Use MDS plots to visualize how samples cluster

Ann Loraine

2023-10-06

Prelude

Define variables:

```
assembly="SL5"
```

Load custom code:

```
source("Common.R")
```

Introduction

To assess sanity-check sample identity and notice possible differential expression patterns, it is useful to perform clustering analysis, in which we plot samples on a plane and look for clusters of points that share the same experimental factor value.

This helps us

- detect possible sample switching problems
- identify comparisons will large effect sizes

In this R Markdown document, we ask:

- How do sample libraries cluster?

Results

Load library with useful data structures and functions:

```
library(edgeR)
```

Read "raw counts" data and sort columns by sample name:

```
counts = getCounts(assembly=assembly,keep_description = F)
o = order(colnames(counts))
counts = counts[,o]
```

Create `edgeR` data structure for convenience:

```
big_DGEList=DGEList(counts=counts,remove.zeros = TRUE)
```

There were 15,253 genes with zero counts that were removed from consideration.

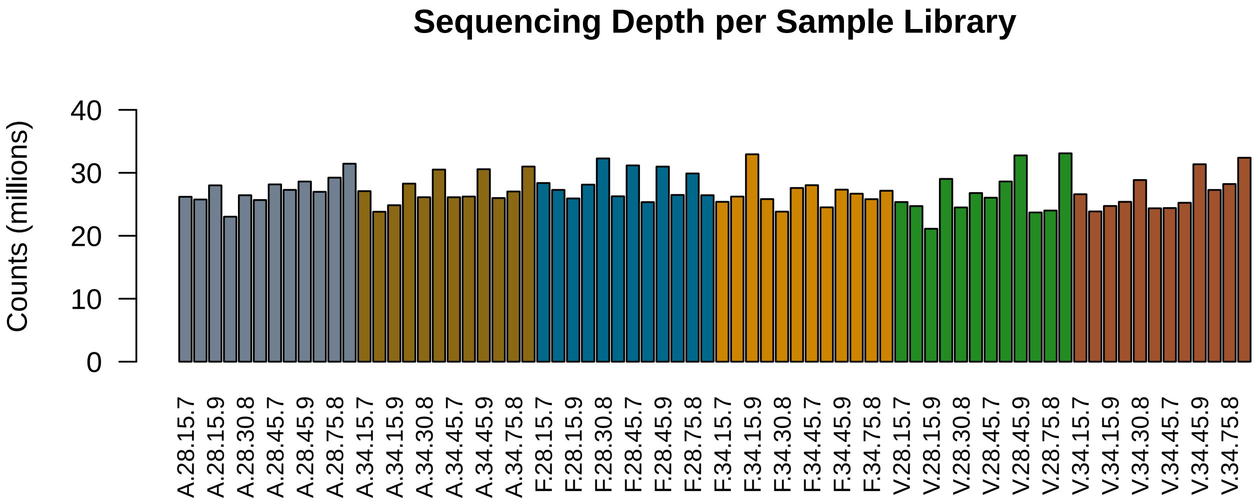
Configure color scheme for plots:

```
the_colors = getSampleColors()
```

For the entire data set with 72 samples, there are 6 unique colors.

Draw a bar plot showing sequencing coverage by sample:

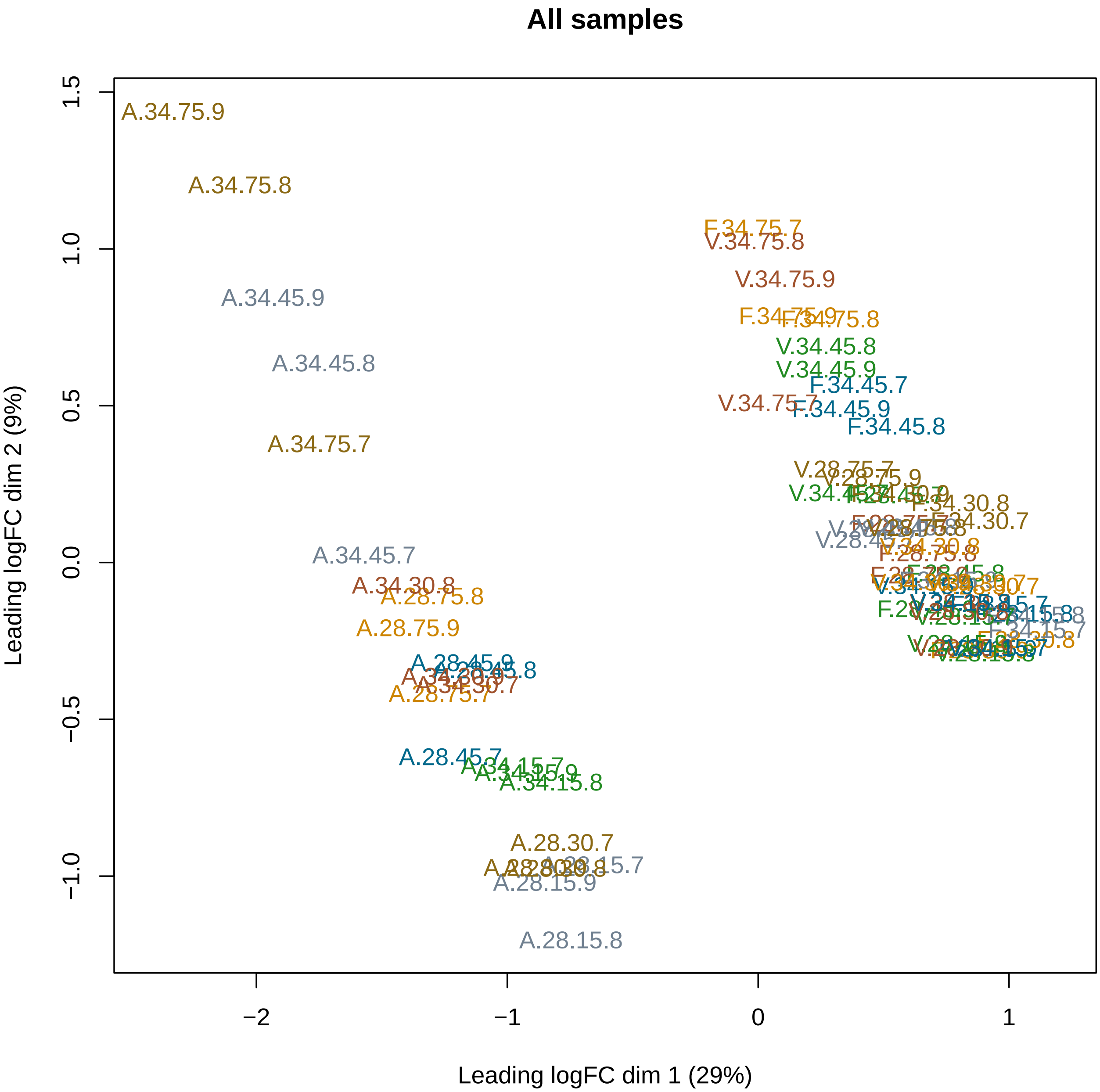
```
main="Sequencing Depth per Sample Library"
ylab="Counts (millions)"
libsizes=big_DGEList$samples$lib.size/10**6
mindepth=round(min(libsizes))
maxdepth=round(max(libsizes))
names(libsizes)=rownames(big_DGEList$samples)
par(mar=c(8,4.1,4.1,2.1))
barplot(libsizes,
        col=the_colors[rownames(big_DGEList$samples)],
        las=2,
        main=main,ylab=ylab,cex.names = 0.8,
        ylim=c(0,maxdepth+10))
```



The preceding figure shows that sequencing depth was similar across libraries. There were between 21 and 33 million counts per sample, across 72 sample libraries.

Use multi-dimensional scaling to examine sample relatedness, showing all samples in the same plot:

```
plotMDS(big_DGEList,col=the_colors,main="All samples")
```



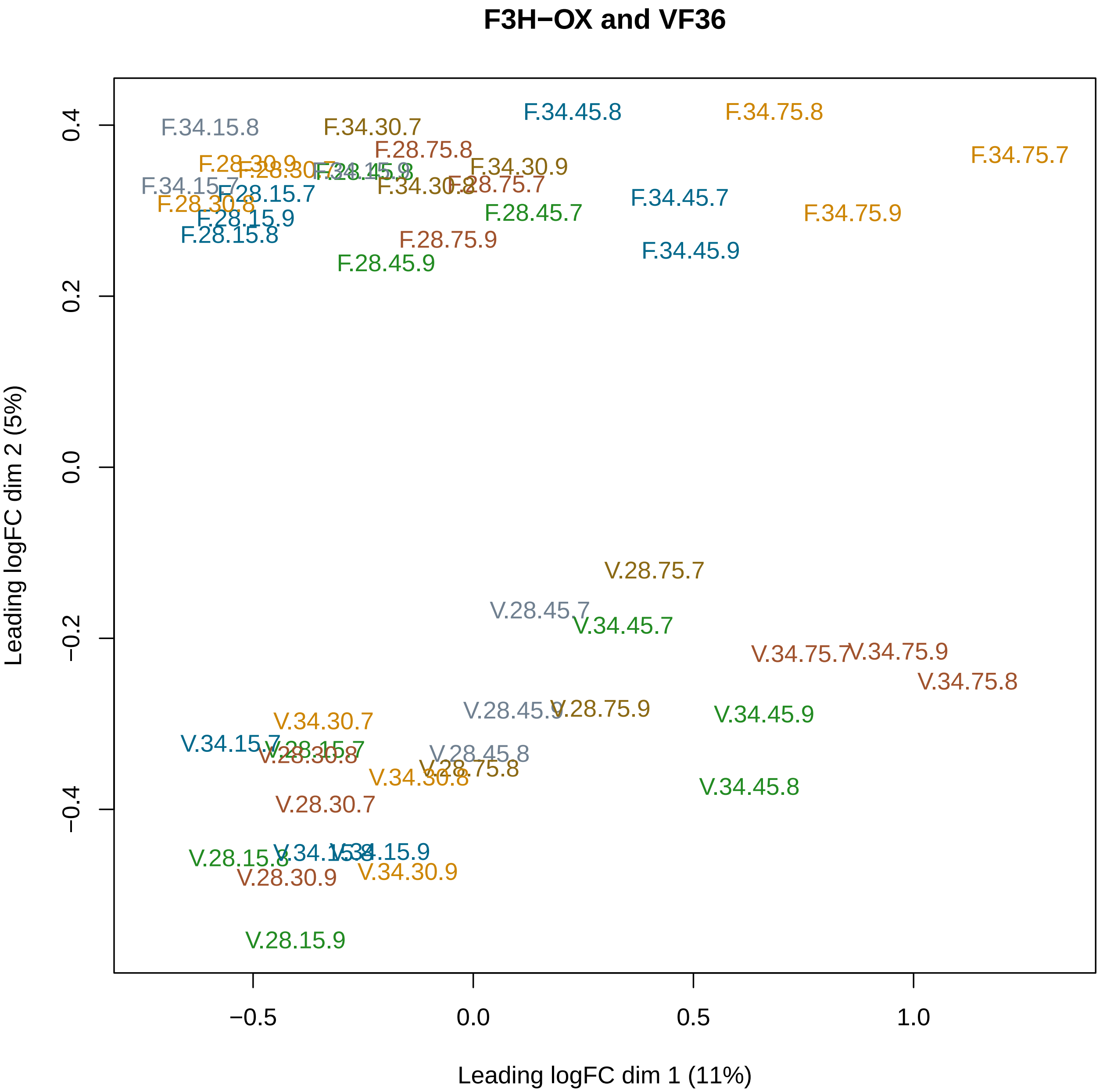
There were two clusters, separated by genotype. ARE genotype ("A") samples cluster together and the other two gentypes (VF36, "V" and F3H-OX3, "F") cluster together.

F3H and VF36 cluster together, but may separate if ARE is removed. To observe possible separation between F3H and VF36, create an MDS plot with F3H and VF36 only:

```
indexes = c(grep("F",names(counts)),
            grep("V",names(counts)))
big_DGEList=DGEList(counts=counts[,indexes],remove.zeros = T)
```

```
## Removing 16709 rows with all zero counts
```

```
sample_colors = the_colors[indexes]
plotMDS(big_DGEList,col=sample_colors,main="F3H-OX and VF36")
```



As shown in the previous plot, VF36 and F3H-OX cluster into two separate clusters by genotype.

To expose trends with respect to treatment and treatment duration, create a plot with one genotype per plot.

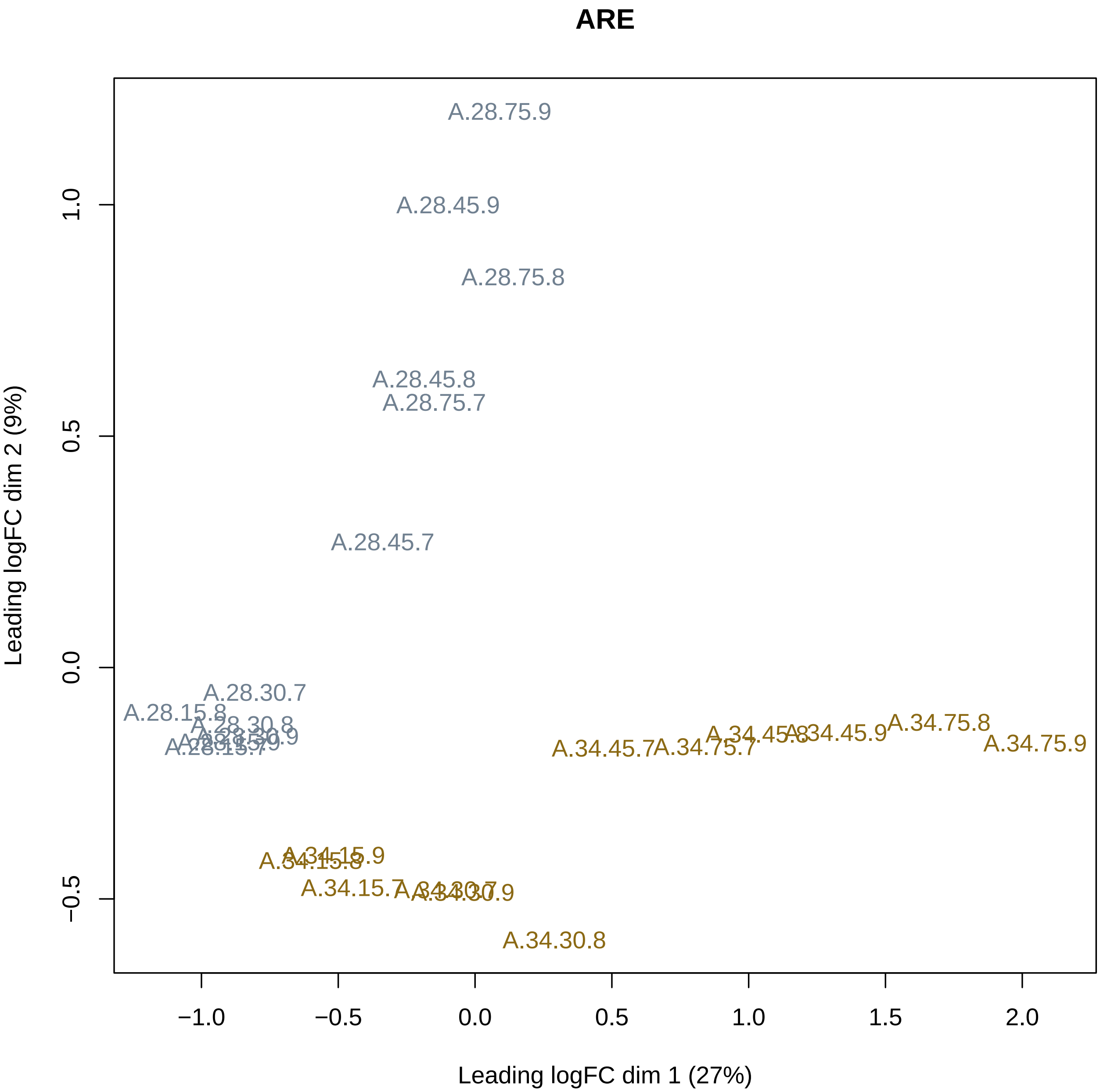
Define a function to cluster samples by genotype:

```
clusterByGenotype = function(genotype,counts,the_colors) {
  toks = strsplit(names(counts),"\\.")
  ggenotype=sapply(toks,function(x){x[[1]]})
  indexes = ggenotype==genotype
  little_DGEList=DGEList(counts[,indexes],
                         remove.zeros = TRUE)
  sample_colors=the_colors[row.names(little_DGEList$samples)]
  display_genotype = ""
  if (genotype == "A") {
    display_genotype = "ARE"
  }
  if (genotype == "F") {
    display_genotype = "F3H-OX"
  }
  if (genotype == "V") {
    display_genotype = "VF36"
  }
  main = paste(display_genotype)
  plotMDS(little_DGEList,col=sample_colors,main=main)
}
```

ARE:

```
clusterByGenotype("A",counts,the_colors)
```

```
## Removing 16862 rows with all zero counts
```

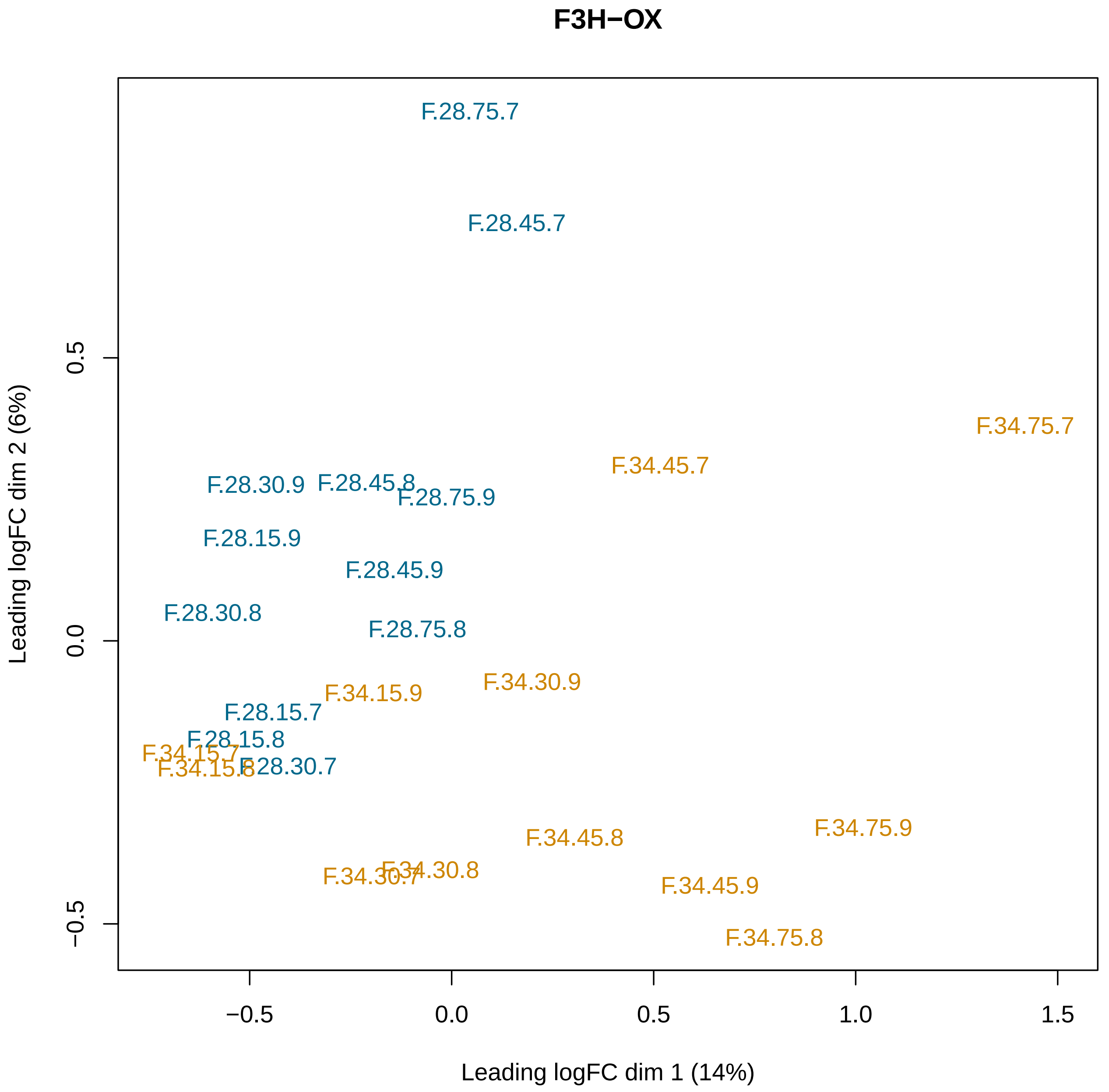


The preceding plot shows that heat-stressed samples cluster together and non-heat-stressed controls sample together. Also, the 45 and 75 minute treatments cluster with each other.

F3H-OX:

```
clusterByGenotype("F",counts,the_colors)
```

```
## Removing 17998 rows with all zero counts
```

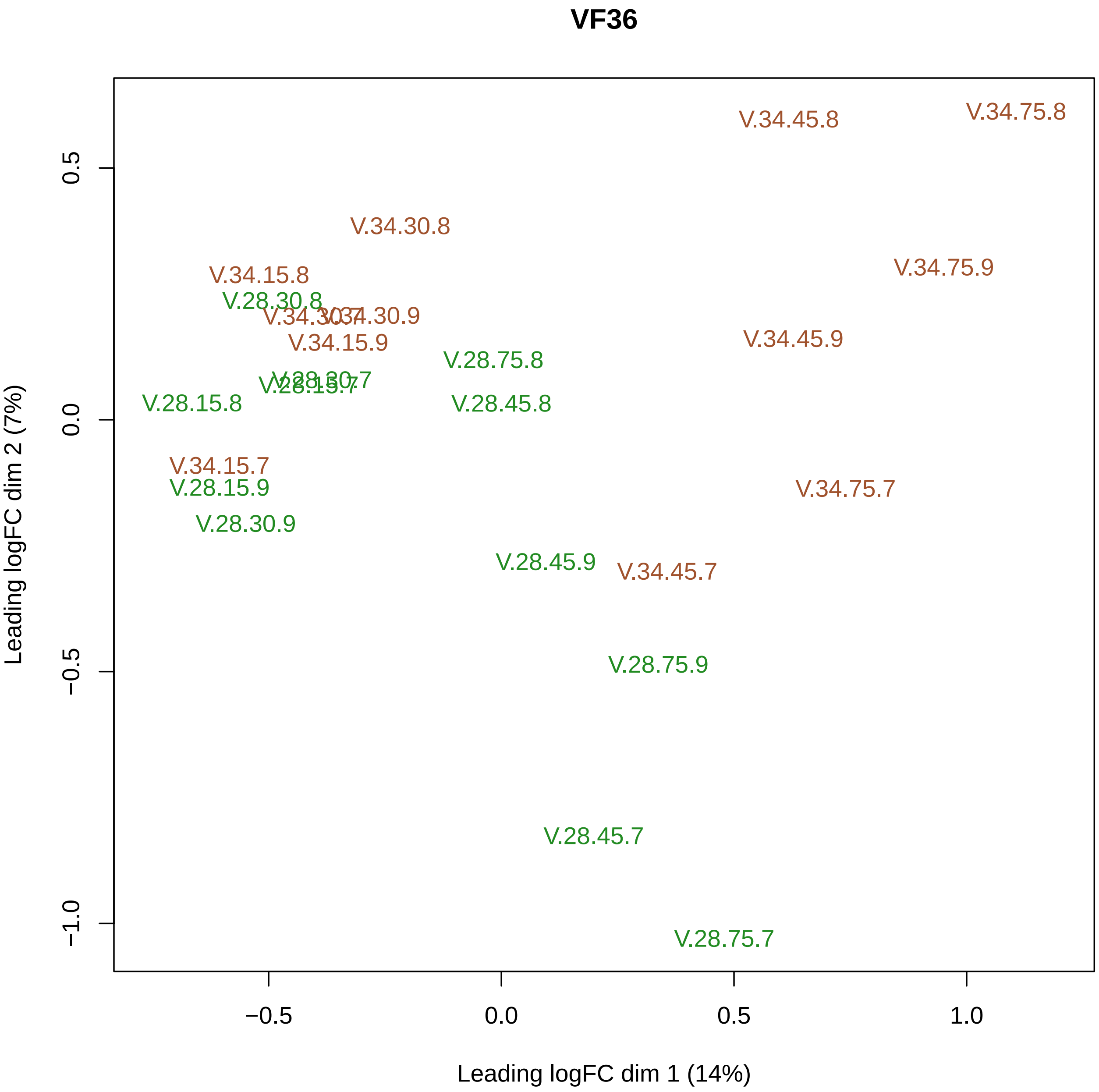


The preceding plot shows that samples cluster by temperature, with some separation by duration.

VF36:

```
clusterByGenotype("V",counts,the_colors)
```

```
## Removing 18133 rows with all zero counts
```



The preceding plot shows that samples cluster by temperature, with some separation by duration.

Next, let's examine the degree to which treatment enables separation of samples within a genotype and time point.

First, create a function to cluster samples from the same time point and genotype:

```
clusterByTimeGenotype = function(timepoint,
                                 genotype,
                                 counts,
                                 the_colors) {
  toks = strsplit(names(counts),"\\.")
  ggenotype=sapply(toks,function(x){x[[1]]})
  time=sapply(toks,function(x){x[[3]]})
  indexes = ggenotype==genotype & time==timepoint
  little_DGEList=DGEList(counts[,indexes],
                         remove.zeros = TRUE)
  sample_colors=the_colors[row.names(little_DGEList$samples)]
  display_genotype = ""
  if (genotype == "A") {
    display_genotype = "ARE"
  }
  if (genotype == "F") {
    display_genotype = "F3H-OX"
  }
  if (genotype == "V") {
    display_genotype = "VF36"
  }
  main = paste(display_genotype,
               "at",
               timepoint,
               "minutes")
  plotMDS(little_DGEList,col=sample_colors,main=main)
}
```

Cluster genotypes and time points separately:

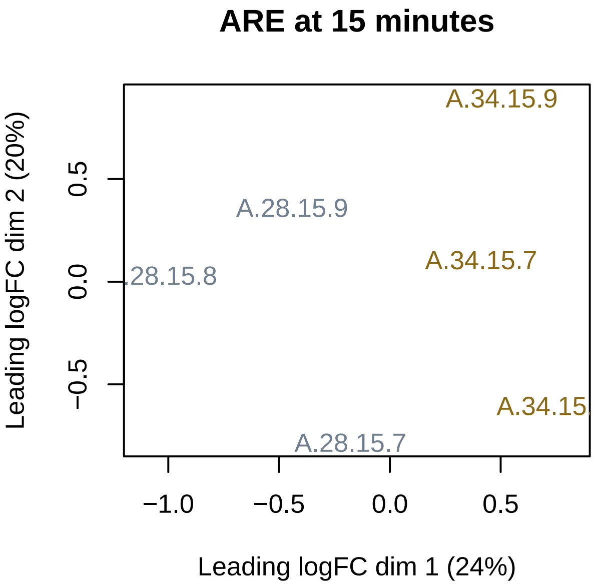
The following plots show temperature-based clustering patterns between samples based on temperature, within genotypes and time points.

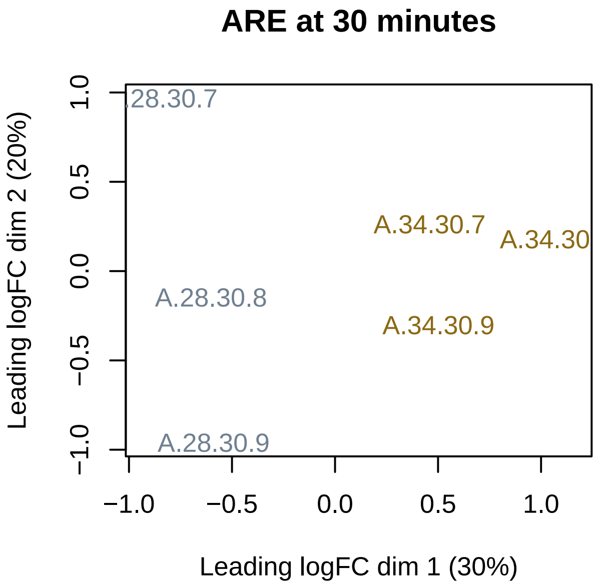
Identify timepoints:

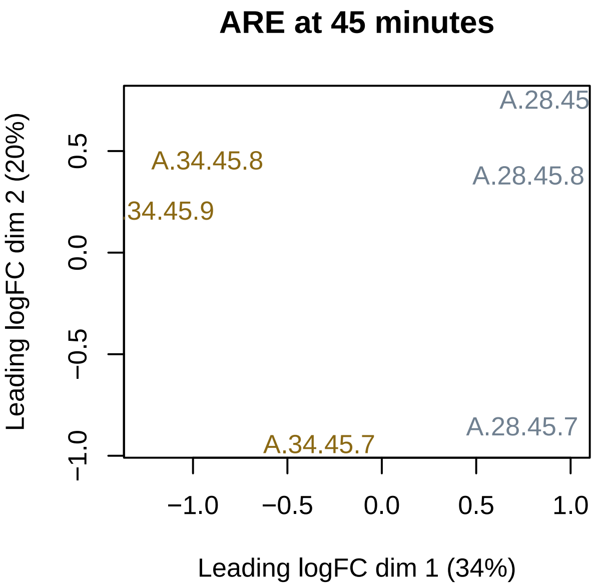
```
toks = strsplit(names(counts),"\\.")
timepoints = unique(sapply(toks,function(x){x[[3]]}))
```

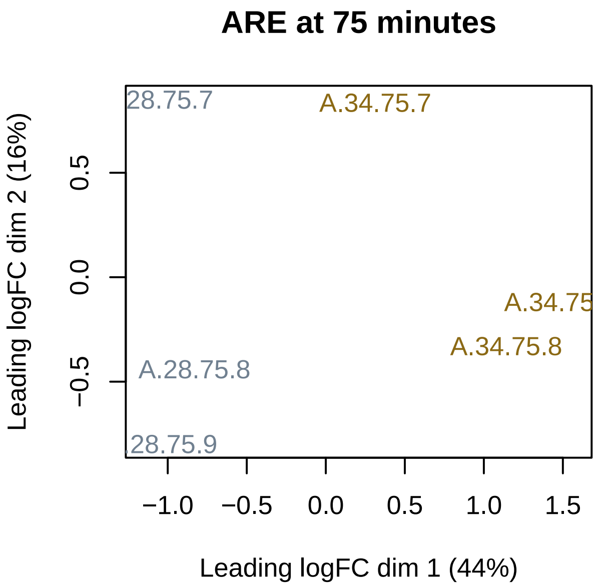
Cluster ARE samples:

```
ncols=2
num_plots = length(timepoints)
nrows = num_plots/ncols
par(mfrow=c(nrows,ncols))
genotype="A"
for (timepoint in timepoints) {
    dge_list = clusterByTimeGenotype(timepoint,genotype,
                                     counts,the_colors)
}
```





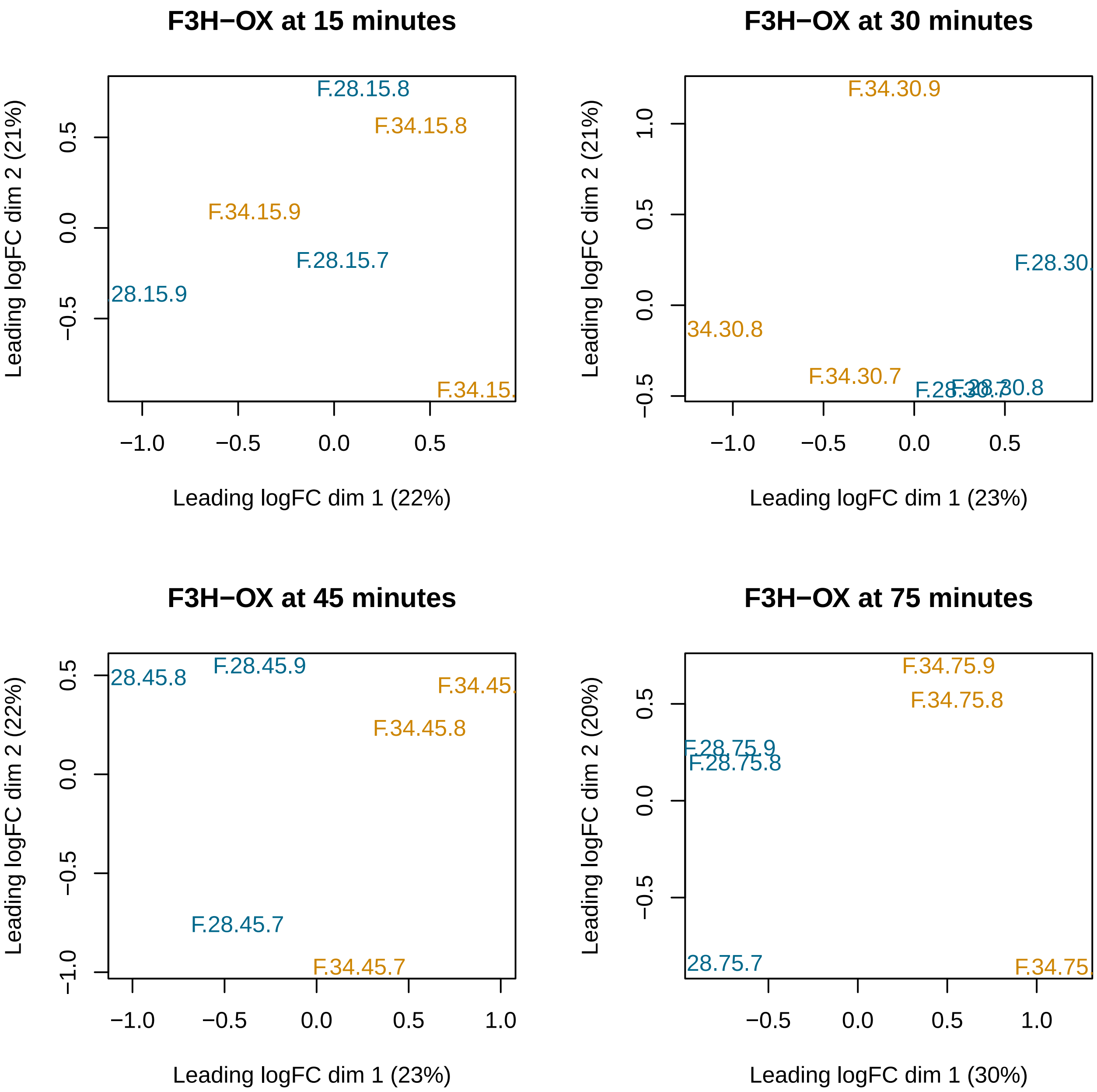




The ARE (mutant, non-transgenic) genotype samples show temperature-based clustering for all time points.

Cluster F3H samples:

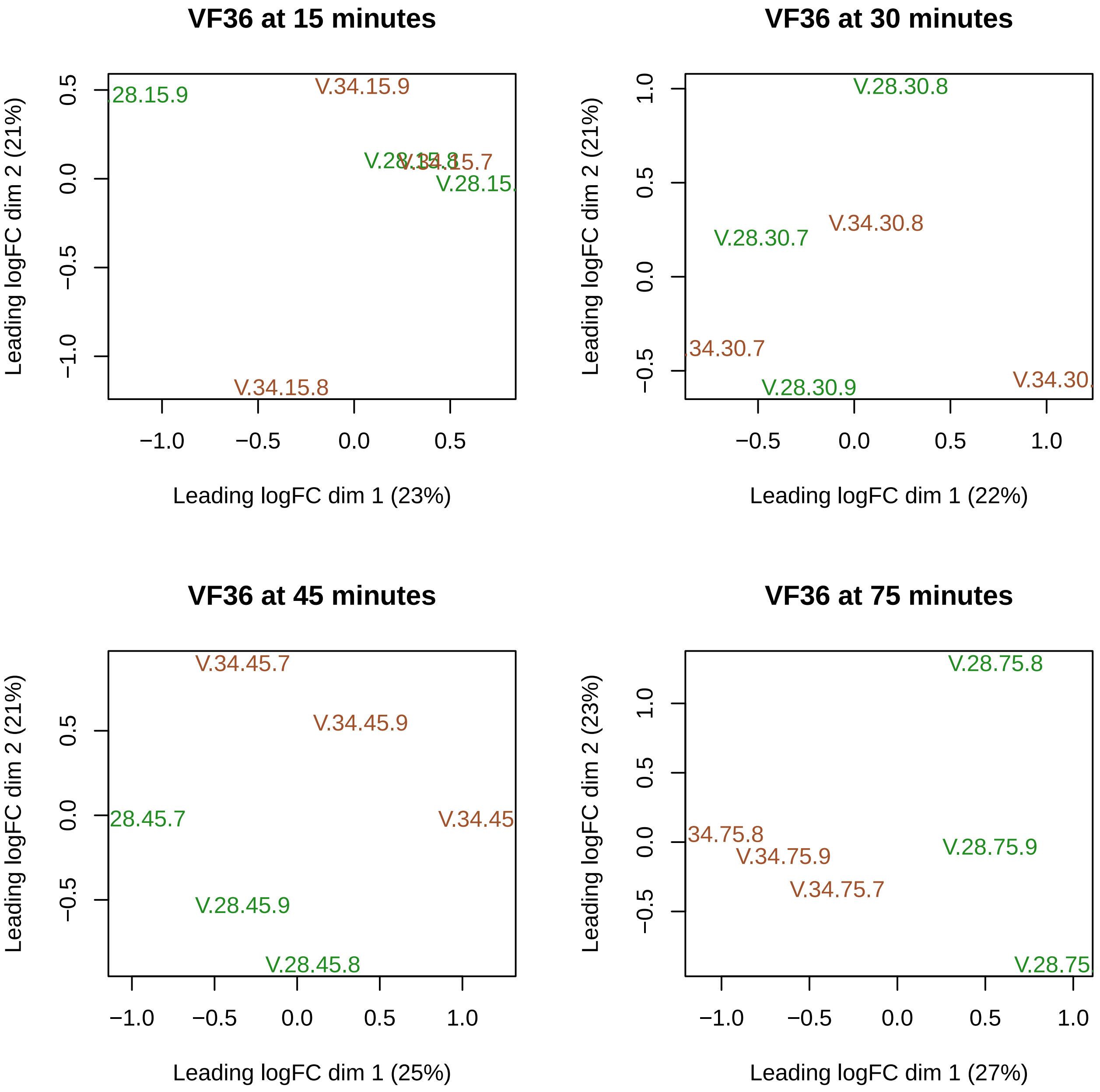
```
par(mfrow=c(nrows,ncols))
genotype="F"
for (timepoint in timepoints) {
    dge_list = clusterByTimeGenotype(timepoint,genotype,
                                     counts,the_colors)
}
```



The F3H overexpression genotype (labeled F3H-OX) samples show separation by temperature for the 30 minute, 45 minute, and 75 minute time points, but not the 15 minute time point.

Cluster VF36 samples:

```
par(mfrow=c(nrows,ncols))
genotype="V"
for (timepoint in timepoints) {
    dge_list = clusterByTimeGenotype(timepoint,genotype,
                                     counts,the_colors)
}
```



The VF36 (wild-type, non-transgenic) genotype samples show separation for the 45 and 75 minute time points, but not the 15 and 30 minute time points.

Discussion

The MDS plot in which all 72 samples appeared contained two clusters, one with ARE samples and the other with VF36 and F3H-OX samples. Co-clustering of VF36 and F3H-OX samples makes sense because F3H-OX is a transgenic VF36 line containing an F3H over-expression transgene.

The MDS plots showing individual genotypes and single time points showed obvious clustering with respect to temperature. The later time points exhibit clustering in all three genotypes.

Two of three genotypes, F3H-OX and VF36, showed no clustering in their 15 minute time point.

One of the genotypes, VF36, showed no clustering in the 30 minute time point.

These results indicate that the renaming of samples was done correctly. We have identified the proper labels for all samples in the experiment.

The time dependence aspect of the clustering results foreshadows results we may obtain in the next step of analyzing these data, where we will attempt to identify differentially expressed genes and quantify the degree of differential expression. We might expect fewer detectable differences in expression in earlier time points, due to the shorter duration of the treatment. The MDS plots exhibiting less treatment-dependent clustering in earlier versus the later time points supports this expectation for differential expression analysis.

Conclusion

- Clustering patterns revealed by MDS (multi-dimensional scaling) analysis strongly suggests that the sample renaming strategy was correct.
- Clustering patterns suggest that we will observe more differential expression in later time points than in earlier time points, reflecting treatment duration.
- Clustering patterns indicate that ARE samples are very different from the others and that ARE versus F3H-OX/VF36 differences are likely to be much greater and more numerous than temperature-based differences.