 1 Confidence Pruning**

This pruning is based on an observation of the row-enumeration tree’s structure. For each node in the tree, we can predict the maximum support [4] and confidence its corresponding itemset can exhibit based on its location within the tree. From this, our first confidence pruning technique is possible. It is based on the following definitions and is performed at Step 2 of Algorithm 1.

As in RERII, in MAXCONF, a node’s support will only increase if it is completely contained within one of its sibling nodes [4] (see Definition 5).

**Definition 7 (Minimum feature).** The item $i_1$ in the itemset $I$ is the minimum feature if

$$\sigma(i_1) \leq \sigma(i_2) \forall i_2 \in I.$$  \hfill (8)

**Definition 8 ($I$-spanning rule).** Given an itemset $I$, a rule $r$ is an $I$-spanning rule if

$$\text{antecedent}(r) \cup \text{consequent}(r) = I \quad \text{and} \quad |\text{antecedent}(r)| = 1.$$  \hfill (9)

**Definition 9 (Maximum confidence).** Given a node $n$ with minimum feature $i$, the maximum confidence of any spanning rule of the itemset at $n$ is

$$\text{conf}_{\text{max}}(n) = \frac{\sigma_{\text{max}}(n)}{\sigma(i)}.$$  \hfill (11)

If $\text{conf}_{\text{max}}(n) < \text{minconf}$, then $n$ can be pruned as any further enumeration below the node will only generate less or equally confident child rules. This is because the maximum support of any child node is bounded above by $\sigma_{\text{max}}(n)$ and the support of its minimum feature can only be greater than or equal to the minimum feature of $n$. Thus, the child node is bounded above by $\text{conf}_{\text{max}}(n)$.

**Example 1.** Consider node $\{5\} (ACEGI)$ in Fig. 1a. This node represents a transaction node; hence, its initial support is 1. As MAXCONF is a depth-first algorithm, when we reach node $\{5\}$, node $\{8\}$ has already been pruned as it was contained within node $\{1\}$ (see Fig. 1b). Similarly, nodes $\{2\}$ and $\{4\}$ are previously pruned. Therefore, when we consider node $\{5\}$, it has two sibling nodes. Thus, from Definition 5, $\sigma_{\text{max}}(ACEGI) = 1 + 2 = 3$. The minimum feature set of $ACEGI$ is $I \{i(1) = 4\}$ and the $\text{conf}_{\text{max}}(ACEGI)$ is thus $3/4$. Assuming $\text{minconf} = 4/5$, node $\{5\}$ can be pruned as it and any of its potential child nodes will not produce confident spanning rules (node $\{56\}$ has a $\text{conf}_{\text{max}}$ of $2/5$).

If the current parent node is not pruned by Level 1, it is expanded to form a subtree of child nodes following the approach of RERII [4]. This is performed at Step 3 of Algorithm 1 and, in doing so, the actual support of the current parent node is determined.

In comparison to FARMER [15] and TOPKRGs [13], our approach generates more complex rules with no restriction on the consequent item. In these algorithms, the consequent is fixed as a class. We effectively restrict our search to mining $I$-spanning rules. It is possible that we may lose confident relationships such as $AB \Rightarrow CD$ if we find that $ABCD$ cannot form any confident $I$-spanning rules and is pruned. This is because the rules $A \Rightarrow BCD$ and $B \Rightarrow ACD$ do not need to be confident for the rule $AB \Rightarrow CD$ to be. This restriction is necessary for any effective pruning based on confidence. To obtain the support of $AB$, we need to
expand the entire row-enumeration tree, which is infeasible. Although some complex rules may be lost, we can still find most complex rules while only testing for I-spanning rules. Our reasoning for this is based on the following lemma:

**Lemma 1.** Given an itemset $I$ and its set of confident spanning rules $CR$, let the set $A$ contain the single antecedents of the rules in $CR$. The rules in $CR$ can be easily combined into one confident rule of the form $A \Rightarrow I - A$.

**Proof 1.** Let $X \Rightarrow I - X$ be a confident spanning rule, then $X \in A$. Therefore, $\sigma(X) = \sigma(A)$. Thus, $\text{conf}(A \Rightarrow I - A) = \text{conf}(X \Rightarrow I - X) \geq \text{minconf}$. □

### 4.2 Level 2 Confidence Pruning

After the support of a current node is determined, maximal confident rules can be identified (Step 4), which is detailed in Section 4.3. Further pruning based on confidence is possible after rule generation. We identified the weak downward closure property of confidence, which can be exploited during the generation of the row-enumeration tree to effectively prune nodes that will provide redundant information. This pruning is performed in Step 5 and is based on the following definitions and lemma:

**Definition 10 (Maximum features).** Given an itemset $I$, let $R_I$ be the set of all confident I-spanning rules. The set of maximum features $M_I$ is the set of all antecedents of the spanning rules.

**Lemma 2 (Weak downward closure of confidence).** Let $M_I$ and $R_I$ be the set of maximum features and I-spanning rules derived from $I$, respectively. Let $k$ be a subset of $M_I$, then the confidence of any $k$-spanning rule is bounded below by the confidence of all rules in $R_I$.

**Proof 2.** Let $x \Rightarrow y$ be a $k$-spanning rule, then

$$\text{conf}(x \Rightarrow y) = \frac{\sigma(x \cup y)}{\sigma(x)} \geq \frac{\sigma(I)}{\sigma(x)} \geq \text{minconf}.$$  

The last inequality follows from the fact that $x \in M_I$. □

**Definition 11 (Subrules).** Given an itemset $I$, let $R_I$ be the set of all rules $\{x \Rightarrow y\}$, where $x \cup y = I$. The set of subrules $SR_I$ is the set of all rules generated from the itemset $I$ such that 1) $I_2 \subseteq I$ and 2) for each $sr \in SR_I$, a) antecedent$(sr) \in \text{antecedent}(R_I)$ and b) conf$(sr) \geq \text{conf}(R_I)$. For example, the rule $A \Rightarrow B$ (90 percent confidence) is a subrule of $A \Rightarrow BCD$ (80 percent confidence).

By extension of Lemma 2, if the maximum feature set $M$ of an itemset at node $n$ is not empty, we can prune all child nodes of $n$ whose itemsets are subsets of $M$ as we are guaranteed that such child nodes will only produce subrules of the confident rules generated by $n$ (Step 5).

**Example 2.** Consider node $\{1234\}$ ($CDEG$) in Fig. 1b. After calculating its support (generating its two child nodes in the process), we find the confident spanning rules $C \Rightarrow DEG, E \Rightarrow CDG$, and $G \Rightarrow CDE$ with confidence $4/6, 4/5$, and $4/4$, respectively. Thus, the maximum features of $CDEG$ is $CEG$. Immediately, the child node $\{12345\}$ ($CEG$) can be pruned as it is a subset of its parent’s maximum features. We can safely prune this node without calculating its support or forming any child nodes as these will only form the confident subrules $C \Rightarrow EG, E \Rightarrow CG$ and $G \Rightarrow CE$. In Fig. 1b, we can see that this effectively prevents the node $\{123456\}$ ($CG$) from being generated, which will also only form confident subrules.

### 4.3 Maximal Confident Rule Generation

We now present another property of confident rules that can be exploited to reduce the number of rules generated without any information loss. If the set of confident rules can be restricted to that of maximal confident rules (Definition 12), the number of rules can be significantly reduced. This approach can only be performed in a row-enumeration algorithm as it exploits the way in which child nodes are constructed and occurs at Step 4.

**Definition 12 (Maximal confident rules).** Let $R$ be the set of confident rules from a data set $D$. The set $MR$ of maximal confident rules is the set of rules from $R$ where, for each rule $r_1$, there does not exist another rule $r_2$ such that 1) antecedent$(r_1) = \text{antecedent}(r_2)$ and 2) consequent$(r_1) \subset \text{consequent}(r_2)$. For example, if the rules $A \Rightarrow BCD$ and $A \Rightarrow BC$ are confident, then $A \Rightarrow BCD$ is a maximal confident rule.

Assume that, during MAXCONF, we reach a node $n$ whose parent node $p$ had a maximum feature set $pM_n$ of cardinality $\geq 1$. We can restrict the rules generated by $n$ to those that are not subrules of rules identified by $p$. First, we identify the maximum feature set of $n$, $nM_n$. Then, for each item $i \in nM_n$ which is not in $pM_n$, we generate a confident spanning rule as any other confident rule from $n$ would be a subrule of one identified from $p$ and would thus be redundant. This simple test successfully restricts our search to mining maximal confident rules.

**Example 3.** Again, consider node $\{1234\}$ ($CDEG$) in Fig. 1b with the maximum feature set $CEG$. The child node $\{12348\}$ ($CDG$) cannot be pruned with Level 2 confidence pruning; however, we only need to consider rules with antecedent $D$. From node $\{1234\}$, we know that $C$ and $G$ produce confident rules and, thus, the rules $C \Rightarrow DG$ and $G \Rightarrow CD$ do not provide any information that is not contained within the maximal rules identified at node $CDEG$.

### 5 EXPERIMENTAL RESULTS AND EVALUATION

We evaluated and compared MAXCONF against RERII [4] on three microarray data sets of *S. cerevisiae* described in Table 2. The first two data sets correspond to perturbation microarrays and the last is a temporal data set. In our experiments, we have not taken into account the sequential nature of this final data set, treating each time measurement as an individual experiment. For each microarray data set, each gene is converted into one of three items, down-regulated, up-regulated, or normal expression, depending on its level of expression in the experiments, as in [3]. This is performed by binning the $\log_2$ of the expression level into the three classes with bounds $\leq -0.2, \geq 0.2$, or in between,
respectively. The final transactions are formed from the items corresponding to the up and down-regulated gene items. All experiments were performed on a PC with a 3.2 GHz Pentium 4 Xeon, 1 Mbyte L3 cache, and 4 Gbyte RAM.

5.1 Rule Generation

The main downfall of RERII is its inability to extract all ARs that satisfy \( \minconf \) due to support pruning. Indeed, using the Hughes et al. [19] data set with \( \minsup = 5 \text{ percent} \), 90.6 percent of the 1-itemsets are pruned in the first stage before row-enumeration begins. This leaves only 502 different items that may be included in the frequent itemsets and confident rules. Without any support cut-off being necessary, MAXCONF mines rules considering all 10,044 items and, as such, is capable of detecting many more rules with high confidence, as shown in Fig. 2b. Figs. 3b and 4b also highlight the drastic effects of support pruning on rule generation. When the support of RERII is lowered to zero in an attempt to find all confident rules, no rules were ever generated as the program required too much memory. RERII also failed when the support was decreased to 10 percent on the Spellman et al. [9] data set (Fig. 4b).

5.2 Scalability

In this set of experiments, we studied the effect of varying \( \minconf \) (and \( \minsup \) with RERII) on the execution time. The results of these are shown in Figs. 2a, 3a, and 4a. Intuitively, with respect to support pruning, a higher \( \minsup \) results in more pruning and, thus, the runtime is decreased. The performance of RERII is not affected by \( \minconf \). This is because confidence is only taken into account after all frequent itemsets are formed. The scalability of MAXCONF, on the other hand, improves as \( \minconf \) is increased. In addition, MAXCONF is significantly more efficient than RERII on both the perturbation data sets (Figs. 2a and 3a). RERII only outperforms MAXCONF on the Spellman et al. [9] data set when \( \minsup \) is increased to 40 percent (Fig. 4a). However, as shown in Fig. 4b, this has little advantage on rule generation with significantly less rules identified. This performance of MAXCONF is based on the fact that an itemset satisfying \( \minsup \) is not guaranteed to produce any confident rules. Therefore, the confidence pruning of MAXCONF can be considered more stringent than support pruning with respect to mining microarray data.

5.3 Biological Rule Analysis

In this section, we report on how the biological significance of the rules mined by MAXCONF and RERII were evaluated. This is not a straightforward task. Since our approach is not a classification task where testing/evaluation data sets are available, we can only evaluate our rules based on documented gene relationships. Many generated rules should correspond to known biological relationships between genes; however, a noncorresponding rule does not imply an incorrect relationship. This is because the mined relationships may not have been hypothesized yet. In fact, biologists often perform microarray experiments to formulate new hypotheses from unknown and/or unexpected gene relationships.

Our evaluation proceeds as follows: First, we concentrate on how effective MAXCONF and RERII are in detecting known direct biological interactions in BIND [7]. As not all gene relationships are direct interactions, we then evaluate our rules with the Gene Ontology (GO) [8]. We show that many of our rules also contain GO gene relationships. Finally, as an example, we address the iron uptake pathway,

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Genes</th>
<th># Items</th>
<th># Trans.</th>
<th>Mean trans. size</th>
<th>Min. trans. size</th>
<th>Max. trans. size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hughes et al. (2000) [19]</td>
<td>6316</td>
<td>10044</td>
<td>300</td>
<td>198</td>
<td>2</td>
<td>2339</td>
</tr>
<tr>
<td>Mnaimneh et al. (2004) [20]</td>
<td>6316</td>
<td>8330</td>
<td>215</td>
<td>228</td>
<td>7</td>
<td>1111</td>
</tr>
<tr>
<td>Spellman et al. (1999) [9]</td>
<td>6178</td>
<td>6179</td>
<td>82</td>
<td>1397</td>
<td>205</td>
<td>2613</td>
</tr>
</tbody>
</table>

Fig. 2. Performance on the Hughes et al. [19] data set of RERII with various supports and MAXCONF as confidence is increased. RERII with \( \minsup = 0\% \) failed to complete due to an out-of-memory error. The key in (b) is also for use in (a). (a) Scalability. (b) Number of rules discovered.
presenting some sample rules identified by MAXCONF that correctly describe gene relationships in this system. From our analysis, we confirm the appropriateness of MAXCONF for mining various types of gene relationships. For brevity, we only discuss our results regarding the Hughes et al. [19] data set.

5.3.1 Direct Interactions
The BIND [7] is an online archive of pairwise information about direct interactions (DIs) that can occur between two biological entities. We have used BIND to analyze the biological relevance of the rules we identify based on precision and recall for DIs.

Using BIND, we determined the percentage of rules mined with MAXCONF that exhibit a DI between at least two of their items, that is, precision. This is based on the rationale that, for a DI to occur between two or more genes/proteins, it is highly probable that their expressions are correlated and, thus, they are likely to be present together in at least one rule. Furthermore, we analyzed the effectiveness of our approach to identify all possible DIs from the data set, that is, recall.

To calculate the precision and recall of DIs, we first need to determine which DIs are actually possible within the microarray data set we analyzed. This is done by forming all pairs of up-regulated genes in each experiment. If a DI is known to occur between the genes and/or their protein products, we store the gene pair as a desired relationship to identify.

Definition 13 (Precision). Let $\mathcal{R}$ be the set of mined ARs and $\mathcal{B}$ be the set of pairwise DIs in the microarray data set in the form of rules. The precision of DIs in $\mathcal{R}$ is

$$\text{Precision} = \frac{\# \text{ rules in } \mathcal{R} \cap \mathcal{B}}{\# \text{ rules in } \mathcal{R}}.$$  \hfill (12)

Definition 14 (Recall). The recall of DIs in $\mathcal{R}$ is

$$\text{Recall} = \frac{\# \text{ rules in } \mathcal{R} \cap \mathcal{B}}{\# \text{ rules in } \mathcal{B}}.$$  \hfill (13)

The recall of a system is the percentage of possible DIs that are contained within at least one rule. For a more detailed analysis, we include two recall measures: Recall 1 and Recall 2. Recall 1 only includes the identified DIs where the antecedent of the rule binds at least one of the
consequents. Recall 2 also includes DIs between genes that are consequents of rules; however, these rules must also satisfy Recall 1.

The results of our BIND analysis are summarized in Table 3. MAXCONF is clearly more effective than the support-based method. The significant improvement from Recall 1 and Recall 2 is expected as more relationships within the rules are considered. The high recall (94 percent) obtained by MAXCONF is superior compared with using RERII (1.5 percent) with minconf = 80 percent (and minsup = 1 percent for RERII). This extremely low recall for RERII is a significant weakness of support-based pruning and highlights the importance of mining high-confidence rules in dense perturbation microarrays. Many of the DIs were not detected by RERII as 96.5 percent of the genes were immediately pruned based on support during preprocessing.

Rules 1, 2, and 3 in Table 4 are example rules displaying DIs. Both rules 1 and 3 would not be identified unless the support threshold for RERII was decreased significantly (if possible). In rule 1, ERG28 binds ERG25; however, there is no known link between these genes and FMP17. Rule 2, with its high support is the most common rule published to validate previous approaches, as in [3], due to the well-documented DI between SNO1 and SNZ1. Inspection of the rules generated by RERII showed that the majority of rules containing a DI contained the genes SNO1 and SNZ1. Rule 3, with 100 percent confidence and 0.33 percent support, correctly describes the relationships between all three genes (CSE1 binds PCL5, which in turn is able to bind CRM1).

Although we achieved high recall with respect to DIs, only a slight improvement in precision was achieved. However, this does not reflect the inappropriateness of mining high-confidence rules. A set of genes can be highly related without interacting and, therefore, will not be mentioned in the BIND. Furthermore, not all gene relationships we identify can convey DIs. For examples, rules 8 and 10 in Table 4 only include down-regulated genes which are not expressed and, thus, a DI between these genes cannot occur. However, we cannot yet confirm, based on this evaluation, that these gene sets are not related and are thus false positives. Therefore, we hypothesize that our low precision is an indication that we are identifying other possible relationships that are not documented DIs. This

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Supp. (%)</th>
<th>Conf. (%)</th>
<th># Rules</th>
<th>Bind Analysis</th>
<th>Gene Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>RERII</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>80</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1.0</td>
<td>80</td>
<td>15669</td>
<td>0.3</td>
<td>0.9</td>
<td>1.5</td>
</tr>
<tr>
<td>2.5</td>
<td>80</td>
<td>12231</td>
<td>0.1</td>
<td>0.3</td>
<td>1.2</td>
</tr>
<tr>
<td>5.0</td>
<td>80</td>
<td>8083</td>
<td>0.5</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>7.5</td>
<td>80</td>
<td>5223</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>10.0</td>
<td>80</td>
<td>2123</td>
<td>1.9</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>MAXCONF</td>
<td>NA</td>
<td>19090</td>
<td>1.2</td>
<td>26.3</td>
<td>94.0</td>
</tr>
<tr>
<td>NA</td>
<td>80</td>
<td>12424</td>
<td>1.6</td>
<td>26.1</td>
<td>93.0</td>
</tr>
<tr>
<td>NA</td>
<td>90</td>
<td>8296</td>
<td>2.3</td>
<td>25.9</td>
<td>86.6</td>
</tr>
<tr>
<td>NA</td>
<td>95</td>
<td>7214</td>
<td>2.6</td>
<td>25.7</td>
<td>83.0</td>
</tr>
<tr>
<td>NA</td>
<td>100</td>
<td>7076</td>
<td>2.7</td>
<td>25.7</td>
<td>81.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Association Rule</th>
<th>Supp. (%)</th>
<th>Conf. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FMP17 (\Rightarrow) ERG28, ERG25</td>
<td>0.60</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>CTF13 (\Rightarrow) SNO1, SNZ1</td>
<td>21.0</td>
<td>80.8</td>
</tr>
<tr>
<td>3</td>
<td>CSE1 (\Rightarrow) CRM1, PCL5</td>
<td>0.33</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>EUG1 (\Rightarrow) BNA2, GCS2, PDH1, TFS1, TH15, TH11, TH13, YGR043c, YML131w</td>
<td>1.30</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>SIL1 (\Rightarrow) AFR1, GCS2, YPS1, YOR289w</td>
<td>2.67</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>FRE6 (\Rightarrow) SIT1, ARN1, ARN2, ENB1, FIT2, FIT3</td>
<td>4.33</td>
<td>100</td>
</tr>
<tr>
<td>7</td>
<td>AKR1 (\Rightarrow) CCC2, SIT1, FTR1, ARN1, ARN2, FIT3, ENB1, FIT2, FIT3</td>
<td>3.33</td>
<td>90</td>
</tr>
<tr>
<td>8</td>
<td>MAC1 (\Rightarrow) FRE7</td>
<td>0.33</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>MEP2 (\Rightarrow) GLK1, GLC3, DMC1, HSP12, PRY1, NCA3, TFS1, MSC1, PGM2, YGP1</td>
<td>1.00</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>ESC8 (\Rightarrow) IMD1, IMD2</td>
<td>1.30</td>
<td>100</td>
</tr>
</tbody>
</table>
forms the basis of our next evaluation scheme using the GO to investigate how biologically relevant our rules are with respect to other gene relationships.

5.3.2 Gene Ontology

In this section, we evaluate the rules based on the GO. The GO [8] is an international standard to annotate genes in three distinct categories: molecular function, biological process, and cellular component. The GO has a hierarchical structure starting with top-level ontologies to specific descriptions with increasing depth. If a rule describes biologically meaningful relationships between its genes, we would expect the genes to share common GO annotations. Based on this, we evaluated the rules we identify using GOstat [21], a Web-based query engine wrapper of the GO database. GOstat determines, for a group of genes, GO annotations that are statistically overrepresented within the group. To take full advantage of this query engine, we developed an automated process using Python and CGI scripts to scrape the HTML results produced by GOstat for each individual rule. Rules that contained an antecedent gene that shared a GO annotation with any genes in the consequent items were said to contain a biologically meaningful relationship. We chose a minimum depth of 4 within the GO hierarchy to ensure that the GO annotations between the items represented more specific gene relationships.

These results are summarized in the last two columns of Table 3. The first of these columns shows the number of rules identified by each system that contain a GO relationship. The second column is the percentage of rules. We chose to show both of these values to highlight the difference between RERII and MAXCONF.

The rules generated by MAXCONF are more biologically meaningful than the rules identified by RERII with 80 percent confidence. Although a high percentage of rules mined by RERII with \( \text{minsup} = 7.5 \) percent contained a GO relationship (81.4 percent), the raw number of rules was significantly less than those mined by MAXCONF. Furthermore, as \( \text{minconf} \) was increased for MAXCONF, the rules mined were more biologically significant. These results strengthen our argument that support pruning is not always ideal for identifying relationships from perturbation microarrays.

Table 5 shows the GO annotation breakdown of three example rules (rules 4, 9, and 10) from Table 4. This table is read as follows: Rule 9 is separated into three related gene groups; for example, the genes DMC1 and MSC1 are both assigned to the ontology term meiotic recombination. This GO term has a depth of 9 within the GO and the genes DMC1 and MSC1 are statistically overrepresented in this group with a P value of 0.0475.

Rule 10 is a perfect example of a biologically significant rule. It cannot express a DI from BIND due to its items corresponding to down-regulated genes; however, the two items share a common GO term. Of interest is the gene IMD1, which is not linked to either of the other genes. Furthermore, this gene has not yet been assigned a GO term. Therefore, we consider this rule a potential candidate for presenting new information to biologists, where, in turn, they may be able to use this rule to hypothesize possible reasons for its association with the other genes.

Rules like these, containing GO but no BIND relationships, also confirm the notion that not all gene relationships are DIs, and hence, rules not depicting DIs can still be biologically interesting. Therefore, our intuition regarding the low precision for DIs is correct. Additionally, rules containing gene sets that are not related with respect to the GO can also be considered interesting. This is because a main goal for generating perturbation microarray data is to identify unknown gene relationships, which can then be further analyzed in other experiments. Thus, MAXCONF may be effectively used as a discovery tool for formulating new hypotheses from microarray experiments.

5.3.3 Iron Uptake Pathway

In this section, we further demonstrate the usefulness of MAXCONF for extracting correct gene relationships by examining the \( S.\text{cerevisiae} \) iron uptake systems. \( S.\text{cerevisiae} \) has two different mechanisms to obtain iron from the external environment, which, combined, form the iron uptake pathway [22], [23]. One system of the iron uptake pathway depends on a family of high-affinity transporter proteins encoded by the genes ARN1, ARN2, SIT1, and ENB1.
Therefore, for this system to function, these genes need to be coexpressed. Another subsystem requires some, if not all, of the proteins FRE1-7, FET3, FIT2-3, and FTR1 [23]. MAXCONF was able to detect similar significant biological patterns, three of which are shown in Table 4 (rules 6, 7, and 8). In particular, rule 8 indicates the strength of MAXCONF for analyzing perturbation microarray experiments. This rule exhibits extremely low support and, thus, it would have been impossible for it to be mined using a support-based approach (unless \textit{minsup} was set to a very low value). Furthermore, although this rule cannot exhibit a DI, it is of biological significance. The gene MAC1 was selectively mutated in the Hughes et al. [19] data set and this rule correctly describes the relationship between the genes MAC1 and FRE7. More specifically, MAC1 activates the expression of the gene FRE7 [24]. Therefore, FRE7 cannot be expressed when MAC1 is not and this rule correctly indicates this causality.

6 Conclusions

In this paper, we introduced the first truly scalable approach for discovering gene relationships from microarray data. Traditional data mining methods, which are optimized for sparse data sets, are impractical for analyzing microarray data. Recently, row-enumeration rule mining algorithms have been developed to facilitate mining in dense data sets. However, until now, all algorithms proposed relied on the support measure to prune the search space. This is a major shortcoming as many potentially interesting gene relationships which have low support and high confidence are pruned. Our proposed rule mining algorithm, MAXCONF, effectively overcomes this, discovering high-confidence rules from dense microarray data. MAXCONF is a support-free row-enumeration algorithm that exploits two new confidence pruning techniques and restricts the rule discovery to maximal confident rules.

We performed experiments on three microarray data sets evaluating the performance of MAXCONF in terms of the number of rules discovered and scalability. Our results demonstrate that support-based pruning drastically reduces the number of gene relationships that can be mined. MAXCONF, on the other hand, can extract significantly more gene relationships with high confidence and low support from microarrays. Our performance study also shows that MAXCONF is more scalable than support-based AR algorithms.

We evaluated the biological significance of the rules discovered by MAXCONF using the BIND and GO resources. Our validation on the BIND shows that, with a \textit{minconf} of 80 percent, we are able to achieve a recall of 94 percent for extracting known DIs. This is superior compared with using a support-based method, where, with a low \textit{minsup} of 1 percent, only 1.5 percent of DIs were discovered. Although precision was considerably lower than recall and only increased slightly using MAXCONF, the majority of the rules discovered depicted other known gene relationships, as highlighted in our GO evaluation. As MAXCONF outperforms other approaches, we consider MAXCONF to be an excellent candidate for discovering gene relationships from microarrays. Therefore, we are convinced that MAXCONF will be a significant contribution to the biomedical and molecular biology domains.

Acknowledgments

This research was partially funded by the Australian Research Council (ARC) Discovery Grant DP0559005. The authors would like to thank James Curran and the anonymous reviewers for their useful comments. Preliminary work reported in this paper was presented at the Fifth ACM SIGKDD Workshop in Data Mining in Bioinformatics (BIOKDD 2005) [25].

References


Tara McIntosh received the BS degree in bioinformatics from the University of Sydney, Australia, in 2005. In March 2006, she started work on the PhD degree at the University of Sydney in computational linguistics and information retrieval for biomedical literature. Her research is supported by the Australian Postgraduate Award and Australia’s Commonwealth Scientific and Industrial Research Organisation (CSIRO). Her research interests include computational linguistics, information retrieval, data mining, machine learning, and their applications to biology.

Sanjay Chawla is an associate professor in the School of Information Technologies, University of Sydney, Australia. He works and publishes in the area of data mining and spatial database management systems. His work has appeared in premier data mining conferences and journals including ACM SIGKDD, IEEE ICDM, SDM, IEEE TKDE, and DMKD. His paper on “Mining for Outliers in Sequential Databases” received the best application paper award at SDM 2006. He is also the research leader of the data mining program of the Capital Markets CRC. He is a coauthor of the text Spatial Databases: A Tour (2002), which has recently been translated into Chinese and Russian. He serves on the program committee of IEEE ICDM, SDM, and PAKDD.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.