Batch effect on covariance structure confounds gene co-expression studies

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

March 6, 2017

Abstract

Systemic biases associated with multiple batches of gene expression experiments have been known to confound results in differential gene expression analyses. Numerous methods have been developed over the past 10 years which 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 potential differential covariance across batches. While this is of lesser concern in the context of standard differential gene expression, analyses that utilize a gene co-expression 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 a maximum likelihood estimation of the conditional covariance matrix. [NAME] controls for continuous and categorical confounders, estimates a reduced set of parameters, is computationally fast, and makes use of the inherently modular structure of features commonly found in genomic analyses.

1 Introduction

While the accessibility of high-throughput assays increases, so too has the 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, is dependent on phenotypes and other genomic features. It can be difficult to ascertain which associations are driven by real biological mechanisms and which associations are observed because of confounding by undesirable batch effects. It's critical to address this confounding in order to reduce the probability of false positive results.

Biological sources of variation are typically of interest, but observed variation is often the result of technical artifacts which may confound associations between experimental groups and gene expression. We can assume 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 $N\left(0, \sigma_j^2\right)$ 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. Many steps of experimental protocols have been shown to lead to batch effects, but it is generally not known what mechanism is at fault for a particular study. Therefore, without knowing which features are susceptible to batch effect, it is typical to estimate γ_{ij} and δ_{ij} for each gene in a study.

Despite widespread literature published regarding the identification and control of confounding due to batch effect [Chen et al.(2011)Chen, Grennan, Badner, Zhang, Gershon, Jin, and Liu, Benito et al.(2004)Benito, Parker, Du, Wu, Xiang, Perou, and Marron, Leek and Storey(2007), Johnson et al.(2007)Johnson, Li, and Rabinovic], 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)Johnson, Li, and Rabinovic] uses an empirical bayes approach to estimate the mean and variance parameters for each gene and then computes an adjusted gene expression which controls for these effects. Another approach, Surrogate Variable Analysis, uses a combination of measured covariates and singular value decomposition to identify unknown sources of variation.

However, in the context of network inference or coexpression analysis, we are often interested in the covariance of genes as opposed to the marginal distribution of each gene. Essentially, we assume that genes which 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.

In estimating coexpression matrices, standard batch correction is critical [Furlotte et al.(2011)Furlotte, Kang, Ye, and Eskin]. Confounding due to batch will reduce power, bias results and inevitably lead to highly significant, but biologically meaningless associations between large volumes of genes. Though existing approaches help mitigate this problem current approaches fail to remove the impact of the type of batch effect which may manifest itself by causing differential coexpression patterns in gene hubs. Removing the impact of differential means and variances across batches is critical to removing differential covariance across batch, but will be insufficient if the covariance itself is associated with batch. While numerous methods consider the correlation of genes in adjusting for batch, no method that we are aware of allows for that correlation to differ according to sample covariates. Similarly, no method currently available applies batch correction directly to the estimated coexpression matrix.

While the impact of this oversight may be negligible for simple differential gene expression analyses, coexpression patterns are widely considered in the field of network inference. The impact of confounding due to differential coexpression in batches remains unexamined.

Estimation of a coexpression matrix which is adjusted by batches or other covariates requires us to allow the gene coexpression to vary by sample. This produces at least two major challenges. The first problem is that it requires the estimation of additional $p \times p$ matrices in a context that is already suffering from the curse of dimensionality. For $n \ll p$, as is the case for high throughput gene expression studies, the fact that $\binom{p}{2}$ parameters must estimated per coexpression matrix is undesirable. Recent work has allowed for the imposition of sparsity on the gene covariance matrix [Bien et al.(2011)Bien, Tibshirani, et al.] or precision matrix [Friedman et al.(2008)Friedman, Hastie, and Tibshirani], but the complexity of biological systems make it an imperfect choice for sparsity and the 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 estimating p weights for eigenvectors rather than pairwise coexpressions. This exploits the modular nature of genomic features, effectively borrowing information from similarly behaved genes to estimate gene coexpression as a function of sample covariates. Our method is presented in a regression framework which allows for the inclusion of continuous and categorical covariates into the adjustment model.

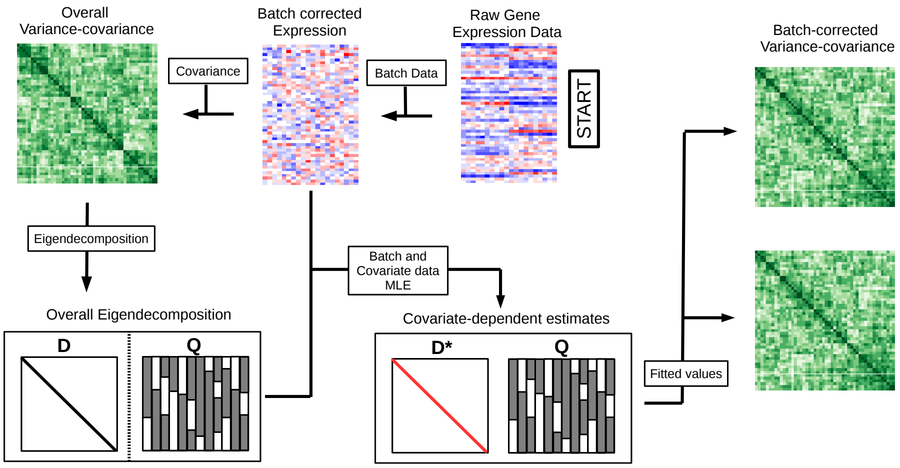

Overall Variance-covariance
Covariance
Batch corrected Expression
Batch Data
Raw Gene Expression Data
START
Batch-corrected Variance-covariance
Eigendecomposition
Batch and Covariate data MLE
Overall Eigendecomposition
D
Q
Covariate-dependent estimates
D*
Q
Fitted values

Figure 1: Workflow of **METHOD_NAME**. Draft 1. Better space usage, less info.

2 Methods

2.1 Approach

The conventional batch correction model is typically given as

$$Y_g = \alpha_g + \beta_g X + \gamma_i g Z + \delta_i g \epsilon_i g$$

where X is the exposure (e.g. treatment/control) and Z is the batch (or other covariates). In the context of network inference, we often want to find $cor(Y_{g1}, Y_{g2})$, independent of Z.

So, in order to model 2nd order batch, what we really want to do is allow for the parameter of interest, β_g to vary by batch. So, now we set

$\beta^*_g = \beta_g + \beta_B g Z$

Where β_B is a new parameter that we need to estimate for each of the $\binom{p}{2}$ comparisons.

We can write out a full model for any two genes. Note that Y_{g2} is another gene in this model.

$$Y_{g2} = \alpha_g + \beta^*_g X + \gamma_i g Z + \delta_i g \epsilon_i g$$

or

$$Y_{g2} = \alpha_g + (\beta_g + \beta_B g Z) X + \gamma_i g Z + \delta_i g \epsilon_i g$$

$$Y_{g2} = \alpha_g + \beta_g X + \beta_B g Z X + \gamma_i g Z + \delta_i g \epsilon_i g$$

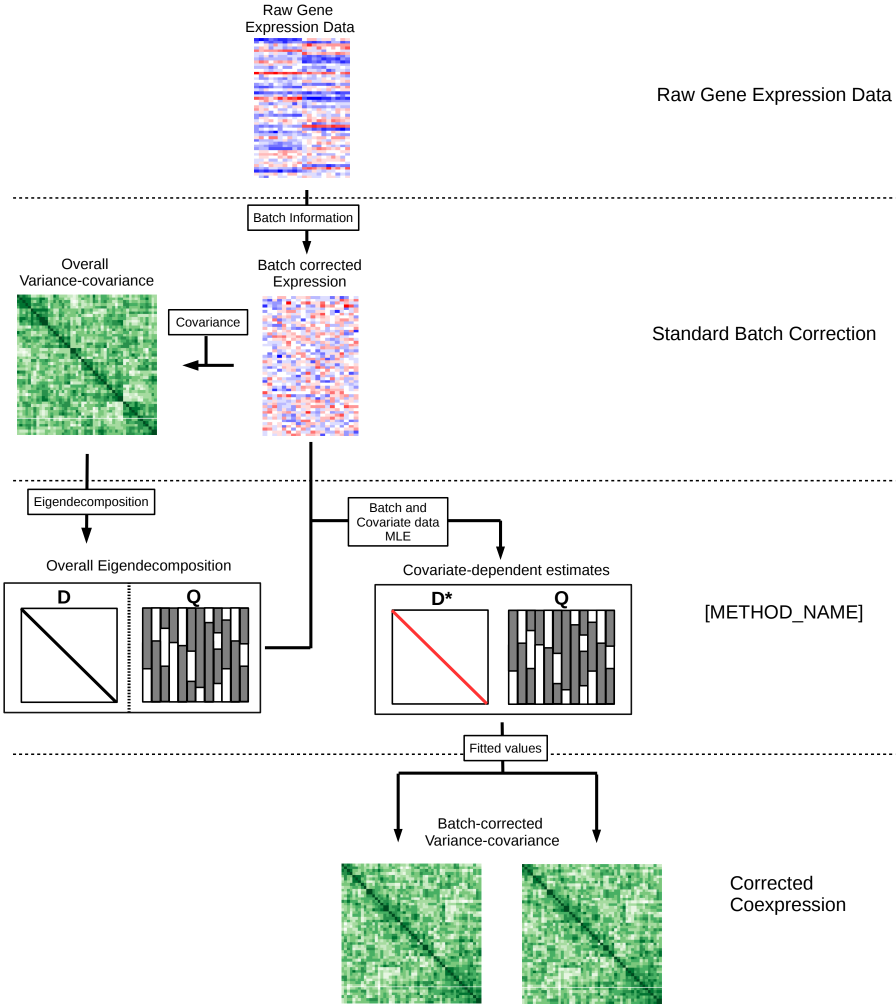

Raw Gene Expression Data
Raw Gene Expression Data
Batch Information
Overall Variance-covariance
Batch corrected Expression
Covariance
Standard Batch Correction
Eigendecomposition
Batch and Covariate data MLE
Overall Eigendecomposition
Covariate-dependent estimates
D
Q
D*
Q
[METHOD_NAME]
Fitted values
Batch-corrected Variance-covariance
Corrected Coexpression

Figure 2: Workflow of **METHODNAME**. Draft 2. Possibly more info, less efficient use of space.

There are many ways to approach this, but I believe the best way is the following steps:

1. Apply conventional batch correction. This will effectively eliminate the $\gamma_i g Z$ term and we can proceed with the simpler model -

$$Y_{g2} = \alpha_g + \beta_g X + \beta_B ZX + \delta_i g \epsilon_i g$$

- on the combat-corrected data. Further standardize each gene expression (this will not impact the actual results, but will aid in interpretation and computation time)

2. Fit the following models Reduced:

$$Y_{g2} = \alpha_g + \beta_g X + \delta_i g \epsilon_i g$$

Full:

$$Y_{g2} = \alpha_g + \beta_g X + \beta_B ZX + \delta_i g \epsilon_i g$$

3. Place estimated coefficients into two separate matrices (

$$S_\beta, S_B$$

). We have tons of options for computing these coefficients. A LASSO-style L1 regularization would probably make the most sense here, but for the purposes of simplicity we will start with OLS.

So, now we have two separate (equal sized) matrices instead of the usual one. S_β is the estimated similarity matrix and S_B is the "batch impact". Intuitively, we can imagine that the expected value of S_B is a zero matrix in the absence of 2nd order batch effects. This lends itself easily for 2nd order batch effect testing - for example, we can compare the two models via likelihood ratio test (LRT). This is nice, but we're much more interested in 2nd order batch effect *correction*.

4. Compute the corrected similarity matrix via:

$$\hat{S_i^*} = \hat{S_\beta} + (\frac{\sum_{j \in X_i} Z_j}{n_i})\hat{S_B}$$

This yields a similarity matrix that is **batch-independent**. In other words, we can now compare networks computed with different proportions of batch membership. We can think of the adjusted similarity matrix as being the estimated similarity matrix given a *standardized representation of batches*. This standardization allows us to compare networks which have been inferred with differing batch composition.

Obviously, the usual caveats apply - this correction is most useful when the batches in each exposure are (a) unequal, (b) not too unequal. Small numbers of samples for batches will result in wild fluctuations in terms of estimating batch effect.

2.2 Model

Consider a set of N samples with q covariates measuring gene expression across p genes. Let $\mathbf{x}_i = (x_{i1}, \ldots, x_{iq})$ denote the confounding 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 independent, and follow distribution, $MVN_p(\mathbf{0}, \Sigma_i)$. 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 & Buhlmann (2006)], or perform penalized maximum likelihood estimation with the penalty on the inverse covariance matrix[Yuan & Lin (2007), Friedman (2007), Banerjee (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 & Tibshirani 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 which 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. Given that 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 maximize the likelihood function below. 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.

Instead of estimating all entries in the covariance matrix, Σ_i, we instead 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 covariance matrix. In this formulation,$\mathbf{\Lambda}$ is a diagonal matrix with entries

$$\mathbf{\Lambda}_{kk}^{(i)} = \mathbf{X}_i\gamma_{\cdot k} \tag{1}$$

where $\mathbf{X}$ is the covariate matrix andγ is a $p \times q$ matrix of coefficients.

Let M_j be a $p \times p$ matrix with 1 at position (j, j) and 0 at all other positions and let $\mathbf{v}_j$ be a $p - vector$ of 0s except for a 1 at position j. Also, let s be a positive integer with $s \leq p$. We can express $\mathbf{\Lambda}^{(i)}$ as

$$\mathbf{\Lambda}^{(i)} = \sum_{j=1}^{s} M_j\gamma\mathbf{X}_i\mathbf{v}_j^T \tag{2}$$

The value taken with q specifies the number of pseudo-eigenvalues which are to be estimated. It is straightforward to show that in the case of a single batch, where $\mathbf{X} = \mathbf{1}_N$, and with $q = p$,γ becomes the vector of eigenvalues from the original covariance matrix.

2.3 Likelihood function

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

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

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 rowmeans yielding $\mathbf{G}_i^* = \mathbf{G}_i - \bar{\mathbf{g}}$. And plugging in our index dependent covariance matrix from equation 2 we have

$$\mathcal{L}(\gamma) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{p}{2}} |\mathbf{Q}\sum_{j=1}^{s}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{G}_i^*)^T \left(\mathbf{Q}\sum_{j=1}^{s}[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T]\mathbf{Q}^T\right)^{-1} (\mathbf{G}_i^*)}$$

and the log-likelihood is:

$$\begin{aligned}
log\mathcal{L}\left(\gamma\right) \propto & \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)\right) \\
& -\frac{1}{2}\sum_{i=1}^{N}\left[\mathbf{G}_i^{*T}\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\right] \\
= & \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)\right) \\
& -\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right] \text{ The trace trick}
\end{aligned}$$

$$\begin{aligned}
& -\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\left[\gamma+d\gamma\right]\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right] \\
& -\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\left[\gamma+d\gamma\right]\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right] \\
& -\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T+\mathbf{Q}diag\left(\mathbf{X}_i d\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right] \\
& -\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T+\mathbf{Q}diag\left(\mathbf{X}_i d\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right]+\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}\right. \\
& =-\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T+\mathbf{Q}diag\left(\mathbf{X}_i d\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}-\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right] \\
& =-\frac{1}{2}\sum_{i=1}^{N} tr\left[\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T+\mathbf{Q}diag\left(\mathbf{X}_i d\gamma\right)\mathbf{Q}^T\right)^{-1}-\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1}\right]\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right] \\
& =-\frac{1}{2}\sum_{i=1}^{N} tr\left[\left[-\left(I_p+\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{Q}diag\left(\mathbf{X}_i d\gamma\right)\mathbf{Q}^T\right)^{-1}\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{Q}diag\left(\mathbf{X}_i d\right.\right.\right. \\
& -(I+A-1B)-1A-1BA-1
\end{aligned}$$

See **http://math.stackexchange.com/questions/17776/inverse-of-the-sum-of-matrices**

$$\begin{aligned}
dln\mathcal{L}\gamma =& \frac{-1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1} diag\left(\sum_{k=1}^{q}\mathbf{X}_{ik}\right)\right] + \frac{1}{2}\sum_{i=1}^{N}\left[\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1} diag\left(\sum_{k=1}^{q}\mathbf{X}_{ik}\right)\left(\mathbf{Q}d\right.\right. \\
0 =& \frac{-1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1} diag\left(\sum_{k=1}^{q}\mathbf{X}_{ik}\right) - \sum_{i=1}^{N}\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1} diag\left(\sum_{k=}^{q}\right.\right.\right. \\
0 =& \sum_{i=1}^{N} tr\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T diag\left(\sum_{k=1}^{q}\mathbf{X}_{ik}\right)d\gamma\right)^{-1}\left(I_p - \mathbf{G}_i^*\mathbf{G}_i^{*T}\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1}\right)\right] \\
I_p =& \sum_{i=1}^{N}\left(\mathbf{Q}diag\left(\mathbf{X}_i\gamma\right)\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^*\mathbf{G}_i^{*T} \\
1 =& \\
\\
\hat{\gamma} =& \mathbf{G_i^*G_i^{*T}}
\end{aligned}$$

$$\begin{aligned}
log\mathcal{L}\left(\gamma\right) \propto & \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)\right) \\
& -\frac{1}{2}\sum_{i=1}^{N}\left[\mathbf{G}_i^{*T}\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\right] \\
= & \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)\right) \\
& -\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right] \text{ The trace trick} \\
= & \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\mathbf{Q}\right)det\left(\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\right)det\left(\mathbf{Q}^T\right)\right)-\frac{1}{2}\sum_{i=1}^{N} tr \\
= & \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\right)\right)-\frac{1}{2}\sum_{i=1}^{N} tr\left[\mathbf{G}_i^{*T}\mathbf{G}_i^{*}\left(\mathbf{Q}\sum_{j=1}^{p}\right.\right. \\
\frac{\partial}{\partial\gamma}log\mathcal{L}\left(\gamma\right) = & \frac{-1}{2}\sum_{i=1}^{N} tr\left[\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\sum_{j=1}^{p}\left[M_j\mathbf{X}_i\mathbf{v}_j^T\right]^T\right]+\frac{1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}\sum_{j=1}^{p}\right.\right. \\
= & \frac{-1}{2}\sum_{i=1}^{N} tr\left[\sum_{j=1}^{p}\left[M_j\mathbf{X}_i^T\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\right]+\frac{1}{2}\sum_{i=1}^{N}\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)\right. \\
= & \frac{-1}{2}\sum_{i=1}^{N}\left[\sum_{j=1}^{p}\left[\mathbf{X}_i^T\gamma\mathbf{X}_i\right]\right]+\frac{1}{2}\sum_{i=1}^{N}\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\left(\mathbf{G}_i^{*}\right.\right. \\
= & \frac{-1}{2}\sum_{i=1}^{N}\left[\sum_{j=1}^{p}\left[\mathbf{X}_i^T\gamma\mathbf{X}_i\right]\right]+\frac{1}{2}\sum_{i=1}^{N}\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\left(\mathbf{G}_i^{*}\right.\right. \\
= & \frac{-1}{2}\sum_{i=1}^{N}\left[\sum_{j=1}^{p}\left[\mathbf{X}_i^T\gamma\mathbf{X}_i\right]\right]+\frac{1}{2}\sum_{i=1}^{N}\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\left(\mathbf{G}_i^{*}\right.\right. \\
\sum_{i=1}^{N}\left[\sum_{j=1}^{p}\left[\mathbf{X}_i^T\gamma\mathbf{X}_i\right]\right] = & \sum_{i=1}^{N}\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\left(\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right)\left(\mathbf{Q}^{-T}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right.\right.\right.
\end{aligned}$$

$$\begin{aligned}
0 = & \sum_{i=1}^{N}\left[tr\left[\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\sum_{j=1}^{p}\left[M_j\mathbf{X}_i\mathbf{v}_j^T\right]^T\right]+\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right.\right.\right.\right. \\
0 = & \sum_{i=1}^{N}\left[tr\left[\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right]\sum_{j=1}^{p}\left[M_j\mathbf{X}_i\mathbf{v}_j^T\right]^T\right]+\left[\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\gamma\mathbf{X}_i\mathbf{v}_j^T\right.\right.\right.\right.
\end{aligned}$$

$$\begin{aligned}
log\mathcal{L}\left(\gamma\right) =& \frac{-1}{2}\sum_{i=1}^{N} log\left(\prod_{j=1}^{p}\left[\mathbf{X}_i\gamma_{\cdot j}\right]\right) - \frac{1}{2}\sum_{i=1}^{N} tr\left[\mathbf{G}_i^{*T}\mathbf{G}_i^{*}\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\right] \text{ det of diagonal mat} \\
=& \frac{-1}{2}\sum_{i=1}^{N}\sum_{j=1}^{p} log\left(\mathbf{X}_i\gamma_{\cdot j}\right) - \frac{1}{2}\sum_{i=1}^{N} tr\left[\mathbf{G}_i^{*T}\mathbf{G}_i^{*}\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\right] \text{ algebra} \\
=& \frac{-1}{2}\sum_{i=1}^{N}\left[\sum_{j=1}^{p} log\left(\mathbf{X}_i\gamma_{\cdot j}\right) - tr\left[\mathbf{G}_i^{*T}\mathbf{G}_i^{*}\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\right]\right] \text{ algebra} \\
\frac{\partial}{\partial\gamma} log\mathcal{L}\left(\gamma\right) =& \frac{-1}{2}\sum_{i=1}^{N}\left[\frac{\partial}{\partial\gamma}\sum_{j=1}^{p} log\left(\mathbf{X}_i\gamma_{\cdot j}\right) + \left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\left(\sum_{i=1}^{N}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right)\left(\mathbf{Q}\sum_{j=1}^{p}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}\right.\right.
\end{aligned}$$

$$d\, log\mathcal{L}\left(\gamma\right) = \frac{-1}{2}\sum_{i=1}^{N} tr\left[\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\right] - \frac{1}{2}\left[-\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\left(\sum_{i=1}^{N}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right)\right.$$

Setting $\mathbf{U}_i = \mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T$ and applying the chain run for differentiating with respect to γ yields

$$\begin{aligned}
log\mathcal{L}\left(\mathbf{U}\right) &= \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\mathbf{U}_i\right)\right) - \frac{1}{2}\sum_{i=1}^{N} tr\left[\mathbf{G}_i^{*}\mathbf{U}_i^{-1}\mathbf{G}_i^{*T}\right] \\
&= \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\mathbf{U}_i\right)\right) - \frac{1}{2}\sum_{i=1}^{N} tr\left[\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\mathbf{U}_i^{-1}\right] \\
&= \frac{-1}{2}\sum_{i=1}^{N} log\left(det\left(\mathbf{U}_i\right)\right) - \frac{1}{2}tr\left(\sum_{i=1}^{N}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\mathbf{U}_i^{-1}\right) \\
\frac{\partial log\mathcal{L}\left(\mathbf{U}\right)}{\partial\mathbf{U}} &= \frac{-1}{2}\sum_{i=1}^{N} tr\left[\mathbf{U}^{-1}\right]\frac{\partial\mathbf{U}}{\partial\gamma} - \frac{1}{2}\left[-\mathbf{U}^{-1}\left(\sum_{i=1}^{N}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}\right)\mathbf{U}^{-1}\right]\frac{\partial\mathbf{U}}{\partial\gamma} \\
\mathbf{U} &= \sum_{j=1}^{s}\left[\mathbf{Q}M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\mathbf{Q}^T\right] \\
\frac{\partial\mathbf{U}}{\partial\gamma} &= \sum_{j=1}^{s}\left[\mathbf{X}_i^T M_j^T\mathbf{Q}^T\mathbf{Q}\mathbf{v}_j\right] \\
\frac{\partial\mathbf{U}}{\partial\gamma} &= \sum_{j=1}^{s}\left[\mathbf{X}_i^T M_j^T\mathbf{v}_j\right] \\
\frac{\partial log\mathcal{L}\left(\gamma\right)}{\partial\gamma} &= \frac{-1}{2}\sum_{i=1}^{N} tr\left[\left(\sum_{j=1}^{s}\left[\mathbf{Q}M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\mathbf{Q}^T\right]\right)^{-1}\right]\sum_{j=1}^{s}\left[\mathbf{X}_i^T M_j^T\mathbf{v}_j\right] - \\
&\quad \frac{1}{2}\left[-\left(\sum_{j=1}^{s}\left[\mathbf{Q}M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\mathbf{Q}^T\right]\right)^{-T}\left(\sum_{i=1}^{N}\mathbf{G}_i^{*T}\mathbf{G}_i^{*}\right)^{T}\left(\sum_{j=1}^{s}\left[\mathbf{Q}M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\mathbf{Q}^T\right]\right)^{-T}\right]\sum_{j=1}^{s}\left[\mathbf{X}_i^T M_j^T\mathbf{v}_j\right]
\end{aligned}$$

Setting equal to 0 and removing terms we get

$$\left(\sum_{j=1}^{s}\left[\mathbf{Q}M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\mathbf{Q}^T\right]\right)^{-T}\left(\sum_{i=1}^{N}\mathbf{G}_i^{*T}\mathbf{G}_i^{*}\right)^{T}\left(\sum_{j=1}^{s}\left[\mathbf{Q}M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\mathbf{Q}^T\right]\right)^{-T} = \sum_{i=1}^{N} tr\left[\left(\sum_{j=1}^{s}\left[\mathbf{Q}M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\mathbf{Q}^T\right]\right)^{-}\right.$$

==========

$$
\begin{aligned}
d\,log\mathcal{L}\left(\gamma\right) = & \frac{-1}{2}\sum_{i=1}^{N} tr\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1} d\gamma \\
& -\frac{1}{2}tr\left[-\sum_{i=1}^{N}\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1} d\gamma\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\mathbf{G}_i^{*T}\mathbf{G}_i^{*}\right]
\end{aligned}
$$

To find the maximum, we set $d\,log\mathcal{L}\left(\gamma\right) = 0$. This is satisfied when

$$
0 = -\frac{1}{2}tr\left(\sum_{i=1}^{N}\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1} d\gamma\left(I_p-\sum_{i=1}^{N}\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right)^{-1}\sum_{i=1}^{N}\mathbf{G}_i^{*T}\mathbf{G}_i^{*}\right)\right)
$$

This is satisfied when the right term is 0, or when

$$
\sum_{i=1}^{N}\left(\mathbf{Q}\sum_{j=1}^{s}\left[M_j\mathbf{X}_i\gamma\mathbf{v}_j^T\right]\mathbf{Q}^T\right) = \sum_{i=1}^{N}\mathbf{G}_i^{*}\mathbf{G}_i^{*T}
$$

Answer:

$$
\begin{aligned}
\hat{\gamma} &= \mathbf{Q}^{-1}\tilde{\mathbf{G}}diag\left(\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\right)\tilde{\mathbf{G}}^T\mathbf{Q}^{-T} \\
\hat{\gamma}_i &= \mathbf{Q}^{-1}\tilde{\mathbf{G}}diag\left(\left[\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\right]_i\right)\tilde{\mathbf{G}}^T\mathbf{Q}^{-T}
\end{aligned}
$$

2.4 Why this solution is awesome

1. This solution is closed form.
   (a) Can be implemented with linear algebra.
   (b) Has an exact answer.
   (c) Is computationally superfast... (2000 genes, 400 samples runs in 11.6s!)
2. This solution is the maximum likelihood estimate
   (a) Sufficiency: The mle depends on the sample observations only through the value of a sufficient statistic.
   (b) Invariance: The maximum likelihood estimate is invariant under functional transformations. That is, if $T = t(X_1, \ldots, X_n)$ is the mle of θ and if $u(\theta)$ is a function of θ, then $u(T)$ is the mle of $u(\theta)$.
   (c) Consistency: The maximum likelihood estimator is consistent.
   (d) Efficiency: If there is a MVB estimator of θ, the method of maximum likelihood will produce it.
   (e) Asymptotic Normality

2.5 Corrected covariance matrix

Given an estimate for γ, $\hat{\gamma}$, we can now estimate the batch-independent covariance structure as

$$\hat{\mathbf{S}} = \mathbf{Q} diag \left(\bar{\mathbf{X}} \hat{\gamma} \right) \mathbf{Q}^T$$

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

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

Computing the differential covariance matrix between two conditions, defined as column 2 of $\mathbf{X}$, can be performed via

$$\hat{\mathbf{W}} = \mathbf{Q} diag \left(\mathbf{y} \hat{\gamma} \right) \mathbf{Q}^T$$

where $\mathbf{y} = \left(0, 1, 0, \ldots 0\right)_q$

3 Results

3.1 Simulated Demonstration

Consider a set of samples from a study involving two batches. Let the expression data be distributed as a set of MVN with mean vector μ and covariance structure Σ.

Existing batch correction methods are designed to identify differentially expressed genes and thus focus on μ and the diagonal of Σ.

For network inference, however, we are less interested in these values and more interesting in the off-diagonal of Σ. Mechanisms for batch effect which act on the covariance structure rather than the mean and variance structure will not be corrected for using existing batch correction methods.

In Figure 4, we see 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. 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.

To demonstrate the general concept of how differential coexpression leads to inflated test statistics, we simulated two studies, each of 200 individuals across 10,000 genes. Each sample was a realization of a multivariate normal distribution with $\mu = \mathbf{0}_p$, and borrowed a coexpression matrix of 10,000 genes from an existing dataset (ECLIPSE) to be $\mathbf{\Sigma}$. For Study 1, we sampled 200 times from this distribution. For Study 2, we sampled 150 times using Σ, and then estimated a new Σ^* using 9000 of the same genes from the ECLIPSE study, but with 1000 new genes. This procedure was intended to mimic the introduction of newly active biological pathways in a subset of the samples as we might expect

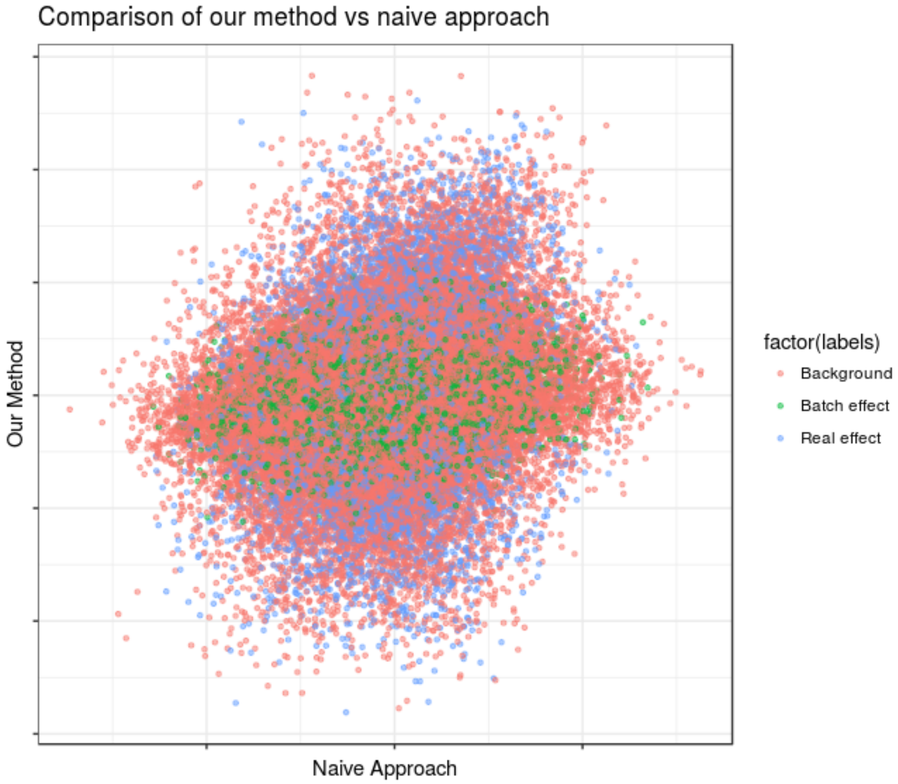


Figure 3: Separation of rewired ECLIPSE data

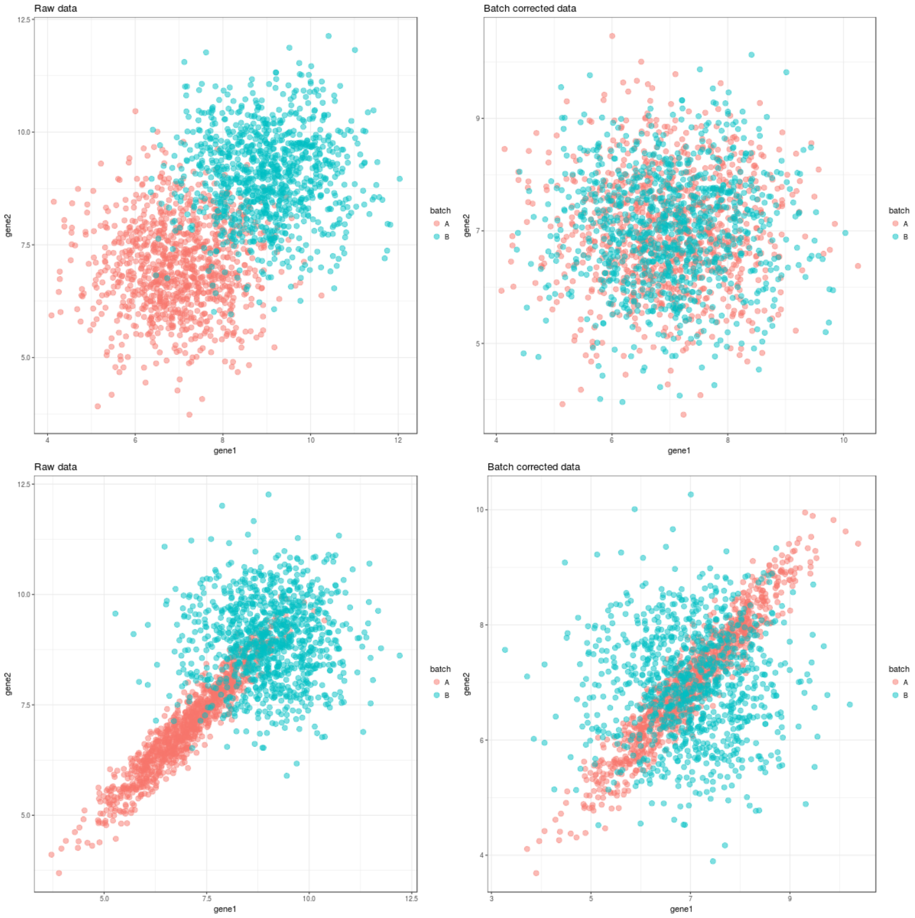


Figure 4: Example of what works and doesn't work for batch effect

with batch effect. Our goal was to observe how the distribution of the estimated coexpression matrix differed according to the different covariance approaches. The results are seen in Figure 2, showing an inflation of significant results... **[describe this in more detail]**

Second order batch effect in GTEx clearly exists (see above), but it can be subtle. For the purposes of a proof on concept, we can generate a much stronger batch effect which clearly demonstrates this approach. We simply select 5000 genes and 5000 separate genes in the other batch, relabeling them Gene1, Gene2,. . . , Gene5000. Basically, we're simply mismatching the genes between the batches! This, obviously, causes a completely random rewiring of the network from one batch to the other.

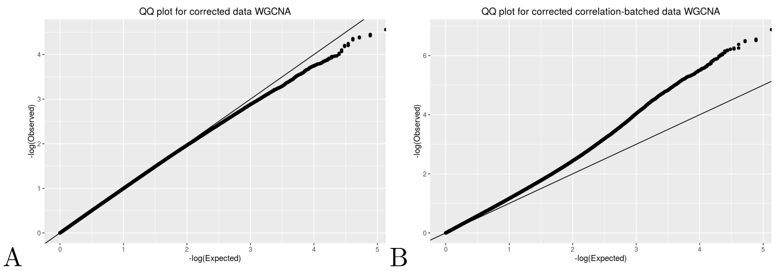


Figure 5: QQ plots for simulated data without differential covariance (A) and with differential covariance (B).

3.2 Batch effect in GTEx Project

GTEx uses WGCNA to find common modules across tissues. [Describe GTEx]

GTEx Consortium uses state-of-the-art methods for attempting to adjust for batch including study design and expression value correction.

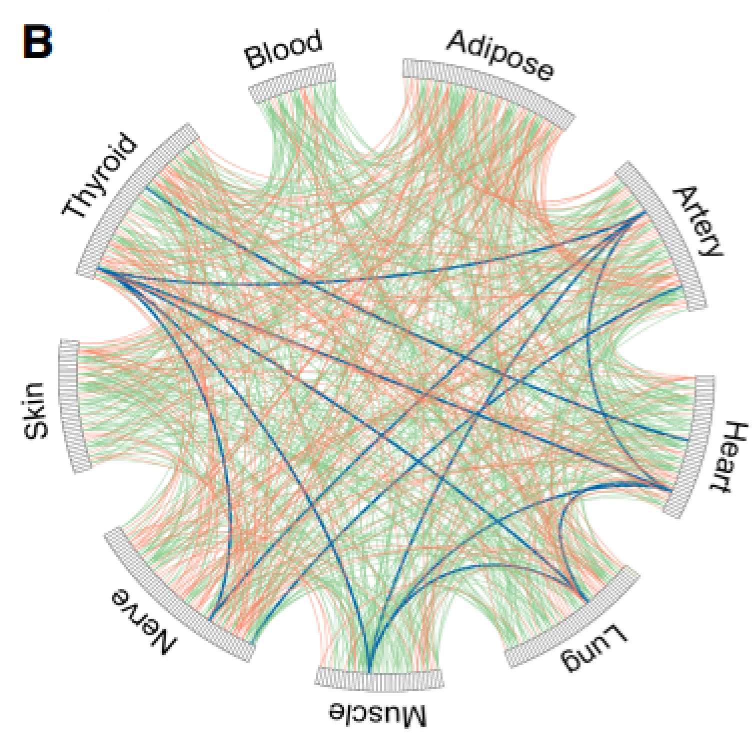
Study design:

"To the extent possible, based on sample availability, batches for library construction were designed to include a range of samples from different tissues and to span multiple donors, so as to minimise donor and tissue batch effects."

Expression correction:

"the effect of top 3 PEER factors, gender, and 3 genotype PCs were removed."

However, neither of these address the "second order" batch issue. These corrections inherently assume that batch affects the location-scale distribution of gene expression independently and thus does not consider the scenario where coexpression is the feature impacted by batch.



[GTEx Supplement] [Reproduce this figure or remove it]

To demonstrate that this study is sensitive to batch, we choose two widely available tissue types (Blood and Lung) from the data [YARN normalization].

We then apply batch correction using "Center" as the batch of interest. We then ran WGCNA on each tissue separately as described in [GTEx paper] and compute modules based on Topological Overlap Map.

We then ran the same procedure, but subsetted the data by each of the 3 centers. In theory, after correcting for batch, the modules observed should have been independent of the batch used to find them. However, we observe dramatic (notable? clear?) differences in the modules identified. We should really quantify this difference somehow. This may require a clever resampling scheme where we select samples both randomly and by center to measure variability.

3.3 Confounding due to sex in ECLIPSE study

In this analysis, we will perform a very common analysis: Build coexpression networks based on *COPD* vs *Smoker control* and identify consensus modules with WGCNA. *Gender* is treated as a confounder, but is only corrected using standard batch correction methods.

1. Run ComBat on gene expression data, *including gender as a covariate*.

2. Sample a set of designs from this study which include varying degrees of gender imbalance.

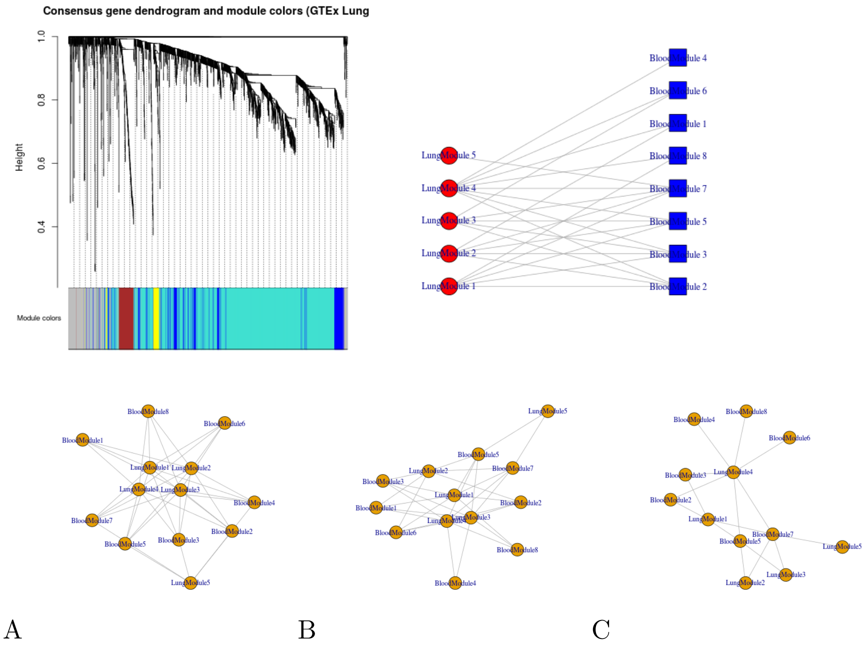


Figure 6: Make these nicer. Combine into single plot?

3. Evaluate the agreement between cases and controls with pseudo-R^2 from multinomial logistic regression.

4. Determine the degree to which the agreement between cases and controls depends on the gender distribution.

While this data is reasonably balanced, we can use it to demonstrate how sensitive our results are to confounding that was *supposedly* accounted for already. Figure 7

Discussion

Thoughts about impact on any analysis involving coexpression with batches...

Thoughts about generality of estimating covariance matrices in the context of confounding.

References

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

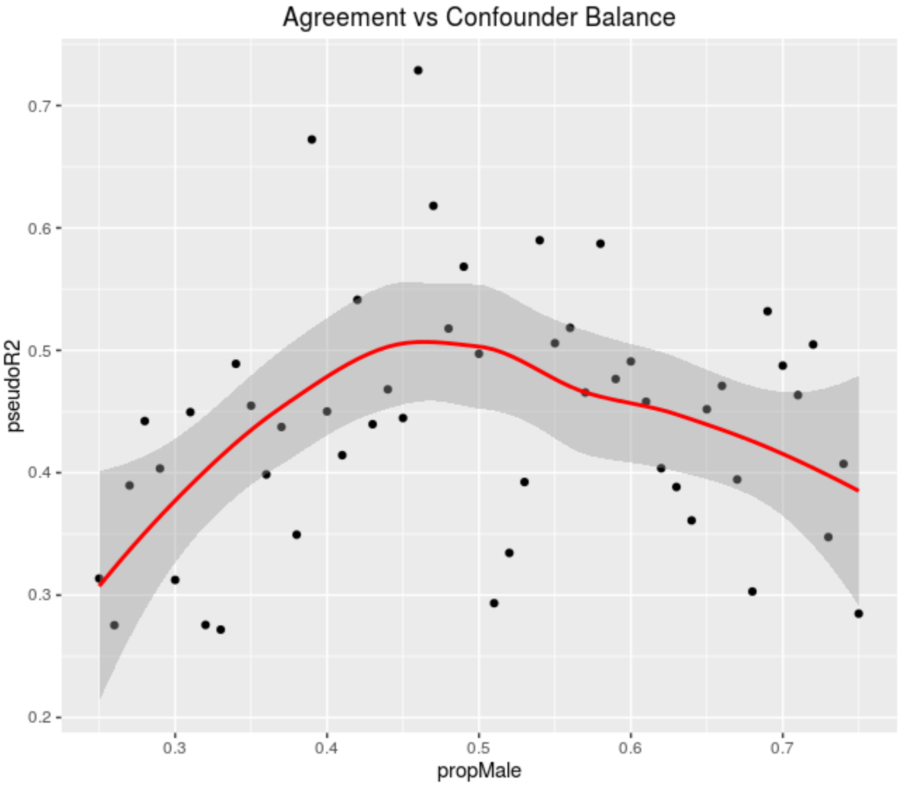


Figure 7: Very clear dependence of agreement between cases and control on the balance. In other words, with a strong lack of balance, we see weaker agreement, indicating that the results we DO see are a function of the confounder and not the case-control partition.

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

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

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

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

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

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