FDR controling method

Jung Da Yeon

2022-08-02

1. BH (Benjamini and Hochberg)

```
if (!require("stats")) install.packages("stats")
p = c(0.0001, 0.0004, 0.0019, 0.0095, 0.0201, 0.0278, 0.0298, 0.0344, 0.0459, 0.3240, 0.4262, 0.5719, 0.6528, 0.7
590, 1.000)
n = length(p)
BH = p.adjust(p, method = "BH", n = length(p))
data.frame(pvals = round(p,3), BH = round(BH, 3))
```

```
##    pvals    BH
## 1  0.000 0.002
## 2  0.000 0.003
## 3  0.002 0.010
## 4  0.010 0.036
## 5  0.020 0.060
## 6  0.028 0.064
## 7  0.030 0.064
## 8  0.034 0.064
## 9  0.046 0.077
## 10 0.324 0.486
## 11 0.426 0.581
## 12 0.572 0.715
## 13 0.653 0.753
## 14 0.759 0.813
## 15 1.000 1.000
```

```
print(paste0('rejection # of FWER is ',sum(p <= 0.05/n, na.rm = TRUE)))
```

```
## [1] "rejection # of FWER is 3"
```

```
print(paste0('rejection # of BH is ',sum(BH <= 0.05, na.rm = TRUE)))
```

```
## [1] "rejection # of BH is 4"
```

2. IHW (Independent Hypothesis Weighting)

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

library(BiocManager)

# BiocManager::install("IHW")
# BiocManager::install("DESeq2")
# BiocManager::install("airway")

library("DESeq2")
library("dplyr")
library("IHW")

data("airway", package = "airway")
dds <- DESeqDataSet(se = airway, design = ~ cell + dex) %>% DESeq # ~ group + condition
deRes <- as.data.frame(results(dds))
ihwRes <- ihw(pvalue ~ baseMean,  data = deRes, alpha = 0.1) #  formula, specified in the form pvalue~covariate

print(paste0('rejection # of FWER is ', sum(deRes$pvalue <= 0.1/length(deRes$pvalue), na.rm = TRUE)))
```

```
## [1] "rejection # of FWER is 1373"
```

```
print(paste0('rejection # of BH is ', sum(p.adjust(deRes$pvalue, method="BH", n = length(deRes$pvalue)) <= 0.1, n
a.rm = TRUE)))
```

```
## [1] "rejection # of BH is 3503"
```

```
print(paste0('rejection # of ihw is ', rejections(ihwRes)))
```

```
## [1] "rejection # of ihw is 4892"
```

```
print(paste0('rejection # of ihw is ', sum(adj_pvalues(ihwRes) <= 0.1, na.rm = TRUE), ', too'))
```

```
## [1] "rejection # of ihw is 4892, too"
```

3. Storey q-value

```
# BiocManager::install("qvalue",force = TRUE)

library(qvalue)

storey_qval <- qvalue(deRes$pvalue, fdr.level = 0.1)


head(data.frame(pval = round(na.omit(storey_qval$pvalues),3), qval = round(na.omit(storey_qval$qvalues),3), signi
ficant = storey_qval$significant))
```

```
##    pval  qval significant
## 1 0.000 0.002        TRUE
## 2 0.065 0.359       FALSE
## 3 0.792 0.996       FALSE
## 4 0.759 0.996       FALSE
## 5 0.694 0.996       FALSE
## 6 0.000 0.000        TRUE
```

```
print(paste0('rejection # of FWER is ', sum(storey_qval$pvalues <= 0.1/length(storey_qval$pvalues), na.rm = TRUE)
))
```

```
## [1] "rejection # of FWER is 1373"
```

```
print(paste0('rejection # of stoery q-value is ', sum(storey_qval$significant)))
```

```
## [1] "rejection # of stoery q-value is 4099"
```

```
print(paste0('rejection # of stoery q-value is ', sum(storey_qval$qvalues <= 0.1, na.rm = TRUE), ', too'))
```

```
## [1] "rejection # of stoery q-value is 4099, too"
```

4. BL (Boka and Leek)

```
# BiocManager::install("swfdr")
library(swfdr)

head(BMI_GIANT_GWAS_sample)
```

```
## # A tibble: 6 × 9
##   SNP        A1    A2    Freq_MAF_Hapmap       b     se     p      N Freq_MAF_…¹
##   <chr>      <chr> <chr>           <dbl>   <dbl>  <dbl> <dbl>  <dbl> <fct>
## 1 rs10510371 T     C             0.025    0.0147 0.0152 0.334 212965 [0.000,0.1…
## 2 rs918232   A     G             0.342   -0.0034 0.0037 0.358 236084 [0.302,0.5…
## 3 rs4816764  A     C             0.00830  0.0163 0.0131 0.213 221771 [0.000,0.1…
## 4 rs17630047 A     G             0.167    0.0004 0.0048 0.934 236177 [0.127,0.3…
## 5 rs4609437  C     G             0.25     0.0011 0.0042 0.793 236028 [0.127,0.3…
## 6 rs11130746 G     A             0.233   -0.0006 0.0042 0.886 235634 [0.127,0.3…
## # … with abbreviated variable name ¹Freq_MAF_Int_Hapmap
```

```
X <- model.matrix(~ splines::ns(N,5) + Freq_MAF_Int_Hapmap, data = BMI_GIANT_GWAS_sample)[,-1]
head(X)
```

```
##   splines::ns(N, 5)1 splines::ns(N, 5)2 splines::ns(N, 5)3 splines::ns(N, 5)4
## 1       7.242962e-01          0.0000000       -0.096623916        0.193411872
## 2       6.214473e-05          0.9873057        0.010529926        0.004207151
## 3       8.482281e-01          0.0000000       -0.054593655        0.109279996
## 4       0.000000e+00          0.9847066        0.012746260        0.005093468
## 5       4.971578e-04          0.9884279        0.009232155        0.003688505
## 6       3.080761e-02          0.9658228        0.002791946        0.001127918
##   splines::ns(N, 5)5 Freq_MAF_Int_Hapmap[0.127,0.302)
## 1      -0.0967879566                                0
## 2      -0.0021049077                                0
## 3      -0.0546863405                                0
## 4      -0.0025463522                                1
## 5      -0.0018457495                                1
## 6      -0.0005644374                                1
##   Freq_MAF_Int_Hapmap[0.302,0.500]
## 1                                0
## 2                                1
## 3                                0
## 4                                0
## 5                                0
## 6                                0
```

```
pi0x <- lm_pi0(BMI_GIANT_GWAS_sample$p, X=X)

# rejection # of FWER
sum(BMI_GIANT_GWAS_sample$p <= 0.05/length(BMI_GIANT_GWAS_sample$p))
```

```
## [1] 64
```

```
# rejection # of BL
sum(pi0x$pi0*BMI_GIANT_GWAS_sample$p <= 0.05)
```

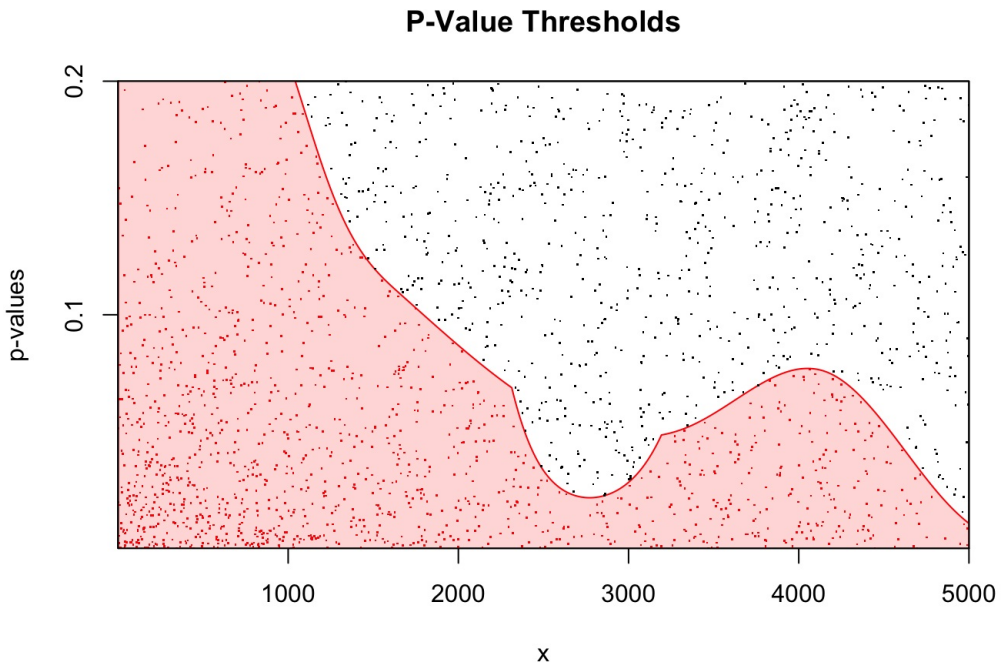
```
## [1] 3981
```

5. AdaPT (Adaptive p-value thresholding)

```
# BiocManager::install("adaptMT")

library(adaptMT)

# Load data
data(estrogen)
pvals <- as.numeric(estrogen$pvals)
x <- data.frame(x = as.numeric(estrogen$ord_high))

# Define the exponential family for AdaPT (Section 4)
dist <- beta_family()

# Subsample the data for convenience (rank 5000)
inds <- (x$x <= 5000)
pvals <- pvals[inds]
x <- x[inds,,drop = FALSE]

# Run adapt_glm
library("splines")
formulas <- paste0("ns(x, df = ", 6:10, ")")
res <- adapt_glm(x = x, pvals = pvals, pi_formulas = formulas,
                 mu_formulas = formulas, dist = dist, alphas = seq(0,0.1, 0.01), nfits = 10)
```

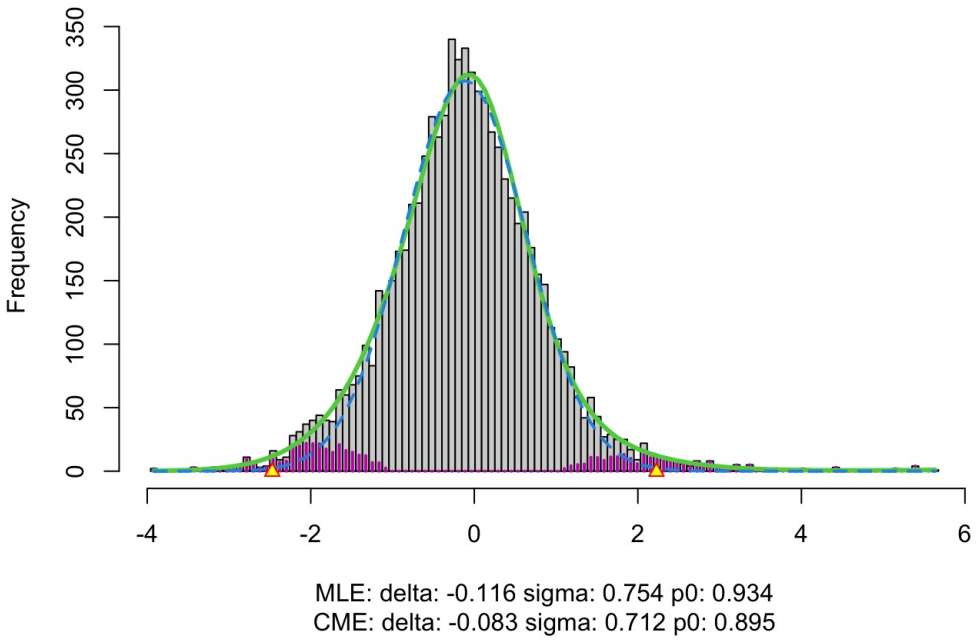
```
## Model selection starts!
## Shrink the set of candidate models if it is too time-consuming.
##
  |
  |                                                      |   0%
  |
  |==========                                            |  20%
  |
  |====================                                  |  40%
  |
  |==============================                        |  60%
  |
  |========================================              |  80%
  |
  |==================================================| 100%
## alpha = 0.1: FDPhat 0.0999, Number of Rej. 1482
## alpha = 0.09: FDPhat 0.0899, Number of Rej. 1390
## alpha = 0.08: FDPhat 0.0798, Number of Rej. 1303
## alpha = 0.07: FDPhat 0.0693, Number of Rej. 1155
## alpha = 0.06: FDPhat 0.06, Number of Rej. 984
## alpha = 0.05: FDPhat 0.0497, Number of Rej. 925
## alpha = 0.04: FDPhat 0.0393, Number of Rej. 839
## alpha = 0.03: FDPhat 0.0289, Number of Rej. 589
## alpha = 0.02: FDPhat 0.019, Number of Rej. 263
```

```
# Plot the threshold curve and the level curves of local FDR
plot_1d_thresh(res, x, pvals, alpha = 0.1, "P-Value Thresholds")
```



6. LFDR (local false discovery rate)

```
# install.packages("locfdr")

library(locfdr)

## HIV data example
data(hivdata)
w <- locfdr(hivdata)
```




```
length(w$fdr)
```

```
## [1] 7680
```

```
sum(w$fdr <= 0.1)
```

```
## [1] 101
```

- local false discovery rate (FDR) conditional on auxiliary covariate information

```
BiocManager::install("calm")
library(calm)

data(pso)
ind.nm <- is.na(pso$tval_mic)
x <- pso$len_gene[ind.nm]

# normalize covariate
x <- rank(x)/length(x) # covariate of length m
y <- pso$zval[ind.nm] # a vector of z-values of length m.

# assign names to the z-values helps to give names to the output variables
names(y) <- row.names(pso)[ind.nm]

fit.nm <- CLfdr(x=x, y=y)
```

```
## Overall pi0:  0.5971049 Tue Aug  2 17:17:26 2022
## Initial bandwidth:  0.088971 0.4551709
## Bandwidth:  0.1074827 0.1782413 Tue Aug  2 17:17:42 2022
```

```
fit.nm$fdr[1:5]
```

```
##      A3GALT2      AADACL3         AARD       AARSD1       ABCA10
## 0.6141581541 0.0009945293 0.7154088171 0.8991229342 0.0685392778
```

7. FDRreg (compared in IHW paper, swfdr paper)

```
# install.packages("devtools")

library(devtools)
install_github('jgscott/FDRreg')
```

위 install 실행이 되지 않음 (install.packages도 안됨)

avaliable in https://github.com/jgscott/FDRreg (https://github.com/jgscott/FDRreg)

```
# Simulated data
P = 2
N = 10000
betatrue = c(-3.5,rep(1/sqrt(P), P))
X = matrix(rnorm(N*P), N,P)
psi = crossprod(t(cbind(1,X)), betatrue)
wsuccess = 1/{1+exp(-psi)}

# Some theta's are signals, most are noise
gammatrue = rbinom(N,1,wsuccess)

# Density of signals
thetatrue = rnorm(N,3,0.5)
thetatrue[gammatrue==0] = 0
z = rnorm(N, thetatrue, 1)

# Fit the model
# fdr1 <- FDRreg(z, features = X, nulltype='theoretical') : error
```

8. ASH (Adaptive SHrinkage)

```
# install.packages("devtools")
library(devtools)

# install_github("stephens999/ashr")
library(ashr)

beta = c(rep(0,100),rnorm(100))
sebetahat = abs(rnorm(200,0,1)) # corresponding standard error
betahat = rnorm(200,beta,sebetahat) # effect size
beta.ash = ash(betahat, sebetahat)

head(beta.ash$result)
```

```
##         betahat  sebetahat NegativeProb PositiveProb      lfsr    svalue
## 1 -0.0005639362 0.01762901  0.002697537  0.002563285 0.9973025 0.7298234
## 2  0.2158330805 0.40735995  0.036737100  0.086475044 0.9135250 0.6916027
## 3  3.3081594456 2.32298625  0.081321874  0.244952221 0.7550478 0.4798684
## 4  0.0396651271 0.21583423  0.026374813  0.035379212 0.9646208 0.7179302
## 5  0.1258178941 0.79933377  0.083904668  0.106067416 0.8939326 0.6753562
## 6 -0.0389120300 0.70524677  0.089913780  0.082605256 0.9100862 0.6876447
##        lfdr    qvalue PosteriorMean PosteriorSD
## 1 0.9947392 0.6805605 -2.966768e-06 0.001279309
## 2 0.8767879 0.6366099  2.656243e-02 0.159439955
## 3 0.6737259 0.3919470  2.197512e-01 0.660791760
## 4 0.9382460 0.6623787  2.449481e-03 0.054478750
## 5 0.8100279 0.6077006  1.989083e-02 0.319970492
## 6 0.8274810 0.6158685 -6.043930e-03 0.278214949
```