Communications for Statistical Applications and Methods
2022, Vol. 29, No. 5, 591–601

https://doi.org/10.29220/CSAM.2022.29.5.591
Print ISSN 2287-7843 / Online ISSN 2383-4757

Comparison of covariance thresholding methods in gene set analysis

Sora Park[a], Kipoong Kim[a], Hokeun Sun[1,a]

[a]Department of Statistics, Pusan National University, Korea

Abstract

In gene set analysis with microarray expression data, a group of genes such as a gene regulatory pathway and a signaling pathway is often tested if there exists either differentially expressed (DE) or differentially co-expressed (DC) genes between two biological conditions. Recently, a statistical test based on covariance estimation have been proposed in order to identify DC genes. In particular, covariance regularization by hard thresholding indeed improved the power of the test when the proportion of DC genes within a biological pathway is relatively small. In this article, we compare covariance thresholding methods using four different regularization penalties such as lasso, hard, smoothly clipped absolute deviation (SCAD), and minimax concave plus (MCP) penalties. In our extensive simulation studies, we found that both SCAD and MCP thresholding methods can outperform the hard thresholding method when the proportion of DC genes is extremely small and the number of genes in a biological pathway is much greater than a sample size. We also applied four thresholding methods to 3 different microarray gene expression data sets related with mutant p53 transcriptional activity, and epithelium and stroma breast cancer to compare genetic pathways identified by each method.

Keywords: covariance thresholding, regularization, gene set analysis, differentially co-expressed genes

1. Introduction

Gene set analysis with microarray expression data aims to identify a group of genes such as a gene regulatory pathway and a signaling pathway that has either differentially expressed (DE) or differentially co-expressed (DC) genes between two biological conditions. In the past, statistical methods for gene set analysis mainly focused on detecting DE genes (Subramanian *et al.*, 2005; Goeman and Bühlmann, 2007; Dinu *et al.*, 2007; Wu *et al.*, 2010). Recently, identification of DC genes has been receiving increasing attention so statistical methods using covariance estimation for a set of genes have been proposed (Choi and Kendziorski, 2009; Rahmatallah *et al.*, 2014; Oh *et al.*, 2020). For example, Oh *et al.* (2020) proposed covariance thresholding for gene set analysis (CTGSA), which is a statistical test based on estimation of co-expression levels of paired genes using covariance regularization by hard thresholding. They demonstrated that CTGSA substantially improves the detection power of DC genes when the proportion of DC genes within a biological pathway is relatively small.

In statistics, thresholding a sample covariance matrix is a natural approach to estimate a sparse covariance matrix. Bickel and Levina (2008) studied the theoretical properties of hard-thresholded covariance. Since covariance thresholding requires essentially no computational burden, the implementation of covariance estimation for high-dimensional gene expression data is simple. Rothman

This work was supported by a 2-Year Research Grant of Pusan National University.

[1] Corresponding author: Department of Statistics, Pusan National University, Busandaehak-ro 63beon-gil, Geumjeong-Gu, Busan 46241, Korea. E-mail: hsun@pusan.ac.kr

Published 30 September 2022 / journal homepage: http://csam.or.kr

et al. (2009) applied generalized thresholding of the sample covariance matrix, where they defined a generalized thresholding operator to conduct elementwise shrinkage and thresholding for the sample covariance matrix. Their thresholding operator includes hard thresholding, soft thresholding, and smoothly clipped absolute deviation (SCAD). In their simulation, the SCAD thresholding of the sample covariance had the best performance to estimate high-dimensional covariance.

The SCAD penalty function is non-convex unlike the lasso penalty, so it essentially compromises the shrinkage lasso estimates and discontinuity of hard thresholding (Fan and Li, 2001). Due to its optimal theoretical properties, the SCAD penalty has been widely used in variables selection problems with high-dimensional data (Zou and Li, 2008; Fan *et al.*, 2009). Similar to the SCAD penlty, Zhang (2010) has proposed a minimax concave plus (MCP) penalty, which is known as the most accurate method among penalized variable selection methods in linear regression framework. Both MCP and SCAD asymptotically achieve the oracle properties in terms of consistent estimation and fully recovery of sparsity patterns. But, Zhang (2010) has demonstrated that MCP has superior selection accuracy over SCAD through his simulation studies with high-dimensional data.

In this article, we extend the generalized thresholding operator to the MCP penalty and compare four different covariance thresholding methods to identify DC genes in gene set analysis. Specifically, we first applied lasso, hard, SCAD and MCP thresholding to estimate co-expression levels of paired genes between two biological conditions. Next, significance of difference in covariance estimation between two conditions is evaluated through a permutation test. Finally, the empirical powers of the statistical test using four different thresholding methods are then compared with each other. We also applied them to 3 different microarray gene expression data sets to compare gene sets and genetic pathways identified by each method.

2. Method

2.1. Covariance thresholding for gene set analysis

Let us denote the gene expression values of the j^{th} gene for the condition $l \in \{1, 2\}$ by $(x_{1j}^{(l)}, x_{2j}^{(l)}, \ldots, x_{n_l j}^{(l)})$, where $j = 1, 2, \ldots, p$, and n_l is the sample size of condition l. Without loss of generality, we assume that

$$\sum_{k=1}^{n_l} x_{kj}^{(l)} = 0 \qquad \text{and} \qquad \sum_{k=1}^{n_l} \left(x_{kj}^{(l)}\right)^2 = n_l - 1$$

for all j. Then, the sample covariance of the i^{th} gene and the j^{th} gene is

$$\hat{\sigma}_{ij}^{(l)} = \frac{1}{n_l - 1} \sum_{k=1}^{n_l} x_{ki}^{(l)} x_{kj}^{(l)}$$

for the condition l. Also, the sample covariance matrix of the condition l is $\hat{\Sigma}^{(l)} = \{\hat{\sigma}_{ij}^{(l)}\}$ for $1 \le i, j \le p$. Next, we define a hard thresholding operator for the sample covariance as

$$s_H\left(\hat{\Sigma}, \tau\right) = \left\{\hat{\sigma}_{ij} I\left(|\hat{\sigma}_{ij}| > \tau\right)\right\},$$

where $I(\cdot)$ is an indicator function and $\tau > 0$ is a tuning parameter to control the sparsity of the covariance matrix. Finally, the test statistic of CTGSA based on the hard thresholding is

$$T_H = \lambda_{\max}\left(s_H\left(\hat{\Sigma}^{(1)}, \tau_1\right)\right) - \lambda_{\max}\left(s_H\left(\hat{\Sigma}^{(2)}, \tau_2\right)\right), \tag{2.1}$$

where $\lambda_{\max}(M)$ is the largest eigenvalue of a matrix M, and $\tau_1 > 0$ and $\tau_2 > 0$ are the tuning parameters for conditions 1 and 2, respectively. It essentially tests

$$H_0 : \text{There are no differentially co-expressed genes in a gene set}$$

against the existence of DC genes. If the numerical value of the test statistic is relatively large, we can reject H_0 and conclude that the gene set contains DC genes.

According to Oh *et al.* (2020), their test statistic can capture the largest variation of weighted linear combinations of correlations among genes using a sparse covariance matrix, which can drastically reduce down the computational cost. CTGSA employed a permutation test to evaluate the significance of the test statistic. The permutation test has been popularly used by statistical methods to detect DC genes such as gene set co-expression analysis (GSCA) and gene sets net correlation analysis (GSNCA) (Choi and Kendziorski, 2009; Rahmatallah *et al.*, 2014). The test statistics of GSCA and GSNCA are based on computation of a sample covariance or correlation matrix. Therefore, the permutation of GSCA and GSNCA requires much greater computational burden than that of CTGSA based on a sparse covariance when the permutation is applied to a gene set with a large number of genes. Moreover, Oh *et al.* (2020) demonstrated that the power of CTGSA is greater than the powers of GSCA and GSNCA when the proportion of DC genes within a gene set is relatively small.

2.2. Generalized covariance thresholding

In covariance estimation with a high-dimensional data, Rothman *et al.* (2009) investigated that SCAD thresholding can outperform hard thresholding when a true covariance is sparse. They also pointed out that hard thresholding tended to threshold too many entries, especially in high-dimensional data. In practice, many biological pathways consist of high-dimensional genes, where the number of genes is much greater the sample size. Additionally, the number of correlated genes within a genetic pathway is limited so true covariance of gene expression is expected to be very sparse. Consequently, we considered to replace hard thresholding of CTGSA by other covariance thresholding methods such as lasso, SCAD and MCP.

Rothman *et al.* (2009) defined a generalized thresholding operator as

$$s(x, \tau) = \arg\min_{\theta} \left\{ \frac{1}{2}(\theta - x)^2 + p_\tau(\theta) \right\},$$

where $p_\tau(\cdot)$ is a penalty function of regularization methods with a tuning parameter $\tau > 0$. For example, we can define a lasso thresholding operator for the sample covariance $\hat{\Sigma}$ as

$$s_L\left(\hat{\Sigma}, \tau\right) = \left\{ \text{sign}\left(\hat{\sigma}_{ij}\right)\left(\left|\hat{\sigma}_{ij}\right| - \tau\right)_+ \right\} = \begin{cases} \hat{\sigma}_{ij} - \tau, & \text{if } \hat{\sigma}_{ij} > \tau, \\ \hat{\sigma}_{ij} + \tau, & \text{if } \hat{\sigma}_{ij} < -\tau, \\ 0, & \text{otherwise.} \end{cases}$$

if we use a l_1-norm penalty $p_\tau(\theta) = \tau|\theta|$. The lasso thresholding is equivalent to a soft thresolding function.

Similarly, the SCAD penalty is written as

$$p_\tau(\theta) = \tau I(|\theta| \le \tau) + \frac{(a_0\tau - |\theta|)_+}{a_0 - 1} I(|\theta| > \tau)$$

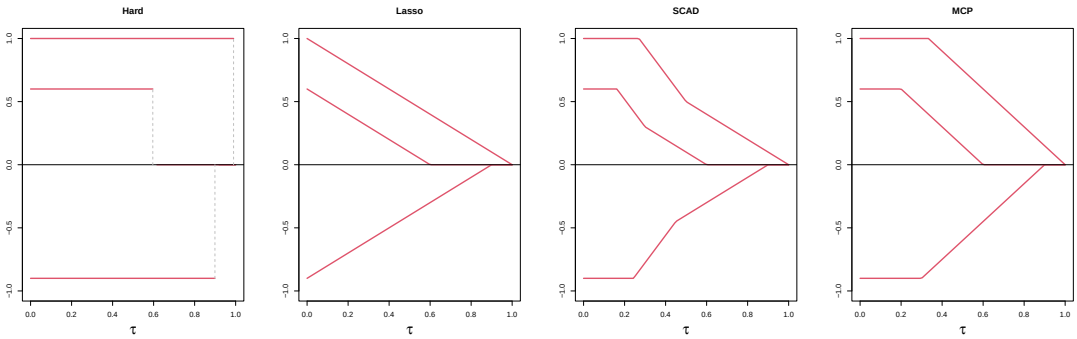

Figure 1: *Regularization paths of four thresholding operators including hard, lasso, SCAD and MCP are displayed along with a tuning parameter* τ.

for some $a_0 > 2$. Subsequently, the SCAD thresholding operator for the sample covariance is then

$$s_C\left(\hat{\Sigma}, \tau\right) = \begin{cases} \text{sign}\left(\hat{\sigma}_{ij}\right)\left(\left|\hat{\sigma}_{ij}\right| - \tau\right)_+, & \text{if } \left|\hat{\sigma}_{ij}\right| \le 2\tau, \\ \dfrac{(a_0 - 1)\hat{\sigma}_{ij} - \text{sign}\left(\hat{\sigma}_{ij}\right) a_0 \tau}{a_0 - 2}, & \text{if } 2\tau < \left|\hat{\sigma}_{ij}\right| \le a_0\tau, \\ \hat{\sigma}_{ij}, & \text{otherwise.} \end{cases}$$

The value $a_0 = 3.7$ was recommended by Fan and Li (2001), so we use it throughout the article.

Finally, we apply the MCP penalty to the generalized thresholding operator. Zhang (2010) defined the MCP penalty function as

$$p_\tau(\theta) = \tau\left(|\theta| - \frac{\theta^2}{2\tau\gamma}\right) I\left(|\theta| < \tau\gamma\right) + \frac{\tau^2\gamma}{2} I\left(|\theta| \ge \tau\gamma\right),$$

where $\gamma > 1$ is another tuning parameter for the MCP penalty. Correspondingly, the MCP thresholding operator for the sample covariance can be expressed as

$$s_M\left(\hat{\Sigma}, \tau\right) = \begin{cases} 0, & \text{if } \left|\hat{\sigma}_{ij}\right| \le \tau, \\ \dfrac{\gamma}{\gamma - 1}\text{sign}\left(\hat{\sigma}_{ij}\right)\left(\left|\hat{\sigma}_{ij}\right| - \tau\right), & \text{if } \tau < \left|\hat{\sigma}_{ij}\right| \le \tau\gamma, \\ \left|\hat{\sigma}_{ij}\right|, & \text{otherwise.} \end{cases}$$

Note that MCP thresholding is simply reduced to lasso thresholding as $\gamma \to \infty$, and hard threshold as $\gamma \to 1+$. Therefore, MCP thresholding can be viewed as the optimized thresholding method combined lasso thresholding and hard thresholding with an intermediate point of $\tau\gamma$, where γ controls concavity of the penalty function. For the equivalent computational cost of four thresholding methods, we just fixed $\gamma = 3$ like the value a_0 of SCAD thresholding. For illustration, Figure 1 shows the behaviors of four thresholding operators through the regularization path in a simple simulation example. We considered three different values of covariance such as 1.0, 0.6 and -0.9. The numerical values of four thresholding operators for these three values are present in the y-axis while the tuning parameter τ is shown in the x-axis. Note that the numerical values of four thresholding operators are the same as 1.0, 0.6 and -0.9 when $\tau = 0$. But, these values are eventually shrinkaged toward 0 as τ is increasing.

It appears that the regularization paths of four thresholding methods are clearly different with each other. So, each covariance thresholding method produces a different estimation result even though zero entries of the covariance matrix can be identical with the same value of τ.

2.3. Tuning parameter and permutation

We have defined four different thresholding operators so far. They are hard, lasso, SCAD and MCP thresholding denoted by s_H, s_L, s_C and s_M, respectively. These thresholding methods require to select the optimal tuning parameter τ. Cross-validation is one of the most popular way to choose the optimal tuning parameter, where samples are randomly split into training sets and validation sets in order to measure an estimation error for each of candidate values of the tuning parameter (Bickel and Levina, 2008; Rothman *et al.*, 2009). However, cross-validation often results in unstable and inconsistent estimation of the tuning parameter due to random split unless there are extremely large samples. In contrast, Oh *et al.* (2020) proposed to use the quantile estimate of the absolute values of sample covariances without random sampling. Specifically, they estimated the proportion of the genes that are not differentially co-expressed, using the largest eigenvalue of the sample covariance matrix.

Let us denote the δ^{th} largest absolute value of the sample covariance among $N = p(p-1)/2$ entries by $r_{[\delta]}$. For example, $r_{[1]} = \max_{i\neq j} |\hat{\sigma}_{ij}|$ and $r_{[N]} = \min_{i\neq j} |\hat{\sigma}_{ij}|$. Since the sample covariance matrix is symmetric, only half of off-diagonal entries of the covariance matrix was considered. The proportion of variance accounted by the first principal component can be computed by $q = \lambda_{\max}(|\hat{\Sigma}|)/p$. The optimal tuning parameter is then

$$\hat{\tau} = r_{[\hat{\delta}]} \quad \text{and} \quad \hat{\delta} = \lceil qN \rceil,$$

where the ceiling function $\lceil x \rceil$ means the smallest integer value greater than or equal to x. Oh *et al.* (2020) demonstrated that CTGSA with the optimal tuning parameter based on the quantile estimate is very robust, regardless of the proportion of DC genes and the numerical value of inter-gene correlation coefficient. Therefore, we also adopt these quantile estimate for the optimal tuning parameter $\hat{\tau}$ of four thresholding methods.

The test statistic of CTGSA, T_H in (2.1) depends on the hard thresholding operator s_H and the tuning parameters τ_1 and τ_2. We consider three additional versions of CTGSA, namely, T_L, T_C and T_M, where s_H of the test statistic T_H is replaced by s_L, s_C and s_M, respectively. We also assume that each threholding method employs the optimal tuning parameters $\hat{\tau}_1$ and $\hat{\tau}_2$ based on the quantile estimates of the sample covariance of the conditions 1 and 2, respectively. Significance of each test statistic is then evaluated by a permutation test. For notational simplicity, we drop the subscript of the test statistic indicating the type of the thresholding method. Let us denote the test statistic based on the l^{th} permuted sample by T_l^*. Then, the empirical p-value for the permutation test can be computed by

$$\frac{1}{K+1}\left(1 + \sum_{l=1}^{K} I\left(T_l^* \geq T\right)\right),$$

where K is the total number of permutation. We fixed $K = 1{,}000$ for both simulation study and real data analysis.

3. Simulation studies

In simulation, the powers of four versions of CTGSA are compared with each other when there are different proportions of DC genes, different inter-correlation coefficients between two genes, and

Table 1: Type I error rates of four thresholding methods at 5% significance level

Method	$n = 20$			$n = 40$		
	$p = 20$	$p = 50$	$p = 100$	$p = 20$	$p = 50$	$p = 100$
Hard	0.0452	0.0491	0.0543	0.0491	0.0476	0.0505
Lasso	0.0504	0.0507	0.0551	0.0481	0.0480	0.0507
SCAD	0.0484	0.0508	0.0510	0.0501	0.0519	0.0488
MCP	0.0510	0.0490	0.0493	0.0493	0.0471	0.0505
Method	$n = 60$			$n = 100$		
	$p = 20$	$p = 50$	$p = 100$	$p = 20$	$p = 50$	$p = 100$
Hard	0.0499	0.0533	0.0470	0.0512	0.0473	0.0494
Lasso	0.0464	0.0551	0.0505	0.0504	0.0480	0.0482
SCAD	0.0516	0.0499	0.0494	0.0477	0.0502	0.0493
MCP	0.0504	0.0503	0.0502	0.0493	0.0517	0.0505

different number of genes within a gene set. Oh *et al.* (2020) have already demonstrated that CT-GSA based on the hard thresholding operator outperforms other mainstream methods such as GSCA (Choi and Kendziorski, 2009), GSNCA (Rahmatallah *et al.*, 2014), and sufficient dimension reduction method (Hsueh and Tsai, 2016). Therefore, we did not consider to compare these methods in our simulation study.

We just followed the simulation settings of Oh *et al.* (2020), which is also based on the simulation studies of Rahmatallah *et al.* (2014) and Hsueh and Tsai (2016). In their simulation, gene expression data was first generated from a multivariate normal distribution,

$$x_i \sim \begin{cases} N\left(0, \Sigma^{(1)}\right), & \text{for condition 1,} \\ N\left(0, \Sigma^{(2)}\right), & \text{for condition 2,} \end{cases}$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ is the p-dimensional vector for $i = 1, \ldots, n$. The sample size was fixed as $n = 20,\ 40,\ 60$ or 100, while the number of genes in a gene set was fixed as $p = 20,\ 50$ or 100. In order to compute type I error rates of the permutation test, we first put an identity matrix into both covariance matrices $\Sigma^{(1)}$ and $\Sigma^{(2)}$. For each combination between n and p, we then calculated the proportion of rejections where the permutation p-value is less than 0.05 among 10,000 simulation replications. Table 1 shows the type I error rates of four thresholding methods. It seems that there is no serious inflation of type I errors in all different n and p.

Next, we conducted the power comparison of four methods, where $\Sigma^{(1)}$ and $\Sigma^{(2)}$ are indeed different with each other. Specifically, $\Sigma^{(1)}$ is still an identity matrix, but

$$\Sigma^{(2)} = \{\sigma_{ij}^{(2)}\} = \begin{cases} 1, & 1 \le i = j \le p, \\ \rho, & 1 \le i \ne j \le gp, \\ 0, & gp + 1 \le i \ne j \le p, \end{cases}$$

where the inter-gene correlation coefficient $\rho = 0.1,\ 0.3,\ 0.5,\ 0.7$ and 0.9, and the proportion of DC genes $g = 0.1,\ 0.2,\ 0.3,\ 0.4$ and 0.5. Note that both ρ and g practically represent the strength of signals for DC genes. Consequently, the detection power of the permutation test is expected to increase as either ρ or g is increasing. Computation of power is the same as that of type I error rates, where the proportion of rejections was calculated for each different combination of n, p, ρ and g. Since we considered too many different simulation settings, including 4 different sample size n, 3 different number of genes p, 5 different inter-gene correlation coefficients ρ and 5 different proportions of DC genes g for each thresholding method, we reduced down the number of simulation replications to 100

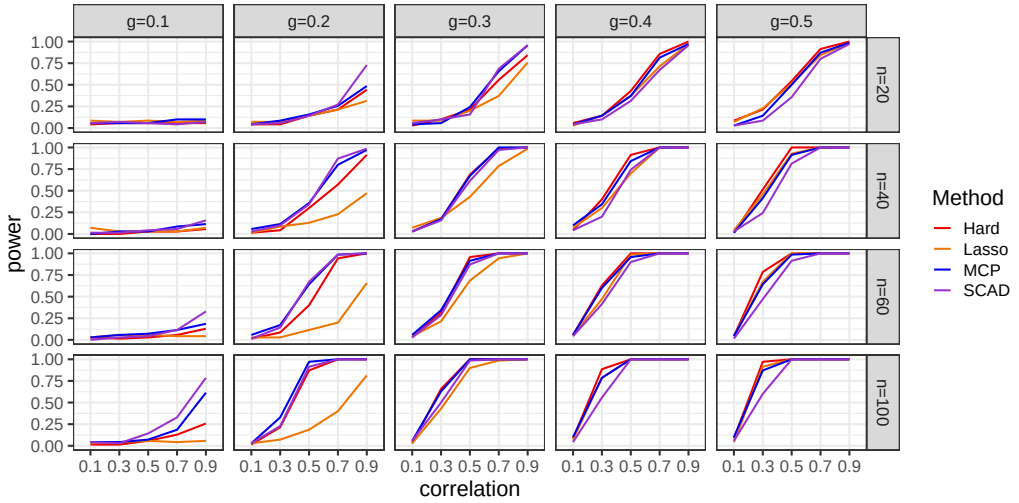

Figure 2: *In a gene set with $p = 20$, power curves of 4 thresholding methods are displayed along with 5 different inter-gene correlation coefficients, 4 different sample sizes (n), and 5 different proportions of DC genes (g).*

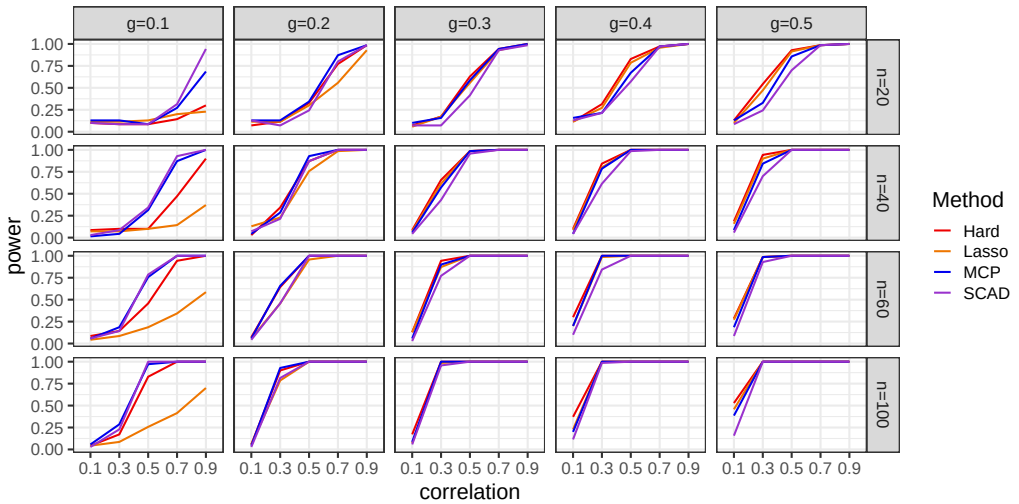

Figure 3: *In a gene set with $p = 50$, power curves of 4 thresholding methods are displayed along with 5 different inter-gene correlation coefficients, 4 different sample sizes (n), and 5 different proportions of DC genes (g).*

for power computation. The power curves of four thresholding methods are shown in Figures 2, 3 and 4 when the number of genes within a gene set is $p = 20,\ 50$ and 100, respectively.

In Figures 2 and 3, it seems that the powers of four thresholding methods are quite different with each other, specially when either ρ or g is relatively small. In contrast, the powers of four thresholding methods are almost identical except $g = 0.1$ in Figure 4. So, if we focus on the case of very sparse DC genes in a gene set, both SCAD and MCP outperforms hard and lasso in all simulation settings with $g = 0.1$. It is noticeable that the power of lasso is not even comparable with the powers of the other

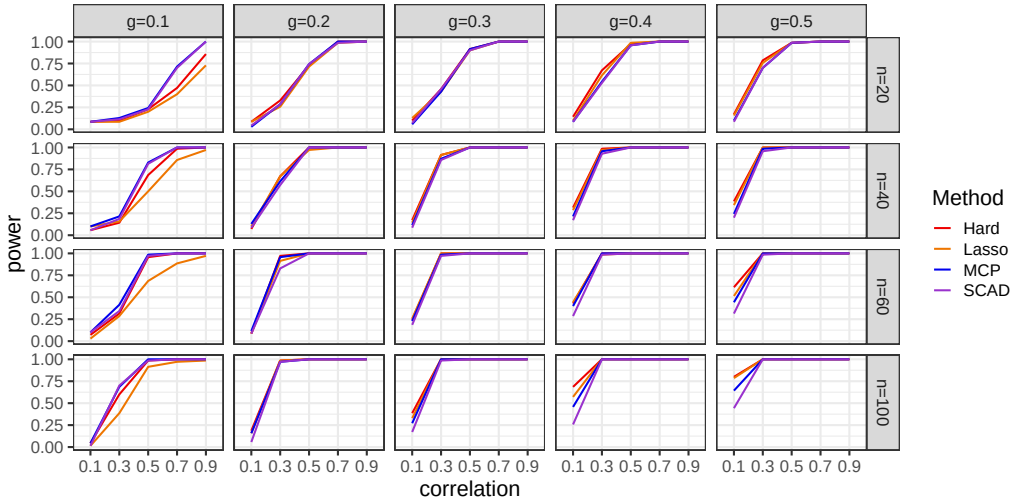

Figure 4: *In a gene set with $p = 100$, power curves of 4 thresholding methods are displayed along with 5 different inter-gene correlation coefficients, 4 different sample sizes (n), and 5 different proportions of DC genes (g).*

Table 2: Power of four thresholding methods when the number of differentially co-expressed genes is 10

Method	$\rho = 0.7,\ n = 20$			$\rho = 0.3,\ n = 40$		
	$p = 20$ $g = 0.5$	$p = 50$ $g = 0.2$	$p = 100$ $g = 0.1$	$p = 20$ $g = 0.5$	$p = 50$ $g = 0.2$	$p = 100$ $g = 0.1$
Hard	0.91	0.77	0.47	0.51	0.34	0.14
Lasso	0.84	0.56	0.40	0.46	0.23	0.16
SCAD	0.80	0.80	0.70	0.24	0.21	0.19
MCP	0.87	0.87	0.71	0.41	0.29	0.21

thresholding methods in this setting. Hard thresholding is slightly better than both SCAD and MCP when $g = 0.4$ or $g = 0.5$. In comparison between SCAD and MCP, SCAD seems to be slightly better than MCP when $g = 0.1$ or 0.2 and $p = 20$, while MCP is better than SCAD for a relatively large value of g. Particularly. SCAD has the worst performance for $g = 0.3,\ 0.4$ and 0.5, $n = 20,\ 40,\ 60$ and 100, and $p = 50$. Considering all different simulation settings, we can select MCP thresodling as the most recommended thresholdimg method to maximize the power of CTGSA.

Next, we additionally explored the change of CTGSA power as the number of genes is increasing while the number of DC genes is fixed. We considered two different scenarios where the first scenario is relatively large inter-gene correlation with the small sample size ($\rho = 0.7, n = 20$) and the second is small inter-gene correlation with the moderate sample size ($\rho = 0.3, n = 40$). For each scenario, we investigated three different cases such as ($p = 20, g = 0.5$), ($p = 50, g = 0.2$) and ($p = 100, g = 0.1$). Note that the number of DC genes is the same as 10 for all cases, although the number of genes in a gene set is different from each other. Table 2 summarizes the power of four thresholding methods for two scenarios including three cases.

In Table 2, it is clear that the power of all thresholding methods is decreased as the number of genes is increasing, which is an expected result due to a high-dimensional problem, i.e., $p > n$. When we compare two scenarios, the power of the first scenario is almost two times greater than that of the second scenario. This indicates that CTGSA is likely to detect DC genes that has relatively large inter-gene correlation coefficient, even if the sample size is not enough. However, if the inter-gene

Table 3: The number of (uniquely) identified gene sets by each thresholding method that have differentially co-expressed genes for 3 different microarray gene expression data sets at a significance level of 0.05

Method	p53 mutant	Epithelium breast cancer	Stroma breast cancer
Hard	112 (29)	64 (28)	54 (20)
Lasso	106 (57)	65 (49)	54 (32)
SCAD	113 (42)	46 (13)	67 (16)
MCP	112 (15)	52 (4)	67 (10)
Total	2442	2190	2190

correlation coefficient is relatively small, CTGSA can fail to detect DC genes even though the sample size is increased. In comparison of four thresholding methods, MCP thresholding has the greatest power when the number of genes is large, while hard thresholding shows the best performance when the number of genes is small. Similarly, SCAD thresholding has the lowest power when the number of genes is small, but it has higher power than both hard and lasso thresholding when the number of genes is large.

4. Real data analysis

For further comparison of four thresholding methods, we applied them to microarray gene expression data such as the p53 mutant and GSE10797 (epithelium and stroma breast cancer) data. The p53 mutant data has been already studied by many researchers (Subramanian *et al.*, 2005; Dinu *et al.*, 2007; Hsueh and Tsai, 2016; Oh *et al.*, 2020). In the dataset, 2,442 genes sets contain 10,100 gene expression profiles for 50 samples, which have 33 cases and 17 controls. 894 gene sets consist of more than 50 genes, so these gene sets have a high-dimensional problem, i.e., $p > n$. Epithelium and stroma breast cancer data was extracted from National Center for Biotechnology Information Gene Expression Omnibus (NCBI GEO). The data set has a total of 2,190 genetic pathways for 22,277 genes, while 33 samples consist of 28 epithelial samples, 5 normal epithelial samples, 28 stromal samples of breast cancer, and 5 normal stromal samples. Also, 900 genetic pathways have more than 33 genes. That is to say, 41.1% of gene sets also has a high-dimensional problem.

For each of gene expression data sets, CTGSA with four thresolding methods identified statistically significant gene sets based on their permutation tests. Specifically, hard, lasso, SCAD and MCP thresholding methods in the p53 mutant data set identified 112, 106, 113 and 112 significant gene sets among 2,442 sets, respectively. For epithelium breast cancer data set, they detected 64, 65, 46 and 52 significant pathways among 2,190 pathways, respectively. For stroma breast cancer data set, four thresholding methods identified 54, 54, 67 and 67 significant pathways among 2,190 pathways, respectively. These results are summarized in Table 3.

We also investigated the number of gene sets that each thresholding method uniquely identified, and they are also included in Table 3. We found that relatively many number of gene sets is not overlapped by four thresholding methods even if the same CTGSA method was applied to the same data. For example, lasso thresholding method uniquely identified more than half of gene sets, which are 57 gene sets among 106 for the p53 mutant data, 49 pathways among 65 for the epithelium breast cancer data, and 32 pathways among 54 for the stroma breast cancer data. It means that the other three thresholding methods failed to detect these gene sets. The number of gene sets uniquely identified by MCP thresholding method is the smallest for all of three data sets. It might seems that MCP thresholding method is relatively stable, compared with other three thresholding methods. However, there is no guarantee that the gene sets MCP thresholding uniquely identified contain truly

differentially co-expressed genes. In other words, they can be either false positives or true positives. In this analysis, we can see that the CTGSA method can produce very different results, depending on which covariance thresholding method is used.

For the power comparison of four thresholding methods, we investigated the permutation p-values of gene sets that all of four methods commonly identified. There are a total of 30 gene sets that the permutation p-values of four methods are all less than 0.05 in three data sets. Hard thresholding has the smallest p-value for 5 gene sets, lasso for 6 gene sets, SCAD for 9 gene sets and MCP for 10 gene sets. We found that both SCAD and MCP methods have relatively small p-values for gene sets which are commonly identified by four methods. Consequently, we can conclude that MCP thresholding has the greatest power in this analysis, followed by SCAD thresholding.

5. Conclusion

In this article, we compared four different covariance thresholding methods such as hard, lasso, SCAD and MCP thresholding when each thresholding method is applied to the test statistic of CTGSA. For detection of DC genes in gene set analysis, CTGSA with the hard thresholding is known to have the greatest power, compared with other mainstream methods such as GSCA (Choi and Kendziorski, 2009), GSNCA (Rahmatallah *et al.*, 2014), and sufficient dimension reduction method (Hsueh and Tsai, 2016). However, in our extensive simulation studies we found that CTGSA using SCAD and MCP thresholding methods can significantly improve the detection power especially when the number of gene is large or the proportion of DC genes is relatively small. Although hard thresholding is better than both SCAD and MCP thresholding when the proportion of weakly correlated DC genes is relatively large, we recommend to use MCP thresholding in order to maximize the power of CTGSA if we have to pick one among four thresholding methods. But, hard thresholding method can be a good alternative particularly when DC genes are weakly correlated with each other. In analysis of real microarray expression data, we also found that CTGSA can identify very different gene sets, depending on the types of the covariance thresholding method.

In recent studies on genetics, statistical methods for RNA-seq data have been developed to identify DE or DC genes since next-generation sequencing technology has facilitated to generate sequencing data rather than microarray data. RNA-seq data can estimate gene expression levels from many sequenced reads, so pairwise correlations of gene expression can be computed. Since it can also quantify the expression levels of the non-coding RNAs, RNA-seq data has generally a higher dimension than microarray data does. CTGSA based on a sparse covariance matrix can take an advantage of the computation of high-dimensional data. Since significance of test statistics for gene set analysis is mostly evaluated by permutation, and there exist many gene sets that contain a large number of genes, computational feasibility and efficiency as well as statistical power is a critical issue for gene set analysis.

References

Bickel PJ and Levina E (2008). Covariance regularization by thresholding, *Annals of Statistics*, **36**, 2577–2604.

Choi Y and Kendziorski C (2009). Statistical methods for gene set co-expression analysis, *Bioinformatics*, **25**, 2780–2786.

Dinu I, Potter JD, Mueller T, *et al.* (2007). Improving gene set analysis of microarray data by SAM-GS, *BMC Bioinformatics*, **8**, 242.

Fan J and Li R (2001). Variable selection via nonconcave penalized likelihood and its oracle proper-

ties, *Journal of the American Statistical Association*, **96**, 1348–1360.

Fan J, Feng Y, and Wu Y (2009). Network exploration via the adaptive Lasso and SCAD penalties, *Annals of Applied Statistics*, **3**, 521–541.

Goeman JJ and Bühlmann P (2007). Analyzing gene expression data in terms of gene sets: methodological issues, *Bioinformatics*, **23**, 980–987.

Hsueh H and Tsai C (2016). Gene set analysis using sufficient dimension reduction, *BMC Bioinformatics*, **17**, 74.

Oh M, Kim K, and Sun H (2020). Covariance thresholding to detect differentially co-expressed genes from microarray gene expression data, *Journal of Bioinformatics and Computational Biology*, **18**, 2050002.

Rahmatallah Y, Emmert-Streib F, and Glazko G (2014). Gene Sets Net Correlations Analysis (GSNCA): a multivariate differential coexpression test for gene sets, *Bioinformatics*, **30**, 360–368.

Rothman AJ, Levina E, and Zhu J (2009). Generalized thresholding of large covariance matrix, *Journal of the American Statistical Association*, **104**, 177–186.

Subramanian A, Tamayo P, Mootha VK, *et al.* (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles, *National Academy of Sciences of the United States of America*, **102**, 15545–15550.

Wu D, Lim E, Vaillant F, Asselin-Labat M-L, Visvader JE, Smyth GK (2010). ROAST: rotation gene set tests for complex microarray experiments, *Bioinformatics*, **26**, 2176–2182.

Zhang C (2010). Nearly unbaised variable selection under minmax concave penalty, *Annals of Statistics*, **38**, 894–942.

Zou H and Li R (2008). One-step sparse estimates in nonconcave penalized likelihood models, *Annals of Statistics*, **36**, 1509–1533.

Received March 22, 2022; Revised May 4, 2022; Accepted May 9, 2022