

Better Rhat and Calibrate with KN Involves

Yixiang Luo

October 28, 2021

[section] [section] [section] equation]{Lemma}

1 A better Rhat: conservative but efficient recursive refinement

Recall we want to decide if

$$\mathbb{E}_{H_j} [f_j \mid S_j] \leq 0,$$

where

$$f_j = \text{DP}_j - b_j, \quad \text{DP}_j = \frac{\mathbf{1}\{j \in \mathcal{R}^{\text{Kn}}\} \vee \mathbf{1}\{p_j \leq c\}}{|\widehat{R} \cup \{j\}|}, \quad b_j = \frac{\alpha \cdot \mathbf{1}\{j \in \mathcal{R}^{\text{EsKn}}\}}{\widehat{\text{FDP}}(\text{EsKn}) |\mathcal{R}^{\text{EsKn}} \cup \{j\}|}.$$

Any \widehat{R} satisfying $\mathcal{R}^{\text{Kn}} \subseteq \widehat{R} \subseteq \mathcal{R}^{\text{cKn}}$ is valid. In principle, we can do reversive refinement like in dBH but it would be computationally infeasible for cKnockoff. However, if we restrict our attention only to those very promising features in calibration, the calculation may be very fast and reliable. And a few selection in \widehat{R} may help a lot if $\mathcal{R}^{\text{Kn}} = \emptyset$. In particular, let

$$\widehat{R} = \mathcal{R}^{\text{pcKn}} := \mathcal{R}^{\text{Kn}} \cup \{j \in S_{\text{prom}} : E_j \leq 0\},$$

$$S_{\text{prom}} = \{j : p_j \leq 0.2\alpha/m, |W_j| \text{ large}\}, \quad \text{enforce } |S_{\text{prom}}| \leq K_{\text{prom}} = \text{say } 5.$$

1.1 tricks in computing pcKn

Computing E_j can be viewed as a almost 1-d integral on the right-sided sampled p-value $p_j^r(mc)$ varies from 0 to 1 (there is no “almost” if we further conditional on the residue direction). We can compute E_j by a simple numerical integral on equal-spaced grid points of $p_j^r(mc)$ with step size h :

$$E_j \approx \sum_{i=1}^{1/h} f_j(p_j^r(mc) = ih) \cdot h.$$

When h is tiny, we expect the error is small.

Note by the definition of DP_j and b_j , we have

$$\int_{\{f_j > 0\}} f_j \leq p_j(\text{obs}).$$

To claim $E_j \leq 0$, it suffices to show

$$p_j(\text{obs}) + \int_{\{p_j(mc) > p_j(\text{obs})\}} f_j \leq 0.$$

Hence we can pick the numerical integral grid points to be (assume $p_j^r(\text{obs}) < 1/2$)

$$p_j^r(mc) = p_j^r(\text{obs}) + h, \quad p_j^r(\text{obs}) + 2h, \quad \dots \quad p_j^r(\text{obs}) + K_{\text{try}}h, \quad K_{\text{try}} = \text{say } 3.$$

Once we see

$$\sum_{i=1}^k f_j(p_j^r(mc) = p_j^r(obs) + kh) \cdot h + p_j(obs) \leq 0,$$

we stop and reject j . Otherwise we don't reject after K_{try} tries.

This trick makes computing \mathcal{R}^{pcKn} to be at most as expensive as $K_{prom} \cdot K_{try}$ times evaluating knockoff statistics.

The question remains is how to choose h . The calling of \mathcal{R}^{pcKn} indicates knockoff is making no rejection. Hence we expect knockoff will also make no rejection when $p_j(mc)$ is even a bit larger than $p_j(obs)$. So We could estimate b_j as

$$b_j \approx \frac{\alpha}{1/\alpha - 1}$$

when $p_j(mc)$ is close to $p_j(obs)$. Then we can set

$$h = p_j(obs) / \frac{\alpha}{1/\alpha - 1}$$

and hope to reject j at the first try. For $m = 100$ and $\alpha = 0.2$, such h is at most roughly 0.01, which I think is the largest h we can trust.

1.2 tricks in using pcKn

Now let's come back to the "outer" calibration computation of E_j and treat \mathcal{R}^{pcKn} as a black box.

There are some tricks when employing \mathcal{R}^{pcKn} to reduce computational burden.

1. Only use $\hat{R} = \mathcal{R}^{pcKn}$ if DP_j has numerator > 0 .
2. Only use $\hat{R} = \mathcal{R}^{pcKn}$ when $\mathcal{R}^{Kn} = \emptyset$.
3. Only use $\hat{R} = \mathcal{R}^{pcKn}$ if we think it would convert a "no reject" to "reject". We can change our mind as the calculation goes on. In particular, we may compute E_j with $\hat{R} = \mathcal{R}^{Kn}$ and
 - if $E_j \leq 0$, reject j ;
 - if $E_j > 0$, compute R^* as the desired new denominator of DP_j for "those MC samples who has $DP_j = 1$ " in order to make the new $E_j \leq 0$. If $R^* > K_{prom}$, don't reject j .
 - if $R^* \leq K_{prom}$, but as we have computed a few \mathcal{R}^{pcKn} , if in average $|\mathcal{R}^{pcKn}| < R^*$, don't reject j .
 - otherwise replace the denominator of DP_j for "those MC samples who has $DP_j = 1$ " by the new denominator $\hat{R} = \mathcal{R}^{pcKn}$ and make rejection decision accordingly.

Trick 1 is of course valid. Trick 2 is valid because it's equivalent to set the screening set $S_{prom} = \emptyset$ if $\mathcal{R}^{Kn} = \emptyset$. And the screening set is allowed to depend on y in an arbitrary way. As for trick 3, imagine a calculation of E_j that employ $\hat{R} = \mathcal{R}^{pcKn}$ for every MC sample. The result will control FDR as we have seen. Now if we replace f_j calculated from some MC samples by some new f_j employing $\hat{R} = \mathcal{R}^{Kn}$, we a.s. make E_j larger. Hence it's conservative and hence safe.

1.3 sampling: pairing samples from (f_j>0) and (rest)?

In generating MC samples for computing E_j , recall that we generate one sample from $\{p_j(mc) \leq p_j(obs)\}$ and one from the rest and take their weighted average to get one sample for f_j . We expect this to reduce the variance of f_j samples as it pairs a positive and a negative f_j . Some exploratory experiments a month ago indicated it requires less MC samples to have the confidence intervals exclude 0. I did a larger-sized experiment on IID-Normal design and confirms it: the number of MC samples required by pairing sampling is in between 1/2 and 1 of the ones required by naive sampling.

A problem for pairing sampling is that we might over sample from the region $\{p_j(mc) \leq p_j(obs)\}$ when it is comparably very small. And it is the region where we apply recursive refinement. This may make the pairing sampling be computationally heavier than naive sampling. Hence in the implementation for the following experiments, I use the naive sampling as a start point.

1.4 heuristic analysis on MCC

One motivation of this idea is to beat the hybrid method cKnockoff(0.8) at MCC design. But heuristic analysis shows this can't be achieved.

Recall in MCC, the normalized $\Sigma = (X^T X)^{-1}$ has diagonal 1 and off-diagonal 0.5. Let's think about z-statistics instead of t-stat for simplicity. Suppose $j = 1$ and the first 10 hypotheses are nonnulls $\beta_{1-10} = 10$. And suppose we observe $Z_{1-10} = 10$ and $Z_{11-100} = 0$. Then conditional on S_1 ,

$$Z_{-1}(mc) = Z_{-1}(obs) + 0.5(Z_1(mc) - Z_1(obs)).$$

Consider two cases in $\{p_j(mc) \leq p_j(obs)\}$.

- $Z_1(mc) \geq Z_1(obs)$. We have $Z(mc) \approx Z(obs)$ and both Bonferroni (in cKnockoff(0.8)) and \mathcal{R}^{pcKn} can make a few rejections.
- $Z_1(mc) \leq -Z_1(obs)$. We have roughly $Z_1 = -10$, $Z_{2-10} = 0$, and $Z_{11-100} = -10$. So Bonferroni can reject 91 hypotheses. However, these 91 hypotheses have very small W -statistics while hypotheses 2-10 have very large p-value. \mathcal{R}^{pcKn} cannot make any rejection here.

1.5 numerical results

Please see the figure below. cKnockoff1 is the one with $\hat{R} = \mathcal{R}^{pcKn}$. It indeed improves the power but still doesn't beat cKnockoff(0.8) under MCC.

As for running time, I pick the settings to make \mathcal{R}^{pcKn} to be employed easily and set $K_{try} = 3$ and $K_{prom} = 5$. The overall running time of 100 trials of y for $m = 100$ increases from about 3min (for cKnockoff) to about 4min (for cKnockoff1).

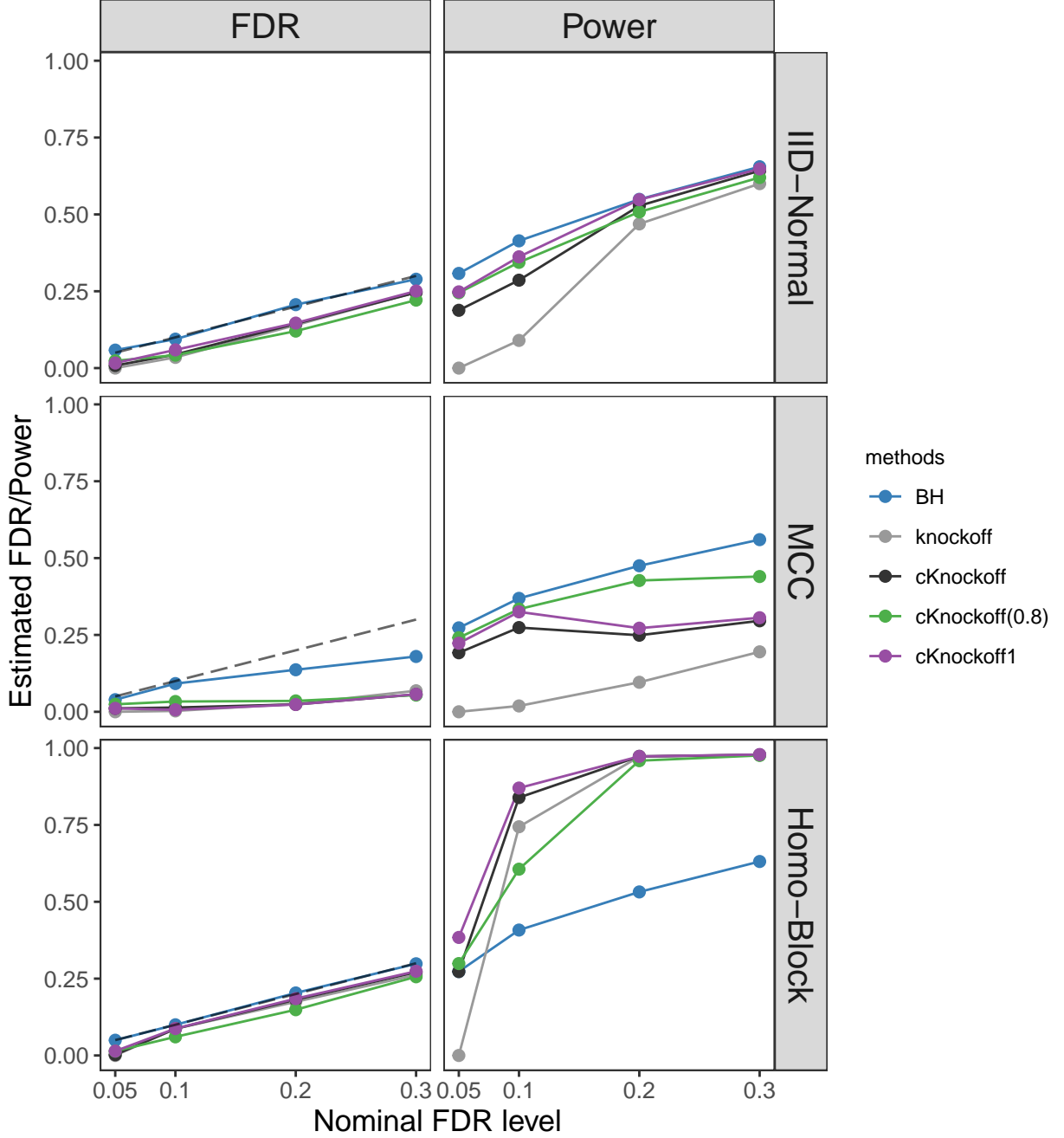


Figure 1: Performance of the improved Rhat. cKnockoff1 is the one with $R_{\text{hat}} = R(\text{cpKn})$. $m = 100$.

2 Calibrate with KN involves

Is it possible to further improve the method and try to beat the hybrid one under MCC? This motivated a new idea I call “calibration with knockoff involves” (definitely a bad name). In the following, I denote it as $\mathcal{R}^{\text{kcKn}}$.

The idea is simple: we reject

$$\mathcal{R}^{\text{kcKn}} := \mathcal{R}^{\text{Kn}} \cup \{j : p_j \leq c_j^*, \text{ knockoff prefer } j\},$$

knockoff prefer $j := |W_j|$ is at least the $(2K_{relax}/\alpha)$ -th largest, **or**
assign $W_j \leftarrow |W_j|$, rerun knockoff with the new W-stats at level $K_{relax}\alpha$, ture if j is rejected,
where $K_{relax} \geq 1$ is say 1.5.

c_j^* is computed from the accordingly adjusted E_j with

$$DP_j = \frac{\mathbf{1}\{j \in \mathcal{R}^{Kn}\} \vee \mathbf{1}\{p_j \leq c, \text{ knockoff prefer } j\}}{|\widehat{R} \cup \{j\}|}.$$

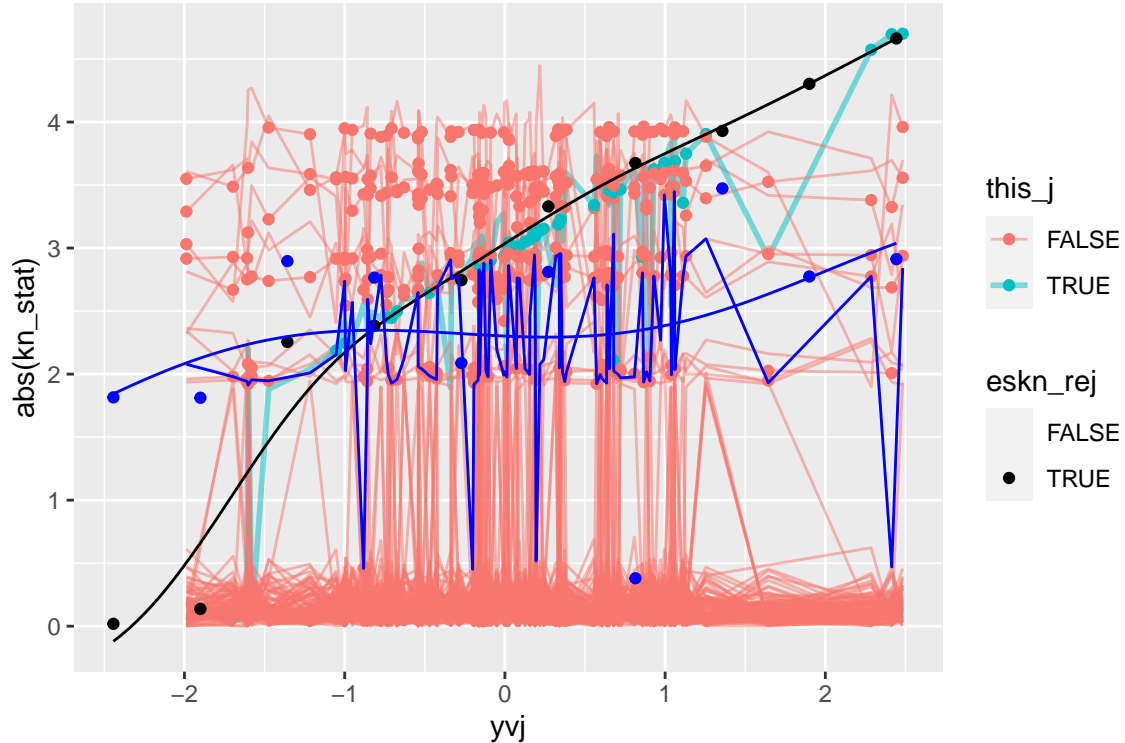
In other words, we use “knockoff prefer” to do a screening and take the screening into consideration when we do calibration.

Before we look at its advantage, I want to first show it’s computationally not harder, as illustrated below. - the W-statistics to decide “knockoff prefer” is already computed for each MC sample. - DP_j is still increasing in c . Hence we don’t need to find c_j^* as well. - Adding the “knockoff prefer” makes $\{f_j > 0\}$ shrink. So our sampling scheme based on the conservative approximation still works.

2.1 why using “knockoff prefer”

In general, the failing cases for knockoff we have seen (small sample issue, MCC) are not because the ordering based on $|W|$ is bad. Instead, the rank of $|W|$ is a very good indicator for the alternatives in all these cases we have seen.

In particular, I want to claim that $\mathcal{R}^{kcKn} \supseteq \mathcal{R}^{cKn}$ is **heuristically/approximately** true. This is based on a both consistently observed and heuristically true phenomenon: conditional on S_j , $|W_j|$ achieve its maximum when $p_j \rightarrow 0$, as illustrated below.



Since

we also do the p-value screening, we only consider j not selected by Knockoff and has p_j small.

The claim consists of two parts:

1. If $j \in \mathcal{R}^{\text{cKn}}$, then j will not be screened out by “knockoff prefer”.

For j to be rejected by calibration in \mathcal{R}^{cKn} , it has two cases.

- Most samples in $\{p_j(mc) \leq p_j(obs)\}$ also have $j \in \mathcal{R}^{\text{Kn}}$. If this is the case, the second part of “knockoff prefer” will not screen out j .
- Most samples in $\{p_j(mc) \leq p_j(obs)\}$ has $j \notin \mathcal{R}^{\text{Kn}}$. To have j rejected, by cKnockoff, we must have many samples in $\{p_j(mc) > p_j(obs)\}$ such that j is rejected by early stopping knockoff (and due to early stopping). If this is the case then the first part of “knockoff prefer” will not screen out j .
- About half of samples in $\{p_j(mc) \leq p_j(obs)\}$ have $j \in \mathcal{R}^{\text{Kn}}$ while the other half not but still $|\mathcal{R}^{\text{Kn}}| > 0$. I expect the second part of “knockoff prefer” with the relaxing factor will not screen out j in this case.

This heuristic analysis also tells a sad story: cKnockoff is only able to make roughly at most $2/\alpha$ more rejections than knockoff. With MCC design and a strong signal, if $m = 2000$, $|\mathcal{H}_1| = 100$, and $\alpha = 0.1$, then we shouldn’t expect cKnockoff to perform comparably with dBH. In other words, we are expecting too much of cKnockoff to solve the whiteout issue.

2. $\mathcal{R}^{\text{kcKn}}$ can make more rejections.

In general, adding the “knockoff prefer” screener will make c_j^* larger. Hence $\mathcal{R}^{\text{kcKn}}$ is more powerful.

In particular, consider the case described in Section 1.4 dot 2 “ $Z_1(mc) \leq -Z_1(obs)$ ”. $|W_j|$ will be small and hence j will be screened out. Therefore we will have $DP_j = 0$ instead of 1 in this case, which brings E_j down.

2.2 numerical results

Please see the results below. $\mathcal{R}^{\text{kcKn}}$ is a bit better than \mathcal{R}^{cKn} . Knockoff-cali-kn is the one with “knockoff prefer” screener. But the power gain is not large and it still doesn’t beat cKnockoff(0.8).

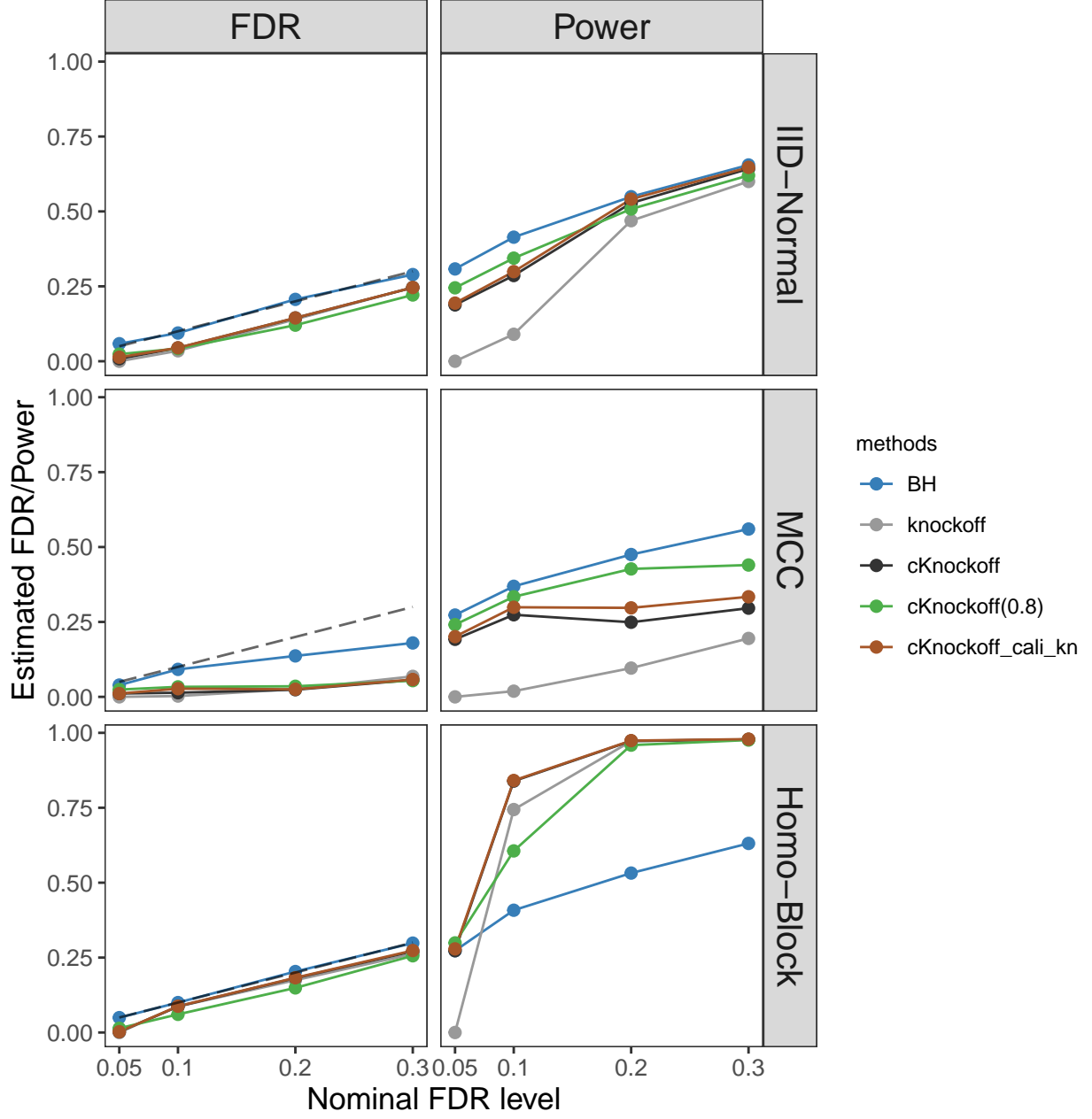


Figure 2: Performance of kcKn. cKnockoff-cali-kn is the one with 'knockoff prefer' screener. $m = 100$.

3 Combine the two ideas

Let's combine the two ideas (\mathcal{R}^{pckn} should be adjusted to be consistent with \mathcal{R}^{kckn}).

Please see the results below. cKnockoff1-cali-kn is the method that combine the two ideas. We have a substantial power gain over cKnockoff but it still doesn't fully beat cKnockoff(0.8).

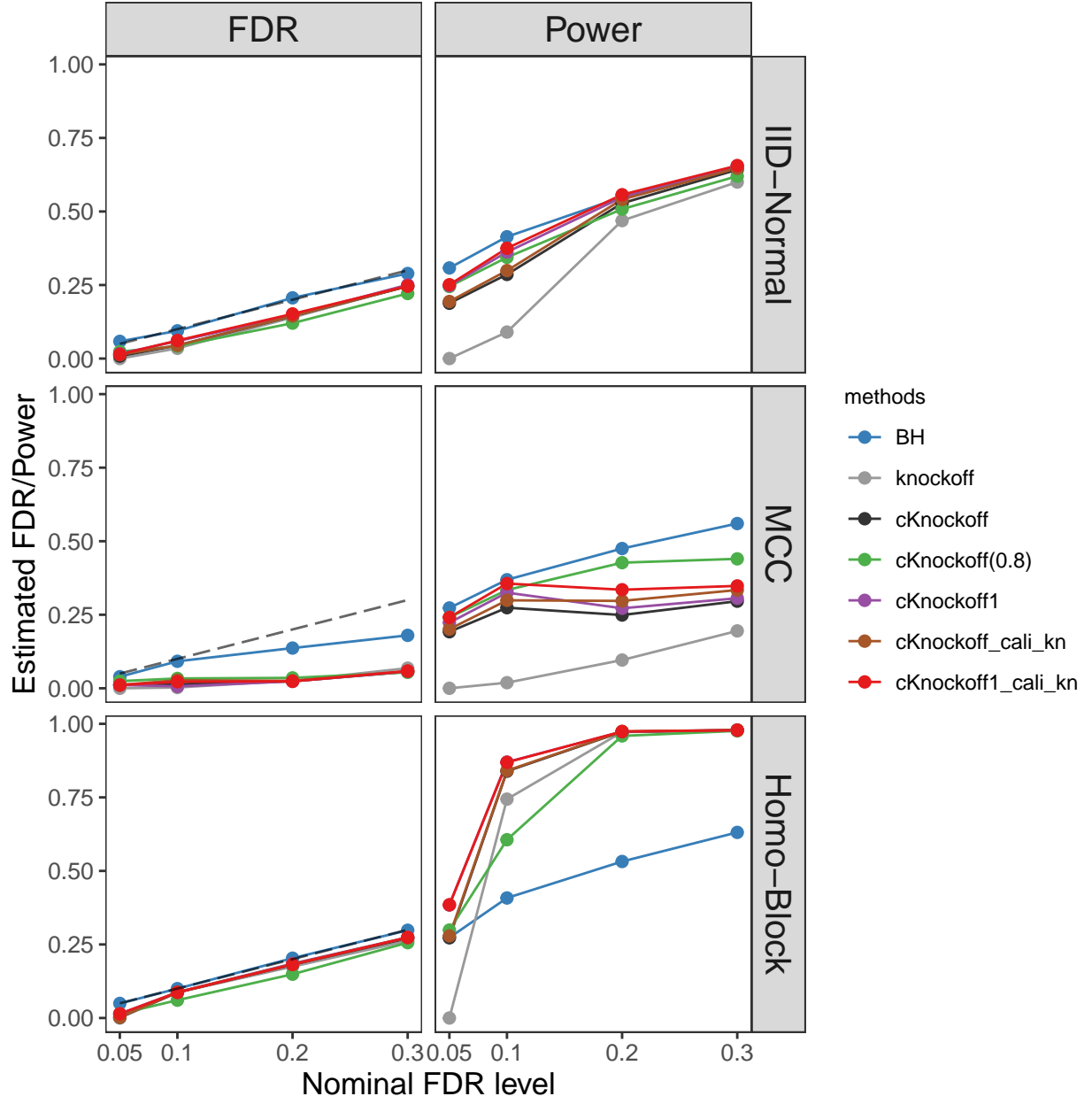


Figure 3: Performance of the combined method, denoted as cKnockoff1-cali-kn. $m = 100$.