Shrinkage of logarithmic fold changes

Michael Love

August 9, 2014

1 Comparing the posterior distribution for two genes

First, we run a DE analysis on the Bottomly *et al.* dataset, once with shrunken LFCs and once with unshrunken LFCs.

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

```
data("bottomly_sumexp")
dds <- DESeqDataSetFromMatrix(assay(bottomly), DataFrame(colData(bottomly)),
    ~strain)
ddsPrior <- DESeq(dds, minReplicatesForReplace = Inf)

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

ddsNoPrior <- nbinomWaldTest(ddsPrior, betaPrior = FALSE)

## you had results columns, replacing these
```

The following code, not run, was used to find two genes with similar average expression strength and unshrunken LFC, but with disparate dispersion estimates (and therefore Wald statistic in the unshrunken case).

```
with(results(ddsNoPrior), plot(baseMean, log2FoldChange, log = "x", ylim = c(0,
    4), xlim = c(10, 1000), col = ifelse(padj < 0.1, "red", "black"), cex = log(abs(stat))))
genes <- rownames(ddsNoPrior)[with(results(ddsNoPrior), identify(baseMean, log2FoldChange))]
```

The two genes chosen are:

```
genes <- c("ENSMUSG00000081504", "ENSMUSG00000092968")
```

We now extract the results from each run. Note that the second gene (with large dispersion) has a high adjusted p-value with or without the shrinkage of LFCs.

```
betaPriorVar <- attr(ddsPrior, "betaPriorVar")
resNoPrior <- results(ddsNoPrior, cooksCutoff = FALSE)
resNoPrior[genes, "padj"]

## [1] 4.123e-24 1.957e-01

resPrior <- results(ddsPrior, cooksCutoff = FALSE)
resPrior[genes, "padj"]

## [1] 3.647e-24 2.077e-01
```

2 Plot

```
cols <- c("green3", "mediumpurple3")
line <- 0.1
adj <- -0.4
cex <- 1.2

par(mar = c(4.5, 4.5, 1.3, 0.5), mfrow = c(2, 2))

# two MA plots

plotMA(resNoPrior, ylim = c(-4, 4), ylab = expression(MLE ~ log[2] ~ fold ~
    change), colNonSig = rgb(0, 0, 0, 0.3), colSig = rgb(1, 0, 0, 0.3), colLine = NULL)
abline(h = 0, col = "dodgerblue", lwd = 2)
with(mcols(ddsNoPrior[genes, ]), points(baseMean, strain_DBA.2J_vs_C57BL.6J,
    cex = 1.7, lwd = 2, col = cols))
legend("bottomright", "adj. p < .1", pch = 16, col = "red", cex = 0.9, bg = "white")
mtext("A", side = 3, line = line, adj = adj, cex = cex)
plotMA(resPrior, ylim = c(-4, 4), ylab = expression(MAP ~ log[2] ~ fold ~ change),
    colNonSig = rgb(0, 0, 0, 0.3), colSig = rgb(1, 0, 0, 0.3), colLine = NULL)
abline(h = 0, col = "dodgerblue", lwd = 2)
with(mcols(ddsPrior[genes, ]), points(baseMean, strainDBA.2J - strainC57BL.6J,
    cex = 1.7, lwd = 2, col = cols))
legend("bottomright", "adj. p < .1", pch = 16, col = "red", cex = 0.9, bg = "white")
mtext("B", side = 3, line = line, adj = adj, cex = cex)

# data plot

k1 <- counts(ddsNoPrior)[genes[1], ]
k2 <- counts(ddsNoPrior)[genes[2], ]
mu1 <- assays(ddsNoPrior)[["mu"]][genes[1], ]
mu2 <- assays(ddsNoPrior)[["mu"]][genes[2], ]
cond <- as.numeric(colData(dds)$strain) - 1
sf <- sizeFactors(ddsNoPrior)

plot(c(cond, cond + 2) + runif(2 * 21, -0.1, 0.1), c(k1/sf, k2/sf), xaxt = "n",
    xlab = "", ylab = "normalized counts", col = rep(cols, each = 21), cex = 0.5)
abline(v = 1.5)
axis(1, at = 0:3, label = rep(levels(colData(ddsNoPrior)$strain), times = 2),
    las = 2, cex.axis = 0.8)
mtext("C", side = 3, line = line, adj = adj, cex = cex)

# prior/likelihood/posterior curves plot

betas <- seq(from = -1, to = 3, length = 500)
disps <- dispersions(ddsNoPrior[genes, ])
a1 <- disps[1]
a2 <- disps[2]

# the prior is on the expanded fold changes (effects for both groups) so to
# translate to a univariate case, multiply the prior standard deviation by
# sqrt(2) why?  sigma_expanded^2 = (.5 beta)^2 + (.5 beta)^2 = 0.5 beta^2
# sigma_standard^2 = (.5 beta + .5 beta)^2 = beta^2
priorSigma <- sqrt(betaPriorVar[2]) * sqrt(2)

likelihood <- function(k, alpha, intercept) {
```

```
    z <- sapply(betas, function(b) {
        prod(dnbinom(k, mu = sf * 2^(intercept + b * cond), size = 1/alpha))
    })
    z/sum(z * diff(betas[1:2]))
}
posterior <- function(k, alpha, intercept) {
    z <- likelihood(k, alpha, intercept) * dnorm(betas, 0, priorSigma)
    z/sum(z * diff(betas[1:2]))
}
points2 <- function(x, y, col = "black", lty) {
    points(x, y, col = col, lty = lty, type = "l")
    points(x[which.max(y)], max(y), col = col, pch = 20)
}

ints <- mcols(ddsNoPrior[genes, ])$Intercept
intsPrior <- mcols(ddsPrior[genes, ])$Intercept + mcols(ddsPrior[genes, ])$strainC57BL.6J

# estimate the LFC using DESeq and test equality
deseqNoPrior <- results(ddsNoPrior[genes, ])$log2FoldChange
deseqPrior <- results(ddsPrior[genes, ])$log2FoldChange
mleFromPlot <- c(betas[which.max(likelihood(k1, a1, ints[1]))], betas[which.max(likelihood(k2,
    a2, ints[2]))])
mapFromPlot <- c(betas[which.max(posterior(k1, a1, intsPrior[1]))], betas[which.max(posterior(k2,
    a2, intsPrior[2]))])
stopifnot(all(abs(deseqNoPrior - mleFromPlot) < 0.01))
stopifnot(all(abs(deseqPrior - mapFromPlot) < 0.01))

# also want SE after prior for plotting
deseqPriorSE <- results(ddsPrior[genes, ])$lfcSE

plot(betas, likelihood(k1, 0.5, 4), type = "n", ylim = c(0, 4.1), xlab = expression(log[2] ~
    fold ~ change), ylab = "density")
lines(betas, dnorm(betas, 0, priorSigma), col = "black")
points2(betas, likelihood(k1, a1, ints[1]), lty = 1, col = cols[1])
points2(betas, likelihood(k2, a2, ints[2]), lty = 1, col = cols[2])
points2(betas, posterior(k1, a1, intsPrior[1]), lty = 2, col = cols[1])
points2(betas, posterior(k2, a2, intsPrior[2]), lty = 2, col = cols[2])
hghts <- c(4, 1.2)
points(deseqPrior, hghts, col = cols, pch = 16)
arrows(deseqPrior, hghts, deseqPrior + deseqPriorSE, hghts, col = cols, length = 0.05,
    angle = 90)
arrows(deseqPrior, hghts, deseqPrior - deseqPriorSE, hghts, col = cols, length = 0.05,
    angle = 90)
mtext("D", side = 3, line = line, adj = adj, cex = cex)
```

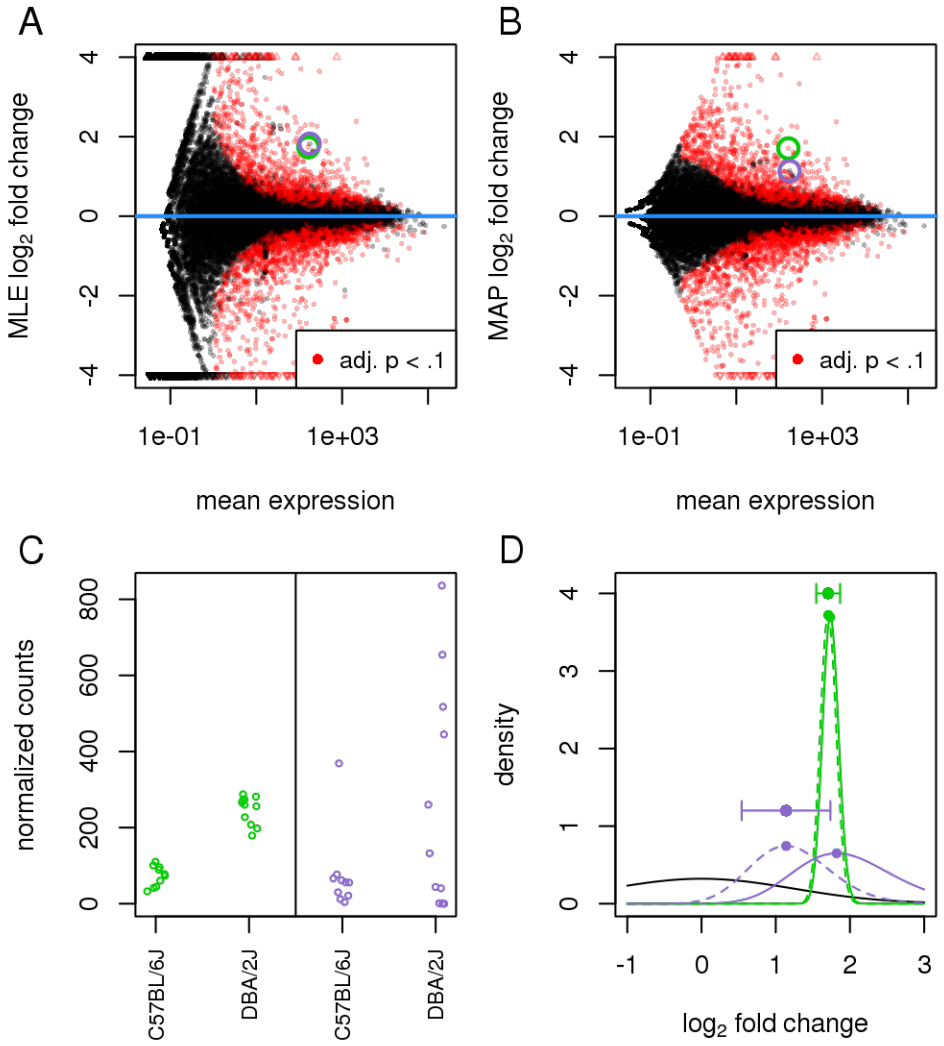

Figure 1: Comparison of logarithmic fold changes without and with zero-centered prior, using the Bottomly et al dataset.

3 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 1.2, 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, gtools 3.3.1, 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, 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