Impact of filtering of protein/peptide lists on FDR estimates

2024-11-26

Introduction

Some may assume that shortening the list of proteins would artificially increase the false discovery rate (FDR). However, the FDR, estimated using the Benjamini-Hochberg correction, depends primarily on the p-value distribution relative to the null hypothesis (H_0, a uniform distribution), not the absolute number of proteins tested. While methods controlling the Family-Wise Error Rate (FWER) are sensitive to the total number of tests, the FDR estimation is unaffected as long as the p-value distribution remains unchanged

The computer Experiment

Example 1, filtering without changing the distribution of p-values

We start by simulating a list of $m = 3000$ p-values. 90 datapoints comes from H0, 10 from H1 (fold change of 2). The group sizes are 4. We store these p-values in the array $pvals24$.

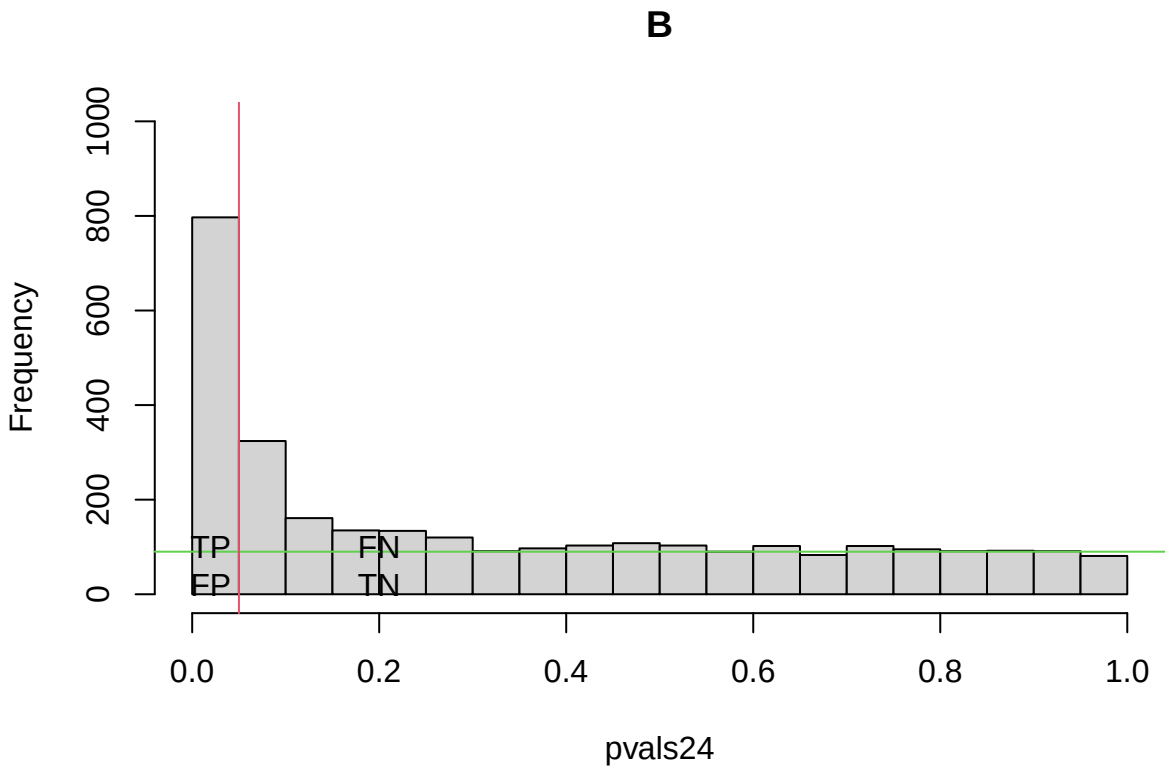
```
m <- 3000
simulate.p.values <- function(
    i, delta = 2, fraction = 0.1, ss = 4){
  control <- rnorm(ss,0,1)
  treatment <- rnorm(ss,0,1)
  if (runif(1) < fraction)
    treatment <- treatment + delta
  return(t.test(treatment,control)$p.value)
}
pvals24 <- sapply(1:m, simulate.p.values,
                  delta = 2, fraction = 0.4 )
```

We plot distribution of the p-values, and illustrate the idea of the FDR or False Discovery Proportion FDP. The

$$FDP = \frac{FP}{FP + TP}$$

We see that the required quantities FP TP FN and TN can be estimated from the histogram.

```
hist(pvals24, breaks = 20, ylim = c(0,m/3), main = "B")
abline(h = (m - m * 0.4)/20, col = 3)
abline(v = 0.05, col = 2)
text(x = c(0.02, 0.02, 0.2, 0.2),y = c(20,100,20,100), labels = c("FP","TP","TN","FN"))
```



We can also compute the FDR using the Benjamini-Hochberg method using the function *p.adjust*.

```
xx <- data.frame(p.vals = pvals24, FDR = p.adjust(pvals24, method = "BH"))
```

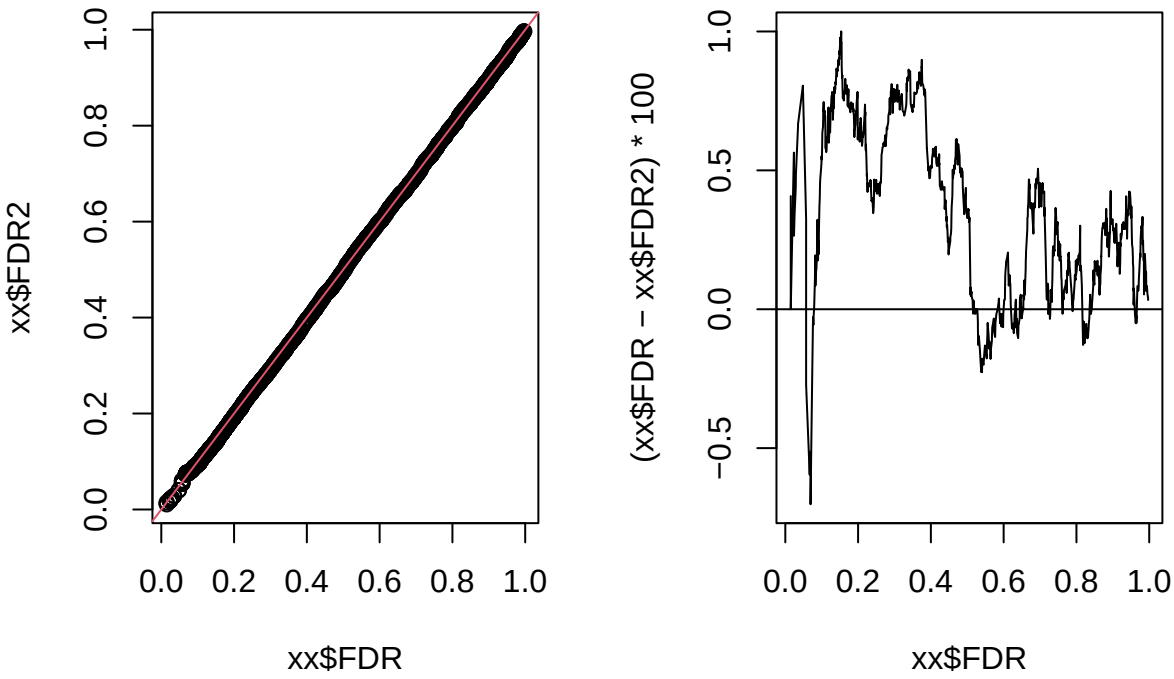
In proteomics experiments, approximately 1/3 of proteins are identified by only a single peptide. If we remove these proteins from the list, and keep only those quantified by 2 or more peptides, do we need to recompute the FDR?

We filter the our simulated p-values, by keeping only 2/3 of it. We randomly choose which rows of the data.frame are kept. We then compute the FDR from the remaining $p - values$.

```
xx <- xx[sample(1:m,floor(m*2/3)),]
xx$FDR2 <- p.adjust(xx$p.vals, method = "BH")
```

We compare the FDR estimates computed for all 3000 proteins, and from the 2000 proteins left after filtering. We can see that the differences in the p-value estimates are less than 1.

```
par(mfrow = c(1,2))
xx <- xx[order(xx$p.vals),]
plot(xx$FDR,xx$FDR2)
abline(c(0,1), col = 2)
plot(xx$FDR, (xx$FDR - xx$FDR2)*100, pch = ".", type = "l")
abline(h = 0)
```



We examine, how this affect the list of selected proteins if we filter for FDR at 5, 10 or 25 %.

```
ll <- data.frame(FDR = c("5" = sum(xx$FDR < 0.05),
"10" = sum(xx$FDR < 0.1), "25" = sum(xx$FDR < 0.25)),

FDR2 = c("5" = sum(xx$FDR2 < 0.05),"10" = sum(xx$FDR2 < 0.1),  "25" = sum(xx$FDR2 < 0.25)))
knitr::kable(ll)
```

	FDR	FDR2
5	16	16
10	235	265
25	723	735

We see that we end up with the same list of proteins, and it does not mattered if we used the FDR computed before, or after removing 1/3 of observations.

Example 2, filtering and changing the distribution of p-values

We now change the distribution of the p-values by filtering the list of proteins and then recompute the FDR.

First let's augment the p-values with 10 of p.values = 1 and compute the FDR.

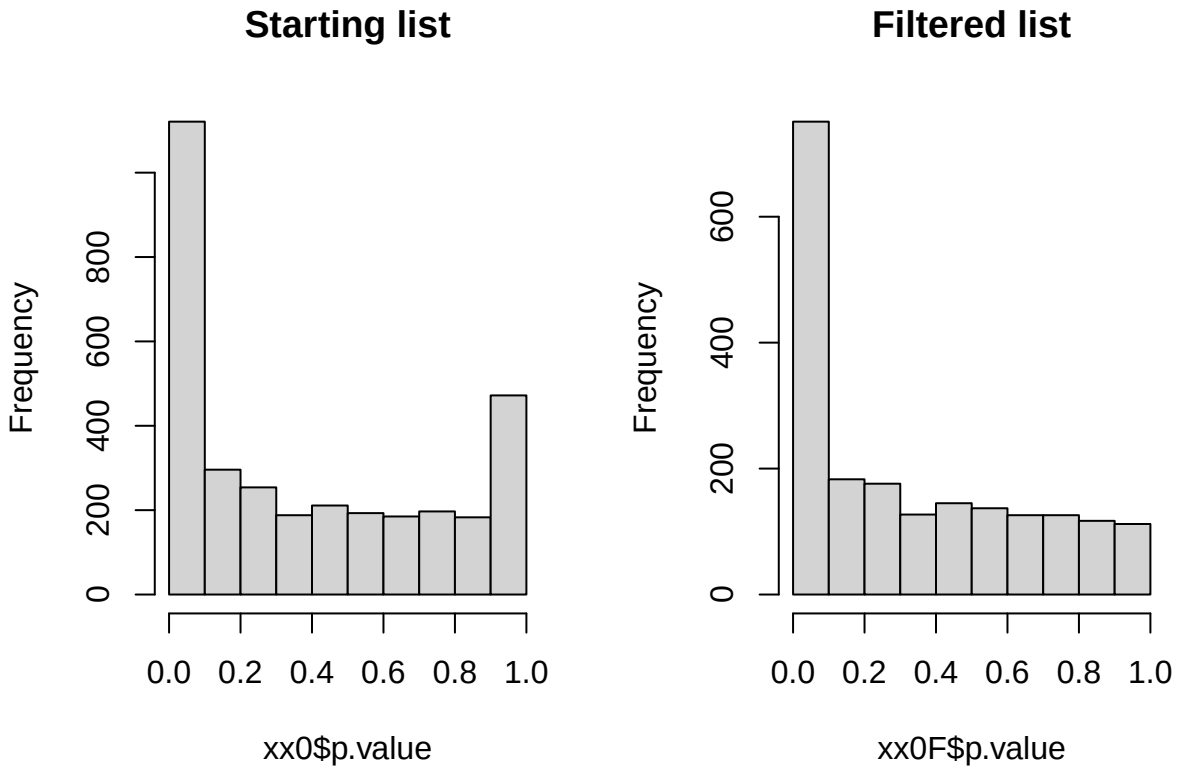
```
addP <- ceiling(m*0.1)
pvals240 <- c(pvals24, rep(1,addP))
xx0 <- data.frame(p.value = pvals240, FDR = p.adjust(pvals240, method = "BH"))
```

Next we remove those p-values equal 1, and in addition 33% of the other p-values, leaving 2000 p-values.

```
xx0F <- xx0[-(3001:(3000 + addP)),]
xx0F <- xx0F[sample(1:m,floor(m*2/3)),]
```

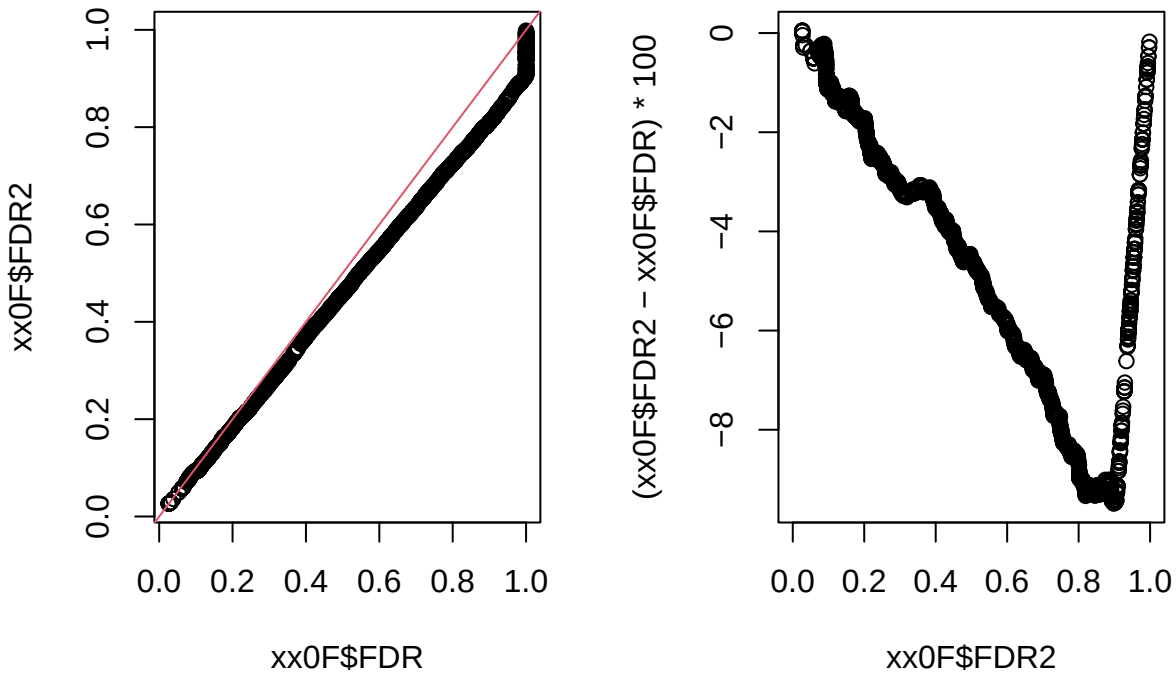
Clearly the p-value distributions does differ befoe and after filtering.

```
par(mfrow = c(1,2))
hist(xx0$p.value, main = "Starting list")
hist(xx0F$p.value, main = "Filtered list")
```



Now we do recompute the FDR from the truncated list of p-values, and compare the FDR estimates computed before with those computed after filtering the lists.

```
xx0F$FDR2 <- p.adjust(xx0F$p.value, method = "BH")
par(mfrow = c(1,2))
plot(xx0F$FDR, xx0F$FDR2)
abline(0,1,col = 2)
plot(xx0F$FDR2,(xx0F$FDR2 - xx0F$FDR)*100, xlim = c(0,1))
```



In this, case the difference of the FDR estimates, between the recomputed and the FDR estimates determined before filtering.

Again, we examine, how this affects the list of selected proteins if we filter for FDR at 5%, 10% or 25%.

```
ll <- data.frame(FDR = c("5" = sum(xx0F$FDR < 0.05),
"10" = sum(xx0F$FDR < 0.1), "25" = sum(xx0F$FDR < 0.25)),

FDR2 = c("5" = sum(xx0F$FDR2 < 0.05),"10" = sum(xx0F$FDR2 < 0.1),  "25" = sum(xx0F$FDR2 < 0.25)))
knitr::kable(ll)
```

	FDR	FDR2
5	11	13
10	137	223
25	671	713

The lists of proteins/peptides filtered by FDR at 5%, 10% and 2 have different lengths, depending if the FDR was computed before or after the experiment. This difference has nothing to do with the length of the list however, but with the difference in the distribution of p-values.