Cluster Analysis of Wasps

Lindsay Rutter

April 26, 2016

Introduction

This script runs three things: 1) DU vs DR with calcNormFactors and no filtering - TopDEG$_DU_DR_NoFilter_BtwnLane.csv-DU_DR_Genes_NoFilter_BtwnLane/2)DUvsDRwithBtwnLaneNormandcpmfiltering-TopDEG_DU_DR_Filter_BtwnLane.csv-DU_DR_Genes_Filter_BtwnLane/3)DUvsDRwithBtwnLaneNormandcpmandLoessfiltering-TopDEG_DU_DR_Filter_Loess_BtwnLane/$

```
> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% NO FILTERING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
> #% We are not filtering or doing Loess
>
> rm(list=ls())
> load("All_wasp.rda")
> listcond = rep(c("DR","DU"),each= 6)
> # create DGEList object
> d = DGEList(counts=countTable[,c(1:12)], group=listcond)
```

```
> ggparcoord(data.frame(d[[1]]), columns=1:12, alphaLines=0, boxplot=TRUE, scale="globalminmax") + coord
```

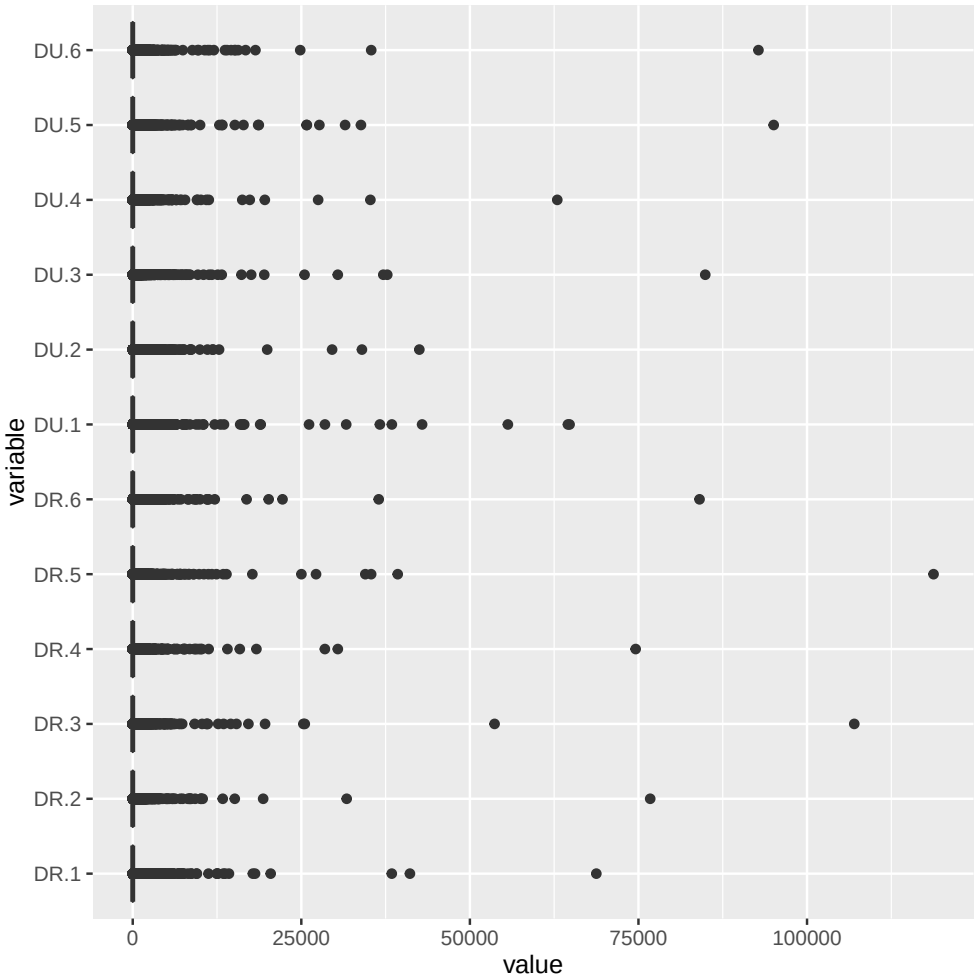

Figure 1: Boxplot of 12 DR and DU samples.

```
> myVec = c("DR", "DU")
> myCol = c(which(colnames(countTable) == grep('DR', colnames(countTable), value=TRUE)), which(colnames

> # estimate normalization factors
> d <- betweenLaneNormalization(d[[1]], which="full", round=FALSE)
> d = DGEList(counts=d, group=listcond)
```

```
> plotMDS(d, labels=colnames(countTable[,c(1:12)]), col = c("red","blue")[factor(listcond)])
```

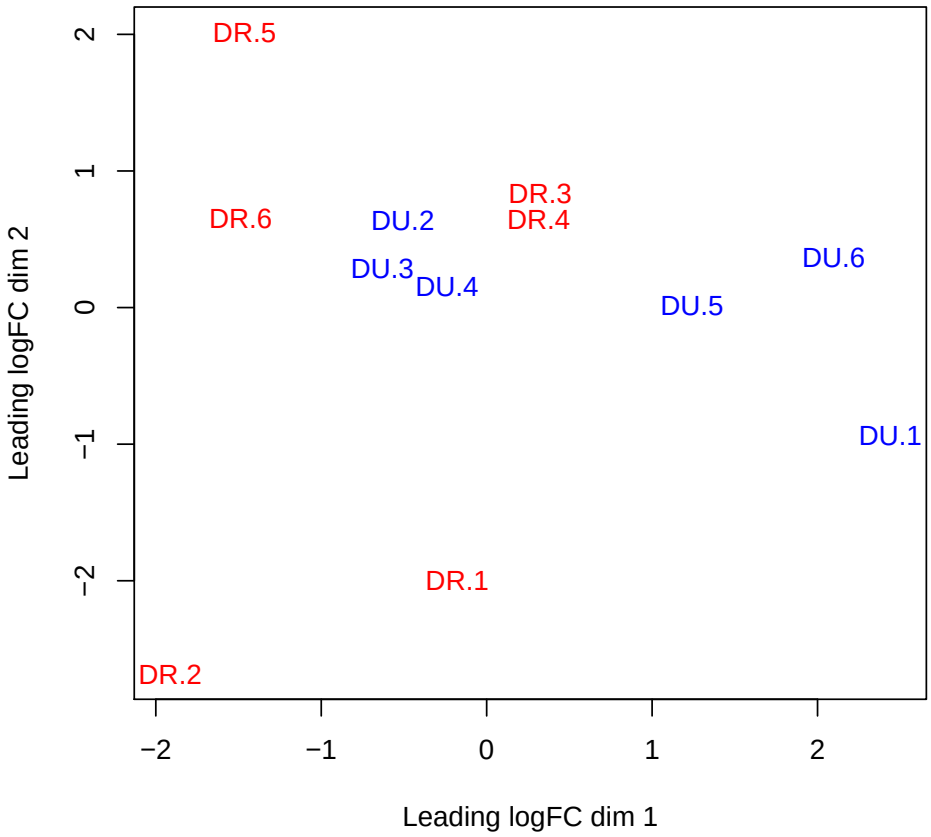

Figure 2: MDS of the 12 DU and DR samples.

```
> # estimate tagwise dispersion
> d = estimateCommonDisp(d)
> d = estimateTagwiseDisp(d)
> # Now, str(d) has raw read counts, norm factors, lib.size, and more
```

```
> plotMeanVar(d, show.tagwise.vars=TRUE, NBline=TRUE)
```

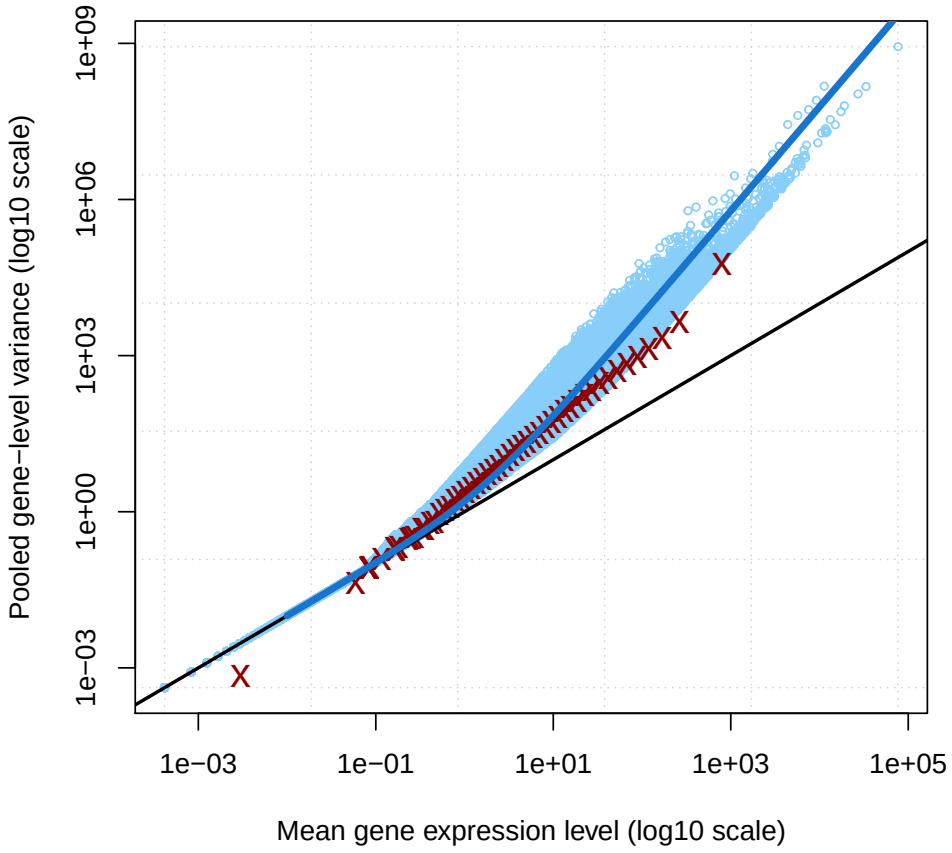

Figure 3: This function is useful for exploring the mean-variance relationship in the data. Raw variances are, for each gene, the pooled variance of the counts from each sample, divided by a scaling factor (by default the effective library size). The function will plot the average raw variance for genes split into nbins bins by overall expression level. The averages are taken on the square-root scale as for count data the arithmetic mean is upwardly biased. A line showing the Poisson mean-variance relationship (mean equals variance) is always shown.

```
> plotBCV(d)
```

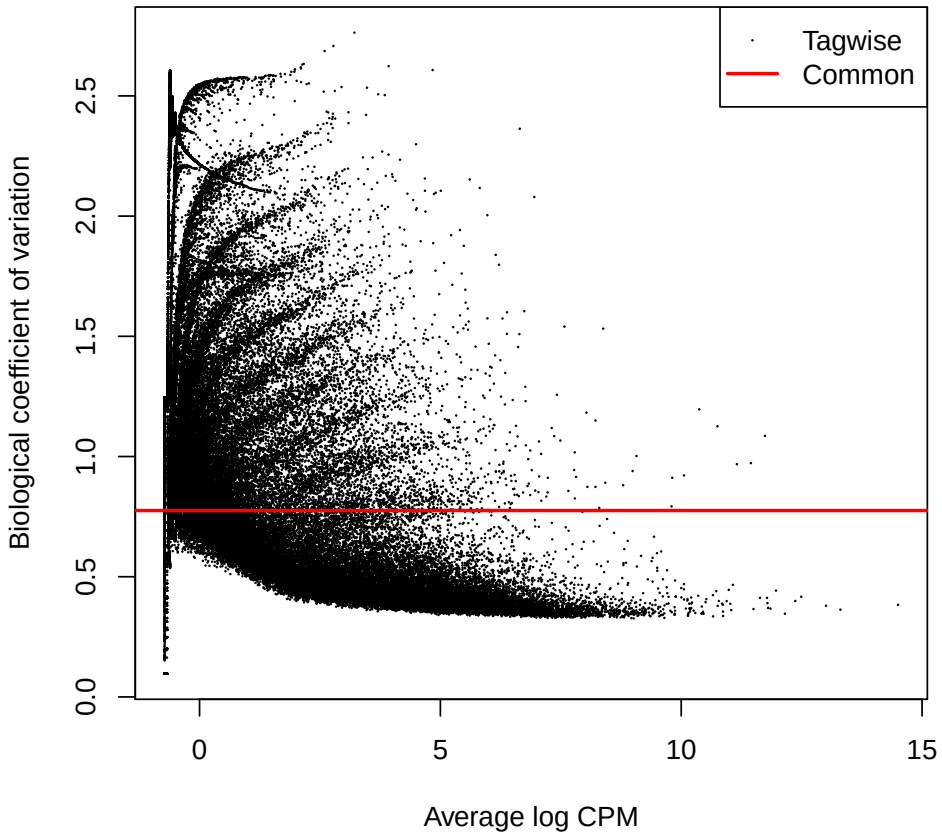

Figure 4: Plots the tagwise biological coefficient of variation (square root of dispersions) against log2-CPM.

```
> # Test for differential expression
> # Compute genewise exact tests for differences in the means between two groups of negative-binomially
> de = exactTest(d, pair=c("DU","DR"))
> #Use the topTags function to present a tabular summary of the differential expression statistics (note
> tt = topTags(de, n=nrow(d))
> head(tt$table)

          logFC   logCPM       PValue          FDR
34127  9.883000 4.203735 7.608463e-30 1.199786e-24
98531  7.259775 1.799679 3.272464e-14 2.580191e-09
91960  9.986667 4.304219 2.426726e-12 1.234494e-07
51991  9.441243 3.778315 3.131425e-12 1.234494e-07
98563  6.633044 1.299308 2.822268e-10 8.900926e-06
56841 -7.777288 2.242107 1.214618e-09 3.192238e-05

> length(which((tt$table)$FDR < 0.05))

[1] 31
```

```
> # There are 34 genes with FDR < 0.05
>
> # Inspect the depth-adjusted reads per million for some of the top differentially expressed genes (jus
> nc = cpm(d, normalized.lib.sizes=TRUE)
> rn = rownames(tt$table)
> # Sorted in order of lowest FDR from DE comparison
> head(nc[rn,order(listcond)],5)

          DR.1      DR.2      DR.3      DR.4      DR.5      DR.6 DU.1 DU.2
34127 43.928151 27.720944 19.070147 47.008760 39.651879 36.486622    0    0
98531  4.232436  5.323800  6.469581  3.688266  5.538445  9.269035    0    0
91960 44.798824 20.311158  0.000000 52.535112 34.471680 77.699968    0    0
51991  0.000000 32.819518 23.093985 36.749637 33.211019 31.522579    0    0
98563  5.308684  1.458679  3.325485  3.558269  5.316242  3.310370    0    0
      DU.3 DU.4 DU.5 DU.6
34127    0    0    0    0
98531    0    0    0    0
91960    0    0    0    0
51991    0    0    0    0
98563    0    0    0    0
```

```
> deg = rn[tt$table$FDR < .05]
> plotSmear(d, de.tags=deg)
```

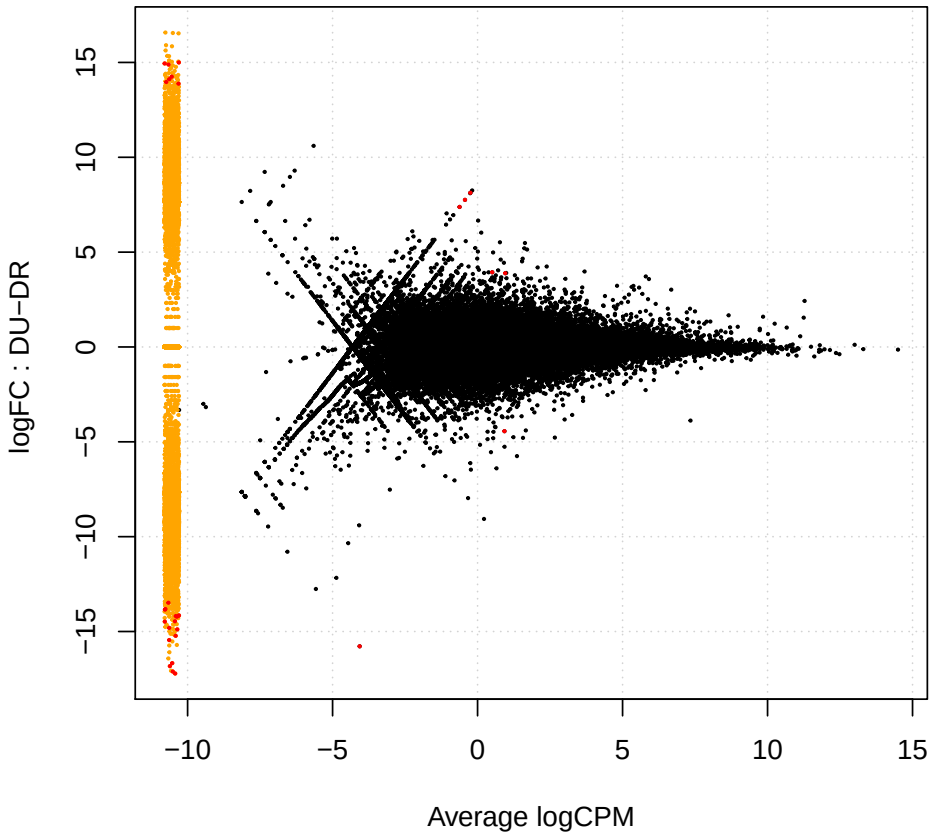

Figure 5: Create a graphical summary, such as an M (log-fold change) versus A (log-average expression) plot, here showing the genes selected as differentially expressed with a 5% false discovery rate. There were 34 in this dataset!

```
> # Would save file
> write.csv(tt$table, file="TopDEG_DU_DR_NoFilter_CalcNorm.csv")
```

```
> topInfo = cbind(nc[rn,order(listcond)], tt$table)
> for (i in 1:100){
+   gene = topInfo[i,1:12]
+   rep = 6
+   fact = 2
+   dat = data.frame(x=rep(1:fact, each=rep),y=t(gene),z=rep(1:rep, times = fact))
+   colnames(dat)=c("x","y","rep")
+   dat$x=as.factor(dat$x)
+   levels(dat$x)=c("DR","DU")
+   genePlot = ggplot(dat, aes(x, y)) + geom_point(aes(colour = factor(x)), shape = 20, size=5) + scale_
+
+   jpeg(file = paste(getwd(), "/DU_DR_Genes_NoFilter_BtwnLane/", "Gene_", i, ".jpg", sep=""), height =
+   print(genePlot)
```

```
+   dev.off()
+ }

> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
> #%%%%%%%%%%%%%%%%%%%%%% START OVER WITH FILTERING NOW %%%%%%%%%%%%%%%%%%%%%%%%%%
> #% We are filtering on cpm values this time (as recommended by EdgeR).
> #% But we are not doing Loess filtering just yet.
> rm(list=ls())
> load("All_wasp.rda")
> listcond = rep(c("DR","DU"),each= 6)
> # 157,691 genes
> y = DGEList(counts=countTable[,c(1:12)], group=listcond)
> keep <- rowSums(cpm(y)>1) >= 6
> y <- y[keep, keep.lib.sizes=FALSE] # it seems library sizes are recalculated (y$samples$lib.size = co
> y <- betweenLaneNormalization(y[[1]], which="full", round=FALSE)
> y <- DGEList(counts=y, group=listcond)


> plotMDS(y, labels=colnames(countTable[,c(1:12)]), col = c("red","blue")[factor(listcond)])
```

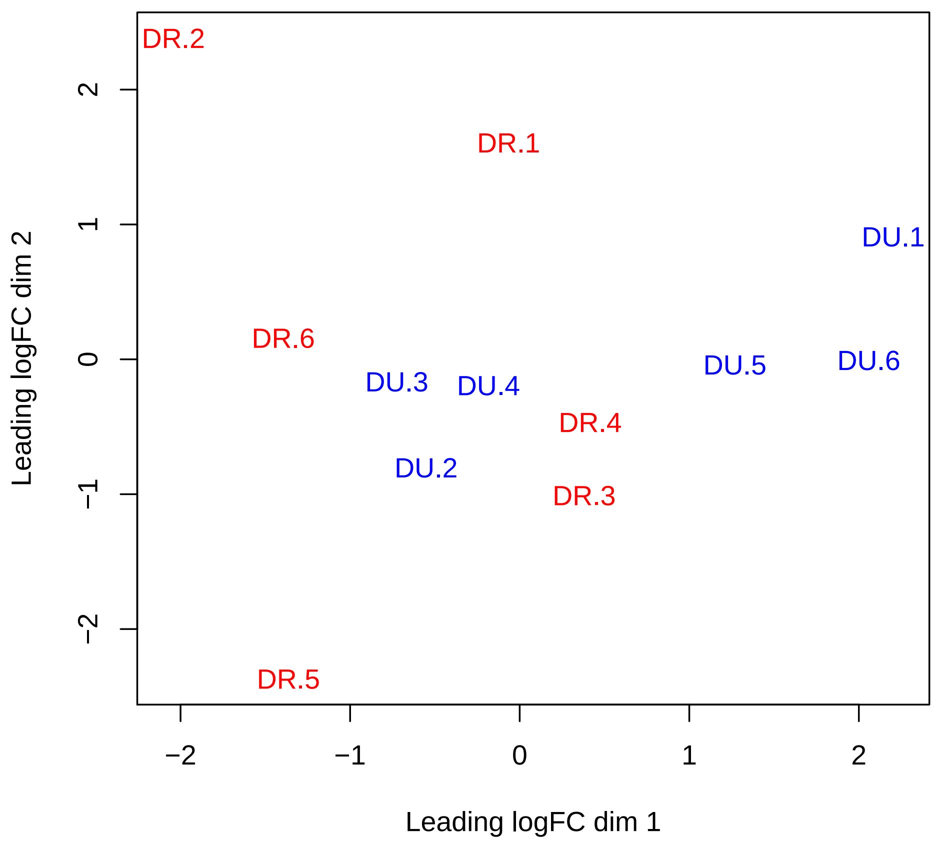

Figure 6: MDS plot 12 DR and DU samples.

```
> y = estimateCommonDisp(y)
> y = estimateTagwiseDisp(y)
```

```
> plotMeanVar(y, show.tagwise.vars=TRUE, NBline=TRUE)
```

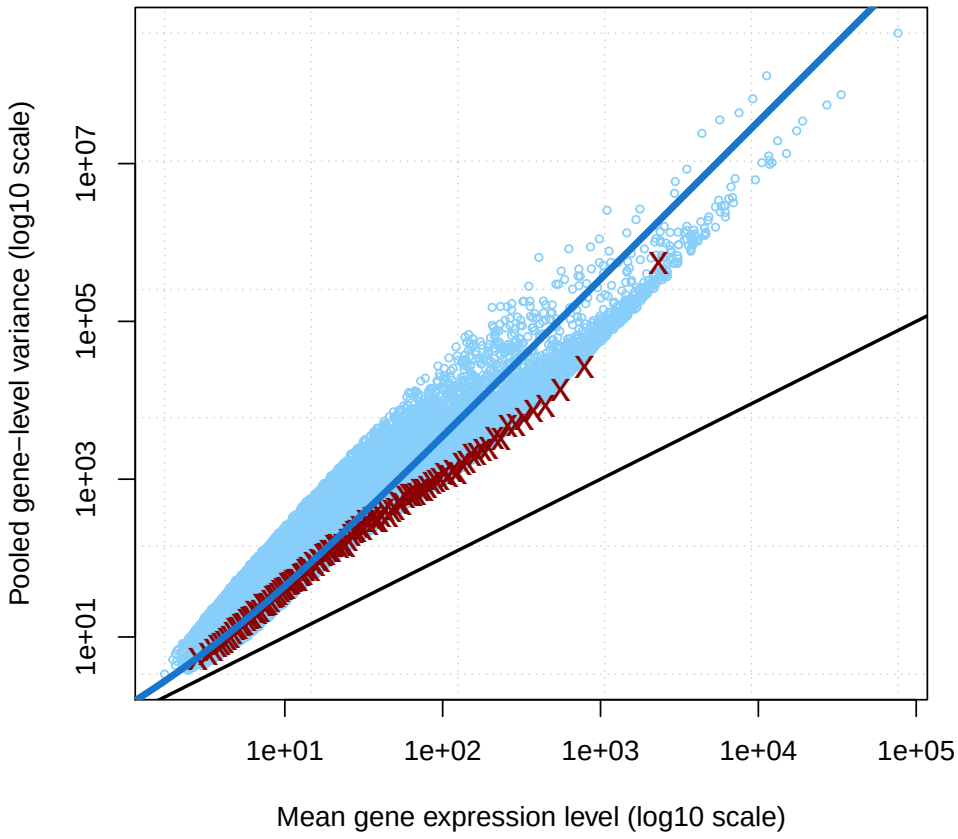

Figure 7: MeanVar plot of 12 DR and DU samples.

```
> plotBCV(y)
```

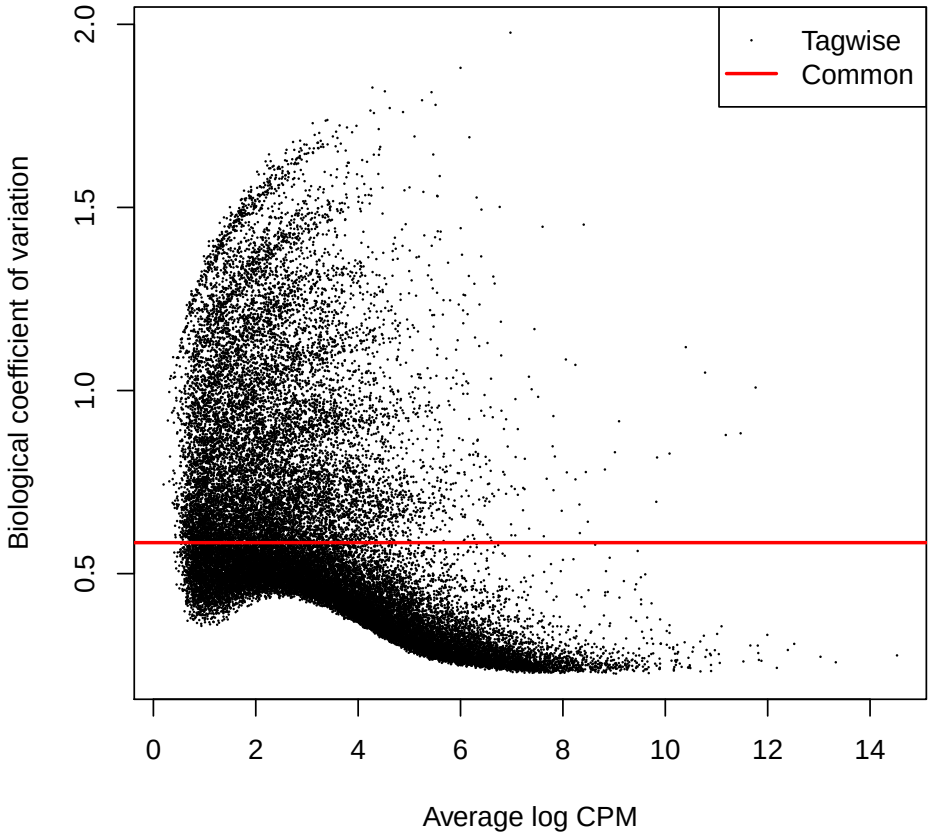

Figure 8: BCV plot of 12 DR and DU samples.

```
> #dim(32813, 3)
> de = exactTest(y, pair=c("DR","DU"))
> tt = topTags(de, n=nrow(y))
> head(tt$table)
```

```
          logFC   logCPM       PValue          FDR
34127 -9.885340 4.231558 1.588679e-36 5.212931e-32
98531 -7.264349 1.829040 3.584232e-14 5.880471e-10
98563 -6.635174 1.326488 1.563742e-11 1.710369e-07
74124  3.818227 1.860505 4.224253e-07 3.465260e-03
59730  3.667433 2.234715 9.993197e-07 6.558135e-03
62812 -5.603138 3.377323 2.081338e-06 1.138249e-02
```

```
> # ONLY 7
> length(which((tt$table)$FDR < 0.05))
```

```
[1] 8
```

```
> nc = cpm(y, normalized.lib.sizes=TRUE)
> rn = rownames(tt$table)
```

```
> # Sorted in order of lowest FDR from DE comparison
> head(nc[rn,order(listcond)],5)

           DR.1      DR.2      DR.3       DR.4       DR.5       DR.6     DU.1
34127 44.834791 28.370237 19.392509 47.8412146 40.4097787 37.1941057 0.000000
98531  4.300386  5.529727  6.523662  3.7280267  5.6774324  9.4916207 0.000000
98563  5.375867  1.538600  3.384919  3.5757054  5.4497196  3.3849190 0.000000
74124  1.846319  0.000000  0.000000  0.3492621  0.0000000  0.0000000 6.154398
59730  2.461759  0.000000  0.000000  0.0000000  0.3077199  0.6154398 9.662405
           DU.2     DU.3     DU.4     DU.5     DU.6
34127  0.000000 0.000000 0.000000 0.000000 0.000000
98531  0.000000 0.000000 0.000000 0.000000 0.000000
98563  0.000000 0.000000 0.000000 0.000000 0.000000
74124  4.194222 4.101906 4.321926 8.314592 6.908312
59730 11.737976 6.769838 4.615799 7.692998 5.231238

> # just for plotting purposes
> deg = rn[tt$table$FDR < .05] # Only 7


> plotSmear(y, de.tags=deg)
```

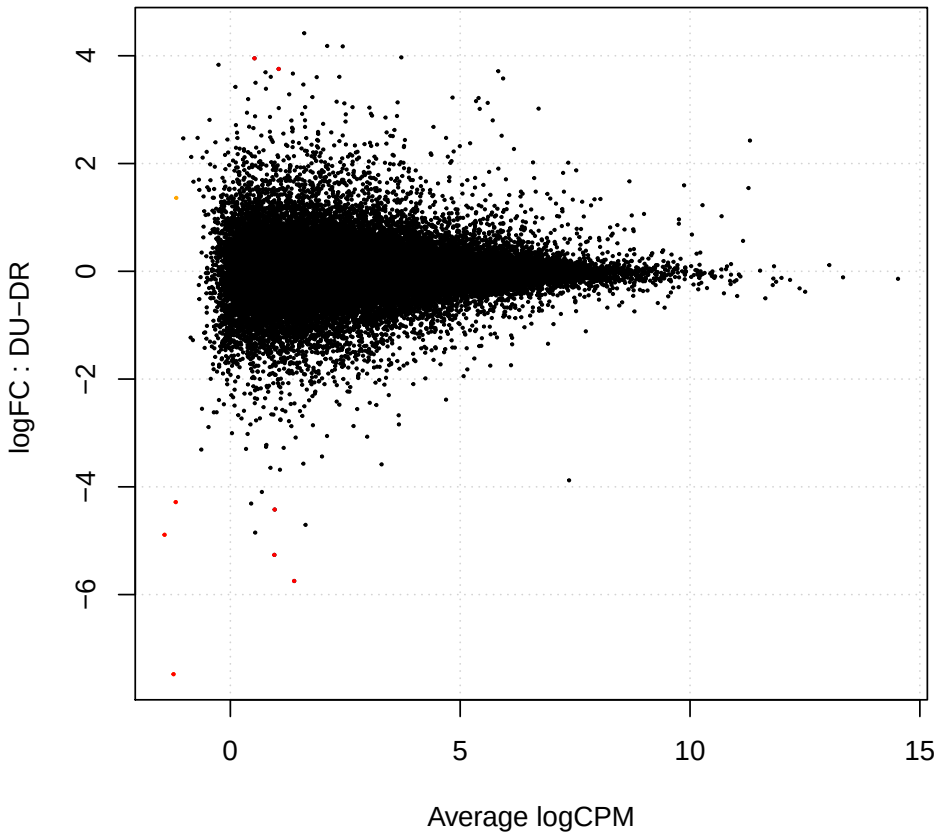

Figure 9: Plot smear of 12 DR and DU samples.

```
> write.csv(tt$table, file="TopDEG_DU_DR_Filter_CalcNorm.csv")

> topInfo = cbind(nc[rn,order(listcond)], tt$table)
> for (i in 1:100){
+   gene = topInfo[i,1:12]
+   rep = 6
+   fact = 2
+   dat = data.frame(x=rep(1:fact, each=rep),y=t(gene),z=rep(1:rep, times = fact))
+   colnames(dat)=c("x","y","rep")
+   dat$x=as.factor(dat$x)
+   levels(dat$x)=c("DR","DU")
+   genePlot = ggplot(dat, aes(x, y)) + geom_point(aes(colour = factor(x)), shape = 20, size=5) + scale_
+
+   jpeg(file = paste(getwd(), "/DU_DR_Genes_Filter_BtwnLane/", "Gene_", i, ".jpg", sep=""), height = 7(
+   print(genePlot)
+   dev.off()
+ }

> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
> #%%%%%%%%%%%%%%%%%%%%%%% START OVER WITH FILTERING NOW %%%%%%%%%%%%%%%%%%%%%%%%%%
> #% We are filtering on cpm values this time (as recommended by EdgeR).
> #% And also now doing Loess filtering.
>
> rm(list=ls())
> load("All_wasp.rda")
> listcond = rep(c("DR","DU"),each= 6)
> # 157,691 genes
> y = DGEList(counts=countTable[,c(1:12)], group=listcond)
> keep <- rowSums(cpm(y)>1) >= 6
> # it seems library sizes are recalculated (y$samples$lib.size = colSums(y$counts))
> y <- y[keep, keep.lib.sizes=FALSE]
> ########################## EXTRA FILTERING AT THIS STEP ########################
>
> RowSD = function(x) {
+   sqrt(rowSums((x - rowMeans(x))^2)/(dim(x)[2] - 1))
+ }
> yt = y
> yt2 = as.data.frame(yt[[1]])
> y = mutate(yt2, mean = (DR.1+DR.2+DR.3+DR.4+DR.5+DR.6+DU.1+DU.2+DU.3+DU.4+DU.5+DU.6)/ncol(yt2), stdev
> rownames(y)=rownames(yt)
> # The first quartile threshold of mean counts across the 12 samples
> q1T = as.numeric(summary(y$mean)["1st Qu."])
> # 24,610 genes
> d2q1 = subset(y,mean>q1T)
> # The first quartile threshold of standard deviation across the 12 samples
> q1Ts = as.numeric(summary(d2q1$stdev)["1st Qu."])
> # 18,458 genes
> d2q1 = subset(d2q1,stdev>q1Ts)
> # 14,355
> filt = subset(y,mean<=q1T|stdev<=q1Ts)
> model = loess(mean ~ stdev, data=d2q1)
> # 8,058 genes
> d2q1 = d2q1[which(sign(model$residuals) == 1),]
> d2q1 = d2q1[,1:(ncol(d2q1)-2)]
```

```
> # (filt 14,355 genes)
> filt = filt[,1:(ncol(filt)-2)]
> colnames(filt)=colnames(d2q1)
> # filt (24,755 genes)
> filt = rbind(filt,d2q1[which(sign(model$residuals) == -1),])
> #filts = t(apply(as.matrix(filt), 1, scale))
> #colnames(filts)=colnames(d2q1)
> colnames(filt)=colnames(d2q1)
> y = DGEList(counts=d2q1, group=listcond)
> y <- betweenLaneNormalization(y[[1]], which="full", round=FALSE)
> y <- DGEList(counts=y, group=listcond)
```

```
> plotMDS(y, labels=colnames(countTable[,c(1:12)]), col = c("red","blue")[factor(listcond)])
```

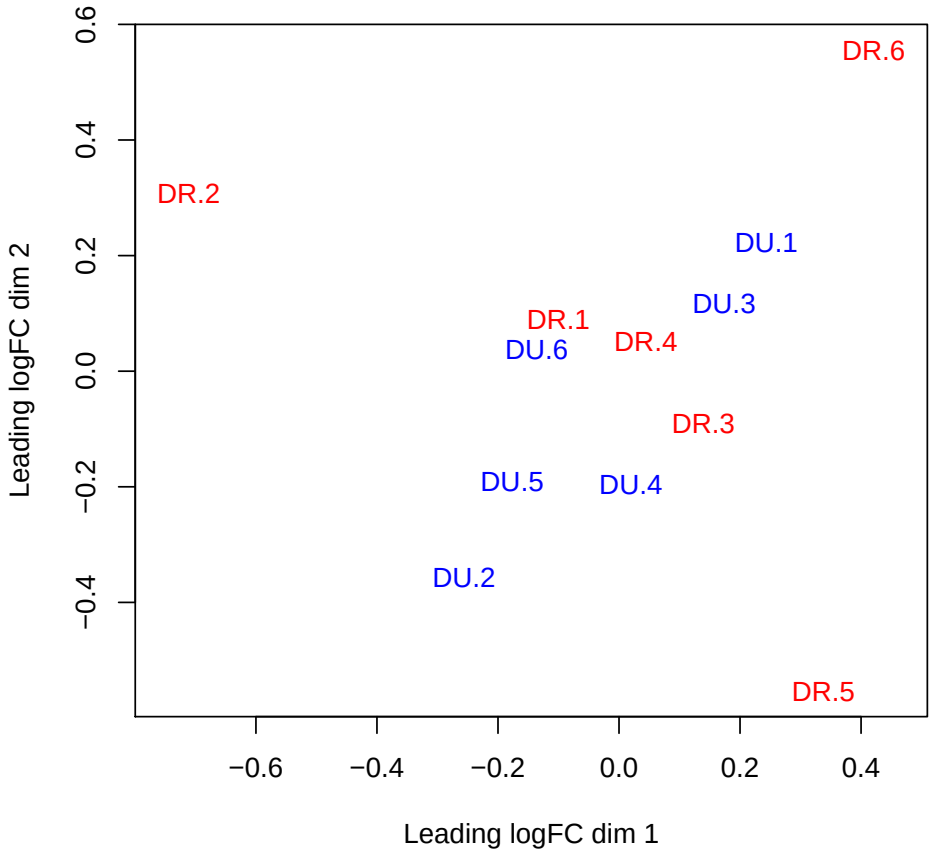

Figure 10: MDS plot 12 DR and DU samples.

```
> y = estimateCommonDisp(y)
> y = estimateTagwiseDisp(y)
```

```
> plotMeanVar(y, show.tagwise.vars=TRUE, NBline=TRUE)
```

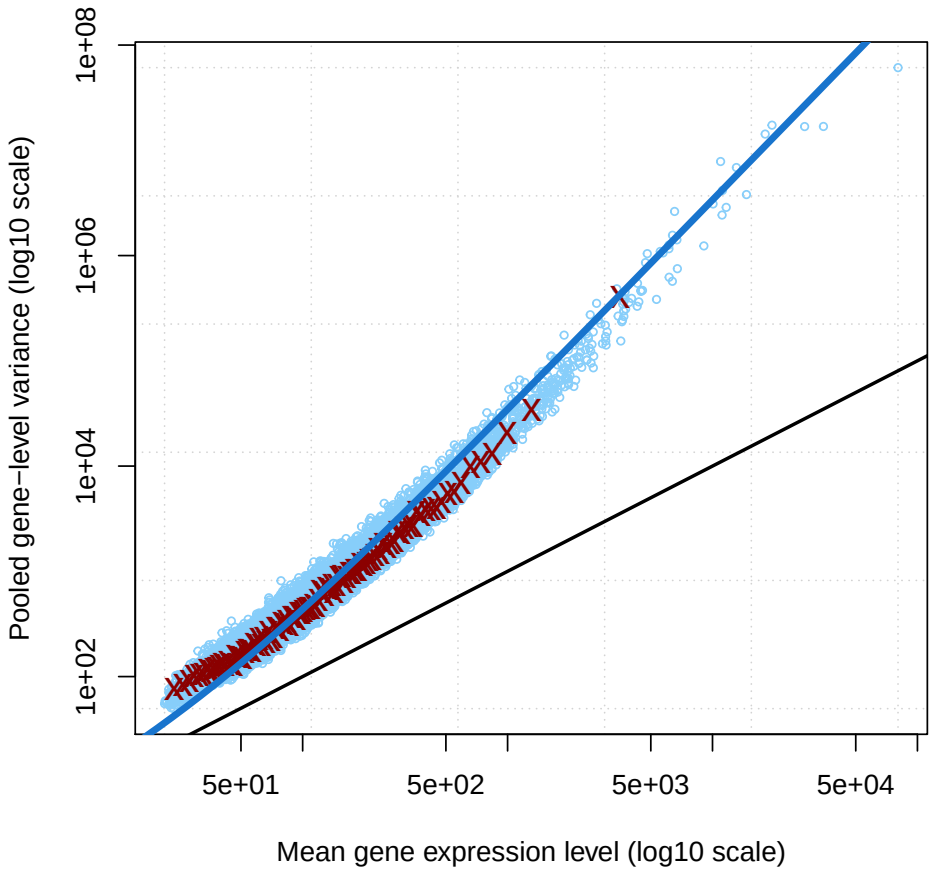

Figure 11: MeanVar plot of 12 DR and DU samples.

```
> plotBCV(y)
```

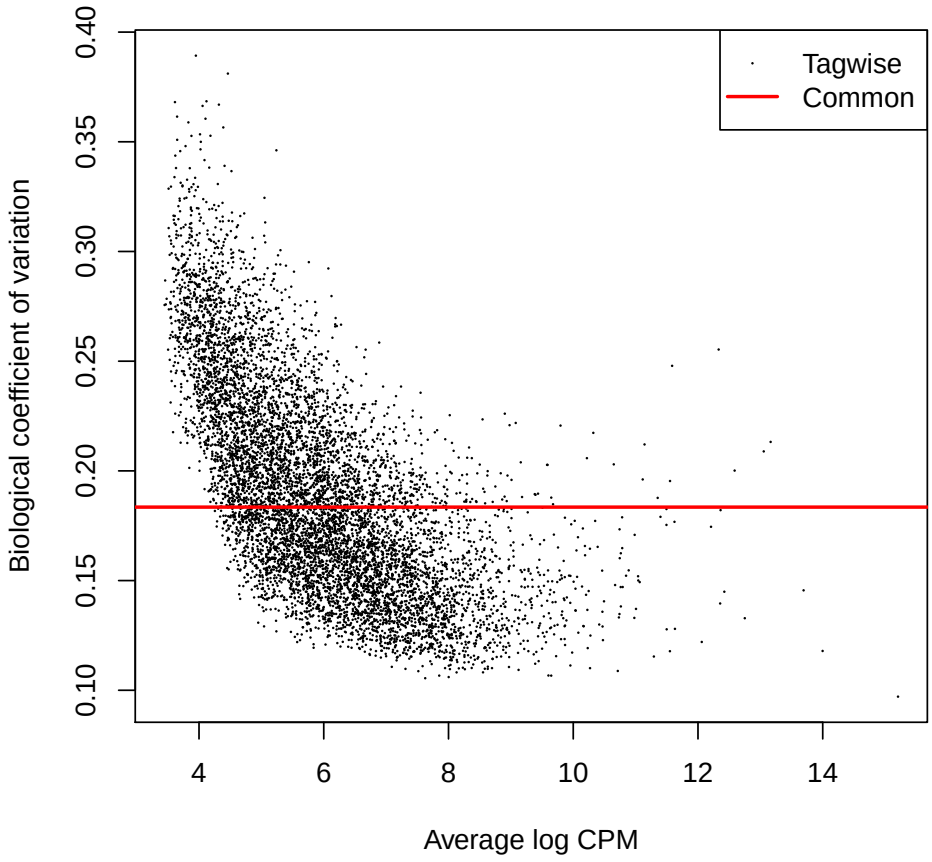

Figure 12: BCV plot of 12 DR and DU samples.

```
> #dim(32813, 3)
> de = exactTest(y, pair=c("DR","DU"))
> tt = topTags(de, n=nrow(y))
> head(tt$table)

           logFC   logCPM       PValue        FDR
51268 -0.7471120 5.278767 3.451014e-06 0.02780827
34025  0.7454142 4.724073 1.465712e-04 0.59053547
39735 -0.4983209 6.637186 2.408833e-04 0.64701242
67325  0.7589919 4.328062 5.432741e-04 0.71669116
93602 -0.8868067 3.618180 6.017719e-04 0.71669116
67022 -0.5366935 6.104398 6.244608e-04 0.71669116

> # ONLY 1
> length(which((tt$table)$FDR < 0.05))

[1] 1

> nc = cpm(y, normalized.lib.sizes=TRUE)
> rn = rownames(tt$table)
```

```
> # Sorted in order of lowest FDR from DE comparison
> head(nc[rn,order(listcond)],5)
          DR.1       DR.2      DR.3      DR.4      DR.5      DR.6      DU.1
51268 51.40133  48.305769  49.87237  39.28255  47.54599  48.53159 20.203460
34025 18.34754  15.750654  15.40017  27.76123  15.28961  21.64068 38.868550
39735 99.29075 124.996118 103.04023 104.30104 141.91349 119.27310 72.914990
67325 12.76330   8.319902  17.43957  14.72507  18.81799  13.17259 19.288435
93602 12.74448  14.045274  11.99647  11.96353  14.60981  23.05203  7.374298
          DU.2      DU.3      DU.4      DU.5     DU.6
51268 25.70066 29.732417 26.072319 31.576580 36.33753
34025 27.96117 34.738002 31.736533 33.980578 24.38811
39735 98.85558 92.709159 70.779149 82.561560 72.54098
67325 19.28844 26.862675 31.597750 22.111133 25.34547
93602  5.65010  9.557184  7.291969  6.487501 11.29079
> # just for plotting purposes
> deg = rn[tt$table$FDR < .05] # Only 1
> write.csv(tt$table, file="TopDEG_DU_DR_Filter_Loess_CalcNorm.csv")

> plotSmear(y, de.tags=deg)
```

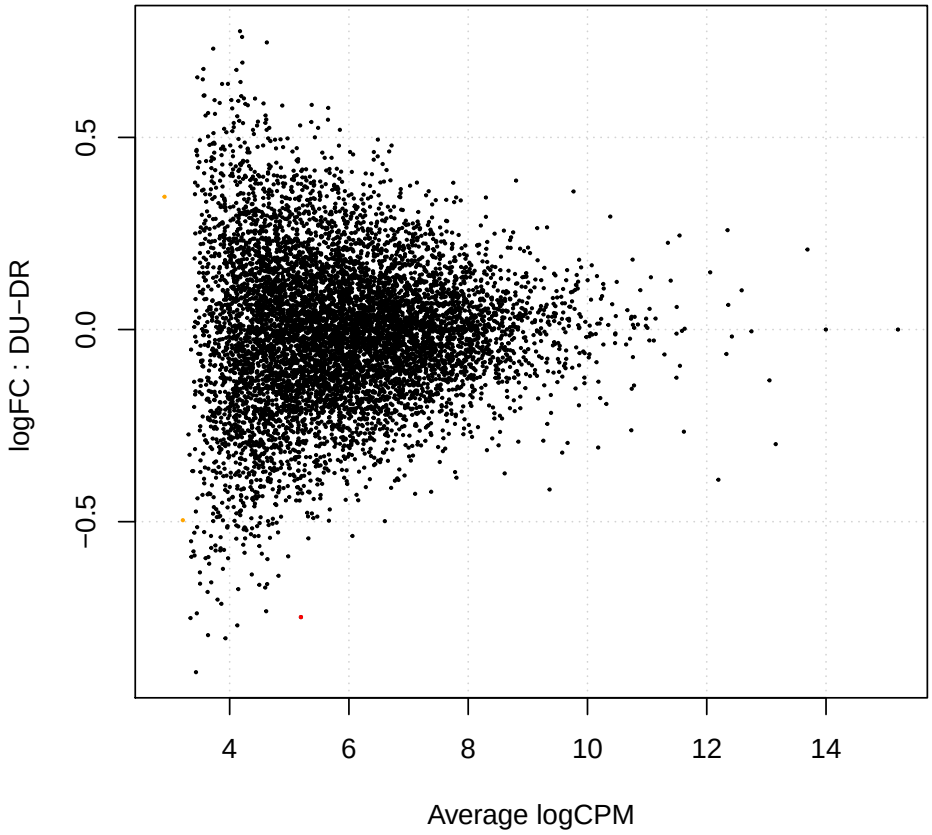

Figure 13: Plot smear of 12 DR and DU samples.

```
> topInfo = cbind(nc[rn,order(listcond)], tt$table)
> for (i in 1:100){
+   gene = topInfo[i,1:12]
+   rep = 6
+   fact = 2
+   dat = data.frame(x=rep(1:fact, each=rep),y=t(gene),z=rep(1:rep, times = fact))
+   colnames(dat)=c("x","y","rep")
+   dat$x=as.factor(dat$x)
+   levels(dat$x)=c("DR","DU")
+   genePlot = ggplot(dat, aes(x, y)) + geom_point(aes(colour = factor(x)), shape = 20, size=5) + scale_
+
+   jpeg(file = paste(getwd(), "/DU_DR_Genes_Filter_Loess_BtwnLane/", "Gene_", i, ".jpg", sep=""), heigh
+   print(genePlot)
+   dev.off()
+ }
```