IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

Mining Order Preserving Submatrices using FP-Growth Algorithm

Yashi Bhardwaj[1], Anubhooti Papola[2]

M.Tech Scholar, CSE Department, Uttarakhand Technical University, Dehradun, India [1]

Assistant Professor, CSE, Uttarakhand Technical University, Dehradun, India [2]

Abstract: Analysis of gene expression data is a significant research field in DNA microarray research. Data mining methods have proven to be beneficial in understanding gene function; gene regulation, cellular processes and subtypes of cells.Microarrays have made it possible to cheaply gather large gene expression datasets. Biclustering has become established as a popular method for mining patterns in these datasets. Biclustering algorithms simultaneously cluster rows and columns of the data matrix; this approach is well suited to gene expression data because genes are not related across all samples, and vice versa. In the past decade many biclustering algorithms that specifically target gene expression data have been published. However, only a few are commonly used in bioinformatics pipelines. To break down the gene expression information, biclustering is generally used to assemble the articles into different groups in light of similarity. Elements in a similar group are as comparative as could be expected under the circumstances. Order Preserving Sub Matrices (OPSM) is highly used these days to analyze DNA microarray data. In this paper, we are proposing a method to find out all the possible OPSM's in the data. Then we find out all common subsequences (ACS), and for this we are going to use FP Growth algorithm to mine frequent sequential patterns.

Keywords: OPSM, microarray data, gene expression analysis, Apriori algorithm, FP-Growth Algorithm.

I. INTRODUCTION

Data mining is the process of analyzing data in a supervised or unsupervised manner to discover useful and interesting information that is hidden within the data. Many data mining approaches have been applied to genomics to aim at understanding the biological systems, by analyzing their structures as well as their functional behaviors. A DNA Microarray is a glass slide covered with a chemical product and DNA samples containing thousands of genes. By placing this glass slide under a scanner, we obtain an image in which colored dots represent the expression level of genes under experimental conditions [1]. This process can be summarized by figure 1 and this figure shows that Generation from a DNA microarray of an image where colored dots represent the expression level of genes under experimental conditions.

Recently developed DNA microarray technology has made it now possible for biologists to monitor simultaneously the expression levels of thousands of genes in a single experiment. Recent numerous high-throughput developments in DNA chips generate massive gene expression results, which are represented as matrix D of real numbers with rows (objects) to represent the genes and columns (attributes) to represent the different environmental conditions, different organs, or even different individuals. Each element or entry represents the expression level of a gene under a specific condition. To analyze the gene expression data, clustering is widely used to gather the objects into different clusters based on similarity. The objects in the same cluster are as similar as possible. Genes in the same cluster may show similar cellular function or expression mode, implying that they are more likely to be involved in the same cellular process. Similarity measurements are mainly based on distance functions, including the Euclidean distance and Manhattan distance.

However, these distance functions are not appropriate to measure the object correlation in the gene matrix. Moreover, only a small subset of genes participate in any cellular process of interest, and a cellular process occurs only in a subset of the samples, requiring biclustering or the subspace clustering to capture clusters formed by a subset of genes across a subset of samples. Biclustering of microarray data can be helpful to discover coexpression of genes and, hence, uncover genomic knowledge such as gene networks or gene interactions. Biclustering is an NP-hard problem [9]. This paper focuses on a model of clustering known as biclustering, which simultaneously clusters both the n objects and their m features. Unlike traditional clustering methods, which consider all features, biclustering algorithms can discover local patterns on only a subset of features. If the n objects are arranged into an m by n feature-object matrix, a bicluster consists of a subset of the m rows and a subset of the n columns. After row and column exchanges, the rows and columns of the bicluster define a contiguous submatrix of the data matrix .This paper focuses on determining common Subsequences using FP growth algorithm.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

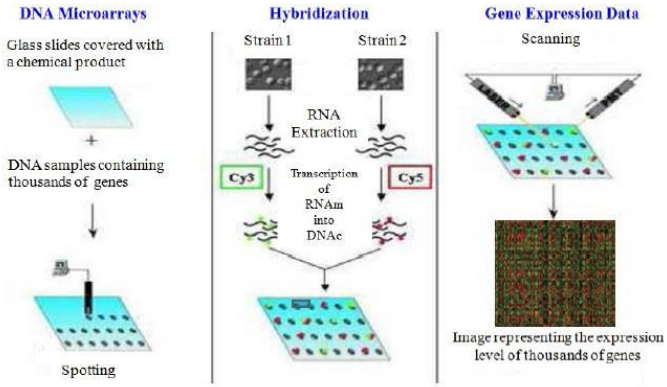

Figure 1 Generation from a DNA Microarray

II. RELATED WORK

The conventional order-preserving submatrix (OPSM) mining problem was motivated and introduced by Ben-Dor et al. [2] to analyze gene expression data without repeated measurements. They proved that the problem is NP hard. A greedy heuristic mining algorithm was proposed, which does not guarantee the return of all OPSM's or the best OPSM's.
Cheng and Church, [3] stated that they begins with an oversized matrix that is original knowledge, and iteratively masks out null values and biclusters that are discovered. Every biclusters is obtained by a series of coarse and fine node deletion, node addition, and also the inclusion of inverted knowledge. In different words, Cheng's work treats the full original knowledge set as a seed, and then they fight to refine it through node deletion and node addition, when refinement the ultimate biclusters are going to be covert with random knowledge. Then within the following iteration, it'll treat the full knowledge set as another seed and refine it once more, so on.

According to P. T. Spellman et al, [4] similarity measure between genes with protein-protein interactions is pro-posed. The chip-chip data are converted into the same form of gene expression data with pear-son correlation as its similarity measure. On the basis of the similarity measures of protein- protein interaction data and chip-chip data, the combined dissimilarity measure is defined. The combined distance measure is introduced into K-means method, which can be considered as an improved K-means method. The improved K-means method and other three clustering methods are evaluated by a real dataset. Performance of these methods is assessed by a prediction accuracy analysis through known gene annotations. Their results show that the improved K-means method outperforms other clustering methods. The performance of the improved K-means method is also tested by varying the tuning coefficients of the combined dissimilarity measure. The results show that it is very helpful and meaningful to incorporate heterogeneous data sources in clustering gene expression data, and those coefficients for the genome-wide or completed data sources should be given larger values when constructing the combined dissimilarity measure.

According to Andrea Califano et al [5] several microarray technologies that monitor the level of expression of a large number of genes have recently emerged. Given DNA-microarray data for a set of cells characterized by a given phenotype and for a set of control cells, an important problem is to identify ―patterns‖ of gene expression that can be used to predict cell phenotype. The potential number of such patterns is exponential in the number of genes. The author's, propose a solution to this problem based on a supervised learning algorithm, which differs substantially from previous schemes. It couples a complex, non-linear similarity metric, which maximizes the probability of discovering discriminative gene expression patterns, and a pattern discovery algorithm called SPLASH. The latter discovers efficiently and deterministically all statistically significant gene expression patterns in the phenotype set. Statistical significance is evaluated based on the probability of a pattern to occur by chance in the control set. Finally, a greedy set covering algorithm is used to select an optimal subset of statistically significant patterns, which form the basis for a standard likelihood ratio classification scheme.

According to Kerby Shedden and Stephen Cooper [6] Microarray analysis of gene expression during the yeast division cycle has led to the proposal that a significant number of genes in Saccharomyces cerevisiae are expressed in a cell-cycle-specific manner. Four different methods of synchronization were used for cell-cycle analysis. Randomized data exhibit periodic patterns of lesser strength than the experimental data. Thus the cyclic ties in the expression measurements in the four experiments presented do not arise from chance fluctuations or noise in the data. However, when the degree of cyclicity for genes in different experiments is compared, a large degree of non-reproducibility is found. Re-examining the phase timing of peak expression, we find that three of the experiments (those using α-factor,

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

CDC28 and CDC15 synchronization) show consistent patterns of phasing, but the elutriation synchrony results demonstrate a different pattern from the other arrest-release synchronization methods. Specific genes can show a wide range of cyclical behaviour between different experiments; a gene with high cyclicity in one experiment can show essentially no cyclicity in another experiment. The elutriation experiment, possibly being the least perturbing of the four synchronization methods, may give the most accurate characterization of the state of gene expression during the normal, unperturbed cell cycle. Under this alternative explanation, the observed cyclicities in the other three experiments are a stress response to synchronization, and may not reproduce in unperturbed cells.

According to Sara C. Madeira and Arlindo L. Oliveira [7] huge amount of clustering methods have been recommended for the study of gene expression data found from microarray experiments. Though, the consequences of the use of standard clustering methods to genes are restricted. These narrow results are executed by the presence of a number of experimental environments where the activity of genes is not correlated. A alike restriction occurs when clustering of conditions is performed. For this reason, a number of algorithms that complete instantaneous clustering on the row and column dimensions of the gene expression matrix have been proposed to date. This instantaneous clustering, typically chosen by biclustering, seeks to find sub-matrices, that is subgroups of genes and subgroups of columns, where the genes show highly associated activities for each condition. This type of algorithms has also been wished-for and used in other fields, such as info recovery and data mining. In their complete survey, they inspect a huge number of prevailing methods to biclustering, and categorize them in accord with the type of biclusters they can treasure, the patterns of biclusters that are learned, the approaches used to make the search and the target applications.

According to Fadhl M. Al-Akwaa, [8] bioinformatics is functional genomics; which concentrates on the interactions and functions of each gene and its items (mRNA, protein) through the whole genome (the entire genetics sequences encoded in the DNA and responsible for the hereditary information). In order to identify the functions of certain gene, the author‘s should able to capture the gene expressions which describe how the genetic information converted to a functional gene product through the transcription and translation processes. Functional genomics uses microarray technology to measure the genes expressions levels under certain conditions and environmental limitations. In the last few years, microarray has become a central tool in biological research. Consequently, the corresponding data analysis becomes one of the important work disciplines in bioinformatics. The analysis of microarray data poses a large number of exploratory statistical aspects including clustering and biclustering algorithms, which help to identify similar patterns in gene expression data and group genes and conditions in to subsets that share biological significance.

III. PROPOSED METHODOLOGY

This paper is based on Frequent Pattern growth algorithm. FP-Growth algorithm compresses the database into a frequent pattern tree (FP-tree) and still maintains the information of associations between data matrix. FP growth algorithm generates frequent item sets from FP-Tree by traversing in bottom up manner.

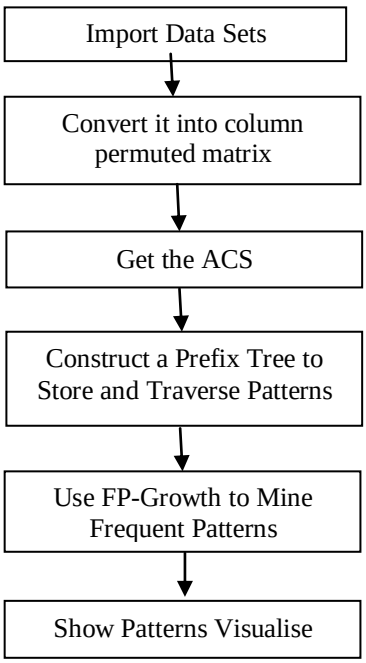

Figure 2 Architecture of proposed methodology

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

Figure 2 gives the overview of the proposed methodology. The data set of ecoli is considered as input to the system .All the dataset is collected from https://archive.ics.uci.edu/ml/datasets/ecoli which is used as an input to the system. In the end, the system will generate the all common subsequences of ecoli genes dataset.

The details about the steps adopted in the methodology are described in the following subsections.

1. Import Data Set: The proposed methodology uses ecoli data set to perform gene expression analysis. E. coli (Escherichia coli) is the name of a germ, or bacterium, which lives in the digestive tracts of humans and animals. The dataset of ecoli is collected from: https://archive.ics.uci.edu/ml/datasets/ecoli

2. Convert It Into Column Permuted Matrix: A column permutated matrix is a square matrix obtained from the same size identity matrix by a permutation of rows. In this step the columns are sorted row wise and then the column permuted matrix where values of the sorted matrix are replaced by their column values depending on their original index number.

3. Get the ACS (All Common Subsequences): The All common subsequence (ACS) problem is the problem of finding the all subsequence common to all sequences in a set of sequences (often just two sequences) and then all common subsequences find of the opsm taking two rows simultaneously.

4. Construct a Prefix Tree to Store and Traverse Patterns: In this step a prefix tree of genes is generated to store the all common subsequences which is already obtained. We use a prefix tree to store and traverse all common sequences. Different from the traditional method to solve OPSM problem, frequent common subsequences can be obtained by traversing frequent prefix tree rather than by the columns joint. The prefix tree, also known as trie, is an ordered tree used to store strings or associative arrays, in which the nodes from the root to the leaf form a path.

5. Use FP-growth to Mine Frequent Patterns: Generating the FP-Tree from this database will lead to a tree containing the sums of the OPSM values in the nodes.
Our proposed work is to find the frequent patterns from gene expression data using FP-growth algorithm which is the enhanced version of Apriori. FP-growth algorithm constructs the conditional frequent pattern (FP) tree and performs the mining on this tree. FP-tree is extended prefix tree structure, storing crucial and quantitative information about frequent sets. FP-growth method transforms the problem of finding long frequent patterns to search for shorter once recursively and then concentrating the suffix.

6. Show Patterns Visualise: Finally, we validate our proposed model by comparing our model with existing apriori models by considering various parameters.

PROPOSED ALGORITHM

1. Import dataset
2. Convert the data into column permuted matrix.
3. If 'Data' is the data matrix and i, j are the row and column sizes, then the column permuted.

Data matrix can be form in the following way:

```
For i=1: r
for j=1: c
if (Data (i, j) ==sorted (i,1))
                    for each item ai in Q do {// Mining multipath     FP- tree
```

- generate pattern ß = ai ∪ a with support = ai .support;

```
ColumnM (i, 1) =j;
end
if (Data (i, j) ==sorted (i,2))
ColumnM (i, 2) =j;
end
if (Data (i,j)==sorted(i,3))
ColumnM (i, 3) =j;
end
if (Data (i,j)==sorted(i,4))
ColumnM (i, 4) =j;
```

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

```
 end
 if (Data(i,j)==sorted(i,5))
 ColumnM (i,5)=j;
 end
 if (Data (i,j)==sorted(i,6))
ColumnM (i, 6) =j;
end
if (Data (i,j)==sorted(i,7))
ColumnM (i, 7) =j;
end
end
end
```

The process is backtracked again to find the longest common subsequences

4. Constructing prefix tree and Using FP Growth to find sequential patterns
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;
- 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)))

}

IV. RESULT AND DISCUSSIONS

In the proposed work, frequent sequential pattern of genes are mined. To mine the frequent sequential patterns, frequent pattern–growth algorithm is implemented. With the help of this algorithm: a prefix tree is generated, biclusters are found, visualization of biclusters in obtained, OPSM is found and then the statistical chart of the overlap distribution is obtained which shows that our proposed algorithm proposed better results as compared to previous results.

At first, the values are sorted in ascending order, and then the column value of each row is replaced with values. This matrix is called column permuted matrix. After that, all Common Subsequences (ACS) are found in the dataset. A prefix tree is used to store and traverse the ACS. A prefix tree of genes data matrix is generated to store the all common subsequences which is already obtained. Prefix tree is a special type of tree used to store associative data structures. Our proposed work implemented FP-Growth algorithm and then finally performance was evaluated by finding Overlap percentages.

The algorithms are implemented on MATLAB 2016a version.
Dataset used: Ecoli Data Set
Source: https://archive.ics.uci.edu/ml/datasets/ecoli

Figure 3 depicts the Prefix Tree generated Using FP Growth algorithm, figure 4 depicts the visualization of bicluster using FP-Growth algorithm, figure 5 shows the linearity of biclusters, figure 6 shows the statistical chart on the overlap distribution.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

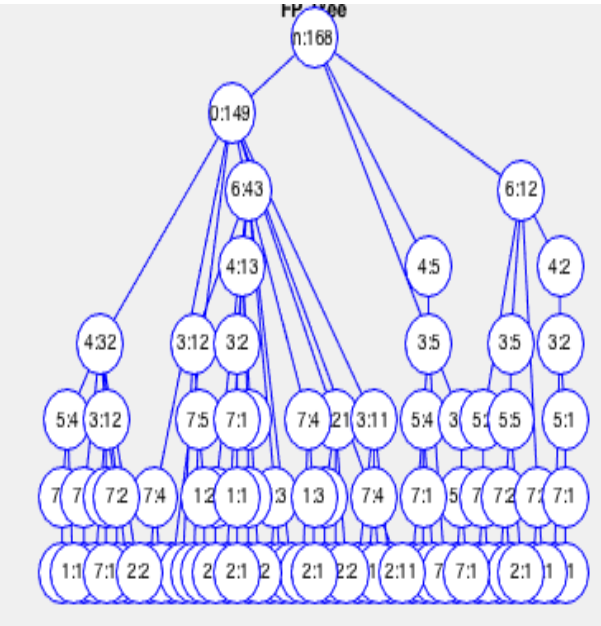

figure 3 Prefix tree using FP-Growth

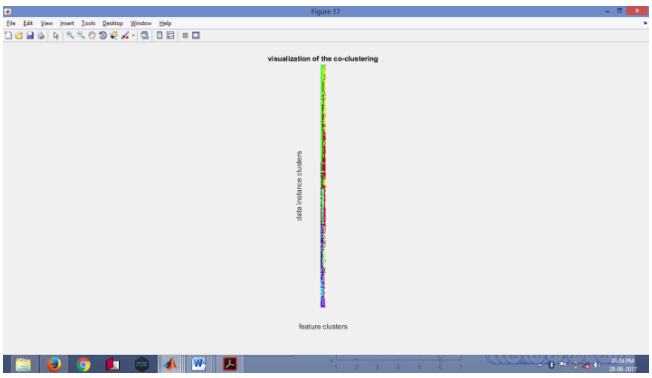

Figure 4 Visualization of Bicluster using FP-Growth

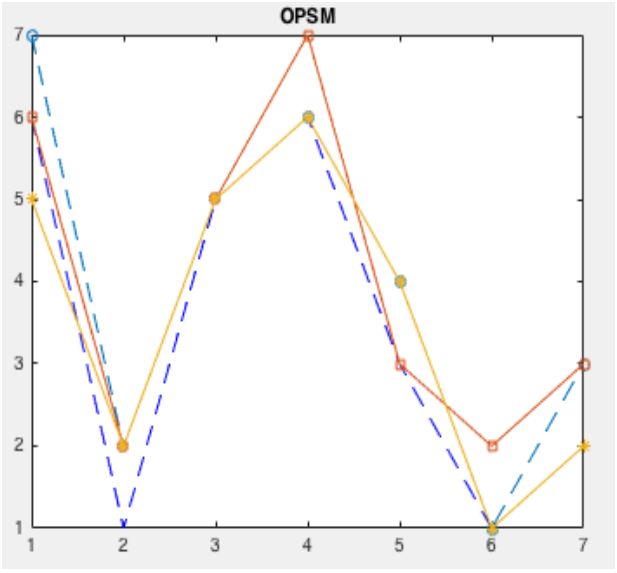

Figure 5 OPSM found using FP –Growth

The performance evaluation is calculated using Overlap Time Comparison. These measures are collected to determine effectiveness of the proposed work.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

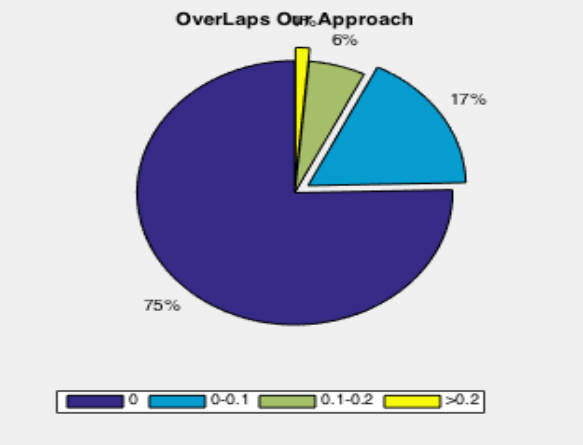

Figure 6 Statistical chart of the overlap distribution.

A) Overlap

Let $G1$, $G2$ be two gene sets in biclusters. The overlap of $G1$ and $G2$ is their intersection divided by their union, and means module identity and 0 means no overlap.

Overlap is calculated by using this formula:

$$S_G(G1, G2) = \frac{|G1 \cap G2|}{|G1 \cup G2|}$$

Figure 6 shows that no-overlap biclusters accounted for 75% of the total, and the degree of overlap between 0 and0.1 (excluding 0).Therefore, the biclusters whose overlap was between 0 and 0.1 (including0) accounted for 92%.

V. CONCLUSION AND FUTURE SCOPE

Biological validation of biclusters of microarray data is one of the most important open issues. In proposed work, very simple yet very effective method proposed to find minimal and optimal subset of genes of well- known microarray data sets, i.e., E. coli data set is used. A new approach is proposed that exploits frequent pattern mining to deterministically generate an initial set of good quality biclusters. In proposed work FP growth algorithm is implemented, and tested it on the Ecoli data set. The result of proposed work shows that, OPSM along with FP growth principle can produce better quality biclusters than apriori in comparable running time. The performance evaluation results show that the overlap was found in our proposed work is 92%.Experimental results indicate that the proposed technique is effective in performing its tasks.

The tool currently supports Ecoli datasets which can be extended with other microarray datasets. To include other type of datasets, the database schema can be modified .As future enhancement; the proposed technique may be extended by using other algorithms to find more overlapping frequent patterns instead of using FP-growth which is proposed to find frequent patterns from gene expression dataset. Other feature selection methods can be included to improve this work.

REFERENCES

[1] Hartigan JA (1972). "Direct clustering of a data matrix". Journal of the American Statistical Association. 67 (337): 123–129. doi:10.1080/01621459.1972.10481214).
[2] Ben-Dor, B. Chor, R. M. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: the order-preserving submatrix problem," Journal of Computational Biology, vol. 10, no. 3-4, pp. 373–384, 2003.
[3] Y. Cheng and G.M. Church. "Biclustering of expression data." In Proc Int Conf Intell Syst Mol Biol. 2000, pages 93-103, 2000.
[4] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," Molecular Biology of the Cell, vol. 9, no. 12, pp. 3273–3297, 1998.
[5] Andrea Califano, Gustavo Stolovitzky, and Yunai Tu. Analysis of gene expression microarrays for phenotype classification. In Proceedings of the International Conference on Computational Molecular Biology, pages 75–85, 2000.
[6] Kerby Shedden and Stephen Cooper "Analysis of cell-cycle gene expression in Saccharomyces cerevisiae using microarrays and multiple synchronization methods" Nucleic Acids Research, 2002, Vol. 30, No. 13.
[7] Sara C. Madeira and Arlindo L. Oliveira "Biclustering algorithm for biological data analysis: A survey" NESC-ID TEC. REP. 1/2004, JAN 2004.
[8] Fadhl M. Al-Akwaa "Analysis of Gene Expression Data Using Biclustering Algorithms" INTECH 2012http://dx.doi.org/10.5772/48150.
[9] Y. Cheng and G.M. Church. "Biclustering of expression data." In Proc Int Conf Intell Syst Mol Biol. 2000, pages 93-103, 2000.
[10] Sudhakar Venanti and Dasari Rajrsh "Discovering Biological Associations in Microarray Data using OPSM-RM" International journal of scientific engineering and technology research (IJETR) SSN 2319-8885 Vol.03, Issue.29 October-2014, Pages: 5745-5749.
[11] Peng sun, Nora k Speicher, Richard Rottger, Jiong Guo and Jan Baumbach" Bi-Force: large-scale biclusters editing and its application to gene expression data biclustering" Vol. 42, No. 9 20 March 2014 doi: 10.1093/nar/gku201.
[12] Haifa BEN SABER and Mourad Elloumi " a comparative study of clustering and biclustering of microarray data " International Journal of Computer Science & Information Technology (IJCSIT) Vol 6, No 6, December 2014.