The paper lacks motivation for using a dynamical system approach. It seems like a solution looking for a problem to solve unfortunately.

根据综述，我们的bicluster应该属于constant value型的，不能找到constant value on row or column 或者 coherent value 或者coherent evoluation的。

Synchronization-inspired Co-clustering of Gene Expression Data

Junming Shao
Big Data Research Center,
University of Electronic
Science and Technology of
China, Chengdu, China
junmshao@uestc.edu.cn

Chongming Gao
Big Data Research Center,
University of Electronic
Science and Technology of
China, Chengdu, China
chongming.gao@gmail.com

Wei Zeng
Big Data Research Center,
University of Electronic
Science and Technology of
China, Chengdu, China
zwei504@uestc.edu.cn

Qinli Yang
Big Data Research Center,
University of Electronic
Science and Technology of
China, Chengdu, China
qinli.yang@uestc.edu.cn

ABSTRACT

Co-clustering has gained growing attentions recently due to its wide practical applications in biological data analysis, text mining and recommender systems. Most existing co-clustering algorithms usually search co-clusters by optimizing some criteria, such as mutual information, residue and graph-cut, and yield good results with the assumed co-cluster structure, e.g. checkerboard. In this paper, we propose a new synchronization-inspired co-clustering algorithm for microarray analysis by dynamic simulation, called CoSync, which aims at discovering all biologically relevant subgroups embedding in a given gene expression data matrix. The basic idea is to view the gene expression data matrix as a dynamical system, and the weighted two-sided interactions are imposed on each entry of the matrix from both aspects of genes and conditions, resulting in the values of all entries in a co-cluster synchronizing together. We demonstrate that our new co-clustering approach has several attractive benefits: (a) CoSync is capable of identifying the biologically relevant co-clusters with high-quality, driven by the intrinsic data structure. (b) Without any co-cluster structure assumption, CoSync supports finding co-clusters of arbitrary size, not limited to disjoint co-clusters. (c) In conjunction with non-negative matrix factorization, CoSync allows analyzing large-scale gene expression data. Experiments show that our algorithm faithfully uncovers co-clusterings embedded in gene expression data sets and has good performance compared to state-of-the-art algorithms.

WOODSTOCK '97 El Paso, Texas USA
© 2016 ACM. ISBN 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

CCS Concepts

•**Applied computing** → *Bioinformatics;* •**Theory of computation** → *Unsupervised learning and clustering;*

Keywords

Co-clustering; Synchronization; Gene Expression Data

1. INTRODUCTION

Microarrays technologies allow measuring the gene expression level for thousands of genes in parallel [9]. The result of a microarray experiment is usually formatted as a gene expression data matrix, whose rows represent genes and columns represent various specific experimental conditions (e.g. different samples, different time points, or different organisms). Currently, how to extract the biologically relevant knowledge or patterns from the information-rich data matrices is still a challenging problem [8]. Recently, co-clustering has been proved to be a powerful tool for knowledge discovery in a large variety of applications, such as text mining [10, 11, 20], bioinformatics [7, 8], and recommendation systems [28, 21]. 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 instances, subgroups in recommender systems represent a group of customers with similar behaviors favoring a subset of similar products, and co-clusters in text mining indicate documents with similar properties with respect to subgroups of terms. For the gene expression data, co-clusters characterize subsets of genes 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 [11], graph cut [16, 13] and residue [7, 8]. However, each criterion comes with specific advantages and drawbacks. For example, the well-known residue-based methods [7, 8] often employ iterative strate-

exaggerate the effect.
The phrase "all biologically relevant" made me hands-shaking. To my understanding, it's impossible to use a computational algorithm to discover such patterns.

全文只在这里一个地方提到shuffle。

改进：应该显示地提出searching patterns can be spread or scattered over the whole data set, rather than be restricted by the neighboring rows or columns. 对于示意图以及人工数据集，只是为了说明的方便，将pattern可视化成块状。

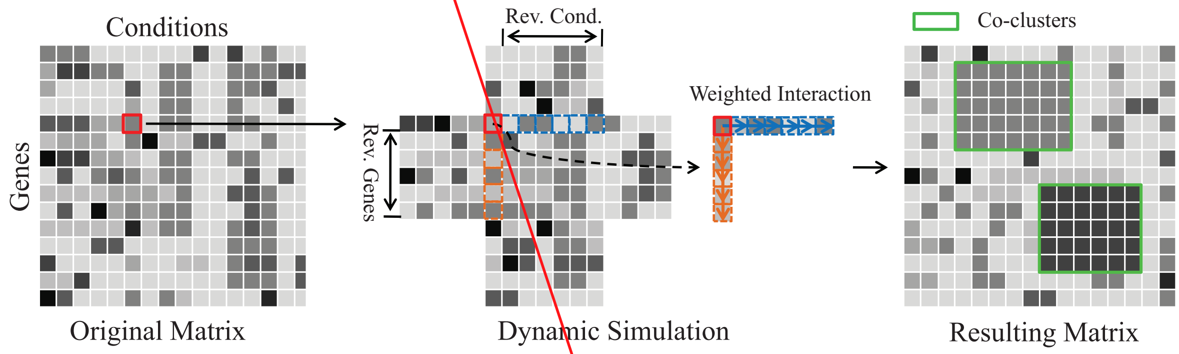


Figure 1: Illustration of co-cluster formation by simulating the dynamics of entries with both interactions of rows and columns. (a) The original gene expression data matrix, where the rows correspond the genes while the columns indicate the different conditions. (b) Relying on the weighted two-sided interactions, the value of each entry will change over time, where the values of entries among each co-cluster tend to synchronize together gradually. (c) The final state of data matrix after shuffling, where co-clusters are intuitively popped up.

gies to simultaneously identify co-clusters with coherent values in both rows and column, allows identifying a good "checkerboard" structure. However, the assumed "checkerboard" structure (also existed in information-theoretical methods and graph-based methods) is not ideal. In the practical microarray analysis, the co-regulated and co-expressed subgroups may exist in an arbitrary location, not limited to disjoint co-clusters. 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, based on the powerful concept of synchronization, we consider the identification of co-clusters embedding in gene expression data matrices from a new aspect: **dynamic simulation**. 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.

1.1 Basic idea

Synchronization is the 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 [5]. To present, many data mining algorithms have been proposed [1, 5, 14], and have been widely used in many applications [15, 2, 24]. In this paper, we aim to identify the natural co-clusters embedding in a given gene expression data matrix based on the synchronization principle. The key idea is to view a target data matrix as a dynamical system, and each entry interacts with its neighboring rows and columns simultaneously. Since a co-cluster characterizes a subset of rows and columns with a similar pattern, we expect all entries in the co-cluster tend to achieve synchronization (entries have same value) through weighted interactions from both sides of rows and columns.

To illustrate the identification of co-clusters by the dynamic simulation, Figure 1 gives a simple example. Considering a given artificial gene expression matrix, for each entry (a_{ij}), we first search its similar genes and conditions at the instance level, respectively (see Figure 1 (b)). Then each entry will interact with its neighboring entries from the sides of rows and columns simultaneously in a weighted way. If its neighboring entries are similar, the interaction will be strong, and vice versa. The detailed two-sided weighted interaction model will be elaborated in Section 3.5. Through the weighted interactions, the difference among values of entries in a co-cluster will decrease and become zero gradually. Finally, we can identify the co-clusters by simply searching genes and conditions with the same values. Figure 1 plots the shuffled resulting data matrix after dynamic simulation, where we can observe that the co-clusters are intuitively popped out.

1.2 Contribution

By simulating the formation of co-clusters based on the weighted two-sided interaction model, CoSync has several attractive benefits for gene expression data analysis, most importantly:

- **New Viewpoint**: CoSync exploits the co-cluster structure of a given data matrix from a new point of view: *dynamic simulation*. Building upon the weighted two-sided interactions, CoSync allows spotting biologically relevant co-clusters embedding in the gene expression data matrices intuitively (cf. Fig. 4, 5, 6, Tables 3, 4).
- **High Performance**: Since the co-cluster formation is driven by the topological structure from both sides of rows and columns, CoSync faithfully finding high-quality co-clusters (cf. Figure 4, Tables 3, 4). Beyond, CoSync approach does not assume any co-cluster structure of the gene data, and thus are not limited to identify the checkerboard clusters (cf. Figure 5, Tables 3-4).
- **High Dimensionality**: In combination with non-negative matrix factorization, the large-scale gene expression data matrix can be transferred to two data matrices with reduced dimensionality. The range search of revelent genes and conditions can be speeded up significantly. This property lends CoSync to handle large-scale gene expression data matrices.

The remainder of this paper is organized as follows: In the following, we briefly survey related work. Section 3 presents our algorithm in detail. Section 4 contains an extensive experimental evaluation. We give a short discussion and finally conclude the paper in Section 5.

关于High Performance的解释，缺乏论证，难以服人，是可以针对特殊的structure？还是具有高efficiency?提到的faithfully指什么？

被指出 was not easy to understand without reading the rest of the paper. It was not obvious what was meant by "neighboring", and suggesting that there is a "detailed two-sided weighted interaction model" that would be elaborated later was also not helping to clear things up.

2. RELATED WORK

During the past several decades, many co-clustering algorithms have been proposed, e.g. [11, 7, 8], to mention a few. Due to space limitations, for a comprehensive survey of co-clustering to gene data analysis, please refer to the recent paper by Pontes et al. [22]. Here, we only provide a very brief survey on some important major research directions. In addition, we briefly introduce the closely related work of synchronization-based data mining.

Information-theoretic methods. The key idea of these methods [12, 11, 4, 27] is to consider the problem of co-clustering 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 [11], which views a nonnegative matrix as the estimate of a 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. A more generalized co-clustering framework is introduced by Banerjee et al. [4], which considers each co-clustering as an approximation of the original data matrix and quantifies the approximation error by a large class of loss functions called Bregman divergences. Recently, Song et al., [27] 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, and is often relaxed and solved by minimizing the edge weights of the vertices in different subgraphs by solving an eigenvalue system. For instance, in [10], 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 [32], where the partition is constructed by minimizing a normalized sum of edge weights between unmatched pairs of vertices of the bipartite graph. Along this line, Gao et al., [13] propose a consistent bipartite graph co-partitioning algorithm for star-structured co-clustering. For gene expression data, Kluger et al. [16] propose a spectral biclustering, which is based on the observation that checkerboard structures in matrices of expression data can be found in eigenvectors corresponding to characteristic expression patterns across genes or conditions. Similarly, QUBIC [19] considers the co-clustering problem as the search of heavy subgraphs in a bipartite graph representation by iterative expansion of a seed edge.

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 [7] 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, several enhanced approaches [30, 29] have been proposed for gene expression analysis. With the similar idea, Cho et al. [8] develop a popular co-clustering algorithm, Minimum Sum-Squared Residue Co-clustering (MSSRCC), which try to escape the local minima and resolve the degeneracy problem in partitional clustering algorithms with binormalization, deterministic spectral initialization, and incremental local search.

Synchronization-based Data Mining: Synchronization phenomena is prevalent in physical, biological, chemical, and social systems. Inspired by the powerful concept of synchronization, many synchronization models and data mining algorithms have been recently proposed and shown many desirable properties. Seliger et al. [23] discuss mechanisms of learning and plasticity in networks of phase oscillators through a generalized Kuramoto model. For bioinformatics, Kim et al. [15] propose a strategy to find groups of genes by analyzing the cell cycle specific gene expression. Aeyels et. al [1] introduce a mathematical model for the dynamics of chaos system. Shao et al. [26] propose a new synchronization-based clustering algorithm by introducing local synchronization and minimum description length principle.

Motivated by previous studies, in this paper, we view co-clustering from a dynamic perspective, and exploit the co-cluster structure of gene expression data based on synchronization principle. To our best knowledge, it is the first time to apply the concept of synchronization for co-clustering.

3. CO-CLUSTERING BY DYNAMIC SIMULATION

In this section, we present our CoSync algorithm for co-clustering of gene expression data. We start with an overview of co-clustering framework by dynamic simulation, and then present the weighted two-sided interaction model. In Section 3.5, we discuss the algorithm CoSync in detail.

3.1 A New Viewpoint for Co-Clustering

Currently, many criteria (e.g. mutual information, residue, graph cut, etc.) have been proposed to qualify co-clusterings from different points of view. In this study, we consider the identification of co-clustering from a new cluster notion: *dynamic simulation*. As stated in Section 1.1, the basic philosophy is to envision a given data matrix as a dynamical system, and exploit the change of values of entries to uncover its co-cluster structure automatically. Specifically, given a gene expression data matrix, for each entry, its similar genes and similar conditions are first searched. However, due to the large number of genes or conditions in many real-world microarray analysis, the non-negative matrix factorization is introduced to alleviate the curse of dimensionality and to speed up the range neighbor research. Afterwards, each entry interacts with its neighboring genes and conditions simultaneously in a weighted fashion. From the side of genes, we examine the distribution of expressed values of these neighboring genes for the given condition. If the expressed values are similar, it tends to believe that these genes under this condition may be co-regulated, and thus the strong interactions should be imposed. In contrast, if these values differ largely, we expect these genes to impose no or weak interactions with the entry. For the side of the condition, the strategy is the same, which we investigate the distribution of expressed values of those similar conditions for the given gene. We expect that the entry interacts with these conditions with similar values strongly, and vice versa. By

Having analyzed gene expression for over 15 years, one of the trickiest problems is that even genes in a pathway may have very expression with diverse scales. The episoln-parameter in section 3.2 does not take scaling into account, which is a very strong assumption. In general, setting this parameter is hard.

the dynamic simulation of the each entry, its value will gradually align with the values of similar genes and conditions, resulting in a same value for all entries in a co-cluster. Finally, the intuitive co-clusters embedding in gene expression data matrix are easily identified by searching blocks with the same value.

3.2 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.[17, 5, 1, 25]) for data mining 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.

Notation. Here we first 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 i^{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 entry A_{ij} corresponding to the i^th row and j^{th} column.

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

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

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 [18, 15, 26], we formulate our two-sided interaction model as follows.

Definition 2 (**Two-sided Interaction Model**) Let a_{ij} be the value of an entry A_{ij}. Given an ϵ-range row and column neighborhood, respectively, the dynamics of the entry A_{ij} of two-sided interaction from both rows and columns is defined as:

$$a_{ij}(t+1) = a_{ij}(t) + \frac{1}{2|N^r_\epsilon(a_{i.}(t))|} \cdot \sum_{p \in N^r_\epsilon(a_{i.}(t))} \sin(a_{pj}(t) - a_{ij}(t)) + \frac{1}{2|N^c_\epsilon(a_{.j}(t))|} \cdot \sum_{q \in N^r_\epsilon(a_{.j}(t))} \sin(a_{iq}(t) - a_{ij}(t)) \tag{2}$$

where $sin(\cdot)$ is the coupling function. $a_{ij}(t+1)$ describes the renewal value of the entry A_{ij} at time stamp $t+1$ $(t = (0, \ldots, T))$ during the dynamic simulation. The interaction model allows investigating the dynamics of each entry by coupling the entry discrepancies from sides of rows and column 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 entries should be considered differently. An intuitive way is to examine the distribution of expressed values of neighboring genes or conditions. Specifically, we first examine the disparities between the neighboring entries and the entry $A_{i,j}$, 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 neighboring genes are co-regulated or the neighboring conditions are similar.

Definition 3 (**Weighting Factor**) Given an ϵ-range row neighborhood $N^r_\epsilon(a_{i.}(t))$ of the entry A_{ij}, the weighting factor for neighboring genes is defined as:

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

where σ_j is the standard deviation of the difference vector $\nu_{pj} = \{abs(a_{pj} - a_{ij}) \big| p \in N^r_\epsilon(a_{i.}(t))\}$. Similarly, the weighting factor for the interactions of neighboring conditions is defined as:

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

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

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

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

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

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

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

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

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:

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

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

The salient feature of synchronization-based co-clustering is its dynamic property. During the process of interaction, the value of each entry changes in a non-linear way driven by the local data structure, and finally the values with similar genes and conditions will become the same. In contrast to search co-clusters by optimizing some criteria, the synchronization-inspired co-clusters are formed automatically without any data structure assumption.

3.3 Synchronized Co-clusters Search

After the simulation of dynamics of entries based on our weighted two-sided interaction model, the values of entries

为什么用sin()作为coupling function？
Definition 4 中的 Order Parameter怎么定义的？

缺少context！

被指出：有点套用搬用NMF算法的意味，并且结果远没有NMF应有的那么好。

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. As we are only interested in the maximum co-clusters and have some constraints on the number of genes and conditions in real-world gene expression data analysis (i.e. the co-clusters are at least have m^* genes and n^* conditions), the sub-coclusters embedding in the bigger co-clusters or very small-sized co-clusters are not needed to search. For this purpose, we first identify all distinct values of entries 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 close frequent itemset mining problem. Here we apply the popular CHARM algorithm [31]. The basic idea of this algorithm is to simultaneously explores both the itemset space (genes) and tidset space (conditions). The exploration of both the itemset and tidset space allows CHARM to use a novel search method that skips many levels to quickly identify the closed frequent itemsets, instead of enumerating many non-closed subsets. Therefore, the search is very efficient and the time complexity is $O(l \cdot |C|)$, where l is the average tidset length, and C is the set of all closed frequent itemsets [31].

3.4 Handling High-dimensional Data via Non-negative Matrix Factorization

For weighted two-sided interactions, the corresponding neighboring rows and columns for a given entry are required. For high-dimensional data (e.g. large number of genes or large number of conditions), due to curse of dimensionality, the neighbor search is usually infeasible as the distances among genes or conditions are almost equal. Currently, some different distance functions, such as cosine distance and distribution-based divergence, are proposed to alleviate the effect. In this paper, we introduce a new strategy via non-negative matrix factorization (NMF), which transfers a data matrix into two non-negative matrices with lower dimensions. The search of similar genes and conditions with given range can be easily computed on the two matrices respectively.

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

$$\mathbf{A} \approx \mathbf{W}\mathbf{H}^T \tag{9}$$

This low-rank approximation can be achieved based on different distance or divergence measures. In this study, we apply the most widely used Frobenius norm by solving the following equation:

$$\min_{\mathbf{W},\mathbf{H}} f(\mathbf{W},\mathbf{H}) = ||\mathbf{A} - \mathbf{W}\mathbf{H}^T||_F^2 \tag{10}$$

$$s.t. \mathbf{W} \geq 0, \mathbf{H} \geq 0$$

In the context of microarray analysis, $\mathbf{A}$ 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 $A(:,i) = W \cdot H(:,i)$ (Figure 2). The two matrices allow capturing the 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. This transformation has two potential benefits: (1) alleviates the effect of curse of dimensionality and more importantly, allows for a good neighboring search on data set with high dimensionality; (2) saves the computation time as the search is performed on a lower dimensional space.

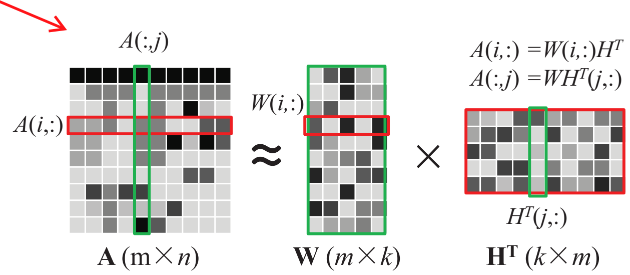


Figure 2: Illustration of range search via non-negative matrix factorization. Here each row vector $A(i,:)$ can be written as the multiplication of the corresponding row vector $W(i,:)$ of W and the whole matrix H^T. Thus, the range search on rows can be performed on the matrix W, and the range search on columns can be worked on matrix H.

3.5 CoSync Algorithm

Building upon the interaction model (cf. Eq. (7)), the dynamical change of values for each entry can be simulated. The identification of co-clusters by dynamic simulation mainly involves the following steps:

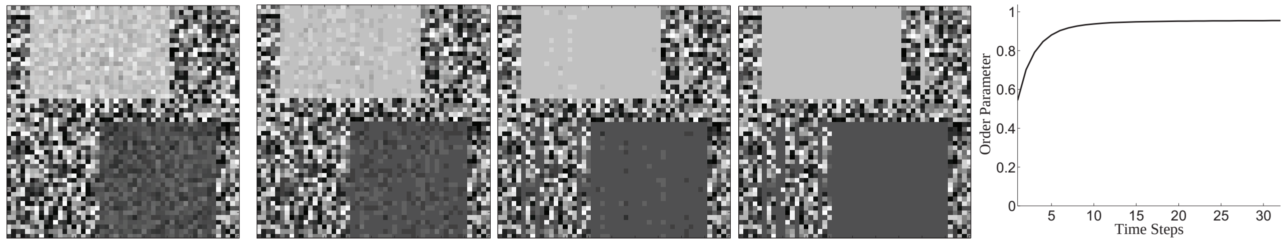
1. **Initialization.** Given a gene expression data matrix A, scale each column to have unit L2-norm. If the size of A is large, it is factorized into two matrices W and H via non-negative matrix factorization.

2. **Dynamic Simulation.** For each entry A_{ij}, we simulate its value change over time. Specifically, its neighboring genes and conditions are first searched based on the matrix A (or the matrix W and H respectively, if necessary). Then the renewal value $a_{ij}(t+1)$ of the entry at time stamp $t+1$ is calculated based on the Eq. (7). Thanks to the topological-driven weighted influences from rows and columns simultaneously, the values of entries in a co-cluster tend to align to the same value.

3. **Co-clusters Search.** The search of co-clusters is transferred into a maximum close frequent itemset mining problem. Here CHARM algorithm is applied to find these synchronized co-clusters efficiently.

For illustration, Fig. 3(a)-(d) shows the detailed process of dynamical change of a artificial data matrix from $T = 0$ to $T = 50$. Fig. 3(a) plots the original data matrix. From $T = 1$ to $T = 30$, the value of each entry will change dynamically based on the influence from its similar genes and similar conditions simultaneously. Finally, entries within a co-cluster will synchronize together and share the same value. During the process, the order parameter, characterizing the

Lack complexity
analysis



(a) Original data matrix (b) T = 1 (c) T = 5 (d) T = 50 (e) Order parameter

Figure 3: Illustration of range search via non-negative matrix factorization. Here each row vector $A(i,:)$ can be written as the multiplication of one vector from $W(i,:)$ and the whole matrix H^T. Thus, the range search on rows can be performed on the matrix W, and the range search on columns can be worked on matrix H.

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 $\epsilon$-neighborhood $N^r_\epsilon(a_{i.})$ on $W$;
        else
            Search its $\epsilon$-neighborhood $N^r_\epsilon(a_{i.})$ 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 $N^c_\epsilon(a_{.j})$ on $H$;
        else
            Search its $\epsilon-$neighborhood $N^c_\epsilon(a_{.j})$ 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: $C$;
```

level of local synchronization will gradually converge (Figure 3)(e). Finally, the Pseudocode of CoSync is given in Algorithm 1.

4. EXPERIMENTAL EVALUATION

Data sets are too old, and the scales are too small!

Table 1: Statistics of four real-world gene expression data sets.

Data Sets	#Genes	#Samples	Class names	Class distri.
Colon	2000	62	Normal/Tumor	20/42
Leukemia	7129	72	ALL/AML	47/25
Lung	12533	181	ADCA/MPM	150/31
MLL	12582	72	ALL/AML/MLL	24/28/20

4.1 Experiment Setup

vary between different clusters!

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 proof the concepts, we generate two types of co-cluster structures: checkerboard co-clusters and a flexible co-clusters in the data matrices. For each implanted co-cluster, the values of entries 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 keep set 0.1. For the other entries, the values follows the uniform distribution $U(0, \frac{\pi}{2})$.

Real-world Data. To evaluate the performance 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 [8]. Table 1 lists statistics of the four gene expression data sets.

Evaluation Metrics. Comparing the results of different clustering algorithms with respect to effectiveness is a non-trivial problem, especially if different algorithms produce results with different numbers of clusters. For evaluating co-clustering algorithms, it becomes more difficult since traditional co-clustering algorithms only tend to produce block clusters while the co-clusters yielded by CoSync is more flexible. Here, to evaluate the clustering results by different algorithms, we follow two ways. For the four gene expression data sets, since the ground truth of conditions (e.g. different types of samples) are available, we report the cluster

[1] Data sets are publicly available at http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html, corresponding gene IDs are provided at: http://www.cs.utexas.edu/users/dml/Data/cocluster/

What the red lines stand for are not noted

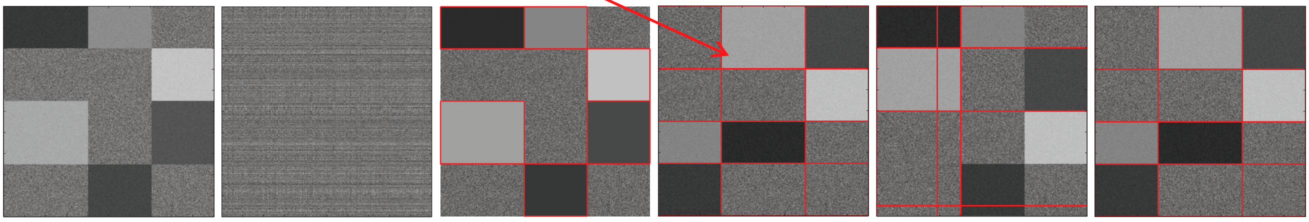

(a) Original Data (b) Shuffled data (c) CoSync (d) ITCC (e) MSSRCC (f) Spectral

Figure 4: Comparing co-clustering algorithms on an artificial synthetic data set with checkerboard co-cluster structure.

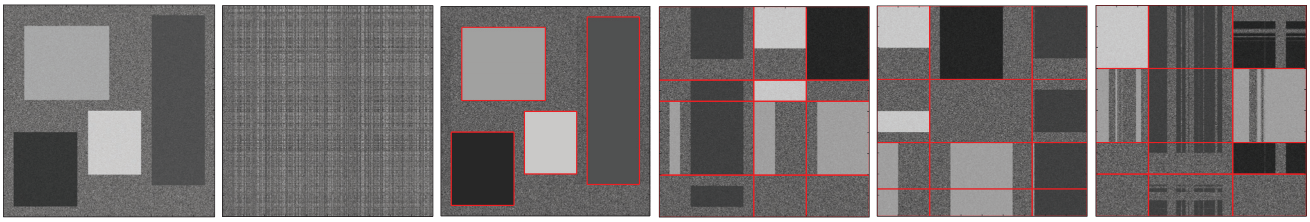

(a) Original Data (b) Shuffled data (c) CoSync (d) ITCC (e) MSSRCC (f) Spectral

Figure 5: Comparing co-clustering algorithms on an artificial synthetic data set with flexible co-cluster structure.

Table 2: 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

size, precision and recall for each co-cluster. To evaluate the genes in a co-cluster, we can evaluate the biological significance of obtained clusters with the help of the Gene Ontology database [3], which provides the ontology of defined terms representing gene product properties on three vocabularies of annotations: Molecular Function, Biological Process and Cellular Component. Researchers can apply P-value to demonstrate the biological significance, which is defined as the probability to observe by chance at least x elements at the intersection between the query set and the reference set [6].

Selection of comparison methods. To extensively evaluate the proposed algorithm CoSync, we compare its performance to several representatives of co-clustering paradigms: the typical information-theoretic co-clustering algorithm: ITCC [11], a well-known residue-based co-clustering algorithm MSSRCC [8] and a graph-based co-clustering algorithm via spectral method [16] (called it as Spectral in this paper). We have implemented CoSync in Java, which is available at: http://staff.uestc.edu.cn/shaojunming/files/2016/02/CoSync.zip. The sourcecode of ITCC and MSSRCC are available at: http://www.cs.utexa-s.edu/users/dml/Software/cocluster.html. The Spectral algorithm is available in the *scikit* package (http://scikit-learn.org/stable/index.html). All experiments have been performed on a workstation with 2.4 GHz CPU and 8 GB RAM.

4.2 Proof of Concept

We start the evaluation with synthetic data sets to facilitate presentation and demonstrate the properties of CoSync.

First, we examine whether CoSync allows finding co-clusters with assumed checkerboard structure like most existing co-clustering algorithms in a synthetic data set (See Figure 4). Based on the synchronization-inspired dynamic simulation, we can observe that the implanted co-clusters on this data set can be effectively identified by CoSync. For comparing algorithms of ITCC, MSSRCC and Spectral, we specify the number of row clusters and column clusters to 4 and 3, respectively. For ITCC and Spectral, we can see that they

Table 3: Co-clustering performance of different algorithms on gene expression data sets from sample-based evaluation.

	CoSync			ITCC		MSSRCC		Spectral	
	#C	Prec.	Recall	#C	Avg.	#C	Avg.	#C	Avg.
Colon	11	0.951	0.661	2	0.815	2	0.857	2	0.726
Leukemia	28	0.956	0.708	2	0.946	2	0.931	2	0.736
Lung	23	1.000	0.945	2	0.854	2	0.997	2	1.000
MLL	23	0.989	0.667	3	0.825	3	0.934	3	0.639

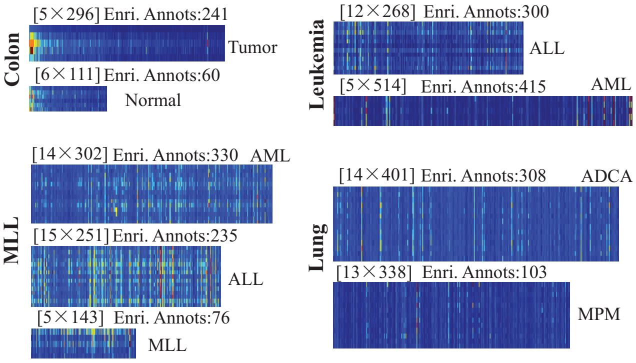


Figure 6: The representative co-clusters discovered by CoSync on the four gene expression data sets. Here each co-cluster corresponds to a particular conditional type, and meanwhile, the corresponding genes are co-regulated, indicating by many enriched functional annotations.

identify these checkerboard co-clusters while MSSRCC fails to find some co-clusters. However, for ITCC and MSSRCC, they view all 12 clusters as co-clusters while there are actually only 6 co-clusters existed.

To further evaluate the performance of CoSync, Figure 5 plots the clustering results of different algorithms on a synthetic data set with a flexible co-cluster structure, where the genes or conditions have the overlapping (yet limited to co-clusters with overlapping of rows and columns simultaneously). 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. However, for CoSync, the formation of co-clusters is driven by intrinsic data structure, and thus supports to find flexible co-clusters, which is essential to practical gene expression data analysis.

4.3 CoSync on Gene Expression Data

In this section, we evaluate the performance of CoSync on several real-world gene expression data sets.

4.3.1 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. Figure 2 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) expect 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. In summary, we notice that CoSync algorithm achieves all we may expect: (1) CoSync automatically finds the number of co-clusters; (2) It detects natural co-clusters, which all samples in each cluster match with corresponding type (almost with perfect precision of 100%) ; (3) The co-clusters are not limited to checkerboard structure, yet the natural co-clusters embedded in gene expression data sets.

被指出：
Could you clarify the phrase "limited to co-clusters with overlapping of rows and columns simultaneously"? I don't understand how else could they overlap?

except

To further evaluate the performance of CoSync, ITCC, MSSRCC and Spectral algorithms are also employed to find co-clusters on these data sets. Table 3 gives a summary of average sample clustering performance. Here, as CoSync allows the one-side overlapping clustering, we use the average precision and recall of all deserved co-clusters as the average sample clustering performance. Based on the experimental setup, we can see CoSync shows its superiority over other comparing algorithms.

derived ?

4.3.2 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 of functional annotations of the gene set in each co-cluster, which is publicly available at https://david.ncifcrf.gov/. Table 4 gives a summary of enriched top four annotations for the first three big co-clusters. Here the Fisher Exact p-value is used, and smaller p-values indicate a better enrichment (P-value < 0.05 are considered statistically significant). Due to space limitation, we list only the first four enriched categories (sorted by P-values) for the top three co-clusters on each data set. We can observe that the gene clusters allow a good enrichment for the three categories (with both a large number of enriched annotations and corresponding small p-values). The results indicate that CoSync also allows finding high-quality co-clusters from gene aspect (biological significant clusters). Figure 6 further depicts top two co-clusters found on each data set, which are evaluated on sample and gene aspects, simultaneously.

for

5. DISCUSSION AND 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 entries with a weighted two-sided mode, the values of these entries of co-regulated genes under a particular set of experimental conditions tend to synchronize together to automatically form co-clusters. Therefore, instead of optimizing some criteria such as mutual information, residue and graph cut, CoSync do search high-quality co-clusters without any cluster structure assumption, only driving by the intrinsic data structure. Beyond, although CoSync is a synchronization-based clustering, it

对于P-value的讨论不够，不能服人

The experimental result is not convincing at all. There needs to be biological validation of the clusters found. Saying that there is a good p-value and have apparently meaningful annotations from enrichment is too weak.

Table 4: Enrichment of the GO functional annotations of Gene Clusters detected by CoSync. Here BP: "Biological Process" and CC: "Cellular Component". ***Annotations*** **indicates the total number of enriched annotations with p-value <0.05.**

Data Sets	CluID	#Genes	#Annotations	Top Enriched Annotations	Count	Percentage (%)	P-Value
Colon	C1	296	214	CC:cytosol	34	23.61%	5.84E-07
				CC:plasma membrane part	42	29.17%	3.01E-05
				BP:negative regulation of molecular function	13	9.03%	7.69E-05
				BP:striated muscle tissue development	8	5.56%	1.34E-04
	C2	138	157	CC:plasma membrane part	22	37.29%	4.81E-05
				CC:plasma membrane	28	47.46%	3.87E-04
				CC:cytosol	15	25.42%	5.20E-04
				CC:actin cytoskeleton	7	11.86 %	6.69E-04
	C3	111	65	BP:regulation of system process	8	15.69%	7.55E-05
				BP:heart development	6	11.76%	7.81E-04
				BP:negative regulation of molecular function	7	13.73%	8.82E-04
				CC:striated muscle thin filament	3	5.88%	1.13E-03
Leukemia	C1	268	300	BP:response to endogenous stimulus	27	10.67%	2.47E-09
				CC:plasma membrane part	71	28.06%	4.54E-09
				CC:integral to plasma membrane	48	18.97%	5.34E-09
				BP:response to hormone stimulus	25	9.88%	7.40E-09
	C2	514	415	CC:cytosol	123	24.45%	2.08E-28
				CC:intracellular organelle lumen	142	28.23%	1.10E-26
				CC:organelle lumen	142	28.23%	1.13E-25
				CC:membrane-enclosed lumen	142	28.23%	8.10E-25
	C2	464	539	CC:plasma membrane part	114	25.79%	2.37E-11
				CC:integral to plasma membrane	73	16.52%	2.72E-10
				CC:intrinsic to plasma membrane	74	16.74%	2.95E-10
				CC:plasma membrane	160	36.20%	2.30E-09
Lung	C1	401	308	CC:cytosol	88	23.34%	4.71E-17
				CC:intracellular organelle lumen	87	23.08%	2.18E-09
				CC:organelle lumen	88	23.34%	2.96E-09
				CC:membrane-enclosed lumen	89	23.61%	3.47E-09
	C2	338	183	CC:cytosol	62	19.68%	1.74E-09
				BP:response to organic substance	37	11.75%	6.39E-07
				BP:negative regulation of apoptosis	24	7.62%	1.07E-06
				BP:negative regulation of programmed cell death	24	7.62%	1.35E-06
	C3	264	105	BP:translational elongation	29	11.69%	2.29E-27
				CC:cytosol	75	30.24%	1.13E-23
				BP:translation	39	15.73%	4.81E-22
				CC:ribosome	30	12.10%	3.84E-19
MLL	C1	302	330	BP:RNA splicing, via transesterification reactions with bulged adenosine as nucleophile	22	7.46%	1.28E-12
				BP:RNA splicing, via transesterification reactions	22	7.46%	1.28E-12
				BP:nuclear mRNA splicing, via spliceosome	22	7.46%	1.28E-12
				CC:membrane-enclosed lumen	73	24.75%	1.58E-10
	C2	251	235	BP:RNA splicing	28	11.43%	3.37E-14
				BP:mRNA metabolic process	31	12.65%	8.00E-14
				BP:mRNA processing	29	11.84%	9.54E-14
				CC:cytosol	57	23.27%	1.05E-12
	C3	211	127	CC:organelle membrane	36	17.48%	3.86E-08
				CC:organelle inner membrane	19	9.22%	6.29E-08
				CC:organelle envelope	26	12.62%	6.34E-08
				CC:envelope	26	12.62%	6.75E-08

largely differs from traditional algorithms. First, to the first time, CoSync aims at discovering the co-clusterings based on the synchronization principle; and secondly, to handle the new problem, CoSync proposes a weighted two-sided interaction model to simulate the synchronization process, which appears to be the main difference. Extensive experiments further demonstrate that CoSync has some advantages compared to several state-of-the-art methods. In future work, we plan to focus on extending our approach to multi-way clustering based on the intuitive interaction model.

Reference 3,7,23 need
capitalisation fixing

6. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (61403062, 61433014, 61502078), China Postdoctoral Science Foundation (2015M580786, 2014M552344, 2015T80973).

7. REFERENCES

[1] D. Aeyels and F. De Smet. A mathematical model for the dynamics of clustering. *Physica D: Nonlinear Phenomena*, 237(19):2517–2530, 2008.

[2] A. Arenas, A. Diaz-Guilera, and C. J. Perez-Vicente. Plasticity and learning in a network of coupled phase oscillators. *Phys. Rev. Lett.*, 96, 2006.

[3] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[4] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *ACM SIGKDD*, pages 509–514, 2004.

[5] C. Böhm, C. Plant, J. Shao, and Q. Yang. Clustering by synchronization. In *ACM SIGKDD*, pages 583–592, 2010.

[6] S. Brohée, K. Faust, G. Lima-Mendez, G. Vanderstocken, and J. van Helden. Network analysis tools: from biological networks to clusters and pathways. *Nature protocols*, 3(10):1616–1629, 2008.

[7] Y. Cheng and G. M. Church. Biclustering of expression data. In *Ismb*, volume 8, pages 93–103, 2000.

[8] H. Cho and I. S. Dhillon. Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. *ACM Transactions on Computational Biology and Bioinformatics*, 5(3):385–400, 2008.

[9] J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, 1997.

[10] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *ACM SIGKDD*, pages 269–274, 2001.

[11] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *ACM SIGKDD*, pages 89–98, 2003.

[12] R. El-Yaniv and O. Souroujon. Iterative double clustering for unsupervised and semi-supervised learning. In *ECML*, pages 121–132. 2001.

[13] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *ACM SIGKDD*, pages 41–50, 2005.

[14] L. Hong, S.-M. Cai, J. Zhang, Z. Zhuo, Z.-Q. Fu, and P.-L. Zhou. Synchronization-based approach for detecting functional activation of brain. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(3):033128, 2012.

[15] C. S. Kim, C. S. Bae, and H. J. Tcha. A phase synchronization clustering algorithm for identifying interesting groups of genes from cell cycle expression data. *BMC bioinformatics*, 9(1):56, 2008.

[16] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, 13(4):703–716, 2003.

[17] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International symposium on mathematical problems in theoretical physics*, pages 420–422, 1975.

[18] Y. Kuramoto. *Chemical oscillations, waves, and turbulence*. Courier Dover Publications, 2003.

[19] G. Li, Q. Ma, H. Tang, A. H. Paterson, and Y. Xu. Qubic: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic acids research*, page gkp491, 2009.

[20] W. Lin, Y. Zhao, S. Y. Philip, and B. Deng. An effective approach on overlapping structures discovery for co-clustering. In *Web Technologies and Applications*, pages 56–67. 2014.

[21] A. L. V. Pereira and E. R. Hruschka. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowledge-Based Systems*, 82:11–19, 2015.

[22] B. Pontes, R. Giráldez, and J. S. Aguilar-Ruiz. Biclustering on expression data: A review. *Journal of biomedical informatics*, 57:163–180, 2015.

[23] P. Seliger, S. C. Young, and L. S. Tsimring. Plasticity and learning in a network of coupled phase oscillators. *Phys. Rev. E*, 65:137–185, Jan. 2002.

[24] J. Shao, Z. Ahmadi, and S. Kramer. Prototype-based learning on concept-drifting data streams. In *ACM SIGKDD*, pages 412–421, 2014.

[25] J. Shao, Z. Han, Q. Yang, and T. Zhou. Community detection based on distance dynamics. In *ACM SIGKDD*, pages 1075–1084, 2015.

[26] J. Shao, X. He, C. Bohm, Q. Yang, and C. Plant. Synchronization-inspired partitioning and hierarchical clustering. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):893–905, 2013.

[27] 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*, 25(6):1227–1239, 2013.

[28] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *WWW*, pages 21–30, 2012.

[29] J. Yang, H. Wang, W. Wang, and P. Yu. Enhanced biclustering on expression data. In *IEEE Symposium on Bioinformatics and Bioengineering*, pages 321–327, 2003.

[30] J. Yang, W. Wang, H. Wang, and P. Yu. δ-clusters: Capturing subspace correlation in a large data set. In *ICDE*, pages 517–528, 2002.

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

[32] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *CIKM*, pages 25–32, 2001.

Ref 32 repeats Technical report. Just
check and clean these up.