Unifying and Generalizing Methods for Removing Unwanted Variation Based on Negative Controls

[1]David Gerard and [2]Matthew Stephens

[1]*Department of Mathematics and Statistics, American University, Washington DC, USA*

[2]*Departments of Human Genetics and Statistics, University of Chicago, Chicago IL, USA*

Supplementary Material

This supplementary document contains proofs, additional theoretical and simulation details, and additional simulation results.

S1 Definition of truncated SVD

The truncated Singular Value Decomposition (SVD) of a matrix $\boldsymbol{Y}$ is defined as follows. Let $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\intercal$ be the SVD of $\boldsymbol{Y}$. Let $\boldsymbol{D}^{(q)} = \text{diag}(d_{11}, \ldots, d_{qq}, 0, \ldots, 0) \in \mathbb{R}^{n\times n}$ be a diagonal matrix whose first q diagonal elements are the same as in $\boldsymbol{D}$ and the last $n - q$ diagonal elements

are 0. Then

$$\hat{\boldsymbol{Z}}(\boldsymbol{Y}) = \boldsymbol{U}[\boldsymbol{D}^{(q)}]^{1-\pi} \tag{S1.1}$$

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{Y}) = [\boldsymbol{D}^{(q)}]^{\pi}\boldsymbol{V}^\intercal \tag{S1.2}$$

$$\hat{\boldsymbol{\Sigma}}(\boldsymbol{Y}) = \text{diag}\left[\boldsymbol{V}(\boldsymbol{D} - \boldsymbol{D}^{(q)})^2\boldsymbol{V}^\intercal\right]/(n-q), \tag{S1.3}$$

for any $\pi \in [0, 1]$. (Without loss of generality, we take $\pi = 1$.)

S2 Proof of theorem concerning equivalence of RUV2 algorithms

We provide a proof of Theorem 1.

Proof. For simplicity, we first assume that there are no nuisance covariates. Then $\binom{\boldsymbol{Y}_{2\mathcal{C}}}{\boldsymbol{Y}_{3\mathcal{C}}} = \boldsymbol{Q}^\intercal\boldsymbol{Y}$, where $\boldsymbol{Q}$ is the orthogonal matrix from the QR decomposition of $\boldsymbol{X}$ (Section 2.1). Thus, $\binom{\hat{\boldsymbol{Z}}_2}{\hat{\boldsymbol{Z}}_3}$ from Procedure 3 results from applying $\mathcal{F}_1$ on $\boldsymbol{Q}^\intercal\boldsymbol{Y}$ while $\hat{\boldsymbol{Z}}$ from Procedure 2 results from applying $\mathcal{F}_2$ on $\boldsymbol{Y}$. From the definitions of (2.13) and (2.14), we thus have that $\hat{\boldsymbol{Z}}$ from step 1 of Procedure 2 is in the same column space as $\boldsymbol{Q}\binom{\hat{\boldsymbol{Z}}_2}{\hat{\boldsymbol{Z}}_3}$ from step 1 of Procedure 3. $\hat{\boldsymbol{\beta}}_2$ from (2.12) contains the partial regression coefficients of $\boldsymbol{X}$ when including $\boldsymbol{Q}\binom{\hat{\boldsymbol{Z}}_2}{\hat{\boldsymbol{Z}}_3}$ as nuisance covariates (to show this, just calculate the MLE's of $\boldsymbol{\beta}_2$ and $\boldsymbol{\alpha}$ using (2.4) and (2.5)). The estimates of $\boldsymbol{\beta}_2$ in step 2 of Procedure 2 are also partial regression coefficients of $\boldsymbol{X}$ when

including $\hat{\boldsymbol{Z}}$ as nuisance covariates. Since the partial regression coefficients in Procedure 2 are only a function of $\hat{\boldsymbol{Z}}$ through its column space, and the partial regression coefficients in Procedure 3 are only a function of $\boldsymbol{Q}\binom{\hat{\boldsymbol{Z}}_2}{\hat{\boldsymbol{Z}}_3}$ through its column space, and these column spaces are the same, we have completed the proof for the case of no nuisance parameters.

To deal with nuisance parameters, Gagnon-Bartsch et al. (2013) rotate $\boldsymbol{X}$ and $\boldsymbol{Y}$ onto the orthogonal complement of the columns of $\boldsymbol{X}$ corresponding to the nuisance parameters prior to applying Procedure 2. If we partition $\boldsymbol{Q} = (\boldsymbol{Q}_1, \boldsymbol{Q}_2, \boldsymbol{Q}_3)$ and $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2)$, this is equivalent to using the model

$$\boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{Y} = \boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{X}_2\boldsymbol{\beta}_2 + \boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{Z}\boldsymbol{\alpha} + \boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{E} \tag{S2.1}$$

where $\boldsymbol{W}$ is some arbitrary (but known) $n-k_1$ by $n-k_1$ orthogonal matrix. Or, using the QR decomposition of $\boldsymbol{X}$, (S2.1) is equal to

$$\boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{Y} = \boldsymbol{W}\begin{pmatrix}\boldsymbol{R}_{22}\\ \boldsymbol{0}\end{pmatrix}\boldsymbol{\beta}_2 + \boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{Z}\boldsymbol{\alpha} + \boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{E}. \tag{S2.2}$$

Let

$$\begin{aligned}\tilde{\boldsymbol{Y}} &:= \boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{Y},\ \tilde{\boldsymbol{X}} := \boldsymbol{W}\begin{pmatrix}\boldsymbol{R}_{22}\\ \boldsymbol{0}\end{pmatrix},\\ \tilde{\boldsymbol{Z}} &:= \boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{Z},\ \text{and}\ \tilde{\boldsymbol{E}} := \boldsymbol{W}\begin{pmatrix}\boldsymbol{Q}_2^\intercal\\ \boldsymbol{Q}_3^\intercal\end{pmatrix}\boldsymbol{E}.\end{aligned} \tag{S2.3}$$

Then, (S2.1) is equal to

$$\tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{X}}\boldsymbol{\beta} + \tilde{\boldsymbol{Z}}\boldsymbol{\alpha} + \tilde{\boldsymbol{E}}. \tag{S2.4}$$

We now just apply the arguments of the previous paragraph to (S2.4), where there are no nuisance parameters and where the QR decomposition of $\tilde{\boldsymbol{X}}$ is just $\tilde{\boldsymbol{X}} = \boldsymbol{W}\binom{\boldsymbol{R}_{22}}{\boldsymbol{0}}$. This completes the proof. □

S3 Proof of corollary concerning equivalence of RUV2 algorithms

We provide a proof of Corollary 1.

Proof. Suppose $\mathcal{F}_2$ from (2.14) is left orthogonally equivariant, then

$$\mathcal{F}_2(\boldsymbol{Y}) = \{\hat{\boldsymbol{\Sigma}}(\boldsymbol{Q}^\intercal\boldsymbol{Y}), \boldsymbol{Q}\hat{\boldsymbol{Z}}(\boldsymbol{Q}^\intercal\boldsymbol{Y})\boldsymbol{A}(\boldsymbol{Y}), \boldsymbol{A}^{-1}(\boldsymbol{Y})\hat{\boldsymbol{\alpha}}(\boldsymbol{Q}^\intercal\boldsymbol{Y})\} \tag{S3.1}$$

$$= \{\hat{\boldsymbol{\Sigma}}(\boldsymbol{Y}), \boldsymbol{Q}\boldsymbol{Q}^\intercal\hat{\boldsymbol{Z}}(\boldsymbol{Y}), \hat{\boldsymbol{\alpha}}(\boldsymbol{Y})\} \tag{S3.2}$$

$$= \{\hat{\boldsymbol{\Sigma}}(\boldsymbol{Y}), \hat{\boldsymbol{Z}}(\boldsymbol{Y}), \hat{\boldsymbol{\alpha}}(\boldsymbol{Y})\} = \mathcal{F}_1(\boldsymbol{Y}). \tag{S3.3}$$

Let $\mathcal{F} := \mathcal{F}_1 = \mathcal{F}_2$. From the results of Theorem 1, we have that

$$\text{RUV2}_{old}(\mathcal{F}) = \text{RUV2}_{old}(\mathcal{F}_2) = \text{RUV2}_{new}(\mathcal{F}_1) = \text{RUV2}_{new}(\mathcal{F}). \quad \text{(S3.4)}$$

□

S4 Proof of theorem connecting RUV2 with RUV4

We provide a proof of Theorem 2.

S4.1 Connection to RUV4

The astute reader will have noticed that (3.2) is the same as (2.7). RUV3 can be viewed as a version of RUV4 with a particular factor analysis. Specifically, any factor analysis applied during RUV4 of the following form will result in RUV3:

$$\{\hat{\boldsymbol{\Sigma}}(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}), \hat{\boldsymbol{Z}}_3(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}), \hat{\boldsymbol{\alpha}}(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}})\} \text{ such that} \quad \text{(S4.1)}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}) = \hat{\boldsymbol{\Sigma}}_{\mathcal{C}}(\boldsymbol{Y}_{3\mathcal{C}}) \quad \text{(S4.2)}$$

$$\hat{\boldsymbol{Z}}_3(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}) = \hat{\boldsymbol{Z}}_3(\boldsymbol{Y}_{3\mathcal{C}}), \text{ and} \quad \text{(S4.3)}$$

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}) = (\hat{\boldsymbol{\alpha}}_{\mathcal{C}}(\boldsymbol{Y}_{3\mathcal{C}}), [\hat{\boldsymbol{Z}}_3^\intercal \hat{\boldsymbol{Z}}_3]^{-1} \hat{\boldsymbol{Z}}_3^\intercal \boldsymbol{Y}_{3\bar{\mathcal{C}}}). \quad \text{(S4.4)}$$

Or, more simply, RUV4 is equal to RUV3 if, in RUV4 one uses any factor analysis such that $\hat{\boldsymbol{\alpha}}_{\bar{\mathcal{C}}} = (\hat{\boldsymbol{Z}}_3^\intercal \hat{\boldsymbol{Z}}_3)^{-1} \hat{\boldsymbol{Z}}_3^\intercal \boldsymbol{Y}_{3\bar{\mathcal{C}}}$ and $\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}$, $\hat{\boldsymbol{Z}}_3$, and $\hat{\boldsymbol{\alpha}}_{\mathcal{C}}$ are functions of $\boldsymbol{Y}_{3\mathcal{C}}$ but *not* $\boldsymbol{Y}_{3\bar{\mathcal{C}}}$

Actually, using a truncated SVD on $(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}})$ (equations (S1.1) to (S1.3)) results in the equalities (S4.2) to (S4.4) if one ignores the functional dependencies. That is, using the truncated SVD on $(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}})$ one can easily prove the relationships

$$\hat{\boldsymbol{Z}}_3(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}) = \hat{\boldsymbol{Z}}_3(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}), \tag{S4.5}$$

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}) = (\hat{\boldsymbol{\alpha}}_{\mathcal{C}}(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}), [\hat{\boldsymbol{Z}}_3^\intercal \hat{\boldsymbol{Z}}_3]^{-1} \hat{\boldsymbol{Z}}_3^\intercal \boldsymbol{Y}_{3\bar{\mathcal{C}}}). \tag{S4.6}$$

The main difference, then, between RUV3 and RUV4 as implemented in the `ruv` R package (Gagnon-Bartsch, 2015) is that in RUV3 $\hat{\boldsymbol{Z}}_3$ has a functional dependence only on $\boldsymbol{Y}_{3\mathcal{C}}$ and *not* $\boldsymbol{Y}_{3\bar{\mathcal{C}}}$.

S4.2 Connection to RUV2

Similar to the relationship between RUV3 and RUV4, RUV3 may also be viewed as a version of RUV2 with a particular factor analysis. Specifically any factor analysis applied during RUV2 of the following form will result

in RUV3:

$$\{\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}), \hat{\boldsymbol{Z}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}), \hat{\boldsymbol{\alpha}}_{\mathcal{C}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}})\} \text{ such that} \tag{S4.7}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}) = \hat{\boldsymbol{\Sigma}}_{\mathcal{C}}(\boldsymbol{Y}_{3\mathcal{C}}) \tag{S4.8}$$

$$\hat{\boldsymbol{\alpha}}_{\mathcal{C}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}) = \hat{\boldsymbol{\alpha}}_{\mathcal{C}}(\boldsymbol{Y}_{3\mathcal{C}}), \text{ and} \tag{S4.9}$$

$$\hat{\boldsymbol{Z}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}) = \begin{pmatrix} \boldsymbol{Y}_{2\mathcal{C}}\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}^{-1}\hat{\boldsymbol{\alpha}}_{\mathcal{C}}^{\intercal}(\hat{\boldsymbol{\alpha}}_{\mathcal{C}}\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}^{-1}\hat{\boldsymbol{\alpha}}_{\mathcal{C}}^{\intercal})^{-1} \\ \hat{\boldsymbol{Z}}_3(\boldsymbol{Y}_{3\mathcal{C}}) \end{pmatrix}. \tag{S4.10}$$

In simpler terms, RUV2 is the same as RUV3 if, in RUV2 one uses any factor analysis such that $\hat{\boldsymbol{Z}}_2 = \boldsymbol{Y}_{2\mathcal{C}}\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}^{-1}\hat{\boldsymbol{\alpha}}_{\mathcal{C}}^{\intercal}(\hat{\boldsymbol{\alpha}}_{\mathcal{C}}\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}^{-1}\hat{\boldsymbol{\alpha}}_{\mathcal{C}}^{\intercal})^{-1}$ and $\hat{\boldsymbol{\alpha}}_{\mathcal{C}}$, $\hat{\boldsymbol{Z}}_3$, and $\boldsymbol{\Sigma}_{\mathcal{C}}$ are functions of $\boldsymbol{Y}_{3\mathcal{C}}$ but not $\boldsymbol{Y}_{2\mathcal{C}}$.

Similar to Section S4.1, using the truncated SVD on $\binom{\boldsymbol{Y}_{2\mathcal{C}}}{\boldsymbol{Y}_{3\mathcal{C}}}$ — assuming homoscedastic variances rather than heteroscedastic variances — will result in equalities (S4.8) to (S4.10) except for the functional dependencies. That is, when using the truncated SVD on $\binom{\boldsymbol{Y}_{2\mathcal{C}}}{\boldsymbol{Y}_{3\mathcal{C}}}$ one can show that the following relationships hold:

$$\hat{\boldsymbol{\alpha}}_{\mathcal{C}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}) = \hat{\boldsymbol{\alpha}}_{\mathcal{C}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}), \text{ and} \tag{S4.11}$$

$$\hat{\boldsymbol{Z}}(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}) = \begin{pmatrix} \boldsymbol{Y}_{2\mathcal{C}}\hat{\boldsymbol{\alpha}}_{\mathcal{C}}^{\intercal}(\hat{\boldsymbol{\alpha}}_{\mathcal{C}}\hat{\boldsymbol{\alpha}}_{\mathcal{C}}^{\intercal})^{-1} \\ \hat{\boldsymbol{Z}}_3(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}) \end{pmatrix}. \tag{S4.12}$$

The main difference, then, between RUV2 and RUV3 is that in RUV3 $\hat{\boldsymbol{\alpha}}_{\mathcal{C}}$ has a functional dependence only on $\boldsymbol{Y}_{3\mathcal{C}}$ and *not* $\boldsymbol{Y}_{2\mathcal{C}}$.

S4.3 RUV2 and RUV4 if and only if RUV3

We have shown in Sections S4.1 and S4.2 that RUV3 can be considered a variant of both RUV2 and RUV4. But the converse is easily proved to also be true.

Proof. If the procedure is a version of RUV2, then (2.10) holds. But if the procedure is a version of RUV4, then (2.7) holds. These are two properties of RUV3 (equations (3.2) and (3.3)).

It remains to show that $\hat{\boldsymbol{Z}}_3$ and $\hat{\boldsymbol{\alpha}}_{\mathcal{C}}$ are functions only of $\boldsymbol{Y}_{3\mathcal{C}}$. But this is clear since if the procedure is RUV4, these quantities are functions only of $\boldsymbol{Y}_{3\mathcal{C}}$ and $\boldsymbol{Y}_{3\bar{\mathcal{C}}}$, while if the procedure is RUV2 these quantities are functions only of $\boldsymbol{Y}_{2\mathcal{C}}$ and $\boldsymbol{Y}_{3\mathcal{C}}$. Since the procedure is both RUV2 and RUV4, this necessarily implies that these quantities are functions only of $\boldsymbol{Y}_{3\mathcal{C}}$. □

To summarize, RUV3 can be viewed as both a version of RUV2 and a version of RUV4 and if a procedure is both a version of RUV2 and a version of RUV4 then it is a version of RUV3.

S5 Relationship to RUVfun

Gagnon-Bartsch et al. (2013) describe a general framework they call RU-

Vfun, for RUV-functional. In our notation, and within the rotated model framework of Section 2.1, RUVfun may be described as

1. Perform factor analysis on $(\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}})$ to obtain an estimate $\hat{\boldsymbol{\alpha}}$,
2. Let $\hat{\boldsymbol{\alpha}}_j$ denote the jth column of $(\hat{\boldsymbol{\alpha}}_{\mathcal{C}}, \hat{\boldsymbol{\alpha}}_{\bar{\mathcal{C}}})$ and $\boldsymbol{y}_j$ denote the jth column of $(\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{2\bar{\mathcal{C}}})$. Train a function f using responses $\boldsymbol{y}_j$ and predictors $\hat{\boldsymbol{\alpha}}_j$ for $j = 1, \ldots, m$ (recall, we have m control genes). That is, fit

$$\boldsymbol{y}_j \approx f(\hat{\boldsymbol{\alpha}}_j), \text{ for } j = 1, \ldots, m, \tag{S5.1}$$

   and call the resulting predictor $\hat{f}$.
3. Estimate $\boldsymbol{Y}_{2\bar{\mathcal{C}}}$ using predictors $\hat{\boldsymbol{\alpha}}_{\bar{\mathcal{C}}}$. That is,

$$\hat{\boldsymbol{y}}_j = \hat{f}(\hat{\boldsymbol{\alpha}}_j), \text{ for } j = m+1, \ldots, p. \tag{S5.2}$$

4. Estimate $\boldsymbol{\beta}_{2\bar{\mathcal{C}}}$ with $\boldsymbol{R}_{22}^{-1}(\boldsymbol{Y}_{2\bar{\mathcal{C}}} - \hat{\boldsymbol{Y}}_{2\bar{\mathcal{C}}})$.

This is the way it is presented in Section 3.8.2 of Gagnon-Bartsch et al. (2013), but typically they take the factor analysis step to mean just setting $\hat{\boldsymbol{\alpha}} = (\boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}})$. A pictorial representation of RUVfun is presented in the second panel of Figure S1. The only difference between the RUVfun diagram in Figure S1 and the RUV4 diagram in Figure 2 is that "regression" was changed to "train any function".

RUVfun, though more general than RUV4, is a special case of RUV*. And RUV* is more general: for example, RUV2 is a version of RUV* but

not of RUVfun. Indeed, RUV* generalizes RUVfun in three key ways. First, RUVfun uses only one column of $\hat{\boldsymbol{\alpha}}$ to estimate one column of $\boldsymbol{Y}_{2\bar{\mathcal{C}}}$ while RUV* allows for joint estimation of $\boldsymbol{Y}_{2\bar{\mathcal{C}}}$. Second, RUVfun assumes that each column of the rotated $\boldsymbol{Y}$ matrix is independent and identically distributed (Gagnon-Bartsch et al., 2013, page 41) while RUV* does not. Indeed, some matrix imputation approaches use column covariances to great effect (Allen and Tibshirani, 2010). Third, RUVfun uses only $\boldsymbol{Y}_{3\mathcal{C}}$ to train the prediction function, whereas RUV* can use all elements in the rotated $\boldsymbol{Y}$ matrix.

S6 Estimating standard errors

For simplicity we have focused our descriptions of RUV2, RUV3, RUV4, and RUV* on point estimation for $\boldsymbol{\beta}_2$. In practice, to be useful, all of these methods must also provide standard errors for these estimates. Several different approaches to this problem exist, and we have found in empirical comparisons (e.g. Section 5) that the approach taken can greatly affect results, particularly calibration of interval estimates. In this section we therefore briefly review some of these approaches.

The simplest approach is to treat the estimated $\hat{\boldsymbol{Z}}$ as the true value of $\boldsymbol{Z}$, and then use standard theory from linear models to estimate the

standard errors of the estimated coefficients. That is, first estimate $\hat{\boldsymbol{Z}}$ using any of the RUV approaches, then regress $\boldsymbol{Y}$ on $(\boldsymbol{X}, \hat{\boldsymbol{Z}})$ and obtain estimated standard errors (for coefficients of $\boldsymbol{X}$) in the usual way. This is the default option in the `ruv` R package. The `cate` R package implements this (with the `nc.var.correction` and `calibrate` parameters both set to `FALSE`), but without the usual degrees of freedom correction in estimated standard errors. Though asymptotically this will not matter, we have found that for small sample sizes this can substantially hurt performance due to downward-biased standard errors.

Gagnon-Bartsch et al. (2013) noted that the standard errors estimated using this simple approach can be poorly behaved, essentially because the $\hat{\boldsymbol{Z}}$ are estimated and not known. They suggested several approaches to calibrating these standard errors using control genes. The approach that we found to work best in our comparisons (at least, when there are many control genes — see Section 5.3) is to multiply the estimated standard errors by a factor λ (i.e. set $\tilde{s}_i = \lambda \hat{s}_i$) which is estimated from control genes by

$$\lambda := \left(\frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \frac{\hat{\beta}_j^2}{\hat{s}_j^2} \right)^{0.5} \tag{S6.1}$$

where $\hat{\beta}_j$ and $\hat{s_j}$ are the estimated coefficients and their standard errors (Equation (236) in Gagnon-Bartsch et al. (2013)). In our empirical com-

parisons below we refer to this procedure as "control gene calibration". Gagnon-Bartsch et al. (2013) use heuristic arguments to motivate (S6.1) in the context of studies with just one covariate of interest. In Section S7, we extend (S6.1) to the case when there is more than one covariate of interest and formally justify it with maximum likelihood arguments.

Sun et al. (2012) take a different approach to calibration, which does not use control genes, but is motivated by the assumption that most genes are null. Specifically they suggest centering the t-statistics $\hat{\beta}_j/\hat{s}_j$ by their median and scaling them by their median absolute deviation (MAD):

$$\tilde{t}_i = \frac{\hat{\beta}_i/\hat{s}_i - \text{median}\left(\hat{\beta}_1/\hat{s}_1, \ldots, \hat{\beta}_p/\hat{s}_p\right)}{\text{MAD}\left(\hat{\beta}_1/\hat{s}_1, \ldots, \hat{\beta}_p/\hat{s}_p\right)}. \tag{S6.2}$$

The motivation for this adjustment is that if most genes are null, then normalizing by robust estimates of the null t-statistics' center and scale will make the null t-statistics more closely match their theoretical distribution. This adjustment of t-statistics is closely connected with the variance calibration (S6.1). Indeed, if we assume that the median of the t-statistics is approximately zero, then it is effectively equivalent to variance calibration with

$$\lambda_{\text{MAD}} := \text{MAD}\left(\frac{\hat{\beta}_1}{\hat{s}_1}, \ldots, \frac{\hat{\beta}_p}{\hat{s}_p}\right) \tag{S6.3}$$

in place of (S6.1).

In addition to MAD calibration, Wang et al. (2017) offer asymptotic arguments for an additive variance adjustment. This additive inflation term is particularly important when there are few control genes, and is specific to the RUV4 estimator (unlike (S6.1) and (S6.3) which can be applied to any estimator).

Finally, before development of RUV-like methods, the benefits of using empirical Bayes variance moderation (EBVM) (Smyth, 2004) were widely recognized in gene expression analyses. Variance moderation can be applied in combination with the variance calibration methods discussed above: for example, the `ruv::variance_adjust` function in R applies EBVM before applying (S6.1). EBVM can similarly be incorporated into other methods. For RUV3 and CATE we can use EBVM either before or after the generalized least squares (GLS) step of their respective algorithms (equation (3.2) for RUV3 and equation (2.7) for CATE).

S7 Calibrated CATE

We can derive a multivariate version of (S6.1) and formally justify it with maximum likelihood arguments via a modification of step 2 of Procedure 1. After step 1, we modify (2.8) and (2.9) to include a variance inflation pa-

rameter, λ.

$$\boldsymbol{Y}_{2\mathcal{C}} = \boldsymbol{Z}_2\hat{\boldsymbol{\alpha}}_{\mathcal{C}} + \boldsymbol{E}_{2\mathcal{C}}, \tag{S7.1}$$

$$e_{2\mathcal{C}ij} \overset{ind}{\sim} N(0, \lambda\hat{\sigma}_j^2). \tag{S7.2}$$

Step 2 of CATE simply calculates the MLE of $\boldsymbol{Z}_2$ in (2.8) and (2.9). Our calibration is simply finding the MLE's of $\boldsymbol{Z}_2$ and λ in (S7.1) and (S7.2), which can be done in closed form. The estimate of $\boldsymbol{Z}_2$ is unchanged (2.7). The MLE of λ is

$$\hat{\lambda} = \frac{1}{k_2 m}\operatorname{tr}\left[(\boldsymbol{Y}_{2\mathcal{C}} - \hat{\boldsymbol{Z}}_2\hat{\boldsymbol{\alpha}}_{\mathcal{C}})\hat{\boldsymbol{\Sigma}}_{\mathcal{C}}^{-1}(\boldsymbol{Y}_{2\mathcal{C}} - \hat{\boldsymbol{Z}}_2\hat{\boldsymbol{\alpha}}_{\mathcal{C}})^{\intercal}\right]. \tag{S7.3}$$

Inference then proceeds by using $\hat{\lambda}\hat{\sigma}_j$ as the estimate of σ_j. Formula (S7.3) is a multivariate version of equation (236) of Gagnon-Bartsch et al. (2013) (and (S6.1)). Though while we derived (S7.3) using an application of maximum likelihood, Gagnon-Bartsch et al. (2013) formulated their calibration using heuristic arguments.

S8 Approximate posterior inference

As discussed in Section 4.2, if we could calculate $p(\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}|\mathcal{Y}_m)$, where $\mathcal{Y}_m := \{\boldsymbol{Y}_{2\mathcal{C}}, \boldsymbol{Y}_{3\mathcal{C}}, \boldsymbol{Y}_{3\bar{\mathcal{C}}}\}$, then inference on $[\boldsymbol{\beta}_2|\boldsymbol{Y}_{2\bar{\mathcal{C}}}, \mathcal{Y}_m]$ would be straightforward — at least in principal if not in practice. That is, suppose $h(\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}) := p(\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}|\mathcal{Y}_m)$, then one would simply use the likelihood $h(\boldsymbol{Y}_{2\bar{\mathcal{C}}} - \boldsymbol{R}_{22}\boldsymbol{\beta}_2)$ and a

user-provided prior $g(\cdot)$ over $\boldsymbol{\beta}_2$ to calculate a posterior and return posterior quantities.

However, to reap the benefits of modularity, we describe a procedure to fit the overall model (4.1) in two discrete steps: A factor analysis step using (4.2) and a regression step using (4.3). We begin with estimating the unwanted variation. Specifically, we suppose that one first assumes the model (4.2) where $\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}$ is *unobserved*. The error $\boldsymbol{E}$ can follow any model a researcher desires. Though, of course, the rotation leading to (4.1) was derived by assuming Gaussian errors with independent rows (Section 2.1) and the appropriateness of different error models should be examined before use. We make the relatively weak assumption that model (4.2) is fit using any Bayesian procedure that yields a proper posterior and that the researcher can obtain samples from the posterior distribution $[\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}|\mathcal{Y}_m]$. Call these posterior draws $\tilde{\boldsymbol{Y}}^{(1)}_{2\bar{\mathcal{C}}}, \ldots, \tilde{\boldsymbol{Y}}^{(t)}_{2\bar{\mathcal{C}}}$.

After estimating the unwanted variation, we have a regression step where we estimate $\boldsymbol{\beta}_2$ using (4.3). Suppose we have any user-provided prior density over $\boldsymbol{\beta}_2$, say $g(\cdot)$. In order to reap the benefits of modularity, we need to derive a Bayesian procedure for approximating the posterior over $\boldsymbol{\beta}_2$ using just the samples $\tilde{\boldsymbol{Y}}^{(1)}_{2\bar{\mathcal{C}}}, \ldots, \tilde{\boldsymbol{Y}}^{(t)}_{2\bar{\mathcal{C}}}$ from the previous step. To do so, we let $\hat{\boldsymbol{\beta}}^{(i)}_2 := \boldsymbol{R}^{-1}_{22}(\boldsymbol{Y}_{2\bar{\mathcal{C}}} - \tilde{\boldsymbol{Y}}^{(i)}_{2\bar{\mathcal{C}}})$ and note that $\hat{\boldsymbol{\beta}}^{(1)}_2, \ldots, \hat{\boldsymbol{\beta}}^{(t)}_2$ are actually

draws from $[\boldsymbol{\beta}_2|\boldsymbol{Y}_{2\bar{\mathcal{C}}}, \mathcal{Y}_m]$ when using an (improper) uniform prior over $\boldsymbol{\beta}_2$. This follows because (4.3) is a location family. We can then weight these samples by the prior information $g(\hat{\boldsymbol{\beta}}_2^{(i)})$ to obtain draws from the posterior $[\boldsymbol{\beta}_2|\boldsymbol{Y}_{2\bar{\mathcal{C}}}, \mathcal{Y}_m]$ when using $g(\cdot)$ as our prior density. This strategy of weighting samples from one distribution to approximate quantities from another distribution was discussed in Trotter and Tukey (1956). What this means in practice is that given any function of $\boldsymbol{\beta}_2$, say $f(\cdot)$, we can approximate its posterior expectation consistently in the number of posterior draws, t, from the first step. That is,

$$\frac{\sum_{i=1}^{t} g(\hat{\boldsymbol{\beta}}_2^{(i)}) f(\hat{\boldsymbol{\beta}}_2^{(i)})}{\sum_{i=1}^{t} g(\hat{\boldsymbol{\beta}}_2^{(i)})} \xrightarrow{P} E[f(\boldsymbol{\beta}_2)|\boldsymbol{Y}_{2\bar{\mathcal{C}}}, \mathcal{Y}_m] \tag{S8.1}$$

Some example calculations of useful posterior quantities are provided in Section S12. We have given intuitive arguments here; formal arguments are given in Section S9. A technical condition required for this approach to work is that $g(\cdot)$ be absolutely continuous with respect to the distribution of $[\boldsymbol{R}_{22}^{-1}(\boldsymbol{Y}_{2\bar{\mathcal{C}}} - \tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}})|\mathcal{Y}_m]$. In the case when the errors $\boldsymbol{E}$ are Gaussian, it suffices to consider priors that are absolutely continuous with respect to Lebesgue measure.

Importantly, this two-step approach, though modular, is actually fitting the full model (4.1) as if done in one step. That is, nothing is lost in taking this two-step approach, except perhaps we could have found more efficient

posterior approximations if the procedure was fit in one step. However, our approach is a contrast to RUV2, RUV3, and RUV4 which do not propagate the uncertainty in estimating the unwanted variation. This allows us to give more accurate quantities of uncertainty (Section 5.3).

To implement this approach in practice, we need to specify both a specific model for the unwanted variation (4.2) and a prior for $\boldsymbol{\beta}_2$. As a proof of concept we use a very simple Bayesian factor model with Gaussian errors (Section S10) and an improper uniform prior on $\boldsymbol{\beta}_2$, which yields a proper proper posterior no matter the model for the unwanted variation (Section S11). We note that although our model for the unwanted variation is based on a factor model, RUVB is neither a version of RUV4 nor RUV2.

S9 Justification for approximate posterior inference

In this section, we prove a general result that given a location family, we can approximate posterior expectations to any arbitrary level of precision using samples from the error distribution. We then connect this to the posterior approximation discussed in Section S8. For data $\boldsymbol{y} \in \mathbb{R}^d$, suppose the model is

$$\boldsymbol{y} = h(\boldsymbol{\theta}) + \boldsymbol{e}, \tag{S9.1}$$

where $h : \mathbb{R}^d \to \mathbb{R}^d$ is bijective. Let $\boldsymbol{J}(\boldsymbol{z})$ be the Jacobian matrix of h. That is

$$[\boldsymbol{J}(\boldsymbol{z})]_{ij} = \frac{\mathrm{d}h_i(\boldsymbol{z})}{\mathrm{d}z_j}, \tag{S9.2}$$

and let $|\boldsymbol{J}(\boldsymbol{z})|$ denote its determinant. Let g be the prior of $\boldsymbol{\theta}$, which we assume is absolutely continuous with respect to the density of $h^{-1}(\boldsymbol{y} - \boldsymbol{e})$.

Theorem 1. *Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K$ be i.i.d. random variables equal in distribution to $\boldsymbol{e}$. Let $u : \mathbb{R}^d \to \mathbb{R}^d$ be a function. Let*

$$\hat{E}[u(\boldsymbol{\theta})|\boldsymbol{y}] := \frac{\sum_{k=1}^K u(h^{-1}(\boldsymbol{y} - \boldsymbol{e}_k)) g(h^{-1}(\boldsymbol{y} - \boldsymbol{e}_k))/|\boldsymbol{J}(h^{-1}(\boldsymbol{y} - \boldsymbol{e}_k))|}{\sum_{k=1}^K g(h^{-1}(\boldsymbol{y} - \boldsymbol{e}_k))/|\boldsymbol{J}(h^{-1}(\boldsymbol{y} - \boldsymbol{e}_k))|}, \tag{S9.3}$$

then

$$\hat{E}[u(\boldsymbol{\theta})|\boldsymbol{y}] \xrightarrow{P} E[u(\boldsymbol{\theta})|\boldsymbol{y}]. \tag{S9.4}$$

Proof. Let f be the density of $\boldsymbol{e}$. Then $p(\boldsymbol{y}|\boldsymbol{\theta}) = f(\boldsymbol{y} - h(\boldsymbol{\theta}))$ since $\boldsymbol{y}$ belongs

to a location family. We have

$$\hat{E}[u(\boldsymbol{\theta})|\boldsymbol{y}] := \frac{\frac{1}{K}\sum_{k=1}^{K} u(h^{-1}(\boldsymbol{y}-\boldsymbol{e}_k))g(h^{-1}(\boldsymbol{y}-\boldsymbol{e}_k))/|\boldsymbol{J}(h^{-1}(\boldsymbol{y}-\boldsymbol{e}_k))|}{\frac{1}{K}\sum_{k=1}^{K} g(h^{-1}(\boldsymbol{y}-\boldsymbol{e}_k))/|\boldsymbol{J}(h^{-1}(\boldsymbol{y}-\boldsymbol{e}_k))|} \tag{S9.5}$$

$$\xrightarrow{P} \frac{E\left[u(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))g(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))/|\boldsymbol{J}(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))|\right]}{E\left[g(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))/|\boldsymbol{J}(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))|\right]} \tag{S9.6}$$

$$= \frac{\int u(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))g(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))/|\boldsymbol{J}(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))f(\boldsymbol{e})\,\mathrm{d}\boldsymbol{e}}{\int g(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))/|\boldsymbol{J}(h^{-1}(\boldsymbol{y}-\boldsymbol{e}))f(\boldsymbol{e})\,\mathrm{d}\boldsymbol{e}} \tag{S9.7}$$

$$= \frac{\int u(\boldsymbol{z})g(\boldsymbol{z})f(\boldsymbol{y}-h(\boldsymbol{z}))\,\mathrm{d}\boldsymbol{z}}{\int g(\boldsymbol{z})f(\boldsymbol{y}-h(\boldsymbol{z}))\,\mathrm{d}\boldsymbol{z}} \tag{S9.8}$$

$$= \frac{\int u(\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{y})p(\boldsymbol{y})\,\mathrm{d}\boldsymbol{z}}{p(\boldsymbol{y})} \tag{S9.9}$$

$$= \int u(\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{z} \tag{S9.10}$$

$$= E[u(\boldsymbol{\theta})|\boldsymbol{y}]. \tag{S9.11}$$

Line (S9.6) follows by two applications of the weak law of large numbers followed by Slutsky's theorem. Line (S9.8) follows by a change of variables $\boldsymbol{e} = \boldsymbol{y} - h(\boldsymbol{z})$, the Jacobian of which is just $\boldsymbol{J}(\boldsymbol{z})$. The condition on the prior g is used in (S9.9) when we start considering $\boldsymbol{z}$ as a dummy variable for $\boldsymbol{\theta}$. To think intuitively about this condition on the prior, if the measure for $\boldsymbol{\theta}$ is non-zero on a set $\mathcal{A}$ in the parameter space where the likelihood is non-zero but the measure is zero, then this approximation procedure would never sample $h^{-1}(\boldsymbol{y}-\boldsymbol{e}_k) \in \mathcal{A}$. This is even though $\mathcal{A}$ does have a non-

zero posterior probability. The absolute continuity condition prohibits this behavior. □

We now connect this general result to Section S8. The $\boldsymbol{y}$, $\boldsymbol{\theta}$, and $\boldsymbol{e}$ in this section are the $\boldsymbol{Y}_{2\bar{\mathcal{C}}}$, $\boldsymbol{\beta}_2$, and $\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}$, respectively, in Section S8. So instead of having draws $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K$, we have that $\tilde{\boldsymbol{Y}}^{(1)}_{2\bar{\mathcal{C}}}, \ldots, \tilde{\boldsymbol{Y}}^{(t)}_{2\bar{\mathcal{C}}}$ are draws from $[\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}|\mathcal{Y}_m]$. We also have that $h(\boldsymbol{\beta}_2) = \boldsymbol{R}_{22}\boldsymbol{\beta}_2$, and so the determinant of the Jacobian is merely $|\boldsymbol{R}_{22}|^p$. Since this is a constant independent of $\boldsymbol{\beta}_2$, the Jacobians in the numerator and denominator cancel in (S9.3).

S10 Simple Bayesian factor analysis

In this section, we present a simple Bayesian factor analysis which we used in our implementation of RUVB. The factor model is

$$\boldsymbol{Y}_{n\times p} = \boldsymbol{L}_{n\times q}\boldsymbol{F}_{q\times p} + \boldsymbol{E}_{n\times p},\ \boldsymbol{E} \sim N_{n\times p}(\boldsymbol{0}, \boldsymbol{\Sigma}\otimes\boldsymbol{I}_n),\ \boldsymbol{\Sigma}^{-1} = \operatorname{diag}(\xi_1, \ldots, \xi_p). \tag{S10.1}$$

We use conditionally conjugate priors on all parameters:

$$[\boldsymbol{L}|\boldsymbol{\Psi}] \sim N_{n\times q}(\mathbf{0}, \boldsymbol{\Psi} \otimes \boldsymbol{I}_n), \tag{S10.2}$$

$$[\boldsymbol{F}|\boldsymbol{\Sigma}] \sim N_{q\times p}(\mathbf{0}, \boldsymbol{\Sigma} \otimes \boldsymbol{I}_q), \tag{S10.3}$$

$$[\xi_i|\phi] \sim \text{Gamma}(\rho_0/2, \rho_0\phi/2), \tag{S10.4}$$

$$\phi \sim \text{Gamma}(\alpha_0/2, \alpha_0\beta_0/2), \tag{S10.5}$$

$$\boldsymbol{\Psi} = \text{diag}(\zeta_1^{-1}, \ldots, \zeta_q^{-1}), \tag{S10.6}$$

$$\zeta_i \sim \Gamma(\eta_0/2, \eta_0\tau_0/2), \tag{S10.7}$$

where all hyper-parameters with a 0 subscript are assumed known. We let $\text{Gamma}(a, b)$ denote the Gamma distribution with mean a/b and variance a/b^2. In the empirical evaluations of Section 5, we set the prior "sample sizes" to be small ($\rho_0 = \alpha_0 = 0.1$) so that the prior is only weakly informative. The prior mean for the precisions (β_0) was set arbitrarily to 1. Following Ghosh and Dunson (2009), we set $\eta_0 = \tau_0 = 1$.

The prior we use is similar in flavor to that of Ghosh and Dunson (2009), and like them we use the parameter expansion (Gelman, 2006), which improves MCMC mixing. However, there are some important differences between our formulation and that of Ghosh and Dunson (2009). First, we chose not to impose the usual identifiability conditions on the $\boldsymbol{F}$ matrix as we are not interested in the actual factors or factor loadings. Rather,

our specifications induce a prior over the joint matrix of interest $\boldsymbol{LF}$, which is identified. Second, the prior specification in Ghosh and Dunson (2009) is not order invariant. That is, their prior is influenced by the arbitrary ordering of the columns of $\boldsymbol{Y}$. This is a known problem (Leung and Drton, 2016) and we circumvent it by specifying an order invariant prior. Third, our prior specifications include hierarchical moderation of the variances, an important and well-established strategy in gene-expression studies (Smyth, 2004). Finally, we chose to link the variances of the genes with those of the factors. This is approximately modeling a mean-variance relationship and we have found it to work well in practice. We emphasize here that we do not actually know the form of the unwanted variation in the simulations in Section 5, and so the good performance of RUVB there is not due to some "unfair advantage" in the choice of prior.

One step of a Gibbs sampler is presented in Procedure 1. Repeated applications of the steps in Procedure 1 will result in a Markov chain whose stationary distribution is the posterior distribution of the parameters in our model. As the calculations of the full conditionals for all parameters are fairly standard, we omit the detailed derivations.

Algorithm 1 One step of Gibbs sampler for Bayesian factor analysis.

1: sample $[\boldsymbol{L}|\boldsymbol{Y},\boldsymbol{F},\boldsymbol{\Sigma}] \sim N_{n\times q}\left[\boldsymbol{Y}\boldsymbol{\Sigma}^{-1}\boldsymbol{F}^{T}(\boldsymbol{F}\boldsymbol{\Sigma}^{-1}\boldsymbol{F}^{\intercal}+\boldsymbol{\Psi}^{-1})^{-1},\ \boldsymbol{I}_n\otimes(\boldsymbol{F}\boldsymbol{\Sigma}^{-1}\boldsymbol{F}^{\intercal}+\boldsymbol{\Psi}^{-1})^{-1}\right]$,

2: sample $[\boldsymbol{F}|\boldsymbol{Y},\boldsymbol{L},\boldsymbol{\Sigma}] \sim N_{q\times p}\left[(\boldsymbol{L}^{\intercal}\boldsymbol{L}+\boldsymbol{I}_q)^{-1}\boldsymbol{L}^{\intercal}\boldsymbol{Y},\ \boldsymbol{\Sigma}\otimes(\boldsymbol{L}^{\intercal}\boldsymbol{L}+\boldsymbol{I}_q)^{-1}\right]$,

3: sample $[\xi_i|\boldsymbol{Y},\boldsymbol{F},\boldsymbol{L},\phi] \sim \text{Gamma}\left[(n+q+\rho_0)/2,(r_i+u_i+\rho_0\phi)/2\right]$, where

$\boldsymbol{r}=\text{diag}\left[(\boldsymbol{Y}-\boldsymbol{L}\boldsymbol{F})^{\intercal}(\boldsymbol{Y}-\boldsymbol{L}\boldsymbol{F})\right]$ and $\boldsymbol{u}=\text{diag}\left[\boldsymbol{F}^{\intercal}\boldsymbol{F}\right]$,

4: sample $[\phi|\boldsymbol{\xi}] \sim \text{Gamma}\left[(p\rho_0+\alpha_0)/2,(\alpha_0\beta_0+\rho_0\sum_{i=1}^{p}\xi_i)/2\right]$,

5: sample $[\zeta_i|\boldsymbol{L}] \sim \text{Gamma}\left[(n+\eta_0)/2,(s_i+\eta_0\tau_0)/2\right]$, where $\boldsymbol{s}=\text{diag}(\boldsymbol{L}^{\intercal}\boldsymbol{L})$.

S11 Propriety of posterior

In this section, we consider the propriety of the posterior $\boldsymbol{\beta}_2$ from Section S8. To prove that the posterior of $\boldsymbol{\beta}_2$ under a uniform prior is proper, it suffices to consider the model (4.3), which we repeat here:

$$\boldsymbol{R}_{22}^{-1}\boldsymbol{Y}_{\bar{\mathcal{C}}}=\boldsymbol{\beta}_2+\boldsymbol{R}_{22}^{-1}\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}. \tag{S11.1}$$

Letting $\boldsymbol{A}:=\boldsymbol{R}_{22}^{-1}\boldsymbol{Y}_{\bar{\mathcal{C}}}$ and $\boldsymbol{C}=\boldsymbol{R}_{22}^{-1}\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}$, we have

$$\boldsymbol{A}=\boldsymbol{\beta}_2+\boldsymbol{C}. \tag{S11.2}$$

Equation (S11.2) represents a general location family with location parameter $\boldsymbol{\beta}_2$. It is not difficult to prove that using a uniform prior on the location parameter in a location family always results in a proper posterior.

Theorem 2. *Let $\boldsymbol{C}$ be a random variable with density f. Let $\boldsymbol{A}=\boldsymbol{\beta}_2+\boldsymbol{C}$. Suppose we place a uniform prior on $\boldsymbol{\beta}_2$. Then the posterior of $\boldsymbol{\beta}_2$ is proper.*

Proof. We first note that

$$\int f(\boldsymbol{C})\,\mathrm{d}\boldsymbol{C} = 1. \tag{S11.3}$$

The density of $\boldsymbol{A}$ is just $f(\boldsymbol{A} - \boldsymbol{\beta}_2)$. Since the prior of $\boldsymbol{\beta}_2$ is uniform, this means that the posterior of $\boldsymbol{\beta}_2$ given $\boldsymbol{A}$ is proportional to the likelihood $f(\boldsymbol{A} - \boldsymbol{\beta}_2)$. Hence, making a change of variables of $\boldsymbol{\eta} = \boldsymbol{A} - \boldsymbol{\beta}_2$, we have

$$\int f(\boldsymbol{A} - \boldsymbol{\beta}_2)\,\mathrm{d}\boldsymbol{\beta}_2 = \int f(\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta} = 1. \tag{S11.4}$$

□

S12 Posterior summaries

Using (S8.1), one can obtain approximations to posterior summaries quite easily. Let $\hat{\boldsymbol{\beta}}_2^{(i)} := \boldsymbol{R}_{22}^{-1}(\boldsymbol{Y}_{2\bar{\mathcal{C}}} - \tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}^{(i)})$. The posterior mean may be approximated by

$$E[\boldsymbol{\beta}_2 | \boldsymbol{Y}_{2\bar{\mathcal{C}}}, \mathcal{Y}_m] \approx \frac{\sum_{i=1}^t \hat{\boldsymbol{\beta}}_2^{(i)} g\left(\hat{\boldsymbol{\beta}}_2^{(i)}\right)}{\sum_{i=1}^t g\left(\hat{\boldsymbol{\beta}}_2^{(i)}\right)}. \tag{S12.1}$$

Suppose one desires a $(1-\alpha)$ credible interval for β_{2jk} for some $0 < \alpha < 1$. Sort the $\hat{\beta}_{2jk}^{(i)}$'s such that $\hat{\beta}_{2jk}^{(1)} < \hat{\beta}_{2jk}^{(2)} < \cdots < \hat{\beta}_{2jk}^{(t)}$. A $(1-\alpha)$ credible interval may be approximated by finding the $\ell, m \in \{1, \ldots, t\}$ such that

$$\frac{\sum_{i=1}^{\ell-1} g\left(\hat{\boldsymbol{\beta}}_2^{(i)}\right)}{\sum_{i=1}^t g\left(\hat{\boldsymbol{\beta}}_2^{(i)}\right)} \leq \alpha/2 \text{ and } \frac{\sum_{i=m+1}^t g\left(\hat{\boldsymbol{\beta}}_2^{(i)}\right)}{\sum_{i=1}^t g\left(\hat{\boldsymbol{\beta}}_2^{(i)}\right)} \leq \alpha/2, \tag{S12.2}$$

then setting the $(1-\alpha)$ interval as $(\hat{\beta}_{2jk}^{(\ell)}, \hat{\beta}_{2jk}^{(m)})$. Note that in (S12.2) we have assumed that the $\hat{\boldsymbol{\beta}}_2^{(i)}$'s all have completely distinct elements. If the error distribution is Gaussian then we may do this without loss of generality as $\tilde{\boldsymbol{Y}}_{2\bar{\mathcal{C}}}^{(i)}$ is drawn from some convolution with a normal, and so is absolutely continuous with respect to Lebesgue measure.

Local false sign rates (lfsr's) (Stephens, 2017) have recently been proposed as a way to measure the confidence in the sign of each effect. The intuition is that the lfsr is the probability of making an error if one makes their best guess about the sign of a parameter. Let

$$p_{jk} := \frac{\sum_{\{i:\hat{\beta}_{2jk}^{(i)}<0\}} g(\hat{\beta}_{2jk}^{(i)})}{\sum_{i=1}^{t} g(\hat{\beta}_{2jk}^{(i)})}. \tag{S12.3}$$

Then the lfsr's may be approximated by

$$lfsr_{jk} := \min(p_{jk}, 1 - p_{jk}), \tag{S12.4}$$

which simplifies under a uniform prior to

$$lfsr_{jk} := \frac{1}{t} \min\left[\#\{\hat{\beta}_{2jk}^{(i)} < 0\}, \#\{\hat{\beta}_{2jk}^{(i)} > 0\}\right]. \tag{S12.5}$$

Though, in many cases in Section 5, the Markov chain during the Bayesian factor analysis did not sample β_{2jk}'s of opposite sign. The estimate for the lfsr using (S12.4) would then be 0. As it is often desirable to obtain a ranking of the most significant genes, this is an unappealing feature. We

instead use a normal approximation to estimate the the lfsr's, using the posterior means and standard deviations from the samples of the β_{2jk}'s.

S13 Simulation approach

We now use simulations based on real data to compare methods (focusing on methods that use control genes, although the same simulations could be useful more generally). In brief, we use random subsets of real expression data to create "null data" that contains real (but unknown) "unwanted variation", and then modify these null data to add (known) signal. We compare methods in their ability to reliably detect real signals and avoid spurious signals. Because they are based on real data, our simulations involve realistic levels of unwanted variation. However, they also represent a "best-case" scenario in which treatment labels were randomized with respect to the factors causing this unwanted variation. They also represent a best case scenario in that the control genes given to each method are simulated to be genuinely null. Even in this best-case scenario unwanted variation is a major issue, and, as we shall see, obtaining well calibrated inferences is challenging.

In more detail: we took the top 1000 expressed genes from the RNA-seq data on muscle samples from the Genotype Tissue Expression Consortium

(GTEx Consortium, 2015). For each dataset in our simulation study, we randomly selected n samples (n = 6, 10, 20, or 40). We then randomly assigned half of these samples to be in one group and the other half to be in a second group. So our design matrix $\boldsymbol{X} \in \mathbb{R}^{n\times 2}$ contains two columns — a column of ones and a column that is an indicator for group assignment.

At this point, all genes are theoretically null, as in the datasets of our introduction (Section). We used this "all-null" scenario as one setting in our simulation studies (similar to the simulations in Rocke et al. (2015)). For other settings, we added signal to a randomly selected proportion of genes $\pi_1 = 1 - \pi_0$ (π_1 = 0.1 or 0.5). To add signal, we first sampled the effect sizes from a $N(0, 0.8^2)$. The standard deviation, 0.8, was chosen by trial and error so that the classification problem would be neither too easy nor too hard. Let

$$a_{j_1}, \ldots, a_{j_{\pi_1 p}} \overset{iid}{\sim} N(0, 0.8^2), \tag{S13.1}$$

be the effect sizes, where $j_\ell \in \Omega$, the set of non-null genes. Then we drew a $\boldsymbol{W}$ matrix of the same dimension as our RNA-seq count matrix $\boldsymbol{Z}$ by

$$w_{ij_\ell}|z_{ij_\ell} \sim \begin{cases} \text{Binomial}(z_{ij_\ell}, 2^{a_{j_\ell} x_{i2}}) & \text{if } a_{j_\ell} < 0 \text{ and } j_\ell \in \Omega, \\ \text{Binomial}(z_{ij_\ell}, 2^{-a_{j_\ell}(1-x_{i2})}) & \text{if } a_{j_\ell} > 0 \text{ and } j_\ell \in \Omega \\ \delta(z_{ij_\ell}) & \text{if } j_\ell \notin \Omega, \end{cases} \tag{S13.2}$$

where $\delta(z_{ij_\ell})$ indicates a point mass at z_{ij_ℓ}. We then used $\boldsymbol{W}$ as our new RNA-seq matrix. Prior to running each method, we took a simple $\log_2$ transformation of the elements in $\boldsymbol{W}$ to obtain our $\boldsymbol{Y}$ matrix.

To motivate this approach, suppose $z_{ij} \sim \text{Poisson}(\lambda_j)$, then

$$[w_{ij}|a_j, a_j < 0, j \in \Omega] \sim \text{Poisson}(2^{a_j x_{i2}} \lambda_j) \tag{S13.3}$$

$$[w_{ij}|a_j, a_j > 0, j \in \Omega] \sim \text{Poisson}(2^{-a_j(1-x_{i2})} \lambda_j). \tag{S13.4}$$

Hence,

$$\begin{aligned} &E[\log_2(w_{ij}) - \log_2(w_{kj})|a_j,\ a_j < 0,\ j \in \Omega] \\ &\quad \approx a_j x_{i2} - a_j x_{k2} = a_j(x_{i2} - x_{k2}),\ \text{and} \end{aligned} \tag{S13.5}$$

$$\begin{aligned} &E[\log_2(w_{ij}) - \log_2(w_{kj})|a_j,\ a_j > 0,\ j \in \Omega] \\ &\quad \approx -a_j(1 - x_{i2}) + a_j(1 - x_{k2}) = a_j(x_{i2} - x_{k2}). \end{aligned} \tag{S13.6}$$

So a_j is approximately the $\log_2$-fold difference between the two groups.

S14 Simulation Method with Correlated Confounders

In this section, we extend the approach in Section S13 to simulate RNA-seq data where group assignment is correlated with latent factors. As in Section S13, this simulation method uses real RNA-seq data and does not assume the form of the unwanted variation — beyond that it can be adequately represented by a low-rank term, as in (2.1). The strength of the correlation

may be controlled by the user under very mild constraints.

As in (2.1), we let $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$ be the matrix of log-count RNA-seq expression data of p genes on n individuals. We first normalize $\boldsymbol{Y}$ by subtracting each gene-specific mean.

$$\boldsymbol{E} := (\boldsymbol{I}_n - \mathbf{1}_n\mathbf{1}_n^\intercal/n)\boldsymbol{Y}, \tag{S14.1}$$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix and $\mathbf{1}_n$ is the n-vector of ones. We then extract q latent factors by taking the SVD of $\boldsymbol{E}$,

$$\boldsymbol{E} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\intercal, \tag{S14.2}$$

and then take the first q columns of $\boldsymbol{U}$ to be the latent factors. If q is not specified ahead of time, then we propose to estimate the number of latent factors either using parallel analysis (Buja and Eyuboglu, 1992), bicross-validation (Owen and Wang, 2016), or the eigenvalue difference method (Onatski, 2010).

Let

$$\tilde{\boldsymbol{U}} := \sqrt{n}\boldsymbol{U}_{[,1:q]} \tag{S14.3}$$

contain the first q columns of $\boldsymbol{U}$, normalized to have variance 1 (this is what multiplying $\boldsymbol{U}$ by $\sqrt{n}$ does). Note that the columns of $\boldsymbol{U}$ are already normalized to have mean 0 since they are constrained to be orthogonal to $\mathbf{1}_n$ by the normalization in (S14.1).

We will generate a random variable, $\boldsymbol{w} \in \mathbb{R}^n$, that is correlated with the columns of $\tilde{\boldsymbol{U}}$. Let $\boldsymbol{r} \in \mathbb{R}^q$ be a vector of user-supplied correlations. That is, r_i is the correlation between $\boldsymbol{w}$ and $\tilde{\boldsymbol{u}}_i$, where $\tilde{\boldsymbol{u}}_i$ is the ith column of $\tilde{\boldsymbol{U}}$ and r_i is the ith element of $\boldsymbol{r}$. The only constraint on $\boldsymbol{r}$ is that $\boldsymbol{r}^\intercal \boldsymbol{r} < 1$, so that the resulting correlation matrix in (S14.5) is positive definite. Then we generate $\boldsymbol{w}$ by

$$\boldsymbol{w} \sim N(\tilde{\boldsymbol{U}}\boldsymbol{r}, (1 - \boldsymbol{r}^\intercal \boldsymbol{r})\boldsymbol{I}_n). \tag{S14.4}$$

The motivation of (S14.4) is by assuming that each row of $(\boldsymbol{w}, \tilde{\boldsymbol{U}})$ is iid multivariate normal with mean $\boldsymbol{0}_{q+1}$ and covariance

$$\begin{pmatrix} 1 & \boldsymbol{r}^\intercal \\ \boldsymbol{r} & \boldsymbol{I}_q \end{pmatrix}, \tag{S14.5}$$

and then simulating the elements of $\boldsymbol{w}$ from a conditional normal, conditional on the values of the elements of $\tilde{\boldsymbol{U}}$.

Once we have $\boldsymbol{w}$, we assign groups based on the sign of $\boldsymbol{w}$.

$$x_i = \begin{cases} 0 & \text{if } w_i < 0 \\ 1 & \text{if } w_i > 0. \end{cases} \tag{S14.6}$$

We then perform Poisson thinning as in Section S13 using these group assignments.

Note that we have specified the correlation (r_i) of $\boldsymbol{w}$ with each latent

factor, not the correlation of the the grouping variable $\boldsymbol{x}$ with each latent factor. Though we can calculate the resulting correlation for a given input $\boldsymbol{r}$ by a simple result.

Proposition 1. *Let* (w, u) *be bivariate normally distributed with mean* $(0, 0)$ *and covariance*

$$\begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}. \tag{S14.7}$$

Then

$$\operatorname{cov}(1(w > 0), u) = \frac{r}{\sqrt{2\pi}}, \text{ and} \tag{S14.8}$$

$$\operatorname{cor}(1(w > 0), u) = r\sqrt{\frac{2}{\pi}} \tag{S14.9}$$

where $1(w > 0)$ *is the indicator function.*

Proof. We use the law of total covariance.

$$\operatorname{cov}(1(w > 0), u) = E[\operatorname{cov}(1(w > 0), u|w)] + \operatorname{cov}(E[1(w > 0)|w], E[u|w]) \tag{S14.10}$$

$$= \operatorname{cov}(E[1(w > 0)|w], E[u|w]) \tag{S14.11}$$

$$= \operatorname{cov}(1(w > 0), rw) \tag{S14.12}$$

$$= rE[(1(w > 0) - E[1(w > 0)])(w - E[w])] \tag{S14.13}$$

$$= rE[w1(w > 0) - w/2] \tag{S14.14}$$

$$= rE[w1(w > 0)] \tag{S14.15}$$

$$= r \int_0^\infty wN(w|0, 1)dw \tag{S14.16}$$

$$= r/\sqrt{2\pi}. \tag{S14.17}$$

Equation (S14.11) follows since $1(w > 0)$ is not random given w. Equation (S14.12) follows by properties of the bivariate normal distribution. Equation (S14.17) follows by basic substitution. Equation (S14.9) is immediate since $\operatorname{var}(1(w > 0)) = 1/4$ and $\operatorname{var}(u) = 1$. □

In Section S16, we will run a simulation study with $r_i \in \{0, 0.25, 0.5, 0.75\}$. This corresponds to correlations with the group assignment variable of approximately 0, 0.2, 0.4, and 0.6.

S15 Additional simulation considerations

To implement the methods in the simulation studies in Section 5, there are additional technicalities to be considered. Here, we briefly list out our choices.

The number of factors, q, is required to be known for all methods that we examined. There are many approaches to estimate this in the literature (for a review, see Owen and Wang, 2016). We chose to use the parallel analysis approach of Buja and Eyuboglu (1992) as implemented in the `num.sv` function in the R package `sva` (Leek et al., 2016). An alternative choice could have been the bi-cross-validation approach described in Owen and Wang (2016), or the eigenvalue difference method of Onatski (2010), both implemented in the `cate` R package (Wang and Zhao, 2015).

RUV2, RUV3, RUV4, and CATE all require specifying a factor analysis (Definition 1) for their respective first steps. For all methods, we used the truncated SVD (Section S1). This was to make the methods more comparable. Though we note that Wang et al. (2017) also suggest using the quasi-maximum likelihood approach of Bai and Li (2012).

For RUVB, we ran each Gibbs sampler (Algorithm 1) for 12,500 iterations, dropping the first 2,500 iterations as burn-in. We kept every 10th sample to retain 1000 posterior draws from which we calculated the poste-

rior summaries of Section S12. Convergence diagnostics and other checks were implemented on a sample of the Markov chains in our simulation studies. We detected no problems with convergence (data not shown).

Specifically, for the non-RUVB methods we considered:

- Variance moderation (EBVM) versus no moderation.
- Variance calibration (both MAD and control-gene based) vs no calibration.
- For RUV3 and CATE: EBVM before GLS or after GLS.
- For CATE: additive variance inflation vs no additive variance inflation.

Altogether this gave 6 different OLS approaches, 6 RUV2 approaches, 9 RUV3 approaches, 6 RUV4 approaches, and 15 CATE approaches. (We did not implement CATE with additive variance inflation and EBVM before GLS because this implementation is not straightforward given the current `cate` software.)

Since we explored many combinations of methods, we adopt the following notation in Supplementary Figures S3 and S4:

- o = original variance estimates,
- m = MAD variance calibration,
- c = control-gene variance calibration,
- l = limma-moderated variances (EBVM),

- lb = limma-moderated variances (EBVM) before GLS (for either CATE or RUV3),
- la = limma-moderated variances (EBVM) after GLS (for either CATE or RUV3),
- d = delta-adjustment from CATE package (additive variance inflation),
- n = t approximation for the likelihood in RUVB,
- nn = normal approximation for the likelihood in RUVB.

When comparing the AUC of different methods, certain combinations of methods theoretically have the same AUC. Specifically, applying MAD variance calibration (m) or control-gene variance calibration (c) does not alter the AUC of a method. Thus, we only need to compare one method of each of these groups to obtain comprehensive results on AUC performance. The members of these groups are those shown in Supplementary Figure S3.

The effect of library size (the total number of gene-reads in a sample) is a well-known source of bias in RNA-seq data (Dillies et al., 2013). We do not explicitly adjust for library size. In this paragraph, we briefly argue that RUV-like methods can effectively account for library size. The usual pipeline to adjust for library size is to choose a constant for each sample, c_i, and divide each count in a sample by c_i. Many proposals have been made

to estimate c_i (Anders and Huber, 2010; Bullard et al., 2010; Robinson and Oshlack, 2010). There are also variations on this pipeline. For example, others choose a constant for each sample, c_i, and include the $\boldsymbol{c} = (c_1, \ldots, c_n)^\intercal$ vector as a covariate in the regression model (Langmead et al., 2010). In terms of our $\log_2$-count matrix of responses $\boldsymbol{Y}$, this corresponds to fitting the model

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{c}\boldsymbol{d}^\intercal + \boldsymbol{E}, \tag{S15.1}$$

where $\boldsymbol{c}$ is estimated independently of $\boldsymbol{X}$ in some (ad-hoc) fashion and $\boldsymbol{d}$ may or may not be assumed to be the vector of ones, $\boldsymbol{1}_p$. However, equation (S15.1) is just a factor-augmented regression model, the same as (2.1). Methods that assume model (2.1) thus need not adjust for library size and need not choose one of the many procedures to estimate $\boldsymbol{c}$. That is, library size can be treated as just another source of unwanted variation.

S16 Simulation Study with Correlated Confounders

In this section, we used the method described in Section S14 to run a simulation study to explore the effect of correlated latent factors on the the procedures discussed in this paper. We simulated data under the following settings:

- Sample size: $n \in \{6, 10, 20, 40\}$.
- Number of genes: $p = 1000$,
- Number of control genes: $m = 100$,
- Proportion of null genes: $\pi = 0.9$.

We also varied the strength of the correlation of the group assignment with the latent factors. We allowed the variable of interest to be correlated only with one latent factor with correlation $r \in \{0, 0.25, 0.5, 0.75\}$. We also allowed the variable of interest to be correlated with two latent factors with correlation $\boldsymbol{r} \in \{(0, 0), (0.25, 0.25), (0.5, 0.5)\}$. We used the same methods described in Section 5.1 and the same means of evaluation as in Sections 5.2 and 5.3.

In Supplementary Figure S5, we took the best performing methods (in terms of mean AUC) within each class of methods (RUV2, RUV3, and RUV4) and compared their mean AUC's to that of RUVB. The results are very similar to those presented in Figure 3a — namely, that RUVB performs better than the other methods in terms of AUC. The major insight from Supplementary Figure S5 is that RUV2 appears to be less robust to correlations with unobserved confounders than the other forms of RUV. RUV3 is also less robust when the correlation is high — which intuitively makes sense since RUV3 is intermediate to RUV2 and RUV4.

We also present box plots of each method's empirical coverage of their 95% confidence intervals in Supplementary Figure S6. As in Figure 3c, all methods have approximately correct coverage. However, for higher levels of correlation, the coverage of RUV3 and RUV4 is more unstable from dataset to dataset.

S17 Simulations with Misspecified Negative Controls

In this section, we ran simulations to assess each method's sensitivity to the negative controls assumption.

We simulated RNA-seq data using the procedure outlined in Section S13 under the following conditions:

- Sample size: $n \in \{6, 10, 20, 40\}$.
- Number of genes: $p = 1000$,
- Number of control genes: $m = 100$,
- Proportion of null genes: $\pi = 0.9$.

We also set a proportion of the specified control genes to be true controls (so their expression levels are not associated with the group assignment) and the remainder of control genes to be false controls (so their expression levels *are* associated with the group assignment). We set this proportion to be 0.25, 0.5, 0.75, or 1 — where "1" indicates that all specified controls

are true controls, "0.75" indicates that only 75 of the 100 controls are true controls, etc. We used the same methods described in Section 5.1 and the same means of evaluation as in Sections 5.2 and 5.3.

Supplementary Figure S7 contains comparisons of mean AUC between RUVB and the best performing methods in each class (RUV2, RUV3, and RUV4). The results here indicate that, in terms of AUC, RUVB and RUV2 are very sensitive to the negative controls assumption, while RUV3 and RUV3 are relatively robust to the negative controls assumption. Though RUVB seems to be less sensitive than RUV2. The sensitivity of RUV2 to the negative controls assumption was studied in Gagnon-Bartsch et al. (2013). The fact that RUV3 is robust to the negative controls assumption seems to indicate that it is more "RUV4-like" than "RUV2-like". This is probably because of the regression step in (3.2) to estimate the latent factors, as hypothesized by an anonymous reviewer.

Coverage results in Supplementary Figure S8 also indicate that that RUVB is more sensitive to the negative controls assumption. For small sample sizes or small levels of misspecified controls, RUVB has approximately unbiased coverage. However, for large amounts of misspecified controls RUVB has coverage much less than 0.95, and this bias is worse for larger sample sizes. The other methods, though not unbiased, have conser-

vative coverage in the presence of misspecified controls. Thus, the use of RUVB should be limited to cases where one has access to very high quality negative controls.

S18 Supplementary figures

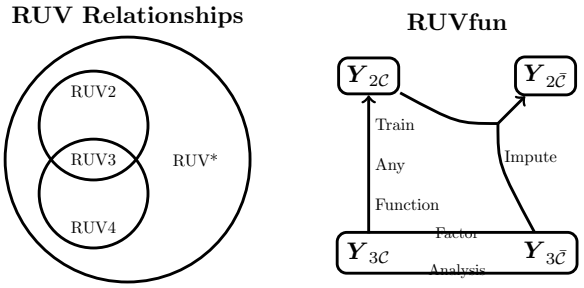

Figure S1: The left panel is a Venn diagram representing the relationships between RUV2, RUV3, RUV4, and RUV*. The right panel is a pictorial representation of RUVfun from Gagnon-Bartsch et al. (2013)

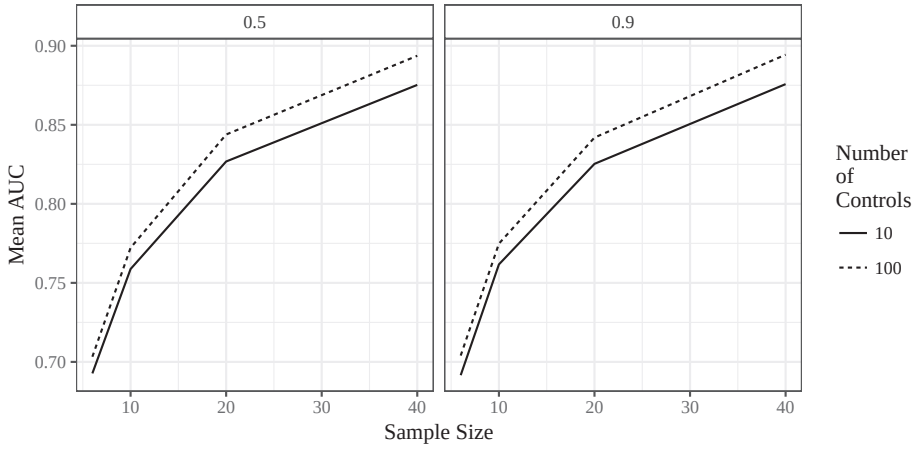

Figure S2: Mean AUC of RUVBnl (see Section S15 for notation) on the y-axis plotted against sample size on the x-axis. The column facets distinguish between the proportion of genes that are null. The line type distinguishes between the number of control genes used.

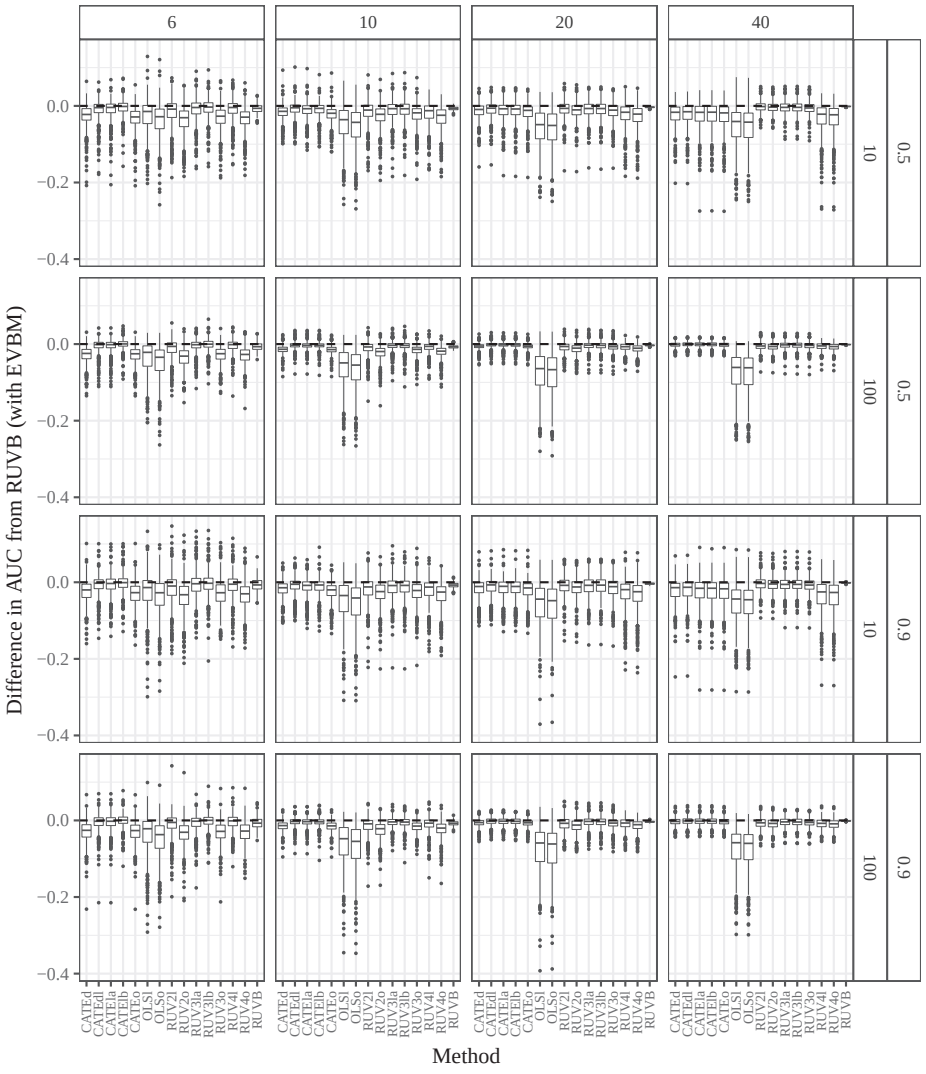

Figure S3: Boxplots of AUC of methods subtracted from the AUC of RUVB (with EBVM). Anything above zero (the dashed horizontal line) indicates superior performance to RUVB. Anything below the line indicates inferior performance to RUVB. Columns are the sample sizes, rows are the number of control genes by the proportion of genes that are null. For the notation of the methods, see Section S15.

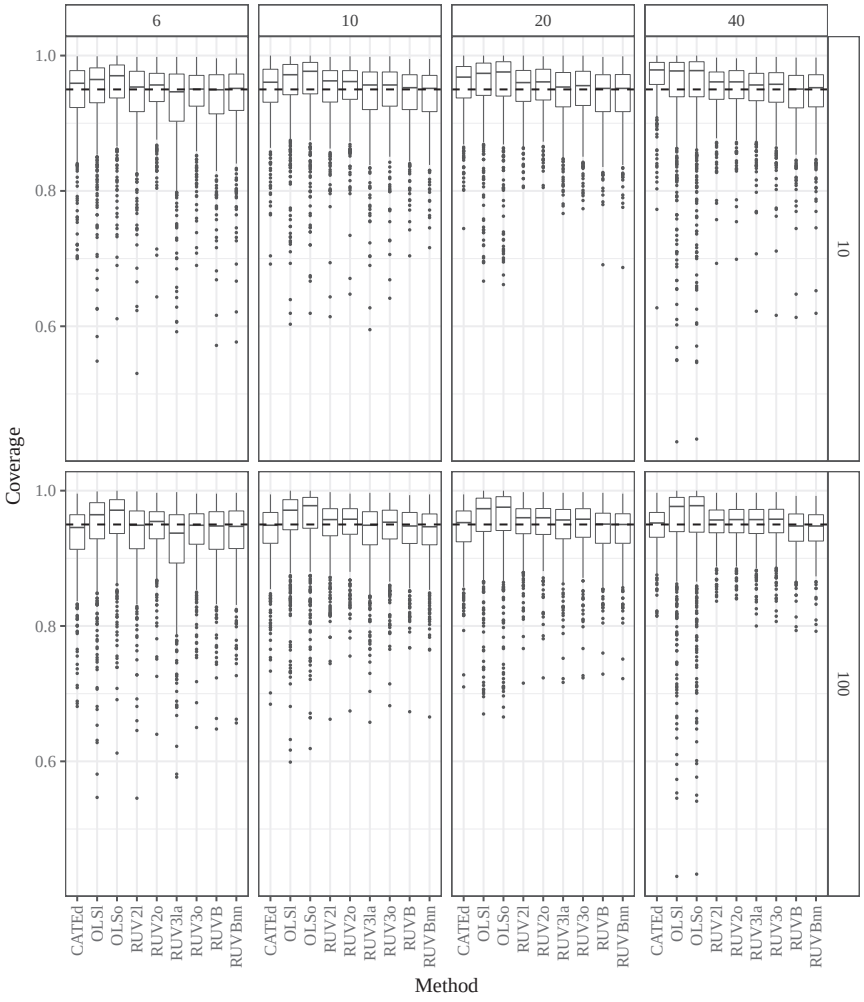

Figure S4: Boxplots for the coverage of the 95% confidence intervals for the best-performing methods when $\pi_0 = 0.5$. The column facets index the sample sizes used while the horizontal facets index the number of control genes used. The horizontal dashed line is at 0.95. For the notation of the methods, see Section S15.

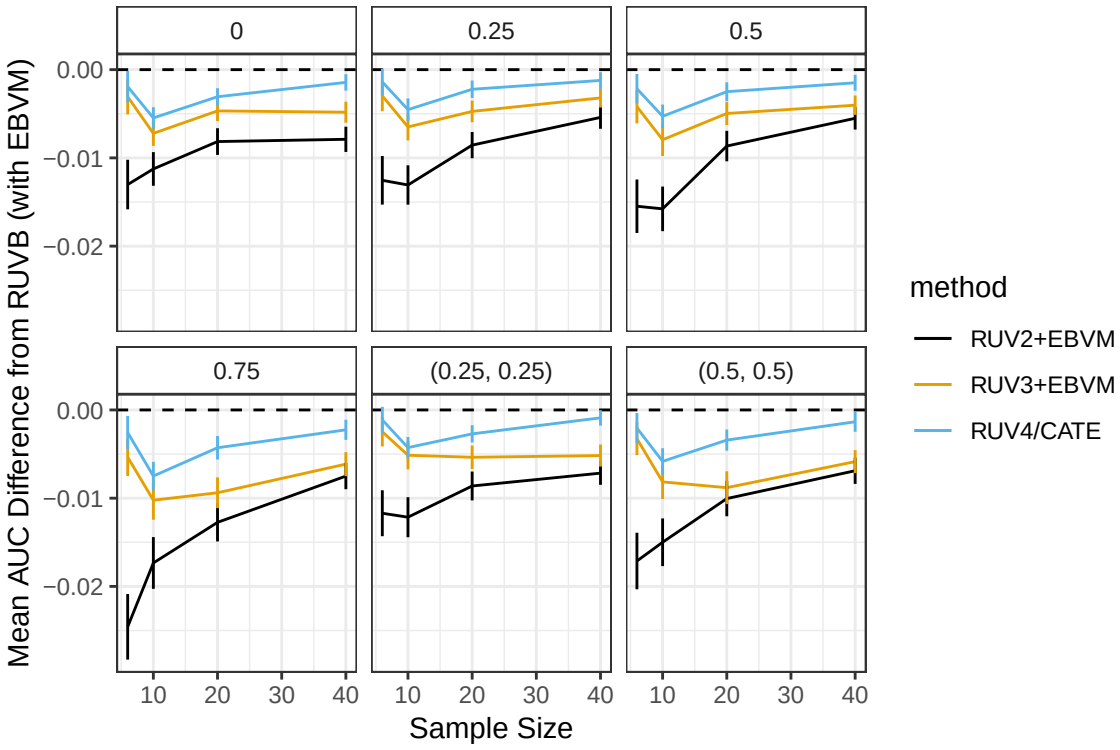

Figure S5: Sample size (x-axis) versus AUC (y-axis) of best-performing methods subtracted from AUC of RUVB (with EBVM). Anything above zero (the dashed horizontal line) indicates superior performance of RUVB. Anything below the line indicates inferior performance to RUVB. The different facets index the different correlation structures of the primary variable with the latent variables.

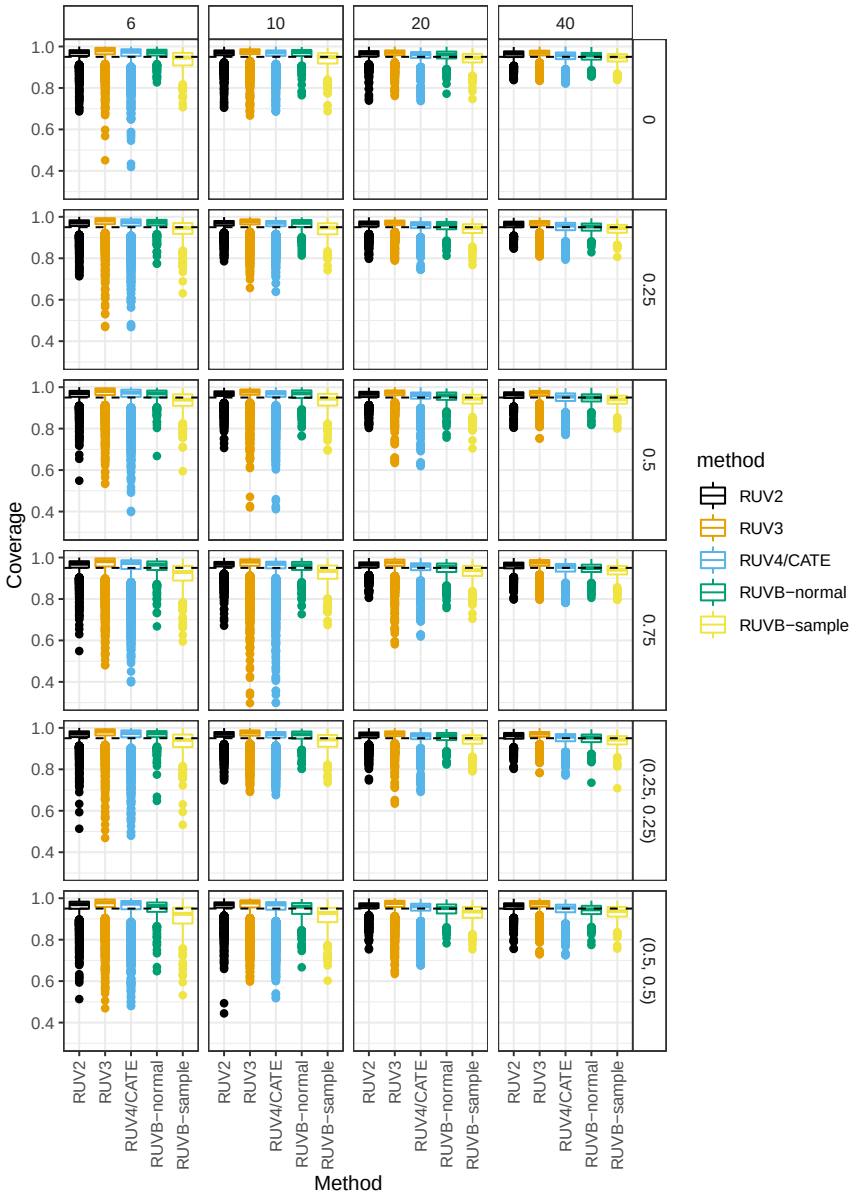

Figure S6: Boxplots of coverage (y-axis) of 95% confidence intervals for the best performing methods (x-axis) in each class of methods. The row facets index the different correlation structures and the column facets index the different sample sizes. The dashed horizontal line is at 0.95. The best performing methods should be close to this dashed line.

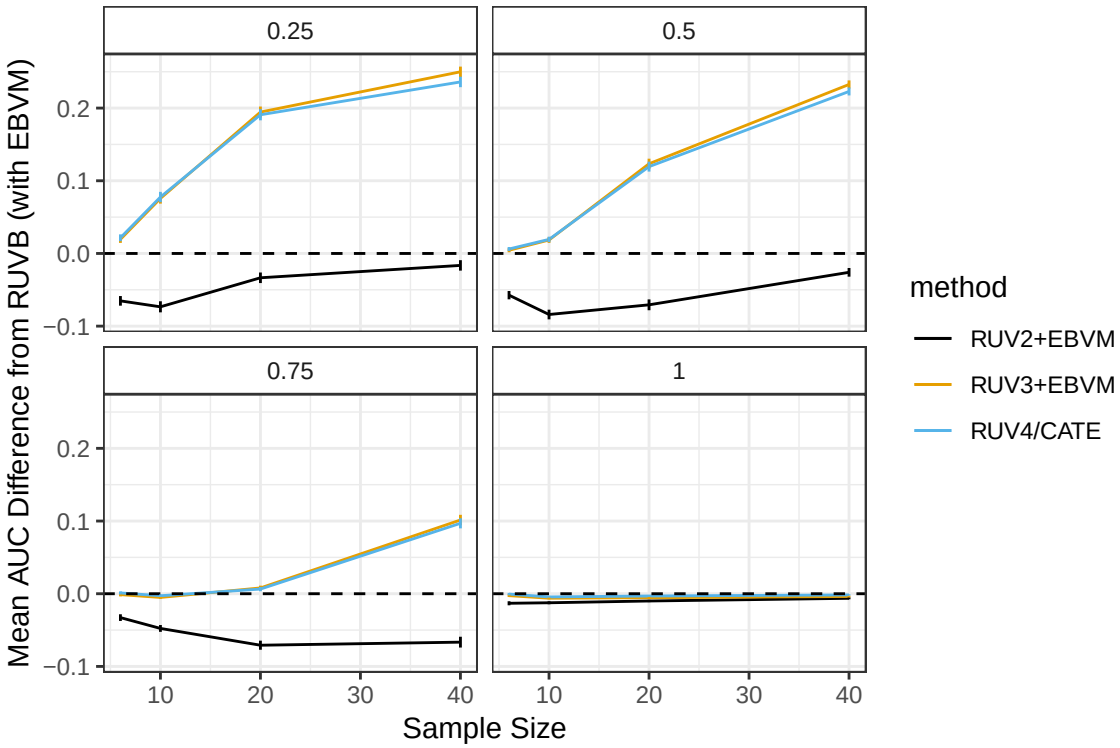

Figure S7: Sample size (x-axis) versus AUC (y-axis) of best-performing methods subtracted from AUC of RUVB (with EBVM). Anything above zero (the dashed horizontal line) indicates superior performance of RUVB. Anything below the line indicates inferior performance to RUVB. The different facets index the proportion of negative controls that are correctly specified.

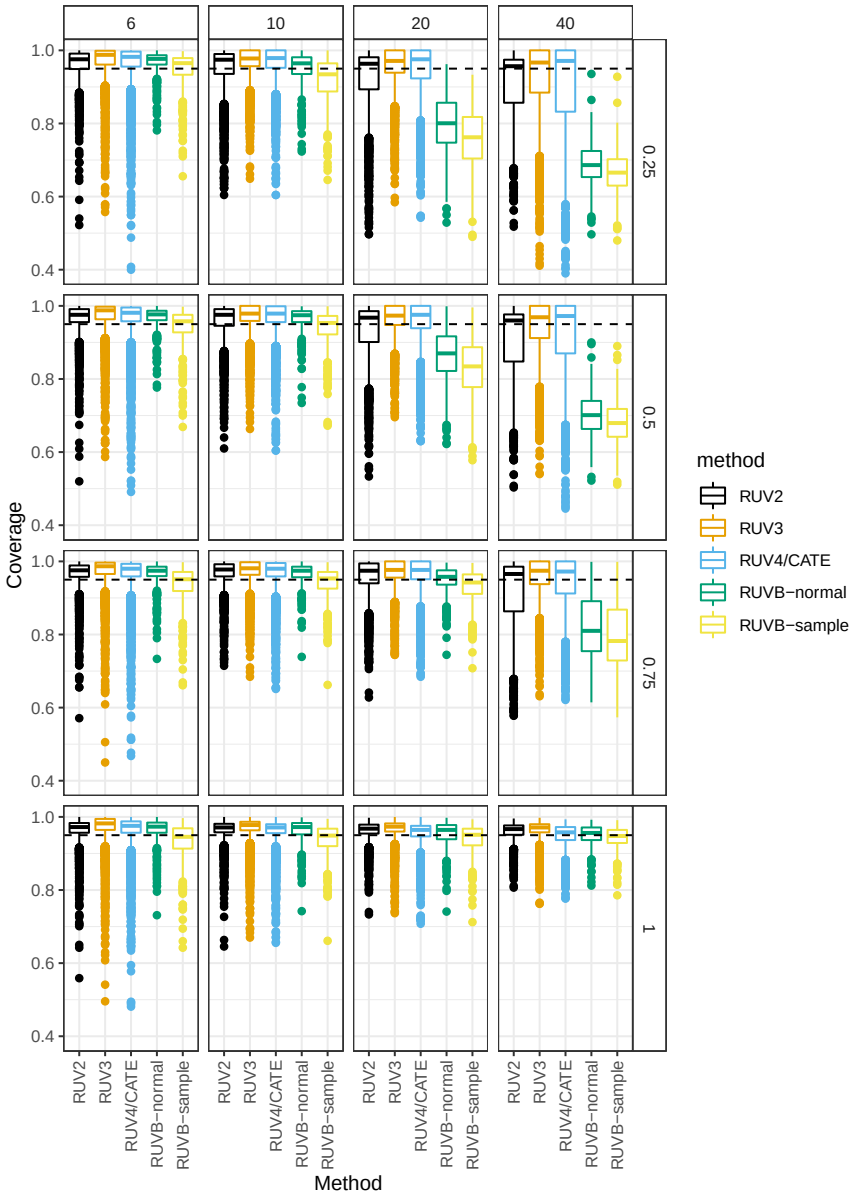

Figure S8: Boxplots of coverage (y-axis) of 95% confidence intervals for the best performing methods (x-axis) in each class of methods. The row facets index the proportion of correctly specified negative controls and the column facets index the different sample sizes. The dashed horizontal line is at 0.95. The best performing methods should be close to this dashed line.

Bibliography

Allen, G. I. and R. Tibshirani (2010). Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics 4*(2), 764–790.

Anders, S. and W. Huber (2010). Differential expression analysis for sequence count data. *Genome biology 11*(10), 1.

Bai, J. and K. Li (2012). Statistical analysis of factor models of high dimension. *Ann. Statist. 40*(1), 436–465.

Buja, A. and N. Eyuboglu (1992). Remarks on parallel analysis. *Multivariate behavioral research 27*(4), 509–540.

Bullard, J. H., E. Purdom, K. D. Hansen, and S. Dudoit (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC bioinformatics 11*(1), 1.

Dillies, M.-A., A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estellé, G. Guernec, B. Jagla, L. Jouneau, D. Laloë, C. L. Gall, B. Schaeffer, S. L. Crom, M. Guedj, and F. Jaffrézic (2013). A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in bioinformatics 14*(6), 671–683.

Gagnon-Bartsch, J. (2015). *ruv: Detect and Remove Unwanted Variation using Negative Controls*. R package version 0.9.6.

Gagnon-Bartsch, J., L. Jacob, and T. Speed (2013). Removing unwanted variation from high dimensional data with negative controls. Technical report, Technical Report 820, Department of Statistics, University of California, Berkeley.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Anal. 1*(3), 515–534.

Ghosh, J. and D. B. Dunson (2009). Default prior distributions and efficient posterior computation in Bayesian factor analysis. *Journal of Computational and Graphical Statistics 18*(2), 306–320.

GTEx Consortium (2015). The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science 348*(6235), 648–660.

Langmead, B., K. D. Hansen, and J. T. Leek (2010). Cloud-scale RNA-sequencing differential expression analysis with myrna. *Genome biology 11*(R83).

Leek, J. T., W. E. Johnson, H. S. Parker, E. J. Fertig, A. E. Jaffe, and

J. D. Storey (2016). *sva: Surrogate Variable Analysis*. R package version 3.20.0.

Leung, D. and M. Drton (2016). Order-invariant prior specification in Bayesian factor analysis. *Statistics & Probability Letters 111*, 60–66.

Onatski, A. (2010). Determining the number of factors from empirical distribution of eigenvalues. *The Review of Economics and Statistics 92*(4), 1004–1016.

Owen, A. B. and J. Wang (2016). Bi-cross-validation for factor analysis. *Statist. Sci. 31*(1), 119–139.

Robinson, M. D. and A. Oshlack (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome biology 11*(3), 1.

Rocke, D. M., L. Ruan, Y. Zhang, J. J. Gossett, B. Durbin-Johnson, and S. Aviran (2015). Excess false positive rates in methods for differential gene expression analysis using RNA-seq data. *bioRxiv*.

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology 3*(1).

Stephens, M. (2017). False discovery rates: a new deal. *Biostatistics 18*(2), 275–294.

Sun, Y., N. R. Zhang, and A. B. Owen (2012). Multiple hypothesis testing adjusted for latent variables, with an application to the AGEMAP gene expression data. *Ann. Appl. Stat. 6*(4), 1664–1688.

Trotter, H. F. and J. W. Tukey (1956). Conditional Monte Carlo for normal samples. In H. A. Meyer (Ed.), *Symposium on Monte Carlo methods*, pp. 64–79. Wiley.

Wang, J. and Q. Zhao (2015). *cate: High Dimensional Factor Analysis and Confounder Adjusted Testing and Estimation*. R package version 1.0.4.

Wang, J., Q. Zhao, T. Hastie, and A. B. Owen (2017). Confounder adjustment in multiple hypothesis testing. *Ann. Statist. 45*(5), 1863–1894.