Accumulation test demo - simulated data

Ang Li and Rina Foygel Barber

May 27, 2015

Introduction

This demo reproduces the ordered hypothesis testing simulation in the paper:

- Ang Li and Rina Foygel Barber, "Accumulation tests for FDR control in ordered hypothesis testing" (2015). Available from http://arxiv.org/abs/1505.07352 (http://arxiv.org/abs/1505.07352)

The demo was run using R version 3.2.0.

The demo considers an ordered hypothesis testing problem with n hypotheses, of which n_1 contain true signals. We produce an ordered list of p-values as follows:

- First, generate $Z_i^{\text{prior}} \sim N(\mu_i, 1)$ where $\mu_i = 0$ for index i corresponding to a null hypothesis, and $\mu_i = \mu_{\text{intermix}}$ for a nonnull (true signal) index i. This z-score represents prior information, e.g. data from a prior experiment.
- Then, reorder the hypotheses according to their z-scores, with the largest z-scores (in absolute value) first. If μ_{intermix} is large, then this yields good separation: the n_1 true signals will mostly be placed early in the list. If μ_{intermix} is small, however, then this yields poor separation: many true signals will be mixed in with the nulls throughout the list.
- Next, for each hypothesis, generate a new z-score $Z_i \sim N(\mu_i, 1)$ where now $\mu_i = 0$ or $\mu_i = \mu_{\text{signal}}$, for either a strong signal strength (μ_{signal} large) or a weak signal strength (μ_{signal} small). Produce p-values p_i with a two-sided z-test. See paper for details.

We then run accumulation tests, specifically, we text the HingeExp method (with parameter $C = 2$), the SeqStep and SeqStep+ (Barber and Candès 2014) (each with parameters $C = 2$), and the ForwardStop method (G'Sell et al 2013).

Setup

We first define some functions for running the simulation and for plotting the results. (Code for this setup hidden in the output, but is visible in .Rmd file)

```
Setup_simulation_functions()
```

```
## [1] "Simulation functions defined."
```

```
Setup_simulation_plotting_functions()
```

```
## [1] "Simulation plotting functions defined."
```

```
source('accumulation_test_functions.R')
```

Parameters for the simulations:

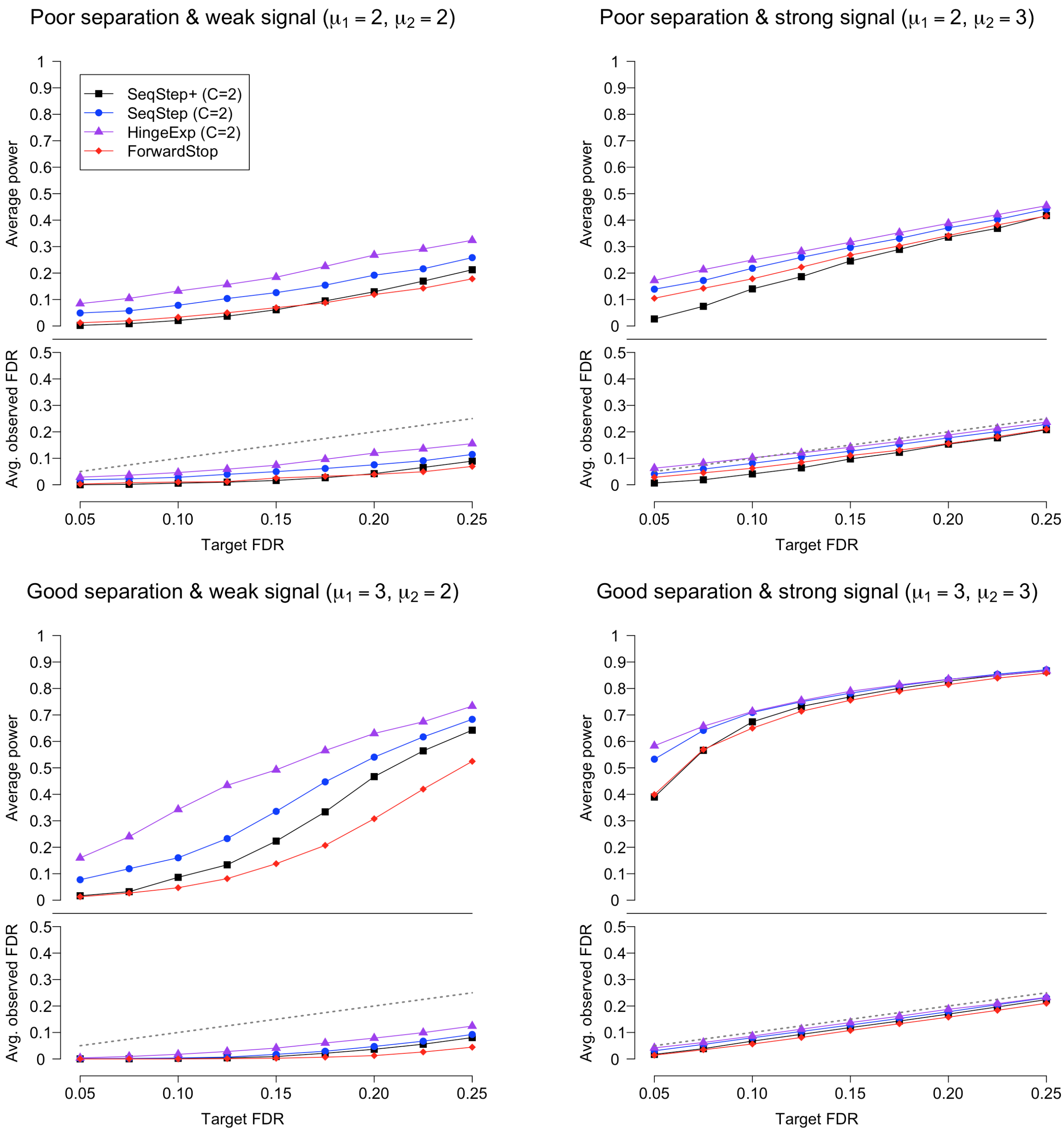
```
hfuns=c(create_SeqStep_function(C=2),create_SeqStep_function(C=2),create_HingeEx
p_function(C=2),create_ForwardStop_function())
numerator_pluses=c(2,0,0,0)
denominator_pluses=c(1,0,0,0)
names=c('SeqStep+ (C=2)','SeqStep (C=2)','HingeExp (C=2)','ForwardStop')
mu_low=2;mu_high=3;
n=1000; ntrials=100;
pr_list=c(0.2,0.1) # proportion of true signals
alphaseq=(2:10)/40
seeds=1:ntrials
```

Simulations part 1

Settings: $n = 1000$ hypotheses, $k^\star = 200$ true signals

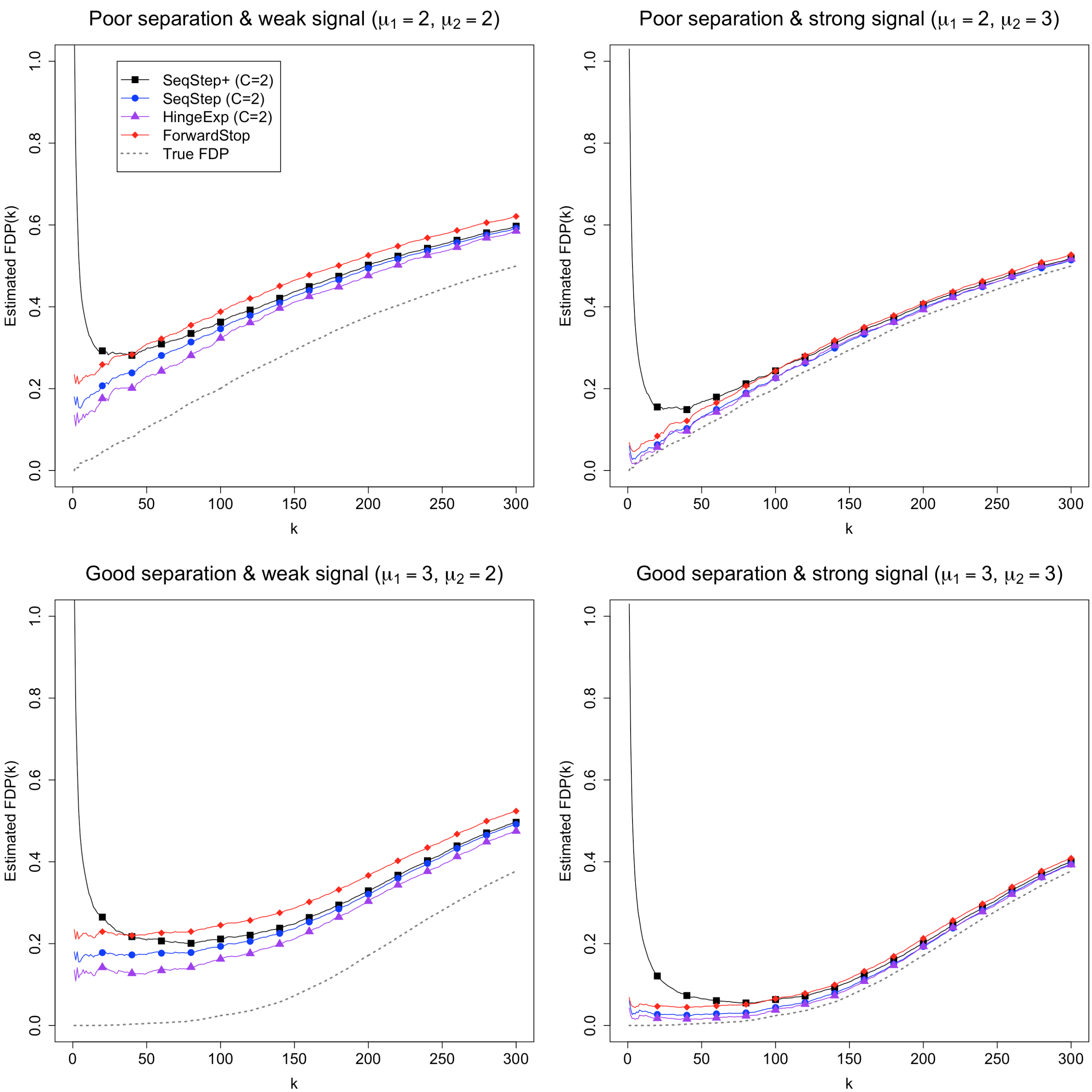
First, we run each method with target FDR level α (ranging in $\alpha = \{0.05, 0.075, 0.1, \ldots, 0.25\}$). For each method and each α, we record the actual FDR attained, and the power to detect signals, averaged over 100 trials.

```
pr=pr_list[1]
#pdf(paste0('FDR_vs_power_plot_n_',n,'_kstar_',n*pr,'.pdf'),14,14)
par(mfrow=c(2,2))
for(i in 1:2){for(j in 1:2){
  mu1=c(mu_low,mu_high)[i];mu2=c(mu_low,mu_high)[j]
  temp=CompEffectFunc(n,pr,mu1,mu2,hfuns,numerator_pluses,denominator_pluses,seed
s, alphaseq)
FDR_and_power_plot(temp[[2]],temp[[1]],names,cols=c('black','blue','purple','re
d'),pchs=15:18,alpha=alphaseq, alpha_display_div=2, title=get_titles(mu_low,mu_hig
h)[i,j],show_legend=(i==1 && j==1))
}};#dev.off()
```



Now we look at how each method estimates the FDP, along the sequence of p-values. At each k, we compare the true FDP among the first k p-values, with the estimated FDP at k for each method. (All results averaged over 100 trials).

```r
#pdf(paste0('FDP_vs_k_plot_n_',n,'_kstar_',n*pr,'.pdf'),14,14)
par(mfrow=c(2,2))
for(i in 1:2){for(j in 1:2){
  mu1=c(mu_low,mu_high)[i];mu2=c(mu_low,mu_high)[j]
    temp=FDPSeqFunc(n,pr,mu1,mu2,hfuns,numerator_pluses,denominator_pluses,seeds)
    FDP_vs_k_plot(temp[[1]],temp[[2]],names,cols=c('black','blue','purple','re
d'),pchs=15:18,title=get_titles(mu_low,mu_high)[i,j],kmax=300,show_legend=(i==1 &&
j==1))
}};#dev.off()
```

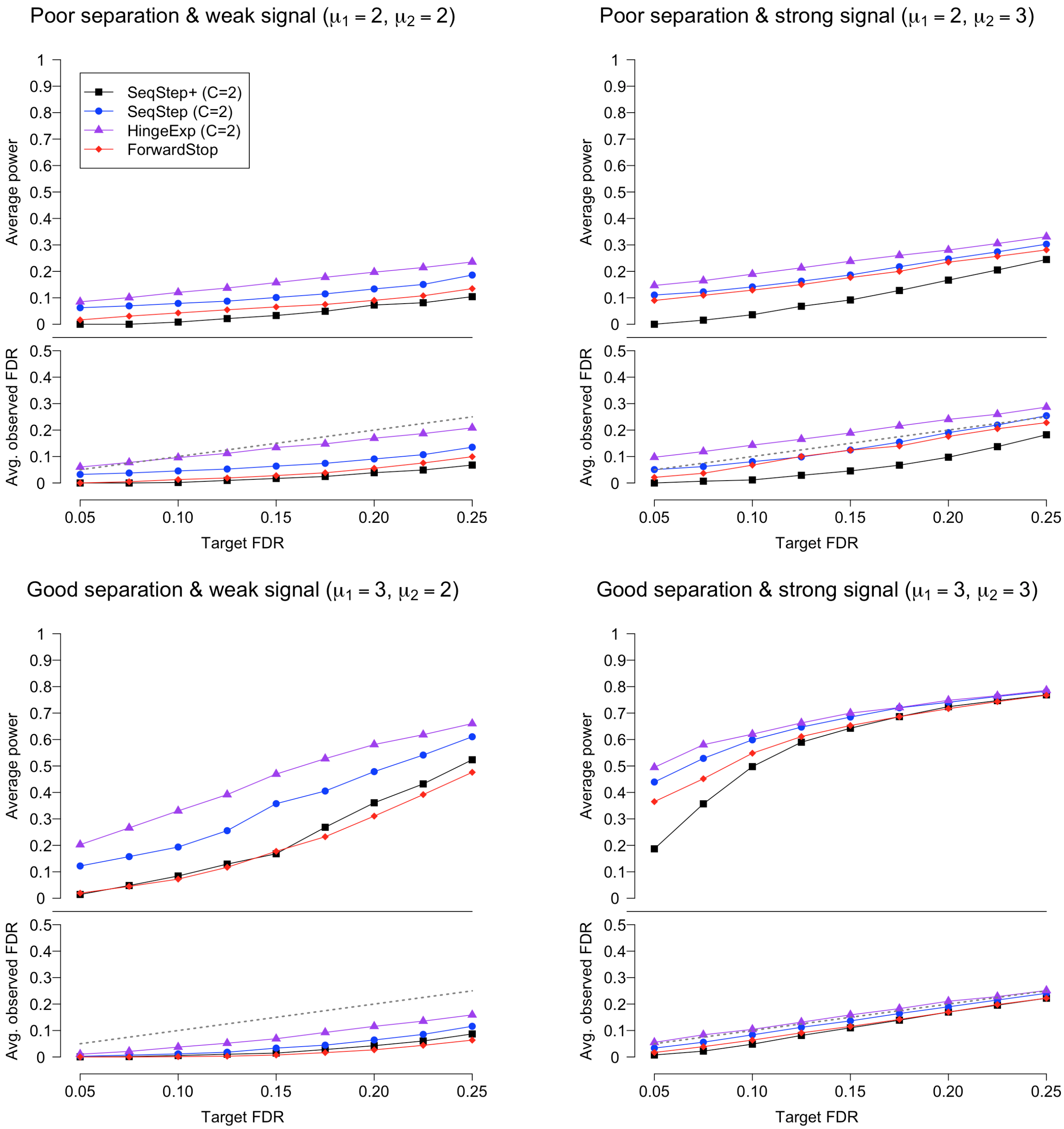


Simulations part 2

Settings: $n = 1000$ hypotheses, $k^\star = 100$ true signals

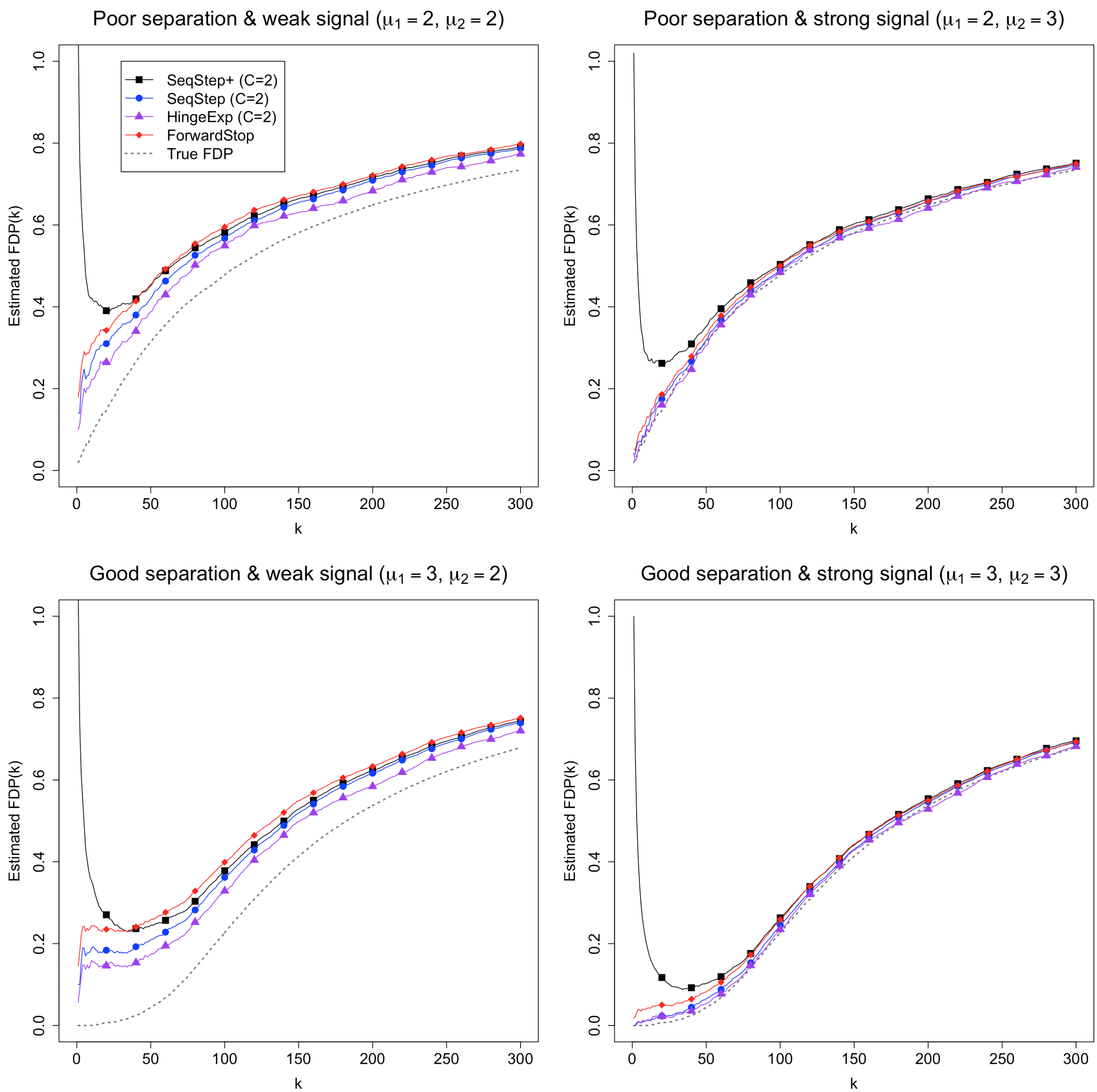
First, we run each method with target FDR level α (ranging in $\alpha = \{0.05, 0.075, 0.1, \dots, 0.25\}$). For each method and each α, we record the actual FDR attained, and the power to detect signals, averaged over 100 trials.

```
pr=pr_list[2]
#pdf(paste0('FDR_vs_power_plot_n_',n,'_kstar_',n*pr,'.pdf'),14,14)
par(mfrow=c(2,2))
for(i in 1:2){for(j in 1:2){
  mu1=c(mu_low,mu_high)[i];mu2=c(mu_low,mu_high)[j]
  temp=CompEffectFunc(n,pr,mu1,mu2,hfuns,numerator_pluses,denominator_pluses,seed
s, alphaseq)
FDR_and_power_plot(temp[[2]],temp[[1]],names,cols=c('black','blue','purple','re
d'),pchs=15:18,alpha=alphaseq, alpha_display_div=2, title=get_titles(mu_low,mu_hig
h)[i,j],show_legend=(i==1 && j==1))
}};#dev.off()
```



Now we look at how each method estimates the FDP, along the sequence of p-values. At each k, we compare the true FDP among the first k p-values, with the estimated FDP at k for each method. (All results averaged over 100 trials).

```r
#pdf(paste0('FDP_vs_k_plot_n_',n,'_kstar_',n*pr,'.pdf'),14,14)
par(mfrow=c(2,2))
for(i in 1:2){for(j in 1:2){
  mu1=c(mu_low,mu_high)[i];mu2=c(mu_low,mu_high)[j]
  temp=FDPSeqFunc(n,pr,mu1,mu2,hfuns,numerator_pluses,denominator_pluses,seeds)
    FDP_vs_k_plot(temp[[1]],temp[[2]],names,cols=c('black','blue','purple','red'),pchs=15:18,title=get_titles(mu_low,mu_high)[i,j],kmax=300,show_legend=(i==1 && j==1))
}};#dev.off()
```



References

Barber, Rina Foygel, and Emmanuel Candès. 2014. "Controlling the False Discovery Rate via Knockoffs." ArXiv Preprint ArXiv:1404.5609.

G'Sell, Max Grazier, Stefan Wager, Alexandra Chouldechova, and Robert Tibshirani. 2013. "False Discovery Rate Control for Sequential Selection Procedures, with Application to the Lasso." ArXiv Preprint ArXiv:1309.5352.