Clustering of Gene Expression Data: Performance and Similarity Analysis

Longde Yin and Chun-Hsi Huang
Dept. of Computer Science & Engineering, University of Connecticut, Storrs, CT 06269, USA
yin@engr.uconn.edu; huang@engr.uconn.edu

Abstract

Recent advances of the DNA Microarray technology allow monitoring gene expression profiles of thousands of genes simultaneously. However, the analysis and handling of such fast growing data is becoming the major bottleneck in the utilization of the technology. Clustering analysis is one of the most effective methods for analyzing such gene expression data. In this paper we first experimentally study three major clustering algorithms: Hierarchical Clustering, Self-Organizing Map (SOM), and Self Organizing Tree Algorithm (SOTA), using Yeast Saccharomyces cerevisiae gene expression data, and compare their performance. Then, we present a data mining tool, Cluster Diff, which allows the similarity analysis of clusters generated by different algorithms. A case study is conducted based on clusters generated by SOTA and SOM.

Keywords: Clustering algorithms, Gene expression, Microarray, Cluster Similarity Analysis, Performance study

1. Introduction

Microarrays are one of the latest breakthroughs in experimental molecular biology. This technology permits the analysis of gene expression, DNA sequence variation, protein levels, tissues, cells and other biological and chemicals in a massive format [1, 17]. However, the analysis and handling of such fast growing data is becoming one of the major bottlenecks in the utilization of the technology. Powerful mathematical and statistical methods are therefore called for this purpose to search for orderly features and logical relationships in such data.

Several clustering methods (algorithms) have been proposed for the analysis of gene expression data, such as hierarchical clustering [14], self-organizing maps (SOM) [13], and k-means approaches [28]. Although many of the proposed algorithms have been reported to be successful, no single algorithm has emerged as a method of choice, and the issues of determining the “correct” number of clusters and the choice of “best” algorithm has yet to be solved [25].

In this paper we first experimentally study three major clustering algorithms: Hierarchical Clustering, Self-Organizing Map (SOM), and Self Organizing Tree Algorithm (SOTA) [12], using Yeast Saccharomyces cerevisiae gene expression data and compare their performance. Then, we present a data mining tool, Cluster Diff, which allows the similarity analysis of clusters generated by different algorithms. A case study is conducted based on clusters generated by SOTA and SOM.

The rest of this paper is structured as follows. In the next section we briefly describe the three clustering algorithms. In section 3 we compare the implementations of each algorithm, their performance, and clustering results. Section 4, we present a data mining tool, Cluster Diff, and applied this tool for the cluster similarity analysis between those clusters identified by SOTA and SOM. Section 5 concludes this paper.

2. Clustering Algorithms

Clustering methods, which determine the natural subgroups in a data set, have some advantages over other methods, because no previous knowledge is necessary for clustering analysis [19, 20]. Several clustering algorithms have been proposed in past few decades [2, 3, 4, 10, 11, 13, 14, 15, 16, 27, 28, 29]. In this section, we briefly describe three such methods, including the classic hierarchical clustering methods, the Self-Organizing Map (SOM) neural networks, and the Self-Organizing Tree Algorithm (SOTA).

2.1. Hierarchical Clustering Methods

Hierarchical clustering methods are useful for analyzing gene expression data as well as many data in other contexts. They are agglomerative (bottom-up) approaches [14]. The clustering process starts with each gene as an individual cluster. These clusters are then
successively merged together to form new, larger clusters until all of the genes are in one big cluster. The sequence of clusters is represented by a hierarchical binary tree, the *dendogram* [9], which can be cut at a specific hierarchy level to obtain a desired number of clusters.

The topology of the clusters is a binary tree. During the clustering process, the number of cluster can only be reduced. The hierarchical clustering methods are deterministic, for each gene will be assigned to one and only one cluster. A great number of clusters will be produced, which is a valuable feature for data structure discovery. The clustering process will also produce an order for the genes, and the order is informative for gene display. However, the order of genes is not unique, because the two branches of each cluster can be switched without any problem. These methods also have some disadvantages. For example, the optimal merge of two clusters at each step may lead to a sub-optimal overall cluster hierarchy. Because of the deterministic characteristics of the hierarchical clustering methods, a bad assignment made earlier cannot be corrected. The detail algorithm

2.2. Self-Organizing Map (SOM) Neural Network

SOM [10] is a neural network with a number of nodes or neurons. Usually the configuration of these nodes is rectangular or hexagonal [15, 21]. The nodes have an associated vector of the same length of the input data. All nodes have initial random values and these reference vectors are adjusted during the training process. After the network is stable, these reference vectors are used to group the genes based on the closeness of the genes to the reference vectors.

During the training stage, the strength of the updating the reference vectors depends on their distances to the winner vector, which is the closest vector to a randomly selected gene. The training length, the training rate, and the size of the updating neighborhood can be customized. Usually the training is performed in two phases: the first one is the ordering phase (strong training rate and large updating radius) and the last one is the fine-tuning phase (long training length with a weak training rate and a smaller radius).

The SOM clustering method is non-deterministic, owing to the random order in which genes are used to move the reference vectors. It is not sensitive to gene outliers (noises), because the effects of outliers can be counter-balanced or corrected through the input of other genes. Once the configuration for partitions of the decision space is chosen, the number of clusters is determined and is fixed during the rest of clustering process. The k-means clustering methods also have a fixed, pre-determined number of clusters at the beginning. However, the Self-Organizing Map method is different in that the cluster centers are restricted to lie in a one or two-dimensional manifold (the decision space).

2.3. Self-Organizing Tree Algorithm (SOTA)

Contrary to the hierarchical clustering methods, which are agglomerative clustering methods, the Self-Organizing Tree Algorithm is a divisive (top-down) clustering method [12, 16, 24]. It starts the clustering process with a binary tree consisting of a root node with two leaves, each of which represents one cluster. The self-organizing process then grows the tree by converting the leaf with the largest resources into a node and attaching two new leaves to it. The resource value for each cluster is defined as the mean value of the distances between the cluster and the genes associated with it.

The Self-Organizing Tree Algorithm combines the tree structure of hierarchy clustering methods and the neural network structure of Self-Organizing Maps for adjusting the cluster vectors. Similar to the SOM algorithm, the SOTA [12] algorithm is non-deterministic and not sensitive to gene outliers (noises). The topology of the clusters is a binary tree, which is similar to that of the hierarchical algorithm except that the number of clusters can only grow. Furthermore, the number of clusters can be customized using the SOTA method by stopping the self-organizing tree growth process after a specific number of loops. Therefore, the SOTA algorithm is more flexible than the hierarchical clustering method and SOM.

3. Performance Study

For gene expression data analysis, we should consider the size of the dataset and the noise contained in the data. Both SOM and SOTA are based on neural networks, so they are more efficient than the hierarchical method (algebraic method) in dealing with large amounts of noisy data.

It is acclaimed in [11] that the SOTA has approximately linear runtime and is much faster than SOM and the Hierarchical methods.

The purpose of our study is to test the performance of the three clustering methods as well as to further compare the results of the SOM and SOTA clustering analysis.

3.1. Software for Performance Study
The software tool we use for experimental study is GEPS (Gene Expression Pattern Analysis Suite) [9]. It includes following servers:

- **Cluster Server**: This is an interface to Hierarchical Clustering. The resulting dendogram is plotted with TreeView.
- **SOM Server**: This is an interface to SOM package. The map is plotted with SomPlot. The resulting clusters can be extracted to continue with the analysis.
- **Sotarray Server**: This is the interface to SOTA for DNA array. The resulting tree can be viewed with TreeView or with SotaTree. The resulting clusters can be extracted to continue with the analysis.
- **SomTree Server**: This tool combines SOM and Hierarchical Clustering. The nodes of the resulting SOM map are clustered and the tree is plotted with SotaTree. The resulting clusters can be extracted to continue with the analysis.

### 3.2. The Data Set and Data Preprocessing

#### 3.2.1. Data Source

We experiment with a subset of the yeast *Saccharomyces cerevisiae* data set that measures the expression level of each of the 6601 different genes of *Saccharomyces cerevisiae* [7, 22]. The data is obtained using an Affymetrix hybridization array, and the values in the subset we selected are measured at 17 time points sampled at every 10 minutes during about two cell division cycles [31].

#### 3.2.2. Data Preprocessing

Our first processing step is to prune Genes with more than one missing value. After this process, 5509 genes remained in our data set. Second step is to randomly select 1000, 2000, 3000, 4000, and 5000 genes from the 5509 genes and save them as in plain text files, respectively.

These five preprocessed data sets are used for comparing the algorithms.

### 3.3. Runtime Comparison and Results

#### 3.3.1. SOTA vs. Hierarchical

Test condition for hierarchical clustering method UPGMA:

*UPGMA is an agglomerative hierarchical method. It starts by calculating the all-to-all distance matrix. The two closest patterns are merged and the all-to-all distance matrix is calculated again but using the new cluster instead of the two merged patterns. This process is repeated until the complete dendrogram is built."

- **Cluster method**: pairwise arithmetic average
- **Distance function**: correlation coefficient
- **Variability threshold**: 90%

The runtime comparison results are shown in Fig 1. For a large number of genes (>1000), SOTA is clearly faster than Hierarchical. For 5000 genes, it is about three orders of magnitude faster.

However, for a relatively small number (<1000) of genes, the performance of the SOTA and that of hierarchical clustering method are similar. In fact, for less than 600 genes the hierarchical clustering method is even slightly faster. This is because the training of the neural network implies a minimum number of presentations [9].

#### 3.3.2. SOTA vs. SOM

Test condition for SOM:

1. **Topology**: Hexagonal lattices
2. **X-Dimension**: 2
3. **Y-Dimension**: 3

Test condition for SOTA:

1. **Cluster method**: pairwise arithmetic average
2. **Distance function**: correlation coefficient
3. **Variability Threshold (%)**: 90

The runtime comparison results are shown in Fig 2. From this figure we know that the runtime of SOTA and SOM are proportional to the sample sizes, and the SOTA is faster than the SOM.
3.4. Clustering Results Comparison

3.4.1. Dataset and test condition. The preprocessed data file with 1k genes is used to compare the clustering results.

Test condition for Self Organizing Map (SOM):
- 2 * 3 hexagonal lattices (This will result in 6 clusters)

Test condition for Self Organizing Tree Algorithm (SOTA)
- (1) Cluster method: pairwise arithmetic average
- (2) Distance function: correlation coefficient
- (3) Unconditional training stop after 5 cycles (It will result in 6 clusters)

3.4.2. Clustering results. The result of SOTA clustering is shown below in Fig 3. In this plot the size of the ratio of the circles is proportional to the amount of genes in that cluster. The patterns of the clusters appear on the right of the circles.

Fig 4 shows the six detailed plots, each including the profile of a cluster and the profiles of the genes in that cluster.

Fig 5. Clustering results of SOM

The clustering result of SOM is shown in Figure 5.

Each rectangle corresponds to a node of the map. The black thick line in the rectangle corresponds to the profile of the node, and the gray lines correspond to the profiles of the genes in that cluster. The black bars on the left of the profiles are proportional to the number of genes in the clusters.

Fig 6 shows the six detailed plots, each of which includes the profile of a cluster and the profiles of the genes in that cluster.
4. Cluster Similarity Analysis

Many clustering algorithms have been proposed for the analysis of gene expression data, but little guidance is available to help choose among them [30]. For example, they lack facilities for estimating the optimal number of clusters, as well as components for evaluating the quality of the clusters obtained. In this section, we present a software tool that offers cluster similarity analysis methods for DNA microarray data analysis.

4.1. Software Introduction

We present a data mining tool, Cluster Diff®, which allows the similarity analysis of clusters generated by different algorithms. This tool may: (1) improve the quality of the data analysis results, (2) support the prediction of the number of relevant clusters in the microarray datasets, and (3) provide cross-reference between different algorithms. The software tool can also be used to analyze cluster similarities from other biomedical data.

The software allows working with two datasets each time. The Main Window (panel) (Fig 7) contains the file, view, and help.

4.2. Data Source and Data Preprocessing

This tool uses the textual tab-delimited data files. The format is similar to the Stanford tab-delimited format (http://genome-www5.stanford.edu/microarray/help/formats.shtml) except that you should put tab [cluster] and [/cluster] between a cluster dataset. An example of the described format is shown in Table 1.
Table 1. Input data file format

| [cluster] |      |      |      |
|-----------|------|------|------|
| YCR008W   | -0.26| 0.22 | -0.2 |
| YDR067C   | -0.13| 0.15 | 0.13 |
| YBR211C   | -0.53| -0.18| -0.33|
| [cluster] |
| YDL228C   | -0.83| -0.31| -0.08|
| YAR075W   | -0.68| -0.19| 0.64 |
| YBL059W   | -0.23| 0.04 | 0.21 |
| [cluster] |

4.3 Cluster Similarity Analysis Results: SOTA vs. SOM

The preprocessed data files with 1k genes, after formatting as Section 4.2, were loaded to the *cluster diff* for the cluster similarity analysis.

Each time, we input a pair of clusters, one by SOTA and one by SOM. One of the Screenshots is shown in Fig 8. The results are summarized in Table 2.

Table 2. Cluster similarity analysis results (SOTA vs. SOM)

|       | SOM11 | SOM12 | SOM13 | SOM21 | SOM22 | SOM23 | Match |
|-------|-------|-------|-------|-------|-------|-------|-------|
| SOTA1 | 0.00  | 0.00  | 0.00  | 0.04  | 0.46  | 0.04  | SOM22 |
| SOTA2 | 0.13  | 0.06  | 0.00  | 0.23  | 0.15  | 0.00  | SOM21 |
| SOTA3 | 0.40  | 0.13  | 0.00  | 0.16  | 0.10  | 0.02  | SOM11 |
| SOTA4 | 0.04  | 0.14  | 0.25  | 0.00  | 0.01  | 0.13  | SOM13 |
| SOTA5 | 0.00  | 0.00  | 0.20  | 0.00  | 0.01  | 0.32  | SOM23 |
| SOTA6 | 0.00  | 0.01  | 0.38  | 0.00  | 0.00  | 0.06  | SOM13 |

The score in bold bears the maximum value in both the row and the column, and the score in italic bares the maximum value in either the row or the column, but not both. From this table, we can find that most SOTA clusters match the SOM clusters well and vice versa. An example of good match (0.46) is SOTA1 with SOM22 (see Fig 9.). The profiles of these two clusters have similar trends, meaning that the most genes in the two clusters are similar.

Two clusters are not matched if the score is 0.00. An example is SOTA6 with SOM11 (see Fig 10.). From this figure, we can tell that their trends are different.
The cluster similarity results can better be viewed by rearranging Table 2 as Table 3.

Table 3. Cluster similarity analysis results (SOTA vs. SOM)

|       | SOM12 | SOM13 | SOM21 | SOM22 |
|-------|-------|-------|-------|-------|
| SOTA3 | 0.40  | 0.15  | 0.16  | 0.10  |
| SOTA4 | 0.04  | 0.14  | 0.25  | 0.01  |
| SOTA6 | 0.01  |       | 0.38  |       |
| SOTA2 | 0.13  | 0.06  | 0.23  | 0.15  |
| SOTA1 |       | 0.04  | 0.46  |       |

5. Conclusions

Hierarchical clustering methods allow a visual, convenient representation of genes. They can also generate an order of the genes, though the order is not unique. However, they are neither robust nor efficient. The SOM, as neural networks, is more robust against noise. The effects of outliers can be counter-balanced or corrected by the sequence of input genes. A disadvantage of SOM is that the number of clusters has to be fixed beforehand. But, in practice, that information may not be known. The SOTA is based on both neural networks and hierarchical clustering methods. It combines the advantages of both hierarchical and SOM clustering. It allows a visual representation of the clusters and their structure and is not sensitive to noises. The SOTA is also more flexible than the other two clustering methods.

Performance study shows that SOTA is more efficient than SOM while hierarchical clustering is the worst. The SOTA is much faster than Hierarchical clustering method. However, this is not always true when the data set is small. The runtimes of SOTA and SOM are approximately proportional to the number of genes. They both can be used to handle very large data sets.

In this paper, we also described a data mining tool, Cluster Diff, which allows the similarity analysis of clusters generated by different algorithms. This tool may: (1) improve the quality of the data analysis results, (2) support the prediction of the number of relevant clusters in the microarray datasets, and (3) provide cross-reference between different algorithms. The software tool can also be used to analyze cluster similarities from other biomedical data.

6. Acknowledgements

We would like to thank Dr. Dong-Guk Shin and Dr. Jae-guon Nam at the Univ. of Connecticut for providing the software for cluster similarity analysis in this work. We also thank Dr. Yanlin Ye for helpful discussions and comments.

7. Reference

[1] R. L. Stears, et al., “Trends in Microarray analysis,” Nature Medicine, Volume 9, 140-145, January 2003.
[2] M. F. Ramoni, et al., “Cluster analysis of gene expression dynamics,” PNAS, Vol.99, July 2002.
[3] J. Herrero, A. Valencia, and J. Dopazo, “A hierarchical unsupervised growing neural network for clustering gene expression patterns,” Bioinformatics, 17, pp. 126-136, 2001.
[4] M.B.Eisen, P.T.Spellman, P.O.Brown, D.Botstein. Cluster analysis and display of genome-wide expression patterns Proc. Natl. Acad. Sci., 95:14863-14868, 1998
[5] http://www.ebi.ac.uk/microarray/biology_intro.html
[6] N.S. Holter, M.Mitra, A.Maritan, et al. Fundamental patterns underlying gene expression profiles: Simplicity from Complexity. Proc. National Academy of Science, V.97: 15, pages 8409-8414, July 2000
[7] R. Cho, M. Campbell, et al. A genome-wide transcription analysis of the mitotic cell cycle, Molecular Cell, 2:65-73, 1998
[8] A. Brazma, J. Vilo. Gene expression data analysis, FEBS Letters, 480, 17-24, 2000
[9] http://bioinfo.cnio.es/sotarray/
[10] Teuvo Kohonen, The self-organizing map. Neurocomputing 21: 1–6. 1998
[11] Herrero, J., Valencia, A. and Dopazo, J., A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics, 17: 126-136, 2001
[12] Joaquin Dopazo and José Maria Carazo, Phylogenetic reconstruction using an unsupervised growing neural network
that adopts the topology of a phylogenetic tree. J. Mol. Evol., 44: 226-233, 1997
[13] Kohonen, T. Self-Organizing Maps, Springer, Berlin. 1995
[14] Sneath and Sokal. Hierarchical Clustering, 1973
[15] Tamayo, P. Dmitrovsky, E., et al. Interpreting patterns of gene expression with self-organizing maps: Methods and applications to hematopoietic differentiation, Proc. Nat. Acad. Sci 96, 2907-2912, 1999
[16] Herrero, J. & Dopazo, J. Combining hierarchical clustering and self-organizing maps for exploratory analysis of gene expression patterns. Journal of Proteome Research, 1(5):467-470. 2002.
[17] Botstein, D. & Brown, P., Exploring the new world of the genome with DNA microarrays, Nature Genetics (supp.) 21, 33-7. 1999
[18] Sokal, R. & Mitchener, C., A statistical method for evaluating systematic relationships, Univ. Kansas Sci. Bull.. 38, 1409-1438. 1958
[19] Everitt, B., Cluster analysis, Halstead, New York. 1980
[20] Hartigan, J., Clustering algorithms, Wiley, New York., 1973
[21] Kohonen, T., Self-Organization and Associative Memory (3rd edition), springer-Verlag, Berlin. 1989.
[22] Spellman, P.T., Eisen, et al. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces by microarray hybridization, Mol. Cell. Biol. 9(12), 3273-975. 1998
[23] Dopazo, J. Microarray Data Processing and Analysis. Microarray data analysis II. Kluwer Academic. Publ. Pp. 43-63, 2002
[24] Tamames, J., Clark, D., Herrero, J., Dopazo, J., Blaschke, C., Fernández, J.M., Oliveros, J.C., Valencia, A. Bioinformatics methods for the analysis of expression arrays: data clustering and information extraction. J. Biotechnol.. 98(2-3), 269-283, 2002
[25] Dopazo, J., Zanders, E., Dragoni, I., Amphlett, G. and Falciani, F. Methods and approaches in the analysis of gene expression data J. Immunol. Meth. 250, 93-112. 2001
[26] http://titan.biotec.uiuc.edu/cs491jh/slides/cs491jh-QJ.ppt
[27] K. Alsabti, S. Ranka, and V. Singh, An Efficient k-means Clustering Algorithm, Proc. First Workshop High Performance Data Mining, Mar. 1998.
[28] http://www.clustan.com/k-means_critique.html , March, 2001.
[29] T. Kanungo, D.M. Mount, N.S. Netanyahu, C. Piatko, R. Silverman, and A.Y. Wu, The Analysis of a Simple k-means Clustering Algorithm, Proc. 16th Ann. ACM Symp. Computational Geometry, pp. 100-109, June 2000.
[30] K.Y. Yeung, W.L.Ruzzo, et al. Validating clustering for gene expression data. Bioinformatics Vol 17, p 309-318, 2001
[31] http://genomics.stanford.edu