fdrcontrol

Dougals Wu

November 2, 2015

The **fdrcontrol** package tends replace *p.adjust(method='BH') or p.adjust(method='fdr')* for a faster filtering step.

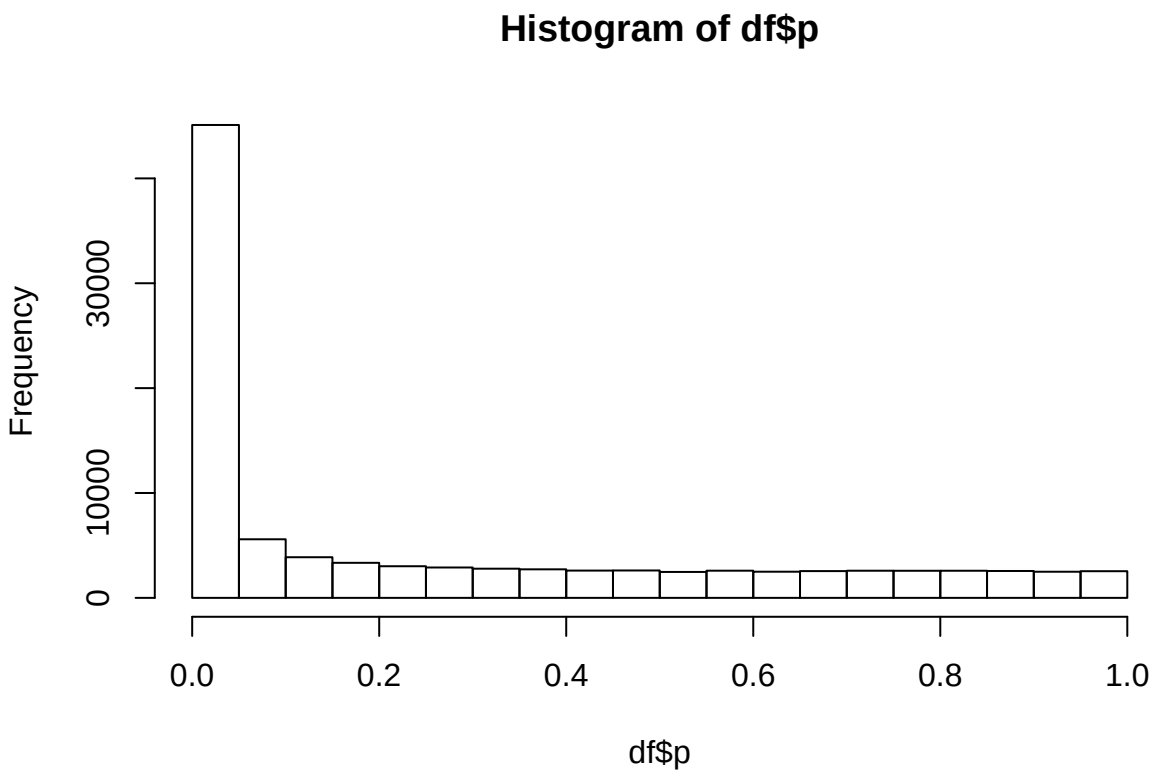
The *FDRcontrol* function takes in a vector of p-values and an alpha value (threshold) for controlling false discovery rate. The algorithm used was described in here.

The function return 1 when the datapoint is below the threshold and 0 if it is above the threshold.

```
set.seed(0)
library(fdrcontrol)
library(ggplot2)
library(dplyr)
library(tidyr)

alpha <- 0.01
no_of_test <- 100000

#simulate data
x <- rnorm(no_of_test, mean = c(rep(0, 25), rep(3, 25)))
p <- 2*pnorm(sort(-abs(x)))
df <- data_frame(p)
hist(df$p)
```



```
df %>%
    mutate(fdr = FDRcontrol(p,alpha))%>%  #usage of the function FDRcontrol
    mutate(padj = p.adjust(p,method='BH')) %>%
    arrange(p) %>%
    mutate(idx = 1:nrow(.)) %>%
    tbl_df -> df

df %>%
    filter(fdr==1,padj > alpha) #none of the data point disagree
```

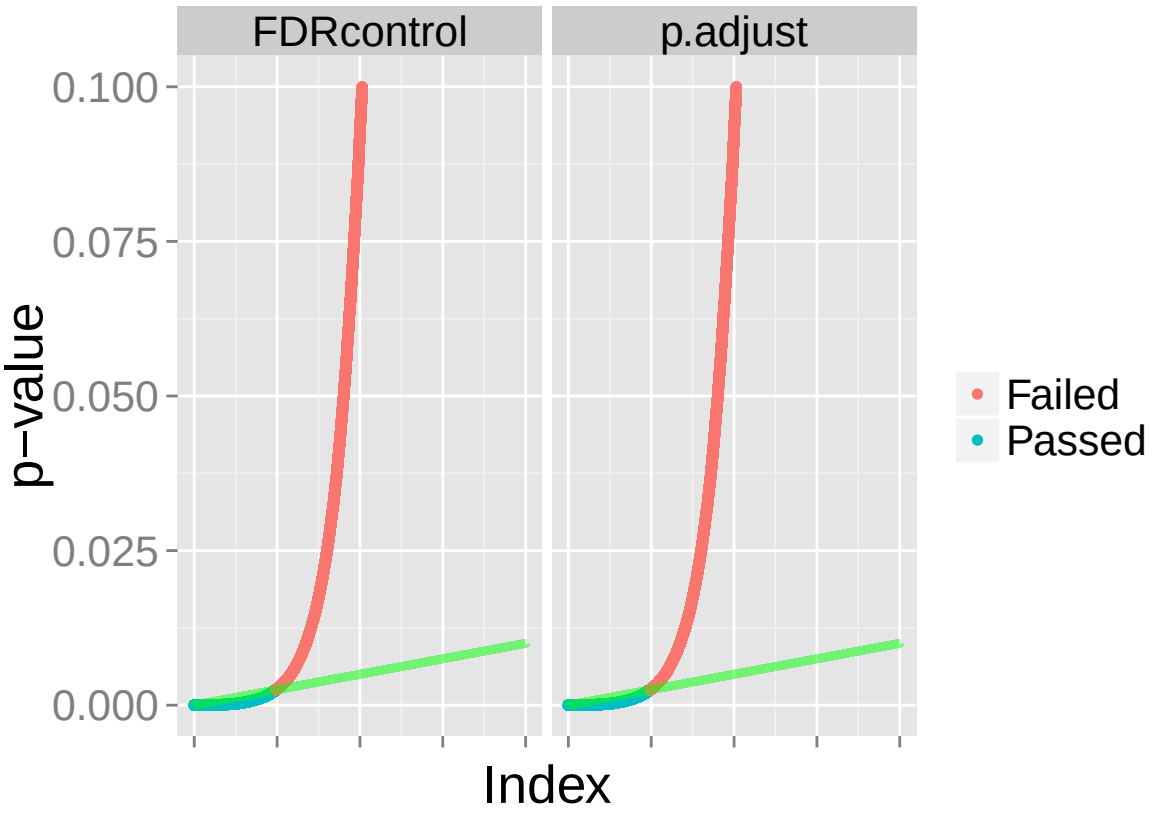
```
## Source: local data frame [0 x 4]
##
## Variables not shown: p (dbl), fdr (dbl), padj (dbl), idx (int)
```

A plot of p-values vs the ranking of the p-values is plotted below. The green line plotted the threshold at y-axis along the ranking of p-values, which follows:

$$y = \frac{\hat{i}}{m}\alpha$$

where $\hat{i}$ is the rank of the p-values, m is the no. of test that were included in the experiment and alpha is false discovery rate that we can tolerate.

```
df %>%
    mutate(FDRcontrol = ifelse(fdr==1,'Passed','Failed')) %>%
    mutate(p.adjust = ifelse(padj< alpha, 'Passed','Failed'))  %>%
    select(-fdr,-padj) %>%
    gather(Method,tag,-p,-idx) %>%
    ggplot() +
        geom_point(aes(y=p,x=idx,color=factor(tag))) +
        geom_line(aes(y = idx/no_of_test * alpha, x=idx),color='green',size = 1.5,alpha=0.5) +
        facet_grid(.~Method)+
        theme(text=element_text(size=20)) +
        theme(axis.text.x = element_blank()) +
        ylim (0,0.1) +
        labs(color=' ', x = 'Index', y = 'p-value')
```



To compare the speed of computation:

```
library(microbenchmark)
microbenchmark(df %>% mutate(FDR = FDRcontrol(p,0.01)),
               df %>% mutate(padj = p.adjust(p,method='fdr')))
```

```
## Unit: milliseconds
##                                                expr       min        lq
##              df %>% mutate(FDR = FDRcontrol(p, 0.01))  2.310848  3.424836
##  df %>% mutate(padj = p.adjust(p, method = "fdr")) 17.606883 21.970987
##       mean    median        uq       max neval cld
##   4.364873  3.834866  4.424162  15.13561   100  a
##  25.218457 24.535672 26.717045 101.75997   100   b
```