

# Active modules for multilayer weighted gene co-expression networks: a continuous optimization approach

DONG LI, SHAN HE

School of Computer Science, The University of Birmingham, UK

ZHISONG PAN, GUYU HU

PLA University of Science and Technology, China

June 3, 2016

## Abstract

**Motivation:** Searching for active connected subgraphs in biological networks has shown important to identifying functional modules. Most existing active modules identification methods need both network structural information and gene activity measures, typically requiring prior knowledge database and high-throughput data. As a pure data-driven gene network, weighted gene co-expression network (WGCN) could be constructed only from expression profile. Searching for modules on WGCN thus has potential values. While traditional clustering based modules detection on WGCN method covers all genes, unavoidable introducing many uninformative ones when annotating modules. We need to find more accurate part of them.

**Results:** We propose a fine-grained method to identify active modules on the multi-layer weighted (co-expression gene) network, based on a continuous optimization approach (AMOUNTAIN). The multilayer network are also considered under the unified framework, as a natural extension to single layer network case. The effectiveness is validated on both synthetic data and real-world data. And the software is provided as a user-friendly R package.

**Availability:** Available at <https://github.com/fairmiracle/AMOUNTAIN>

**Contact:** s.he@cs.bham.ac.uk

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

# 1 Introduction

As a well-known fact, a group of genes may get involved into a biological process other than act alone [3], thus identifying a group of genes and associating it with certain biological functions is of important. In this paper, we define a **functional module** in a biological network as a subnetwork which may involve a common function in biological processes. Another important but different concept is **topological module**, may also be referred as community, within which the interactions are much more intensive compared with those outside [15]. Topological modules have been studied intensively and the modular structure is easy to be detected in general sense, but functional modules are of real interest [3].

Although a function module may overlap with a topological module, only using the network structural information is not enough to find the function modules. The topology of a biological network does not always precisely reflects the function or even disease-determined regions [3], which are the real concerns in biology. To bridge the gap between the topological module and functional module, searching for **active modules**, i.e. connected regions of the molecular interaction network showing striking changes in molecular activity or phenotypic signatures that are associated with a given cellular response [28], has become a central challenge in system biology. Active modules were shown to be able to reveal regulatory mechanisms [19], which closely to function modules, i.e., these modules might connect multiple function modules. The activities of network nodes are usually measured by high-throughput omics data. In recent years, many active module identification algorithms have been developed to solve this problem, and most of them are applied on a *skeleton networks* plus *muscle* paradigm. The skeleton networks like protein-protein interaction networks or metabolic networks are constructed from prior knowledge database [19, 27, 10]. However, compared with increasing vast amount of high-throughput omics data, the speed of constructing reliable and complete skeleton networks, which heavily rely on experiment and human validation, is quite slow. For some non-model species, or even some new model species such as *Daphnia*, the PPI network even does not exist. The lack of reliable skeleton networks posts a challenge to the detection of active module to reveal certain key mechanisms in the biological systems.

In contrast, gene co-expression network is a pure data-driven gene network, which could be constructed only from expression profile. In such networks a nodes is a single gene and an edge is the correlation relationship between a pair of genes. And a weighted gene co-expression network is a fully connected graph. Modules in such networks are also considered to participate into some biological process [44], and those modules with significant biological meaning

are essentially functional modules.

As the first but crucial step for modules functional annotation analysis, module identification on gene networks is an important but less studied topic. Traditional module detection on gene co-expression networks was based on gene clustering, i.e. putting similar genes based on their correlations or edge weights into clusters as modules [44]. The coarse-grained clustering technique basically covers all genes in the network. As a results, the identified functional module including those genes show very little activities, which might not be very informative to reveal the biological mechanisms. We hypothesize that by identifying active modules which consider gene activities in the co-expression network, more precise biological mechanisms would be obtained. How to rigorously define **active modules** in such weighted network is still an open problem, but the module itself should be be more compact and informative compared with random subnetwork or clusters from two perspectives: 1) From the topological view, active modules are supposed enjoy high module scores measured by nodes and edges. 2) From the functional view, active modules are supposed to be more significantly associated with some biological process. The active modules identification on gene co-expression network is also a new problem, especially for the multilayer gene networks. A better understanding of modules in such network is expected to establish a pipeline from pervasive gene expression data to reasonable biological interpretation on a systemic level.

As a generalization of the single network, there are various reasons to model the interactions in living organisms as multilayer networks. Different layers may represent different time points, multiple conditions or various species. Modules across multiple layers in this report may reveal some properties of weighted gene co-expression networks such as time-invariant component genes, general responsive functional modules, and species conservation biological process. Similar but more general topics are also called multiplex networks [29, 22]. An existing work [24] mined recurrent heavy subgraphs in multi-slice networks, where each network shares the same set of genes without interactions between them. Conversed modules identification examples include [48, 13] which will be mentioned later. Inspired by [24], we develop a unified optimization framework to identify active modules on the multi-layer weighted co-expression gene network (AMOUNTAIN), and the layers could cover all three cases mentioned above.

## 2 Methods

In the early settings, active module identification was proposed to find significantly changed subnetwork in modular interaction networks [19]. Most following works developed methods based on the "skeleton+muscle" paradigm, where the "skeleton" is basic molecular interaction network constructed from prior knowledge database, and the "muscle" comes from the widely available high-throughput data which measure the genes activities. In general the skeleton biological network is represented as an undirected graph  $G = (V, E)$ , where nodes in  $V$  represents components like genes (or gene products), proteins or metabolites, and edges in  $E$  represents the interaction between two nodes. Each node  $i$  is assigned a single score to denote the activity of corresponding component in certain condition, such as fold-change or p-value of gene expression level. The simplified problem of finding highest score module in unweighted network, which consider the subnetwork score is the sum of each node's score, is formally defined as following:

**Problem 1.** *Given a graph  $G = (V, E)$  with vertices weights  $\mathbf{z} \in R^n$  for each  $v \in V$ , find a connected subnetwork  $S = (V_S, E_S)$  of  $G$  with maximal weight  $f(S) = \sum_{v \in V_S} z_v$ .*

This combinatorial optimization problem is also called Maximum-Weight Connected Subgraph Problem (MWCSP), which is equivalent to finding a maximum weight clique in a weighted graph, being referred as a famous NP-complete problem [20]. The proof is provided as supplementary materials of Ideker et al. [19]. As effective tools to solve combinatorial problems, metaheuristic algorithms have been widely applied to search satisfied solutions. The original paper [19] proposed to use simulated annealing, a generic probabilistic metaheuristic to solve this problem. Other methods include extended simulated annealing [17], greedy algorithm [40, 41], graph-based heuristic algorithm [34], genetic algorithm [27] and some exact approaches based on integer linear programming [33, 10, 46, 2].

### 2.1 Single-layer network

Compared with increasing vast amount of high-throughput data, the speed of constructing or confirming precise molecular interactions, which heavily rely on experiment and human validation, is quite slow. With only gene expression data, we could build gene co-expression network (GCN). We generalize the idea of active modules in problem (1) to GCN case, and consider the node score as gene importance criterion in gene co-expression network, which can be evaluated by the expression level changes (such as fold change or other

more comprehensive statistics) under certain conditions. Meanwhile the network structure is also determined by gene expression data and expressed as a weighted network. In practice, we get a bit more different problem to find a subgraph of size  $k$  (otherwise it corresponds to a trivial case containing all nodes) which aims to has both maximal node weights and closely connected to has large edge weights, formally defined as:

**Problem 2.** *Given a complete graph  $G = (V, E)$ , with vertex weight  $\mathbf{z}_v \in R$  for each  $v \in V$  and edge weights  $W = [w_{ij}]$  for each edge  $(i, j)$ , find a subgraph  $T$  of size  $k$  with large vertices weight  $\sum_{i \in T} z_i$  and also edges weights  $\sum_{i, j \in T} w_{ij}$ .*

Problem (2) is actually a simplified problem of  $(K_1, K_2)$ -Recurrent Heavy Subgraph (RHS) problem [24] but with additional node scores.  $(K_1, K_2)$ -RHS considers multiple co-expression networks, which is also discussed in details at next section. The module can be represented by membership vector  $\mathbf{x} \in \{0, 1\}^n$ , where  $x_i = 1$  means  $i$ -gene belongs to the module. Thus the optimization is naturally expressed as:

$$\begin{aligned} \max_{\mathbf{x}} S &= \mathbf{x}^T W \mathbf{x} + \lambda \mathbf{z}^T \mathbf{x} \\ \text{Subject to} & \\ \sum_{i=1}^n x_i &= k \\ x_i &\in \{0, 1\}, \quad i = 1, \dots, n. \end{aligned} \tag{1}$$

The NP-hardness can be proved by reducing the well-known NP-complete problem  $k$ -clique to this problem, for the details of the proof refer to the supplementary materials. Although integer programming methods [33, 10, 46] can be applied, it may cause high computational complexity and be lack of theoretical guarantee w.r.t running time and accuracy. Alternatively, if we relax the integer constraints of  $\mathbf{x}$  to continuous constraints [42, 24] and control the module size by introducing a vector norms of  $\mathbf{x}$ , it becomes a nonnegative and equality constrained quadratic programming (QP) problem (2), which can be solved by various existing continuous optimization techniques in polynomial time.

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}_+^n} F(\mathbf{x}) &= -\mathbf{x}^T W \mathbf{x} - \lambda \mathbf{z}^T \mathbf{x} \\ \text{Subject to,} & \\ f(\mathbf{x}) &= 1, \end{aligned} \tag{2}$$

where  $f(\mathbf{x})$  is the vector norm. The  $\ell_p$ -norm ( $p > 0$ ) of  $\mathbf{x}$  is defined as  $(\sum_i |x_i|^p)^{1/p}$ .

The choice of vector norm has an impact on how to solve the problem (2). For example, the  $\ell_0$ -norm can produce a sparse solution which is consistent with the fact that only a few members belong to the module. The  $\ell_1$ -norm is widely used as an alternative to  $\ell_0$  since the optimal solution of the latter corresponds to a combinatorial problem [11]. The  $\ell_2$ -norm is also widely used for closed-form solution but cannot lead to a sparse solution. A linear combination of  $\ell_1$  and  $\ell_2$ , i.e.  $(1 - \alpha)\|\mathbf{x}\|_1 + \alpha\|\mathbf{x}\|^2$  is also called elastic net penalty [49], when the objective is least square and  $\alpha = 0$  corresponds to lasso [38] and  $\alpha = 1$  corresponds to ridge regression. Elastic net is considered to enjoy the characteristics of both lasso and ridge regression. Besides, the  $\ell_\infty$ -norm is  $\max\{x_1, x_2, \dots, x_n\}$  which makes the values in vector smooth and all the entries are roughly identical. All the mentioned vector norms and the existing corresponding optimization techniques may be applied in the problem (2) as long as the constraints of vector norms can reveal some essence of module membership in weighted co-expression network. While the fundamental one is  $\ell_1$ -norm since the module size needs to be constrained, otherwise, the problem becomes trivial with all nodes included in the target module.

The  $\ell_1$ -norm constraint of optimization problem (2) can be converted to  $f(\mathbf{x}) = \sum_i^n x_i$  since  $x_i \geq 0$ . This problem has been intensively studied in mathematical programming [7, 4] and could be solved by a lot of existing method. But only using  $\ell_1$ -norm constraint tends to produce very sparse solution for problem (2), even a vector  $\mathbf{x}$  which contains only one non-zero element may be the optimal solution. [24] used the mixed norm  $\ell_{0,\infty}(\mathbf{x}) = \alpha\|\mathbf{x}\|_0 + (1 - \alpha)\|\mathbf{x}\|_\infty$  ( $0 < \alpha < 1$ ) to encode the characteristics of gene membership, where the optimal vector should contain equal non-zero values and the rest zero values. In practise  $\ell_{0,\infty}$  was approximated by  $\ell_{p,2}(\mathbf{x}) = \alpha\|\mathbf{x}\|_p + (1 - \alpha)\|\mathbf{x}\|_2$  ( $0 < p < 1$ ), and they solved the non-convex problem (1) with only edge weights by Multi-Stage Convex Relaxation (MSCR) [45].

It is a natural idea to use the elastic net penalty [49] to control the sparsity and achieve desirable membership, i.e.  $f(\mathbf{x}) = (1 - \alpha)\|\mathbf{x}\|_1 + \alpha\|\mathbf{x}\|^2$ . And a general strategy follows the gradient projection method [25] and generate a sequence to approximate the accurate solution like

$$\mathbf{x}^{(k+1)} = \Pi_C(\mathbf{x}^{(k)} - \alpha^{(k)}\nabla F(\mathbf{x}^{(k)})), \quad (3)$$

where  $\Pi_C$  is the Euclidean projection of a vector on convex set  $C$ , defined as (4):

$$\Pi_C(\mathbf{g}) = \arg \min_{\mathbf{x} \in \mathbb{R}_+^n} \frac{1}{2} \|\mathbf{x} - \mathbf{g}\|_2^2 \text{ s.t. } (1 - \alpha)\|\mathbf{x}\|_1 + \alpha\|\mathbf{x}\|^2 \leq t, \quad (4)$$

where  $\mathbf{g}$  is a constant vector and  $t$  is the radius which has little impact on the solution in practice. The step size  $\alpha^{(k)}$  in  $k$ -step should satisfy the following

condition (5) in order to make the objective function:

$$F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k)}) \leq \sigma \nabla_{\mathbf{x}} F(\mathbf{x}^{(k)})^T (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}), \quad (5)$$

where  $\sigma$  is a small positive constant. Searching for optimal  $\alpha^{(k)}$  is time consuming. Here we adopt the same as in [25] that scale  $\alpha^{(k)}$  by a fixed factor  $\beta$  until  $\alpha^{(k)}$  satisfies (5). Thus the algorithm is guaranteed to converge.

Solving subproblem (4) involves a root finding procedure [16] which can be done in linear time, and the total iterative procedure can be improved up by Nesterov's method [30], which replace the current step  $\mathbf{x}^{(k)}$  in (3) with a linear combination of previous two steps,  $\mathbf{s}^{(k)} = \mathbf{x}^{(k)} + t_k(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$  where  $t_k$  is another parameter to make it convergence. Nesterov's method has been shown to have optimal convergence rate for first-order method. Refer to the supplementary material the details about how to solve the convex optimization problem (2) with Elastic net penalty.

The Euclidean projections optimization for problem (2) is summarized as algorithm 1.

---

**Algorithm 1** Euclidean projections optimization

---

**Input:** Network edge weight  $W \in \mathbb{R}^{n \times n}$ , node score  $\mathbf{z} \in \mathbb{R}^n$  and initial solution  $\mathbf{x}^{(0)} \in \mathbb{R}_+^n$

**Output:** Module indicator vector  $\mathbf{x}$

- 1: **repeat**
  - 2:     Update  $\mathbf{g}$  in (4) by the gradient of  $F(\mathbf{x})$  in (2).
  - 3:     Solve optimal  $\mathbf{x}$  in (4) by algorithm in supplementary material.
  - 4: **until** Convergence or reach maximal iterations
- 

Generally we may want to identify multiple modules from one network. Similar to [47, 26], we can find  $N$  modules by running algorithm 1  $N$  times, with each time simply extracting the resulted module from background network. The resulting sequences of modules may indicates the importance in terms of node activities and correlation similarities, in a descending order. The general procedure for identifying  $N$  modules from given gene expression profile can be summarized as algorithm 2.

## 2.2 Two-layer network

The biological system has been modeled as a multi-layer network before [5], while more detailed analysis on multi-layer networks raises up in recent years [21], after the intensive research on single layer networks. The multi-layer network provides a general framework to model temporal and spatial change of

---

**Algorithm 2** Active modules identification of GCN

---

**Input:** Gene expression profile  $X \in \mathbb{R}^{n \times p}$ , number of modules  $M$

**Output:**  $M$  modules

- 1: **Construction:** Use certain algorithm to construct pair-wise gene network  $G$ .
  - 2: **Nodes scores:** Calculate the fold-change or other gene statistics according to experimental design.
  - 3: **for** iterations less than  $M$  **do**
  - 4:     **Solving:** Find solution  $\mathbf{x}$  for (1) using algorithm (1)
  - 5:     **Musk:** Delete the nodes in  $\mathbf{x}$  and corresponding edges from  $G$ .
  - 6: **end for**
- 

interactions for cellular networks, and contributing three aspects for current computational biology research: 1) Modeling dynamic properties for biological process as multiple snapshots, 2) Modeling different responses to multiple conditions of the same species and 3) The identification of conserved genes across species as well as specific genes.

Although the behaviors of living organisms were considered to be dynamic, traditional network-based methods primarily focused on static network, which is a snapshot of the real case. Multi-layer networks provide a powerful tool for modeling this time series networks, with each layer standing for a time point. The differential analysis on this time series networks may reveal several important concepts related to time changing on cells belongs to certain species or tissues, such as the key components that were not be affected or some vanishing structures.

The integration of genomic techniques into environmental toxicology has shown potential application values to develop exposure biomarkers and investigate the mode of toxicity [31]. Representing the interaction of a set of genes' response under different conditions as multiple layer network may shed light no this evolutionary conservation of living organism.

Besides the dynamic change of networks in the same set of components, a multi-layer network can also capture the core set across species, such as conserved active modules [37, 9]. By constructing a multi-layer network with each layer representing one species, we may find similar patterns in a species that has relatively more prior information, thus to gain biological knowledge of interested species. Finding conserved modules may also improve our understanding about the evolutionary biological procedure by highlighting the similarities and differences in key patterns between species [48].

Although [24] was proposed for multiple gene co-expression networks, which is similar to multi-layer networks. They can be distinguished from two aspects:



1) Multiple networks share exactly the same set of nodes while multi-layer networks do not necessarily to, which makes later can be applied in multiple species orthology. 2) Multi-layer networks consider inter-layer interactions while multiple networks do not to.

Another related work is **xHeinz** [13], which mines cross-species network modules using an integer linear programming approach. It extends the single network module identification algorithm **heinz** [10] to two species case and takes the same optimization technique for the problem. As discussed before, this kind of algorithm requires the skeleton interaction network plus high-throughput data, which is the main difference between our work. The network node scoring function in **xHeinz** is inherited from that in **heinz**, which requires the parameters estimation in a beta-uniform mixture (BUM) model. While our algorithm simply uses the fold-change information since node score is only part of the whole objective. In theory we can also use the adjusted log-likelihood ratio score in **heinz**. The optimization methodology we adopt is also straightforward, while **xHeinz** relies on external integer programming solver CPLEX.

Being similar to single layer network situation, an active module in a two layers-network can be represented as a connection of two modules in two different networks  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$ . The inter-layer interactions were measured by  $A = [a]_{ij} \in \mathbb{R}^{n_1 \times n_2}$  where  $n_1$  and  $n_2$  are the numbers of nodes in  $G_1$  and  $G_2$ . The basic two layer network module identification problem is formally defined as

**Problem 3.** *Given two complete graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$ , with vertices weights  $\mathbf{z}_{1v} \in \mathbb{R}$  for each  $v \in V_1$  and  $\mathbf{z}_{2v} \in \mathbb{R}$  for each  $v \in V_2$ . And edges weights  $W_1$  for edges in  $G_1$  and  $W_2$  for edges in  $G_2$ . The inter-layer interactions were measured by  $A = [a]_{ij} \in \mathbb{R}^{n_1 \times n_2}$ . The goal is to find two subgraphs  $T_1 \in G_1$  and  $T_2 \in G_2$  which both have large vertices weights and edges weights as well as intensive interaction with each other.*

We use two variables  $\mathbf{x}$  and  $\mathbf{y}$  to represent the memberships of active modules in two different networks,  $\mathbf{x}_i = 1$  means the  $i$ -th node in the first network is in the module. Thus the optimization problem can be expressed as an extension to (2),

$$\begin{aligned}
 & \min_{\mathbf{x} \in \mathbb{R}_+^{n_1}, \mathbf{y} \in \mathbb{R}_+^{n_2}} F = -\mathbf{x}^T W_1 \mathbf{x} - \lambda_1 \mathbf{z}_1^T \mathbf{x} - \mathbf{y}^T W_2 \mathbf{y} \\
 & - \lambda_2 \mathbf{z}_2^T \mathbf{y} - \lambda_3 \mathbf{x}^T A \mathbf{y} \\
 & \text{Subject to} \\
 & f_1(\mathbf{x}) = 1 \\
 & f_2(\mathbf{y}) = 1,
 \end{aligned} \tag{6}$$

where  $f_1(\mathbf{x})$  and  $f_2(\mathbf{y})$  are the vector norms on two vectors respectively. For simplicity we use the same Elastic net penalty  $f(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + (1 - \alpha)\|\mathbf{x}\|_2^2$  or mix norm penalty  $f(\mathbf{x}) = \alpha\|\mathbf{x}\|_p + (1 - \alpha)\|\mathbf{x}\|_2$  for both  $\mathbf{x}$  and  $\mathbf{y}$ . The general idea for solving (6) is alternating optimization, i.e. a iteratively optimize one variable while fixing another each time [25]. When optimizing one variable, it has the same form as in (2). Each iteration in the procedure can be simply expressed as:

- Find  $\mathbf{x}^{(k+1)}$  such that  $F(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k)}) \leq F(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$  and,
- Find  $\mathbf{y}^{(k+1)}$  such that  $F(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}) \leq F(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k)})$

The complete algorithm to find multiple modules in the two-layer network shares the same structure of algorithm 2. There is another parameter  $\lambda_3$  in (6) controlling how much degree the inter-layer links affect the resulting modules. Take multi-species for example, large  $\lambda_3$  can leads to conserved modules across different species which may reveal some gene conservation in response to certain changes. Conversely, small  $\lambda_3$ , e.g  $\lambda_3 = 0$  makes the inter-layer information playing no role, thus leading to two independent module identification processes. If we have multiple layers more than two, the rational keeps the same. As long as each layer has different set of nodes, alternating optimization can be used as the same way as in two-layer networks. Otherwise a more compact tensor computational paradigm [24] can be more efficient without inter-layer links consideration.

## 3 Results

### 3.1 Synthetic data

Several related works have used artificially generated data [34, 40, 23, 33] in order to test their algorithms in single network module identification. Being different from previous networks, the simulated networks here should have clear topological structure as well as node scores. We follow [24] to construct gene co-expression networks for simulation study. Let  $n$  be the number of genes, and edge weights as well as node score follow the uniform distribution in range  $[0, 1]$ . A module contains  $k$  genes inside which the edge weights as well as node score follow the uniform distribution in range  $[\theta, 1]$ , where  $\theta = \{0.5, 0.6, 0.7, 0.8, 0.9\}$ . Figure 1 shows the weighted co-expression network when  $n = 100, k = 20$  and red nodes indicate module members and wider edges mean larger similarities. Visualization is based on `qgraph` [14].

With ground-truth in hand, we can define the following performance measurement for the problem (2). From the topological view, the accuracy for

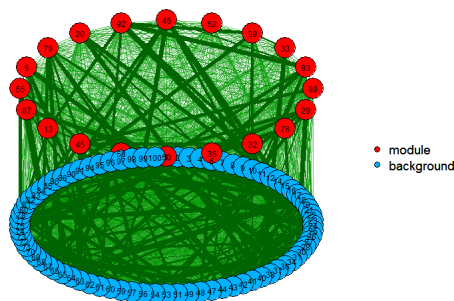


Figure 1: Simulated weighted gene co-expression network.

each single module identification is considered as a binary classification problem. The performance is defined by the  $2 \times 2$  confusion matrix, including the number of genes correctly detected (True positive,  $tp$ ), the number of genes in identified module but not in the real module (False positive,  $fp$ ), the number of genes in the real module but not in the identified module (False negative,  $fn$ ), and the number of genes neither in identified module or in the real module (True negative,  $tn$ ).

$$\begin{aligned}
 Recall &= \frac{tp}{tp + fn} \\
 Precision &= \frac{tp}{tp + fp} \\
 F &= 2 \cdot \frac{Precision \cdot Recall}{Recall + Precision}.
 \end{aligned} \tag{7}$$

We conduct the simulation study on a relatively large network when  $n = 10000$ , and we consider a sparse case where one module contains  $k = 100$  genes and node score follow the uniform distribution in range  $[\theta, 1]$ , where  $\theta = 0.5$ . The algorithm which uses mixed norm in (2) is outperformed by our algorithm 1 using elastic net penalty, in terms of both the running time and predictive accuracy. By choosing proper parameters from grid search combining  $\alpha = \{0.1 \sim 0.9\}$  in elastic net penalty and  $\lambda = 2^{\{-5 \sim 5\}}$  in (2), we can almost exactly find the target model nodes. The optimal  $\alpha = \{0.3 \sim 0.4\}$  and  $\lambda = 2^{\{-5 \sim -1\}}$  for this network which makes  $F = 1$ . Figure 2 shows how these parameters affect F-score (7) in this case. A similar result is also found in two-layer network simulation.

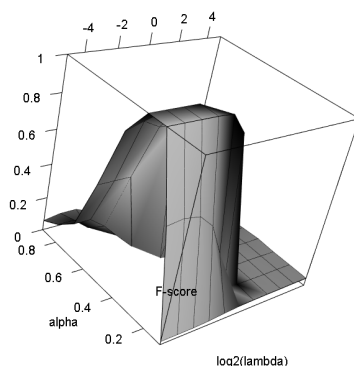


Figure 2: Parameters selection for algorithm 1 on large network.

### 3.2 Real-world data for single layer network

We mainly use gene expression datasets from Gene Expression Omnibus (GEO) [12] as real-world data examples. GSE3635 and GSEGSE5283 are two expression profiling by array from *Saccharomyces cerevisiae*. The original goal of the dataset is study the the regulation of transcription factor YOX1 and YHP1 during the cell cycle of *Saccharomyces cerevisiae* [32], with deleted YOX1 and YHP1 deleted. The wild-type and mutant cells were collected at 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, and 120 minutes after synchronization with alpha factor. We take the formal as control group and second as experimental group since they share the same set of genes. And we assume the biological processes with YOX1 and YHP1 deleted was measured by gene expression values. The goal is to find several set of genes, referred as modules identification using algorithm 1, which are closely related to these biological processes. The gene expression values has been normalized as log ratio using Rosetta Resolver, we only need to deal with missing or invalid values. The strategy is simple, discarding probes with more than 20% missing values or NAs and repalcing missing or NAs positions in a valid probe with mean value of the rest samples of that probe. We donot pick up signigicantly expressed genes using linear model like many other methods, since the algorithm requires as complete information about gene correlation and gene activities only contribute part of the whole objective.

First, algorithm 1 requires the a weighted gene coexpression network as input. We employ the same differential analysis method of gene pairs as [18]. The difference of coexpression, also the edge score of gene  $i$  and  $j$  between two

conditions  $a$  (control) and  $b$  (experimental) was quantified by,

$$W_{ij}^{a \rightarrow b} = (R_{ij}^b - R_{ij}^a) \times \max(|R_{ij}^b|, |R_{ij}^a|), \quad (8)$$

where  $R_{ij}^b$  is the correlation between gene  $i$  and  $j$  under condition  $b$ . The node score which reflects the gene expression activity degree is measured by the ratio of expression level under two conditions,

$$z_i^{a \rightarrow b} = \frac{\frac{1}{m} \sum_{j=1}^m D_{ij}^b}{\frac{1}{n} \sum_{j=1}^n D_{ij}^a}, \quad (9)$$

where  $D_{ij}^b$  is the gene expression value of gene  $i$  in  $j$ -th sample under condition  $b$ , and  $m, n$  are the number of sample in condition  $b$  and  $a$  respectively. Here we let  $N = 10$  in algorithm. The parameter  $\lambda$  represents the trade-off between edges weight and nodes weight which seems to play a slight role in performance. Here we simply fix  $\lambda = 1$  and use a binary search method to select  $\alpha$  for elastic net penalty which controls the sparsity of the module. See supplementary text section 4 for usage of parameter selection. Here we desire each module size with around 100 200 genes. These modules are provided as gene list (S1.xlsx).

We performed functional enrichment analysis of these modules. The basic of functional enrichment of a module is to assign the biological process annotations in Gene Ontology [1] to the genes (proteins) in that module. The probability that a module of size  $n$  have the same function as an existing functional module can be calculated by a hypergeometric distribution with Gene Ontology Term Finder<sup>1</sup>. The P-value is calculated by the following formula,

$$P = \sum_{x=1}^n \frac{\binom{M}{x} \binom{N-M}{n-x}}{\binom{N}{n}} \quad (10)$$

where  $N$  is the total number of genes (proteins) within the genome and  $M$  is the total number of genes (proteins) within a category. A low p-value indicates genes have high overlap with enriched functional categories thus are biologically significant. Results show that module identification using algorithm 1 can find consistent biological processes only from expression data which help to understand the underlying mechanisms related to biological conditions. Since YOX1 and YHP1 are important transcription factors in the regulation of the cell cycle, the identified modules are enriched by corresponding biological processes such as single-organism cellular process (GO:0044763), cellular macromolecule metabolic process (GO:0006139) and nucleobase-containing compound metabolic process (GO:0044260). Furthermore, the enriched GO terms are less significant as they are later identified, which corresponds with the algorithmic settings.

---

<sup>1</sup><http://www.yeastgenome.org/help/analyze/go-term-finder#pvalue>

### 3.3 Real-world data for two-layer network

Inspired by xHeinz [13], we chose two expression data for two species *mus musculus* and *homo sapiens*: GSE43955 and GSE35103 for multilayer case. The original studies [43] and [39] reported the expression profiles identification controlled by the differentiation of Th17 cell. Here we expect the proposed algorithm could find consistent results from a two-layer gene co-expression network. Each layer of the network is constructed from the gene expression of a species, and the edge weights and node weights are defined by (8) and (9) as well. In each layer the two conditions are with or without Th17. Specifically, we use the expression value of two conditions in different time points to check whether the effect may vary along the time. The inter-layer connections are defined by the orthology information, obtained from Ensembl 84<sup>2</sup>. We use the associated gene name as the unique identifier for each gene (node) in both human and mouse, and the corresponding orthologous mapping table are embedded into this two-layer network. After gene expression data pre-processing and orthologous selection, we get 19332 genes in human layer and 13656 genes in mouse layer. There are 8066 links between two layers, standing for confident orthologous mapping pairs.

As the same in single layer case, we use the binary search for parameter  $\alpha$  in elastic net penalties to get the desired modules size for both layers. It turns out to be a grid search process in order to get two desired modules at the same time. Here we only consider the large  $\lambda_3$  in (2) which aims to find a conserved module across these two species. We use samples from all conditions to construct the basic weighted co-expression networks for two species, but different expression values under different time to define node activities. Because correlation based network construction requires as many samples while gene activities are closely related to certain conditions, including the exposed time period.

Figure 3 shows the most active module (the first identified) for human and mouse at the time point 2 hours, visualized by muxViz [8]. The conserved module is acquired when  $\lambda_3 = 1000$  in (2), where inter-layer links mean the orthologous gene pairs. The gene lists of two modules are attached in table file (S2.xlsx). Gene ontology enrichment analysis indicate that there are several GO terms such as response to virus (GO:0009615) and cellular response to type I interferon (GO:0071357) are significantly enriched for mouse, while GO terms like response to endoplasmic reticulum stress (GO:0034976) and topologically incorrect protein (GO:0035966) for human. Both modules shows some cellular response to topologically incorrect protein (GO:0035967) or virus (GO:0051607), which is closely related to the functional role of Th17 differen-

---

<sup>2</sup><http://www.ensembl.org/>

tiation played in pathogenesis of autoimmune and inflammatory diseases [39].

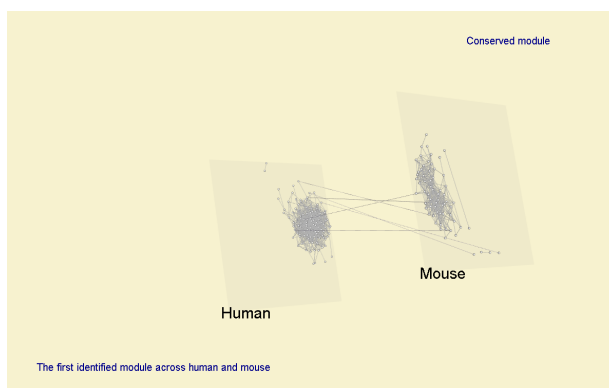


Figure 3: Conserved modules for human and mouse. Visualized by muxViz [8]

Besides the early stage of Th17, we also explore the characteristics of other time points (12h, 24h, 48h and 72h). Figure 5 shows the most active module (the first identified) for human across these four time points. We can see that some shared genes show activity along the time and play important roles in these networks. All gene lists are attached in table file (S3.xlsx). Gene ontology enrichment analysis shows that all modules are significantly related to several biological processes, which are consistent with previous studies [1].

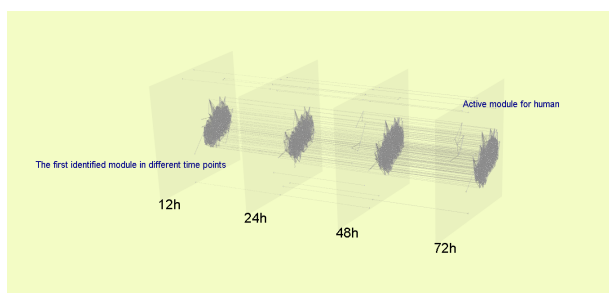


Figure 4: Active modules for human across time points. Visualized by muxViz [8]

The first active modules identified in each time point themselves show some differences, indicating that modification of Th17 would consistently have an impact on the cell along the time. From the algorithmic point of view, when fixing  $\lambda_3$  in (2) we need to seek the optimal parameters in the elastic penalties for both species. i.e.  $f_1(\mathbf{x}) = \alpha_1 \|\mathbf{x}\|_p + (1 - \alpha_1) \|\mathbf{x}\|_2$  and  $f_2(\mathbf{y}) = \alpha_2 \|\mathbf{y}\|_p + (1 -$

$\alpha_2\|\mathbf{y}\|_2$ . The grid search uses a binary search for each to seek a combination of two  $\alpha$ s for desired modules. Table 1 shows the results,

	12h	24h	48h	72h
$\alpha_1$	0.675	0.675	0.563	0.563
$\alpha_2$	0.042	0.028	0.018	0.028

Table 1: Best parameters for equation (2)

We use an integrative gene list enrichment analysis tool **Enrichr** for human [6], which does not only provide the pathway and gene ontology enrichment analysis, but also has a visualization tool with each. Figure 5 shows top GO items enriched by the most active modules different in different time points and their relationships, check supplementaty file part 3 for other time points (Figure S1-S3). Although these modules have a lot genes in common, functions enriched by the first identified module changed slightly along the time. Structural constituent of ribosome (GO:0003735) appeared frequently as a top enriched term in all time points, served as a fundamental function.

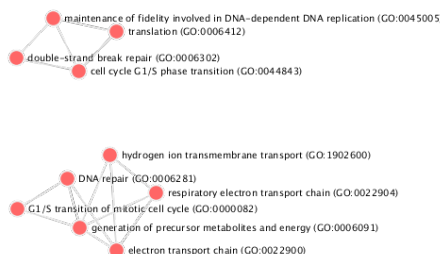


Figure 5: GO terms network of the identified module at 12h

## 4 Discussion and Conclusion

There have been many works discussing about key individual components such as transcription factors which play important roles in a biological process. Popular tools such as limma [35] uses statistical models to find significantly expressed genes, which almost becomes a standard pre-processing for further study. Take Th17 cell differentiation for example, previous works [36, 43, 39] reveals important transcription factors involved and related mechanism. Our model is designed not for individual gene detection tasks such transcription



factors identification or gene prioritization, but for providing a complementary from modular perspective. Modules identified from genome-wide network may reveal system-level properties of related biological mechanism. The goal of algorithm 1 is to establish a computational approach that only uses gene expression data from different conditions, to find biological modules which show significant responses caused by expression changes.

The continuous optimization method, especially the convex optimization also offers high efficient computational tools other than widely used heuristic algorithms [19, 48] or discrete optimization [10, 13] for active modules identification. Furthermore, convex optimization methods always enjoy the guarantee with respect to running time and accuracy. On the one hand, this guarantee makes the solution more reliable even unique given precise input. On the other side, the so-called optimal solution fully relies on algorithmic input which poses a higher demand on data preprocessing and model assumption. Take the real-world data studies in section 3.2 and 3.3 for example, slight differences on how to compute node scores or edge weights may lead totally different results. Conversely the uncertainty in heuristic algorithms may offer flexibility about model assumption and algorithmic input. In other words, the gap between computational model and real case does exist. From the software design and implementation view, open source and user-friendly tools have more advantages. A large number of reliable open source libraries can be easily found for mature convex optimization techniques. And it is not difficult to implement the core part of them. In contrast, the implementation of specific purpose heuristic algorithms or integer programming is challenging.

This paper describes a general continuous optimization based active modules identification method for multilayer gene coexpression networks. With proper replacement of node (gene) similarity matrix and node activities, the proposed methods can be easily extended to other applications. The idea of formulating the conserved modules identification across multiple layers under a uniform framework can also be enriched with more sophisticated considerations, such as multiple data source fusion instead of using single gene expression profiles. Future works may be more integrated with specific applications.

## Funding

This work has been supported by the...

## References

- [1] 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.
- [2] C. Backes, A. Rurainski, G. W. Klau, O. Müller, D. Stöckel, A. Gerasch, J. Küntzer, D. Maisel, N. Ludwig, M. Hein, et al. An integer linear programming approach for finding deregulated subgraphs in regulatory networks. *Nucleic acids research*, 40(6):e43–e43, 2012.
- [3] A.-L. Barabási, N. Gulbahce, and J. Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011.
- [4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [5] P. Brazhnik, A. de la Fuente, and P. Mendes. Gene networks: how to put the function in genomics. *TRENDS in Biotechnology*, 20(11):467–472, 2002.
- [6] E. Y. Chen, C. M. Tan, Y. Kou, Q. Duan, Z. Wang, G. V. Meirelles, N. R. Clark, and A. Maayan. Enrichr: interactive and collaborative html5 gene list enrichment analysis tool. *BMC bioinformatics*, 14(1):128, 2013.
- [7] T. F. Coleman and J. Liu. An interior newton method for quadratic programming. *Mathematical programming*, 85(3):491–523, 1999.
- [8] M. De Domenico, M. A. Porter, and A. Arenas. Muxviz: a tool for multilayer analysis and visualization of networks. *Journal of Complex Networks*, page cnu038, 2014.
- [9] R. Deshpande, S. Sharma, C. M. Verfaillie, W.-S. Hu, and C. L. Myers. A scalable approach for discovering conserved active subnetworks across species. *PLoS computational biology*, 6(12):e1001028, 2010.
- [10] M. T. Dittrich, G. W. Klau, A. Rosenwald, T. Dandekar, and T. Müller. Identifying functional modules in protein–protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–i231, 2008.
- [11] D. L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.
- [12] R. Edgar, M. Domrachev, and A. E. Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002.

- [13] M. El-Kebir, H. Soueidan, T. Hume, D. Beisser, M. Dittrich, T. Müller, G. Blin, J. Heringa, M. Nikolski, L. F. Wessels, et al. xheinz: An algorithm for mining cross-species network modules under a flexible conservation model. *Bioinformatics*, page btv316, 2015.
- [14] S. Epskamp, A. O. Cramer, L. J. Waldorp, V. D. Schmittmann, and D. Borsboom. qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software*, 48(4):1–18, 2012.
- [15] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.
- [16] P. Gong, K. Gai, and C. Zhang. Efficient euclidean projections via piecewise root finding and its application in gradient projection. *Neurocomputing*, 74(17):2754–2766, 2011.
- [17] Z. Guo, Y. Li, X. Gong, C. Yao, W. Ma, D. Wang, Y. Li, J. Zhu, M. Zhang, D. Yang, et al. Edge-based scoring and searching method for identifying condition-responsive protein–protein interaction sub-network. *Bioinformatics*, 23(16):2121–2128, 2007.
- [18] C.-L. Hsu, H.-F. Juan, and H.-C. Huang. Functional analysis and characterization of differential coexpression networks. *Scientific reports*, 5, 2015.
- [19] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18(suppl 1):S233–S240, 2002.
- [20] R. M. Karp. *Reducibility among combinatorial problems*. Springer, 1972.
- [21] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.
- [22] N. E. Kouvaris, S. Hata, and A. Díaz-Guilera. Pattern formation in multiplex networks. *Scientific reports*, 5, 2015.
- [23] P. Langfelder and S. Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559, 2008.
- [24] W. Li, C.-C. Liu, T. Zhang, H. Li, M. S. Waterman, and X. J. Zhou. Integrative analysis of many weighted co-expression networks using tensor computation. *PLoS Comput Biol*, 7(6):e1001106, 2011.
- [25] C.-b. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

- [26] Y. Liu, D. A. Tennant, Z. Zhu, J. K. Heath, X. Yao, and S. He. Dime: a scalable disease module identification algorithm with application to glioma progression. *PLoS one*, 9(2), 2014.
- [27] H. Ma, E. E. Schadt, L. M. Kaplan, and H. Zhao. Cosine: Condition-specific sub-network identification using a global optimization method. *Bioinformatics*, 27(9):1290–1298, 2011.
- [28] K. Mitra, A.-R. Carvunis, S. K. Ramesh, and T. Ideker. Integrative approaches for finding modular structure in biological networks. *Nature Reviews Genetics*, 14(10):719–732, 2013.
- [29] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878, 2010.
- [30] Y. Nesterov. A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ . In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.
- [31] H. C. Poynton, J. M. Lazorchak, C. A. Impellitteri, B. J. Blalock, K. Rogers, H. J. Allen, A. Loguinov, J. L. Heckman, and S. Govindasmaw. Toxicogenomic responses of nanotoxicity in daphnia magna exposed to silver nitrate and coated silver nanoparticles. *Environmental science & technology*, 46(11):6288–6296, 2012.
- [32] T. Pramila, S. Miles, D. GuhaThakurta, D. Jemiole, and L. L. Breiden. Conserved homeodomain proteins interact with mads box protein mcm1 to restrict ecb-dependent transcription to the m/g1 phase of the cell cycle. *Genes & development*, 16(23):3034–3045, 2002.
- [33] Y.-Q. Qiu, S. Zhang, X.-S. Zhang, and L. Chen. Detecting disease associated modules and prioritizing active genes based on high throughput data. *BMC bioinformatics*, 11(1):26, 2010.
- [34] D. Rajagopalan and P. Agarwal. Inferring pathways from gene lists using a literature-derived network of biological relationships. *Bioinformatics*, 21(6):788–793, 2005.
- [35] G. K. Smyth. Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*, pages 397–420. Springer, 2005.
- [36] B. Stockinger and M. Veldhoen. Differentiation and function of th17 t cells. *Current opinion in immunology*, 19(3):281–286, 2007.
- [37] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim. A gene-coexpression network for global discovery of conserved genetic modules. *science*, 302(5643):249–255, 2003.

- [38] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- [39] S. Tuomela, V. Salo, S. K. Tripathi, Z. Chen, K. Laurila, B. Gupta, T. Äijö, L. Oikari, B. Stockinger, H. Lähdesmäki, et al. Identification of early gene expression changes during human th17 cell differentiation. *Blood*, 119(23):e151–e160, 2012.
- [40] I. Ulitsky and R. Shamir. Identification of functional modules using network topology and high-throughput data. *BMC systems biology*, 1(1):8, 2007.
- [41] I. Ulitsky and R. Shamir. Identifying functional modules using expression profiles and confidence-scored protein interactions. *Bioinformatics*, 25(9):1158–1164, 2009.
- [42] Y. Wang and Y. Xia. Condition specific subnetwork identification using an optimization model. *Proc Optim Syst Biol*, 9:333–340, 2008.
- [43] N. Yosef, A. K. Shalek, J. T. Gaublot, H. Jin, Y. Lee, A. Awasthi, C. Wu, K. Karwacz, S. Xiao, M. Jorgolli, et al. Dynamic regulatory network controlling th17 cell differentiation. *Nature*, 496(7446):461–468, 2013.
- [44] B. Zhang and S. Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1), 2005.
- [45] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *The Journal of Machine Learning Research*, 11:1081–1107, 2010.
- [46] X.-M. Zhao, R.-S. Wang, L. Chen, and K. Aihara. Uncovering signal transduction networks from high-throughput data by integer linear programming. *Nucleic acids research*, 36(9):e48–e48, 2008.
- [47] Y. Zhao, E. Levina, and J. Zhu. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.
- [48] G. E. Zinman, S. Naiman, D. M. O’Dee, N. Kumar, G. J. Nau, H. Y. Cohen, and Z. Bar-Joseph. Moduleblast: identifying activated sub-networks within and across species. *Nucleic acids research*, 43(3):e20–e20, 2015.
- [49] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.