

AP-based consensus clustering for gene expression time series

Tai-Yu Chiu, Ting-Chieh Hsu, and Jia-Shung Wang

Department of Computer Science, National Tsing Hua University, HsinChu, Taiwan
 tychiu@vc.cs.nthu.edu.tw, tchsu@vc.cs.nthu.edu.tw, jswang@cs.nthu.edu.tw

Abstract

We propose an unsupervised approach for analyzing gene time-series datasets. Our method combines Affinity Propagation (AP) and the spirit of consensus clustering—extracting multiple partitions from different time intervals. Without priori knowledge of total number of clusters and exemplars, this method holds the relationship between genes through different time intervals, and eliminates the influence from noises and outliers. We demonstrate our method with both synthetic and real gene expression datasets showing significant improvement in accuracy and efficiency.

1. Introduction

The DNA microarray technology allows monitoring the expression levels for thousands of genes crossing various time points. During distinct biological processes, the gene expression data are collected to understand the complex dynamics of biological systems and reveal gene activities of conditional processes such as cell-cycle, diseases' progression, and the responses to external stimuli. The similar expression patterns would be organized into a group under the assumption that co-expressed genes share co-regulation or common functional tasks in correlated pathways. In addition, phenotypic responses with the production of proteins coded by the expressed mRNAs have causal relationship with the gene expression [1]. Therefore, numerous efficient clustering algorithms have been developed for analyzing gene expression data. Early developed methods such as k-means, hierarchical clustering and self-organizing maps are popular for their simplicity and availability. However, due to the noises and omissions from neglecting relationships between successive time points, these common algorithms have less accuracy of classifying genes. With regarding the relationships, Model-based clustering methods, for example, use statistical and probabilistic models to present the charac-

teristic of data [12, 7, 10, 9, 6]. The model-based methods gain advantages in high tolerance toward experimental errors, though; fail in computation inefficiency while modeling.

In this paper we propose an approach which combines Affinity Propagation [4] and the spirits of consensus clustering [8] to analyze the dependency between genes at different time intervals. We select multiple various numbers of time-points (called window sizes) as the subset of feature vectors for Affinity Propagation clustering in sliding-window mechanism. All clustering solutions obtained from Affinity Propagation would be merged for final partition by the concept of consensus clustering. While participating in the vote of the relationship between genes at different time patterns, each gene grade with supports (votes) – the stronger supports they show, the higher correlations they have.

2 Methods

In this section, the proposed algorithm based on Affinity Propagation [4], significant multi-class membership (SiMM) rule [2] and consensus clustering [8] is described. We formulate the time-courses expression clustering problem as follows: Given a set of genes $G = \{G_1, G_2, \dots, G_n\}$ where n is the number of genes, and each gene G_i includes τ time points for gene expression values. These n genes would be grouped into K disjoint clusters C_1, C_2, \dots, C_K . Figure 1 provides the framework for the proposed algorithm.

2.1 Interval Selection for Clustering

To take the temporal relationship into account, we use a sliding-window mechanism. For each window with different sub-intervals, we choose these time-points as feature vectors for clustering.

Affinity Propagation [4] based on *message passing* is a powerful clustering algorithm without the anticipative number of clusters K as input. Therefore, we use Affinity Propagation to cluster genes with each window

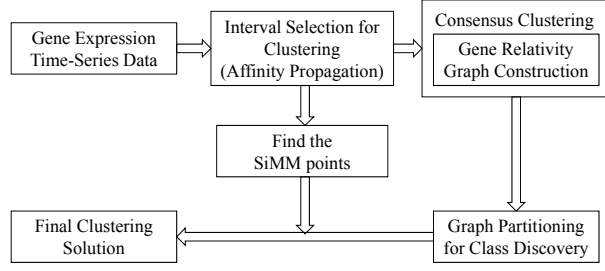


Figure 1. The framework for the proposed algorithm.

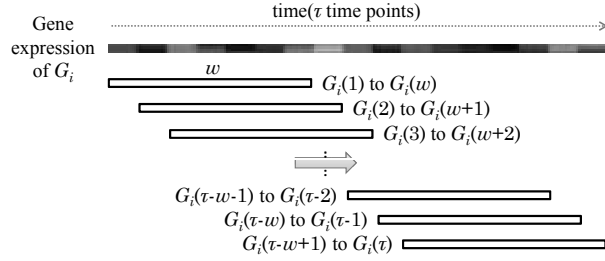


Figure 2. The sliding-window mechanism for interval selection.

while discovering the relationship between genes in different time intervals.

For each gene G_i with τ time points for gene expression values, we use a window sized w to represent the time pattern from $G_i(t)$ to $G_i(t+w-1)$ as in Figure 2. Then, the sliding-window mechanism is applied to extract different time interval as feature vectors. That is, we will have $(\tau-w+1)$ time patterns for each gene. When extracting sub-interval time-series pattern from $G(t)$ to $G(t+w+1)$ as feature vectors, we calculate the Pearson correlation coefficient for measuring similarity/dissimilarity between gene expression patterns of two genes G_i and G_j , which is defined as:

$$\begin{aligned} & Corr(G_i, G_j) \\ &= \frac{\sum_{l=t}^{t+w-1} (G_i(l) - \bar{G}_i)(G_j(l) - \bar{G}_j)}{\sqrt{\sum_{l=t}^{t+w-1} (G_i(l) - \bar{G}_i)^2 \sum_{l=t}^{t+w-1} (G_j(l) - \bar{G}_j)^2}} \end{aligned} \quad (1)$$

where $G_i(l)$ and $G_j(l)$ are expression value at l th time point of genes G_i and G_j , respectively. \bar{G}_i and \bar{G}_j are mean values of w expression data from genes G_i and G_j , respectively. We construct the correlation matrix (CM) whose entries r_{ij} is the similarity $Corr(G_i, G_j)$ between genes G_i and G_j , and then choose the median

of the similarities as preferences for the inputs of Affinity Propagation.

Then, we are able to obtain the predicted labels y of the genes G . The adjacency matrix M is formed by the predicted labels with entries m_{ij} defined as:

$$m_{ij} = \begin{cases} 1, & \text{if } y_i = y_j. \\ 0, & \text{if } y_i \neq y_j. \end{cases} \quad (2)$$

where y_i and y_j are the predicted labels of the genes G_i and G_j , respectively.

For each gene G_i , Affinity Propagation assigns a predicted label y_i as well as the most similar exemplar. Moreover, considering the genes which have significant multi-class membership [2], we applied the SiMM concept to filter $\lceil n \times \rho\% \rceil$ genes as SiMM list after clustering where ρ is set as SiMM rate.

2.2 Gene-Relativity Graph Construction

For each window sized w in Section 2.1, the adjacency matrix M to represent the relationship between genes will be constructed. We construct an $n \times n$ consensus matrix M_w^c by merging the adjacency matrix M_u ($u \in \{1, \dots, (\tau-w+1)\}$) as follows:

$$M_w^c = \frac{1}{(\tau-w+1)} \sum_{u=1}^{\tau-w+1} M_u \quad (3)$$

where $(\tau-w+1)$ is the number of adjacency matrices because of the sliding-window mechanism for temporal dynamics, and the element m_{ij} in the consensus matrix M_w^c indicates the possibilities that genes G_i and genes G_j are in the same class.

Besides, we can experiment with multiple window sizes to inspect the relationship between genes, and combine these consensus matrices with each window size, respectively. Suppose that we apply l window sizes w_1, w_2, \dots, w_l , then we acquire l consensus matrices $M_{w_1}^c, M_{w_2}^c, \dots, M_{w_l}^c$. We further define an aggregated consensus matrix R as:

$$R = \frac{1}{l} \sum_{u=1}^l M_{w_u}^c \quad (4)$$

where l is the number of windows with different sizes, and each entry m_{ij} in the aggregated consensus matrix R denotes the probability of two genes, G_i and G_j , appearing in the same class.

Afterwards, we construct a graph P to represent the relationship between genes from the aggregated consensus matrix R , called gene-relativity graph ($P =$

(G, R)). The vertices of the gene-relativity graph correspond to the genes in G , and the edges indicate the probability that two genes appear in the same class.

Suppose that we experiment with l window sizes w_1, w_2, \dots, w_l , we have η SiMM lists where η is the number of times of the Affinity Propagation clustering and derived as:

$$\eta = \sum_{i=1}^l (\tau - w_i + 1) = l\tau + l - \sum_{i=1}^l w_i \quad (5)$$

Next, we examine η SiMM lists and verify whether the gene which appears more than $\lceil \frac{\eta}{2} \rceil$ times should be removed. Finally, we obtain a new gene-relativity graph P to replace the original one.

2.3 Graph Partitioning for Class Discovery

After the reconstruction of gene-relativity graph P , we can investigate the relationship between genes. A relativity threshold $\sigma \geq 0.5$ is chosen to convert the graph P into a binary graph P^b :

$$p_{ij}^b = \begin{cases} 1, & \text{if } p_{ij} \geq \sigma. \\ 0, & \text{if } p_{ij} < \sigma. \end{cases} \quad (6)$$

where p_{ij} and p_{ij}^b denotes the edge weight between genes G_i and genes G_j of P and P^b , respectively.

Next, a Depth-first search algorithm is employed to find connected components of the undirected graph P^b , which is described above. Suppose that we find L connected components C_1, C_2, \dots, C_L , we would claim that these L connected components are L disjoint clusters C_1, C_2, \dots, C_L for results. However, some edges weighting more than 0.5 but slightly less than relativity threshold σ in the gene-relativity graph P would be eliminated from the binary graph P^b , and it would produce several connected components (i.e. clusters) with a few vertices (i.e. genes) when the Depth-first search algorithm applied. These connected components with a few vertices, called *tiny-clusters*, would influence the clustering results, so we should rearrange the previous clusters for final partition.

In order to determine which cluster is *tiny-cluster*, we use a parameter φ to restrict the number of genes in one cluster. A cluster is denoted as *tiny-cluster* if the number of genes in this cluster is less than φ . For each cluster $C_p, 1 \leq p \leq L$, the number of genes is compared with parameter φ to find out L' *tiny-clusters* $C_1, C_2, \dots, C_{L'}$ with the number of genes $n_1, n_2, \dots, n_{L'}$, respectively.

Afterwards, we begin merging each of L' *tiny-clusters* with $L - L'$ *main-clusters* whose number of

Table 1. The parameters setting.

Parameter	values
SiMM rate (ρ)	0% to 15%
genes in one cluster (φ)	1 to 15
relativity threshold (σ)	0.5 to 0.8
# of clustering times (η)	20 to 30

genes are more than φ . Consider each *tiny-cluster* $C_q, 1 \leq q \leq L'$ with the number of genes n_q . If n_q is equal to 1, this is a *singleton* cluster with one gene, says G_q . Our algorithm first finds out the gene G_h which has the highest relativity to G_q in the gene-relativity graph P . Then, merges C_q by the following rules: (1) If the relativity between G_q and G_h is below 0.5, the merging process stops. C_q survives as a *singleton* cluster. (2) Otherwise, if C_h is a *main-cluster*, we merge C_q with C_h . And (3) Otherwise (C_h is a *tiny-cluster*), we merge C_q and C_h to a new cluster whose number of genes is $n_q + n_h$; and check the type, *tiny-cluster* or *main-cluster*. On the other hand, if C_q is not *singleton*, our algorithm calculates the mean values G_q^m of these n_q gene profiles. Then, compares the maximum Pearson correlation coefficient with the mean values G_h^m which belongs to C_h in all *main-clusters*. Then, merges C_q if the Pearson correlation coefficient between G_q^m and G_h^m is above 0.

Repeatedly perform the above procedure and finally obtain K number of disjoint clusters C_1, C_2, \dots, C_K until no *tiny-cluster* exists. Finally, we assign each SiMM gene to one of K disjoint clusters C_1, C_2, \dots, C_K based on the Nearest Cluster-Centroid rule.

3 Results and Discussion

We first use both synthetic and real gene expression datasets to judged clustering accuracy with ARI (the adjusted Rand index) [5]. Then, the parameter settings would be exhibited in Table 1 and finally demonstrate the comparison of performance with other approaches.

Following [12] and [10], we synthesized two artificial datasets named *Sim_Data1* and *Sim_Data2*, respectively. In addition, two well-known gene expression datasets, the Yeast galactose dataset and the Yeast cell-cycle dataset [10].

Both synthetic datasets achieve great clustering accuracy 0.9051 and 1.0000 of ARI, and the result of *Sim_Data2* is slightly better because the *Sim_Data2* dataset excludes the linear function for being a non cell-cycle regulation function. Based on the observation, we can draw the conclusion that the clustering ac-

Table 2. The comparison of performance.

Method	The ARI
The Yeast galactose dataset	
Our algorithm	0.9746
CRF [6]	0.9478
CORE [11]	~ 0.7
Ng <i>et al.</i> 's [9]	0.9780
Yeung <i>et al.</i> 's [12]	0.9680
The Yeast cell-cycle dataset	
Our algorithm	0.5129
k-means	0.4300
Splines [3]	0.3620
HMM [10]	0.4670
CRF [6]	0.4808

curacy of the proposed algorithm relies on the choice of relativity threshold σ , and we suggest that the relativity threshold σ ranging from 0.55 to 0.65 be appropriate.

Then, we compared our results with other methods in two real gene expression datasets which was showed in Table 2. With the Yeast galactose dataset(205 genes, 20 time points), our algorithm has a maximum value 0.9746 of the ARI and only takes 45 seconds for clustering including loading time. Although the performance of Ng *et al.*'s method is slightly better than ours, the method has its inevitable limitation. That is, compared to our algorithm without the number of cluster as input, Ng *et al.*'s method would fit random-effects model with the number of components and apply BIC (Bayesian information criterion) for model selection to indicate the number of clusters [9]. Users have to decide the number of components in advance when using Ng *et al.*'s method, which causes various clustering results and thus sways the performance.

In the Yeast cell-cycle dataset(384 genes, 17 time points), our method achieved a peak value 0.5129 of ARI and only takes about 170 seconds each clustering solution against varied relativity threshold σ , including loading time. The performance of our algorithm is better than those of k-means algorithm, the Splines model [3], and probabilistic sequence data models such as HMM (Hidden Markov models) [10] and CRF (conditional random fields) [6].

4 Conclusion

In this paper, we presented an unsupervised clustering algorithm for analyzing time-series gene expression data, which requires no *priori* knowledge. The idea

combines Affinity Propagation and consensus clustering with various time intervals providing progressive robustness and accuracy.

Owing to the assistance in the efficiency of Affinity Propagation, our proposed algorithm provides appropriate and effective analysis on time-series gene expression experiments. Through the experimental results on two real gene expression datasets, our method illustrates the biological relevance between these genes and performs well.

References

- [1] I. P. Androulakis, E. Yang, and R. R. Almon. Analysis of time-series gene expression data: Methods, challenges, and opportunities. *Annual Review of Biomedical Engineering*, 9:205–228, 2007.
- [2] S. Bandyopadhyay, A. Mukhopadhyay, and U. Maulik. An improved algorithm for clustering gene expression data. *Bioinformatics*, 23(21):2859–2865, 2007.
- [3] Z. Bar-Joseph, G. Gerber, D. K. Gifford, and T. S. Jaakkola. A new approach to analyzing gene expression time series data. In *Proceedings of the Annual International Conference on Computational Molecular Biology, RECOMB*, pages 39–48, 2002.
- [4] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.
- [5] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.
- [6] C.-T. Li, Y. Yuan, and R. Wilson. An unsupervised conditional random fields approach for clustering gene expression time series. *Bioinformatics*, 24(21):2467–2473, 2008.
- [7] M. Medvedovic, K. Yeung, and R. Bumgarner. Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8):1222–1232, 2004.
- [8] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118, 2003.
- [9] S. K. Ng, G. J. McLachlan, K. Wang, L. B.-T. Jones, and S.-W. Ng. A mixture model with random-effects components for clustering correlated gene-expression profiles. *Bioinformatics*, 22(14):1745–1752, 2006.
- [10] A. Schliep, I. G. Costa, C. Steinhoff, and A. Schonhuth. Analyzing gene expression time-courses. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(3):179–193, 2005.
- [11] B. Tjaden. An approach for clustering gene expression data with error information. *BMC Bioinformatics*, 7:17, 2006.
- [12] K. Y. Yeung, M. Medvedovic, and R. E. Bumgarner. Clustering gene-expression data with repeated measurements. *Genome Biology*, 4:R34, 2003.