Marginal independence of the test statistic of the filter statistic

Michael Love

October 14, 2014

1 Simulate null data from real RNA-Seq estimates

We constructed simulated datasets with samples divided into 2 groups, with no true difference between the means of the two groups. The means and dispersions of the Negative Binomial simulated data were drawn from the estimates from the Pickrell et al dataset.

```
library("DESeq2")
library("DESeq2paper")
makeSimScript <- system.file("script/makeSim.R", package = "DESeq2paper", mustWork = TRUE)
source(makeSimScript)
data("meanDispPairs")
```

We then run the differential expression pipeline, excluding genes with row sum of one read or less: due to the discreteness of the data, these genes lead to discrete p-value spikes.

```
set.seed(1)
n <- 20000

m <- 6
condition <- factor(rep(c("A", "B"), each = m/2))
x <- model.matrix(~condition)
beta <- rep(0, n)
mat <- makeSim(n, m, x, beta, meanDispPairs)$mat
dds1 <- DESeqDataSetFromMatrix(mat, DataFrame(condition), ~condition)
dds1 <- dds1[rowSums(counts(dds1)) > 1, ]
dds1 <- DESeq(dds1)

## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing

res1 <- results(dds1, independentFiltering = FALSE)

m <- 12
condition <- factor(rep(c("A", "B"), each = m/2))
x <- model.matrix(~condition)
beta <- rep(0, n)
mat <- makeSim(n, m, x, beta, meanDispPairs)$mat
dds2 <- DESeqDataSetFromMatrix(mat, DataFrame(condition), ~condition)
dds2 <- dds2[rowSums(counts(dds2)) > 1, ]
dds2 <- DESeq(dds2)
```

```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing

res2 <- results(dds2, independentFiltering = FALSE)
```

```
plotMarginalPvalueDensity <- function(df, cuts) {
    qs <- cut(df$mean, c(0, cuts, max(df$mean) + 1))
    plot(0, 0, type = "n", xlim = c(0, 1), ylim = c(0, 2), xlab = "p-value",
        ylab = "density")
    abline(h = 1)
    cols <- colorRampPalette(c("purple", "blue"))(nlevels(qs))
    for (i in seq_along(levels(qs))) {
        h <- hist(df$pvalue[qs == levels(qs)[i]], breaks = 0:16/16, plot = FALSE)
        points(h$mids, h$density, type = "o", col = cols[i], pch = i)
    }
    legend("bottomright", legend = levels(qs), pch = seq_along(levels(qs)),
        title = "bin by row mean", cex = 0.7, ncol = 2, col = cols)
}
```

```
line <- 0.5
adj <- -0.15
cex <- 1.5

par(mar = c(4.5, 4.5, 2, 1), mfrow = c(1, 2))
nq <- 7

df <- data.frame(mean = res1$baseMean, pvalue = res1$pvalue)
cuts <- c(10, round(quantile(df$mean[df$mean > 10], 1:(nq - 1)/nq)))
plotMarginalPvalueDensity(df, cuts)
mtext("A", side = 3, line = line, adj = adj, cex = cex)

df <- data.frame(mean = res2$baseMean, pvalue = res2$pvalue)
cuts <- c(10, round(quantile(df$mean[df$mean > 10], 1:(nq - 1)/nq)))
plotMarginalPvalueDensity(df, cuts)
mtext("B", side = 3, line = line, adj = adj, cex = cex)
```

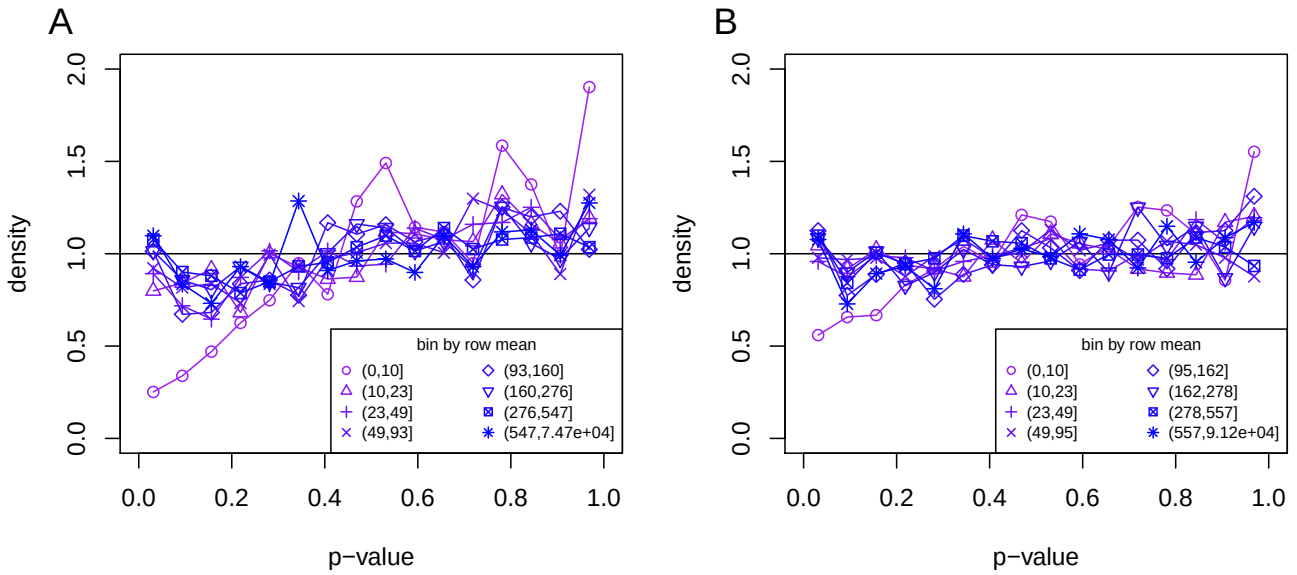

Figure 1: Marginal null histogram of the test statistic, p-values, conditioning on the filter statistic, the row mean of normalized counts across all samples, used for independent filtering. A simulated dataset was constructed with (A) 6 samples or (B) 12 samples. In either case the samples were equally divided into 2 groups with no true difference between the means of the two groups. The means and dispersions of the Negative Binomial simulated data were drawn from the estimates from the Pickrell et al dataset, and the standard DESeq2 pipeline was run. The histogram of p-values was estimated at 16 equally spaced intervals spanning [0,1]. The marginal distributions of the test statistic were generally uniform while conditioning on various quantiles of filter statistic. The row mean bin with the smallest mean of normalized counts (mean count 0-10) was depleted of small p-values. The black line indicates the expected frequency for a uniform distribution.

2 Session information

- R version 3.1.0 (2014-04-10), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, utils
- Other packages: BiocGenerics 0.10.0, DESeq2 1.4.0, DESeq2paper 1.3, GenomeInfoDb 1.0.0, GenomicRanges 1.16.0, IRanges 1.21.45, LSD 2.5, MASS 7.3-31, RColorBrewer 1.0-5, Rcpp 0.11.1, RcppArmadillo 0.4.200.0, colorRamps 2.3, ellipse 0.3-8, ggplot2 0.9.3.1, gridExtra 0.9.1, gtools 3.3.1, hexbin 1.27.0, knitr 1.5, schoolmath 0.4, xtable 1.7-3
- Loaded via a namespace (and not attached): AnnotationDbi 1.26.0, Biobase 2.24.0, DBI 0.2-7, RSQLite 0.11.4, XML 3.98-1.1, XVector 0.4.0, annotate 1.42.0, codetools 0.2-8, colorspace 1.2-4, dichromat 2.0-0, digest 0.6.4, evaluate 0.5.5, formatR 0.10, genefilter 1.46.0, geneplotter 1.42.0, gtable 0.1.2, highr 0.3, labeling 0.2, lattice 0.20-29, locfit 1.5-9.1, munsell 0.4.2, plyr 1.8.1, proto 0.3-10, reshape2 1.4, scales 0.2.3, splines 3.1.0, stats4 3.1.0, stringr 0.6.2, survival 2.37-7, tools 3.1.0