Synchronization-inspired Co-clustering and Its Application to Gene Expression Data

Junming Shao, Chongming Gao, Wei Zeng, Jingkuan Song, and Qinli Yang
Big Data Research Center, School of Computer Science and Engineering
University of Electronic Science and Technology of China, Chengdu 611731, China
Email: {junmshao@uestc.edu.cn, qinli.yang@uestc.edu.cn}

***Abstract*—In this paper, we propose a new synchronization-inspired co-clustering algorithm by dynamic simulation, called CoSync, which aims to discover biologically relevant subgroups embedding in a given gene expression data matrix. The basic idea is to view a gene expression data matrix as a dynamical system, and the weighted two-sided interactions are imposed on each element of the matrix from both aspects of genes and conditions, resulting in the values of all element in a co-cluster synchronizing together. Experiments show that our algorithm allows uncovering high-quality co-clusterings embedded in gene expression data sets and has its superiority over many state-of-the-art algorithms.**

***Keywords*-co-clustering; gene expression data; synchronization;**

I. Introduction

Co-clustering has been proved to be a powerful tool for knowledge discovery in a large variety of applications, such as text mining [1] and bioinformatics [2]. Instead of clustering one set of objects, co-clustering algorithms aim at finding subgroups by clustering rows and columns simultaneously. The derived subgroups (also referred as co-clusters) usually bring some deep insights into the data. For the gene expression data, co-clusters characterize subsets of genes which are co-regulated under a particular subset of experimental conditions. Discovering such patterns may be the key to uncover many genetic pathways. To present, many co-clustering algorithms have been proposed for microarray analysis based on different criteria, such as mutual information [1], graph cut [3] and residue [2]. However, each criterion comes with specific advantages and drawbacks. For example, the well-known residue-based methods [2] often employ iterative strategies to simultaneously identify co-clusters with coherent values in both rows and columns, allow identifying a good "checkerboard" structure. However, the assumed "checkerboard" structure (also existed in information-theoretic methods and graph-based methods) is not ideal. Beyond, the number of gene clusters and condition clusters need to be specified, which are usually not available in real-world applications.

In this paper, we consider the co-clustering problem from a new aspect: **Synchronization**. We will see this new viewpoint supplements an intuitive way to discover co-clusters, and has several attractive properties. But let us first illustrate the basic idea.

A. Basic idea

Synchronization is a phenomenon that a group of events spontaneously come into co-occurrence with a common rhythm, despite of the differences between individual rhythms of the events [4]. To present, synchronization phenomena in nature have been widely investigated and many models concisely describing the dynamical synchronization process have been proposed [4], [5]. The basic philosophy of synchronization-based models is to dynamically simulate the synchronization process by imposing the interactions on local or global objects, resulting in objects are gradually synchronized together and have the same phase. Therefore, motivated by the existing models, we extend the notion of synchronization into the context of co-clustering, and propose a new algorithm, called CoSync.

To illustrate the formation of synchronized co-clusters by dynamic simulation, Figure 1 gives a simple example. Considering a given artificial gene expression matrix, for each element (a_{ij}), we first use its associated gene (i.e. $a_{i.}$) and condition (i.e. $a_{.j}$) to search similar genes and conditions, respectively (see Figure 1 (b)). Subsequently, each element will interact with its similar genes and conditions simultaneously in a weighted way. Specifically, if its similar genes or conditions are similar, the interaction will be strong, and vice versa. The detailed two-sided weighted interaction model will be elaborated in Section III-D. Through the weighted interactions, the difference among values of elements in a co-cluster will decrease and become zero gradually. Finally, after the dynamic simulation, we can identify the co-clusters by simply searching the subsets of genes and conditions with the same values. For better visualization, Figure 1 plots the **shuffled** resulting data matrix after dynamic simulation, where we can observe that the co-clusters are intuitively popped out as all elements in the same co-clusters are synchronized together (i.e. all elements in the green rectangles in Figure 1).

II. Related Work

Due to space limitation, for a comprehensive survey of co-clustering to gene data analysis, please refer to the recent paper by Pontes et al. [6]. Here, we only provide a very brief survey on some major research directions.

Information-theoretic methods. The key idea of these methods [1], [7] is to consider the problem of co-clustering

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.141

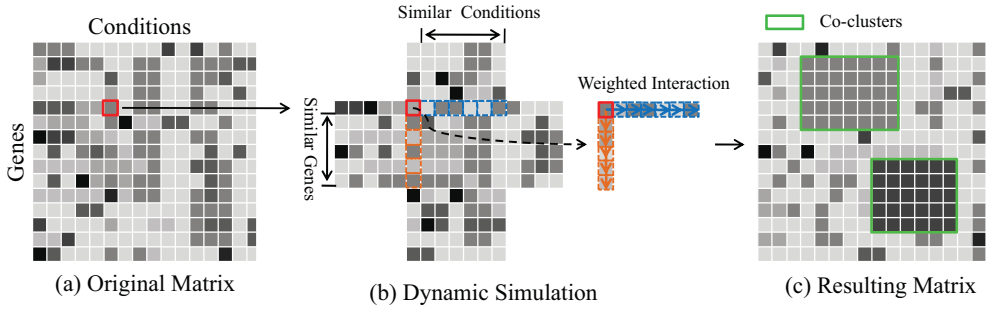


Figure 1: Illustration of co-cluster formation by simulating the synchronization process with a weighted two-sided interaction model. (a) The original gene expression data matrix, where the rows correspond the genes while the columns indicate the different conditions, and each element with different levels of gray colors indicate its value. (b) Relying on the weighted two-sided interactions, the value of each element will change over time, where the values of elements among each co-cluster tend to synchronize together gradually. The blue and yellow arrows indicate each element interact with its similar conditions and genes, respectively. (c) The final state of data matrix after **shuffling**, where co-clusters are popped up visually.

as a data compression problem based on information theory. The identification of co-clusters is to optimize some criteria, such as mutual information, Bregman divergence, subject to some constraints. The most fundamental technique in this line is ITCC [1], which views a nonnegative matrix as the estimate of an empirical joint probability distribution of two discrete random variables, and presents an algorithm to reduce the loss of mutual information between the original data matrix and the compressed representation provided by the co-clustering. Recently, Song et al. [7] propose an approach called constrained information-theoretic co-clustering, which integrates constraints into the information theoretic co-clustering framework and employs KL-divergence to improve clustering performance.

Graph-based Methods. In graph-based co-clustering approaches, a data matrix is constructed as a bipartite graph between rows and columns. The identification of co-clusterings is thus formulated as a problem of graph partitioning. For instance, in [8], Dhillon formalizes this idea by modeling document collection as a bipartite graph between documents and words, using the second left and right singular vectors to yield good bipartitionings. The similar idea is also employed in [9], where the partition is constructed by minimizing a normalized sum of edge weights between unmatched pairs of vertices of the bipartite graph.

Residue-based Methods. The type of residue-based methods refers to a class of techniques by optimizing the objective function of residue, which is widely used in expression data analysis. Cheng and Church [10] are considered to be the first to apply co-clustering to gene expression data to generate co-clusters that satisfy mean squared residue. Following this idea, Cho et al. [2] develop a popular co-clustering algorithm, Minimum Sum-Squared Residue Co-clustering (MSSRCC), which tries to escape the local minima and resolve the degeneracy problem in partitioned clustering algorithms. Lazzeroni and Owen [11] propose the plaid models for gene expression data analysis, which is viewed as a merger of clustering and ANOVA methods, and allows exploring the overlapping clusters.

III. Co-Clustering by Dynamic Simulation

In this section, we present our CoSync algorithm for co-clustering of gene expression data. In the following, we start with some basic notations.

Notations. Here we introduce some notations used in the remaining of the section. Formally, given the matrix $A = (A_I, A_J)$ with set of rows A_I and set of columns A_J, $a_{i.} \in A_I$ indicates the i^{th} row vector, and $a_{.j} \in A_J$ represents the j^{th} column vector. A co-cluster is a submatrix $A(I_S, J_S)$, where I_S is the indices of a subset of the rows A_I, J_S is the indices of a subset of the columns A_J. a_{ij} is the value of element A_{ij} corresponding to the i^{th} row and j^{th} column.

A. Two-sided Weighted Interaction Model

To uncover the co-cluster structure of a given gene expression data by dynamic simulation, the interaction model is essential. Currently, most existing interaction models (e.g.[4], [12], [5]) are often one-sided and in an unweighted way. However, since we are interested in co-clusters, the interactions should be imposed on both sides of genes and conditions in a local weighted fashion. In the following, we will formulate our interaction model based on the above two aspects.

Definition 1 (ϵ-**Range Neighborhood**) Given a data matrix $\mathcal{A}$ and $\epsilon \in \mathcal{R}$, the ϵ-range neighborhood of a row vector $a_{i.} \in A_I$ (or a column vector $a_{.j} \in A_J$), denoted as $N^r_\epsilon(a_{i.})$ (or $N^c_\epsilon(a_{.j})$), is defined as:

$$N^r_\epsilon(a_{i.}) = \{p \big| dist(a_{p.}, a_{i.}) \leq \epsilon, p \in I\} \quad (1)$$

where $dist(\cdot,\cdot)$ is a metric distance function, and the Euclidean distance is used in this study.

Based on the Kurumoto model, like existing synchronization-based models [13], we formulate our two-sided interaction model as follows.

DEFINITION 2 (TWO-SIDED INTERACTION MODEL) Let a_{ij} be the value of an element A_{ij}. Given an ϵ-range row and column neighborhood, respectively, the dynamics of the element A_{ij} of two-sided interaction from both rows and columns is defined as:

$$a_{ij}(t+1) = a_{ij}(t) + \frac{1}{2|N_\epsilon^r(a_{i.}(t))|} \cdot \sum_{p \in N_\epsilon^c(a_{i.}(t))} \sin(a_{pj}(t) - a_{ij}(t)) + \frac{1}{2|N_\epsilon^c(a_{.j}(t))|} \cdot \sum_{q \in N_\epsilon^r(a_{.j}(t))} \sin(a_{iq}(t) - a_{ij}(t)) \quad (2)$$

where $sin(\cdot)$ is the coupling function, which is a wide spreading function in almost all synchronization-based models (e.g. [4], [14]). $a_{ij}(t+1)$ describes the renewal value of the element A_{ij} at time stamp $t+1$ ($t = (0, \ldots, T)$) during the dynamic simulation. The interaction model allows investigating the dynamics of each element by coupling the element discrepancies from sides of rows and columns simultaneously.

However, as we state above, we only expect the similar genes under similar conditions to group together by dynamic interaction. Therefore, the interactions among elements should be considered differently. An intuitive way is to examine the distribution of expressed values of similar genes or conditions. Specifically, we first examine the disparities between the elements of similar genes (or conditions) and the element A_{ij}, and then use the standard deviation of the disparities to determine the coupling strength. We expect that the deviation tend to be small if the similar genes are co-regulated or the similar conditions are similar.

DEFINITION 3 (WEIGHTING FACTOR) Given an ϵ-range row neighborhood $N_\epsilon^r(a_{i.}(t))$ of the element A_{ij}, the weighting factor for similar genes is defined as:

$$w^r(j) = e^{-\lambda \cdot \sigma_j} \quad (3)$$

where σ_j is the standard deviation of the difference vector $\nu_{pj} = \{abs(a_{pj} - a_{ij}) | p \in N_\epsilon^r(a_{i.}(t))\}$. λ is a constant. Empirical experiments indicate that $\lambda = [50 - 150]$ often produces promising results. In this paper, we set $\lambda = 100$ for all experiments. Similarly, the weighting factor for the interactions of similar conditions is defined as:

$$w^c(i) = e^{-\lambda \cdot \sigma_i} \quad (4)$$

where σ_i is the standard deviation of the difference vector $\nu_{iq} = \{abs(a_{iq} - a_{ij}) | q \in N_\epsilon^c(a_{.j}(t))\}$.

Based on the weighting factor, the row and column interactions at time stamp t can be computed as follows, respectively.

$$I_{row}(t) = \frac{w^r(j)}{2|N_\epsilon^r(a_{i.}(t))|} \cdot \sum_{p \in N_\epsilon^r(a_{i.}(t))} \sin(a_{pj}(t) - a_{ij}(t)) \quad (5)$$

$$I_{col}(t) = \frac{w^c(i)}{2|N_\epsilon^c(a_{.j}(t))|} \cdot \sum_{q \in N_\epsilon^c(a_{.j}(t))} \sin(a_{iq}(t) - a_{ij}(t)) \quad (6)$$

Finally, the dynamics of each element is govern as follows.

$$a_{ij}(t+1) = a_{ij}(t) + I_{row}(t) + I_{col}(t) \quad (7)$$

To characterize the level of synchronization among elements during the dynamic simulation process, an order parameter r is defined to measure the coherence of the local population of elements.

DEFINITION 4 (ORDER PARAMETER) The order parameter r is used to terminate the dynamic simulation by investigating the degree of local synchronization, which is defined as:

$$r = \frac{1}{2|I|} \sum_{i=1}^{|I|} \frac{1}{|N_\epsilon^r(a_{i.})|} \sum_{p \in N_\epsilon^r(a_{i.})} \mathbf{e}^{-||a_{p.} - a_{i.}||} + \frac{1}{2|J|} \sum_{i=1}^{|J|} \frac{1}{|N_\epsilon^c(a_{.j})|} \sum_{q \in N_\epsilon^c(a_{.j})} \mathbf{e}^{-||a_{.q} - a_{.j}||} \quad (8)$$

The dynamic simulation terminates when $r(t)$ converges, which indicates local coherence.

B. Synchronized Co-clusters Search

After the simulation of dynamics of elements based on our weighted two-sided interaction model, the values of elements with co-regulated genes under a certain set of conditions will synchronize together. The search of these co-clusters is to find the particular subset of rows and columns that share the same value. For this purpose, we first identify all distinct values of elements in the resulting data matrix after dynamic simulation. If the count of one distinct value is smaller than a given size, e.g. $minSize = 100$, the value is removed. Otherwise, for each distinct value (e.g. c), we search the maximal block of same values with constraints of minimum rows and columns, which is actually a maximum closed frequent itemset mining problem. Here we apply the popular CHARM algorithm [15].

C. Speed up Range Search for High-dimensional Data via Non-negative Matrix Factorization

Nonnegative matrix factorization: Given a nonnegative matrix $A \in \mathbb{R}^{m \times n}$ and an integer $k < min(m, n)$, the goal of NMF is to find two factor matrices $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{n \times k}$, so that

$$\min_{W \geq 0, H \geq 0} f(W, H) = ||A - WH^T||_F^2 \quad (9)$$

Algorithm 1 CoSync

Input: A, $minRow$, $minCol$
$A = norm(A)$; //row or column normalization
if LargeFlag == TRUE **then**
  [W, H] = NMF(A); //non-negative matrix factorization
end if
while LoopFlag == TRUE **do**
  // Interactions of genes
  for each gene vector $a_{i.} \in A$ **do**
    if LargeFlag == TRUE **then**
      Search its ϵ-neighborhood $N_\epsilon^r(a_{i.})$ on W;
    else
      Search its ϵ-neighborhood $N_\epsilon^r(a_{i.})$ on A;
    end if
    for each condition $j \in J$ **do**
      Calculate the weighting factor $w(j)$ with Eq. (3);
      Compute row interactions with Eq. (5);
    end for
  end for
  // Interactions of conditions
  for each condition vector $a_{.j} \in A$ **do**
    if LargeFlag == TRUE **then**
      Search its $\epsilon-$neighborhood $N_\epsilon^c(a_{.j})$ on H;
    else
      Search its $\epsilon-$neighborhood $N_\epsilon^c(a_{.j})$ on A;
    end if
    for each gene $i \in I$ **do**
      Calculate the weighting factor $w(i)$ with Eq. (4);
      Compute column interactions with Eq. (6);
    end for
  end for
  Update the matrix A with Eq. (7);
  Compute order parameter r with Eq. (8);
  if r converges **then**
    LoopFlag = false;
  end if
end while
//Find co-clusters
for each distinct value c **do**
  Find the maximum co-cluster with CHARM;
end for
find all co-clusters C;
Output: C;

Where $||\dot{|}|_F^2$ is the Frobenius norm. In the context of microarray analysis, A corresponds to the gene expression data matrix. With this matrix factorization, we can observe that each gene (i.e. $A(i,:)$) can be written by $A(i,:) = W(i,:) \cdot H^T$, and each condition is represented as $A(:,i) = W \cdot H(:,i)$. The two matrices allow capturing the genes similarity and condition similarity, respectively. For instance, given a particular gene, if we want to search its similar genes with a given range, we do not need to search on the original data matrix, instead of performing range search on the matrix W. It is important to note that we just use NMF for speeding up the similar conditions or genes search. Therefore, this step is optional. In this study, we set $k = 5$ for all experiments.

D. CoSync Algorithm

Building upon the interaction model (cf. Eq. (7)), the dynamical change of values for each element can be simulated. For illustration, Figure 2(a)-(d) shows the detailed process of dynamical change of an artificial data matrix from $T = 0$ to $T = 30$. Figure 2(a) plots the original data matrix. From $T = 1$ to $T = 30$, the value of each element will change dynamically based on the influence from its similar genes and similar conditions simultaneously. Finally, elements within a co-cluster will synchronize together and share the same value. During the process, the order parameter, characterizing the level of local synchronization will gradually converge (Figure 2)(e). Finally, the Pseudocode of CoSync is given in Algorithm 1.

E. Time Complexity

For synchronization-based co-clustering, the runtime complexity with respect to the number of genes M and the number of conditions N is $O(T \cdot (M^2N + N^2M))$, where T is the iterations. If there exists an efficient index, the complexity reduces to $O(T \cdot (M \log MN + N \log NM))$. For NMF, its complexity is O(MN). The computational cost of searching co-cluster is $O(l \cdot |C|)$ [15], where l is the average row and column length and C is the set of all identified co-clusters.

IV. Experimental Evaluation

A. Experimental Setup

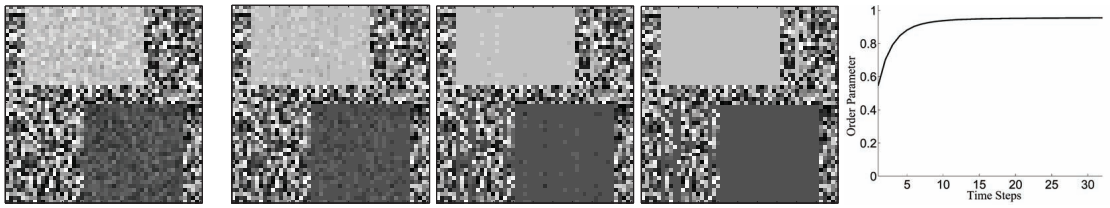
Data sets. We evaluate the proposed method CoSync on synthetic data and different genres of real-world gene expression data sets.

Synthetic Data. Here, to prove the concepts, we generate a flexible co-clusters in the data matriX. For each implanted co-cluster, the values of elements are generated based on the Gaussian distribution $N(\mu, \sigma)$, of which the mean values μ vary in the range of $(0, \frac{\pi}{2})$ and the standard deviations σ keeps set 0.1. For the other elements, the values follow the uniform distribution $U(0, \frac{\pi}{2})$.

Real-world Data. To evaluate the performance of our co-clustering algorithm, we further perform the experiments on four different genres of gene expression data sets[1]: Colon Cancer, Leukemia, Lung, and MLL, which are widely used for previous evaluation of co-clustering algorithms for gene expression data analysis. The preprocessing has been done with the same procedure in MSSRCC [2].

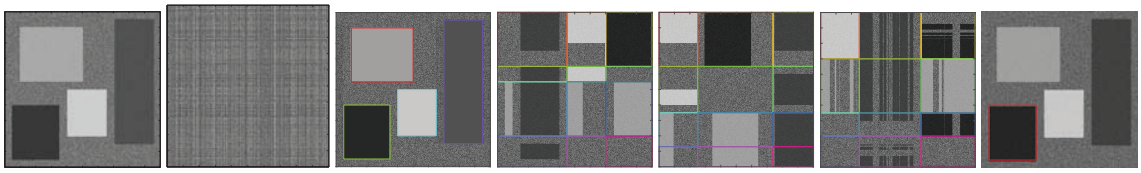
Selection of comparison methods. We compare CoSync to several representatives of co-clustering paradigms: the information-theoretic co-clustering algorithm: ITCC [1], a well-known residue-based co-clustering algorithm MSSRCC [2], the well-known overlapping co-clustering algorithm based on plaid models [11] and a graph-based co-clustering algorithm via spectral method [3].

[1]Data sets and corresponding gene names and IDs are publicly available at http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html



(a) Original data matrix (b) T = 1 (c) T = 5 (d) T = 30 (e) Order parameter

Figure 2: Illustration of the co-clusters formation based on dynamic simulation. (a). The original data matrix, where two co-clusters are embedded. (b) - (c). The dynamical changes of elements over time, where the values are aligned with its similar genes and conditions over time. (d). The final state of data matrix at time stamp T = 30, the elements in the same co-clusters synchronize together. (e). The order parameter.



(a) Original Data (b) Shuffled data (c) CoSync (d) ITCC (e) MSSRCC (f) Spectral (g) Plaid

Figure 3: Comparing co-clustering algorithms on an artificial synthetic data set with flexible co-cluster structure. Here the blocks with colors means the detected co-clusters by different algorithms.

Table I: The quality of co-clusters found by CoSync algorithm, which is evaluated from sample clustering. Here P and R represent the precision and recall for each co-cluster. $No.G.$ and $No.S.$ are the number of genes and samples in this co-cluster. N and T represent normal and tumor tissues, respectively. A and M represent ADCA and MPM respectively. AL, AM and ML represent ALL, AML and MLL respectively.

	Colon					Leukemia					Lung					MLL				
cID	**Size**	**No.G.**	**No.S.**	**P**	**R**	**Size**	**No.G.**	**No.S.**	**P**	**R**	**Size**	**No.G.**	**No.S.**	**P**	**R**	**Size**	**No.G.**	**No.S.**	**P**	**R**
1	1480	296	5(N)	1.00	0.23	3216	268	12(7AL/5AM)	0.58	0.15	5614	401	14(A)	1.00	0.09	4228	302	14(AM)	1.00	0.50
2	966	138	7(5N/2T)	0.71	0.23	2570	514	5(4AM/1AL)	0.80	0.16	4394	338	13(M)	1.00	0.42	3765	251	15(AL)	1.00	0.63
3	666	111	6(T)	1.00	0.15	2320	464	5(AL)	1.00	0.11	3960	264	15(A)	1.00	0.10	2954	211	14(AL)	1.00	0.58
4	510	85	6(5N/1T)	0.83	0.23	1480	296	5(AL)	1.00	0.11	3806	346	11(M)	1.00	0.36	2071	109	19(AM)	1.00	0.68
5	420	84	5(N)	1.00	0.13	1215	243	5(AM)	1.00	0.20	2344	293	8(A)	1.00	0.05	1584	99	16(AM)	1.00	0.57
6	290	58	5(T)	1.00	0.13	625	125	5(AL)	1.00	0.11	2210	221	10(M)	1.00	0.32	918	102	9(AL)	1.00	0.38
7	205	41	5(T)	1.00	0.13	320	64	5(AL)	1.00	0.11	1770	177	10(M)	1.00	0.32	890	89	10(AL)	1.00	0.42
8	125	25	5(T)	1.00	0.13	294	49	6(AM)	1.00	0.24	1035	115	9(M)	1.00	0.29	715	143	5(3ML/2AL)	0.60	0.15
9	65	13	5(T)	1.00	0.13	264	22	12(AL)	1.00	0.26	950	190	5(A)	1.00	0.03	533	41	13(AM)	1.00	0.46
10	49	7	7(T)	1.00	0.13	242	22	11(AL)	1.00	0.23	420	30	14(M)	1.00	0.45	510	34	15(AM)	1.00	0.54

B. Proof of Concept

Here, we examine whether CoSync allows finding flexible co-cluster structure on a synthetic data set. Figure 3 plots the clustering results of different algorithms. CoSync allows finding all the four co-clusters with distinct sizes successfully. However, ITCC, MSSRCC and Spectral are difficult to find these co-clusters. The reason behind it is that these algorithms assume a potential checkerboard structure, and they tend to fail if the truly co-clusters do not follow such assumption. For Plaid, although it aims at exploring overlapping co-clusters, only one cluster is correctly found.

Table II: Co-clustering performance of different algorithms on gene expression data sets from sample-based evaluation.

	CoSync			**Plaid**			**ITCC**		**MSSRCC**		**Spectral**	
	#C	Pre.	Rec.	#C	Pre.	Rec.	#C	Avg.	#C	Avg.	#C	Avg.
Colon	11	0.95	0.66	4	0.71	0.66	2	0.82	2	0.86	2	0.73
Leukemia	28	0.96	0.71	2	0.81	0.43	2	0.95	2	0.93	2	0.74
Lung	23	1.00	0.96	3	1.00	0.50	2	0.85	2	0.99	2	1.00
MLL	23	0.99	0.67	4	0.88	0.88	3	0.83	3	0.93	3	0.64

C. CoSync on Gene Expression Data

1) Sample-based Evaluation of Co-clusterings: For the four gene expression data sets, since we have known the sample categories, we first evaluate the co-clusterings generated by different algorithms from the sample aspect. Table

I summarizes the precisions and recalls of the top ten co-clusters detected by CoSync on the four data sets. We can observe that CoSync allows finding high-quality co-clusters. For instance, on the Colon data set, CoSync finds 11 clusters and all clusters correspond to a perfect match with a corresponding type (normal tissue or tumor issue) except 2 clusters, where in total 3 samples are wrongly grouped. The good clustering results can also be found on the Leukemia and MLL data sets. More impressively, CoSync allows a perfect grouping of samples on the Lung data set. To further evaluate the performance of CoSync, ITCC, MSSRCC, Plaid and Spectral algorithms are also employed to find co-clusters on these data sets. Table II gives a summary of average sample clustering performance, and we can see CoSync also shows its superiority over other comparing algorithms.

2) Gene-based Evaluation of Co-clusterings: In this section, we evaluate the statistical significance of the interesting gene clusters (i.e. the enrichment of functional annotations) generated by CoSync with the help of the Gene Ontology database on three categories of annotations: "Molecular Function", "Biological Process" and "Cellular Component". We use the DAVID software to find the significantly enriched for functional annotations of the gene set in each co-cluster, which is publicly available at https://david.ncifcrf.gov. We notice that all generated gene clusters allow a good enrichment for the three categories (with both a large number of enriched annotations and corresponding small p-values (all p-values are lower than 0.05)). The results indicate that CoSync also allows finding high-quality co-clusters from gene aspect (biological significant clusters).

V. Conclusion

In this paper, we introduce a new co-clustering algorithm, CoSync, to uncover the co-cluster structure of gene expression data sets based on synchronization-inspired dynamic simulation. By coupling the elements with a weighted two-sided mode, the values of these elements of co-regulated genes under a particular set of experimental conditions tend to synchronize together to automatically form co-clusters.

VI. Acknowledgement

This work is supported by the National Natural Science Foundation of China (61403062, 61433014, 41601025), China Postdoctoral Science Foundation (2014M552344, 2015M580786, 2015T80973), Science-Technology Foundation for Young Scientist of SiChuan Province (2016JQ0007), National key research and development program (2016YFB0502300).

References

[1] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 89–98.

[2] H. Cho and I. S. Dhillon, "Coclustering of human cancer microarrays using minimum sum-squared residue coclustering," *ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 3, pp. 385–400, 2008.

[3] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, "Spectral biclustering of microarray data: coclustering genes and conditions," *Genome research*, vol. 13, no. 4, pp. 703–716, 2003.

[4] C. Böhm, C. Plant, J. Shao, and Q. Yang, "Clustering by synchronization," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 583–592.

[5] J. Shao, Q. Yang, H.-V. Dang, B. Schmidt, and S. Kramer, "Scalable clustering by iterative partitioning and point attractor representation," *ACM Transactions on Knowledge Discovery from Data*, vol. 11, no. 1, p. 5, 2016.

[6] V. A. Padilha and R. J. Campello, "A systematic comparative evaluation of biclustering techniques," *BMC bioinformatics*, vol. 18, no. 1, p. 55, 2017.

[7] Y. Song, S. Pan, S. Liu, F. Wei, M. X. Zhou, and W. Qian, "Constrained text coclustering with supervised and unsupervised constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1227–1239, 2013.

[8] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 269–274.

[9] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Bipartite graph partitioning and data clustering," in *International conference on Information and knowledge management*, 2001, pp. 25–32.

[10] Y. Cheng and G. M. Church, "Biclustering of expression data," in *International Conference on Intelligent Systems for Molecular Biology*, vol. 8, 2000, pp. 93–103.

[11] L. Lazzeroni and A. Owen, "Plaid models for gene expression data," *Statistica Sinica*, pp. 61–86, 2002.

[12] J. Shao, Z. Ahmadi, and S. Kramer, "Prototype-based learning on concept-drifting data streams," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 412–421.

[13] J. Shao, X. He, C. Bohm, Q. Yang, and C. Plant, "Synchronization-inspired partitioning and hierarchical clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 893–905, 2013.

[14] J. Shao, X. Wang, Q. Yang, C. Plant, and C. Böhm, "Synchronization-based scalable subspace clustering of high-dimensional data," *Knowledge and Information Systems*, vol. 52, no. 1, pp. 83–111, 2017.

[15] M. J. Zaki, C.-J. Hsiao *et al.*, "Charm: An efficient algorithm for closed association rule mining," in *Technical Report 99*, 1999.