Differential Association Rule Mining for the Study of Protein-Protein Interaction Networks

Christopher Besemann[*]
Computer Science Dept
North Dakota State University
Fargo, North Dakota 58105
christopher.besemann

Anne Denton
Computer Science Dept
North Dakota State University
Fargo, North Dakota 58105
anne.denton

Ajay Yekkirala
Biology Dept
North Dakota State University
Fargo, North Dakota 58105
ajay.yekkirala

ABSTRACT

Protein-protein interactions are of great interest to biologists. A variety of high-throughput techniques have been devised, each of which leads to a separate definition of an interaction network. The concept of differential association rule mining is introduced to study the annotations of proteins in the context of one or more interaction networks. Differences among items across edges of a network are explicitly targeted. As a second step we identify differences between networks that are separately defined on the same set of nodes. The technique of differential association rule mining is applied to the comparison of protein annotations within an interaction network and between different interaction networks. In both cases we were able to find rules that explain known properties of protein interaction networks as well as rules that show promise for advanced study.

General Terms

association rule mining, protein interactions, relational data mining, graph-based data mining

1. INTRODUCTION

Association Rule Mining (ARM) is a popular technique for the discovery of frequent patterns within item sets [1; 2; 13]. The technique has been generalized to the relational setting [18; 10; 22] including the study of annotations of proteins within a protein-protein interaction network [22]. In many bioinformatics problems, biologists are interested in comparing different sets of items. Rather than identifying patterns among protein annotations, biologists often want to contrast annotations of interacting proteins [25]. Going one step further, is also a want to contrast different network definitions to understand which experimental technique to use for which purpose.

Several definitions of protein-protein interactions have been introduced. For our study we concentrate on three: Physical interactions are determined through experiments such as the yeast-two-hybrid method [16; 30] and indicate a level of biochemical interaction. Genetic interactions are derived from in-vivo experiments in which the lethality associated with mutation of two genes is tested [26]. Domain-fusion interactions are detected in silico by comparing different species [19; 28]. Two genes in one species are labeled as interacting if they have homologs in another species and those homologs are exons of the same gene. Previous approaches to network comparison have studied each network in isolation and have compared statistics between networks [25; 27]. We use differential association rule mining techniques to identify rules that directly contrast the differences in annotations across interactions, and between different types of interactions.

Can differences be identified from standard ARM output? Assume, for example, that proteins with ”transcription” as annotation are found to frequently interact with proteins that are localized in the ”nucleus”. This rule may be due to two independent rules, one that associates ”transcription” and ”nucleus” within a single protein, and others that represent a correlation of ”transcription” and/or ”nucleus” between interacting proteins. We would not consider this a sign of a difference between interacting proteins. The same type of rule could, however, indeed stand for a difference. Consider the rule that proteins in the ”nucleus” are found to interact with proteins in the ”mitochondria”. It can be expected that a single protein would not simultaneously be located in the ”nucleus” and in the ”mitochondria”. We can therefore assume that the rule highlights a difference between interacting proteins and may identify an instance of compartmental crosstalk. This rule is significantly more interesting to a biologist than the rule relating ”nucleus” and ”transcription”. It is much more expressive of the properties of the respective interaction network.

So far we have distinguished between the two examples on the basis of our biological background knowledge. Two approaches could be taken to translate the idea into a useful ARM algorithm. We could devise a difference criterion involving correlations between neighboring nodes and/or rules found within individual nodes. Such an approach would not benefit from any of the pruning that has made ARM an efficient and popular technique. Our algorithm takes an approach that makes significant use of pruning: Only those items are considered for the ARM algorithm for which each item in a set is unique to only one of the interacting nodes. The rule associating ”transcription” and ”nucleus” would thereby only be evaluated on those ”transcription” proteins that are not themselves in the ”nucleus”, and those ”nucleus” proteins, that are not themselves involved in ”transcription”.

There are other reasons why a focus on differences is more effective for association rule mining in networks than a stan-

[*]Authors’ email: @ndsu.nodak.edu

dard application of ARM on joined relations. Traditionally association rule mining is performed on sets of items with no known correlations. Interacting proteins are, however, known to often have matching annotations [27]. Using association rule mining on such data, in which items are expected to be correlated may lead to output in which the known correlations dominate all other observations either directly or indirectly. This problem has been observed when relational association rule mining is directly applied to protein networks [22; 4]. Excluding matching items of interacting proteins is therefore commonly advisable in the interest of getting meaningful results alone [4]. Matching annotations can be studied by simple correlation analysis, in which co-occurrence of an annotation in interacting proteins is tested. In the presence of such correlations, association rules are likely to reflect nothing but similarities between interacting proteins.

We use the concept of including only items that are unique to one of a set of interacting nodes to further address the task of comparing different interaction networks. In principle networks can be compared by studying each individually and comparing the results. When applying association rule mining to annotations in protein interaction networks, such an approach faces two difficulties. First, not all biological experiments have been done on all proteins. It is, therefore, safest to base a comparison of two networks only on proteins that show both types of interaction. Second, association rule mining gains its computational efficiency from item set pruning. Any test that is done at a later time removes rules that were produced unnecessarily. If the selection process can be converted to act on item sets themselves, pruning is restored. We demonstrate how the concept of unique items can be used to extract differences between networks.

2. DIFFERENTIAL ASSOCIATION RULES

We assume a relational framework to discuss differences within and between networks. The concept of a network may suggest use of graph-based techniques. Graph-theory typically assumes that nodes and edges have at most one label. Relational algebra on the other hand has the tools for the manipulation of data associated with nodes and edges. A relational representation of a graph with one type of nodes requires one relation for data associated with nodes, which we will call node relation, and a second relation that describes the reflexive relationship between nodes, the edge relation. To compare networks we will use multiple edge relations. Association rule mining is commonly defined and implemented over sets of items. We combine the concept of sets with the relational algebra framework by choosing an extended relational model similar to [13] . Attributes within this model are allowed to be set-valued, thereby violating first normal form. We go one step further by allowing sets of tuples, i.e. relations themselves, as attribute values. Consider a database with node relations $R_N(T, D)$ where T is a tuple identifier and D is a set of descriptors. Tuples in R_N have the form $< t_i, D_i >$ where D_i is a relation of descriptors $< d_j >$ (see Table 2 for representation). Descriptors are tuples with just one attribute of domain $\mathcal{D}$. We call the $< d_j >$ descriptors to distinguish them from items. Items have a second attribute to identify their node of origin, see definition (3). We will call the sets of items that form the basis for association rule mining *basis sets*.

Table 2: Node

ORF	Annotations
YPR184W	$\{< \mathit{cytoplasm} >\}$
YER146W	$\{< \mathit{cytoplasm} >\}$
YNL287W	$\{< \mathit{SensitivityTOaaaod} >\}$
YBL026W	$\{< \mathit{transcription} >, < \mathit{nucleus} >\}$
YMR207C	$\{< \mathit{nucleus} >\}$

Table 3: Edge

ORF0	ORF1
YPR184W	YER146W
YNL287W	YBL026W
YBL026W	YMR207C

Definition 1. A *single-node basis set* is identical to a set of descriptors $D_i \subseteq \mathcal{D}$. This definition is equivalent to the basic definition of an item set used in association rule mining [1].

Our goal is to mine relational basis sets that will be constructed from multiple descriptor sets that belong to the same tuple of a joined relation. An edge relation has two attributes $R_E(T_l, T_r)$, with T_l as well as T_r being foreign keys that refer to identifiers in one or more node relations (see Table 3 for representation). Edge relations can, in principle, have the alternate form $R_E(T_l, T_r, D^{(E)})$ with $D^{(E)}$ being a set of edge descriptors. We could split such a relation into a separate node relation as well as a standard edge relation as in [7].

Joined-relation basis sets are formed in multiple steps. Edge and node relations are joined through a natural join operation ($*$). Attribute names are changed [11] such that they are unique. We use this step to ensure that information about the origin of different attributes is maintained. Attributes are identified by consecutive integers to which we will refer as origin identifiers $g \in \mathcal{G} = \{0, ..., (n-1)\}$ where n is the number of node relations. This information will be used in a later step to actually modify the descriptors according to their origin before joined-relation basis sets are constructed from multiple descriptor sets.

Definition 2. A *joined-relation basis set* is derived through the following steps. A 2-node joined-relation is created by

$$\begin{aligned} R_{2N} \;\leftarrow\; & \rho_{0.T,0.D}(R_N(T,D)) * \rho_{0.T,1.T}(R_E(T_l,T_r)) \\ & *\rho_{1.T,1.D}(R_N(T,D)). \qquad (1) \end{aligned}$$

Generalization to n-node joined-relations is straight forward. Note, however that we can have multiple alternatives. For a 4-node joined-relation we can have

$$\begin{aligned} R_{4Nl} \;\leftarrow\; & \rho_{0.T,0.D}(R_N(T,D)) * \rho_{0.T,1.T}(R_E(T_l,T_r)) \\ & *\rho_{1.T,1.D}(R_N(T,D)) * \rho_{1.T,2.T}(R_E(T_l,T_r)) \\ & *\rho_{2.T,2.D}(R_N(T,D)) * \rho_{2.T,3.T}(R_E(T_l,T_r)) \\ & *\rho_{3.T,3.D}(R_N(T,D)) \qquad (2) \\ R_{4Ng} \;\leftarrow\; & \rho_{0.T,0.D}(R_N(T,D)) * \rho_{0.T,1.T}(R_E(T_l,T_r)) \\ & *\rho_{1.T,1.D}(R_N(T,D)) * \rho_{1.T,2.T}(R_E(T_l,T_r)) \\ & *\rho_{2.T,2.D}(R_N(T,D)) * \rho_{1.T,3.T}(R_E(T_l,T_r)) \\ & *\rho_{3.T,3.D}(R_N(T,D)). \qquad (3) \end{aligned}$$

Table 1: Join and Unique

TID	Join	
1	$\{< 0, cytoplasm >\}$	$\{< 1, cytoplasm >\}$
2	$\{< 0, SensitivityTOaaaod >\}$	$\{< 1, transcription >, < 1, nucleus >\}$
3	$\{< 0, transcription >, < 0, nucleus >\}$	$\{< 1, nucleus >\}$
TID	Unique	
1	NULL	NULL
2	$\{< 0, SensitivityTOaaaod >\}$	$\{< 1, transcription >, < 1, nucleus >\}$
3	$\{< 0, transcription >\}$	NULL

Notice that in equation (2) the joining corresponds to a chain of 0-1-2-3 and in equation (3) there is a branch 1-2 and 1-3. Attribute renaming $\rho_{A_0 \ldots A_n}$ is used as defined in [11]. We then apply a Cartesian product of a relation consisting of a single tuple containing the origin identifier $< g >$ with each descriptor set individually. It converts the descriptors d_j into tuples $< g, d_j >$. g is the same origin identifier that is used as prefix in the attribute name

$$\begin{aligned} g.I_i &= \; < g > \times \{< d_0 >, ..., < d_k >\} \\ &= \; \{< g, d_0 >, ..., < g, d_k >\}. \end{aligned} \tag{4}$$

Definition 3. An *item* is defined as a tuple $< g{,}d_j >$ where g is an integer which is the origin identifier and d_j is the descriptor value of an attribute.

Note that we will use an abbreviated notation for items in the results section ($g.d_j$ instead of $< g, d_j >$). A joined-relation basis set B_i is derived as the union of descriptor sets for each tuple identified by t_i of the joined relation. For a 2-node joined-relation basis set or 2-node basis set we have

$$\forall t_i \;\; B_i = 0.I_i \cup 1.I_i. \tag{5}$$

The set of all basis sets is $C = \{B_0, ..., B_m\}$ where m is the number of tuples in the joined relation an example of the product can be seen in Table (1 Join) as the result of the operations to Tables (2 and 3).

Definition 4. A *uniqueness operator* U is defined as follows. For each set-valued attribute on which it operates the set difference is computed between that attribute and the union of all other attributes of that domain.

$$U(R_{nN}(t_i, \{0.I, ..., (n-1).I\})) :$$
$$\forall t_i \;\; \forall_{j=0}^{(n-1)} \; j.I_i^U = j.I_i - \bigcup_{k=0,k\neq j}^{(n-1)} k.I_i \tag{6}$$

with $g.I_i$ defined as in equation (4).

Table (1 Unique) shows the results of the unique operation on the joined portion. In this paper the uniqueness operator is applied to all set-valued attributes of a joined-relation but other choices are possible, such as requiring uniqueness only across a subset of edges.

Definition 5. A *unique item basis set* is defined through the following steps. An n-node joined-relation is created as described in definition (2). The uniqueness operator is applied to all set-valued attributes. Then the Cartesian product is used to create item tuples, and the process continues as for joined-relation basis sets.

Definition 6. A *network comparison basis set* differs from a unique node item basis set through the use of different edge relations. In the current paper we limit ourselves to 3-node network comparison basis sets. We only consider those edges that are unique to one of the network definitions. Edges that are represented in both networks are removed since they cannot give us information on differences between networks.

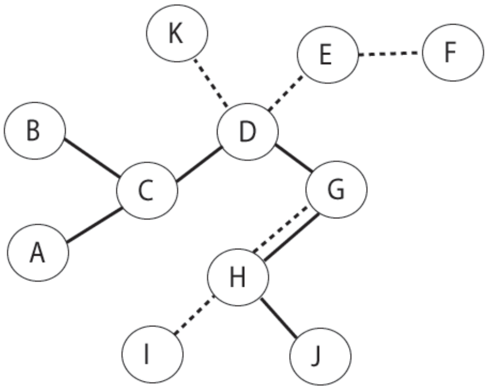
$$\begin{aligned} R_{3NC} \;\; \leftarrow \;\; & \rho_{0.T,0.D}(R_N(T,D)) * \rho_{0.T,1.T}(R_{E1}(T_l,T_r)) \\ & * \rho_{1.T,1.D}(R_N(T,D)) * \rho_{1.T,2.T}(R_{E2}(T_l,T_r)) \\ & * \rho_{2.T,2.D}(R_N(T,D)) \end{aligned} \tag{7}$$

The other steps are done as for unique node item basis sets. The uniqueness operator is applied to all nodes. That means that if an item exists on node 2 which interacts with node 1 through $E1$, and on node 1 as well, it will not be considered for network comparison basis set. Rules that we may observe between node 0 and 1 will strictly relate to interaction $E1$ between those nodes and not to interaction $E2$ between node 1 and 2. We limit the scope of our algorithm to rules that involve only one of the networks as definition (9). Any such rule will automatically represent a property that is in contrast to the other network. Compare Figure (1) for a graphical representation of the extraction of a network comparison basis set.

Definition 7. Given the above definitions of basis sets, association rules are defined in their standard way. A rule has the form $X \rightarrow Y$ where X and Y are sets of items (see definition 3). The *support* of a rule is the probability $P(X \cup Y)$ within the set of all basis sets C. The *confidence* of a rule is the conditional probability $P(Y|X)$. The set of all items in the rule is an item set $I = X \cup Y$.

It is important to understand that any relational association rule depends on the context in which it was generated. A rule that involves only two nodes related by one edge can, in principle, be found in a 2-node join-relation and any higher order relation. The support and confidence will however vary depending on that context, and a rule that is strong in one context may not be so in another. We follow [7] in always using the lowest order possible. For network comparison purposes we need three entities to derive 2-node rules. See definition (6). The problems associated with multiple contexts leads us to the following definitions.

Definition 8. An item set J is *out-of-scope* if one or more nodes are not represented, i.e., if $|\pi_G(J)| < n$ where $||$ indicates the cardinality, π is the relational projection operation, G is the identifier attribute of the item tuples, and n is the number of node relations that were joined. In Table (1 Unique) item sets for TID 1 and 3 are considered out-of-scope on the transaction level.



Transactions		
C	D	E
C	D	K
G	D	E
G	D	K
J	H	I

Figure 1: Left: Two graphs defined over the same set of nodes, Right: Network comparison basis set

Definition 9. An item set J has *network comparison scope* if it represents all nodes that are related through one edge relation and no nodes that are related through a different edge relation. If the item set is furthermore unique, support and confidence based on this item set will reflect network properties that are specific to one type of network and not to any other network involved in the comparison.

Definition 10. An item set J is *repetitious* if at least one descriptor occurs more than once, i.e., if $|\pi_D(I)| < |J|$ where π_D is the projection on the descriptor attribute. Two items are considered *repetitious* if they belong to the same joined-relation basis set, their origin identifier differs, and their descriptors are equal. Table (1 Join) item sets for TID 1 and 3 have repetitious items.

3. RELATED WORK

Oyama et al. [22] apply association rule mining to joined-relations of physical protein interactions and their annotations. This work notes the problem of what we term repetitious item sets but does not resolve it. Relational association rule mining has more generally been addressed in the context of inductive logic programming [10; 18; 17]. These approaches are very flexible and leave most choices up to the user. This paper, on the other hand, addresses the question of what specifications allow extracting meaningful rules. It is useful to notice that the major portions of differential rule mining can be imported to different frameworks including ILP.

Some biological publications have touched on the concept of comparing networks. The authors in [27] address aspects such as density of the networks and how well the genetic interactions predict physical interactions. Another work [23] looks at correlation and interdependency characteristics between the genetic and physical networks. The distribution of annotations on an individual network is discussed in [25]. These approaches fall short of contrasting annotations in different networks. A further related research area is graph-based ARM [15; 21; 31; 6]. Graph-based ARM does not typically consider more than one label on each node or edge. The goal of graph-based ARM is to find frequent substructures in that setting.

Removal of a class of redundant rules is an important part of differential rule mining. Redundant rules have been studied, and closed sets [8; 33] have proven a successful approach to their elimination. Closed sets alone do not, however, address the problem of contrasting different nodes or networks. Since we know what kinds of rules we want to eliminate, it is significantly more efficient to do so at the relational join level. This strategy has the added benefit of correcting support and confidence of all rules to reflect only the contribution that is non-redundant to a combination of repetitious and out-of-scope item sets.

There are other areas of research on ARM in which related transactions are mined in some combined fashion. Sequential pattern or episode mining [2; 32; 24; 34] and inter-transaction mining [29] are two main categories. Some similarities in the formalism can be observed since we are also interested in mining across what can be considered transactions. A tuple in a joined-relation can ultimately be compared with sequences of transactions. Overall the goals of these approaches are too different to be applicable to our setting in any direct way.

4. IMPLEMENTATION

The differential association rule mining algorithm was implemented in a modular fashion. Three major parts are distinguished. Preprocessing (steps 1.-3.) includes application of the uniqueness operator U (see definition 4 in section 2). The actual item set generation (step 4.) is done based on sets of items that appear as regular sets to the ARM program. Results in this paper use the Apriori algorithm from Christian Borgelt [5]. Postprocessing (steps 5.,6.) does additional filtering at the item set and rule level.

Preprocessing includes the following tasks. For undirected graphs only one direction is typically included in data sets. We create both directions to ensure correct representation and then join the relations. Joined relations were created with different methods depending on the comparison type for input.

The uniqueness operator, U, from equation (6) was applied to all basis set relations (step 8.). If the operator U has removed all items related to any one of the entities the basis set is marked as deleted (steps 9.,10.). Such basis sets can never contribute to in-scope item sets or rules. The basis set is therefore not passed to the ARM method. We do, however, calculate support and confidence based on the full set of joined table basis sets by counting all basis sets. Once the basis sets are processed into the unique basis sets, standard

Figure 2: Differential ARM Algorithm

```
Number of nodes in the join relation: $n$
n-entity joined relation basis set: $B_i$
Set of basis sets $C$:$\{B_0,...,B_m\}$

**Diff-ARM**($n$,$minconf$,$minsup$,$C$)
1. For undirected graphs represent each direction
2. Join relations and eliminate cycles
3. $C^U$=U_OP($n$,$C$)
4. FreqSets=Apriori:FreqItemset_Gen($C^U$,$minsup$)
5. For undirected graphs remove symmetric contributions
6. U_SCOPERULE($FreqSet$,$n$,$minconf$)

**U_OP**($n$,$C$) Returns→ $C^U$
7. foreach transaction, $B_i \in C$
8.    $B_i^U = U(B_i(\{0.I_i, ..., (n-1).I_i\}))$
9.    foreach $j.I_i^U \in B_i^U$
10.        if($j.I_i^U == \emptyset$) → mark tuple as deleted
11.  $C^U + = B^U$

**U_SCOPERULE**($FreqSet$, $n$, $minconf$)
12. foreach $J_i \in FreqSet$
13.  if($|\pi_G(J_i)| == n$ )
14.        Apriori:Rule_Gen($J_i$,$minconf$)
15. Apply rule filtering
```

Apriori is applied (step 4.).

Frequent item sets or closed item sets are returned as the usual result of Apriori. For undirected graphs symmetric versions of each item set are returned and have to be removed (step 5.). Input from Apriori is sent to the rule generation phase (step 6.). Item sets are tested if all entities are represented (step 13.). If not, the item set is removed as being out-of-scope. Rules are then produced as in standard ARM by processing the frequent item sets (step 14.). The algorithm concludes with a set of rules that satisfy the requirements from section 2. Rule results are additionally filtered so that any node does not have items in both the antecedent and the consequent of the rule after the final set (step 15.). The following equation defines this step for a given rule A→C:

$$\pi_G(A) \cap \pi_G(C) == \emptyset \tag{8}$$

4.1 Data sets

Our data consist of one node relation gathered from the Comprehensive Yeast Genome Database at MIPS [20; 9], gene_orf. The gene_orf node relation represents gene annotation data. Annotations are hierarchically structured, with hierarchies for function, localization, protein class, complex, enzyme commission, phenotype and motif. In any category, attributes are multi-valued and we pick the highest level in each hierarchy as descriptors. The relation contains the ORF identifier as key and the set of annotations related to that ORF as attribute (descriptor set).

We used three different definitions for protein-protein interactions which are undirected edges for yeast: physical, genetic and domain fusion. The physical edge relation was built from the ppi table at CYGD [9] where all tuples with type label of ”physical” were used. The genetic edge relation was taken from supplemental table S1 of genetic interactions from [27] where both Synthetic Sick and Synthetic Lethal entries are used. Our third edge relation was the domain fusion set built from the unfiltered results posted from [28; 14]. The set was filtered to reflect only ORFs contained in our node relation.

4.2 Performance

Three contributions to the complexity have to be distinguished: preprocessing, Apriori and postprocessing. The most important contribution is the Apriori step. Since we did not modify the algorithm itself, changes in performance come from data reduction. The resulting improvement is highly significant. Figure (3) shows the processing time of the Apriori algorithm under a performance trial. Recorded is the time to generate frequent item sets for unique item basis sets of one to 4 nodes. We did not include time to load the database or print the rules. As seen, the differential ARM algorithm outperforms ARM by a factor of 100 in the 4-node setting. The reduction in the number of rules is even more significant. The difference between the number of rules in differential and standard ARM demonstrate how correlations dominate standard ARM output and thereby render it useless.

5. RESULTS

We will first look at an example of a rule that is strong based on the application of a standard ARM algorithm on joined tables but not so if only unique items are considered. A clear example is the rule mentioned in the introduction. Standard ARM on joined tables returns mostly rules that are repetitious or out-of-scope. We can look at a rule that is simple in meaning:

$$\begin{array}{lcl} \{0.\text{transcription}\} & \rightarrow & \{1.\text{nucleus}\} \\ \text{support} = 0.29\% & & \text{confidence} = 28.38\% \end{array} \tag{9}$$

This rule is a consequence of a strong single-node rule together with correlations that are documented by a repititious rule

$$\begin{array}{lcl} \{0.\text{transcription}\} & \rightarrow & \{0.\text{nucleus}\} \\ \text{support} = 0.70\% & & \text{confidence} = 69.59\% \end{array}$$

$$\begin{array}{lcl} \{0.\text{nucleus}\} & \rightarrow & \{1.\text{nucleus}\} \\ \text{support} = 5.74\% & & \text{confidence} = 29.02\% \end{array}$$

Using the uniqueness operator changes the support of rule (9) to 0.02% and a confidence of 2.08%. We expect support and confidence to be lower when the uniqueness operator is applied, since annotations are removed. Strong rules in our data set do, however, in general have a support around 2-4% and confidence around 20%. Based on these numbers the rule (9) cannot be considered strong and ranks much lower in the new results.

For the remainder of this section we will report differential association rules and no standard ARM results. The following rule was found to be strong in the physical interaction network

$$\begin{array}{lcl} \{1.\text{mitochondria}\} & \rightarrow & \{0.\text{cytoplasm}\} \\ \text{support} = 1.2\% & & \text{confidence} = 27.3\% \end{array}$$

This rule clearly corresponds to annotations that would not be expected to hold within a single protein but may hold

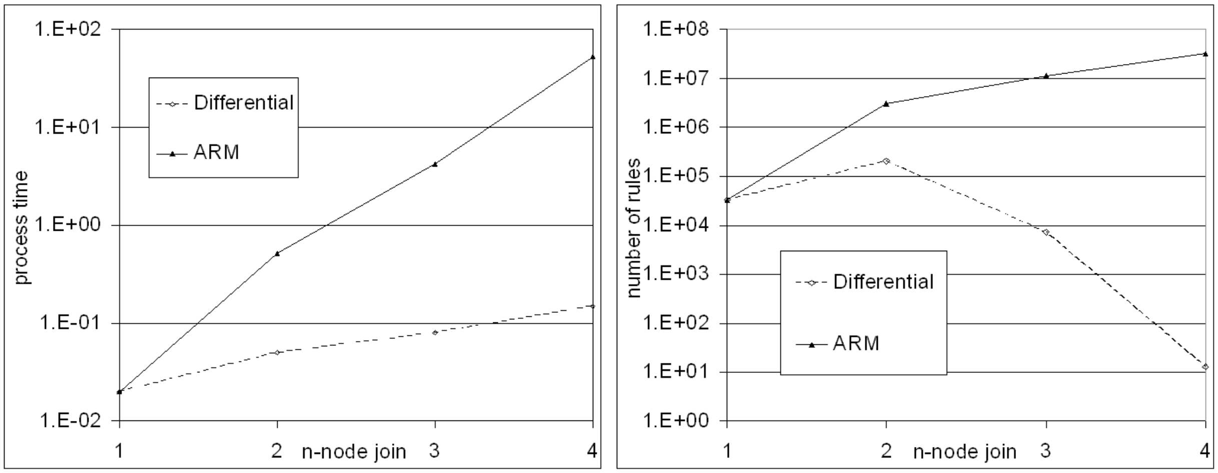


Figure 3: Left: Processing time, Right: Reduction in Number of Rules

between interacting ones. A protein located in the mitochondria would not have localization cytoplasm. We do, however expect compartmental crosstalk as studied in a paper by Schwikowski et al.[25] between those two locations. The observation confirms to us that we see rules that are sensible from a biological perspective. Comparison with [25] further helped us confirm some less expected rules such as

{1.mitochondria} → {0.nucleus}
support = 0.72% confidence = 16%.

We also found rules that have not yet been reported in the literature. The following rule was also observed within the physical interaction network

{1.ER} → {0.mitochondria}
support = 0.21% confidence = 6%

This rule was of interest particularly due to its comparatively high support. From a biological perspective one would not expect proteins in the endoplasmatic reticulum (ER) to physically interact with proteins in the mitochondria. To analyze the significance of the result we looked at some ORFs that support the rule. One pair was

(0.YLR423C: ER)
(1.YOR232W: mitochondria,
GrpE_protein_signature(PDOC00822),
Molecular_chaperones).

On further investigation it was found that GrpE along with a Molecular_chaperone is involved in protein import into the mitochondria [3]. This information leads to a hypothesis that YLR423C could be aiding the import mechanism or be interacting with the chaperone. This example demonstrates how differential association rules can provide insights into the functioning of the cell and can lead to further studies.

5.1 Differences Between Interaction Types

We will now look at rules that derive from the network comparison formalism of definitions (6) and (9) (inter-network comparison). Given multiple types of protein-protein interactions we look for significant differences to aid in the understanding of cellular function and as well as the properties and uses of the networks. In this paper we consider pairs of networks for inter-network comparisons (physical and genetic, physical and domain fusion, domain fusion and genetic) and join the two edge relations to form a network comparison joined relation (definition 6).

Table 4: Statistics

Table	int/orf	max int	#>20	#int
physical	3.55	289	73	14672
genetic	7.88	157	93	8336
domain fusion	44.6	231	305	28040

The networks do not show a significant overlap, i.e., it is very common that for any given physical interaction between two proteins there will be no genetic interaction [27]. Table 4 shows that even the statistical properties of the networks differ significantly: the average number of interactions of proteins that show at least one interaction varies from 3.55 in the physical network to 44.5 in the domain fusion network. Comparison of annotations across those networks has to compensate for such differences. The process of joining relations ensures that each protein that is considered for a physical interaction will also be considered for a genetic interaction.

Before looking at details of individual rules we will make some general observations regarding the number of rules we observed for different combinations of networks. When comparing physical and genetic networks we found about one order of magnitude more strong rules relating to the physical network compared with the genetic network. Physical interactions also produce the stronger rules when compared with domain fusion networks. That means that the physical network allows the most precise statements to be made. When comparing the domain fusion and the genetic network no major difference was found. That suggests that physical interactions reflect properties of the proteins better than either of the other two.

These rules are among the top 100 generated for the physical-domain fusion set. Some specific examples of interesting rules from this study are as follows:

{1.Fungal_Zn(PDOC00378)} →
{2.Zinc_finger_C2H2_type_domain(PDOC00028)}
support = 0.48% confidence = 76%

This rule was found to be supported in the domain fusion interaction set but not among the physical interactions. The motif of ORF 1 is a fungal Zinc-cysteine domain present in many transcription activator proteins which bind DNA in a zinc-dependent fashion. The motif of ORF 2 is a zinc finger which also binds DNA and commonly has cysteines and Histidine residues in them [12]. This rule tells us that the confidence of assuming a domain-fusion interaction between the fungal zinc domain and the zinc finger motif is 76%, not considering cases in which a zinc finger is also involved in a physical interaction. Further studies would be necessary to decide if the absence of a physical interaction is due to a problem with annotations or if those two proteins really do not interact. The second rule is supported by the physical network but not the domain fusion network

{0.ABC_trans_family_signature(PDOC00185)} →
{1.ATP/GTP_binding_site_motif_A(PDOC00017)}
support = 0.45% confidence = 90%

ORF 0 has the motif of an ABC transporter signature which implies it is an ABC transporter coding sequence. ABC transporters have conserved ATP binding domains as the motif in ORF 1 and help in either the import or export of molecules utilizing ATP as the energy molecule for the process [12]. From the rule we can see that these two domains physically interact but are never represented by a single gene. This supports the observation that the ATP binding domain is found in many other proteins as well [12] and both functions are combined through interactions at the protein level rather than at the genetic level. This observation would also warrant further studies.

6. CONCLUSIONS

We have described the novel concept of differential association rules. The goal of this technique is to highlight differences between items belonging to different interacting nodes or different networks. We demonstrate that such differences would not be identified by application of standard relational ARM techniques. Our technique is highly efficient and effective. It follows the ARM spirit by gaining its efficiency from a pruning step that is included even before the frequent item set generation step. We apply our framework to real examples of protein annotations and interactions. Results were able to confirm expected biological knowledge as well as identifying as yet unknown associations that were successfully supported by further inspection of the data. We have thereby provided a new tool that has potential for most network settings, and have demonstrated its successful application to bioinformatics.

7. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. #01322899. Additional thanks are expressed for valuable feedback from the anonymous reviewers of this paper.

8. ADDITIONAL AUTHORS

Ron Hutchison & Marc Anderson
Biology Department NDSU
email: `ron.hutchison & marc.Anderson @ndsu.nodak.edu`

9. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[3] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Studholme, C. Yeats, and S. R. Eddy. The pfam protein families database. *Nucleic Acids Research: Database Issue*, 32:D138–D141, 2004.

[4] C. Besemann and A. Denton. Unic: Unique item counts for association rule mining in relational data. Technical report, North Dakota State University, 6, 2004.

[5] C. Borgelt. Apriori. *http://fuzzy.cs.uni-magdeburg.de/˜borgelt/software.html*, accessed August 2003.

[6] D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[7] L. Cristofor and D. Simovici. Mining association rules in entity-relationship modeled databases. Technical report, University of Massachusetts Boston, 2001.

[8] L. Cristofor and D. Simovici. Generating an informative cover for association rules. In *Proceedings of International Conference on Data Mining*, Maebashi, Japan, 2002.

[9] CYGD. *http://mips.gsf.de/genre/proj/yeast/index.jsp*, accessed March 2004.

[10] L. Dehaspe and L. D. Raedt. Mining association rules in multiple relations. In *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297, pages 125–132, Prague, Czech Republic, 1997.

[11] Elmasri and Navathe. *Fundamentals of Database Systems*. Pearson, Boston, 4th edition, 2004.

[12] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. J. Sigrist, K. Hofmann, and A. Bairoch. The prosite database, its status in 2002. *Nucleic Acids Research*, 30:235–238, 2002.

[13] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, San Francisco, CA, 1995.

[14] O. C. I. Ikura Lab. Domain fusion database. *http://calcium.uhnres.utoronto.ca/pi/pub_pages/download/index.htm*, accessed March 2004.

[15] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 13–23, Lyon, France, 2000.

[16] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci U S A*, 98(8):4569–74, 2001.

[17] V. C. Jensen and N. Soparkar. Frequent itemset couting across multiple tables. In *Proceedings of PAKDD*, pages 49–61, 2000.

[18] A. J. Knobbe, H. Blockeel, A. Siebes, and D. M. G. van der Wallen. Multi-relational data mining. Technical Report INS-R9908, Maastricht University, 9, 1999.

[19] E. M. Marcotte, M. Pellegrini, H. L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–3, 1999.

[20] H. Mewes, D. Frishman, U. Gldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Mnsterkoetter, S. Rudd, and B. Weil. Mips: a database for genomes and protein sequences. *Nucleic Acids Research*, 30(1):31–44, 2002.

[21] K. Michihiro and G. Karypis. Frequent subgraph discovery. In *Proceedings of the International Conference on Data Mining*, pages 313–320, San Jose, California, 2001.

[22] T. Oyama, K. Kitano, K. Satou, and T. Ito. Extraction of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics*, 18(8):705–14, 2002.

[23] O. Ozier, N. Amin, and T. Ideker. Global architecture of genetic interactions on the protein network. *Nat Biotechnol*, 21(5):490–1, 2003.

[24] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan mining sequential patterns efficiently by prefix projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering*, pages 215–226, Heidelberg, Germany, 2001.

[25] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnol.*, 18(12):1242–3, 2000.

[26] A. H. Y. Tong, M. Evangelista, A. B. Parsons, H. Xu, G. D. Bader, N. Pag, M. Robinson, S. Raghibizadeh, C. W. V. Hogue, H. Bussey, B. Andrews, M. Tyers, and C. Boone. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294(5550):2364–8, 2001.

[27] A. H. Y. Tong, M. Evangelista, A. B. Parsons, H. Xu, G. D. Bader, N. Pag, M. Robinson, S. Raghibizadeh, C. W. V. Hogue, H. Bussey, B. Andrews, M. Tyers, and C. Boone. Global mapping of the yeast genetic interaction network. *Science*, 303(5695):808–815, 2004.

[28] K. Truong and M. Ikura. Domain fusion analysis by applying relational algebra to protein sequence and domain databases. *BMC Bioinformatics*, 4:16, 2003.

[29] A. K. H. Tung, H. Lu, J. Han, and L. Feng. Breaking the barrier of transactions: Mining inter-transaction association rules. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 1999.

[30] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403(6770):623–7, 2000.

[31] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the International Conference on Data Mining*, Maebashi City, Japan, 2002.

[32] X. Yan, J. Han, and R. Afshar. Clospan: Mining closed sequential patterns in large datasets. In *Proceedings 2003 SIAM Int.Conf. on Data Mining*, San Francisco, California, 2003.

[33] M. J. Zaki. Generating non-redundant association rules. In *Knowledge Discovery and Data Mining*, pages 34–43, Boston, MA, 2000.

[34] M. J. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning Journal*, 42:31–60, 2001.