MODA: MOdule Differential Analysis for weighted gene co-expression network

Dong Li
School of Computer Science, The University of Birmingham, UK
James B. Brown
Department of Statistics, University of California Berkeley, USA
Luisa Orsini
School of Biosciences, The University of Birmingham, UK
Zhisong Pan,Guyu Hu
PLA University of Science and Technology, China
Shan He
School of Computer Science, The University of Birmingham, UK

May 31, 2016

1 Summary

Gene co-expression network differential analysis is designed to help biologists understand gene expression patterns under different conditions. We have implemented an R package called MODA (Module Differential Analysis) for gene co-expression network differential analysis. Based on transcriptomic data, MODA can be used to estimate and construct condition-specific gene co-expression networks, and identify differentially expressed subnetworks as conserved or condition specific modules which are potentially associated with relevant biological processes. The usefulness of the method is also demonstrated by synthetic data as well as Daphnia magna gene expression data under different environmental stresses.
Availability: Available at https://github.com/fairmiracle/MODA
Contact: s.he@cs.bham.ac.uk

2 Introduction

Gene co-expression network attracts much attention nowadays. In such a network, nodes represent genes and each edge connecting two genes stands for how much degree may this pair of genes are co-expressed across several samples. The presence of these edges is commonly based on the correlation coefficients between each gene pairs. The higher of correlation between a pair of genes, the higher probability that there exists a co-functionality relationship between them. Weighted correlation network analysis (WGCNA) [1, 2] has been widely used for this case, mostly as a tool for single network analysis. Traditional gene differential analysis has covered identification of important individual genes [3] which shows significant changes across multiple conditions. Even so-called network differential analysis still focus on isolated nodes (genes) in networks. However, based on the fact that genes interact with each other to exert some biological function instead of acting alone, it may be more informative to identify a subnetwork of genes which are conserved across multiple conditions or just active in certain conditions.

Several previous works went beyond individual gene differential analysis. GSCA [4] detects a set of differentially co-expressed (DC) genes. DICER [5] uses a probabilistic framework to detect DC gene sets. Both of take genes as individuals and did not provide a systematic view (at network level) of expression profiles. DINGO [6] estimates group-specific networks by calculating differential scores between each pair of two genes, which is focused on individual edges in the networks.

Here we present MODA (Module Differential Analysis), a Bioconductor (ref Huber) package for co-expression network differential analysis, which can (i) estimate and construct condition-specific co-expression networks with limited samples for each biological condition from gene expression profile; (ii) identify conserved and condition-specific co-expression modules by comparing networks; (iii) perform functional annotation enrichment analysis on the identified modules.

3 Methods

The first step is condition-specific network reconstruction. In a gene co-expression network, the edge weights are defined by correlation coefficients of gene pairs. However, it is well known that the accurate correlation coefficient is approximated by $1/sqrt(n)$ where n is the number of samples, which makes it impossible to get reliable correlation coefficients with only several replicates under each experimental condition in practice. We use a sample-saving approach to construct condition-specific co-expression networks for each single condition, which works as follows. Assume

network N_1 is background, normally containing samples from all conditions, is constructed based on the correlation matrix from all samples. Then condition D specific network N_2 is constructed from all samples minus samples belong to certain condition D [7]. The differences between network N_1 and N_1 is supposed to reveal the effects of condition D. The rationale behind this criteria is based on the mechanism of correlation, i.e. which samples can make impact on the correlation coefficient while others may not? More details can be found in supplementary file part 1. Finally we get a set of condition specific networks as such.

The second step is module identification for each network. Similar to WGCNA, we also employ hierarchical clustering as the basic method [1, 2]. However, in order to obtain good module identification results, it is crucial to set an optimal cutting height of hierarchical clustering tree, which is usually tune by the users in WGCNA. In our MODA, We propose an automatic method to determine the optimal cutting height based on the quality of modules. Inspired by the concept of partition density of link communities [8], our method search for the optimal cutting height which maximize the average density of resulting modules. Here we simply define the module density as the average edge weights in one module (equation (1) in supplementary file), same as in [2]. We also provide other criterion such as average modularity for weighted network [9] of resulting clusters to determine the cutting height.

The third step is network differential analysis. We compare two networks by comparing two set of modules. The similarity of each pair of modules is measured by a Jaccard similarity coefficient. With all condition-specific networks compared with the background network, we get a similarity matrix A, where each entry A_{ij} means the Jaccard similarity coefficient between the i-th module from the network N_1 and j-th module from the network N_2. Then the elements in row sum of A (vector denoted by $\mathbf{s}$) indicate how much degree that modules in N_1 can be affected by corresponding condition. The higher $\mathbf{s}_i$ means the module i in N_1 may just be responsible for general stress. Especially when some $\mathbf{s}_i$ in N_1 keeps relatively high row sum of A compared with all other N_2 (remove one condition each time), showing these modules have little association with any specific conditions. While lower $\mathbf{s}_i$ means module i in N_1 is very different from the modules in N_2, which may indicate the module has some connection with corresponding condition.

After determining which module may be condition specific or conserved, we can associate biological process with module by functional annotation enrichment analysis. The input can be gene list from the module, or overlapping just part much with others. Here we use DAVID [10] to conduct integrative functional annotation enrichment analysis of gene list based on an R Webservice interface [11]. We implemented a module differential analysis pipeline, from gene expression profile of multiple con-

ditions to enrichment analysis results. Figure shows the general process of each step mentioned above.

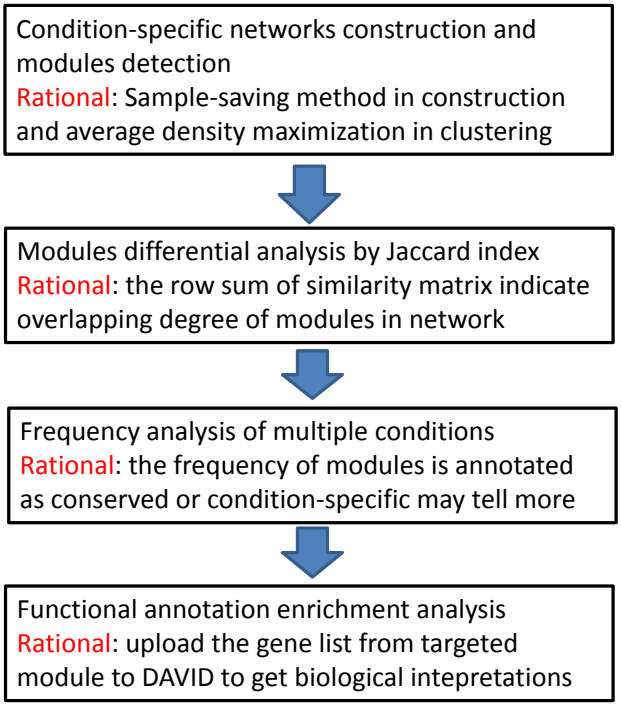

Figure 1: Overview of MODA.

4 Result

We evaluated the effectiveness of proposed methods on both synthetic data and real-world data. By comparing two gene expression profiles generated by different desired correlation matrices of the same set of genes, we can determine the genes affected by a groups definition, which is consistent with the generator. The details for simulation as well as the usage of package can be found in supplementary file part

2. The method is also used on a comprehensive RNA-Seq data set obtained from two natural genotypes fo D. magna, to detect condition-specific as well as conserved responsive genes and biological functions. Several biological meaningful results show the capability of the algorithm, and more details can be found in *Characterization of early stress transcriptional response in the waterflea Daphnia magna* (in preparation).

5 Supplementary

5.1 Concept part

Given gene expression profile $X \in \mathbb{R}^{n\times p}$, where n is the number of experimental samples and p is the number of genes. X_{ij} means the expression value of the j-th gene in i-th sample. The popular tool WGCNA [1] conducts the module detection by hierarchical clustering, i.e. putting similar gene together. The definition of similarity ranges from basic correlation to more complex topological overlap measure [2]. While how to determine the cutting height of hierarchical clustering tree remains an open problem. Here we give the option to chose the height based on the quality of partition. Inspired by the concept of partition density of link communities [8, 12], we choose the cutting height to make the average density of resulting modules to be optimal. The density of one module A is defined as:

$$Density(A) = \frac{\sum_{i\in A}\sum_{j\in A, j\neq i} a_{ij}}{n_A(n_A-1)} \tag{1}$$

where a_{ij} is the similarity between gene i and gene j, and n_A is the number of genes in A. We can also use the modularity Q of weighted network A [9] as the criterion to pick the height of hierarchical clustering tree:

$$Q = \frac{1}{2m}\sum_{ij}[a_{ij} - \frac{k_i k_j}{2m}]\sigma(c_i, c_j) \tag{2}$$

where m is the number of edges and k_i is the connectivity (degree) of gene i, defined as $\sum_j a_{ij}$. And $\sigma(c_i, c_j) = 1$ only when gene i and j are in the same module. The complete module detection and average density is shown in Figure 2.

After the module detection, the co-expression network is represented as a collection of modules (see Figure 3), which makes the differential analysis more focused on the modules other than the nodes or links. By comparing all module pairs from N_1 and N_2, we can get a similarity matrix B, where each entry B_{ij} means the similarity between the i-th module from the network N_1 (denoted by $N_1(A_i)$) and j-th

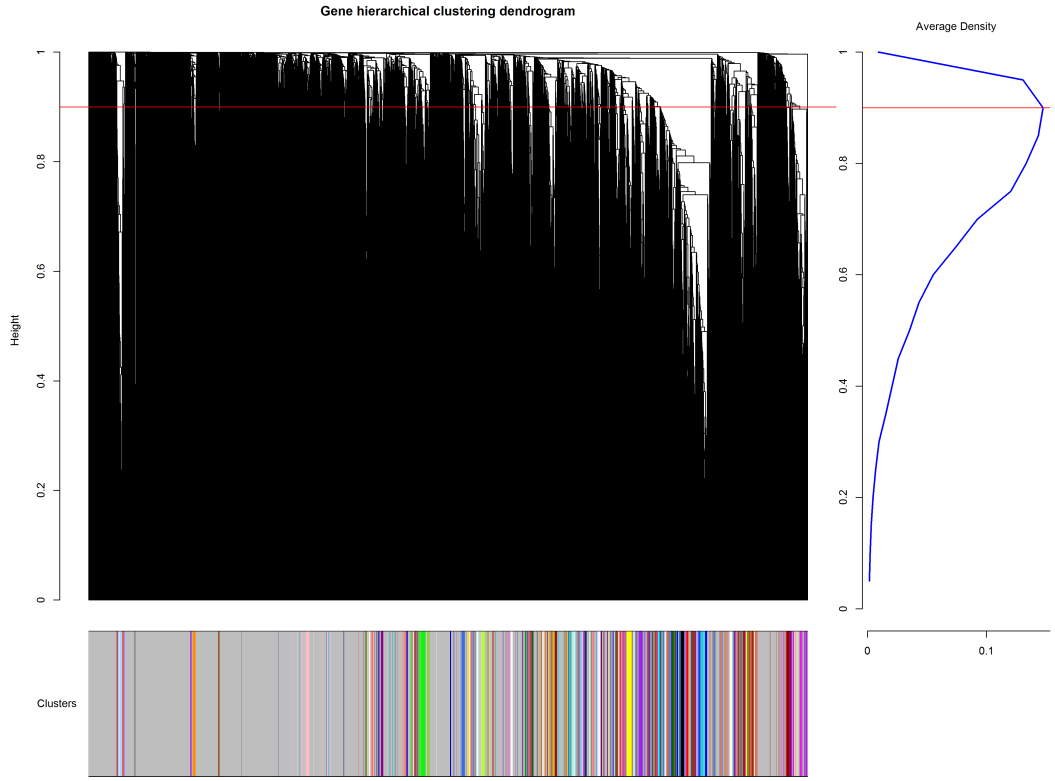

Figure 2: Maximal partition density based hierarchical clustering

module from the network N_2 (denoted by $N_2(A_j)$). The similarity is evaluated by the Jaccard index.

$$B_{ij} = \frac{N_1(A_i) \cap N_2(A_j)}{N_1(A_i) \cup N_2(A_j)} \tag{3}$$

Assume N_1 is background, normally containing samples from all conditions, and the N_2 is constructed from all samples except samples belonging to certain condition D. Let $\mathbf{s}$ is the sums of rows in B, i.e. $\mathbf{s}_i = \sum_j B_{ij}$. The value of $\mathbf{s}_i$ indicates how much the i-th module from network N_1 might be affected by condition D. The rationale behind this statistics is based on the mechanism of correlation, i.e. which samples could make an impact on the correlation while others may not? Figure 3 illustrates an extreme example about how the additional two samples may affect the correlation between X and Y.

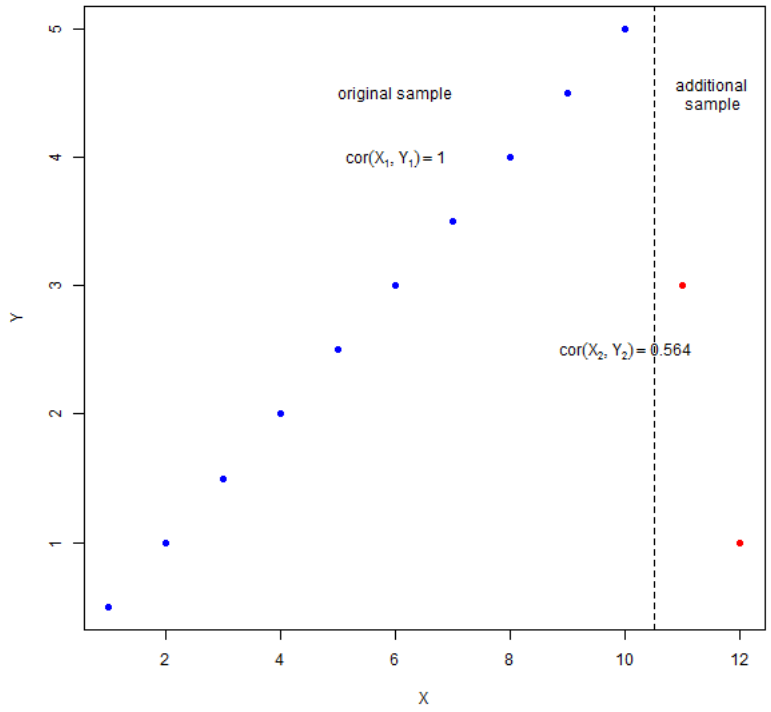

Figure 3: Scatter plot of varibale X and Y

As Figure 4 shows, we use two threshold values here: θ_1 is the threshold to define $min(\mathbf{s}) + \theta_1$, less than which is considered as condition specific module. θ_2 is

the threshold to define $max(\mathbf{s}) - \theta_2$, greater than which is considered as condition conserved module.

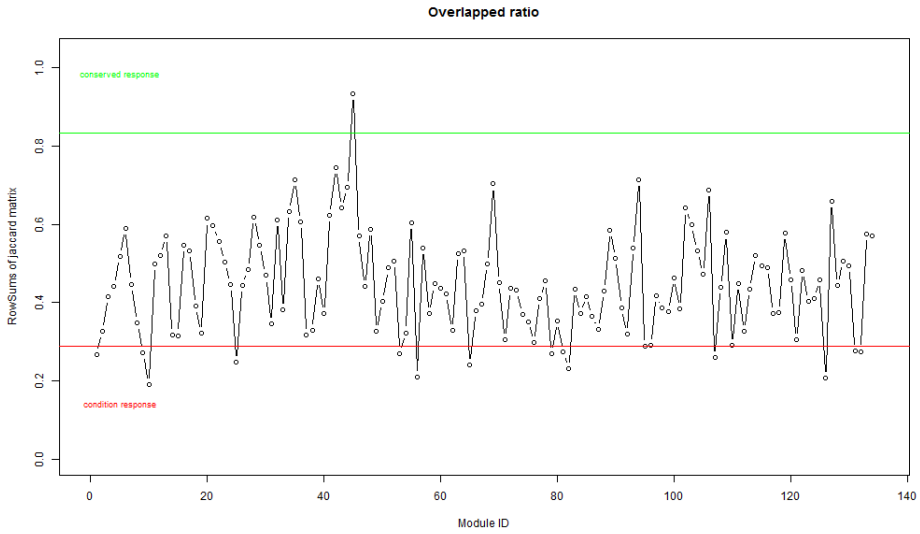

Figure 4: Overlap degree of modules in N_1 with N_2

We also calculate the frequency of each module is annotated as conserved or condition specific and compare all the conditions together. The rationale behind this statistics is based on the mechanism of correlation, i.e. which samples could make an impact on the correlation while others may not? The package visualizes it with a bar plot as Figure 4. A similar plot about the conserved module is also available. The module id is stored as a plain text file for functional enrichment analysis. Here we send one module as gene list to DAVID [10, 11] for integrative analysis.

5.2 Evaluation

We evaluate the effectiveness of proposed methods on both synthetic data and real-world data. The basic synthetic gene expression data is generated by the following logic: given desired correlation matrix $C \in \mathbb{R}^{n \times p}$ with p genes which has a clear modular structure that all genes are equally divided into 5 groups according to the similarities. Then we conduct the Cholesky decomposition on C such that $C = LL^T$, where L is the lower triangular matrix. Finally we project L on random matrix $A \in \mathbb{R}^{n \times p}$ to get desired gene expression matrix $X \in \mathbb{R}^{n \times p}$, which has the rough

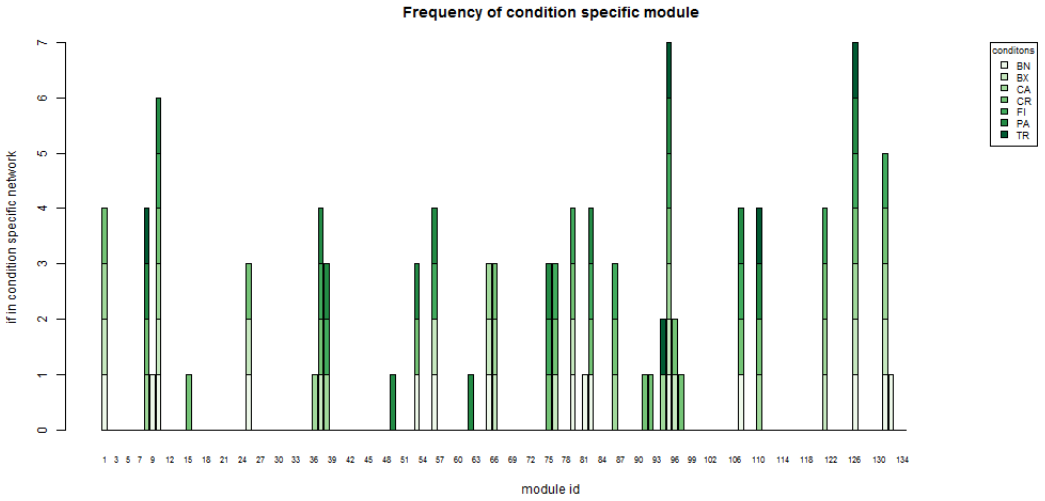

Figure 5: Statistics about which module can be condition specific

modular structure defined by correlation C. Let $n = 500$ and each group has 100 genes in the simulation. In each group, we allocate the gene id from 1-100, 101-200, 201-300, 301-400 and 401-500 respectively. The correlation matrix of genes in X is shown in Figure 6. In another matrix Y, we merge the last two groups into one by adding more samples to X, and the correlation matrix is shown in Figure 7. The we can compare these two networks with proposed method to see which genes were affected. Gene lists in target fold show that modules that contain gene id from 1-100, 101-200 and 201-300 have large overlap with network 2, while module gene id from 301-500 which were merged have least overlap with network 2. The facts are consistent with experimental settings.

Here is the example code to use `MODA` given two gene expression profiles. Results of modules are stored under the newly created folder *ResultFolder* as gene lists. The condition-specific and conserved module ids are stored as plain texts in next directory with the name of indicator which need to be compared. Other materials such as figure 2 and 4 are also available in the folder.

```
library(MODA)
data(synthetic)
ResultFolder = 'ForSynthetic' # where middle files are stored
CuttingCriterion = 'Density'  # could be Density or Modularity
indicator1 = 'X'              # indicator for data profile 1
indicator2 = 'Y'              # indicator for data profile 2
specificTheta = 0.1           #threshold to define condition specific modules
conservedTheta = 0.1          #threshold to define conserved modules
##modules detection for network 1
intModules1 <- WeightedModulePartitionDensity(datExpr1,ResultFolder,indicator1,CuttingCriterion)
##modules detection for network 2
intModules2 <- WeightedModulePartitionDensity(datExpr2,ResultFolder,indicator2,CuttingCriterion)
```

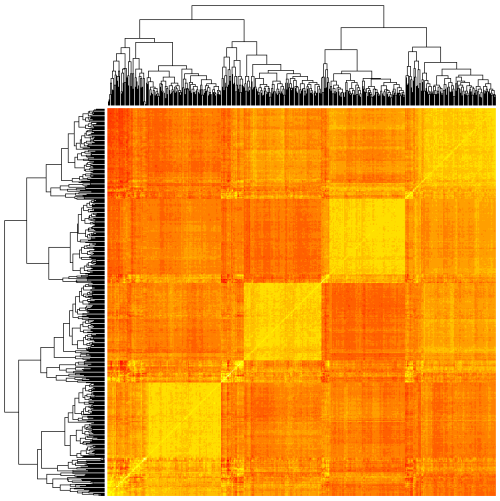

Figure 6: Correlation matrix of X

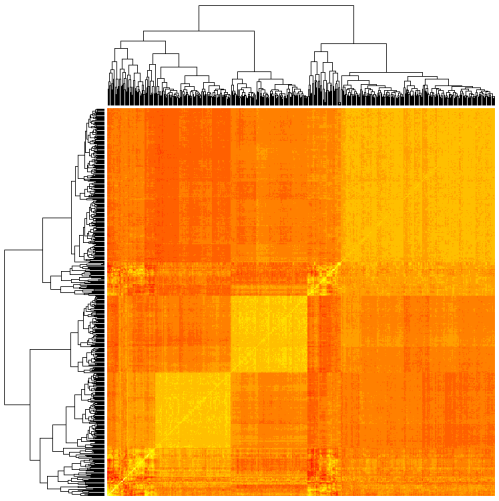

Figure 7: Correlation matrix of Y

```
##modules differential analysis
CompareAllNets(ResultFolder,intModules1,indicator1,intModules2,indicator2,specificTheta,conservedTheta)
```

References

[1] Peter Langfelder and Steve Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559, 2008.

[2] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1), 2005.

[3] Gordon K Smyth. Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*, pages 397–420. Springer, 2005.

[4] YounJeong Choi and Christina Kendziorski. Statistical methods for gene set co-expression analysis. *Bioinformatics*, 25(21):2780–2786, 2009.

[5] David Amar, Hershel Safer, and Ron Shamir. Dissection of regulatory networks that are altered in disease via differential co-expression. *PLoS Comput Biol*, 9(3):e1002955, 2013.

[6] Min Jin Ha, Veerabhadran Baladandayuthapani, and Kim-Anh Do. Dingo: differential network analysis in genomics. *Bioinformatics*, 31(21):3413–3420, 2015.

[7] Marieke Lydia Kuijjer, Matthew Tung, GuoCheng Yuan, John Quackenbush, and Kimberly Glass. Estimating sample-specific regulatory networks. *arXiv preprint arXiv:1505.06440*, 2015.

[8] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[9] Mark EJ Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004.

[10] Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature protocols*, 4(1):44–57, 2008.

[11] Cristóbal Fresno and Elmer A Fernández. Rdavidwebservice: a versatile r interface to david. *Bioinformatics*, page btt487, 2013.

[12] Alex T Kalinka and Pavel Tomancak. linkcomm: an r package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. *Bioinformatics*, 27(14), 2011.