Mining Frequent Patterns from Microarray Data

Baris Yildiz
Department of Computer Engineering
Dokuz Eylul University
Izmir, Turkey
barisyildiz@computer.org

Hatice Selale
Department of Molecular Biology and Genetics
Izmir Institute of Technology
Izmir, Turkey
haticeselale@iyte.edu.tr

Abstract— **The rapid development of computers and increasing amount of collected data made data mining a popular analysis tool. Data mining research is interrelated to many fields and one of the most important ones is bioinformatics. Among many techniques, mining association rules or frequent patterns is one of the most studied techniques in computer science and it is applicable to bioinformatics. Association analysis of gene expressions may be used as decision support mechanism for finding genes that are in same pathway. In this work, publicly available yeast microarray data has been analyzed using an efficient frequent pattern mining algorithm Matrix Apriori and frequently co-over-expressed genes have been identified.**

Keywords—frequent pattern mining; microarray

I. Introduction

The process of discovering knowledge or patterns from massive amounts of data is defined as data mining [1]. In last decades data mining has become a popular research area in computer science and also it is applied to many problems in other disciplines. With the increasing power of computers, data mining applications became capable of handling huge amount of data and this attracted people in many disciplines as well as bioinformatics. Classification, clustering and association rule or frequent pattern mining are three main types of data mining techniques [2].

Association analysis is studied by many computer scientists and applied to many fields. It came into prominence by the help of barcode technology which resulted construction of transactional databases in markets. Later it was thought that it would be beneficial to find frequently purchased items in the markets in order to increase sales. In [3], association rule mining was introduced as a new data mining technique which could be used for market basket analysis. Association rule mining, searches for items frequently purchased together when the market domain is considered. For instance, through frequent pattern mining, it can be found that sugar and tea are purchased together frequently. Placing these items closer or going on a discount may increase sales. Mining frequent patterns is used in many applications, however, it is not that much applied to bioinformatics.

Microarray is an effective technique used in molecular biology to analyze gene expression in different conditions such as control versus drug treatment or healthy versus disease conditions in many organisms. Huge amount of data has been generated already and it is difficult to handle such large amount of data for analysis. Mining frequent patterns in gene expression data seems applicable and may help bioinformatics researchers. In the gene expression analysis context, we are looking for which genes are frequently expressed together in different experiment conditions. This will lead us to knowledge of genes that are expressed association in defined conditions hereby this will facilitate the molecular biologists to define components of biological pathways with further studies.

In this study we applied a frequent pattern mining algorithm to a DNA microarray gene expression dataset. A publicly available dataset of “saccharomyces cerevisiae” [4] is used. To apply frequent pattern mining, firstly, gene expression data will be transformed into transactional dataset like market basket datasets [5]. Over-expressed genes will be considered as purchased items and experiment conditions as transactions. After transformation, frequent pattern mining algorithm will be applied on this generated transactional dataset. At last, we will gain the knowledge to help us determine the possible interactions between different genes. For our study we used Matrix Apriori algorithm [6] for mining frequent patterns.

The organization of the paper is as follows. Next section introduces DNA microarray and how it is obtained. Following that in section 3, frequent pattern mining and Matrix Apriori algorithm are explained with examples. In section 4, results of case study is given and at last paper is concluded.

II. DNA Microarray

DNA microarray technology is an advanced technique to study gene expression in different conditions like drug treatment versus control or healthy versus disease conditions. DNA microarray technology is based on hybridization of DNA or oligonucleotide probes which represent specific genes with fluorescently or chemiluminescence-labeled DNA or cDNA fragments. Tens of thousands of DNA or oligonucleotide probes covalently bound to glass, silicone chips or microscopic beads to form DNA micro arrays in a precise and known pattern and representing thousands of genes in the genome [7].

Microarray technique is simply depicted in Fig. 1. mRNA is isolated from organism of interest in control and test conditions than mRNA is converted to cDNA with reverse transcription reaction than, cDNA samples from control and test conditions labeled with different florescent tags. The two pools of labeled cDNA are mixed and hybridized to the DNA microarray containing a full set of tens of thousands of DNA sequences based on genomic or complimentary DNA (cDNA)

978-1-4673-4431-9/11//$31.00 ©2012 IEEE

sequences after hybridization, the chip is washed. Finally, the microarray array is scanned using a specialized fluorimager, and the color and intensity of each spot is determined. According to the colors and intensities of each spot expression levels of genes can be calculated by contrasting control versus treatment arrays [8]. As a result of a microarray experiment, datasets containing long lists of measurements of spot intensities and intensity ratios which are generated either by pair wise comparison of two samples or by comparing several samples to a common control are generated. Differentially expressed genes or co-regulated genes with related function can be identified by analyzing the generated dataset with some complicated softwares [9].

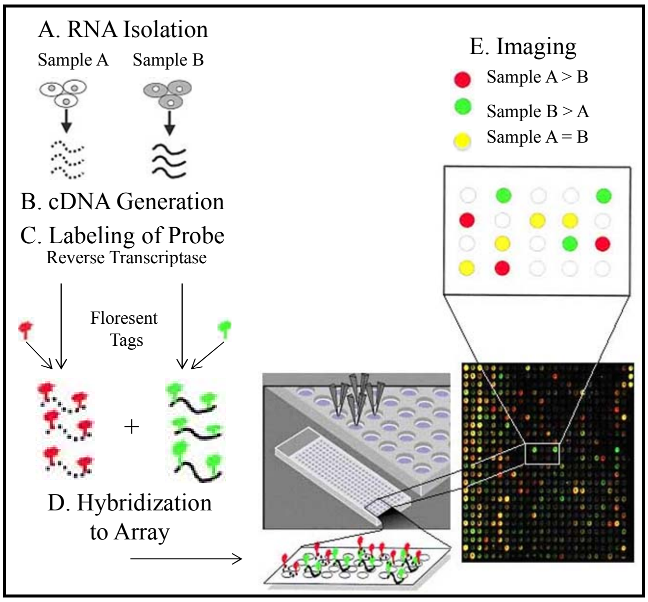


Figure 1. Steps of microarray technique (source:http://www.fastol.com/~renkwitz/microarray_chips.htm)

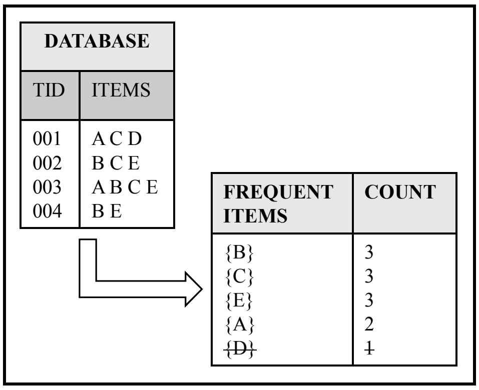
III. Frequent Pattern Mining

Frequent pattern or itemset mining is simply finding which things go together. It is the most important and time consuming part of association rule mining. Researchers are focused on efficiently finding frequent itemsets such that frequent pattern mining and frequent itemset mining began to be used as synonym of association rule mining. Association rule mining was firstly introduced in [3] and it is one of the most important and well defined techniques of data mining. Aim of association rule mining is to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transactional databases or other data repositories [10]. If you consider market basket data, the purchasing of one product(X) while another product(Y) is purchased represents an association rule [11] and is displayed as X→Y. Association rule mining process consists of two steps: finding frequent itemsets and generating association rules. The rules are generated from frequent itemsets. An itemset is a set of items in the database. A frequent itemset is one of which support value (percentage of transactions in the database that contain both X and Y) is above the threshold defined as minimum support. The main concentration of most association rule mining algorithms is to find frequent item-sets in an efficient way to reduce the overall cost of the process.

A. Matrix Apriori Algorithm

After introduction of association rules, Apriori algorithm proposed in [12] and it happened to be one of the most popular frequent pattern mining algorithms. In spite of its popularity it has a bottleneck of multiple database scans for generation and testing of candidate patterns. FP-Growth algorithm [13] is alternative to Apriori since it finds frequent patterns without candidate generation. Another efficient algorithm without candidate generation is Matrix Apriori algorithm [6]. It was shown to perform better than FP-Growth in [14]. It uses a compact data structure representing the original database. Data structure build for Matrix Apriori is a matrix representing frequent items (MFI) and a vector holding support of candidates (STE). The search for frequent patterns is executed on these two structures, which is faster than accessing to the original database [14].

In Fig. 2, first phase of Matrix Apriori algorithm is demonstrated. For the example transactional dataset, support value is 2 (%50). Firstly, a database scan to determine frequent items is executed and a frequent items list is obtained. The list is in descending order.



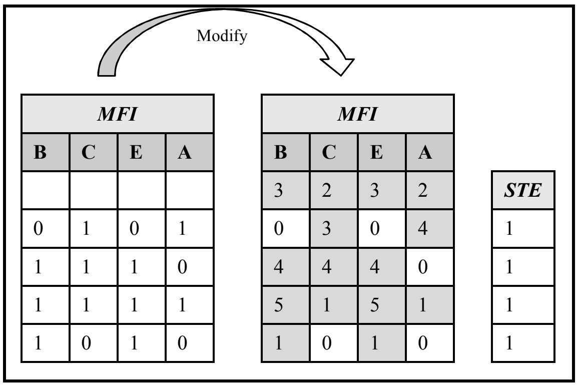
DATABASE	
TID	ITEMS
001 002 003 004	A C D B C E A B C E B E

FREQUENT ITEMS	COUNT

{B}
{C}
{E}
{A}
~~{D}~~
	3
	3
	3
	2
	~~1~~

Figure 2. First scan of Matrix Apriori algorithm

Following this, a second scan on database is executed. During the scan, MFI and STE are constructed as follows. Each transaction is read. If the transaction has any item that is in the frequent item list then it is represented as "1" and other-wise "0". This pattern is added as a row to MFI matrix and its occurrence is set to 1 in STE vector. While reading remaining transactions if the transaction is already included in MFI then in STE its occurrence is incremented. Otherwise it is added to MFI and its occurrence in STE is set to 1. After reading all transactions the MFI matrix is modified to speed up frequent pattern search. For each column of MFI, beginning from the first row, the value of a cell is set to the row number in which the item is "1". If there is not any "1" in remaining rows then the value of the cell is set to "1" which means down to the bottom of the matrix, no row contains this item (see Fig. 3).



MFI			
B	C	E	A
0	1	0	1
1	1	1	0
1	1	1	1
1	0	1	0

MFI			
B	C	E	A
3	2	3	2
0	3	0	4
4	4	4	0
5	1	5	1
1	0	1	0

STE
1
1
1
1

Figure 3. Second scan for building data structure for Matrix Apriori algorithm

After constructing the MFI matrix finding patterns is simple. Beginning from the least frequent item, create candidate itemsets and count their support value. The support value of an itemset is the sum of the items at STE of which index are rows where all the items of the candidate itemset are included in MFI's related row. Frequent itemsets found can be seen in Fig. 4.

FREQUENT ITEMSETS	SUPPORT
C,A	2
C,E	2
B,E	3
B,C	2
B,C,E	2

Figure 4. Frequent itemsets/patterns found

IV. Case Study

In case study, publicly available yeast microarray dataset from [4] is analyzed. There are 173 experimental conditions and 6152 genes. Frequently over-expressed genes are found for given minimum support values. Firstly, dataset is transformed into appropriate format that developed Matrix Apriori mining software can process. The over-expressed genes are considered (like items purchased in market basket data) in order to obtain transactional dataset. In Fig. 5, the structure of a simple dataset file can be seen. For each gene a number is assigned (6152 for our dataset) and between "BEGIN_DATA" and "END_DATA" microarray data is given. Each row represents the genes over-expressed in that experiment condition (173 for our dataset).

```
1 Gene1
2 Gene2
3 Gene3
4 Gene4
5 Gene5
BEGIN_DATA
1 3 4
2 3 5
1 2 3 5
2 5
END_DATA
```

Figure 5. File structure for developed program

After transformation, frequent pattern mining is performed. Results are obtained for four minimum support values. Table 1 below displays the runtime (milliseconds), number of frequent patterns found for given minimum support.

TABLE I. Results for different minimum support values

MINIMUM SUPPORT (%)	NUMBER OF FREQUENT PATTERNS	TIME TO FIND FREQUENT PATTERNS (ms)
90	0	226
80	9	302
70	8343	10147
60	150507538	33317405

Results of frequent pattern mining will be useless if there is unmanageable number of patterns found. For 70% and 60% minimum support values, this situation occurs. The number of frequent patterns found is very high. It will be difficult to use these results for decision support. Time costs also reveal the difficulty of pattern mining in microarray data. Therefore results of 80% minimum support are given below in Table 2. First column displays UIDs of frequently over-expressed genes. Second column displays support count and support percentage of frequent gene patterns. Third column gives functions of the genes (explanations are obtained from dataset file and give biological process, function/protein name) for the patterns found.

What can be inferred from patterns below is that the possibility of some genes being in same pathway is high. These genes are related and they will function at similar processes or over-expression of one affects one another.

TABLE II. FREQUENT GENE PATTERNS FOUND

FREQUENT GENE PATTERNS	SUPPORT	GENE FUNCTION
YPL154C YGR209C	138(~80%)	Protein degradation, Vacuolar aspartyl protease DNA replication, Thioredoxin II
YKL065C YGR209C	139(~80%)	Unknown, ER 25 KDA transmemrane protein DNA replication, Thioredoxin II
YLR250W YGR209C	142(~82%)	Secretion, Unknown DNA replication, Thioredoxin II
YJR104C YGR209C	138(~80%)	Oxidative stress response, Copper-zinc superoxide dismutase DNA replication, Thioredoxin II
YDL124W YGR209C	139(~80%)	Unknown, Unknown DNA replication, Thioredoxin II
YIL124W YIR037W	142(~82%)	Unknown, Unknown; similar to insect-type alcohol/ribitol dehydrogenase Oxidative stress response, Glutathione peroxidase
YIL124W YIR037W YGR209C	139(~80%)	Unknown, Unknown; similar to insect-type alcohol/ribitol dehydrogenase Oxidative stress response, Glutathione peroxidase DNA replication, Thioredoxin II
YIR037W YGR209C	142(~82%)	Oxidative stress response, Glutathione peroxidase DNA replication, Thioredoxin II
YIL124W YGR209C	145(~84%)	Unknown, Unknown; similar to insect-type alcohol/ribitol dehydrogenase DNA replication, Thioredoxin II

V. CONCLUSION

In conclusion, this work demonstrates frequent pattern mining application on yeast microarray data. In [15] Apriori algorithm is used for analysis of gene expression data. However, Matrix Apriori algorithm is shown to be faster than the popular Apriori and FP-Growth algorithms. Therefore, we used Matrix Apriori algorithm for data mining. Firstly, data is transformed into appropriate format and frequent pattern analysis is applied. Most frequent genes are defined in the case study. The applied approach for microarray data analysis will be helpful to molecular biologist for analyzing the microarray data and identify the candidate genes acting in same or related biological pathways. In this study, only over-expressed genes are analyzed and it would be nice future work to analyze association of under-expressed genes and also both under and over-expressed genes together. As a result of our study, we obtained some information in agreement with published results. As is known in literature, in increased stress conditions oxidative stress defense mechanisms are activated as a result of increased oxidative stress [16]. In our results three oxidative stress defense related genes are detected to be over-expressed in association with several genes. The obtained information seems more interesting when we consider this microarray data was obtained in stress treatment conditions.

REFERENCES

[1] L. Liu, M. Özsu, Encyclopedia of Database Systems. Springer, 2009.

[2] J. Han, M. Kamber, Data Mining Concepts and Tech-niques. Morgan Kaufmann, 2006

[3] R. Agrawal, T. Imielinski and A. Swami, “Mining asso-ciation rules between sets of items in large databases,” in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993, pp.207-216

[4] Microarray dataset of “saccharomyces cerevisiae” [Online]. Available: http://www-genome.stanford.edu/yeast-stress/ (last access July 2010)

[5] R. Alves, D. Rodriguez-Baena and J. Aguilar-Ruiz, “Gene association analysis: a survey of frequent pattern mining from gene expression data,” Briefings in Bioinfor-matics, vol. 11, pp. 210-224, 2009.

[6] J. Pavón, S. Viana & S. Gómez, “Matrix Apriori: Speed-ing up the Search for Frequent Patterns,” in Proceedings of the 24th IASTED international Conference on Database and Applications, Innsbruck, Austria, Feb 13-15, 2006, pp. 75-82.

[7] M. Schena, D. Shalon, R.W. Davis, P.O. Brown, “Quanti-tative monitoring of gene expression patterns with a com-plementary DNA microarray,” Science, vol. 270, pp. 467–470, 1995.

[8] S. Albelda, D. Sheppard, “Functional Genomics and Expression Profiling,” Am. J. Respir. Cell Mol. Biol., vol. 23, pp. 265-269, 2000.

[9] A. Schulze and J. Downward, “Navigating gene expres-sion using microarrays — a technology review,” Nature Cell Biology, vol. 3, pp. 190-195, 2001.

[10] S. Kotsiantis, D. Kanellopoulos, “Association Rules Mining: A Recent Overview,” International Transactions on Computer Science and Engineering, vol. 32, pp.71-82, 2006.

[11] M.H. Dunham, Data Mining Introductory and Ad-vanced Topics. Pearson Education, 2003.

[12] R. Agrawal, R.Srikant, “Fast Algorithms for Mining Association Rules in Large Databases,” in Proceedings of 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, September 12-15, 1994, pp. 487-499.

[13] J. Han, J. Pei, Y. Yin, “Mining Frequent Patterns without Candidate Generation,” ACM SIGMOD Record, vol. 29, pp. 1-12, 2000.

[14] B. Yıldız, B. Ergenç, “Comparison of Two Association Rule Mining Algorithms without Candidate Generation,” in Proceedings of the 10th IASTED international Conference on Artificial Intelligence and Applications, Innsbruck, Austria, Feb 15-17, 2010, pp. 450-457.

[15] C. Creighton, S. Hanash, “Mining gene expression databases for association rules,” Bioinformatics, vol. 19, pp. 79-86, 2003.

[16] J. Jamieson, “Oxidative stress responses of the yeast Saccharomyces cerevisiae,” Yeast, vol. 14, pp 1511-1527, 1998.