Clustering of Time-Course Gene Expression Data

Ya Zhang[1], Hongyuan Zha[2], James Z. Wang[3], Chao-Hsien Chu[4]
The Pennsylvania State University, University Park, PA 16802

Keywords: Microarray, gene expression, time series, clustering

1 Introduction.

Microarray experiments have been used to measure genes' expression levels under different cellular conditions or along certain time course. Initial attempts to interpret these data begin with grouping genes according to similarity in their expression profiles. The widely adopted clustering techniques for gene expression data include hierarchical clustering, self-organizing maps, and K-means clustering. Bayesian networks and neural networks have also been applied to gene clustering. Sharan & Shamir [3] provided a survey on this topic. Clustering techniques typically discover the inherent structure of the genes expression profiles based on some similarity measures. The clustering results largely depend on how the similarity measure corresponds to the biological correlation between genes. Before reliable conclusion about biological functions can be drawn from the data, the gene clusters obtained from microarray analysis must be investigated with respect to known biological roles of those clusters.

The current analysis of whole-genome expression focuses on relationships based on global correlation over a whole time-course, identifying clusters of genes whose expression levels simultaneously rise and fall. However, genes may be regulated by different regulators in a long time course. Co-regulating in part of the long time course does not guarantee a global similarity in gene profiles.

Biclustering of microarray gene expression data has recently been introduced by Chen & Church [1] as a means to discover sets of genes that co-expressed in only part of the experiment conditions under study. Essentially, overlapping in gene clusters is allowed, and many subtle gene clusters are revealed. Since then, several other algorithms have been developed to bicluster gene expression data [4]. However, existing biclustering algorithms do not consider the differences between time-series gene expression data and multi-condition gene expression data. The relations between time points are ignored, and the time points are clustered independently. It is marginally biologically meaningful if two genes show similar expression pattern in non-consecutive time points. It is therefore necessary to preserve the time locality in time-course gene expression data.

2 Method.

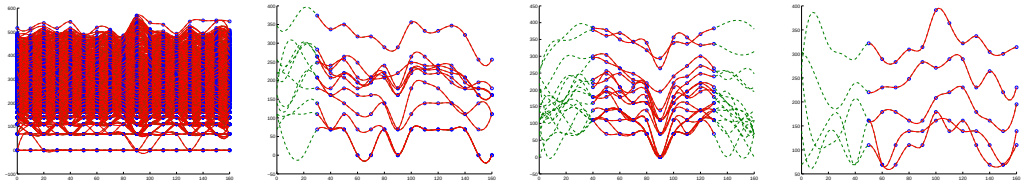
We present our time series biclustering algorithm to cluster time course microarray data. The aim of this clustering is to discover genes that are co-regulated in an interim of the time course but do not show highly correlated gene expression over the whole time course. The mean square residue score H is used as a measurement for the biclustering. While enforcing H to be smaller than a user-selected parameter δ, we try to simultaneously maximize H,

[1]School of Information Sciences and Technology. E-mail: yzhang@ist.psu.edu
[2]Department of Computer Science and Engineering. E-mail: zha@cse.psu.edu
[3]School of Information Sciences and Technology. E-mail: jwang@ist.psu.edu
[4]School of Information Sciences and Technology. E-mail: chu@ist.psu.edu

the number of genes, and the length of the time course in the cluster. The bicluster is first initialized as the entire data matrix. We adopt a deletion-based method to eliminate genes with expression profiles deviating from those of the majority. A row (gene) is removed from the bicluster if the ratio of the mean square residue score of the row to that of the bicluster larger than a user-defined threshold. Similarly, time points are removed. To ensure that the time points in a bicluster are always consecutive, in time point deletion, we only allow the deletion to be exerted on the border time points – the first and the last column in the bicluster. The deletion has demonstrated the capability to reduce the mean square residue score of the resulting bicluter [1]. The deletion stops when the mean square residue score of the resulting bicluter is less than δ. Some previously deleted genes may have strong correlation with genes in the bicluster in terms of similarity in the interim of expression profile in the bicluster. These genes are then add into the bicluster, and it is guaranteed that the addition will reduce the mean square residue score. Similarly, the time points adjacent to the border of the bicluster may be considered for addition into the bicluster.

3 Results and Discussion.

We test our algorithm on the yeast cell cycle data provided by [2]. Figure 1 presents some clusters obtained by our time series biclustering algorithm. The solid lines represent the interims of gene profiles that are in the biclusters, and the dash lines represent the deleted time points. Clearly, the variability of expression profile in the biclustered range is smaller than the that in the range of deleted time points. By deleting some time points in the two ends, we are able to discover some subtle genes clusters. However, further investigation of the gene clusters with respect to known biological roles of cluster members is desired. Further experimental confirmation may be required to reveal the true 'biological relationships' among genes.



cluster 1 cluster 35 cluster 59 cluster 83

Figure 1: Expression profile of gene clusters.

References

[1] Cheng,Y., and Church,G. 2000. Biclustering of expression data. In *Proceedings of 8th International Conference on Intelligent System for Molecular Biology (ISMB)*, pp.93-103.

[2] Cho,R.J., Campbell,M.J., Winzeler,E.A., Steinmetz,L., Conway,A., Wodicka,L. et al. (1998) A genome-wide transcriptional analysis of the cell cycle. *Mol. Cell,* 2, pp.65-73.

[3] Sharan,R. and Shamir,R. 2002. Algorithmic approaches to clustering gene expression data. In T. Jiang *et al.* (eds), *Current Topics in Computational Molecular Biology.* The MIT Press, pp. 269-300.

[4] Yang,J., Wang,H., Wang,W., and Yu,P. 2003. Enhanced biclustering on expression data. In *Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 321-327.