

Different Alternative Types Simulated Data

David Gerard

2016-06-07

Abstract

I compare various competitors under the same alternative scenarios in Stephens (2016) but using simulated data rather than the GTEx data. The results are more muddled than in the GTEx simulations. Overall, it seems RUVASH does best. SUCCOTASH does much worse than in the GTEx simulations.

Simulation Setup

I ran through 200 repetitions of generating data from the factor-augmented Gaussian regression model:

$$Y_{n \times p} = X_{n \times k} \beta_{k \times p} + Z_{n \times q} \alpha_{q \times p} + E_{n \times p} \quad (1)$$

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

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

under parameter conditions:

- $n \in \{10, 20, 40\}$,
- $p = 1000$,
- $k = 2$,
- $q = 5$
- $\beta_{1j} \stackrel{iid}{\sim} N(0, 1)$,
- β_{2j} being iid a mixture of normals with null proportion π_0 ,
- $Z_{ij} \stackrel{iid}{\sim} N(0, 1)$,
- $\alpha_{ij} \stackrel{iid}{\sim} N(0, 1)$,
- $X_{j1} = 1$ for all $j = 1, \dots, n$,
- $X_{j2} = 1$ for $j = 1, \dots, n/2$ and $X_{j2} = 0$ for $j = n/2, \dots, n$,
- $\sigma_j^2 = 1$ for all $j = 1, \dots, p$,
- $\pi_0 \in \{0.5, 0.9, 1\}$,
- The alternative distribution of β_{2j} being either spiky, near-normal, flattop, skew, big-normal, or bimodal, where these are the same alternatives defined in Stephens (2016) and the following table. New alternatives are generated every iteration.

Scenario	Alternative Distribution
Spiky	$0.4N(0, 0.25^2) + 0.2N(0, 0.5^2) + 0.2N(0, 1^2), 0.2N(0, 2^2)$
Near Normal	$2/3N(0, 1^2) + 1/3N(0, 2^2)$
Flattop	$(1/7)N(-1.5, .5^2) + N(-1, .5^2) + N(-.5, .5^2) + N(0, .5^2) + N(0.5, .5^2) + N(1.0, .5^2) + N(1.5, .5^2)$
Skew	$(1/4)N(-2, 2^2) + (1/4)N(-1, 1.5^2) + (1/3)N(0, 1^2) + (1/6)N(1, 1^2)$
Big-normal	$N(0, 4^2)$
Bimodal	$0.5N(-2, 1^2) + 0.5N(2, 1^2)$

At each iteration, I generated new values of X , Z , α , β , E , and thus also Y .

Questions on simulation settings

- Right now, I am simulating Z independently of X . Should I be generating Z to have some pre-specified correlation with X ?
- I have $q = 5$ for all n . Should I increase q as n increases?
- I assume q is known. Should I estimate it every iteration, as I do in the GTEX simulations? Maybe SUCCOTASH and RUVASH are more robust to the choice of q ? Or maybe the true q isn't the best q to use when doing estimation?
- I didn't choose the variances of β_{1j} , Z_{ij} , and E_{ij} carefully. Should I vary them so that proportion of variance explained by β_{2j} is different?
- Should I even include β_{1j} in the simulations?

Methods

The confounder adjustment methods I look at in this write-up are:

- OLS + qvalue.
- OLS + ASH
- SUCCOTASH using normal mixtures and heteroscedastic PCA as the factor-analysis method. This is the two-step version that does variance inflation.
- The robust regression version of CATE using PCA as the factor analysis method + qvalue.
- SVA + qvalue.
- RUVASH with normal likelihood.
- RUV4 (GLS version) using the variance inflation.
- Negative control version of CATE using PCA as the factor analysis method + qvalue.
- RUV2 + qvalue.
- RUV4 + qvalue.
- RUV4 + ASH (without variance inflation).

Results

Estimates of π_0

- There isn't a clear winner like there was during the GTEX Poisson-thinning simulations. RUVASH appears to do the best when $n = 40$.
- RUV4 + qvalue doesn't do too bad.
- SUCCOTASH no longer works as well as it did during the GTEX simulations. Under some alternatives it works OK, but mostly only for $n = 40$.

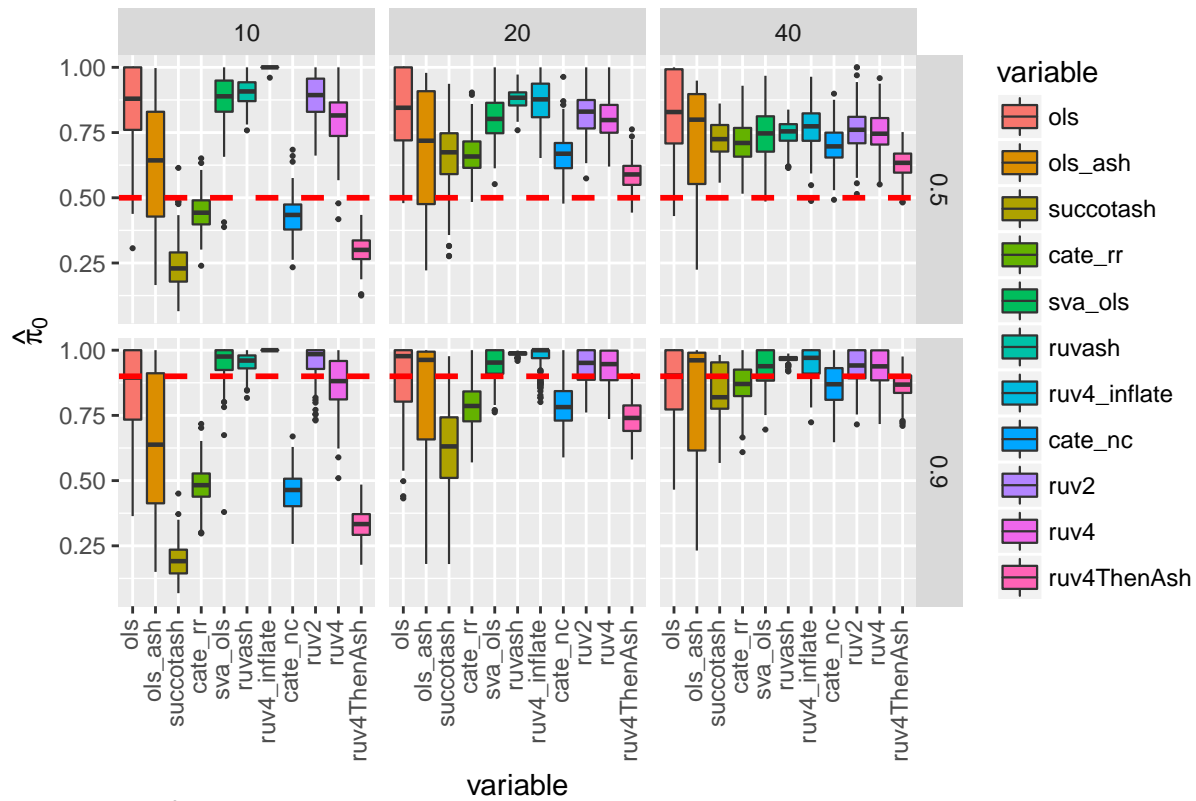
AUC performance.

- SUCCOTASH, CATE_rr, CATE_nc, RUV4, and RUV4thenASH seem to be the winners in terms of AUC.
- RUVASH is among the winners when $n = 20$ or 40 , but not when $n = 5$.
- RUV4_inflate has very poor AUC when $n = 10$. Since RUV4_inflate uses GLS instead of OLS, like RUV4, this must be what's causing this poor performance. Otherwise, the ordering of the p-values is unchanged by inflation. I think that using limma-shrunk variances would correct for this poor performance of AUC. If I used OLS, then theoretically I should get the *exact* same results as RUV4.
- Limmashrunk variances would probably also help RUVASH for small n .

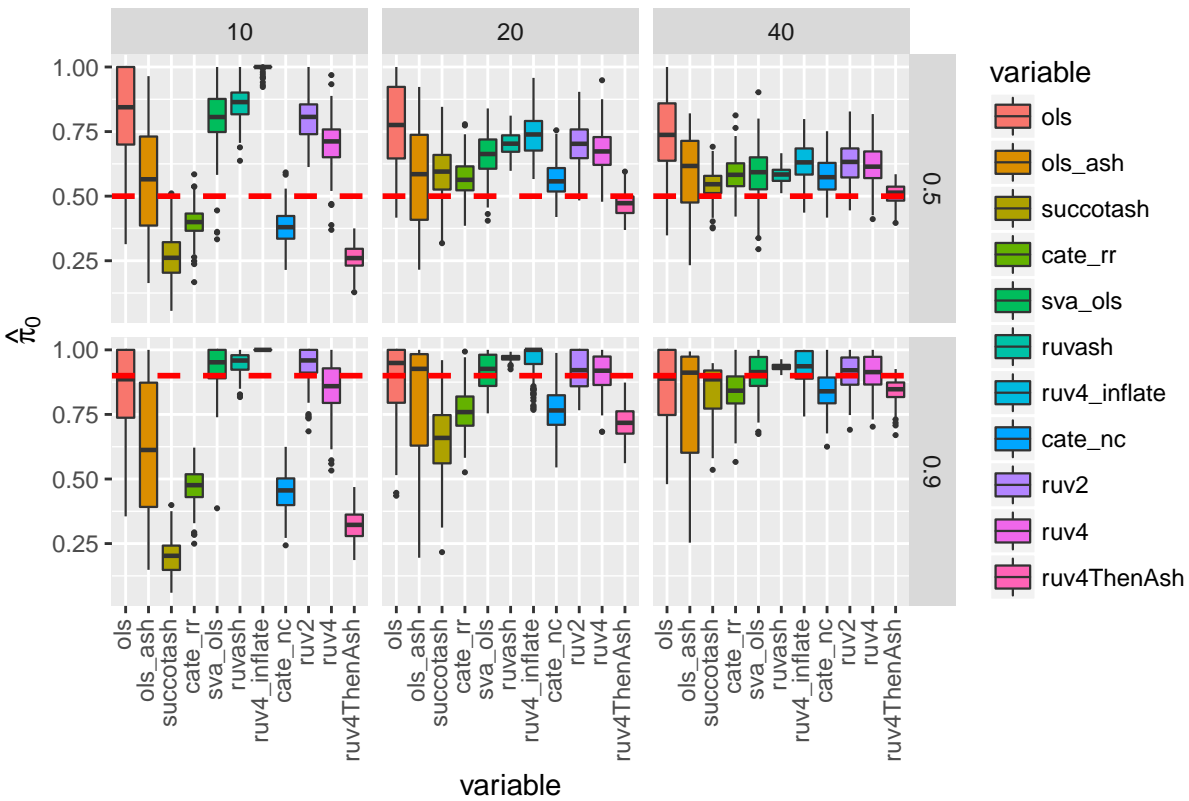
MSE

- SUCCOTASH, RUVASH, and RUV4 then ASH are the winners in terms of MSE.

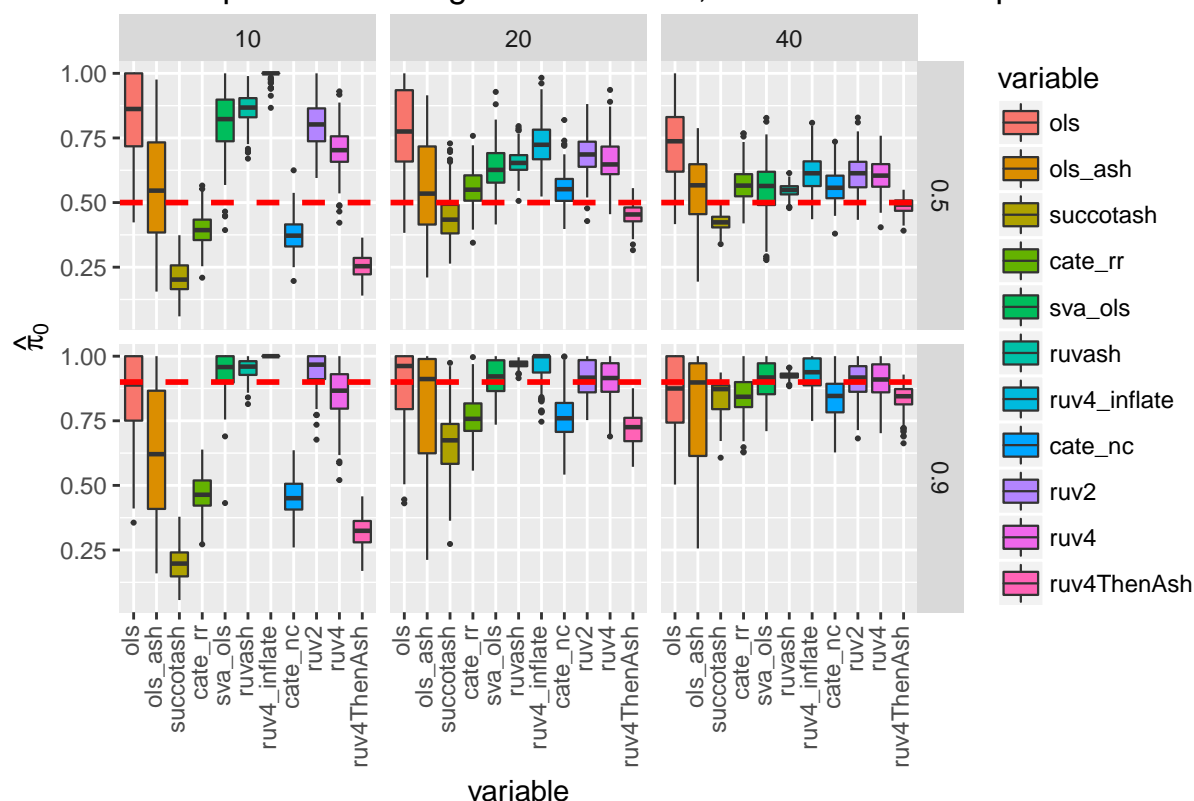
Estimates of π_0 When Using Muscle Tissue, Alternative = spiky



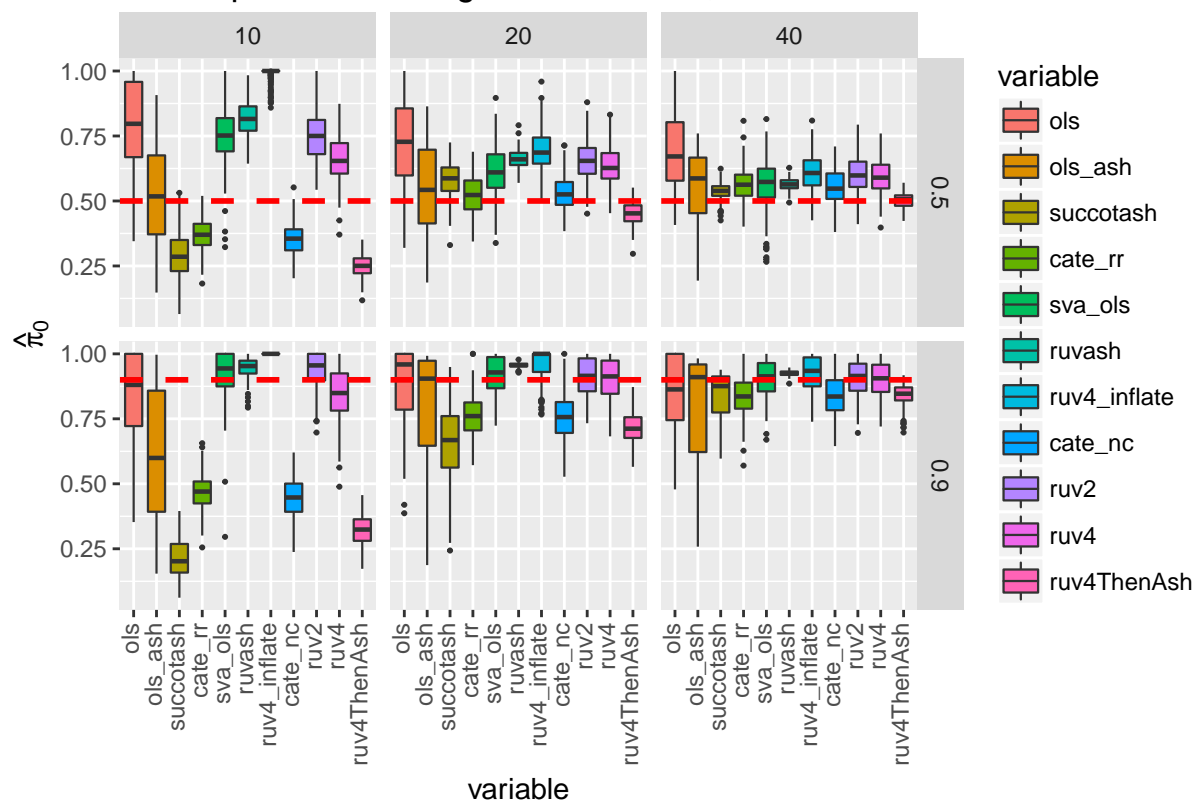
Estimates of π_0 When Using Muscle Tissue, Alternative = near_normal



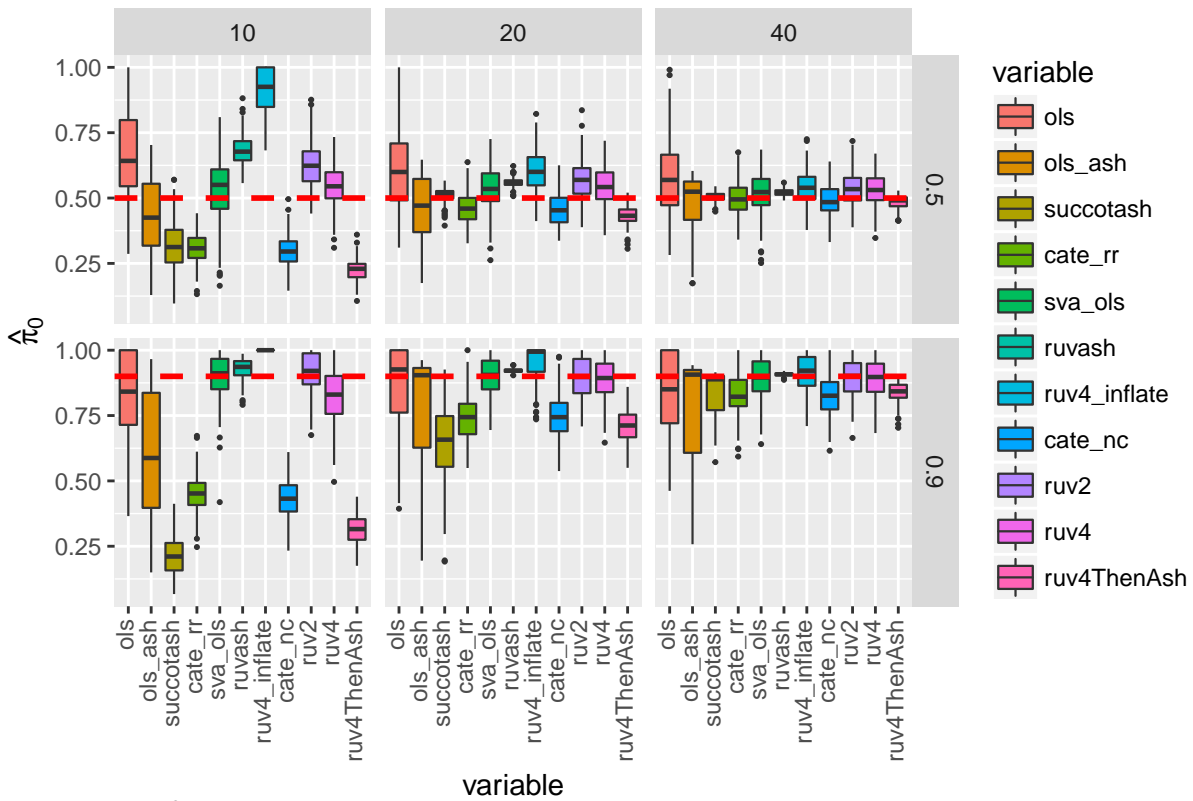
Estimates of π_0 When Using Muscle Tissue, Alternative = flattop



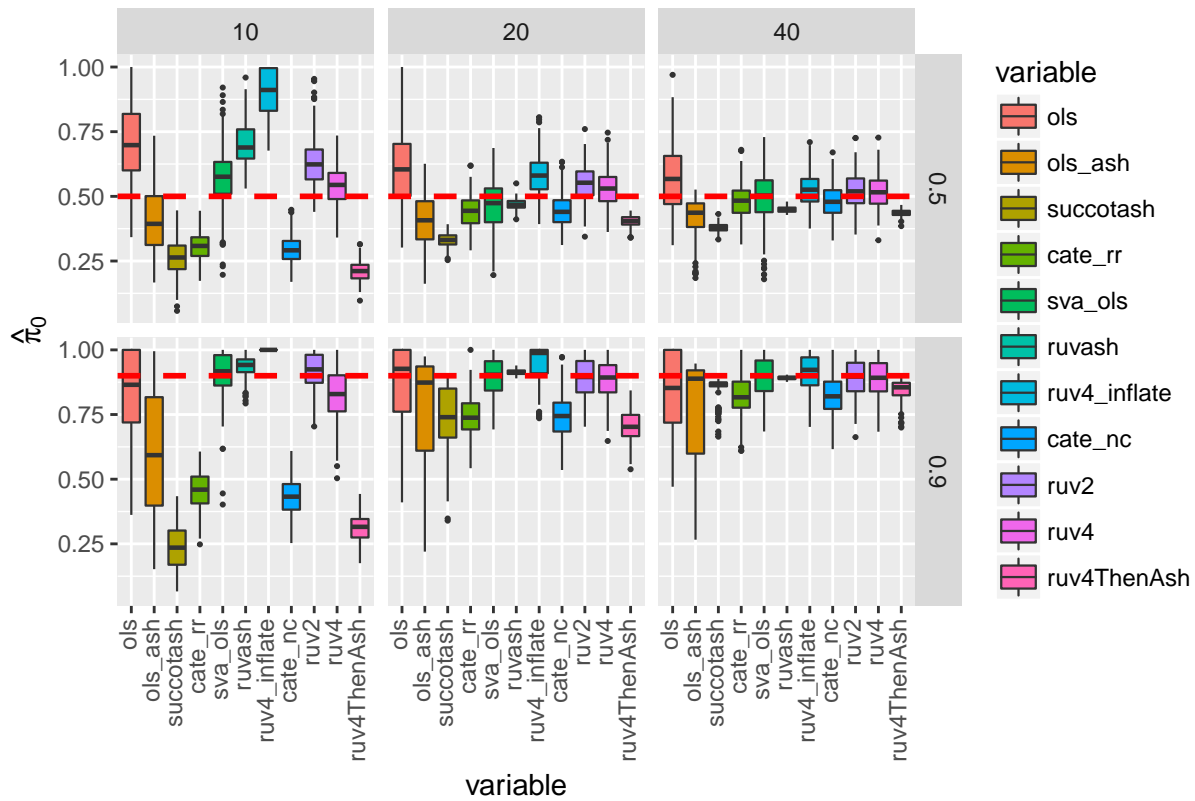
Estimates of π_0 When Using Muscle Tissue, Alternative = skew



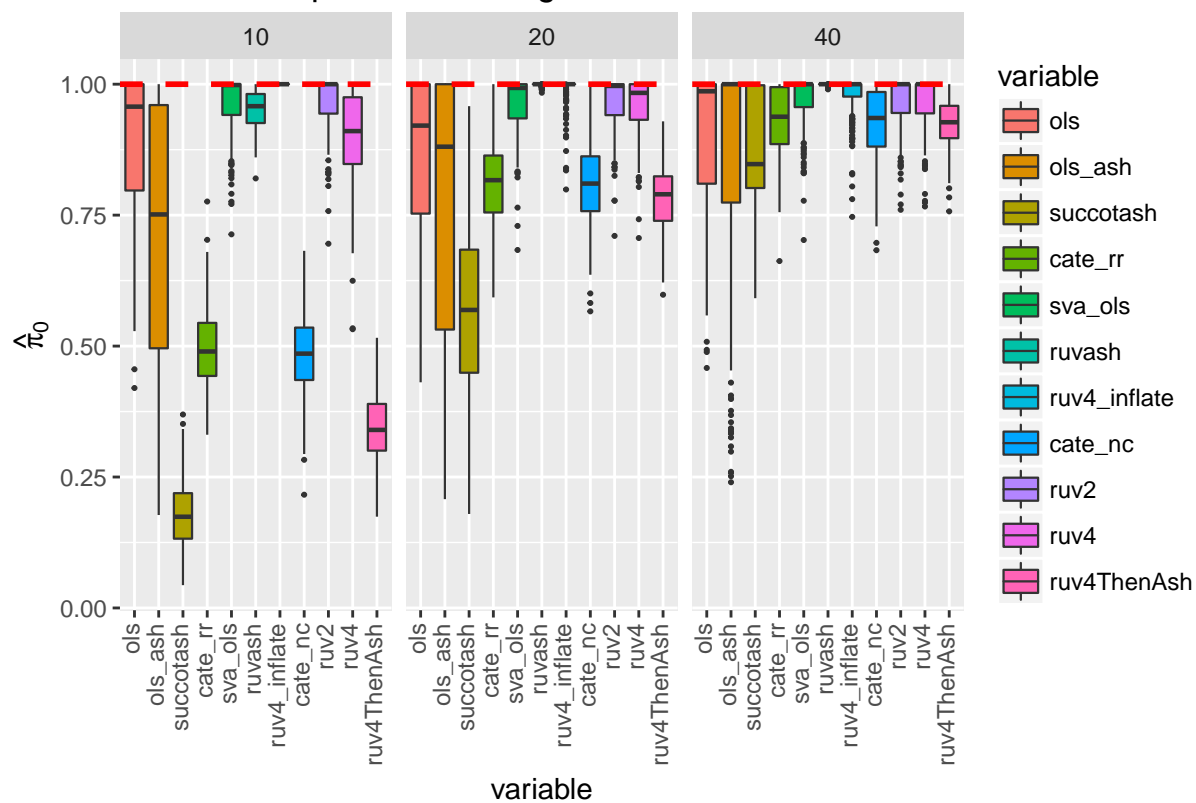
Estimates of π_0 When Using Muscle Tissue, Alternative = big_normal



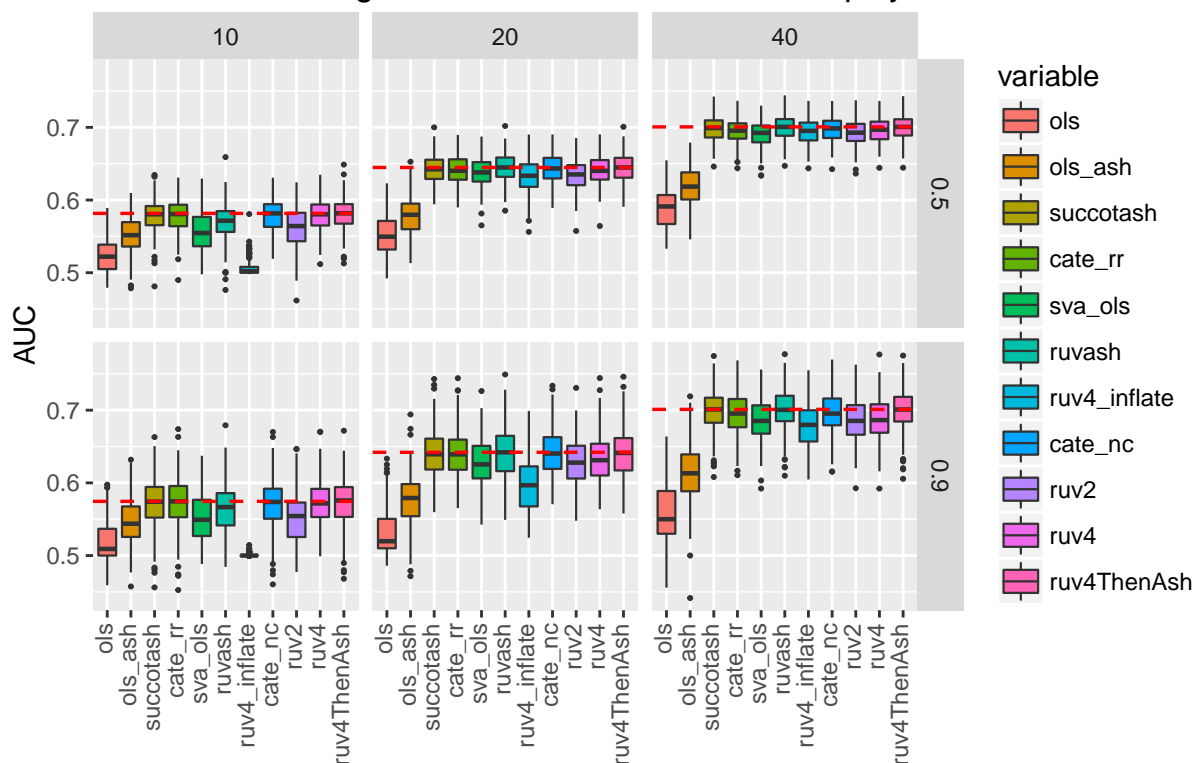
Estimates of π_0 When Using Muscle Tissue, Alternative = bimodal



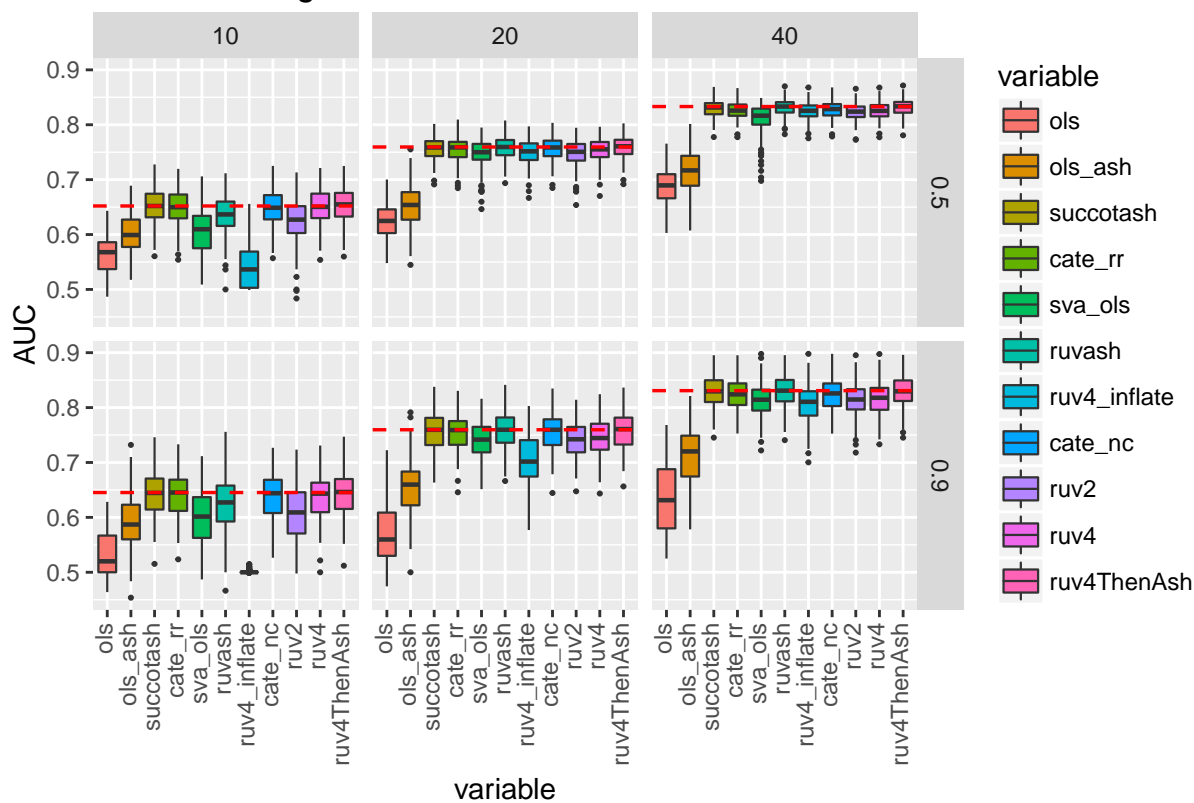
Estimates of π_0 When Using Muscle Tissue and All Null



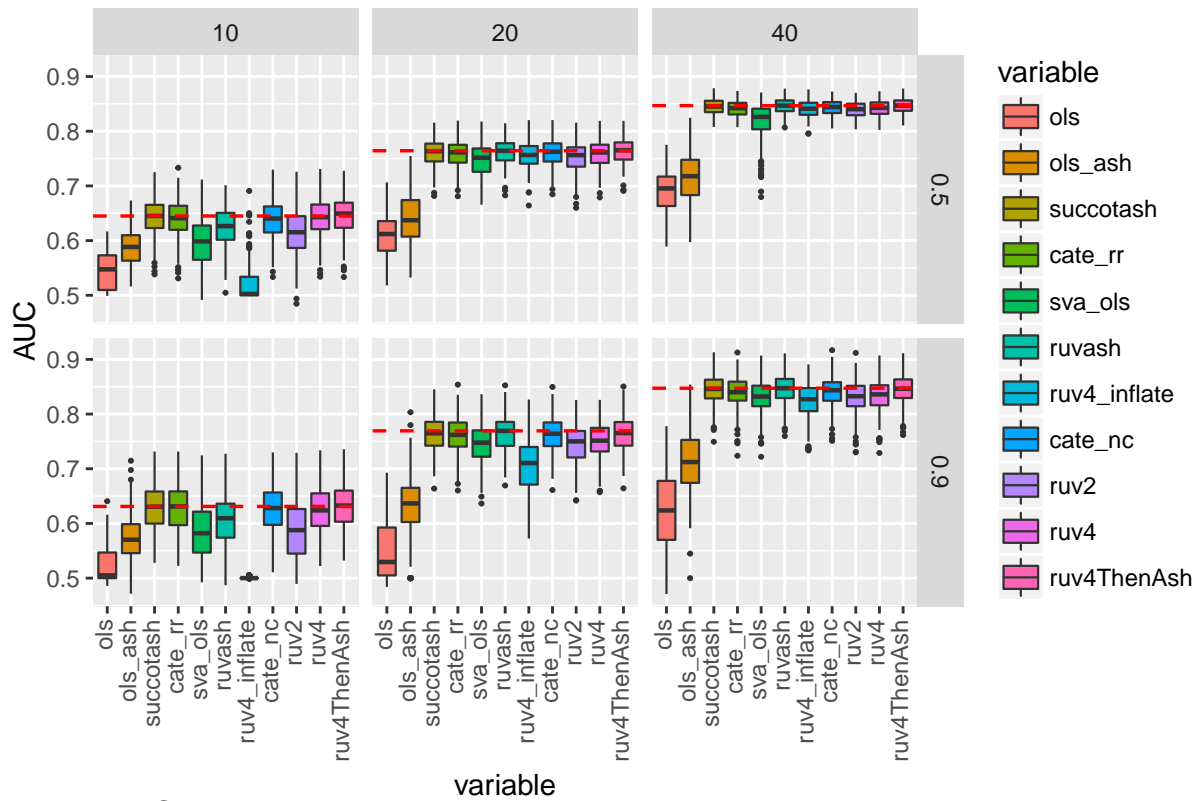
AUC When Using Muscle Tissue, Alternative = spiky



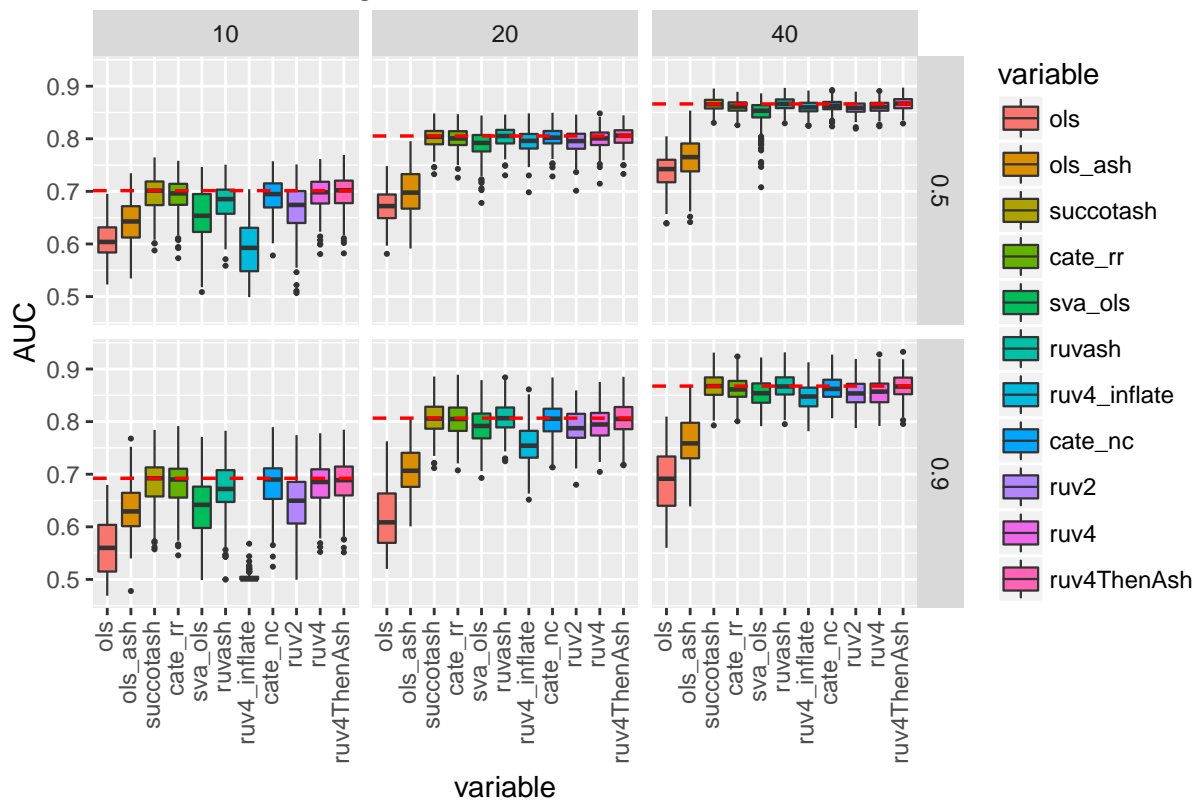
AUC When Using Muscle Tissue, Alternative = near_normal



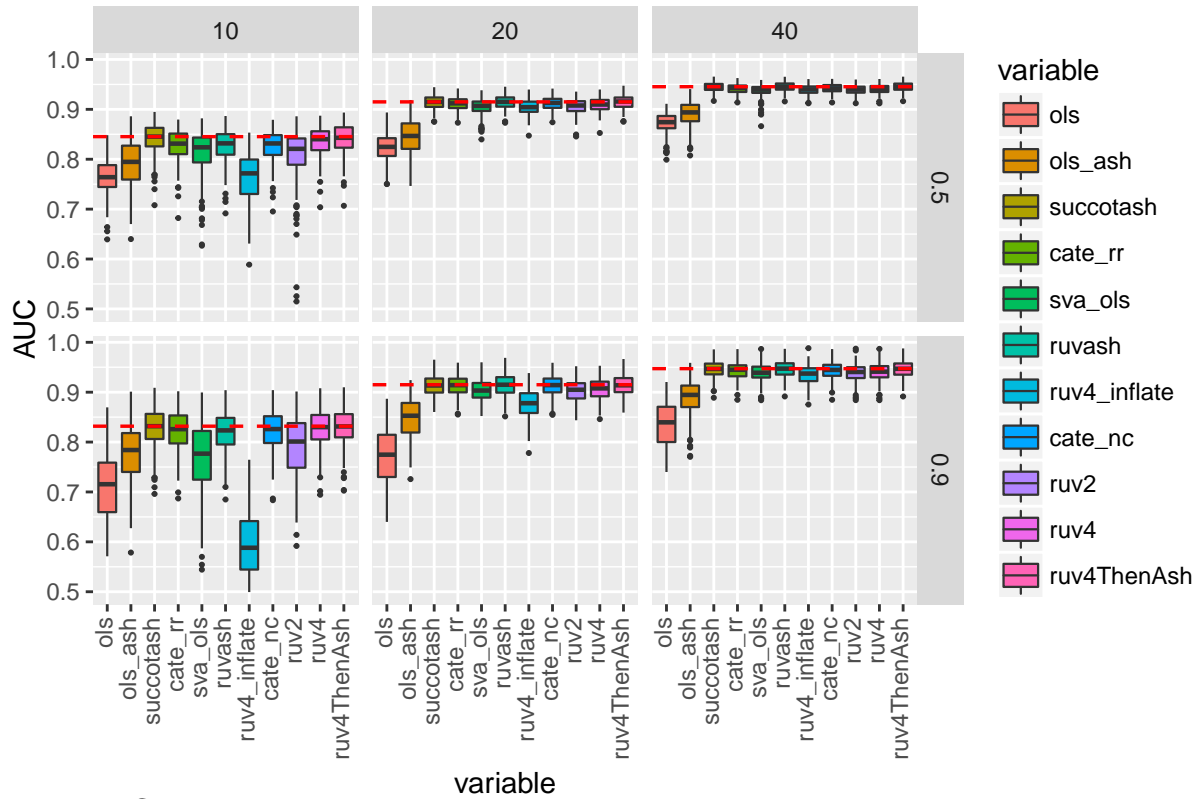
AUC When Using Muscle Tissue, Alternative = flattop



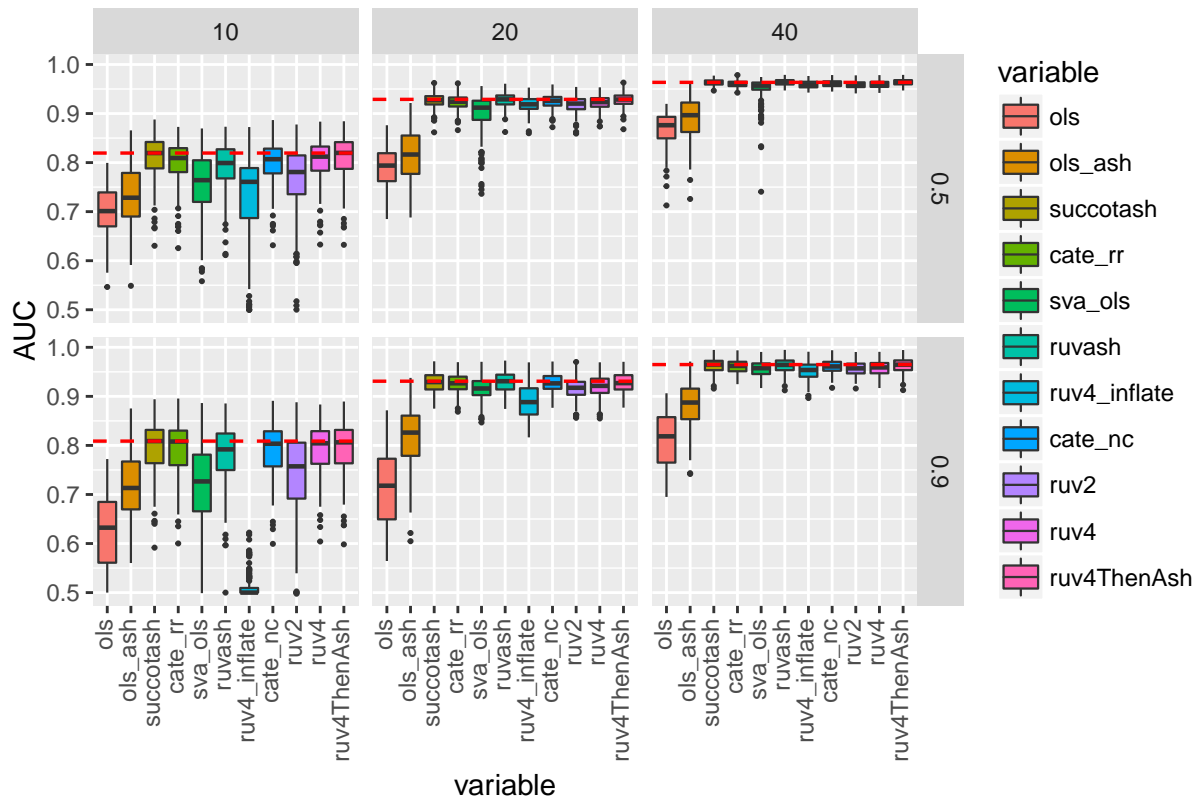
AUC When Using Muscle Tissue, Alternative = skew



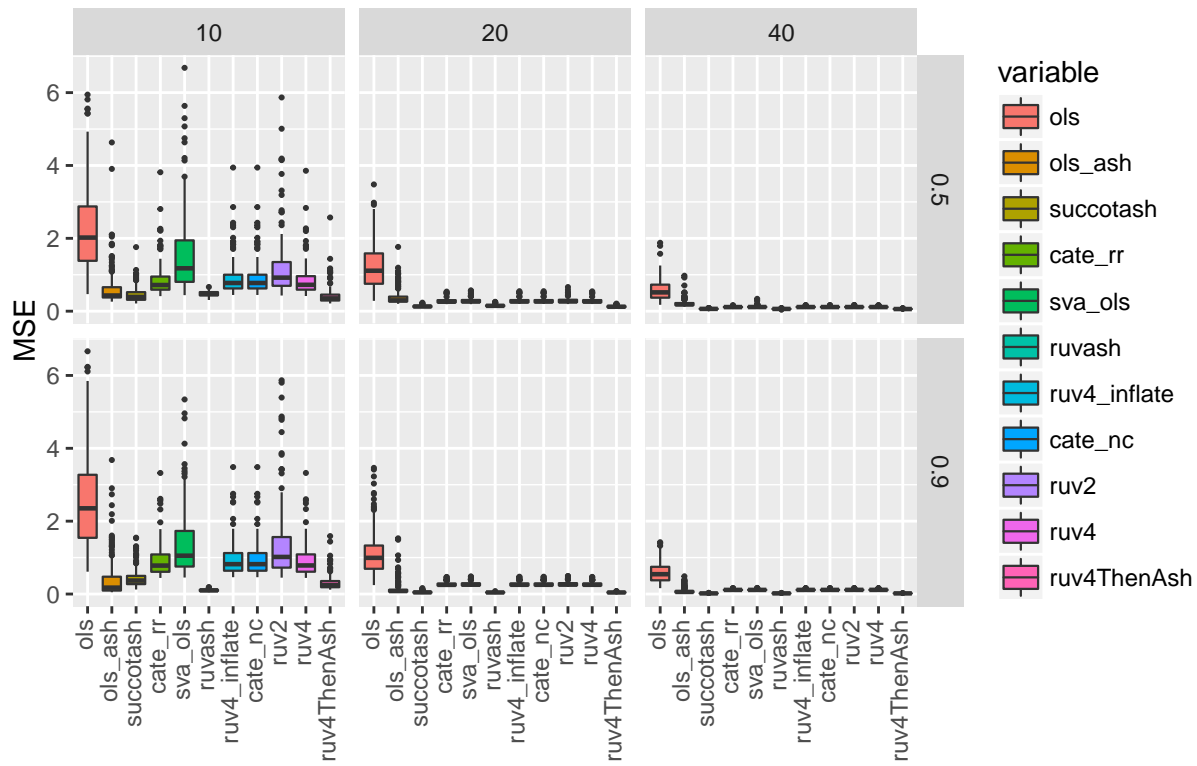
AUC When Using Muscle Tissue, Alternative = big_normal



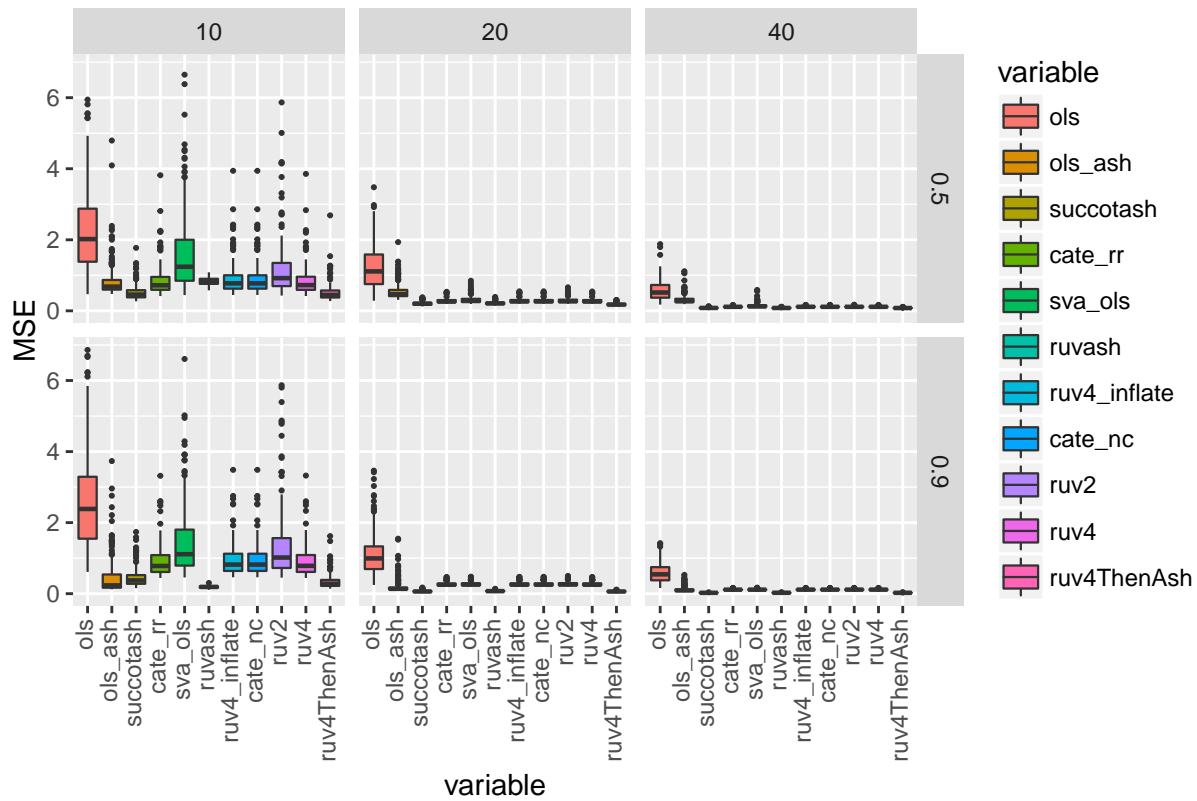
AUC When Using Muscle Tissue, Alternative = bimodal



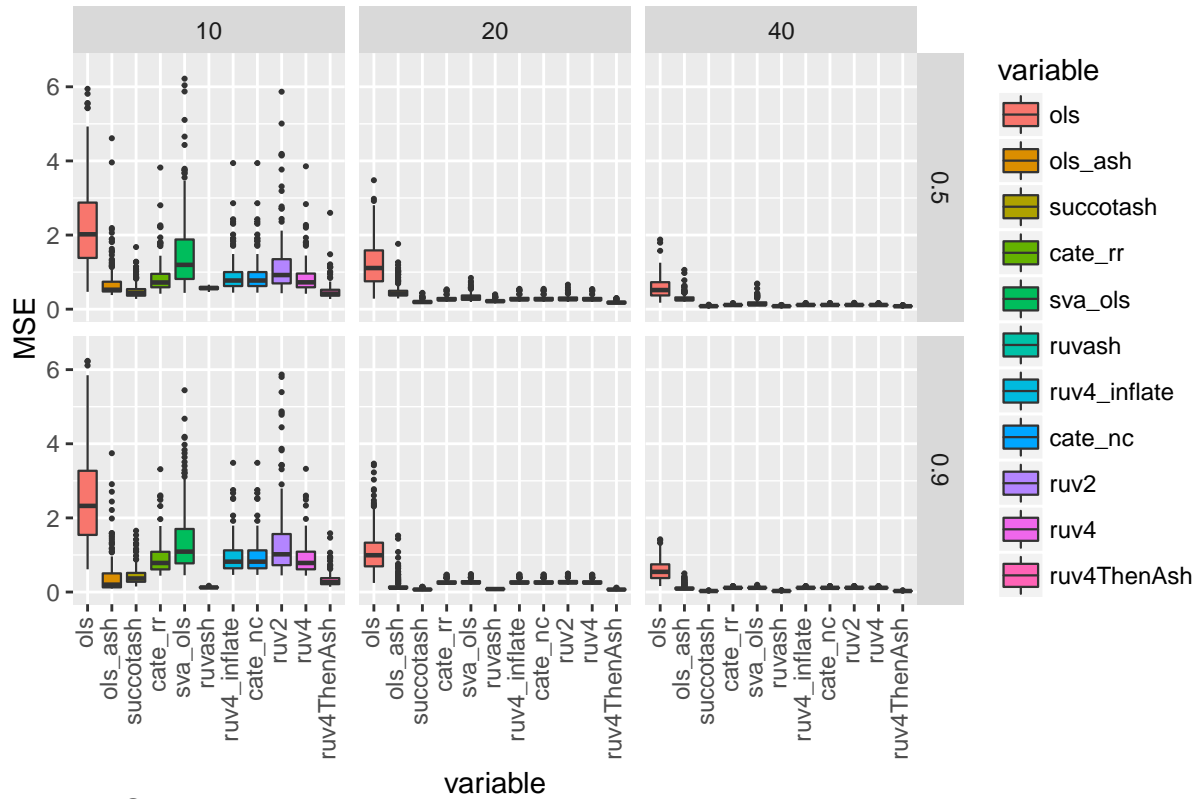
MSE When Using Muscle Tissue, Alternative = spiky



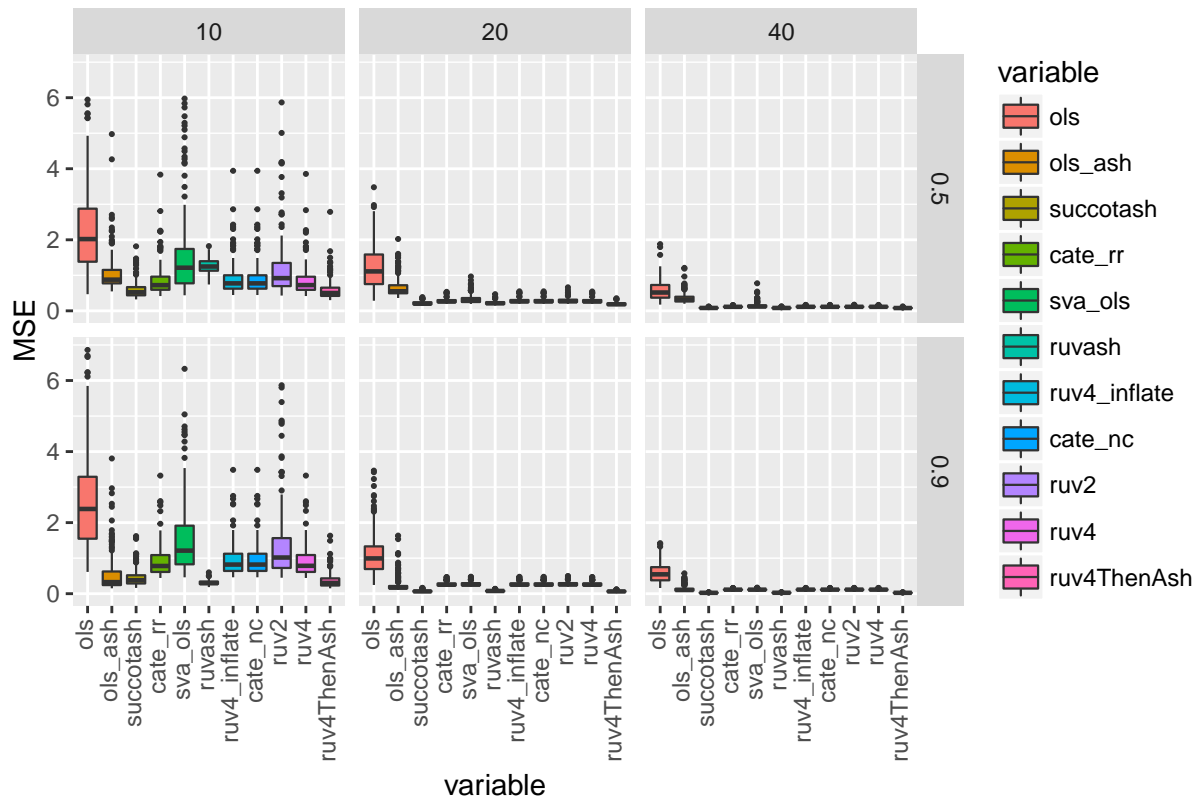
MSE When Using Muscle Tissue, Alternative = near_normal



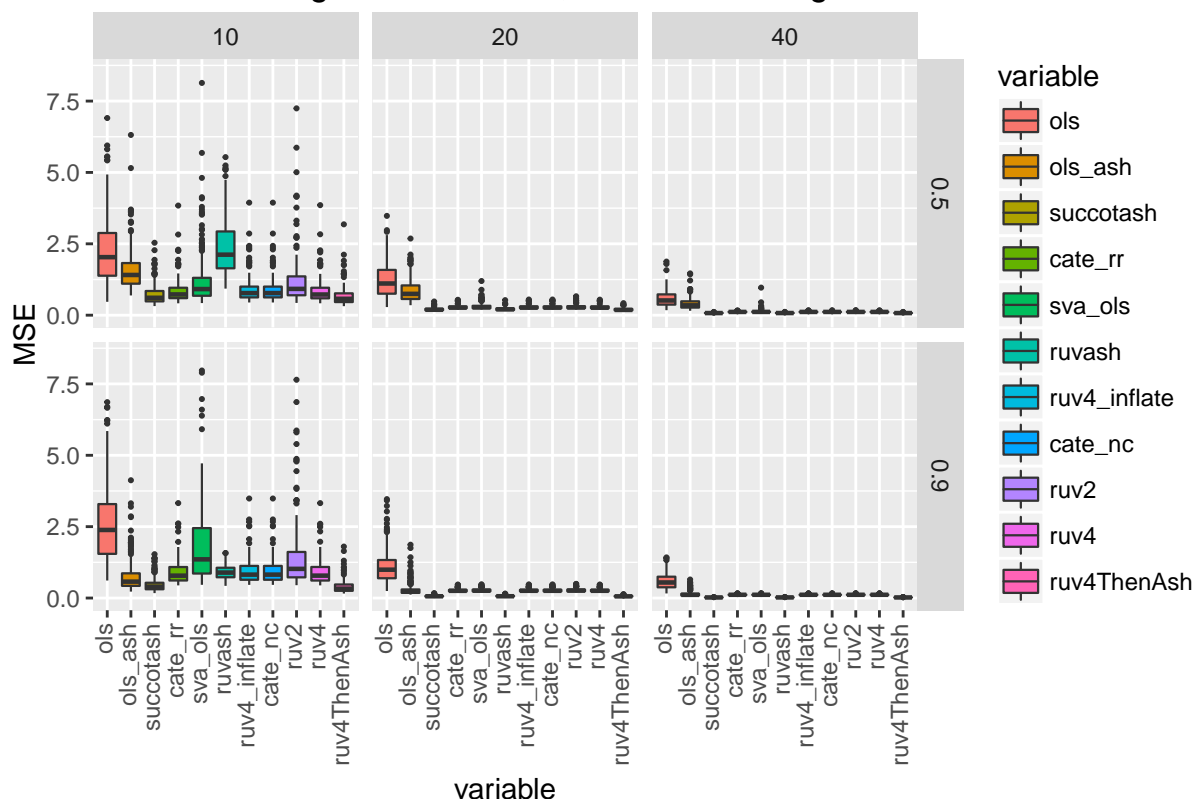
MSE When Using Muscle Tissue, Alternative = flattop



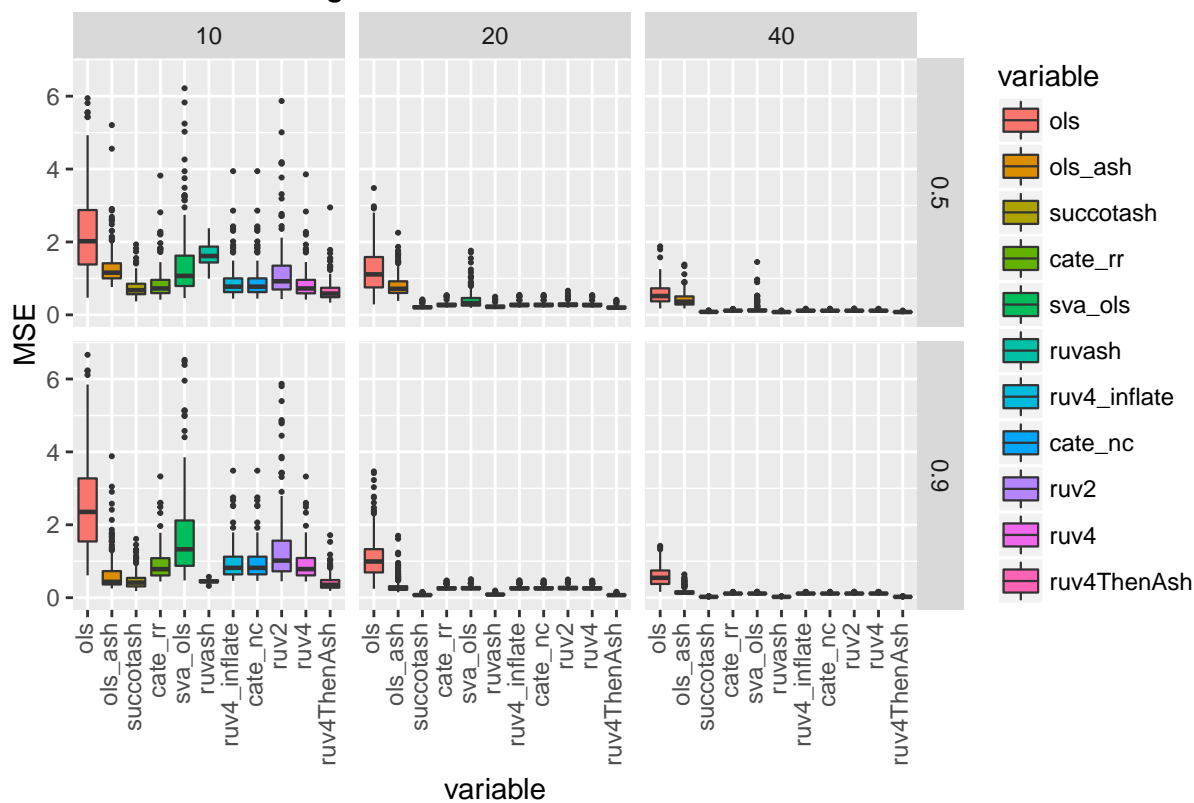
MSE When Using Muscle Tissue, Alternative = skew



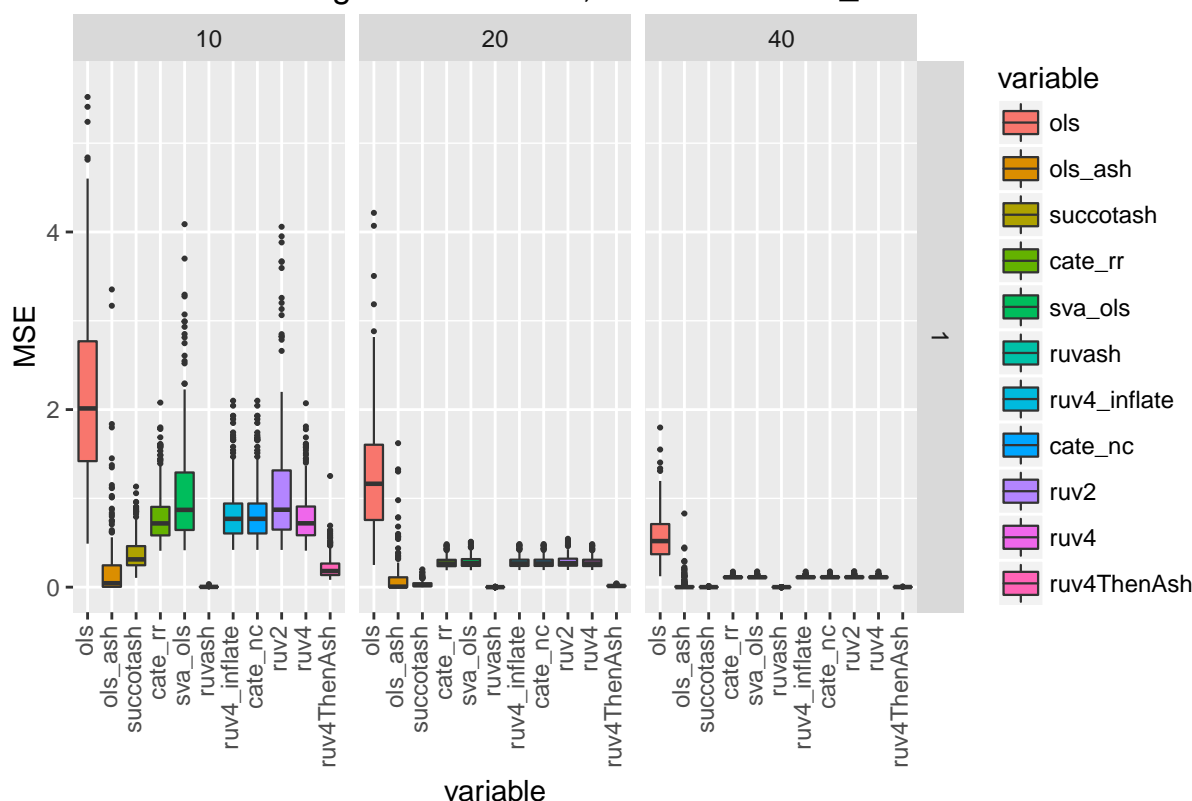
MSE When Using Muscle Tissue, Alternative = big_normal



MSE When Using Muscle Tissue, Alternative = bimodal



MSE When Using Muscle Tissue, Alternative = all_null



`sessionInfo()`

```
## R version 3.3.0 (2016-05-03)
## 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] pROC_1.8      dplyr_0.4.3  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_1.8.3
##  [9] tools_3.3.0      parallel_3.3.0 grid_3.3.0      gtable_0.2.0
## [13] DBI_0.4          htmltools_0.3.5 yaml_2.1.13     lazyeval_0.1.10
## [17] assertthat_0.1   digest_0.6.9   formatR_1.3     codetools_0.2-14
## [21] evaluate_0.9     rmarkdown_0.9.6 labeling_0.3     stringi_1.0-1
```

[25] compiler_3.3.0 scales_0.4.0

Stephens, Matthew. 2016. “False Discovery Rates: A New Deal.” *BioRxiv*. Cold Spring Harbor Labs Journals, 038216.