Diagnostic plots for independent filtering

Richard Bourgon

25 October 2009

Contents

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:

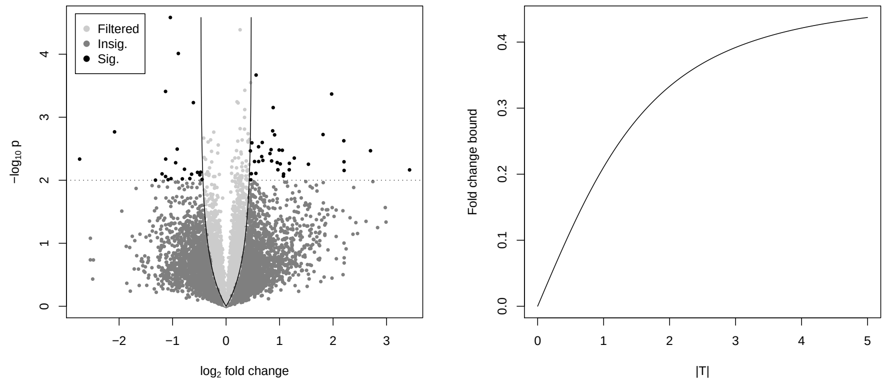

Figure 1: Left panel: plot produced by the `filter_volcano` function. Right panel: graph of the `kappa_t` function.

```
> 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)
> 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 devation filter statistics as well as standard two-sample t and releated 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.

```
> S_cutoff <- quantile(S, .50)
> filter_volcano(d, p, S, n1, n2, alpha=.01, S_cutoff)
```

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

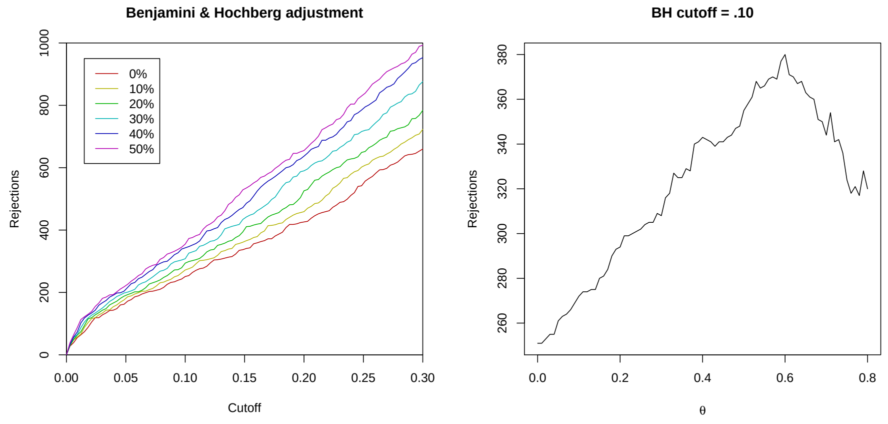

Figure 2: Left panel: plot produced by the `rejection_plot` function. Right panel: graph of `theta`.

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 κ 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.

```
> t <- seq(0, 5, length=100)
> plot(t, kappa_t(t, n1, n2) * S_cutoff,
+      xlab="|T|", ylab="Fold change bound", type="l")
```

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 simulataneous calculation of unadjusted or adjusted p-values over a range of filtering thresholds (θ). 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 θ 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 2.11.0 (2010-04-22), `x86_64-unknown-linux-gnu`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils

- Other packages: ALL 1.4.7, Biobase 2.8.0, class 7.3-2, genefilter 1.30.0
- Loaded via a namespace (and not attached): annotate 1.26.0, AnnotationDbi 1.10.0, DBI 0.2-5, RSQLite 0.8-4, splines 2.11.0, survival 2.35-8, xtable 1.5-6

References

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