Count outlier detection using Cook's distance

Michael Love

October 14, 2014

1 Run DE analysis with and without outlier removal

The following vignette produces the Supplemental Figure of the effect of replacing outliers based on Cook's distance. First, we load the Bottomly *et al.* dataset, and subset the dataset to allow a 7 vs 7 sample comparison based on strain. Because we have 7 replicates per condition, the *DESeq* function automatically replaces outlier counts and refits the GLM for these genes. The argument controlling this behavior is `minReplicatesForReplace` which is set by default to 7.

```
library("DESeq2")

## Loading required package:  GenomicRanges
## Loading required package:  BiocGenerics
## Loading required package:  parallel
##
## Attaching package:  'BiocGenerics'
##
## The following objects are masked from 'package:parallel':
##
##    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##    clusterExport, clusterMap, parApply, parCapply, parLapply,
##    parLapplyLB, parRapply, parSapply, parSapplyLB
##
## The following object is masked from 'package:stats':
##
##    xtabs
##
## The following objects are masked from 'package:base':
##
##    Filter, Find, Map, Position, Reduce, anyDuplicated, append,
##    as.data.frame, as.vector, cbind, colnames, do.call,
##    duplicated, eval, evalq, get, intersect, is.unsorted, lapply,
##    mapply, match, mget, order, paste, pmax, pmax.int, pmin,
##    pmin.int, rank, rbind, rep.int, rownames, sapply, setdiff,
##    sort, table, tapply, union, unique, unlist
##
## Loading required package:  IRanges
## Loading required package:  GenomeInfoDb
## Loading required package:  Rcpp
## Loading required package:  RcppArmadillo

library("DESeq2paper")
```

```
data("bottomly_sumexp")
dds <- DESeqDataSetFromMatrix(assay(bottomly), DataFrame(colData(bottomly)),
    ~strain)
```

```r
dds <- dds[, c(8:11, 15:17, 12:14, 18:21)]
as.data.frame(colData(dds))

##           experiment sample.id num.tech.reps   strain experiment.number
## SRR099230  SRX033480 SRX033480             1 C57BL/6J                 6
## SRR099231  SRX033481 SRX033481             1 C57BL/6J                 6
## SRR099232  SRX033482 SRX033482             1 C57BL/6J                 6
## SRR099233  SRX033483 SRX033483             1 C57BL/6J                 6
## SRR099237  SRX033488 SRX033488             1 C57BL/6J                 7
## SRR099238  SRX033489 SRX033489             1 C57BL/6J                 7
## SRR099239  SRX033490 SRX033490             1 C57BL/6J                 7
## SRR099234  SRX033484 SRX033484             1   DBA/2J                 6
## SRR099235  SRX033485 SRX033485             1   DBA/2J                 6
## SRR099236  SRX033486 SRX033486             1   DBA/2J                 6
## SRR099240  SRX033491 SRX033491             1   DBA/2J                 7
## SRR099241  SRX033492 SRX033492             1   DBA/2J                 7
## SRR099242  SRX033493 SRX033493             1   DBA/2J                 7
## SRR099243  SRX033494 SRX033494             1   DBA/2J                 7
##           lane.number submission     study    sample       run
## SRR099230           1  SRA026846 SRP004777 SRS140391 SRR099230
## SRR099231           2  SRA026846 SRP004777 SRS140392 SRR099231
## SRR099232           3  SRA026846 SRP004777 SRS140393 SRR099232
## SRR099233           5  SRA026846 SRP004777 SRS140394 SRR099233
## SRR099237           1  SRA026846 SRP004777 SRS140398 SRR099237
## SRR099238           2  SRA026846 SRP004777 SRS140399 SRR099238
## SRR099239           3  SRA026846 SRP004777 SRS140400 SRR099239
## SRR099234           6  SRA026846 SRP004777 SRS140395 SRR099234
## SRR099235           7  SRA026846 SRP004777 SRS140396 SRR099235
## SRR099236           8  SRA026846 SRP004777 SRS140397 SRR099236
## SRR099240           5  SRA026846 SRP004777 SRS140401 SRR099240
## SRR099241           6  SRA026846 SRP004777 SRS140402 SRR099241
## SRR099242           7  SRA026846 SRP004777 SRS140403 SRR099242
## SRR099243           8  SRA026846 SRP004777 SRS140404 SRR099243

dds <- DESeq(dds)

## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## -- replacing outliers and refitting for 33 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds))
## estimating dispersions
## fitting model and testing
```

Here we run again without outlier replacement, in order to obtain the uncorrected fitted coefficients.

```r
ddsNoReplace <- DESeq(dds, minReplicatesForReplace = Inf)

## using pre-existing size factors
## estimating dispersions
## you had estimated dispersions, replacing these
## gene-wise dispersion estimates
## mean-dispersion relationship
```

```
## final dispersion estimates
## fitting model and testing
```

2 Select the gene with highest Cook's distance

Now we pick the gene which had one of the highest Cook's distance in the initial fit. The inital Cook's distances are available as a matrix in the `assays` slot of the *DESeqDataSet*.

```
names(assays(dds))

## [1] "counts"        "mu"            "cooks"         "replaceCounts"
## [5] "replaceCooks"

maxCooks <- apply(assays(dds)[["cooks"]], 1, max)
idx <- which(rownames(dds) == "ENSMUSG00000076609")
unname(counts(dds)[idx, ])

##  [1]  0  1  3  0  4  0  1 50  0  0  1  0  3  4
```

3 Plot

The following code produces the plot. Note that the original counts are accessible via the *counts* function, and the replacement counts are accessible via the `replaceCounts` slot of the assays of `dds`. We find the expected normalized values q_{ij} by accessing the model coefficients in the metadata columns of the object.

```
makeColors <- function(y = c(-1e+06, 1e+06)) {
    polygon(c(-1, -1, 7.5, 7.5), c(y, y[2:1]), col = rgb(0, 1, 0, 0.1), border = NA)
    polygon(c(7.5, 7.5, 50, 50), c(y, y[2:1]), col = rgb(0, 0, 1, 0.1), border = NA)
}
line <- 0.6
adj <- -0.5
cex <- 1

par(mfrow = c(1, 3), mar = c(4.3, 4.3, 3, 1))
out <- assays(dds)[["cooks"]][idx, ] > qf(0.99, 2, ncol(dds) - 2)
plot(counts(dds, normalized = TRUE)[idx, ], main = "With outlier", ylab = "normalized counts",
    xlab = "samples", pch = as.integer(colData(dds)$strain) + 1, ylim = c(0,
        max(counts(dds, normalized = TRUE)[idx, ])), col = ifelse(out, "red",
        "black"))
makeColors()
q0 <- 2^(mcols(ddsNoReplace)$Intercept[idx] + mcols(ddsNoReplace)$strainC57BL.6J[idx])
q1 <- 2^(mcols(ddsNoReplace)$Intercept[idx] + mcols(ddsNoReplace)$strainDBA.2J[idx])
segments(1, q0, 7, q0, lty = 3)
segments(8, q1, 14, q1, lty = 3)
mtext("A", side = 3, line = line, adj = adj, cex = cex)

plot(assays(dds)[["cooks"]][idx, ], main = "Cook's distances", ylab = "", xlab = "samples",
    log = "y", pch = as.integer(colData(dds)$strain) + 1, col = ifelse(out,
        "red", "black"))
makeColors(y = c(1e-05, 1e+05))
abline(h = qf(0.99, 2, ncol(dds) - 2))
mtext("B", side = 3, line = line, adj = adj, cex = cex)

plot(assays(dds)[["replaceCounts"]][idx, ]/sizeFactors(dds), main = "Outlier replaced",
```

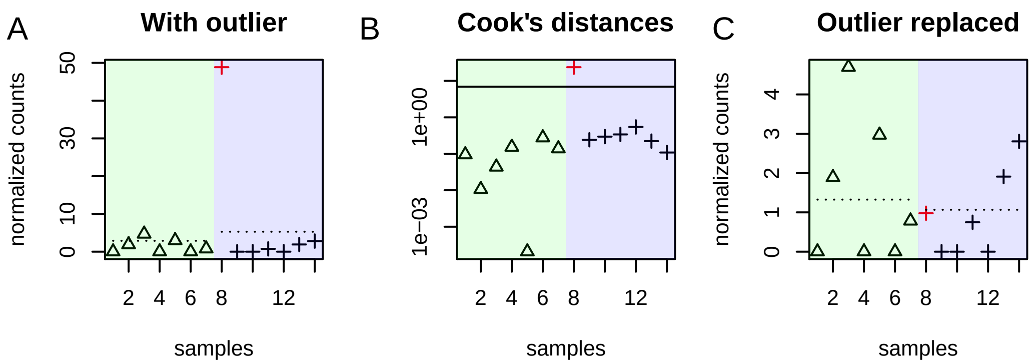

Figure 1: Demonstration using the Bottomly et al dataset, of detection of outlier counts (A) using Cook's distances (B), and refitting after outliers have been replaced by the trimmed median over all samples (C). The dotted line indicates the fitted mean on the common scale.

```
    ylab = "normalized counts", xlab = "samples", ylim = c(0, max(assays(dds)[["replaceCounts"]][idx,
        ]/sizeFactors(dds))), pch = as.integer(colData(dds)$strain) + 1, col = ifelse(out,
        "red", "black"))
makeColors()
q0 <- 2^(mcols(dds)$Intercept[idx] + mcols(dds)$strainC57BL.6J[idx])
q1 <- 2^(mcols(dds)$Intercept[idx] + mcols(dds)$strainDBA.2J[idx])
segments(1, q0, 7, q0, lty = 3)
segments(8, q1, 14, q1, lty = 3)
mtext("C", side = 3, line = line, adj = adj, cex = cex)
```

4 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, 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, Rcpp 0.11.1, RcppArmadillo 0.4.200.0
- Loaded via a namespace (and not attached): AnnotationDbi 1.26.0, Biobase 2.24.0, DBI 0.2-7, RColorBrewer 1.0-5, RSQLite 0.11.4, XML 3.98-1.1, XVector 0.4.0, annotate 1.42.0, codetools 0.2-8, digest 0.6.4, evaluate 0.5.5, formatR 0.10, genefilter 1.46.0, geneplotter 1.42.0, grid 3.1.0, highr 0.3, knitr 1.5, lattice 0.20-29, locfit 1.5-9.1, splines 3.1.0, stats4 3.1.0, stringr 0.6.2, survival 2.37-7, tools 3.1.0, xtable 1.7-3