DE_Cyt_vs_Nuc

Núria Rivera Brugués

2023-10-02

```
counts_brain_mouse<-read.delim("./GSE159919_PolyA_read_counts.txt")
counts_brain_mouse<-counts_brain_mouse[,-c(2:6)]
rownames(counts_brain_mouse)<-counts_brain_mouse[,1]
counts_brain_mouse<-counts_brain_mouse[,-1]
colnames(counts_brain_mouse)<-c("ESC_Ch1","ESC_Ch2","ESC_Ch3","ESC_N1","ESC_N2","ESC_N3","ESC_C1","ESC_

targets_brain_human<-read.delim("./SraRunTable.txt", header=T, sep=",")
targets_brain_human<-targets_brain_human[, c(1,13,24:26)]
info<-targets_brain_human
Sub1<-rep( c("Ch1", "N1", "C1"),each=1, len= 3)
Sub2<-rep( c("Ch2", "N2", "C2"),each=1, len= 3)
Sub3<-rep( c("Ch3", "N3", "C3"),each=1, len= 3)
info$Subgroup<-c(Sub1,Sub2,Sub3)
info$Fraction<-rep(c("Chr", "Nuc", "Cyto"), each=1, len= 9)
info$Source<-c(rep("ESC",9), rep("NPC",9), rep("Ctx",9))
info$Sample<-paste(info$Source,info$Subgroup, sep="_")
info$Group<-paste(rep(c("Chr", "Nuc", "Cyto"), each=1, len= 9), info$Source, sep="")
write.csv(info, "./info_brain.csv")
```

```
# redueixo el dataframe (nomes Cyt vs Nuc del neurones corticals primaries)
counts_primarycorticalneuron<-counts_brain_mouse[,-c(1:21)]
info<-info[-c(1:18),]
info<-info[-c(1,4,7),]
info$Subgroup<-c(1,1,2,2,3,3)
rownames(info)<-info$Sample
barcode=factor(info$Sample)
subgroup=factor(info$Subgroup)
group=factor(info$Group)
source=factor(info$Source)
fraction<-factor(info$Fraction)
c<-info[c(2,4,6),]
n<-info[c(1,3,5),]
info<-rbind(n,c)
subgroup=factor(info$Subgroup)
group=factor(info$Group)
source=factor(info$Source)
fraction<-factor(info$Fraction)

View(info)
```

```
y=DGEList(counts_primarycorticalneuron)
isexpr <- rowSums(cpm(y) > 1) >= 3
y=y[isexpr,keep.lib.size=FALSE]
y=calcNormFactors(y)
y$samples
```

```
##        group lib.size norm.factors
## Ctx_N1     1 17429224    0.9700342
## Ctx_N2     1 15973042    0.9809899
## Ctx_N3     1 23877953    0.9579471
## Ctx_C1     1 15584676    1.0241706
## Ctx_C2     1 22267101    1.0289512
## Ctx_C3     1 24501987    1.0409739
```

```
dim(y)
```

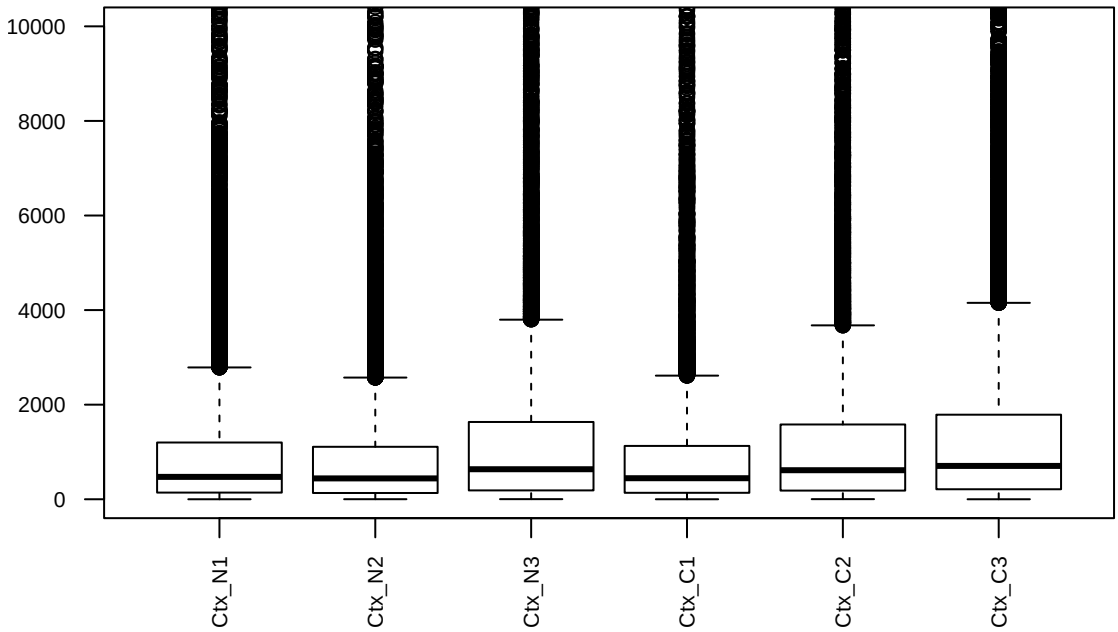
```
## [1] 13456     6
```

Exploración de los datos

Una vez descartados los genes poco expresados y con los recuentos almacenados en un objeto DGEList, podemos‘proceder a realizar algunos gráficos exploratorios para determinar si los datos aparentan buena calidad y/o si presentan algun problema.

Distribución de los contajes

```
boxplot(y$counts, col = y$samples$cols, las = 2, cex.axis = 0.7,
    main = "Contajes normalizados", ylim = c(0, 10000))
```



Análisis de similaridad entre las muestras

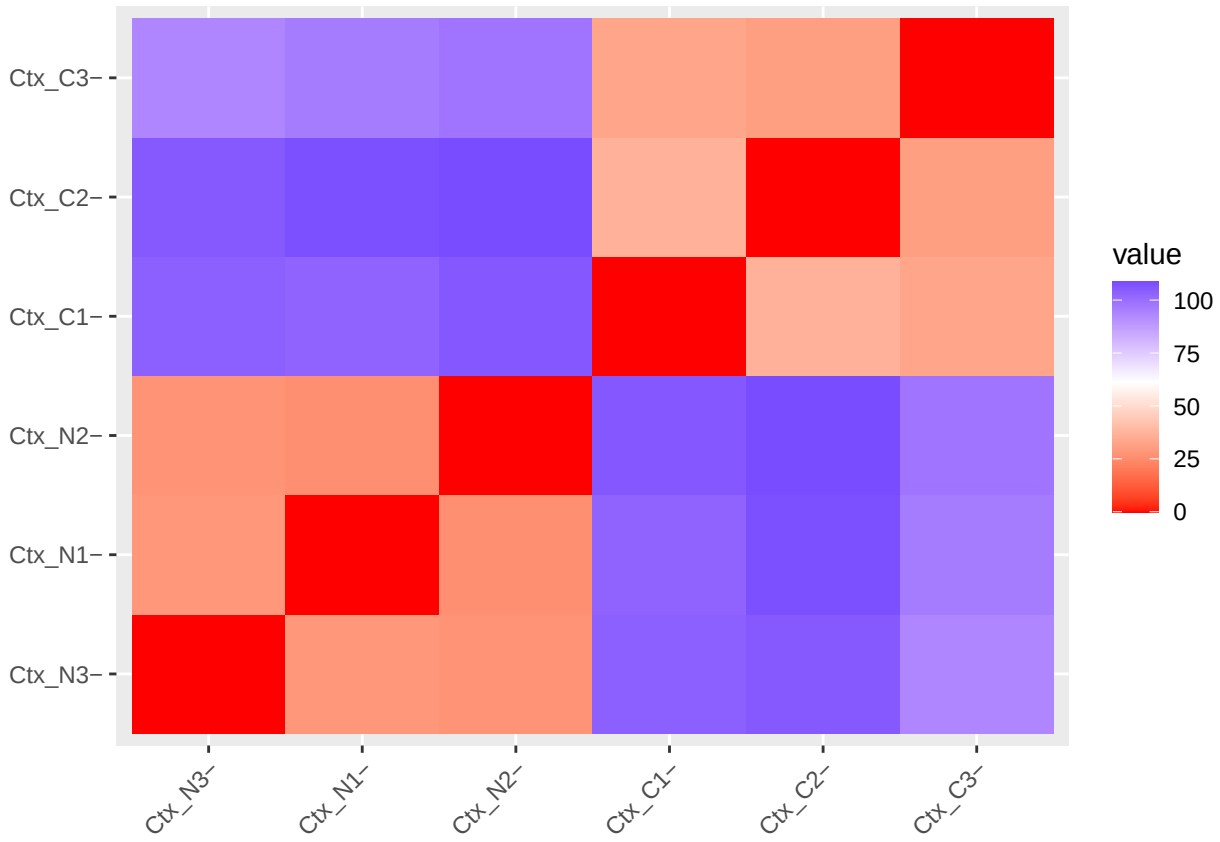
Distancia entre muestras

La función dist permite calcular una matriz de distancias que contiene las comparaciones dos a dos entre todas las muestras. Por defecto se utiliza una distancia euclídea.

```
log2count_norm <- cpm(y, log = TRUE)
sampleDists <- dist(t(log2count_norm))
round(sampleDists, 1)
```

```
##        Ctx_N1 Ctx_N2 Ctx_N3 Ctx_C1 Ctx_C2
## Ctx_N2   26.2
## Ctx_N3   28.7   27.4
## Ctx_C1  102.6  105.6  103.6
## Ctx_C2  107.5  108.5  105.2   36.6
## Ctx_C3   96.0   98.0   93.3   32.8   30.8
```

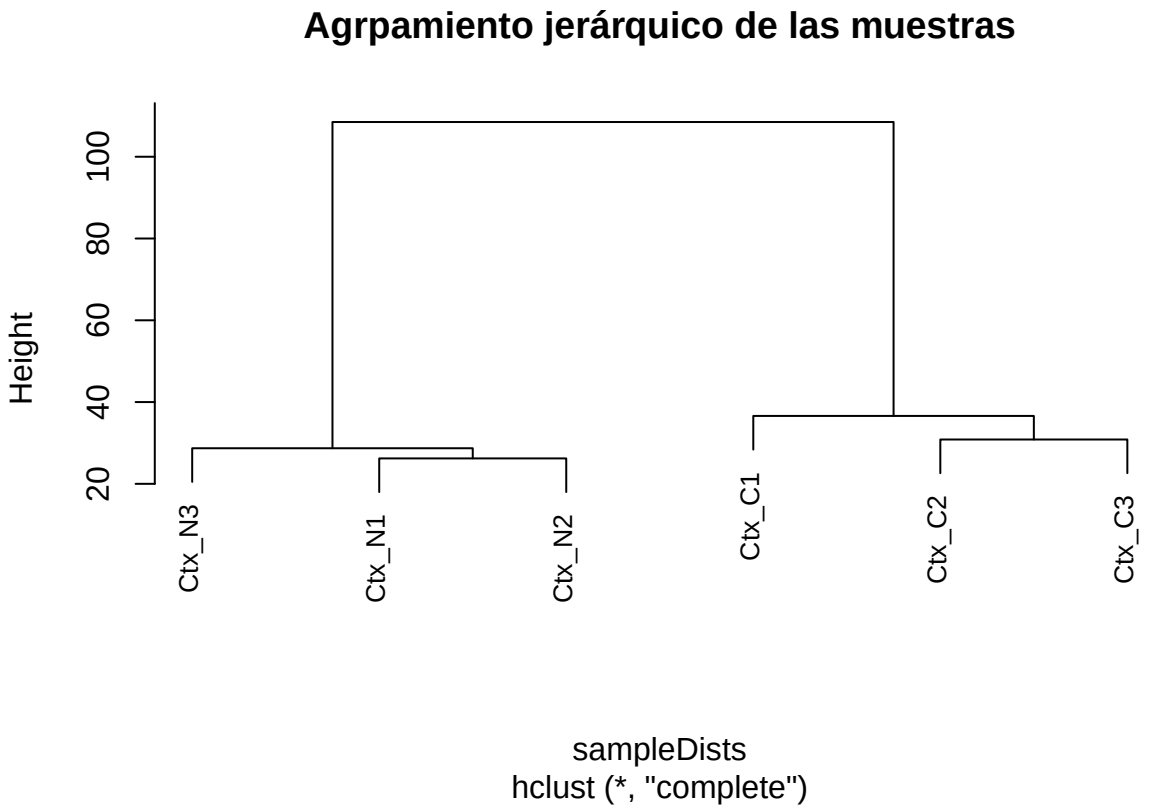
```
par(mfrow = c(1, 1))
fviz_dist(sampleDists)
```



Agrupamiento jerárquico

Un agrupamiento jerárquico proporciona una representación alternativa, también basada en la matriz de distancias.

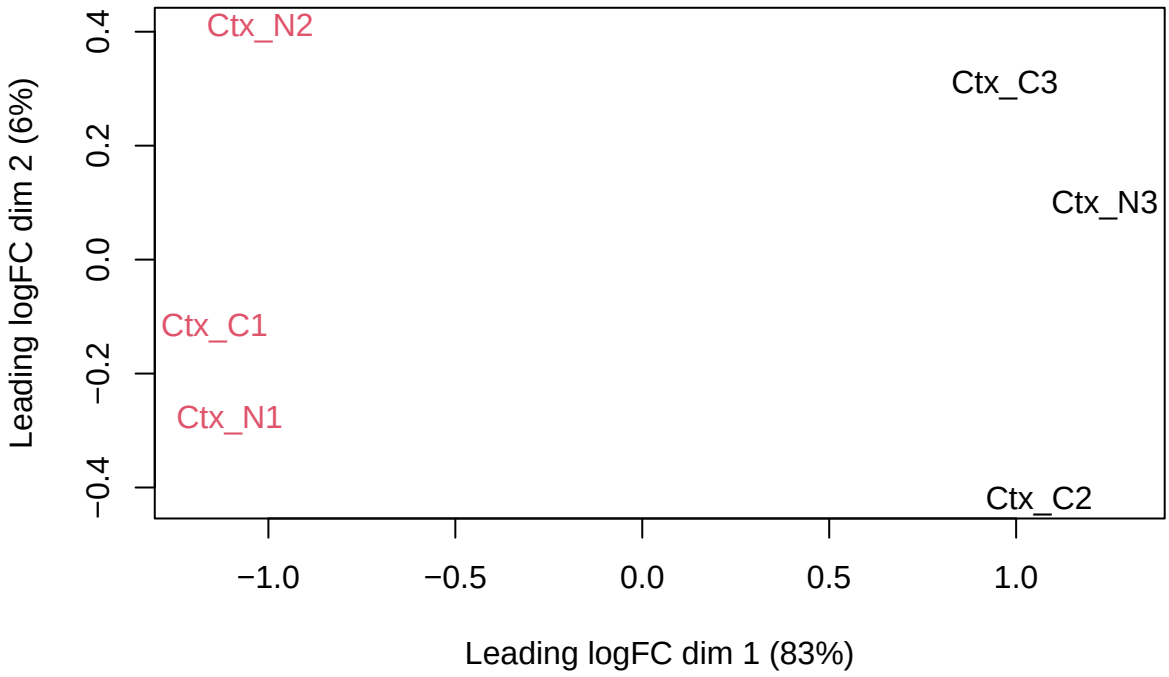
```
hc <- hclust(sampleDists)
plot(hc, labels = colnames(log2count_norm), main = "Agrpamiento jerárquico de las muestras",
    cex = 0.8)
```



Análisis de Escalamiento Multidimensional (MDS)

Reducción dimensional

```
plotMDS(y, col=as.numeric(fraction), labels=barcode, cex = 1 )
```



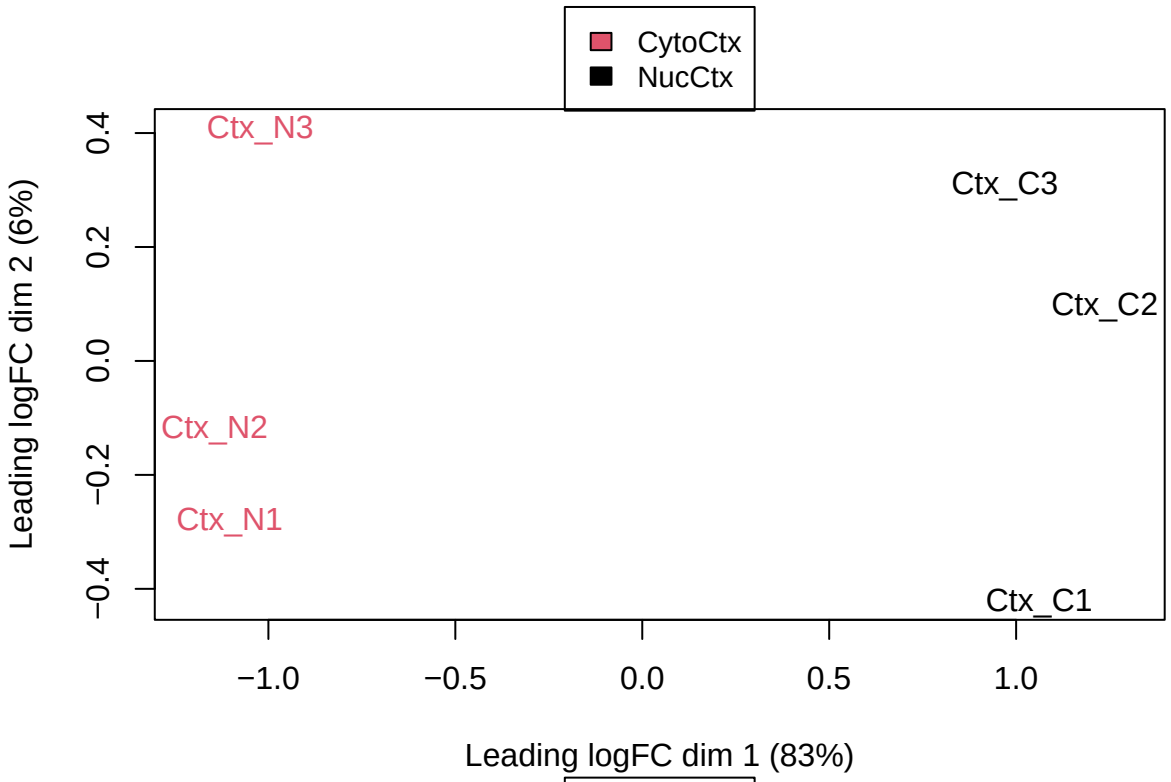

```
pdf(paste("plotMDS.pdf",sep=""))
plotMDS(y, col=as.numeric(fraction), labels=barcode, cex = 1 )
dev.off()
```

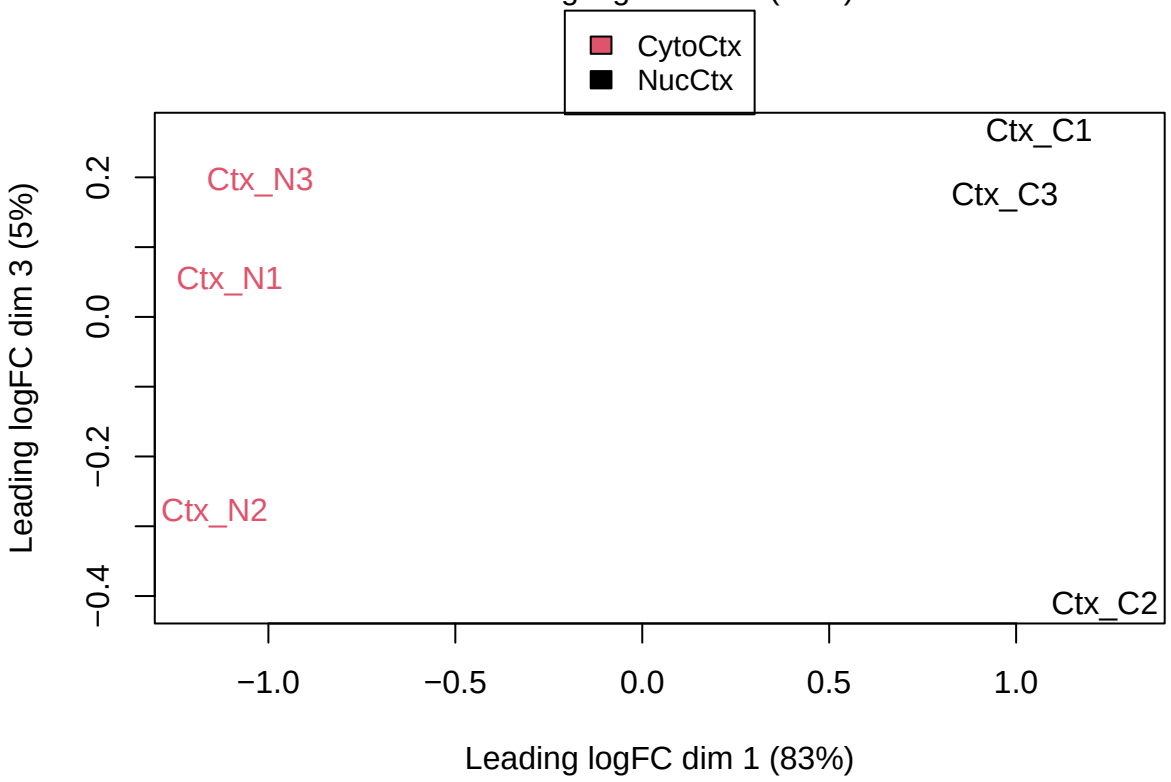
```
## pdf
##   2
```

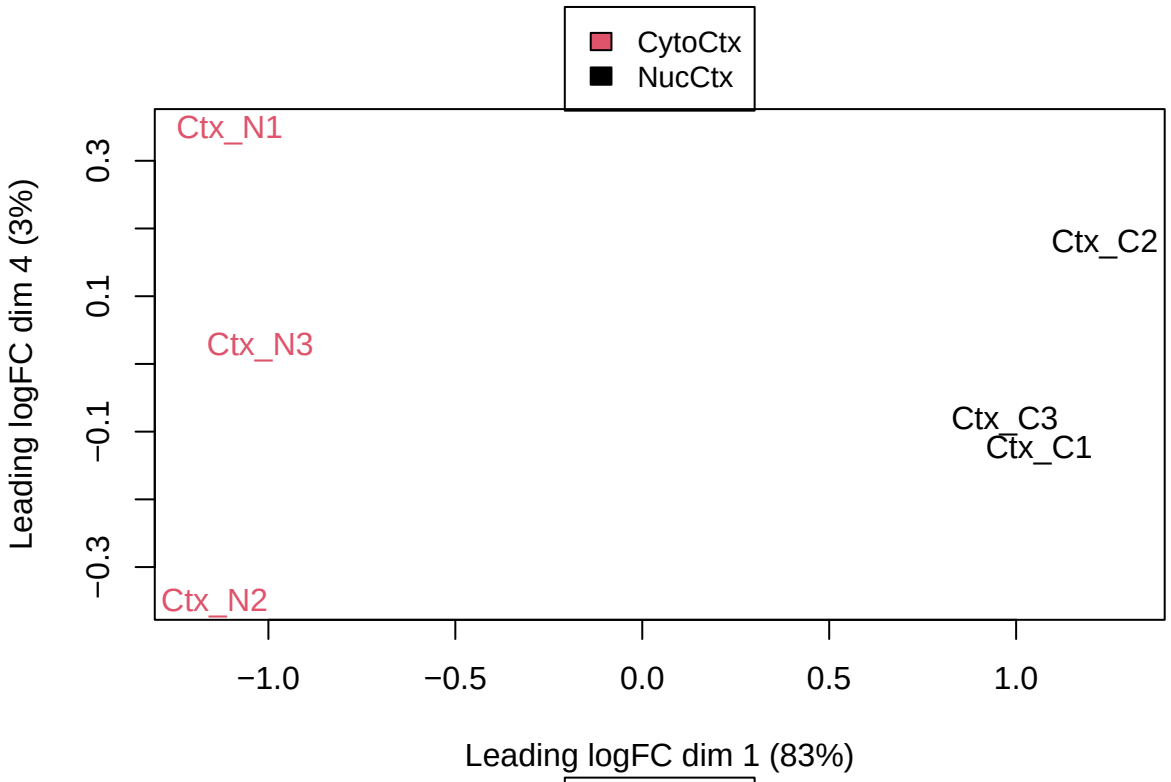
```
combi=combn(unique(c(1:6)), 2)
pdf("plotMDS_01_wo_outliers.pdf")
par(mfrow=c(2,3))
for (i in 1:ncol(combi)){
  plotMDS(y,dim.plot = combi[,i],col=as.numeric(fraction),pch=16, labels=colnames(y) )
  legend(x = "top",inset = c(0, -0.20 ), legend =levels(unique(group)), cex = 0.9 ,fill= as.numeric(unic
}
dev.off()
```

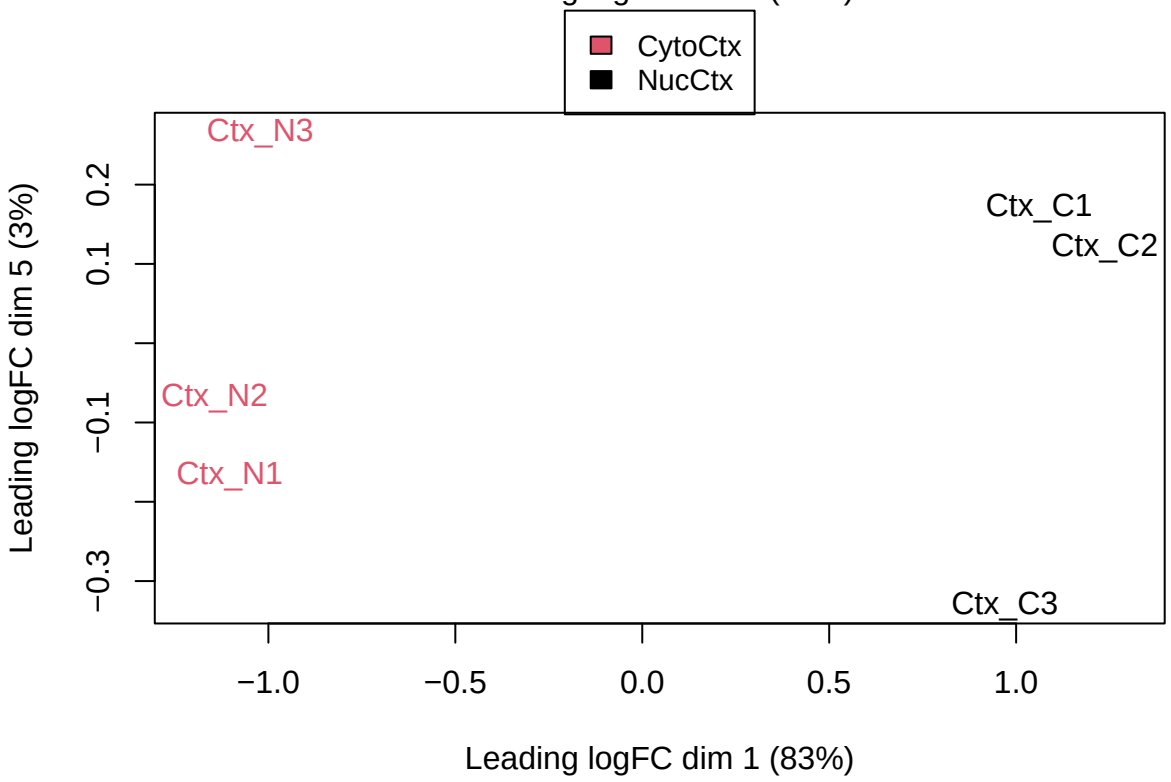
```
## pdf
##   2
```

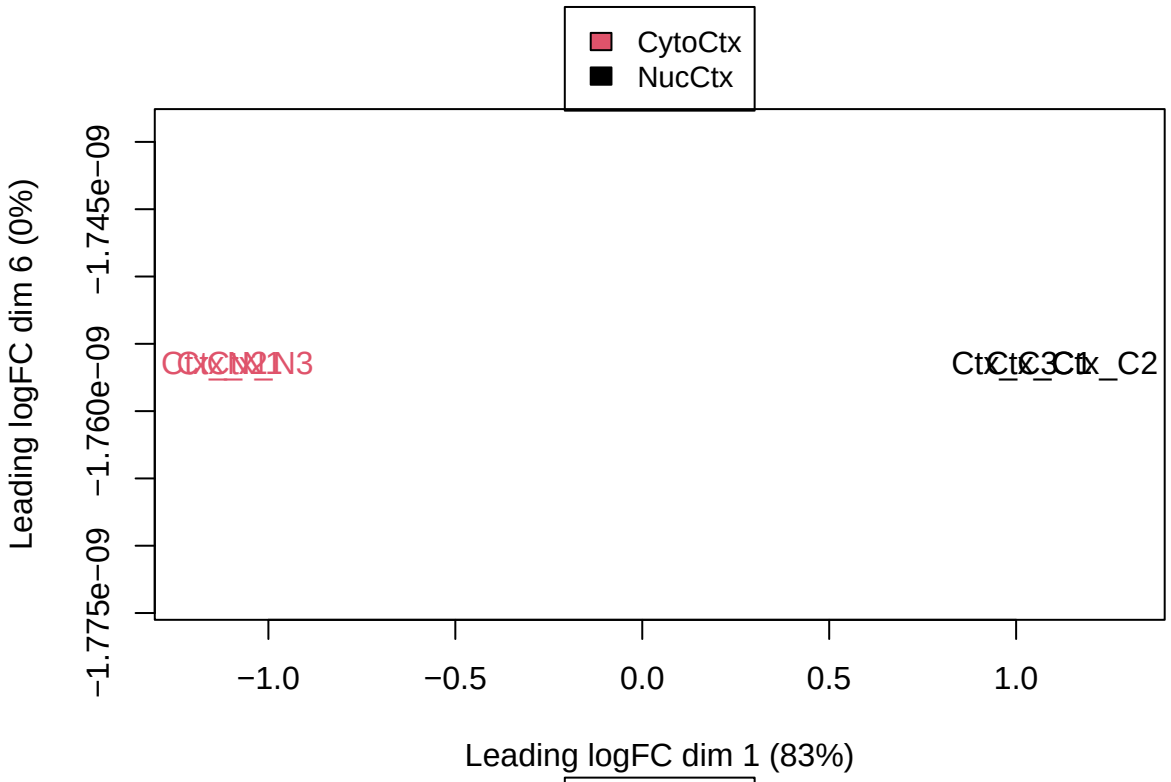
```
for (i in 1:ncol(combi)){
  plotMDS(y,dim.plot = combi[,i],col=as.numeric(fraction),pch=16, labels=colnames(y) )
  legend(x = "top",inset = c(0, -0.20 ), legend =levels(unique(group)), cex = 0.9 ,fill= as.numeric(unic
}
```

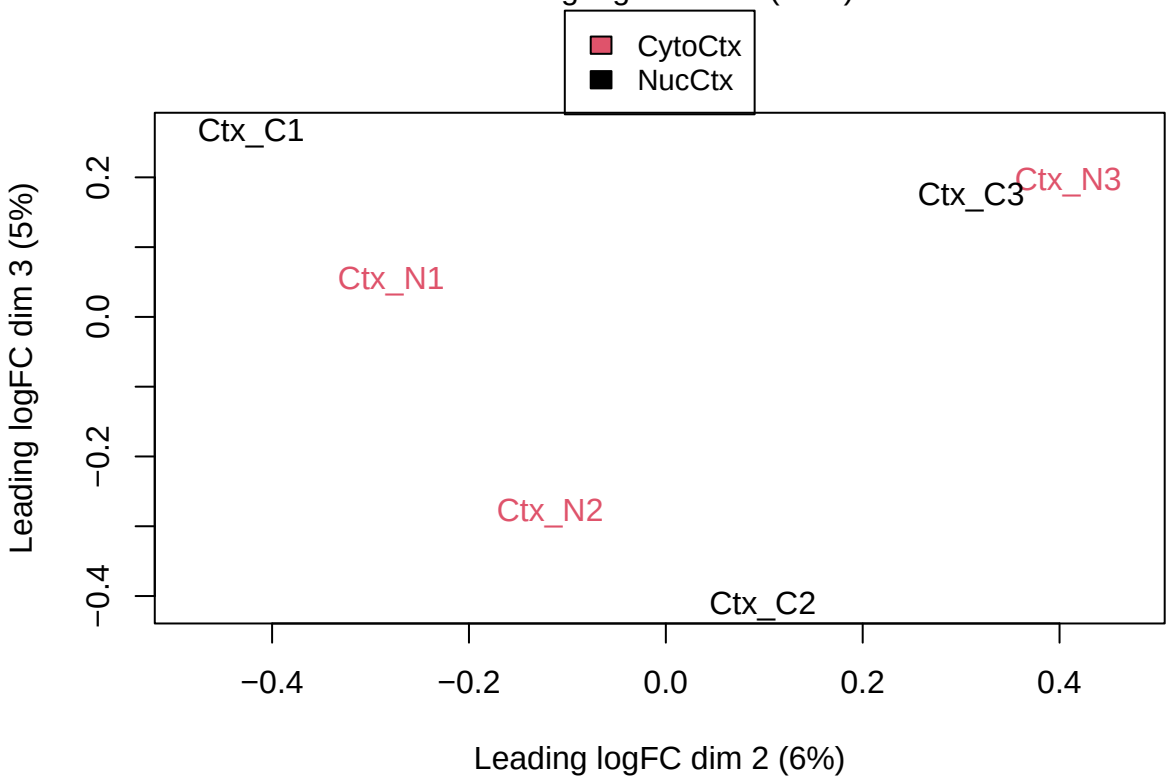

CytoCtx
NucCtx
Ctx_N3
Ctx_C3
Ctx_C2
Ctx_N2
Ctx_N1
Ctx_C1
Leading logFC dim 2 (6%)
0.4
0.2
0.0
−0.2
−0.4
−1.0
−0.5
0.0
0.5
1.0
Leading logFC dim 1 (83%)

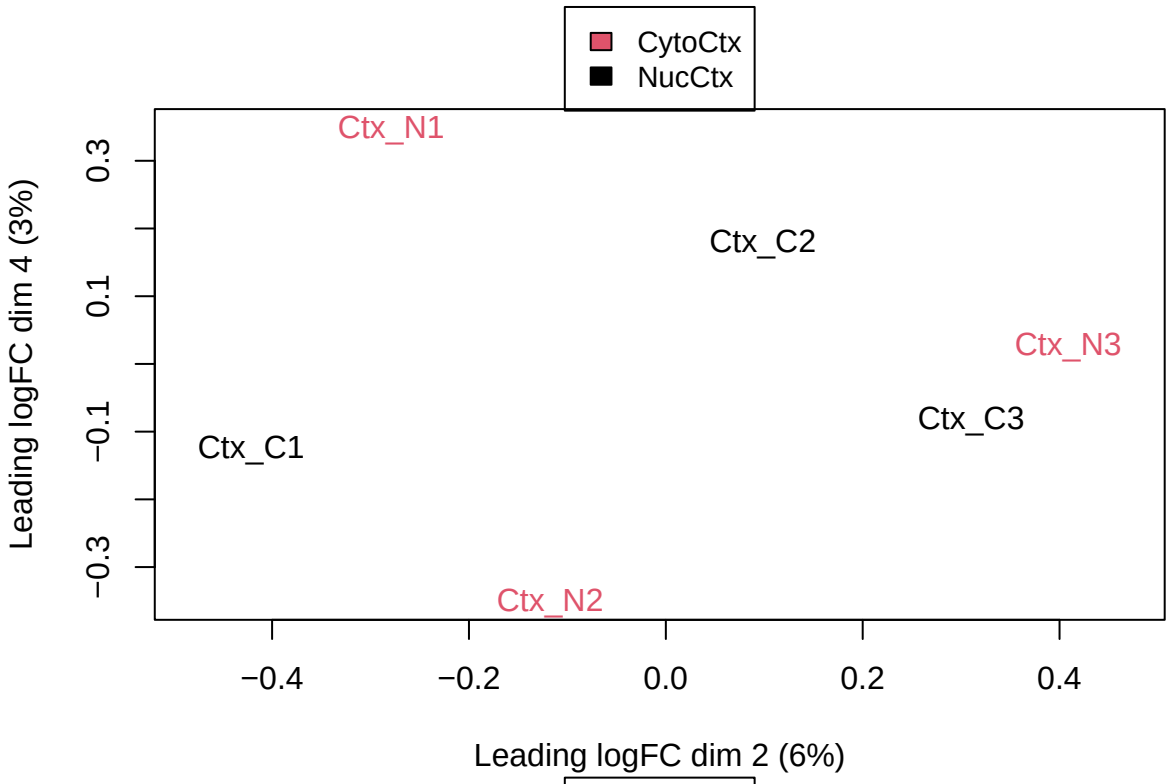

CytoCtx
NucCtx
Ctx_C1
Ctx_N3
Ctx_C3
Ctx_N1
Ctx_N2
Ctx_C2
Leading logFC dim 3 (5%)
0.2
0.0
−0.2
−0.4
−1.0
−0.5
0.0
0.5
1.0
Leading logFC dim 1 (83%)

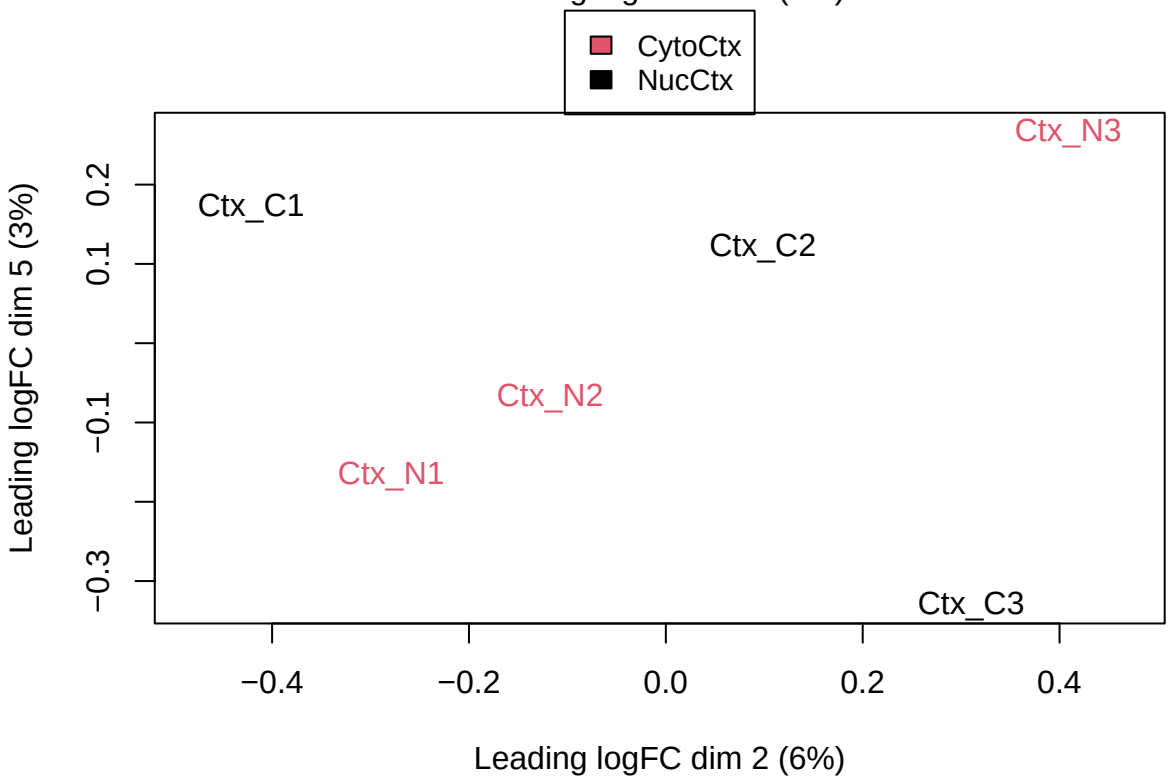

CytoCtx
NucCtx
Ctx_N1
Ctx_C2
Ctx_N3
Ctx_C3
Ctx_C1
Ctx_N2
Leading logFC dim 4 (3%)
0.3
0.1
−0.1
−0.3
−1.0
−0.5
0.0
0.5
1.0
Leading logFC dim 1 (83%)

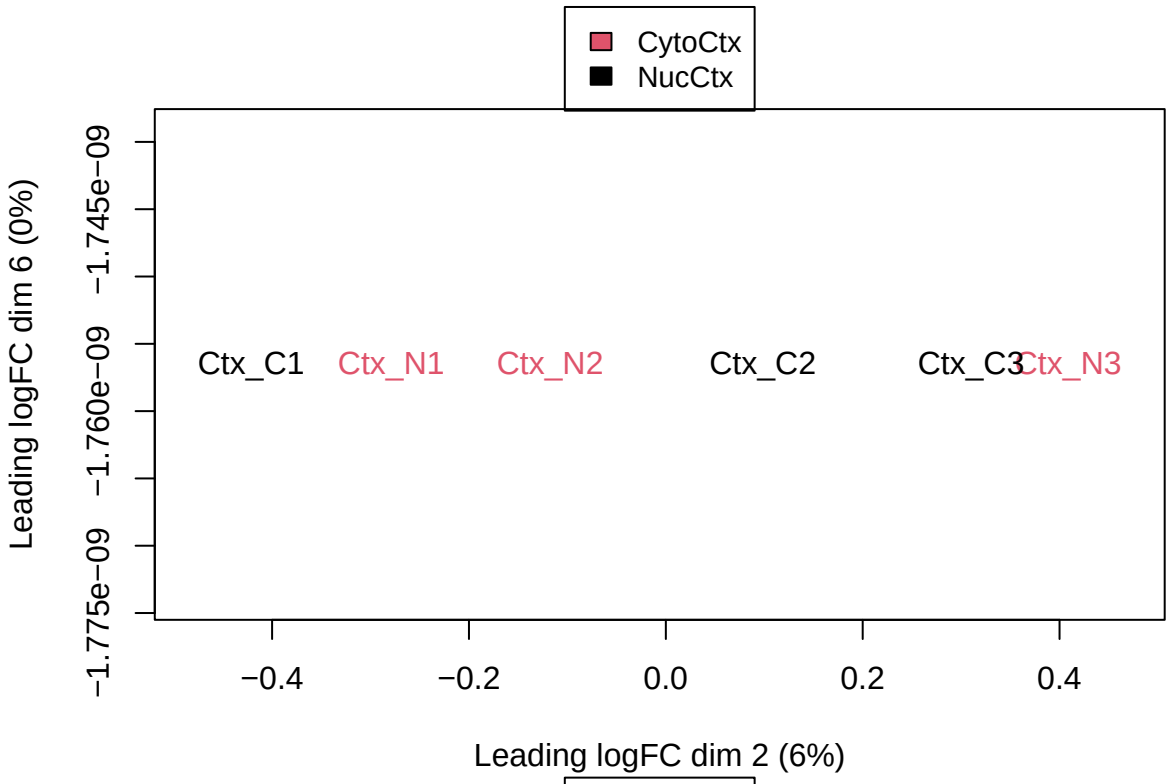

CytoCtx
NucCtx
Ctx_N3
Ctx_C1
Ctx_C2
Ctx_N2
Ctx_N1
Ctx_C3
Leading logFC dim 5 (3%)
0.2
0.1
−0.1
−0.3
−1.0
−0.5
0.0
0.5
1.0
Leading logFC dim 1 (83%)

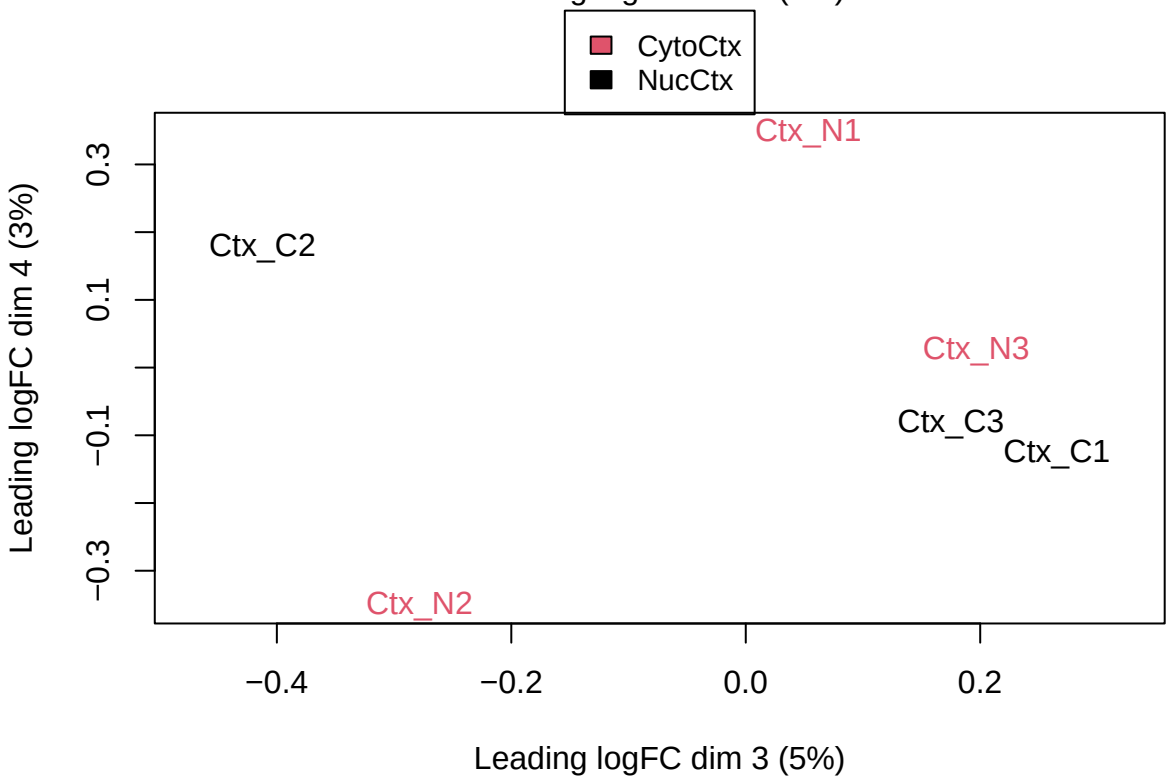

CytoCtx
NucCtx
Leading logFC dim 6 (0%)
−1.775e−09
−1.760e−09
−1.745e−09
−1.0
−0.5
0.0
0.5
1.0
Leading logFC dim 1 (83%)

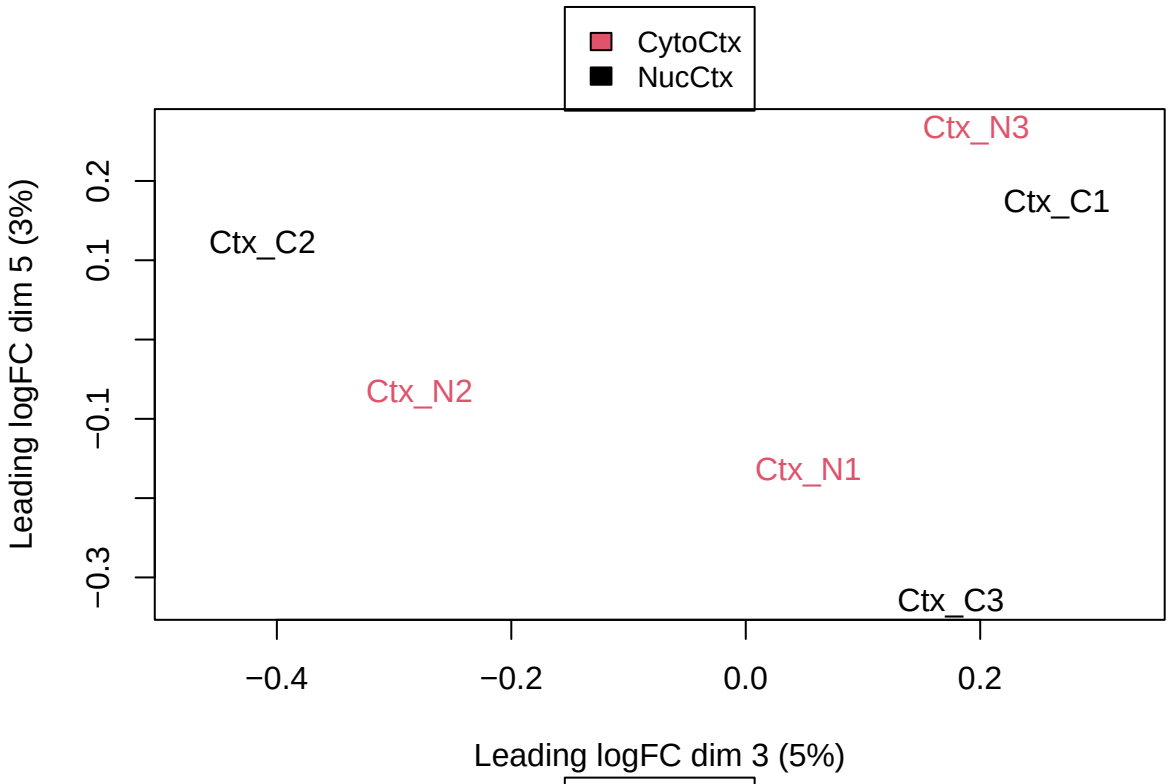

CytoCtx
NucCtx
Ctx_C1
Ctx_C3
Ctx_N3
Ctx_N1
Ctx_N2
Ctx_C2
Leading logFC dim 3 (5%)
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
Leading logFC dim 2 (6%)

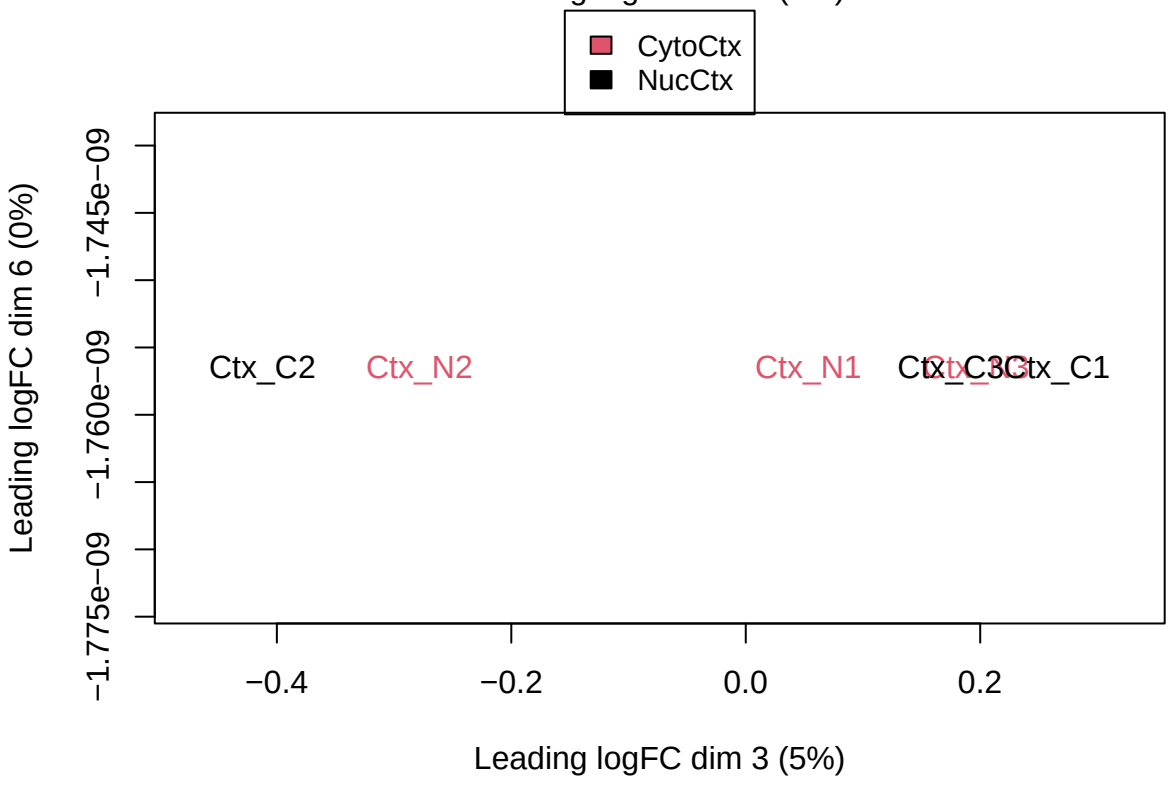

CytoCtx
NucCtx
Ctx_N1
Ctx_C2
Ctx_N3
Ctx_C3
Ctx_C1
Ctx_N2
Leading logFC dim 4 (3%)
0.3
0.1
−0.1
−0.3
−0.4
−0.2
0.0
0.2
0.4
Leading logFC dim 2 (6%)

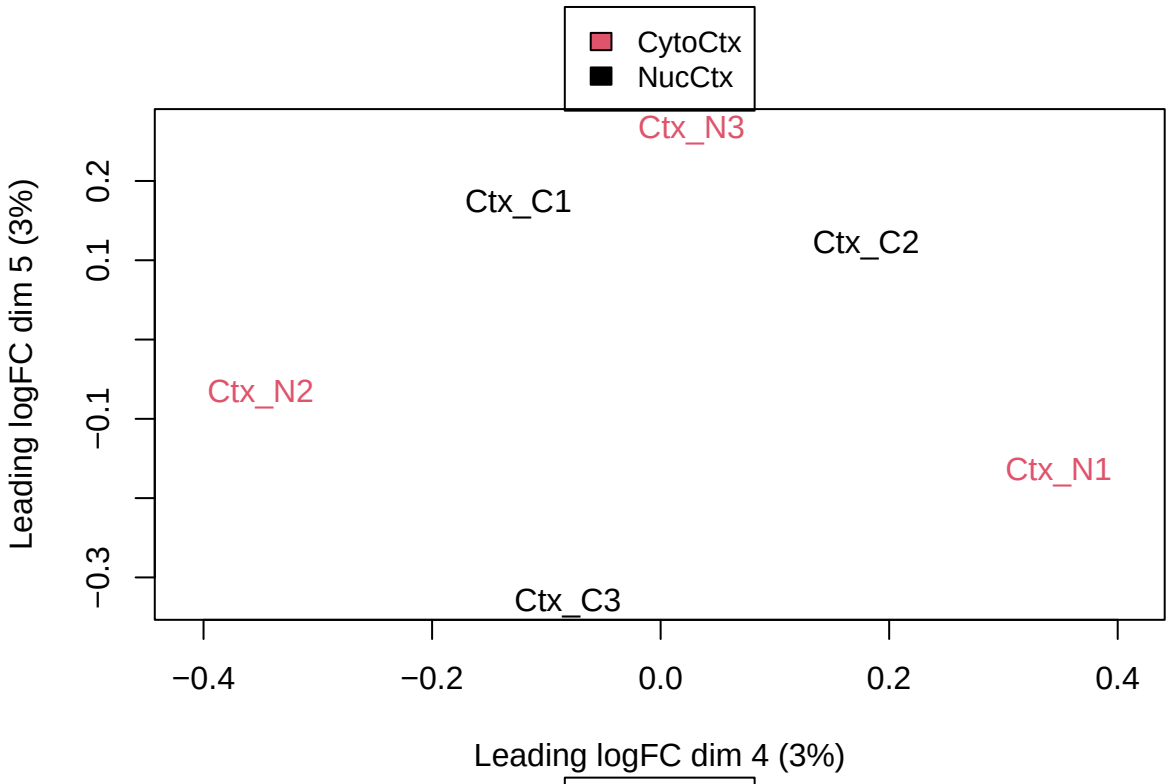

CytoCtx
NucCtx
Ctx_N3
Ctx_C1
Ctx_C2
Ctx_N2
Ctx_N1
Ctx_C3
Leading logFC dim 5 (3%)
0.2
0.1
−0.1
−0.3
−0.4
−0.2
0.0
0.2
0.4
Leading logFC dim 2 (6%)

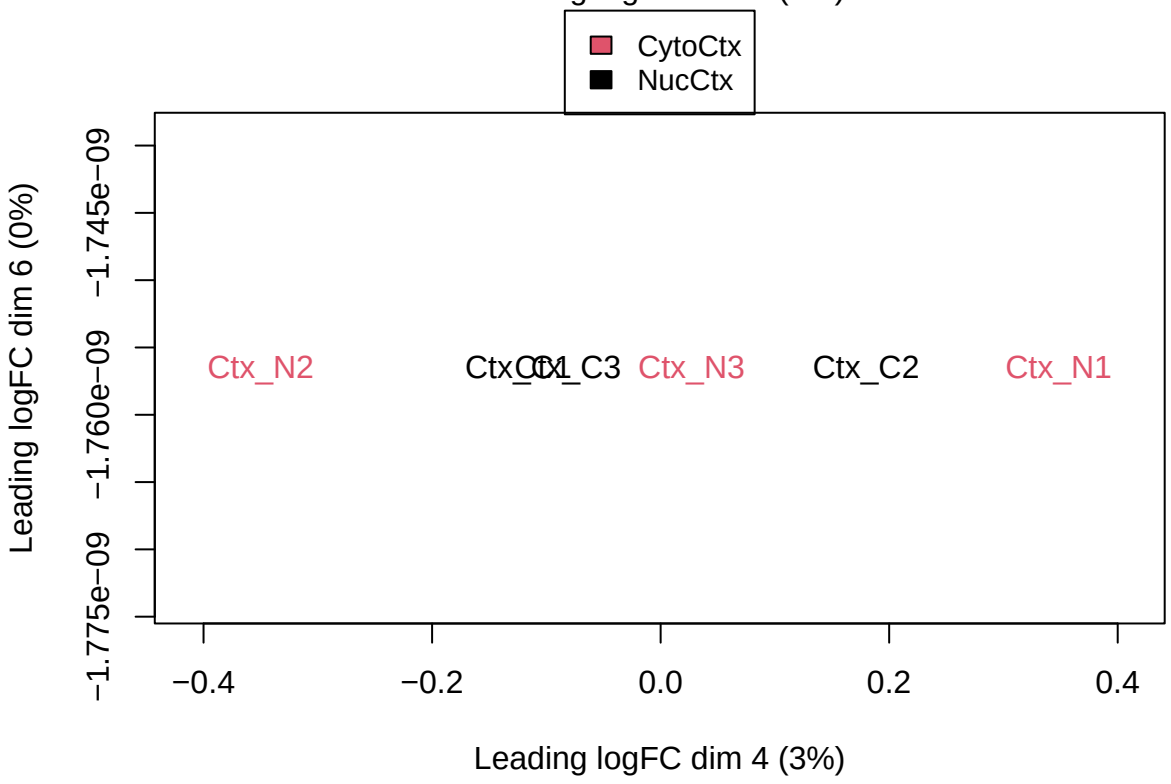

CytoCtx
NucCtx
Leading logFC dim 6 (0%)
−1.775e−09
−1.760e−09
−1.745e−09
Ctx_C1
Ctx_N1
Ctx_N2
Ctx_C2
Ctx_C3
Ctx_N3
−0.4
−0.2
0.0
0.2
0.4
Leading logFC dim 2 (6%)

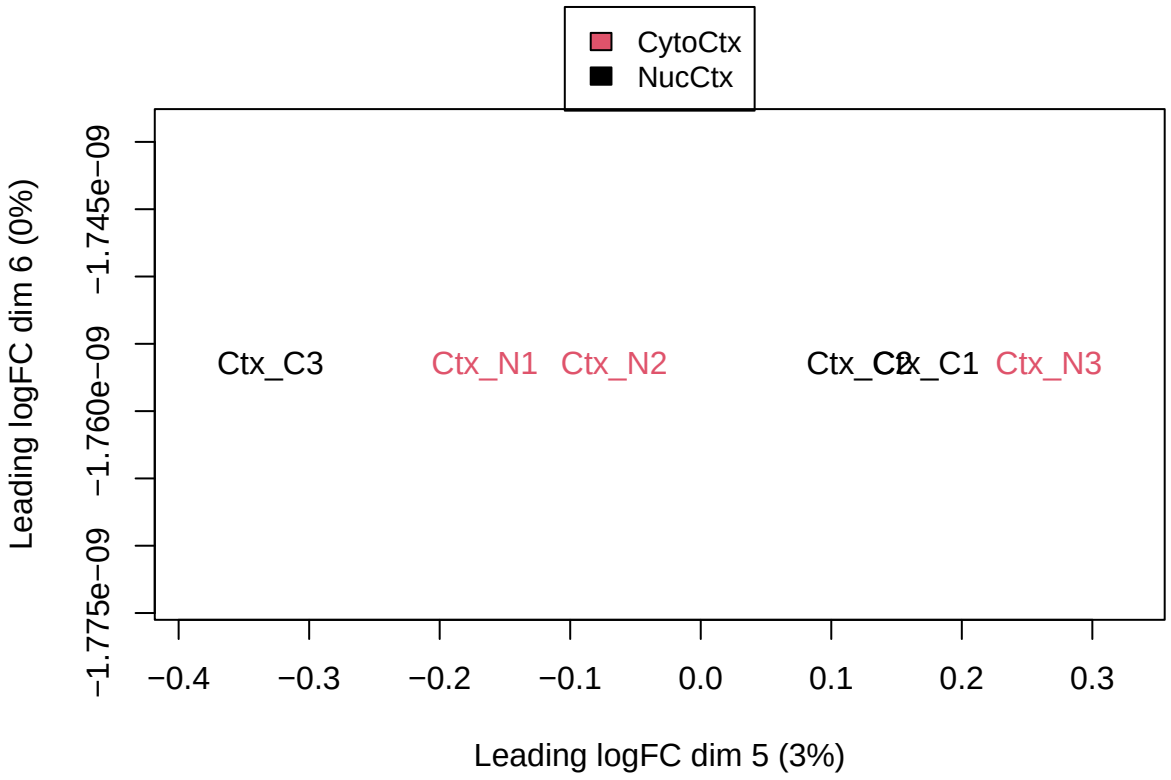

CytoCtx
NucCtx
Leading logFC dim 4 (3%)
−0.3
−0.1
0.1
0.3
Ctx_N1
Ctx_C2
Ctx_N3
Ctx_C3
Ctx_C1
Ctx_N2
−0.4
−0.2
0.0
0.2
Leading logFC dim 3 (5%)


CytoCtx
NucCtx
Ctx_N3
Ctx_C1
Ctx_C2
Ctx_N2
Ctx_N1
Ctx_C3
Leading logFC dim 5 (3%)
0.2
0.1
−0.1
−0.3
−0.4
−0.2
0.0
0.2
Leading logFC dim 3 (5%)


CytoCtx
NucCtx
Ctx_C2
Ctx_N2
Ctx_N1
Ctx_C1
Leading logFC dim 6 (0%)
−1.745e−09
−1.760e−09
−1.775e−09
−0.4
−0.2
0.0
0.2
Leading logFC dim 3 (5%)


CytoCtx
NucCtx
Ctx_N3
Ctx_C1
Ctx_C2
Ctx_N2
Ctx_N1
Ctx_C3
Leading logFC dim 5 (3%)
0.2
0.1
−0.1
−0.3
−0.4
−0.2
0.0
0.2
0.4
Leading logFC dim 4 (3%)


CytoCtx
NucCtx
Ctx_N2
Ctx_C1
Ctx_C3
Ctx_N3
Ctx_C2
Ctx_N1
Leading logFC dim 6 (0%)
−1.745e−09
−1.760e−09
−1.775e−09
−0.4
−0.2
0.0
0.2
0.4
Leading logFC dim 4 (3%)

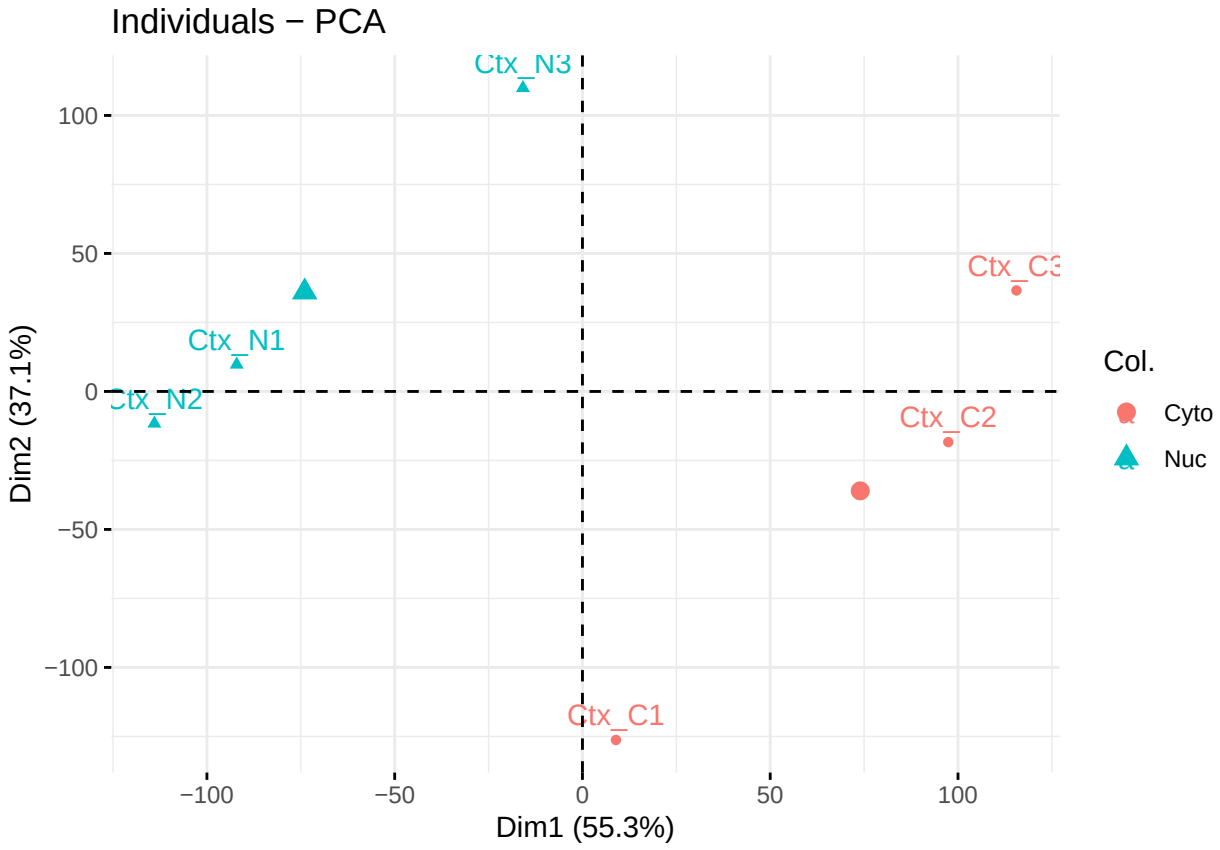


PCA

```
library(FactoMineR)

pca.raw.y <- log2(y$counts+1)

pca.y <- PCA(t(pca.raw.y),graph = F)
fviz_pca_ind(pca.y, col.ind = fraction)
```




```
pdf(paste("PCA.pdf",sep=""))
fviz_pca_ind(pca.y, col.ind = fraction)
dev.off()
```

```
## pdf
##   2
```

Análisis de expresión diferencial (DE)

El objetivo del análisis de expresión diferencial es seleccionar genes cuya expresión difiere entre grupos.

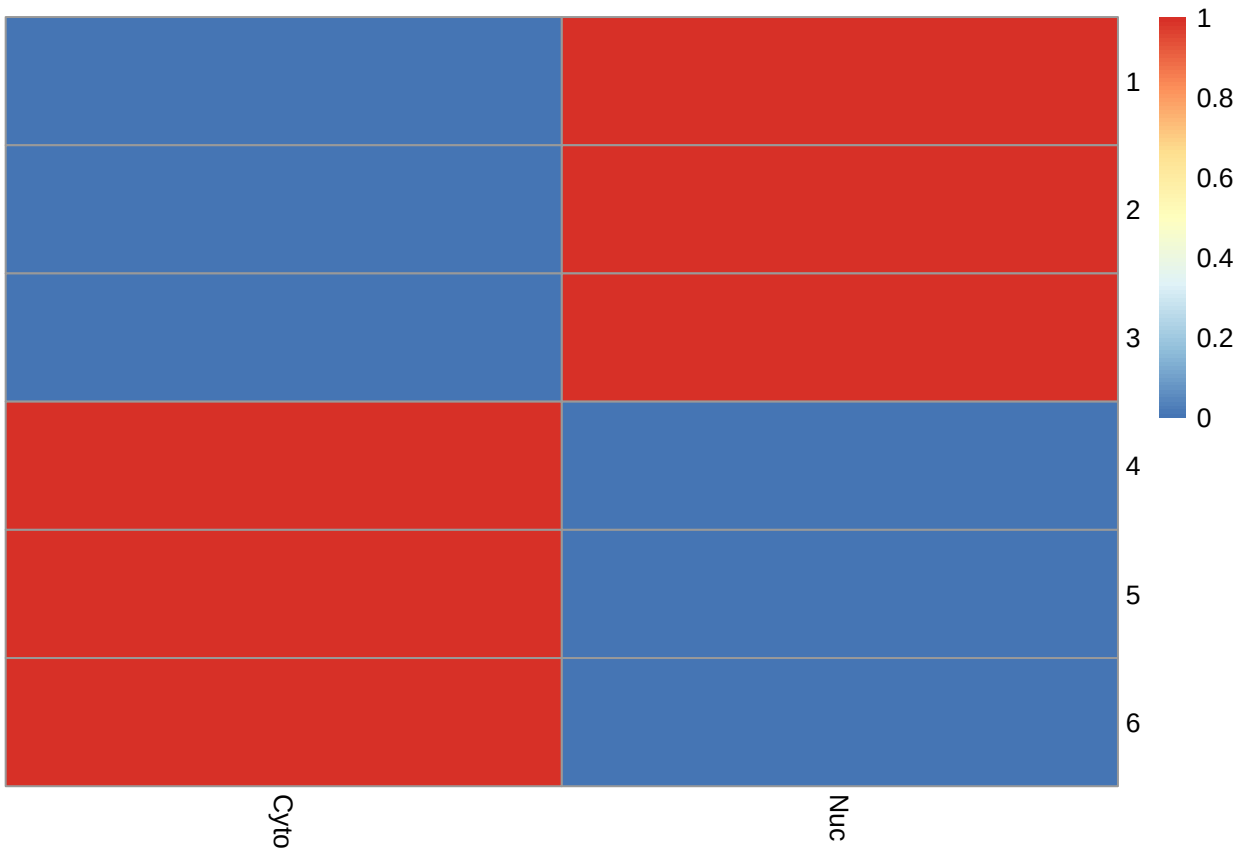
Selección de genes usando limma-Voom

La ventaja principal de esta aproximación es que permite trabajar con toda la flexibilidad de los modelos lineales para representar diseños experimentales, y, en muchos casos , aprovechar la experiencia previa del usuario en el manejo de limma.

Matriz de diseño

Utilizando la variable group podemos definir una matriz de diseño y, sobre ésta, los contrastes que nos interesan.

```
mod <- model.matrix(~0+fraction)
colnames(mod)=gsub("fraction","",colnames(mod))
pheatmap(mod,cluster_rows = FALSE,cluster_cols = FALSE)
```




```
mod
```

```
##   Cyto Nuc
## 1    0   1
## 2    0   1
## 3    0   1
## 4    1   0
## 5    1   0
## 6    1   0
## attr(,"assign")
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$fraction
## [1] "contr.treatment"
```

Matriz de contrastes

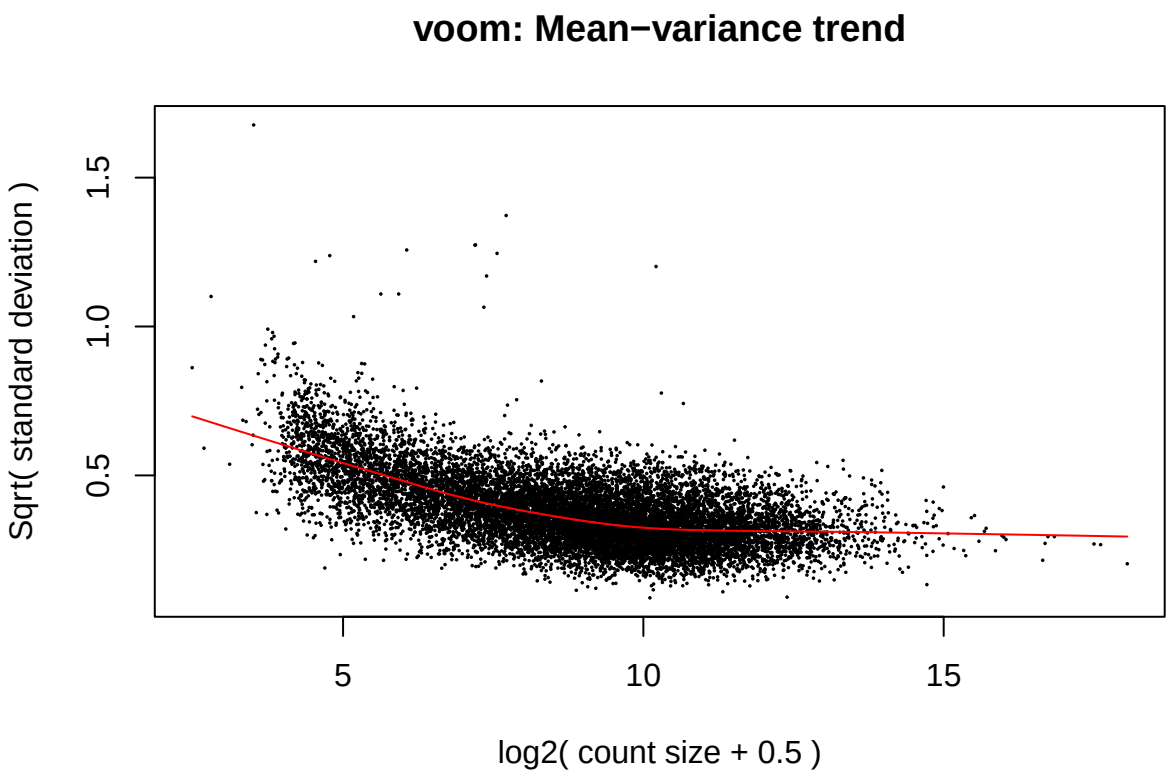
```
contr.matrix <- makeContrasts(
  Cyto_vs_Nuc = Cyto-Nuc,
```

```
levels=colnames(mod))
contr.matrix
```

```
##       Contrasts
## Levels Cyto_vs_Nuc
##   Cyto           1
##   Nuc           -1
```

Transformación de los contajes

```
v=voom(y,mod, plot = T)
```




```
v
```

```
## An object of class "EList"
## $targets
##        group lib.size norm.factors
## Ctx_N1     1 16906943    0.9700342
## Ctx_N2     1 15669393    0.9809899
## Ctx_N3     1 22873815    0.9579471
## Ctx_C1     1 15961367    1.0241706
## Ctx_C2     1 22911761    1.0289512
## Ctx_C3     1 25505928    1.0409739
##
## $E
##                     Ctx_N1   Ctx_N2   Ctx_N3   Ctx_C1   Ctx_C2   Ctx_C3
```

```
## ENSMUSG00000051951 5.209322 4.944508 5.018678 5.134058 4.248512 5.327944
## ENSMUSG00000033845 5.370636 5.335044 5.310129 5.852893 5.997191 5.801452
## ENSMUSG00000025903 3.999941 4.187469 4.086146 6.213550 5.955182 6.023772
## ENSMUSG00000033813 4.798507 4.773274 4.709582 6.833607 6.519874 6.527218
## ENSMUSG00000002459 2.627815 2.804910 2.285275 3.561908 3.479163 3.150606
## 13451 more rows ...
##
## $weights
##          [,1]     [,2]     [,3]     [,4]      [,5]      [,6]
## [1,] 71.94695 69.36192 82.06288 66.37668  78.65919  82.12865
## [2,] 78.58302 76.00643 87.72334 88.17369  95.86449  97.46106
## [3,] 50.62680 48.50259 59.76268 91.20824  97.70441  98.89640
## [4,] 64.97664 62.49110 75.24044 98.08869 100.93620 101.66574
## [5,] 25.98371 24.56360 32.09084 36.83788  45.79962  48.67236
## 13451 more rows ...
##
## $design
##   Cyto Nuc
## 1    0   1
## 2    0   1
## 3    0   1
## 4    1   0
## 5    1   0
## 6    1   0
## attr(,"assign")
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$fraction
## [1] "contr.treatment"
```

Selección de genes diferencialmente expresados

Como en el caso de los microarrays el objeto v y las matrices de diseño y contrastes se utilizaran para ajustar un modelo y, a continuación realizar las comparaciones especificadas sobre el modelo ajustado. El proceso finaliza con la regularización del estimador del error usando la función eBayes.

```
fit=lmFit(v,mod)
fit2 <- contrasts.fit(fit, contr.matrix)
fit2 <- eBayes(fit2)
(results<-topTable(fit2, coef = 1, adjust="BH"))
```

```
##                       logFC   AveExpr         t      P.Value    adj.P.Val
## ENSMUSG00000018707 -3.458027  9.394118 -34.25716 1.490925e-13 1.055366e-09
## ENSMUSG00000014602 -3.346214  9.774906 -32.74156 2.582927e-13 1.055366e-09
## ENSMUSG00000063077 -3.239485  9.010497 -32.40273 2.930533e-13 1.055366e-09
## ENSMUSG00000026764 -3.335681  9.072041 -31.72330 3.789737e-13 1.055366e-09
## ENSMUSG00000028364 -2.381221  7.692305 -31.16199 4.705851e-13 1.055366e-09
## ENSMUSG00000034243 -3.313622  5.748709 -31.55674 4.039615e-13 1.055366e-09
## ENSMUSG00000061983  2.431092  8.596411  30.54217 6.003565e-13 1.079689e-09
## ENSMUSG00000023830 -2.663222  5.801741 -30.37332 6.420776e-13 1.079689e-09
## ENSMUSG00000084899 -2.781841  6.556294 -30.04438 7.326312e-13 1.079689e-09
## ENSMUSG00000057738 -2.359876 10.258145 -29.36990 9.645199e-13 1.079689e-09
##                           B
```

```
## ENSMUSG00000018707 21.54248
## ENSMUSG00000014602 21.03362
## ENSMUSG00000063077 20.90532
## ENSMUSG00000026764 20.66371
## ENSMUSG00000028364 20.44143
## ENSMUSG00000034243 20.42183
## ENSMUSG00000061983 20.22159
## ENSMUSG00000023830 20.04695
## ENSMUSG00000084899 19.98405
## ENSMUSG00000057738 19.78011
```

```
summary(decideTests(fit2))
```

```
##        Cyto_vs_Nuc
## Down          4846
## NotSig        3480
## Up            5130
```

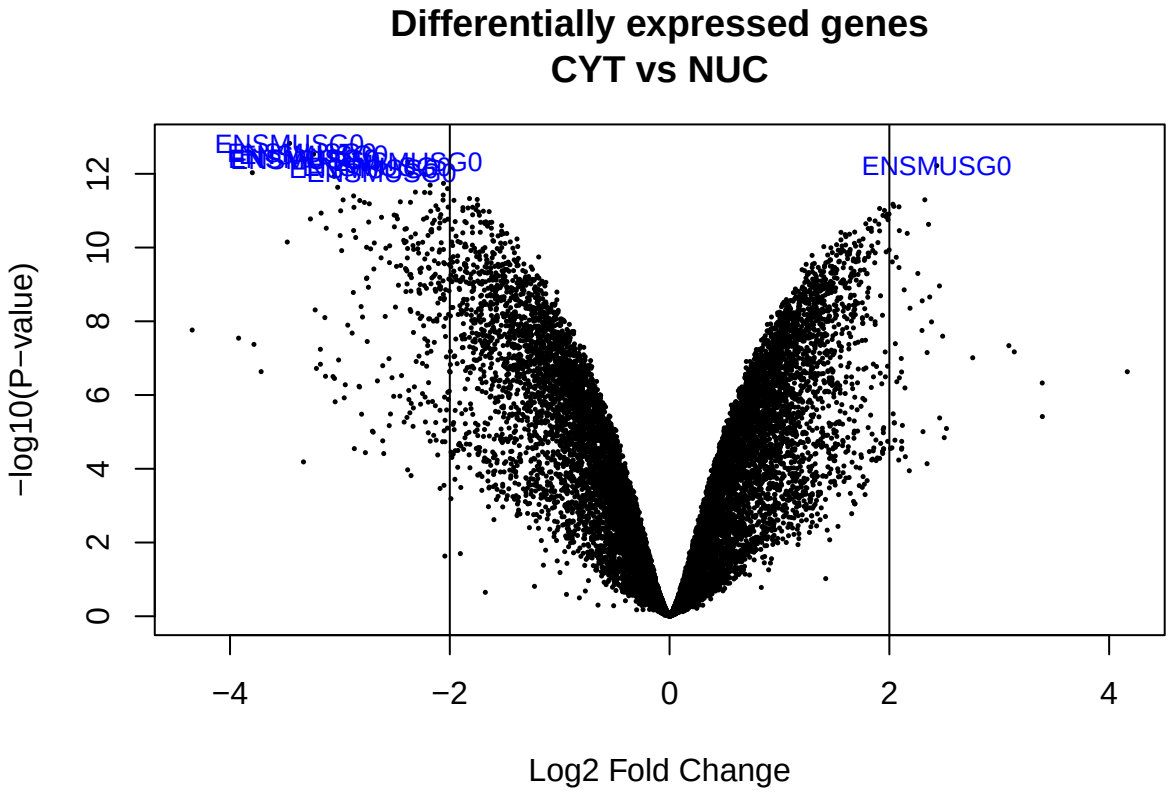
```
summa.fit <- decideTests(fit2, p.value = 0.01, lfc = 2)
summary(summa.fit)
```

```
##        Cyto_vs_Nuc
## Down           255
## NotSig       13150
## Up              51
```

Visualización de los resultados

Volcano Plot

```
volcanoplot(fit2, coef = 1, highlight = 10,names=rownames(fit2) ,main =paste( "Differentially expressed
abline(v=c(-2,2))
```




```
pdf(paste("volcanoplot.pdf",sep=""))
volcanoplot(fit2, coef = 1, highlight = 10,names=rownames(fit2) ,main =paste( "Differentially expressed
abline(v=c(-2,2))
dev.off()
```

```
## pdf
##   2
```

Perfiles de expresión

Con el fin de observar si existen perfiles de expresión diferenciados podemo realizar un mapa de colores con los genes más diferencialmente expresados.

Es decir, fijamos un criterio de selección de genes y retenemos aquellos componentes de la tabla de resultados que lo cumplen. Por ejemplo: Genes con un p-valor ajustado inferior a 0.001 y un 'fold-change' superior a 6 o inferior a -6.

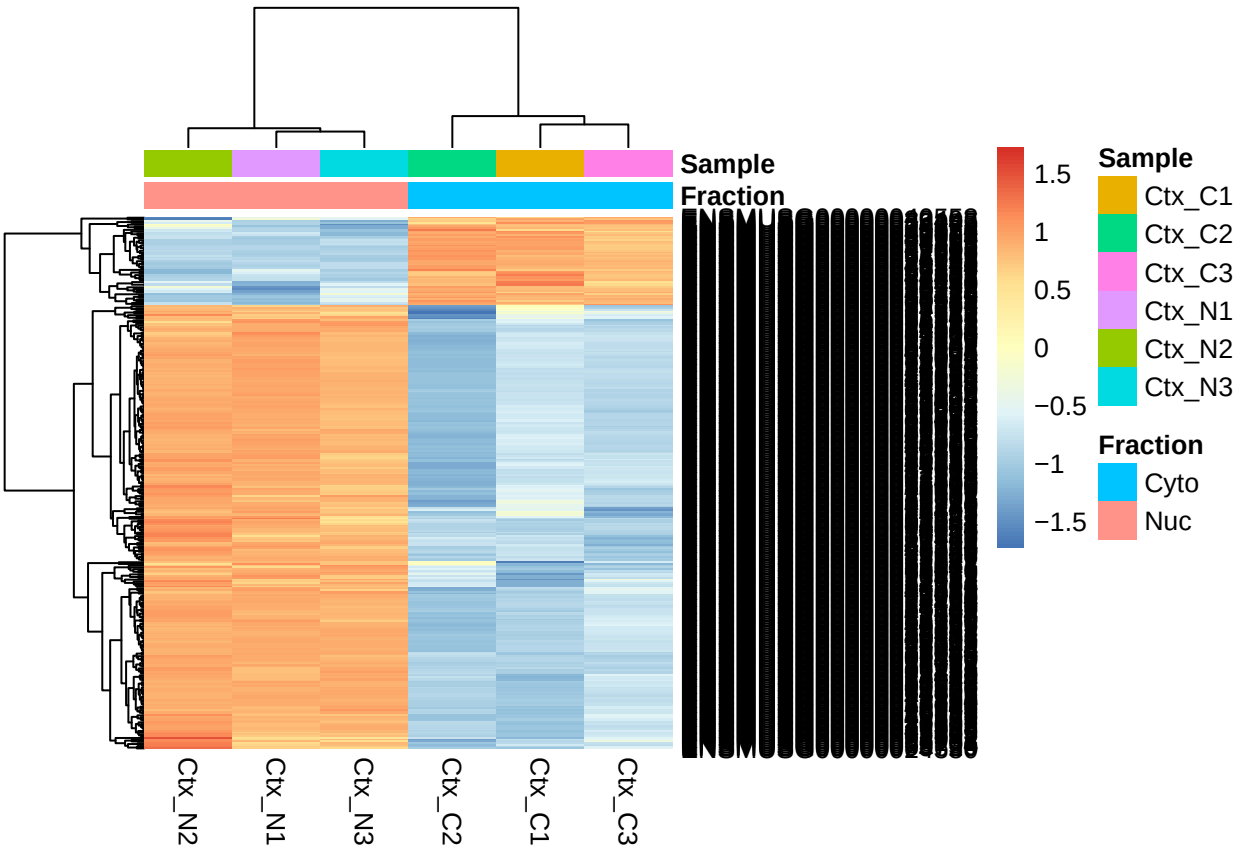
mapa de colores

```
for (i in  colnames(fit2$coefficients)){
  top=topTable(fit2,coef=i,sort="p", n=13456)
  genes=rownames(top[which(top$adj.P.Val<0.01 & abs(top$logFC)>2),])
  write.table(top,paste(i,"_limma_voom.txt",sep=""),quote=F)
  term1=strsplit(i,split="_vs_")[[1]][1]
  term2=strsplit(i,split="_vs_")[[1]][2]
```

```
  samples=rownames(subset(info,fraction==term1 | fraction==term2))
  expr=v$E[genes,samples]
  rownames(expr)=do.call(rbind, strsplit(genes, ','))[,1]
  if (length(genes) >1) {
    pdf(paste("pheatmap_DE_genes__01_",i,".pdf",sep=""), width = 10, height = 12)
    pheatmap(expr,scale="row",annotation_col=info[,c("Fraction","Sample")], border_color = "NA",show_ro
    dev.off()
  }}

write.table(v$E,"logcpm.txt",quote=F)
```

```
for (i in  colnames(fit2$coefficients)){
  top=topTable(fit2,coef=i,sort="p", n=13456)
  genes=rownames(top[which(top$adj.P.Val<0.01 & abs(top$logFC)>2),])
  write.table(top,paste(i,"_limma_voom.txt",sep=""),quote=F)
  term1=strsplit(i,split="_vs_")[[1]][1]
  term2=strsplit(i,split="_vs_")[[1]][2]
  samples=rownames(subset(info,fraction==term1 | fraction==term2))
  expr=v$E[genes,samples]
  rownames(expr)=do.call(rbind, strsplit(genes, ','))[,1]
  if (length(genes) >1) {
    pheatmap(expr,scale="row",annotation_col=info[,c("Fraction","Sample")], border_color = "NA",show_ro
  }}
```

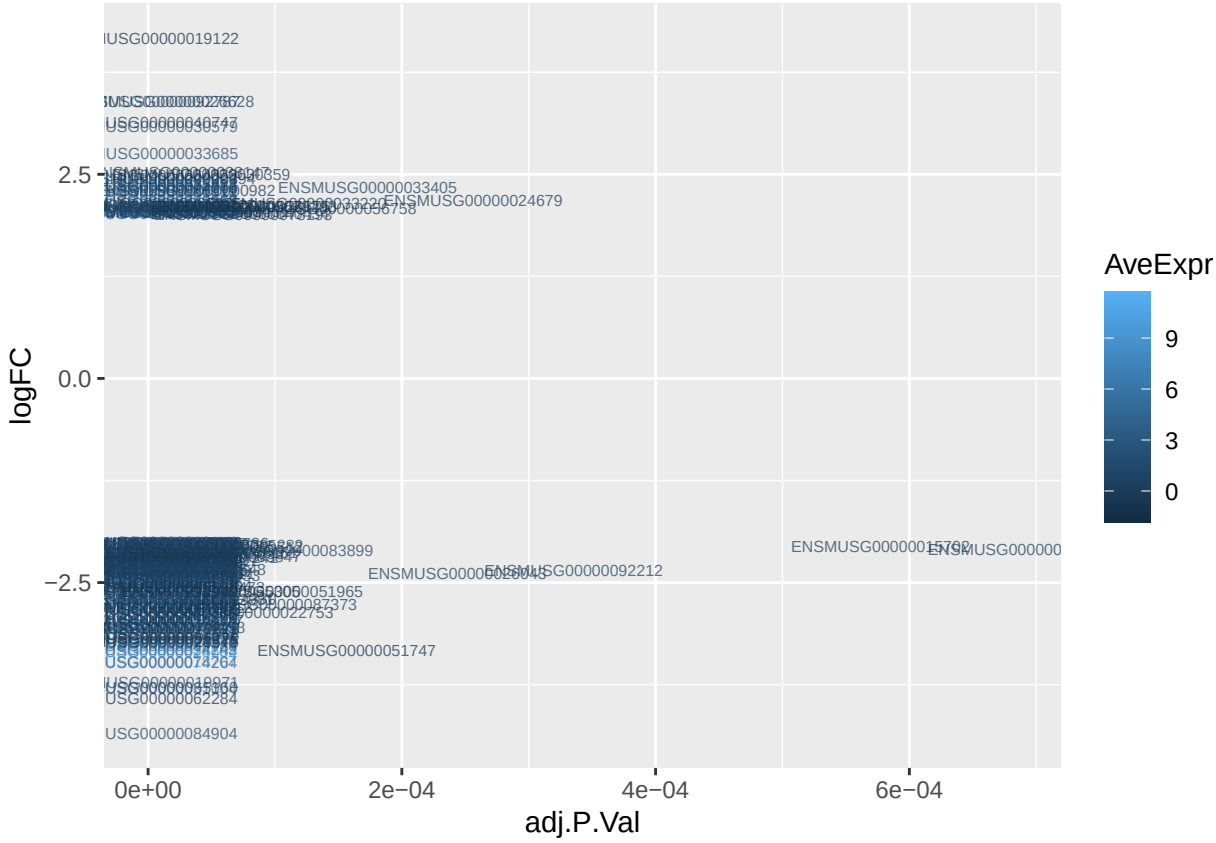


```
length(which(top$adj.P.Val < 0.01 & abs(top$logFC) > 2))
```

```
## [1] 306
```

```
p_data <-   top  %>% filter(adj.P.Val <0.01 & abs(logFC) > 2)

p_data %>%  ggplot(aes(x=adj.P.Val,y=logFC)) +
  geom_text(label=rownames(p_data), size=2.2,alpha=0.7, aes(col=AveExpr))
```



Top tables

```
top$Gene <- rownames(top)
DEGs <- top %>%  arrange(logFC) %>% filter(adj.P.Val <0.01 & abs(logFC) > 2)
head(DEGs)
```

```
##                         logFC     AveExpr         t      P.Value    adj.P.Val
## ENSMUSG00000084904 -4.344373  0.45675725 -12.87227 1.726451e-08 1.890246e-07
## ENSMUSG00000062284 -3.922046 -1.60764566 -12.31525 2.864710e-08 2.703193e-07
## ENSMUSG00000031004 -3.796871  5.29118919 -29.45542 9.311549e-13 1.079689e-09
## ENSMUSG00000065160 -3.782379 -0.03038683 -11.90032 4.232075e-08 3.595126e-07
## ENSMUSG00000019971 -3.716758  1.01713028 -10.22523 2.321258e-07 1.335964e-06
## ENSMUSG00000074264 -3.479048  3.55091389 -20.55095 7.090956e-11 6.626104e-09
```

```
##                            B               Gene
## ENSMUSG00000084904  9.465442 ENSMUSG00000084904
## ENSMUSG00000062284  8.710028 ENSMUSG00000062284
## ENSMUSG00000031004 19.559172 ENSMUSG00000031004
## ENSMUSG00000065160  8.636020 ENSMUSG00000065160
## ENSMUSG00000019971  7.415789 ENSMUSG00000019971
## ENSMUSG00000074264 15.303904 ENSMUSG00000074264
```

```
write.table(DEGs, file = "./DEG.txt", row.names = F, sep = "\t", quote = F)
```

```
#genes_sin_version <- sub("\\.\\d+$", "", rownames(top))
top$Gene <- rownames(top)
top <- top[,c("Gene", names(top)[1:6])]
write.table(top, file = "./Cyt_v_Nuc.txt", row.names = F, sep = "\t", quote = F)
```

Análisis de significació biológica

Nos centraremos únicamente en la lista de genes "up-regulados" y "down-regulados" es decir diferencialmente expresados con un logFC mayor que seis (más expresados en "cytosol" que en "nucleo").

Para el análisis de enriquecimiento utilizaremos la función `enrichGO` del paquete `clusterProfiler muy parecida a las de otros paquetes como`GOstats`.

```
head(top)
```

```
##                                   Gene     logFC  AveExpr         t      P.Value
## ENSMUSG00000018707 ENSMUSG00000018707 -3.458027 9.394118 -34.25716 1.490925e-13
## ENSMUSG00000014602 ENSMUSG00000014602 -3.346214 9.774906 -32.74156 2.582927e-13
## ENSMUSG00000063077 ENSMUSG00000063077 -3.239485 9.010497 -32.40273 2.930533e-13
## ENSMUSG00000026764 ENSMUSG00000026764 -3.335681 9.072041 -31.72330 3.789737e-13
## ENSMUSG00000034243 ENSMUSG00000034243 -3.313622 5.748709 -31.55674 4.039615e-13
## ENSMUSG00000028364 ENSMUSG00000028364 -2.381221 7.692305 -31.16199 4.705851e-13
##                       adj.P.Val        B
## ENSMUSG00000018707 1.055366e-09 21.54248
## ENSMUSG00000014602 1.055366e-09 21.03362
## ENSMUSG00000063077 1.055366e-09 20.90532
## ENSMUSG00000026764 1.055366e-09 20.66371
## ENSMUSG00000034243 1.055366e-09 20.42183
## ENSMUSG00000028364 1.055366e-09 20.44143
```

```
allEntrezs <- rownames(top)
selectedEntrezsUP <- rownames(subset(top, (abs(logFC) > 2) & (adj.P.Val < 0.01)))
length(allEntrezs); length(selectedEntrezsUP)
```

```
## [1] 13456
```

```
## [1] 306
```

```
library(clusterProfiler)
library(org.Mm.eg.db)
ego <- enrichGO(gene = selectedEntrezsUP,
                universe = allEntrezs,
                keyType = "ENSEMBL",
                OrgDb = org.Mm.eg.db,
                ont = "BP",
                pAdjustMethod = "BH",
                qvalueCutoff = 0.01,
                readable = TRUE)
```

El objeto resultante almacena las categorías GO enriquecidas, los genes anotados en ellas y los valores de los estadísticos que llevan a afirmar que dichas categorías se encuentran significativamente sobre-representadas como resultado de un test de enriquecimiento.

```
head(ego)
```

```
##                    ID                                   Description GeneRatio
## GO:0030199 GO:0030199                  collagen fibril organization     9/263
## GO:0008347 GO:0008347                          glial cell migration    10/263
## GO:0030198 GO:0030198             extracellular matrix organization    17/263
## GO:0043062 GO:0043062          extracellular structure organization    17/263
## GO:0045229 GO:0045229 external encapsulating structure organization    17/263
## GO:0051493 GO:0051493       regulation of cytoskeleton organization    26/263
##              BgRatio       pvalue    p.adjust       qvalue
## GO:0030199  44/12308 2.986168e-07 0.001015894 0.0008747900
## GO:0008347  62/12308 6.763865e-07 0.001150533 0.0009907282
## GO:0030198 207/12308 2.151050e-06 0.001267688 0.0010916107
## GO:0043062 207/12308 2.151050e-06 0.001267688 0.0010916107
## GO:0045229 207/12308 2.151050e-06 0.001267688 0.0010916107
## GO:0051493 436/12308 2.235782e-06 0.001267688 0.0010916107
##
## GO:0030199
## GO:0008347
## GO:0030198                                                  Mia3/App/Lamb1/Fn1/Lamb2/Col4a2/Acan/Hspg2/
## GO:0043062                                                  Mia3/App/Lamb1/Fn1/Lamb2/Col4a2/Acan/Hspg2/
## GO:0045229                                                  Mia3/App/Lamb1/Fn1/Lamb2/Col4a2/Acan/Hspg2/
## GO:0051493 Dync1h1/Sptan1/Ckap5/Nes/Mycbp2/Tmsb10/Cltc/Tpr/Plxna3/Sptbn1/Agrn/Tmsb4x/Tenm1/Akap9/Map
##            Count
## GO:0030199     9
## GO:0008347    10
## GO:0030198    17
## GO:0043062    17
## GO:0045229    17
## GO:0051493    26
```

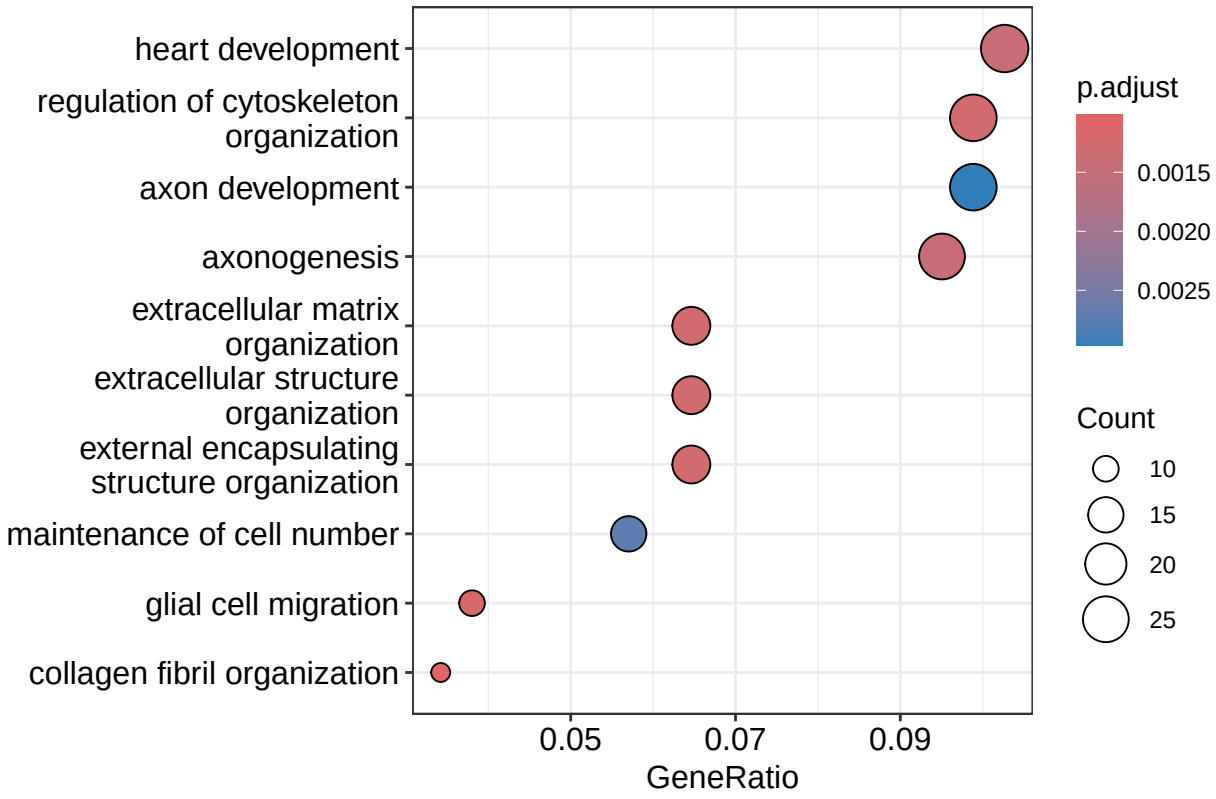
```
ego_results <- data.frame(ego)
write.csv(ego_results, "clusterProfiler_ORAresults_UpGO.csv")
```

Visualización de los resultados del análisis de enriquecimiento

Uno de los aspectos interesantes del paquete `clusterProfiler` es que permite visualizar los resultados mediante algunos gráficos creados específicamente para tal fin.

##Dotplot de los 9 términos más enriquecidos Este gráfico compara visualmente las categorías enriquecidas (de más a menos enriquecidas) visualizando simultáneamente cuan enriquecidas estan y el p-valor del test de enriquecimiento.

```
dotplot(ego, showCategory=10)
```




```
pdf(paste("dotplot.pdf",sep=""))
dotplot(ego, showCategory=10)
dev.off()
```

```
## pdf
##   2
```

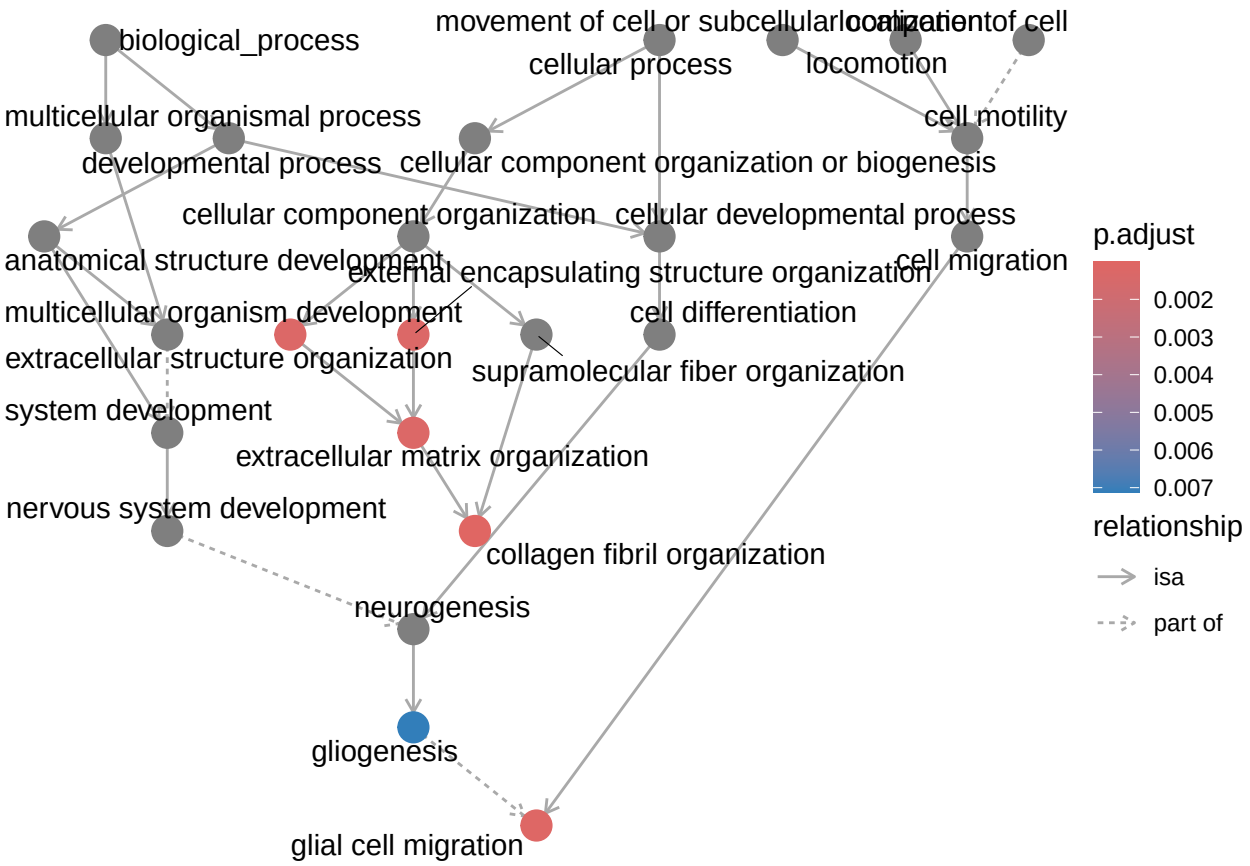
Visualización jerárquica de los términos GO

Este gráfico permite visualizar los términos seleccionados dentro del sub-grafo de la GO que los contiene. Esto nos, permite por ejemplo, hacernos una idea de si estan muy dispersos, o no, en la jerarquía y de si se trata de términos muy generales o más específicos.

```
pdf(paste("GO.pdf",sep=""))
goplot(ego, showCategory=5, cex=0.5)
dev.off()
```

```
## pdf
##   2
```

```
goplot(ego, showCategory=5, cex=0.5)
```

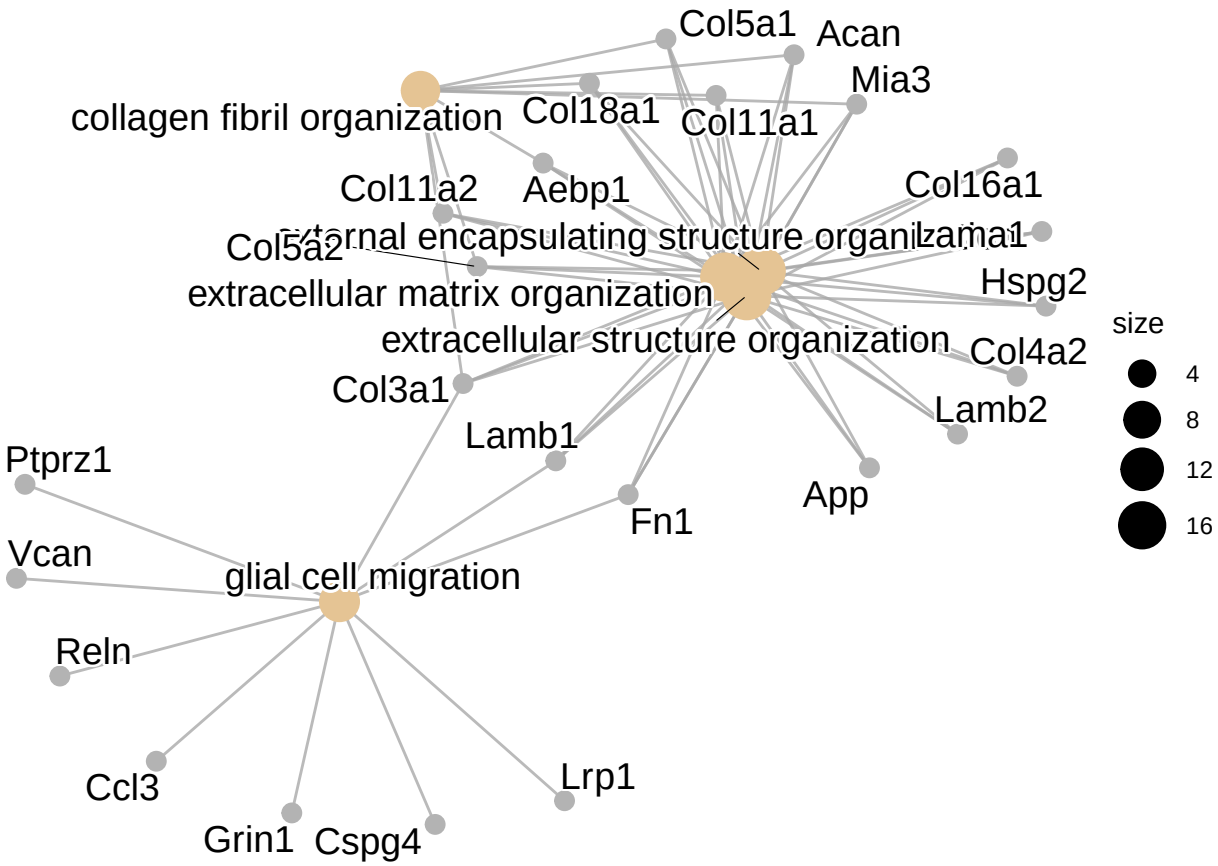


De forma parecida una red de genes nos permite visualizar la asociación entre los genes y las categorías seleccionadas en las que éstos genes estan anotados.

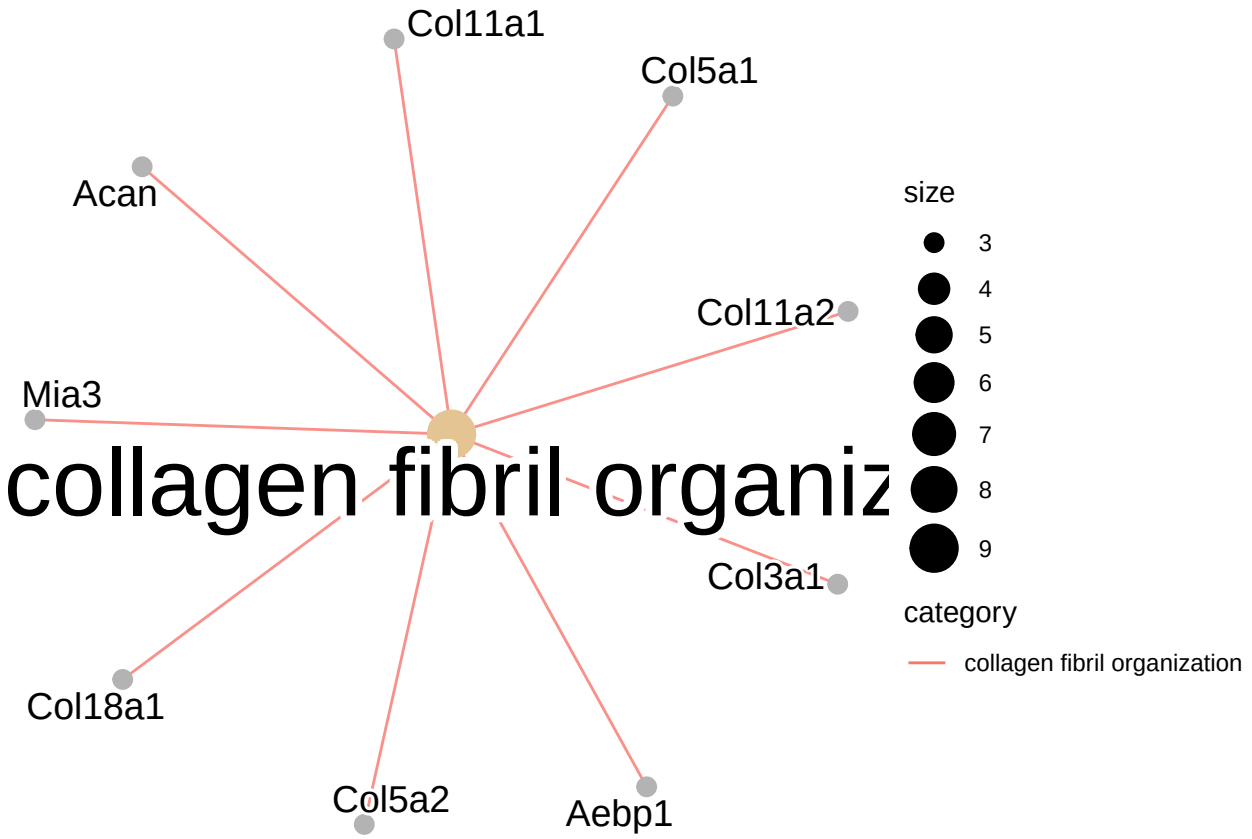
```
## Gene network para los términos seleccionados
pdf(paste("cneplot.pdf",sep=""))
cnetplot(ego)
dev.off()
```

```
## pdf
##   2
```

```
cnetplot(ego)
```




```
library(clusterProfiler)
library(ggplot2)
ego2 = clusterProfiler::simplify(ego, cutoff = 0.01, by = "p.adjust")
png("./cnetplot_transp.png", units = "in", width = 24, height = 16, res = 600,
    bg = "transparent")
par(bg = NA)
a <- cnetplot(ego2, showCategory = 5, cex_category = 1, cex_label_category = 2.5,
    cex_gene = 1, cex_label_gene = 1, circular = FALSE, colorEdge = TRUE)
a
invisible(dev.off())
a
```

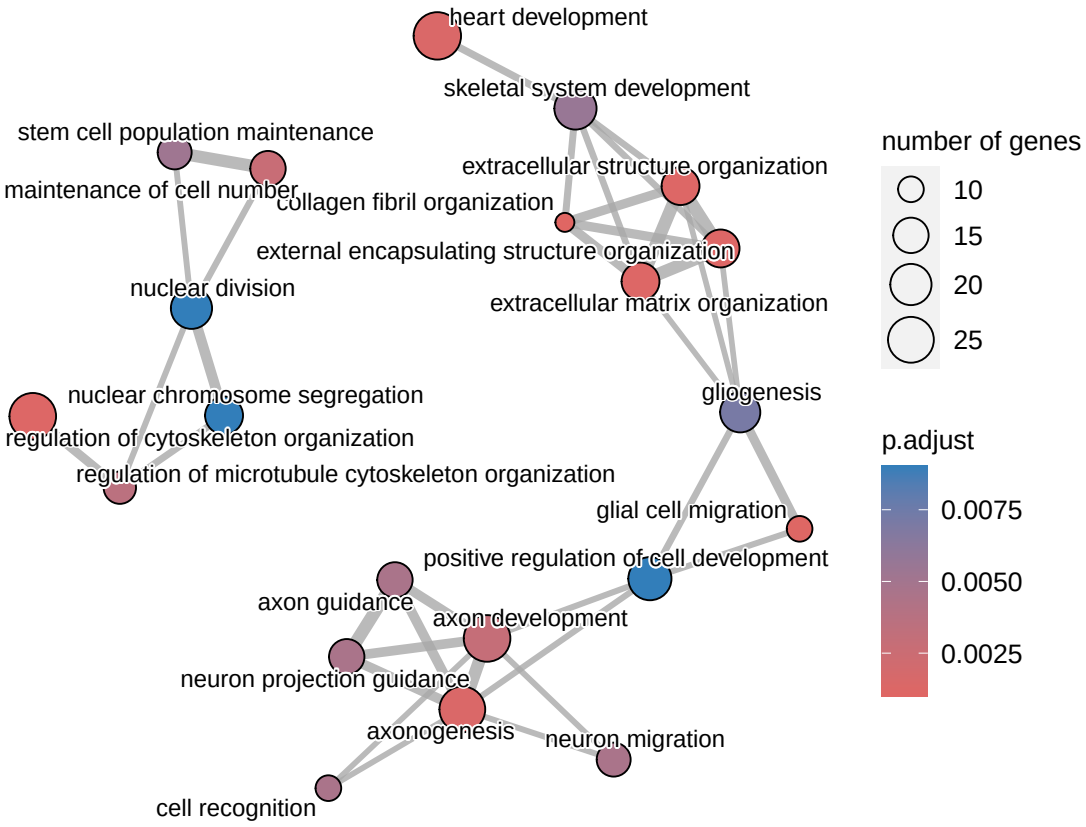


Finalmente este gráfico permite simplificar las visualizaciones y agrupa los 104 términos más significativos basándose en alguna medida de similaridad entre los mismos (por ejemplo "similaridad semántica" definida a partir de su interdistancia dentro del grafo).

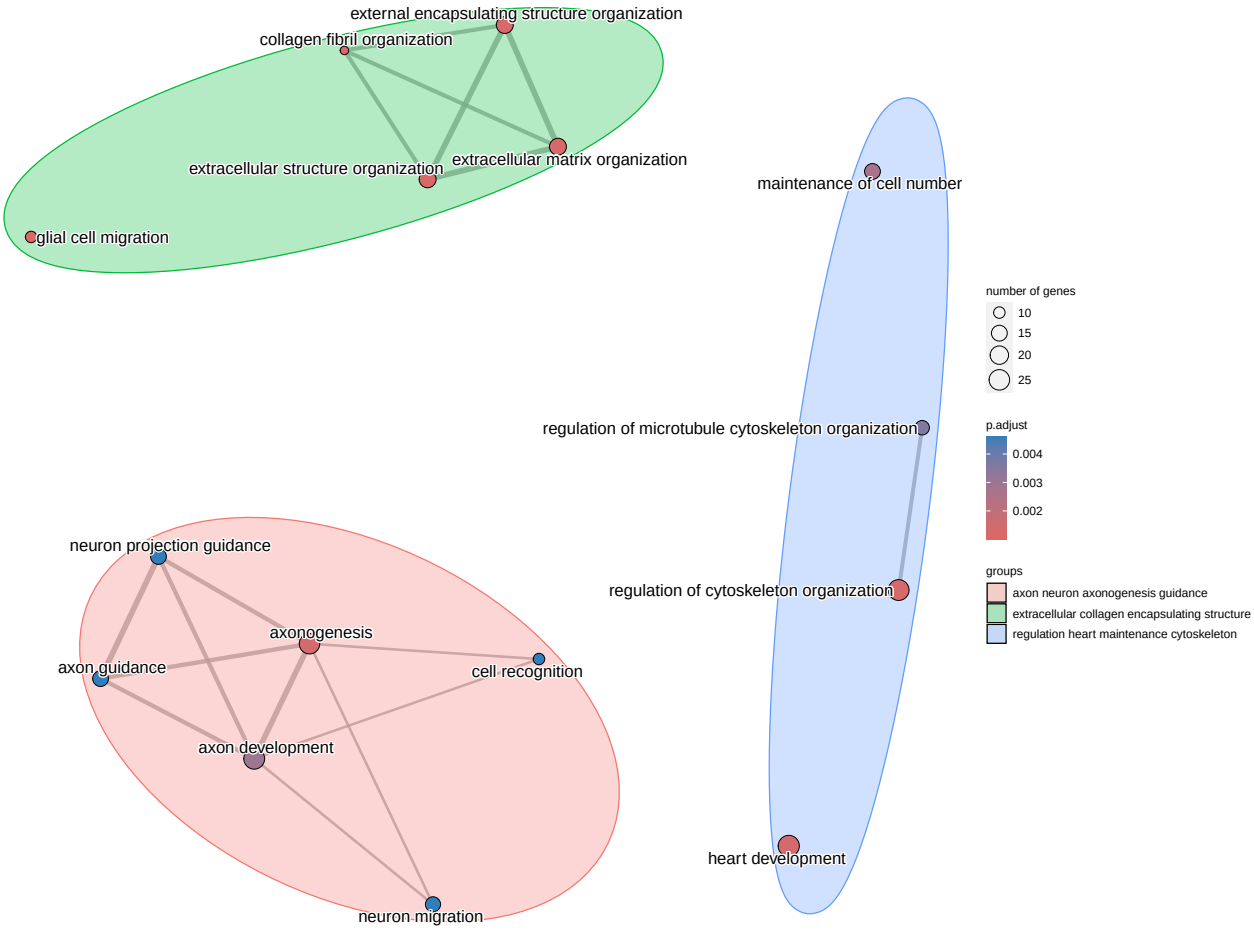
```
## Enrichment Map
library(enrichplot)
ego_sim <- pairwise_termsim(ego)
pdf(paste("emaplot.pdf",sep=""))
emapplot(ego_sim, cex_label_category=0.6)
dev.off()
```

```
## pdf
##   2
```

```
emapplot(ego_sim, cex_label_category=0.6)
```



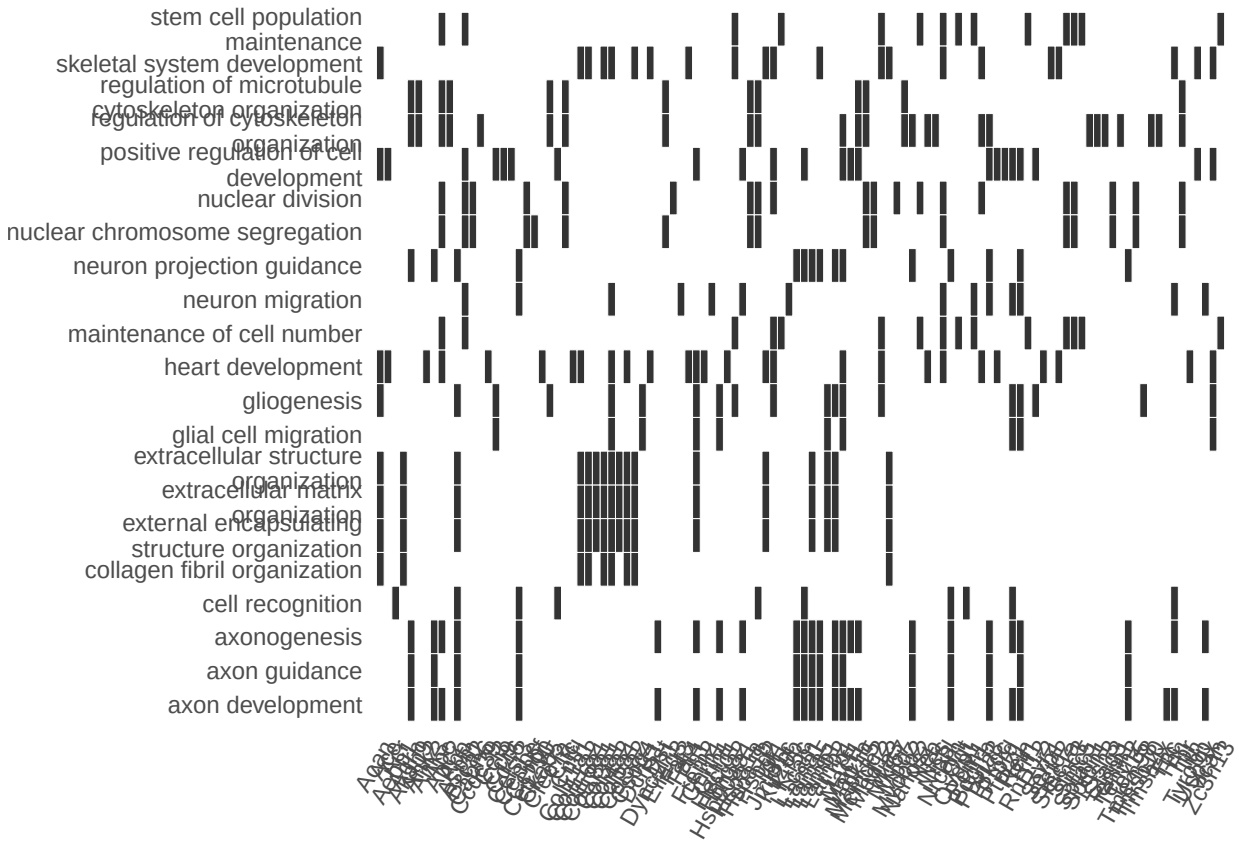

```
term_similarity_matrix = pairwise_termsim(ego)
emapplot(ego_sim, showCategory = 15, group_category = TRUE, group_legend = TRUE)
```




```
pdf(paste("emaplot_grouped.pdf",sep=""),width = 15, height = 17)
emapplot(term_similarity_matrix, showCategory = 15, group_category = TRUE, group_legend = TRUE)
dev.off()
```

```
## pdf
##   2
```

```
library(enrichplot)
heatplot(ego)
```




```
pdf(paste("heatplot_ego.pdf",sep=""))
heatplot(ego)
dev.off()
```

```
## pdf
##   2
```