

2018 SISG Bayesian Statistics for Genetics

R Notes: Multiple Testing

Jon Wakefield

Departments of Statistics and Biostatistics, University of
Washington

2018-07-27

Prostrate Cancer Example

These data come from Efron and Hastie book website:

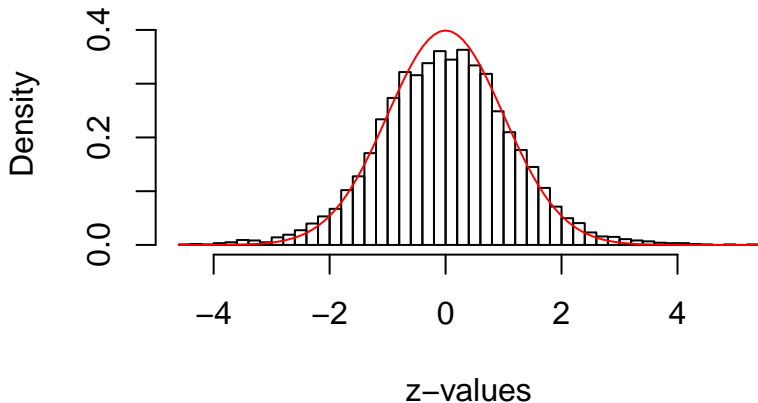
<https://web.stanford.edu/~hastie/CASI/data.html>

```
library(knitr)
library(qvalue)
library(locfdr)
library(ashr)
prostz <- read.table("prostz.txt", header = F, sep = ",")
prostz = unlist(prostz)
n = length(prostz)
n
## [1] 6033
```

Prostrate Cancer Example

Histogram of z-values, with an $N(0,1)$ added is lower in the middle, with perhaps an excess of mass in the tails.

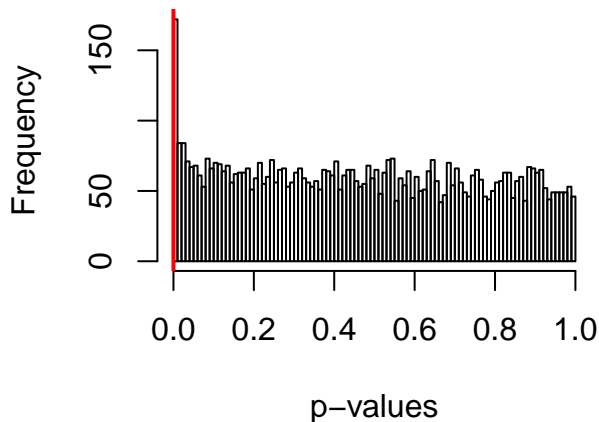
```
hist(prostz, nclass = 60, freq = F, ylim = c(0, 0.45),  
     main = "", xlab = "z-values")  
curve(dnorm(x), add = T, col = "red")
```



Prostrate Cancer: Histogram of p -values

```
##      V1332      V1610      V11720  
## 4.473521 5.291964 4.834436
```

Prostrate Cancer: Bonferroni



Hard to see the number of small p -values

Prostrate Cancer: Bonferroni

```
flag.index = which(pvalues < alpha.bon)
flag.index
## V1332 V1610 V11720
## 332 610 1720
# If we don't want to do 'by hand' use this
# command:
flag.index = which(p.adjust(pvalues, method = "bonferroni") <
0.05) #or use p.adjust function directly
```

Prostrate Cancer Example

Holm's procedure

```
pvalues = 2 * pnorm(abs(prostz), mean = 0, sd = 1,
  lower.tail = F)
# hist(pvalues, breaks = sqrt(n))
levels = alpha/(n + 1 - seq(1, n))
pvalues.order = order(pvalues)
flag.index = pvalues.order[which(pvalues[pvalues.order] <=
  levels)]
flag.index
## [1] 610 1720 332
# If we don't want to do 'by hand' use this
# command: flag.index = which(p.adjust(pvalues,
# method = 'holm') < 0.05) #or use p.adjust function
# directly
```

Prostrate Cancer Example: EFD

Control expected number of false discoveries (EFD)

```
EFD = 1
alpha.EFD = EFD/n
flag.index.1 = which(pvalues < alpha.EFD)
flag.index.1
## V1332 V1364 V1579 V1610 V1914 V11068 V11077 V11089 V11113 V11557
## 332 364 579 610 914 1068 1077 1089 1113 1557
## V11720 V13375 V13647 V13940 V13991 V14073 V14088 V14316 V14331 V14518
## 1720 3375 3647 3940 3991 4073 4088 4316 4331 4518
## V14546
## 4546
```


Prostrate Cancer Example: EFD

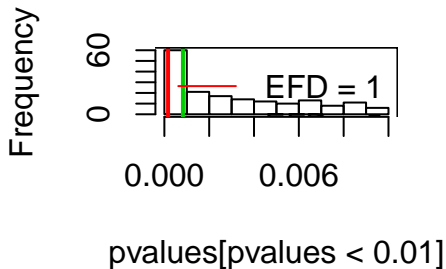
Control expected number of false discoveries (EFD)

```
EFD = 5
alpha.EFD = EFD/n
flag.index.5 = which(pvalues < alpha.EFD)
flag.index.5
##      V12      V111  V1332  V1364  V1579  V1610  V1694  V1698  V1702  V1721
##         2        11     332    364    579    610    694    698     702     721
##  V1735  V1739  V1905  V1914  V1921  V11068  V11077  V11089  V11113  V11130
##     735    739    905    914    921    1068    1077    1089    1113    1130
## V11314 V11346 V11557 V11588 V11589 V11720 V12370 V12856 V12897 V12945
##   1314   1346   1557   1588   1589   1720   2370   2856   2897   2945
## V13017 V13260 V13282 V13292 V13375 V13600 V13647 V13665 V13930 V13940
##   3017   3260   3282   3292   3375   3600   3647   3665   3930   3940
## V13991 V14000 V14073 V14088 V14104 V14154 V14316 V14331 V14396 V14518
##   3991   4000   4073   4088   4104   4154   4316   4331   4396   4518
## V14546 V14549 V14552 V14981
##   4546   4549   4552   4981
```

Prostrate Cancer Example: EFD

Control expected number of false discoveries (EFD)

```
hist(pvalues[pvalues < 0.01], main = "", xlim = c(0,
  0.01))
abline(v = 1/n, col = 2, lwd = 2)
abline(v = 5/n, col = 3, lwd = 2)
legend("topright", legend = c("EFD = 1", "EFD = 5"),
  col = c(2, 3), lty = c(1, 1))
```

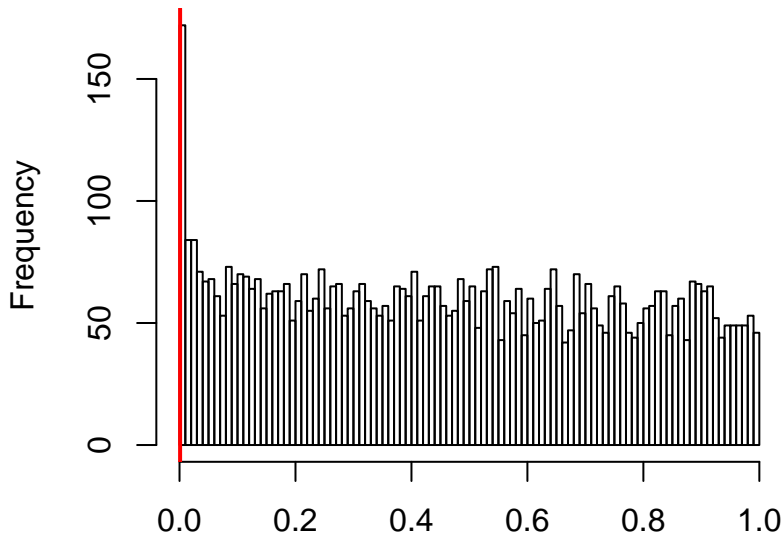


Prostrate Cancer Example: Benjamini Hochberg

##	V12	V111	V1332	V1364	V1377	V1579	V1610	V1694	V1698	V1702
##	2	11	332	364	377	579	610	694	698	702
##	V1721	V1735	V1739	V1805	V1905	V1914	V1921	V11068	V11077	V11089
##	721	735	739	805	905	914	921	1068	1077	1089
##	V11113	V11130	V11314	V11346	V11557	V11588	V11589	V11720	V12370	V12856
##	1113	1130	1314	1346	1557	1588	1589	1720	2370	2856
##	V12897	V12945	V13017	V13260	V13269	V13282	V13292	V13375	V13505	V13600
##	2897	2945	3017	3260	3269	3282	3292	3375	3505	3600
##	V13647	V13665	V13930	V13940	V13991	V14000	V14040	V14073	V14088	V14104
##	3647	3665	3930	3940	3991	4000	4040	4073	4088	4104
##	V14154	V14316	V14331	V14396	V14518	V14546	V14549	V14552	V14981	
##	4154	4316	4331	4396	4518	4546	4549	4552	4981	

Prostrate Cancer Example: Benjamini Hochberg

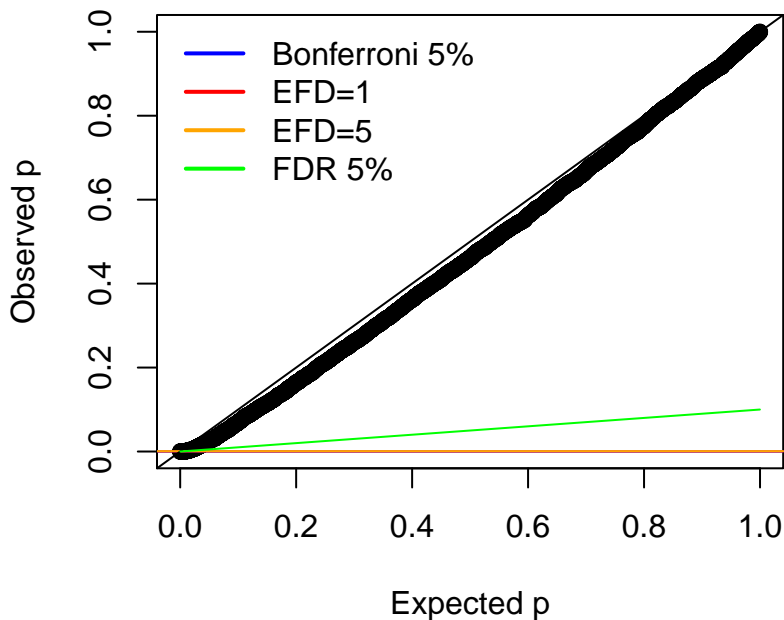
Histogram of pvalues



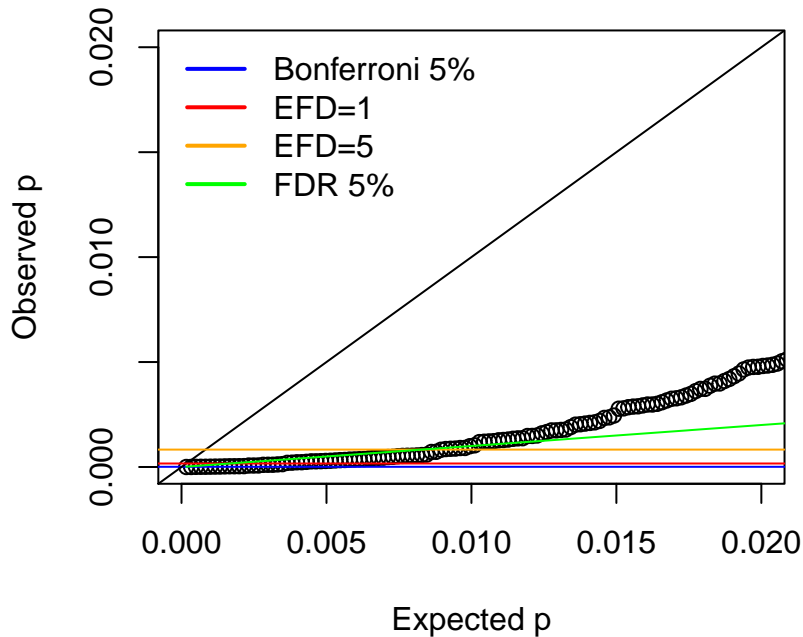
Multiple testing example

```
expp <- seq(1, n)/(n + 1)
p <- sort(pvalues)
plot(expp, p, xlim = c(0, 1), ylim = c(0, 1), ylab = "Observed p",
      xlab = "Expected p")
abline(h = 0.05/n, col = "blue")
abline(h = 1/n, col = "red")
abline(h = 5/n, col = "orange")
abline(a = 0, b = 1)
# points(expp, alpha*seq(1,m)/m, pch=25)
lines(expp, alpha * seq(1, n)/n, col = "green")
legend("topleft", legend = c("Bonferroni 5%", "EFD=1",
  "EFD=5", "FDR 5%"), col = c("blue", "red", "orange",
  "green"), bty = "n", lwd = 2)
```

Multiple testing example



Multiple testing example

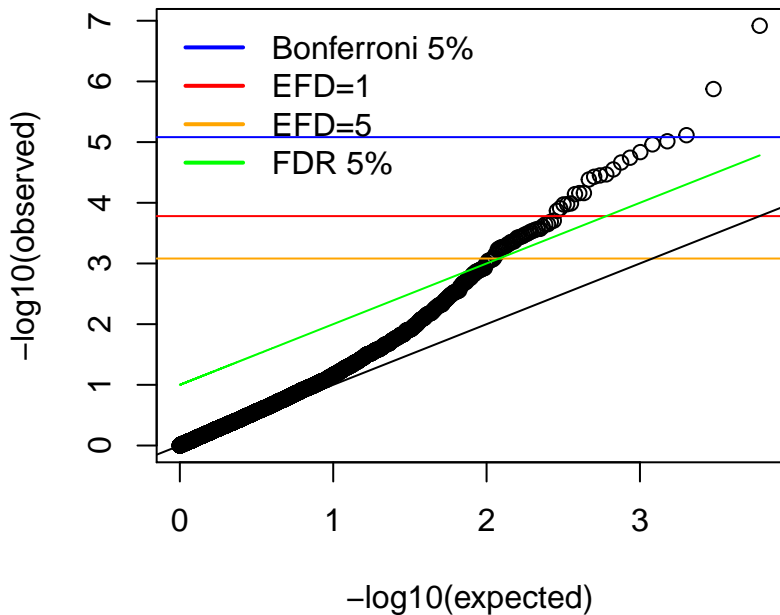


Multiple testing example

Evaluate the number passing each condition

```
sum(-log10(p) > -log10(0.05/n)) # Bonferroni
## [1] 3
sum(-log10(p) > -log10(1/n)) # EFD=1
## [1] 21
sum(-log10(p) > -log10(5/n)) # EFD=5
## [1] 54
li <- seq(1, n) * 0.05/n
cond <- rep(0, n)
for (i in 1:n) {
  if (p[i] < li[i]) {
    cond[i] <- 1
  }
}
sum(cond) # FDR
## [1] 21
```

Multiple testing example

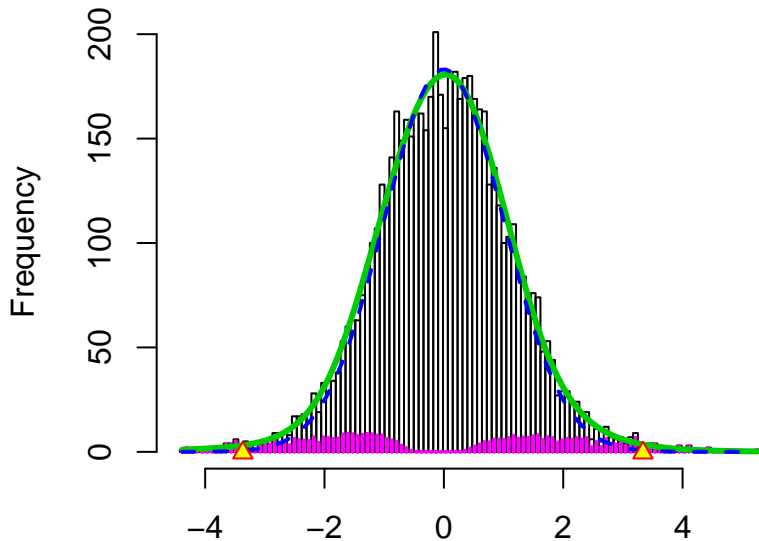


Prostrate Cancer Example: Storey's q-values

```
qobj = qvalue(p = pvalues, fdr.level = 0.1)
pi0.est = qobj$pi0
flag.index = which(qobj$significant)
flag.index
## [1] 2 11 292 298 332 364 377 452 579 610 637 694 698 702
## [15] 721 735 739 805 905 914 921 1068 1077 1089 1113 1130 1314 1346
## [29] 1491 1557 1588 1589 1647 1659 1720 1966 2370 2856 2897 2945 3017 3208
## [43] 3260 3269 3282 3292 3375 3505 3600 3647 3665 3930 3940 3991 4000 4040
## [57] 4073 4088 4104 4154 4316 4331 4396 4492 4496 4515 4518 4546 4549 4552
## [71] 4981
```

Prostrate Cancer Example: Efron's local fdr

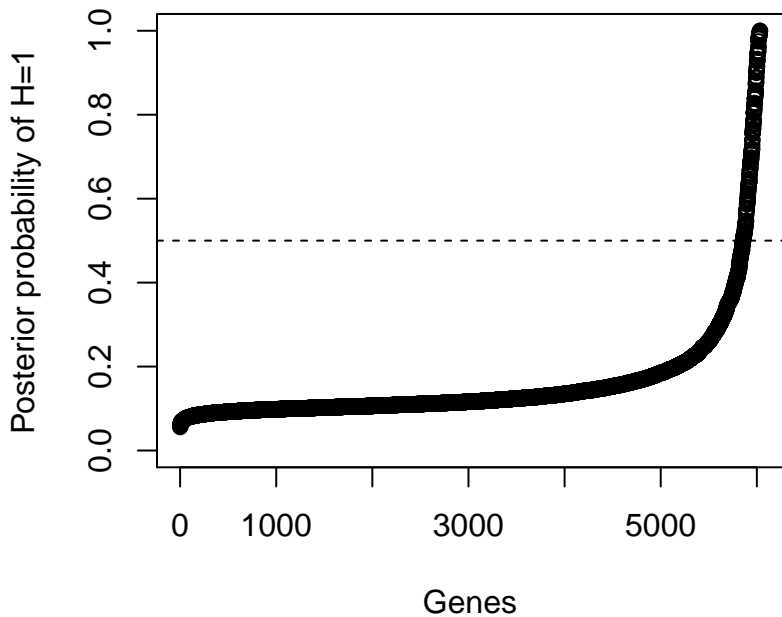
```
out = locfdr(prostz, nulltype = 0)
```



Prostrate Cancer Example:

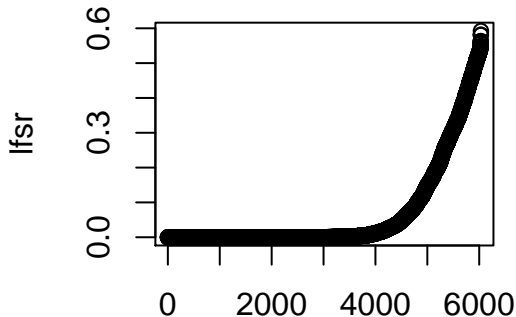
```
prostmat <- read.csv("prostmat.csv")
n = dim(prostmat)[1]
X.bar = apply(prostmat, 1, function(x) sum(x[1:50])/50)
Y.bar = apply(prostmat, 1, function(x) sum(x[51:102])/52)
Y = Y.bar - X.bar
s02 = apply(prostmat, 1, function(x) var(x[1:50]))
s12 = apply(prostmat, 1, function(x) var(x[51:102]))
sigma2 = s02/50 + s12/52
alpha = 0.05
```

Prostrate Cancer Example:



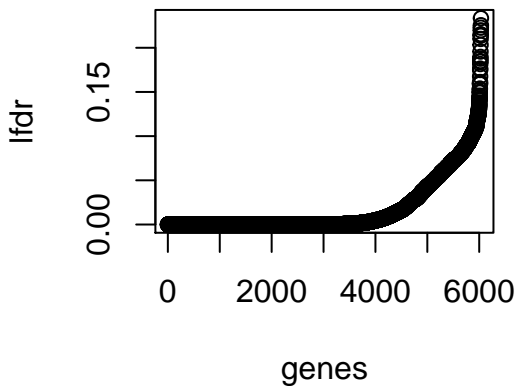
local false sign rate approach

```
out = ash.workhorse(betahat = Y, sebetahat = sigma2,  
  mixcompdist = c("normal"), method = c("fdr"))  
result = out$result #it is a large dataframe containing positive  
names(result)  
## [1] "betahat"          "sebetahat"         "NegativeProb"      "PositiveProb"  
## [5] "lfsr"              "svalue"            "lfdr"              "qvalue"  
## [9] "PosteriorMean"    "PosteriorSD"  
plot(sort(result[, 5]), xlab = "genes", ylab = "lfsr")
```



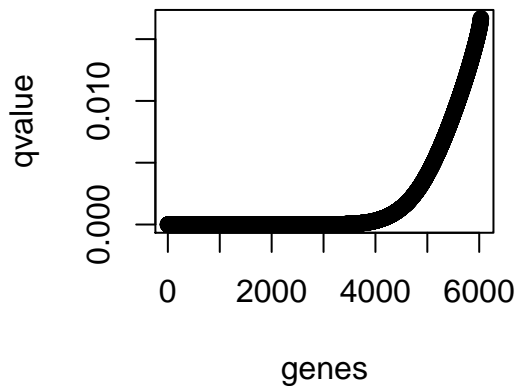
local false sign rate approach

```
plot(sort(result[, 7]), xlab = "genes", ylab = "lfdr")
```



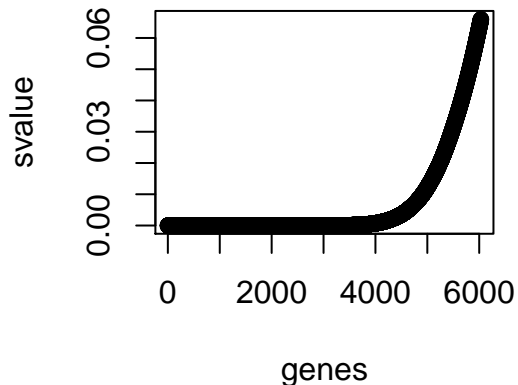
local false sign rate approach

```
plot(sort(result[, 8]), xlab = "genes", ylab = "qvalue")
```



local false sign rate approach

```
plot(sort(result[, 6]), xlab = "genes", ylab = "svalue")
```



local false sign rate approach

```
# plot the sorted local false sign rate.  
# lower the better  
flag.index = which(p > 0.5) #flag the genes with lfsr larger than  
flag.index  
## [1] 2 11 73 78 98 292 298 332 341 348 354  
## [15] 381 452 476 478 489 493 579 594 610 611 637  
## [29] 692 694 698 702 718 721 735 739 742 758 805  
## [43] 914 918 921 926 987 995 1003 1018 1019 1068 1073  
## [57] 1090 1097 1113 1117 1130 1185 1193 1223 1314 1329 1345  
## [71] 1491 1507 1557 1566 1572 1588 1589 1620 1628 1643 1647  
## [85] 1751 1810 1908 1918 1966 2370 2391 2764 2785 2811 2852  
## [99] 2872 2886 2897 2908 2912 2923 2945 2968 3017 3187 3200  
## [113] 3265 3269 3282 3292 3313 3343 3375 3505 3557 3567 3585  
## [127] 3665 3696 3712 3746 3761 3793 3835 3879 3917 3922 3930  
## [141] 4000 4013 4040 4073 4088 4104 4134 4154 4163 4282 4315  
## [155] 4378 4396 4428 4492 4496 4499 4500 4515 4518 4538 4539  
## [169] 4552 4554 4671 4981 4997 5159 5242 5287 5305
```