

Different Implementations of RUVASH

Abstract

I run RUVASH under six different scenarios and RUV + ASH under three different scenarios.

Motivation

The model is

$$Y = X\beta + Z\alpha + E, \quad (1)$$

$$E \sim N_{n \times p}(0, \Sigma \otimes I_n) \quad (2)$$

$$\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_p^2). \quad (3)$$

RUVASH will obtain estimates \hat{Z} , $\hat{\alpha}$, $\hat{\Sigma}$, and a variance inflation term $\hat{\lambda}$. In order to obtain estimates of the variances of $\hat{\beta}$ (in order to use in running ASH), I've been multiplying $\hat{\lambda}\hat{\Sigma}$ by the diagonal elements of $(X^T X)^{-1}$. I think this is wrong and what I should be doing is multiplying $\hat{\lambda}\hat{\Sigma}$ by the diagonal elements of $([X, \hat{Z}]^T [X, \hat{Z}])^{-1}$. I run comparisons of this approach here.

Simulation Setup

I ran through 100 repetitions of generating data from GTEX muscle data under the following parameter conditions:

- $n \in \{10, 20, 40\}$,
- $p = 1000$.
- $\pi_0 \in \{0.5, 0.9, 1\}$,
- The alternative distribution being just a standard normal. New alternatives are generated every iteration.

I extracted the most expressed p genes from the GTEX muscle data and n samples are chosen at random. Half of these samples are randomly given the “treatment” label 1, the other half given the “control” label 0. Of the p genes, $\pi_0 p$ were chosen to be non-null. Signal was added by a Poisson-thinning approach, where the log-2 fold change was sampled from a standard normal. That is

$$A_1, \dots, A_{p/2} \sim N(0, 1) \quad (4)$$

$$B_i = 2^{A_i} \text{ for } i = 1, \dots, p/2, \quad (5)$$

If $A_i > 0$ then we replace $Y_{[1:(n/2), i]}$ with $\text{Binom}(Y_{[j, i]}, 1/B_i)$ for $j = 1, \dots, n/2$. If $A_i < 0$ then we replace $Y_{[(n/2+1):n, i]}$ with $\text{Binom}(Y_{[j, i]}, B_i)$ for $j = n/2 + 1, \dots, n$.

I now describe the justification for this. Suppose that

$$Y_{ij} \sim \text{Poisson}(\lambda_j). \quad (6)$$

Let x_i be the indicator of treatment vs control for individual i . Let Ω be the set of non-null genes. Let Z be the new dataset derived via the steps above. That is

$$Z_{ij}|Y_{ij} = \begin{cases} \text{Binom}(Y_{ij}, 2^{A_j x_i}) & \text{if } A_j < 0 \text{ and } j \in \Omega \\ \text{Binom}(Y_{ij}, 2^{-A_j(1-x_i)}) & \text{if } A_j > 0 \text{ and } j \in \Omega \\ Y_{ij} & \text{if } j \notin \Omega. \end{cases} \quad (7)$$

Then

$$Z_{ij}|A_j, A_j < 0, j \in \Omega \sim \text{Poisson}(2^{A_j x_i} \lambda_j) \quad (8)$$

$$Z_{ij}|A_j, A_j > 0, j \in \Omega \sim \text{Poisson}(2^{-A_j(1-x_i)} \lambda_j), \quad (9)$$

and

$$E[\log_2(Z_{ij}) - \log_2(Z_{kj})|A_j, A_j < 0, j \in \Omega] \approx A_j x_i - A_j x_k, \text{ and} \quad (10)$$

$$E[\log_2(Z_{ij}) - \log_2(Z_{kj})|A_j, A_j > 0, j \in \Omega] \approx -A_j(1-x_i) + A_j(1-x_k). \quad (11)$$

if individual i is in the treatment group and individual k is in the control group, then this just equals A_j . I treat the A_j 's as the true coefficient values when calculating the MSE.

Methods

The notation in the plots below is:

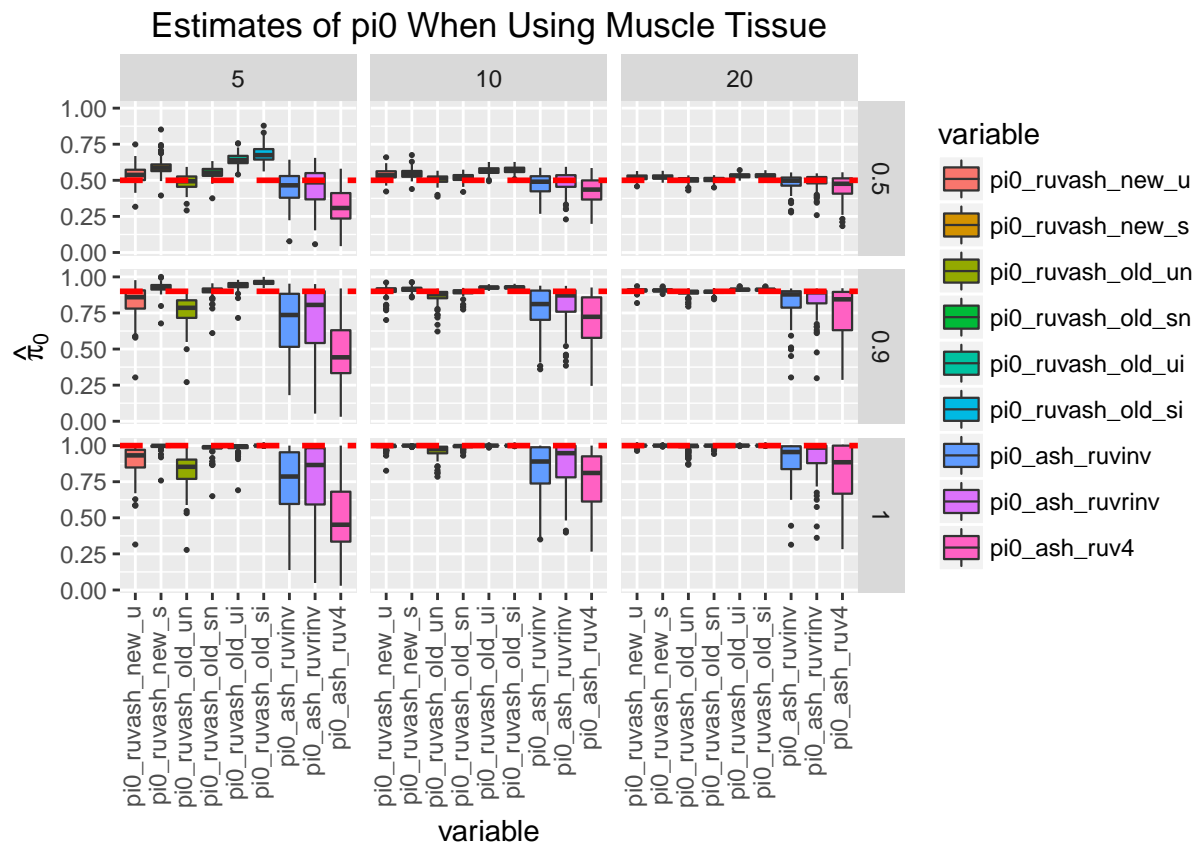
- `ruvash_new_u`: $\hat{\Sigma}$ is not limmashrunk and multiply $\hat{\lambda}\hat{\Sigma}$ by the diagonal elements of $([X, Z]^T[X, Z])^{-1}$.
- `ruvash_new_u`: $\hat{\Sigma}$ is limmashrunk and multiply $\hat{\lambda}\hat{\Sigma}$ by the diagonal elements of $([X, Z]^T[X, Z])^{-1}$.
- `ruvash_old_un`: $\hat{\Sigma}$ is not limmashrunk and multiply $\hat{\lambda}\hat{\Sigma}$ by the diagonal elements of $(X^T X)^{-1}$, but not $n/(n-k-q)$.
- `ruvash_old_sn`: $\hat{\Sigma}$ is limmashrunk and multiply $\hat{\lambda}\hat{\Sigma}$ by the diagonal elements of $(X^T X)^{-1}$, but not $n/(n-k-q)$.
- `ruvash_old_ui`: $\hat{\Sigma}$ is not limmashrunk and multiply $\hat{\lambda}\hat{\Sigma}$ by the diagonal elements of $(X^T X)^{-1}$ and $n/(n-k-q)$.
- `ruvash_old_ui`: $\hat{\Sigma}$ is limmashrunk and multiply $\hat{\lambda}\hat{\Sigma}$ by the diagonal elements of $(X^T X)^{-1}$ and $n/(n-k-q)$.
- `ash_ruvinv`: Run 'ruv::RUVinv' then ASH. RUVinv is RUV4 using the maximum allowed number of factors given the number of control genes, followed by a method-of-moments approach to estimating the variances that the RUV folks call the "inverse-method".
- `ash_ruvrinv`: A ridged version of RUV-inverse.
- `ash_ruv4`: Run 'ruv::RUV4' followed by ASH (no variance inflation).

Results

The new multiplier seems to work slightly better than not including it. I don't think the ad-hoc inflation is really needed.

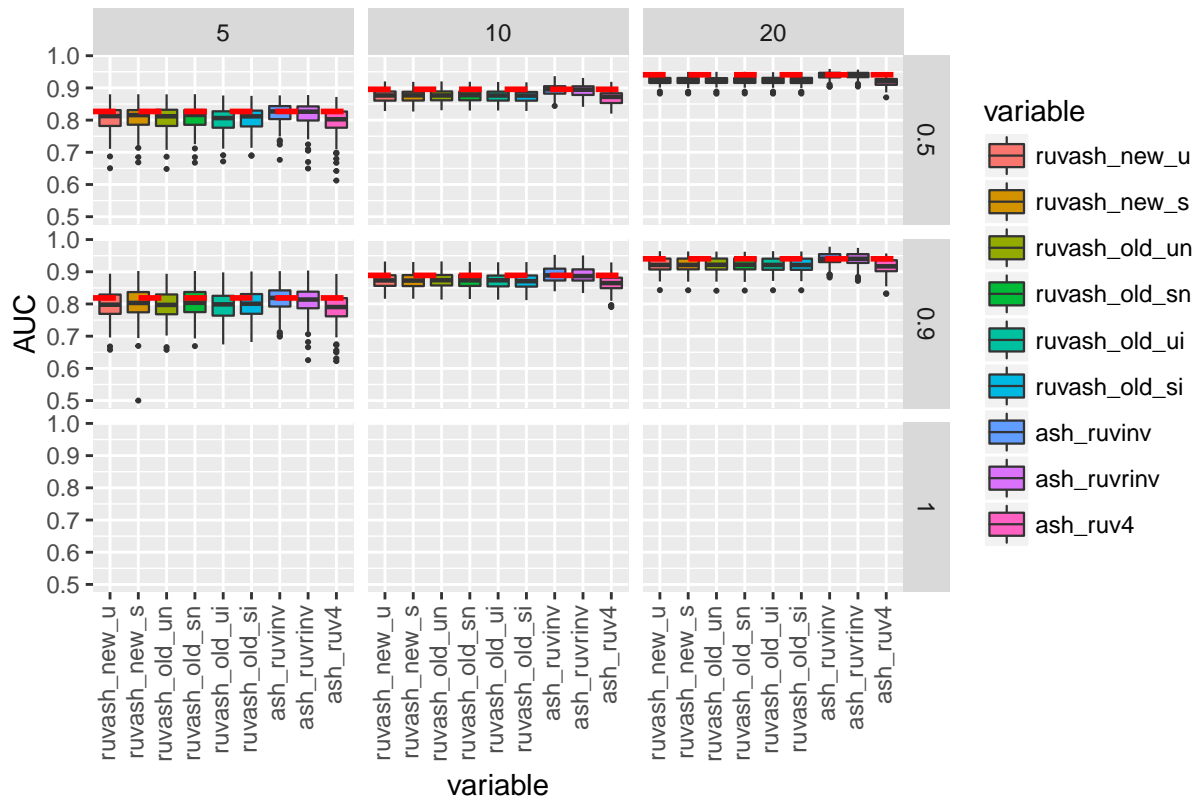
Interestingly, AUC improved a little bit when you use all of the factors you can (for RUVinv). I wonder if RUVASH would work really well in this scenario.

Plots



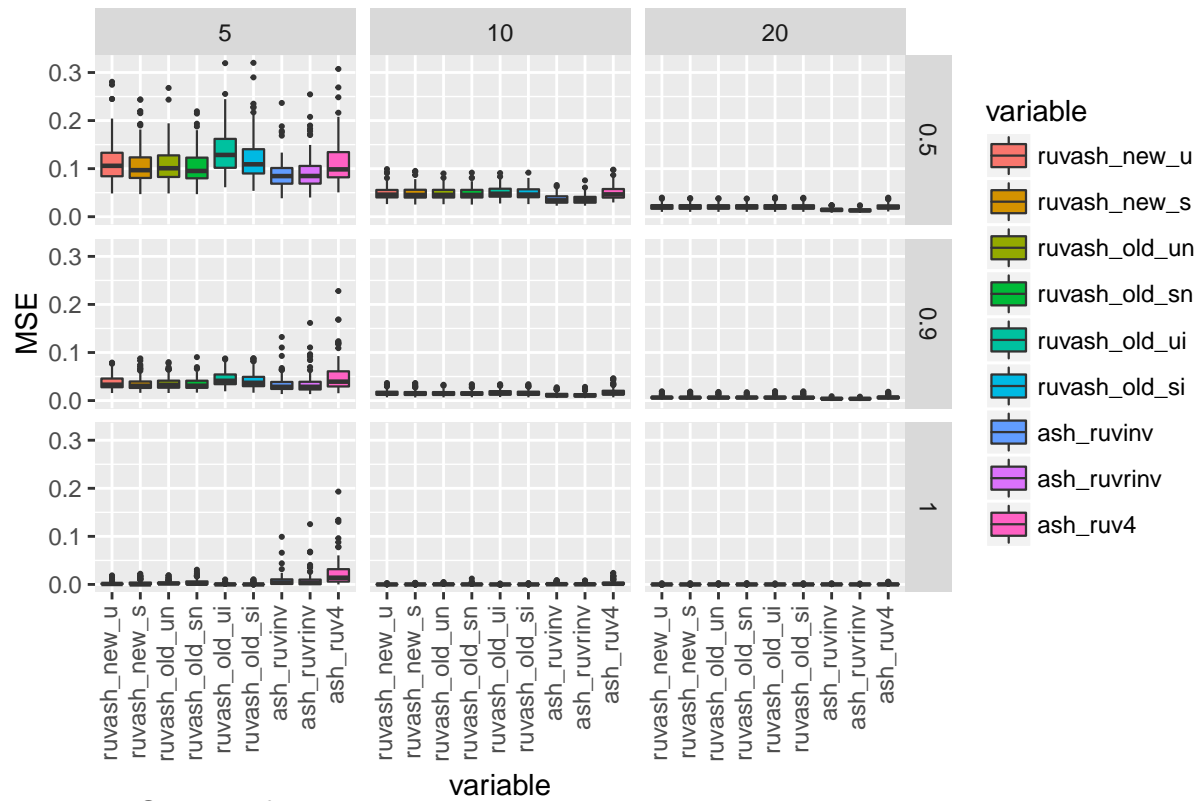
Warning: Removed 2700 rows containing non-finite values (stat_boxplot).

AUC When Using Muscle Tissue

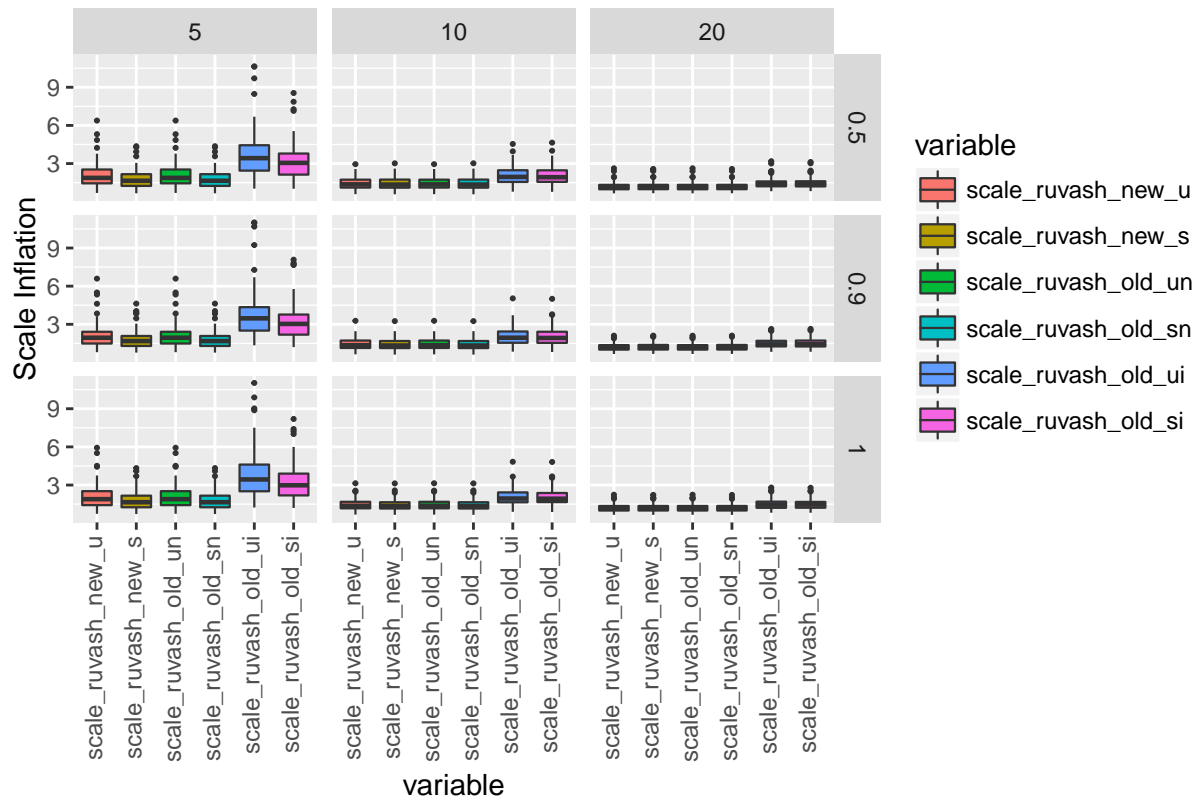


Warning: Removed 1 rows containing non-finite values (stat_boxplot).

MSE When Using Muscle Tissue



Scale Inflation When Using Muscle Tissue



```
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.4 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] tidyr_0.4.1    reshape2_1.4.1 ggplot2_2.1.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.5      knitr_1.12.28    magrittr_1.5      munsell_0.4.3
##  [5] colorspace_1.2-6 R6_2.1.2         stringr_1.0.0     plyr_1.8.4
##  [9] dplyr_0.4.3      tools_3.3.1      parallel_3.3.1    grid_3.3.1
## [13] gtable_0.2.0     DBI_0.4          htmltools_0.3.5   yaml_2.1.13
## [17] digest_0.6.9     assertthat_0.1   formatR_1.3       evaluate_0.9
## [21] rmarkdown_0.9.6  labeling_0.3     stringi_1.0-1     compiler_3.3.1
## [25] scales_0.4.0
```