Stability of shrunken logarithmic fold changes

Michael Love

October 14, 2014

1 DE analysis on an equally split dataset

The following code takes the Bottomly *et al.* dataset and splits into two equally sized groups, using the `createDataPartition` function of the *caret* package to balance the split across the experimental batches. Then the *DESeq* function is run in three different variations on the two splits: (1) a standard run including shrinkage of LFCs, (2) a run without shrinkage of LFCs (by setting the argument `betaPrior=FALSE`), and (3) a run without shrinkage of LFCs, but after adding a pseudocount of one read to each sample.

```
library("DESeq2")
library("DESeq2paper")
library("caret")

##
## Attaching package:  'caret'
##
## The following object is masked from 'package:survival':
##
##    cluster
```

```
data("bottomly_sumexp")
dds <- DESeqDataSetFromMatrix(assay(bottomly), colData(bottomly), ~experiment.number +
    strain)
dds <- dds[rowSums(counts(dds)) > 0, ]
cond1 <- which(colData(dds)$strain == "C57BL/6J")
cond2 <- which(colData(dds)$strain == "DBA/2J")

set.seed(1)
idx1 <- cond1[createDataPartition(colData(dds)$experiment.number[cond1], p = 0.5)[[1]]][1:5]
idx2 <- cond2[createDataPartition(colData(dds)$experiment.number[cond2], p = 0.5)[[1]]][1:5]
idx <- c(idx1, idx2)
table(1:ncol(dds) %in% idx, colData(dds)$strain)

##
##         C57BL/6J DBA/2J
##   FALSE        5      6
##   TRUE         5      5

table(1:ncol(dds) %in% idx, colData(dds)$experiment.number)

##
##         4 6 7
##   FALSE 3 3 5
##   TRUE  4 4 2

ddsList <- list(dds[, idx], dds[, -idx])
```

```
ddsListNoPrior <- list()
ddsListPC <- list()

for (i in 1:2) {
    ddsList[[i]] <- DESeq(ddsList[[i]])
    ddsListNoPrior[[i]] <- nbinomWaldTest(ddsList[[i]], betaPrior = FALSE)
    ddsListPC[[i]] <- ddsList[[i]]
    counts(ddsListPC[[i]]) <- counts(ddsListPC[[i]]) + 1L
    ddsListPC[[i]] <- DESeq(ddsListPC[[i]], betaPrior = FALSE)
}
```

```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## you had results columns, replacing these
## 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
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## you had results columns, replacing these
## 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 Calculate root mean squared error

Here we calculate four different root mean squared errors, comparing: (1) the unshrunken LFCs to each other, (2) the shrunken LFCs to each other, (3) the unshrunken LFCs of group I to the shrunken LFCs of group II, (4) the unhrunken LFCs of group I to the unshrunken LFCs of group II using a pseudocount. The first two errors are used in the first plot, while the first, third and fourth errors are used in the second plot.

```
rmseNoPriorNoPrior <- sqrt(mean((results(ddsListNoPrior[[1]])$log2FoldChange -
    results(ddsListNoPrior[[2]])$log2FoldChange)^2, na.rm = TRUE))
rmsePriorPrior <- sqrt(mean((results(ddsList[[1]])$log2FoldChange - results(ddsList[[2]])$log2FoldChange)^
    na.rm = TRUE))
rmseNoPriorPrior <- sqrt(mean((results(ddsListNoPrior[[1]])$log2FoldChange -
    results(ddsList[[2]])$log2FoldChange)^2, na.rm = TRUE))
rmseNoPriorPC <- sqrt(mean((results(ddsListNoPrior[[1]])$log2FoldChange - results(ddsListPC[[2]])$log2Fold
    na.rm = TRUE))
```

3 Plots

```
line <- 0.4
adj <- -0.3
cex <- 1.5

plotMM <- function(x, y, s, l = 4, ...) {
    idx <- abs(x) < 10 & abs(y) < 10
    x <- x[idx]
    y <- y[idx]
    s <- s[idx]
    plot(x, y, xlim = c(-l, l), ylim = c(-l, l), type = "n", ...)
    abline(0, 1, col = rgb(0, 0, 0, 1))
    abline(v = 0, h = 0, col = rgb(0, 0, 0, 1))
    cols <- ifelse(s, rgb(1, 0, 0, 0.5), rgb(0, 0, 0, 0.2))
    points(x, y, cex = 0.6, col = cols, pch = 20)
}

rmselg <- function(rmse) {
    legend("bottomright", legend = paste("RMSE:", round(rmse, 2)), bg = "white",
        adj = c(0.2, 0.5), cex = 0.9)
}


par(mfrow = c(1, 2), mar = c(4.5, 4.5, 2, 1))
plotMM(results(ddsListNoPrior[[1]])$log2FoldChange, results(ddsListNoPrior[[2]])$log2FoldChange,
    (results(ddsListNoPrior[[1]], ind = FALSE)$padj < 0.1 & results(ddsListNoPrior[[2]],
        ind = FALSE)$padj < 0.1), xlab = expression(MLE ~ log[2] ~ fold ~ change),
    ylab = expression(MLE ~ log[2] ~ fold ~ change))
rmselg(rmseNoPriorNoPrior)
mtext("A", side = 3, line = line, adj = adj, cex = cex)

plotMM(results(ddsList[[1]])$log2FoldChange, results(ddsList[[2]])$log2FoldChange,
    (results(ddsList[[1]], ind = FALSE)$padj < 0.1 & results(ddsList[[2]], ind = FALSE)$padj <
        0.1), xlab = expression(MAP ~ log[2] ~ fold ~ change), ylab = expression(MAP ~
        log[2] ~ fold ~ change))
rmselg(rmsePriorPrior)
mtext("B", side = 3, line = line, adj = adj, cex = cex)
```

```
catFn <- function(n, l) {
    top1 <- head(order(-abs(results(l[[1]])$log2FoldChange)), n)
    top2 <- head(order(-abs(results(l[[2]])$log2FoldChange)), n)
    length(intersect(top1, top2))
}

ns <- c(50, 100, 200, 1000, 2000)
priorCat <- sapply(ns, catFn, ddsList)/ns
noPriorCat <- sapply(ns, catFn, ddsListNoPrior)/ns
pcCat <- sapply(ns, catFn, ddsListPC)/ns

data.frame(n = ns, MAP = priorCat * ns, MLE = noPriorCat * ns, PC = pcCat *
    ns)
```

```
##      n MAP MLE  PC
## 1   50  40  18  38
```

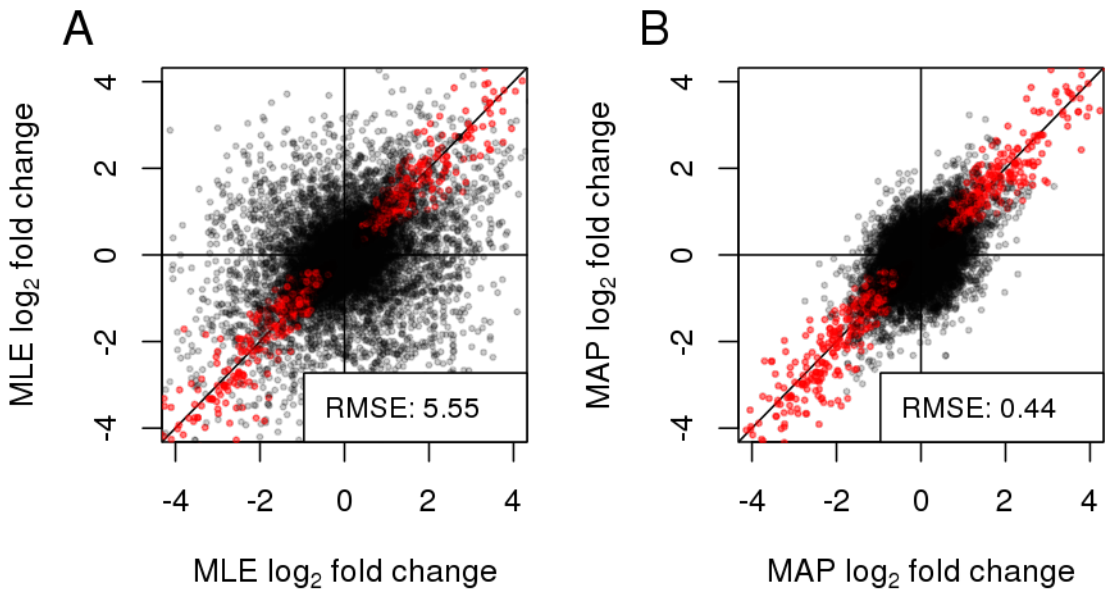

Figure 1: Comparing stability of logarithmic fold changes across two balanced, random subsets of the Bottomly et al dataset, (A) without a prior on logarithmic fold changes, (B) with a zero-centered Normal prior on logarithmic fold changes

```
## 2  100  80  21  81
## 3  200 156  29 156
## 4 1000 543 163 529
## 5 2000 979 545 963

plot(ns, priorCat, type = "b", col = "blue", ylim = c(0, 1), lwd = 2, log = "x",
    xlab = "size of list", ylab = "proportion in common")
points(ns, noPriorCat, type = "b", col = "purple", lwd = 2)
points(ns, pcCat, type = "b", col = "forestgreen", lwd = 2)
legend("topright", legend = c("MAP", "pseudocount", "MLE"), col = c("blue",
    "forestgreen", "purple"), lwd = 2, pch = 1)
```

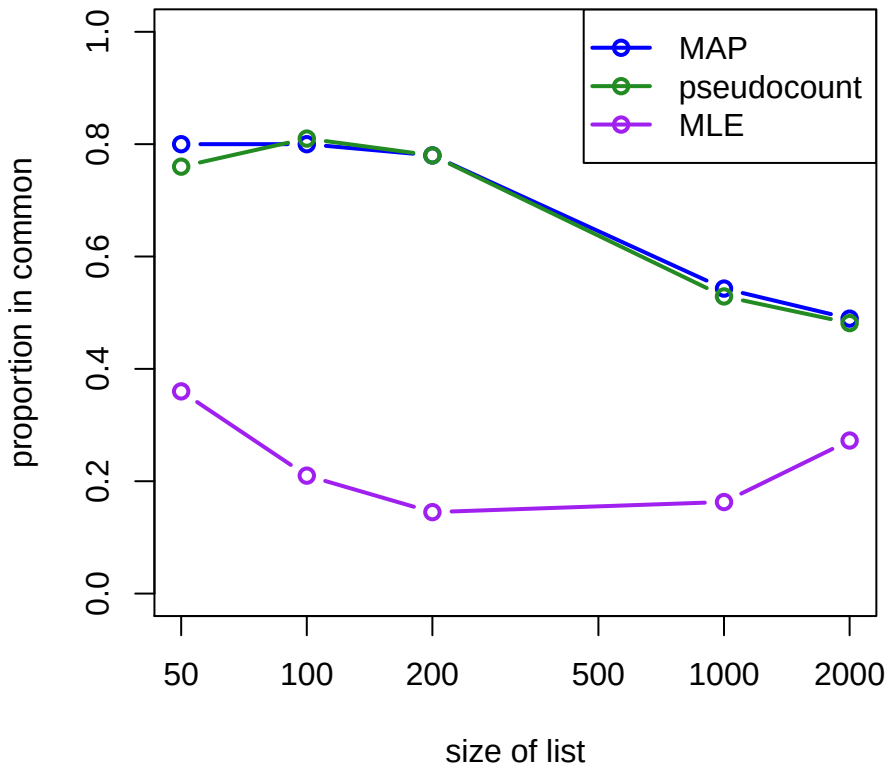

Figure 2: Concordance at the top (CAT) plot showing that the shrunken LFCs and and pseudocount-based estimators for logarithmic fold change provided more stable rankings compared to ranking based on the unhsrunken LFCs.

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, grid, methods, parallel, splines, stats, utils
- Other packages: Biobase 2.24.0, BiocGenerics 0.10.0, DESeq2 1.4.0, DESeq2paper 1.3, Formula 1.1-1, GenomeInfoDb 1.0.0, GenomicRanges 1.16.0, Hmisc 3.14-4, IRanges 1.21.45, LSD 2.5, MASS 7.3-31, RColorBrewer 1.0-5, Rcpp 0.11.1, RcppArmadillo 0.4.200.0, abind 1.4-0, caret 6.0-24, colorRamps 2.3, ellipse 0.3-8, ggplot2 0.9.3.1, gplots 2.13.0, gridExtra 0.9.1, gtools 3.3.1, hexbin 1.27.0, knitr 1.5, lattice 0.20-29, reshape 0.8.5, schoolmath 0.4, survival 2.37-7, vsn 3.32.0, xtable 1.7-3
- Loaded via a namespace (and not attached): AnnotationDbi 1.26.0, BiocInstaller 1.14.2, DBI 0.2-7, KernSmooth 2.23-12, RSQLite 0.11.4, XML 3.98-1.1, XVector 0.4.0, affy 1.42.2, affyio 1.32.0, annotate 1.42.0, bitops 1.0-6, caTools 1.16, car 2.0-20, cluster 1.15.2, codetools 0.2-8, colorspace 1.2-4, dichromat 2.0-0, digest 0.6.4, evaluate 0.5.5, foreach 1.4.2, formatR 0.10, gdata 2.13.3, genefilter 1.46.0, geneplotter 1.42.0, gtable 0.1.2, highr 0.3, iterators 1.0.7, labeling 0.2, latticeExtra 0.6-26, limma 3.20.1, locfit 1.5-9.1, munsell 0.4.2, nnet 7.3-8, plyr 1.8.1, preprocessCore 1.26.1, proto 0.3-10, reshape2 1.4, scales 0.2.3, stats4 3.1.0, stringr 0.6.2, tools 3.1.0, zlibbioc 1.10.0