Research Article

Statistics in Medicine

Received 13 July 2013, Accepted 28 February 2014 Published online 28 March 2014 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/sim.6157

Covariance adjustment for batch effect in gene expression data

Jung Ae Lee,[a] Kevin K. Dobbin[b] and Jeongyoun Ahn[c*†]

Batch bias has been found in many microarray gene expression studies that involve multiple batches of samples. A serious batch effect can alter not only the distribution of individual genes but also the inter-gene relationships. Even though some efforts have been made to remove such bias, there has been relatively less development on a multivariate approach, mainly because of the analytical difficulty due to the high-dimensional nature of gene expression data. We propose a multivariate batch adjustment method that effectively eliminates inter-gene batch effects. The proposed method utilizes high-dimensional sparse covariance estimation based on a factor model and a hard thresholding. Another important aspect of the proposed method is that if it is known that one of the batches is produced in a superior condition, the other batches can be adjusted so that they resemble the target batch. We study high-dimensional asymptotic properties of the proposed estimator and compare the performance of the proposed method with some popular existing methods with simulated data and gene expression data sets. Copyright © 2014 John Wiley & Sons, Ltd.

Keywords: batch effect; factor model; gene expression; high-dimensional covariance estimation

1. Introduction

In microarray experiments, the sample size of a data set is limited because of practical complications. Sometimes, data are collected at several independent institutes. Analyzing these data sets individually would limit the scope of the study because of a relatively small sample size compared to tens of thousands of genes. Combining several data sets is often considered a remedy to this problem, as it can artificially enlarge the sample size for the sake of a potential benefit of increased statistical power. For example, in [1–4], researchers have attempted merging data sets or an inter-laboratory study in order to discover reliable biological markers and establish more robust prognostic models.

However, it has been observed that there exist significant biases among different batches in the merged data, which offsets the advantages of the increased sample size. This is because data sets collected at different sites or times may have a systematic bias, called ‘batch effect,’ that possibly comes from subtle inconsistencies in sample maintenance, RNA extraction techniques, hybridization protocols, or many other experimental conditions [5]. Ignoring these batch effects can lead to erroneous conclusions (see, e.g., [6]). In order to make the combining serve its purpose, there is a pressing need to find a batch effect removal method that can create a merged data set free of any batch bias.

Batch effects also have been found in microarray reproducibility studies. Dobbin *et al.* [7] found laboratory batch effects when the same samples were assayed in technical replicates at four different laboratories using the identical set of detailed protocols and equipment. Using different levels of replication, they isolated sources of variability and found that the largest lab-to-lab variation was attributable to the lowest level of chip processing—that is, the RNA reverse-transcription, labeling, hybridization, and scanning. In a subsequent study, Irizarry *et al.* [8] found similar effects under less controlled conditions. After that, the MAQC I study [9] confirmed these earlier results, finding that making laboratory protocols uniform could greatly reduce, but not eliminate, batch effects. Recently,

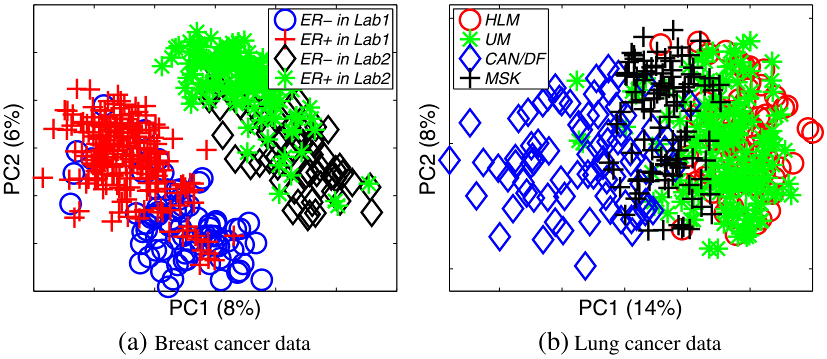
[a] *Division of Public Health Sciences, Washington University in St. Louis, St. Louis, MO 63110, U.S.A.*
[b] *Department of Epidemiology and Biostatistics, University of Georgia, Athens, GA 30605, U.S.A.*
[c] *Department of Statistics, University of Georgia, Athens, GA 30602, U.S.A.*
**Correspondence to: Jeongyoun Ahn, Department of Statistics, University of Georgia, Athens, GA 30602, U.S.A.*
[†] *E-mail: jyahn@uga.edu*

Parker and Leek [10] found that batch effect associated with the prediction outcome can cause a serious bias in prediction studies.

Batch effects exist not only in microarray but also in other newer technologies. Leek *et al*. [11] found significant batch effects in mass spectrometry data, copy number abnormality data, methylation array data, and DNA sequencing data. Even though most of the existing methods, including the proposed work in this paper, have been developed for microarray, finding general methods to correct batch effects continues to be a critical endeavor that may have substantial impact on the future success of these technologies.

In the following example, we examine two breast cancer microarray batches collected at different laboratories; the sample sizes are 286 and 198, respectively. The detailed description of these data sets can be found in Section 5. With a goal of predicting the estrogen receptor (ER) status, we want to create a combined data set in order to increase the statistical power. Figure 1(a) displays projections of the data onto the first two principal component (PC) directions obtained from the whole data set. Inside the parentheses is the proportion of variation explained by a PC. We can see that the separation between the batches is more apparent than the separation between ER+ and ER− groups, suggesting that the batch effect dominates the biological signal. Clearly, there is a need to fix this problem prior to any statistical analysis with the combined data. Another example of batch effect can be found in Figure 1(b), where four lung cancer microarray batches from different laboratories are shown. Shedden *et al*. [12] used these data sets for a survival prediction study. The detailed description of the data set can also be found in Section 5. In the figure, four different symbols represent their laboratory memberships. Visible gaps among the batches are noted in the direction of the first PC.

There exist several popular batch bias adjustment methods. The simplest method is to make each batch have the same centroid. Despite the simplicity of its idea, the mean-centering method seems to be effective in reducing batch biases, but by no means in eliminating them. Sample standardization makes each gene within a batch have a unit variance as well as zero mean. Another popular approach is to utilize linear discrimination methods while treating the batch membership as target labels for classification. A common choice for a discrimination method is the distance-weighted discrimination (DWD) that was proposed by Marron *et al*. [13] for high-dimensional classification problem. Benito *et al*. [14] proposed a batch adjustment method using DWD, with which they find the optimal separating hyperplane that maximizes the separation between batches, and then moves each batch along the normal direction vector until its mean reaches the hyperplane. Johnson *et al*. [15] proposed the empirical Bayes (EB) method, which sets up a mixed-effect model for each gene, with the batch bias as a random block effect and biological signals as fixed treatment effects. The surrogate variable analysis (SVA) by Leek and Storey [16] identifies the effect of the hidden factors that may be the sources of data heterogeneity, thus useful when there are unknown biases other than batch membership. Shabalin *et al*. [17] proposed the cross-platform normalization (XPN), which was originally designed for merging data sets from different microarray platforms. The XPN procedure is based on a block-linear model under the assumption that a microarray heatmap can be segmented into several blocks according to homogeneous genes and



(a) Breast cancer data (b) Lung cancer data

Figure 1. Illustration of a batch effect in microarray data. In (a), breast cancer data sets from two different laboratories are projected on the first two PC directions. It is clear that the batch effect dominates the biological (ER+, ER−) signal. In (b), the projections of four lung cancer data sets are shown. We can see strong batch bias especially in the first PC direction.

samples. Gagnon-Bartsch and Speed [18] proposed the RUV-2 method that uses control genes to remove unwanted variation. We note that DWD, EB, and XPN are designed for direct adjustment when the source of batch bias is known, while SVA and RUV-2 are suitable when the source is unknown. We include DWD, EB, and XPN in the simulation study in Section 4 and additionally include SVA and RUV-2 in the real data examples in Section 5.

Some existing methods are essentially gene-wise in the sense that the batch effect is assumed to affect each gene independently, and thus, they ignore the possibly altered inter-gene relationships. In particular, both the mean-centering and discrimination-based approaches suggest that the batch bias only exists in the shift of the mean of each gene. One might argue that the discrimination-based approaches such as DWD are not gene-wise because the inter-gene relationship is taken into account when determining the separating hyperplane. However, as the resulting adjustment is parallel to the coordinate axes, it does not alter variance–covariance structures of genes.

It has been noted that batch effects are often multivariate, that is, the variance–covariance structure is altered due to the batch effect [11]. In this paper, we propose a multivariate batch adjustment method that can correct the variance–covariance of the batches in order to create a combined data set with a better homogeneity. The rest of the paper is organized as follows. In Section 2, we introduce the probabilistic framework under which we derive an ideal batch effect removal scenario and propose a new method based on a high-dimensional covariance estimation. In Section 3, we provide some theoretical understanding of the estimated covariance adjustment factor. The proposed method is compared with some existing methods with simulated examples and real gene expression data sets in Sections 4 and 5, respectively. The paper ends with discussion in Section 6.

2. Batch effect removal by covariance adjustment

2.1. *Ideal batch adjustment*

Let us assume an imaginary situation where batch effect does not exist or that all current and future data are from the same batch. Define the (unobservable) random vector of p gene expression values by $\mathbf{Y}^* = (Y_1^*, \dots, Y_p^*)^{\mathrm{T}}$, which is assumed to be from a multivariate distribution with mean vector $\mathrm{E}(\mathbf{Y}^*) = \boldsymbol{\mu}^*$ and nonsingular covariance matrix $\mathrm{Var}(\mathbf{Y}^*) = \boldsymbol{\Sigma}^*$. We also assume that the affine transformed vector $\mathbf{Z} = \boldsymbol{\Sigma}^{*-1/2}(\mathbf{Y}^* - \boldsymbol{\mu}^*)$ has mean $\mathrm{E}(\mathbf{Z}) = \mathbf{0}$ and variance $\mathrm{Var}(\mathbf{Z}) = \mathbf{I}$.

In a more realistic scenario, we would observe array vectors $\mathbf{Y}_{ij} = (Y_{ij1}, \dots, Y_{ijp})^{\mathrm{T}}$ from batch i with sample size n_i, $i = 1, \dots, K$, $j = 1, \dots, n_i$. We assume that each sample array from the ith batch follows a multivariate distribution with mean vector $\mathrm{E}(\mathbf{Y}_{ij}) = \boldsymbol{\mu}_i$ and nonsingular covariance matrix $\mathrm{Var}(\mathbf{Y}_{ij}) = \boldsymbol{\Sigma}_i$. Then we can express

$$\begin{aligned}\mathbf{Y}_{ij} &= \boldsymbol{\Sigma}_i^{1/2}\mathbf{Z}_j + \boldsymbol{\mu}_i \\ &= \boldsymbol{\Sigma}_i^{1/2}\left(\boldsymbol{\Sigma}^{*-1/2}\left(\mathbf{Y}_j^* - \boldsymbol{\mu}^*\right)\right) + \boldsymbol{\mu}_i \\ &= f_i(\mathbf{Y}_j^*),\end{aligned}$$

where a function f_i represents the ith batch effect and $\mathbf{Y}_j^*$ is a realization of $\mathbf{Y}^*$. Thus, the function f_i is an affine transformation of the unobservable $\mathbf{Y}^*$, that is, $f_i(\mathbf{Y}^*) = \mathbf{A}_i\mathbf{Y}^* + \mathbf{b}_i$, where $\mathbf{A}_i = \boldsymbol{\Sigma}_i^{1/2}\boldsymbol{\Sigma}^{*-1/2}$ and $\mathbf{b}_i = -\boldsymbol{\Sigma}_i^{1/2}\boldsymbol{\Sigma}^{*-1/2}\boldsymbol{\mu}^* + \boldsymbol{\mu}_i$. Then the batch effect can be adjusted by applying the inverse function f_i^{-1} such that $f_i^{-1}(\mathbf{Y}) = \mathbf{A}_i^{-1}(\mathbf{Y} - \mathbf{b}_i)$. Note that in this way we can adjust for the batch effects that possibly have distorted inter-gene relationship as well as mean and variance for individual genes.

2.2. *Covariance estimation based on factor model*

Now, the remaining important question is how to estimate $\mathbf{A}_i^{-1} = \boldsymbol{\Sigma}^{*1/2}\boldsymbol{\Sigma}_i^{-1/2}$, which is a function of the two unknown population covariance matrices. Estimation of high-dimensional covariance matrix has gained much importance over the recent years, as the classical sample covariance matrix is not applicable when the dimension is large relative to the sample size. Note that in the ideal batch adjustment suggested in the previous section, the multiplicative factor matrix $\mathbf{A}_i^{-1}$ has two components. The first is related to the covariance of the 'true' batch, that is, the data without a batch bias. The second component is the inverse of the covariance of an observed batch. In the literature, estimating inverse covariance matrix, called the precision matrix, is often treated as a different problem from estimating a covariance matrix.

For this reason, we need a unified framework under which both covariance and precision matrices are estimated. In this paper, we employ the factor model proposed by Fan *et al.* [19].

One advantage of using the factor model is that gene categorization can be taken into account, by assuming that the behaviors of observed variables are determined by a much smaller number of 'factor variables.' A common belief in gene expression analysis is that there exist groups of genes within which the genes act together. There are a few different approaches for defining factors for genes. The most popular approach is to use the Gene Ontology (GO), grouping genes that belong in the same pathway [20]. For example, the so-called gene set enrichment analysis [21] is based on this approach. A possible drawback is that at the current moment, the pathway information is not complete and may be inaccurate [22]. One can also use evolutionary information and group the genes that share a similar DNA sequence [23]. Another approach is to use the data set at hand to create clusters of the genes, utilizing clustering algorithms such as K-means. In this paper, we used K-means clustering for the simulated data in Section 4 and used the pathway information for real data examples in Section 5.

The factor model by Fan *et al.* [19] is a multiple regression model on each gene expression level Y_i, $i = 1, \dots, p$,

$$Y_i = \sum_{\ell=1}^{q} \beta_{i\ell} X_\ell + \varepsilon_i, \tag{1}$$

where $X_1, \dots, X_q$ are q known factors and $\beta_{i\ell}$, $\ell = 1, \dots, q$, are regression coefficients. Note X_ℓ represents the mean of expression values that belong to the ℓth group of genes. Letting $\mathbf{y} = (Y_1, \dots, Y_p)^{\mathrm{T}}$, $\mathbf{x} = (X_1, \dots, X_q)^{\mathrm{T}}$, and $\mathbf{e} = (\varepsilon_1, \dots, \varepsilon_p)^{\mathrm{T}}$, we can rewrite (1) in a matrix form

$$\mathbf{y} = \mathbf{B}\mathbf{x} + \mathbf{e}, \tag{2}$$

where $\mathbf{B} = \{\beta_{i\ell}\}$ is a $p \times q$ regression coefficient matrix. We also make the common assumption that $\mathrm{E}(\mathbf{e}|\mathbf{x}) = \mathbf{0}$, and $\mathrm{cov}(\mathbf{e}|\mathbf{x})$ is diagonal.

Using a matrix form $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]_{p\times n}$ and $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]_{q\times n}$, the least squares estimator of $\mathbf{B}$ in (2) is given by $\hat{\mathbf{B}} = \mathbf{Y}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}})^{-1}$. Let $\mathbf{\Sigma} = \mathrm{cov}(\mathbf{y})$ and $\mathbf{\Sigma}_0 = \mathrm{cov}(\mathbf{e}|\mathbf{x})$. The estimation of the covariance matrix of $\mathbf{y}$ from n samples can be derived from model (2):

$$\hat{\mathbf{\Sigma}} = \hat{\mathbf{B}}\widehat{\mathrm{cov}}(\mathbf{x})\hat{\mathbf{B}}^{\mathrm{T}} + \hat{\mathbf{\Sigma}}_0, \tag{3}$$

where $\widehat{\mathrm{cov}}(\mathbf{x})$ is a $q \times q$ nonsingular sample covariance matrix from the given factor matrix $\mathbf{X}(q < n)$. Last, $\hat{\mathbf{\Sigma}}_0$ is obtained by $\mathrm{diag}(n^{-1}\hat{\mathbf{E}}\hat{\mathbf{E}}^{\mathrm{T}})$, where $\hat{\mathbf{E}}$ is the residual matrix, $\mathbf{Y} - \hat{\mathbf{B}}\mathbf{X}$. The estimator $\hat{\mathbf{\Sigma}}$ in (3) is always invertible even when the dimension p exceeds n. Fan *et al.* [19] showed that $\hat{\mathbf{\Sigma}}^{-1}$ is asymptotically better than the inverse of the regular sample covariance matrix as $(p, q, n) \to \infty$.

For the purpose of batch adjustment, we propose to use the estimator in (3) for a covariance estimator for each batch $\hat{\mathbf{\Sigma}}_i$, $i = 1, \dots, K$. Recall that another matrix to estimate, $\mathbf{\Sigma}^*$, is the covariance of the 'true' batch, under the no-batch-effect assumption. When one of the observed batches, say the i^*th batch, can be regarded as close to the ideal batch, one can replace $\hat{\mathbf{\Sigma}}^*$ by $\hat{\mathbf{\Sigma}}_{i^*}$. This batch may have been produced under the best conditions. More specifically, the facility may have the most experience with the technology, so that the batch shows the best quality metrics or the best reproducibility on technical replicates, such as Strategene Universal Human Reference samples. For these cases, the proposed adjustment method transforms the data so that all the other batches mimic the ideal batch.

In general, when it is difficult to pinpoint a better batch, we can pool the covariance estimates for each batch as the following:

$$\hat{\mathbf{\Sigma}}^* = \sum_{i=1}^{K} (n_i - 1)\hat{\mathbf{\Sigma}}_i / (N - K),$$

where $N = \sum_{i=1}^{K} n_i$. Note that this assumes that $\mathbf{\Sigma}^*$ is reasonably close to $\sum_{i=1}^{K}(n_i - 1)\mathbf{\Sigma}_i/(N - K)$.

2.3. Sparse estimation

In practice, the suggested covariance adjustment estimator $\hat{\mathbf{A}}_i^{-1}$ in the previous section can induce a substantial amount of uncertainty because the estimation involves a multiplication of high-dimensional covariance estimates, one of which is even inverted. In high-dimensional data analysis, it is a common assumption that not all variables are signal variables. Thus, some degree of sparsity is usually imposed in the estimation process to achieve a more stable estimator, especially when high-dimensional covariance matrix is being estimated. See, for example, [24, 25].

In this work, we use a hard thresholding, that is, the entries that are smaller than some tuning parameter, say δ, in the estimated matrix are forced to be zero. The (j,k)th element of the sparse estimate $\hat{\mathbf{A}}_i^{-1}(\delta)$ is given by $a_{jk}(\delta) = a_{jk} I(|a_{jk}| > \delta),\ \ j \neq k$, where a_{jk} is the (j,k)th element of $\mathbf{A}_i^{-1}$ and $I(\cdot)$ is the indicator function. In order to choose δ, we consider similarity between covariances of the adjusted batches. Let $\mathbf{S}_i$ and $\mathbf{S}_i^{\delta}$ be the sample covariance matrices of the ith batch before and after the adjustment, respectively. Note that $\mathbf{S}_i^{\delta} = \hat{\mathbf{A}}_i^{-1}(\delta)\mathbf{S}_i(\hat{\mathbf{A}}_i^{-1}(\delta))^{\mathrm{T}}$. We propose to choose δ that makes $\mathbf{S}_i^{\delta}$ as similar to each other as possible. In particular, we consider the equal covariance test statistic for high-dimensional data proposed by Srivastava and Yanagihara [26]. Their test statistic Q_K^2 for comparing ith and jth batches is based on the difference between $\mathrm{tr}(\mathbf{S}_i^{\delta})^2/[\mathrm{tr}(\mathbf{S}_i^{\delta})]^2$ and $\mathrm{tr}(\mathbf{S}_j^{\delta})^2/[\mathrm{tr}(\mathbf{S}_j^{\delta})]^2$. A smaller value of the test statistic indicates greater similarity of two population covariance matrices. This test is also applicable for comparison of more than two batches. See [26] for more details.

Figure 2 displays the test statistic Q_K^2 for both the breast cancer data and the lung cancer data in Section 5 for a range of δ. It can be seen that $\hat{\delta} = 0.02$ and $\hat{\delta} = 0.03$ are the best choices for respective data sets. In Section 5, we separately choose the level of sparsity for each batch. For computational efficiency, the search is performed around the common $\hat{\delta}$ found in Figure 2. As a result, $\hat{\boldsymbol{\delta}} = (0.01, 0.03, 0.02)$ are used for the breast cancer data, and $\hat{\boldsymbol{\delta}} = (0.03, 0.02, 0.06, 0.05)$ for the lung cancer data.

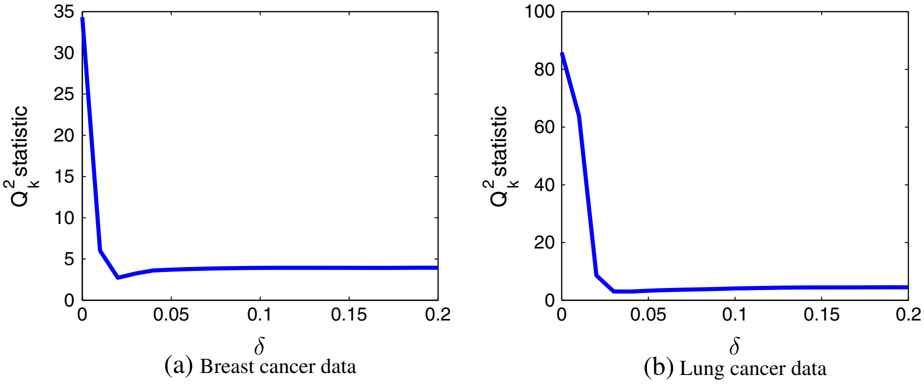
3. Theoretical properties

In this section, we study some theoretical properties of $\hat{\mathbf{A}}_i^{-1} = \hat{\boldsymbol{\Sigma}}^{*1/2}\hat{\boldsymbol{\Sigma}}_i^{-1/2}$, $i = 1, \ldots, K$, with growing dimensionality (p), number of factors (q), and sample size (n_i). The rate of convergence is studied in terms of Frobenius norm. For a matrix $\mathbf{C}$, its Frobenius norm is given by $\|\mathbf{C}\| = \{\mathrm{tr}(\mathbf{C}\mathbf{C}^{\mathrm{T}})\}^{1/2}$. For the sake of simplicity, we impose the same set of assumptions for each batch so that we can omit the subscript i when discussing the estimation of $\hat{\boldsymbol{\Sigma}}_i$. Also, we assume that the sample size n_i is all equal to n for all batches. In the following, we use some basic assumptions in [19]. Let $b_n = \mathrm{E}\|\mathbf{y}\|^2$, $c_n = \max_{1\leqslant \ell \leqslant q} \mathrm{E}\left(X_\ell^4\right)$, and $d_n = \max_{1 \leqslant j \leqslant p} \mathrm{E}\left(\varepsilon_j^4\right)$.

(i) $(\mathbf{y}_1, \mathbf{x}_1), \ldots, (\mathbf{y}_n, \mathbf{x}_n)$ are i.i.d. samples of $(\mathbf{y}, \mathbf{x})$. $\mathrm{E}(\mathbf{e}|\mathbf{x}) = \mathbf{0}$ and $\mathrm{cov}(\mathbf{e}|\mathbf{x}) = \boldsymbol{\Sigma}_0$ is diagonal. Also, the distribution of $\mathbf{x}$ is continuous, and the number of factors q is less than the dimension p.

(ii) $b_n = O(p)$ and the sequences c_n and d_n are bounded. Also, there exists a constant $\sigma_1 > 0$ such that $\lambda_q(\mathrm{cov}(\mathbf{x})) \geqslant \sigma_1$ for all n, where $\lambda_j(\mathbf{C})$ is jth eigenvalue of $\mathbf{C}$.

(iii) There exists a constant $\sigma_2 > 0$ such that $\lambda_p(\boldsymbol{\Sigma}_0) \geqslant \sigma_2$ for all n.



(a) Breast cancer data

(b) Lung cancer data

Figure 2. Change of the test statistic Q_K^2 over $\hat{\delta}$. The lowest value is obtained at $\hat{\delta} = 0.02$ and $\hat{\delta} = 0.03$ in (a) and (b), respectively.

In this paper, we further assume the following:

(iv) Denote the eigenvectors and the corresponding eigenvalues of $\mathbf{\Sigma}$ as $(\mathbf{u}_j, \lambda_j)$, and those of $\hat{\mathbf{\Sigma}}$'s as $(\hat{\mathbf{u}}_j, \hat{\lambda}_j)$, $j = 1, \ldots, p$. The conditional expectation $\mathrm{E}\left(\hat{\mathbf{u}}_j \hat{\mathbf{u}}_j^{\mathrm{T}} \mid \hat{\lambda}_j\right) = \mathbf{u}_j \mathbf{u}_j^{\mathrm{T}}$ for all j, and

$$P\left(\sum_{j=1}^{p} (\hat{\lambda}_j - \lambda_j)^2 \geqslant \sum_{j=1}^{p} \left(\hat{\lambda}_j^{1/2} - \lambda_j^{1/2}\right)^2\right) = 1.$$

Note that the preceding inequality inside the probability notation can be rewritten as

$$\sum_{j=1}^{p} \left(\hat{\lambda}_j(\hat{\lambda}_j - 1) + \lambda_j(\lambda_j - 1) - 2\hat{\lambda}_j^{1/2}\lambda_j^{1/2}\left(\hat{\lambda}_j^{1/2}\lambda_j^{1/2} - 1\right)\right) \geqslant 0,$$

and is easily met in general unless most eigenvalues are less than one. In what follows, we list our theoretical findings. Proofs are given in the Appendix.

Theorem 1 (Convergence rate for the covariance adjustment estimator)
Under assumptions (i)–(iv),

$$\|\hat{\mathbf{\Sigma}}^{*1/2}\hat{\mathbf{\Sigma}}_i^{-1/2} - \mathbf{\Sigma}^{*1/2}\mathbf{\Sigma}_i^{-1/2}\| = o_p\left((p^4 q^5 \log(n)/n)^{1/2}\right).$$

The following corollary shows the rate of convergence of the after-adjustment covariance.

Corollary 1
Under the same conditions as in Theorem 1,

$$\|\widehat{\mathrm{cov}}(\mathbf{Y}_i^*) - \mathbf{\Sigma}^*\| = O_p(n^{-1/2}pq),$$

where $\mathbf{Y}_i^*$ is the adjusted data in the ith batch.

4. Simulation study

In this section, we carry out some simulations to compare the performance of the proposed method with that of existing methods. As the first step, we generate two heterogeneous data sets in both location and covariance. Then, six methods are attempted to adjust two data sets to make them homogeneous. The methods are mean centering (MC), DWD method, standardization (Z-score), EB method, XPN, and the proposed multi-batch covariance adjustment (MBCA) method. We also attempted the targeted adjustment, MBCA(B1) and MBCA(B2), that treats batches 1 and 2 as the target, respectively.

We simulated two batches of size $n_1 = n_2 = 50$ from multivariate normal distributions of dimension $p = 800$. Each element of the mean vector was randomly drawn from Uniform(0, 1.4) for the first batch and from Uniform(−1.4, 0) for the second batch. The endpoints for the uniform distribution are determined such that the discrimination-based method, DWD, can find a reasonable separating hyperplane between the batches. Note that the location of the mean vector does not matter for other methods because they all adjust for the mean. For the covariance for the two batches, we use $\mathbf{\Sigma} = \mathbf{UDU}^{\mathrm{T}}$, where $\mathbf{U}$ is a $p \times p$ orthonormal matrix and $\mathbf{D}$ is a diagonal matrix containing eigenvalues. The eigenvalues are set to $\lambda_k = pk^{-1}/3$ for the first batch and $\lambda = p\exp(-0.042k)/6$ for the second batch, $k = 1, \ldots, p$. The heterogeneity of the simulated data is illustrated in the PC plots in Figure 3, where it can be seen that the two batches are quite different in terms of not only the variance–covariance structures but also in the means. We ran the simulation with 100 repetitions to evaluate the performance of each method. As for the factors for MBCA, we used K-means gene clustering. The choice of K may be an important consideration in general; however, in this particular simulation, $K = 3, 5, 7$ behaved similarly; thus, we present the result when $K = 5$.

First, we use some graphical measures to check the batch homogeneity for one realization of simulated data. Figure 4 shows adjusted data on the first two PC directions. Both MC and DWD adjust only the location of the batches as the first two panels indicate. Other methods adjust the variance as well as the location, which the rest of the panels reflect. In particular, XPN and MBCA appear to mix the batches well. We also compare sample eigenvalues obtained separately in two batches in Figure 5. The proposed

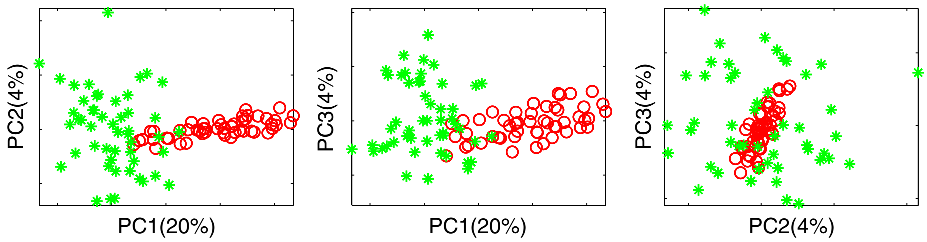


Figure 3. PC projection plot of two simulated batches. Green circles are batch 1, and red stars are batch 2. We can see the two batches are different in both location and shape.

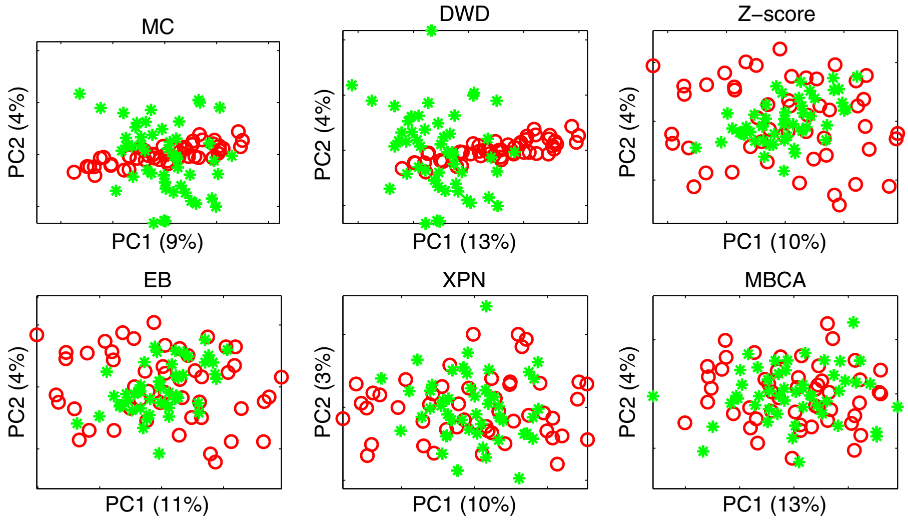


Figure 4. PC plot after various adjustments for simulated data. Red circles are batch 1, and green stars are batch 2.

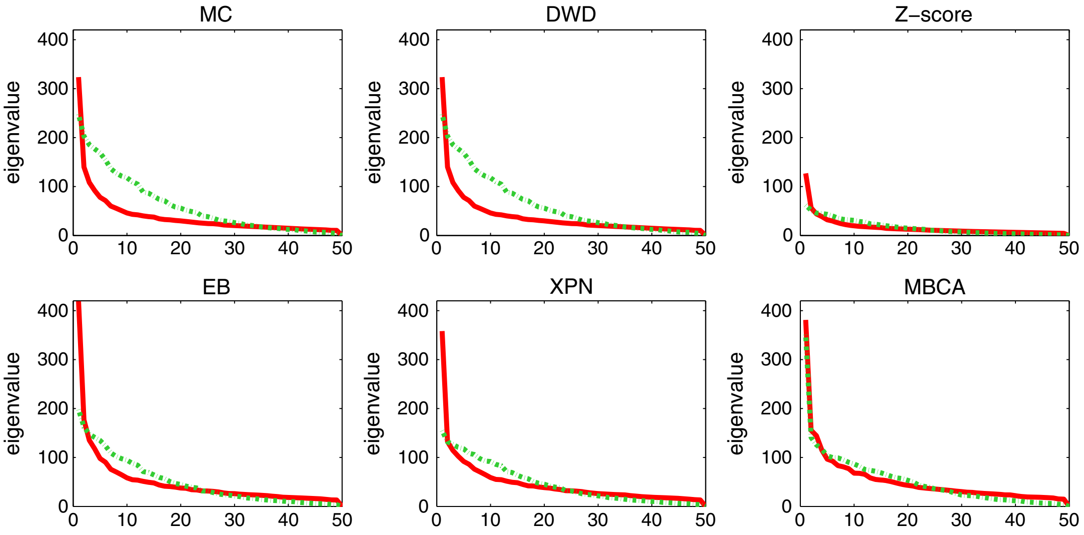


Figure 5. Sample eigenvalues after batch adjustment for simulated data. Solid red lines are for batch 1, and dotted green lines are for batch 2.

MBCA method achieves the best agreement of the two sets of eigenvalues. Note again that both MC and DWD do not alter variance–covariance structure of each batch; thus, both sets of eigenvalues are identical.

The equal covariance test statistic developed by Srivastava and Yanagihara [26] can be useful to measure the homogeneity of covariance among batches. We obtain Q_2^2 test statistic ($H_0 : \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$) discussed in Section 2.3 and corresponding p-values. In Figure 6, we show the box plots of the p-values of the test, from 100 repetitions. Just as the sample eigenvalues results above, the MC and DWD methods have exactly the same p-values as the *Before*. Overall, the box plot of the original MBCA is the highest, while the targeted adjustments (B1) and (B2) are close second and third, respectively. We acknowledge that this criterion favors the proposed method because the MBCA aims to achieve the most similar covariances; however, it should be informative to see how other methods measure up each other in this aspect.

Other than covariance similarity, we also consider some geometric measures. Figure 7 shows box plots of the nearest pairwise distances to the opposite batch [17]. The horizontal line across the figure is the baseline value that is the median nearest distance with ten random splits of the data before adjustment. While all the compared methods show similar box plots, the distances by the MBCA with the target batch 1 are closest to the baseline value.

The next criterion is based on the fact that if the batches are well mixed, the distance among data points within a batch will be similar to the distances between batches. For each repetition, we

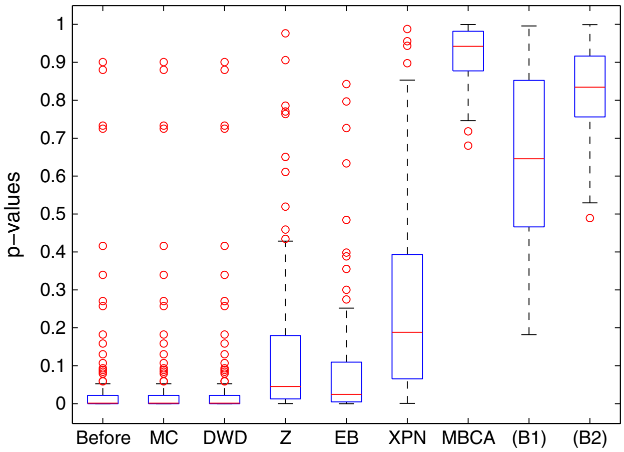


Figure 6. Box plots of p-values from equal covariance test for simulated data.

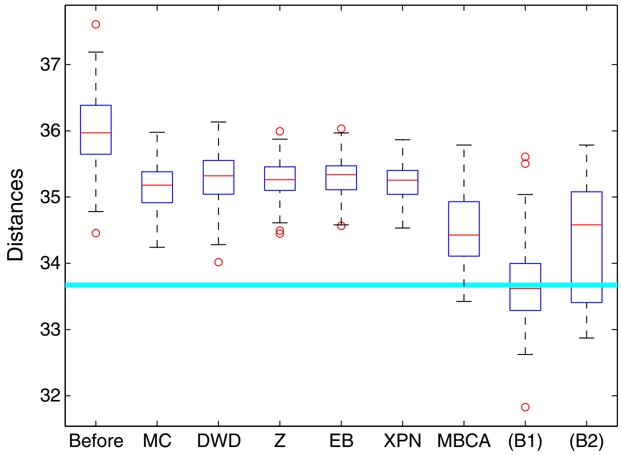


Figure 7. Box plots of average distances of the nearest array in the opposite batch for simulated data. The horizontal line across the plot is the baseline based on the average of random splits within a batch.

calculated Kullback–Leibler divergence between densities of within-batch and between-batch pairwise distances. Figure 8 shows box plots of the log divergence values from 100 repetitions. The last three box plots, the original MBCA and the two targeted versions, are the lowest, indicating that the batches are merged well.

5. Gene expression examples

In this section, we use two cancer microarray gene expression data sets: breast cancer data and lung cancer data. Originally, there were 22,283 genes from which 12,526 genes are extracted by averaging the replicates as identified by the UniGene cluster. Then we select 4364 genes that are assigned by GO; mapping of the GO annotations for genes can be found at http://go.princeton.edu/cgi-bin/GOTermMapper.

The breast cancer data set consists of three batches that were independently collected in three different places in Europe at different times. All three batches were preprocessed by MAS5.0 for Affymetrix U133a GeneChip. The data are available at http://www.ncbi.nlm.nih.gov/projects/geo/ with the GEO accession numbers GSE2034, GSE4922, and GSE7379. As for the biological signal, we have an estrogen status (ER+, ER−) for each subject. The four batches in the lung cancer data set were collected at four different laboratories: Moffitt Cancer Center (HLM), University of Michigan Cancer Center (UM), the Dana-Farber Cancer Institute (CAN/DF), and Memorial Sloan-Kettering Cancer Center (MSK). These four institutes formed a consortium with the U.S. National Cancer Institute to develop and validate gene expression signatures of lung adenocarcinomas. This experiment was designed to characterize the performance of several prognostic models across different subjects and different laboratories. A relevant study has been published by Shedden *et al*. [12]. The CEL files are available at http://caarraydb.nci.nih.gov/caarray/. The data sets for our analysis are preprocessed by the robust multi-array average (RMA) method. As for the biological signal, we use overall survival status (dead and alive) reported at last follow-up time. Table I displays a brief summary of the data.

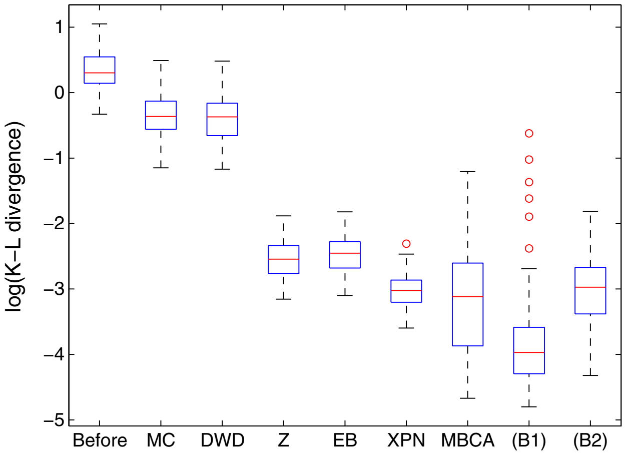


Figure 8. Kullback–Leibler divergence between densities of within-batch and between-batch pairwise distances for simulated data.

Table I. A summary of the two gene expression data sets.

Breast cancer data				Lung cancer data			
Batch	Sample size	ER−	ER+	Batch	Sample size	Live	Die
GSE2034	286	77	209	HLM	79	19	60
GSE4922	245	34	211	UM	178	76	102
GSE7379	198	64	134	CAN/DF	82	47	35
				MSK	104	65	39

In addition to the existing methods in the previous section, we added two more methods, namely SVA and RUV-2, for comparison. We used the SVA R package called for implementing SVA. In implementing RUV-2, one needs to choose control genes that are known to be irrelevant to the biological signal, which are supposed to be determined outside the observed data. In this study, without such information, we used the genes that are least correlated with biological signal as the control genes. As this approach overuses the biological signal, it may have created a bias in the prediction result.

First, we measured the inter-batch homogeneity in terms of their covariance matrices. Note that most methods adjust for the means by default, so comparing means are pointless. We obtain Q_K^2 test statistic ($H_0 : \mathbf{\Sigma}_1 = \cdots = \mathbf{\Sigma}_K$) discussed in Section 2.3 and corresponding p-values and report the averages in Table II. The EB, XPN, and MBCA have higher p-values for both data, which indicates that these methods alter covariances and effectively homogenize the batches. However, it does not necessarily imply that the covariances mutually agree. We tested whether the covariance produced by the three methods is the same or not using the same test procedure and obtained almost zero p-value. For the lung cancer data, it is notable that standardization has a lower p-value (0.0002) than the Before (0.0456). It is notable that SVA and RUV-2 work well for the breast cancer data but not for the lung cancer data.

As the batch bias usually interferes with the biological signal, a successful adjustment can improve separability of the biological classes, making a classifier produce better prediction performance. A PC plot can also be useful to see how separable biological classes become after adjustment. However, one should use caution when using the strengthened biological signal as a criterion for a successful batch effect adjustment. If the adjustment is overly done, it may force the data to be spuriously more separable than the underlying population can allow. Methods that use the biological signal in the adjustment process may be prone to this problem.

In this paper, rather than the performance itself, we use cross-batch prediction. That is, we see if a classification rule based on one batch can be effectively applied to other batches. If the adjustment is reasonable, we expect that prediction performance of a classifier would be similar from batch to batch. As for the classification method, we choose the regularized linear discriminant analysis [27], because of its known superior performance for high-dimensional data such as gene expression.

As each batch can have different proportions of biological signals, we use Matthews correlation coefficient (MCC) in order to measure prediction performance [28]. The MCC is calculated directly from the confusion matrix using the following formula:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively. The MCC value of $+1$ indicates perfect prediction, -1 reverse prediction, and 0 neutral prediction.

For the breast cancer data, we use two batches as training data to determine the discrimination rule, with which the tuning parameter is chosen with fivefold cross-validation. Then we predict the ER status in the left-out batch and report the MCC. The results are shown in Table III. It is evidenced that cross-

Table II. Results of equality covariance test for both gene expression data examples.

	Breast cancer data		Lung cancer data	
Method	Q_K^2	p-value	Q_K^2	p-value
Before	7.7697	0.0206	8.0182	0.0456
MC	7.7697	0.0206	8.0182	0.0456
DWD	7.7697	0.0206	8.0182	0.0456
Z-score	4.8767	0.0873	19.4446	0.0002
EB	0.7093	0.7014	3.3300	0.3435
XPN	0.2802	0.8693	1.8651	0.6009
SVA	0.4473	0.7996	6.5814	0.0865
RUV-2	0.0797	0.9609	40.0763	1.0266e-08
MBCA	0.2421	0.8860	0.1585	0.9840

Test statistics and the corresponding p-values are shown.

Table III. MCC values from cross-batch prediction.

	Breast cancer data			Lung cancer data			
Method	(23) → 1	(13) → 2	(12) → 3	(234) → 1	(134) → 2	(124) → 3	(123) → 4
Before	0.6267	0.5597	0.7191	0.0492	−0.0554	0.0416	0.1933
MC	0.6955	0.5794	0.7317	0.1268	0.1431	0.0812	0.2334
DWD	0.7133	0.6140	0.6941	0.0654	0.1421	0.1183	−0.0201
Z-score	0.6711	0.6125	0.7178	0.0367	0.1658	0.1656	0.1300
EB	0.6623	0.5159	0.7317	0.0479	0.1298	0.1183	0.0852
XPN	0.7080	0.5962	0.7209	0.1410	0.1502	0.0175	0.1654
SVA	0.5974	0.5719	0.6176	0.1247	0.2109	0.0529	0.1260
RUV-2	0.6959	0.5692	0.7317	0.3797	0.3700	0.2625	0.2400
MBCA	0.7106	0.5877	0.7317	0.1696	0.1334	0.0812	0.1442

The numbers in the parentheses indicate batches used for training.

Table IV. MCC values from cross-batch prediction for the lung cancer data with MBCA applied with a target batch.

	Lung cancer data			
Method	(234) → 1	(134) → 2	(124) → 3	(123) → 4
Before	0.0492	−0.0554	0.0416	0.1933
MBCA(B1)	0.1410	0.1693	0.1027	0.2636
MBCA(B2)	0.0654	0.1658	0.1027	0.1788
MBCA(B3)	0.0675	0.1817	0.0386	0.1689
MBCA(B4)	0.0533	0.1466	0.0745	0.1992

batch prediction performance has been generally improved by all methods, and the proposed method shows competitive performance. For the lung cancer data, we train the classifier with three batches and test it on the left-out batch. The results for the lung cancer data with overall survival as the biological signal are shown in the table as well. Note that the MCC values for the lung cancer data are much smaller than those for the breast cancer data because the prediction of the overall survival is notoriously difficult [28], while the ER status is relatively easy to predict. The exceptional performance by RUV-2 can be explained by the prediction bias mentioned earlier in this section.

Table IV shows the MCC values when MBCA is applied with a target batch for the lung cancer data. The results suggest that some batches work better as a target batch. In particular, MBCA(B1), with batch 1 as the target, excels any results in Table III, which is consonant with the fact that this site was a high volume facility experienced with microarrays.

6. Discussion

In this paper, we propose a multivariate batch adjustment method for gene expression data. The goal of the proposed method is to transfer the batches so that they become as homogeneous as possible. We use the first two moments as criteria for homogeneity. The proposed MBCA method is useful especially when there exists an ideal batch obtained in the best experimental conditions.

Preprocessing is an important step for gene expression data analysis because this step reduces technical variation across arrays, although it may not completely eliminate a batch effect. From literature, we have found that both RMA and MAS5.0 are popular and preferred by many researchers. In this work, we use RMA for one data set and MAS5.0 for the other because we intend to demonstrate that either of the two normalization methods does not entirely eliminate the batch bias. Both preprocessing methods are performed under the default setting provided at http://www.bioconductor.org. In particular, RMA is applied with a single reference distribution for all the batches, which essentially follows the principles of the more recent frozen RMA normalization [29].

Often, a great attention is given to the prediction performance in the combined data after adjustment. This is because many microarray studies are attempted with the purpose of exploring predictors (e.g., clinically useful prognostic markers for cancer), and thus, it is expected to discover more reliable predictors and increase predictive power by integrating data sets [1, 4]. Unfortunately, eliminating a batch bias in the merged data set is not always followed by improved prediction performance. For the lung cancer data in Table III, most methods (except the mean centering and RUV-2) fail to improve the cross-batch prediction performance for the fourth batch, and this weak predictive power is probably due to unpredictability of overall survival or uncertainty of bias in the batch. Luo *et al.* [28] and Yasrebi *et al.* [30] pointed out that there are more factors affecting predictive power other than a batch adjustment technique such as classification methods, sample sizes, and biological natures. Therefore, even though a batch effect removal method has a positive (or negative) impact on prediction performance in a combined data set or validation data set, this may not directly imply that the batch adjustment is effectively (or ineffectively) done.

There are some practical issues with the proposed MBCA method. The first is the choice of factors. One can use data-driven clustering results with genes, for example, K-means, which we include in the provided software as an option. However, choosing the number of clusters is another non-trivial problem. In this paper, we use the GO to avoid such argument, and it is reasonable in the biological sense. The second is computational burden, because the MBCA method involves calculation of high-dimensional covariances. We suggest to use the SVD technique proposed by Shedden *et al.* [31], which reduces the computing time from $O(p^3)$ to $O(pn^2)$.

Software in the form of Matlab code is available at https://faculty.franklin.uga.edu/jyahn/mbca

Appendix

Lemma 1 (Convergence rate for pooled covariance matrix)
Under the assumptions (i) and (ii), we have

$$\|\hat{\boldsymbol{\Sigma}}^* - \boldsymbol{\Sigma}^*\| = O_p(n^{-1/2}pq).$$

Lemma 2 (Inequality for the rates of convergence)
Under the assumptions (i), (ii), and (iv), we have

$$\mathrm{E}\|\hat{\boldsymbol{\Sigma}}^{1/2} - \boldsymbol{\Sigma}^{1/2}\|^2 \leqslant \mathrm{E}\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\|^2.$$

Lemma 3 (Convergence rate for the inverse of square root covariance estimator)
Suppose that $q = O(n^{\alpha_1})$ and $p = O(n^{\alpha})$, where $\alpha_1, \alpha \geqslant 0$. Under the assumptions (i)–(iv), we have

$$\|\hat{\boldsymbol{\Sigma}}^{-1/2} - \boldsymbol{\Sigma}^{-1/2}\| = o_p\left((p^3q^4\log(n)/n)^{1/2}\right).$$

In Lemma 1, the convergence rate of the pooled covariance estimator is determined by the term $n^{-1/2}pq$. Note that this rate is the same for the individual covariance estimators $\hat{\boldsymbol{\Sigma}}_i$, as shown in [19]. From Lemma 2, the convergence rate of $\hat{\boldsymbol{\Sigma}}^{1/2}$ is bounded by that of $\hat{\boldsymbol{\Sigma}}$. Furthermore, in Lemma 3, we show the weak convergence of the estimator $\hat{\boldsymbol{\Sigma}}^{-1/2}$ as $(p, q, n) \to \infty$. Note that p and q increase as n increases; thus, the impact of dimensionality can considerably slow down the convergence rate. The idea of Lemma 3 originated from Fan *et al.* [19], who obtained the convergence rate of $\hat{\boldsymbol{\Sigma}}^{-1}$. In a comparison to the convergence rate for $\hat{\boldsymbol{\Sigma}}^{-1}$: $\|\hat{\boldsymbol{\Sigma}}^{-1} - \boldsymbol{\Sigma}^{-1}\| = o_p\left((p^2q^4\log(n)/n)^{1/2}\right)$, it can be seen that $\hat{\boldsymbol{\Sigma}}^{-1/2}$ converges slightly slower than $\hat{\boldsymbol{\Sigma}}^{-1}$ by the order of $p^{1/2}$. From Lemmas 1, 2, and 3, the next theorem follows.

Proof of Lemma 1.
It immediately follows from $\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\| = O_p(n^{-1/2}pq)$ shown by Fan *et al.* [19]. □

Proof of Lemma 2.

$$\begin{aligned}\mathrm{E}\|\hat{\boldsymbol{\Sigma}}^{1/2} - \boldsymbol{\Sigma}^{1/2}\|^2 &= \mathrm{E}\,\mathrm{tr}\left(\hat{\boldsymbol{\Sigma}}^{1/2} - \boldsymbol{\Sigma}^{1/2}\right)^2 \\ &= \mathrm{E}\left\{\mathrm{tr}(\hat{\Lambda}) + \mathrm{tr}(\Lambda) - 2\,\mathrm{tr}\left(\hat{\mathrm{U}}\hat{\Lambda}^{1/2}\hat{\mathrm{U}}^{\mathrm{T}}\mathrm{U}\Lambda^{1/2}\mathrm{U}^{\mathrm{T}}\right)\right\},\end{aligned}$$

where $\mathbf{U}=[\mathbf{u}_1,\ldots,\mathbf{u}_p]_{p\times p}$ and $\Lambda=\mathrm{diag}(\lambda_1,\ldots,\lambda_p)$. From condition (iv),

$$\begin{aligned}
\mathrm{E}\ \mathrm{tr}\left(\hat{\mathbf{U}}\hat{\Lambda}^{1/2}\hat{\mathbf{U}}^{\mathrm{T}}\mathbf{U}\Lambda^{1/2}\mathbf{U}^{\mathrm{T}}\right) &= \mathrm{E}\ \mathrm{tr}\left\{\left(\hat{\Lambda}^{1/2}\hat{\mathbf{U}}^{\mathrm{T}}\mathbf{U}\right)(\Lambda^{1/2}\mathbf{U}^{\mathrm{T}}\hat{\mathbf{U}})\right\}\\
&= \mathrm{E}\left\{\textstyle\sum_{i=1}^{p}(\hat{\mathbf{u}}_i^{\mathrm{T}}\mathbf{u}_i)^2\hat{\lambda}_i^{1/2}\lambda_i^{1/2}+\sum_{i\neq j}^{p}\left(\hat{\mathbf{u}}_i^{\mathrm{T}}\mathbf{u}_j\right)^2\hat{\lambda}_i^{1/2}\lambda_j^{1/2}\right\}\\
&= \mathrm{E}\left\{\textstyle\sum_{i=1}^{p}\mathbf{u}_i^{\mathrm{T}}\mathrm{E}\left(\hat{\mathbf{u}}_i\hat{\mathbf{u}}_i^{\mathrm{T}}\mid\hat{\lambda}_i\right)\mathbf{u}_i\hat{\lambda}_i^{1/2}\lambda_i^{1/2}+\sum_{i\neq j}^{p}\mathbf{u}_j^{\mathrm{T}}\mathrm{E}\left(\hat{\mathbf{u}}_i\hat{\mathbf{u}}_i^{\mathrm{T}}\mid\hat{\lambda}_i\right)\mathbf{u}_j\hat{\lambda}_i^{1/2}\lambda_j^{1/2}\right\}\\
&= \mathrm{E}\left(\textstyle\sum_{i=1}^{p}\hat{\lambda}_i^{1/2}\lambda_i^{1/2}\right).
\end{aligned}$$

Similarly, $\mathrm{E}\ \mathrm{tr}\left(\hat{\mathbf{U}}\hat{\Lambda}\hat{\mathbf{U}}^{\mathrm{T}}\mathbf{U}\Lambda\mathbf{U}^{\mathrm{T}}\right)=\mathrm{E}\left(\sum_{i=1}^{p}\hat{\lambda}_i\lambda_i\right)$. Therefore,

$$\begin{aligned}
&\mathrm{E}\|\hat{\boldsymbol{\Sigma}}^{1/2}-\boldsymbol{\Sigma}^{1/2}\|^2-\mathrm{E}\|\hat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}\|^2\\
&=\mathrm{E}\left\{\textstyle\sum_{i=1}^{p}\hat{\lambda}_i+\sum_{i=1}^{p}\lambda_i-2\sum_{i=1}^{p}\hat{\lambda}_i^{1/2}\lambda_i^{1/2}\right\}-\mathrm{E}\left\{\sum_{i=1}^{p}\hat{\lambda}_i^2+\sum_{i=1}^{p}\lambda_i^2-2\sum_{i=1}^{p}\hat{\lambda}_i\lambda_i\right\}\\
&=\mathrm{E}\left\{\textstyle\sum_{i=1}^{p}(\hat{\lambda}_i^{1/2}-\lambda_i^{1/2})^2-\sum_{i=1}^{p}(\hat{\lambda}_i-\lambda_i)^2\right\}\leqslant 0.
\end{aligned}$$

□

Proof of Lemma 3.
Fan *et al*. [19] showed that

$$\hat{\boldsymbol{\Sigma}}^{-1}=\hat{\boldsymbol{\Sigma}}_0^{-1}-\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}.$$

Multiplying $\hat{\boldsymbol{\Sigma}}^{1/2}$ to both sides, we obtain

$$\hat{\boldsymbol{\Sigma}}^{-1/2}=\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\boldsymbol{\Sigma}}^{1/2}-\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\boldsymbol{\Sigma}}^{1/2}.$$

Then the estimation error of $\hat{\boldsymbol{\Sigma}}^{-1/2}$ is as follows:

$$\begin{aligned}
\|\hat{\boldsymbol{\Sigma}}^{-1/2}-\boldsymbol{\Sigma}^{-1/2}\| \leqslant\ & \|\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\boldsymbol{\Sigma}}^{1/2}-\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{1/2}\|\\
&+\|\left(\hat{\boldsymbol{\Sigma}}_0^{-1}-\boldsymbol{\Sigma}_0^{-1}\right)\hat{\mathbf{B}}\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\boldsymbol{\Sigma}}^{1/2}\|\\
&+\|\boldsymbol{\Sigma}_0^{-1}\hat{\mathbf{B}}\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}\hat{\mathbf{B}}^{\mathrm{T}}\left(\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\boldsymbol{\Sigma}}^{1/2}-\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{1/2}\right)\|\\
&+\|\boldsymbol{\Sigma}_0^{-1}\left(\hat{\mathbf{B}}-\mathbf{B}\right)\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}\hat{\mathbf{B}}^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{1/2}\|\\
&+\|\boldsymbol{\Sigma}_0^{-1}\mathbf{B}\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}\left(\hat{\mathbf{B}}^{\mathrm{T}}-\mathbf{B}^{\mathrm{T}}\right)\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{1/2}\|\\
&+\|\boldsymbol{\Sigma}_0^{-1}\mathbf{B}\left\{\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}-\left[\mathrm{cov}(\mathbf{x})^{-1}+\mathbf{B}^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}\mathbf{B}\right]^{-1}\right\}\mathbf{B}^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{1/2}\|\\
\hat{=}\,&\mathcal{H}_1+\mathcal{H}_2+\mathcal{H}_3+\mathcal{H}_4+\mathcal{H}_5+\mathcal{H}_6.
\end{aligned}$$

In what follows, we obtain bounds for $\mathcal{H}_i$, $i=1,\ldots,6$. First, from $\mathrm{E}\|\hat{\boldsymbol{\Sigma}}^{1/2}\|^2=\mathrm{E}\ \mathrm{tr}(\hat{\boldsymbol{\Sigma}})=O(qp)$, we have $\|\hat{\boldsymbol{\Sigma}}^{1/2}\|=O_p(p^{1/2}q^{1/2})$, and from Lemma 2, we have $\|\hat{\boldsymbol{\Sigma}}^{1/2}-\boldsymbol{\Sigma}^{1/2}\|=O_p(n^{-1/2}pq)$; thus, $\mathcal{H}_1\leqslant\|\hat{\boldsymbol{\Sigma}}_0^{-1}-\boldsymbol{\Sigma}_0^{-1}\|\|\hat{\boldsymbol{\Sigma}}^{1/2}\|+\|\boldsymbol{\Sigma}_0^{-1}(\hat{\boldsymbol{\Sigma}}^{1/2}-\boldsymbol{\Sigma}^{1/2})\|=O_p(n^{-1/2}pq)$. The second term is

$$\begin{aligned}
\mathcal{H}_2 &\leqslant \|\left(\hat{\boldsymbol{\Sigma}}_0^{-1}-\boldsymbol{\Sigma}_0^{-1}\right)\hat{\boldsymbol{\Sigma}}_0^{1/2}\|\times\|\hat{\boldsymbol{\Sigma}}_0^{-1/2}\hat{\mathbf{B}}\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1/2}\|\times\|\hat{\boldsymbol{\Sigma}}_0^{-1/2}\hat{\boldsymbol{\Sigma}}^{1/2}\|\\
&= O_p(n^{-1/2}p^{1/2})\,O_p(q^{1/2})\,O_p(p^{1/2}q^{1/2})\\
&= O_p(n^{-1/2}pq).
\end{aligned}$$

Similarly, we can bound $\mathcal{H}_3$

$$\begin{aligned}\mathcal{H}_3 &\leqslant \|\boldsymbol{\Sigma}_0^{-1/2}\|\|\boldsymbol{\Sigma}_0^{-1/2}\hat{\mathbf{B}}\left[\widehat{\mathrm{cov}}(\mathbf{f})^{-1}+\hat{\mathbf{B}}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\mathbf{B}}\right]^{-1}\hat{\mathbf{B}}^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1/2}\|\|\boldsymbol{\Sigma}_0^{1/2}\left(\hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\boldsymbol{\Sigma}}^{1/2}-\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{1/2}\right)\| \\ &= O(p^{1/2})O(q^{1/2})o_p((n/\log(n))^{-1/2}pq) \\ &= o_p((n/\log(n))^{-1/2}p^{3/2}q^{3/2}).\end{aligned}$$

The rest of the terms are, by a slight modification of the calculations in [19],

$$\mathcal{H}_4 = O_p(n^{-1/2}p^{3/2}q), \quad \mathcal{H}_5 = O_p(n^{-1/2}p^{3/2}q), \quad \text{and} \quad \mathcal{H}_6 = o_p((n/\log(n))^{-1/2}p^{3/2}q^2).$$

In conclusion,

$$\sqrt{np^{-3}q^{-4}/\log(n)}\,\|\hat{\boldsymbol{\Sigma}}^{-1/2}-\boldsymbol{\Sigma}^{-1/2}\| \xrightarrow{P} 0 \quad \text{as } n\to\infty.$$

□

Proof of Theorem 1.
From the lemmas and assumption (iii), we can show that

$$\begin{aligned}\|\hat{\boldsymbol{\Sigma}}^{*1/2}\hat{\boldsymbol{\Sigma}}_i^{-1/2}-\boldsymbol{\Sigma}^{*1/2}\boldsymbol{\Sigma}_i^{-1/2}\| &\leqslant \|\hat{\boldsymbol{\Sigma}}^{*1/2}\|\|\hat{\boldsymbol{\Sigma}}_i^{-1/2}-\boldsymbol{\Sigma}_i^{-1/2}\|+\|\hat{\boldsymbol{\Sigma}}^{*1/2}-\boldsymbol{\Sigma}^{*1/2}\|\|\boldsymbol{\Sigma}_i^{-1/2}\| \\ &= O_p(p^{1/2}q^{1/2})o_p\{(p^3q^4\log(n)/n)^{1/2}\}+O_p(n^{-1/2}pq)O(p^{1/2}).\end{aligned}$$

As $p^{3/2}q = o((n/\log(n))^{1/2})$, we have the following result:

$$\sqrt{np^{-4}q^{-5}/\log(n)}\,\|\hat{\boldsymbol{\Sigma}}^{*1/2}\hat{\boldsymbol{\Sigma}}_i^{-1/2}-\boldsymbol{\Sigma}^{*1/2}\boldsymbol{\Sigma}_i^{-1/2}\| \xrightarrow{P} 0 \quad \text{as } n\to\infty.$$

□

Acknowledgement

This research is supported by National Institutes of Health (R21CA152460-01A1).

References

1. Xu L, Tan AC, Naiman DQ, Geman D, Winslow RL. Robust prostate cancer marker genes emerge from direct integration of inter-study microarray data. *Bioinfometics* 2005; **21**(20):3905–3911.
2. Michiels S, Koscielny S, Hill C. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet* 2005; **365**(9458):488–492.
3. Ein-Dor L, Zuk O, Domany E. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences of the USA* 2006; **103**:5923–5928.
4. Cheng C, Shen K, Song C, Luo J, Tseng GC. Ratio adjustment and calibration scheme for gene-wise normalization to enhance microarray inter-study prediction. *Bioinformatics* 2009; **25**(13):1655–1661.
5. Scherer A. *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*. Wiley: Chichester, United Kingdom, 2009.
6. Baggerly KA, Morris JS, Coombes KR. Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments. *Bioinformatics* 2004; **20**:777–785.
7. Dobbin KK, Beer DG, Meyerson M, Yeatman TJ, Gerald WL, Jacobson JW, Conley B, Buetow KH, Heiskanen M, Simon RM, Minna JD, Girard L, Misek DE, Taylor JMG, Hanash S, Naoki K, Hayes DN, Ladd-Acosta C, Enkemann SA, Viale A, Giordano TJ. Interlaboratory comparability study of cancer gene expression analysis using oligonucleotide microarrays. *Clinical Cancer Research* 2005; **11**:565–572.
8. Irizarry RA, Warren D, Spencer F, Kim IF, Biswal S, Gabrielson BC, Frank E, Garcia JGN, Geoghegan J, Germino G, Griffin C, Hilmer SC, Hoffman E, Jedlicka AE, Kawasaki E, Martinez-Murillo F, Morsberger L, Lee H, Petersen D, Quackenbush J, Scott A, Wilson M, Yang Y, Ye SQ, Yu W. Multiple-laboratory comparison of microarray platforms. *Nature Methods* 2005; **2**(5):345–350.
9. MAQC Consortium. The microarray quality control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nature Biotechnology* 2006; **24**:1151–1161.
10. Parker HS, Leek JT. The practical effect of batch on genomic predictions. *Statistical Applications in Genetics and Molecular Biology* 2012; **11**(3):Article 10.
11. Leek JT, Scharpf RB, Bravo HC, Simcha D, Langmead B, Johnson WE, Geman D, Baggerly K, Irizarry RA. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics* 2010; **11**:733–739.

12. Shedden K, Taylor JMG, Enkemann SA, Tsao MS, Yeatman TJ, Gerald WL, Eschrich S, Jurisica I, Venkatraman SE, Meyerson M, Kuick R, Dobbin KK, Lively T, Jacobson JW, Beer DG, Giordano TJ, Misek DE, Chang AC, Zhu CQ, Strumpf D, Hanash S, Shepherd FA, Ding K, Seymour L, Naoki K, Pennell N, Weir B, Verhaak R, Ladd-Acosta C, Golub T, Gruidl M, Szoke J, Zakowski M, Rusch V, Kris M, Viale A, Motoi N, Travis W, Sharma A. Gene-expression-based survival prediction in lung adenocarcinoma: a multi-site, blinded validation study. *Nature Medicine* 2008; **14**(8):822–827.
13. Marron JS, Todd M, Ahn J. Distance weighted discrimination. *Journal of the American Statistical Association* 2007; **102**:1267–1271.
14. Benito M, Parker J, Du Q, Wu J, Xiang D, Perou CM, Marron JS. Adjustment of systematic microarray data biases. *Bioinformatics* 2004; **20**(1):105–114.
15. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007; **8**(1):118–127.
16. Leek JT, Storey JD. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics* 2007; **3**(9):e161.
17. Shabalin AA, Tjelmeland H, Fan C, Perou CM, Nobel AB. Merging two gene-expression studies via cross-platform normalization. *Bioinformatics* 2008; **24**(9):1154–1160.
18. Gagnon-Bartsch JA, Speed TP. Using control genes to correct for unwanted variation in microarray data. *Biostatistics* 2012; **13**(3):539–552.
19. Fan J, Fan Y, Lv J. High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics* 2008; **147**:186–197.
20. Carvalho CM, Chang J, Lucas JE, Nevins JR, Wang Q, West M. High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association* 2008; **103**(484):1438–1456.
21. Efron B, Tibshirani R. On testing the significance of sets of genes. *The Annals of Applied Statistics* 2007; **1**:107–129.
22. Montaner D, Minguez P, Al-Shahrour F, Dopazo J. Gene set internal coherence in the context of functional profiling. *BMC Genomics* 2009; **10**:1–13. DOI: 10.1186/1471–2164–10–197.
23. Claesson MJ, O'Sullivan O, Wang Q, Nikkila J, Marchesi JR, Smidt H, de Vos WM, Ross RP, O'Toole PW. Comparative analysis of pyrosequencing and a phylogenetic microarray for exploring microbial community structures in the human distal intestine. *PLoS ONE* 2009; **4**:1–15. DOI:10.1371/journal.pone.0006669.
24. Bickel PJ, Levina E. Covariance regularization by thresholding. *The Annals of Statistics* 2008; **36**(6):2577–2604.
25. Cai T, Liu W. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association* 2011; **106**(494):672–684.
26. Srivastava MS, Yanagihara H. Testing the equality of several covariance matrices with fewer observations than the dimension. *Journal of Multivariate Analysis* 2010; **101**:1319–1329.
27. Guo Y, Hastie T, Tibshirani R. Regularized discriminant analysis and its application in microarrays. *Biostatistics* 2005; **1**(1):1–18.
28. Luo J, Schumacher M, Scherer A, Sanoudou D, Megherbi D, Davison T, Shi T, Tong W, Shi L, Hong H, Zhao C, Elloumi F, Shi W, Thomas R, Lin S, Tillinghast G, Liu G, Zhou Y, Herman D, Li Y, Deng Y, Fang H, Bushel P, Woods M, Zhang J. A comparison of batch effect removal methods for enhancement of prediction performance using MARQ-II microarray gene expression data. *The Pharmacogenomics Journal* 2010; **10**:278–291.
29. McCall MN, Bolstad BM, Irizarry RA. Frozen robust multiarray analysis(fRMA). *Biostatistics* 2010; **11**(2):242–253.
30. Yasrebi H, Sperisen P, Praz V, Bucher P. Can survival prediction be improved by merging gene expression data sets?. *PLoS ONE* 2009; **4**(10):e7431.
31. Hastie T, Tibshirani R. Efficient quadratic regularization for expression arrays. *Biostatistics* 2004; **5**(3):329–340.