Different applications of CATE + ASH

David Gerard

February 17, 2016

Abstract

I look at the performance of different options of CATE. In terms of estimating π_0, if control genes are not known, it looks like the robust regression version of CATE using PCA to estimate the latent variables and then using ASH to estimate π_0 works the best when the sample size is at least 20. In terms of AUC, any of the CATE + ASH methods will work.

1 Methods

For CATE, I varied three parameters.

1. The factor analysis method: Either quasi-maximimum likelihood ("ml"), PCA ("pc"), or an early stopping method I haven't read about but is an option ("esa").
2. Whether the p-values are calibrated using maximum absolute deviation (TRUE) or not (FALSE). This only matters for the qvalue methods and shouldn't affect the ASH methods.
3. Whether we used the robust-regression version of CATE ("rr") or the negative controls verison of CATE ("nc") using half of the null genes as the negative controls.

For each setting in CATE, I performed two methods. The first method consisted of a two-step procedure:

1. Estimate $\hat{\beta}_{[2,i]}$ and it's corresponding standard error $\hat{s}_i$.
2. Run ASH on $\hat{\beta}_{[2,i]}$ and $\hat{s}_i$.

The second method was to use the p-values output by CATE.

I always ran CATE on $\log(COUNTS + 1)$.

The ASH methods provide an estimate of π_0. I obtained an estimate of π_0 from the p-values by the `qvalue` package in `R` [Storey, 2002].

The number of hidden confounders was estimated using the methods of Buja and Eyuboglu [1992] implemented in the `num.sv()` function in the `sva` package in `R`. CATE doesn't work sometimes when there is only one confounder, so I set the minimum number to 2 confounders.

2 Simulation Study

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

- $n \in \{10, 20, 40\}$,
- $p = 1000$,
- $\pi_0 \in \{0.5, 0.9\}$,

- $\sigma_{log2} \in \{1, 5\}$.

I extracted the most expressed p genes (excluding the top 5 expressed genes) from the GTEX lung data and n samples are chosen at random. Half of these samples are randomly given the "treatment" label 1, the other half given the "treatment" label 0. Of the p genes, $\pi_0 p$ were chosen to be non-null. Signal was added by the Poisson-thinning approach in Mengyin's code with a mean log2-fold change of 0 and a standard deviation log2-fold change of σ_{log2}. That is

$$A_1, \ldots, A_{p/2} \sim N(0, \sigma^2_{log2}) \tag{1}$$

$$B_i = 2^{A_i} \text{ for } i = 1, \ldots, p/2. \tag{2}$$

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

For each iteration, I calculated two things:

1. The AUC using either the lfdrs or p-values.
2. The estimates of π_0.

3 Results

For the frequentist procedures, I used the vector of p-values as the predictions and I used the vector of lfdr's from the ASH-like procedures for prediction. These were used to create ROC curves and calculate AUCs. In general, ASH procedures performed better than just using the p-values and using negative controls worked better than the robust regression version. This seems more pronounced when the sample size is larger and when π_0 is smaller.

From the p-values, I used the `qvalue` package [Storey, 2002] to estimate π_0. Estimates of π_0 are given from `ashr` for the ASH-like methods.

For a sample size of 10 (5 vs 5), none of the methods perform well at estimating π_0. But when the sample size is 20 or 40, negative control version of CATE + ASH performs remarkably well. Interestingly, the robust regression version of CATE + ASH works almost as well when we use PCA to estimate the hidden confounders instead of the "ML" or "ESA" options in CATE.

Inserting the calibrated p-values of CATE into `qvalue` always resulted in very high estimates of π_0. An example histogram of these p-values is in Figure 1. I think the mode near 1 is causing the very high estimates of π_0. The smother method fits a cubic spline to $(\lambda, \hat{\pi}_0)$. Call this function (of lambda) $sp(\cdot)$. The final estimate of π_0 is $sp(1)$. The bump makes the $(\lambda, \hat{\pi}_0)$ look really weird (Figure 2).

The bootstrap method to estimate the tuning parameter used to estimate π_0 yielded an estimate of π_0 of 0.88 in this case. The "smoother" method yielded a estimate of 0.99. All of the q-value methods used in the simulation study use the "smoother" because that's the default option.

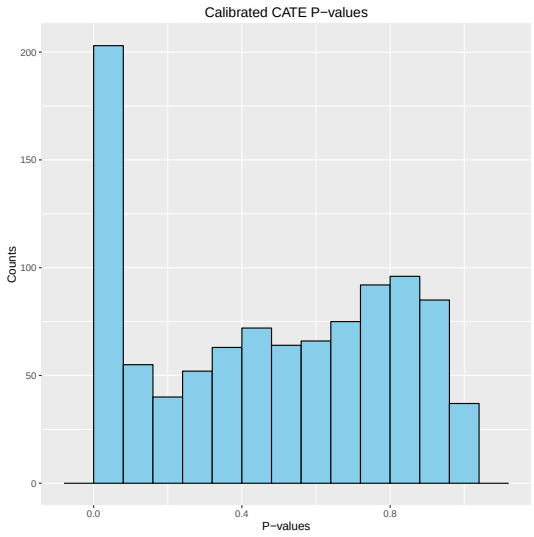


Figure 1: Calibrated p-values from CATE rr with ml option when $n = 5$, $\pi_0 = 0.5$, and $\sigma^2_{log2} = 1$.

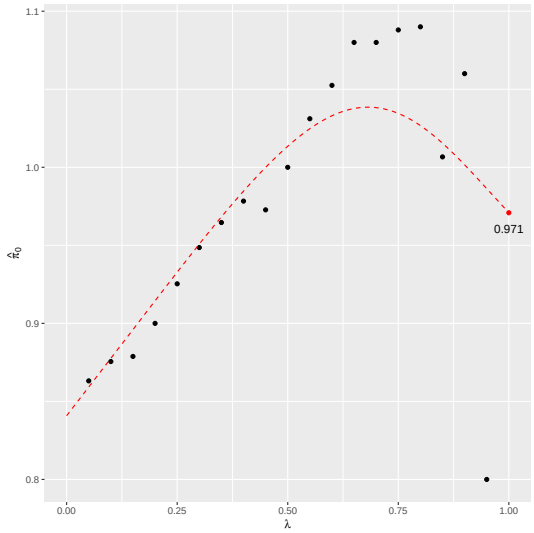
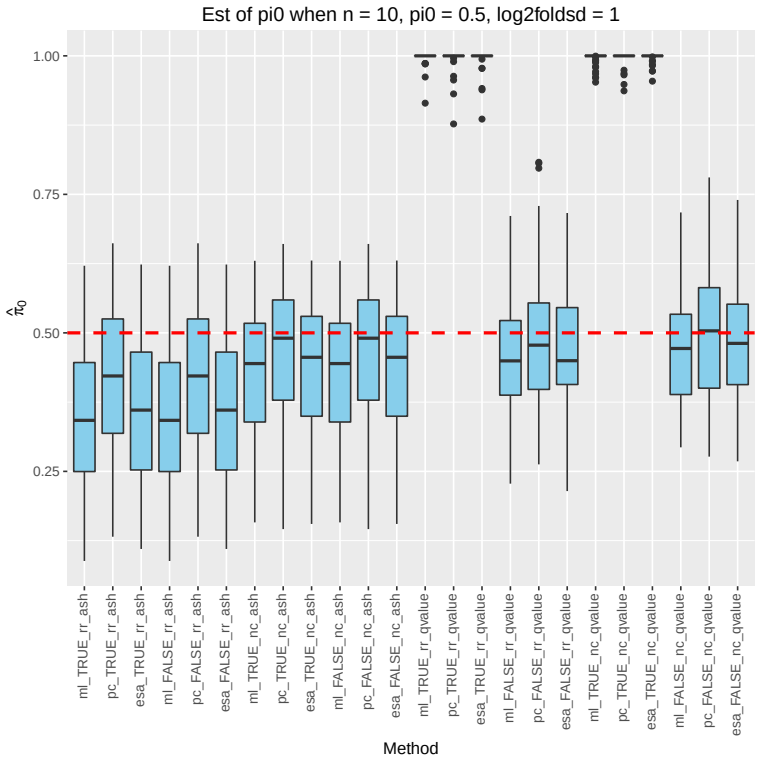
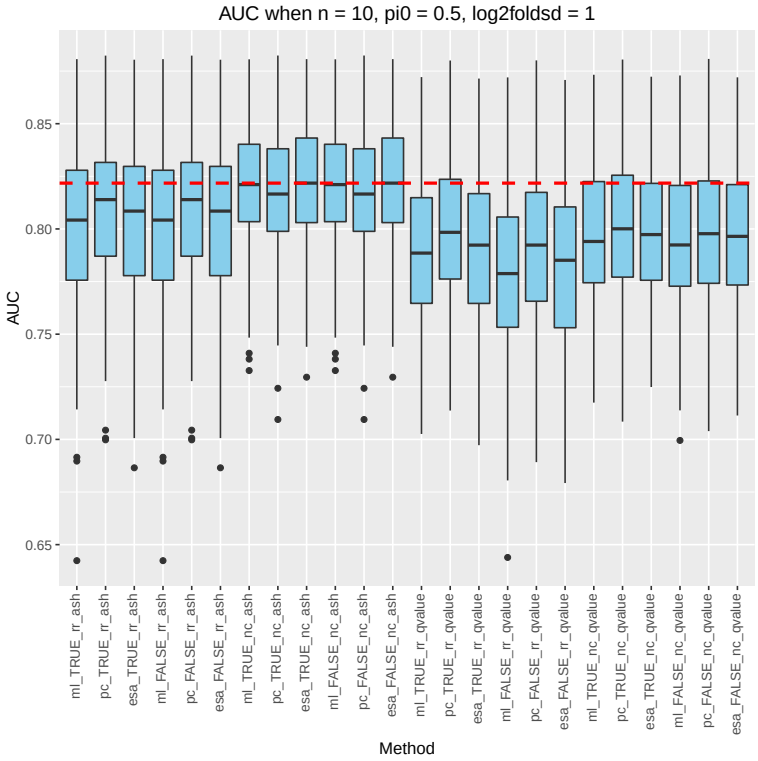
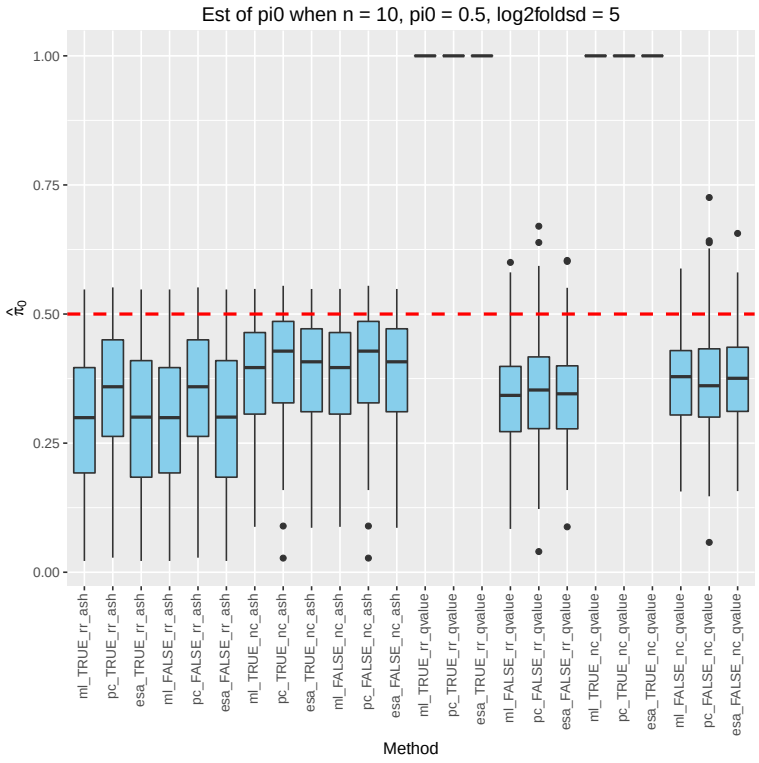
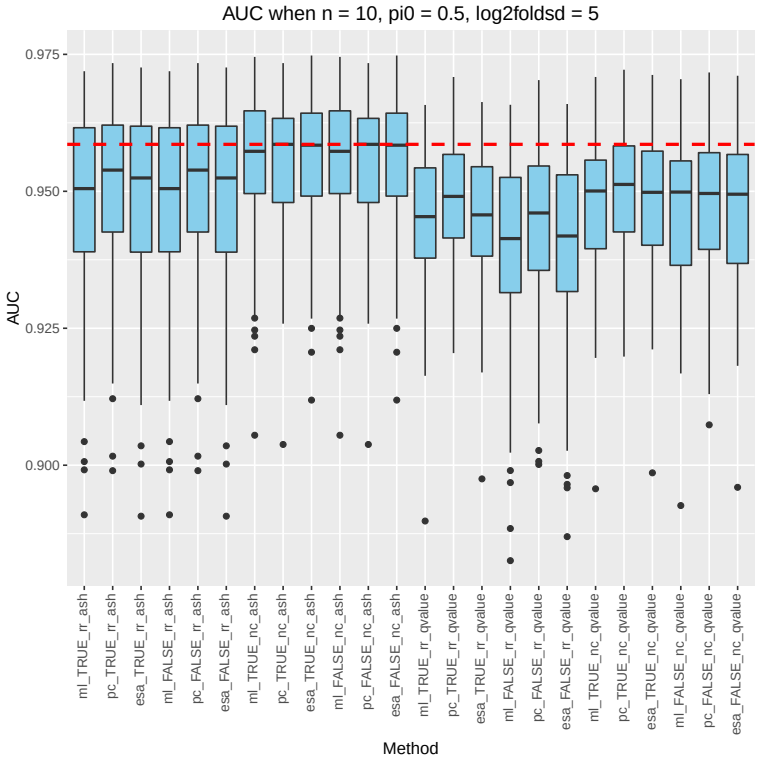


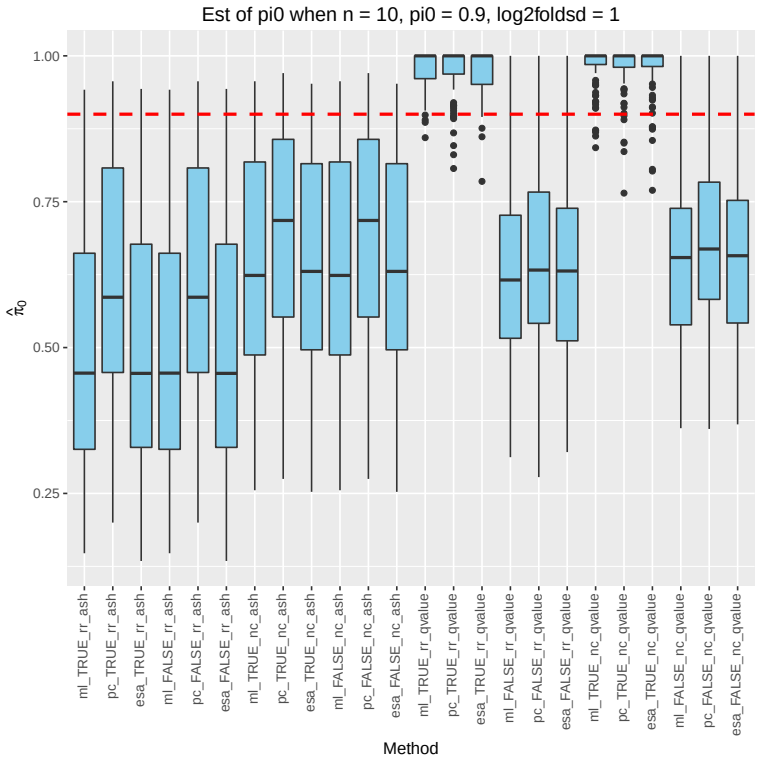
Figure 2: λ vs $\hat{\pi}_0$ with a cubic smoothing spline fit with 3 degrees of freedom. The spline value at $\lambda = 1$ is the red point.

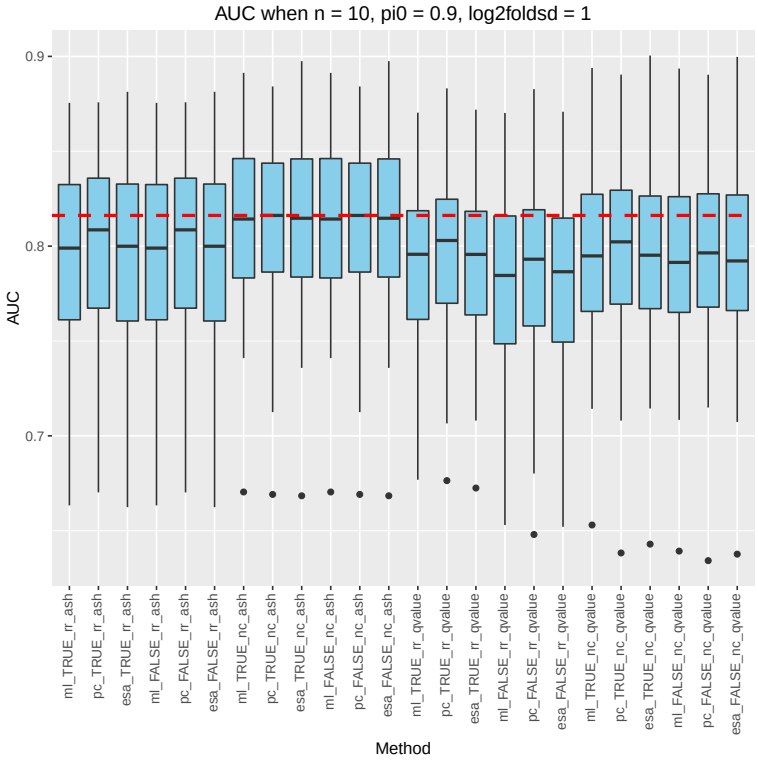

Est of pi0 when n = 10, pi0 = 0.5, log2foldsd = 1
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

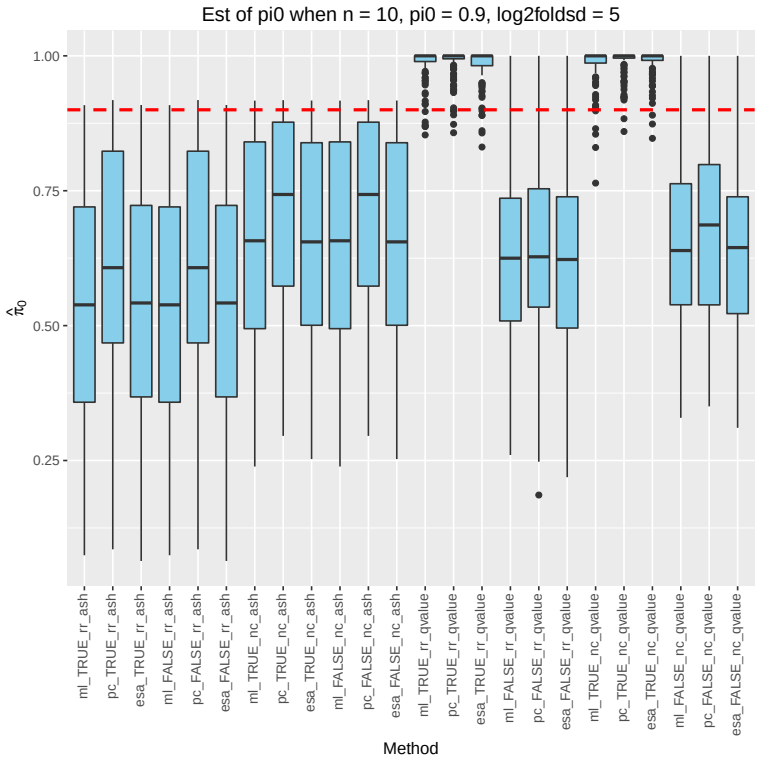

AUC when n = 10, pi0 = 0.5, log2foldsd = 1
0.85
0.80
0.75
0.70
0.65
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

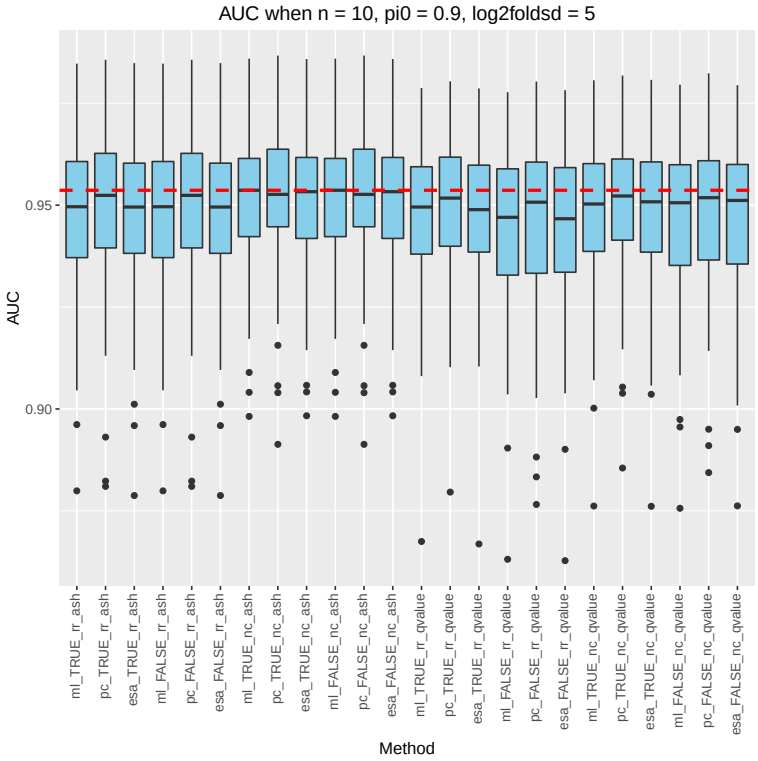

Est of pi0 when n = 10, pi0 = 0.5, log2foldsd = 5
1.00
0.75
0.50
0.25
0.00
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

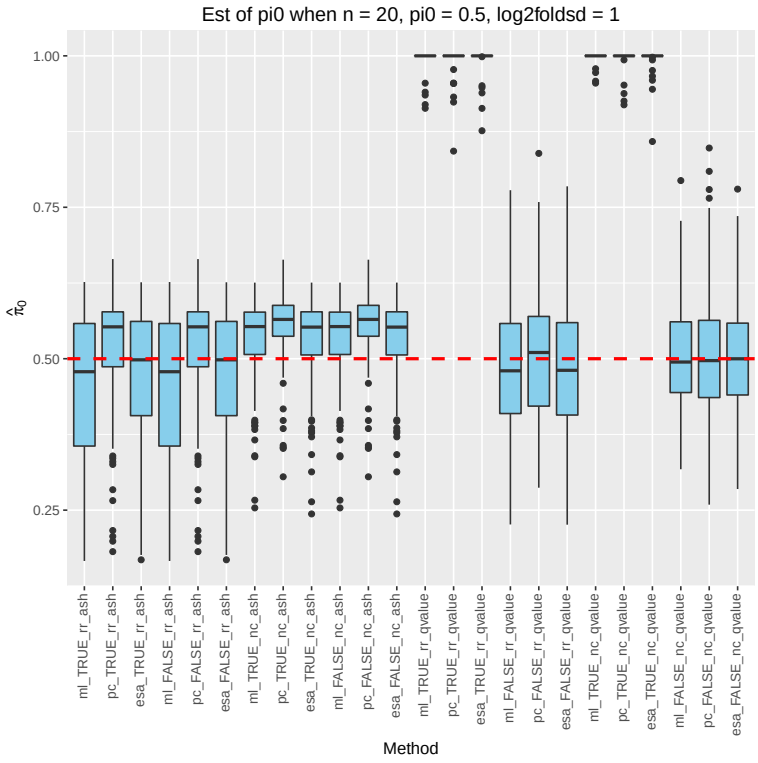

AUC when n = 10, pi0 = 0.5, log2foldsd = 5
0.975
0.950
0.925
0.900
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

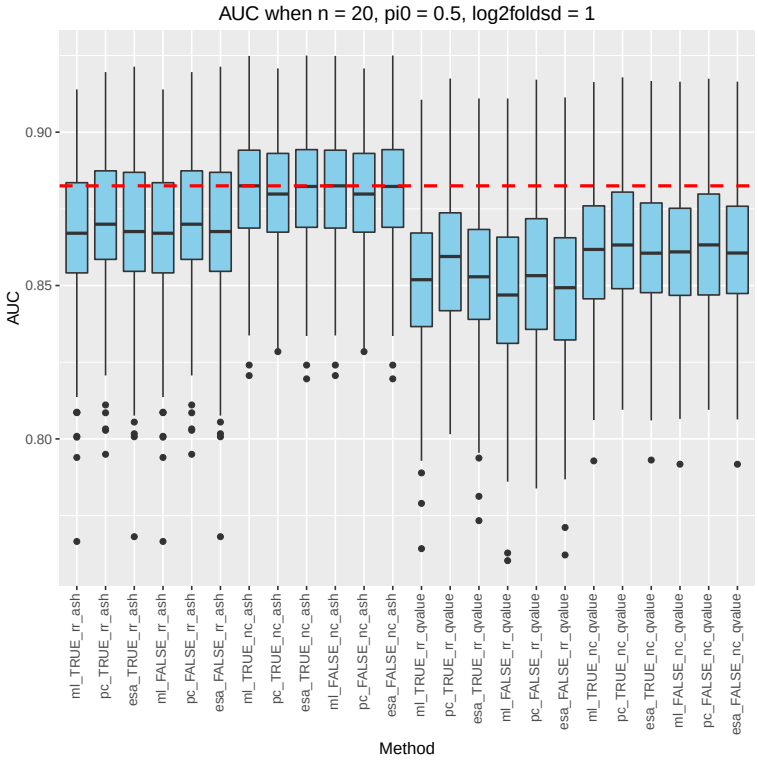

Est of pi0 when n = 10, pi0 = 0.9, log2foldsd = 1
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

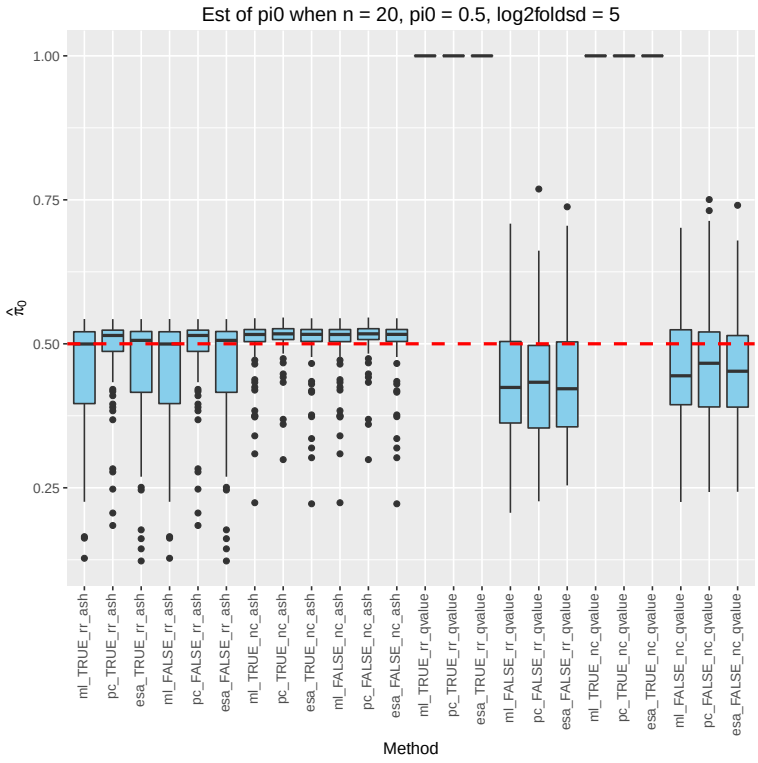

AUC when n = 10, pi0 = 0.9, log2foldsd = 1
0.9
0.8
0.7
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

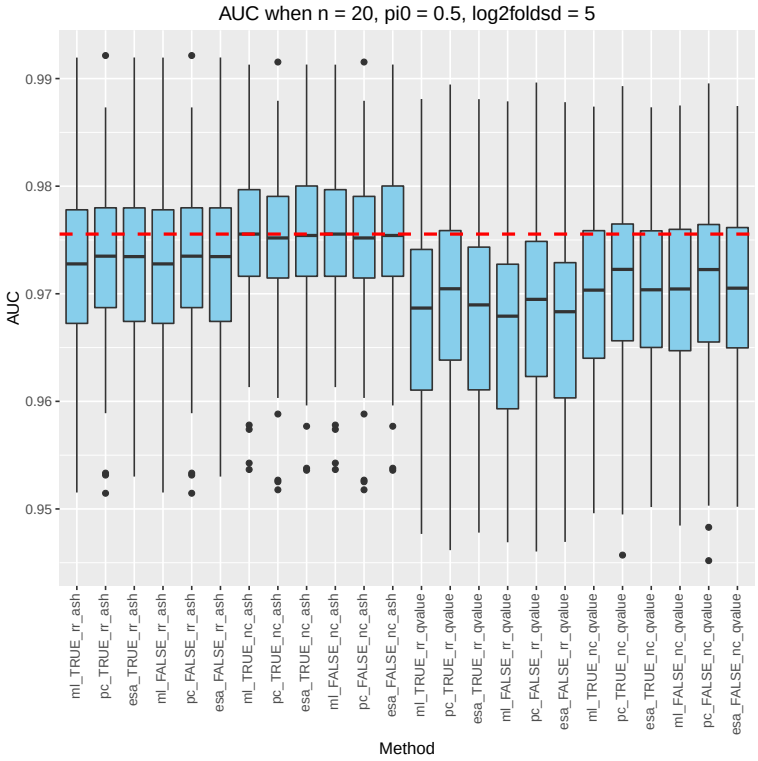

Est of pi0 when n = 10, pi0 = 0.9, log2foldsd = 5
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

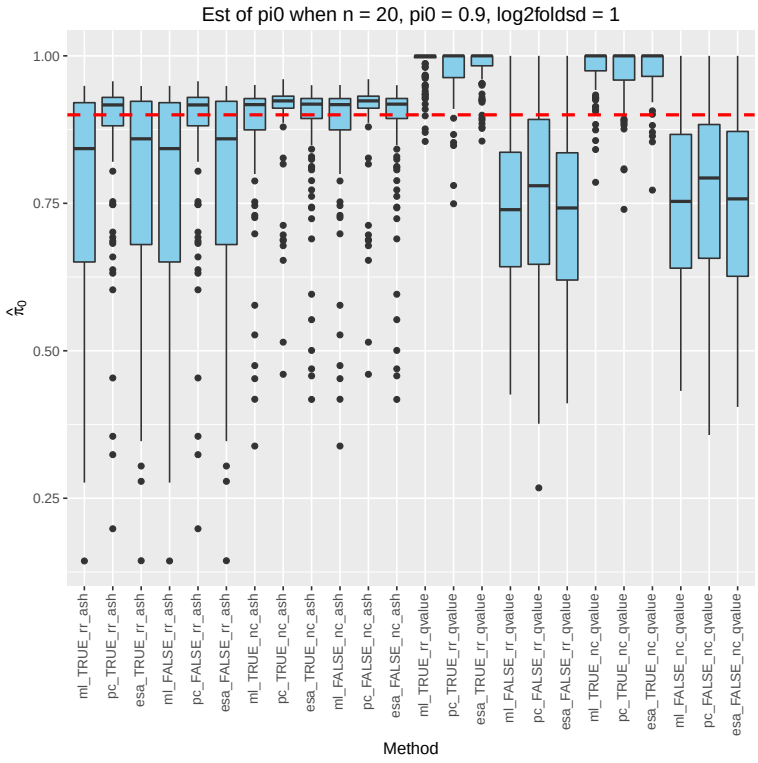

AUC when n = 10, pi0 = 0.9, log2foldsd = 5
0.95
0.90
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

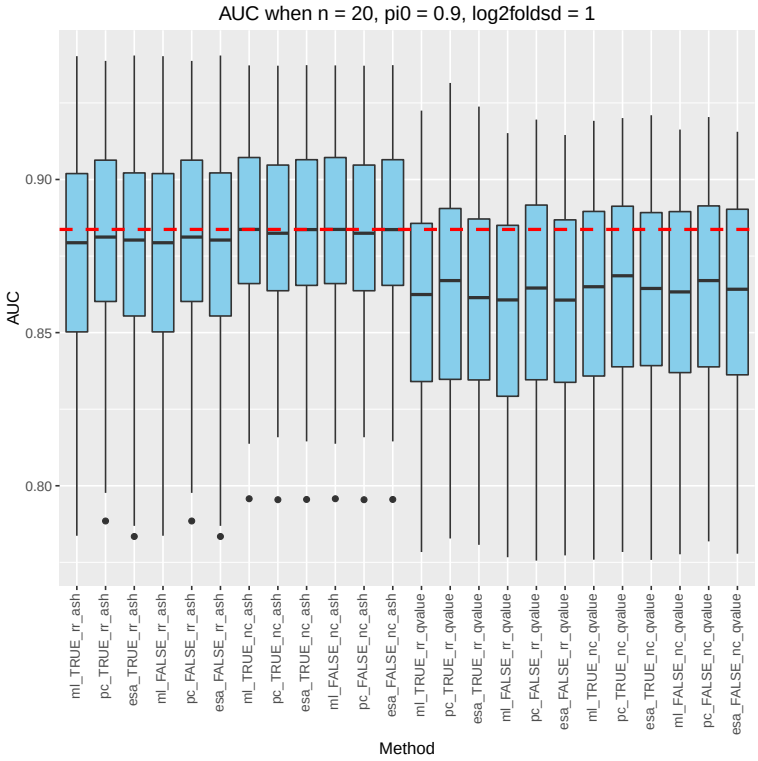

Est of pi0 when n = 20, pi0 = 0.5, log2foldsd = 1
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

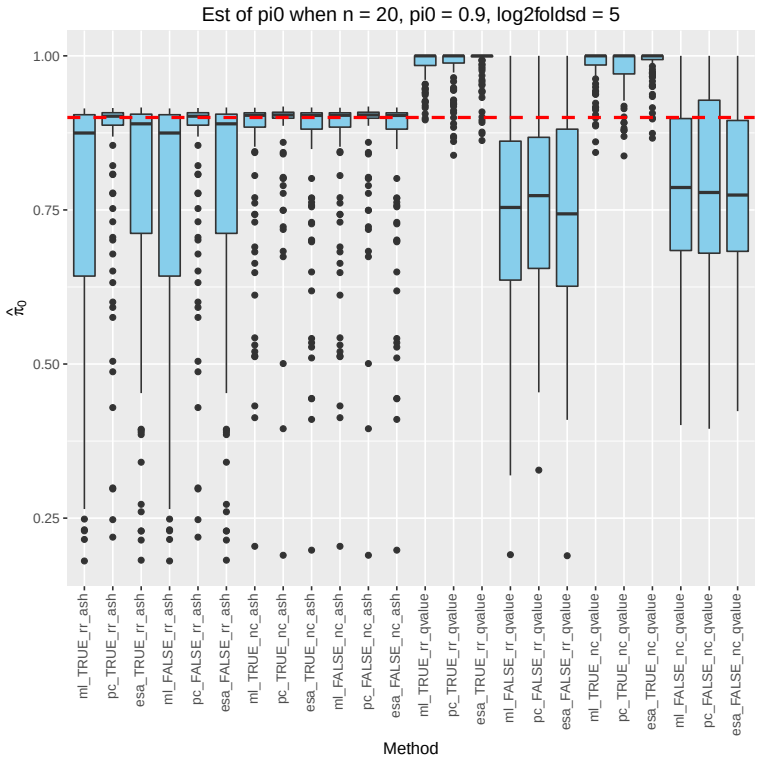

AUC when n = 20, pi0 = 0.5, log2foldsd = 1
0.90
0.85
0.80
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

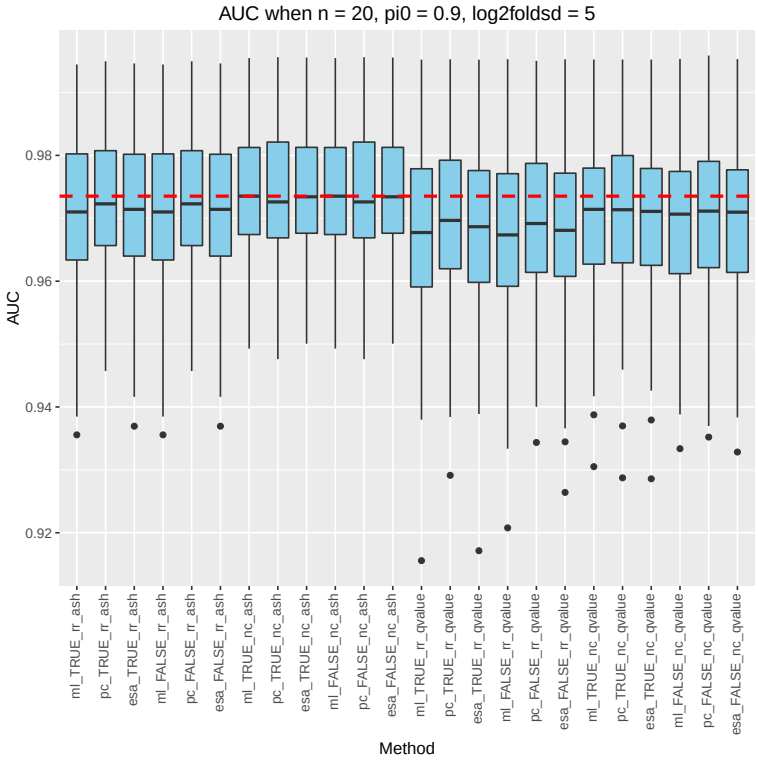

Est of pi0 when n = 20, pi0 = 0.5, log2foldsd = 5
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

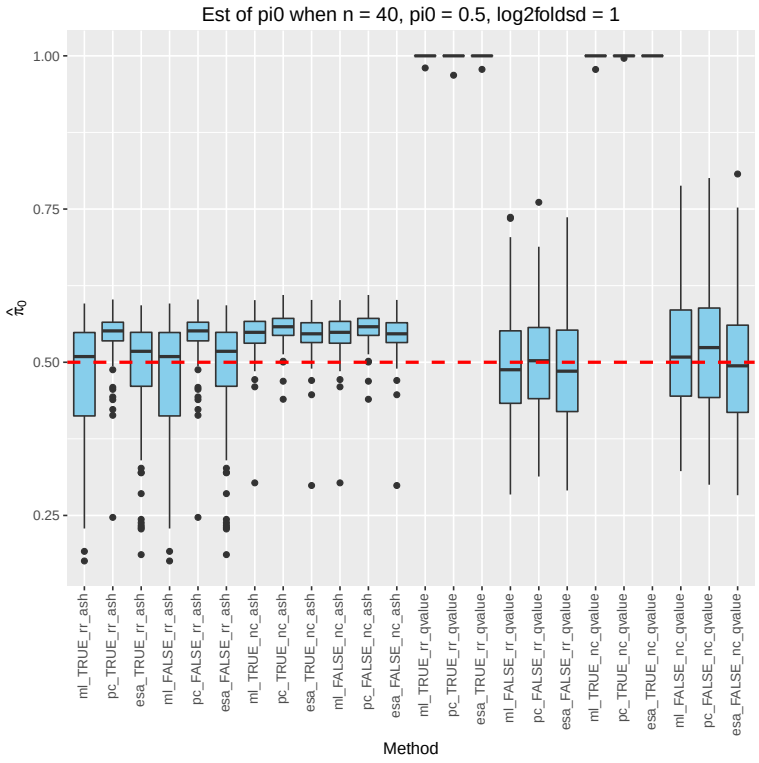

AUC when n = 20, pi0 = 0.5, log2foldsd = 5
0.99
0.98
0.97
0.96
0.95
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

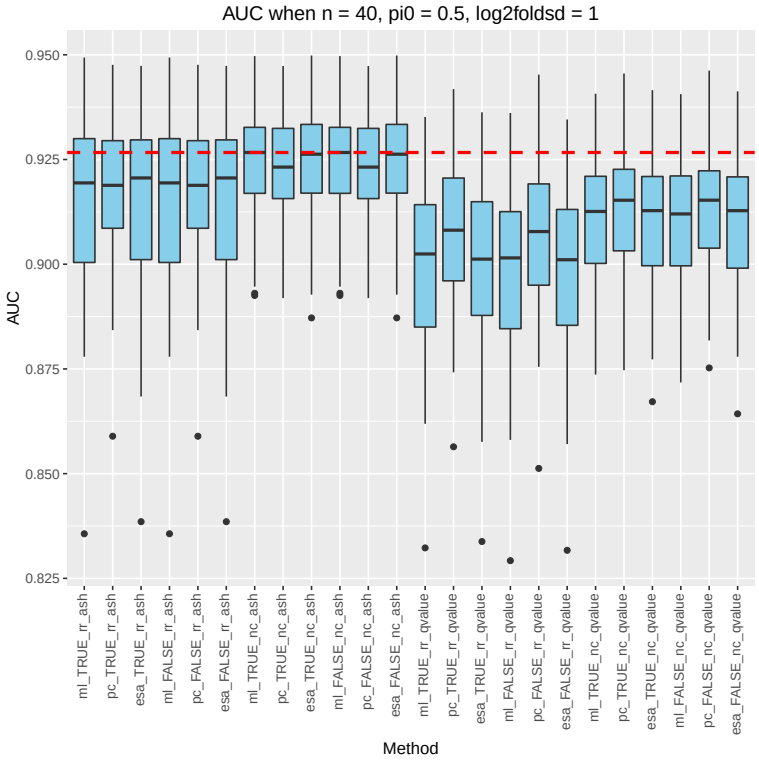

Est of pi0 when n = 20, pi0 = 0.9, log2foldsd = 1
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

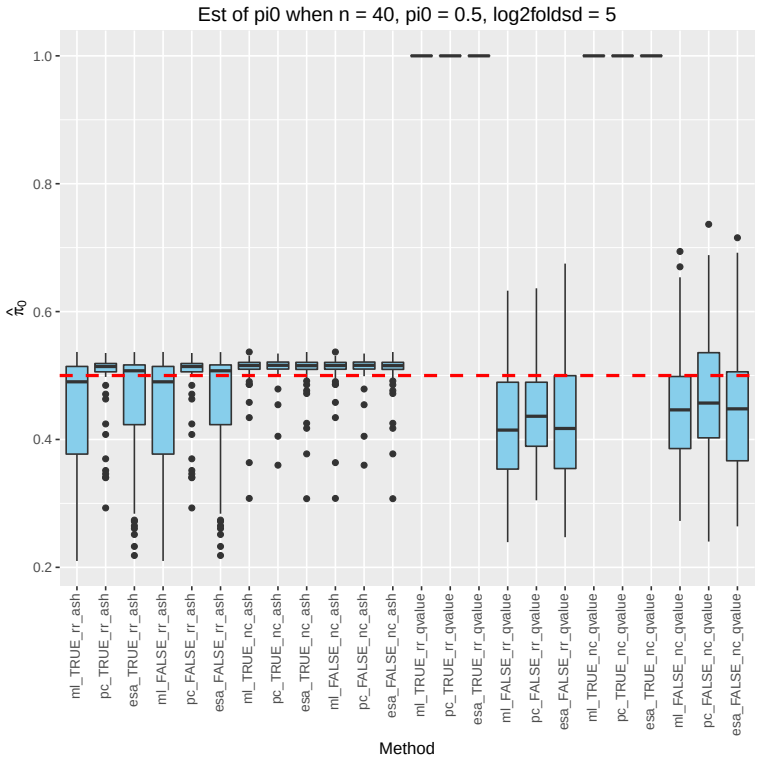

AUC when n = 20, pi0 = 0.9, log2foldsd = 1
0.90
0.85
0.80
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

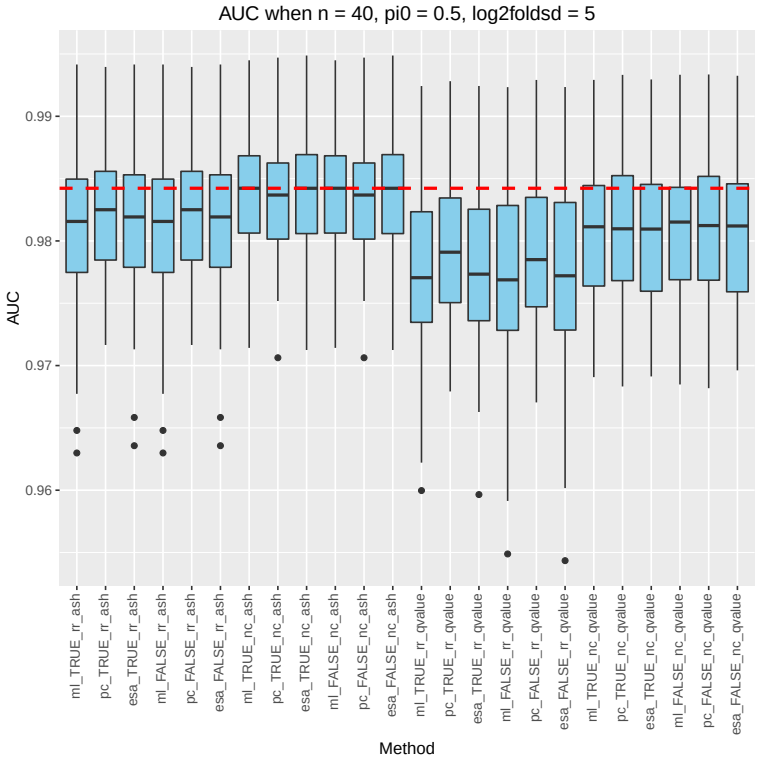

Est of pi0 when n = 20, pi0 = 0.9, log2foldsd = 5
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

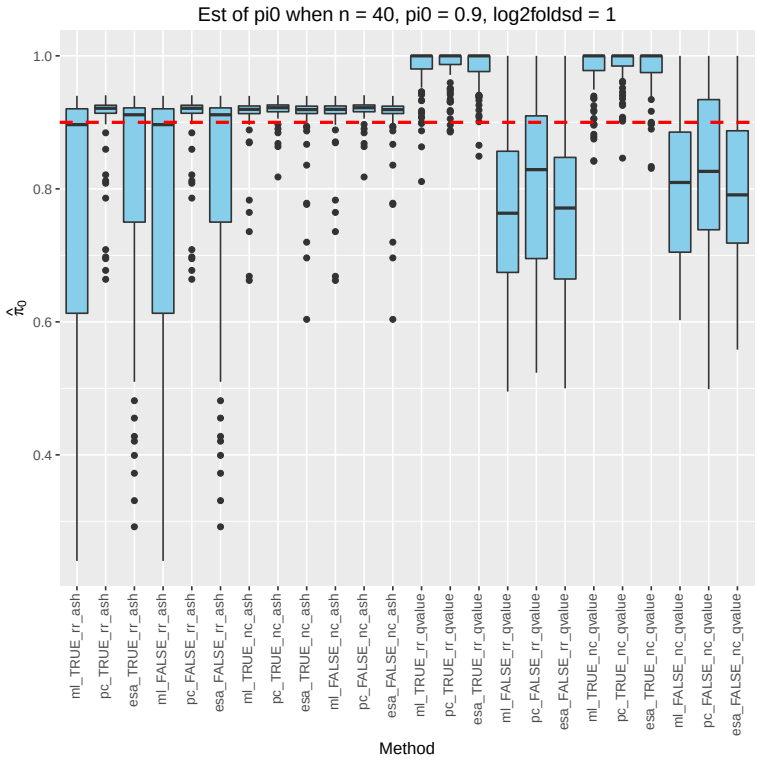

AUC when n = 20, pi0 = 0.9, log2foldsd = 5
0.98
0.96
0.94
0.92
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

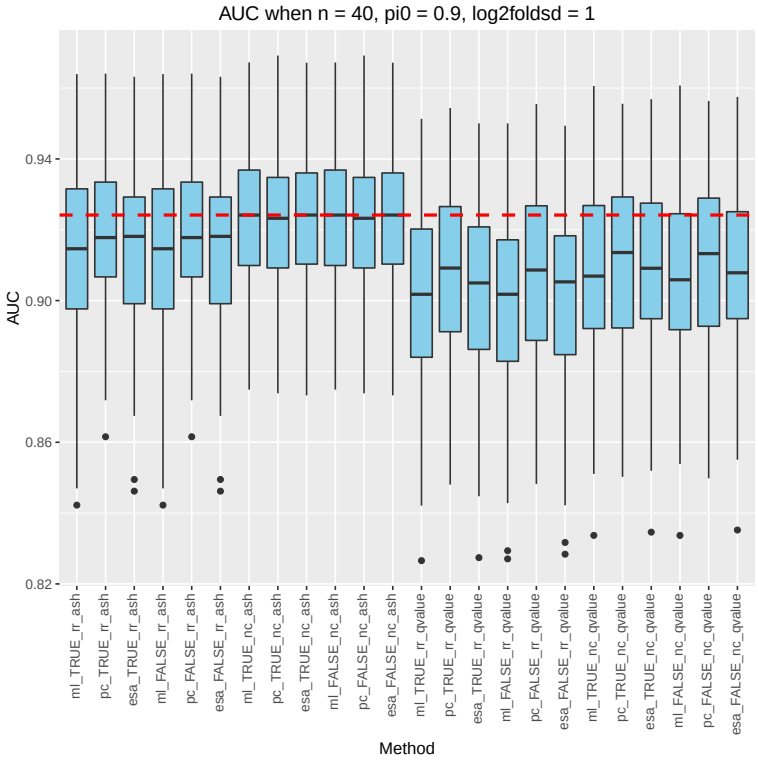

Est of pi0 when n = 40, pi0 = 0.5, log2foldsd = 1
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

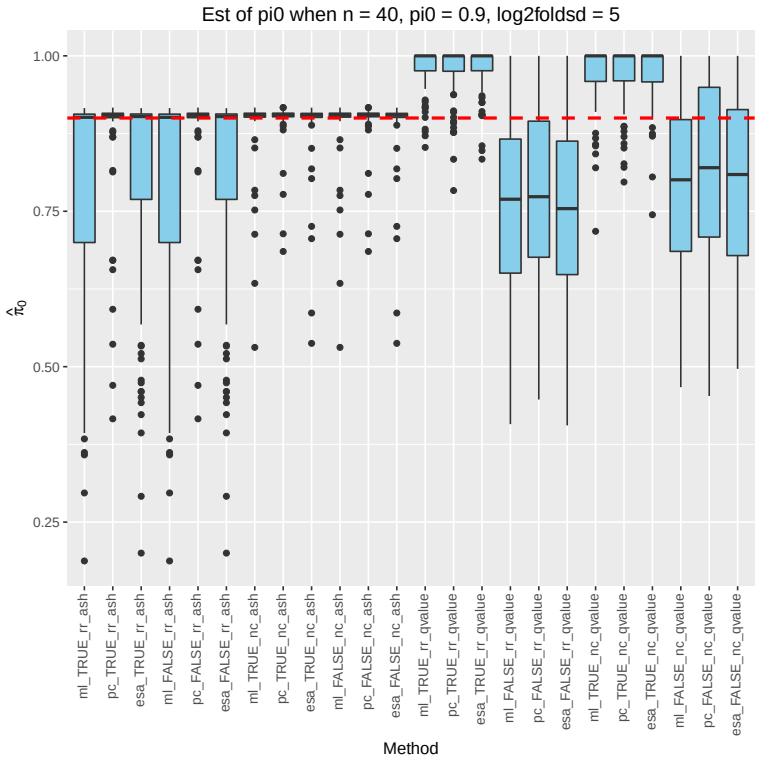

AUC when n = 40, pi0 = 0.5, log2foldsd = 1
0.950
0.925
0.900
0.875
0.850
0.825
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

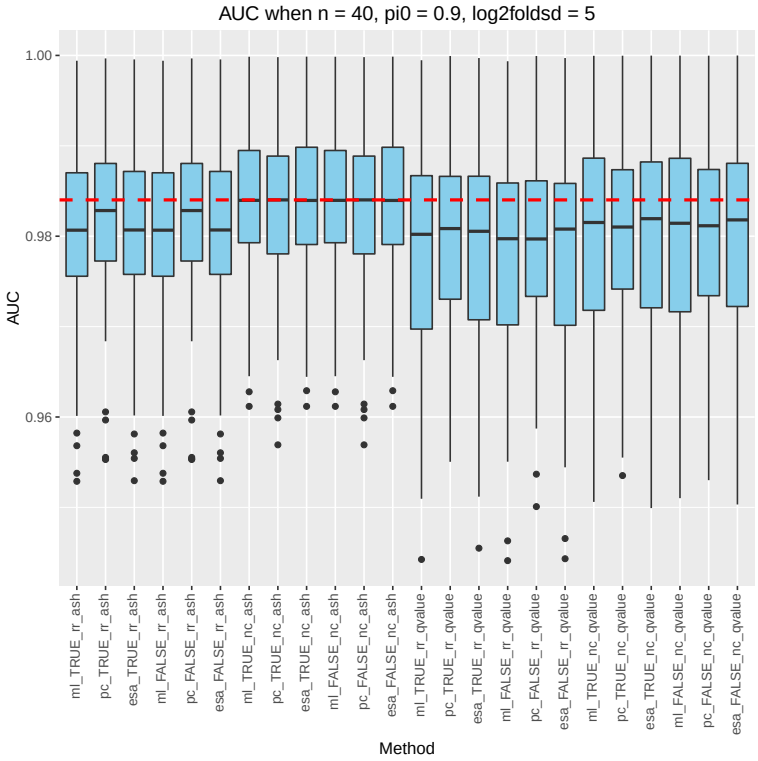

Est of pi0 when n = 40, pi0 = 0.5, log2foldsd = 5
1.0
0.8
0.6
0.4
0.2
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method


AUC when n = 40, pi0 = 0.5, log2foldsd = 5
0.99
0.98
0.97
0.96
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method


Est of pi0 when n = 40, pi0 = 0.9, log2foldsd = 1
1.0
0.8
0.6
0.4
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method


AUC when n = 40, pi0 = 0.9, log2foldsd = 1
0.94
0.90
0.86
0.82
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method


Est of pi0 when n = 40, pi0 = 0.9, log2foldsd = 5
1.00
0.75
0.50
0.25
$\hat{\pi}_0$
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method


AUC when n = 40, pi0 = 0.9, log2foldsd = 5
1.00
0.98
0.96
AUC
ml_TRUE_rr_ash
pc_TRUE_rr_ash
esa_TRUE_rr_ash
ml_FALSE_rr_ash
pc_FALSE_rr_ash
esa_FALSE_rr_ash
ml_TRUE_nc_ash
pc_TRUE_nc_ash
esa_TRUE_nc_ash
ml_FALSE_nc_ash
pc_FALSE_nc_ash
esa_FALSE_nc_ash
ml_TRUE_rr_qvalue
pc_TRUE_rr_qvalue
esa_TRUE_rr_qvalue
ml_FALSE_rr_qvalue
pc_FALSE_rr_qvalue
esa_FALSE_rr_qvalue
ml_TRUE_nc_qvalue
pc_TRUE_nc_qvalue
esa_TRUE_nc_qvalue
ml_FALSE_nc_qvalue
pc_FALSE_nc_qvalue
esa_FALSE_nc_qvalue
Method

References

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

John D Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):479–498, 2002.