

# Additional plots for: Independent filtering increases power for detecting differentially expressed genes, Bourgon et al., PNAS (2010)

Richard Bourgon

*genefilter* version 1.50.0 (Last revision 2014-10-15 )

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Data preparation</b>	<b>1</b>
<b>3</b>	<b>Filtering volcano plot</b>	<b>2</b>
<b>4</b>	<b>Rejection count plots</b>	<b>3</b>
4.1	Across <i>p</i> -value cutoffs . . . . .	3
4.2	Across filtering fractions . . . . .	4

## 1 Introduction

---

This vignette illustrates use of some functions in the *genefilter* package that provide useful diagnostics for independent filtering [1]:

- `kappa_p` and `kappa_t`
- `filtered_p` and `filtered.R`
- `filter_volcano`
- `rejection_plot`

## 2 Data preparation

---

Load the ALL data set and the *genefilter* package:

```
library("genefilter")
library("ALL")
data("ALL")
```

Reduce to just two conditions, then take a small subset of arrays from these, with 3 arrays per condition:

```
bcell <- grep("^B", as.character(ALL$BT))
moltyp <- which(as.character(ALL$mol.biol) %in%
               c("NEG", "BCR/ABL"))
ALL_bcrneg <- ALL[, intersect(bcell, moltyp)]
ALL_bcrneg$mol.biol <- factor(ALL_bcrneg$mol.biol)
```

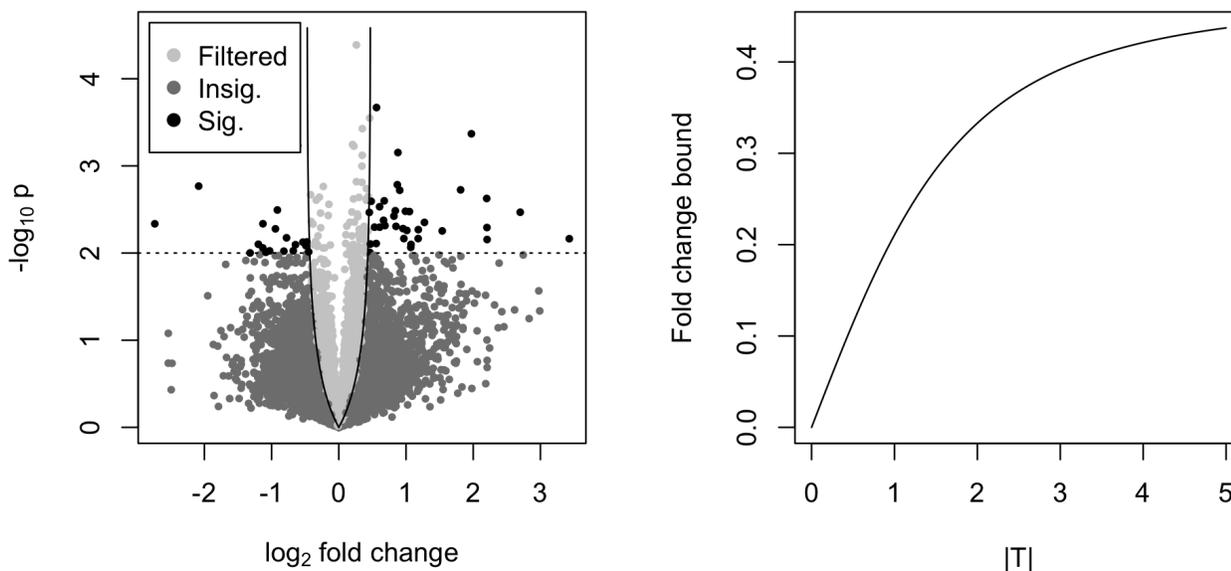


Figure 1: Left panel: plot produced by the `filter_volcano` function. Right panel: graph of the `kappa_t` function.

```
n1 <- n2 <- 3
set.seed(1969)
use <- unlist(tapply(1:ncol(ALL_bcrneg),
                    ALL_bcrneg$mol.biol, sample, n1))
subsample <- ALL_bcrneg[,use]
```

We now use functions from *genefilter* to compute overall standard deviation filter statistics as well as standard two-sample  $t$  and related statistics.

```
S <- rowSds( exprs( subsample ) )
temp <- rowttests( subsample, subsample$mol.biol )
d <- temp$dm
p <- temp$p.value
t <- temp$statistic
```

### 3 Filtering volcano plot

---

Filtering on overall standard deviation and then using a standard  $t$ -statistic induces a lower bound of fold change, albeit one which varies somewhat with the significance of the  $t$ -statistic. The `filter_volcano` function allows you to visualize this effect.

The output is shown in the left panel of Fig. 1.

The `kappa_p` and `kappa_t` functions, used to make the volcano plot, compute the fold change bound multiplier as a function of either a  $t$ -test  $p$ -value or the  $t$ -statistic itself. The actual induced bound on the fold change is  $\kappa$  times the filter's cutoff on the overall standard deviation. Note that fold change bounds for values of  $|T|$  which are close to 0 are not of practical interest because we will not reject the null hypothesis with test statistics in this range.

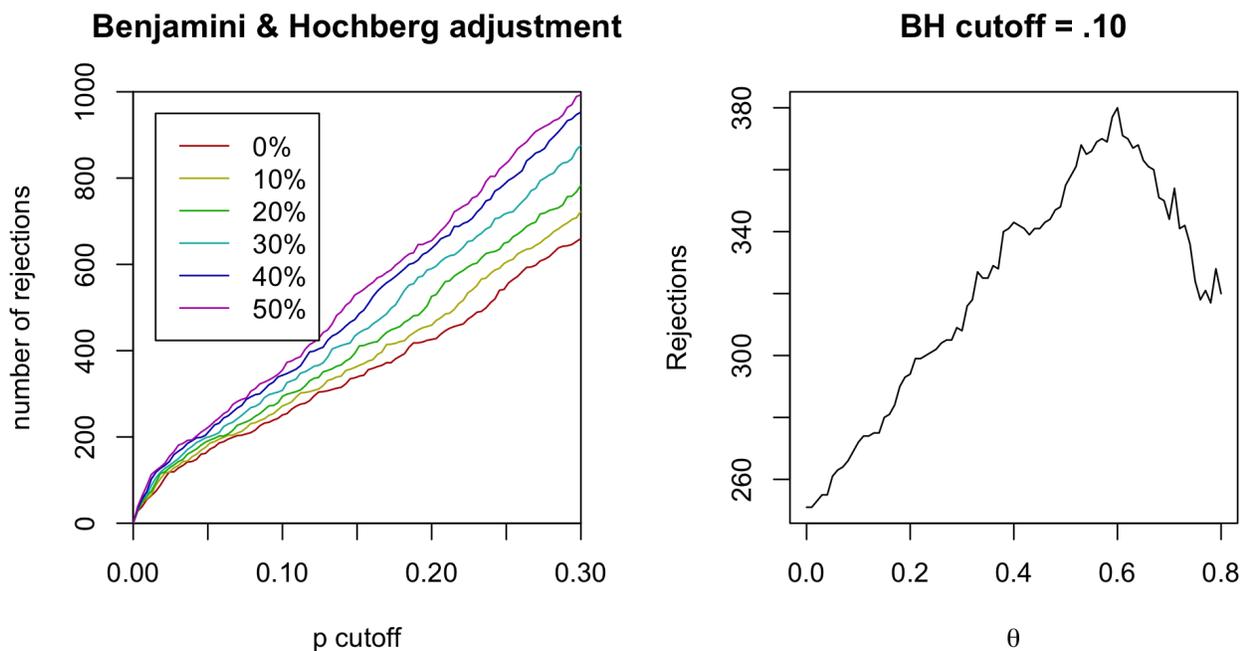


Figure 2: Left panel: plot produced by the `rejection_plot` function. Right panel: graph of  $\theta$ .

The plot is shown in the right panel of Fig. 1.

## 4 Rejection count plots

### 4.1 Across $p$ -value cutoffs

The `filtered_p` function permits easy simultaneous calculation of unadjusted or adjusted  $p$ -values over a range of filtering thresholds ( $\theta$ ). Here, we return to the full “BCR/ABL” versus “NEG” data set, and compute adjusted  $p$ -values using the method of Benjamini and Hochberg, for a range of different filter stringencies.

```
table(ALL_bcrneg$mol.biol)
```

```
##
## BCR/ABL    NEG
##      37     42
```

```
S2 <- rowVars(exprs(ALL_bcrneg))
p2 <- rowttests(ALL_bcrneg, "mol.biol")$p.value
theta <- seq(0, .5, .1)
p_bh <- filtered_p(S2, p2, theta, method="BH")
```

```
head(p_bh)
```

```
##           0%          10%          20%          30%          40%          50%
## [1,] 0.9185626 0.8943104 0.8624798 0.8278077          NA          NA
## [2,] 0.9585758 0.9460504 0.9304104 0.9059466 0.8874485 0.8709793
## [3,] 0.7022442          NA          NA          NA          NA          NA
```

```
## [4,] 0.9806216 0.9747555 0.9680574 0.9567131      NA      NA
## [5,] 0.9506087 0.9349386 0.9123998 0.8836386      NA      NA
## [6,] 0.6339004 0.5896890 0.5440851 0.4951371 0.4497915 0.4102711
```

The `rejection_plot` function takes sets of  $p$ -values corresponding to different filtering choices — in the columns of a matrix or in a list — and shows how rejection count ( $R$ ) relates to the choice of cutoff for the  $p$ -values. For these data, over a reasonable range of FDR cutoffs, increased filtering corresponds to increased rejections.

```
rejection_plot(p_bh, at="sample",
               xlim=c(0,.3), ylim=c(0,1000),
               main="Benjamini & Hochberg adjustment")
```

The plot is shown in the left panel of Fig. 2.

## 4.2 Across filtering fractions

If we select a fixed cutoff for the adjusted  $p$ -values, we can also look more closely at the relationship between the fraction of null hypotheses filtered and the total number of discoveries. The `filtered_R` function wraps `filtered_p` and just returns rejection counts. It requires a  $p$ -value cutoff.

```
theta <- seq(0, .80, .01)
R_BH <- filtered_R(alpha=.10, S2, p2, theta, method="BH")
```

```
head(R_BH)
```

```
## 0% 1% 2% 3% 4% 5%
## 251 251 253 255 255 261
```

Because overfiltering (or use of a filter which is inappropriate for the application domain) discards both false and true null hypotheses, very large values of  $\theta$  reduce power in this example:

```
plot(theta, R_BH, type="l",
      xlab=expression(theta), ylab="Rejections",
      main="BH cutoff = .10"
      )
```

The plot is shown in the right panel of Fig. 2.

## Session information

---

- R version 3.2.0 RC (2015-04-08 r68161), x86\_64-apple-darwin10.8.0
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, utils
- Other packages: ALL 1.9.1, Biobase 2.28.0, BiocGenerics 0.14.0, genefilter 1.50.0, knitr 1.9
- Loaded via a namespace (and not attached): annotate 1.46.0, AnnotationDbi 1.30.0, BiocStyle 1.6.0, codetools 0.2-11, DBI 0.3.1, digest 0.6.8, evaluate 0.6, formatR 1.1, GenomInfoDb 1.4.0, highr 0.4.1, IRanges 2.2.0, RSQLite 1.0.0, S4Vectors 0.6.0, splines 3.2.0, stats4 3.2.0, stringr 0.6.2, survival 2.38-1, tools 3.2.0, XML 3.98-1.1, xtable 1.7-4

## References

---

- [1] Richard Bourgon, Robert Gentleman and Wolfgang Huber. Independent filtering increases power for detecting differentially expressed genes.