# PCM: A Pairwise Correlation Mining Package for Biological Network Inference

Hao Liang[1], Feiyang Gu[2], Chaohua Sheng[1], Qiong Duan[1], Bo Tian[1], Jun Wu[3], Bo Xu[*,1], and Zengyou He[*,1]

[1]School of Software, Dalian University of Technology, Dalian, China
[2] Baidu Inc., Beijing, China
[3]School of Information Engineering, Zunyi Normal University, Zunyi, China
zyhe@dlut.edu.cn

**Abstract.** One fundamental task in molecular biology is to understand the dependency among genes or proteins to model biological networks. Numerous methods have been developed for reconstructing different types of networks using data sets generated from high-throughput technologies. One widely used method is to calculate the pairwise correlation or association scores between genes or proteins. To date, a software package supporting various types of correlation measures has been lacking. In this paper, we present a pairwise correlation mining package, termed PCM, which supports the commonly used marginal correlation measures, together with two algorithms enabling the estimation of conditional correlations. Two example data sets are used to illustrate how to use this package and demonstrate the importance of having an integrated software package that incorporates various correlation measures. It is anticipated that the PCM package will become a versatile tool for supporting the correlation-based inference of biological networks. The package and source codes of the implementations are available at https://github.com/FeiyangGu/PCM.

**Keywords:** Pairwise Correlation, Network Inference, Correlation Mining

## 1 Background

To understand the relationships between DNA, RNA, proteins or other cellular molecules, it is necessary to infer biochemical networks from genomic data and proteomic data by exploring various types of computational and statistical methods. Such network inference or reverse engineering problem is quite fundamental in bioinformatics, which has drawn much attention during the past decades [1, 5, 9].

The key issue in the network inference procedure is to obtain the interaction relationships among the molecules such as genes and proteins. This issue can be tackled from various angles, leading to different types of computational and statistical network inference models. Probably the most straightforward and

commonly used formulation is to cast the network inference problem as a pairwise correlation mining problem, i.e., calculating the correlation or association scores among each gene or protein pair. Although many correlation measures have been explored in different network inference applications, there is still no consensus on the best one even for one specific application such as the gene regulatory network inference [5]. Therefore, it is highly necessary to have a pairwise correlation mining package that supports various types of correlation measures.

In this paper, we provide PCM, an open-source implementation of pairwise correlation mining algorithms. PCM enables the pairwise correlation mining with a series of marginal correlation measures under the same umbrella. In addition, it implements two low-order conditional correlation mining methods and provides the functionality of clustering-based approximate correlation mining. Examples are provided to illustrate how to use the package and performance evaluation on several real data sets are used to justify the rationale for developing such a package.

The rest of this paper is organized as follows. In Section 2, we describe the main modules of the PCM package with a specific emphasis on the correlation measures and mining algorithms. Section 3 presents examples on package usage and the experimental results. Section 4 concludes the paper.

## 2 Correlation Measures and Algorithms

PCM was implemented in C++, which uses a matrix as the input. Columns and rows in the input matrix correspond to variables (e.g. genes, proteins) and samples (e.g. gene expression profiles), respectively. PCM consists of three main modules.

### 2.1 Marginal Correlation Measures

The Marginal Correlation module calculates the marginal pairwise correlation scores among all candidate pairs and returns a set of pairs whose correlation scores are above the user-specified threshold. Furthermore, it is further divided into two sub-modules: one is used for handling continuous variables and another one is designed to mine pairwise correlations among binary variables.

In the current version, we have implemented 9 correlation measures for continuous variables and 27 correlation measures for binary variables. The correlation measures for continuous variables can be applied to quantify the association strength between gene profiles in gene regulatory network inference, and those measures for binary variables can be used for network inference from qualitative affinity purification-mass spectrometry (AP-MS) data [9].

The names and detailed definitions of these correlation measures are presented in Table 1 and Table 2, respectively. For some measures that need further explanations, we also provide some illustrations at the end of each table.

**Table 1.** The correlation measures for continuous variables implemented in the PCM package. More details and properties about these measures can be found in [12, 7, 2]. In this table, $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ is a set of joint observations from two univariate continuous variables $X$ and $Y$, $\overline{x}$ and $\overline{y}$ denote the mean value of $X$ and $Y$, respectively.

| ID | Measure | Definition |
|---|---|---|
| 1 | Pearson | $\frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2}}$ |
| 2 | Cosine | $\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2}\sqrt{\sum_{i=1}^{n} {y_i}^2}}$ |
| 3 | Jaccard | $\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2}+\sqrt{\sum_{i=1}^{n} {y_i}^2}-\sum_{i=1}^{n} x_i y_i}$ |
| 4 | Overlap | $\frac{\sum_{i=1}^{n} x_i y_i}{min(\sqrt{\sum_{i=1}^{n} {x_i}^2},\sqrt{\sum_{i=1}^{n} {y_i}^2})}$ |
| 5 | Dice | $\frac{2\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2}+\sqrt{\sum_{i=1}^{n} {y_i}^2}}$ |
| 6 | Spearman | $1-\frac{6\sum_{i=1}^{n}(r_i-s_i)^2}{n(n-1)(n+1)}$ |
| 7 | Dot Product | $\sum_{i=1}^{n} x_i y_i$ |
| 8 | Kendall Rank | $\frac{n_c-n_d}{\sqrt{(n_0-n_1)(n_0-n_2)}}$ |
| 9 | Hoeffding's D measure | $\frac{(n-2)(n-3)D_1+D_2-2(n-2)D_3}{n(n-1)(n-2)(n-3)}$ |

*Spearman.* The Spearman's correlation coefficient is defined as the Pearson's correlation coefficient between two variables that have been transformed into ranks, where $r_i$ and $s_i$ are the ranks of $x_i$ and $y_i$, respectively.

*Kendall Rank.* The variables appeared in the Kendall's Rank correlation coefficient are defined as:

$n_c$ = The number of concordant pairs, where two pairs are said to be concordant if $(x_i - x_j)(y_i - y_j) > 0$,

$n_d$ = The number of discordant pairs, where two pairs are said to be disconcordant if $(x_i - x_j)(y_i - y_j) < 0$,

$n_0 = n(n-1)/2$, $n_1 = \sum_i t_i(t_i - 1)/2$,

$n_2 = \sum_j u_j(u_j - 1)/2$,

$t_i$ = The number of tied values in the $i^{th}$ group of ties for $X$,

$u_j$ = The number of tied values in the $j^{th}$ group of ties for $Y$.

*Hoeffding's D measure.* In Hoeffding's D measure, $r_i$ and $s_i$ are the ranks of $x_i$ and $y_i$, respectively. And $q_i = \sum_{j=1}^{n} \phi(x_j, x_i)\phi(y_j, y_i)$, where $\phi(a, b) = 1$ if

**Table 2.** The correlation measures for binary variables provided in the PCM package. Further details and discussions on these measures can be found in [8, 3]. To simplify the notations, we use $X$ ($Y$) to denote the event that $X = 1$ ($Y = 1$) and $\overline{X}$ ($\overline{Y}$) to denote the event that $X = 0$ ($\overline{Y} = 0$), respectively. $P(\cdot)$ represents the probability that one event will occur and $N$ is the number of samples in the data set.

| ID | Measure | Definition |
|---|---|---|
| 1 | Odds Ratio | $\frac{P(X,Y)P(\overline{X},\overline{Y})}{P(X,\overline{Y})P(\overline{X},Y)}$ |
| 2 | Cosine | $\frac{P(X,Y)}{\sqrt{P(X)P(Y)}}$ |
| 3 | Support | $P(X,Y)$ |
| 4 | Yule's Q | $\frac{P(X,Y)P(\overline{X},\overline{Y})-P(X,\overline{Y})P(\overline{X},Y)}{P(X,Y)P(\overline{X},\overline{Y})+P(X,\overline{Y})P(\overline{X},Y)}$ |
| 5 | Yule's Y | $\frac{\sqrt{P(X,Y)P(\overline{X},\overline{Y})}-\sqrt{P(X,\overline{Y})P(\overline{X},Y)}}{\sqrt{P(X,Y)P(\overline{X},\overline{Y})}+\sqrt{P(X,\overline{Y})P(\overline{X},Y)}}$ |
| 6 | Kappa | $\frac{P(X,Y)+P(\overline{X},\overline{Y})-P(X)P(Y)-P(\overline{X})P(\overline{Y})}{1-P(X)P(Y)-P(\overline{X})P(\overline{Y})}$ |
| 7 | J-Measure | $max\{P(X,Y)log(\frac{P(Y\|X)}{P(Y)}) + P(X,\overline{Y})log(\frac{P(\overline{Y}\|X)}{P(\overline{Y})}),$ $P(X,Y)log(\frac{P(X\|Y)}{P(X)}) + P(\overline{X},Y)log(\frac{P(\overline{X}\|Y)}{P(\overline{X})})\}$ |
| 8 | Gini Index | $max\{P(X)[P(Y\|X)^2+P(\overline{Y}\|X)^2]+P(\overline{X})[P(Y\|\overline{X})^2+P(\overline{Y}\|\overline{X})^2]-$ $P(Y)^2 - P(\overline{Y}^2), P(Y)[P(X\|Y)^2 + P(\overline{X}\|Y)^2] + P(\overline{Y})[P(X\|\overline{Y})^2 +$ $P(\overline{X}\|\overline{Y})^2] - P(X)^2 - P(\overline{X}^2)\}$ |
| 9 | Confidence | $max\{P(Y\|X), P(X\|Y)\}$ |
| 10 | Laplace | $max\{\frac{N\times P(X,Y)+1}{N\times P(X)+2}, \frac{N\times P(X,Y)+1}{N\times P(Y)+2}\}$ |
| 11 | Conviction | $max\{\frac{P(X)P(\overline{Y})}{P(X,\overline{Y})}, \frac{P(Y)P(\overline{X})}{P(Y,\overline{X})}\}$ |
| 12 | Interest | $\frac{P(X,Y)}{P(X)P(Y)}$ |
| 13 | Piatetsky-Shapiro's | $P(X,Y) - P(X)P(Y)$ |
| 14 | Certainty Factor | $max\{\frac{P(Y\|X)-P(Y)}{1-P(Y)}, \frac{P(X\|Y)-P(X)}{P(X)}\}$ |
| 15 | Added Value | $max\{P(Y\|X) - P(Y), P(X\|Y) - P(X)\}$ |
| 16 | Collective Strength | $\frac{P(X,Y)+P(\overline{X},\overline{Y})}{P(X)P(Y)+P(\overline{X})P(\overline{Y})} \times \frac{1-P(X)P(Y)-P(\overline{X})P(\overline{Y})}{1-P(X,Y)-P(\overline{X},\overline{Y})}$ |
| 17 | Jaccard | $\frac{P(X,Y)}{P(X)+P(Y)-P(X,Y)}$ |
| 18 | Klosgen | $\sqrt{P(X,Y)}max\{P(Y\|X) - P(Y), P(X\|Y) - P(X)\}$ |
| 19 | $\phi$-coefficient | $\frac{P(X,Y)-P(X)P(Y)}{\sqrt{P(X)P(Y)(1-P(X))(1-P(Y))}}$ |
| 20 | Probability Ratio | $log\frac{P(X,Y)}{P(X)P(Y)}$ |
| 21 | BCPNN | $log(\frac{P(X,Y)+cc}{P(X)P(Y)+cc})$ |
| 22 | SCWCC | $N\frac{(P(X,Y)-P(X)P(Y))^2}{P(X)P(Y)+cc}$ |
| 23 | Two-way Support | $P(X,Y)log(\frac{P(X,Y)}{P(X)P(Y)})$ |
| 24 | SCWS | $N \times P(X,Y)\frac{(P(X,Y)-P(X)P(Y))^2}{P(X)P(Y)}$ |
| 25 | Simplified $\chi^2$-statistic | $N\frac{(P(X,Y)-P(X)P(Y))^2}{P(X)P(Y)}$ |
| 26 | Likelihood Ratio | $N[P(X,Y)log(\frac{P(X,Y)}{P(X)P(Y)}) + (1 - P(X,Y))log(\frac{1-P(X,Y)}{1-P(X)P(Y)})]$ |
| 27 | Mutual Information | $P(X,Y)log(\frac{P(X,Y)}{P(X)P(Y)}) + P(\overline{X},Y)log(\frac{P(\overline{X},Y)}{P(\overline{X})P(Y)}) +$ $P(X,\overline{Y})log(\frac{P(X,\overline{Y})}{P(X)P(\overline{Y})}) + P(\overline{X},\overline{Y})log(\frac{P(\overline{X},\overline{Y})}{P(\overline{X})P(\overline{Y})})$ |

$a < b$ and $\phi(a, b) = 0$ otherwise. $D_1$, $D_2$ and $D_3$ are defined as:

$$D_1 = \sum_{i=1}^{n} q_i(q_i - 1),$$
$$D_2 = \sum_{i=1}^{n} (r_i - 1)(r_i - 2)(s_j - 1)(s_j - 2),$$
$$D_3 = \sum_{i=1}^{n} (r_i - 2)(s_i - 2)q_i.$$

*Odds ratio*. It is defined as the ratio of the odds of an event occurring in one group to the odds of it occurring in another group. Here one group corresponds to $Y = 1$ and another group corresponds to $Y = 0$.

*Cosine*. The cosine of two variables can be considered as the normalized inner product, which ranges from -1 to +1.

*Support*. It is originally proposed for association rule mining in the data mining society, which has been widely used in different applications.

*Yule's Q* and *Yule's Y*. These two measures are normalized variants of the odds ratio, whose values fall into the interval [-1, 1].

*Kappa*. This measure captures the degree of consistency between two binary variables. If these two variables are highly correlated with each other, then the values for $P(X, Y)$ and $P(\overline{X}, \overline{Y})$ will be large, which in turn, leads to a higher correlation value.

*Mutual Information*, *J-measure* and *Gini*. The entropy is the key concept in information theory, which is indeed related to the variance of a probability distribution. Intuitively, the entropy of a uniform distribution is larger than that of a skewed distribution. The mutual information evaluates the dependencies among variables by the amount of reduction in the entropy of one variable when the value of another variable is given. The measures such as J-Measure and Gini index are defined according to the same principle.

*Confidence*, *Laplace* and *Conviction*. The confidence measure is originally used in association rule mining for measuring the accuracy of a given rule. The Laplace function and the conviction function are two variants of the confidence measure.

*Interest*, *Piatetsky-Shapiro's*, *Certainty factor*, *Collective strength* and *Added value*. The Interest measure is widely used in data mining for measuring the deviation from statistical independence. Piatetsky-Shapiro's measure, Certainty factor, Collective strength and Added value are four variants that extend the Interest measure from different angles.

*Jaccard*. The Jaccard measure has been used extensively in information retrieval to measure the similarity between two documents.

*$\phi$-coefficient*. This measure is the variant of Pearson's correlation coefficient for binary variables, which is closely related to the $\chi^2$ statistic since $\phi^2 = \chi^2/N$.

*BCPNN* and *SCWCC*. In these two measures, *cc* is a user-specified parameter, which is usually assigned a larger value if the data set is noisy and a smaller value otherwise.

### 2.2 Clustering-based Approximate Correlation Mining

To conduct an exhaustive pairwise correlation mining, the time complexity is at least $O(n^2)$, where $n$ is the number of variables. To handle the data sets with large number of variables, an alternative approach is to generate an approximate set of correlated pairs so as to reduce the time complexity. We also provide a fast algorithm for generating an approximate set of correlated pairs. This algorithm first uses clustering algorithms such as $k$-means to partition the variables into different clusters, then the correlation coefficients of all candidate pairs within each cluster are calculated. Finally, the mining results across all clusters are gathered to generate the final set of correlated pairs.

In the PCM package, we implement a clustering-based method for this purpose. This method has two steps: clustering and correlation mining. In the first step, it uses clustering algorithms to partition the variables into different groups. For continuous variables, the clustering algorithm employed is the well-known $k$-means method, which requires the number of clusters as an input parameter. For binary variables, the CLOPE algorithm [10] is utilized, which also has an user-specified parameter called repulsion for controlling the level of intra-cluster similarity. It is expected that highly correlated variables will be put into the same cluster so that it is sufficient to perform correlation mining within each cluster. Therefore, in the second step, the correlation coefficients of all candidate pairs within each group are calculated. And the mining results across all clusters are merged to generate the final set of correlated pairs.

Suppose the number of generated clusters is $k$ (all clusters has the same size) and the clustering algorithm has a linear time complexity, then this method will reduce the time complexity of pairwise correlation mining from $O(n^2)$ to $O(n^2/k)$. Apparently, such performance gain in running efficiency is at the cost of missing some really correlated pairs of variables if these variables are not assigned into the same cluster. For instance, if we handle the example data set used in the experimental section when the number of cluster is set to be 2 and the threshold for the Pearson's correlation coefficient is 0.5, 9 of the 49 correlated pairs that are above the threshold will be missed.

### 2.3 Algorithms for Conditional Correlation Mining

Since the marginal correlation cannot distinguish direct and indirect associations (the induced associations due to other variables), the Conditional Correlation module implements two existing algorithms for mining low-order conditional pairwise correlations: LOPC [13] and CMI2NI [11]. The conditional correlation measures the correlation between two variables after their dependence on other variables is removed. Thus, these algorithms are able to remove pairs whose correlation relationship may mainly come from their mutual dependencies on other variables.

**CMI2NI** For the CMI2NI method, we provide two implementations for different types of variables: CMI2NIC and CMI2NIB. The former implementation

strictly follows [11], which can be used to handle data sets with continuous variables. While the latter implementation is designed for data sets with binary variables, which computes the conditional mutual information between two binary variables directly.

The CMI2NI [11] algorithm takes a matrix and a user-specified threshold as the input and outputs a set of variable pairs (i.e., a network with a set of detected edges), which are correlated even on condition the existence of other variables.

In this algorithm, the correlation measure CMI2 is defined based on the conditional mutual information. To make the computation fast and exact, the authors further assume a Gaussian distribution on the target data. This makes it possible to calculate the conditional mutual information with a concise formula involving the covariance matrix of the data set.

To handle data sets with binary variables, we also provide an implementation that follows the same procedure. In this implementation named CMI2NIB, we can perform the conditional pairwise correlation mining from the binary data set.

**LOPC** The LOPC [13] algorithm is similar to that of CMI2NI. In this method, the zero-th, first and second order partial correlation coefficients for a candidate variable pair are calculated in an iterative manner. For instance, if the zero-th order partial correlation coefficient of one variable pair is less than the given threshold, then this pair would not be considered in the subsequent evaluation with respect to higher order partial correlations.

## 3 Results

Two example data sets were delivered within the PCM package. The first data set, "data_DREAM3.txt", is a DREAM3 data set about the Yeast knock-out gene expressions, which is composed of 100 continuous variables and 100 samples. The second data set is produced from the AP-MS data in the reference [4], which has 2761 binary variables (the presence/absence of each protein in a purification) and 2166 samples (each sample corresponds to a purification).

### 3.1 An Example Application

In this section, we use the first data set as the example data. The following commands can be used to find all gene pairs whose Pearson's correlation coefficients are no less than 0.2 from this data set and put the mining results into the file "out.txt":

```
PCM PearsonC data_DREAM3.txt out.txt 0.2.
```

To rapidly obtain a set of correlated pairs, we can use the following commands:

```
PCM PearsonC data_DREAM3.txt out.txt 0.2 1 5,
```

where "1" means that we use clustering methods to obtain an approximate mining results, and "5" is the number of clusters specified by the users.

Furthermore, the following commands will use the algorithm LOPC in [13] to retain only pairs whose conditional correlation coefficients (from 0th order up to 2nd order) are significant enough.

`PCM LOPCC data_DREAM3.txt out.txt`.

### 3.2 Comparison Results

In this section, we use the second data set, the AP-MS data set from Gavin et al [4], as an example to illustrate the necessity of implementing a software package that are composed of many correlation measures and algorithms.

To validate network inference results from the AP-MS data, we use three reference sets of experimentally validated binary protein interactions for the performance assessment. These reference sets are denoted as Y2H, PCA, and BGS, respectively [6]. As shown in Fig.1, different correlation measures may yield significantly different network inference results. Moreover, we do not know which measure will achieve the best performance for a specific data set. Therefore, it is necessary to have software package that can provide various correlation measures for the end users.

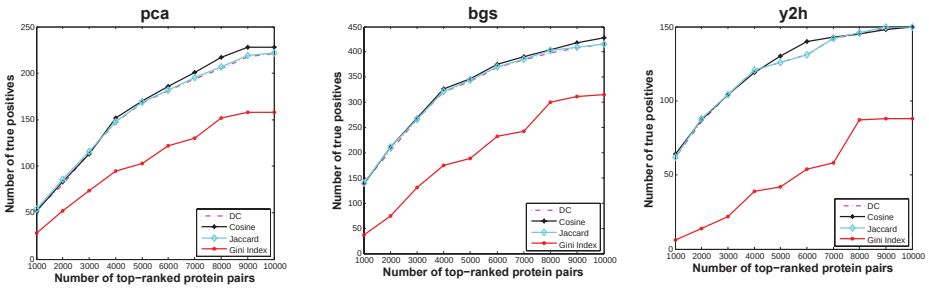


**Fig. 1.** The performance comparison on the Gavin data set when DC (Dice Coefficient), Cosine, Jaccard and Gini index are used as the correlation measures. For each reference set and correlation measure, we report a set of top-ranked interactions (x-axis) to check how many interactions are contained in the the reference set (y-axis).

## 4 Conclusions

PCM enables automated pairwise correlation mining using various types of correlation measures. Different correlation mining methods can be applied in a single framework, enabling easy and fast comparison and selection of the most suitable correlation mining method for a biological network inference task.

## Availability and requirements

**Project name**: PCM.
**Project home page**: https://github.com/FeiyangGu/PCM.
**Operating system**: Windows platform.
**Programming language**: C++.
**Other requirements**: Eigen library.
**License**: GNU GPL.
**Any restrictions to use by non-academics**: Licence needed.

## Acknowledgements

This work was partially supported by the Natural Science Foundation of China (Nos.61572094, 61502071), the Fundamental Research Funds for the Central Universities (No.DUT2017TB02) and the Science-Technology Foundation for Youth of Guizhou Province (No.KY[2017]250).

## References

1. De Smet, R., Marchal, K.: Advantages and limitations of current network inference methods. Nature Reviews Microbiology 8(10), 717–729 (2010)
2. Deshpande, R., VanderSluis, B., Myers, C.L.: Comparison of profile similarity measures for genetic interaction networks. PLoS One 8(7), e68664 (2013)
3. Duan, L., Street, W.N., Liu, Y., Xu, S., Wu, B.: Selecting the right correlation measure for binary data. ACM Transactions on Knowledge Discovery from Data 9(2), 13 (2014)
4. Gavin, A.C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Rau, C., Jensen, L.J., Bastuck, S., Dümpelfeld, B., et al.: Proteome survey reveals modularity of the yeast cell machinery. Nature 440(7084), 631–636 (2006)
5. Kurt, Z., Aydin, N., Altay, G.: A comprehensive comparison of association estimators for gene network inference algorithms. Bioinformatics 30(15), 2142–2149 (2014)
6. Schelhorn, S.E., Mestre, J., Albrecht, M., Zotenko, E.: Inferring physical protein contacts from large-scale purification data of protein complexes. Molecular & Cellular Proteomics 10(6), M110–004929 (2011)
7. de Siqueira Santos, S., Takahashi, D.Y., Nakata, A., Fujita, A.: A comparative study of statistical methods used to identify dependencies between gene expression signals. Briefings in Bioinformatics 15(6), 906–918 (2014)
8. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Information Systems 29(4), 293–313 (2004)
9. Teng, B., Zhao, C., Liu, X., He, Z.: Network inference from AP-MS data: computational challenges and solutions. Briefings in bioinformatics 16(4), 658–674 (2015)
10. Yang, Y., Guan, X., You, J.: CLOPE: a fast and effective clustering algorithm for transactional data. In: Proceedings of the Eighth ACM SIGKDD International conference on Knowledge Discovery and Data Mining. pp. 682–687. ACM (2002)
11. Zhang, X., Zhao, J., Hao, J.K., Zhao, X.M., Chen, L.: Conditional mutual inclusive information enables accurate quantification of associations in gene regulatory networks. Nucleic acids research 43(5), e31–e31 (2015)

12. Zhao, Y., Zou, Q., Jiang, Y., Wang, G.: A graphic processing unit web server for computing correlation coefficients for gene expression data. Journal of Computational and Theoretical Nanoscience 12(4), 582–584 (2015)
13. Zuo, Y., Yu, G., Tadesse, M.G., Ressom, H.W.: Biological network inference using low order partial correlation. Methods 69(3), 266–273 (2014)