

Mark Dunning

Analysis of the Pasilla dataset

29th July 2015

The counts for the pasilla dataset were read from the [pasilla data package](#).

The first few lines of the file are shown.

```
pasillaCountTable = read.table( datafile, header=TRUE, row.names=1 )  
  
head(pasillaCountTable)
```

```
##          untreated1 untreated2 untreated3 untreated4 treated1 treated2  
## FBgn0000003         0         0         0         0         0         0  
## FBgn0000008        92        161         76         70        140        88  
## FBgn0000014         5         1         0         0         4         0  
## FBgn0000015         0         2         1         2         1         0  
## FBgn0000017       4664       8714      3564      3150      6205      3072  
## FBgn0000018        583        761        245        310        722        299  
##          treated3  
## FBgn0000003         1  
## FBgn0000008        70  
## FBgn0000014         0  
## FBgn0000015         0  
## FBgn0000017      3334  
## FBgn0000018       308
```

A design matrix was used to compare treated and untreated samples.

```
pasillaDesign = data.frame(  
  row.names = colnames( pasillaCountTable ),  
  condition = c( "untreated", "untreated", "untreated",  
                 "untreated", "treated", "treated", "treated", "treated" ),  
  libType = c( "single-end", "single-end", "paired-end",  
               "paired-end", "single-end", "paired-end", "paired-end", "paired-end" ) )
```

```
pasillaDesign
```

```
##          condition  libType  
## untreated1 untreated single-end  
## untreated2 untreated single-end  
## untreated3 untreated paired-end  
## untreated4 untreated paired-end  
## treated1      treated single-end  
## treated2      treated paired-end  
## treated3      treated paired-end
```

```
pairedSamples = pasillaDesign$libType == "paired-end"
countTable = pasillaCountTable[ , pairedSamples ]
condition = pasillaDesign$condition[ pairedSamples ]
```

The analysis will use 3 Treated and 4 Untreated samples. Normalisation was performed with a standard edgeR protocol

```
y <- DGEList(counts=countTable,group=condition)
y <- calcNormFactors(y)
y <- estimateCommonDisp(y)
y <- estimateTagwiseDisp(y)
```

Differential expression was performed between treated and untreated samples using the exact test in [edgeR](#). The top hits are shown below.

```
et <- exactTest(y)
topTags(et)
```

```
## Comparison of groups: untreated-treated
##          logFC      logCPM      PValue      FDR
## FBgn0039155  4.378187  5.587721  1.988561e-183  2.903100e-179
## FBgn0003360  2.961327  8.058804  2.725221e-156  1.989275e-152
## FBgn0025111 -2.943074  7.158666  3.007218e-154  1.463412e-150
## FBgn0026562  2.446889  11.903496  1.955728e-106  7.137917e-103
## FBgn0039827  4.129115  4.281292  1.646830e-105  4.808413e-102
## FBgn0035085  2.499390  5.542361  1.617570e-96   3.935818e-93
## FBgn0029167  2.225726  8.062840  4.295075e-93   8.957687e-90
## FBgn0000071 -2.564871  5.033671  5.473558e-79   9.988558e-76
## FBgn0029896  2.545682  5.131528  1.268123e-77   2.057037e-74
## FBgn0034897  2.061625  6.096982  3.430337e-75   5.007949e-72
```

```
p <- 0.05
summary(de <- decideTestsDGE(et, p=p))
```

```
##      [,1]
## -1    625
##  0   13349
##  1     625
```

```
detags <- rownames(y)[as.logical(de)]
```

The number of down-regulated genes at a cutoff of 0.05 was 1250, and 625 genes were up-regulated. The logFC and CPM of these differentially-expressed genes is shown below.

```
plotSmeare(et, de.tags=detags)
abline(h = c(-2, 2), col = "blue")
```

