Discovering Distinct Patterns in Gene Expression Profiles

Li Teng[1,2] and Laiwan Chan[1]

[1]Department of Computer Science and Engineering The Chinese University of Hong Kong, Hong Kong

Summary

Traditional analysis of gene expression profiles use clustering to find groups of coexpressed genes which have similar expression patterns. However clustering is time consuming and could be difficult for very large scale dataset. We proposed the idea of Discovering Distinct Patterns (DDP) in gene expression profiles. Since patterns showing by the gene expressions reveal their regulate mechanisms. It is significant to find all different patterns existing in the dataset when there is little prior knowledge. It is also a helpful start before taking on further analysis. We propose an algorithm for DDP by iteratively picking out pairs of gene expression patterns which have the largest dissimilarities. This method can also be used as preprocessing to initialize centers for clustering methods, like K-means. Experiments on both synthetic dataset and real gene expression datasets show our method is very effective in finding distinct patterns which have gene functional significance and is also efficient.

1 Introduction

Microarray techniques allow the expression level of thousands of genes to be monitored in paralle [1, 2, 3, 4, 5]. The resulted microarray data are expression profiles of genes under different experimental conditions. They provide characteristics information about the expressed genes (functioning genes) of a genome or a biological probe. Microarray data has properties of high noise, high variance, high dimensionality and high correlations [6, 7, 8]. A typical microarray data set would have thousands of genes and a very small number of conditions.

Clustering is one of the first steps in data analysis of high-throughput expression measurements. A variety of techniques (hierarchical clustering [9], K-means [10], Self-Organizing Maps [11]) have been implemented and successfully used on analyzing high-dimensional gene expression data. Since gene expression profiles are encoded in real vectors, these algorithms intend to group gene expression vectors that are sufficiently close to each other (according to certain distance or similarity measurement). However, most algorithms have several notable weaknesses. Algorithms such as K-means and Self-Organizing Maps require the predefinition of the number of clusters which is usually unknown in advance. Hierarchical clustering suffers from lack of robustness, nonuniqueness. It is not practical to do clustering on very large dataset. And the idea of forcing each gene into a cluster is a significant drawback of these implementations. Adaptive Quality-based Clustering [12] and DSF_Clust [13] were proposed to do adaptive clustering on the genes. They only find clusters of gene that are highly correlated but ignore the other genes that scatter in data space. However, those genes which have been excluded always have typical patterns which need special care. Finding genes with similar expression patterns would be helpful for gene function prediction based on their temporal association with genes of

[2]To whom correspondence should be addressed. E-mail: lteng@cse.cuhk.edu.hk

known function. While, when there is little prior knowledge, it is difficult to pick up interesting genes from large dataset. And it is time consuming or even intractable for any unsupervised learning analysis.

Discovering Distinct Patterns (DDP) means to find typical gene expression patterns which show different variations and are representative in the whole gene expression profiles. Since genes show different patterns in the biological process, the patterns they show are important for analysis of their functions. There would be many genes sharing similar expression patterns. However, the number of different patterns and the number of genes sharing the similar patterns is unknown in advance. If we discover the different patterns existing in the dataset it will be much easier to find groups of genes with similar patterns. The pattern shared by a group of genes is crucial for the analysis of theirs functions, while, the size of a group is not important. So instead of grouping genes into clusters with similar patterns by clustering we find collection of distinct gene expression patterns which are typical in the whole set of genes. And the distinct patterns can be regarded as representatives for other genes with similar expressions patterns. It is useful to find a good start and scale-down the cost for gene function analysis. In some other work [14, 15] it is called prototype finding which has a similar idea. And the distinct patterns we find can be further used to construct gene regulatory network [16, 17].

In this paper we propose a method of finding distinct gene expression patterns by iteratively picking out pairs of genes which have the lowest similarities. The algorithm is tested on both synthetic data and real gene expression data for its effectiveness and efficiency.

In the following sections, we first present the algorithm and how we handle large scale dataset by a divide and conquer scheme in section 2. Experiments on both synthetic dataset and real gene expression datasets were given in section 3. We make conclusions in section 4.

2 Algorithm

We aim to find distinct patterns in the whole dataset. Those patterns are picked up from the dataset and are supposed to be representative for all patterns. To reduce redundancy they should have very low similarities between each other. We propose a DDP algorithm by iteratively picking out the pair of genes which have the lowest similarity and we regard them as the distinct patterns when threshold is satisfied. Detail will be discussed in the following. Firstly we introduce the format of gene expression profiles and the similarity matrix in section 2.1. Details of the algorithm are given in section 2.2-2.3.

2.1 Gene expression profiles

Gene expression profiles could be represented by an $n \times m$ numerical data matrix $G = (g_{ij})$. Each row of G stands for a gene and each column stands for a condition (different time points or samples). Element g_{ij} stands for the expression of gene i under condition j. There are usually thousands of genes compared to dozens of conditions. It is believed that genes with related functions tend to have similar expression patterns [2]. Similarities of gene expression patterns could be represented by an $n \times n$ matrix $A = (a_{ij})$ as Fig. 1 shows. We call it the similarity matrix in this paper. And a_{ij} stands for the similarity between gene i and gene j.

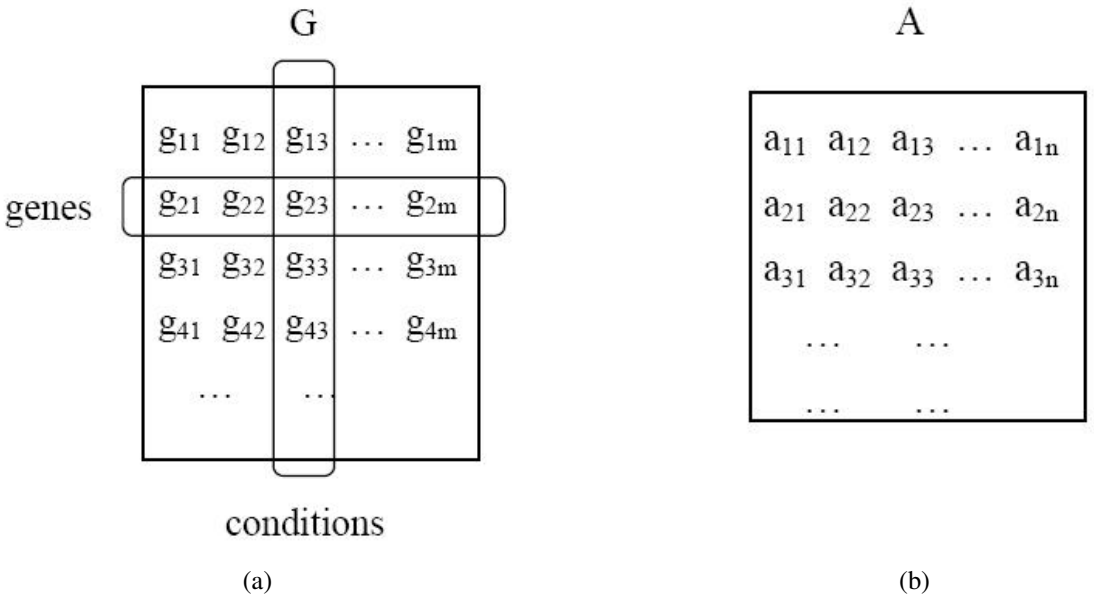


Figure 1: (a) Original gene expression profiles where each row stands for a gene and each column stands for a condition. (b) Similarity matrix for the genes.

There are several often used similarity/dissimilarity measurements for gene expression data. We use Pearson correlation coefficient in our experiments. Correlation coefficient between gene i and j is,

$$r(g_i, g_j) = \frac{\sum_{k=1}^{m}(g_{ik} - \overline{g_i})(g_{jk} - \overline{g_j})}{\sqrt{\sum_{k=1}^{m}(g_{ik} - \overline{g_i})^2}\sqrt{\sum_{k=1}^{m}(g_{jk} - \overline{g_j})^2}} \quad (1)$$

, where $\overline{g_i}$ and $\overline{g_j}$ is the average expression level over all conditions. This similarity measurement depends only on the trend and not on the absolute magnitude of two expression vectors.

2.2 Discovering Distinct Patterns by Iteratively Picking out the Largest Dissimilarities

The basic idea is that each time we find the pair of genes which have the lowest similarities among all genes. We record them as the distinct patterns and delete the genes which have high similarities with them. Iteration goes on when necessary. There are two thresholds, θ_1 and θ_2 to be defined. Only when the similarity between two genes is smaller than θ_1 the two genes can be regarded as distinct patterns. And only when two genes have similarity higher than θ_2 we regard them as similar patterns. The following are the main steps.

Step 1. Compute the similarity matrix A for all genes. This is the start point of our algorithm.

Step 2. Pick up the minimum item a_{ij} from the similarity matrix A. If $a_{ij} < \theta_1$, add gene i and j to the collection of distinct patterns. If $a_{ij} > \theta_1$, stop the algorithm and output the distinct patterns.

Step 3. Update the similarity matrix as following, If $a_{ip} > \theta_2$ for any $a_{ip} \in A$, mark gene p as neighbor of gene i. If $a_{jq} > \theta_2$ for any $a_{jq} \in A$, mark gene q as neighbor of gene j. Update matrix A by deleting all rows and columns corresponding to the neighbors of gene i and gene j. Then go back to step 2 if A is not empty.

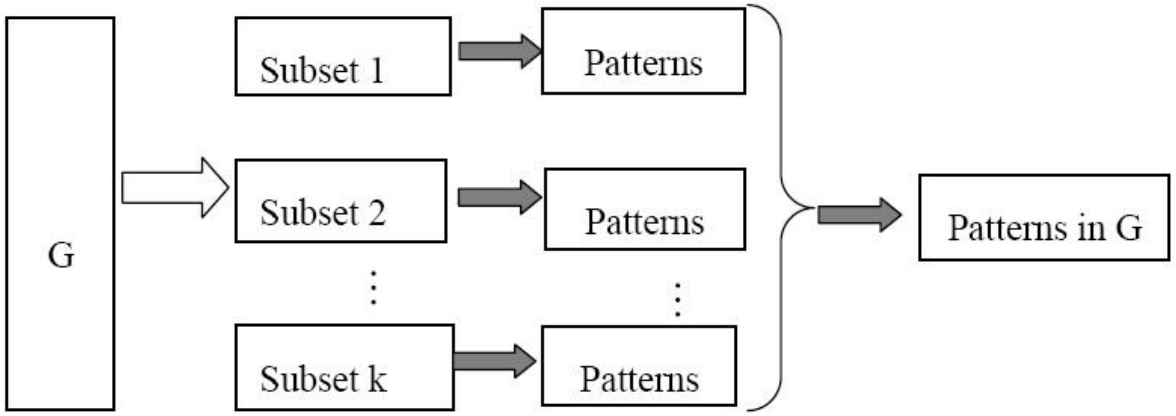


Figure 2: Handling large dataset by divide and conquer. The gray arrows show when the DDP algorithm is carried out.

Other similarity/dissimilarity measurements can be easily fitted to this algorithm. For example when using Euclidian distance, one just has to change the signs of inequality which appears in step 2 and 3. Little prior knowledge is needed to set the value of θ_1 and θ_2. They are only relevant to the average similarity among the genes. Generally when two genes expression patterns have a correlation coefficient higher than 0.85, they are regarded as having similar expressions. Increase the value of θ_2 will include more patterns in result distinct patterns since the criterion on similar patterns becomes more restrict. And setting θ_1 to any value around 0 would not apparently affect the result. And also one can stop the algorithm when getting enough distinct patterns. And no matter when the algorithm is stopped the most significant distinct patterns will be output first.

2.3 Handling large dataset by divide and conquer

With very large dataset it could be impractical to compute the similarity matrix for all samples, especially when we are using a common PC. We have to divide the problem into smaller sub-problem before it can be handled. Fig. 2 shows the procedure of how we handle large dataset in DDP. G is the original dataset. Firstly, the large dataset is partitioned into some smaller pieces. Then the algorithm above is taken on every pieces of G to find distinct patterns for each subset respectively. And we find distinct patterns for the original dataset by performing the algorithm on a collection of distinct patterns for all the subsets. The gray arrows show when to carry out the processes of DDP. If the large dataset is divided into k parts then the processes would be taken for $k + 1$ times. The size of the squares in the figure is not proportional to the size of the data.

Theoretically as iteration goes on, all distinct patterns in the dataset could be found. Pair of patterns with lowest similarities would be discovered first regardless of the threshold settings. Cost for the whole algorithm is hard to estimate since number of iteration varies a lot for different datasets. Suppose the largest partition of G is a $s \times m$ subset (s equals to n if no partition happens). In step 1 computing similarity matrix A requires time in $O(s2)$ and this only take place for once, where s is the number of genes of the biggest partition of G. In step 2 finding the minimum item of A in the first iteration is an $O(s2)$ effort. Updating similarity matrix A in the first iteration is an $O(s)$ effort. Cost for step 2 and 3 drops dramatically in after

iterations since size of A decreases very quickly. Space complexity of the whole algorithm is $O(s2)$ which is very small comparing to most of the clustering methods. Similar patterns could exist in the collection of distinct patterns for all partitions after the first implementation was taken as Fig. 2 shows. However, the similar patterns would be excluded after the second step. And distinct patters in a globe view would not be excluded in local views. So this scheme can guarantee a result comparable to that from single implementation on the whole dataset.

3 Experiment

The algorithm is tested with both synthetically generated datasets and real gene expression datasets for effectiveness and efficiency. The experiments are implemented with MATLAB and executed on a PC with a 3.2 GHz and 0.99 GB main memory.

3.1 Synthetic dataset

The experiment on synthetic dataset is to verify that our algorithm is able to find distinct patterns which are representative for the dataset. For easy visualization we firstly constructed a 3D dataset with 60 samples. The samples in the dataset can be grouped into 6 clusters as Fig. 3 shows. Distinct patters we found are marked with squares. The distinct patterns are representative for the 6 groups in the dataset. Then we generated 3 random datasets in 1D, 2D and 3D, respectively. Each of the datasets has 100 samples. In these experiments we use Euclidean Distance to measure the similarities between samples. The results on the three datasets are shown in Fig. 4(a), (b) and (c), respectively, where the distinct patterns were marked with squares as well. The distinct patterns well scattered in the space and representative for the global distribution of the dataset as the result shows.

3.2 Real gene expression profiles

3.2.1 Dataset of rat heart cell cultivation

Firstly we use a rat dataset (not open for public access right now) from Li Ming et al.. This dataset measures more than 31000 genes on 6 time points across the period of stem cells growing into heart cells. We picked up 28 genes with know functions without absent values. Firstly, hierarchical clustering with complete linkage and single linkage was implemented on the dataset, respectively. Fig. 5(a) and (b) shows the result.

In hierarchical clustering we can find clusters by cutting the tree at a certain level. Here we cut the tree at 0.05 as the red horizontal line shows. The gene which has the largest dissimilarity with the genes in adjacent cluster is regarded as the center of a cluster. We found 7 clusters (with centers {3, 5, 8, 10, 21, 24}) by single linkage and 12 clusters (with centers {1, 3, 8, 11, 14, 15, 16, 21, 23, 24, 25}) by complete linkage. And we found 8 distinct patterns ({5, 8, 10, 11, 14, 21, 23, 24}) when $\theta_2 = 0.95$. Most of the 8 distinct patters overlap with the centers of the clusters from hierarchical clustering. This verifies the representativeness of the patterns we found.

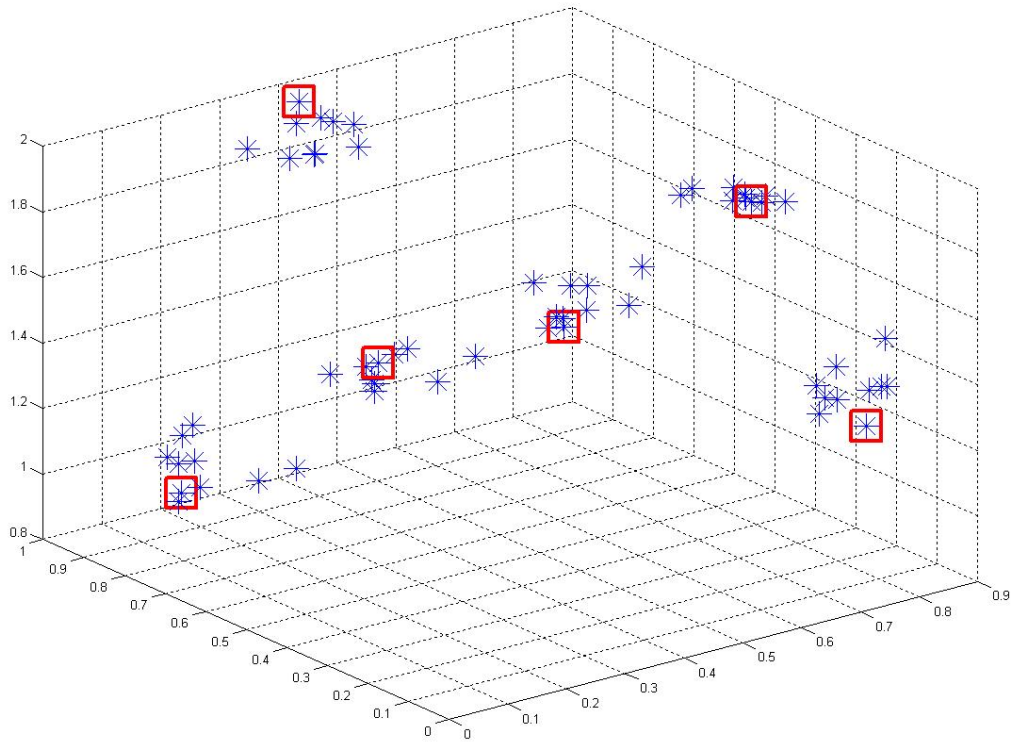
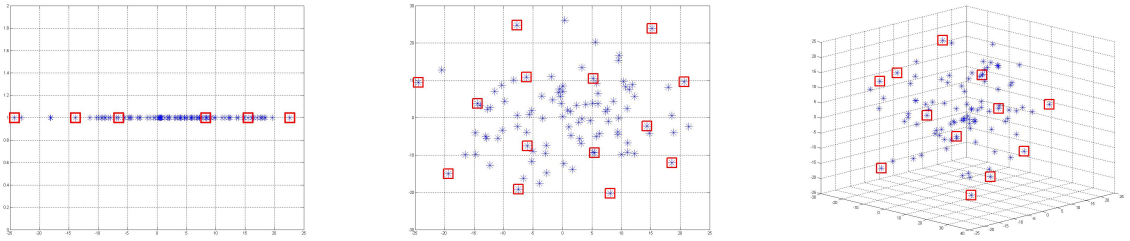


Figure 3: Result on synthetic dataset. Stars are the 3D samples and squares are the distinct patterns we found.

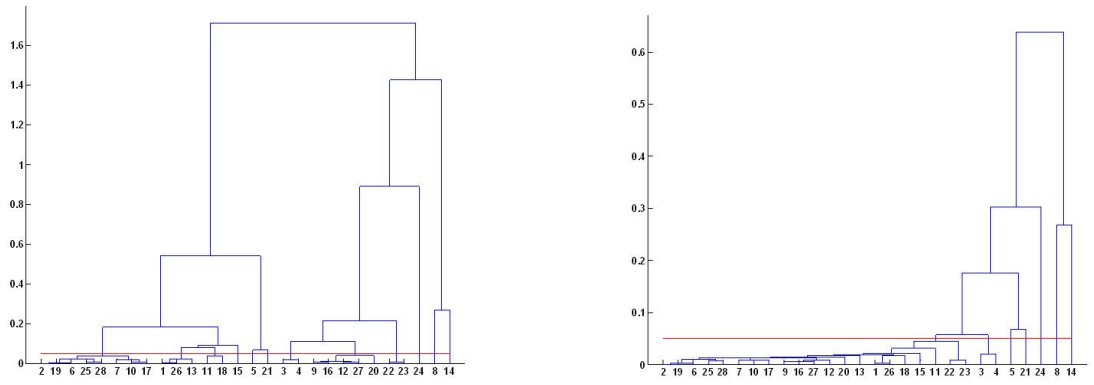


(a) (b) (c)

Figure 4: Result on synthetic dataset. Stars are the 3D samples and squares are the distinct patterns we found. (a) result for 1D dataset, (b) result for 2D dataset, (c) result for 3D dataset.

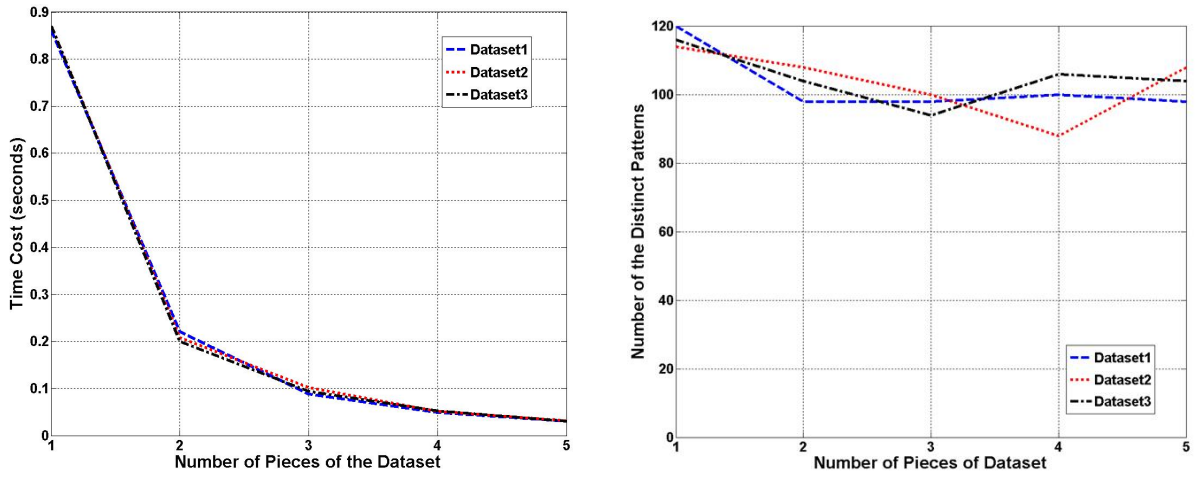
3.2.2 Sporulation data on budding yeast

We used the sporulation data on budding Yeast [2] to test the performance when large dataset is partitioned into different number of pieces. This dataset measures the expression of 6118 genes on 7 time points. The dataset was firstly divided into 3 subsets Dataset1 (2039 genes), Dataset2 (2039 genes) and Dtaset3 (2040 genes) to test the consistency. The same implementations were carried on the three datasets respectively. Multiple partitions were tried on them (number of partition varies from 1 to 5 and 1 means no division took place).



(a) (b)

Figure 5: Result of hierarchical clustering on 28 genes of rat data. (a) with complete linkage, (b) with single linkage. The red line cut the tree at 0.05 and partition the gene into several clusters.



(a) (b)

Figure 6: (a) Time cost for finding the distinct patterns when increasing the number of partitions, (b) Number of different patterns found when increasing the number of partitions.

Fig. 6(a) shows the time cost for experiments on three datasets. The three lines have exactly the same trend. It shows that when the dataset was divided into more pieces with smaller size, running time for finding the distinct patterns will drop significantly. Fig. 6(b) shows the number of distinct patterns we found in each experiment. The total number of distinct patterns decreases a little when dataset was partitioned into more pieces. But there is no apparent relationship between the number of distinct patterns and the number of partitions. And the average similarity among the distinct patterns for all cases is very low (range from -0.009 to 0.003), which means the distinct patterns we found have totally unrelated expressions.

We selected a subset of 1600 genes which have higher variance than the others. High variance across the conditions suggests the gene has significantly expressed during the cell cycle. The first 20 distinct patterns we found by using our method was showed in Fig. 7. The left subfigure on the first row shows the seven distinct temporal patterns of induction defined by Chu. et al. [2]. These seven temporal patterns are selected mainly by the time the gene peaked during the cell cycle and Chu. et al. have assigned them to 7 cell stages as Metabolic, Early I, Early

II, Early-Mid, Middle, Mid-Late and Late, respectively. Our algorithm is able to find very similar patterns to all those seven temporal patterns. There are 6 distinct patterns matched with the 7 temporal patterns with Pearson correlation coefficient no lower than 0.9. They were shown in the subfigures of Fig. 7. with titles specifying the matching. Since in [2] the patterns for Mid-Late and Late have high similarity of 0.86 they match to the same distinct pattern YGR044C in our result. Among the other 14 distinct patterns, some of them have already been verified to have different functions, such as YJR069W for transport cell wall, YKL106W for amino acid transport and metabolism, YJR096W for cotactor biosynthesis, YNL239W for starvation and YBR166C for Aromatic et.. Some are still under research and shows interesting patterns, such as YOR007C, which has highly prohibited patterns across the whole sporulation. It's reasonable to believe that these distinct patterns can be a good start to study. This result shows that our algorithm is capable of finding genes in different functional categories. And our methods find more than 7 distinct patterns. This is helpful to sophisticate the cell stages. Since we have reduced the number of genes by a large number and kept the most significant expressed genes patterns at the same time, the result can be further used to construct gene regulatory network.

3.2.3 Dataset of mouse cerebellum development

We use the time series gene expression data generated by Kagami et al. [18]. The data is publicly available through GEO repository [19]. The dataset consists 5 series of expression levels for 897 genes. Those are only genes Kagami et al. have selected as being significantly expressed during the development of mouse cerebellum. Among the 897 genes, there are 450 genes with clear function description. They were clustered into 7 functional categories: CGDD (Cell Growth, Differentiation, and Death), CLAM (Carbohydrate, Lipid, and Amino acid Metabolism), CSC (Cell Structure and Communication), IMTN (Intra/Inter-cellular Molecular Transport and Neurotransmission), NNM (Nucleotide and Nucleic Acid Metabolism), ST (Signal Transduction) and TTPM (Transcription, Translation, and Protein Modification). The rest are either EST (expressed Sequence Tags) or genes unclassified. Table 1 shows the profiles of the 450 genes. Setting θ_2 as 0.8, our method was able to quickly find 6 genes belonging to 6 different categories from the 450 genes, respectively, as Fig. 8 shows. Genes in the CSC category was not discovered in the first 10 patterns. However, genes in the CSC category appear to be required throughout the development of cerebellum [18]. They show a rather steady expression pattern across the 5 stages and this makes them harder to be distinguished from genes in other categories.

4 Conclusion

In this paper we propose the idea of discovering distinct patterns in gene expression profiles. For large scale gene expression data with little prior knowledge, it is difficult and could be aimless on beginning any analysis. Usually there will be a lot of genes sharing similar expression patterns. How many genes are sharing similar expression patterns is not as important as what patterns they are sharing. It will be a good start to find the distinct patterns in the dataset. The distinct patterns are representative for the whole dataset and could reveal the mechanism of genes with different function. And reducing the number of significant genes would be great

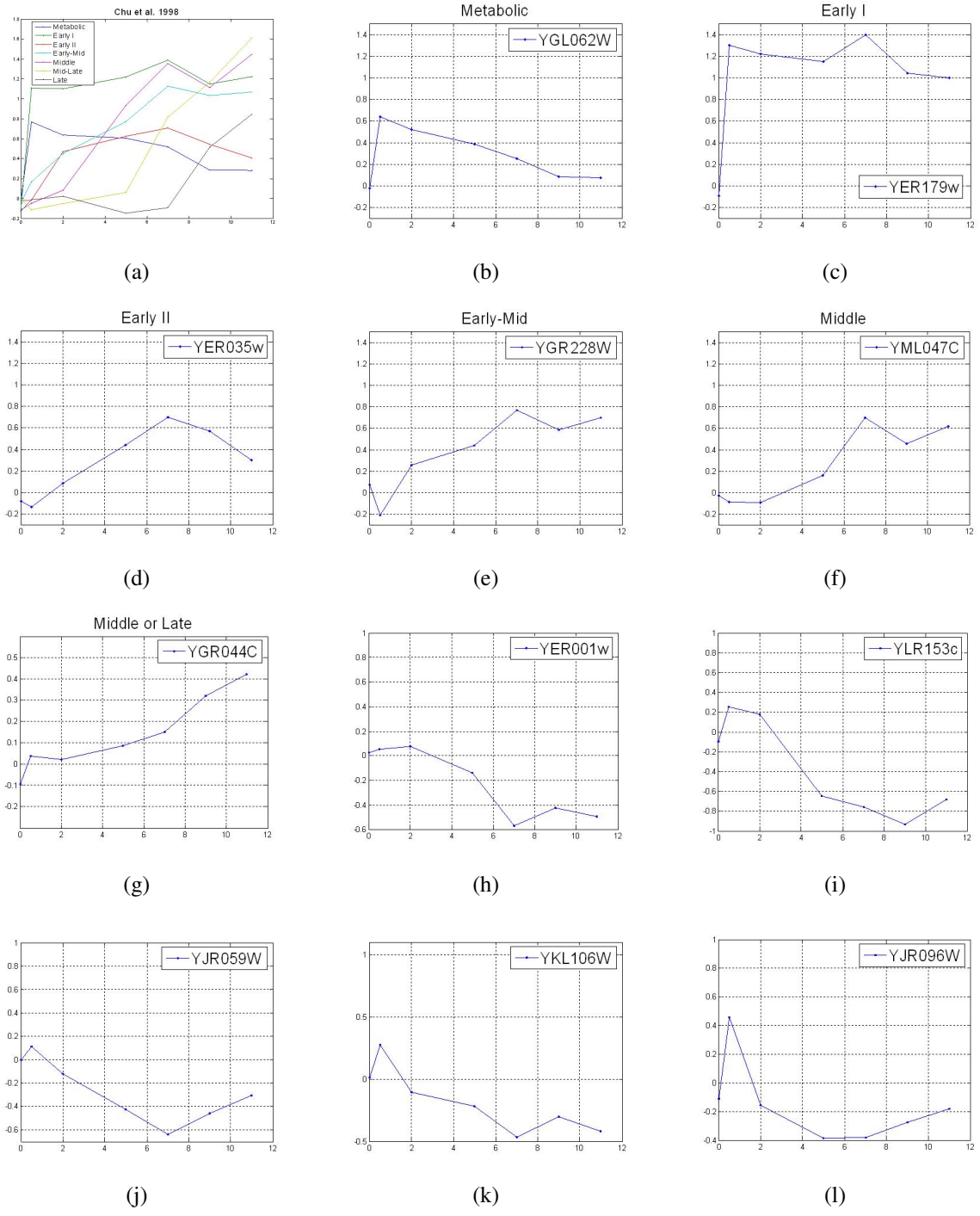


Figure 7: The first 20 distinct patterns we found in yeast data. The left subfigure on the first row shows the seven temporal patterns defined by Chu. et al. [2]. Distinct patterns which have the best match to the corresponding temporal patterns were shown with a title.

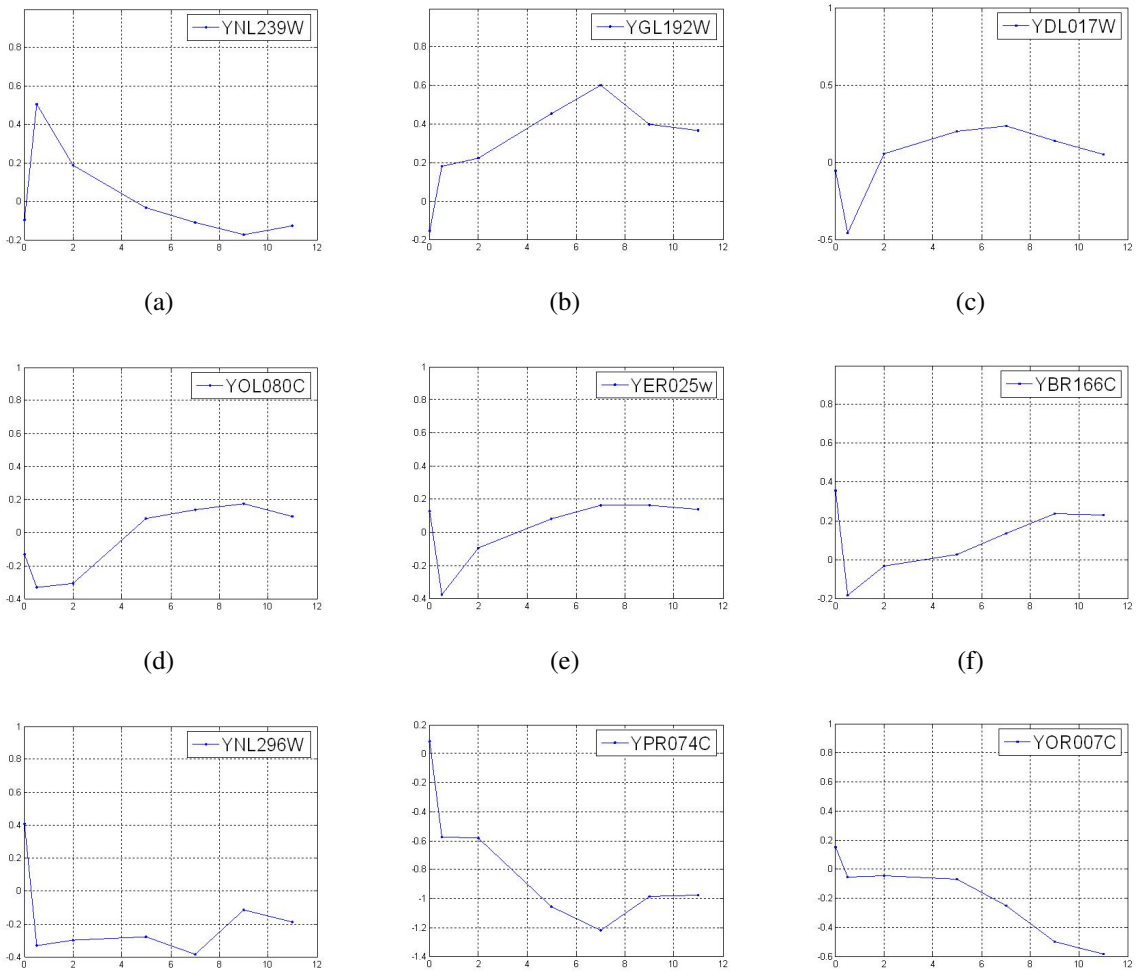


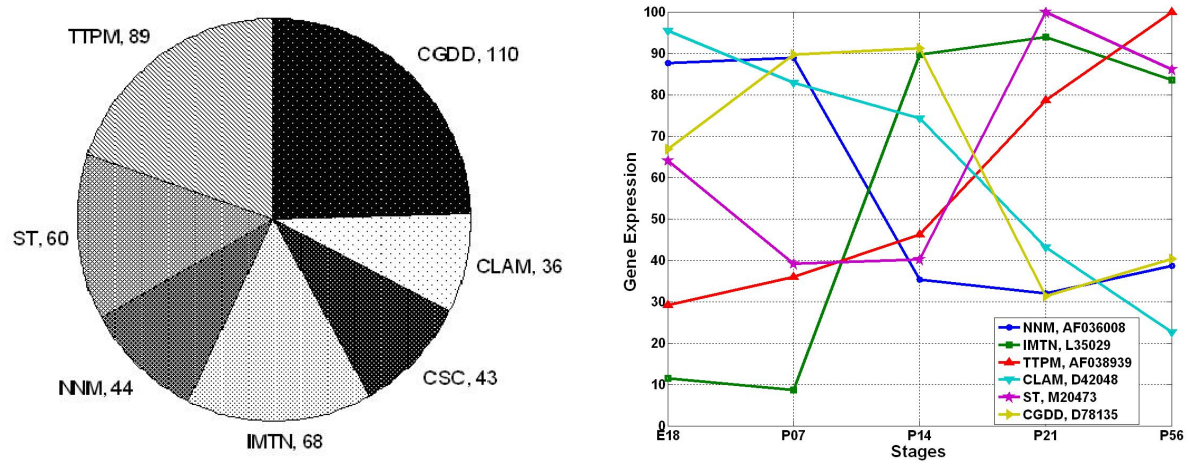
Figure 8: Figure 7 (continued)

helpful for constructing gene regulatory networks. An algorithm is then proposed for the DDP problem. By iteratively picking up pairs of genes with lowest similarities, distinct patterns can be found very effectively. Very large dataset can be handled through a divide and conquer scheme by which we implement the algorithm on each partition of the original dataset respectively. The effectiveness and efficiency of our algorithms is tested. Experiment on synthetic dataset shows our algorithm is capable of finding distinct patterns which are representative for the underlying clusters in the dataset. Results on the rat data were compared with that from hierarchical clustering to show the effectiveness in finding typical patterns. Experiment on the budding yeast dataset shows that when partition happened on the dataset, time cost would decrease a lot and little effect would happen to the result. Comparison with Chu. et al. [2] and result on mouse cerebellum development data shows the effectiveness in finding functional significant genes.

By finding distinct patterns we reduce the problem of large gene expression profiles analysis to a treatable scale before further work can be done. Also the distinct patterns can be used as the initial centers for clustering methods, like K-means, so that time cost for clustering would be reduced greatly. Since similarity between genes is the only basis for our algorithm, it

	CGDD	CLAM	CSC	IMTN	NNM	ST	TTPM	Size
Cluster1	4	4	2	2	0	4	1	17
Cluster2	10	4	5	6	3	2	8	38
Cluster3	10	1	1	9	2	2	9	34
Cluster4	45	14	23	26	20	30	36	194
Cluster5	27	8	10	19	9	16	26	115
Cluster6	10	5	0	4	8	3	8	38
Cluster7	4	0	2	2	2	3	1	14

Table 1: Profile of the 450 genes from the mouse cerebellum development dataset



(a) (b)

Figure 9: (a) Profile of the genes in 7 function groups, (b) the first 6 distinct genes which belong to the 6 function groups respectively.

could mistakenly include unimportant outliers into the result distinct patterns. In future work the distribution of the whole dataset could be taken into consideration to increase efficiency and avoid outlier. And the distinct patterns can be further used to construct gene regulatory network.

Acknowledgments

The authors thank Professor Ming Li from Hong Kong Institute of Biotechnology for providing the rat data. We also thank Xu Song and other colleagues for their valuable suggestions on the experiments and the previews.

References

[1] R. J. Cho, J. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis. A genome-wide transcription analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.

[2] S. Chu, J. DeRisi, M. Eisen, J. Mulbolland, D. Botstein, P. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998.

[3] D. J. Lockhart, H. Dong, M. Byrne, M. Follettie, M. Gallo, M. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.

[4] J. D. Risi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genome scale. *Science*, 278:680–686, 1997.

[5] L. Wodicka, H. Dong, M. Mittmann, M. Ho, and D. Lockhart. Genome-wide expression monitoring in Saccharmoyces cerevisiae. *Nature Biotechnology*, 15:1359–1367, 1997.

[6] D. Hwang, W. Schmitt, and G. Stephanopoulos. Determination of minimum sample size and discriminatory expression patterns in microarray data. *Bioinformatics*, 18:1184–1193, 2002.

[7] W. Pan, J. Lin, and C. Le. How many replicates of arrays are required to detect gene expression changes in microarray experiments? A mixture model approach. *Genome Biology*, 3:research0022.1–0022.10, 2002.

[8] G. K. Smyth, Y. H. Yang, and T. Speed. Statistical issues in cDNA microarray data analysis. *Methods in Molecular Biology*, 224:111–36, 2003.

[9] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

[10] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[11] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences*, 96(6):2907–2912, 1999.

[12] F. D. Smet, J. Mathys, K. Marchal, G. Thijs, B. De Moor, and Y. Moreau. Adaptive quality-based clustering of gene expression profiles. *Bioinformatics*, 18:735–746, 2002.

[13] X. Fu, L. Teng, Y. Li, W. Chen, Y. Mao, I. F. Shen, and Y. Xie. Finding dominant sets in microarray data. *Frontiers in Bioscience*, 10:3068–3077, 2005.

[14] C. L. Chang. Finding prototypes for nearest neighbor classifier. *IEEE Transactions on Computers*, 23(11):1179–1184, 1974.

[15] L. Wai, C. K. Keung, and D. Liu. Discovering useful concept prototypes for classification based on filtering and abstraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1075–1090, 2002.

[16] L. F. A. Wessels, E. P. Van Someren, and M. J. T. Reinders. A comparison of genetic network models. In *Pacific Symposium on Biocomputing*, pages 508–19, 2001.

[17] E. P. van Someren, L. F. Wessels, and M. J. Reinders. Linear modeling of genetic networks from experimental data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, volume 8, pages 355–366, 2000.

[18] Y. Kagami and T. Furuichi. Investigation of differentially expressed genes during the development of mouse cerebellum. *Gene Expression Patterns*, 1(1):39–59, 2001.

[19] National Center for Biotechnology Information. Gene Expression Omnibus. `http://www.ncbi.nlm.nih.gov/geo/`.