RESEARCH ARTICLE OPEN ACCESS

Association Rule Mining for Gene Expression Data -A Neural Network Approach

Pallabi Das, Rafiqul Islam, Nirupam Saha, Sourish Mitra, Sayani Chandra
Department of Computer Science and Engineering
GNIT, Kolkata
India

ABSTRACT

A systematic approach for learning and extracting rule-based knowledge from gene expression data has become an important research area. Using computation techniques such as data mining to find the association relationship among these gene data is a challenging aspect. The aim of this paper is to get the set of genes which are responsible for the expression of another particular set of genes. As we are working with biological datasets the concern is solely depended on the type of data. To evaluate the dependency of the healthy or the diseased genes we have selected this association rule mining technique which is generally used in market baskets analysis. After getting the rules we compare the support and confidence of each rule. Instead of using traditional conditional probability approach neural network is used to take the decision which rule is strongly expressed. Depending upon the type of dataset we have plotted a graph to show that by tuning the activation function of the neural network we can get the fittest rule having minimum support and minimum confidence threshold.

Keywords:- Association rule, neural network, support, confidence, activation function.

I. INTRODUCTION

The 20th Century is frequently referred as the Century of Biology, given the huge developments of this scientific area that concluded that century with the great success of the Human Genome Project [1,2] producing the full human DNA sequencing. It is widely believed that thousands of genes and their products (i.e. RNA and proteins) in a given living organism function in a complicated and orchestrated way. However, classical methods in molecular biology generally worked on a 'one gene in one experiment' basis and it implies a very limited throughput so the overall picture of gene function is hard to accomplish.

Global gene expression profiling, both at the transcript level and at the protein level, can be a valuable tool in the understanding of genes, biological networks, and cellular states. As larger and larger gene expression data sets become available, data mining techniques can be applied to identify patterns of interest in the data. Association rules, used widely in the area of market basket analysis, can be applied to the analysis of expression data as well. Association rules can reveal biologically relevant associations between different genes or between environmental effects and gene expression. An association rule has the form Antecedent ⇒ consequence, where Antecedent and consequence are disjoint sets of items, the consequence set being likely to occur whenever the Antecedent set occurs. Items in gene expression data can include genes that are highly expressed or repressed, as well as relevant facts describing the cellular environment of the genes.

Using association rule mining approach, we can analyze:

1. The expression of one gene leads to the induction of a serial of target gene expressions. This expression pattern is denoted regulation of gene expression. The relationship between one gene and the other target genes can be viewed as an associative relation.

2. Several gene expressions lead to the expression of one target gene. Transcription factors and their target gene is one of many examples in this category (Morishita, 1999).

3. Gene expression leads to the induction of new biological function (Nakaya et al., 2000).

II. ASSOCIATION RULE MINING

Here we introduce some basic definitions of the association rule mining. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of distinct items.

Definition 1: A database *D* is a set of transactions, where each transaction *T* contains a set of items in *I*.

Definition 2: A subset of *I* is called an *itemset*. An itemset is called a *k*-itemset if it contains *k* items.

Definition 3: The support of an itemset $X \rightarrow I$ for a database, denoted as *support*(*X*), is the number of transactions in the database that contain all items in *X*.

Definition 4: An itemset is called a frequent itemset if its support is greater than or equal to some user-specified minimum support. Otherwise, it is an infrequent itemset. The set of all frequent *k*-itemsets is denoted as L_k.

Definition 5: Given L_{k-1}, C_k is defined as $L_{k-1} \times L_{k-1} = \{X \cup Y \mid X, Y \text{€} L_{k-1}, \mid X \cap Y \mid = k-2\}$, where $k > 1$.

Definition 6: A frequent itemset is called a maximal frequent itemset if it is not a subset of any other frequent itemsets.

Definition 7: An association rule is defined as $X \rightarrow Y$, where $X, Y \subseteq I$, $X, Y \neq$ Ǿ and $X \cap Y =$ Ǿ.

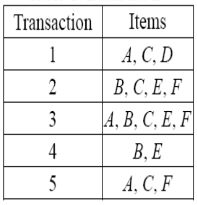
Definition 8: The support of an association rule $X \rightarrow Y$ is defined as *support*(*X* U*Y*).

Definition 9: An association rule is frequent if its support is greater than or equal to some user-specified minimum support.

Definition 10: The confidence of an association rule is defined as *support*(*X*U*Y*)/*support*(*X*). An association rule is reliable if its confidence is greater than or equal to some user-specified minimum confidence.

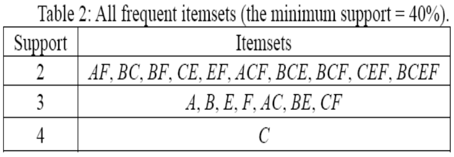
The task of the association rule mining is to discover all association rules that satisfy the minimum support and the minimum confidence. We now give an example to explain the terms described above. Consider the database shown in Table 1. Assume that *I* = {*A*, *B*, *C*, *D*, *E*, *F*} and there are five transactions in the database *D*. Let the minimum support and the minimum confidence are 40% and 100%, respectively. All frequent itemsets are shown in Table. The itemsets whose support smaller than 5 * 40% = 2 are infrequent itemsets. In this example, itemsets *D*, *AB* and *AE* are infrequent itemsets. Note that *D*, *AB* and *AE* are the short-cut of {*D*}, {*A*, *B*}, and {*A*, *E*}, respectively. We have *L*1 = {*A*, *B*, *C*, *E*, *F*}, *L*2 = {*AF*, *BC*, *BF*, *CE*, *EF*, *AC*, *BE*, *CF*}, *L*3 = {*ACF*, *BCE*, *BCF*, *CEF*} and *L*4 = {*BCEF*}. The maximal frequent itemsets are *ACF* and *BCEF*. *BC*→*EF* is one of the association rules that can be derived from the database shown in the Table. Its confidence is *support*(*BC* U *EF*)/*support*(*BC*) = 2/2*100% = 100%. *BC*→*EF* is a frequent and reliable association rule.

Table 1: Database *D*.



Transaction	Items
1	*A, C, D*
2	*B, C, E, F*
3	*A, B, C, E, F*
4	*B, E*
5	*A, C, F*

Table 2: All frequent itemsets (the minimum support = 40%).



Support	Itemsets
2	*AF, BC, BF, CE, EF, ACF, BCE, BCF, CEF, BCEF*
3	*A, B, E, F, AC, BE, CF*
4	*C*

III. DIFFERENT ASSOCIATION RULE MINING ALGORITHMS

A number of efficient association rule mining algorithms have been proposed in the last few years. Among these, the Apriori algorithm (Agrawal & Srikant, 1994) has been very influential. Since its inception, many scholars have improved and optimized the Apriori algorithm and have presented new Apriori-like algorithms.

The Apriori-like algorithms adopt an iterative method to discover frequent itemsets. The algorithm starts from frequent 1-itemsets until all maximum frequent itemsets are discovered. The Apriori-like algorithms consist of two major procedures: the join procedure and the prune procedure. The join procedure combines two frequent k-itemsets, which have the same (k-1)-prefix, to generate a (k+1)-itemset as a new preliminary candidate. Following the join procedure, the prune procedure is used to remove from the preliminary candidate set all itemsets whose k-subset is not a frequent itemset. A huge calculation and a complicated transaction process are required during the two procedures. Therefore, the mining efficiency of the Apriori-like algorithms is very unsatisfactory when transaction database is very large. It was improved by partition [4] and sampling [5], but both of these approaches were inefficient when the database was dense. PASCAL [6] is an optimization of Apriori but when all the candidate patterns are candidate key patterns, then the algorithm behaves exactly like apriori.

Here we have described some of the popular data mining algorithm in brief.

Apriori: The main contribution of the *Apriori* algorithm is it utilizes the downward closure property, i.e., any superset of an infrequent itemset must be an infrequent itemset, to efficiently generate candidate itemsets for the next database scan.By scanning a database *k* times, the *Apriori* algorithm can find all frequent itemsets of a database, where *k* is the length of the longest frequent itemset in the database. Many methods based on the *Apriori* algorithm have been proposed in the literature. In general, they can be classified into three categories, reduce the number of candidate itemsets, reduce the number of database scans, and the combination of bottom-up and top-down search.

The Apriori algorithm is given as follows.

```
Algorithm Apriori()
1.  Scan D to obtain L1, the set of frequent 1-itemsets;
2.  for (k = 2; $L_{k-1} \neq Ø$; k++) do
3.      $C_k$ = apriori-gen($L_{k-1}$); // Generate new candidates from $L_{k-1}$
4.      for all transactions t € D do
5.        $C_t$ = subset($C_k$, t); // Candidates contained in t
6.        for all c €Ct do
7.          c. count++;
8.      $L_k$ = {c € $C_k$ | c. count ≥ minimum support};
9.  All frequent itemsets = $U_k L_k$;
end_of_Apriori
```

In the first round, the Apriori algorithm scans the database to determine L1 (line 1). In the kth round, where k ≥ 2, the process of the Apriori algorithm can be divided into the following three steps.

Step 1: Line 3 constructs Ck from Lk-1, which was determined in the (k-1)th round.

Step 2: Lines 4-7 scan the database to count the support of each k-itemset in C_k.

Step 3: Line 9 determines the L_k, whose support is greater than or equal to the user-specified minimum support, from C_k. The algorithm terminates when no more candidate itemsets can be constructed for next round. The algorithm needs to do multiple database scans as many times as the length of the longest frequent itemset. Therefore, its performance decreases dramatically when the length of the longest frequent itemset is relatively long.

FP growth: FP-tree growth (Han et al.) adopts a divide-and-conquer strategy that mines the complete set of frequent itemsets without candidate generation. FP growth algorithm constructs the conditional frequent pattern (FP)-tree and performs the mining on this tree. FP-tree is an extended prefix tree structure, storing crucial and quantitative information about frequent sets. The tree nodes are frequent items and are arranged in such a way that more frequently occurring nodes will have better chances of sharing nodes than the less frequently occurring ones. The method starts from frequent 1- itemsets as an initial suffix pattern and examines only its conditional pattern base (a subset of the database), which consists of set of frequent items co-occurring with the suffix pattern. The algorithm involves two phases. In phase I, it constructs the FP-tree with respect to a given support factor . The construction of this tree requires two passes over the whole database. In phase II, the algorithm does not use the transaction database anymore, but it uses the FP-tree. Interestingly, the FP-tree contains all the information about frequent itemsets with respect to the given. The FP-tree growth method transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. The method substantially reduces the search costs .When the database is large, it is sometimes unrealistic to construct a main memory-base FP-tree. A study on the performance of the FP method shows that it is efficient and scalable for mining both long and short frequent patterns.

The original algorithm to construct the FP-Tree defined by Han is presented below in Algorithm 1.

Algorithm 1: FP-tree construction

Input: A transaction database DB and a minimum support threshold ?.

Output: FP-tree, the frequent-pattern tree of DB.

Method: The FP-tree is constructed as follows.

1. Scan the transaction database DB once. Collect F, the set of frequent items, and the support of each frequent item. Sort F in support-descending order as FList, the list of frequent items.
2. Create the root of an FP-tree, T, and label it as “null”. For each transaction Trans in DB do the following:

- Select the frequent items in Trans and sort them according to the order of FList. Let the sorted frequent-item list in Trans be [p | P], where p is the first element and P is the remaining list. Call insert tree([p | P], T).
- The function insert tree([p | P], T) is performed as follows. If T has a child N such that N.item-name = p.item-name, then increment N ’s count by 1; else create a new node N , with its count initialized to 1, its parent link linked to T , and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree(P, N) recursively.

By using this algorithm, the FP-tree is constructed in two scans of the database. The first scan collects and sort the set of frequent items, and the second constructs the FP-Tree.

FP-Growth Algorithm

After constructing the FP-Tree it’s possible to mine it to find the complete set of frequent patterns. To accomplish this job, Han in [1] presents a group of lemmas and properties, and thereafter describes the FP-Growth Algorithm as presented below in Algorithm 2.

Algorithm 2: FP-Growth

Input: A database DB, represented by FP-tree constructed according to Algorithm 1, and a minimum support threshold.
Output: The complete set of frequent patterns.
Method: call FP-growth (FP-tree, null).

```
Procedure FP-growth(Tree, a) {
 if Tree contains a single prefix path then // Mining single prefix-path FP-tree {
let P be the single prefix-path part of Tree;
let Q be the multipath part with the top branching node replaced by a null root;
for each combination (denoted as ß) of the nodes in the path P do
generate pattern ß ? a with support = minimum support of nodes in ß;
let freq pattern set(P) be the set of patterns so generated;
}
 Else let Q be Tree;
for each item ai in Q do { // Mining multipath FP-tree
generate pattern ß = ai ? a with support =$a_i$. support;
 construct ß’s conditional pattern-base and then ß’s conditional FP-tree Tree ß;
 if Tree ß = Ø then
call FP-growth(Tree ß , ß);
 let freq pattern set(Q) be the set of patterns so generated;
}
return(freq pattern set(P) , freq pattern set(Q) , (freq pattern set(P) × freq pattern set(Q)))
}
```

When the FP-tree contains a single prefix-path, the complete set of frequent patterns can be generated in three parts: the single prefix-path P, the multipath Q, and their combinations . The resulting patterns for a single prefix path are the enumerations of its subpaths that have the minimum support. Thereafter, the multipath Q is defined and the resulting patterns from it are processed. Finally, the combined results are returned as the frequent patterns found.

IV. NEURAL NETWORK

Artificial neural networks are inspired by the operation of the human brain. It is a model of the biological neuron as a circuit component to perform computational tasks. Artificial neural networks consist of a number of simple computing elements called neurons that are modeled after the human nerve cell. Each neuron receives a number of input signals and performs a simple operation on this set of inputs. The output of each neuron is fanned out to the inputs of other neurons.[18]

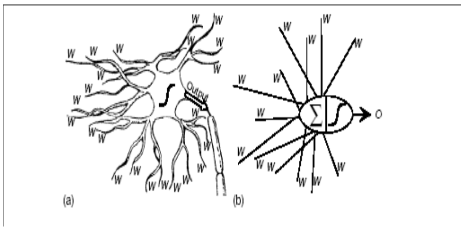


Figure: Human nerve cell (a) and its model (b). Weighted (*w*) input signals are added. The resulting sum is compared to a threshold as is depicted with the nonlinear, S-shaped neural response function in the cell body.[18]

In Fig. 1 a human nerve cell, or neuron, (a) and its artificial equivalent (b) are sketched. The neuron receives a set of input signals via a number of tentacles or dendrites. At the tip of each dendrite the input signal is weighted with a factor w, which can be positive or negative. All the signals from the dendrites are added in the cell body to contribute to a weighted sum of inputs of the neuron. If a weight is positive the corresponding input will have an excitatory influence on the weighted sum. With a negative weight, an input decreases the weighted sum and is inhibitory. In the cell body the weighted sum of inputs is compared to a threshold value. If the weighted sum is above this threshold, the neuron sends a signal via its output to all connected neurons. The threshold operation

is essentially a nonlinear response function as is indicated in the figure with an S-shaped, sigmoid, curve. The function of a neuron can be described in mathematical form with:

$$O = F\left(\sum w_i \,.\, I_i \right)$$

where, O is the output signal of the neuron and Ii are the input signals to the neuron, weighted with a factor w_i. F is some nonlinear function representing the threshold operation on the weighted sum of inputs.

V. ASSOCIATION RULE MINING USED IN GENE EXPRESSION DATA

Using association rule mining approach, we can analyze:

1. The expression of one gene leads to the induction of a serial of target gene expressions. This expression pattern is denoted regulation of gene expression. The relationship between one gene and the other target genes can be viewed as an associative relation.

2. Several gene expressions lead to the expression of one target gene. Transcription factors and their target gene is one of many examples in this category (Morishita, 1999).

3. Gene expression leads to the induction of new biological function (Nakaya et al., 2000).

The gene expression data in microarray are presented in M×N matrix where M is the number of microarray experiments and N being the number of genes [1]. The number of experiments M can range from dozens to thousands. On the other hand, the number of genes N can range from hundred to tens of thousands. In some context, M can be referred to as number of transactions or item sets where each gene represents an item. To add to the complexity of representation, each gene is measured in terms of absolute values. However, biologists are more interested in how gene expression changes under different environments in each respective experiment. Thus, these absolute values are discretized according to some predetermined thresholds and grouped under three different levels, namely unchanged, up regulated and down regulated.

In market basket analysis, an association rule represents a set of items that are likely to be purchased together; for example, the rule {cereal}→{milk, juice} would state that whenever a customer purchases cereal, he or she is likely to purchase both milk and juice as well in the same transaction. In the analysis of gene expression data, the items in an association rule can represent genes that are strongly expressed or repressed, as well as relevant facts describing the cellular environment of the genes (e.g. a diagnosis for a tumor sample that was profiled, or a drug treatment given to cells in the sample before profiling). An example of an association rule mined from expression data might be {gene A,→gene B↑},{Gene A→gene C↑}, meaning that, if gene B is expressed highly then gene A is the reason for the particular disease.

VI. IMPLEMENTATION

We developed a database application that implements a version of the Apriori algorithm as described in Methods for first finding frequent genesets and then generating association rules from those sets using artificial neural network. As input, the algorithm accepts a dataset of breast cancer samples of some genes responsible for the disease. Each data is measured according to their maximum range.Then depending upon some user defined constraints the genesets are computed. The application then mines the database for frequent genesets that exist within the database. The application proceeds iteratively using Apriori until all frequent genesets have been found. The user can also specify a minimum support in selecting frequent itemsets of interest. Once the data set has been mined for frequent itemsets, the application then generates association rules from these genesets.After the rule generation procedure has been completed , neural network selects the strongest rule responsible for the disease

Once the frequent itemsets in the database have been mined for association rules, the user can export the results from the database into a spreadsheet. The user can limit the exported results to rules of a specified number of items or which include at least one item within a specified set of items.

VII. DATA SETS

To demonstrate the algorithm, we used the Gene Omnibus Dataset(GEO) of the Database institute named NCBI NLM NIH.

TheDatabase_web_link is http://www.ncbi.nlm.nih.gov/projects/geo .

Database email is geo@ncbi.nlm.nih.gov.

The reference of the Database is Nucleic Acids Res. 2005 Jan 1;33 Database Issue:D562-6 Dataset title is Breast cancer cell expression profiles (HG-U133B). Dataset type is gene expression array-based. Dataset_pubmed_id is 15161944. Platform of the dataset is GPL97. Name of dataset platform organism is Homo sapiens. Name of dataset_platform_technology in situ oligonucleotide. dataset_feature_count = 22645. Dataset_sample_organism is Homo sapiens. Dataset sample type is RNA. Dataset_sample_count is 6. Dataset value type is transformed count. Dataset_reference_series - GSE1299 . Dataset order none. Dataset_update_date = Nov 02 2004. SUBSET is GDS823_1. Subset dataset id = GDS823.Subset description is normal.

Subset sample id : GSM21252,GSM21253.Subset type is disease state. SUBSET = GDS823_2. Subset dataset id = GDS823. Subset description :cancer. Subset_sample_id = GSM21248,GSM21249,GSM21250,GSM21251. Subset type :disease state. SUBSET = GDS823_3. Subset dataset id : GDS823. Subset description MDA-MB-436.

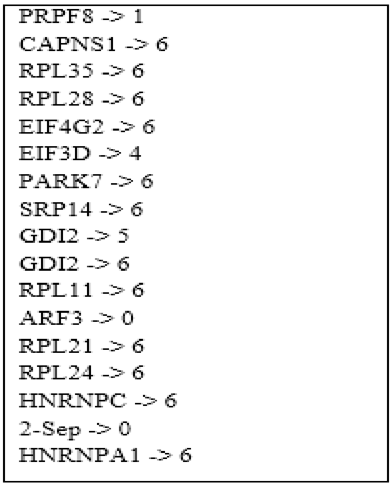
Subset_sample_id : GSM21248,GSM21249. Subset type : cell line. SUBSET : GDS823_4.

Subset dataset id is GDS823.Subset description is HCC 1954. Subset sample id : GSM21250,GSM21251. Subset_type : cell line. SUBSET = GDS823_5. Subset dataset id: GDS823.Subset description is mammary epithelium. Subset sample id : GSM21252,GSM21253. Subset type : cell line. DATASET = GDS823. ID_REF = Platform reference identifier. IDENTIFIER = identifier. GSM21252 = Value for GSM21252: Normal Breast Epithelium Control replicate 1 133B; src: Human Mammary Epithelial Cells. GSM21253 = Value for GSM21253: Normal Breast Epithelium Control replicate 2 133B; src: Human Mammary Epithelial Cells. GSM21248 = Value for GSM21248: Breast Cancer cells MDA-MB-436 replicate 1 133B; src: MDA-MB436 Breast Cancer cell line GSM21249 = Value for GSM21249: Breast Cancer cells MDA-MB-436 replicate 2

133B; src: MDA-MB436 Breast Cancer cell line GSM21250 = Value for GSM21250: Breast Cancer cells HCC1954 replicate 1 133B; src: HCC1954 Breast Cancer cell line GSM21251 = Value for GSM21251: Breast Cancer cells HCC1954 replicate 2 133B; src: HCC1954 Breast Cancer cell line.

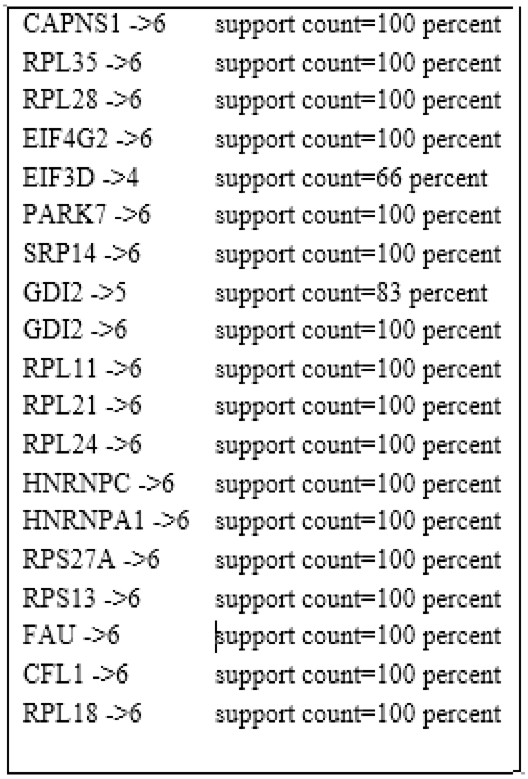
VIII. RESULTING DATASETS

Few of the datasets are shown in this table.



```
PRPF8 -> 1
CAPNS1 -> 6
RPL35 -> 6
RPL28 -> 6
EIF4G2 -> 6
EIF3D -> 4
PARK7 -> 6
SRP14 -> 6
GDI2 -> 5
GDI2 -> 6
RPL11 -> 6
ARF3 -> 0
RPL21 -> 6
RPL24 -> 6
HNRNPC -> 6
2-Sep -> 0
HNRNPA1 -> 6
```

Scan database for count of each gene



```
CAPNS1 ->6      support count=100 percent
RPL35 ->6       support count=100 percent
RPL28 ->6       support count=100 percent
EIF4G2 ->6      support count=100 percent
EIF3D ->4       support count=66 percent
PARK7 ->6       support count=100 percent
SRP14 ->6       support count=100 percent
GDI2 ->5        support count=83 percent
GDI2 ->6        support count=100 percent
RPL11 ->6       support count=100 percent
RPL21 ->6       support count=100 percent
RPL24 ->6       support count=100 percent
HNRNPC ->6      support count=100 percent
HNRNPA1 ->6     support count=100 percent
RPS27A ->6      support count=100 percent
RPS13 ->6       support count=100 percent
FAU ->6         support count=100 percent
CFL1 ->6        support count=100 percent
RPL18 ->6       support count=100 percent
```

After comparing candidate support count with minimum support count

CAPNS1	RPL35	6	support count=100 percent
CAPNS1	RPL28	6	support count=100 percent
CAPNS1	EIF4G2	6	support count=100 percent
CAPNS1	EIF3D	4	support count=66 percent
CAPNS1	PARK7	6	support count=100 percent
CAPNS1	SRP14	6	support count=100 percent
CAPNS1	GDI2	5	support count=83 percent
CAPNS1	GDI2	5	support count=83 percent
CAPNS1	RPL11	6	support count=100 percent
CAPNS1	RPL21	6	support count=100 percent
CAPNS1	RPL24	6	support count=100 percent

Two genes dataset after comparing with minimum support count

CAPNS1	RPL35	RPL28	6	support count=100 percent
CAPNS1	RPL35	EIF4G2	6	support count=100 percent
CAPNS1	RPL35	EIF3D	4	support count=66 percent
CAPNS1	RPL35	PARK7	6	support count=100 percent
CAPNS1	RPL35	SRP14	6	support count=100 percent
CAPNS1	RPL35	GDI2	5	support count=83 percent
CAPNS1	RPL35	GDI2	5	support count=83 percent
CAPNS1	RPL35	RPL11	6	support count=100 percent
CAPNS1	RPL35	RPL21	6	support count=100 percent
CAPNS1	RPL35	RPL24	6	support count=100 percent
CAPNS1	RPL35	HNRNPC	6	support count=100 percent
CAPNS1	RPL35	HNRNPA1	6	support count=100 percent
CAPNS1	RPL35	RPS27A	6	support count=100 percent
CAPNS1	RPL35	RPS13	6	support count=100 percent

Three genes dataset after comparing with minimum support count

CAPNS1	RPL35	RPL28	EIF4G2	6	support count=100 percent
CAPNS1	RPL35	RPL28	EIF3D	4	support count=66 percent
CAPNS1	RPL35	RPL28	PARK7	6	support count=100 percent
CAPNS1	RPL35	RPL28	SRP14	6	support count=100 percent
CAPNS1	RPL35	RPL28	GDI2	5	support count=83 percent
CAPNS1	RPL35	RPL28	GDI2	5	support count=83 percent
CAPNS1	RPL35	RPL28	RPL11	6	support count=100 percent
CAPNS1	RPL35	RPL28	RPL21	6	support count=100 percent
CAPNS1	RPL35	RPL28	RPL24	6	support count=100 percent
CAPNS1	RPL35	RPL28	HNRNPC	6	support count=100 percent
CAPNS1	RPL35	RPL28	HNRNPA1	6	support count=100 percent
CAPNS1	RPL35	RPL28	RPS27A	6	support count=100

Four genes dataset after comparing with minimum support count

IX. GENERATED RULES

Some of the Association rules generated from the dataset
1 { CAPNS1}→{RPL35, RPL28, SRP14}
2 { KHDRBS,1BAT1}→{ERH, URG4}
3 { RPL19→MRPL51}

X. SCOPE AND LIMITATIONS

Association rule generation for gene expression data is used for gene prediction and expression identification. It is useful to clarify the robustness of the bonding in case of gene-gene interaction. The identification of desease mediating gene is also done using association rule for gene expression data.

The association rules that we have mined from the Breast cancer data certainly represent only a fraction of all of the possible gene-to-gene interactions that remain to be discovered in this disease. More rules could be found by using different search criteria (e.g. a lower minimum support) or another large data set. The rules that we have found, however, do represent a considerable number of non-random patterns of interest that could lead to the generation of new hypotheses to explain them, hypotheses that could ultimately be confirmed experiments.

XI. CONCLUSION

In this paper, we have proposed the Apriori algorithm for gene expression data with the help of neural network, So far we have implemented Apriori algorithm and single layer feed forward network. Our goal is to improve the accuracy of the traditional Apriori algorithm implementing it using neural network.As we know that the neural network is applicable for supervised, unsupervised as well as reinforcement learning, so it is expected to provide better accuracy, efficiency as well as less time and space complexity.

XII. BIBLIOGRAPHY

[1] F.S. Collins, V.A. McKusick, Implications of the Human Genome Project for Medical Science, The Journal of the American Medical Association, vol. 285,pp. 540{544, 2001.

[2] J.D. Watson, The human genome project: past, present, and future, Science (1990) vol. 248, pp. 44{49, 1990.

[3] Francesco Masulli(*Department of Computer and Information Sciences, University of Genova, Via Dodecaneso 35, 16146 Genoa, ITALY*) and Sushmita Mitra(*Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, INDIA*)" Natural Computing Methods in Bioinformatics: A Survey". *Preprint submitted to Information Fusion. 20 November 2008.*

[4] *M. Anandhavalli Gauthaman." Analysis of DNA Microarray Data usingAssociation Rules: A Selective Study" .World Academy of Science, Engineering and Technology 42 2008*

[5] J. Han M.Kamber, "Data Mining Concepts and Techniques". Morgan Kaufmann Publishers, San Francisco, USA, 2001, ISBN 1558604898.

[6] M.H. Dunham. "Data Mining – Introductory and Advanced Topics". Prentice Hall, 2003, ISBN 0-13-088892-3.

[7] C.Gyorodi, R.Gyorodi. "Mining Association rules in Large Databases". Proc. of Oradea EMES'02: 45-50, Oradea, Romania, 2002.

[8] Website: [http://genomebiology.com/2002/3/12/research/0067]

[9] Céline Becquet, Sylvain Blachon1, Baptiste Jeudy, Jean-Francois Boulicaut, Olivier Gandrillon: "Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data"published:21st November 2002 © 2002 Becquet et al., licensee BioMed Central Ltd

[10] Amit Bhagat (Department of Computer Applications), Dr. Sanjay Sharma (Associate Prof. Deptt. of Computer Applications), Dr. K.R.Pardasani (Professor Deptt. of Mathematics);"Feed Forward Neural Network Algorithm for Frequent Patterns Mining", Maulana Azad National Institute of Technology, Bhopal (M.P.)462051, India. (IJCSIS) International Journal of Computer Science and Information Security,Vol. 8, No.8, November 2010.

[11] Website:- [http://www.biomedcentral.com/1471-2105/7/54]

[12] Reference: Pedro Carmona-Saez, Monica Chagoyen, Andres Rodriguez, Oswaldo Trelles, Jose M Carazo, Alberto Pascual-Montano: "Integrated analysis of gene expression by association rules discovery" published 7 February 2006 © 2006 Carmona-Saez et al; licensee BioMed Central Ltd.

[13] Ah-Hwee Tan of School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore, Hong Pan of Genome Institute of Singapore, 60 Biopolis Street #02-01, Genome, Singapore 138672, Singapore; "Predictive neural networks for gene expression data analysis" accepted 17 January 2005. A.-H. Tan, H. Pan / Neural Networks 298 18 (2005) 297–306. www.elsevier.com/locate/neunet.

[14] R.Agrawal, T.Imielinki and A.Swami, "Mining association rules between set of item of large databases" in Proc. Of the ACM SIGMOD Intl'l Conf. on Management of data, Washington, D.C.,USA, 1993, pp 207-216.

[15] M.Anandhavalli *Member, IACSIT, IAENG*, M.K.Ghose, K.Gauthaman." Association Rule Mining in Genomics" International Journal of Computer Theory and Engineering, Vol. 2, No. 2 April, 2010. 1793-8201 pages:12-15.

[16] Chad Creighton, Samir Hanash. Bioinformatics Program and 2Pediatrics and Communicable Diseases, University of Michigan, Ann Arbor, MI 48109, USA. " Mining gene expression databases for association rules ". Received on April 19, 2002; revised on July 1, 2002; accepted on July 10, 2002. Vol. 19 no. 1 2003 Pages 79–86

[17] S.Brin, R.Motawani, J.D.Ullman and S. Tsur, "Dynamic Itemset counting and implication rules for market basket data" in Proc. of the ACM SIGMOD Intl'l Conf. on Management of data, Tucson, Arizona, USA, 1997, pp. 255-264.

[18] Mos, Evert C.: Optical Neural Network based on Laser Diode Longitudinal Modes.