

Mark Dunning

Revised 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 ]
```

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

Prior to filtering, there are 14599 genes

```
countsPerMillion <- cpm(y)
summary(countsPerMillion)
```

```
##      untreated3      untreated4      treated2
## Min.   :  0.000   Min.   :  0.000   Min.   :  0.000
## 1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:  0.000
## Median :  2.154   Median :  1.931   Median :  2.089
## Mean   : 68.498   Mean   : 68.498   Mean   : 68.498
## 3rd Qu.: 41.994   3rd Qu.: 42.017   3rd Qu.: 43.461
## Max.   :15701.760   Max.   :16981.030   Max.   :15293.843
##      treated3
## Min.   :  0.000
## 1st Qu.:  0.000
## Median :  2.127
## Mean   : 68.498
## 3rd Qu.: 44.084
## Max.   :15869.130
```

```
#'summary' is a useful function for exploring numeric data; eg. summary(1:100)
countCheck <- countsPerMillion > 1
head(countCheck)
```

```
##      untreated3 untreated4 treated2 treated3
## FBgn0000003      FALSE      FALSE      FALSE      FALSE
## FBgn0000008       TRUE       TRUE       TRUE       TRUE
## FBgn0000014      FALSE      FALSE      FALSE      FALSE
## FBgn0000015      FALSE      FALSE      FALSE      FALSE
## FBgn0000017       TRUE       TRUE       TRUE       TRUE
## FBgn0000018       TRUE       TRUE       TRUE       TRUE
```

```
keep <- which(rowSums(countCheck) >= 2)
y <- y[keep,]
summary(cpm(y))
```

```
##      untreated3      untreated4      treated2
## Min.   :  0.12   Min.   :  0.102   Min.   :  0.104
## 1st Qu.: 10.53   1st Qu.: 10.873   1st Qu.: 11.283
## Median : 36.13   Median : 36.377   Median : 37.401
## Mean   : 126.91   Mean   : 126.925   Mean   : 126.918
## 3rd Qu.: 103.25   3rd Qu.: 98.767   3rd Qu.: 102.593
## Max.   :15701.76   Max.   :16981.030   Max.   :15293.843
##      treated3
## Min.   :  0.00
```

```
## 1st Qu.: 11.31
## Median : 37.90
## Mean : 126.92
## 3rd Qu.: 102.28
## Max. :15869.13
```

After filtering, 7873 genes remain.

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

```
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  4.385238  5.589470  5.135176e-197  4.042924e-193
## FBgn0025111 -2.935480  7.156088  4.089145e-154  1.609692e-150
## FBgn0003360  2.968460  8.060906  2.963278e-146  7.776628e-143
## FBgn0039827  4.136724  4.282611  1.518899e-103  2.989573e-100
## FBgn0035085  2.506729  5.543742  3.977713e-103  6.263307e-100
## FBgn0026562  2.453695  11.905324  4.727494e-95   6.203260e-92
## FBgn0029167  2.234275  8.065584  1.403281e-82   1.578290e-79
## FBgn0000071 -2.557948  5.031865  3.418496e-79   3.364227e-76
## FBgn0029896  2.553932  5.133395  8.862470e-77   7.752691e-74
## FBgn0034897  2.069850  6.098928  2.280255e-76   1.795245e-73
```

```
p <- 0.01
summary(de <- decideTestsDGE(et, p=p))
```

```
##      [,1]
## -1  477
##  0  6891
##  1   505
```

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

The number of down-regulated genes at a cutoff of 0.01 was 982, and 505 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")
```

