Batch effect on covariance structure confounds gene coexpression

Daniel Schlauch[1,2], Joseph Paulson[1], Kimberly Glass[2,3], and John Quackenbush[1,3]

[1]Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute and Department of Biostatistics, Harvard TH Chan School of Public Health, Boston, MA 02115
[2]Channing Division of Network Medicine, Brigham and Women's Hospital, Boston, MA 02115
[3]Department of Medicine, Harvard Medical School, Boston, MA 02115
[4]Pulmonary and Critical Care Division, Brigham and Women's Hospital and Harvard Medical School, Boston, USA

April 3, 2017

Abstract

Systemic biases associated with multiple batches of gene expression experiments have been known to confound results in differential gene expression analyses and numerous methods have been developed over the past 10 years that address this phenomenon. Commonly, these approaches adjust expression values such that the mean and variance of each gene is conditionally independent of a set of batch covariates. However, methods published to date have not addressed the potential for differential coexpression across batches. While this is of lesser concern in the context of standard differential gene expression, analyses that utilize a gene coexpression or correlation matrix will continue to see confounding due to batch effect even when applied to a properly batch-corrected gene expression matrix. In this article, we demonstrate the persistence of confounding at the covariance level after standard batch correction using simulation studies and real biological examples. We present an approach for computing a corrected gene expression coexpression matrix, called [NAME], based on estimation of the conditional covariance matrix. [NAME] estimates a reduced set of parameters that express the coexpression as a function of the sample covariates and can be used to control for continuous and categorical confounders. The method is computationally fast, and makes use of the inherently modular structure of features commonly found in genomic analyses.

1 Introduction

High-throughput data generation, including RNA-sequencing and microarrays, have revolutionized molecular biology. These technological advancements allow for the measurement of tens of thousands of gene expression patterns at once, giving us a window into the molecular activity of living cells. But as promising as these data generation methods are, the deluge of data that has arisen from these tools has revealed an extraordinarily level of complexity in linking the information at the gene level to the higher level observations of phenotypes.

Plummeting costs have lead to increased accessibility of high-throughput genomic assays and with that we gain ability to investigate numerous hypotheses simultaneously. At the heart of most genomic studies is the analysis of the manner in which the biological variability of genomic features, such as RNA expression, differs in the context of phenotypes and/or other genomic features. We hope that understanding the joint distribution of gene expression, conditional on phenotypes, will lead to an understanding of the core biology. However, it can be difficult to distinguish which associations are driven by real biological mechanisms and which associations are observed because of confounding by undesirable batch effects or other extraneous experimental variables. It is critical to address this confounding in order to reduce the probability of false positive results.

In the context of gene expression studies, the measurement of biological sources of variation RNA abundance are typically of interest. Commonly, observed variation is often the result of technical artifacts that may confound associations between experimental groups and gene expression [Leek et al., 2010,Lander, 1999].

Batch effects are known to come from many sources. Some sources are obvious, such as the array platform or the experimental reagents used, but others may be more unexpected. Timing, ozone [Fare et al., 2003], technician and lab humidity have all been identified as sources of unwanted variation and undoubtedly many other sources remain undiscovered [Scherer, 2009]. In other words, variation attributed to batch is unavoidable. Ideally, experimental design will allow for single-batch experiments to be performed, but experiments may be too large for this to be practical. Randomized batch assignment is recommended as an experimental design [Conesa et al., 2016], but many investigations involve the use of publicly available data (e.g. Gene Expression Omnibus, Genomics Data Commons) for which no randomization is possible.

A common way to approach the batch correction problem is to consider the model $G_{ij} = \alpha_j + X\beta_j + B\gamma_{ij} + \delta_{ij}\epsilon_{ij}$, where G_{ij} is the gene expression of gene j for sample i, X is the design matrix, β_j is a vector of regression coefficients for gene j for the columns of X. The next two terms specify the additive and multiplicative impacts of batch. B is an matrix of indicators for each of the batches, and γ_j is a vector of additive batch effects on gene j. ϵ_{ij} is the error term and δ_{ij} is the multiplier of that error term. Controlling for batch necessarily involves estimating the impact of batch on the mean expression and the variance of that expression, specifically γ_{ij} and δ_{ij}, for each gene. It is generally not

known what mechanism for batch effect is at fault for a particular study and consequently, it is unknown which set of genes and the magnitude of the effect on those genes. Therefore, without knowing which features are susceptible to batch effect, it is typical to estimate γ_{ij} and δ_{ij} for each separately gene in a study.

Despite widespread literature published regarding the identification and control of confounding due to batch effect [Chen et al., 2011,Benito et al., 2004,Leek and Storey, 2007,Johnson et al., 2007,Nygaard et al., 2016], batch effect correction has focused on adjusting for the effects of batch on gene expression mean and variance at an individual level. For example, ComBat [Johnson et al., 2007] uses an empirical bayes approach to estimate the mean and variance parameters for each gene and then computes an adjusted gene expression that controls for these effects. Another approach, Surrogate Variable Analysis [Leek and Storey, 2007], uses a combination of measured covariates and singular value decomposition to identify unknown sources of variation. These variables are estimated and their effects regressed out of the gene expression matrix. These approaches amount to a location-scale adjustment that is critical for promoting the conditional independence of gene expression. For the purposes of differential gene expression analysis, this approach is effective for both microarrays and RNA-seq data [Conesa et al., 2016].

However, as our understanding of genomics grows, we recognize that finding differentially expressed genes do not give a complete picture of relationship between the transcriptome and phenotype. Cellular states involve the complex combination of numerous biological processes that are characterized by the behavior of large sets of interacting genes. This understanding has lead to increased interest measuring gene coexpression, the degree to which the expression between two gene is correlated, to gain an understanding of the network biology where simple differential gene expression falls short. Analogous to differential expression and complementary to network inference, we are naturally interested in differential coexpression - the difference in gene correlation across experimental conditions.

The difference between differential coexpression analysis and differential expression is that we focus on the pairwise joint distribution of genes as opposed to the marginal distribution of each gene. Essentially, we assume that genes that are functionally related will exhibit a correlated expression pattern across a set of experimental conditions or samples. A significant association may indicate a potential functional interaction. With this in mind, a natural goal is the identification of those genes that are differentially correlated. Gene pairs or gene sets that gain or lose a common expression pattern in differing experimental conditions may implicate the biological pathways or functional mechanisms that drive a particular phenotypic change.

Many methods have been proposed in for differential coexpression analysis, most commonly in the field of gene network inference. Many of these proposed algorithms start with the computation of a correlation matrix from the gene expression data [Tesson et al., 2010, Langfelder and Horvath, 2008, Langfelder and Horvath, 2012, Glass et al., 2013, Southworth et al., 2009, Choi et al., 2005,

Siska and Kechris, 2017]. There is an assumption, often implicit, that the gene expression data lacks heterogeneity or has heterogeneity sufficiently corrected. The biases that persist in the coexpression matrix when homogeneity is violated are rarely discussed or considered in the literature. Each of the methods in the field would benefit from an estimated coexpression matrix that has had the impact of batch effect reduced compared to a standard Pearson correlation matrix.

In estimating coexpression matrices, standard batch correction is critical [Furlotte et al., 2011], but not sufficient. Location-scale confounding on gene expression will reduce power and bias results. This will inevitably lead to highly significant, but biologically meaningless associations between large volumes of genes. Though the common batch correction practices help mitigate this problem, they fail to remove the impact of the type of batch effect that causes differential coexpression patterns among gene. Current methods treat batch effect as acting on the marginal distribution of each gene and ignore the possibility of changes to joint distributions. While some impact on the joint distribution is addressed by removing the impact of differential means and variances across batches it is insufficient if the covariance itself is associated with batch.

It is easy to conceive of scenarios where this phenomenon plays out. For example, different experimental protocols across batches may induce a coexpression difference by preferentially sampling cells with certain active biological pathways for cell cycle or stress response. But even simpler, batch effect for coexpression may be introduced merely by differential biological variability. To illustrate this, recognize that correlation is roughly interpreted as the square root of proportion of variability explained by true expression (as opposed to other sources of error). Then two genes which are functionally related will only be detected as such if the biological variability is equal between batches. Subtle differences in protocol that lead to differences in biological variability can not be removed with standard batch correction methods.

The demonstration in Figure 1 shows two examples of uncorrected batch effect (left) impacting two genes in a study. In the top row, batch effect alters the means and variances of the two genes (location-scale model). In the bottom row, the means, variances *and* coexpression is impacted. Upon application of ComBat (Right) to the uncorrelated genes, the two genes become independent as desired. However, when applied to the conditionally coexpressed case (Bottom row) we continue to observe differential coexpression across batches.

Though methods exist to consider the correlation of genes in the presence of location-scale batch effect, no method exists that further allows for the coexpression itself to be a function of batch. Similarly, no method currently available returns a corrected coexpression matrix rather than a corrected expression matrix.

While the impact of ignoring pairwise interactions may be negligible for simple differential gene expression analyses, coexpression patterns are widely considered in the field of network inference [de la Fuente, 2010,Chen et al., 2011, Fukushima, 2013]. The impact of confounding due to differential coexpression in batches remains unexamined.

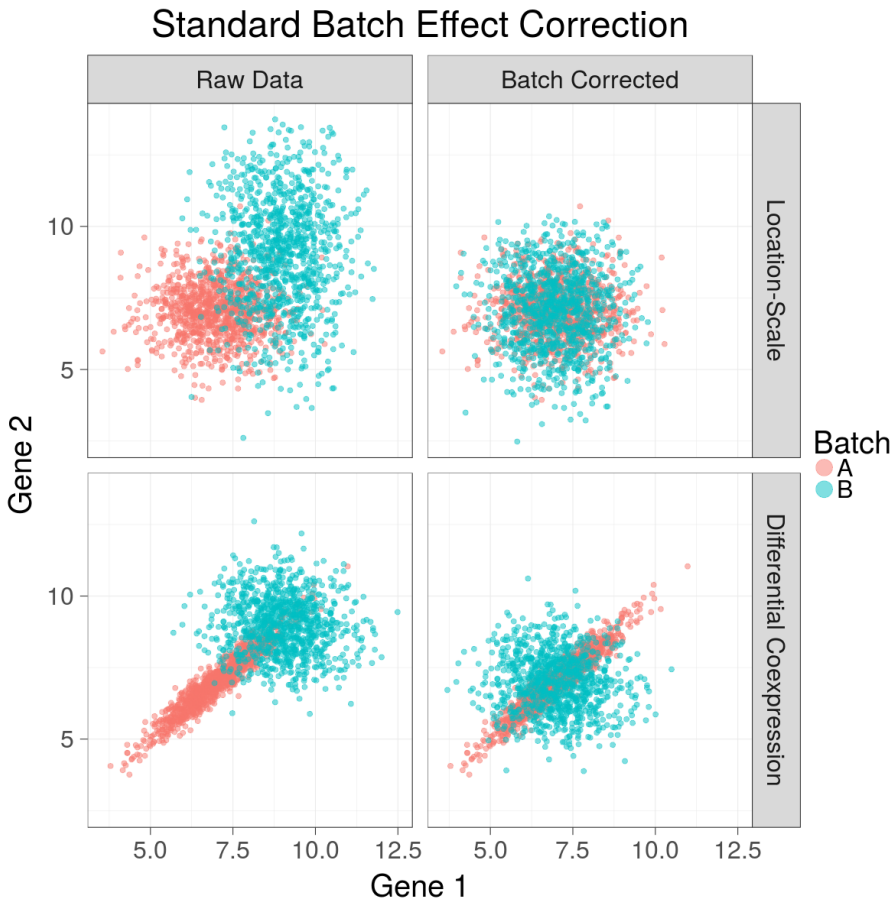

Figure 1: In this toy example, we demonstrate which artifacts standard batch correction is capable of correcting and which artifacts will remain. In A-D, we show plots of two example genes before (left) and after (right) correction, colored by their batch. In the top row (A,B), we show a comparison of two genes which are conditionally independent and demonstrate that location-scale batch correction appropriately removes the marginal dependence between the genes. In the bottom row (C,D), we show two genes that are conditionally coexpressed and illustrate that batch correction may help mitigate the measured coexpression, but the resulting coexpression is still a function of the batch membership. Importantly, when comparing coexpression matrices, differing batch proportions will bias the differential coexpression. In simulations we demonstrate that in the absence of batched differential coexpression, ComBat sufficiently controls the type I error. However, when coexpression differs by batch, our false positive rate increases above the expectation of the null model.

In order to solve this problem, we need to create a model that describes the coexpression matrix as a function of the experimental conditions and batches. Classical regression models predict the expectation of a response variable, but in coexpression analyses, we are interested in the covariance. Some work has recently been published on this subject [Hoff and Niu, 2012], but little has been studied in high dimensional or biological settings.

This problem faces at least two major challenges. The first problem is that it requires the estimation of a very large number of parameters. Given p genes, there are $\binom{p}{2}$ pairwise correlations, and each of these must be a function of the number of covariates. For most high throughput gene expression studies where $N \ll p$, we want to limit this parameter space in some way. Previous work has shown the increased difficulty in reproducing coexpression across studies [Schlauch et al., 2016] likely owing to the high number of parameters to estimate in noisy data. Recent work has allowed for the imposition of sparsity on the gene covariance matrix [Bien et al., 2011] or precision matrix [Friedman et al., 2008], but the complexity of biological systems make sparsity an imperfect choice and computationally burdensome to implement. Second, in the case of numerous batches or continuous covariates, it may not be possible to estimate a coexpression matrix using the sample covariance matrix form, $\frac{1}{n}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(X_i - \bar{X}\right)^T$, where X_i is the set of all gene expression values for sample i.

In the method we describe here, [NAME], we reduce the parameter space by exploiting the modular nature of gene expression, estimating only N variables for each covariate, with each weight corresponding to a eigenvector. This collects the information from many similarly expressed genes by effectively borrowing information from similarly patterned features. This allows us to estimate gene coexpression matrix as a function of sample covariates. Our method is presented in a regression framework that allows for the inclusion of continuous and categorical covariates into the adjustment model.

2 Methods

2.1 Approach

In this manuscript we present a method for estimating the coexpression matrix by modeling the matrix as a function of the largest components of variation. Critical to our approach is the idea that although there are $\binom{p}{2}$ pairwise gene-gene relationships, the true biology can be predominantly explained by a much smaller set of variance components. One way to identify these components is to compute the eigendecomposition of the gene correlation matrix. We can then write the coexpression matrix as a function of the experimental covariates and these eigenvectors. Solving this formulation by minimizing the squared error will yield a set of parameter estimates from which we can compute corrected coexpression estimates.

Consider a set of N samples with q covariates measuring gene expression

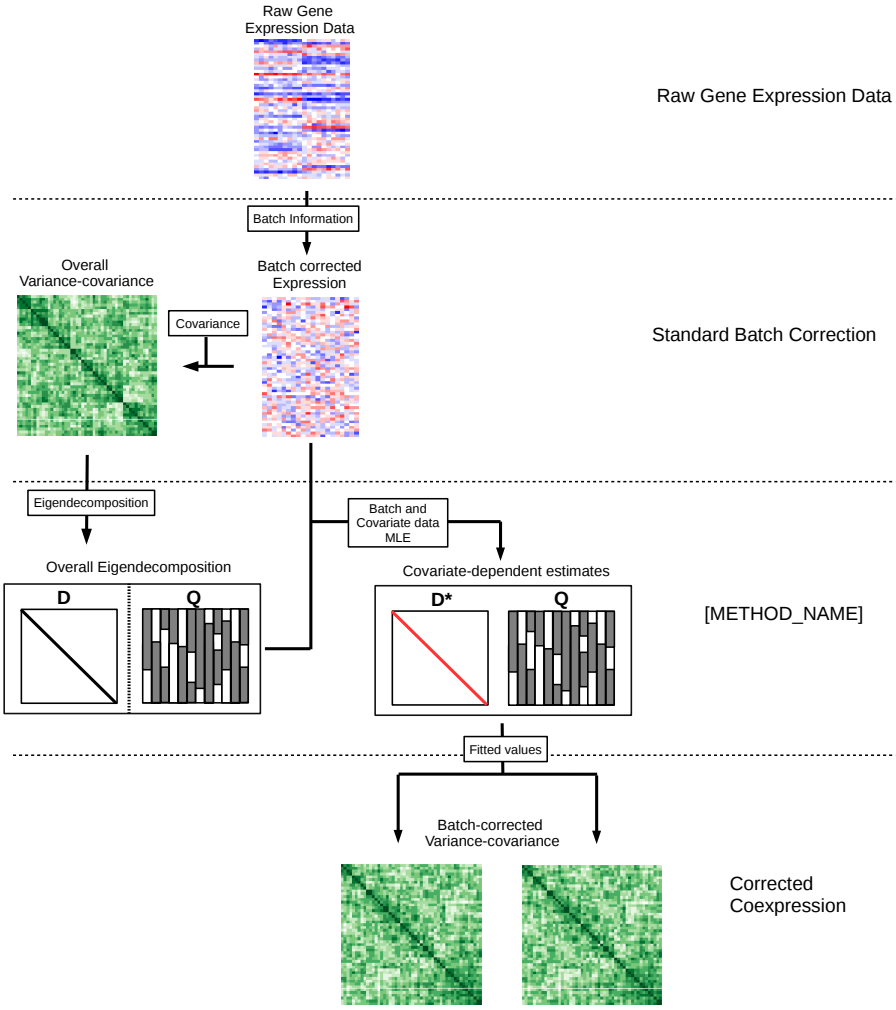

Figure 2: **Workflow of [NAME]**. [NAME] begins with a raw or normalized gene expression dataset. (1) Standard batch correction (ComBat) is applied to remove location-scale batch effect. (2) The overall coexpression matrix is calculated. (3) An eigendecomposition of the overall coexpression matrix is computed. The eigenvectors from this decomposition are then used to re-estimate "pseudo-eigenvalues" that minimize the coexpression error from the batch corrected expression data. (4) Fitted values obtained from this estimation, in combination with the eigenvector matrix, $\mathbf{Q}$, are used to estimated covariate-dependent coexpression matrices such as for batch corrected network inference or differential coexpression analysis.

across p genes. Let $\mathbf{x}_i = (x_{i1}, \ldots, x_{iq})$ denote the covariates for sample i and let $\mathbf{g}_i = (g_{i1}, \ldots, g_{ip})$ denote the gene expression values for sample i for the p genes.

In multivariate regression form we can express this as

$$\mathbf{g}_i = \beta^T \mathbf{x}_i + \epsilon_i \text{ for } i = 1, \ldots, N$$

where β is a $q \times p$ matrix of coefficients.

Equivalently,

$$\mathbf{G} = \mathbf{X}\beta + \mathbf{E}$$

where $\mathbf{G}$, $\mathbf{X}$, and $\mathbf{E}$ are each matrices with row i corresponding to $\mathbf{g}_i$, $\mathbf{x}_i$, and ϵ_i respectively.

Here, we make the usual multivariate assumption for $\mathbf{E}$ that the rows $\epsilon_i, \ldots, \epsilon_N$ are conditionally independent, and follow distribution, $MVN_p(\mathbf{0}_p, \Sigma_i)$. Notably in this paper, the covariance of ϵ_i differ according to i.

Estimating the covariance structure for a set of p genes typically involves computing the sample covariance matrix, S, with entries $s_{jk} = \frac{1}{N-1} \sum_{i=1}^{N} (G_{ij} - \bar{G_{\cdot j}})(G_{ik} - \bar{G_{\cdot k}})$. However, as is typical in high-throughput settings, $p \gg N$, producing an estimated covariance matrix $p \times p$ with column rank $\leq N$.

To address this "curse of dimensionality", numerous methods have been proposed. One might use a series of LASSO regressions to estimate parameters in the inverse covariance matrix [Meinshausen and Bühlmann, 2006], or perform penalized maximum likelihood estimation with the penalty on the inverse covariance matrix [Banerjee et al., 2008, Yuan and Lin, 2007, Friedman et al., 2008]. Each of these approaches imposes sparsity on the precision matrix, effectively assuming a large degree of conditional independence between genes. More recent work has explored imposing sparsity on the covariance matrix itself, rather than the precision matrix [Bien et al., 2011], which allows us to assume widespread marginal independence of genes.

The approach we take here involves estimating a covariance matrix Σ_i which depends on the batch and experimental design features of sample i. An estimate of Σ_i that allows all elements of the matrix to vary freely can be obtained by separately estimating the covariance matrix for each unique row of $\mathbf{X}$. However, this approach in impractical for a large number of categorical covariates or any continuous covariates. Additionally, it neglects the information in other samples and other genes which can be used to gain a better estimate of the coexpression. Given that groups of genes often behave in distinct patterns, it is inefficient to estimate coexpression values for every pairwise combination of genes.

Instead, we approach the problem by making use of the fact that genes commonly behave in coexpressed modules, and that the dimensional space is effectively much smaller than p^2. To do this, we decompose the gene expression correlation matrix and find a set of eigenvectors which explain the variation. We then attempt to infer a diagonal matrix of "pseudo-eigenvalues", which minimize the square error. This procedure allows us to reduce the parameter space from p^2 to p or less while still considering the bulk of the variability in the data.

Furthermore, in the application to the gene expression data, the column rank of the coexpression matrix will be $N-1$, and the number of non-zero eigenvalues will also be only $N-1$. Therefore, we need only estimate the parameters corresponding to eigenvectors with non-zero eigenvalues substantially reducing the parameter space from p to $N-1$.

Formally, for Σ_i we estimate $\Sigma_i = \mathbf{Q}\mathbf{\Lambda}_i\mathbf{Q}^T$, where $\mathbf{Q}$ is held constant as the set of eigenvectors from the full coexpression matrix. In this formulation,$\mathbf{\Lambda}_i$ is a diagonal matrix with entries

$$\mathbf{\Lambda}_{i,kk} = \mathbf{x}_i\mathbf{\Psi}_{\cdot k} \tag{1}$$

where $\mathbf{x}_i$ is the predictors for sample i and$\mathbf{\Psi}$ is a $p \times q$ matrix of coefficients.

Because we don't estimate the pseudo-eigenvalues after $k = N-1$, we set $\mathbf{\Psi}_{\cdot k} = \mathbf{0}_q$ for all $k \geq N$.

Intuitively, we can think of the parameter matrix $\mathbf{\Psi}$ as adjusting the eigenvalues as a function of the covariates to minimize the coexpression error. It is straightforward to show that in the case of a single batch and no experimental conditions, i.e. $\mathbf{x}_i = 1$ for all $i \in N$, then$\mathbf{\Psi}$ becomes identical to the vector of eigenvalues from the original covariance matrix.

2.2 Likelihood function

The likelihood function for a multivariate normal with mean μ and variance-covariance Σ is

$$\mathcal{L}(\mu, \Sigma) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{G}_i - \mu)^T \Sigma_i^{-1} (\mathbf{G}_i - \mu)}$$

The maximum likelihood estimation of μ is simply the vector $\bar{\mathbf{g}} = \frac{\sum_{i=1}^{N} \mathbf{g}_i}{N}$ and since μ is independent of Σ we can subtract off the row means, yielding $\mathbf{G}_i^* = \mathbf{G}_i - \bar{\mathbf{g}}$. And plugging in our index dependent covariance matrix from equation 1 we have

$$\mathcal{L}(\gamma) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{p}{2}} |\mathbf{Q}\text{diag}(\mathbf{x}_i\Psi)\mathbf{Q}^T|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{G}_i^*)^T \left(\mathbf{Q}\text{diag}(\mathbf{x}_i\Psi)\mathbf{Q}^T\right)^{-1} (\mathbf{G}_i^*)}$$

where $\text{diag}(\mathbf{x}_i\Psi)$ is defined as a matrix with 0's in all off-diagonal entries and diagonal equal to $\mathbf{x}_i\Psi$.

2.3 Estimator

In estimating the parameters in the matrix Ψ, we may consider that each row, i, of Ψ corresponds to the vector of contributions from the i^{th} eigenvector of $\mathbf{Q}$. With $\mathbf{Q}_i$ specifying the i^{th} column of $\mathbf{Q}$ we have that $\mathbf{Q}_i^T\mathbf{Q}_j = 0$ for all $i \neq j$ and $\mathbf{Q}_i^T\mathbf{Q}_i = 1$ for all $i, j \in 1, 2, \ldots p$.

For some $h \in 1, 2, \ldots p$, we seek to find the estimates $\hat{\Psi}_h$ which minimize the squared error of the estimated correlation matrices defined as $\mathbf{G}_i\mathbf{G}_i^T$ for each sample $i \in 1, 2, \ldots, N$. By the Orthogonal Decomposition Theorem, the "error residuals" $\mathbf{Q}_h^T \left[\mathbf{G}_i\mathbf{G}_i^T - \mathbf{Q}_h\mathbf{X}_i\hat{\Psi}_h\mathbf{Q}_h^T\right]\mathbf{Q}_h$ will be minimized when they are orthogonal to the hyperplane spanned by $\mathbf{X}$. Therefore, we can set the product below (Equation 2) equal to the zero vector to solve for our estimator $\hat{\Psi}$.

$$\mathbf{0}_q = \sum_{i=1}^{N} \mathbf{X}_i^T \left[\mathbf{Q}_h^T \left[\mathbf{G}_i\mathbf{G}_i^T - \mathbf{Q}_h\mathbf{X}_i\hat{\Psi}_h\mathbf{Q}_h^T\right]\mathbf{Q}_h\right] \tag{2}$$

$$\mathbf{0}_q = \sum_{i=1}^{N} \left[\mathbf{X}_i^T\mathbf{Q}_h^T\mathbf{G}_i\mathbf{G}_i^T\mathbf{Q}_h - \mathbf{X}_i^T\mathbf{Q}_h^T\mathbf{Q}_h\mathbf{X}_i\hat{\Psi}_h\mathbf{Q}_h^T\mathbf{Q}_h\right]$$

$$\mathbf{0}_q = \sum_{i=1}^{N} \left[\mathbf{X}_i^T\mathbf{Q}_h^T\mathbf{G}_i\mathbf{G}_i^T\mathbf{Q}_h - \mathbf{X}_i^T\mathbf{X}_i\hat{\Psi}_h\right]$$

$$\sum_{i=1}^{N} \left[\mathbf{X}_i^T\mathbf{X}_i\right]\hat{\Psi}_h = \sum_{i=1}^{N} \left[\mathbf{X}_i^T\mathbf{Q}_h^T\mathbf{G}_i\mathbf{G}_i^T\mathbf{Q}_h\right]$$

$$\mathbf{X}^T\mathbf{X}\hat{\Psi}_h = \sum_{i=1}^{N} \left[\mathbf{X}_i^T\mathbf{Q}_h^T\mathbf{G}_i\mathbf{G}_i^T\mathbf{Q}_h\right]$$

$$\hat{\Psi}_h = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \sum_{i=1}^{N} \left[\mathbf{X}_i^T\mathbf{Q}_h^T\mathbf{G}_i\mathbf{G}_i^T\mathbf{Q}_h\right] \tag{3}$$

$$\hat{\Psi} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \sum_{i=1}^{N} \left[\mathbf{X}_i^T\mathbf{Q}^T\mathbf{G}_i\mathbf{G}_i^T\mathbf{Q}\right]$$

Equation 3 provides an estimate for Ψ_h, a q-vector specifying the contribution of eigenvector h and the q covariates to the correlation structure in the N samples.

The estimate $\hat{\Psi}$ represents the least squares estimate for Ψ, which is equivalent to the maximum likelihood estimate under normal error. Given the generous assumption of a properly specified model, this estimate will be the most efficient estimator and will asymptotically converge to the true parameter Ψ.

This provides a closed form solution to our problem. Given that the computationally intensive steps involve matrix inversion, the computational complexity is $\mathcal{O}(n^3)$ or less, depending on the specific implementation. This allows for relatively fast computation of corrected coexpression matrix that is comparable to the simple Pearson correlation computation, which has similar complexity. Using a computer with Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz, and Microsoft R Open 3.2.5 linked with multi-threaded BLAS/LAPACK libraries, the R implementation of this method finished in 8.8 seconds on a dataset of 4000 genes, 400 samples and 2 covariates.

2.4 Corrected covariance matrix

With the estimates obtained with our method, it is straightforward to see how fitted values for the covariance matrix for each sample or experimental condition can be obtained. Using the usual interpretations of . Given an estimate for Ψ, $\hat{\Psi}$, we can now estimate the batch-independent covariance structure as

$$\hat{\mathbf{S}} = \mathbf{Q} diag\left(\bar{\mathbf{x}}\hat{\Psi}\right)\mathbf{Q}^T \text{ or } \hat{\mathbf{S}} = \sum_{i=1}^{p} \bar{\mathbf{x}}\hat{\Psi}_i \mathbf{Q}_i \mathbf{Q}_i^T \tag{4}$$

where $\bar{\mathbf{x}}$ is a q-vector specifying the column means of $\bar{\mathbf{x}}$,

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N}$$

The differential coexpression matrix between two conditions, defined in binary as column 2 of $\mathbf{X}$, is computed

$$\hat{\mathbf{W}} = \mathbf{Q} diag\left(\mathbf{v}\hat{\Psi}\right)\mathbf{Q}^T \tag{5}$$

where $\mathbf{v} = (0, 1, 0, \ldots 0)_q$

3 Results

3.1 *In Silico* analysis

We performed a simulation study to determine the relative performance of our method in identifying differential coexpression in the presence of coexpression batch effect. Gene expression for 400 samples were simulated across 4,000 genes. The simulation study contained a balance Cases/Control design with 200 samples per group. Similarly, "Batch A" and "Batch B" were each assigned 200 samples. To generate an unbalanced batch effect, 150/200 samples in Batch A were control group samples, whereas 150/200 samples in Batch B were cases.

Each gene was randomly assigned to one of 10 distinct modules, labeled A-J. Modules A,B were labeled as background modules with the coexpression pattern present in all samples, Module C was present in all Batch A samples, Module D was present in all Batch B samples, Module E was present in controls, Module F and G were present in cases. The coexpression pattern of all other modules were present in no samples. Within each module each gene was assigned a continuous value, γ_i, uniformly random between $-a$ and a. For case/control modules, a was chosen to be $\sqrt{0.1}$ and for all other modules a was set at $\sqrt{0.2}$. The true coexpression between any two genes was defined as $\rho_{i,j} = \gamma_i \gamma_j$. This yielded within module correlation values in the range $(-0.1, 0.1)$ for cases/controls and $(-0.2, 0.2)$ for batch and background modules. The average absolute correlation between two case-control coexpressed genes was $\rho = 0.025$ with $R^2 = 0.000625$. The average absolute correlation between two batch or background coexpressed genes was $\rho = 0.05$ with $R^2 = 0.0025$.

The simulated study was run by generating 400 samples from a multivariate normal distribution with 0-vector mean and covariance equal to the correlation matrix described above for each sample.

The eigenvectors obtained demonstrate the tendency to isolate distinct gene modules. Figure 3 shows this feature along with the pseudo-eigenvalue contribution of each covariate. It is important to note that the top eigenvectors do not necessarily identify genes of interest in the case/control context. The estimate $\hat{\Psi}$ is a $3 \times p$ matrix with the first 20 columns plotted in (Figure 3B). The i^{th} column and j^{th} row can be interpreted as the additional contribution of the i^{th} eigenvector for a 1 unit increase in the value of the j^{th} variable. This is analogous to standard regression, where we can identify the estimated mean differences associated with a change in a predictor. To identify differential coexpression for the j^{th} variable, such as case/control, controlling for batch we need only examine the values that deviate significantly from zero. The parameter corresponding to the case/control variable successfully finds the eigenvectors which best describe the genes differentially coexpressed across cases/controls.

We evaluated the ability of [NAME] to capture case/control differential coexpression relative to batch coexpression and background coexpression (Figure 4). For 4,000 genes there are 7,998,000 pairwise coexpression estimates of which 319,600 (4%) are considered case/control gene-pairs, and 159,600 (2%) are considered batch

3.2 ComBat-corrected expression data still contains batch-associated coexpression in ENCODE

Above, we outline a theoretical basis for adjusting for differential coexpression by batch. In short, we demonstrate how this particular form of batch-effect could, in theory, lead to reduced power and biased results. However, it remains to be seen whether this purported phenomenon actually occurs in real gene expression datasets. One might hope that the impact of batch on gene expression data occurs on each gene independently, altering the distribution of expression within each batch. In that scenario, existing approaches would be sufficient for removal of batch effect and batch-associated differential coexpression would be virtually absent.

To demonstrate that differential coexpression by batch exists in the wild, we selected publicly available data from the ENCODE project. This dataset (GSE19480) contains 153 RNA-seq samples across 57820 ENSEMBL IDs from lymphoblastoid cell lines obtained from Yoruban HapMap individuals. Reads were aligned using Bowtie [Langmead et al., 2009], and counts were produced using *featureCounts* from the *Subread* program [Liao et al., 2014]. Replicates were removed, yielding samples from 63 individuals who were each sequenced at both the Yale University and Argonne National Laboratory (126 samples in total). Both centers used the Illumina Genome Analyzer II, which helps reduce, but is known to not eliminate, batch effect. These two centers represent the two batches to consider for correction. Since each batch contains RNA-seq experiments on the same group of 63 individuals, one would hope that in the

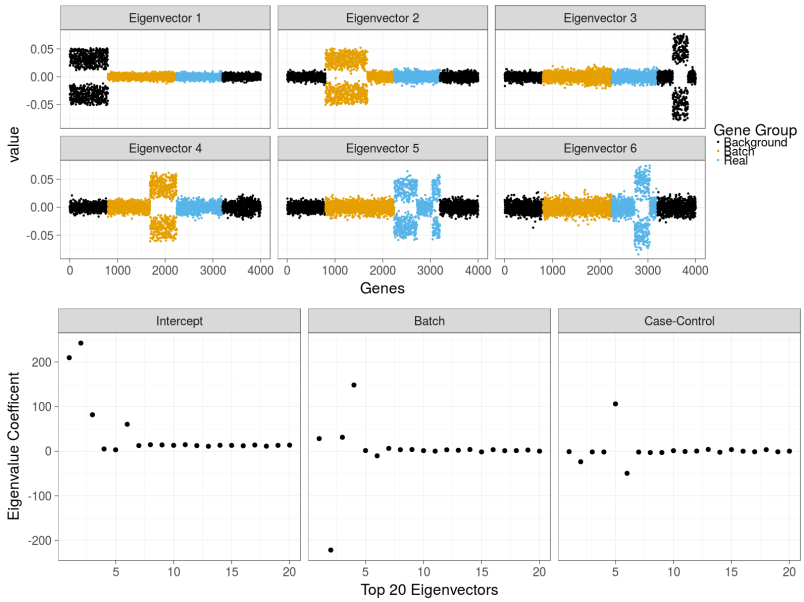

Figure 3: Eigenvector plots. [NAME] is designed to estimate sample specific coexpression as a function of the sample covariates and the overall coexpression eigenvectors. **(A)** Here we see the top six eigenvectors plotted for all 4000 genes. Each point is colored according to that gene's membership in a batch, case/control or background module. We see that the eigenvectors tend to separate along with coexpression modules. **(B)** Pseudo-eigenvalues for the top 20 eigenvectors corresponding to the three covariates (intercept, batch, case/control). Deviations from zero on the y-axis are indications of an unequal contribution of the corresponding eigenvector to the fitted coexpression estimate. Note that eigenvectors 5 and 6 have notable non-zero pseudo-eigenvalues corresponding to case/control parameter.

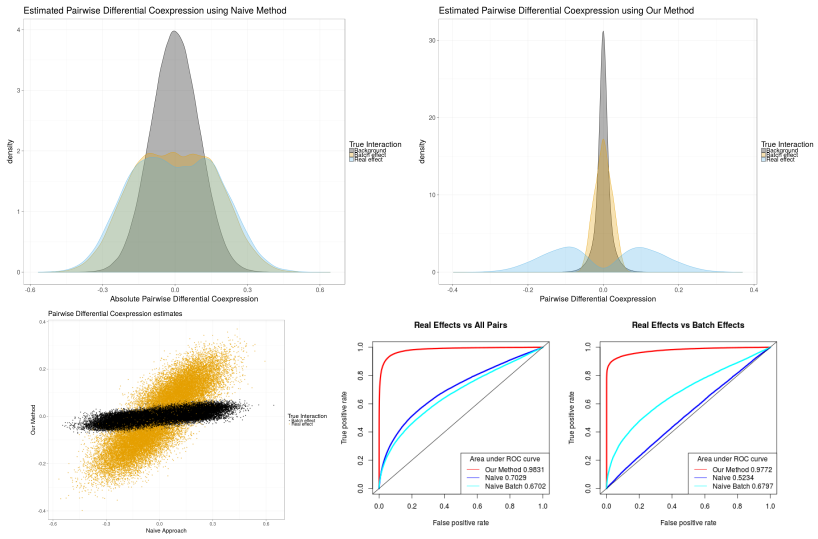

Figure 4: *In Silico* Results. Fitted differential coexpression scores for standard differential coexpression (upper left) vs [NAME] (upper right) separated by true relationship. Pearson difference failed to generate much power to predict true coexpression compared to background and found batch effect at approximately the same rate as true effect. [NAME] found true effects at vastly superior rates compared to both background and batch effect. The predicted scores of non-background genes for Pearson difference (x-axis) vs [NAME] (y-axis) demonstrate improved ability to separate case/control effects (orange) from batch effects (black). ROC curves show the relative performance in identifying case/control genes compared to background genes (left) and batch genes (right).

absence of batch effect (or in the presence of satisfactory batch correction) that there would be minimal differential expression and coexpression between the batches.

We first ran LIMMA on the uncorrected data between Yale and Argonne and observed 495 significant (FDR<.01) genes (Figure 5A). We then applied ComBat to the data with Yale/Argonne as the batch. ComBat uses an empirical bayes approach to make location/scale adjustments for each gene, returning a gene expression matrix of corrected values. We then reran LIMMA using the same Center partition and observed 43 (FDR<.01) significant genes. Unsurprisingly, this procedure removed the differential expression across batches which is critical from the removal of confounding effects in differential expression.

Next we examined the distribution of differential coexpression between the two batches (Figure 5B). We also generated null distributions by randomly swapping the two centers for each of the 63 individuals 1000 times. Interestingly, despite the absence of differential expression across batches, differential coexpression persists after batch correction.

3.3 [NAME] allows for separation of covariate specific modules with WGCNA in COPDGene study

Weighted Gene Coexpression Network Analysis (WGCNA) is one of the most popular network reconstruction methods in use today [Langfelder and Horvath, 2008], with 1730 citations as of March 31, 2017. It's use continues to grow with 140 citations in the first 3 months of 2017. Like many other methods in the field, WGCNA begins with a standard Pearson correlation matrix of gene expression data. We were interested in whether the use of [NAME] could provide covariate-specific differential coexpression estimates that could integrate with WGCNA to find functionally relevant coexpression modules. While our method is motivated by the idea of removing batch, it is general enough to be applied to any confounding variable. In this application, we chose to treat sex as a confounder because it is clearly labeled, dichotomous, and likely to impart some coexpression bias on the study due to sexual dimorphism.

Gene expression data from the ECLIPSE study (GSE54837) [Singh et al., 2014] was collected using blood samples from 226 subjects classified as non-smokers (6), smoker controls (84) or COPD (136). Blood samples from each individual were profiled using Affymetrix Human Genome U133 Plus 2.0 microarrays. CEL data files from these assays were RMA-normalized [Irizarry et al., 2003] in R using the BioConductor package 'affy' [Gautier et al., 2004]. Array probes were collapsed to 19,765 Entrez-gene IDs using a custom CDF [Dai et al., 2005] and the 220 samples for COPD or smoker control subjects were retained for analysis.

Previous work applying WGCNA to this data has identified network modules associated with COPD diagnosis [Morrow et al., 2015,Morrow et al., 2017]. These studies involved applying topological overlap to an overall similarity score (e.g. Pearson, Euclidean, biweight midcorrelation) after standard batch correction (Surrogate Variable Analysis [Leek and Storey, 2007]). The similarity

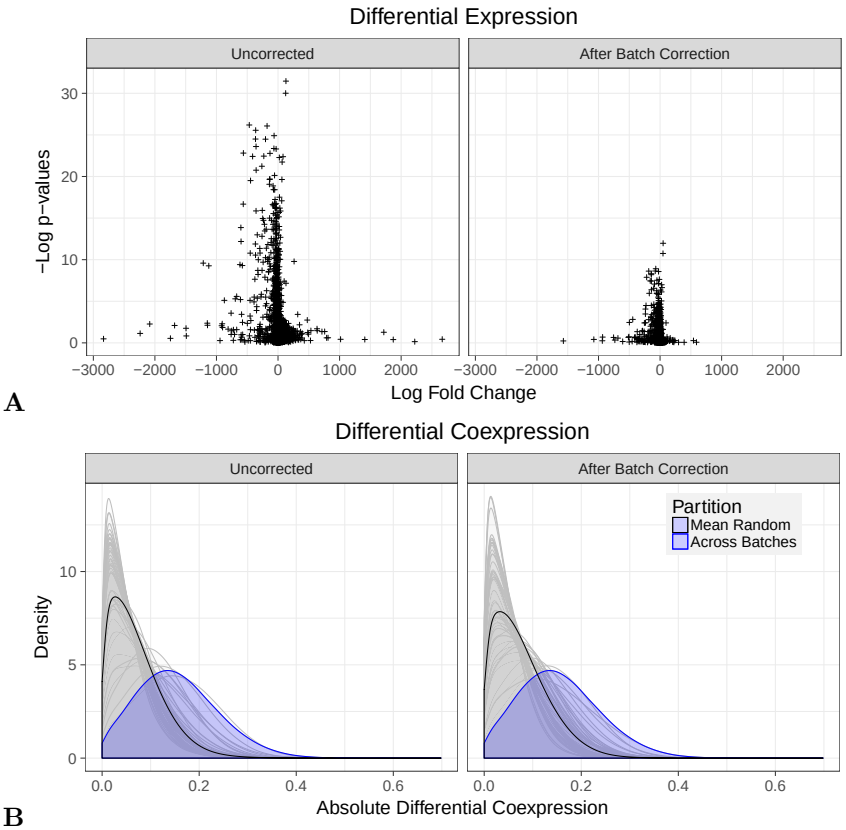

Figure 5: Differential expression and absolute differential coexpression in ENCODE data with batch correction. ComBat effectively mitigates the differential expression between samples run in two separate centers (**A**). However, with this same batch correction, differential coexpression continues to be strongly influenced by processing center. The lower plot (**B**) shows the distribution of differential coexpression when comparing groups that are randomly assigned (grey) compared to assignments based on batch. Despite the fact that ComBat helps mitigate differential expression between batches, these results show that batch-associated differential coexpression remains uncorrected.

matrix typically does not consider sample covariates and consequently yields a collection of modules which are generally coexpressed, not necessarily differentially coexpressed. To identify which modules might be relevant to phenotypes of interest, an eigendecomposition by samples of each module is performed and the top eigenvector (eigengene) is regressed against the phenotypes and other covariates. This approach, while effective at identifying associated modules has limitations. The eigengene obtained through this method will capture the greatest axis of variation across the samples, not the greatest axis of covariation. By design, the eigengene will only be associated with a phenotype of interest if there is differential expression within the module across phenotypes. Given the wide availability of methods for differential expression analysis, the greatest value coming from the investigation of coexpression necessarily focuses on discovery of genes and modules which are not differentially expressed. In any scenario where we wish to consider differential coexpression as a potential driver of disease needs to consider these concepts.

We applied [NAME] to the COPDGene data and included Sex as a covariate in the model.

The sex distribution for cases and controls in this study were uneven, potentially leading to confounded results. Using Equation 5 we generated $p \times p$ matrix interpreted as the differential correlation for the case-control partition, holding sex constant. We applied a soft thresholding power of 6 and computed the topological overlap matrix, as described in [Langfelder and Horvath, 2008]. Modules generated were analyzed for functional enrichment using DAVID [Huang et al., 2009a, Huang et al., 2009b].

[Edit this after new results come in] The top module was found to be enriched for mitochondrial translational elongation $\left(\text{adjusted } p = 7 \times 10^{-7}\right)$. Recent work has identified several mitochondrial mechanisms for disease progression [Cloonan et al., 2016, Cloonan and Choi, 2016].

Discussion

This manuscript makes two important contributions to gene correlation networks. First, we identify the problem of confounding by differential coexpression, provide a theoretical basis for that artifact and demonstrate its presence in real data. Second, we propose a method for estimating coexpression matrices in the context of covariates which serve as coexpression confounders.

Incremental improvements in high-throughput data collection have vastly increased the availability of large scale gene expression data. As we dive deeper into this data, we recognize that cellular states are rarely driven by the additive impacts of sets of suspect genes. Rather, it is the relationships, pairwise and higher, that these genes have with each other and their environment that leads to the phenotypes we seek to explain. Technological and methodological advancements in genomics allow us unprecedented ability to study these interactions. But with this new data come new statistical challenges that were not as impactful in differential expression analyses.

We argue that the batch correction methods that are designed for and are ubiquitous in differential expression are important, but not sufficient, for removing unwanted variation from the data in gene coexpression.

References

[Banerjee et al., 2008] Banerjee, O., Ghaoui, L. E., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine learning research*, 9(Mar):485–516.

[Benito et al., 2004] Benito, M., Parker, J., Du, Q., Wu, J., Xiang, D., Perou, C. M., and Marron, J. S. (2004). Adjustment of systematic microarray data biases. *Bioinformatics*, 20(1):105–114.

[Bien et al., 2011] Bien, J., Tibshirani, R. J., et al. (2011). Sparse estimation of a covariance matrix. *Biometrika*, 98(4):807.

[Chen et al., 2011] Chen, C., Grennan, K., Badner, J., Zhang, D., Gershon, E., Jin, L., and Liu, C. (2011). Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PloS one*, 6(2):e17238.

[Choi et al., 2005] Choi, J. K., Yu, U., Yoo, O. J., and Kim, S. (2005). Differential coexpression analysis using microarray data and its application to human cancer. *Bioinformatics*, 21(24):4348–4355.

[Cloonan and Choi, 2016] Cloonan, S. M. and Choi, A. M. (2016). Mitochondria in lung disease. *Journal of Clinical Investigation*, 126(3):809.

[Cloonan et al., 2016] Cloonan, S. M., Glass, K., Laucho-Contreras, M. E., Bhashyam, A. R., Cervo, M., Pabón, M. A., Konrad, C., Polverino, F., Siempos, I. I., Perez, E., et al. (2016). Mitochondrial iron chelation ameliorates cigarette smoke-induced bronchitis and emphysema in mice. *Nature medicine*.

[Conesa et al., 2016] Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., et al. (2016). A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):13.

[Dai et al., 2005] Dai, M., Wang, P., Boyd, A. D., Kostov, G., Athey, B., Jones, E. G., Bunney, W. E., Myers, R. M., Speed, T. P., Akil, H., et al. (2005). Evolving gene/transcript definitions significantly alter the interpretation of genechip data. *Nucleic acids research*, 33(20):e175–e175.

[de la Fuente, 2010] de la Fuente, A. (2010). From 'differential expression'to 'differential networking'–identification of dysfunctional regulatory networks in diseases. *Trends in genetics*, 26(7):326–333.

[Fare et al., 2003] Fare, T. L., Coffey, E. M., Dai, H., He, Y. D., Kessler, D. A., Kilian, K. A., Koch, J. E., LeProust, E., Marton, M. J., Meyer, M. R., et al. (2003). Effects of atmospheric ozone on microarray data quality. *ANALYTICAL CHEMISTRY-WASHINGTON DC-*, 75(17):4672–4675.

[Friedman et al., 2008] Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

[Fukushima, 2013] Fukushima, A. (2013). Diffcorr: an r package to analyze and visualize differential correlations in biological networks. *Gene*, 518(1):209–214.

[Furlotte et al., 2011] Furlotte, N. A., Kang, H. M., Ye, C., and Eskin, E. (2011). Mixed-model coexpression: calculating gene coexpression while accounting for expression heterogeneity. *Bioinformatics*, 27(13):i288–i294.

[Gautier et al., 2004] Gautier, L., Cope, L., Bolstad, B. M., and Irizarry, R. A. (2004). affy-analysis of affymetrix genechip data at the probe level. *Bioinformatics*, 20(3):307–315.

[Glass et al., 2013] Glass, K., Huttenhower, C., Quackenbush, J., and Yuan, G.-C. (2013). Passing messages between biological networks to refine predicted interactions. *PloS one*, 8(5):e64832.

[Hoff and Niu, 2012] Hoff, P. D. and Niu, X. (2012). A covariance regression model. *Statistica Sinica*, pages 729–753.

[Huang et al., 2009a] Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009a). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1):1–13.

[Huang et al., 2009b] Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009b). Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature protocols*, 4(1):44–57.

[Irizarry et al., 2003] Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264.

[Johnson et al., 2007] Johnson, W. E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127.

[Lander, 1999] Lander, E. S. (1999). Array of hope. *Nature genetics*, 21:3–4.

[Langfelder and Horvath, 2008] Langfelder, P. and Horvath, S. (2008). Wgcna: an r package for weighted correlation network analysis. *BMC Bioinformatics*, (1):559.

[Langfelder and Horvath, 2012] Langfelder, P. and Horvath, S. (2012). Fast R functions for robust correlations and hierarchical clustering. *Journal of Statistical Software*, 46(11):1–17.

[Langmead et al., 2009] Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome biology*, 10(3):R25.

[Leek et al., 2010] Leek, J. T., Scharpf, R. B., Bravo, H. C., Simcha, D., Langmead, B., Johnson, W. E., Geman, D., Baggerly, K., and Irizarry, R. A. (2010). Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739.

[Leek and Storey, 2007] Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet*, 3(9):e161.

[Liao et al., 2014] Liao, Y., Smyth, G. K., and Shi, W. (2014). featurecounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, 30(7):923–930.

[Meinshausen and Bühlmann, 2006] Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, pages 1436–1462.

[Morrow et al., 2015] Morrow, J. D., Qiu, W., Chhabra, D., Rennard, S. I., Belloni, P., Belousov, A., Pillai, S. G., and Hersh, C. P. (2015). Identifying a gene expression signature of frequent copd exacerbations in peripheral blood using network methods. *BMC medical genomics*, 8(1):1.

[Morrow et al., 2017] Morrow, J. D., Zhou, X., Lao, T., Jiang, Z., DeMeo, D. L., Cho, M. H., Qiu, W., Cloonan, S., Pinto-Plata, V., Celli, B., et al. (2017). Functional interactors of three genome-wide association study genes are differentially expressed in severe chronic obstructive pulmonary disease lung tissue. *Scientific Reports*, 7.

[Nygaard et al., 2016] Nygaard, V., Rødland, E. A., and Hovig, E. (2016). Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics*, 17(1):29–39.

[Scherer, 2009] Scherer, A. (2009). *Batch effects and noise in microarray experiments: sources and solutions*, volume 868. John Wiley & Sons.

[Schlauch et al., 2016] Schlauch, D., Glass, K., Hersh, C. P., Silverman, E. K., and Quackenbush, J. (2016). Estimating drivers of cell state transitions using gene regulatory network models. *bioRxiv*, page 089003.

[Singh et al., 2014] Singh, D., Fox, S. M., Tal-Singer, R., Bates, S., Riley, J. H., and Celli, B. (2014). Altered gene expression in blood and sputum in copd frequent exacerbators in the eclipse cohort. *PloS one*, 9(9):e107381.

[Siska and Kechris, 2017] Siska, C. and Kechris, K. (2017). Differential correlation for sequencing data. *BMC Research Notes*, 10(1):54.

[Southworth et al., 2009] Southworth, L. K., Owen, A. B., and Kim, S. K. (2009). Aging mice show a decreasing correlation of gene expression within genetic modules. *PLoS Genet*, 5(12):e1000776.

[Tesson et al., 2010] Tesson, B. M., Breitling, R., and Jansen, R. C. (2010). Diffcoex: a simple and sensitive method to find differentially coexpressed gene modules. *BMC bioinformatics*, 11(1):497.

[Yuan and Lin, 2007] Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, pages 19–35.