International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 01 | Jan-2016 www.irjet.net p-ISSN: 2395-0072

CCC-BICLUSTER ANALYSIS FOR TIME SERIES GENE EXPRESSION DATA

[1]D.Soundaravalli [2] S.Thilagavathi

[1] Associate Professor, Computer Science and Engineering, Aksheyaa College of Engg.,Tamilnadu,India
[2]Assistant Professor, Computer Science and Engineering, Aksheyaa College of Engg.,Tamilnadu,India

---

Abstract - *Many of the biclustering problems have been shown to be NP-complete. However, when they are interested in identify biclusters in time series expression data, it can limit the problem by finding only maximal biclusters with contiguous columns. This restriction leads to a well-mannered problem. Its motivation is the fact that biological processes start and conclude in an identifiable contiguous period of time, leading to increased (or decreased) activity of sets of genes forming biclusters with adjacent columns. In this context, propose an algorithm that finds and reports all maximal adjacent column coherent biclusters (CCC-Biclusters), in time linear in the size of the expression matrix. Each relevant CCC-Bicluster identified corresponds to the detection of a coherent expression pattern shared by a group of genes in a adjacent subset of time-points and identifies a potentially relevant regulatory module. The linear time complexity of CCC-Biclustering is acquired by manipulating a discretized version of the gene expression matrix and using efficient string dealing out techniques based on suffix trees.*

The effectiveness was obtained by applying the algorithm to the transcriptomic expression patterns stirring in Saccharomyces cerevisiae in response to heat stress.

Key Words*— *Biclustering, gene expression data, biology computing

1. INTRODUCTION

It has the primary function to display clusters (the output of any clustering algorithm) computed from experimental data in a suitable manner that will aid in their interpretation. The assumption here is that experiments are presented under the form of a large matrix where each line represents a variable over a series of conditions (the columns). The data is usually clustered and the acquired clusters are used for interpretation. Beside the choice of the clustering algorithm, program, parameters need to be attuned (for instance, filtering parameters or numbers of nodes for Self-Organizing Maps).

In this method an integrated algorithm is used to group the clusters into a tree, each tree level being an experiment. In corresponding, a secondary tree is constructed with the gene membership. The primary goal is to analyze the manners of a clustering algorithm applied on experimental microarray data that increases the number of basis vectors. To demonstration conclusions, genes can be displayed along with their IDs, descriptions, and ontology.

Generally, the cellular processes move toward on subsets of genes to be co-regulated and co-expressed only under certain experimental conditions, but to behave almost separately under other conditions. Discovering such local expression patterns may be the key to uncover many genetic mechanisms that are not apparent. As a result, bi-clustering algorithms have been presented as an different approach to standard clustering techniques to identify local structures from gene expression data sets, such as mining coherent patterns, inferring global regulatory networks, uncovering statistically significant sample subclasses, and associated marker genes.

The efficiency of the method is validated by two benchmark gene data sets: Human organs and the yeast Saccharomyces cerevisiae. In order to explore the statistical and biological characteristics of bi-clusters, we compute the coherence index to demonstrate the performance of the algorithm. It also use gene ontology (GO) terms of the genes in different bi-clusters to infer the biological processes, molecular functions, and cellular components of the genes.

2. OBJECTIVE

Extracting biologically relevant information from DNA microarrays is a very important task for drug development and test, function annotation, and cancer diagnosis. Various clustering methods have been proposed for the analysis of gene expression data, but when analyzing the large and heterogeneous collections of gene expression data, predictable clustering algorithms often cannot produce a satisfactory solution. Bi-clustering algorithm has been presented as an another approach to standard clustering techniques to identify local structures from gene expression data set.

3. APPLICATIONS

Cancer Diagnosis

In cancer research, where microarrays are most widely used, clustering has successfully been used to classify different cancer types, and identify new cancer types which were not recognized in the original data.

Pharmaceutical Industry

There are three major tasks with which the pharmaceutical industry deals on a regular basis:

(1) to discover a drug for an already definite target,
(2) to assess drug toxicity, and
(3) to monitor drug safety and effectiveness.

Clustering can help in all the above tasks. Drug safety, effectiveness and toxicity also may be examined through the use of clustering the microarray.

4. EXISTING SYSTEM

Explore how to locate the low-score correlated bi-clusters on gene expression data. The proposed technique uses singular value decomposition as its framework which will be introduced in the first part. The process of bi-clustering technique will be discussed in the second part. The whole process includes three steps: The first step, based on the property of SVD, the problem of identifying correlated bi-clusters from gene expression matrix is transformed into two global clustering problems. The second step, the Bidirectional Mixed Clustering algorithm motivated by agglomerative hierarchical clustering is applied to discover the original bi-clusters which are not mutually exclusive. The last step, based on original bi-clusters, the inclusion-maximal biclusters are revealed by Lift algorithm. About the Lift algorithm, it is similar with the node-deletion and node-addition algorithm proposed. In our algorithm, Use our defined residue score to obtain correlated bi-clusters.

4.1 UNDERLYING TECHNOLOGIES

4.1.1 SVD (singular value decay)

Singular value decay (SVD) and non-negative matrix factorization (NMF) are two important matrix decomposition methods for microarray data analysis.

In this method we use an Singular value decay (SVD) to decompose the gene expression matrix into a cluster of basis genes and a group of basis conditions. The spectral bi-clustering method proposed by Kluger et al., only finds the distinctive "checkerboard" pattern which is a special type of our algorithm. It only utilizes the eigenvectors corresponding to the first two or three eigen values in matrices of gene expression data while our method uses the eigenvectors related to the eigen values that account for most of energy. It can also identify non-overlapping bi-clusters with checkerboard structure while our method can discover arbitrarily positioned overlapping bi-clusters.

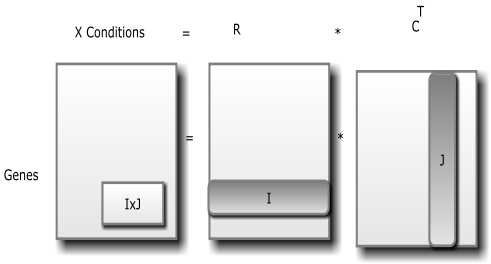

Fig 1.1 SVD Processing

4.1.2 Mixed Clustering

Algorithm: Mixed Clustering

Input: $\Omega = \{w_1, w_2, \ldots\ldots, w_N\}$ a sample set; D, the intergroup dissimilarity; k, the minimal number of one cluster.

Output: t, the number of clusters, and the clusters G1; G2; . . .;Gt. // initialization

The initial clusters G1; G2; . . .; GN, where $G_i := \{w_i\}$;

Repeat For i: = 1 to N do

For cluster Gi, find the closest observation xi from other observations $\Omega \setminus G_i$

Compute the corresponding distance di, and get the object set $G_i\left[x_i^*, d_i^*\right]$

End

Compute the smallest value d^* and the corresponding index j^* from $d_1^*, d_2^*, \ldots\ldots, d_N^*$

If $\left(d^* \le D\right)$ then (// Euclidean distance)

$G_{J*} := G_{j*} \bigcup x_{j*}^*$

If (G_{J*} has identical observations with one of other Clusters) then get rid of the cluster G_{J*}

End

End

Until $\left(d^* \ge D\right)$

4.1.3 Lift Algorithm

In the first step, we calculate the correlation score S for each possible row/ column deletion and choose the action that decreases score S the lot. The process is iterated until S≤ δ . In the second step, we add other potential nodes and compute the score S for each possible row/column addition and choose the node which corresponds to the smallest score S. If S≤ δ , this node is added. The process is iterated until no node can be added. The biclusters obtained in this way are inclusion maximal.

Algorithm: Lift Algorithm

Input: X, a gene expression matrix; $\delta \in (0,1)$ the maximum acceptable correlation score; I and J signifying an original sub matrix of X; nc, a minimal number of conditions; nr, a minimal number of genes.

Output: I' and J' J0 with the property that the score $s(I', J') \leq \delta$

While $(s(I, J) \succ \delta$ do // Single-Node Deletion

Compute the score S_{iJ} for all $i \in I$, the score S_{Ij} for all $j \in J$;

Find the row $i \in I$, with the largest S_{iJ} and the column $j \in J$ with the largest S_{Ij};

If (SiJ > SIj) then Remove the row i from I;
Else remove the column j from J;
End
// other potential nodes Addition

While $(s(I, J) \succ \delta$ do // Single-Node Addition
Compute SiJ for all i but not in I, SIj for all j but not in J;
Get the S for each possible row/column addition;
Choose the node which corresponds to the smallest S;
If ($S \leq \delta$) then add this node.
End

If $(\#I \geq nr AND \#J \geq nc)$ then Return the final I and J as I' and J'.

5. PROPOSED SYSTEM

The efficiency can be improved by using a CCC-Biclustering (Contiguous column coherence) algorithm and time series gene expression data (i.e. A special type of gene expression data obtained from microarray experiments performed in successive time periods) in terms of the number of the Biclusters.

The central idea of this approach is based on the relation between adjacent column coherent Biclusters (CCC-Biclusters) and nodes in a generalized suffix tree constructed using Ukkonen's algorithm. After performing a simple alphabet transformation, which appends the column number to each symbol in the matrix (as a pre processing step in the algorithm), all nodes in the generalized suffix tree constructed with the set of strings equivalent to each row in the transformed matrix correspond to CCC-Biclusters.

The proposed system includes three modules namely

1) Preprocessing and Dicretization
2) Biclustering
3) Gene annotation

5.1 Preprocessing

Pre-processing includes
- Image Analysis
- Filtering
- Normalization

a) Image Analysis
Convert raw data to useful biological data that is image data to intensities values. Pre-processing can be used to distinguish noise and the actual biological data and also able to compare data from multiple arrays.

b) Filtering
Remove data that will contribute to noise or bias. K-nearest neighbor algorithm can identify other genes with expression most similar to the genes of interest (Euclidean distance). Here weighted typical values for those genes are used to estimate the missing values.

c) Normalization
Aims to correct dissimilarities in intensities between samples

5.2 BICLUSTERING

5.2.1 Ukkonen's algorithm

Ukkonen's algorithm established by constructing an implicit suffix tree *Ti* for each prefix of *S*[1..*i*] of a string *S*, starting from *T*1 and incrementing *i* by one until *T*|*S*| is built, where |*S*| is the number of characters in *S*. The true suffix tree from *S* is then constructed from *T*|*S*|. Ukkonen's algorithm to construct suffix trees uses the concepts of *implicit suffix tree* and *suffix link* to attain a linear time construction

An *implicit suffix tree* for a string *S* is a tree acquired from the suffix tree *T* constructed for the string *S*$ by removing every copy of the symbol $ from the edge labels of the tree, then removing any node *v* that does not have at least two children

- **Text**: S[1..*m*]
- *m* phases
  Phase *j* is divided into *j* extensions

The suffix tree [3] for the string S of length n is defined as a tree such that

- The paths from the root to the leaves have a one-to-one association with the suffixes of S
- Edges spell non-empty strings,
- All internal nodes (except possibly the root) have at least two children.

5.3 GENE ANNOTATION

Analyzing the raw sequence of a genome and describing relevant genetic and genomic features such as genes, mobile elements, repetitive elements, duplications, and polymorphisms

6. CONCLUSION

CCC-Biclustering can provide three additional extensions that identify biclusters with shifted/scaled, anti-correlated and time lagged patterns. Sometimes, distinct genes demonstrate similar expression evolutions at different expression levels, thus not shiny a similar pattern after discretization. This problem is addressed by identifying biclusters with shifted patterns.

Anti-correlation allows genes with conflicting expression patterns, in a set of consecutive time points, to be included in the same bicluster. The time lagged move toward identifies genes that exhibit similar expression patterns starting at dissimilar time points, enabling the detection of activation/inhibition delays.

So, Thus the efficiency of the algorithm is enhanced by using time series data set and CCC-biclustering algorithm by concentrating on contiguous columns.

REFERENCES

[1] *A. Hartigan (1972), "Direct Clustering of a Data Matrix," J. Am. Statistical Assoc., vol. 67, no. 337, pp. 123-129.*

[2] *Y.H. Zhao, J.X. Yu, G.R. Wang, L. Chen, B. Wang, and G. Yu (2008), "Maximal Subspace Co- regulated Gene Clustering," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 1, pp. 83-98.*

[3] *D. Gusfield(1997) Algorithms on strings, trees, and sequences. Computer Science and Computational Biology Series. Cambridge University Press.*

[4] *Y. Zhang, H. Zha, and C.H. Chu (2005), "A Time-Series Biclustering Algorithm for Revealing Co-regulated Genes," Proc. Int'l Conf. Information Technology: Coding and Computing (ITCC '05), pp. 32-37.*

[5] *Sara C. Madeira, Miguel C. Teixeira, Isabel S´a-Correia and Arlindo L. Oliveira(2007) Supplementary Material to TCBB Manuscript: Identification of Regulatory Modules in Time-Series Gene Expression Data using a Linear Time Biclustering Algorithm.*

[6] *Sara C Madeira and Arlindo L Oliveira (2009) A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series.*