High Confidence Rule Mining for Microarray Analysis

Tara McIntosh, Sanjay Chawla

T. McIntosh and S. Chawla are with The School of Information Technologies, The University of Sydney, Australia. Email: {tara, chawla}@it.usyd.edu.au

Abstract

We present an association rule mining method for mining high confidence rules, which describe interesting gene relationships from microarray datasets. Microarray datasets typically contain an order of magnitude more genes than experiments, rendering many data mining methods impractical as they are optimised for sparse datasets. A new family of row-enumeration rule mining algorithms have emerged to facilitate mining in dense datasets. 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 which directly uses the confidence measure to effectively prune the search space. Experiments on three microarray datasets 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.

I. 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 disease and developing specific gene-targeting treatments.

Microarrays have revolutionised the way in which biological research is carried out. They allow biologists to analyse the behaviour 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 which render many traditional data mining techniques infeasible to extract and explore 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% 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 which allows for the simultaneous discovery of relationships between attributes. AR algorithms can extract associations among genes from microarray datasets, 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% of the time GENE2 is also expressed, and that GENE1 and GENE2 are expressed together in 10% of the microarray experiments.

In comparison to 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 which are considered uncommon/infrequent. As a result, AR algorithms search for common gene relationships within experiments. These AR algorithms however were developed for sparse datasets, 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 row-enumeration AR mining algorithms have been introduced to prevent itemset explosion, allowing the mining of dense datasets [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[1], 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 to the recently introduced support-based row-enumeration algorithm RERII [4]. Our evaluation on three microarray datasets demonstrates how MAXCONF outper-

[1]The MAXCONF implementation and source code are available by request from the authors.

forms 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 database [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 organised as follows: In Section II we introduce microarrays and their characteristics which make analysis difficult. In Section III we present the relevant work in the literature, which motivated the development of our MAXCONF algorithm described in Section IV. The experimental results of our evaluation are outlined in Section V. In Section VI we conclude this paper with a summary.

II. 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.

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 a NxM 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 $\gg$ 6000). This data form renders many traditional data mining algorithms ineffective as these algorithms are designed to mine sparse data, where the number of non-zero columns is a small fraction of the number of rows. This aspect will be further detailed in Section III.

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] dataset 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 dataset [10] contains the expression values of 12,600 genes from 52 cancerous and 50 healthy prostate cells.

In this paper, we concentrate on analysing 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 which 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 analysing 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 not only contain common relationships but also rare relationships describing the affects of the perturbations. Therefore, methods designed to analyse duplicate microarrays are not particularly effective for analysing 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.

III. Related Work

A. Association Rule Mining

AR mining was originally designed to examine the behaviour 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 dataset $\mathcal{D} = \{t_1, t_2, \ldots, t_n\}$ be a set of n transactions and let $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ be the set of all possible items (m). Each transaction t consists of a set of items I from $\mathcal{I}$. 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 $\mathcal{D}$. 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

TABLE I

EXAMPLE TRANSACTION DATASET AND RULES (MINSUP ≥ 3 AND MINCONF $\geq 4/5$)

(a) Transaction set

Transaction	Items
1	A B C D E G
2	A C D E G
3	C D E F G H I
4	B C D E G
5	A C E G I
6	A D I
7	D I J
8	A B C D G

(b) Rules found by MAXCONF and RERII

MAXCONF Rule	Confidence	Support	RERII Found
C $\Rightarrow$ DEG	4/6	4	Yes
E $\Rightarrow$ CDG	4/5	4	Yes
G $\Rightarrow$ CDE	4/4	4	Yes
A $\Rightarrow$ CG	4/5	4	Yes
C $\Rightarrow$ AG	4/6	4	Yes
G $\Rightarrow$ AC	4/6	4	Yes
A $\Rightarrow$ D	4/5	4	Yes
B $\Rightarrow$ CDEG	2/3	2	No
B $\Rightarrow$ CDG	3/3	3	No
I $\Rightarrow$ D	3/4	3	No
J $\Rightarrow$ DI	1/1	1	No
F $\Rightarrow$ CDEGHI	1/1	1	No
H $\Rightarrow$ CDEFGI	1/1	1	No

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 $I \subseteq \mathcal{I}$ be a set of items from $\mathcal{D}$. The *support* of an *itemset* I in $\mathcal{D}$, denoted by $\sigma(I)$, is the proportion of transactions that contain I:

$$\sigma(I) = \frac{\text{\# of transactions containing } I}{\text{\# of transactions}} \tag{1}$$

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 $conf(I_1 \Rightarrow I_2)$, refers to the strength of the association and is given by:

$$\frac{\sigma(I_1 \cup I_2)}{\sigma(I_1)} \tag{2}$$

For example, the support and confidence of the rule $A \Rightarrow CD$ in Table I(a) 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 dataset. We refer to any itemset which is generated and whose support is counted during this process as a *candidate itemset*. To naïvely 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 dataset with items $\mathcal{I}$, let I_1 and I_2 be two itemsets such that $I_1, I_2 \subseteq \mathcal{I}$, then:

$$I_1 \subseteq I_2 \implies \sigma(I_1) \geq \sigma(I_2) \tag{3}$$

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 I(a) 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 remaining 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 datasets. 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 which 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 V.

B. 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 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:

$$I_1 \subset I_2 \quad \text{and} \quad \sigma(I_1) = \sigma(I_2) \tag{4}$$

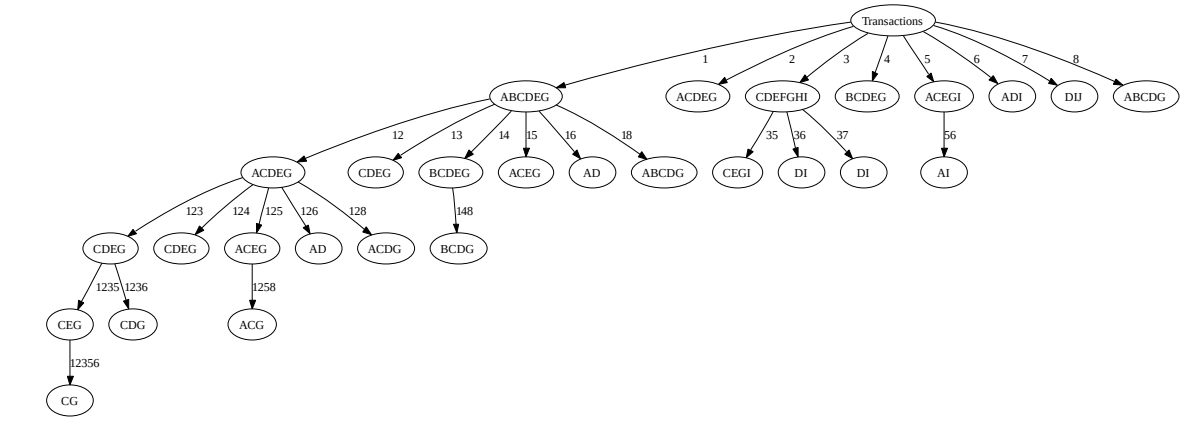
FARMER and TOPKRGS were specifically designed to generate *classification* ARs of the form $X \Rightarrow C$, where C is a class label [13], [15]. These two algorithms require duplicate microarray data, where each microarray experiment is classified into one of two classes prior to mining.

The algorithms CARPENTER and it's extension RERII are designed to mine *frequent closed itemsets* (FCI) from microarray data which 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.

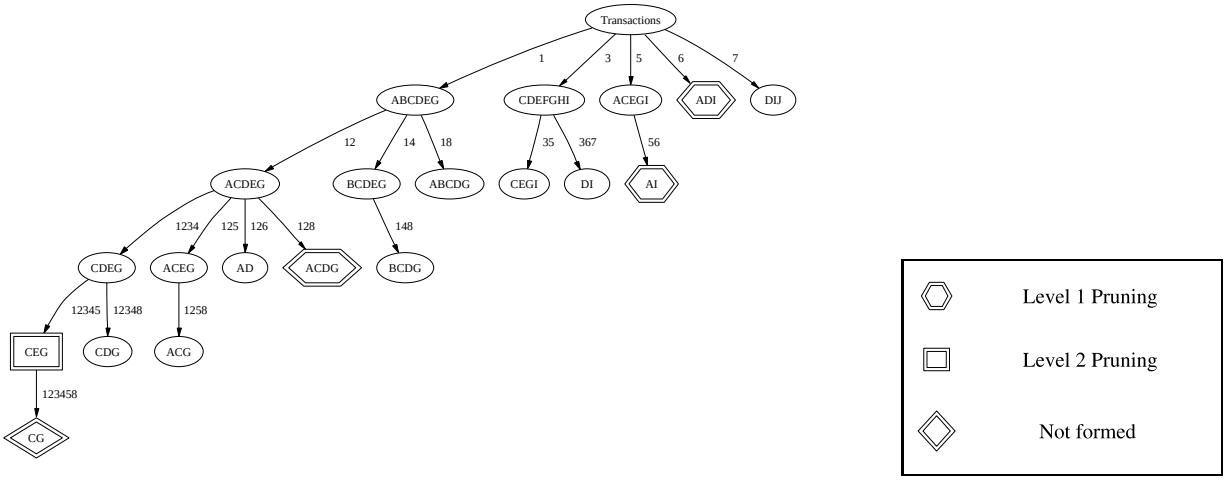
RERII extracts all FCI by searching the row-enumeration space depth-first. It begins by removing all infrequent 1-itemsets from the transactions. These transactions are then considered as individual itemsets, each assigned a support of 1. These itemsets are then intersected with one another, iteratively generating sub-itemsets 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 I(a) is represented as a row-enumeration tree in Fig. 1(a). Here, each node n corresponds to an itemset whose child nodes, $c(n)$, correspond to sub-itemsets 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. 1(a), the node $\{12\}$ (indicated by the edge label)



(a) Complete row-enumeration tree



(b) Pruned row-enumeration tree

(c) Key

Fig. 1. MAXCONF row-enumeration tree before and after pruning

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 2. There are two situations when a resulting intersection does not form a child node:

1) If the intersection is a 1-itemset, the child node is not formed as this simply cannot form an association rule. 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 $c(n)$ are 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 initialisation, 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 $\sigma_{max}(n)$, or any of n's potential child nodes is:

$$\sigma_{max}(n) = n.\text{initial\_support} + k \tag{5}$$

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 2, will only be intersected with one node ($|s(\{16\})| = 1$). Thus, the support of $\{16\}$ and all $c(\{16\})$ will be at most 3. If a node's maximum support is less than *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 *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.

C. 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 datasets. 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 $I_1 \subset I_2$, we are not guaranteed that MAXPI(I_1) $\geq$ 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 (Maximal Participation Index): Given an itemset I, the *maximal participation index* of I is defined as the maximal participation ratio (pr) of all items $i \in I$:

$$\text{MAXPI}(I) = max_{i \in I}\{ pr(I, i) \} \text{ where} \quad (6)$$

$$pr(I, i) = conf(i \Rightarrow (I/i)) \quad (7)$$

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%. Therefore, Apriori-MAXPI algorithms must deal with all the candidate 1-itemsets and the $|\mathcal{I}|^2$ 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.

IV. 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 association rule mining: support pruning and itemset explosion. The main challenge is that no support pruning can

take place to reduce the search space. A naïve 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 datasets it will require unnecessary expensive computations and memory. We applied this naïve approach on the Hughes et al. [19] perturbation microarray dataset, and an error was reported after using up all available memory (4GB RAM) when only 30% of the transactions had been processed.

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 which cannot generate confident I-spanning rules. *Level 2 pruning* removes nodes which 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 the following sections and we continue with our running example dataset in Table I(a) for detailed explanations. The rules generated by MAXCONF on this example dataset are shown in Table I(b). Rules which are not identified using RERII when *minsup* = 3 are indicated in the last column. As can be seen in Table I(b), if support pruning takes place on this small dataset almost 50% of the rules are not identified.

A. 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 Def. 5).

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

$$\sigma(i_1) \leq \sigma(i_2) \,|\, \forall i_2 \in I \tag{8}$$

Algorithm 1: MAXCONF - High Confidence Rule Mining

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

Output: High confidence spanning rules satisfying *minconf*

Initialisation:

Let N = set of parent nodes corresponding to each transaction in $\mathcal{D}$. Let $n.items$ = 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:= getMaxFeatures(n_i);

foreach $m \in M$ **do**
if $m \notin n_i$.parentMaxFeatures **then** add rule $m \Rightarrow \{n_i.items - m\}$ into R

5 *Level 2 Confidence Pruning*;

foreach child $c \in n_i.children$ **do**
if $c.items \subset M$ **then** delete c

6 **if** $n_i.childen \neq \emptyset$ **then** MAXCONF_depthfirst($n_i.children$)

Procedure: getMaxFeatures(n)

7 maxFeatures := $\emptyset$;

foreach item $i \in n.items$ **do**
if $\sigma(n)/\sigma(i) \geq$ *minconf* **then** maxFeatures.insert(i)

return maxFeatures

Definition 8 (I-Spanning Rule): Given an itemset I, a rule r is an *I-spanning rule* if:

$$\text{antecedent}(r) \cup \text{consequent}(r) = I \text{ and} \tag{9}$$

$$|\text{antecedent}(r)| = 1 \tag{10}$$

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:

$$conf_{max}(n) = \frac{\sigma_{max}(n)}{\sigma(i)} \tag{11}$$

If $conf_{max}(n) < minconf$, then n can be pruned as any further enumeration below the node will only generate less than or equally confident child rules. This is because the maximum support of any child node is bounded above by $\sigma_{max}(n)$, 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 $conf_{max}(n)$.

Example 1: Consider node $\{5\}$ ($ACEGI$) in Fig. 1(a). 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 being pruned as it was contained within node $\{1\}$ (see Fig. 1(b)). Similarly nodes $\{2\}$ and $\{4\}$ are previously pruned. Therefore, when we consider node $\{5\}$, it has 2 sibling nodes. Thus from Def. 5, $\sigma_{max}(ACEGI) = 1 + 2 = 3$. The minimum feature set of $ACEGI$ is I ($\sigma(I) = 4$) and the $conf_{max}(ACEGI)$ is thus $3/4$. Assuming *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 $conf_{max}$ 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 loose 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 $\sigma(X) \geq \sigma(A)$. Thus, $conf(A \Rightarrow I - A) \geq conf(X \Rightarrow I - X) \geq$ ***minconf***.

B. 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 IV-C. 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 which will provide redundant information. This pruning is performed in Step 5 and is based on the following definitions and Lemma 2.

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 (Confidence weak downward closed): 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 ***conf***$(x \Rightarrow y) = \sigma(x \cup y)/\sigma(x) \geq \sigma(I)/\sigma(x) \geq$ ***minconf***. The last inequality follows from the fact that $x \in M_I$.

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

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 sub-rules of the confident rules generated by n (Step 5). After Level 2 pruning MAXCONF continues recursively (Step 6).

Example 2: Consider node $\{1234\}$ ($CDEG$) in Fig. 1(b). 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 sub-rules $C \Rightarrow EG, E \Rightarrow CG$ and $G \Rightarrow CE$. From Fig. 1(b) we can see that this effectively prevents the node $\{123456\}$ (CG) from being generated, which will also only form confident sub-rules.

C. Maximal Confident Rule Generation

We now present another property of confident rules which 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 $\mathcal{R}$ be the set of confident rules from a dataset $\mathcal{D}$. The set $\mathcal{MR}$ of *maximal confident rules* is the set of rules from $\mathcal{R}$, where for each rule r_1 there does not exist another rule, r_2, such that: (1) antecedent(r_1) = antecedent(r_2) and (2) consequent(r_1) $\subset$ 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 p_M of cardinality > 1. We can restrict the rules generated by n to those which are not sub-rules of rules identified by p. Firstly, we identify the maximum feature set of n, n_M. Then, for each item $i \in n_M$, which is not in p_M, we generate a confident spanning rule, as any other confident rule from n would be a sub-rule of one identified from p, and thus be redundant. This simple test successfully restricts our search to mining maximal confident rules.

Example 3: Again, consider node $\{1234\}$ ($CDEG$) in Fig. 1(b), 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 C and G produce confident rules, and thus the rules $C \Rightarrow DG$ and $G \Rightarrow CD$ do not provide any information which is not contained within the maximal rules identified at node $CDEG$.

TABLE II

MICROARRAY DATASETS USED IN EXPERIMENTS

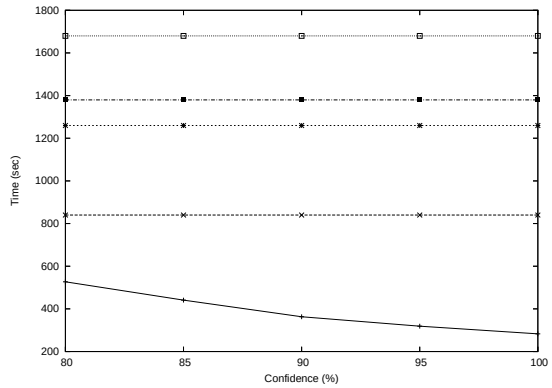
Dataset	# Genes	# Items	#Trans.	Mean trans. size	Min. trans. size	Max. trans. size
Hughes *et al.* (2000) [19]	6316	10044	300	198	2	2339
Mnaimneh *et al.* (2004) [20]	6316	8330	215	228	7	1111
Spellman *et al.* (1999) [9]	6178	6179	82	1397	205	2613

V. EXPERIMENTAL RESULTS AND EVALUATION

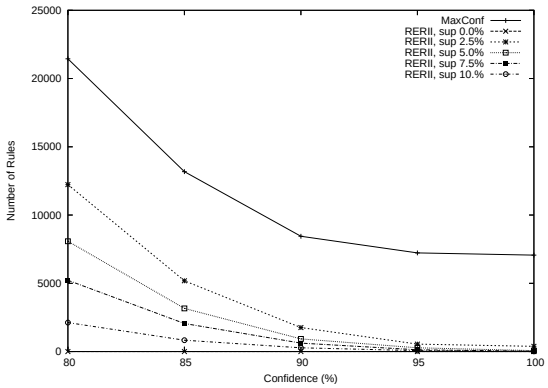
We evaluated and compared MAXCONF against RERII [4] on three microarray datasets of *S.cerevisiae* described in Table II. The first two datasets correspond to perturbation microarrays and the last is a temporal dataset. In our experiments we have not taken into account the sequential nature of this final dataset, treating each time measurement as an individual experiment. For each microarray dataset, 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 3.2Ghz Pentium 4 Xeon, 1MB L3 cache, and 4GB RAM.

A. Rule Generation

The main downfall of RERII is its inability to extract all association rules that satisfy *minconf* due to support pruning. Indeed, using the Hughes *et al.* [19] dataset with *minsup* = 5%, 90.6% of the 1-itemsets are pruned in the first stage before row-enumeration begins. This leaves only 502 different items which may be included in the frequent itemsets and confident rules. Without any support cut-off necessary MAXCONF mines rules considering all 10044 items, and as such is capable of detecting many more rules with high confidence, as shown in Fig. 2(b). Fig. 3(b) and 4(b) 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% on the Spellman *et al.* [9] dataset (Fig. 4(b)).

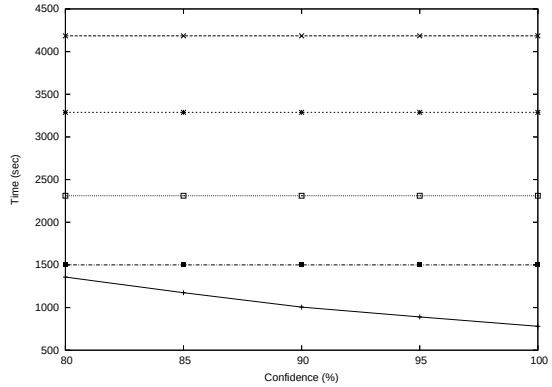


(a) Scalability

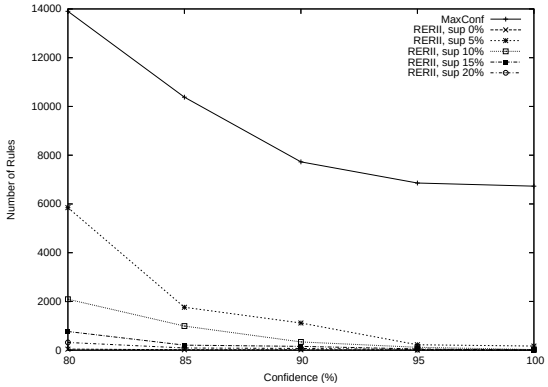


(b) # Rules Discovered

Fig. 2. Performance, on the Hughes *et al.* [19] dataset, 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).

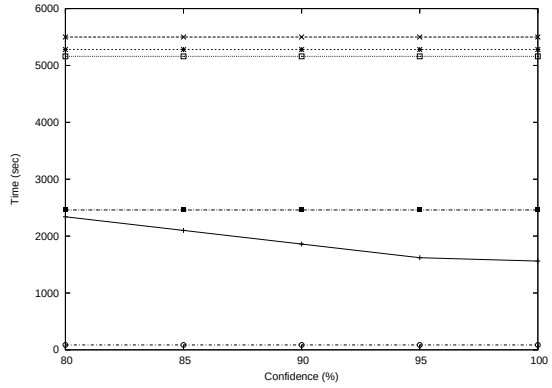


(a) Scalability

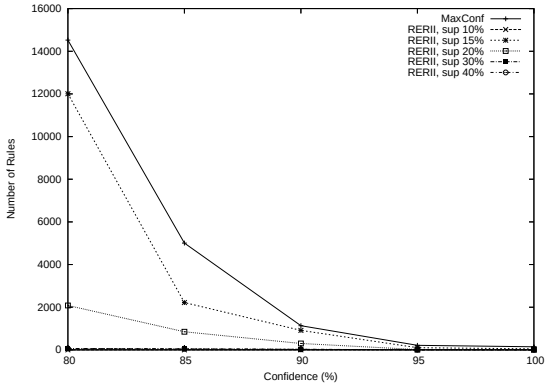


(b) # Rules Discovered

Fig. 3. Performance, on the Mnaimneh et al. [20] dataset, 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) # Rules Discovered

Fig. 4. Performance, on the Spellman et al. [9] dataset, of RERII with various supports, and MAXCONF as confidence is increased.The key in (b) is also for use in (a).

B. 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 Fig. 2(a), 3(a) and 4(a). Intuitively, with respect to support pruning, a higher *minsup* results in more pruning and thus the run time 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 datasets (Fig. 2(a) and 3(a)). RERII only outperforms MAXCONF on the Spellman et al. [9] dataset when *minsup* is increased to 40% (Fig. 4(a)). However, as shown in Fig. 4(b), 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.

C. 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 datasets 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 non-corresponding rule does not imply an incorrect relationship. This is because the mined relationships may not have been hypothesised yet. In fact, biologists often perform microarray experiments to formulate new hypotheses from unknown and/or unexpected gene relationships.

Our evaluation proceeds as follows. Firstly 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*, presenting some sample rules identified by MAXCONF that correctly describe gene relationships in this system. From our analysis we confirm the

appropriateness of MAXCONF to mine various types of gene relationships. For brevity, we only discuss our results regarding the Hughes et al. [19] dataset.

1) Direct Interactions: The Biomolecular Interaction Network Database (BIND) [7] is an online archive of pairwise information about *direct interactions* (DI) which can occur between two biological entities. We have used BIND to analyse the biological relevance of the rules we identify, based on precision and recall for direct interactions.

Using BIND we determined the percentage of rules mined with MAXCONF that exhibit DI between at least two of their items i.e. *precision*. This is based on the rationale that for a direct interaction to occur between two or more genes/proteins, it is highly probable their expressions are correlated and thus they are likely to be present together in at least one rule. Furthermore, we analysed the effectiveness of our approach to identify all possible DIs from the dataset, i.e. *recall*. For example, we say that the rule $X \Rightarrow YZ$ describes a DI if their is a known DI in BIND between X and Y or X and Z.

To calculate the precision and recall of DIs we first need to determine which DIs are actually possible within the microarray dataset we analysed. 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 association rules and $\mathcal{B}$ be the set of pairwise DIs in the microarray dataset, in the form of rules. The *precision* of DIs in $\mathcal{R}$ is:

$$\text{Precision} = \frac{\text{\# rules in } \mathcal{R} \cap \mathcal{B}}{\text{\# rules in } \mathcal{R}} \tag{12}$$

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

$$\text{Recall} = \frac{\text{\# rules in } \mathcal{R} \cap \mathcal{B}}{\text{\# rules in } \mathcal{B}} \tag{13}$$

The recall of a system is the percentage of possible DIs which 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 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 summarised in Table III. MAXCONF is clearly more effective than the support-based methods. The significant improvement from Recall 1 and Recall

TABLE III

BIOLOGICAL RULE ANALYSIS FOR RERII AND MAXCONF ON THE Hughes *et al.* [19] DATASET

Algorithm	Supp. (%)	Conf. (%)	# Rules	Bind Analysis			Gene Ontology	
				Precision (%)	Recall 1 (%)	Recall 2 (%)	# Rules	% Rules
RERII	0	80	–	–	–	–	–	–
	1.0	80	15669	0.3	0.9	1.5	11548	73.7
	2.5	80	12231	0.1	0.3	1.2	9136	74.7
	5.0	80	8083	0.5	0.1	0.4	6102	75.5
	7.5	80	5223	0.7	0.1	0.1	4251	81.4
	10.0	80	2123	1.9	0.1	0.1	1619	76.3
MAXCONF	NA	80	19090	1.2	26.3	94.0	14298	74.9
	NA	85	12424	1.6	26.1	93.0	9727	78.3
	NA	90	8296	2.3	25.9	86.6	6860	82.7
	NA	95	7214	2.6	25.7	83.0	5980	82.9
	NA	100	7076	2.7	25.7	81.9	5866	82.9

TABLE IV

ASSOCIATION RULES EXTRACTED USING MAXCONF ON THE Hughes *et al.* [19] DATASET

#	Association Rule	Supp. (%)	Conf. (%)
1	FMP17 $\Rightarrow$ ERG28, ERG25	0.60	100
2	CTF13 $\Rightarrow$ SNO1, SNZ1	21.0	80.8
3	CSE1 $\Rightarrow$ CRM1, PCL5	0.33	100
4	EUG1 $\Rightarrow$ BNA2, GCS2, PDH1, TFS1, THI5, THI11, THI13, YGR043c, YML131w	1.30	100
5	SIL1 $\Rightarrow$ AFR1, GCS2, YPS1, YOR289w	2.67	100
6	FRE6 $\Rightarrow$ SIT1, ARN1, ARN2, ENB1, FIT2, FIT3	4.33	100
7	AKR1 $\Rightarrow$ CCC2, SIT1, FTR1, ARN1, ARN2, FET3, ENB1, FIT2, FIT3	3.33	90
8	$\overline{\text{MAC1}}$ $\Rightarrow$ $\overline{\text{FRE7}}$	0.33	100
9	MEP2 $\Rightarrow$ GLK1, GLC3, DMC1, HSP12, PRY1, NCA3, TFS1, MSC1, PGM2, YGP1	1.00	100
10	$\overline{\text{ESC8}}$ $\Rightarrow$ $\overline{\text{IMD1}}$, $\overline{\text{IMD2}}$	1.30	100

2 is expected as more relationships within the rules are considered. The high recall (94%) obtained by MAXCONF is superior compared with using RERII (1.5%) with *minconf* = 80% (and *minsup* = 1% 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% of the genes were immediately pruned based on support during preprocessing.

Rules 1, 2 and 3 in Table IV 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% confidence and 0.33% support, correctly describes the relationships between all three genes (CSE1 binds PCL5, which in turn PCL5 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 database. Furthermore, not all gene relationships we identify can convey DIs. For examples, rules 8 and 10 in Table IV 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 hypothesise that our low precision is an indication that we are identifying other possible relationships which are not documented DIs. This forms the basis of our next evaluation scheme using the Gene Ontology, to investigate how biologically relevant our rules are with respect to other gene relationships.

2) Gene Ontology: In this section we evaluate the rules based on the the Gene Ontology (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 over-represented 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 which 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 the GO annotations between the items represented more specific gene relationships.

These results are summarised in the last two columns of Table III. The first of these columns shows the number of rules identified by each system which contain a GO relationship. The second column is the percentage of rules. We chose to show both 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% confidence. Although a high percentage of rules mined by RERII, with *minsup* = 7.5, contained a GO relationship (81.4%), the raw number of rules was significantly less than those mined by MAXCONF. Furthermore, as *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 V shows the GO annotation break down of three example rules (rules 4, 9 and 10) from Table IV. 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 *meotic recombination*. This GO term has a depth of 9 within the GO, and the genes DMC1 and MSC1 are statistically over-represented 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 hypothesise 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 analysed in other experiments. Thus, MAXCONF may be effectively used as a discovery tool for formulating new hypotheses from microarray experiments.

TABLE V

SAMPLE RULES WITH GO INFORMATION

Rule #	Gene Ontology Cluster Information				
	Gene set		GO term	Depth	P-value
4	1	THI5, BNA2, THI13, THI11	water soluble vitamin biosynthesis	7	8.93e-06
	2	BNA2, GSC2, THI5, THI13, THI11	cellular biosynthesis	5	0.067
	3	THI5, EUG1, BNA2, TFS1, GSC2, THI13, THI11, PDH1	cellular metabolism	4	0.2
9	1	DMC1, MSC1	meotic recombination	9	0.0475
	2	GLK1, GLC3, PGM2	carbohydrate metabolism	5	0.0475
	4	HSP12, MEP2	plasma membrane	4	0.172
10	1	ESC8, IMD2	nuclear acid metabolism	5	0.02
	2	IMD1	unknown	–	–

3) Iron Uptake Pathway: In this section we further demonstrate the usefulness of MAXCONF for extracting correct gene relationships by examining the *S.cerevisiae* iron uptake systems. *S.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 co-expressed. Another sub-system 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 IV (rules 6, 7 and 8). In particular, rule 8 indicates the strength of MAXCONF for analysing 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 *minsup* was set to a very low value). Furthermore, although this rule cannot exhibit a direct interaction, it is of biological significance. The gene MAC1 was selectively mutated in the Hughes *et al.* [19] dataset, 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.

VI. CONCLUSIONS

In this paper, we introduced the first truly scalable approach for discovering gene relationships from microarray data. Traditional data mining methods, which are optimised for sparse datasets,

are impractical for analysing microarray data. Recently, row-enumeration rule mining algorithms have been developed to facilitate mining in dense datasets. 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 which exploits two new confidence pruning techniques, and restricts the rule discovery to maximal confident rules.

We performed experiments on three microarray datasets 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 which 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 database shows that with a *minconf* of 80%, we are able to achieve a recall of 94% for extracting known direct interactions. This is superior compared with using a support-based method, where with a low *minsup* of 1% only 1.5% of direct interactions 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.

ACKNOWLEDGEMENTS

This research was partially funded by the 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 BIOKDD 2005 [25].

REFERENCES

[1] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model." *Theoretical Computer Science*, vol. 298, pp. 235–251, 2003.

[2] T. Akutsu, S. Miyano, and S. Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways." *Bioinformatics*, vol. 16, no. 8, pp. 727–734, 2000.

[3] C. Creighton and S. Hanash, "Mining gene expression databases for association rules." *Bioinformatics*, vol. 19, no. 1, pp. 79–86, 2003.

[4] G. Cong, K.-L. Tan, A. Tung, and F. Pan, "Mining frequent closed patterns in microarray data." in *Fourth IEEE Int'l Conf. on Data Mining (ICDM)*, vol. 4, pp. 363–366, 2004.

[5] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the boolean network model." in *Pacific Symposium on Biocomputing*, vol. 4, pp. 17–28, 1999.

[6] F. Pan, G. Cong, K. Tung, J. Yang, and M. Zaki, "CARPENTER: Finding closed patterns in long biological datasets.", in *ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 637–642, 2003.

[7] C Alfarano et.al., "The Biomolecular Interaction Network Database and Related Tools 2005 update." *Nucleic Acids Res*, vol. 33, pp. D418–24, 2005.

[8] The Gene Ontology Consortium, "The Gene Ontology (GO) database and informatics resource." *Nucleic Acids Res*, vol. 32, pp. D258–D261, 2004.

[9] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K.Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization." *Molecular Biology of the Cell*, vol. 9, pp. 3273–3297, 1998.

[10] D. Singh et al., "Gene expression correlates of clinical prostate cancer behavior." *Cancer Cell*, vol. 1, pp. 203–209, 2002.

[11] A. Gasch, P. Spellman, C. Kao, O. Carmel-Harel, M. Eisen, G. Storz, D. Botstein, and P. Brown, "Genomic expression changes in the response of yeast cells to environmental changes." *Mol Biol Cell*, vol. 11, no. 12, pp. 4241–4257, 2000.

[12] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: a survey." *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 11, pp. 1370–1386, Nov 2004.

[13] G. Cong, K.-L. Tan, A. K. Tung, and X. Xu, "Mining TOP-K covering rule groups for gene expression data." in *ACM SIGMOD Int'l Conf. on Management of data*, pp. 670–681, 2005.

[14] R. Agrawal, T. Imielinksi, and A. N. Swami, "Mining association rules between sets of items in large databases." in *ACM SIGMOD Int'l Conf. on Management of Data*, pp. 207–216, 1993.

[15] G. Cong, A. Tung, X. Xu, F. Pan, and J. Yang, "FARMER: Finding interesting rule groups in microarray datasets." in *ACM SIGMOD Int'l Conf. on Management of Data*, pp. 143–154, 2004.

[16] M. Zaki and C. Hsiao, "CHARM: An efficient algorithm for closed association rule mining." in *SIAM Int'l Conf. on Data Mining (SDM)*, pp. 457–473, 2002.

[17] J. Pei, J. Han, and R. Mao, "CLOSET: An efficient algorithm for mining frequent closed itemsets." in *ACM SIGMOD Int'l Workshop on Data Mining and Knowledge Discovery (DMKD '00)*, pp. 21–30, 2000.

[18] Y. Huang, H. Xiong, S. Shekhar, and J. Pei, "Mining confident co-location rules without a support threshold." in *18th ACM Symposium on Applied Computing (ACM SAC)*, pp. 407–501, 2003.

[19] T. Hughes et al., "Functional discovery via a compendium of expression profiles." *Cell*, vol. 102, pp. 109–126, 2000.

[20] S. Mnaimneh et al., "Exploration of essential gene functions via titratable promoter alleles." *Cell*, vol. 118, pp. 31–44, 2004.

[21] T. Beissbarth and T. Speed, "GOstat: Find statistically overrepresented gene ontologies within gene groups." *Bioinformatics*, vol. 20, no. 9, pp. 1464–1465, 2004.

[22] R. Hassett, A. Romeo, and D. Kosman, "Regulation of high affi nity iron uptake in the yeast Saccharomyces cerevisiae." *J Biol Chem*, vol. 273, no. 13, pp. 7628–7636, 1998.

[23] V. Haurie, H. Boucherie, and F. Sagliocco, "The Snf1 protein kinase controls the induction of genes of the iron uptake pathway at the diauxic shift in Saccharomyces cerevisiae." *J Biol Chem*, vol. 278, no. 46, pp. 45 391–6, 2003.

[24] L. Martins, L. Jensen, J. Simon, G. Keller, and D. Winge, "Metalloregulation of FRE1 and FRE2 homologs in Saccharomyces cerevisiae." *J Biol Chem*, vol. 273, no. 37, pp. 23 716–23 721, 1998.

[25] T. McIntosh and S. Chawla, "On discovery of maximal confi dent rules without support pruning in microarray data." in *5th ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD '05)*, 2005, pp. 37–45.

Tara McIntosh received the BS degree in Bioinformatics from the University of Sydney, Australia in 2005. In March 2006 Tara started a 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 Scientifi c 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 including ACM SIGKDD, IEEE ICDM and SIAM International Conference on Data Mining (SDM). His paper on "Mining for Outliers in Sequential Databases" received the best application paper award in the SDM, 2006. He is also the research leader of the data mining program of the Capital Markets CRC. He is a co-author on 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.