Cluster Analysis of Wasps

Lindsay Rutter

April 19, 2016

Introduction

```
> 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
```

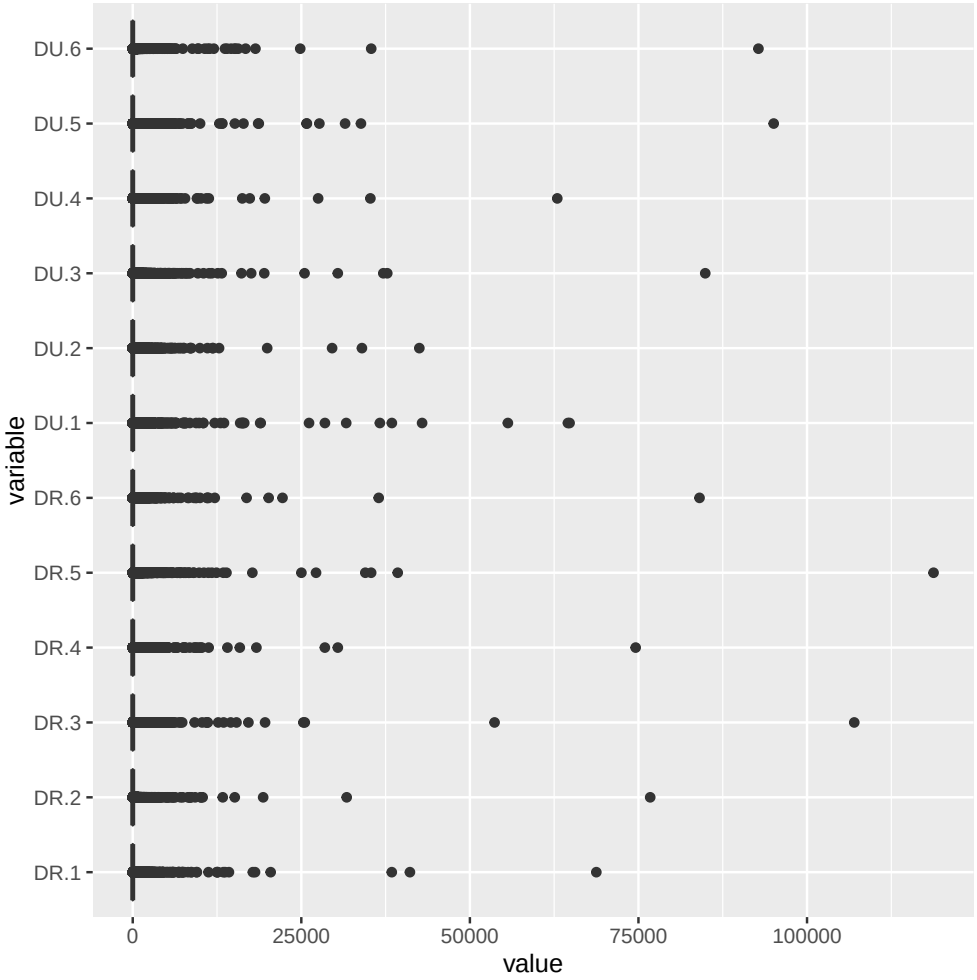

Figure 1: Boxplot of 12 DR and DU samples.

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

> # estimate normalization factors
> d = calcNormFactors(d)


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

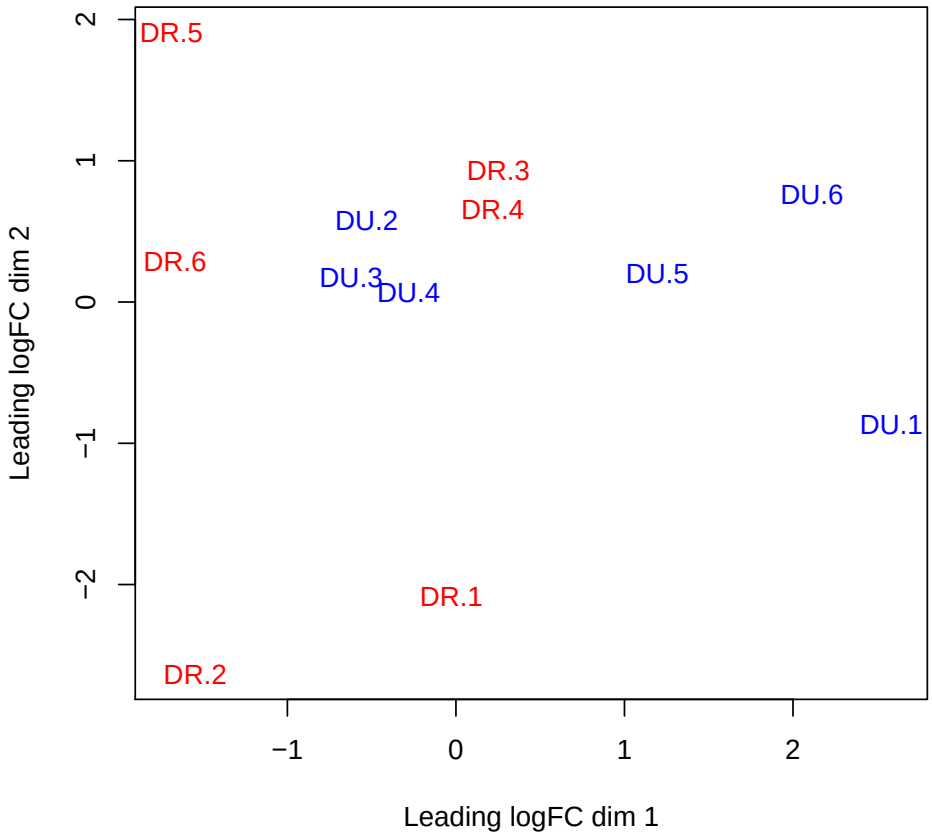

Figure 2: MDS of the 12 DU and DR samples.

```
> # 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)
```

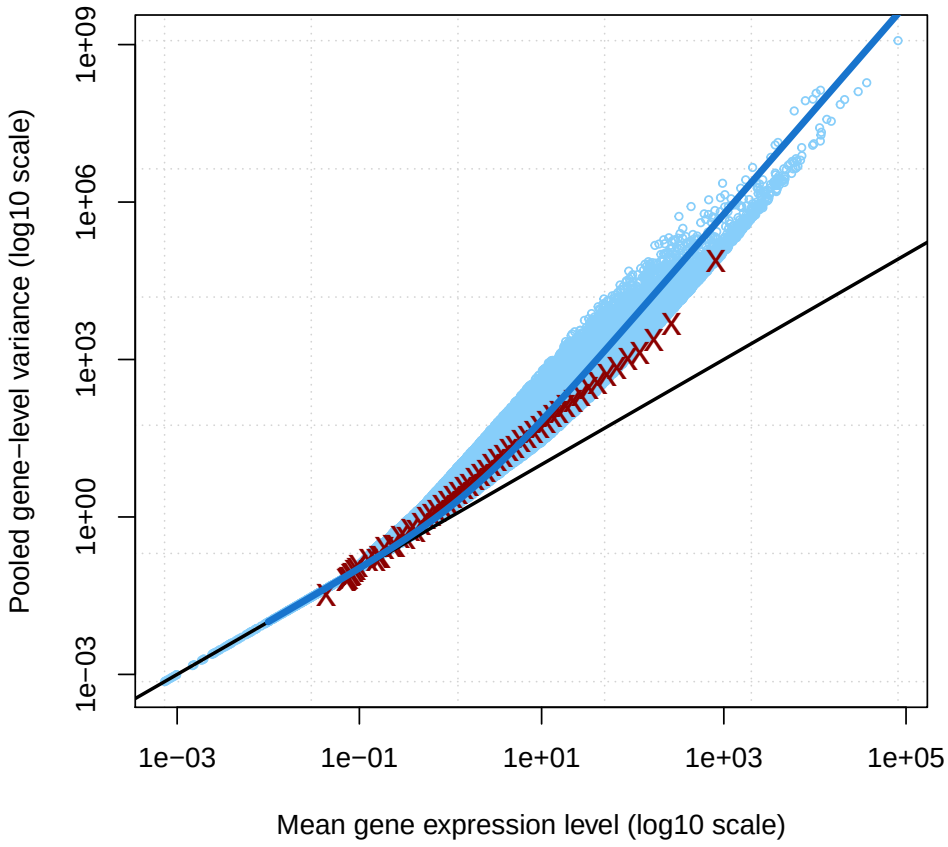

Figure 3: This function is useful for exploring the mean-variance relationship in the data. Raw variances are, for each gene, the pooled variance of the counts from each sample, divided by a scaling factor (by default the effective library size). The function will plot the average raw variance for genes 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)
```

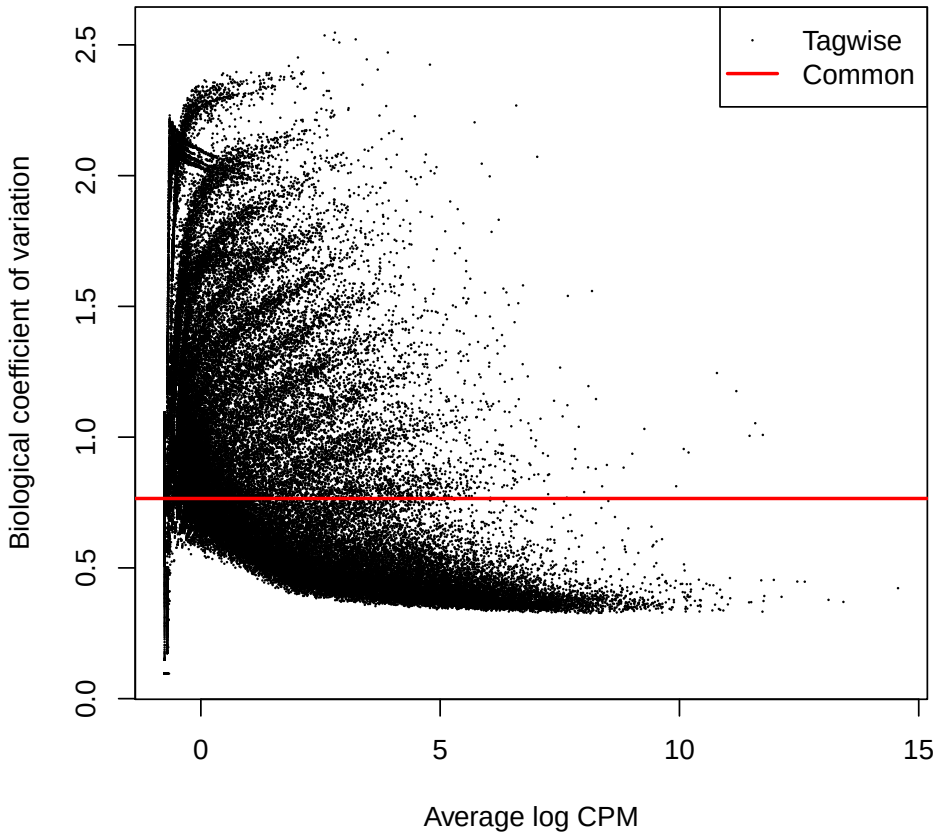

Figure 4: Plots the tagwise biological coefficient of variation (square root of dispersions) against log2-CPM.

```
> # 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  9.892378 4.180810 8.570717e-30 1.351525e-24
98531  7.242351 1.753758 2.498226e-14 1.969739e-09
91960 10.009367 4.292985 2.449598e-12 1.058102e-07
51991  9.441423 3.745375 2.683988e-12 1.058102e-07
98563  6.610023 1.250211 2.908486e-10 9.172840e-06
56841 -7.798801 2.228502 8.518851e-10 2.238910e-05

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

[1] 34
```

```
> # 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 43.171984 25.701993 18.890110 46.481883 39.504769 36.843917    0    0
98531  4.164191  5.562410  6.315535  3.583096  5.155995  8.623044    0    0
91960 43.964297 18.812537  0.000000 52.088373 33.969573 79.579269    0    0
51991  0.000000 30.000912 22.993424 36.789961 32.625393 31.608620    0    0
98563  5.276027  1.540594  3.270757  3.488249  4.963565  2.880104    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)
```

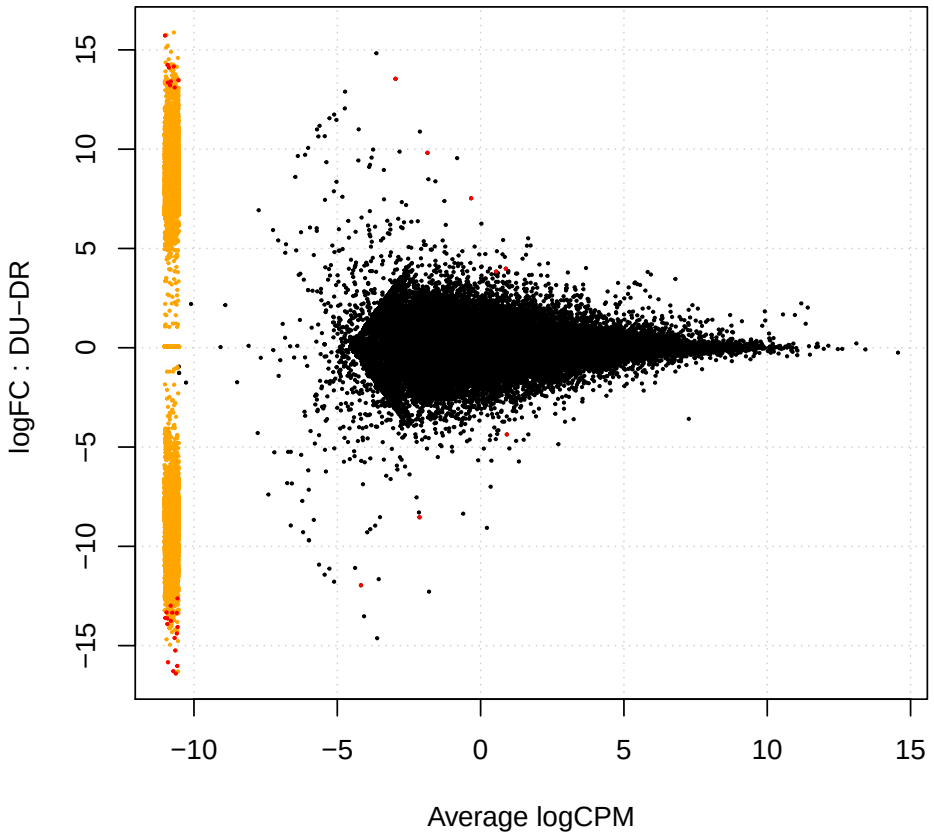

Figure 5: Create a graphical summary, such as an M (log-fold change) versus A (log-average expression) plot, here showing the genes selected as differentially expressed with a 5% false discovery rate. There were 34 in this dataset!

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

> #%%%%%%%%%%%%%%%%%%%%%% START OVER WITH FILTERING NOW %%%%%%%%%%%%%%%%%%%%%%
> #% We are filtering on cpm values this time (as recommended by EdgeR). But we are not doing Loess filt
>
> 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 <- calcNormFactors(y)
```

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

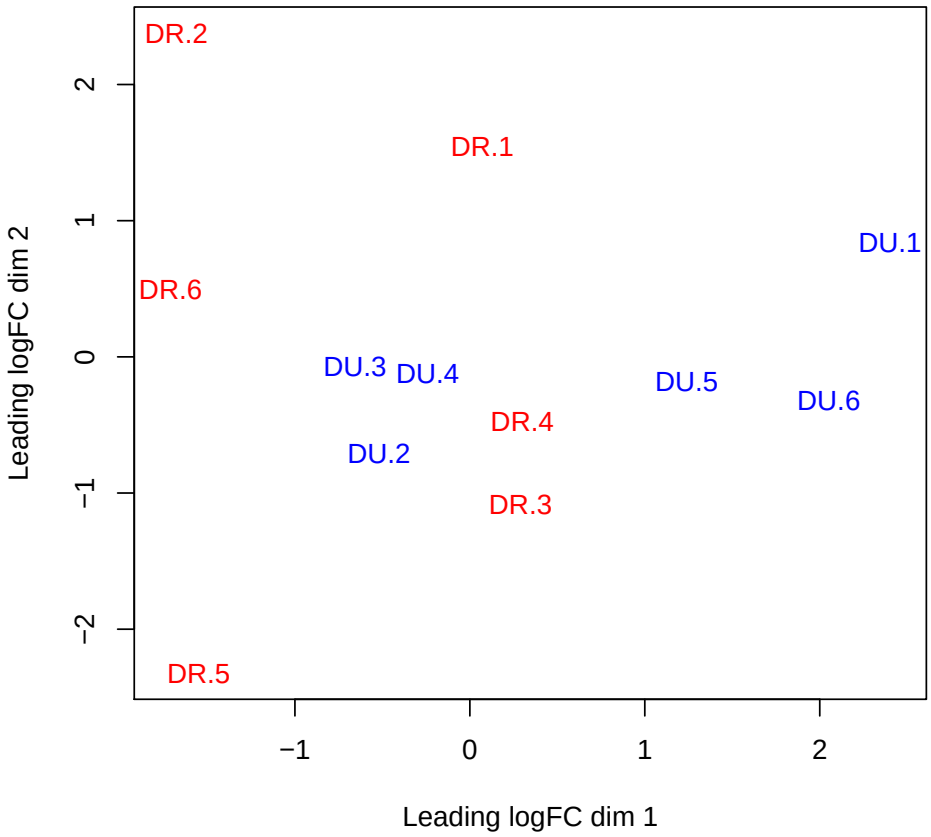

Figure 6: MDS plot 12 DR and DU samples.

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

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

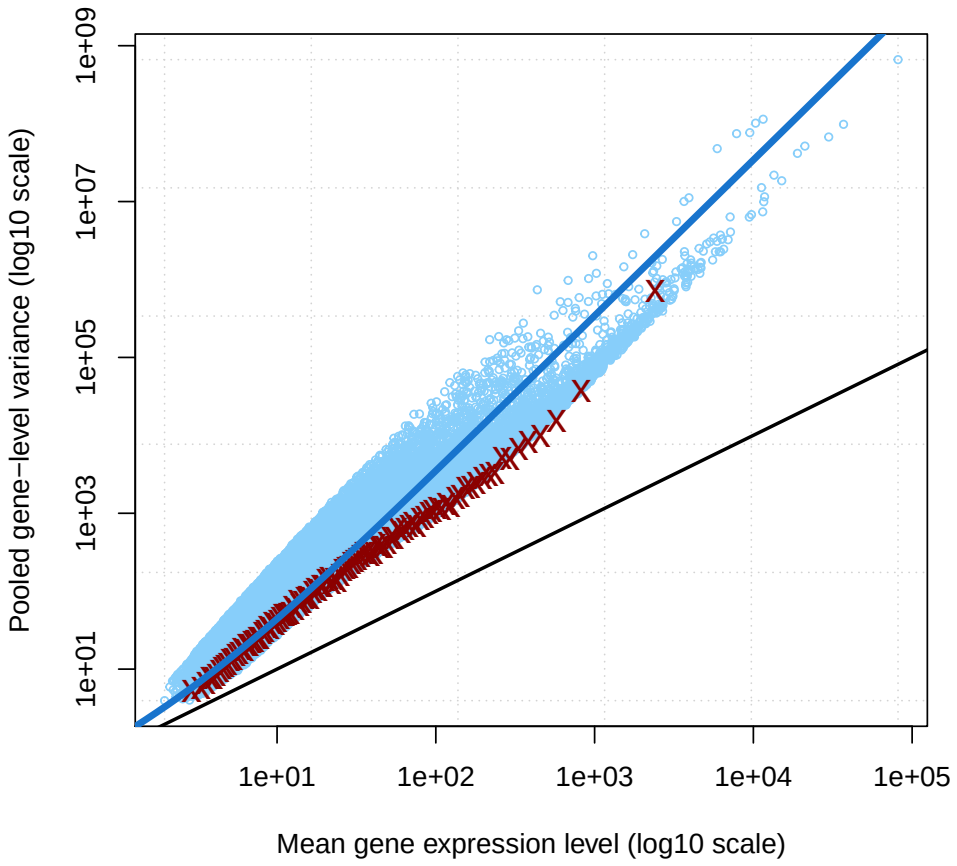

Figure 7: MeanVar plot of 12 DR and DU samples.

```
> plotBCV(y)
```

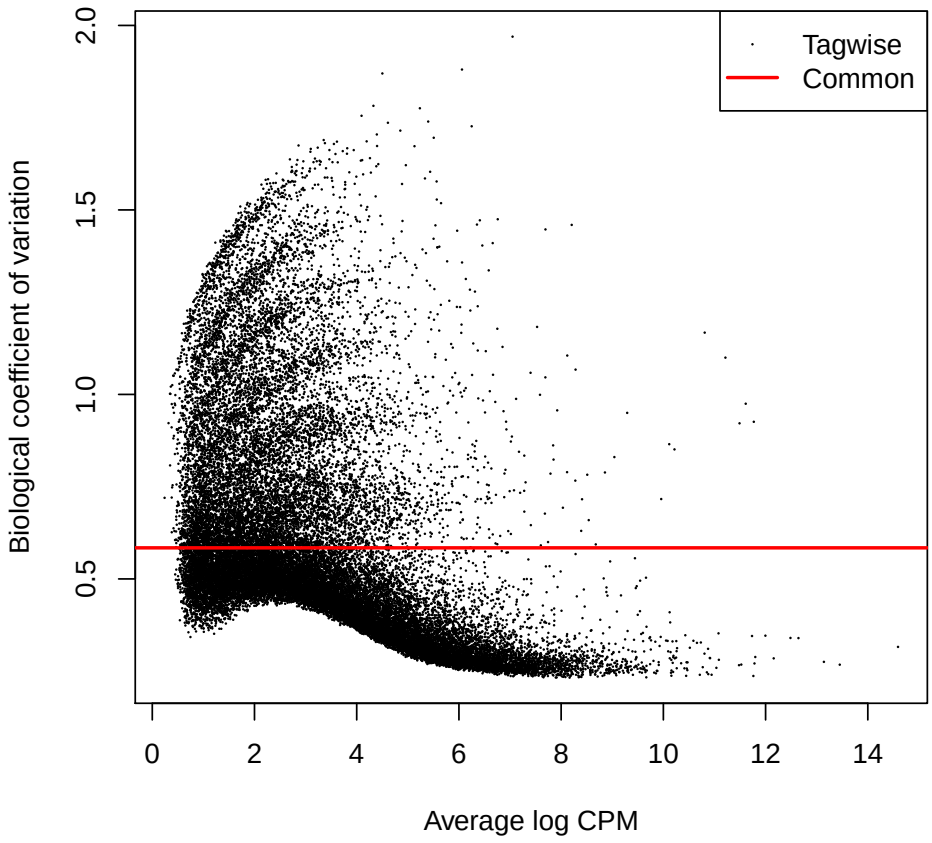

Figure 8: BCV plot of 12 DR and DU samples.

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

```
          logFC   logCPM       PValue          FDR
34127 -9.887517 4.202938 3.432570e-36 1.126329e-31
98531 -7.238876 1.777650 3.164574e-14 5.191958e-10
98563 -6.604045 1.273889 1.566621e-11 1.713517e-07
59730  3.819932 2.195888 1.274089e-06 1.045167e-02
62812 -5.560674 3.334869 2.176178e-06 1.303127e-02
74124  3.649912 1.850164 2.382824e-06 1.303127e-02
```

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

```
[1] 7
```

```
> 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 43.811465 26.655983 19.017882 46.998468 39.9230884 37.4221165 0.000000
98531  4.225873  5.768871  6.358253  3.622917  5.2105922  8.7583677 0.000000
98563  5.354178  1.597777  3.292881  3.527016  5.0161244  2.9253018 0.000000
59730  2.491235  0.000000  0.000000  0.000000  0.1000937  0.2279911 9.291491
62812  0.000000 20.568083  9.100325 28.091814 15.2628598 39.2285079 0.000000
           DU.2     DU.3    DU.4     DU.5     DU.6
34127  0.000000 0.000000 0.00000 0.000000 0.000000
98531  0.000000 0.000000 0.00000 0.000000 0.000000
98563  0.000000 0.000000 0.00000 0.000000 0.000000
59730 11.614545 6.684968 4.54099 7.761995 4.925656
62812  2.146035 0.000000 0.00000 0.000000 0.000000

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


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

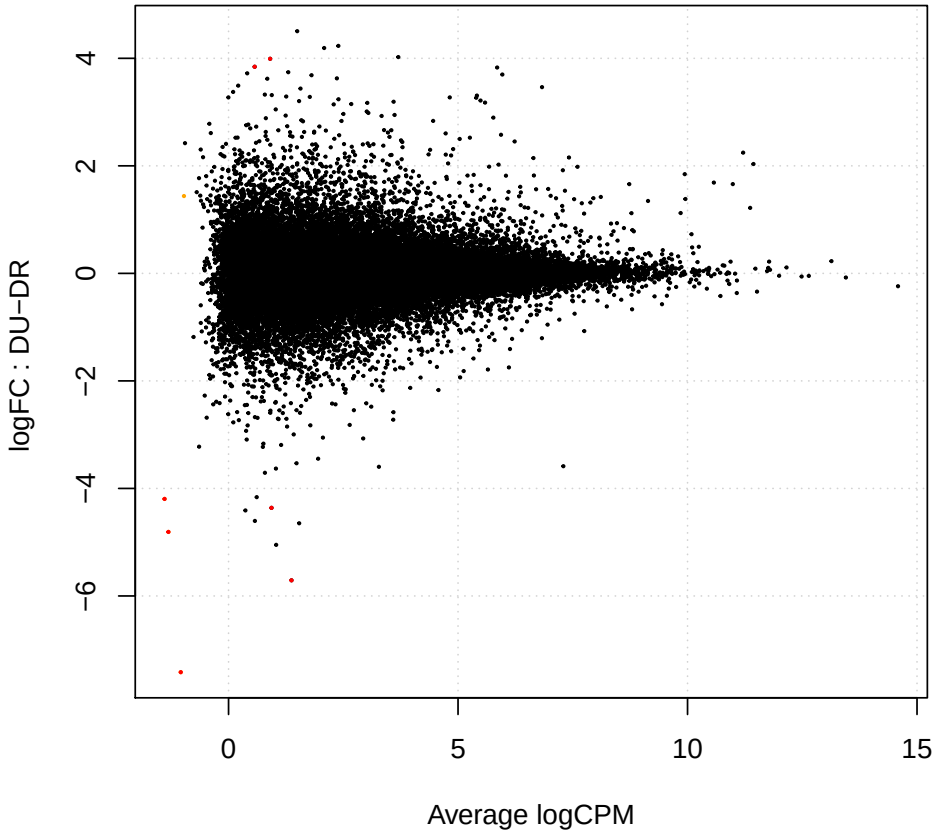

Figure 9: Plot smear of 12 DR and DU samples.

```
> topInfo = cbind(nc[rn,order(listcond)], tt$table)
> dev.off()
> for (i in 1:100){
+   gene = topInfo[i,1:12]
+   rep = 6
+   fact = 2
+   dat = data.frame(x=rep(1:fact, each=rep),y=t(gene),z=rep(1:rep, times = fact))
+   colnames(dat)=c("x","y","rep")
+   dat$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_FDR/", "Gene_", i, ".jpg", sep=""), height = 700, width = 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 filter
> 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 = col
> ########################## 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 = calcNormFactors(y)

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

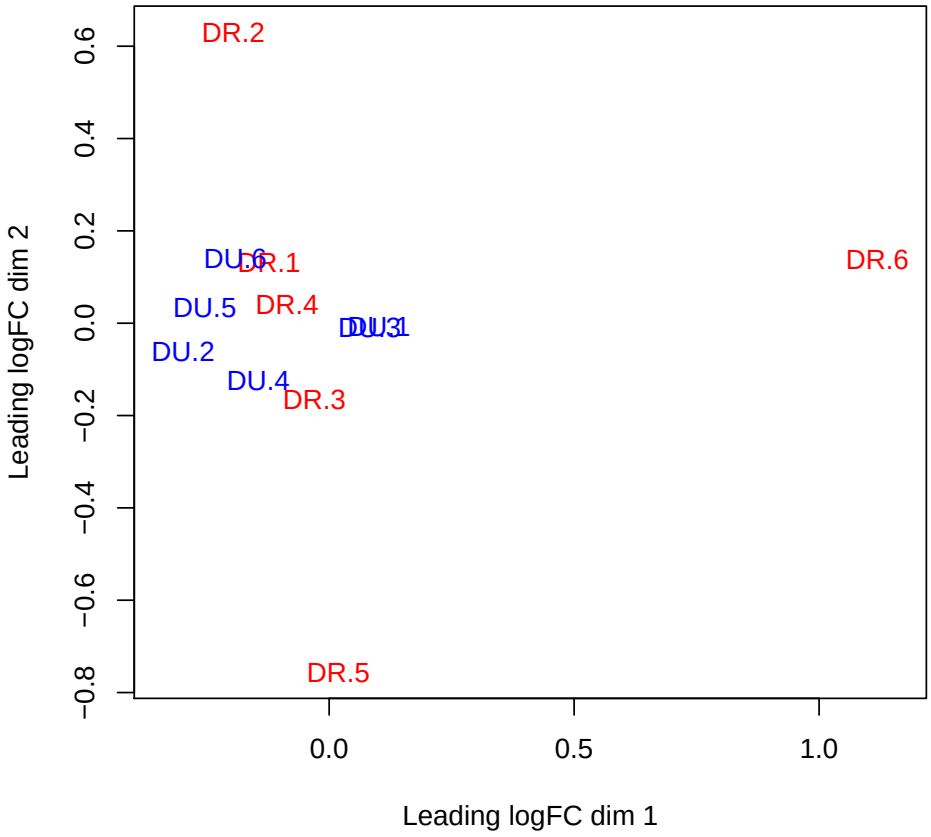

Figure 10: MDS plot 12 DR and DU samples.

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

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

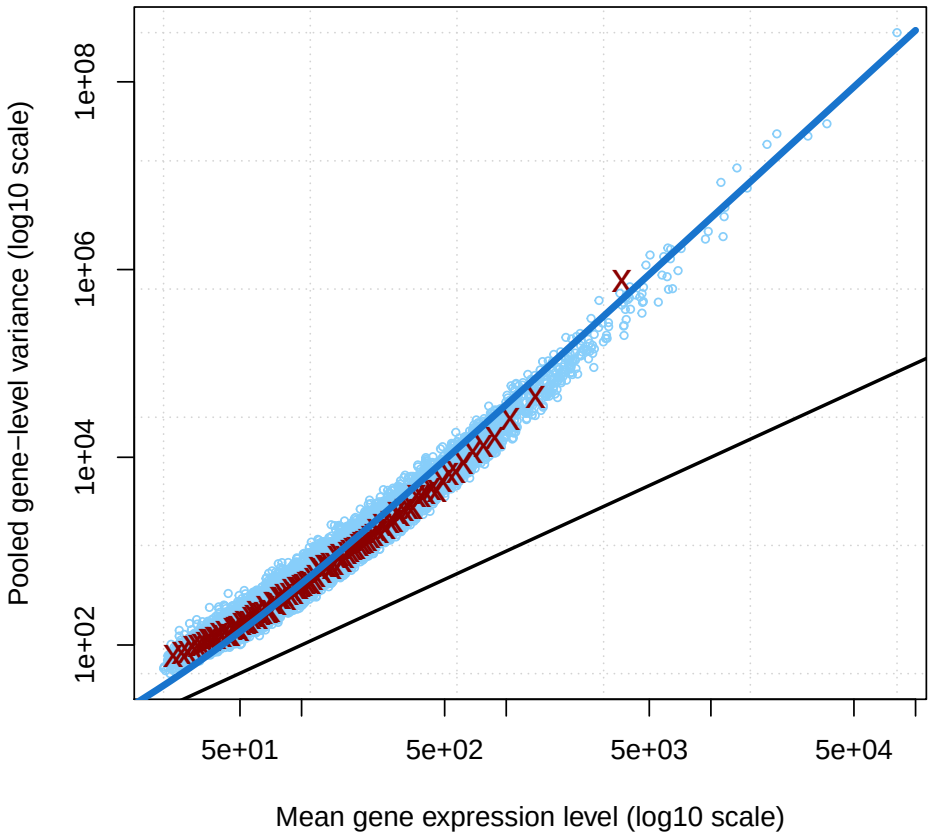

Figure 11: MeanVar plot of 12 DR and DU samples.

```
> plotBCV(y)
```

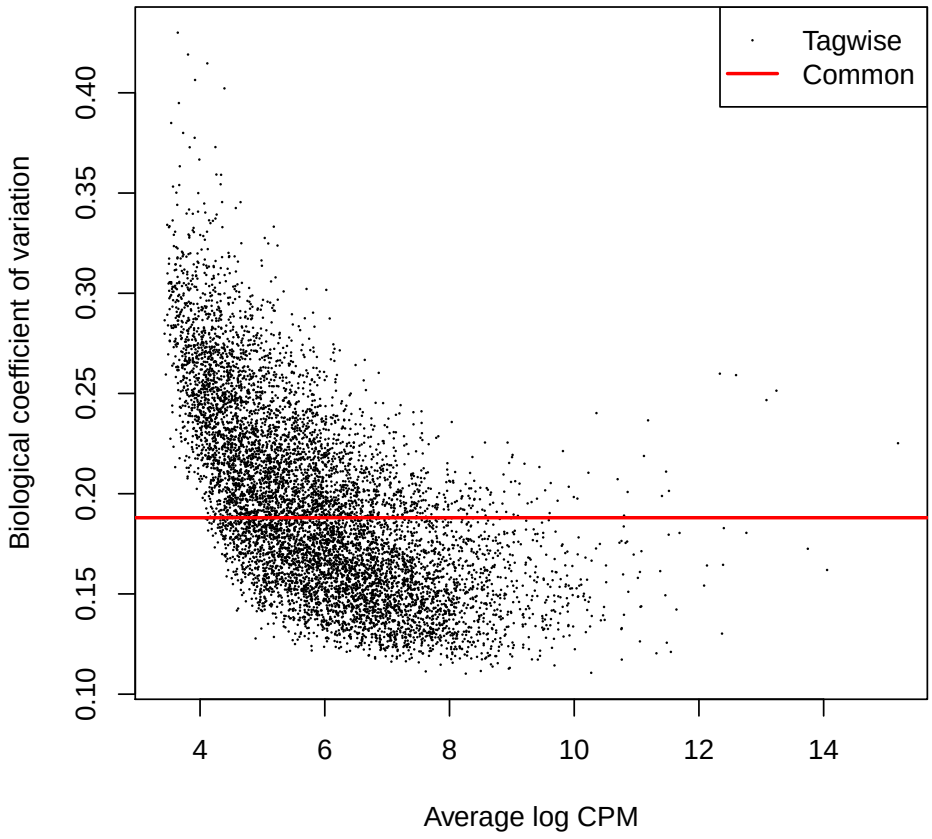

Figure 12: BCV plot of 12 DR and DU samples.

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

```
           logFC   logCPM       PValue        FDR
51268 -0.7547529 5.224029 4.668924e-06 0.03762219
34025  0.7662029 4.661857 1.160403e-04 0.46752632
39735 -0.5023912 6.602661 2.448115e-04 0.65756366
67325  0.7771275 4.264229 3.793611e-04 0.76422298
58430 -0.6494361 4.668783 7.329703e-04 0.80560812
38835 -0.4360339 7.119246 7.597844e-04 0.80560812
```

```
> # 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 48.87832  47.112531  48.25206  38.23963  44.47141  47.12474 18.36293
34025 17.94042  16.083303  14.59745  27.04349  13.91737  19.69846 36.33197
39735 97.06786 119.909046 100.95933 102.58930 137.75851 118.64444 70.74587
67325 13.37023   8.216799  16.63399  14.48471  17.27844  10.82616 17.55374
58430 27.95593  20.758229  31.91928  29.50589  30.85435  39.18040 17.12560
          DU.2     DU.3     DU.4     DU.5     DU.6
51268 25.85023 28.51817 24.77008 30.85262 34.68346
34025 27.84451 33.49669 30.25161 33.06801 23.04242
39735 96.94661 89.99831 68.60723 80.32141 70.91702
67325 19.80138 25.92561 30.20128 21.70982 23.96534
58430 22.63015 22.68491 18.12077 19.73620 13.76732

> # just for plotting purposes
> deg = rn[tt$table$FDR < .05] # Only 1


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

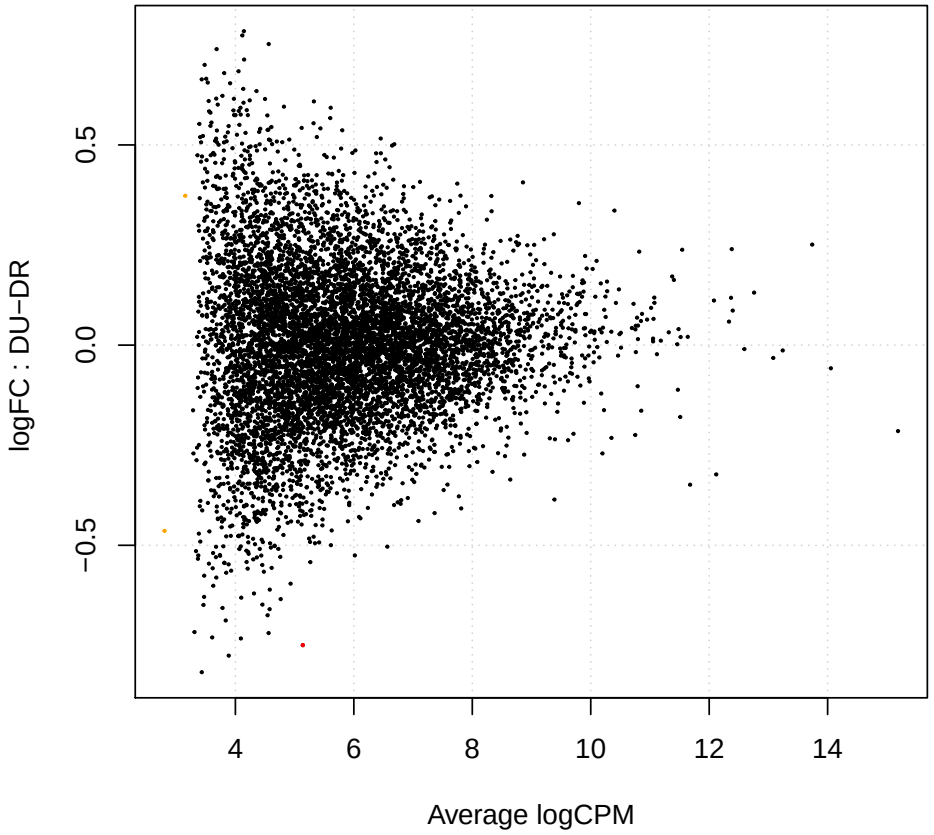

Figure 13: Plot smear of 12 DR and DU samples.

```
> topInfo = cbind(nc[rn,order(listcond)], tt$table)
> dev.off()
> for (i in 1:100){
+   gene = topInfo[i,1:12]
+   rep = 6
+   fact = 2
+   dat = data.frame(x=rep(1:fact, each=rep),y=t(gene),z=rep(1:rep, times = fact))
+   colnames(dat)=c("x","y","rep")
+   dat$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_FDR_LOESS/", "Gene_", i, ".jpg", sep=""), height = 700, wi
+   print(genePlot)
+   dev.off()
+ }

> rm(list=ls())
> load("All_wasp.rda")
> listcond = rep(c("DR","DU","F","NR","NU"),each= 6)
> # 157,691 genes
> y = DGEList(counts=countTable, 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
> ########################## 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+F.1+F.2+F.3+F.4+F
> 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)
> # 11,998 genes
> d2q1 = d2q1[which(sign(model$residuals) == 1),]
> d2q1 = d2q1[,1:(ncol(d2q1)-2)]
> # (filt 14,355 genes)
> filt = filt[,1:(ncol(filt)-2)]
> colnames(filt)=colnames(d2q1)
> # filt (30,670 genes)
> filt = rbind(filt,d2q1[which(sign(model$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 = calcNormFactors(y)
```

```
> ggparcoord(data.frame(y[[1]]), columns=1:ncol(y[[1]]), alphaLines=0, boxplot=TRUE, scale="globalminmax
```

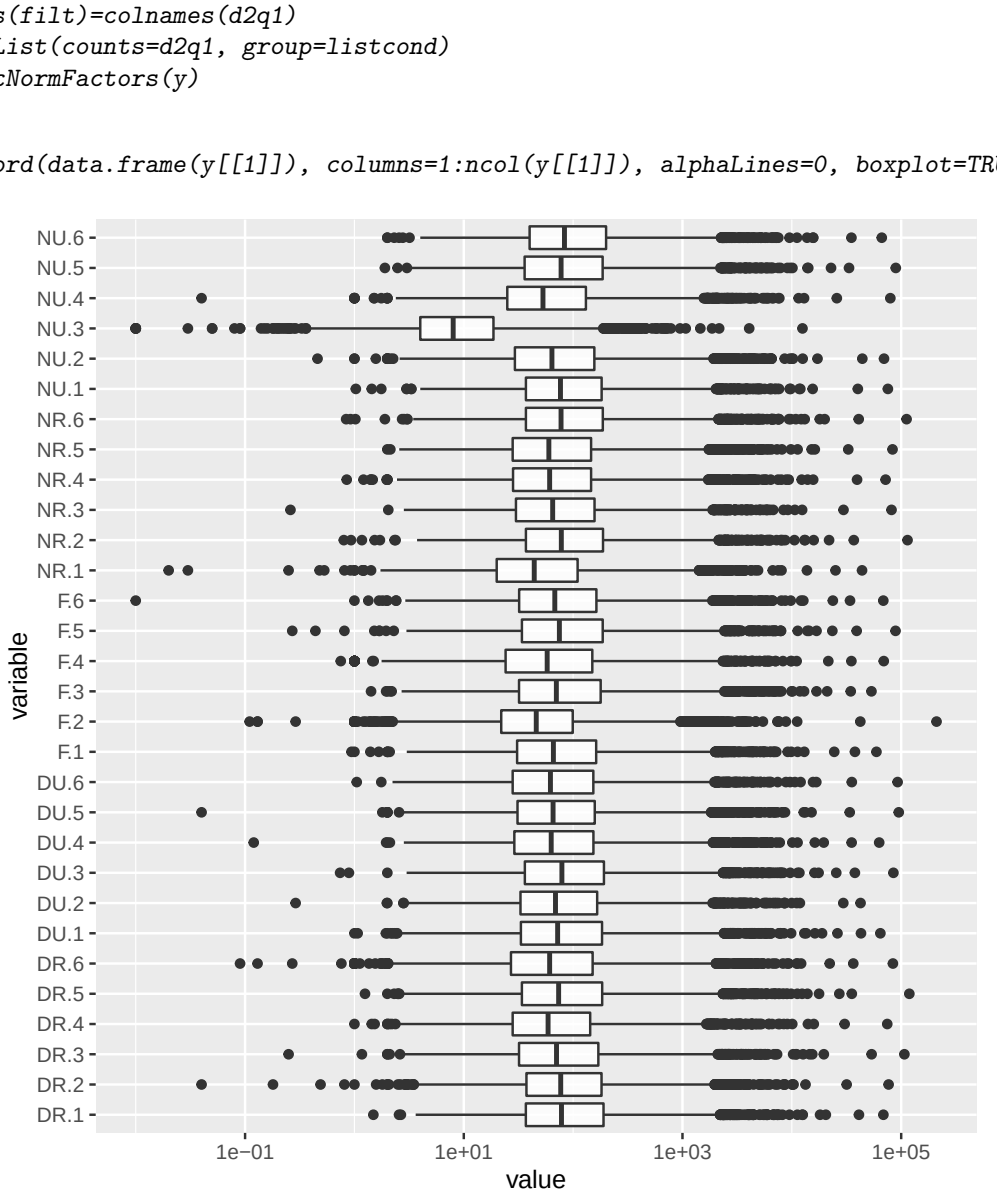

Figure 14: This is all data filtered on CPM and filtered by Loess, then normalized.

```
> ggparcoord(data.frame(y[[1]]), columns=1:6, scale="globalminmax", alphaLines = 0.07)
```

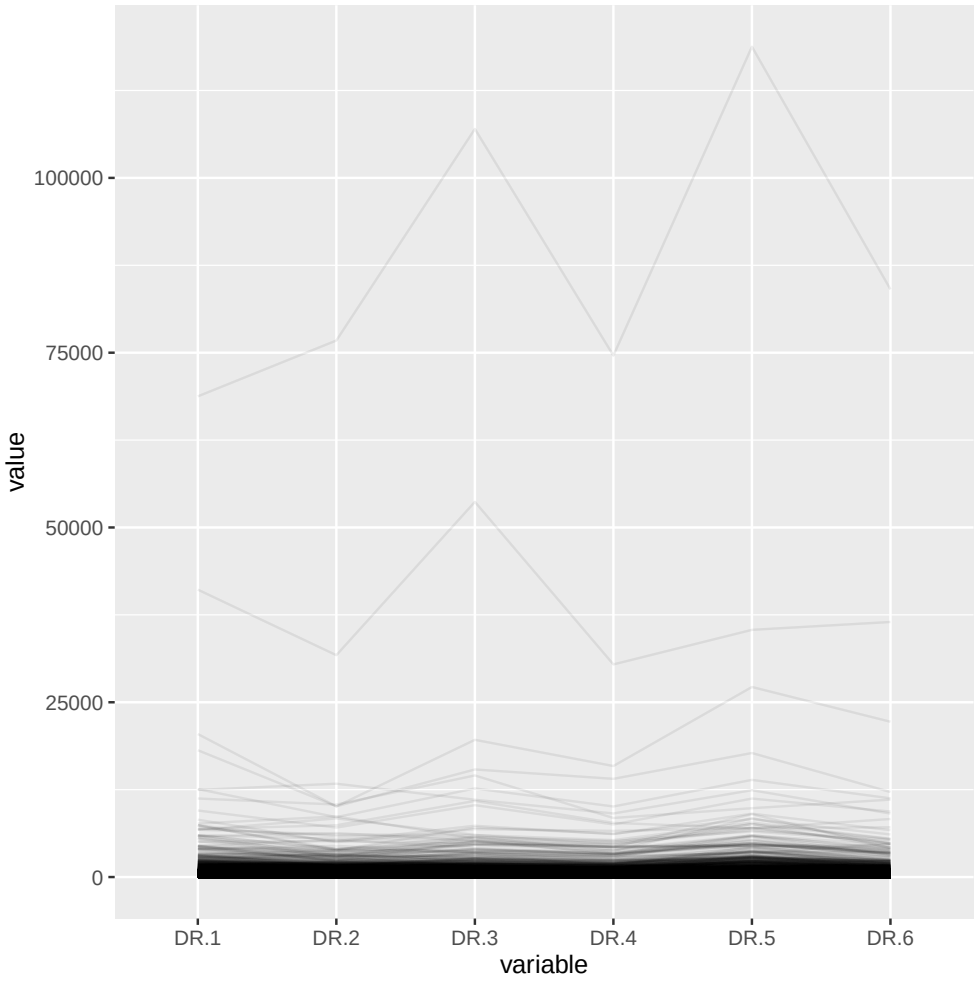

Figure 15: DR PCP

```
> ggparcoord(data.frame(y[[1]]), columns=7:12, scale="globalminmax", alphaLines = 0.07)
```

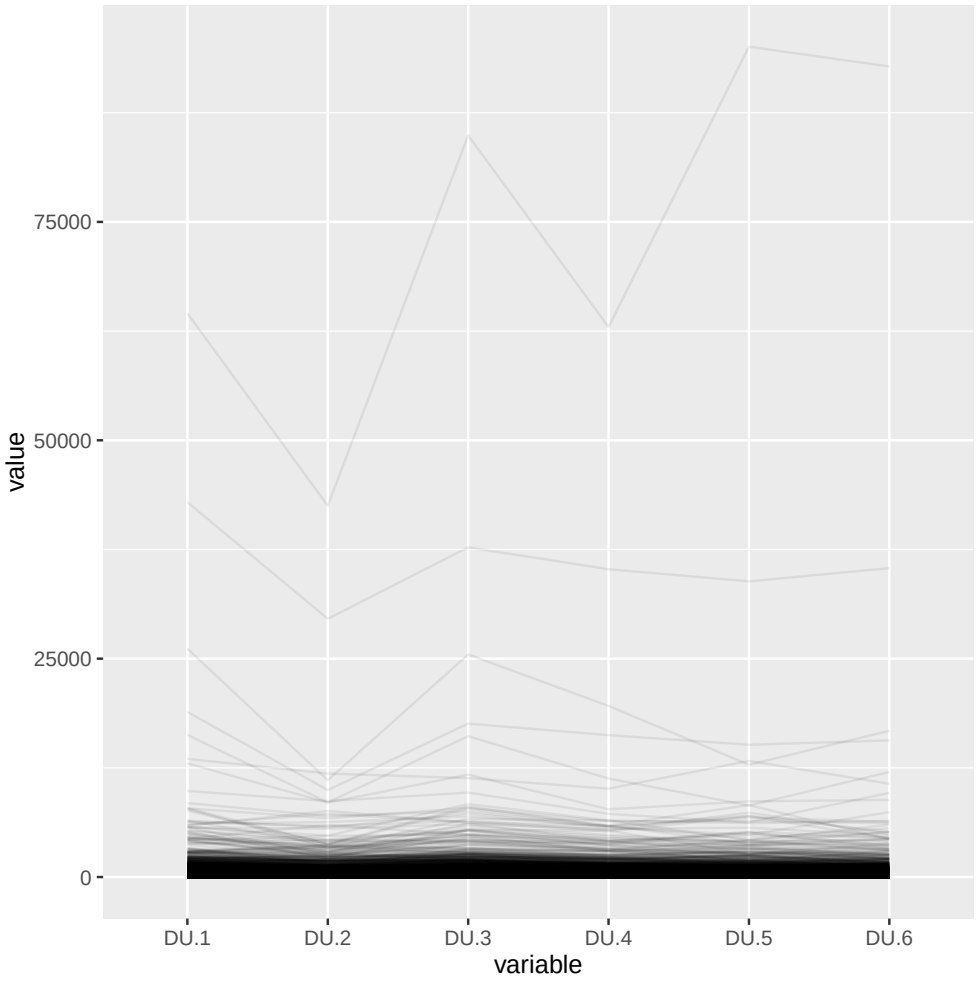

Figure 16: DU PCP

```
> ggparcoord(data.frame(y[[1]]), columns=13:18, scale="globalminmax", alphaLines = 0.07)
```

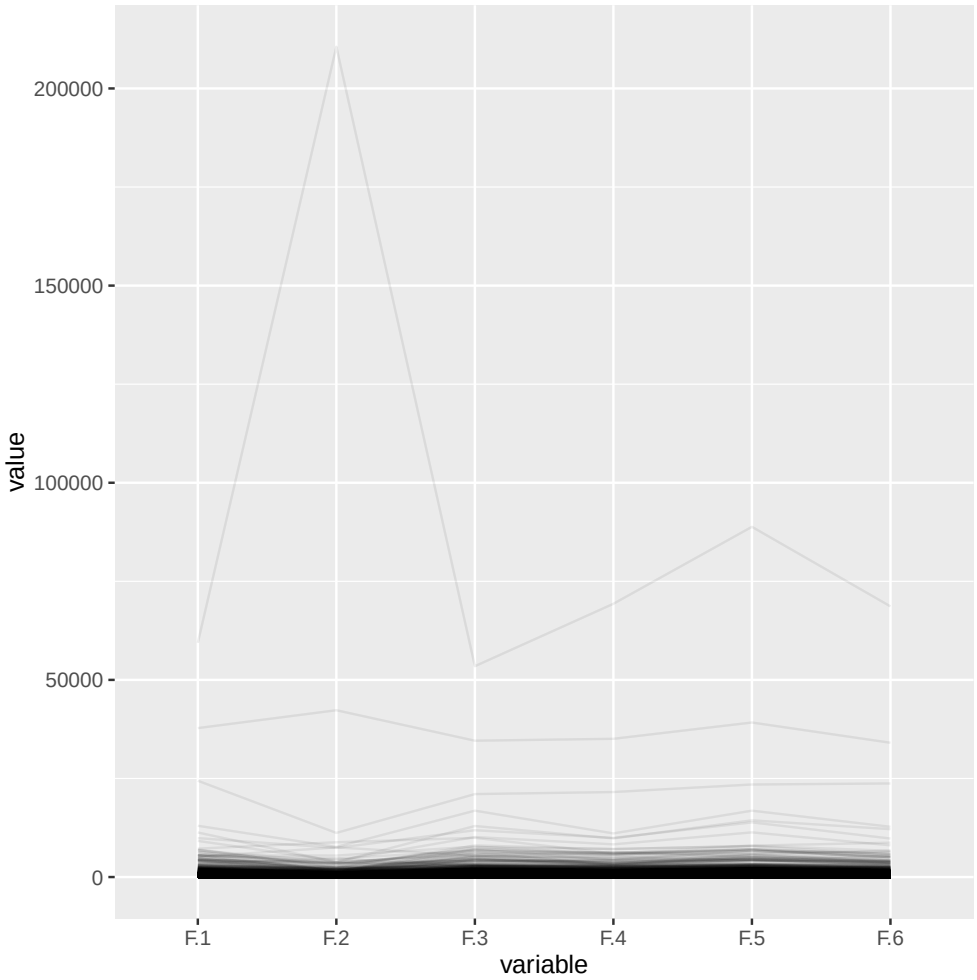

Figure 17: F PCP

```
> ggparcoord(data.frame(y[[1]]), columns=19:24, scale="globalminmax", alphaLines = 0.07)
```

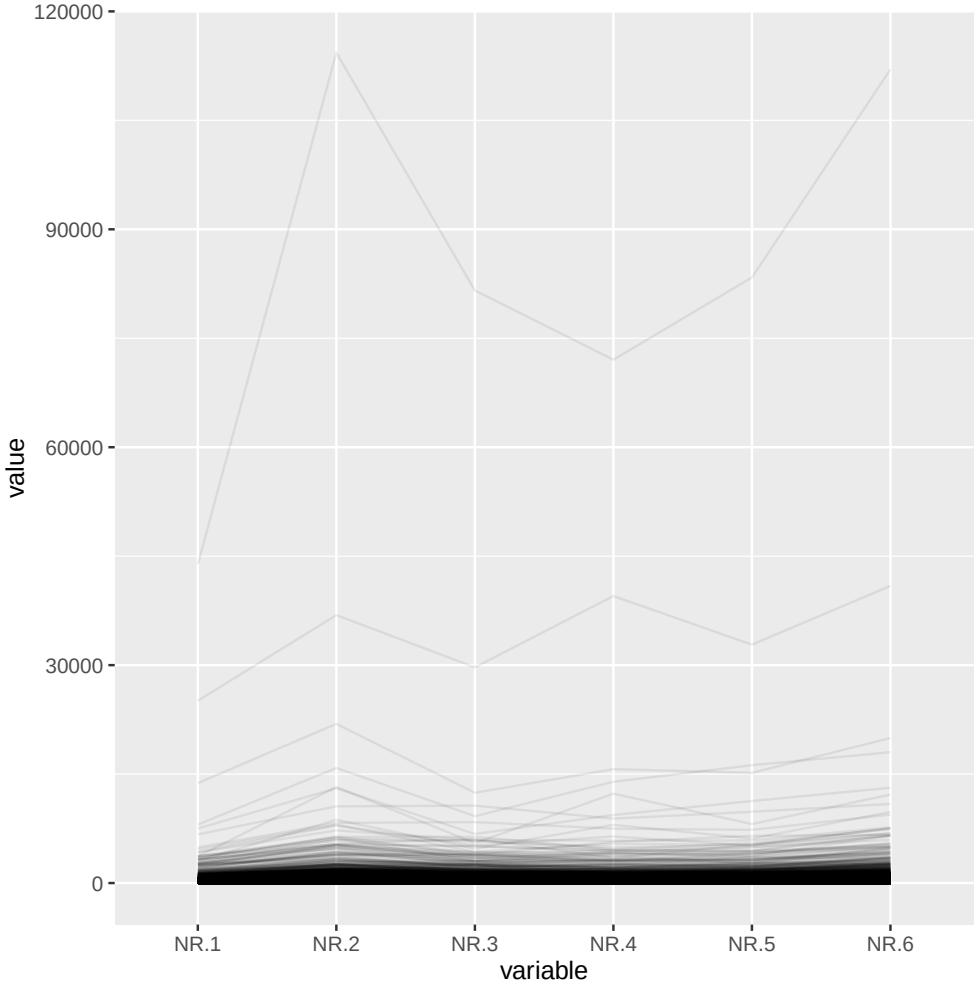

Figure 18: NR PCP

```
> ggparcoord(data.frame(y[[1]]), columns=25:30, scale="globalminmax", alphaLines = 0.07)
```

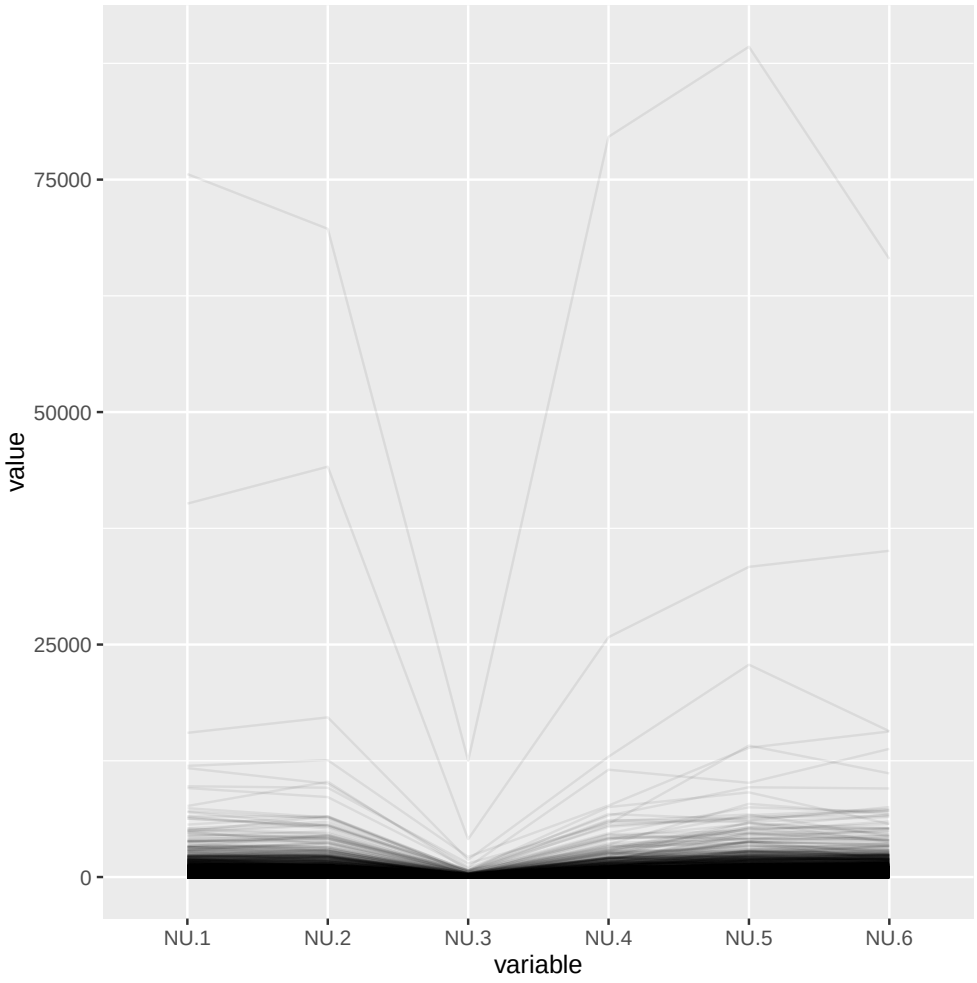

Figure 19: NU PCP