Shrinkage of dispersion estimates

Michael Love

October 14, 2014

1 Run DE analysis and select a subset of genes

Here examine dispersion estimates from the Bottomly et al. dataset, and the Pickrell et al. dataset. For the Bottomly dataset, we estimate dispersions for a 3 vs 3 comparison of the two strains. For the Pickrell dataset, we estimate dispersions for 5 samples with a model specifying only an intercept. These two comparisons have the same residual degrees of freedom: $df = 6 - 2$ and $df = 5 - 1$.

For plotting, we pick 40 genes equally spaced along average expression strength, and 5 dispersion outliers.

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

```
data("pickrell_sumexp")
idx <- 1:5
ddsP <- DESeqDataSetFromMatrix(assay(pickrell)[, idx], DataFrame(colData(pickrell)[idx,
    ]), ~1)
ddsP <- estimateSizeFactors(ddsP)
ddsP <- estimateDispersions(ddsP)

## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates

data("bottomly_sumexp")
idx <- c(1:3, 5:7)
ddsB <- DESeqDataSetFromMatrix(assay(bottomly)[, idx], DataFrame(colData(bottomly)[idx,
    ]), ~strain)
ddsB <- estimateSizeFactors(ddsB)
ddsB <- estimateDispersions(ddsB)

## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
```

```
plotDispShrink <- function(dds) {
    # pick 40 equally spaced genes along the base mean
    bins <- 10^seq(from = 0, to = 5, length = 20)
    pickone <- function(x) {
        if (sum(x) == 0)
            return(NULL)
        if (sum(x) == 1)
            return(which(x))
        sample(which(x), 1)
    }
    up <- sapply(seq_along(bins[-1]), function(i) pickone(mcols(dds)$dispGeneEst >
```

```
        1e-04 & !mcols(dds)$dispOutlier & mcols(dds)$dispGeneEst > mcols(dds)$dispFit &
        mcols(dds)$baseMean > bins[i] & mcols(dds)$baseMean < bins[i + 1]))
    down <- sapply(seq_along(bins[-1]), function(i) pickone(mcols(dds)$dispGeneEst >
        1e-04 & !mcols(dds)$dispOutlier & mcols(dds)$dispGeneEst < mcols(dds)$dispFit &
        mcols(dds)$baseMean > bins[i] & mcols(dds)$baseMean < bins[i + 1]))
    # pick 5 outliers
    bins <- 10^seq(from = 1, to = 4, length = 6)
    outliers <- do.call(c, lapply(seq_along(bins[-1]), function(i) pickone(mcols(dds)$dispGeneEst/mcols(dds
        2 & mcols(dds)$dispOutlier & mcols(dds)$baseMean > bins[i] & mcols(dds)$baseMean <
        bins[i + 1])))
    s <- c(up, down, outliers)
    s <- s[!is.na(s)]
    with(mcols(dds[s, ]), plot(baseMean, dispGeneEst, log = "xy", pch = 16,
        xlab = "mean of normalized counts", ylab = "dispersion estimate", yaxt = "n",
        ylim = c(0.001, 100)))
    axis(2, at = 10^(-3:2), label = 10^(-3:2))
    xs <- 10^(-20:50/10)
    lines(xs, dispersionFunction(dds)(xs), col = "red", lwd = 2)
    with(mcols(dds[s, ][!mcols(dds[s, ])$dispOutlier, ]), arrows(baseMean, dispGeneEst,
        baseMean, dispersion, length = 0.075, col = "dodgerblue", lwd = 2))
    with(mcols(dds[s, ][mcols(dds[s, ])$dispOutlier, ]), segments(baseMean,
        dispGeneEst, baseMean, dispMAP, col = "dodgerblue", lwd = 2, lty = 3))
    with(mcols(dds[s, ][mcols(dds[s, ])$dispOutlier, ]), points(baseMean, dispersion,
        cex = 2, col = "dodgerblue", lwd = 2))
    legend("topright", c("MLE", "prior mean", "MAP"), pch = 20, col = c("black",
        "red", "dodgerblue"), bg = "white")
}
```

2 Plot

The following code produces the plot, breaking out the different values based on whether the gene-wise estimate, trend fit, or the final MAP estimate is being plotted. Additionally, the dispersion outliers are plotted separately.

```
line <- -0.1
adj <- -0.3
cex <- 1.5
par(mfrow = c(1, 2), mar = c(4.5, 4.5, 1.5, 1.5))
set.seed(1)
plotDispShrink(ddsB)

## Warning:  zero-length arrow is of indeterminate angle and so skipped

mtext("A", side = 3, line = line, adj = adj, cex = cex)
set.seed(1)
plotDispShrink(ddsP)
mtext("B", side = 3, line = line, adj = adj, cex = cex)
```

```
line <- -0.1
adj <- -0.3
cex <- 1.5
par(mfrow = c(1, 2), mar = c(4.5, 4.5, 1.5, 1.5))
set.seed(1)
plotDispEsts(ddsB, cex = 0.1, ylim = c(1e-08, 10))
asymptDispB <- attr(dispersionFunction(ddsB), "coefficients")["asymptDisp"]
```

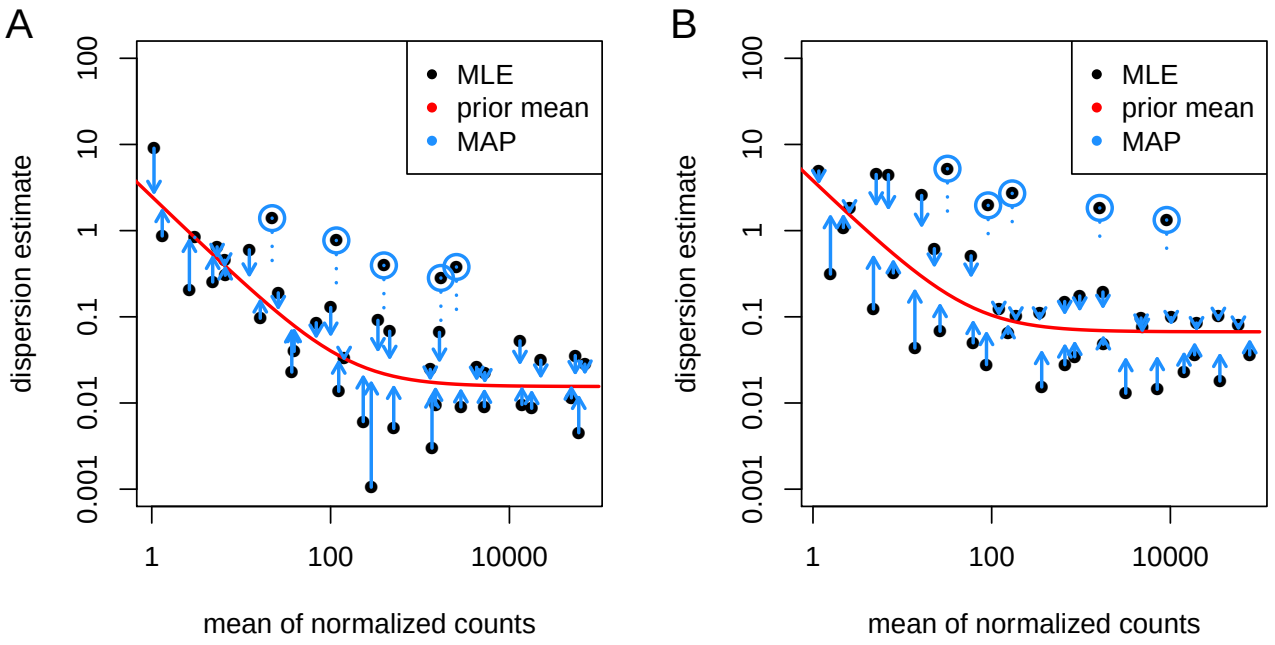

Figure 1: Plot of dispersion estimates over the base mean for a subset of the (A) Bottomly et al and (B) Pickrell et al dataset. Dispersion outliers are circled in blue with dotted lines indicating the effect shrinkage would have had on the estimate. Genes were selected for ease of visualization, including an enrichment of dispersion outliers.

```
abline(v = 1/(asymptDispB * range(sizeFactors(ddsB))))
mtext("A", side = 3, line = line, adj = adj, cex = cex)
set.seed(1)
plotDispEsts(ddsP, cex = 0.1, ylim = c(1e-08, 10))
asymptDispP <- attr(dispersionFunction(ddsP), "coefficients")["asymptDisp"]
abline(v = 1/(asymptDispP * range(sizeFactors(ddsP))))
mtext("B", side = 3, line = line, adj = adj, cex = cex)
```

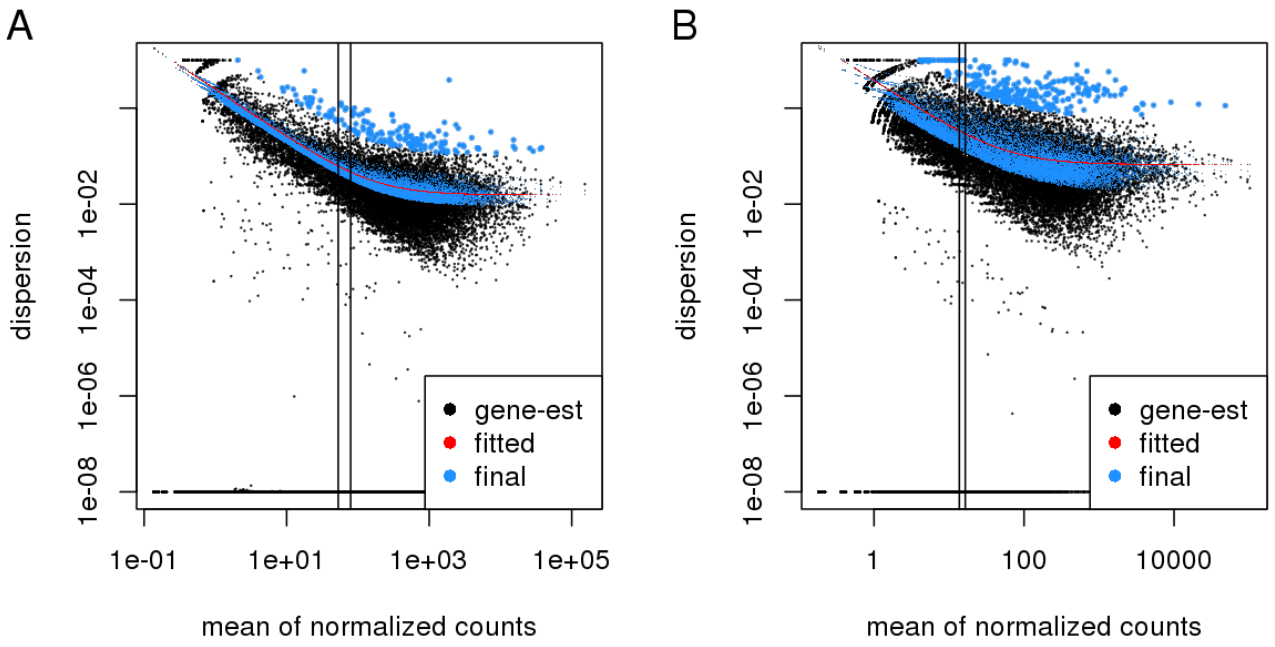

Figure 2: Plot of dispersion estimates over the base mean for a subset of the (A) Bottomly et al and (B) Pickrell et al dataset. Dispersion outliers are circled in blue with dotted lines indicating the effect shrinkage would have had on the estimate. Vertical lines indicate the reciprocal of dispersion on the scale of the samples with the smallest and largest size factor; the estimation of dispersion to the left of this line is difficult as described in the Methods section.

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 1.3, 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, gtools 3.3.1, schoolmath 0.4
- 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, xtable 1.7-3