A Tensor-Based Markov Chain Method for Module Identification from Multiple Networks

Chenyang Shen
Department of Mathematics
Hong Kong Baptist University
Hong Kong
Email: 12466743@life.hkbu.edu.hk

Shuqin Zhang
School of Mathematical Sciences
Fudan University
Shanghai, 200433, China
Email: zhangs@fudan.edu.cn

Michael K. Ng
Department of Mathematics
Hong Kong Baptist University
Hong Kong
Email: mng@math.hkbu.edu.hk

***Abstract*—The interactions among different genes, proteins and other small molecules are becoming more and more significant and have been studied intensively nowadays. One general way that helps people understand these interactions is to analyze networks constructed from genes/proteins. In particular, module structure as a common property of most biological networks has drawn much attention of researchers from different fields. In most cases, biological networks can be corrupted by noise in the data and the corruption may cause mis-identification of module structure. Besides, some structure may be destroyed when improper experimental settings are built up. Thus module structure may be unstable when one single network is employed. In this paper, we consider employing multiple networks for consistent module detection in order to reduce the effect of noise and experimental setting. Instead of considering different networks separately, our idea is to combine multiple networks together by building them into tensor structure data. Then given any node as prior label information, tensor-based Markov chains are constructed iteratively for identification of the modules shared by the multiple networks. In addition, the proposed tensor-based Markov chain algorithm is capable of simultaneously evaluating the contribution from each network. It would be useful to measure the consistency of modules in the multiple networks. In the experiments, we test our method on two groups of gene co-expression networks from human beings. We also validate the modules identified by the proposed method.**

I. Introduction

Genes/proteins function inference from complex biological systems has become a very important problem. In the past few years, it has been studied intensively by people from multiple fields. Among a large number of diverse research tasks for genes/proteins, one important aspect is to figure out the complex relations between genes/proteins. Biological networks representing the co-expression, regulation, interaction and so on seem to have become a general tool for computational genes/proteins relation analysis. More specifically, the network is constructed by making use of genes (or proteins) as nodes while the edges as connections. By analyzing such genes (or proteins) network, it is possible for researchers to identify and interpret relations between genes (or proteins) such that some important phenomena in a complex biological systems can be inferred correspondingly, see [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. Among a large number of approaches, one substantial way in network analysis is to explore a group of similar nodes which are closely related with each other. These nodes combining together form a densely connected subgraph which is usually called module. The module structure is common in biological networks and it is very useful especially in the study on function of genes and proteins in some complex biological systems. For instance, the large size of genes (or proteins) network can be reduced by dividing the network into different modules and thus much easier to handle. Once a module is identified, we may infer the function of unknown genes by other genes which we know well since the genes/proteins in the same module are more likely to play similar roles. In the literature, there are a lot of works accomplished for identifying the functional module for a single graph [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

However, biological networks constructed by real data are often corrupted by noise occurring in data: some edges which should exist are removed due to the noise while in other place, some extra edges are introduced into the networks. If so, the modules are possibly to be mis-identified. Moreover, the selection of different thresholding values during the process of constructing networks may introduce the inconsistency of module structure. One intuitive idea to reduce the effect of both noise and parameters is integrating multiple networks composed of the same set of nodes, see [21], [22], [23], [24], [25], [26].

In this paper, we consider the module identification problem based on multiple networks at the same time. Our main idea is to formulate multiple networks into order three tensor which is actually a three dimensional data array. Note that each network is usually expressed by matrix which can be also treated as two dimensional data array. Then by aligning each network as a slice, it is natural to integrate them into a cubic like tensor. Following the structure of networks, two directions of the tensor data are based on genes while the rest one dimension corresponds to different networks. Therefore, tensor data is capable of preserving all the information in the multiple networks. Then, our goal is to identify the module structure embedded in tensor-based multiple networks. Our numerical algorithm is mainly motivated by [27] in which the authors formulate a Markov chain by normalizing the adjacency matrix among objects and iteratively learn a label indicator of objects for multi-instance multi-label learning (MIML) tasks. We remark that each column of the label indicator is guaranteed

2014 The 8th International Conference on Systems Biology (ISB)
978-1-4799-7294-4/14/$31.00 ©2014 IEEE

Qingdao, China, October 24–27, 2014

to be a probability distribution vector. It is also possible to formulate module identification task in the classification point of view: by considering each node as an object in classification problem, we are seeking for nodes similar to each other as a module in a broader network. Thus, we may also determine the module structures by labeling the nodes. The main difficulty lies in is that instead of formulating the problem under the matrix framework, we target at algorithm that can handle the tensor data. To overcome this difficulty, a novel two-stage iterative scheme is proposed. As preparation for the algorithm, two probability transition tensors should be generated. One of them should be normalized along the direction corresponding to nodes while the other is normalized with respect to different networks. Then in the first stage, we may multiply two probability transition tensors by the initial guess of label indicator vector such that two probability transition matrices are generated with one dimension of them corresponding to nodes and the other referring to multiple networks. Then in the second stage, by fixing these two matrices, we are able to form two Markov chains: one is to determine the transition probabilities from multiple networks to nodes incorporating with prior information and the other computes the transition probabilities from nodes to different networks. After convergence is achieved in the second stage, we may make use of the converged label indicator for module to update the two transition probability matrices. Repeating the two-stage process until global steady state is reached, we may identify the module structure in the label indicator and also, we are able to tell the contributions of different networks as well as the consistency of the modules across multiple networks. Later we illustrate the effect of the proposed tensor-based module identification method by the experimental results on two different gene data sets. One is co-expression networks constructed from three cancers and the other is from different tissues of morbidly obese patients. It can be seen from the results that our proposed method can identify the valid consistent module structures in multiple networks.

The rest of this paper is organized as follows: in section two, details of the proposed methodology will be introduced. Then, we will report the experimental results by employing the proposed method in section 3. Finally, in section 4, some concluding remarks will be given.

II. Proposed Method

Before we introduce proposed method, some basic notations based on tensor multiplication will be given. Then we will briefly review a Markov chain based multi-instance multi-label learning algorithm (Markov-MIML) [27], which inspires us to tackle the module identification problem by formulating Markov chains. After that we will present our proposed tensor-based Markov chain method for module identification in multiple networks.

A. Preliminary notations

Here we introduce some preliminary notations which will be used later in this paper. First, let $\mathcal{A}$ indicates tensor data while $\mathbf{A}$ and $\mathbf{a}$ are utilized to represent matrix and vector, respectively. An order m tensor is expressed by $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$ and $\mathcal{A}_{i_1,i_2,\cdots,i_m}$ are employed to represent the element at $(i_1, i_2, \cdots, i_m)$ of tensor $\mathcal{A}$. On the other hand, $[\mathbf{A}]_{i,j}$ indicates the element at (i, j) position of matrix A and $[\mathbf{a}]_i$ represents the i-th element of vector $\mathbf{a}$. In addition, we introduce the mode multiplication between tensor and matrix: for $p = 1, 2, \cdots, m$, let $\mathbf{B} \in \mathbb{R}^{n_p \times q}$, the mode-$p$ multiplication of $\mathcal{A}$ and $\mathbf{B}$ is written as

$$\mathcal{C} = \mathcal{A} \times_p \mathbf{B}.$$

where $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_{p-1} \times q \times n_{p+1} \times \cdots \times n_m}$ and

$$\mathcal{C}_{i_1,i_2,\cdots,i_m} = \sum_{j=1}^{n_p} \mathcal{A}_{i_1,\cdots,i_{p-1},j,i_{p+1},\cdots,i_m} [\mathbf{B}]_{j,i_p}.$$

Note that here $i_p = 1, 2, \cdots, q$. Similarly, mode multiplication can be also performed between tensor and vector. For $\mathbf{b} \in \mathbb{R}^{n_p}$,

$$\mathcal{D} = \mathcal{A} \times_p \mathbf{b}$$

where $\mathcal{D} \in \mathbb{R}^{n_1 \times \cdots n_{p-1} \times n_{p+1} \times \cdots n_m}$. Moreover,

$$\mathcal{D}_{i_1,\cdots,i_{p-1},i_{p+1},\cdots,i_m} = \sum_{j=1}^{n_p} \mathcal{A}_{i_1,\cdots,i_{p-1},j,i_{p+1},\cdots,i_m} [\mathbf{b}]_j.$$

B. Markov-MIML algorithm

Markov-MIML algorithm is designed to tackle the so called multi-instance multi-label problem: a kind of classification problem in which each object is described by several different instances and can simultaneously belong to more than 1 class. In the Markov-MIML algorithm [27], the first step is to construct the adjacency matrix among instances and then transfer the affinity information from instance level to object level. Assume $\mathbf{S}$ is the similarity matrix among objects which can be also treated as nearest neighbors affinity for all the objects. Then by normalizing each column sum of $\mathbf{S}$ to be one, a transition probability matrix $\mathbf{M}$ is obtained. Then the Markov-MIML iterative algorithm is given in (1):

$$\mathbf{X}(t+1) = (1-\alpha)\mathbf{M}\mathbf{X}(t) + \alpha\mathbf{P}, \quad t = 0, 1, 2, \cdots, \quad (1)$$

where $\mathbf{P}$ contains label information of training objects and α is a parameter to balance the label information from neighbors and the given label training data $\mathbf{P}$. $\mathbf{X}(t)$ is the label-indicator at the t-th iteration. The convergence analysis given in [27] demonstrates that the iterate $\mathbf{X}(t)$ converges to a limiting vector $\hat{\mathbf{X}}$ which can be used for MIML testing object prediction. In [27], it has been shown this algorithm is very effective.

The Markov-MIML method can be also applied to biological network analysis. When replacing the affinity matrix among objects by co-expression adjacency matrix in biological network, we may extract the module structure embedded in an individual network by incorporating the prior label information. However, due to the concern that corruption of noise may affect the result in single network analysis, we are more interested in considering the module identification task for multiple networks. Together, these two reasons motivate us

978-1-4799-7294-4/14/$31.00 ©2014 IEEE

to propose tensor-based Markov chain method from multiple networks analysis.

C. Tensor-based module identification method

In this subsection, we will introduce the proposed tensor-based module identification method. Assume we have a number of n_2 co-expression networks constructed by the same set of n_1 nodes which can be genes, proteins and some other small molecules. Note that any individual network can be expressed by n_1-by-n_1 adjacency matrix $\mathbf{A}_k$ $(k = 1, 2, \cdots, n_2)$. Then we may utilize order 3 tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_1 \times n_2}$ to express the multiple networks. More precisely, the tensor data is generated by setting the i, j element of k-th network to i, j position of k-th layer in the tensor $\mathcal{A}$: $\mathcal{A}_{i,j,k} = [\mathbf{A}_k]_{i,j}$. Note that our idea is to use Markov chains to extract consistent module structure embedded in the multiple networks, the transition probability tensors are required to be constructed. In fact, by normalizing $\mathcal{A}$ along different directions, we generate normalized tensors $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$ as transition probability tensors:

$$\sum_{i=1}^{n_1} \mathcal{A}^{(1)}_{i,j,k} = 1, \sum_{k=1}^{n_2} \mathcal{A}^{(2)}_{i,j,k} = 1,$$

Note that one dimension we chosen corresponds to nodes while the other one refers to multiple networks. In addition, incorporating with some prior information, we can one step further formulate the problem by high order Markov Chains:

$$\mathbf{x} = (1-\alpha)\mathcal{A}^{(1)} \times_2 \mathbf{x} \times_3 \mathbf{y} + \alpha\mathbf{p}; \quad (2)$$
$$\mathbf{y} = \mathcal{A}^{(2)} \times_1 \mathbf{x} \times_2 \mathbf{x}. \quad (3)$$

Here, our aim is to seek for $\mathbf{x}$ and $\mathbf{y}$ satisfying (2) and (3) when $\mathbf{x}$ identifies the nodes that belong to the same module as the prior nodes while $\mathbf{y}$ offers evaluations of contributions from different networks to the module. We remark that proposed tensor based model is unsupervised, such that no label information of the prior nodes is required. In addition, any node (or nodes) in networks can be selected as prior, for instance, if a set of nodes $S = \{s_1, s_2, s_3, \cdots, s_l\} \subset 1, 2, \cdots n_1$ are chosen to be prior in the Markov chains, then set $[\mathbf{p}]_S = 1/l$ and all the other entries to be zero. The resulting $\mathbf{x}$ will identify the modules containing nodes in set S. One step further, if all the module structures are required to be detected, we may go through all the nodes in the multiple networks to be prior information. More precisely, denote $\mathbf{I}_{\mathbf{n_1}} \in \mathbb{R}^{n_1 \times n_2}$ as the identical matrix, we are able to extract all the modules embedded in the networks by setting $\mathbf{p}$ to be each column of $\mathbf{I}_{\mathbf{n_1}}$ one by one. Moreover, $\alpha \in [0, 1]$ here is set to balance the contribution between multiple networks and prior information.

In order to reach the global steady state of system in (2)(3), we propose a two stage iterative scheme. More precisely, when we fix $\mathbf{M}_1 = \mathcal{A}^{(1)} \times_2 \mathbf{x}$ and $\mathbf{M}_2 = (\mathcal{A}^{(2)} \times_2 \mathbf{x})^T$ We may rewrite (2)(3) into the following system:

Algorithm 2.1:

Input: $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \mathbf{x}(0), \mathbf{y}(0), \mathbf{p}, \alpha$ and tolerance ε
Output: $\mathbf{x}^*$ and $\mathbf{y}^*$
Procedure
1. Set $t = 1$;
2. Compute $\mathbf{M_1}(t-1) = \mathcal{A}^{(1)} \times_2 \mathbf{x}(t-1); \mathbf{M_2}(t-1) = (\mathcal{A}^{(2)} \times_2 \mathbf{x}(t-1))^T$;
3. Set $k = 0, \mathbf{x}_0 = \mathbf{x}(t-1), \mathbf{y}_0 = \mathbf{y}(t-1)$;
4. Compute $\mathbf{x}_k = (1-\alpha)\mathbf{M}_1(t-1)\mathbf{y}_{k-1} + \alpha\mathbf{p}$, and $\mathbf{y}_k = \mathbf{M}_1(t-1)\mathbf{x}_k$;
5. If $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| < \varepsilon$ and $\|\mathbf{y}_k - \mathbf{y}_{k-1}\| < \varepsilon$, set $\mathbf{x}(t) = \mathbf{x}_k$, $\mathbf{y}(t) = \mathbf{y}_k$;
Otherwise set $k = k + 1$ and goto Step 4.
6. If $\|\mathbf{x}(t) - \mathbf{x}(t-1)\| < \varepsilon$ and $\|\mathbf{y}(t) - \mathbf{y}(t-1)\| < \varepsilon$, set $\mathbf{x}^* = \mathbf{x}(t)$, $\mathbf{y}^* = \mathbf{y}(t)$;
Otherwise set $t = t + 1$ and goto Step 2.

$$\mathbf{x} = (1-\alpha)\mathbf{M}_1\mathbf{y} + \alpha\mathbf{p}; \quad (4)$$
$$\mathbf{y} = \mathbf{M}_2\mathbf{x}. \quad (5)$$

$\mathbf{M}_1$ and $\mathbf{M}_2$ are two transition probability matrices, thus (4) and (5) give two standard Markov chains which can be solved iteratively according to following recursive formulas:

$$\mathbf{x}_{k+1} = (1-\alpha)\mathbf{M}_1\mathbf{y}_k + \alpha\mathbf{p}; \quad (6)$$
$$\mathbf{y}_{k+1} = \mathbf{M}_2\mathbf{x}_k. \quad (7)$$

Once (6), (7) converge with results $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, it is possible to update the probability transition matrices by:

$$\hat{\mathbf{M}}_1 = \mathcal{A}^{(1)} \times_2 \hat{\mathbf{x}}; \hat{\mathbf{M}}_2 = (\mathcal{A}^{(2)} \times_2 \hat{\mathbf{x}})^T.$$

Replacing $\mathbf{M}_1, \mathbf{M}_2$ in (6) and (7) by $\hat{\mathbf{M}}_1, \hat{\mathbf{M}}_2$, we are allowed to once again update $\mathbf{x}, \mathbf{y}$ accordingly. Then by repeating the above process until convergence, the nonlinear systems in (2), (3) can be solved eventually. Note that once the initial guess of $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{p}$ are given by probability distribution vectors, the converging results are also guaranteed to be probability distribution vectors.

Now, we are ready to present the proposed algorithm, see Algorithm 2.1, for solving $\mathbf{x}$ and $\mathbf{y}$ in (2)(3).

Based on the resulting probability distribution vectors $\mathbf{x}^*$ and $\mathbf{y}^*$, we are able to identify the common module structure across multiple networks by two simple steps. The first step is to identify the module structure by observing $\mathbf{x}^*$: a module should be constructed by the group of nodes $\{q_1, q_2, \cdots, q_r\} \subseteq \{1, 2, \cdots, n_1\}$ which receive higher values compared with the others. The reason is simply that nodes receiving higher probabilities should be more closely related with the prior points compared with others and the subgraph constructed by these closely related nodes is exactly the

978-1-4799-7294-4/14/$31.00 ©2014 IEEE

module structure we need to detect. In order to see the way of tracking these nodes which form module, we make use of Table III in Section III as an illustration: we first sort the entry values of $\mathbf{x}^*$ in decreasing order and it's easy to see the 7 nodes with highest values receive at least 0.0769 when the others receive at most 7.2×10^{-18}. Thus we may conclude that the first 7 nodes listed in Table III receive much higher values compared with others and they might be considered to construct a module structure. In the second step, we employ $\mathbf{y}^*$ to decide if the module structure is consistent among the multiple networks. Each entry of $\mathbf{y}^*$ represents the contribution score of each networks to the module structure. If the module structure observed in the first step is common across all the considered networks, the corresponding contribution scores should be more or less the same with each other. For instance, the contribution scores of 3 networks are exactly same with each other in Table IV which indicates that the module structure is highly consistent across the 3 networks. On the other hand, if the entry values in $\mathbf{y}^*$ are extremely unbalance, the module structure detected may be inconsistent across multiple networks. In Table VIII, the contribution scores of the first two networks are less than 5×10^{-3} while score of the 3rd one is larger than 0.99. This fact tells us that the module structure in the first 2 networks are not clear while the 3rd network takes the dominant place for module structure, such that we may conclude the module structure is not common across the 3 networks.

Moreover, we provide the computational complexity of the proposed algorithm listed in Algorithm 2.1. In the first stage of each iteration, we are required to compute $\mathbf{M_1}$ and $\mathbf{M_2}$ with complexity of $\mathrm{O}(n_1^2 n_2)$. Then, during the second stage, it takes $\mathrm{O}(n_1 n_2)$ to update $\mathbf{x}$ and $\mathbf{y}$ at each step. Suppose that for fixed $\mathbf{M_1}$ and $\mathbf{M_2}$, it requires $k = Iter_1$ steps for both $\mathbf{x}$ and $\mathbf{y}$ to converge and we further assume that when t reaches the number of $Iter_2$, the whole algorithm converges. The total computational cost of Algorithm 2.1 should be $\mathrm{O}(Iter_2 * (n_1^2 n_2 + Iter_1 * (n_1 n_2)))$.

III. Experimental Results

We mainly focus on the performance on identifying common module of the proposed algorithm in this section. Both synthetic data and real data are considered in order to show the efficiency as well as the effectiveness of the proposed tensor based method. Although there are methods proposed to address the common module identification problem [30], [31], they are developed under the assumption that the underlying modules (clusters) are the same across different networks (data sets). In our considered networks, this assumption does not hold. We only report the experimental results of our proposed tensor based Markov Chain algorithm.

A. Module identification on synthetic data

In this subsection, we illustrate the efficiency as well as the effectiveness of the proposed algorithm by conducting experiments on synthetic data. As we have mentioned before, the proposed tensor based algorithm is capable of handling any number of multiple networks simultaneously. In this case, We implement proposed method 5 different synthetic networks corresponding to a same set of 100 vertices. In each network, the subgraph formed by 10 vertices are more densely connected compared with the others. Thus, when considering the multiple networks together, this subgraph should be exactly the common module structure which proposed algorithm targets on. Practically, we set the subgraph constructed by the first 10 vertices to be complete graph for all the networks except the third one. In the third networks, we try to add noise to the networks by removing some of the edges in the subgraph of the first 10 vertices (see Fig. 1) such that we can test if the proposed algorithm is capable of addressing the noised networks successfully. For the rest vertices the networks, we randomly put edges to connect vertices and finally, the average degree of each vertex in networks achieves around 18.

TABLE I
THE 11 LARGEST VALUES IN $\mathbf{x}$ RESPECT TO THE VERTICES FOR SYNTHETIC NETWORKS

Vertex	Value of $\mathbf{x}$
1	0.5085
10	0.0374
4	0.0354
2	0.0318
8	0.0315
7	0.0306
5	0.0304
3	0.0295
6	0.0276
9	0.0274
73	0.0078

TABLE II
CONTRIBUTION SCORES IN $\mathbf{y}$ FOR SYNTHETIC DATA.

Networks	1-st	2-nd	3-rd	4-th	5-th
Value of $\mathbf{y}$	0.2192	0.2225	0.1046	0.2231	0.2306

In the experiment, we set $\alpha = 0.5$ and employ the first vertex as prior information. The running time of the algorithm is around 0.02 second. Fig. 2 gives the value of $\mathbf{x}$ generated by performing the proposed algorithm on the synthetic networks. Obviously, the values received by first 10 vertices in $\mathbf{x}$ are larger than the others. More specifically, in TABLE I we list the largest 11 entries of resulting $\mathbf{x}$. We may also check the contribution of each network by checking value of each entry in $\mathbf{y}$, see TABLE II. Recover that the third network is corrupted by noise. It is easy to find out that the contribution score of the third network comes out to be the smallest while the contribution of the others is almost on the same level. Base on the these observations, we may draw the conclusion that proposed algorithm not only identifies common module embedding in multiple networks correctly, but also addresses the noised networks successfully.

B. Module identification on three cancer gene co-expression networks

In this subsection, we report the experimental results conducted on three gene data sets downloaded from The Cancer

978-1-4799-7294-4/14/$31.00 ©2014 IEEE

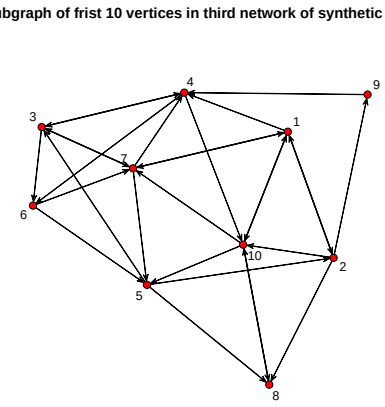

Fig. 1. The subgraph formed by first 10 vertices in the 3-rd network synthetic data.

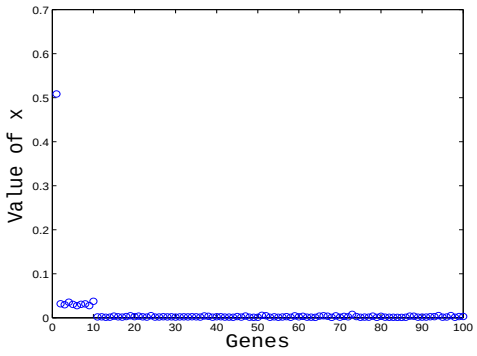

Fig. 2. Value of $\mathbf{x}$ with the 1-st vertex as prior for synthetic data.

Genome Atlas (TCGA). The three cancers are ovarian cancer (OV), glioblastoma multiforme (GBM), and lung squamous cell carcinoma (LUSC), respectively. All the data are generated with Affymetrix HT HG-U133A by Broad Institute. There are 558 OV samples, 594 GBM samples, and 134 LUSC samples in total 22277 different genes. For each cancer, we compute the variance of all the genes across the samples. In the experiments, we only select 1500 genes with largest variance for each cancer. Combining all 3 cancers, we actually select the number of 2756 different genes for further study.

The next step, for each of the three cancers, we calculate the Pearson correlation coefficients across all the genes and construct the affinity matrices by taking the hard thresholding. More precisely, if the Pearson correlation coefficient of two genes is greater than some pre-defined value, we assign an edge to link them up; otherwise, there should be no edge between them. Practically, we set thresholding 0.65, 0.60, 0.52 for OV, LUSC and GBM respectively such that all there networks have approximately scale free property. Moreover, the average degree of each gene across all 3 networks is about 18. Then, by removing the common genes in 3 cancer networks that have no connection to any other genes, we finally construct three networks for number of 2297 common genes.

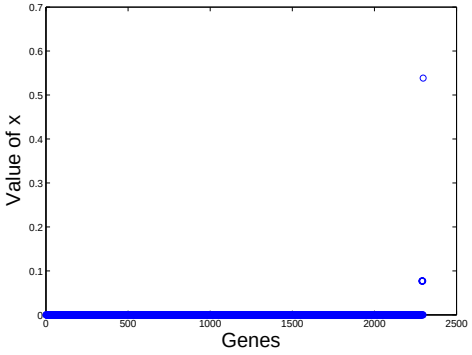

Fig. 3. Value of $\mathbf{x}$ with gene AFFX-r2-Ec-bioB-M at as prior information for example 1 in the three gene co-expression networks of cancers.

We apply our proposed tensor-based Markov chain method to the networks to identify the module structures. With d-ifferent nodes being selected as prior information, we are able to generate a probability distribution vector in which the nodes similar to them receive clearly much higher value than the others. Note that the parameter α is tuned based on seeking for an obvious jump between them. Moreover, another probability distribution vector $\mathbf{y}$ is also generated to measure the contribution of different networks. Based on $\mathbf{y}$, we may also check the consistency of the modules across multiple networks. When the difference in contribution scores of multiple networks are relatively small, we may conclude that module structure is consistent. Otherwise, the module structure may only appear in some of the networks. Thus, we may identify the consistent modules based on both converged $\mathbf{x}$ and $\mathbf{y}$. In order to illustrate the effectiveness of the proposed method more clearly, we state and discuss several examples in the following part.

1) Example 1: We select the control probe AFFX-r2-Ec-bioB-M as prior information $\mathbf{p}$ in this example. When we set α to be 0.5, our algorithm converges within about 19.8 seconds and the resulting $\mathbf{x}$ can be seen in the Fig. 3. The gene receiving largest value is AFFX-r2-Ec-bioB-M itself. In addition, it is clear to find out in the figure that some genes receive larger values compared with the others. To make it easier to check, we sort $\mathbf{x}$ in descending order and list the largest 8 values in TABLE III with the gene IDs accordingly. Obviously, a jump can be found between the value of gene AFFX-BioDn-5-at (0.0769) and gene 201348-at (7.2426×10^{-18}) and the first 7 genes receives much higher value than the others.

Moreover, we also concern about the converged $\mathbf{y}$ which is the contribution scores of multiple cancers as listed in TABLE IV. We may see for this example, the contribution of each cancer is on the same level. We get back to check the networks and find that the subgraphs constructed by the 7 identified

978-1-4799-7294-4/14/$31.00 ©2014 IEEE

TABLE III
THE 8 LARGEST VALUES IN $\mathbf{x}$ WITH THEIR GENE IDS FOR EXAMPLE 1 IN THE THREE GENE CO-EXPRESSION NETWORKS OF CANCERS.

Gene IDS	Value of $\mathbf{x}$
AFFX-r2-Ec-bioB-M-at	0.5386
AFFX-BioB-5-at	0.0769
AFFX-BioB-M-at	0.0769
AFFX-BioC-3-at	0.0769
AFFX-BioC-5-at	0.0769
AFFX-BioDn-5-at	0.0769
AFFX-r2-Ec-bioB-M-at	0.0769
201348-at	7.2426×10^{-18}

TABLE IV
CONTRIBUTION SCORES IN $\mathbf{y}$ OF THREE CANCERS FOR EXAMPLE 1.

Cancers	CBM	LUSC	OV
Value of $\mathbf{y}$	0.3333	0.3333	0.3333

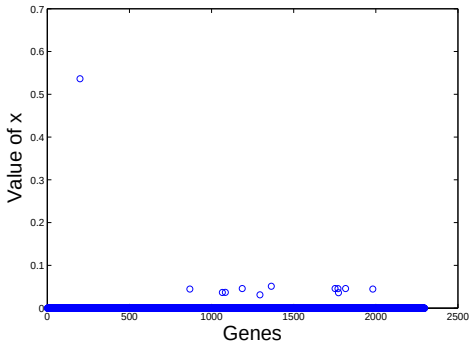

Fig. 4. Value of $\mathbf{x}$ with gene 202708-s-at as prior information for example 2 in the three gene co-expression networks of cancers.

genes are all complete graphs (thus exactly the same with each other). Hence it is reasonable that the contribution scores of the 3 networks are the same. In addition, the complete subgraphs extracted also explain the reason why all the other nodes identified receive the same score in $\mathbf{x}$.

Based on the results in TABLE III and TABLE IV, we may conclude that AFFX-r2-Ec-bioB-M-at, AFFX-BioB-5-at, AFFX-BioB-M-at, AFFX-BioC-3-at, AFFX-BioC-5-at, AFFX-BioDn-5-at and AFFX-r2-Ec-bioB-M-at should form a consistent module across networks of three cancers. Biologically, these nodes selected in this module are all known to be control probes which should be closely related with each other. This evidence supports that the module identified is meaningful, which validates the effectiveness of the proposed algorithm.

2) Example 2: In this example, we employ the gene 202708-s-at as prior information. α here is again set to be 0.5 and the computational time is around 20.0 seconds. In Fig. 4, we plot the converged $\mathbf{x}$ to make it clear to see.

Obviously, several genes which should be closely related with gene 202708-s-at receive much higher values compared with the others. Similarly, we sort converged $\mathbf{x}$ in descending order and list the largest 15 values with their gene IDs

Gene IDS	Value of $\mathbf{x}$
202708-s-at	0.5364
210387-at-at	0.0511
209398-at	0.0457
214290-s-at	0.0457
215071-s-at	0.0457
214455-at	0.0456
218280-x-at	0.0446
206110-at	0.0446
208579-x-at	0.0368
208180-s-at	0.0367
214469-at	0.0360
209911-x-at	0.0312
206640-x-at	1.3683×10^{-17}
207086-x-at	1.3683×10^{-17}
207663-x-at	1.3683×10^{-17}

TABLE V
THE 15 LARGEST VALUES IN $\mathbf{x}$ WITH THEIR GENE IDS FOR EXAMPLE 2 IN THE THREE GENE CO-EXPRESSION NETWORKS OF CANCERS.

TABLE VI
CONTRIBUTION SCORES IN $\mathbf{y}$ OF THREE CANCERS FOR EXAMPLE 2.

Cancers	CBM	LUSC	OV
Value of $\mathbf{y}$	0.6008	0.3672	0.0320

respectively in TABLE V. We may see a gap appears between value 0.0312 of gene 209911-x-at and 1.3683×10^{-17} of gene 206640-x-at.

Consider the converged results of $\mathbf{y}$ in TABLE VI. We may see different from the previous example, the contribution scores of three cancers are different.

In order to figure out the reason, we check the subgraphs formulated by identified nodes in all the three networks see Fig. 5.

We may see that vertices are closely related with each other in GBM and LUSC cancer networks while the subgraph in OV cancer does not contrain as many edges as which in GBM and LUSC networks. It turns out that the proposed algorithm assigns 0.6008 as contribution score to the GBM cancer network and 0.3672 to LUSC cancer network while OV cancer network only receives 0.0320. We do the enrichment analysis for Gene Ontology (GO, biological process) and KEGG pathways for this identified module. All the genes in this module belong to histone cluster 1 or histone cluster 2. 10 of the twelve genes in the module enrich 12 GO terms and they cover all the genes belonging to these GO terms among all the genes we consider. In TABLE VII, the related functions of these 12 GO terms can be found with P-value less than 10^{-11}. In addition, 8 in the 12 genes in this module enrich the pathway: hsa05322: Systemic lupus erythematosus. According to [28], [29], this pathway is related to several cancers such as liver cancers, lung cancers and kidney cancers. It is interesting that genes in this module identified in considered cancers networks also enrich this pathway. Based on the previous discussions, we may draw the conclusion that the module discovered by the proposed algorithm is meaningful in biological study. Therefore, we may also claim that the edges are preserved well in GBM and LUSC cancer networks while corrupted in the OV cancer network.

978-1-4799-7294-4/14/$31.00 ©2014 IEEE

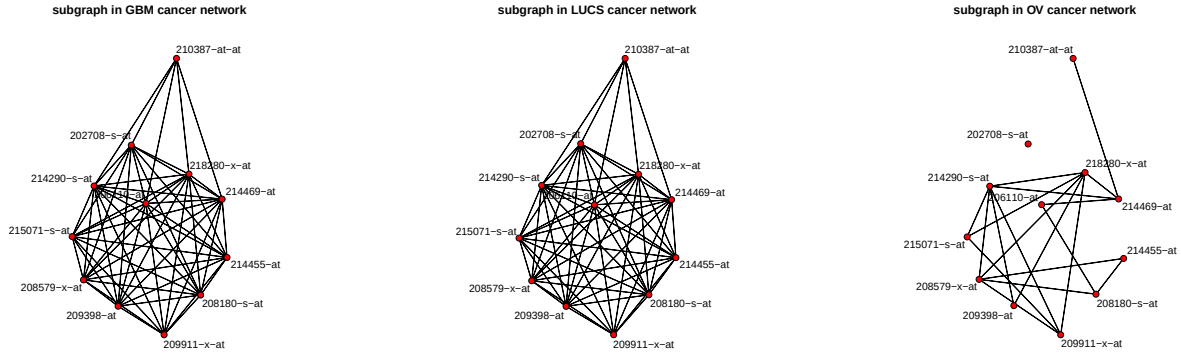

Fig. 5. Subgraphs constructed by 12 genes selected for example 2 in the three gene co-expression networks of cancers, left: subgraph in GBM cancer netowrk; middle: subgraph in LUCS cancer network; right: subgraph in OV cancer network.

Enriched GO terms	%	P-value
GO:006334 nucleosome assembly	100	8.83×10^{-21}
GO:0031497 chromatin assembly	100	1.21×10^{-20}
GO:0065004 protein-DNA complex assembly	100	1.88×10^{-20}
GO:0034728 nucleosome organization	100	2.31×10^{-20}
GO:0006323 DNA packaging	100	1.98×10^{-19}
GO:0006333 chromatin assembly or disassembly	100	4.25×10^{-19}
GO:0034622 cellular macromolecular complex assembly	100	1.96×10^{-15}
GO:0034621 cellular macromolecular complex subunit organization	100	5.62×10^{-15}
GO:0006325 chromatin organization	100	9.46×10^{-15}
GO:0051276 chromosome organization	100	9.11×10^{-14}
GO:0065003 macromolecular complex assembly	100	1.59×10^{-12}
GO:0043933 macromolecular complex subunit organization	100	2.88×10^{-12}

TABLE VII

GENE ONTOLOGY ENRICHMENT OF THE MODULE IDENTIFIED FOR EXAMPLE 2 IN THE THREE GENE CO-EXPRESSION NETWORKS OF CANCERS.

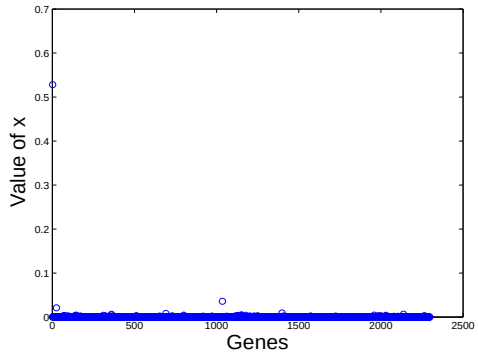

Fig. 6. Value of $\mathbf{x}$ with gene 200606-at as prior information for example 3 in the three gene co-expression networks of cancers.

3) Example 3: In this example, we consider the case that gene 200606-at is employed as prior. Similar to the previous examples, we set $\alpha = 0.5$ and run the algorithm on multiple networks of three cancers. The time cost of the algorithm is around 20.4 seconds and the converged $\mathbf{x}$ can be found in Fig. 6. We explore that there are in total 8 genes receiving much larger values than the others. When considering $\mathbf{y}$, which indicates the contribution of each network in TABLE VIII,

TABLE VIII

CONTRIBUTION SCORES IN $\mathbf{y}$ OF THREE CANCERS FOR EXAMPLE 3.

Cancers	CBM	LUSC	OV
Value of $\mathbf{y}$	0.0009	0.0030	0.9961

we may find the contribution scores of both GBM and LUSC cancer are very small. Then we check the networks of all three cancers and discover that the subgraphs in both GMB and LUSC contain no edge, see Fig. 7. Thus the module structure is not consistent in both GMB and LUSC cancers. Then edges appearing in subgraph of OV cancer networks are considered to be mis-linked due to the noise or parameter effect.

C. Module identification from co-expression network of different tissues of morbidly obese patients

In this part we present conducted experiments and results on co-expression network constructed by gene expression profile of liver (LIV), ometal (OME) and subcutaneous (SUBC) tissues for morbidly obese patients (GEO Accession number: GSE24294). There are in total 459 subjects with data across all three tissues and all the data are measured on the number of 40638 probes. We select the genes covered by more than one probe and employ the mean value as expression of that gene. Moreover, the genes with greater than 10% missing

978-1-4799-7294-4/14/$31.00 ©2014 IEEE

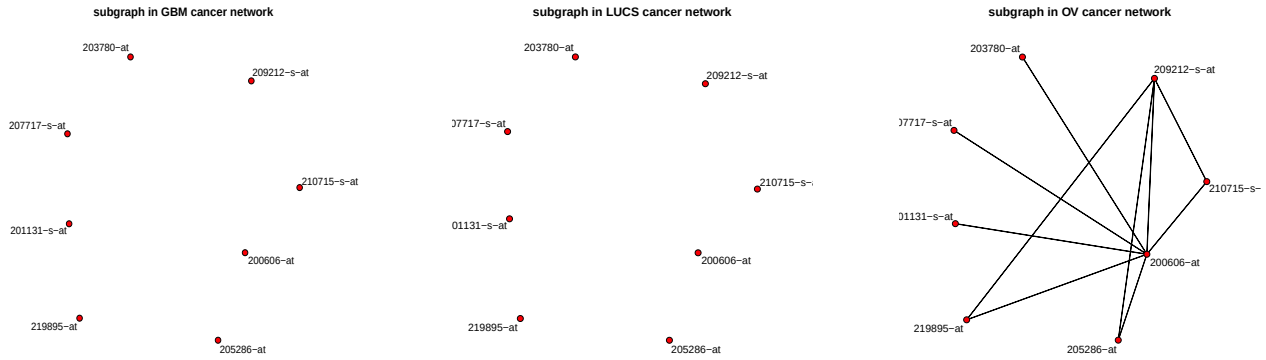

Fig. 7. Subgraphs constructed by 12 genes selected for example 3 in the three gene co-expression networks of cancers, left: subgraph in GBM cancer netowrk; middle: subgraph in LUCS cancer network; right: subgraph in OV cancer network.

observations are excluded, and we use mean of available data to express other missing values. Among the rest 17282 common genes, we select 1800 with largest variance across samples for each tissue and thus, we construct three gene networks which evolve the number of 2637 genes. Then we perform hard thresholding at 0.5 on multiple networks of all three tissues. Similar to the previous experiments, we remove the nodes which do not connect with any other nodes in the network and finally, each network consists of 1873 common genes.

By formulating the multiple networks into tensor data, we are able to employ the proposed method to handle the module identification problem. For the same purpose as previous experiments, we select a satisfied α which brings clear gap between values received by genes in $\mathbf{x}$. To validate the proposed method, we present some examples to clarify the effect and efficiency of the proposed tensor based method.

1) Example 1: In this example, we select gene SAA1 as prior information to run the proposed algorithm. The parameter α is set to be 0.5 and it takes around 13.7 seconds for our method to converge. The resulting $\mathbf{x}$ is plotted in Fig. 8. Obviously, some of the genes receive much higher values than the others.

It can be seen more clearly in TABLE IX that the first 4 genes listed in the table receive much larger scores (at least 0.1035 received by SAA2) than the others (at most only 0.0192 for CPR). Moreover, we also report evaluation on contribution of multiple networks in TABLE X.

According to the converged $\mathbf{y}$, the differences on contribution of multiple networks are not very significant which indicate that the module structure is quite consistent across three networks of different tissues. This module of 4 genes enriches 2 GO terms: GO:000695 acute-phase response and GO:0002526 acute inflammatory response with P-value 2.39×10^{-8} and 3.69×10^{-7} respectively. SAA is a well known protein in inflammation-associated reactive amyloidosis (AA-type). These facts indicate this module identified by the proposed method is reasonable biologically.

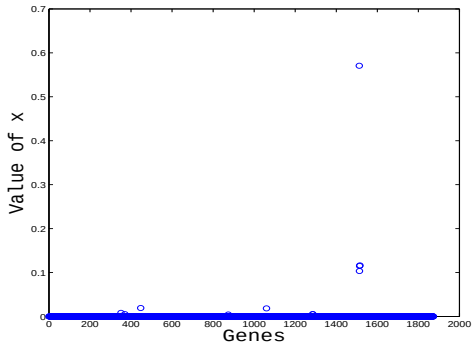

Fig. 8. Value of $\mathbf{x}$ with gene SAA1 as prior information for example 1 in the three gene co-expression networks of different tissues of morbidly obese patients.

TABLE IX
THE 5 LARGEST VALUES IN $\mathbf{x}$ WITH THEIR GENE IDS FOR EXAMPLE 1 IN THE THREE GENE CO-EXPRESSION NETWORKS OF DIFFERENT TISSUES OF MORBIDLY OBESE PATIENTS.

Gene IDS	Value of $\mathbf{x}$
SAA1	0.5706
SAA4P	0.1162
SAA3	0.1149
SAA2	0.1035
CPR	0.0192

TABLE X
CONTRIBUTION SCORES OF THREE TISSUES IN $\mathbf{y}$ FOR EXAMPLE 1

Cancers	LIV	OME	SUBC
Value of $\mathbf{y}$	0.4014	0.3091	0.2895

2) Example 2: In this example, we would like to use multiple nodes as prior information to test the effect of the proposed algorithm. More specifically, we select ALAS2 and HBA1 as prior information in $\mathbf{p}$, and perform the proposed tensor-based Markov chain algorithm with $\alpha = 0.5$. Computational time for the algorithm is 12.5 seconds and the converged $\mathbf{x}$ can be

978-1-4799-7294-4/14/$31.00 ©2014 IEEE

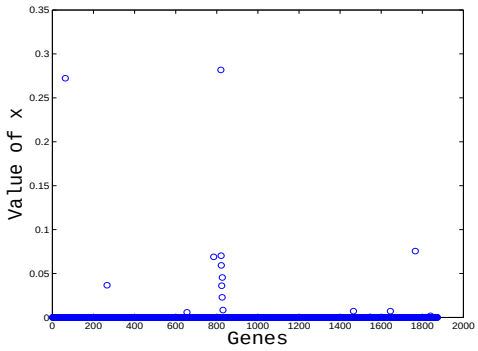

Fig. 9. Value of $\mathbf{x}$ with gene ALAS2 and HBA1 as prior information for example 2 in the three gene co-expression networks of different tissues of morbidly obese patients.

TABLE XI
THE 12 LARGEST VALUES IN $\mathbf{x}$ WITH THEIR GENE IDS FOR EXAMPLE 2 IN THE THREE GENE CO-EXPRESSION NETWORKS OF DIFFERENT TISSUES OF MORBIDLY OBESE PATIENTS.

Gene IDS	Value of $\mathbf{x}$
HBA1	0.2817
ALAS2	0.2723
TRIM58	0.0755
HBA2	0.0703
GPR144	0.0690
HBB	0.0592
HBG2	0.0454
CA1	0.0366
BHD	0.0361
HBG1	0.0228
HEMGN	7.084×10^{-3}
RHAG	7.084×10^{-3}

TABLE XII
CONTRIBUTION SCORES OF THREE TISSUES IN $\mathbf{y}$ FOR EXAMPLE 2.

Cancers	LIV	OME	SUBC
Value of $\mathbf{y}$	0.4120	0.3622	0.2258

found in Fig. 9.

Moreover, it is listed in TABLE XI that first 10 genes with largest value in $\mathbf{x}$ receive at least 0.0228 when the value of others is at most 7.084×10^{-3}. We check the contribution scores of networks in TABLE XII. The result indicates that the contribution scores received by LIV and OME networks are a little higher than SUBC. Still, it could tell that the module constructed by the chosen nodes is quite consistent across multiple networks.

This module enriches the GO:0015671 oxygen transport with P-value 3.58×10^{-11} and GO:0015669 gas transport with P-value 1.53×10^{-10} in the GO enrichment analysis. Among the considered genes in three tissues, half number of genes which carry out the function of oxygen transport and gas transport are located in this module.

In conclusion, our proposed method is capable of identifying the module structure both efficiently and effectively. Moreover, it also allows the users to check the contributions of different networks in $\mathbf{y}$. In the above experiments, both data sets contain three different networks while under our formulation, we can handle any number of multiple networks. Remark that the noise level in different networks can be also evaluated by examining the values in $\mathbf{y}$. If contribution scores of only a few networks are apparently distinct with the others, we may infer them as being noised severely or taking improper thresholding values. These results can be more convincible when more networks are involved in the tensor data.

IV. CONCLUSION

In this paper, we propose a novel tensor-based method for identification of module structures. The main idea of the proposed algorithm is to construct two Markov chains and iteratively update both label indicator $\mathbf{x}$ and contribution indicator $\mathbf{y}$ until global steady state is reached. Then by considering converged $\mathbf{x}$ and $\mathbf{y}$ we may tell the consistent module embedded in multiple networks. In order to illustrate the efficiency and effectiveness of proposed algorithm, we conduct experiments on two gene data sets. The results support that proposed algorithm is capable of identifying modules in multiple networks.

ACKNOWLEDGMENT

S. Zhang's research is supported in part by NSFC grants 10901042, 91130032,11471082 and Shanghai Natural Science Foundation 13ZR1403600. M. Ng's research is supported in part by Hong Kong Research Grant Council GRF Grant No. 201812.

REFERENCES

[1] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya, "Development and implementation of an algorithm for detection of protein complexes in large interaction networks", *BMC Bioinformatics*, 7:207, (2006).
[2] C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Guenoche, and B. Jacq, "Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network", *Genome Biol.*, 5, p. R6, (2003).
[3] J. Chen and B. Yuan, "Detecting functional modules in the yeast protein-protein interaction network", *Bioinformatics*, 22, p. 2283C2290, (2006).
[4] M. Chikina, C. Huttenhower, C. Murphy, and O. Troyanskaya, "Global prediction of tissue-specific gene expression and context-dependent gene networks in caenorhabditis elegans", *PLoS Comput. Biol.*, 5:e1000417, (2009).
[5] H. Chua, W. Sung, and L. Wong, "Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions", *Bioinformatics*, 22, pp. 1623-1630, (2006).
[6] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns", *Proc. Natl Acad. Sci. USA.*, 95, p. 14863C14868, (1998).
[7] S. Li, C. M. Armstrong, N. Bertin, H. Ge, S. Milstein, and et al., "A map of the interactome network of the metazoan c. elegans", *Science*, 303, pp. 540-543, (2004).
[8] E. Segal, N. Friedman, D. Koller, and A. Regev, "A module map showing conditional activity of expression modules in cancer", *Nat. Genet.*, 36, p. 1090C1098, (2004).
[9] R. Sharan, I. Ulitsky, and R. Shamir, "Network-based prediction of protein function", *Mol. Syst. Biol.*, 3:88, (2007).
[10] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani, "Global protein function prediction from protein-protein interaction networks", *Nat. Biotechnol.*, 21, pp. 697-700, (2003).
[11] L. Danon et al., "Comparing community structure identification", *Journal of Statistical Mechanics: Theory and Experiment*, 2005, p. P09008, (2005).

978-1-4799-7294-4/14/$31.00 ©2014 IEEE

[12] J. Dong and S. Horvath, “Understanding network concepts in modules”, *BMC Systems Biology*, 1, (2007).
[13] E. Estrada and N. Hatano, “Communicability in complex networks”, *Physical Review E.*, 77, p. 036111, (2008).
[14] S. Fortunato, “Community detection in graphs”, *Physics Reports*, 486, pp. 75-174, (2010).
[15] M.E.J. Newman, “Modularity and community structure in networks”, *Proc. Natl. Acad. Sci. USA*, 103, pp. 8577-8582, (2006).
[16] ———, “Finding community structure in networks using the eigenvectors of matrices”, *Physical Review E*, 74, p. 036104, (2006).
[17] M.A. Porter et al., “Communities in networks”, *Notices of the AMS*, 56, pp. 1082-1102, (2010).
[18] F. Radicchi et al., “Defining and identifying communities in networks”, *Proc. Natl. Acad. Sci. USA*, 101, pp. 2658-2663, (2004).
[19] S. Zhang and H. Zhao, “Community identification in networks with unbalanced structure”, *Physical Review E*, 85, p. 066114, (2012).
[20] ———, “Normalized modularity optimization method for community identification with degree adjustment”, *Physical Review E*, 88, p. 052802, 2013.
[21] M. Koyuturk, A. Grama, and W. Szpankowski, “An efficient algorithm for detecting frequent subgraphs in biological networks”, *Bioinformatics*, 20, p. i200Ci207, (2004).
[22] H. Hu, X. Yan, Y. Huang, J. Han, and X.J. Zhou, “Mining coherent dense subgraphs across massive biological networks for functional discovery”, *Bioinformatics*, 21, p. i213Ci221, (2005).
[23] Y. Huang, H. Li, H. Hu, X. Yan, M.S. Waterman, and et al., “Systematic discovery of functional modules and context-specific functional annotation of human genome”, *Bioinformatics*, 23, p. i222Ci229, (2007).
[24] W. Li, C.-C. Liu, T. Zhang, H. Li, M.S. Waterman, and et al., “Integrative analysis of many weighted co-expression networks using tensor computation”, p. e1001106, (2012).
[25] P.J. Mucha, T. Richardson, K. Macon, and M.A.P. and J.-P. Onnela, “Community structure in time-dependent, multiscale, and multiplex networks”, *Science*, 328, pp. 876-878, (2010).
[26] M. Girvan and M.E.J. Newman, “Community structure in social and biological networks”, *Proc. Natl. Acad. Sci. USA*, 99, pp. 7821- 7826, (2002).
[27] Q. Wu, MK. Ng and Y. Ye, “Markov-miml: A markov chain-based multi-instance multi-label learning algorithm”, *Knowledge and Information System*, 37, pp. 83–104, (2013).
[28] S. Bernatsky, R. Ramsey-Goldman, and A. Clarke, “Exploring the links between systemic lupus erythematosus and cancer”, *Rheumatic disease clinics of north America*, 31(2), pp. 387–402, (2005).
[29] A. Parikh-Patel, R. H. White, M. Allen, and R. Cress, “Cancer risk in a cohort of patients with systemic lupus erythematosus (sle) in california”, *Cancer Causes Control*, 19(8), pp. 887–894, (2008).
[30] W. Tang, Z. Lu and I. S. Dhillon”, “Clustering with Multiple Graphs”, *IEEE International Conference on Data Mining (ICDM)*, pp. 1016–1021”,(2009).
[31] S. Yu, X. Liu, L-C. Tranchevent, W. Glanzel, J. A. K. Suykens, B. D. Moor and Y. Moreau”, “Optimized data fusion for K-means Laplacian clustering”,*Bioinformatics*, 27(1), pp. 118–126, (2010).

978-1-4799-7294-4/14/$31.00 ©2014 IEEE