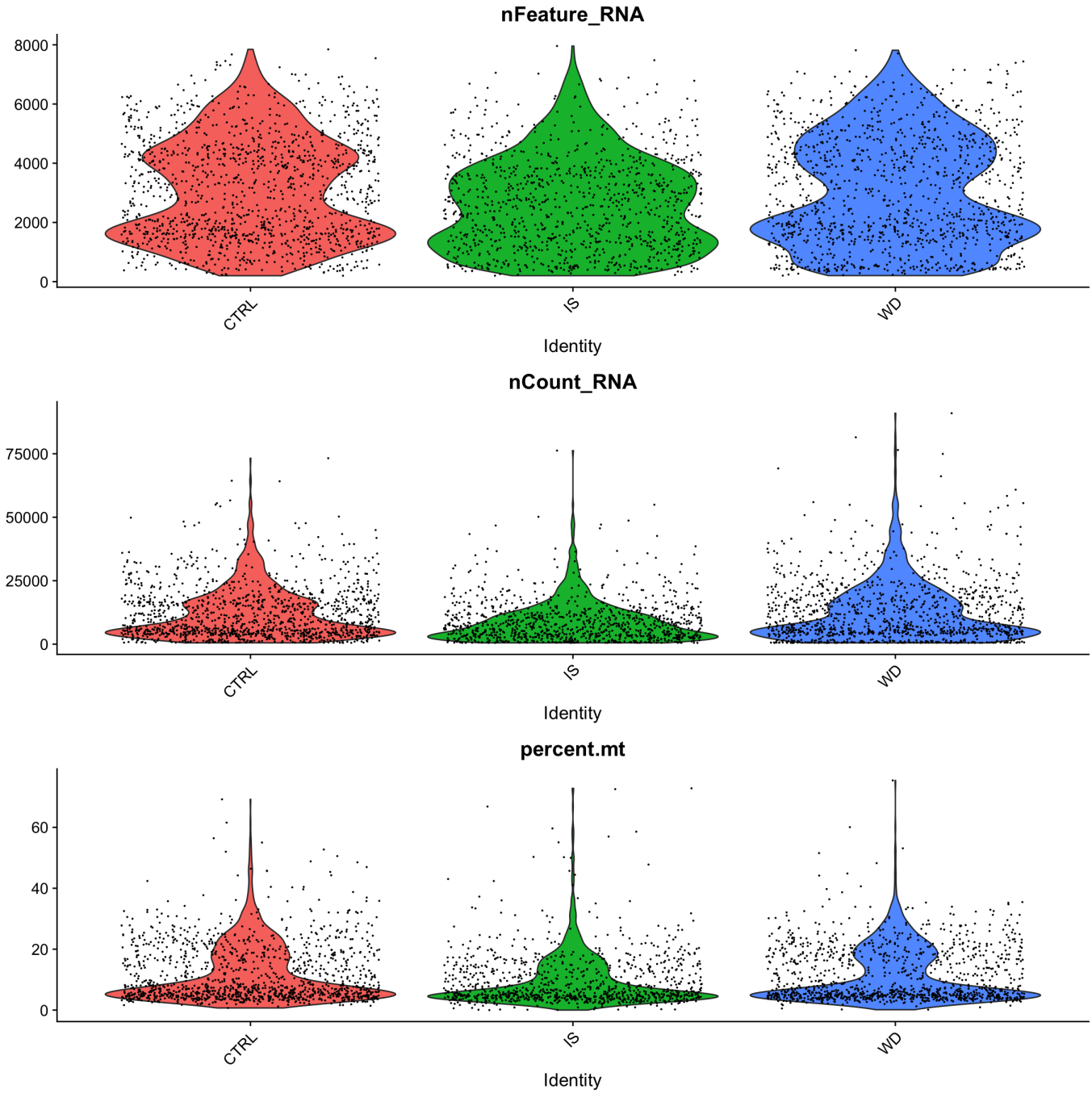
R Notebook

Code

```
Samples
CTRL   IS   WD
1303 1033 1181
```

Hide

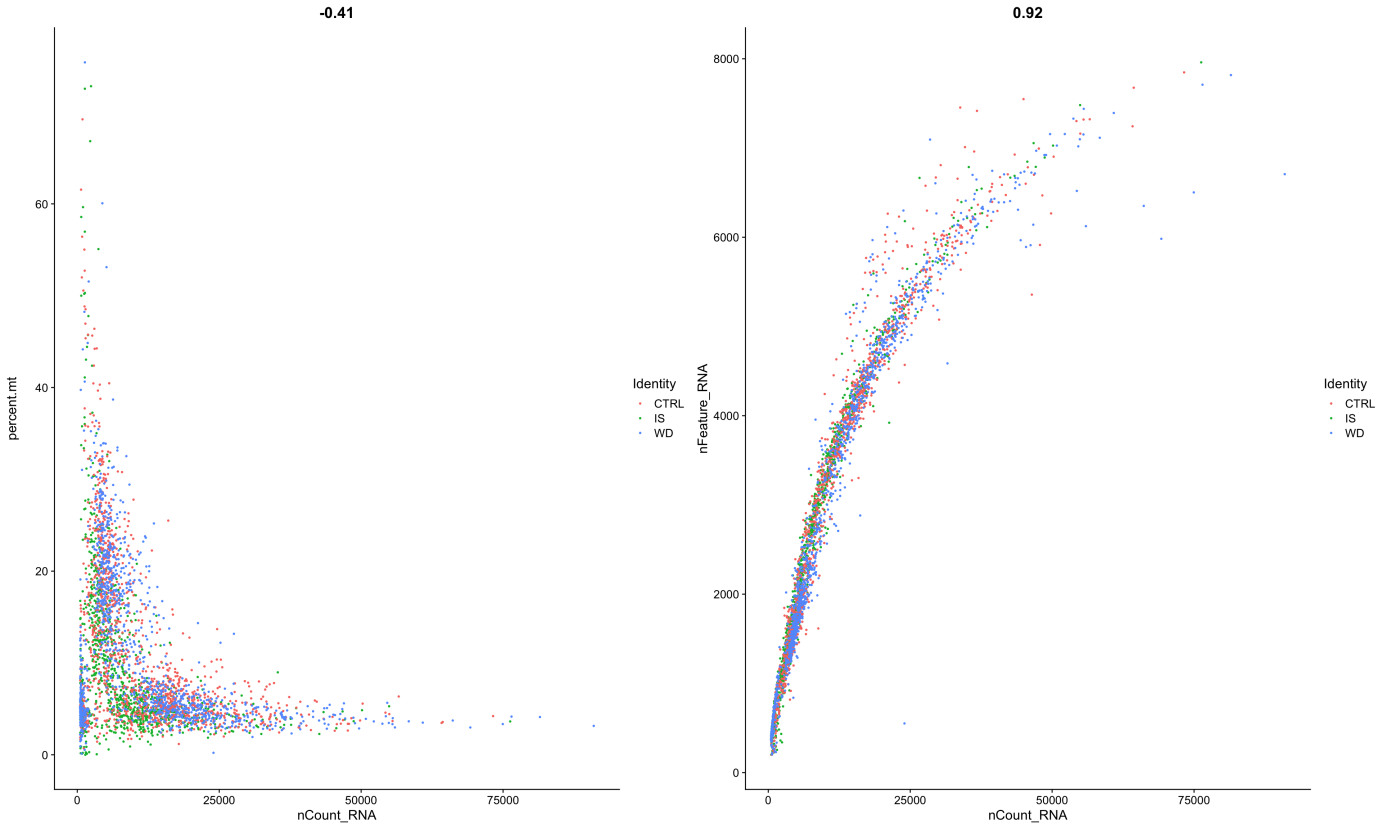
```
# The [[ operator can add columns to object metadata. This is a great place to stash
 QC stats
oligos[["percent.mt"]] <- PercentageFeatureSet(oligos, pattern = "^mt-")
# Visualize QC metrics as a violin plot
VlnPlot(oligos, group.by = "Sample",features = c("nFeature_RNA", "nCount_RNA", "perce
nt.mt"), ncol = 1,pt.size = 0.1)
```



Hide

```
# FeatureScatter is typically used to visualize feature-feature relationships, but ca
n be used
# for anything calculated by the object, i.e. columns in object metadata, PC scores e
tc.
plot1 <- FeatureScatter(oligos, group.by = "Sample",feature1 = "nCount_RNA", feature2
= "percent.mt",pt.size = 0.5)
plot2 <- FeatureScatter(oligos, group.by = "Sample", feature1 = "nCount_RNA", feature
2 = "nFeature_RNA",pt.size = 0.5)
CombinePlots(plots = list(plot1, plot2))
```

```
CombinePlots is being deprecated. Plots should now be combined using the patchwork sy
stem.
```



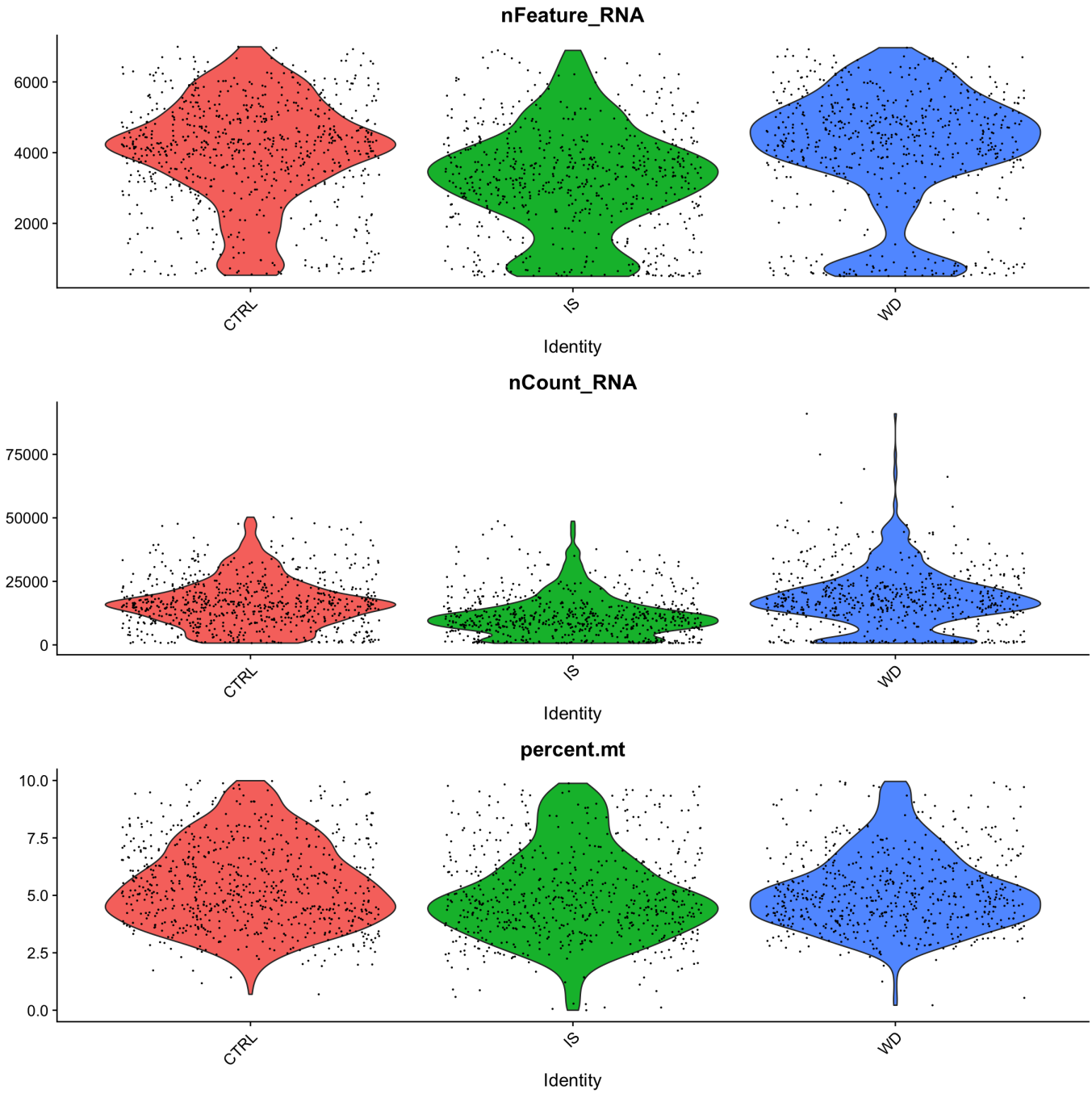
Hide

```
#Clean up the data
oligos <- subset(oligos, subset = nFeature_RNA > 500 & nFeature_RNA < 7000 & percent.
mt < 10)
ncol(oligos)
```

```
[1] 1975
```

Hide

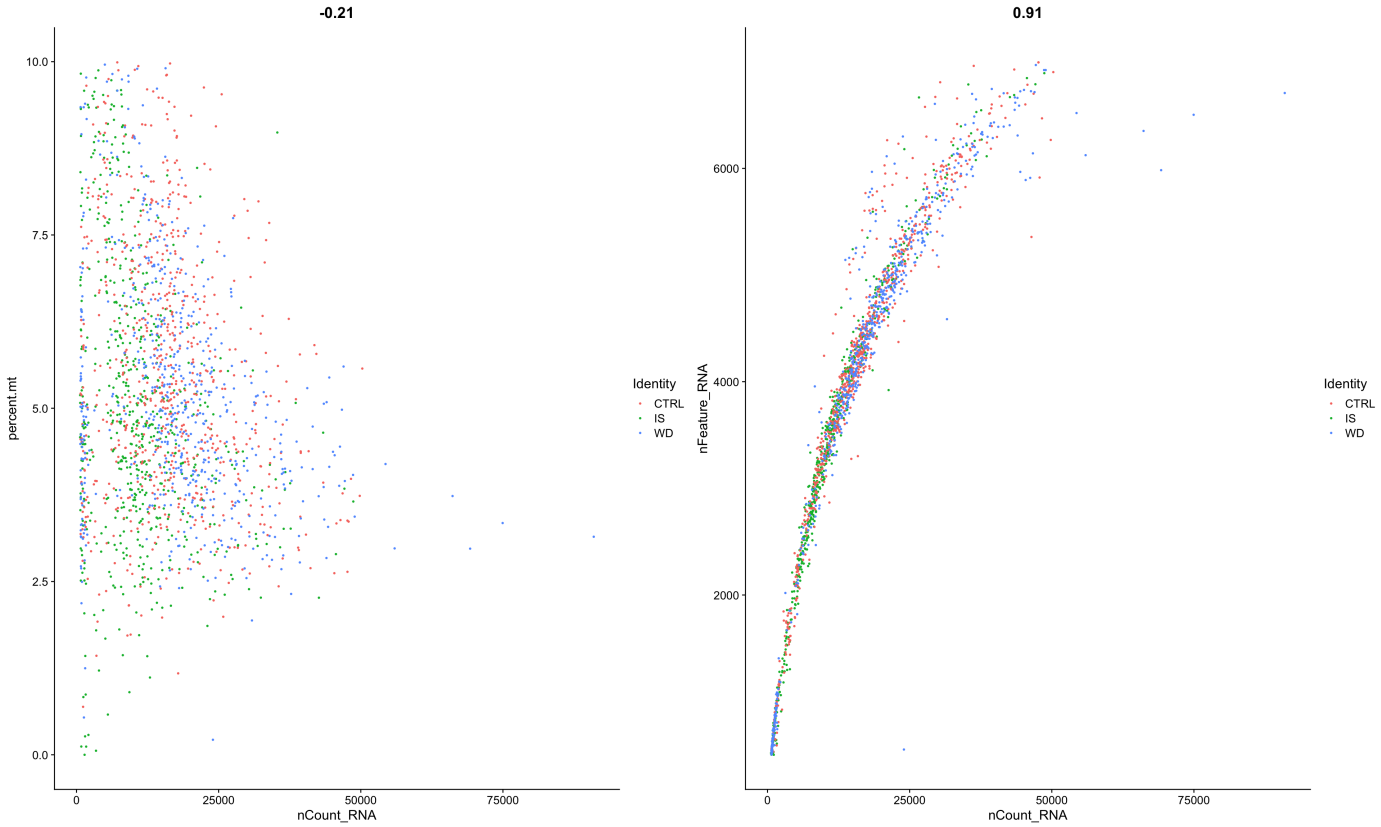
```
# The [[ operator can add columns to object metadata. This is a great place to stash
 QC stats
oligos[["percent.mt"]] <- PercentageFeatureSet(oligos, pattern = "^mt-")
# Visualize QC metrics as a violin plot
VlnPlot(oligos, group.by = "Sample",features = c("nFeature_RNA", "nCount_RNA", "perce
nt.mt"), ncol = 1,pt.size = 0.1)
```



Hide

```
# FeatureScatter is typically used to visualize feature-feature relationships, but ca
n be used
# for anything calculated by the object, i.e. columns in object metadata, PC scores e
tc.
plot1 <- FeatureScatter(oligos, group.by = "Sample",feature1 = "nCount_RNA", feature2
= "percent.mt",pt.size = 0.5)
plot2 <- FeatureScatter(oligos, group.by = "Sample", feature1 = "nCount_RNA", feature
2 = "nFeature_RNA",pt.size = 0.5)
CombinePlots(plots = list(plot1, plot2))
```

```
CombinePlots is being deprecated. Plots should now be combined using the patchwork sy
stem.
```



Now we normalize the dataset.

Generating the UMAP and TSNE.

Hide

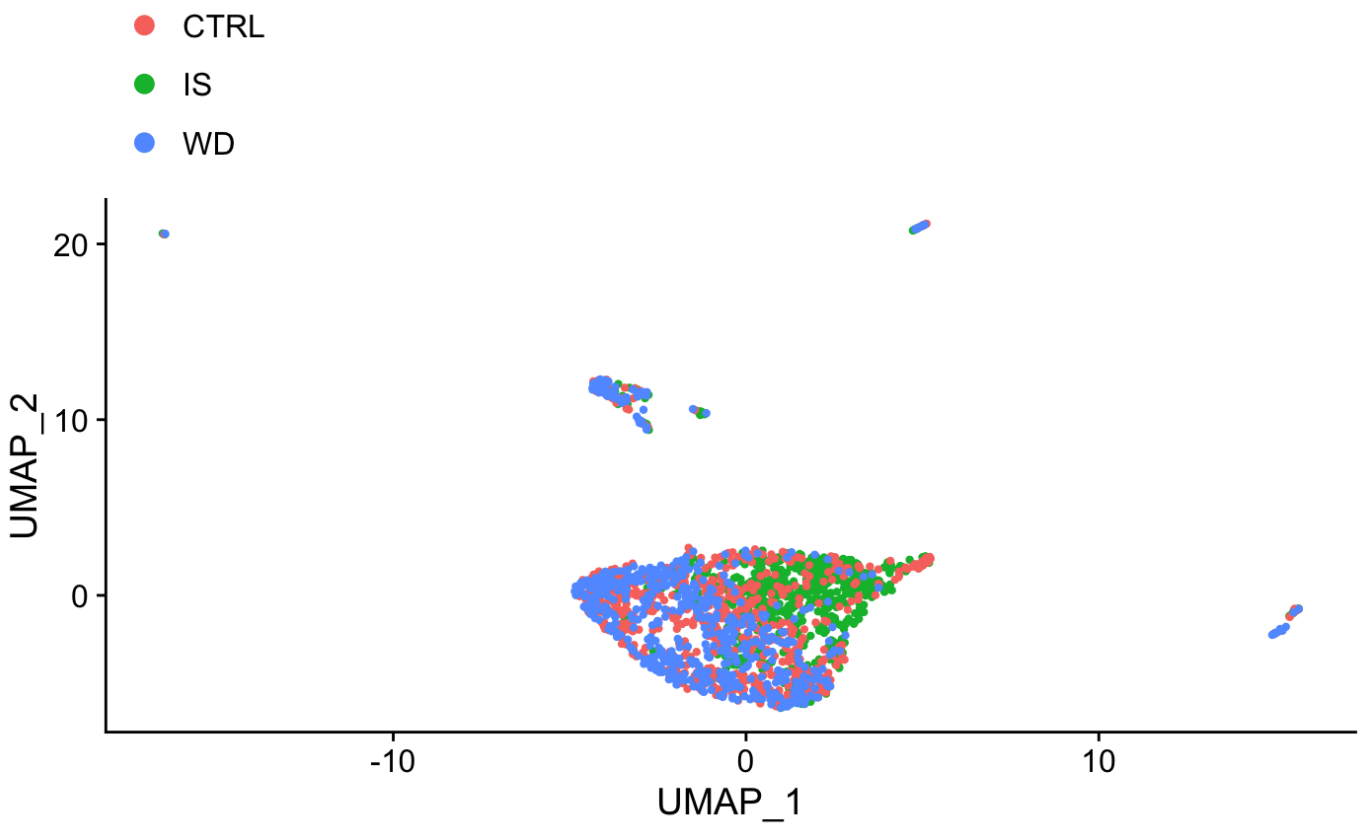
```
#DefaultAssay(oligos.integrated) <- "integrated"
oligos.integrated <- RunPCA(oligos.integrated, verbose = FALSE)
oligos.integrated <- RunUMAP(oligos.integrated, dims = 1:30)
```

```
The default method for RunUMAP has changed from calling Python UMAP via reticulate to
the R-native UWOT using the cosine metric
To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'cor
relation'
This message will be shown once per session16:37:23 UMAP embedding parameters a = 0.9
922 b = 1.112
16:37:23 Read 1975 rows and found 30 numeric columns
16:37:23 Using Annoy for neighbor search, n_neighbors = 30
16:37:23 Building Annoy index with metric = cosine, n_trees = 50
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
16:37:24 Writing NN index file to temp file /var/folders/98/j14xl9ln5190l9fvq32g5rkw0
000gn/T//RtmpSdWnAZ/file3f2e39773684
16:37:24 Searching Annoy index using 1 thread, search_k = 3000
16:37:24 Annoy recall = 100%
16:37:25 Commencing smooth kNN distance calibration using 1 thread
16:37:25 Initializing from normalized Laplacian + noise
16:37:28 Commencing optimization for 500 epochs, with 77718 positive edges
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
16:37:31 Optimization finished
```

Hide

```
#oligos.integrated <- RunTSNE(oligos.integrated, dims = 1:30)
plots <- DimPlot(oligos.integrated, group.by = c("Sample"), combine = FALSE)
plots <- lapply(X = plots, FUN = function(x) x + theme(legend.position = "top") + gui
des(color = guide_legend(nrow = 3,
    byrow = TRUE, override.aes = list(size = 3))))
CombinePlots(plots)
```

```
CombinePlots is being deprecated. Plots should now be combined using the patchwork sy
stem.
```



Hide

```
# plots <- TSNEPlot(oligos.integrated, group.by = c("Sample"), combine = FALSE)
# plots <- lapply(X = plots, FUN = function(x) x + theme(legend.position = "top") + g
uides(color = guide_legend(nrow = 3,
#     byrow = TRUE, override.aes = list(size = 3))))
# CombinePlots(plots)
```

Label transfer

Now we attempt to transfer the cluster labels of the Science dataset onto the 10X dataset.

Hide

```
oligos.integrated <- FindNeighbors(oligos.integrated, dims = 1:30)
```

```
Computing nearest neighbor graph
Computing SNN
```

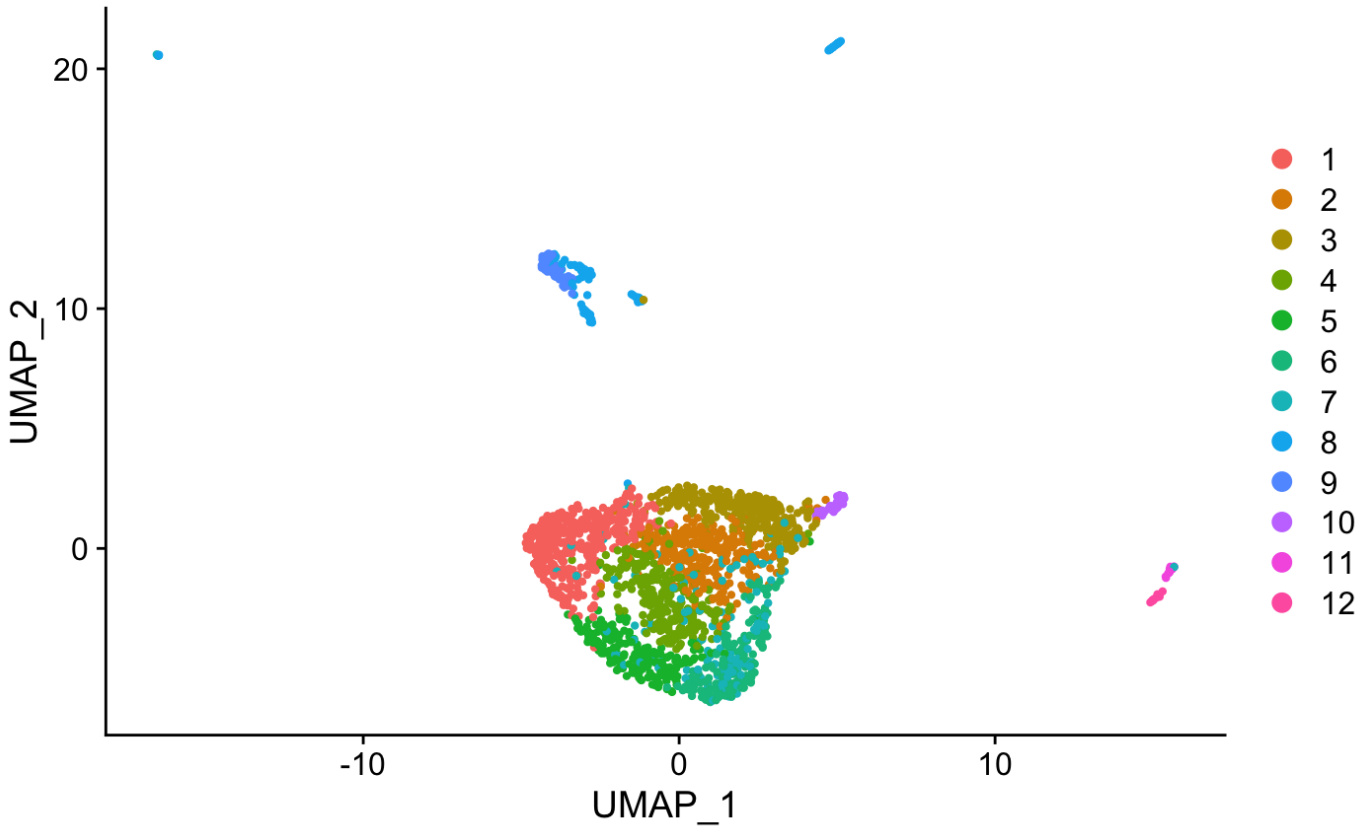
Hide

```
oligos.integrated <- FindClusters(oligos.integrated,algorithm = 4,resolution = 0.6)
#0.6
```

Hide

```
oligos.integrated$predicted.id <- factor(oligos.integrated$predicted.id,levels=c("OP
C","COP","NFOL1","MFOL1","MFOL2","MOL1","MOL2","MOL3","MOL4","MOL5","MOL6","PPR"))
DimPlot(oligos.integrated, group.by = c("seurat_clusters"), combine = FALSE)
```

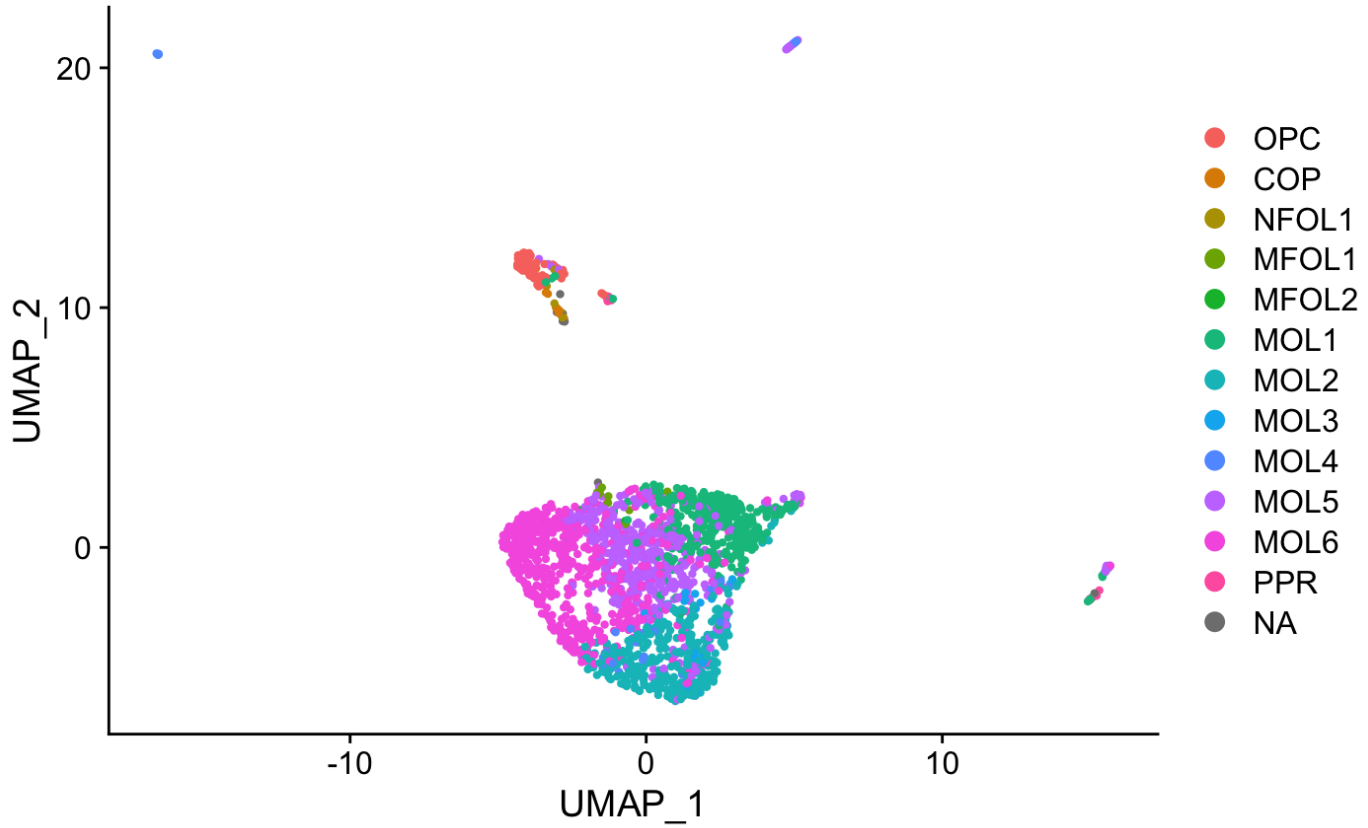
```
[[1]]
```



Hide

```
DimPlot(oligos.integrated, group.by = c("predicted.id"), combine = FALSE)
```

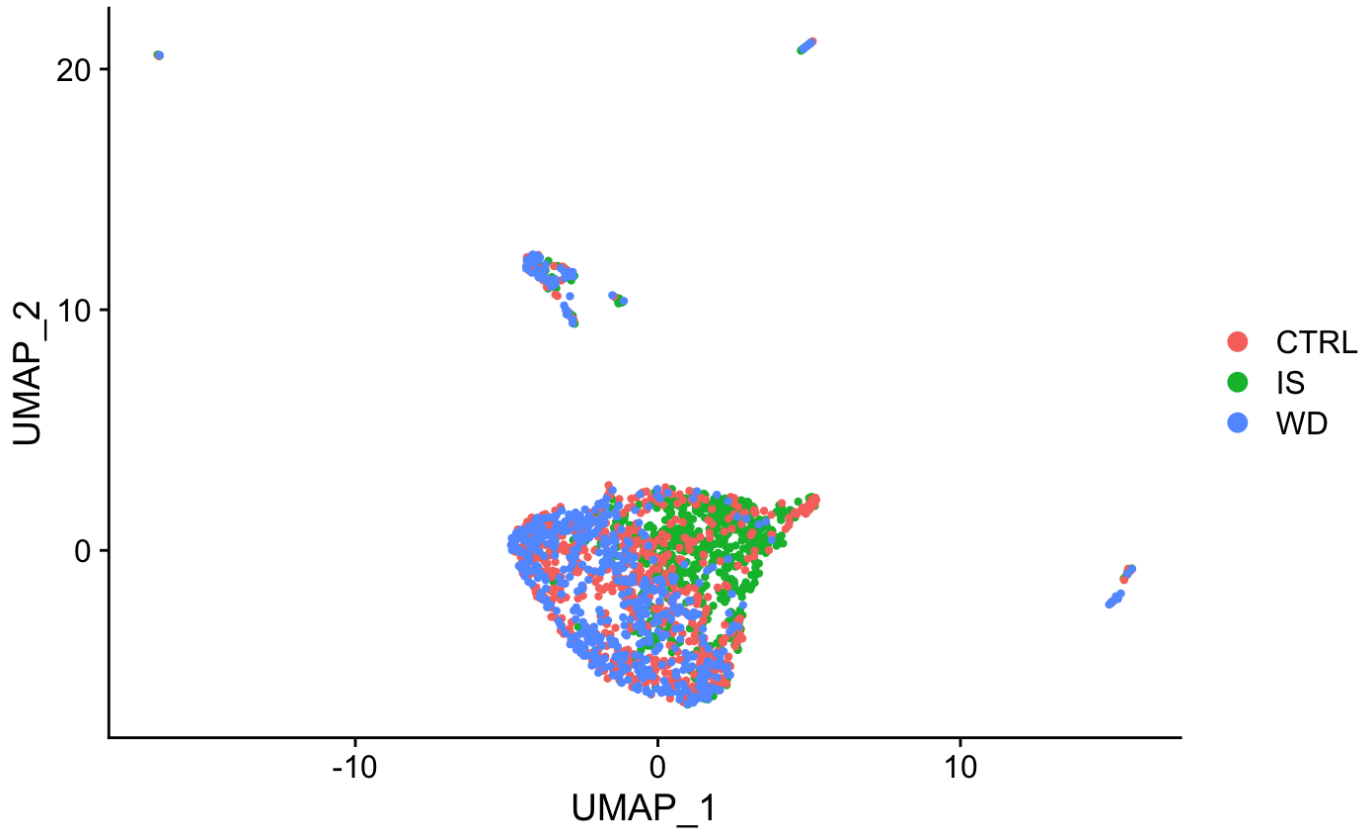
```
[[1]]
```



Hide

```
DimPlot(oligos.integrated, group.by = c("Sample"), combine = FALSE)
```

```
[[1]]
```



Hide

```
table(oligos.integrated$Sample,oligos.integrated$predicted.id)
```

```
      OPC COP NFOL1 MFOL1 MFOL2 MOL1 MOL2 MOL3 MOL4 MOL5 MOL6 PPR
  CTRL  37   5     2     6     2   77  153     8   15  156  264   1
  IS     8   1     5     1     1  272   80     6   15  180   62  12
  WD    33   5     3     2     0   26  122     5   10  118  269   4
```

Hide

```
#subset OPCs
oligos.integratedIm <- subset(oligos.integrated,seurat_clusters %in% c(12,11,8,9))
oligos.integratedIm <- FindVariableFeatures(oligos.integratedIm)
```

```
Calculating gene variances
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Calculating feature variances of standardized and clipped values
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
```

Hide

```
DefaultAssay(oligos.integratedIm) <- "SCT"
```

#Spatially filter genes with auto bootstrapping #Script will give error when running out of genes, this is normal

Hide

```
networkExpressionFile=oligos.integratedIm@assays$SCT@scale.data
expr_limit <- min(networkExpressionFile)-0.01
featureselection <- row.names(networkExpressionFile)
gc()
```

```
           used   (Mb) gc trigger   (Mb)  max used   (Mb)
Ncells   2854600  152.5    5203605  278.0   3953528  211.2
Vcells 350219390 2672.0  648326055 4946.4 645570740 4925.4
```

Hide

```
library(umap)
useUMAP <- "TRUE"
UMAPdim <- 3
adjustforbias <- "FALSE"
GeneMarkovList <- list()
GeneFilterProgression <- as.data.frame(featureselection)
row.names(GeneFilterProgression) <- featureselection
corenumber <- 4
usemagic <- "FALSE"
pcathresh <- 20
ncompGF <- 20
whentobootstrap <- 20000 #cellnumber
howmanyboots <- 10+1 #bootstrap repetitions
samplesize <- 2000 #how many cells to take for sampling
threshold <- 3 #use mean spatial correlation of geneset of the featureselection (1) o
r after the first round of filtering (2) (more strict, which is default)
nsim <- 100
nnquant <- 0.995 #lower for small datasets, higher for speedups in larger datasets
set.seed(1234)
tri.to.squ<-function(x)
{
rn<-row.names(x)
cn<-colnames(x)
an<-unique(c(cn,rn))
myval<-x[!is.na(x)]
mymat<-matrix(1,nrow=length(an),ncol=length(an),dimnames=list(an,an))
for(ext in 1:length(cn))
{
 for(int in 1:length(rn))
 {
 if(is.na(x[row.names(x)==rn[int],colnames(x)==cn[ext]])) next
 mymat[row.names(mymat)==rn[int],colnames(mymat)==cn[ext]]<-x[row.names(x)==rn[int],c
olnames(x)==cn[ext]]
 mymat[row.names(mymat)==cn[ext],colnames(mymat)==rn[int]]<-x[row.names(x)==rn[int],c
olnames(x)==cn[ext]]
 }
}
return(mymat)
}
SCN3Egeneset <- featureselection
originalnetworkdf <- networkExpressionFile
if(ncol(networkExpressionFile)>whentobootstrap){
originalnetworkdf <- networkExpressionFile
bootstrap<-1
sample<-1
moransIsampled <- as.character(NULL)
}
if(ncol(networkExpressionFile)<=whentobootstrap)
{
  bootstrap<-howmanyboots-1
  sample<-0
  }
while(bootstrap<howmanyboots){
SCN3Egeneset <- featureselection
iter=0
meanMoran<-0
if(sample==1){
```

```
networkExpressionFile  <- originalnetworkdf[,sample(ncol(originalnetworkdf), samplesi
ze) ]
print(paste("bootstrapping...","round",bootstrap))
}
OldGeneset <- as.character(seq_len(100000))
while(length(OldGeneset) > (length(SCN3Egeneset)+10) ){ # uncomment this and below to
enable iterative filtering
 iter=iter+1
  #library(amap)
#library(MASS)
#library(destiny)
#library(dpt)
#library(Matrix)
#library(diffusionMap)
print(paste("Preparing genefiltering on", length(SCN3Egeneset), "genes."))
#print("Making celllandscape...Filtering possible duplicated cells in original file")
#if(length(OldGeneset)==100000)
#{
#  pca <- prcomp(t(log(networkExpressionFile[featureselection,]+1)),scale. = FALSE)
#  cd_diffusionplot <-pca$x[,c(1:30)]
#  }
#if(length(OldGeneset)!=100000){cd_diffusionplot <- t(log(networkExpressionFile[SCN3E
geneset,]+1))}
  if(length(SCN3Egeneset) > pcathresh){
#library(rsvd)
    print("Making celllandscape...Performing dimensionality reduction")
#pca <- rpca(t(log(networkExpressionFile[SCN3Egeneset,]+1)))
#pca <- prcomp(t(log(networkExpressionFile[SCN3Egeneset,]+1)),scale. = TRUE)
pca <- prcomp(t(networkExpressionFile[SCN3Egeneset,]),scale. = TRUE)
if(length(SCN3Egeneset) < 100) {ncompGF <- length(SCN3Egeneset)}
cd_diffusionplot <-pca$x[,c(1:which(cumsum(pca$sdev[1:ncompGF])/sum(pca$sdev[1:ncompG
F]) > 0.8)[1])]
  }
   if(length(SCN3Egeneset) <= pcathresh){
#cd_diffusionplot <-t(log(networkExpressionFile[SCN3Egeneset,]+1))
cd_diffusionplot <-t(networkExpressionFile[SCN3Egeneset,])
}
cd_diffusionplot <- cd_diffusionplot[!duplicated(cd_diffusionplot),]
print("Making celllandscape...Making diffusionmap")
# gc()
#ts <- Transitions(cd_diffusionplot,k=20)
if(useUMAP==TRUE)
{
  library(umap)
umap.settings <- umap.defaults
umap.settings$n_neighbors <-10
umap.settings$min_dist <- 0
umap.settings$n_components <- UMAPdim
umap.settings$metric <- "manhattan"
#umap.settings$local_connectivity <- 0.00001
umap.settings$n_epochs <- 1000
umap_out <- umap(cd_diffusionplot,config = umap.settings,method="umap-learn")
}
if(useUMAP != TRUE){
library(destiny)
ts <- DiffusionMap(cd_diffusionplot, verbose = TRUE)
}
if(adjustforbias==TRUE){
```

```
testwilcox <- as.matrix(apply(t(ts@eigenvectors),1,function(x) apply(tri.to.squ(pairw
ise.wilcox.test(x,biasedfactor,p.adjust.method = "fdr")$p.value),2,function(x) min(x,
na.rm=TRUE))))
testwilcoxdifference <- scale(t(testwilcox))
biasedcomponents <- unique(unlist(apply(testwilcoxdifference,2,function(x) which(abs
(x) > 1.96))))
}
#Gene filtering
print(paste("Filtering", length(SCN3Egeneset), "genes."))
OldGeneset <- SCN3Egeneset
#NetworkDist <- as.matrix(ts@transitions)
library(amap)
range01 <- function(x){(x-min(x))/(max(x)-min(x))}
if(useUMAP==TRUE)
  {
  if(adjustforbias==TRUE){
NetworkDist <- 1-range01((as.matrix(Dist(umap_out$layout[,1:UMAPdim],method = "manhat
tan",nbproc = 8))))}
else{NetworkDist <- 1-range01((as.matrix(Dist(umap_out$layout[,1:UMAPdim],method = "m
anhattan",nbproc = 8))))}
}
if(useUMAP != TRUE){
if(adjustforbias==TRUE){
NetworkDist <- 1-range01((as.matrix(Dist(ts@eigenvectors[,-c(biasedcomponents)],metho
d = "manhattan",nbproc = 8))))}
else{NetworkDist <- 1-range01((as.matrix(Dist(ts@eigenvectors,method = "manhattan",nb
proc = 8))))}
}
#NetworkDist <- distcells*distcells2
colnames(NetworkDist) <- row.names(cd_diffusionplot)
row.names(NetworkDist) <- row.names(cd_diffusionplot)
emat_expressed <- apply(networkExpressionFile,1,function(x) any ((x) >expr_limit))
emat_expressed <- networkExpressionFile[emat_expressed,row.names(NetworkDist)]
emat_expressed <- as.matrix(emat_expressed[intersect(row.names(emat_expressed),OldGen
eset),])
if(usemagic=="TRUE"){
emat_expressed <- magic(ts,emat_expressed[,row.names(cd_diffusionplot)], power = 1, k
= 20, n_eigs = 20, n_local = 10)
}
#library(doParallel)
#library(foreach)
library(parallel)
library(spdep)
cores <- corenumber
# cl <- makeCluster(cores)
# registerDoParallel(cl)
SCN3Egenefilter <- matrix(nrow=nrow(emat_expressed),ncol = 3)
cellnames <- colnames(emat_expressed)
# library(Matrix)
# NetworkDistsparse <- Matrix(NetworkDist, sparse = TRUE)
print("Making celllandscape...Generating Spatial Weights Matrix...")
i=1
NetworkDist2 <- matrix(nrow=nrow(NetworkDist),ncol=ncol(NetworkDist))
  for(i in 1:ncol(NetworkDist2))
    {
  x <-NetworkDist[,i]
 x[x < as.numeric(quantile(x,nnquant))] <- 0
 NetworkDist2[,i] <- x
```

```
  }
NetworkDist <- NetworkDist2
rm(NetworkDist2)
colnames(NetworkDist) <- row.names(cd_diffusionplot)
row.names(NetworkDist) <- row.names(cd_diffusionplot)
spatial.weights <- mat2listw(NetworkDist[])
print("Making celllandscape...Generating Spatial Weights Matrix...done")
numbsim <- nsim
i=1
test <- as.data.frame(mclapply(1:nrow(emat_expressed), function(i) {
   return(m <- c(unlist(moran.mc(emat_expressed[i,],spatial.weights,nsim=numbsim,zer
o.policy = TRUE))[1:3],row.names(emat_expressed)[i]))
}, mc.cores=corenumber))
SCN3Egenefilter <- t(test)
SCN3Egenefilter <- SCN3Egenefilter[! apply(SCN3Egenefilter,1,function(x) any(x=="NaN"
)),]
row.names(SCN3Egenefilter) <- SCN3Egenefilter[,4]
SCN3Egenefilter_clean <- as.data.frame(SCN3Egenefilter[complete.cases(SCN3Egenefilte
r),c(1:4)])
 r <- row.names(SCN3Egenefilter_clean)
SCN3Egenefilter_clean <- apply(SCN3Egenefilter_clean,2,function(x) as.numeric(x))
row.names(SCN3Egenefilter_clean) <- r
SCN3Egeneset <- row.names(subset(SCN3Egenefilter_clean,SCN3Egenefilter_clean[,3] <=
0.01))
HSEgenes <- row.names(subset(SCN3Egenefilter_clean,SCN3Egenefilter_clean[,3] <= 0.01
 & SCN3Egenefilter_clean[,1] >= as.numeric(quantile(SCN3Egenefilter_clean[SCN3Egenese
t,1],0.1))))
SCN3Egeneset <- row.names(subset(SCN3Egenefilter_clean,SCN3Egenefilter_clean[,3] <=
0.01 & SCN3Egenefilter_clean[,1] >= as.numeric(quantile(SCN3Egenefilter_clean[SCN3Ege
neset,1],0.1))))
GeneFilterProgression <- cbind(GeneFilterProgression[SCN3Egeneset,],SCN3Egenefilter_c
lean[SCN3Egeneset,1])
if(iter<=threshold){meanMoran<-mean(abs(SCN3Egenefilter_clean[,1]),na.rm=TRUE)}
GeneMarkovList <- c(list(GeneMarkov=SCN3Egeneset),GeneMarkovList)
} #uncomment this to enable iterative filtering
bootstrap <- bootstrap+1
if(sample==1){
moransIsampled <- c(moransIsampled,SCN3Egeneset)
}
}
```

```
[1] "Preparing genefiltering on 3000 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 3000 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 1835 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 1835 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 1553 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 1553 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 1375 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 1375 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 1229 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 1229 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 1101 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 1101 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 991 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 991 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 880 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 880 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 792 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 792 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 709 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 709 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 638 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 638 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 574 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 574 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 515 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 515 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 463 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 463 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 416 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 416 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 374 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 374 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 336 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 336 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 302 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 302 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 271 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 271 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 244 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 244 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 219 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 219 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 197 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 197 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 177 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 177 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 159 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 159 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 143 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 143 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 128 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 128 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 115 genes."
[1] "Making celllandscape...Performing dimensionality reduction"
[1] "Making celllandscape...Making diffusionmap"
[1] "Filtering 115 genes."
[1] "Making celllandscape...Generating Spatial Weights Matrix..."
[1] "Making celllandscape...Generating Spatial Weights Matrix...done"
```

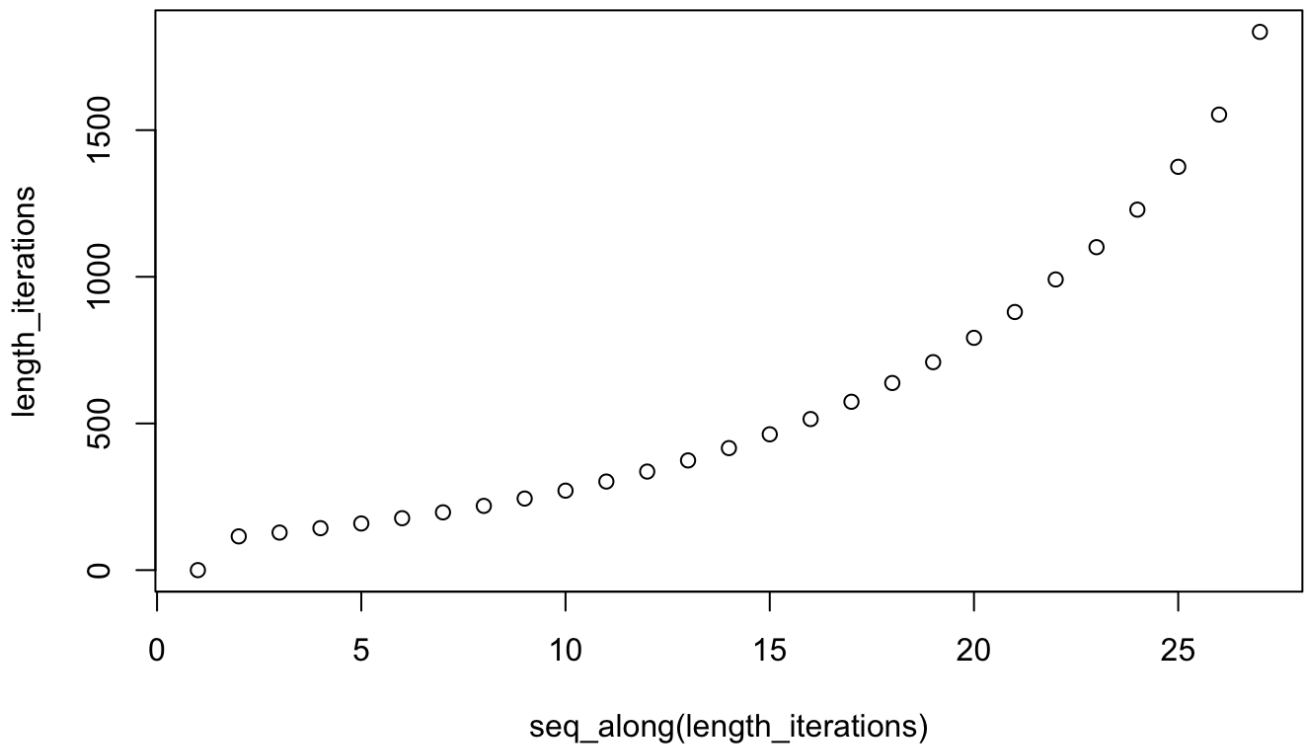
```
NAs introduced by coercion
```

```
[1] "Preparing genefiltering on 0 genes."
[1] "Making celllandscape...Making diffusionmap"
```

```
Error in d[, 1] : subscript out of bounds
```

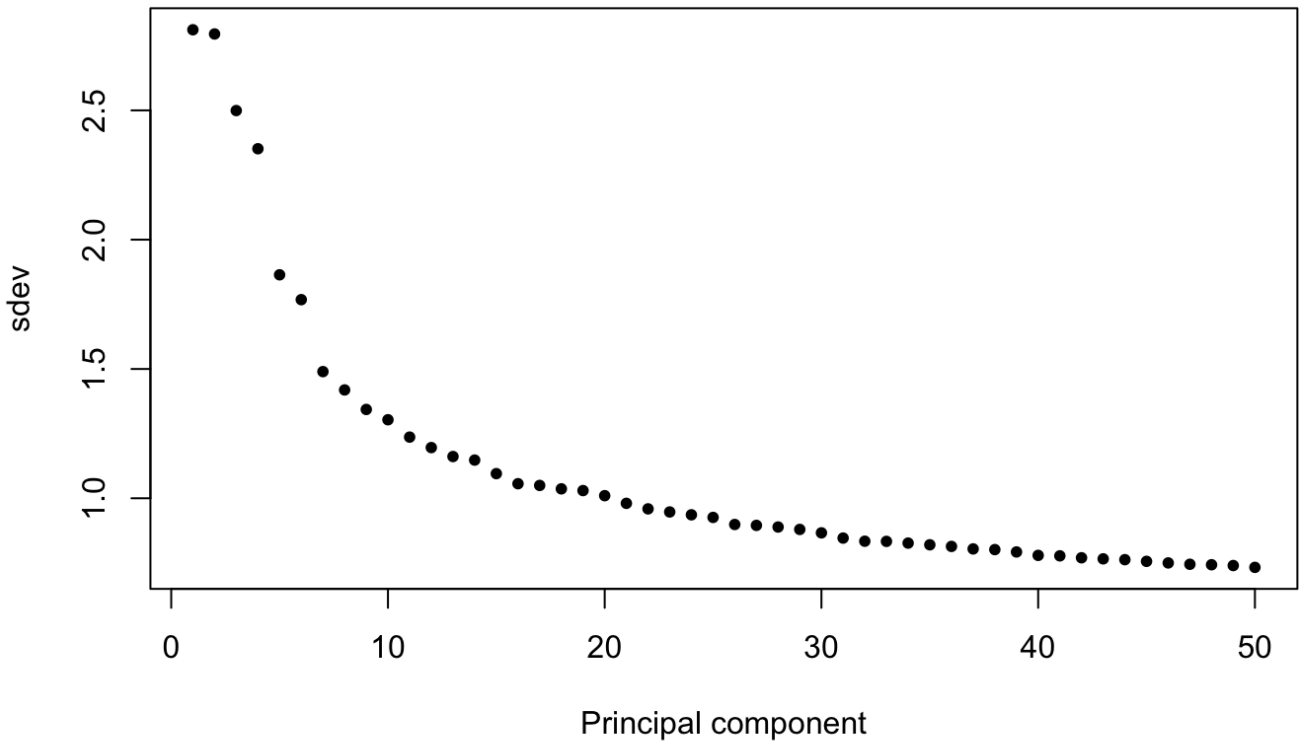
Hide

```
length_iterations <-  unlist(lapply(GeneMarkovList,function(x) length(x)))
plot(seq_along(length_iterations),length_iterations)
```



Hide

```
pcatsne <- prcomp(t(as.matrix(networkExpressionFile[unlist(GeneMarkovList[26]),])),sc
ale. = TRUE)
pca <- pcatsne
plot(log(pcatsne$sdev[1:50]),pch = 20,
xlab = 'Principal component', ylab = 'sdev')
```

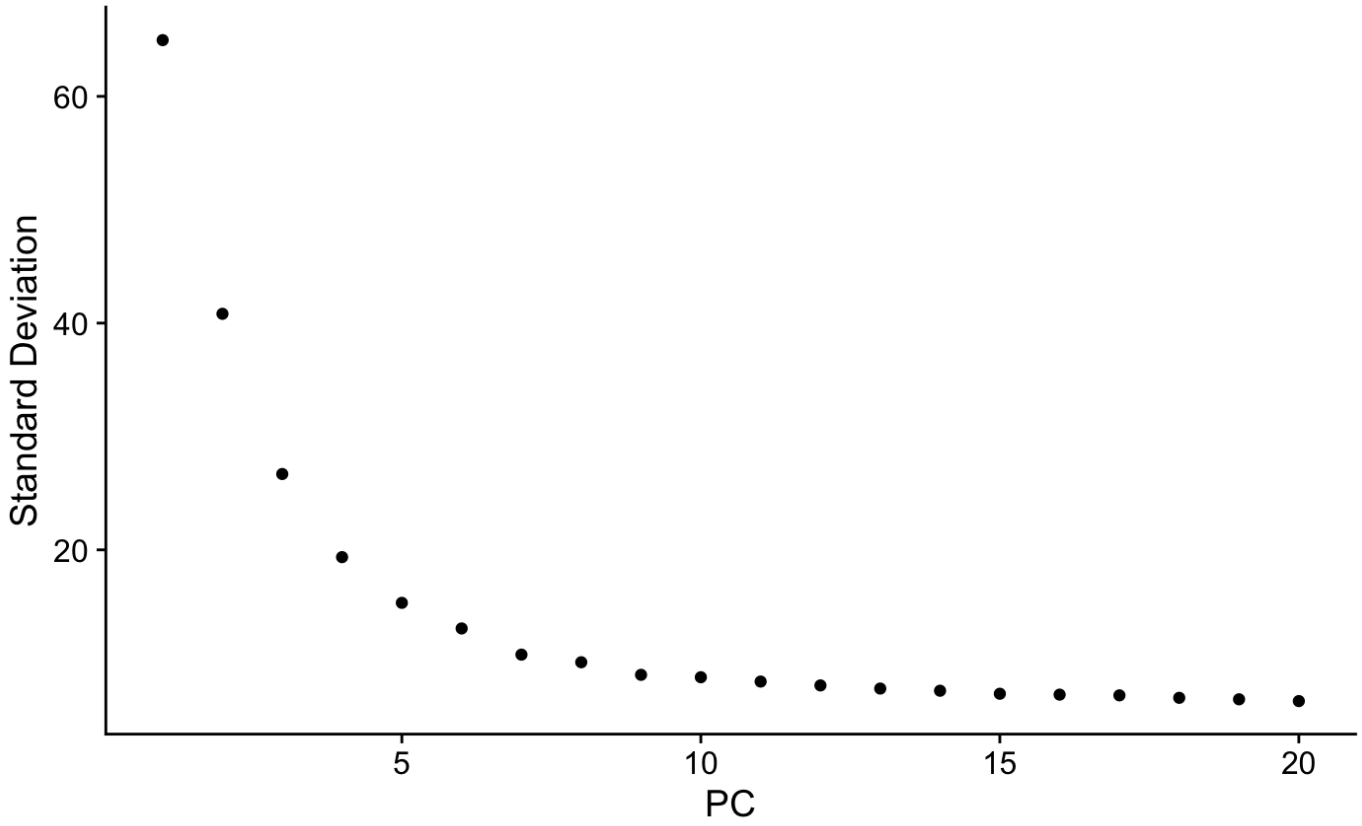


Hide

```
ncompL1tsne <- which(cumsum(pca$sdev[1:50])/sum(pca$sdev[1:50]) > 0.6)[1]
tsne <- t(pca$x[,c(1:ncompL1tsne)])
number_of_clusters <- 8
NetworkDist <- Dist(t(tsne),method = "manhattan",nbproc = 8)
GeneMarkov_hc <- hclust(NetworkDist, method = "ward.D2")
GMcluster <- rbind(groups = cutree(GeneMarkov_hc, k=number_of_clusters))
names(GMcluster) <- colnames(networkExpressionFile)
oligos.integratedIm$seurat_clusters <- as.factor(GMcluster)
GeneMarkov  <- unlist(GeneMarkovList[27])
```

Hide

```
DefaultAssay(oligos.integratedIm) <- "SCT"
oligos.integratedIm <- RunPCA(oligos.integratedIm, verbose = FALSE,features = GeneMar
kov)#npcs = 20)
ElbowPlot(oligos.integratedIm)
```



Hide

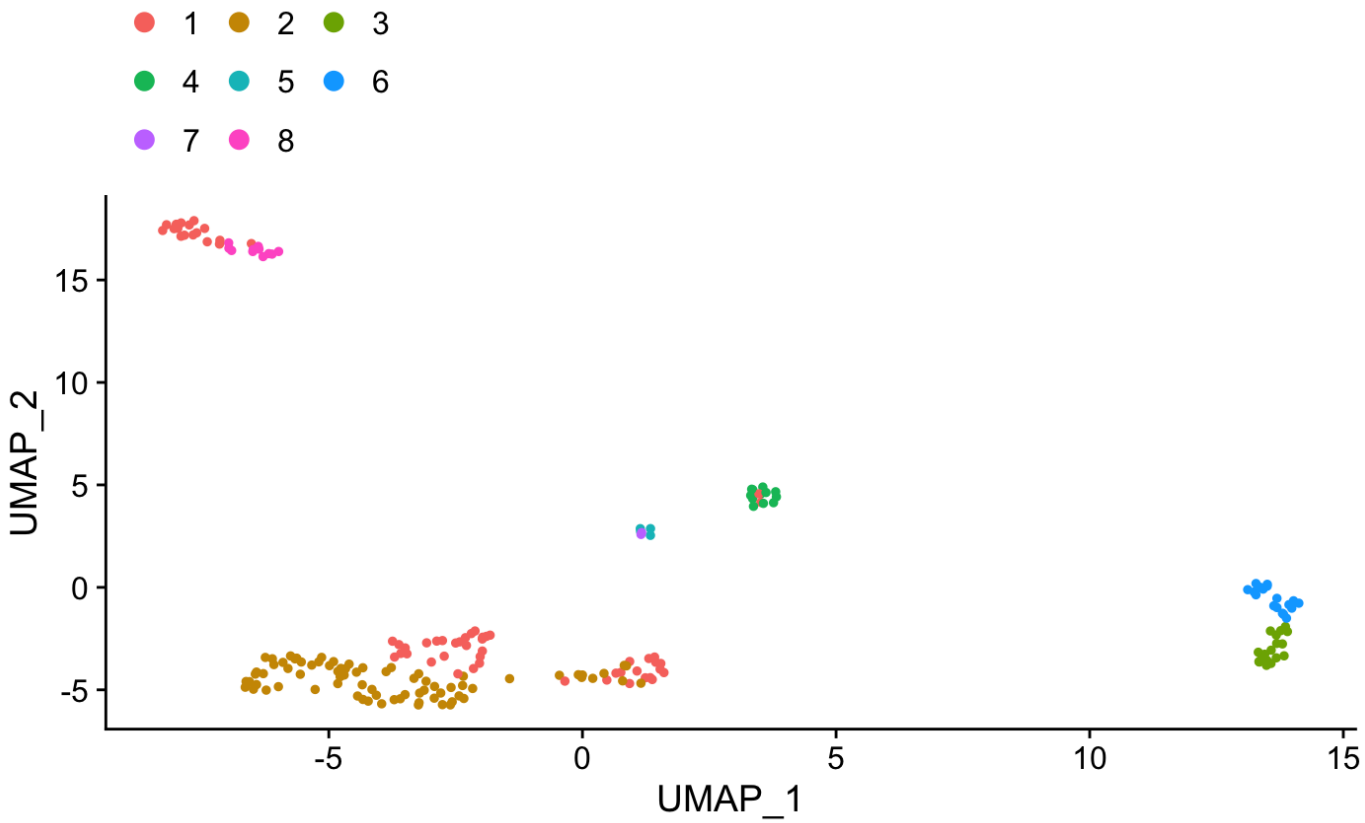
```
oligos.integratedIm <- RunUMAP(oligos.integratedIm, dims = 1:10)
```

```
16:50:59 UMAP embedding parameters a = 0.9922 b = 1.112
16:50:59 Read 203 rows and found 10 numeric columns
16:50:59 Using Annoy for neighbor search, n_neighbors = 30
16:50:59 Building Annoy index with metric = cosine, n_trees = 50
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
16:51:00 Writing NN index file to temp file /var/folders/98/j14xl9ln5190l9fvq32g5rkw0
000gn/T//RtmpSdWnAZ/file3f2e7275c4d0
16:51:00 Searching Annoy index using 1 thread, search_k = 3000
16:51:00 Annoy recall = 100%
16:51:00 Commencing smooth kNN distance calibration using 1 thread
16:51:01 Initializing from normalized Laplacian + noise
16:51:01 Commencing optimization for 500 epochs, with 6014 positive edges
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
16:51:01 Optimization finished
```

Hide

```
#oligos.integratedScience <- RunTSNE(oligos.integratedScience, dims = 1:10)
plots <- DimPlot(oligos.integratedIm, group.by = c("seurat_clusters"), combine = FALS
E)
plots <- lapply(X = plots, FUN = function(x) x + theme(legend.position = "top") + gui
des(color = guide_legend(nrow = 3,
    byrow = TRUE, override.aes = list(size = 3))))
CombinePlots(plots)
```

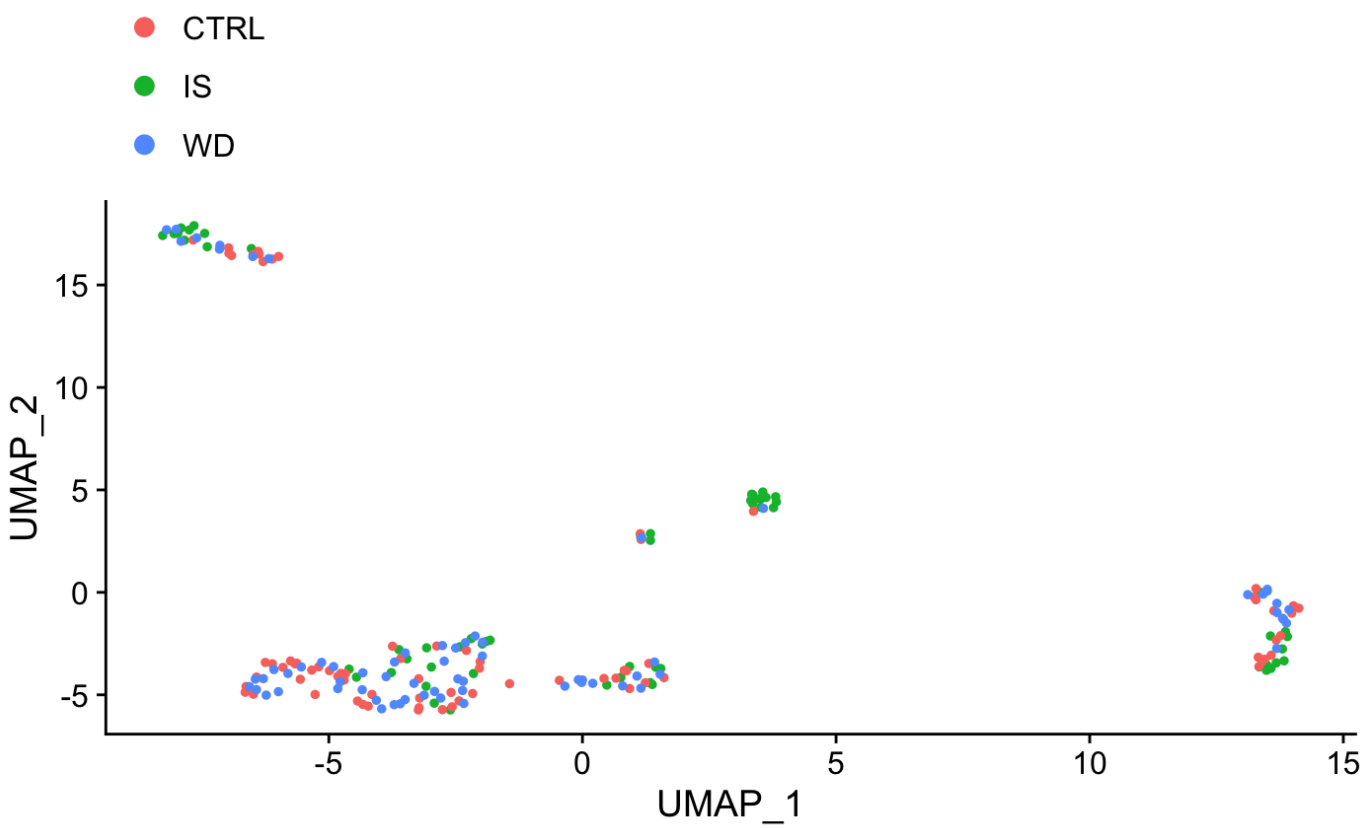
```
CombinePlots is being deprecated. Plots should now be combined using the patchwork sy
stem.
```



Hide

```
plots <- DimPlot(oligos.integratedIm, group.by = c("Sample"), combine = FALSE)
plots <- lapply(X = plots, FUN = function(x) x + theme(legend.position = "top") + gui
des(color = guide_legend(nrow = 3,
    byrow = TRUE, override.aes = list(size = 3))))
CombinePlots(plots)
```

```
CombinePlots is being deprecated. Plots should now be combined using the patchwork sy
stem.
```



Hide

```
oligos.integrated.markersIm %>% group_by(cluster) %>% top_n(n = 2, wt = avg_logFC)
```

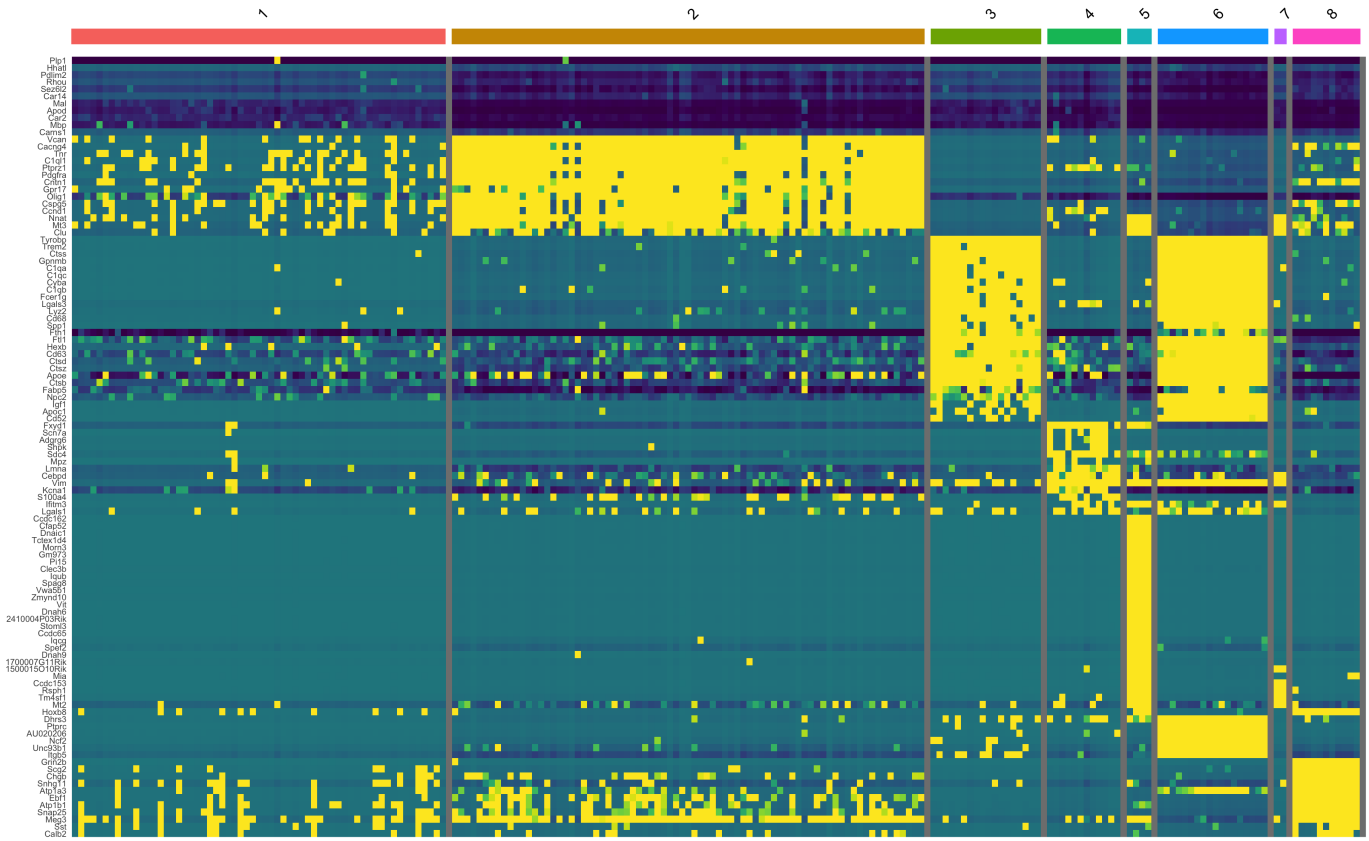
p_val <dbl>	avg_logFC <dbl>	pct.1 <dbl>	pct.2 <dbl>	p_val_adj <dbl>	cluster <fctr>	gene <chr>
1.949640e-24	1.566897	1.000	0.549	3.075752e-20	1	Mag
6.110968e-23	1.431195	1.000	0.831	9.640663e-19	1	Mbp
1.143881e-32	2.578188	0.870	0.056	1.804587e-28	2	Pdgfra
2.536822e-26	2.618041	0.896	0.278	4.002091e-22	2	Cspg5
1.367789e-11	2.175040	1.000	1.000	2.157824e-07	3	Fth1
8.316481e-10	1.736269	1.000	1.000	1.312008e-05	3	Ftl1
3.164079e-25	2.716292	0.917	0.042	4.991651e-21	4	Fxyd1
1.578773e-10	3.121956	0.333	0.010	2.490672e-06	4	Mpz
4.715871e-31	3.426160	1.000	0.010	7.439759e-27	5	1500015O10Rik
2.698587e-04	3.353688	1.000	0.497	1.000000e+00	5	Nnat

1-10 of 14 rows Previous **1** 2 Next

Hide

```
library(viridis)
DefaultAssay(oligos.integratedIm) <- "SCT"
top10 <- oligos.integrated.markersIm %>% group_by(cluster) %>% top_n(n = 10, wt = avg
_logFC)
# DoHeatmap(oligos.integratedIm, features = intersect(oligos.integrated.markersIm$gen
e,GeneMarkov),disp.max=7,) + NoLegend() +scale_fill_viridis()
DoHeatmap(oligos.integratedIm, features =intersect(oligos.integrated.markersIm$gene,u
nique(c(unlist(GeneMarkovList[5]),top10$gene))),disp.max=3) + NoLegend() +scale_fill_
viridis()
```

```
The following features were omitted as they were not found in the scale.data slot for
the SCT assay: Cldn11, Mag, Mog, Mobp, ErmnScale for 'fill' is already present. Addin
g another scale for 'fill', which will replace the existing
scale.
```



Hide

```
nonOL <- colnames(oligos.integratedIm)[oligos.integratedIm@meta.data$seurat_clusters
%in% c(3:6,8)]
oligos.integrated <- oligos.integrated[,! colnames(oligos.integrated) %in% nonOL]
data <- as.data.frame(table(oligos.integrated$Sample,droplevels(as.factor(oligos.inte
grated$predicted.id))))
colnames(data) <- c("Condition","Cluster","Freq")
library(plyr)
data$Cluster  <- factor(data$Cluster,levels=c("OPC","COP","NFOL1","MFOL1","MFOL2","MO
L1","MOL2","MOL3","MOL4","MOL5","MOL6","PPR"))
data <- data[which(data$Cluster %in% c("MFOL1","MFOL2","MOL1","MOL2","MOL3","MOL4","M
OL5","MOL6")),]
#data$Cluster  <- factor(data$Cluster,levels=c("MFOL1","MFOL2","MOL1","MOL2","MOL
3","MOL4","MOL5","MOL6"))
library(reshape2)
datacasted <- dcast(data,Cluster ~ Condition)
```

```
Using Freq as value column: use value.var to override.
```

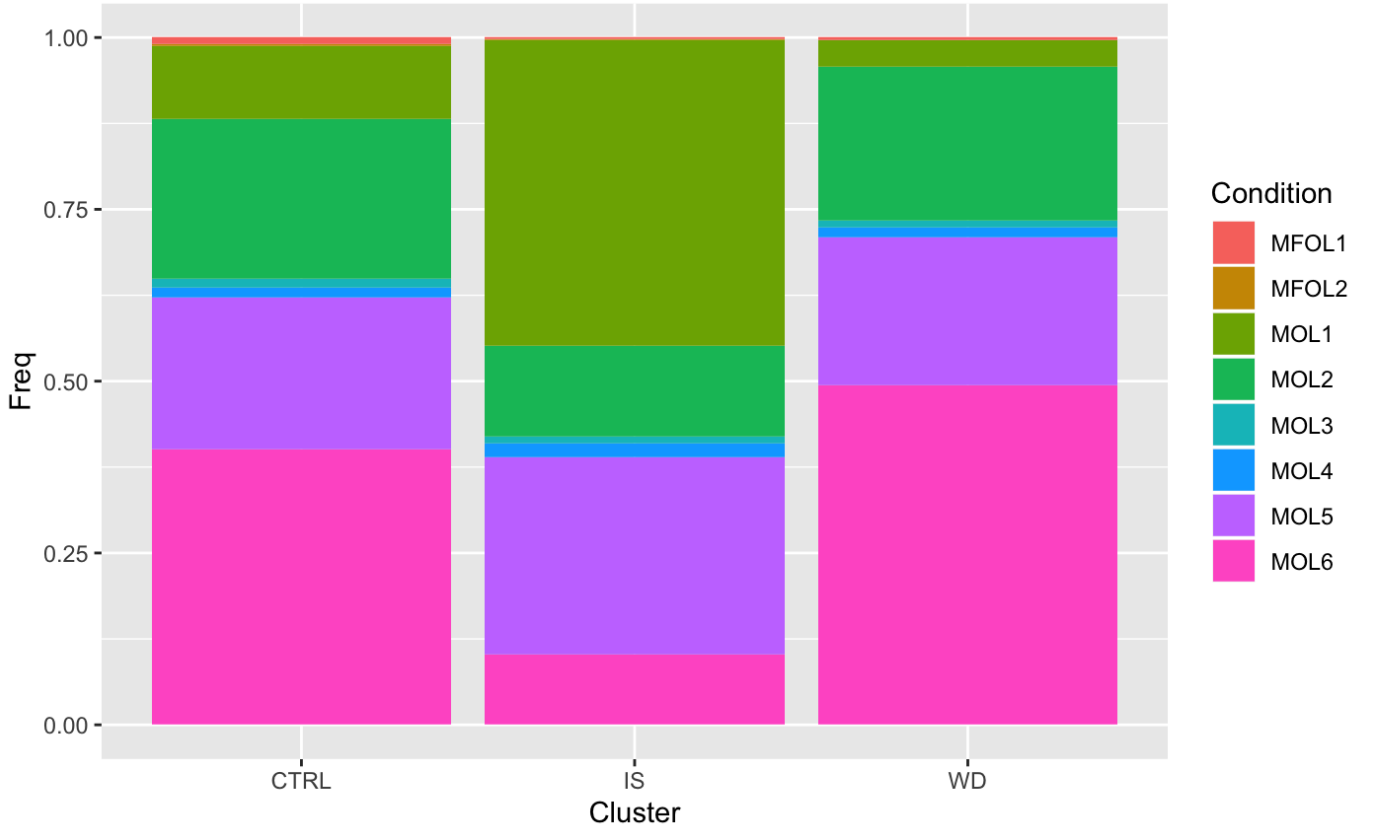
Hide

```
calc_cpm <-function (expr_mat)
{
    norm_factor <- colSums(expr_mat)
    return(t(t(expr_mat)/norm_factor))
}
datacasted[,2:4] <- calc_cpm(datacasted[,2:4])
data <- melt(datacasted)
```

```
Using Cluster as id variables
```

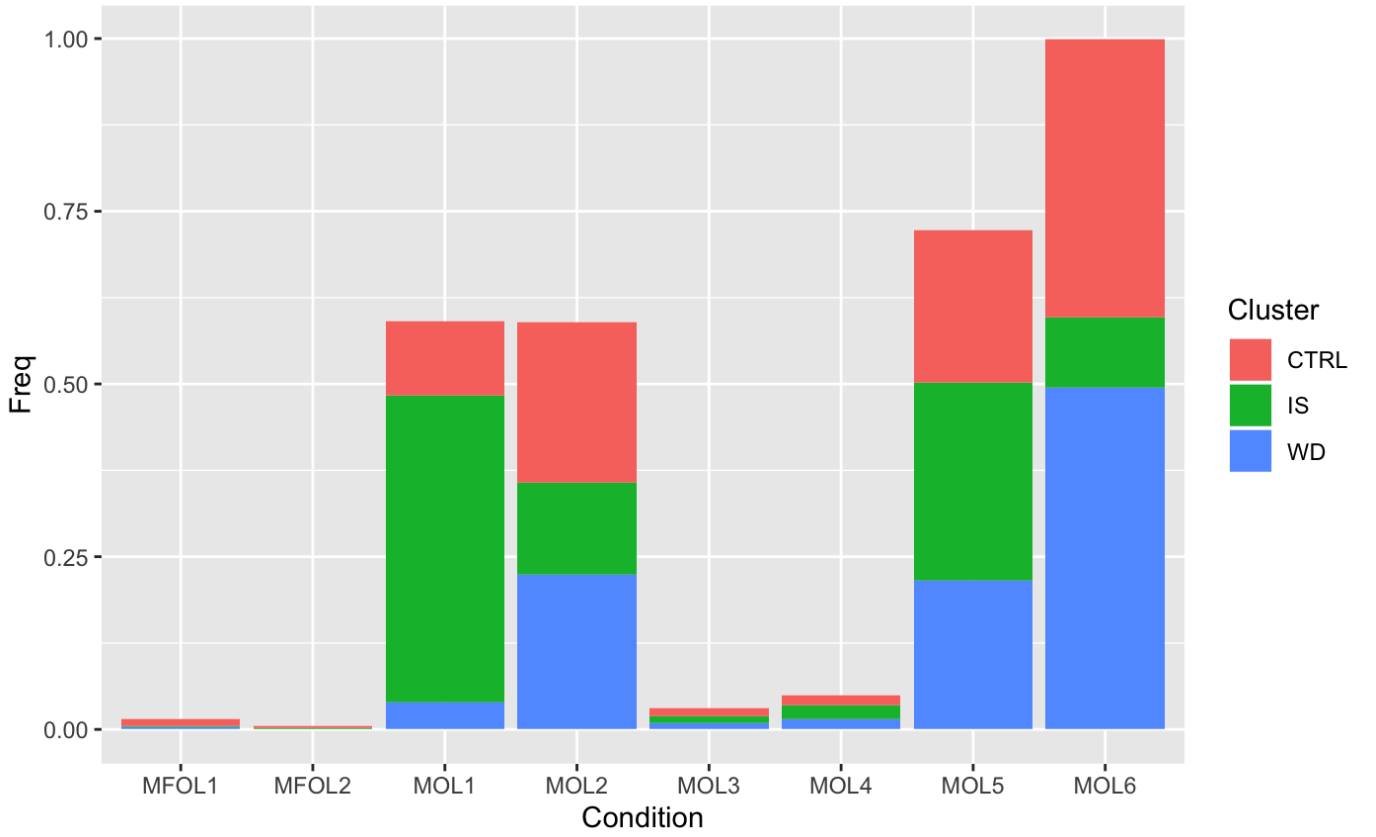
Hide

```
colnames(data) <- c("Condition","Cluster","Freq")
#data$Cluster  <- revalue(as.factor(data$Cluster),c("PPR"="VLMC"))
# Stacked + percent
ggplot(data, aes(fill=Condition, y=Freq, x=Cluster)) +
    geom_bar(position="fill", stat="identity")
```



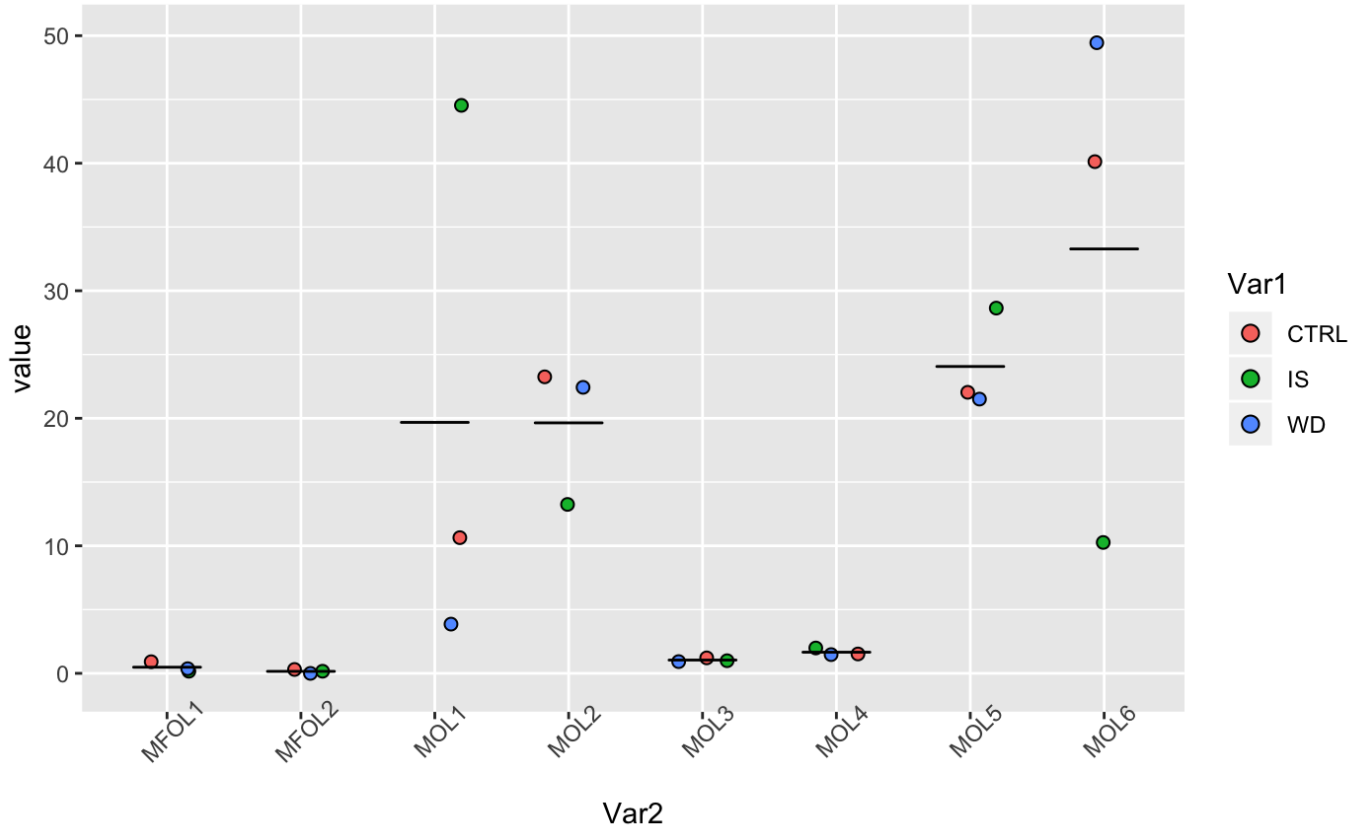
Hide

```
ggplot(data, aes(fill=Cluster, y=Freq, x=Condition)) +
    geom_bar( stat="identity")
```



Hide

```
row.names(datacasted) <- datacasted[,1]
datacasted <- datacasted[,2:4]*100
datamelted <- melt(t(datacasted))
ggplot(datamelted, aes(y = value, x = Var2)) + # Move y and x here so than they can b
e used in stat_*
    geom_dotplot(aes(fill = Var1),   # Use fill = Species here not in ggplot()
                 binaxis = "y",         # which axis to bin along
                 binwidth = 1,        # Minimal difference considered diffeerent
                 stackdir = "center",
                 position = position_jitter(0.2)# Centered
                 ) +  # scale_y_log10() +
    stat_summary(fun.y = mean, fun.ymin = mean, fun.ymax = mean,
                 geom = "crossbar", width = 0.5,fatten = 0.01) + theme(axis.text.x =
 element_text(angle = 45))
```



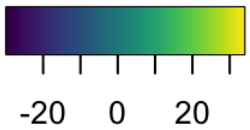
Hide

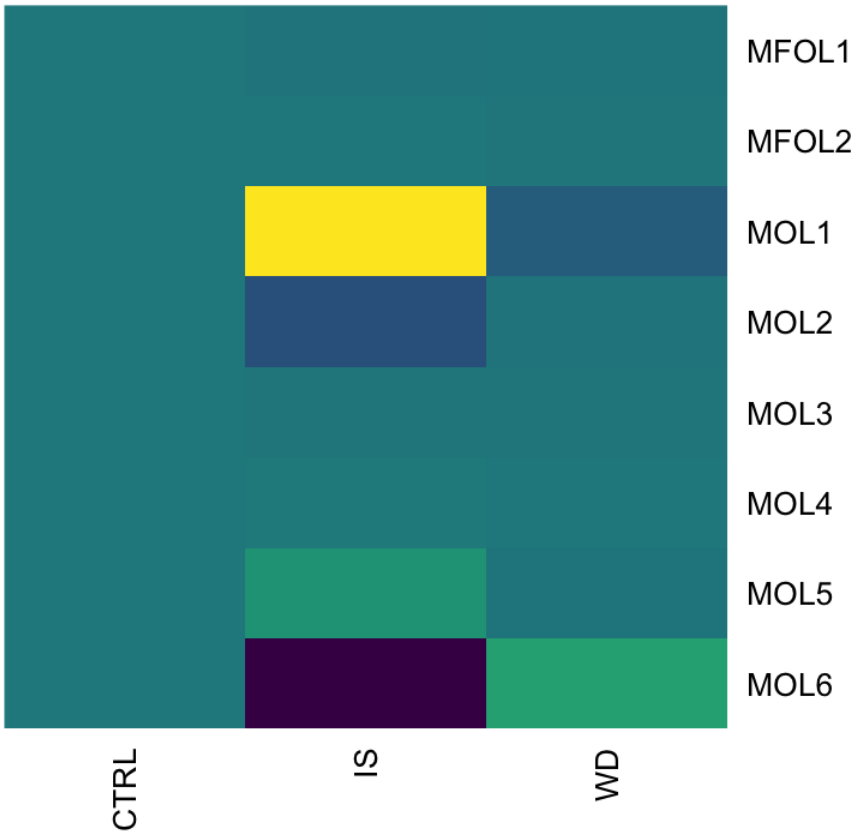
```
library(heatmap3)
library(viridis)
data <- as.data.frame(table(oligos.integrated$Sample,droplevels(as.factor(oligos.inte
grated$predicted.id))))
colnames(data) <- c("Condition","Cluster","Freq")
library(plyr)
data$Cluster  <- factor(data$Cluster,levels=c("OPC","COP","NFOL1","MFOL1","MFOL2","MO
L1","MOL2","MOL3","MOL4","MOL5","MOL6","PPR"))
data <- data[which(data$Cluster %in% c("MFOL1","MFOL2","MOL1","MOL2","MOL3","MOL4","M
OL5","MOL6")),]
#data$Cluster  <- factor(data$Cluster,levels=c("MFOL1","MFOL2","MOL1","MOL2","MOL
3","MOL4","MOL5","MOL6"))
library(reshape2)
datacasted <- dcast(data,Cluster ~ Condition)
```

```
Using Freq as value column: use value.var to override.
```

Hide

```
calc_cpm <-function (expr_mat)
{
    norm_factor <- colSums(expr_mat)
    return(t(t(expr_mat)/norm_factor))
}
datacasted[,2:4] <- calc_cpm(datacasted[,2:4])*100
row.names(datacasted) <- datacasted[,1]
datacasted <- datacasted[,2:4]
comparison <-datacasted-apply(datacasted,1,function(x) mean(x))
comparison <-datacasted-datacasted[,1]
heatmap3(comparison[rev(row.names(comparison)),], Rowv = NA , Colv = NA ,scale = "non
e",symm = F, method = "ward.D2",col=viridis(1000),balanceColor =F,cexRow = 1,cexCol =
1,margins = c(10, 10))
```

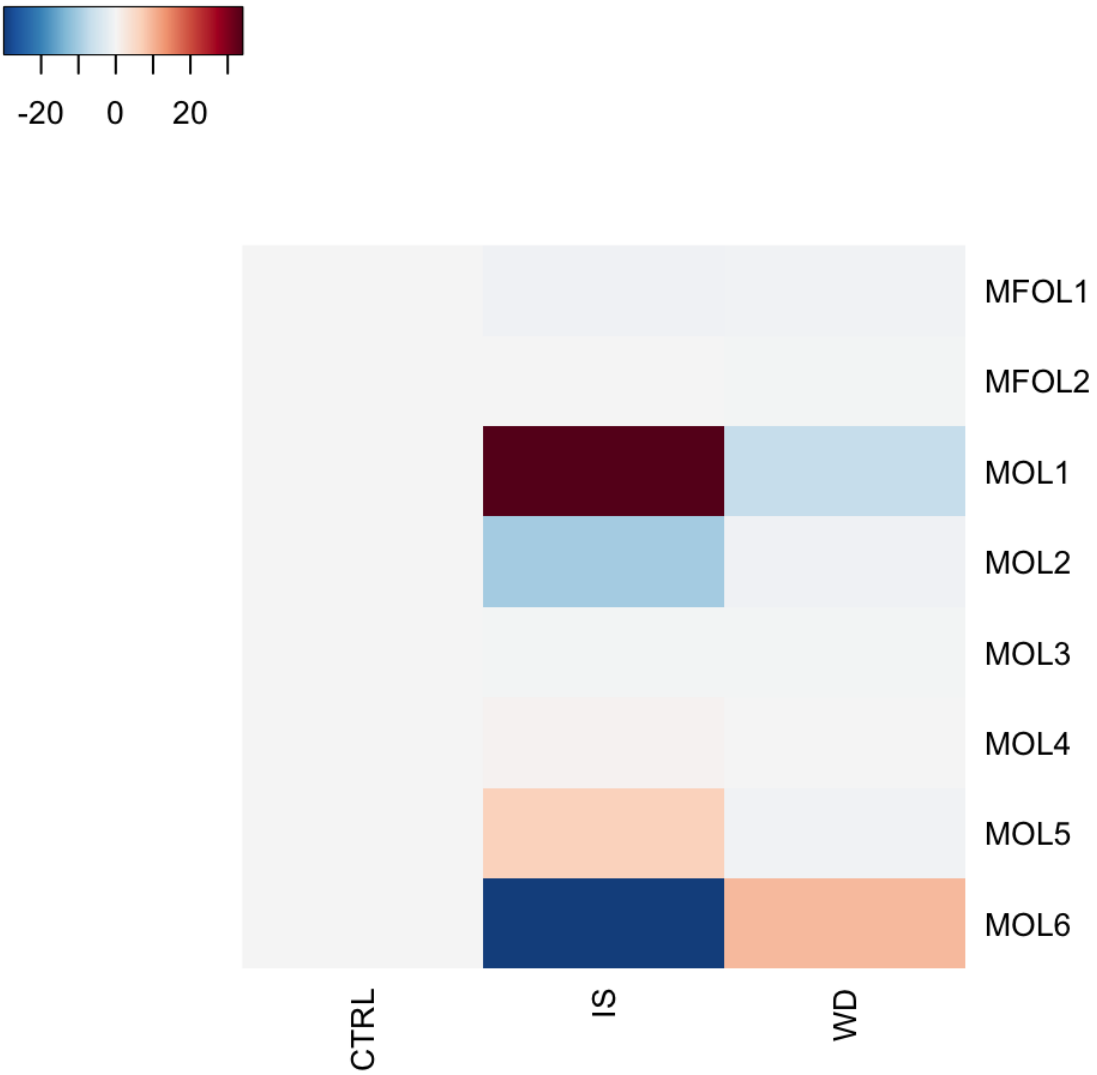




Hide

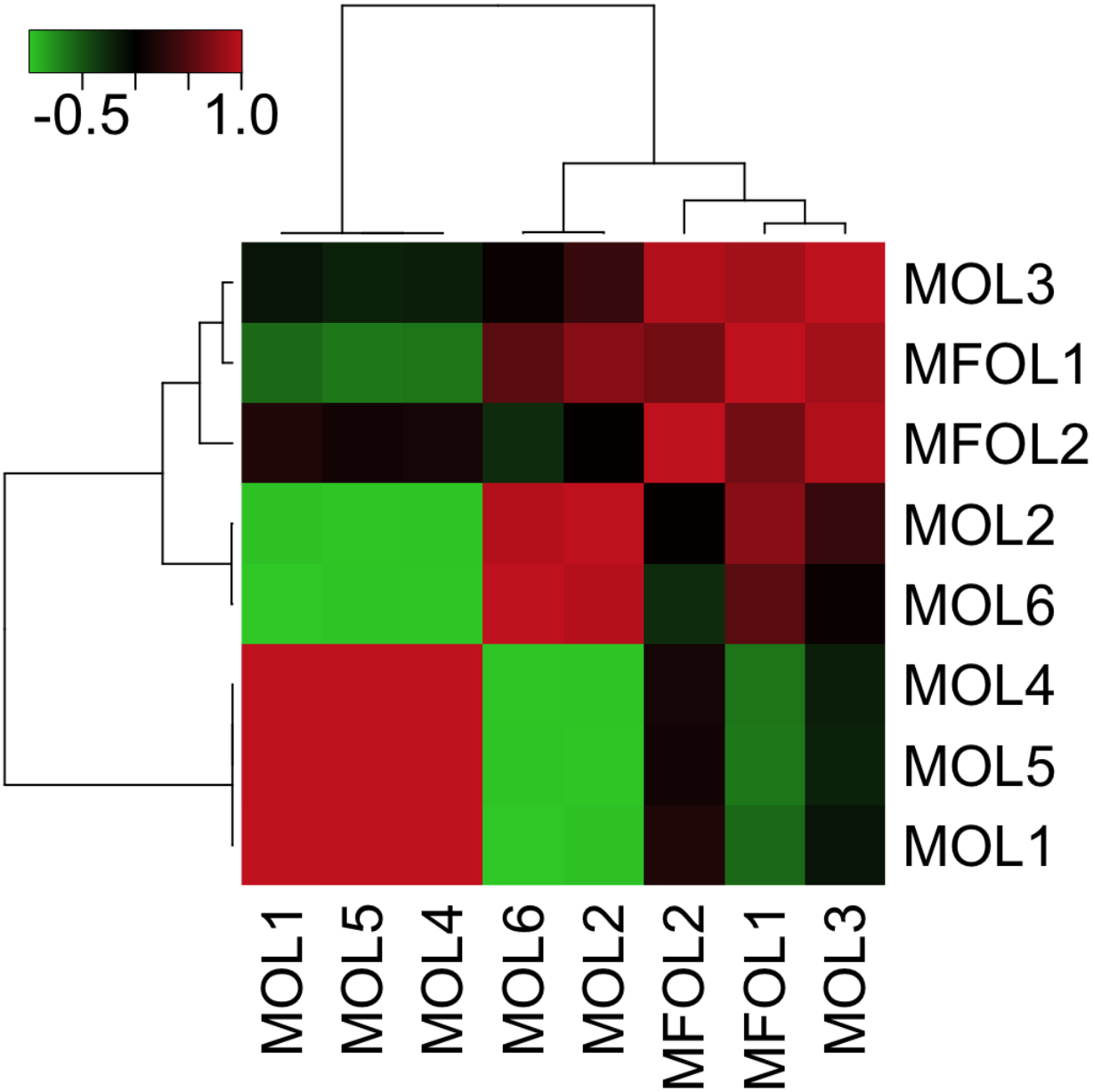
```
library(RColorBrewer)
heatmap3(comparison[rev(row.names(comparison)),], Rowv = NA , Colv = NA ,scale = "non
e",symm = F, method = "ward.D2",col=rev(colorRampPalette(brewer.pal(1024,"RdBu"))(102
4)),balanceColor =T,cexRow = 1,cexCol = 1,margins = c(10, 10))
```

```
n too large, allowed maximum for palette RdBu is 11
Returning the palette you asked for with that many colors
```



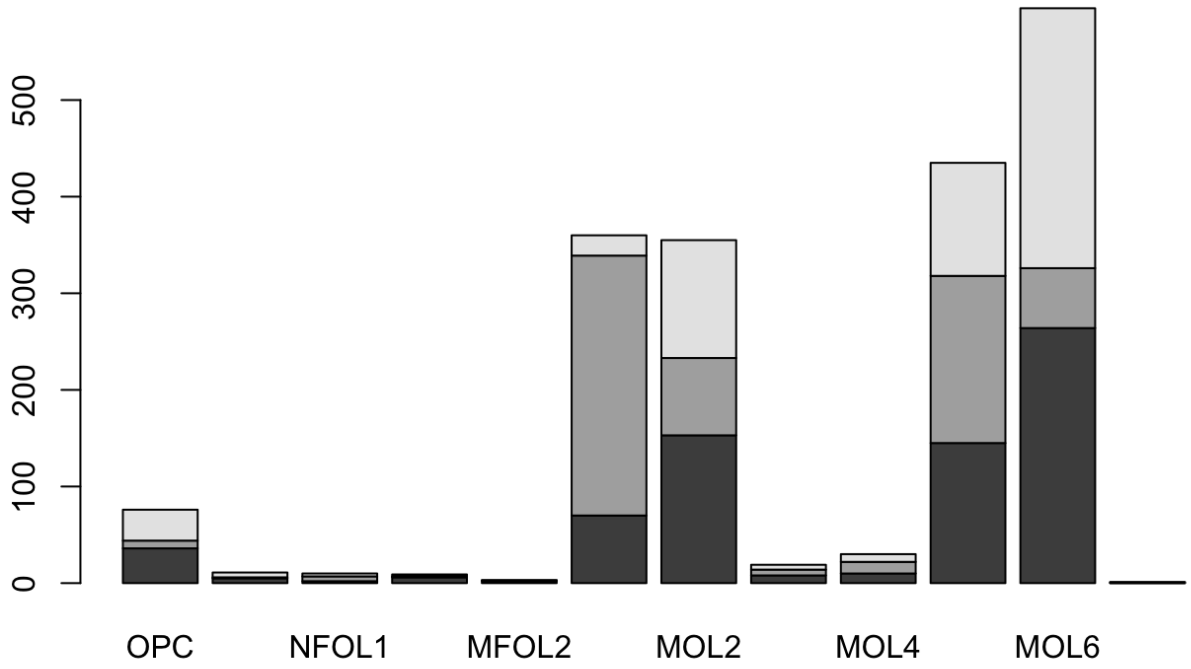
Hide

```
  relationshipratio <- cor(t(comparison),method="pearson")
heatmap3(relationshipratio[rev(row.names(comparison)),], Rowv = NULL , Colv = NULL ,s
cale = "none",symm = F, method = "ward.D2",col=colorRampPalette(c("limegreen","black"
,
"firebrick3"))(1024),balanceColor =F,cexRow = 2,cexCol = 2,margins = c(10, 10))
```



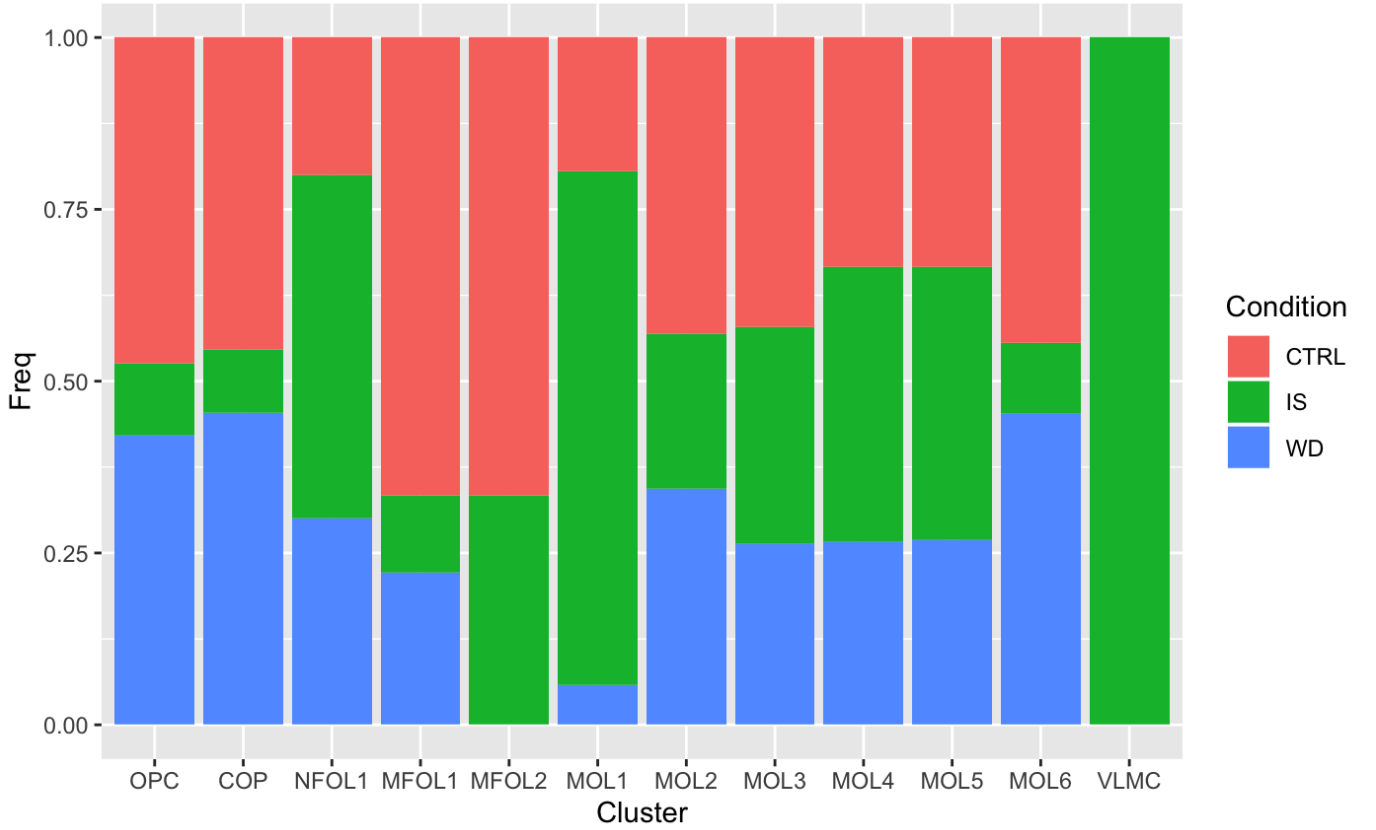
Hide

```
barplot(table(oligos.integrated$Sample,oligos.integrated$predicted.id))
```



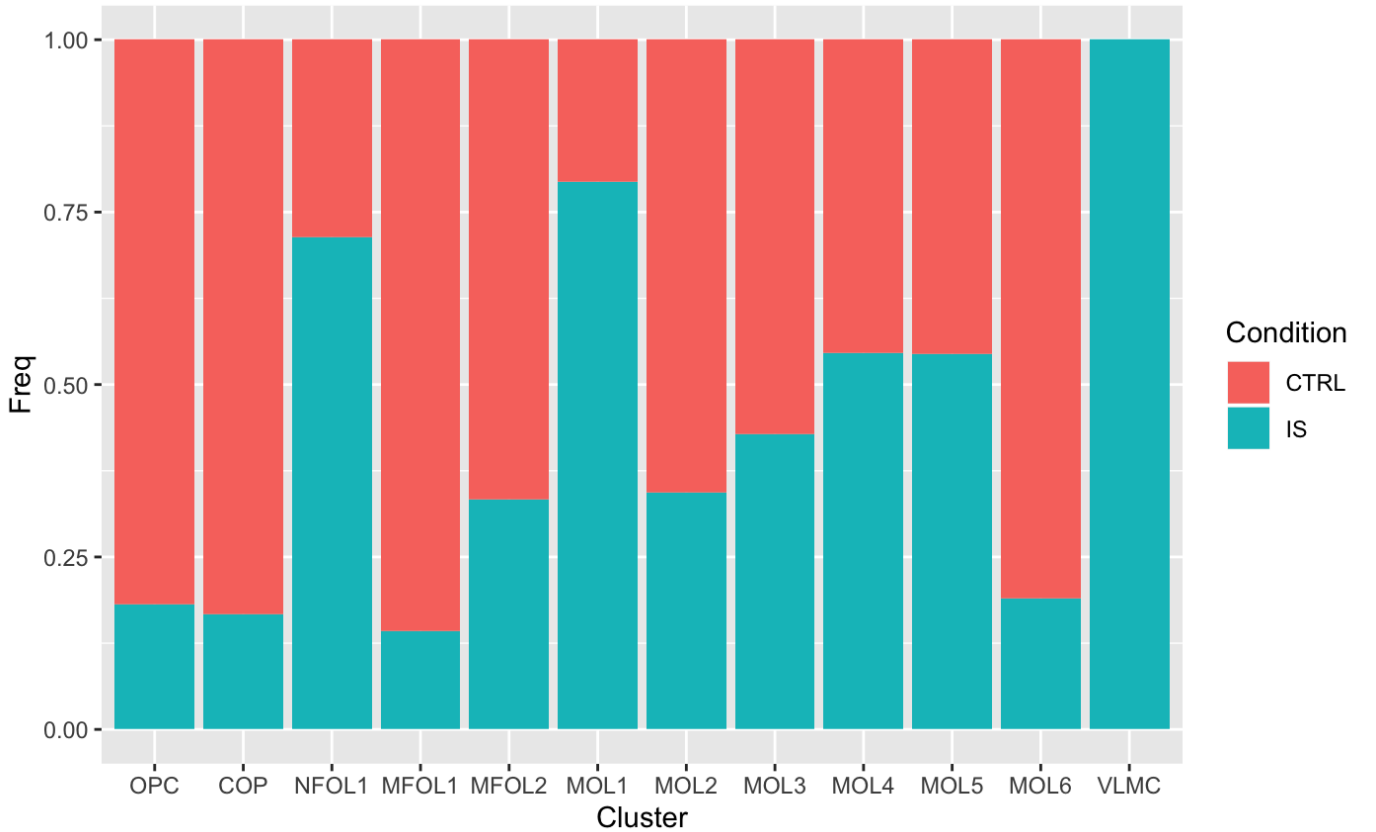
Hide

```
data <- as.data.frame(table(oligos.integrated$Sample,oligos.integrated$predicted.id))
colnames(data) <- c("Condition","Cluster","Freq")
library(plyr)
data$Cluster  <- factor(data$Cluster,levels=c("OPC","COP","NFOL1","MFOL1","MFOL2","MO
L1","MOL2","MOL3","MOL4","MOL5","MOL6","PPR"))
data$Cluster  <- revalue(as.factor(data$Cluster),c("PPR"="VLMC"))
dataIS_CTRL <- data[which(data$Condition %in% c("IS","CTRL")),]
dataWD_CTRL <- data[which(data$Condition %in% c("WD","CTRL")),]
dataIS_WD <- data[which(data$Condition %in% c("IS","WD")),]
# Stacked + percent
ggplot(data, aes(fill=Condition, y=Freq, x=Cluster)) +
    geom_bar(position="fill", stat="identity")
```



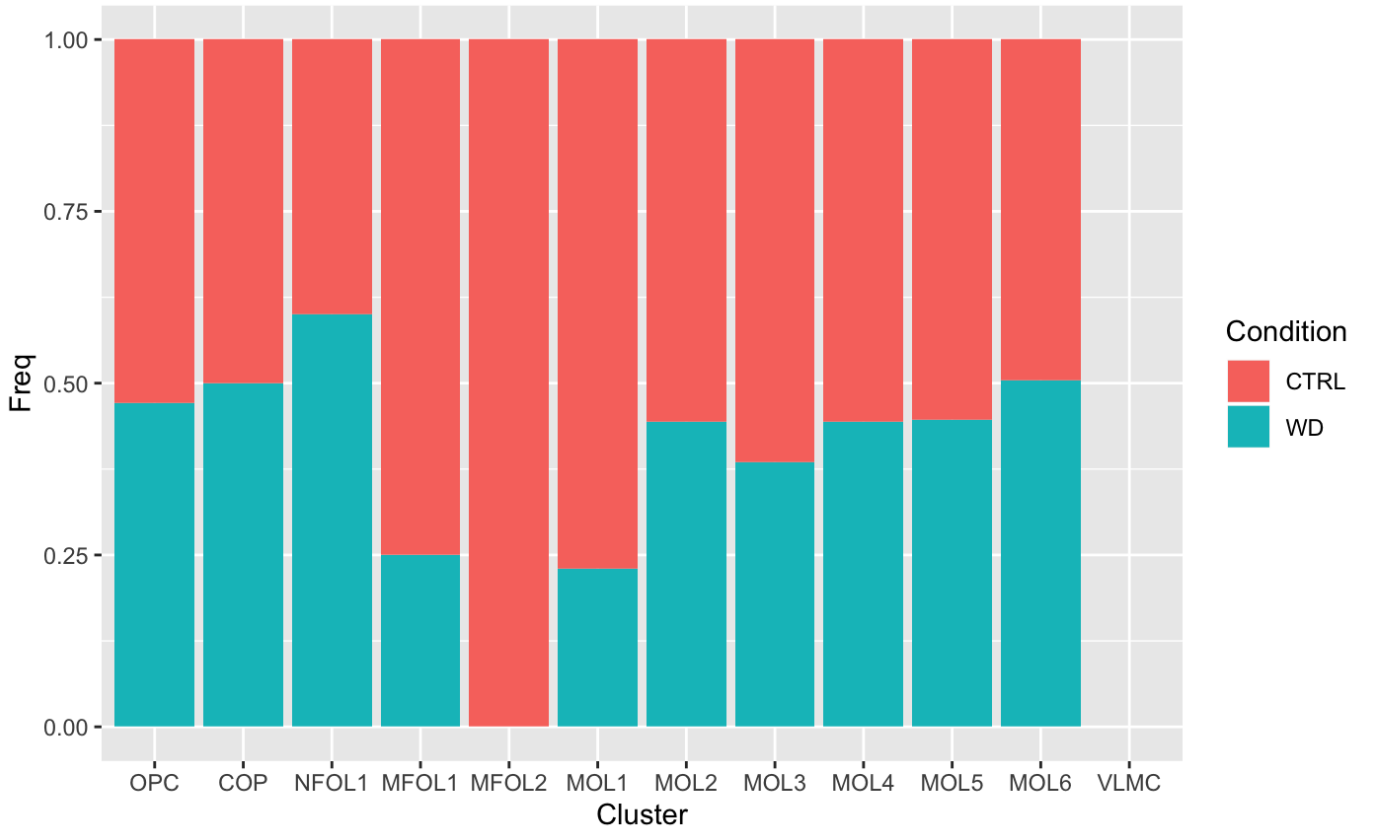
Hide

```
ggplot(dataIS_CTRL, aes(fill=Condition, y=Freq, x=Cluster)) +
    geom_bar(position="fill", stat="identity")
```



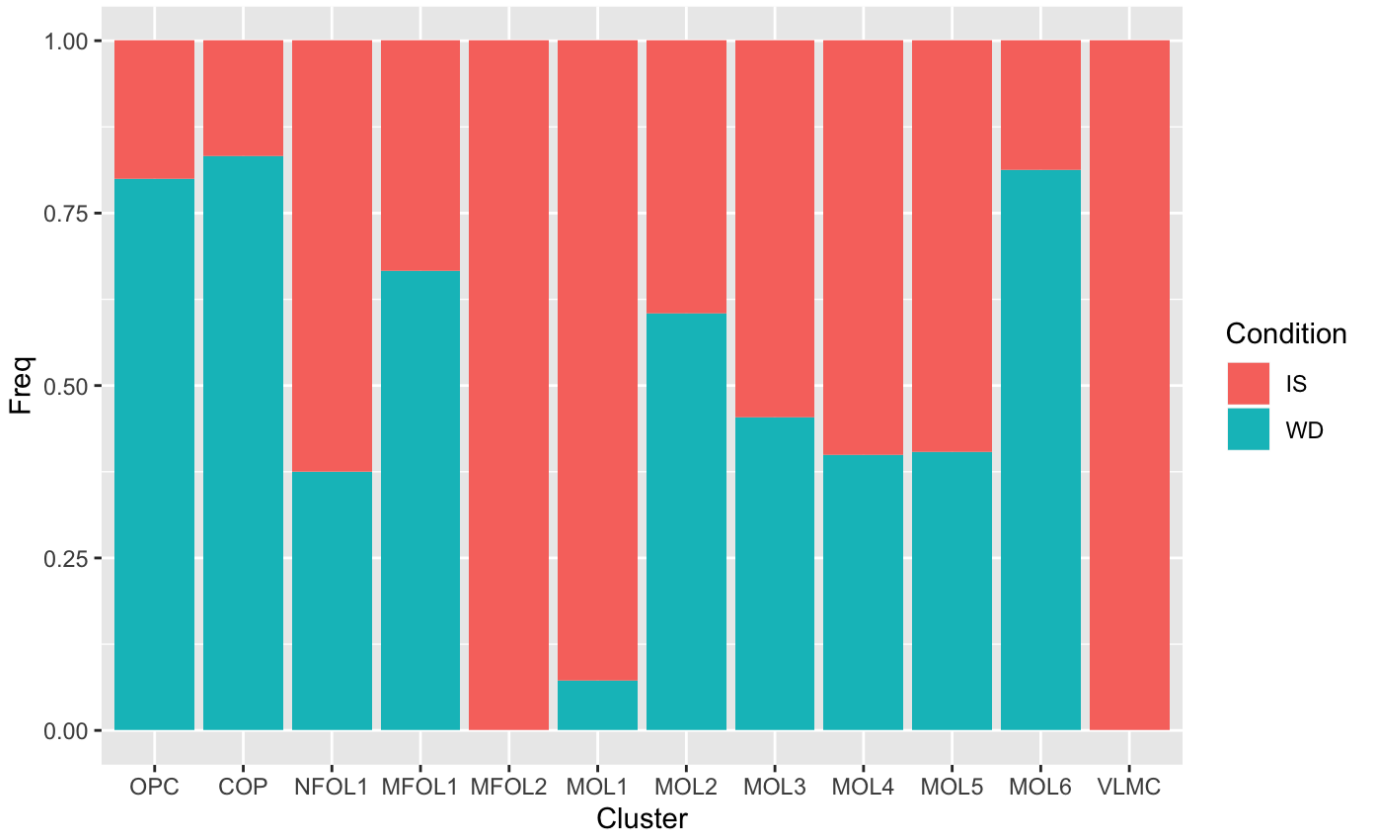
Hide

```
ggplot(dataWD_CTRL, aes(fill=Condition, y=Freq, x=Cluster)) +
    geom_bar(position="fill", stat="identity")
```



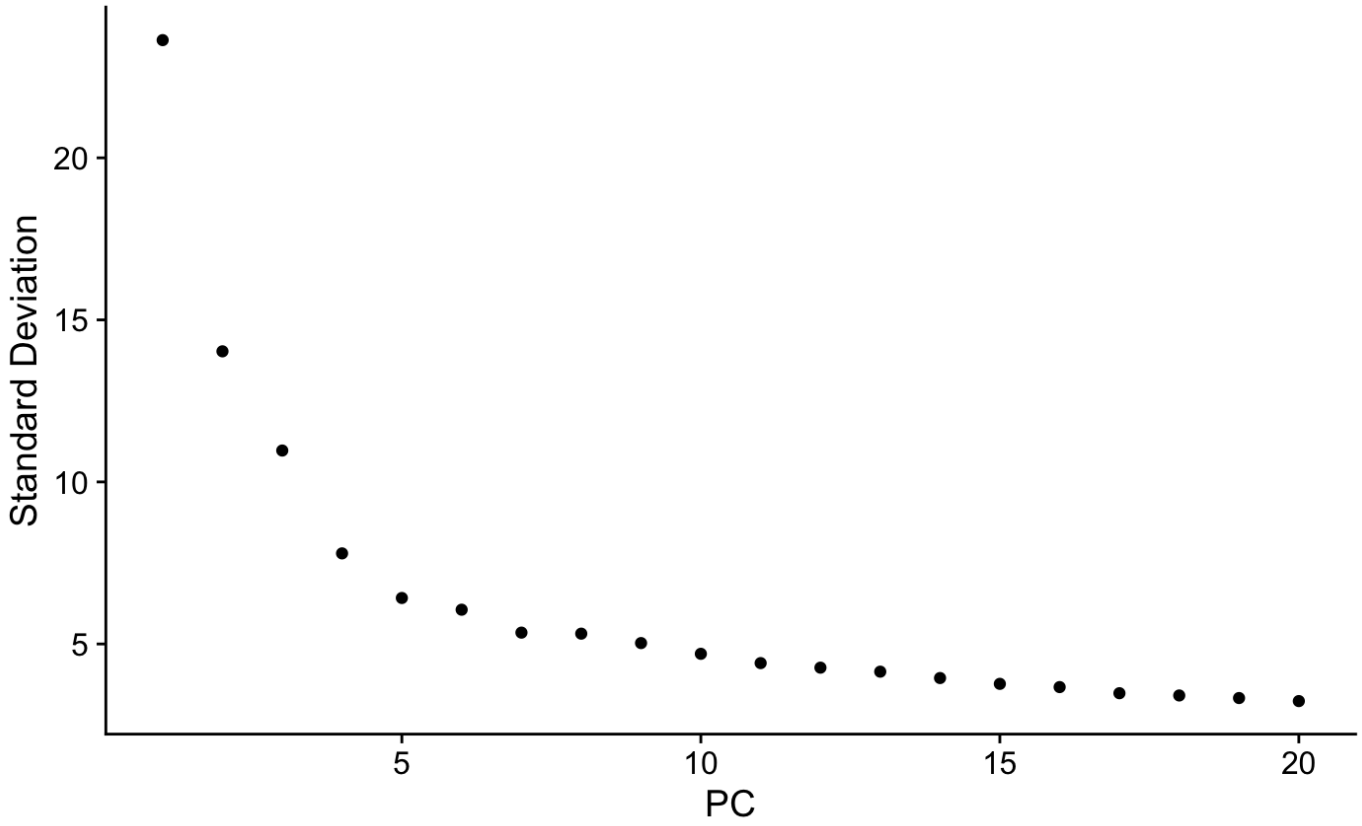
Hide

```
ggplot(dataIS_WD, aes(fill=Condition, y=Freq, x=Cluster)) +
    geom_bar(position="fill", stat="identity")
```



Hide

```
#for MT=10%
oligos.integratedOL <- subset(oligos.integrated,seurat_clusters %in% c(1:6,8,9))
#subset nonOLlineage cells
nonOL <- colnames(oligos.integratedIm)[oligos.integratedIm@meta.data$seurat_clusters
 %in% c(3:6,8)]
oligos.integratedOL <- oligos.integratedOL[,! colnames(oligos.integratedOL) %in% nonO
L]
#non-integrated
#oligos.integratedOL <- subset(oligos.integrated,seurat_clusters %in% c(1:6,10))
#for MT=5%
#oligos.integratedOL <- subset(oligos.integrated,seurat_clusters %in% c(1:5))
DefaultAssay(oligos.integrated) <- "SCT"
# oligos.integratedOL <- RunPCA(oligos.integratedOL, verbose = FALSE,features=c("Opal
in","Ptgds","Apoe","S100b","Apod","Lamp1","Fos","Sepp1"),npcs=5,approx=FALSE)
oligos.integratedOL <- RunPCA(oligos.integratedOL, verbose = FALSE)#,features=GeneMar
kov)
ElbowPlot(oligos.integratedOL)
```



Hide

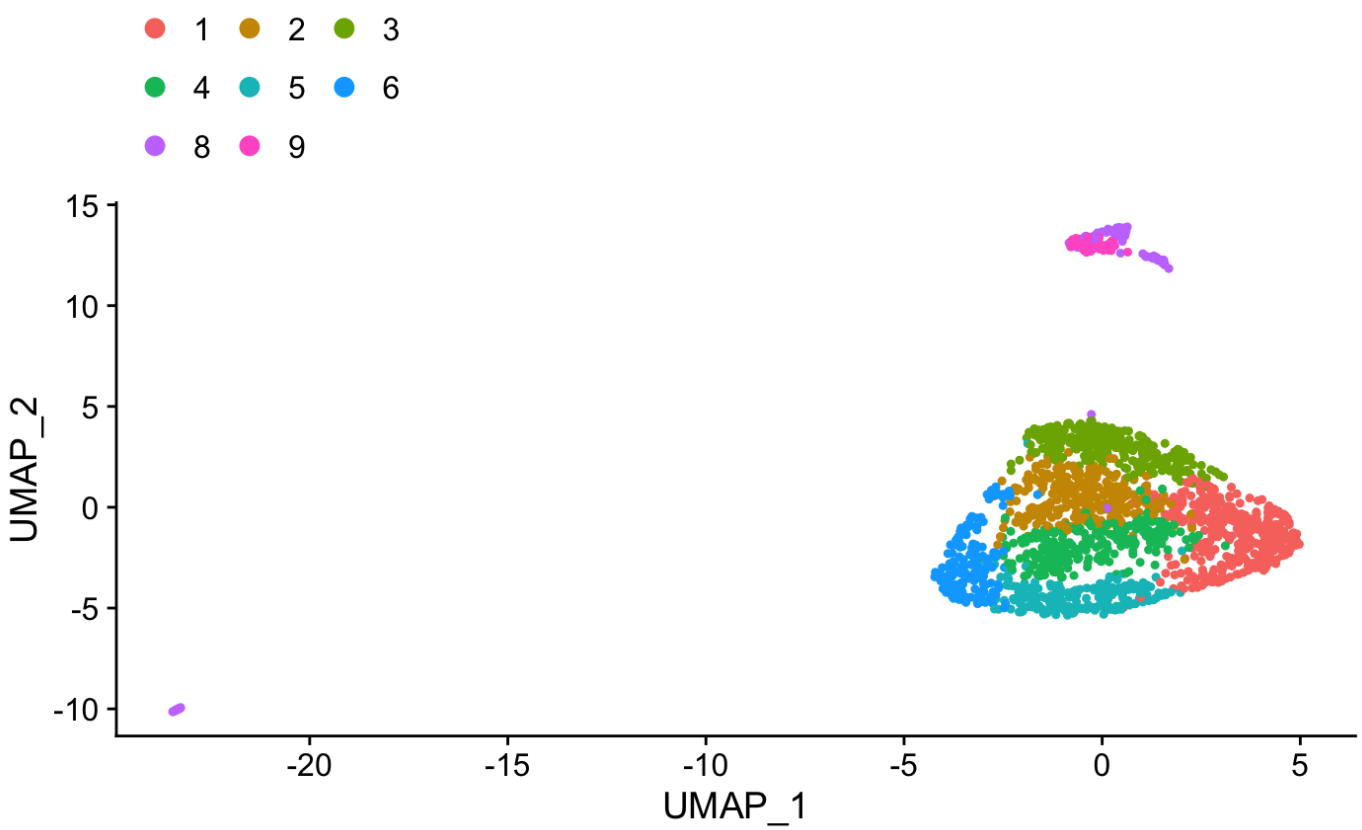
```
oligos.integratedOL <- RunUMAP(oligos.integratedOL, dims = 1:30)
```

```
16:51:45 UMAP embedding parameters a = 0.9922 b = 1.112
16:51:45 Read 1731 rows and found 30 numeric columns
16:51:45 Using Annoy for neighbor search, n_neighbors = 30
16:51:45 Building Annoy index with metric = cosine, n_trees = 50
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
16:51:45 Writing NN index file to temp file /var/folders/98/j14xl9ln5190l9fvq32g5rkw0
000gn/T//RtmpSdWnAZ/file3f2e463b204
16:51:45 Searching Annoy index using 1 thread, search_k = 3000
16:51:46 Annoy recall = 100%
16:51:46 Commencing smooth kNN distance calibration using 1 thread
16:51:47 Initializing from normalized Laplacian + noise
16:51:48 Commencing optimization for 500 epochs, with 68344 positive edges
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
16:51:51 Optimization finished
```

Hide

```
#oligos.integratedScience <- RunTSNE(oligos.integratedScience, dims = 1:10)
plots <- DimPlot(oligos.integratedOL, group.by = c("seurat_clusters"), combine = FALS
E)
plots <- lapply(X = plots, FUN = function(x) x + theme(legend.position = "top") + gui
des(color = guide_legend(nrow = 3,
    byrow = TRUE, override.aes = list(size = 3))))
CombinePlots(plots)
```

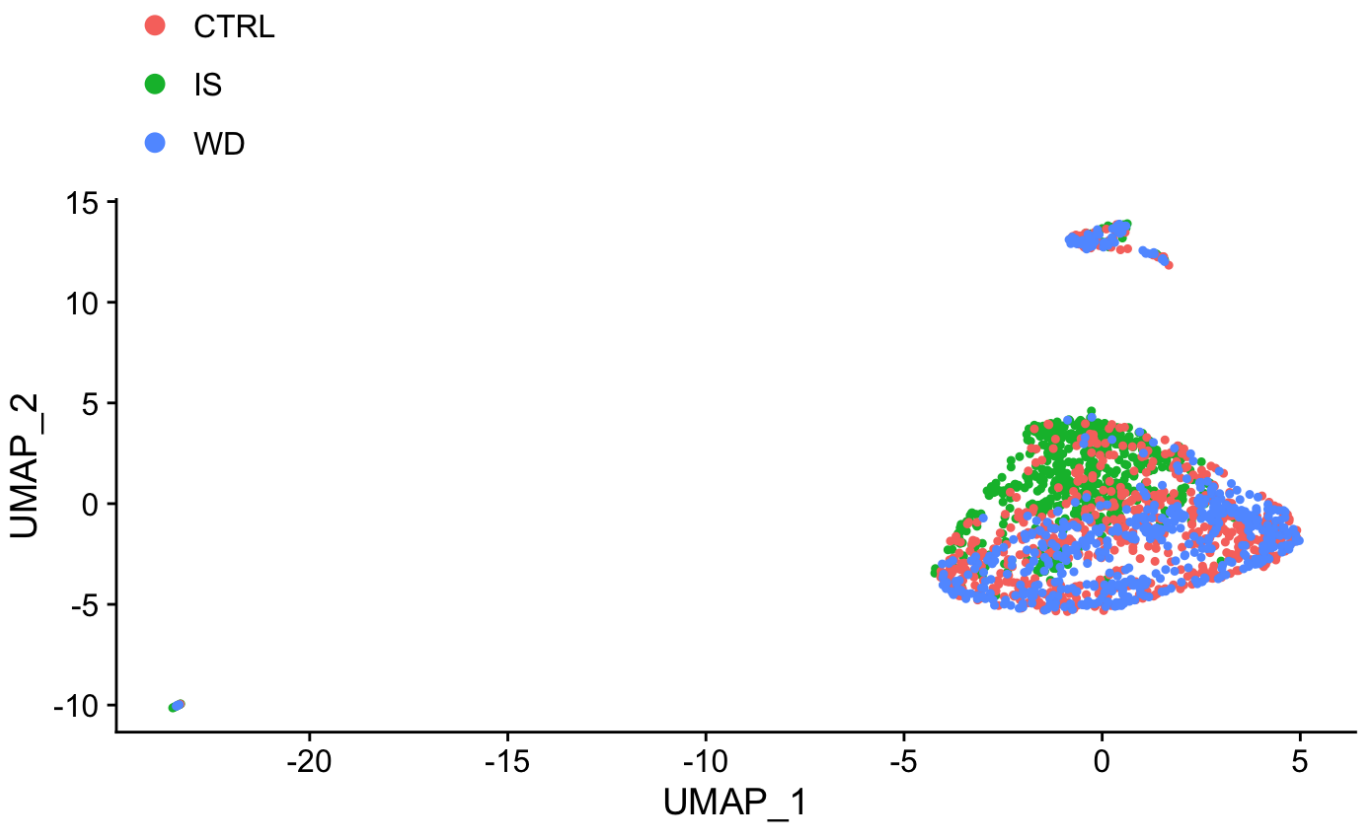
```
CombinePlots is being deprecated. Plots should now be combined using the patchwork sy
stem.
```



Hide

```
plots <- DimPlot(oligos.integratedOL, group.by = c("Sample"), combine = FALSE)
plots <- lapply(X = plots, FUN = function(x) x + theme(legend.position = "top") + gui
des(color = guide_legend(nrow = 3,
    byrow = TRUE, override.aes = list(size = 3))))
CombinePlots(plots)
```

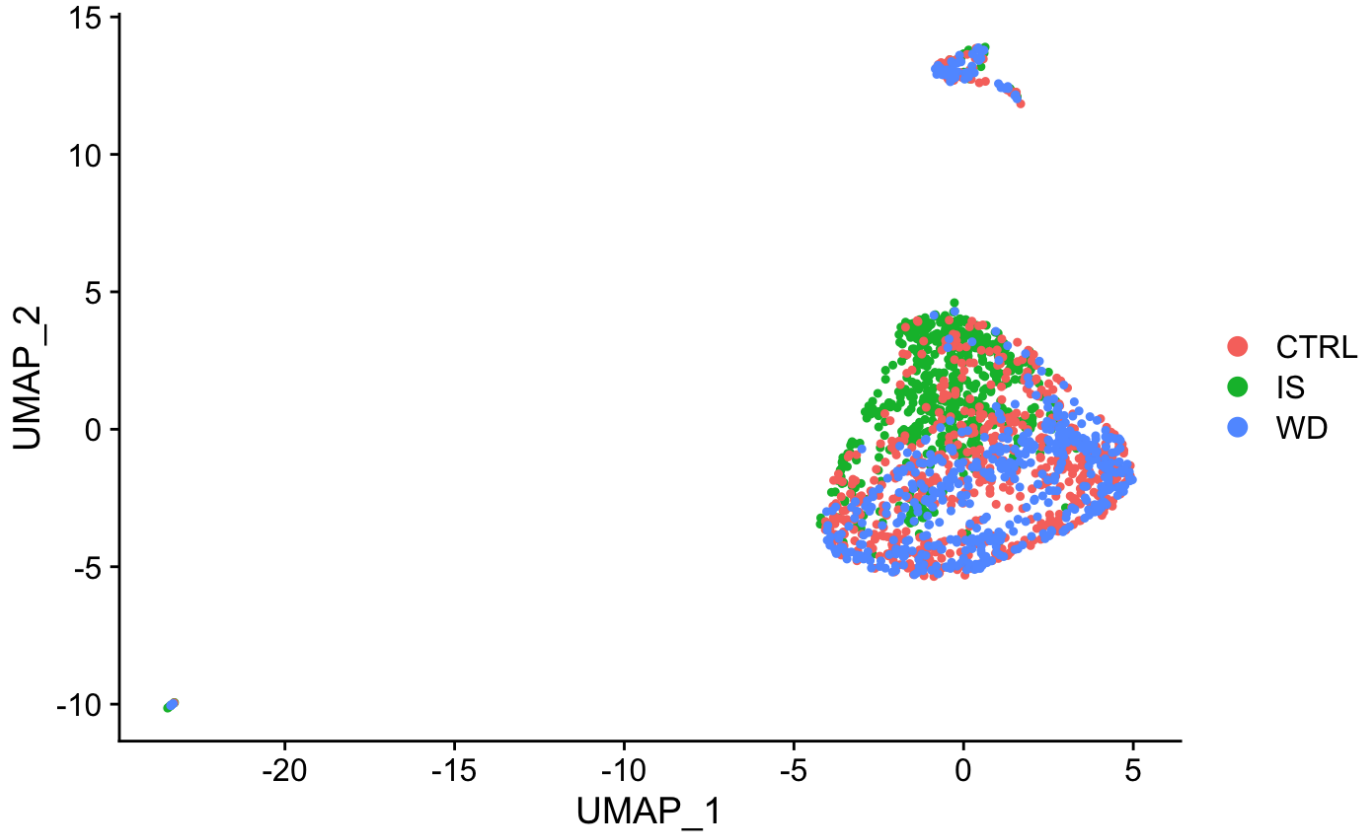
```
CombinePlots is being deprecated. Plots should now be combined using the patchwork sy
stem.
```



Hide

```
DimPlot(oligos.integratedOL, group.by = c("Sample"), combine = FALSE)
```

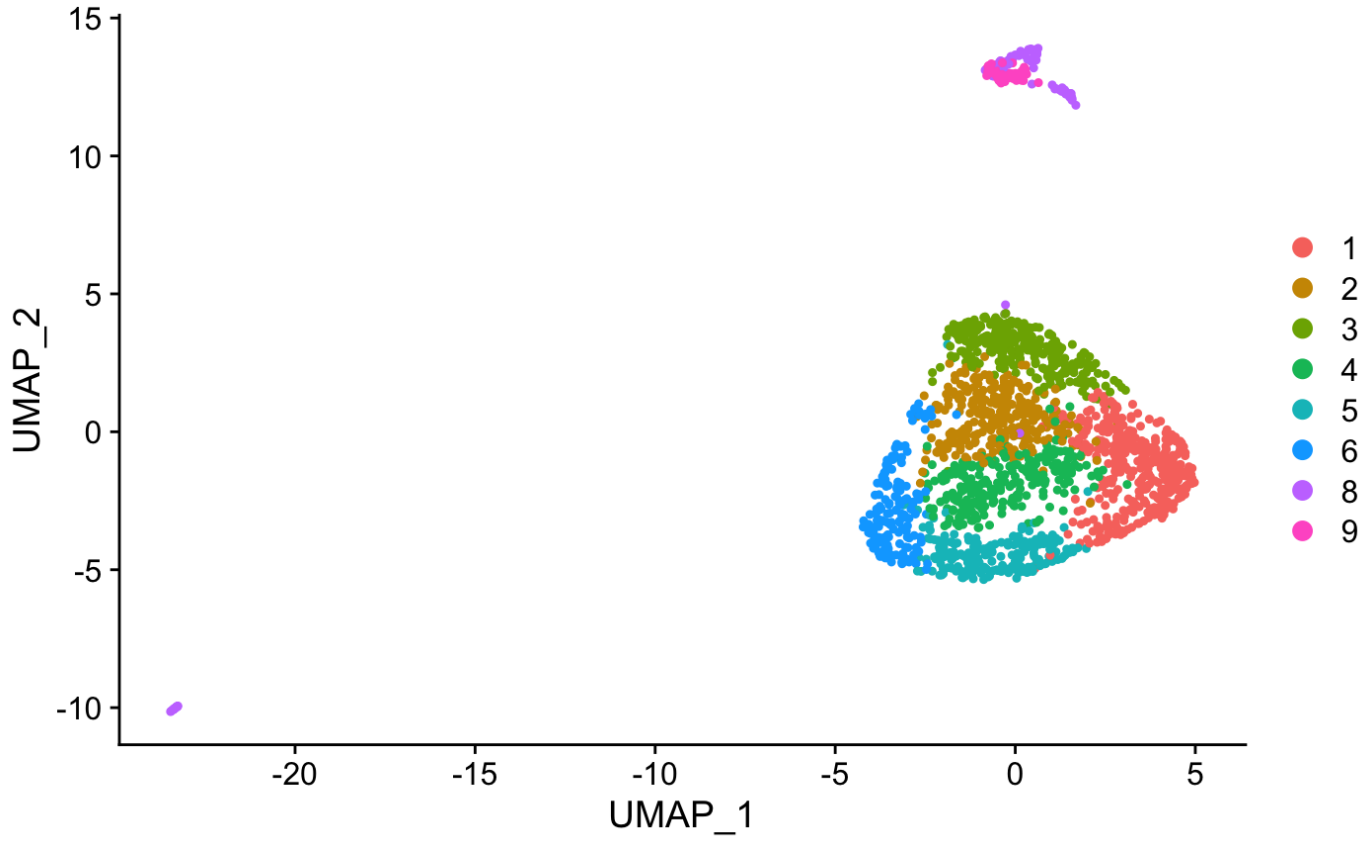
```
[[1]]
```



Hide

```
DimPlot(oligos.integratedOL, group.by = c("seurat_clusters"), combine = FALSE)
```

```
[[1]]
```

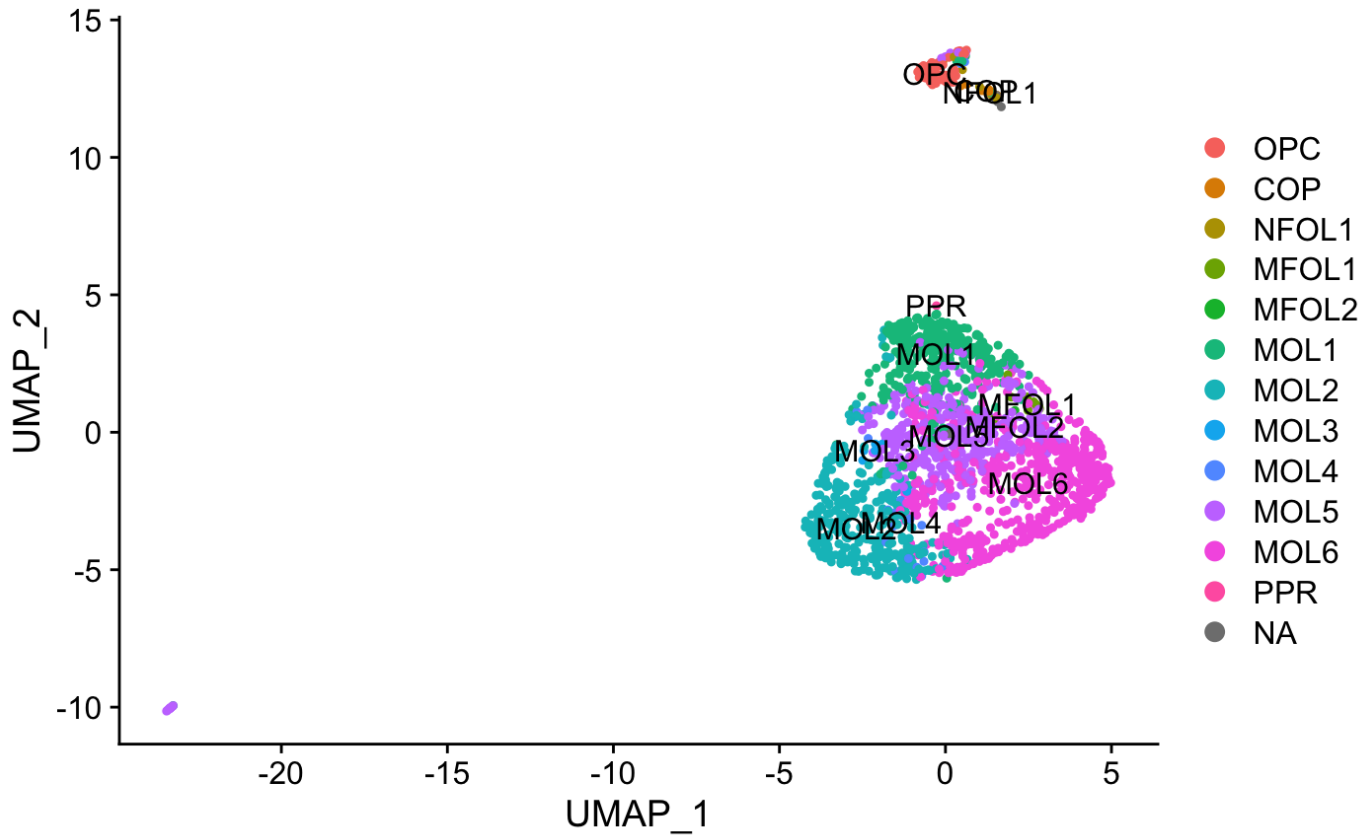


Hide

```
DimPlot(oligos.integratedOL, group.by = c("predicted.id"), combine = FALSE,label=TRUE
)
```

```
Using `as.character()` on a quosure is deprecated as of rlang 0.3.0.
Please use `as_label()` or `as_name()` instead.
 [90mThis warning is displayed once per session. [39m
```

```
[[1]]
```



Hide

```
oligos.integrated.markersOL %>% group_by(cluster) %>% top_n(n = 2, wt = avg_logFC)
```

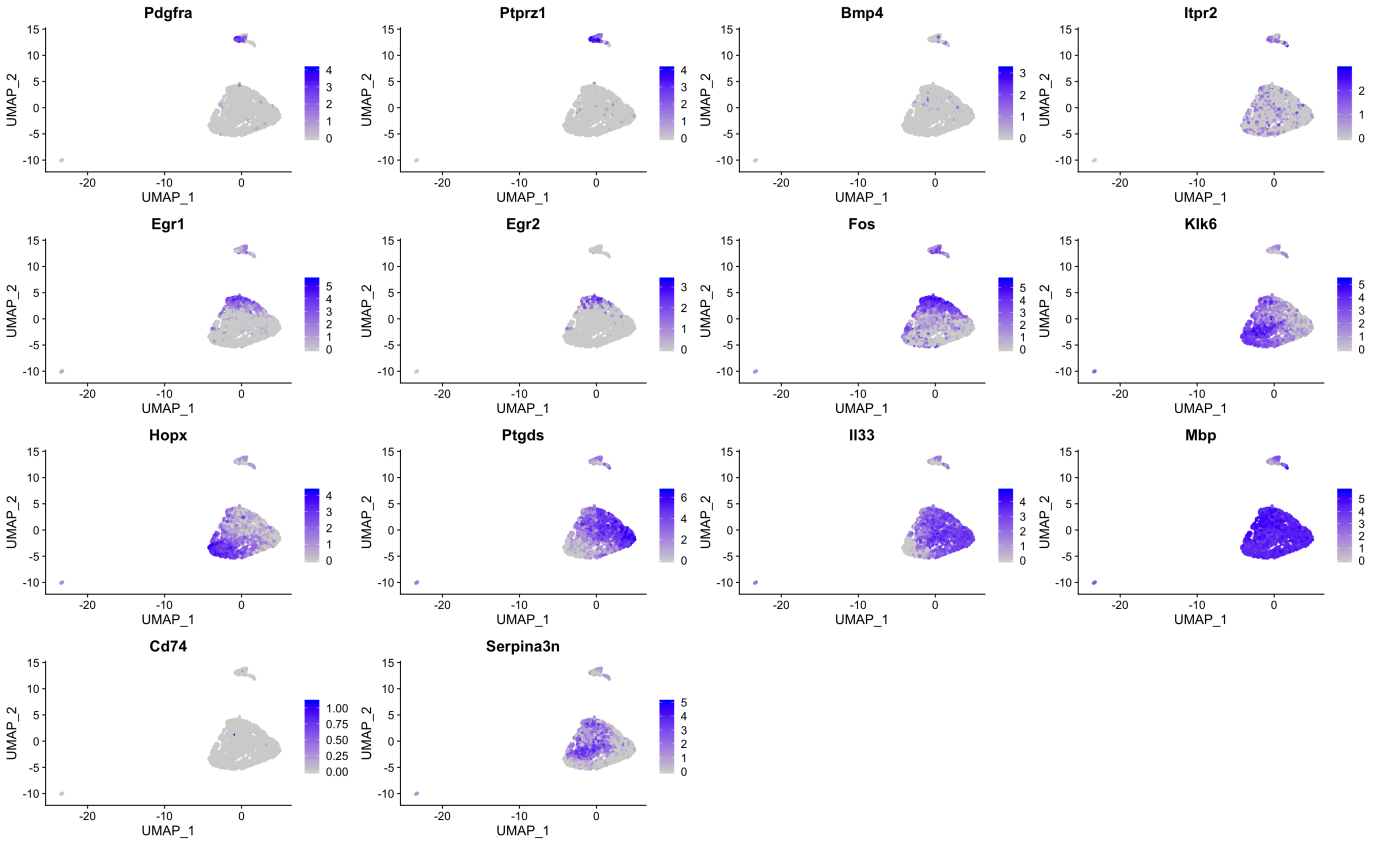
p_val <dbl>	avg_logFC <dbl>	pct.1 <dbl>	pct.2 <dbl>	p_val_adj <dbl>	cluster <fctr>	gene <chr>
1.802119e-212	3.2458801	0.974	0.035	2.843023e-208	OPC	Ptprz1
3.781718e-119	3.6580958	1.000	0.120	5.966039e-115	OPC	Cspg5
3.072570e-42	3.0340503	1.000	0.055	4.847286e-38	COP	Gpr17
2.399446e-09	3.1370126	1.000	0.578	3.785367e-05	COP	Fyn
2.814614e-06	1.5253263	1.000	0.579	4.440335e-02	NFOL1	Fyn
3.899398e-06	1.7224180	0.700	0.197	6.151690e-02	NFOL1	Tubb2b
8.038929e-09	1.9302515	1.000	0.285	1.268221e-04	MFOL1	Slc9a3r2
3.351630e-08	1.8818568	1.000	0.325	5.287532e-04	MFOL1	Ttyh1
2.992190e-03	2.0752835	0.333	0.033	1.000000e+00	MFOL2	Syt4

	p_val	avg_logFC	pct.1	pct.2	p_val_adj	cluster	gene
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fctr>	<chr>
	3.257815e-03	1.2654297	0.333	0.034	1.000000e+00	MFOL2	Fgl2

1-10 of 22 rows Previous **1** 2 3 Next

Hide

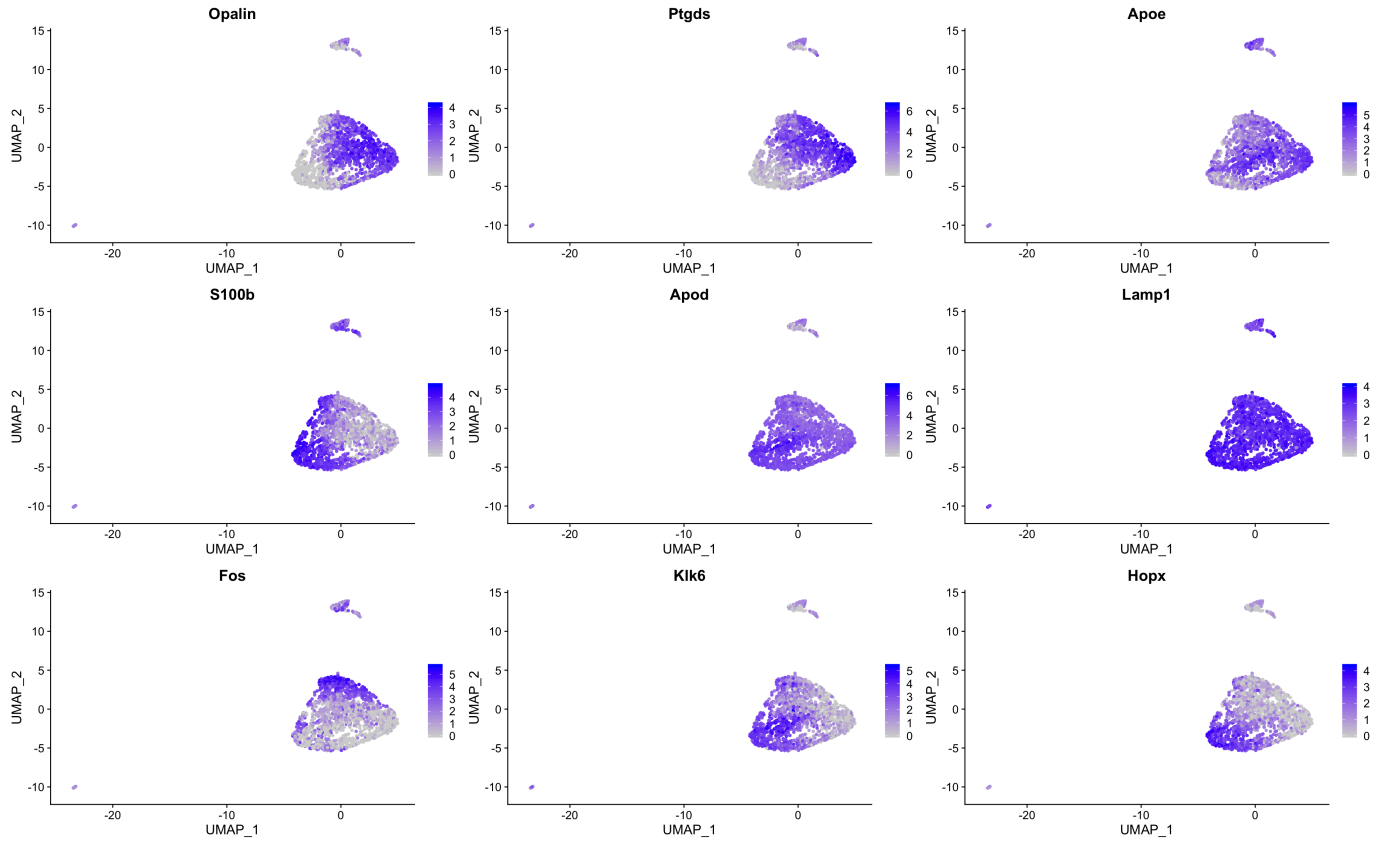
```
DefaultAssay(oligos.integratedOL) <- "SCT"
# Normalize RNA data for visualization purposes
#oligos.integrated <- NormalizeData(oligos.integrated, verbose = FALSE)
FeaturePlot(oligos.integratedOL, c("Pdgfra", "Ptprz1","Bmp4","Itpr2", "Egr1","Egr2",
"Fos","Klk6", "Hopx", "Ptgds","Il33","Mbp","Cd74","Serpina3n"),pt.size = 1)
```



Hide

```
FeaturePlot(oligos.integratedOL, c("Opalin","Ptgds","Apoe","S100b","Apod","Lamp1","Fo
s","Sepp1","Klk6","Hopx"),pt.size = 1)
```

```
The following requested variables were not found: Sepp1
```

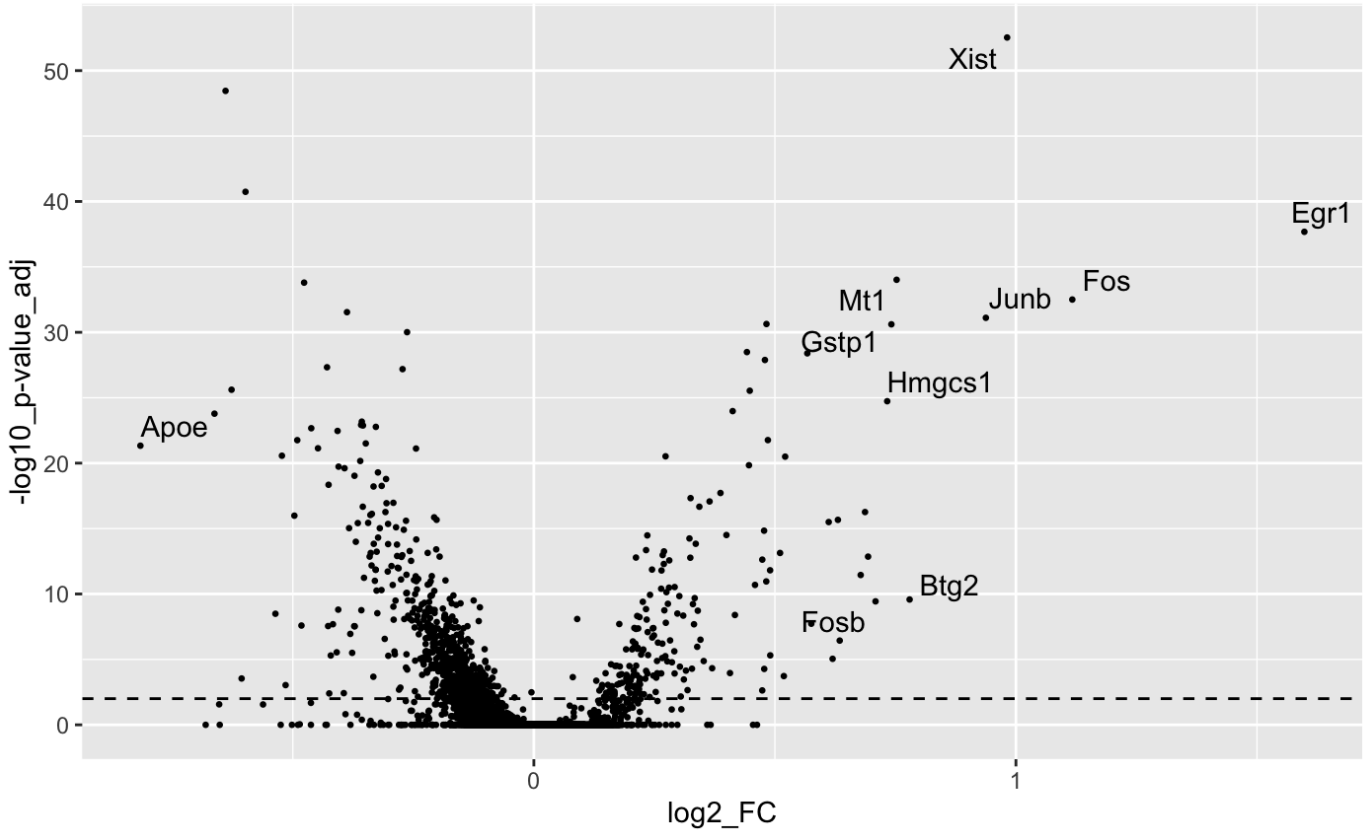


Hide

```
DefaultAssay(oligos.integrated) <- "SCT"
```

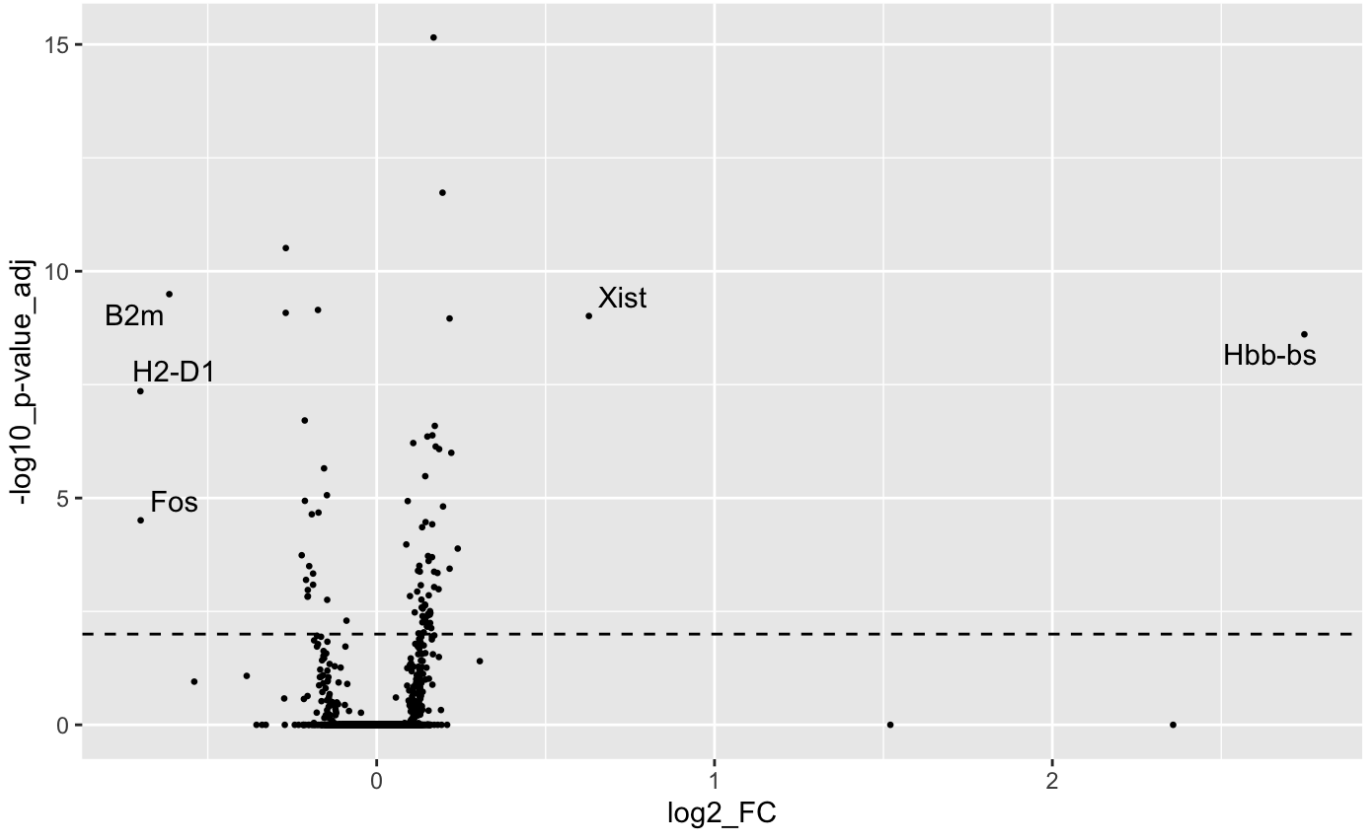
Hide

```
library(ggrepel)
DefaultAssay(oligos.integratedOL) <- "SCT"
Idents(oligos.integratedOL) <- "Sample"
oligos.integrated.samplediffIS <- FindMarkers(oligos.integratedOL, ident.1 = "IS", id
ent.2 = "CTRL", verbose = FALSE,logfc.threshold = 0,min.pct=0)
#head(oligos.integrated.samplediffAllRNA, n = 50)
diffmatrix <- oligos.integrated.samplediffIS
diffmatrix$logp_val <- -log10(diffmatrix$p_val_adj)
ggplot(diffmatrix,aes(avg_logFC,y=logp_val,label=row.names(diffmatrix)))+ geom_point
(size=0.5)+ geom_text_repel(data=subset(diffmatrix, p_val_adj < 0.01 & abs(avg_logFC)
> 0.7),                              label=row.names(subset(diffmatrix, p_val_adj <
0.01 & abs(avg_logFC) > 0.7)))+xlab("log2_FC") + ylab("-log10_p-value_adj") + geom_hl
ine(yintercept=-log10(0.01),linetype="dashed",size=0.5)
```



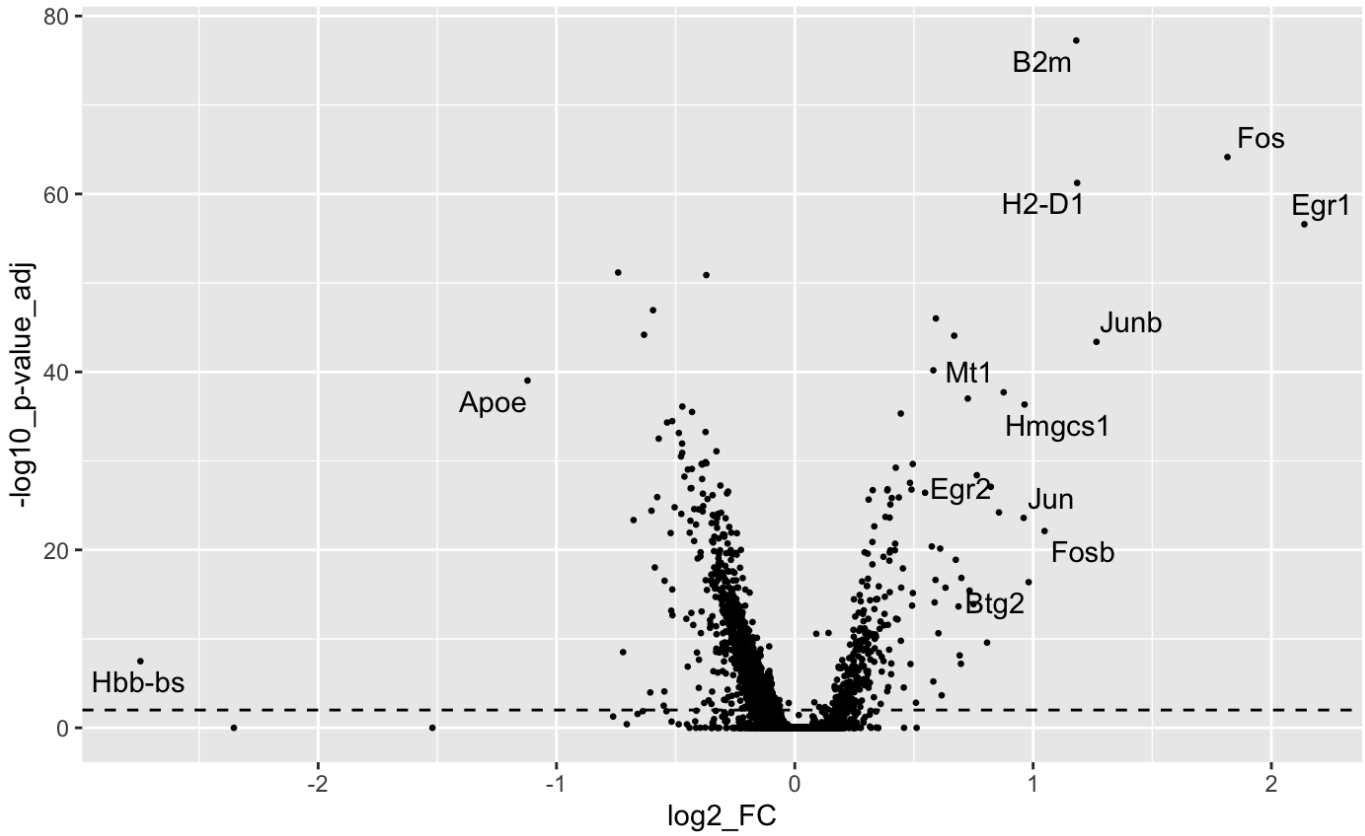
Hide

```
oligos.integrated.samplediffWD <- FindMarkers(oligos.integratedOL, ident.1 = "WD", id
ent.2 = c("CTRL"), verbose = FALSE,logfc.threshold = 0,min.pct=0)
#head(oligos.integrated.samplediffAllRNA, n = 50)
diffmatrix <- oligos.integrated.samplediffWD
diffmatrix$logp_val <- -log10(diffmatrix$p_val_adj)
ggplot(diffmatrix,aes(avg_logFC,y=logp_val,label=row.names(diffmatrix)))+ geom_point
(size=0.5)+ geom_text_repel(data=subset(diffmatrix, p_val_adj < 0.01 & abs(avg_logFC)
> 0.35),                              label=row.names(subset(diffmatrix, p_val_adj <
0.01 & abs(avg_logFC) > 0.35)))+xlab("log2_FC") + ylab("-log10_p-value_adj") + geom_h
line(yintercept=-log10(0.01),linetype="dashed",size=0.5)
```



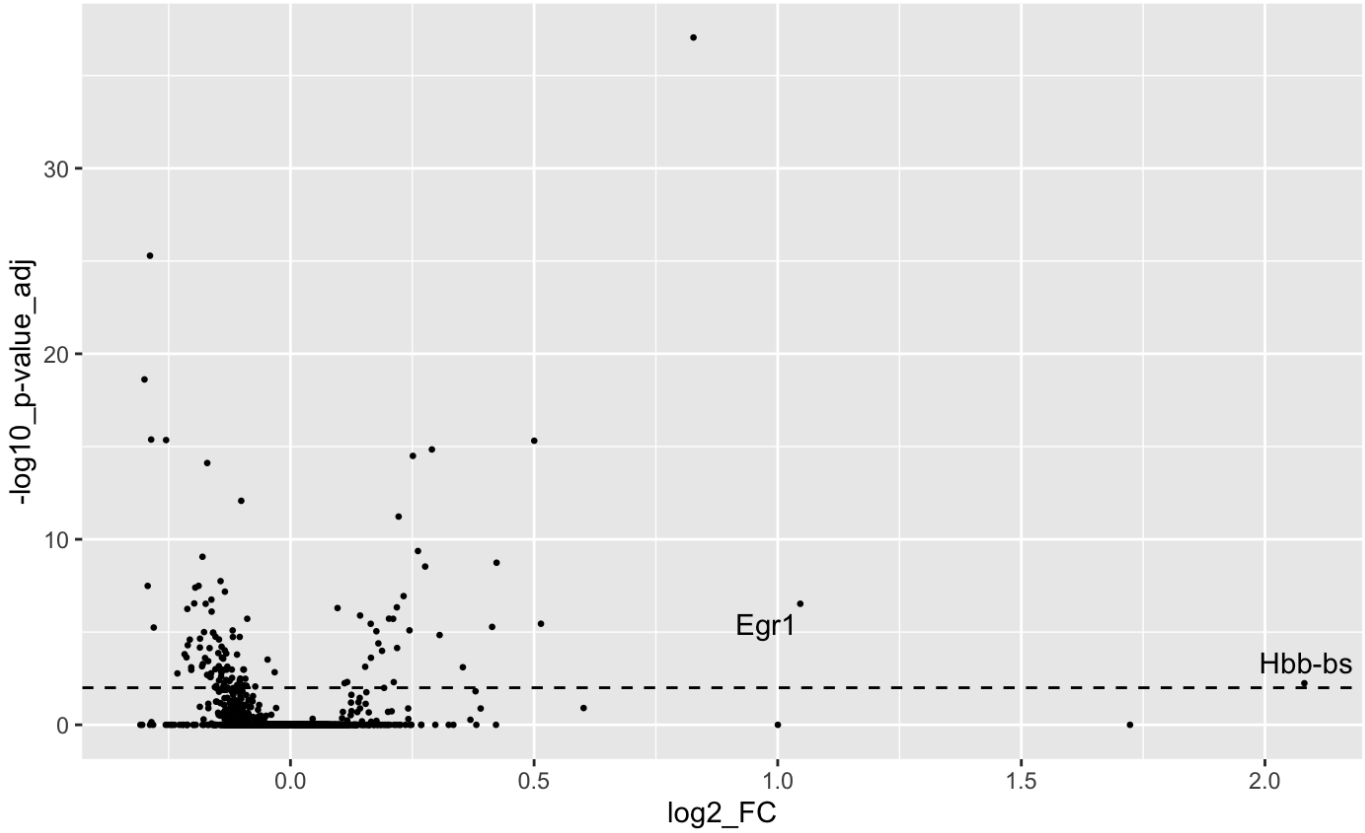
Hide

```
oligos.integrated.samplediffISWD <- FindMarkers(oligos.integratedOL, ident.1 = "IS",
 ident.2 = "WD", verbose = FALSE,logfc.threshold = 0,min.pct=0)
#head(oligos.integrated.samplediffAllRNA, n = 50)
diffmatrix <- oligos.integrated.samplediffISWD
diffmatrix$logp_val <- -log10(diffmatrix$p_val_adj)
ggplot(diffmatrix,aes(avg_logFC,y=logp_val,label=row.names(diffmatrix)))+ geom_point
(size=0.5)+ geom_text_repel(data=subset(diffmatrix, p_val_adj < 0.01 & abs(avg_logFC)
> 0.85),                              label=row.names(subset(diffmatrix, p_val_adj <
0.01 & abs(avg_logFC) > 0.85)))+xlab("log2_FC") + ylab("-log10_p-value_adj") + geom_h
line(yintercept=-log10(0.01),linetype="dashed",size=0.5)
```



Hide

```
oligos.integrated.samplediffISWDvsCNTRL <- FindMarkers(oligos.integratedOL, ident.1 =
c("IS","WD"), ident.2 = "CTRL", verbose = FALSE,logfc.threshold = 0,min.pct=0)
#head(oligos.integrated.samplediffAllRNA, n = 50)
diffmatrix <- oligos.integrated.samplediffISWDvsCNTRL
diffmatrix$logp_val <- -log10(diffmatrix$p_val_adj)
ggplot(diffmatrix,aes(avg_logFC,y=logp_val,label=row.names(diffmatrix)))+ geom_point
(size=0.5)+ geom_text_repel(data=subset(diffmatrix, p_val_adj < 0.01 & abs(avg_logFC)
> 0.85),                              label=row.names(subset(diffmatrix, p_val_adj <
0.01 & abs(avg_logFC) > 0.85)))+xlab("log2_FC") + ylab("-log10_p-value_adj") + geom_h
line(yintercept=-log10(0.01),linetype="dashed",size=0.5)
```



Hide

```
DefaultAssay(oligos.integrated) <- "SCT"
```

Hide

```
DiffMatrix <- list()
diffmatrixnames <- c("oligos.integrated.samplediffISWD",
                     "oligos.integrated.samplediffISWDvsCNTRL")

do.call(head,as.list(as.name(diffmatrixnames[1])))
```

	p_val <dbl>	avg_logFC <dbl>	pct.1 <dbl>	pct.2 <dbl>	p_val_adj <dbl>
B2m	3.576127e-82	1.1811544	0.951	0.509	5.641698e-78
Fos	4.468996e-69	1.8155714	0.911	0.459	7.050288e-65
H2-D1	3.650141e-66	1.1848700	0.919	0.573	5.758462e-62
Egr1	1.596893e-61	2.1383206	0.604	0.132	2.519259e-57
Gng11	4.263997e-56	-0.7412645	0.951	0.977	6.726882e-52
Aplp1	8.250248e-56	-0.3712214	1.000	1.000	1.301559e-51

6 rows

Hide

```
DiffMatrix <- list()
diffmatrixnames <- c("oligos.integrated.samplediffISWD",
                     "oligos.integrated.samplediffISWDvsCNTRL")

do.call(head,as.list(as.name(diffmatrixnames[1])))
```

	p_val <dbl>	avg_logFC <dbl>	pct.1 <dbl>	pct.2 <dbl>	p_val_adj <dbl>
B2m	3.576127e-82	1.1811544	0.951	0.509	5.641698e-78
Fos	4.468996e-69	1.8155714	0.911	0.459	7.050288e-65
H2-D1	3.650141e-66	1.1848700	0.919	0.573	5.758462e-62
Egr1	1.596893e-61	2.1383206	0.604	0.132	2.519259e-57
Gng11	4.263997e-56	-0.7412645	0.951	0.977	6.726882e-52
Aplp1	8.250248e-56	-0.3712214	1.000	1.000	1.301559e-51

6 rows

Hide

```
library(clusterProfiler)
```

```
clusterProfiler v3.10.1  For help: https://guangchuangyu.github.io/software/clusterProfiler

If you use clusterProfiler in published research, please cite:
Guangchuang Yu, Li-Gen Wang, Yanyan Han, Qing-Yu He. clusterProfiler: an R package for comparing biological themes among gene clusters. OMICS: A Journal of Integrative Biology. 2012, 16(5):284-287.
```

Hide

```
#Convert to gencode using biomart
library(biomaRt)
listMarts()
```

biomart <chr>	version <chr>
ENSEMBL_MART_ENSEMBL	Ensembl Genes 101
ENSEMBL_MART_MOUSE	Mouse strains 101
ENSEMBL_MART_SNP	Ensembl Variation 101
ENSEMBL_MART_FUNCGEN	Ensembl Regulation 101

4 rows

Hide

```
ensembl = useMart("ensembl",dataset="mmusculus_gene_ensembl")
listDatasets(ensembl)
```

dataset	desc
<S3: AsIs>	<S3
acalliptera_gene_ensembl	Eastern happy genes (fAst
acarolinensis_gene_ensembl	Anole lizard genes (Ano
acchrysaetos_gene_ensembl	Golden eagle genes (bAqu
acitrinellus_gene_ensembl	Midas cichlid genes (Mic
amelanoleuca_gene_ensembl	Panda genes (a
amexicanus_gene_ensembl	Mexican tetra genes (Astyanax_mexican
ampachon_gene_ensembl	Pachon cavefish genes (Astyanax_mexicanus
anancymaae_gene_ensembl	Ma's night monkey genes (Ana
aplatyrhynchos_gene_ensembl	Mallard genes (ASM874
applatyrhynchos_gene_ensembl	Duck genes (CAU_du

1-10 of 203 rows | 1-2 of 3 columns Previous **1** 2 3 4 5 6 … 21 Next

Hide

```
attributes = listAttributes(ensembl)
Biomart_gencode_ensembl84_biotypes <- getBM(attributes=c("mgi_symbol","ensembl_gene_i
d","entrezgene_id","gene_biotype"), filters = "", values = "", ensembl)
Biomart_gencode_ensembl84_biotypes[, 'gene_biotype'] <- as.factor(Biomart_gencode_ens
embl84_biotypes[,'gene_biotype'])
#Filter for only our genes
 Biotype_All_dataset <- subset(Biomart_gencode_ensembl84_biotypes, mgi_symbol %in% ol
igos.integrated@assays$SCT@var.features)
entrezID <-  subset(Biotype_All_dataset, Biotype_All_dataset$mgi_symbol %in% oligos.i
ntegrated@assays$SCT@var.features)
```

Hide

```
# if (!requireNamespace("BiocManager", quietly = TRUE))
#     install.packages("BiocManager")
#
# BiocManager::install("reactome.db")
library(ReactomePA)
```

```
ReactomePA v1.26.0  For help: https://guangchuangyu.github.io/ReactomePA

If you use ReactomePA in published research, please cite:
Guangchuang Yu, Qing-Yu He. ReactomePA: an R/Bioconductor package for reactome pathwa
y analysis and visualization. Molecular BioSystems 2016, 12(2):477-479
```

Hide

```
library(org.Mm.eg.db)
```

```
Loading required package: AnnotationDbi
Loading required package: stats4
Loading required package: BiocGenerics

Attaching package: 'BiocGenerics'

The following objects are masked from 'package:dplyr':

    combine, intersect, setdiff, union

The following objects are masked from 'package:Matrix':

    colMeans, colSums, rowMeans, rowSums, which

The following objects are masked from 'package:parallel':

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ, clusterExport, clusterMa
p,
    parApply, parCapply, parLapply, parLapplyLB, parRapply, parSapply, parSapplyLB

The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

    Filter, Find, Map, Position, Reduce, anyDuplicated, append, as.data.frame, basena
me, cbind,
    colMeans, colSums, colnames, dirname, do.call, duplicated, eval, evalq, get, gre
p, grepl,
    intersect, is.unsorted, lapply, lengths, mapply, match, mget, order, paste, pmax,
pmax.int,
    pmin, pmin.int, rank, rbind, rowMeans, rowSums, rownames, sapply, setdiff, sort,
table,
    tapply, union, unique, unsplit, which, which.max, which.min

Loading required package: Biobase
Welcome to Bioconductor

    Vignettes contain introductory material; view with 'browseVignettes()'. To cite
    Bioconductor, see 'citation("Biobase")', and for packages 'citation("pkgname")'.

Loading required package: IRanges
Loading required package: S4Vectors

Attaching package: 'S4Vectors'

The following objects are masked from 'package:dplyr':

    first, rename

The following object is masked from 'package:Matrix':

    expand

The following object is masked from 'package:plyr':
```

```
    rename

The following object is masked from 'package:base':

    expand.grid


Attaching package: 'IRanges'

The following objects are masked from 'package:dplyr':

    collapse, desc, slice

The following object is masked from 'package:sp':

    %over%

The following object is masked from 'package:plyr':

    desc


Attaching package: 'AnnotationDbi'

The following object is masked from 'package:dplyr':

    select
```

Hide

```
ReactomeTerms <- list()
i=1
#UP
pvaladj <- 0.01
logfc <- 0.25
for(i in 1:length(diffmatrixnames)){
diffmatrix <- do.call("as.data.frame",as.list(as.name(diffmatrixnames[i])))
diffmatrix <- subset(diffmatrix, p_val_adj < pvaladj & avg_logFC > logfc)
siggenes <- head(row.names(diffmatrix),50)
entrezmatched <- entrezID[entrezID$mgi_symbol %in% siggenes,]
#entrezID <- entrezID[! apply(entrezID[,c(1,3)], 1,function (x) anyNA(x)),]
allLLIDs <- entrezmatched$entrezgene
modulesReactome <- enrichPathway(gene=allLLIDs,organism="mouse",pvalueCutoff=0.01,qva
lueCutoff = 0.3,pAdjustMethod = "none", readable=T)
ReactomeTerms[[i]] <- modulesReactome
head(as.data.frame(modulesReactome))
print(i)
}
```

```
[1] 1
[1] 2
```

Hide

```
ReactomeTerms[which(lapply(ReactomeTerms,function(x) is.null(x))==TRUE)] <- "No_Gene
s"
#Add DOWN
pvaladj <- 0.01
logfc <- -0.25
offset <- length(ReactomeTerms)
for(i in 1:length(diffmatrixnames)){
  i=i+offset
diffmatrix <- do.call("as.data.frame",as.list(as.name(diffmatrixnames[i-offset])))
diffmatrix <- subset(diffmatrix, p_val_adj < pvaladj & avg_logFC < logfc)
siggenes <- head(row.names(diffmatrix),50)
entrezmatched <- entrezID[entrezID$mgi_symbol %in% siggenes,]
#entrezID <- entrezID[! apply(entrezID[,c(1,3)], 1,function (x) anyNA(x)),]
allLLIDs <- entrezmatched$entrezgene
modulesReactome <- enrichPathway(gene=allLLIDs,organism="mouse",pvalueCutoff=0.01,qva
lueCutoff = 0.3,pAdjustMethod = "none", readable=T)
ReactomeTerms[[i]] <- modulesReactome
head(as.data.frame(modulesReactome))
print(i)
}
```

```
[1] 3
[1] 4
```

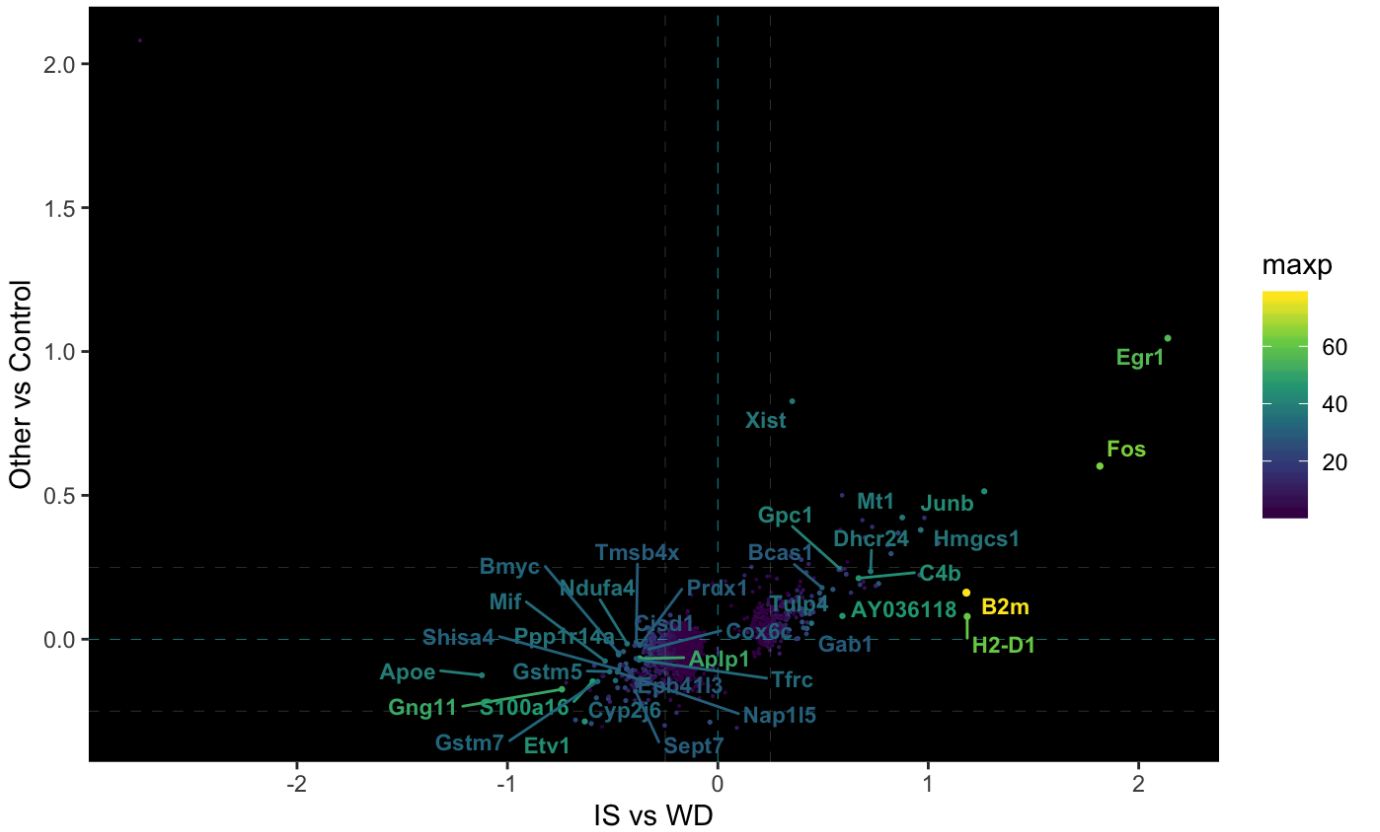
Hide

```
ReactomeTerms[which(lapply(ReactomeTerms,function(x) is.null(x))==TRUE)] <- "No_Gene
s"
```

Hide

```
Upper_diff <- subset(oligos.integrated.samplediffISWD, p_val_adj < 0.01 & abs(avg_log
FC) > 0)
Lower_diff <- subset(oligos.integrated.samplediffISWDvsCNTRL, p_val_adj < 0.01 & abs
(avg_logFC) > 0)
AlldiffgenesHetMOL5 <- intersect(intersect(row.names(oligos.integrated.samplediffISW
D),row.names(oligos.integrated.samplediffISWDvsCNTRL)),unique(c(row.names(Upper_dif
f),row.names(Lower_diff))))
subset2 <- oligos.integrated.samplediffISWD[AlldiffgenesHetMOL5,]
subset3 <- oligos.integrated.samplediffISWDvsCNTRL[AlldiffgenesHetMOL5,]
subsetMOL5 <- cbind(subset2,subset3)
colnames(subsetMOL5) <- make.unique(colnames(subsetMOL5))
diffmatrix <- subsetMOL5
diffmatrix$log_p_val <- -log10(diffmatrix$p_val_adj)
q95pgenes1 <- row.names(diffmatrix[which(diffmatrix$log_p_val >= quantile(diffmatrix
$log_p_val,0)),])
diffmatrix$log_p_val.1 <- -log10(diffmatrix$p_val_adj.1)
q95pgenes2 <- row.names(diffmatrix[which(diffmatrix$log_p_val.1 >= quantile(diffmatri
x$log_p_val.1,0)),])
q95pgenes <- unique(c(q95pgenes1,q95pgenes2))
diffmatrix <- diffmatrix[q95pgenes,]
diffmatrix$avg_logFC[is.infinite(diffmatrix$avg_logFC)] <- max(diffmatrix$avg_logFC[!
is.infinite(diffmatrix$avg_logFC)])
diffmatrix$avg_logFC.1[is.infinite(diffmatrix$avg_logFC.1)] <- max(diffmatrix$avg_log
FC.1[!is.infinite(diffmatrix$avg_logFC.1)])
#diffmatrix$avg_logFC.1 <- 2*diffmatrix$avg_logFC.1
diffmatrix$combp <- -log10(diffmatrix$p_val_adj*diffmatrix$p_val_adj.1)
diffmatrix$maxp <- apply(cbind(diffmatrix$log_p_val,diffmatrix$log_p_val.1),1,functio
n(x) max(x))
diffmatrix$minp <- apply(cbind(diffmatrix$p_val_adj,diffmatrix$p_val_adj.1),1,functio
n(x) min(x))
diffmatrix$maxp[is.infinite(diffmatrix$maxp)] <- max(diffmatrix$maxp[!is.infinite(dif
fmatrix$maxp)])
diffmatrix$maxFC <- apply(cbind(diffmatrix$avg_logFC,diffmatrix$avg_logFC.1),1,functi
on(x) max(abs(x)))
diffmatrix$Genes <- factor(row.names(diffmatrix),levels=row.names(diffmatrix))
ggplot(diffmatrix,aes(avg_logFC,y=avg_logFC.1,colour=maxp,label=row.names(diffmatri
x)))+ geom_point(size=diffmatrix$maxp/100) + scale_colour_viridis_c(direction = +1,op
tion ="viridis" ) + geom_hline(yintercept= 0,linetype="dashed",size=0.1,color="cyan")
+
  geom_hline(yintercept= 0.25,linetype="dashed",size=0.1,color="grey",alpha=0.5)+
  geom_hline(yintercept= -0.25,linetype="dashed",size=0.1,color="grey",alpha=0.5)+
  geom_vline(xintercept= 0,linetype="dashed",size=0.1,color="cyan")+
  geom_vline(xintercept= 0.25,linetype="dashed",size=0.1,color="grey",alpha=0.5)+
  geom_vline(xintercept= -0.25,linetype="dashed",size=0.1,color="grey",alpha=0.5)+
  geom_text_repel(size=3,fontface = "bold",force=1,data=subset(diffmatrix,
maxp > quantile(diffmatrix$maxp,0.98) #|
# avg_logFC > 0 |
# avg_logFC < -0|
# avg_logFC.1 > 0 |
# avg_logFC.1 < -0)
),label=row.names(subset(diffmatrix,
maxp > quantile(diffmatrix$maxp,0.98) #|
# avg_logFC > 0 |
# avg_logFC < -0 |
# avg_logFC.1 > 0 |
)# avg_logFC.1 < -0)
))+xlab("IS vs WD") + ylab("Other vs Control") +theme(
```

```
# get rid of panel grids
panel.grid.major = element_blank(),
#panel.grid.major = element_line(color="darkgrey",size=0.1),
panel.grid.minor = element_blank(),
#panel.grid.minor = element_line(color="darkgrey",size=0.05),
# Change plot and panel background
plot.background=element_rect(fill = "white"),
panel.background = element_rect(fill = 'black'),
# Change legend
legend.background = element_rect(fill = "white", color = NA),
legend.key = element_rect(color = "gray", fill = "white"),
legend.title = element_text(color = "Black"),
legend.text = element_text(color = "black")
)
```



Hide

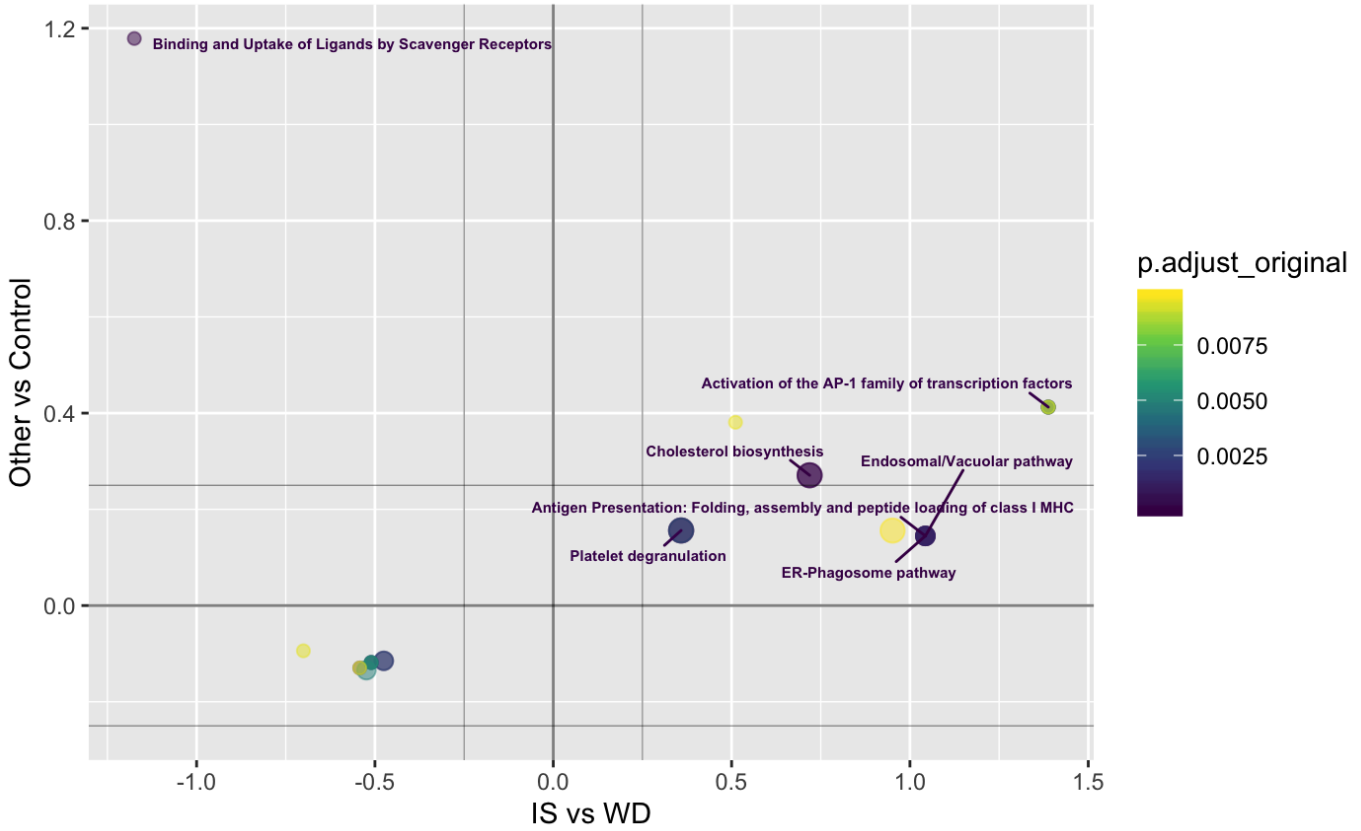
```
#magma,inferno, plasma,viridis
#scale_colour_gradient(low = "darkgreen", high = "red")
#Do reactome analysis at the bottom of script
i=1
j=1
#for(i in 1:length(ReactomeTerms)){
for(i in 1:4){
pwydata <- as.data.frame(ReactomeTerms[[i]])
geneset <- strsplit(pwydata$geneID, "/")
FCmeans <- data.frame()
for(j in 1:length(geneset)){
 geneset2FC <- which(row.names(diffmatrix) %in% geneset[[j]])
 FC <- mean(diffmatrix$avg_logFC[geneset2FC],na.rm=T)
 FCvar <- var(diffmatrix$avg_logFC[geneset2FC],na.rm=T)
 FC.1 <- mean(diffmatrix$avg_logFC.1[geneset2FC],na.rm=T)
 FC.1var <- var(diffmatrix$avg_logFC.1[geneset2FC],na.rm=T)
FCmeans <- rbind(FCmeans,cbind(FC,FC.1,FCvar,FC.1var))
 print(j)
}
ReactomeTerms[[i]] <- cbind(ReactomeTerms[[i]],FCmeans)
print(i)
}
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
[1] 16
[1] 1
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
[1] 20
[1] 21
[1] 22
[1] 23
[1] 24
[1] 25
[1] 26
[1] 27
[1] 28
[1] 29
[1] 30
[1] 31
[1] 32
[1] 33
[1] 34
[1] 35
[1] 36
[1] 37
[1] 38
[1] 39
[1] 40
```

```
[1] 41
[1] 42
[1] 43
[1] 44
[1] 45
[1] 46
[1] 47
[1] 48
[1] 2
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 3
[1] 1
[1] 2
[1] 3
[1] 4
```

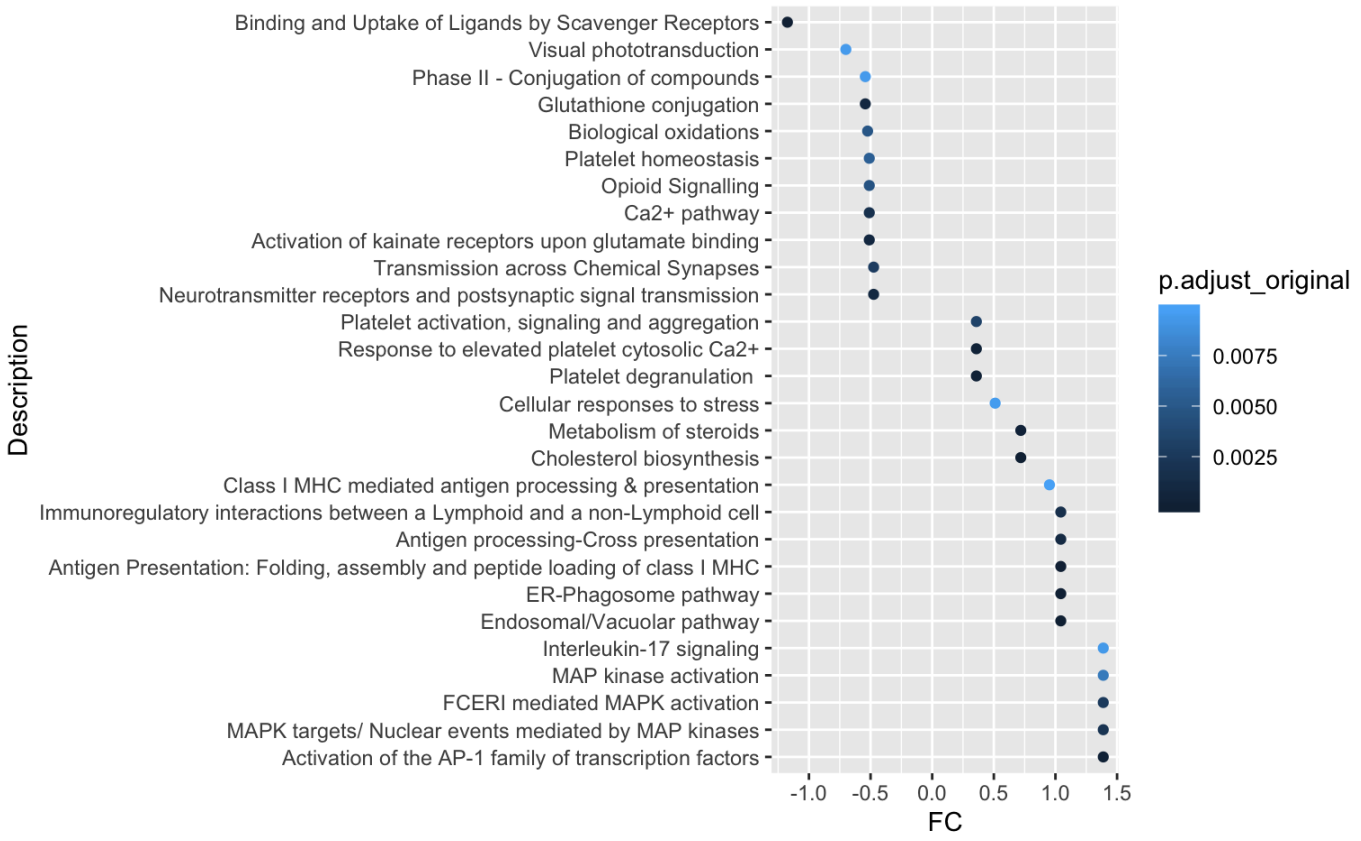
Hide

```
pathmatrix <- rbind(as.data.frame(ReactomeTerms[[1]]),as.data.frame(ReactomeTerms[[2
]]),as.data.frame(ReactomeTerms[[3]]),as.data.frame(ReactomeTerms[[4]]))
#pathmatrix <- rbind(as.data.frame(ReactomeTerms[[1]]),as.data.frame(ReactomeTerms
[[2]]))
#pathmatrix <- rbind(as.data.frame(ReactomeTerms[[3]]),as.data.frame(ReactomeTerms
[[4]]))
pathmatrix$p.adjust_original <- pathmatrix$p.adjust
pathmatrix$p.adjust <- -log10(pathmatrix$p.adjust )
pathmatrix$maxFC <- sum(abs(pathmatrix$FC),abs(pathmatrix$FC.1))
pathmatrix <- subset(pathmatrix, pathmatrix$Count > 1)
pathmatrix$AdjSelect <- pathmatrix$p.adjust*(500*(0.2+abs(pathmatrix$FC)))
#scale_colour_gradient(low = "yellow", high = "red") +
#scale_colour_viridis_c(direction = -1)
#scale_colour_gradient(low = "black", high = "red")
ggplot(pathmatrix,aes(FC,y=FC.1,colour=p.adjust_original),label=pathmatrix$Descriptio
n)+ geom_point(size=pathmatrix$Count,alpha=0.5) +scale_colour_viridis_c(direction = +
1,option = "viridis") +
  geom_hline(yintercept= 0,linetype="solid",size=0.5,color="black",alpha=0.5)+
  geom_hline(yintercept= 0.25,linetype="solid",size=0.2,color="black",alpha=0.5)+
  geom_hline(yintercept= -0.25,linetype="solid",size=0.2,color="black",alpha=0.5)+
  geom_vline(xintercept= 0,linetype="solid",size=0.5,color="black",alpha=0.5)+
  geom_vline(xintercept= 0.25,linetype="solid",size=0.2,color="black",alpha=0.5)+
  geom_vline(xintercept= -0.25,linetype="solid",size=0.2,color="black",alpha=0.5)+
  geom_text_repel(size=2,fontface="bold",force=20,data=
subset(pathmatrix,
abs(pathmatrix$AdjSelect) > quantile(
abs(pathmatrix$AdjSelect),1,na.rm=T) | abs(pathmatrix$p.adjust) > quantile(
abs(pathmatrix$p.adjust),0.75,na.rm=T) |
  abs(pathmatrix$FC.1) > quantile(abs(pathmatrix$FC.1),1,na.rm=T)),
label=subset(pathmatrix,
abs(pathmatrix$AdjSelect) > quantile(abs(pathmatrix$AdjSelect),1,na.rm=T) |
  abs(pathmatrix$p.adjust) > quantile(abs(pathmatrix$p.adjust),0.75,na.rm=T) |
  abs(pathmatrix$FC.1) > quantile(abs(pathmatrix$FC.1),1,na.rm=T))$Description,box.pa
dding = 0.5)+xlab("IS vs WD") + ylab("Other vs Control")
```



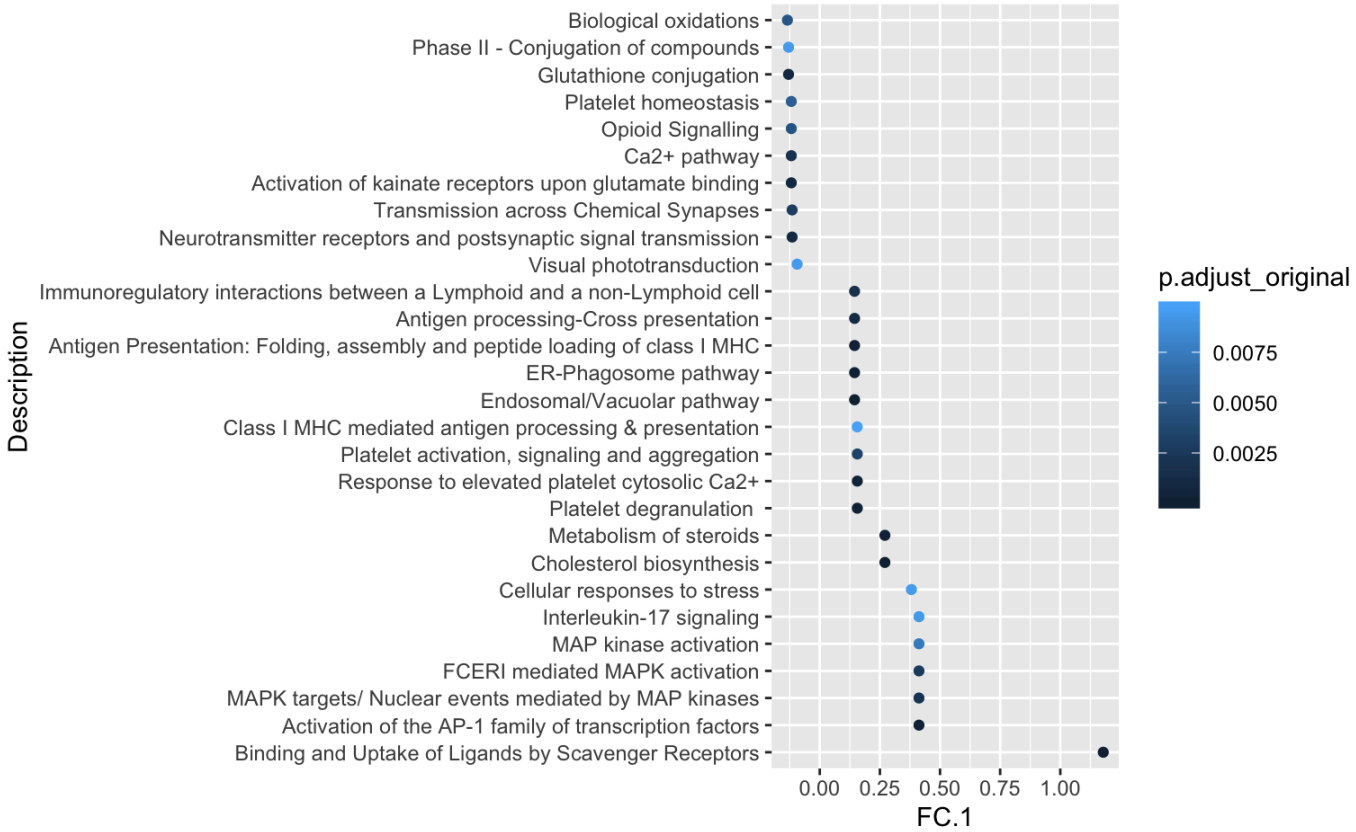
Hide

```
pathmatrixsort <- pathmatrix[order(pathmatrix$FC,decreasing=T),]
pathmatrixsort$Description <- factor(pathmatrixsort$Description, levels = unique(path
matrixsort$Description))
ggplot(pathmatrixsort, aes(x=FC, y=Description)) +
        geom_point(aes(color = p.adjust_original))
```



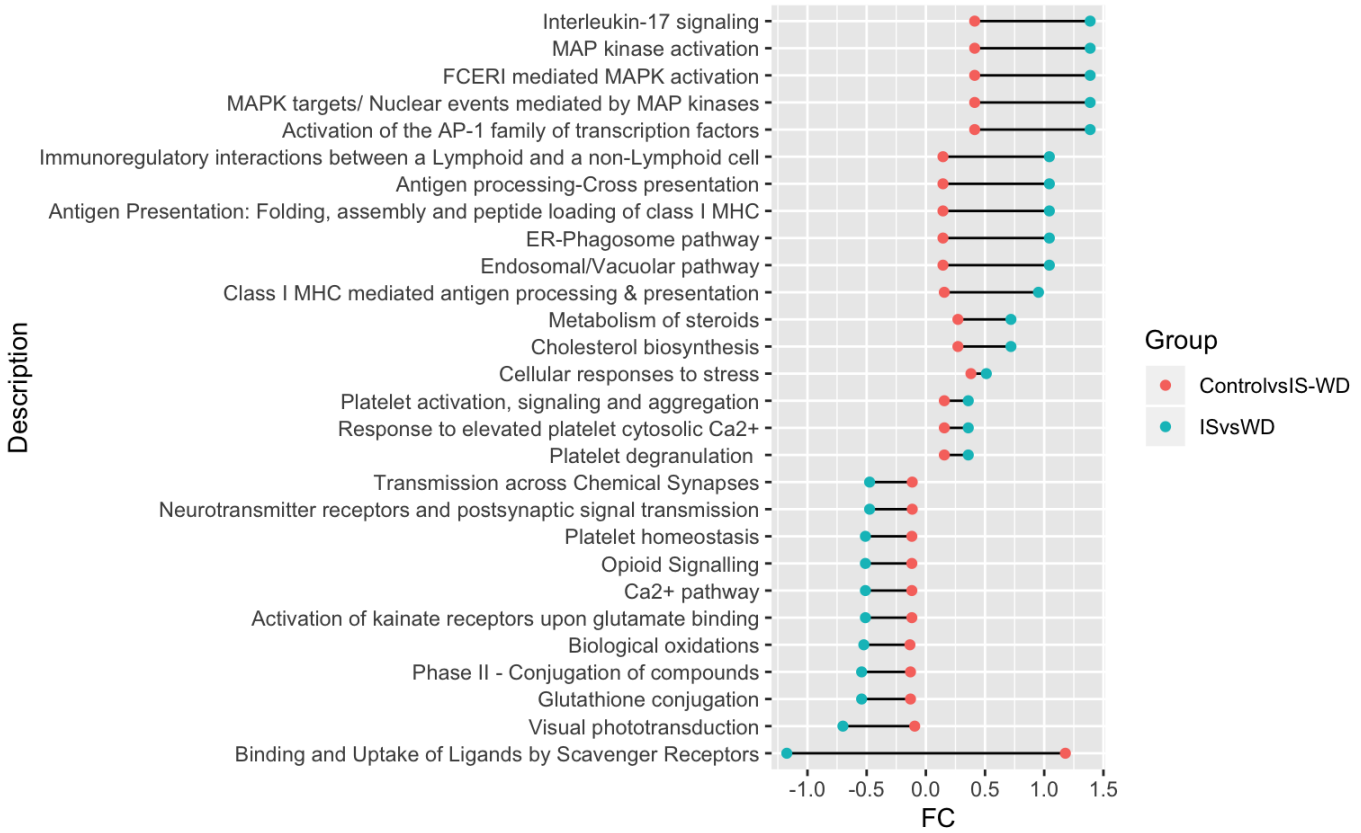
Hide

```
pathmatrixsort <- pathmatrix[order(pathmatrix$FC.1,decreasing=T),]
pathmatrixsort$Description <- factor(pathmatrixsort$Description, levels = unique(path
matrixsort$Description))
ggplot(pathmatrixsort, aes(x=FC.1, y=Description)) +
        geom_point(aes(color = p.adjust_original))
```



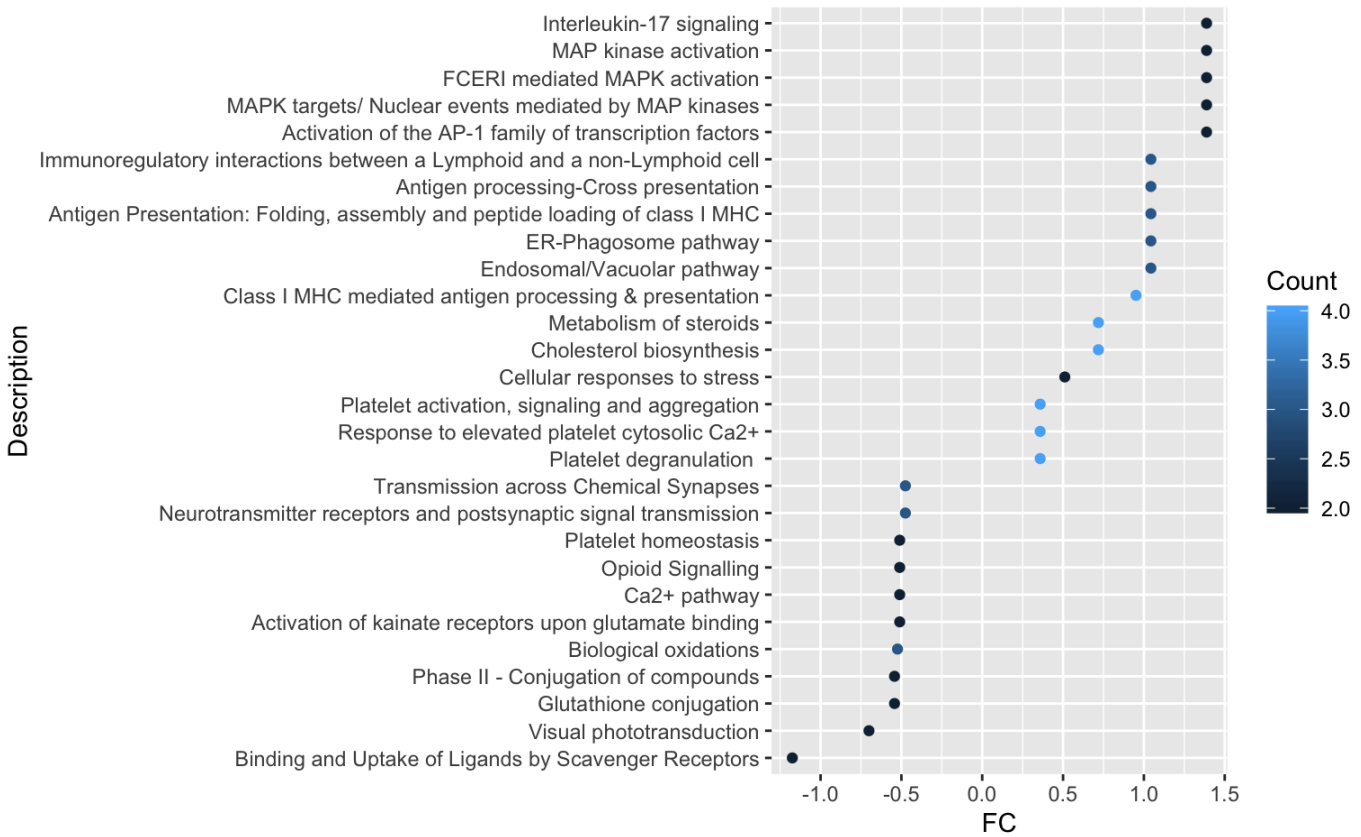
Hide

```
pathmatrixsort <- pathmatrix[order(pathmatrix$FC,decreasing=F),]
pathmatrixsort$Description <- factor(pathmatrixsort$Description, levels = unique(path
matrixsort$Description))
pathmatrixsort <- pathmatrixsort[!pathmatrixsort$p.adjust_original > 0.01,]
library(reshape2)
pathmatrixsortISWD <- pathmatrixsort[,c(2,10,14)]
pathmatrixsortISWD$Group <- rep("ISvsWD",nrow(pathmatrixsortISWD))
pathmatrixsortCntrlISWD <- pathmatrixsort[,c(2,11,14)]
colnames(pathmatrixsortCntrlISWD)[2] <- "FC"
pathmatrixsortCntrlISWD$Group <- rep("ControlvsIS-WD",nrow(pathmatrixsortCntrlISWD))
pathmatrixsort <- rbind(pathmatrixsortISWD,pathmatrixsortCntrlISWD)
ggplot(pathmatrixsort, aes(x=FC, y=Description)) +
        geom_line(aes(group = Description)) +
        geom_point(aes(color = Group))
```



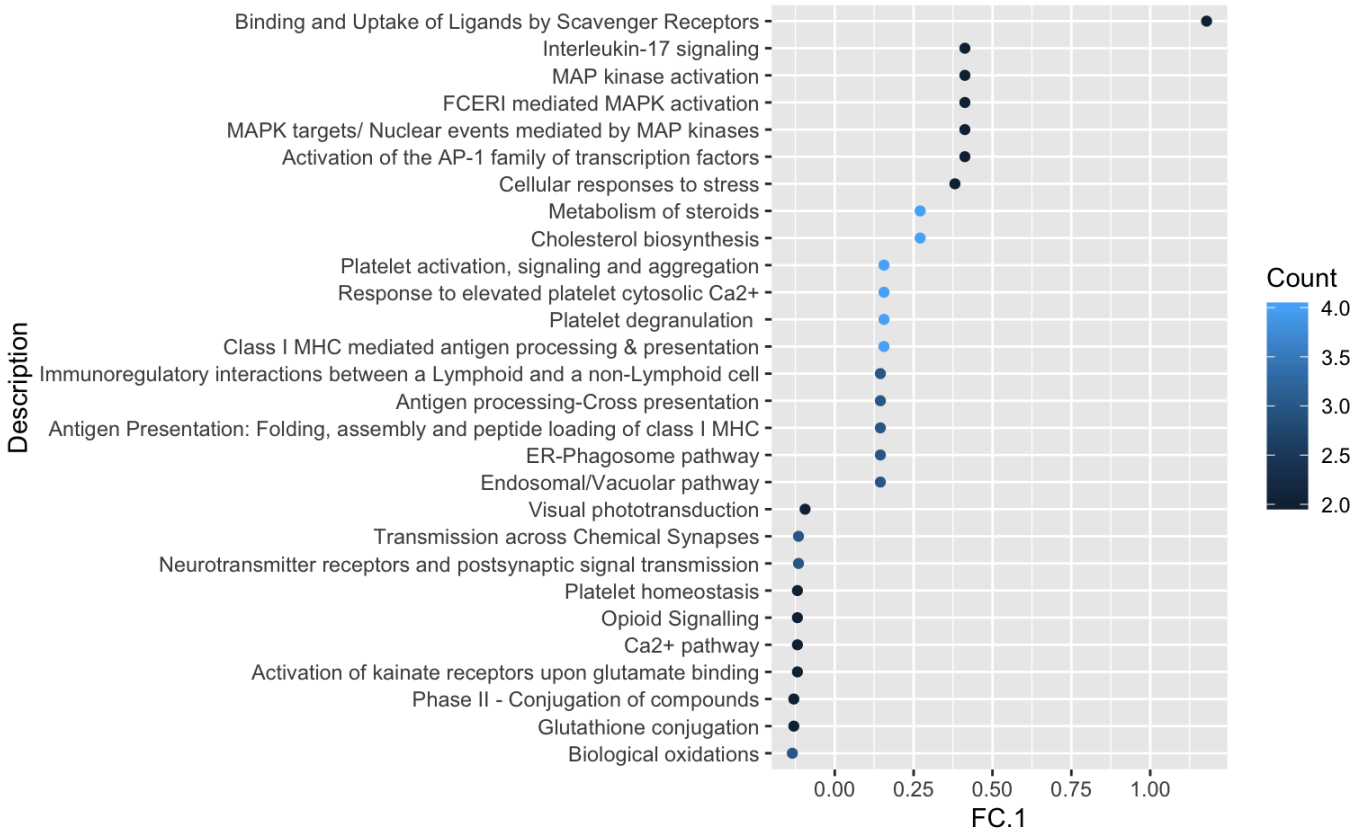
Hide

```
pathmatrixsort <- pathmatrix[order(pathmatrix$FC,decreasing=F),]
pathmatrixsort$Description <- factor(pathmatrixsort$Description, levels = unique(path
matrixsort$Description))
pathmatrixsort <- pathmatrixsort[!pathmatrixsort$p.adjust_original > 0.01,]
library(reshape2)
ggplot(pathmatrixsort, aes(x=FC, y=Description)) +
        geom_point(aes(color = Count))
```



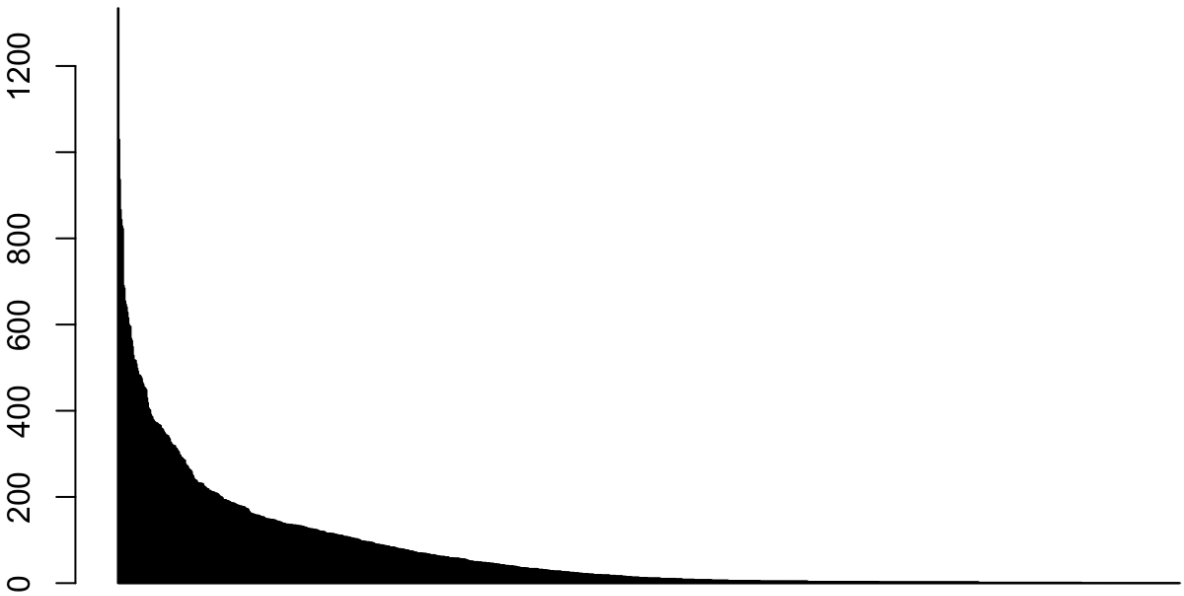
Hide

```
pathmatrixsort <- pathmatrix[order(pathmatrix$FC.1,decreasing=F),]
pathmatrixsort$Description <- factor(pathmatrixsort$Description, levels = unique(path
matrixsort$Description))
pathmatrixsort <- pathmatrixsort[!pathmatrixsort$p.adjust_original > 0.01,]
library(reshape2)
ggplot(pathmatrixsort, aes(x=FC.1, y=Description)) +
        geom_point(aes(color = Count))
```



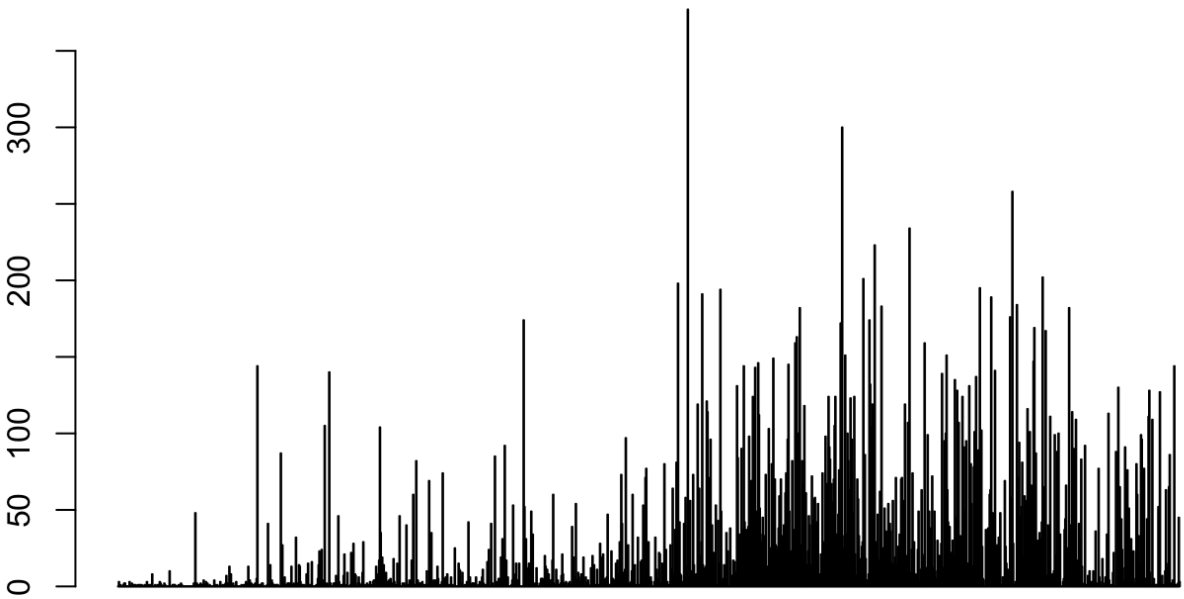
Hide

```
#Coexpression MOL5 and MOL6 vs MOL1
oligos.integratedOL256 <- subset(oligos.integrated,predicted.id %in% c("MOL2","MOL5",
"MOL6"))
Ptgdsexpression <- oligos.integratedOL256@assays$RNA@counts["Ptgds",]
Klk6expression <- oligos.integratedOL256@assays$RNA@counts["Klk6",]
ExpressionCombo <- as.data.frame(t(oligos.integratedOL256@assays$RNA@counts[c("Ptgds"
,"Klk6"),]))
#plot(x=log(ExpressionCombo$Ptgds+1),y=log(ExpressionCombo$Klk6+1))
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Ptgds,decreasing = TRU
E),]
barplot(ExpressionCombosorted$Ptgds)
```



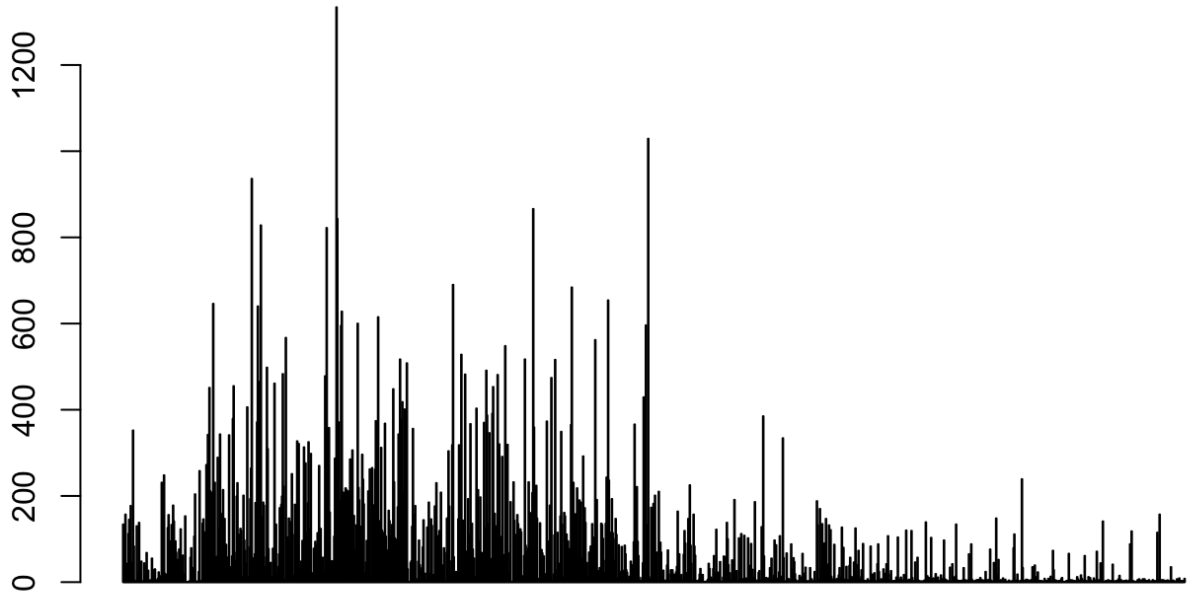
Hide

```
barplot(ExpressionCombosorted$Klk6)
```



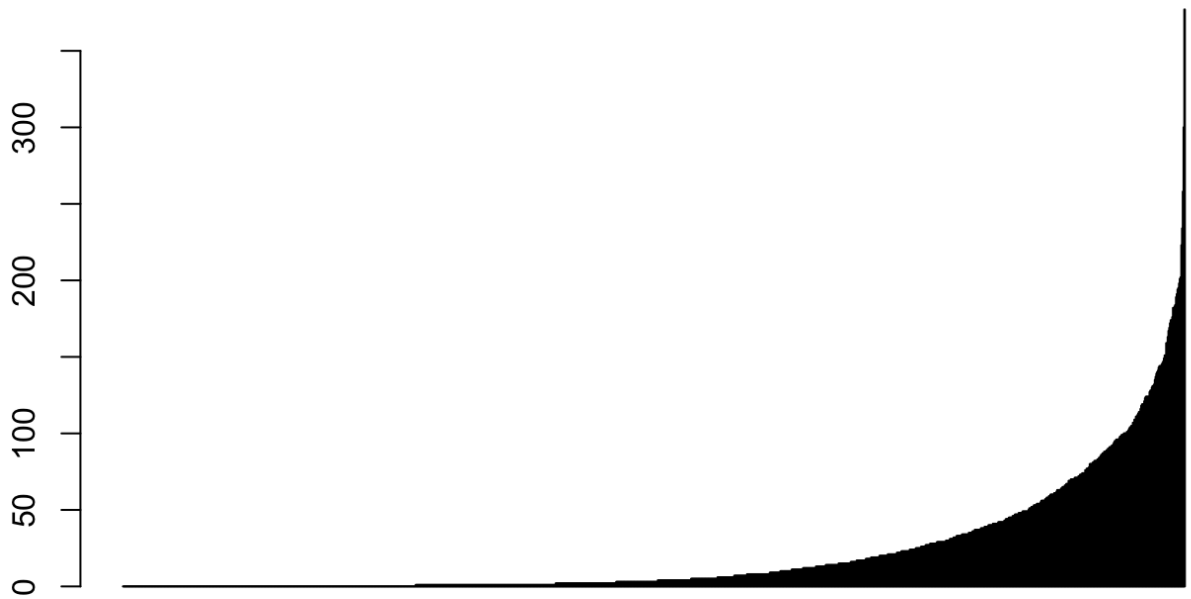
Hide

```
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Klk6,decreasing = FALS
E),]
barplot(ExpressionCombosorted$Ptgds)
```



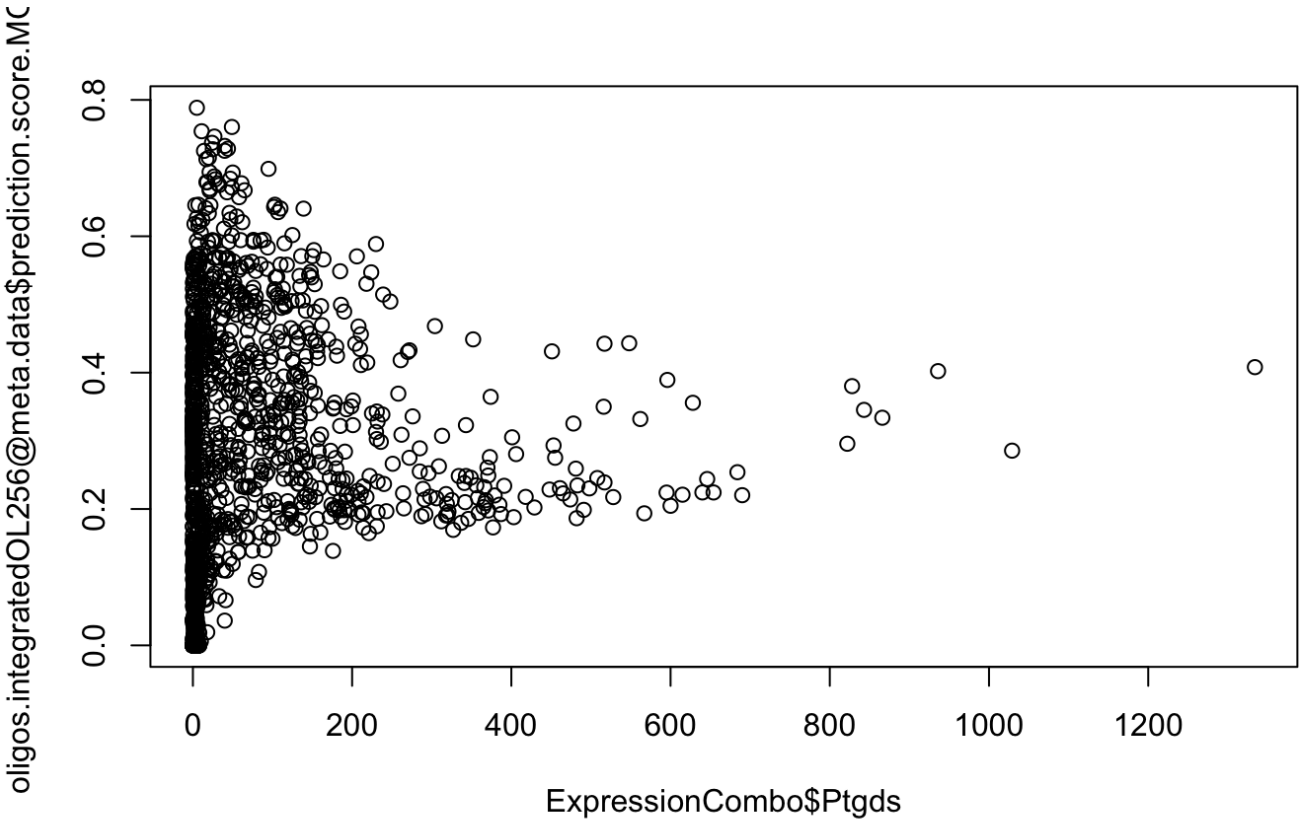
Hide

```
barplot(ExpressionCombosorted$Klk6)
```



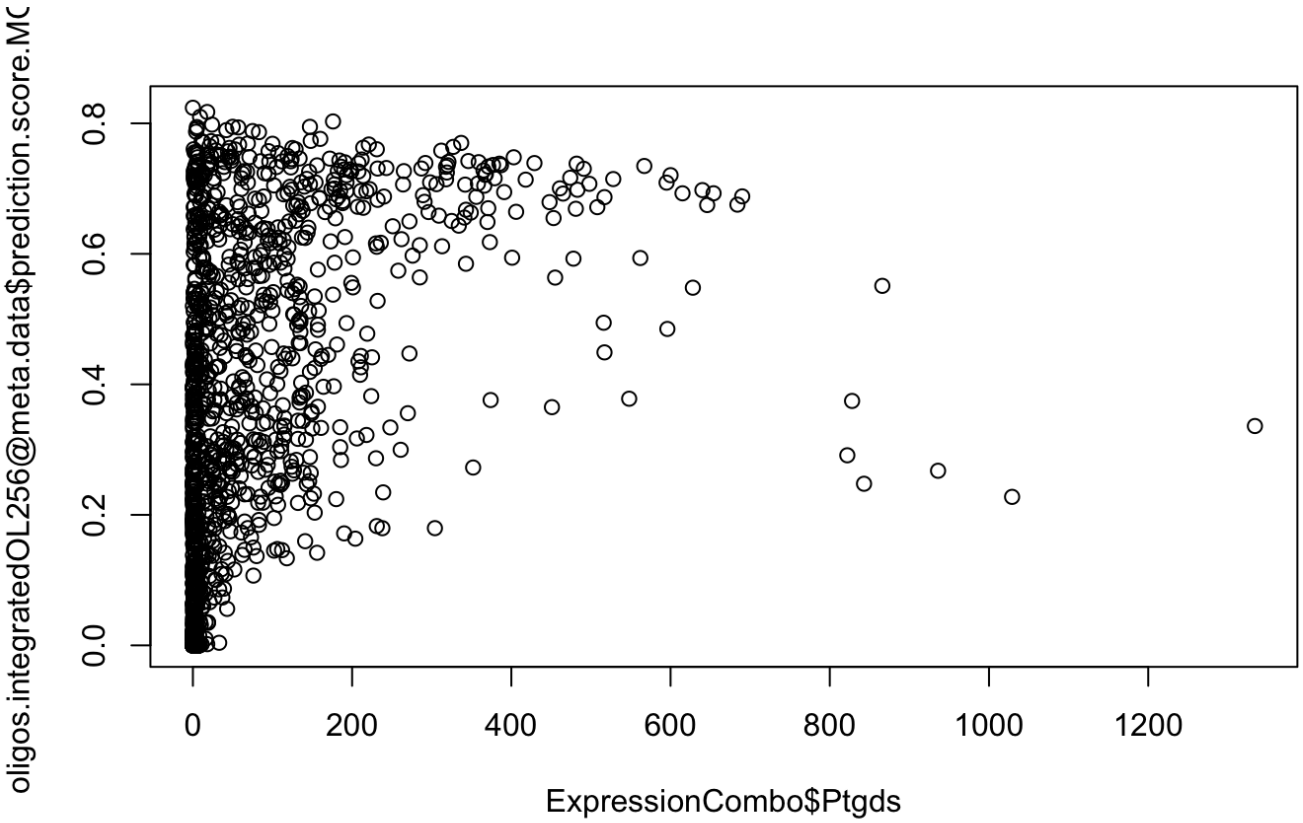
Hide

```
namesMOL56 <- oligos.integratedOL256@meta.data$predicted.id
plot(x=ExpressionCombo$Ptgds,y=oligos.integratedOL256@meta.data$prediction.score.MOL
5)
```



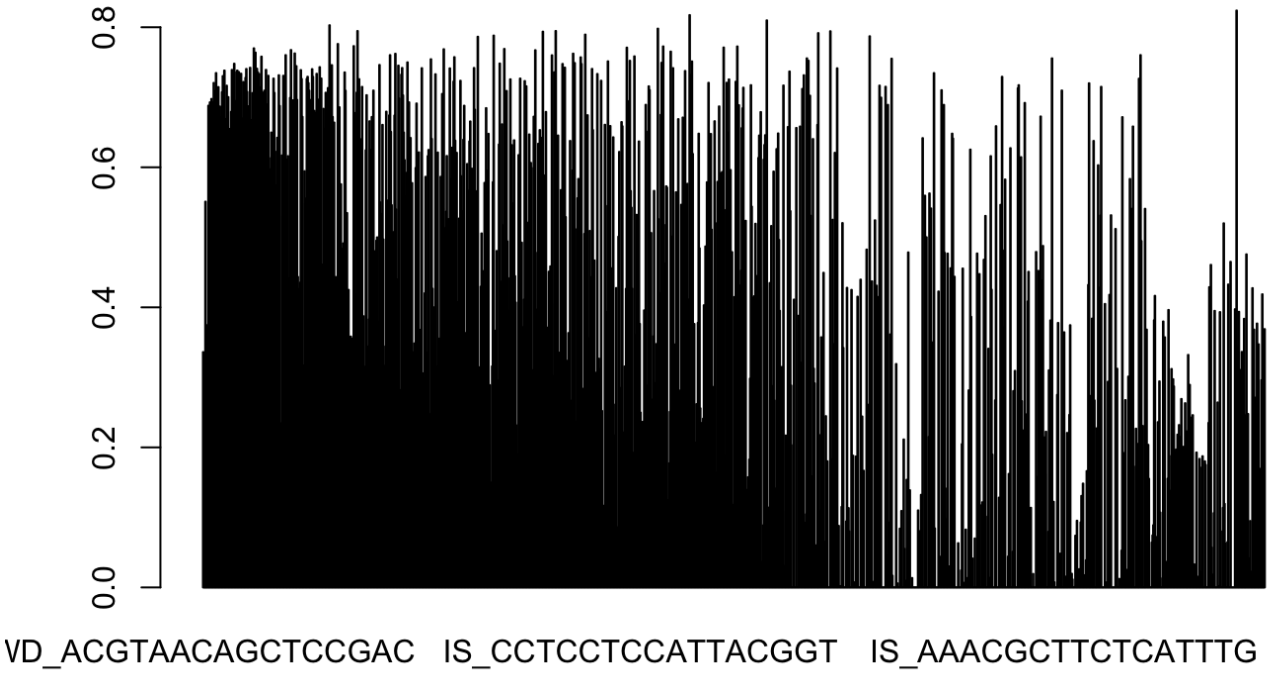
Hide

```
plot(x=ExpressionCombo$Ptgds,y=oligos.integratedOL256@meta.data$prediction.score.MOL
6)
```



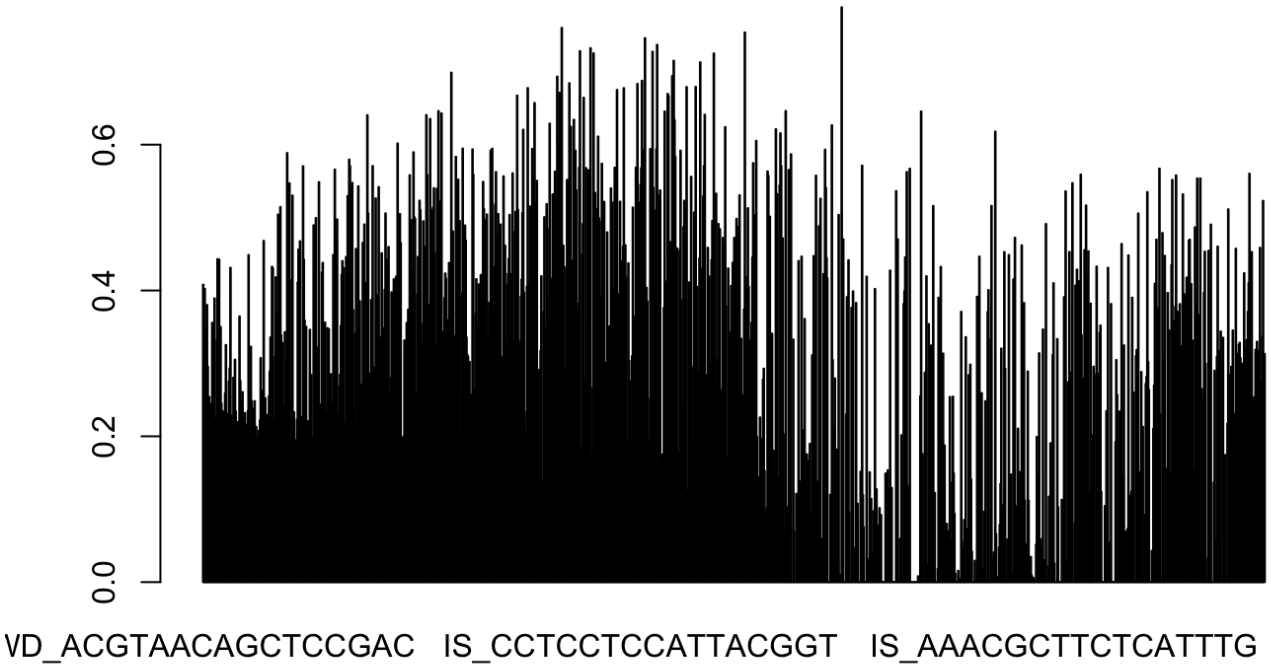
Hide

```
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Ptgds,decreasing = TRU
E),]
MOL6pred <- oligos.integratedOL256@meta.data$prediction.score.MOL6
names(MOL6pred) <- row.names(oligos.integratedOL256@meta.data)
barplot(MOL6pred[row.names(ExpressionCombosorted)])
```



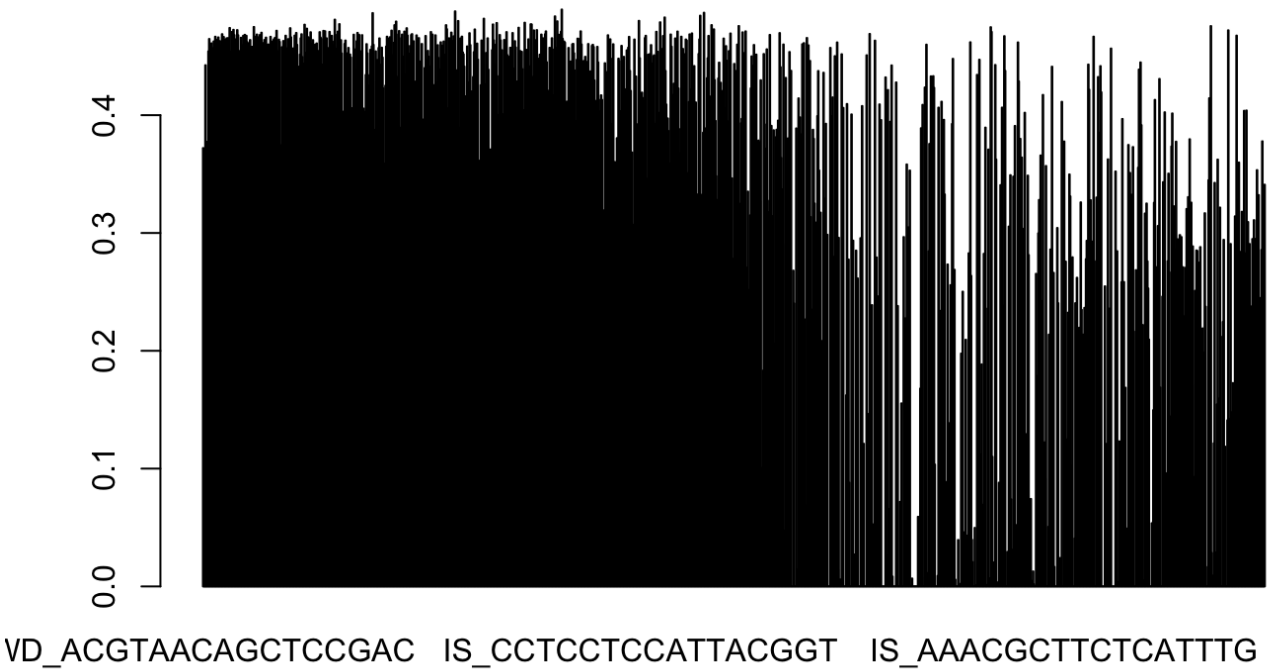
Hide

```
MOL5pred <- oligos.integratedOL256@meta.data$prediction.score.MOL5
names(MOL5pred) <- row.names(oligos.integratedOL256@meta.data)
barplot(MOL5pred[row.names(ExpressionCombosorted)])
```



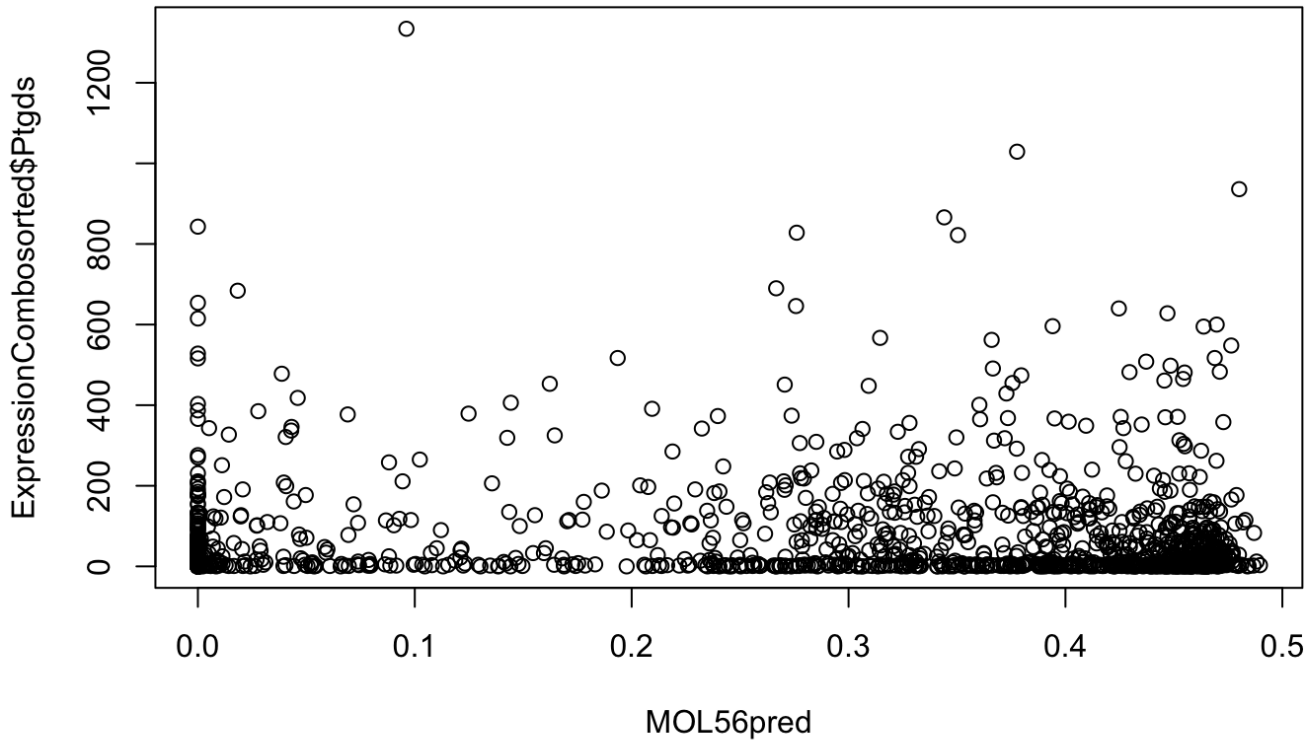
Hide

```
MOL56pred <- rowMeans(cbind(MOL5pred,MOL6pred))
barplot(MOL56pred[row.names(ExpressionCombosorted)])
```



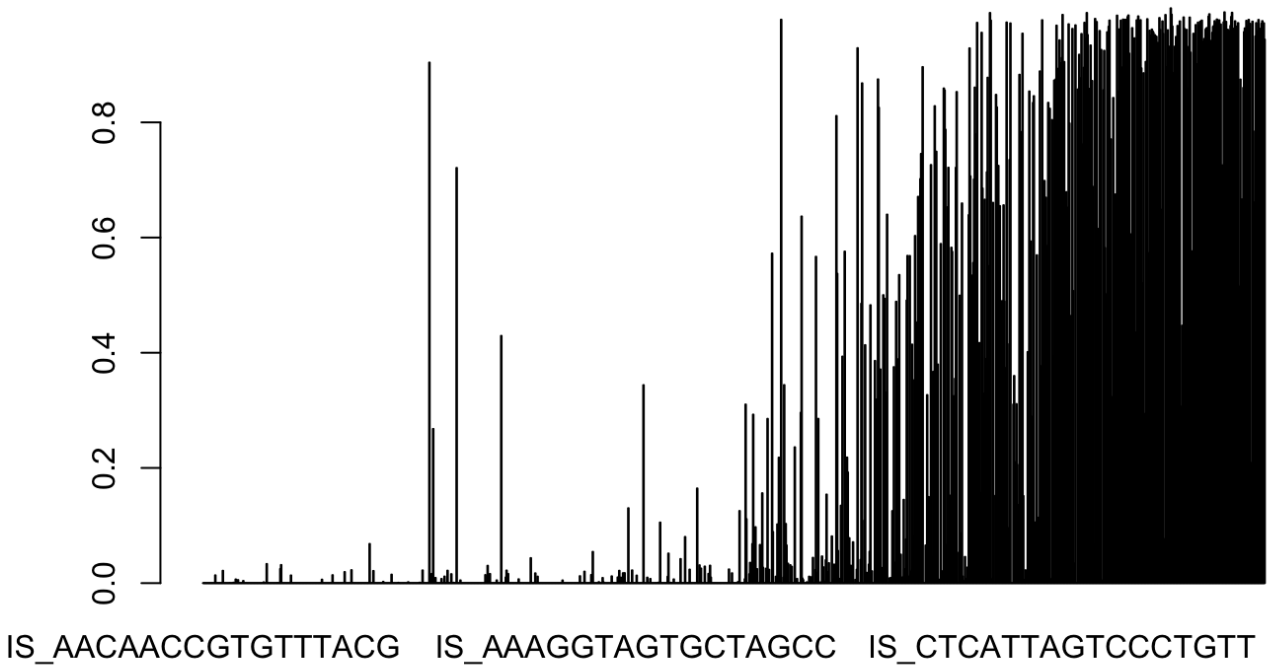
Hide

```
plot(MOL56pred,ExpressionCombosorted$Ptgds)
```



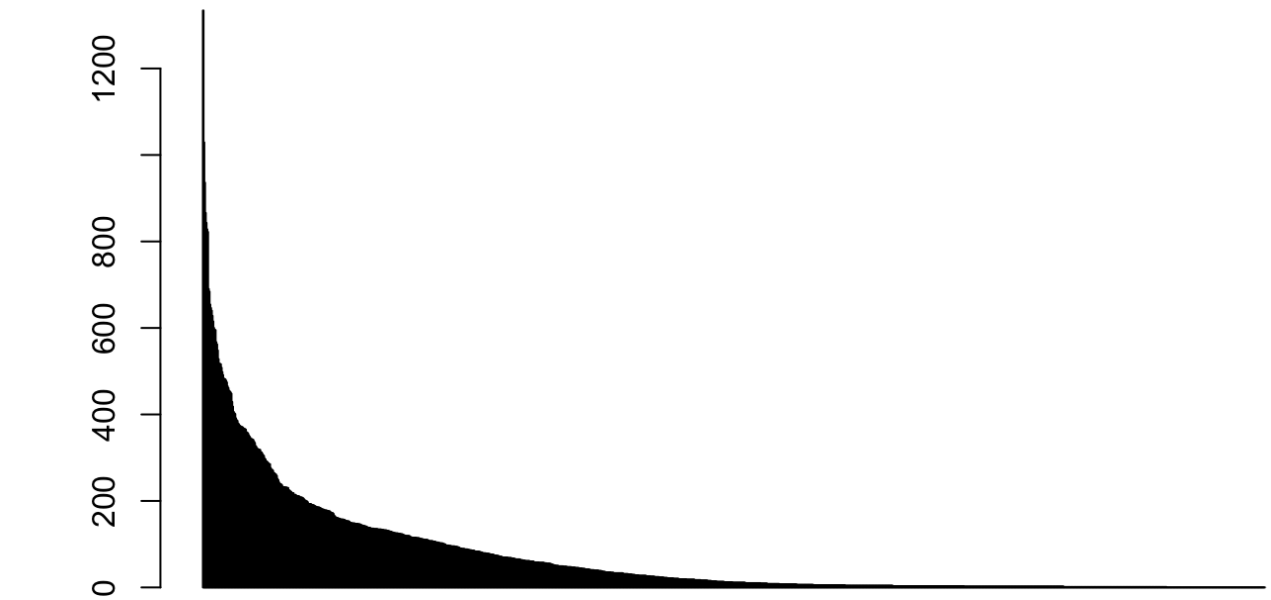
Hide

```
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Klk6,decreasing = FALS
E),]
MOL2pred <- oligos.integratedOL256@meta.data$prediction.score.MOL2
names(MOL2pred) <- row.names(oligos.integratedOL256@meta.data)
barplot(MOL2pred[row.names(ExpressionCombosorted)])
```



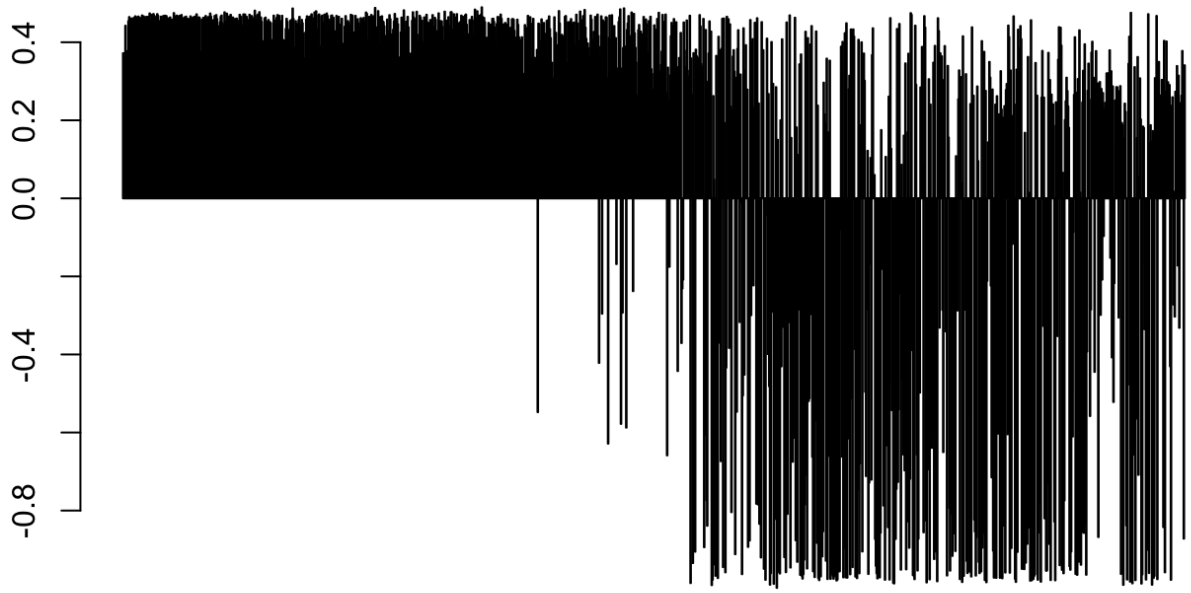
Hide

```
ExpressionCombo <- as.data.frame(t(oligos.integratedOL256@assays$RNA@counts[c("Ptgds"
,"Klk6"),]))
pred <- (MOL56pred)-(MOL2pred)
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Ptgds,decreasing = TRU
E),]
barplot(ExpressionCombosorted$Ptgds)
```



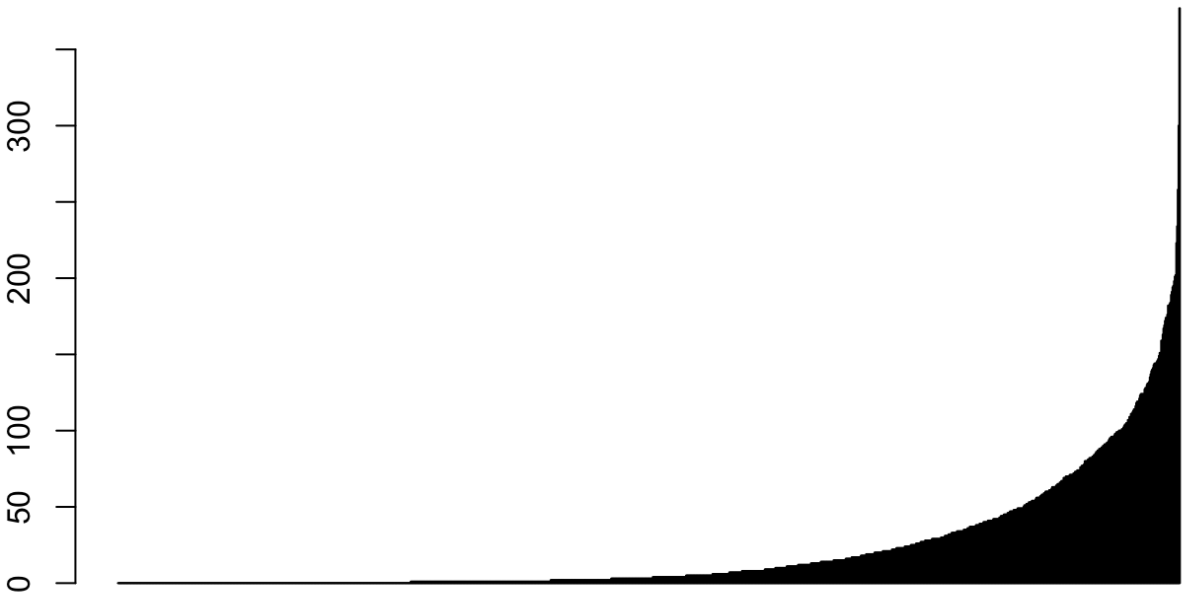
Hide

```
barplot(pred[row.names(ExpressionCombosorted)])
```



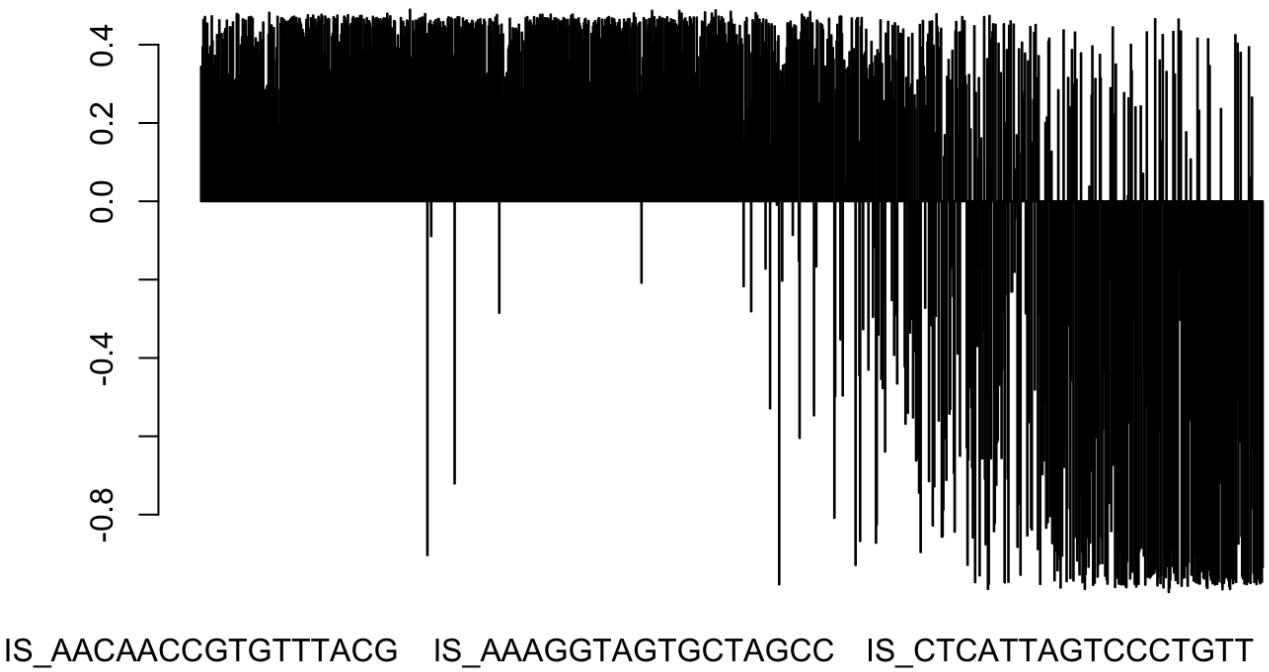
Hide

```
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Klk6,decreasing = FALS
E),]
barplot(ExpressionCombosorted$Klk6)
```



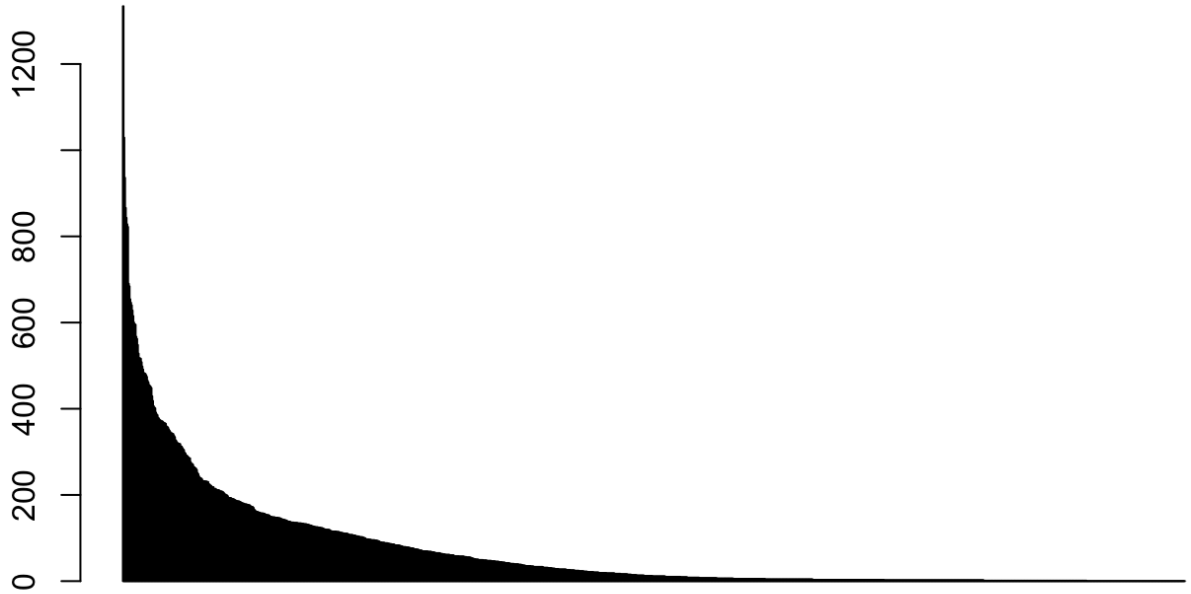
Hide

```
barplot(pred[row.names(ExpressionCombosorted)])
```



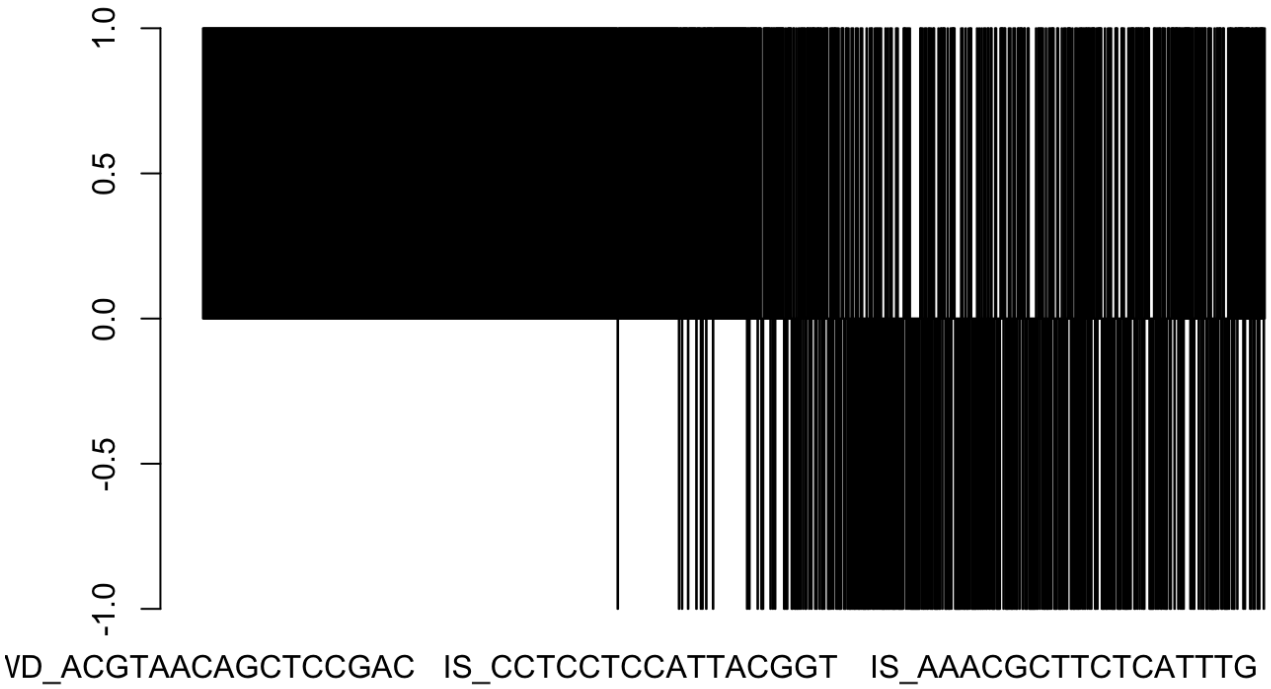
Hide

```
ExpressionCombo <- as.data.frame(t(oligos.integratedOL256@assays$RNA@counts[c("Ptgds"
,"Klk6"),]))
MOL56ID <-  1*oligos.integratedOL256@meta.data$predicted.id %in% c("MOL5","MOL6")
MOL2ID <-  -1*oligos.integratedOL256@meta.data$predicted.id %in% c("MOL2")
MOLID <- MOL56ID+MOL2ID
names(MOLID) <- row.names(oligos.integratedOL256@meta.data)
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Ptgds,decreasing = TRU
E),]
barplot(ExpressionCombosorted$Ptgds)
```



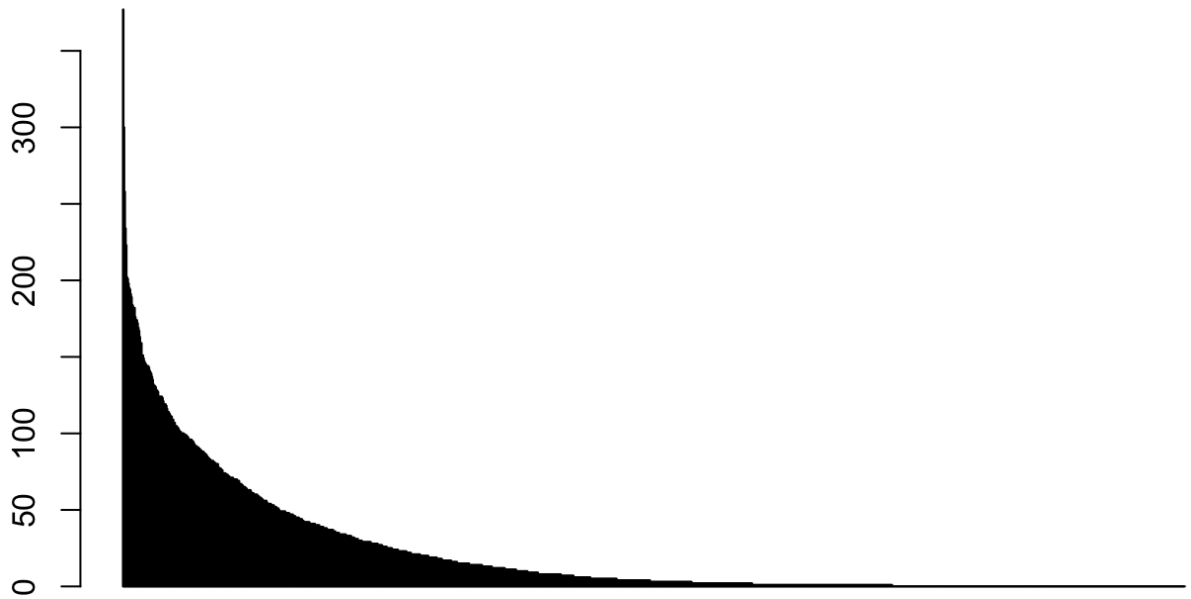
Hide

```
barplot(MOLID[row.names(ExpressionCombosorted)])
```



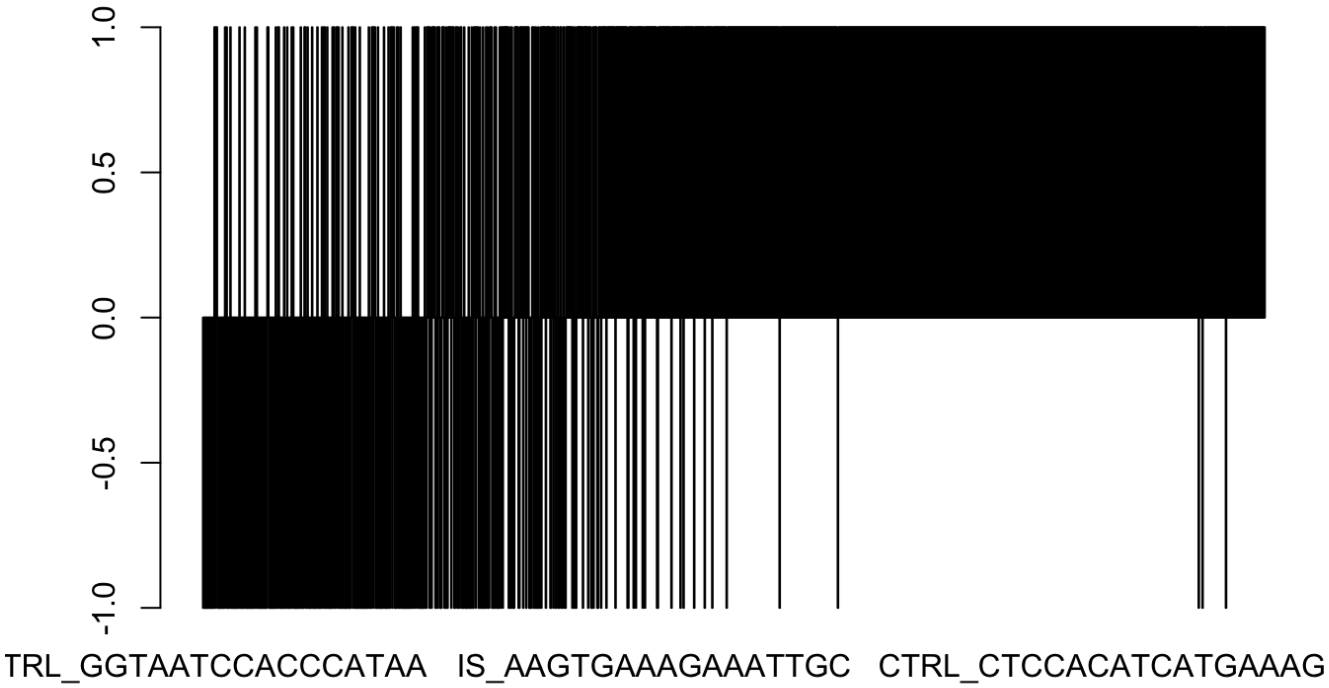
Hide

```
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Klk6,decreasing = TRUE
),]
barplot(ExpressionCombosorted$Klk6)
```



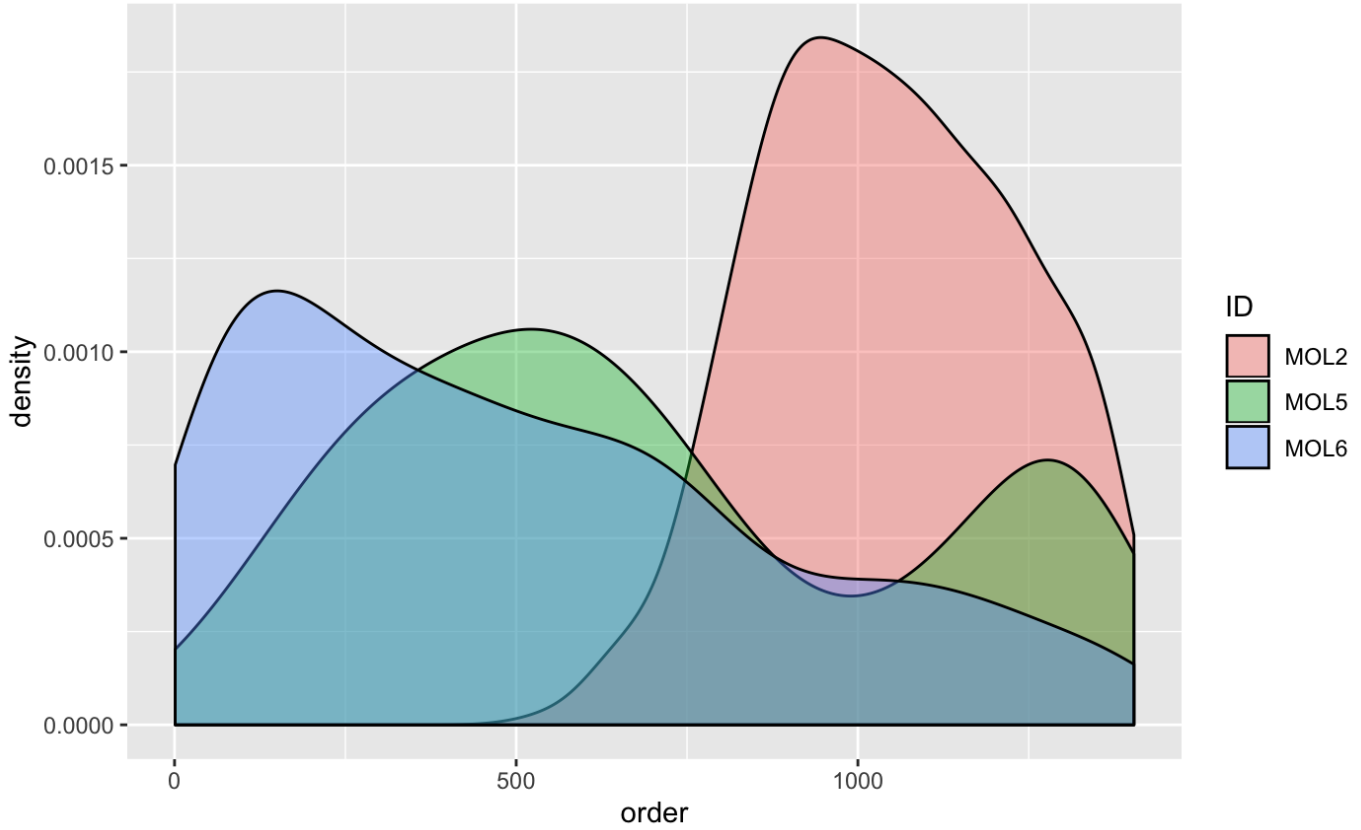
Hide

```
barplot(MOLID[row.names(ExpressionCombosorted)])
```



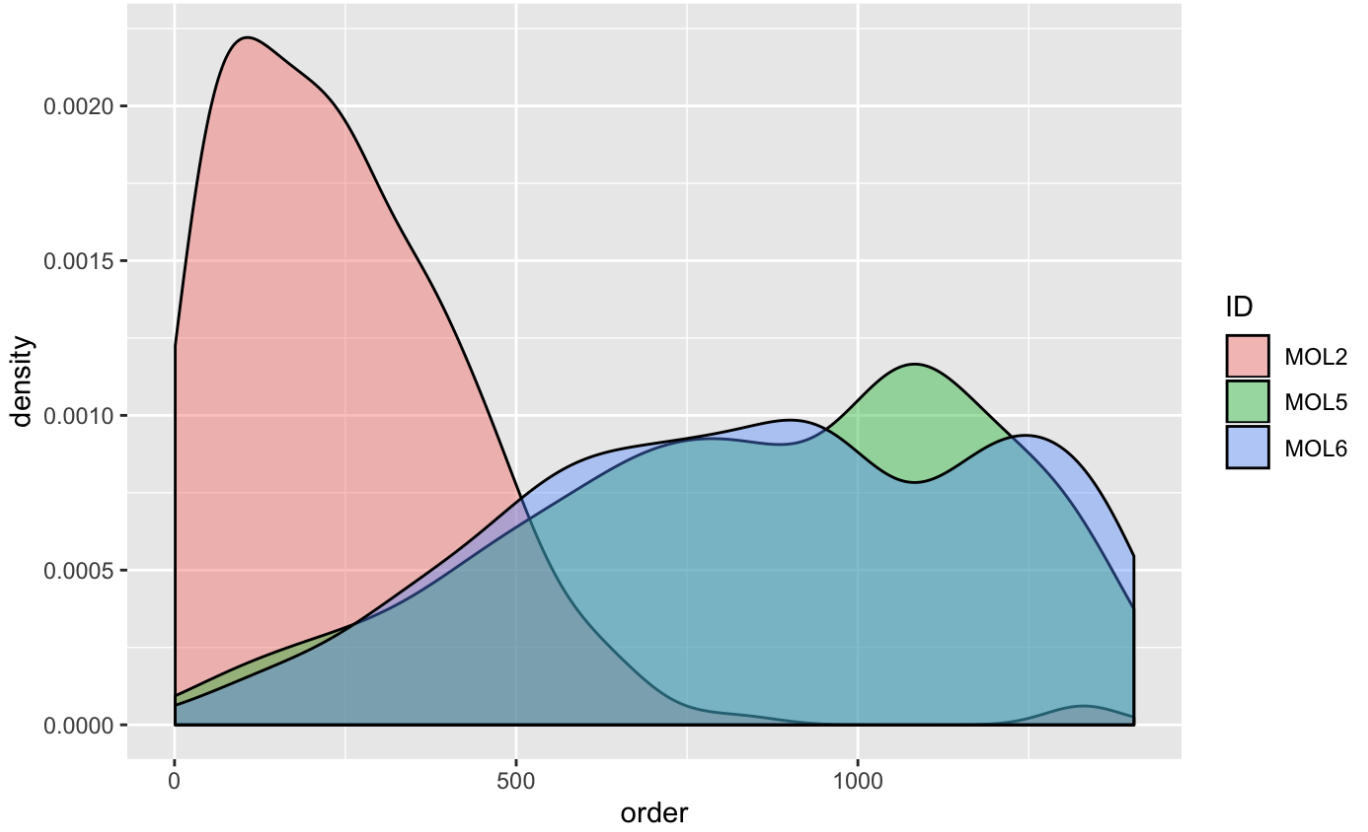
Hide

```
ExpressionCombo <- as.data.frame(t(oligos.integratedOL256@assays$RNA@counts[c("Ptgds"
,"Klk6"),]))
MOLID <- droplevels(oligos.integratedOL256@meta.data$predicted.id)
names(MOLID) <- row.names(oligos.integratedOL256@meta.data)
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Ptgds,decreasing = TRU
E),]
ExpressionCombosorted$ID <- MOLID[row.names(ExpressionCombosorted)]
ExpressionCombosorted$order <- seq_len(nrow(ExpressionCombosorted))
ggplot(ExpressionCombosorted, aes(x=order, fill=ID)) +
  geom_density(alpha=0.4)
```



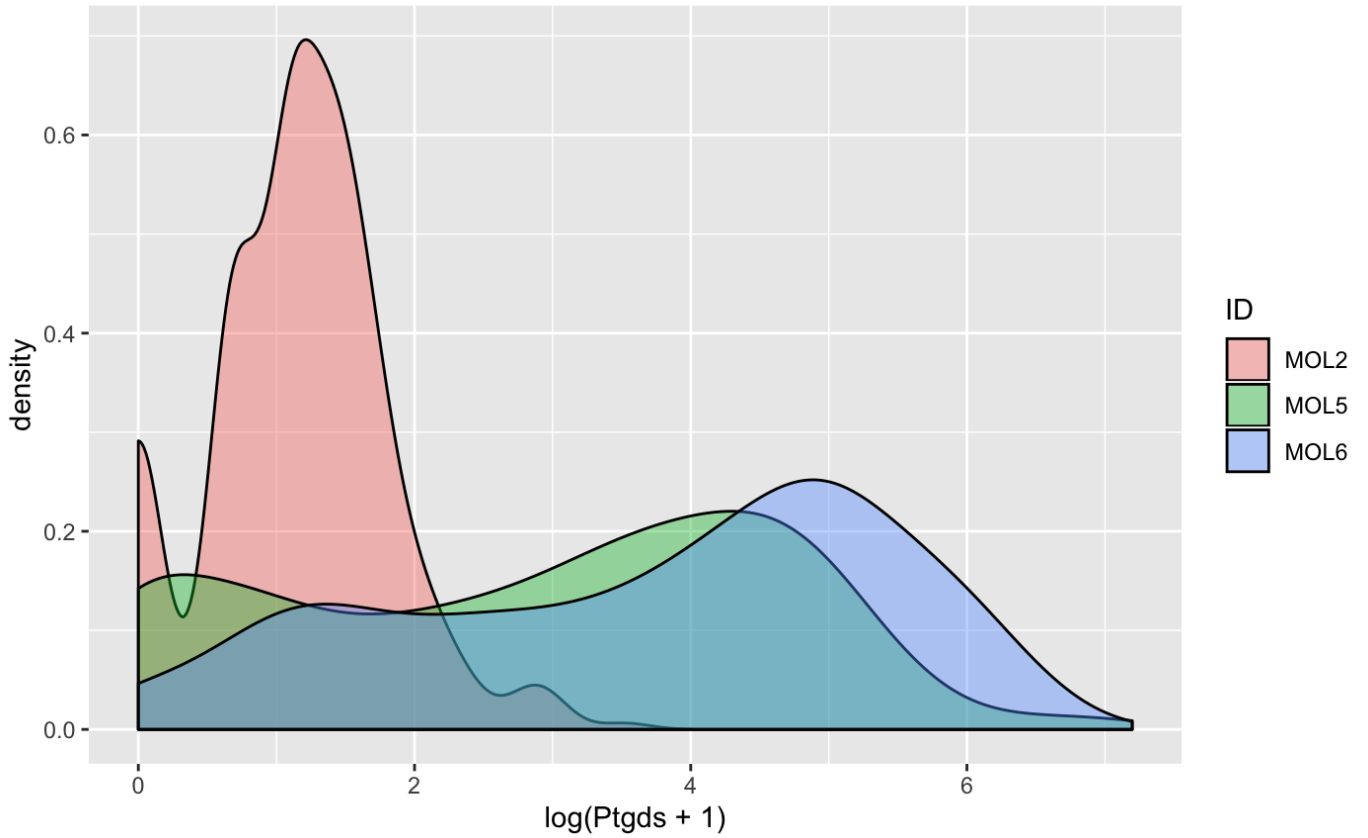
Hide

```
ExpressionCombo <- as.data.frame(t(oligos.integratedOL256@assays$RNA@counts[c("Ptgds"
,"Klk6","S100b","Opalin","Hopx","Apoe","Apod"),]))
MOLID <- droplevels(oligos.integratedOL256@meta.data$predicted.id)
names(MOLID) <- row.names(oligos.integratedOL256@meta.data)
ExpressionCombosorted <- ExpressionCombo[order(ExpressionCombo$Klk6,decreasing = TRUE
),]
ExpressionCombosorted$ID <- MOLID[row.names(ExpressionCombosorted)]
ExpressionCombosorted$order <- seq_len(nrow(ExpressionCombosorted))
ggplot(ExpressionCombosorted, aes(x=order, fill=ID)) +
  geom_density(alpha=0.4)
```



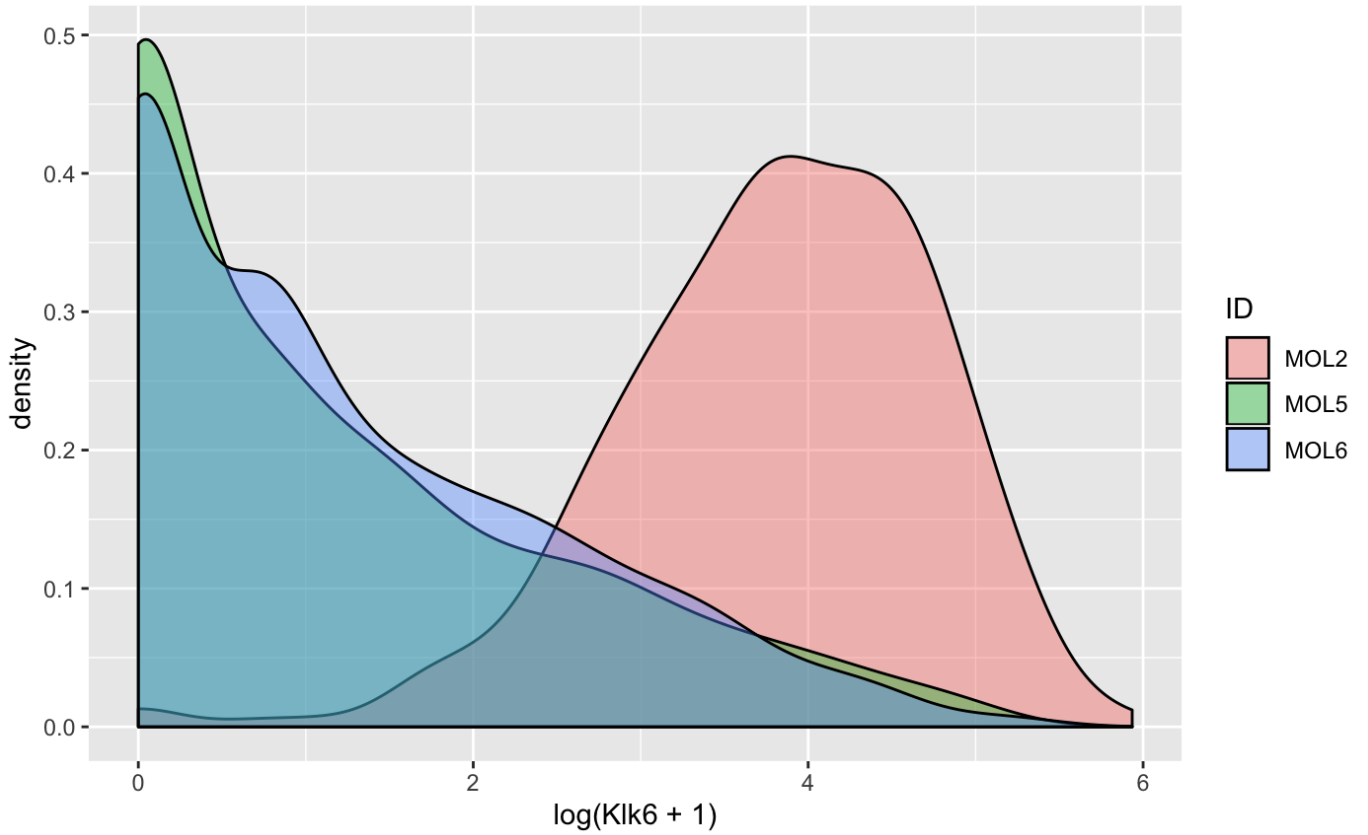
Hide

```
ggplot(ExpressionCombosorted, aes(x=log(Ptgds+1), fill=ID)) +
  geom_density(alpha=0.4)
```



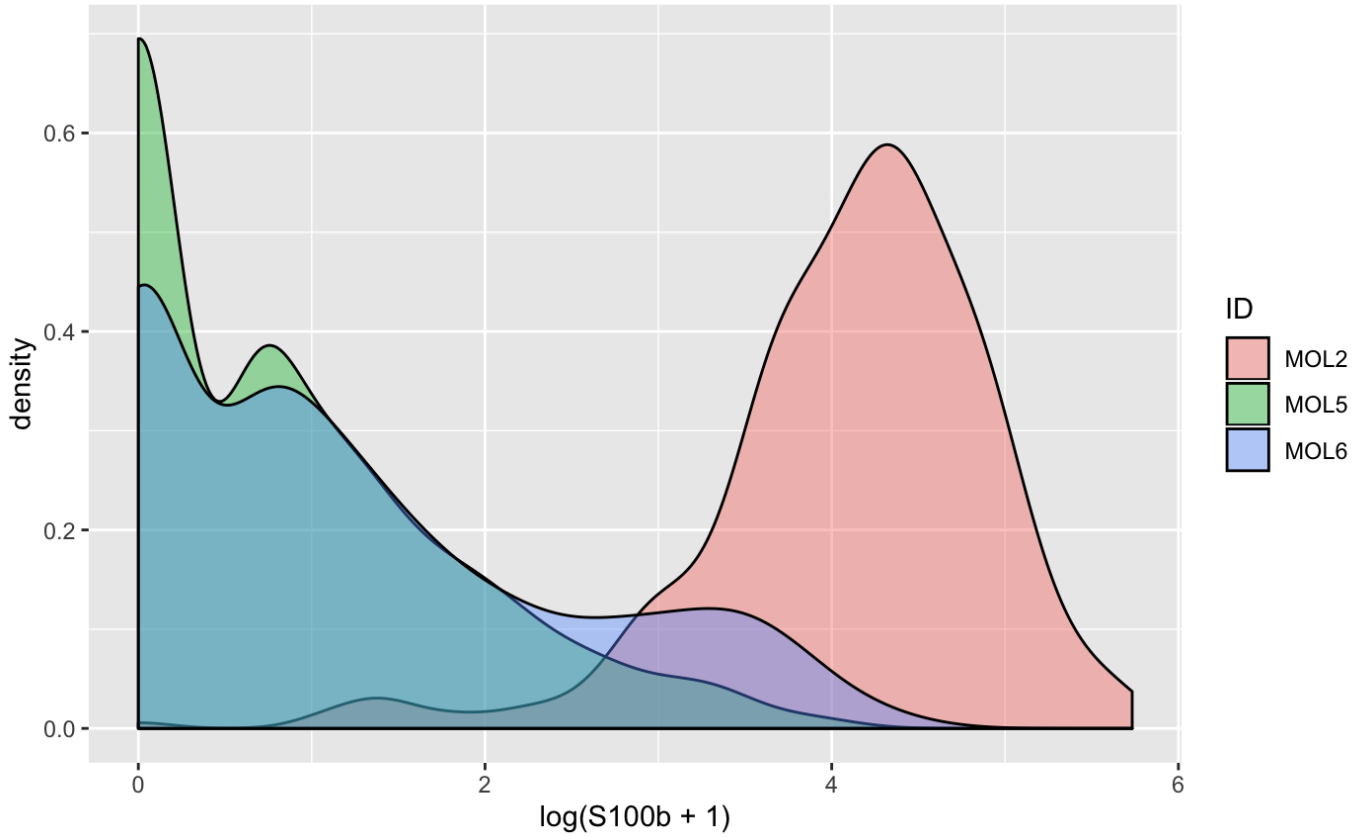
Hide

```
ggplot(ExpressionCombosorted, aes(x=log(Klk6+1), fill=ID)) +
  geom_density(alpha=0.4)
```



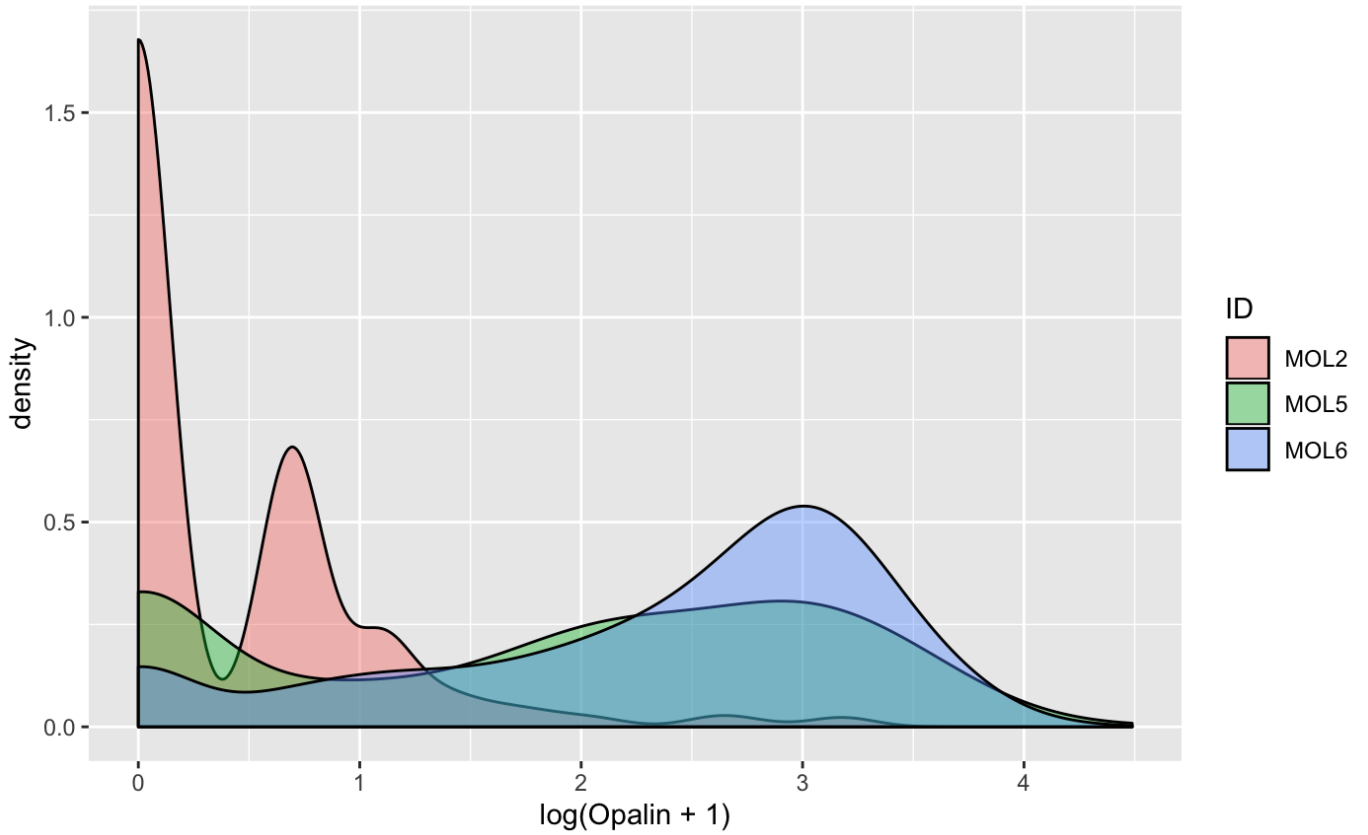
Hide

```
ggplot(ExpressionCombosorted, aes(x=log(S100b+1), fill=ID)) +
  geom_density(alpha=0.4)
```



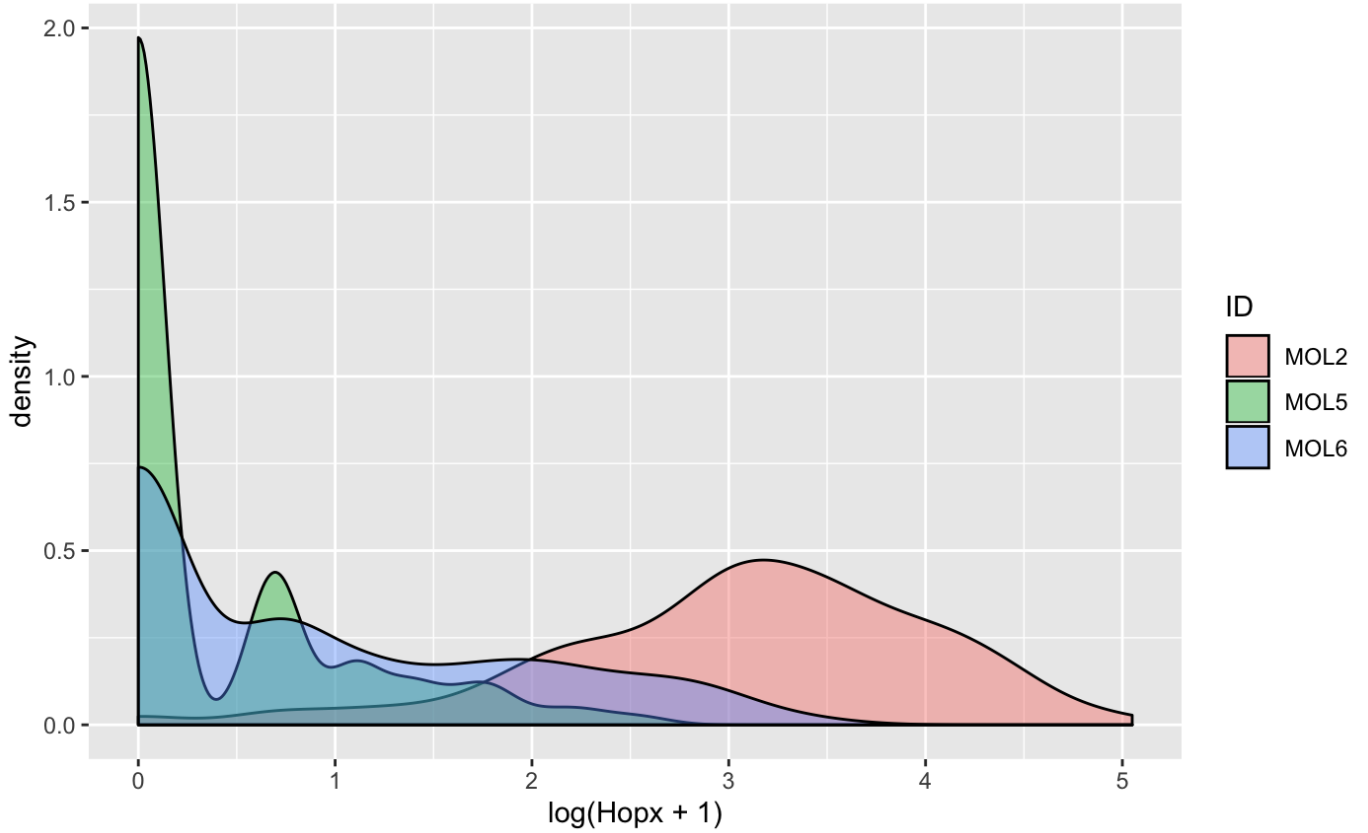
Hide

```
ggplot(ExpressionCombosorted, aes(x=log(Opalin+1), fill=ID)) +
  geom_density(alpha=0.4)
```



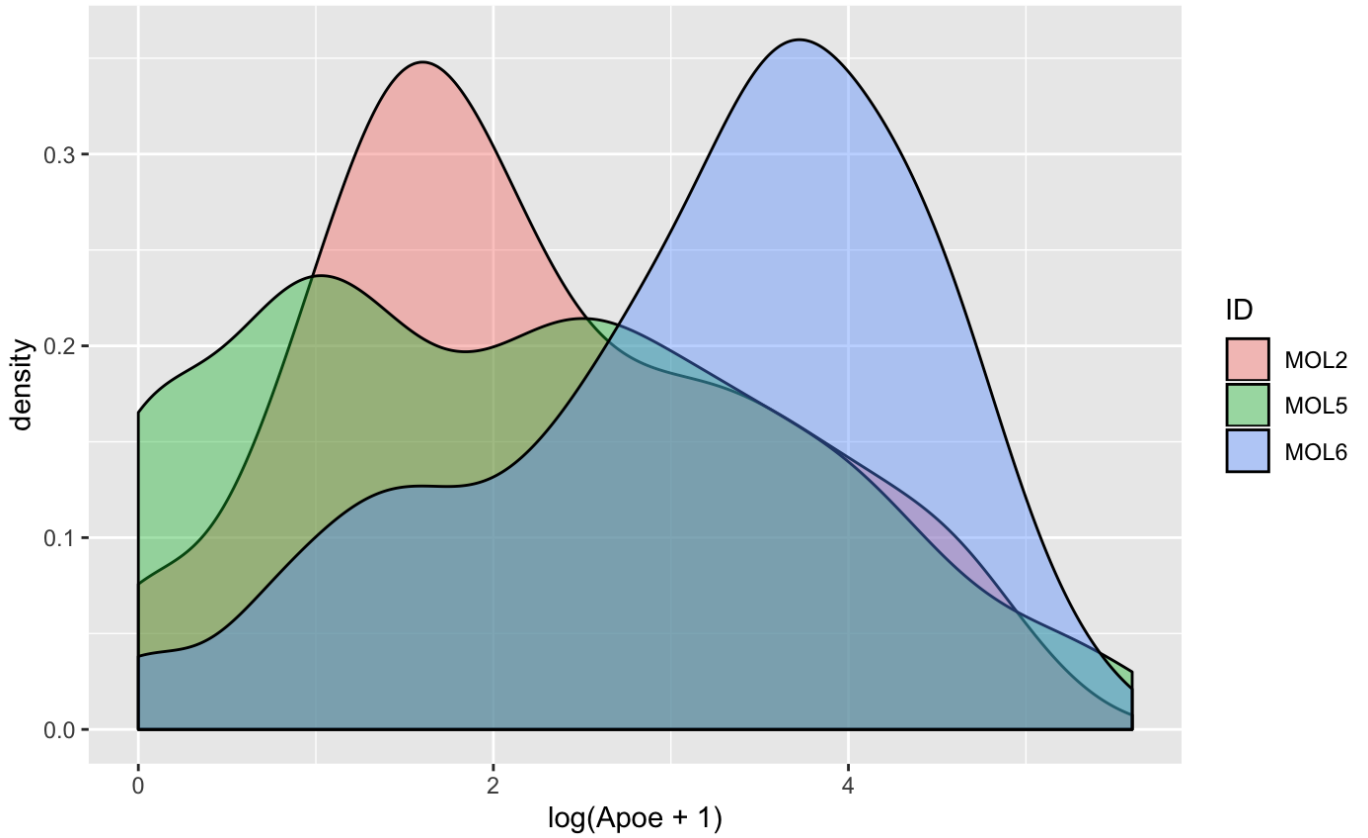
Hide

```
ggplot(ExpressionCombosorted, aes(x=log(Hopx+1), fill=ID)) +
  geom_density(alpha=0.4)
```



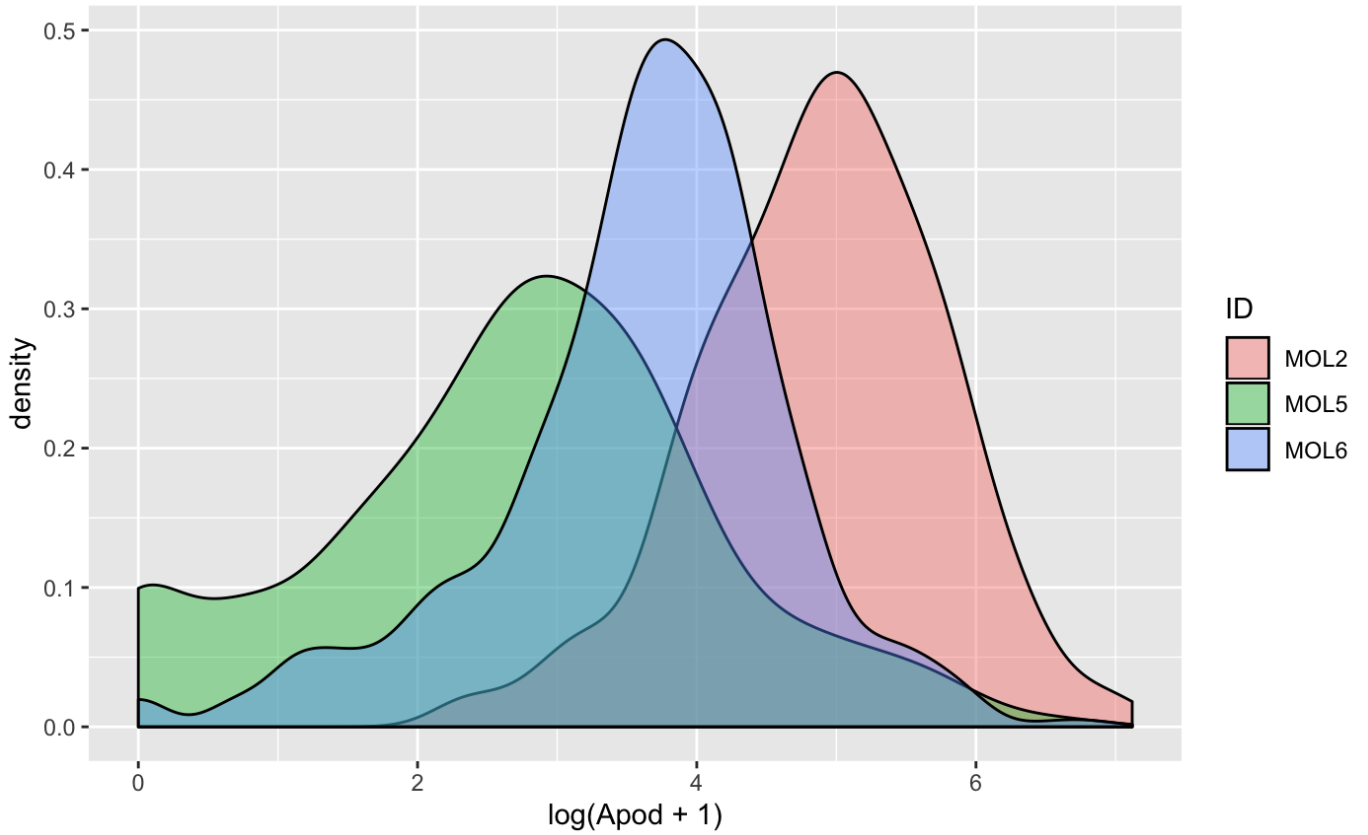
Hide

```
ggplot(ExpressionCombosorted, aes(x=log(Apoe+1), fill=ID)) +
  geom_density(alpha=0.4)
```



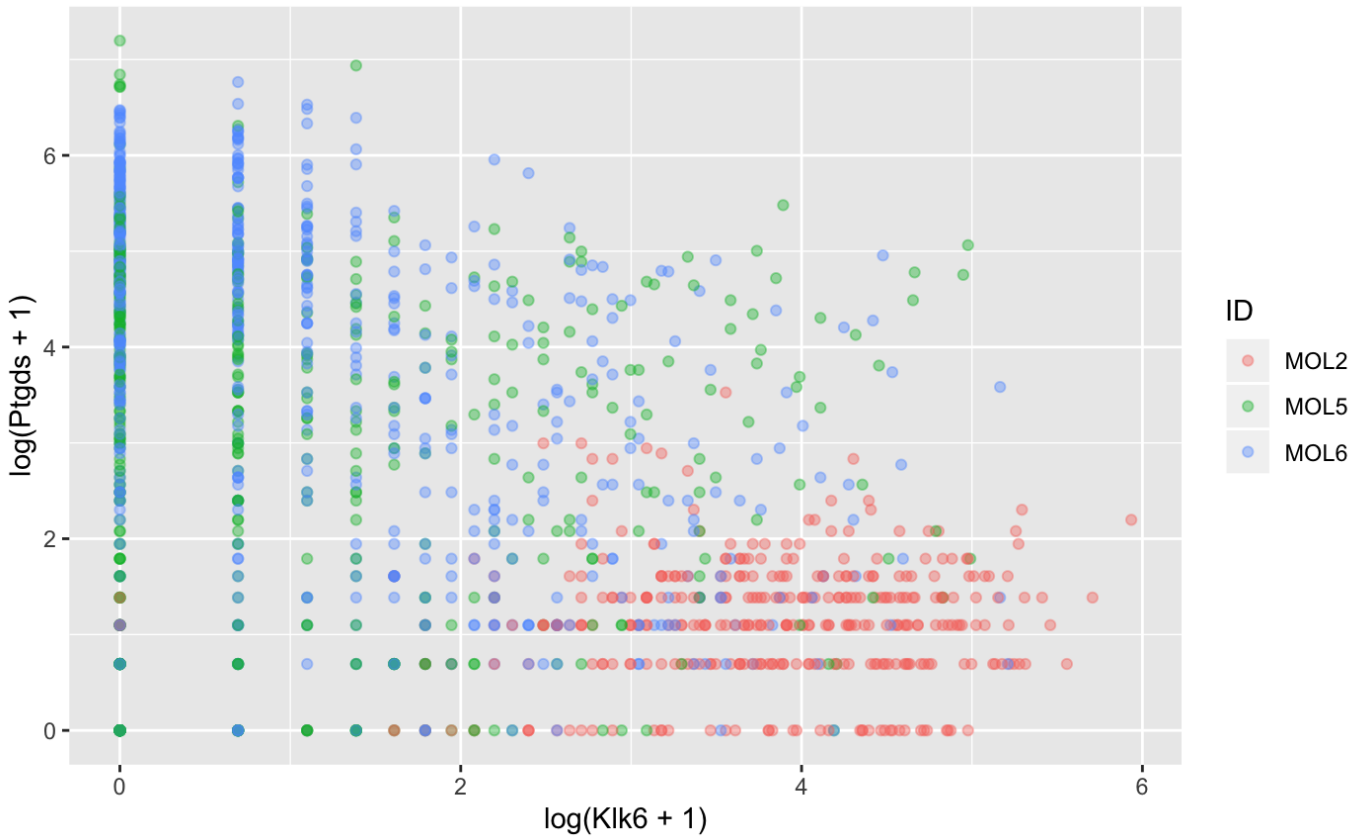
Hide

```
ggplot(ExpressionCombosorted, aes(x=log(Apod+1), fill=ID)) +
  geom_density(alpha=0.4)
```



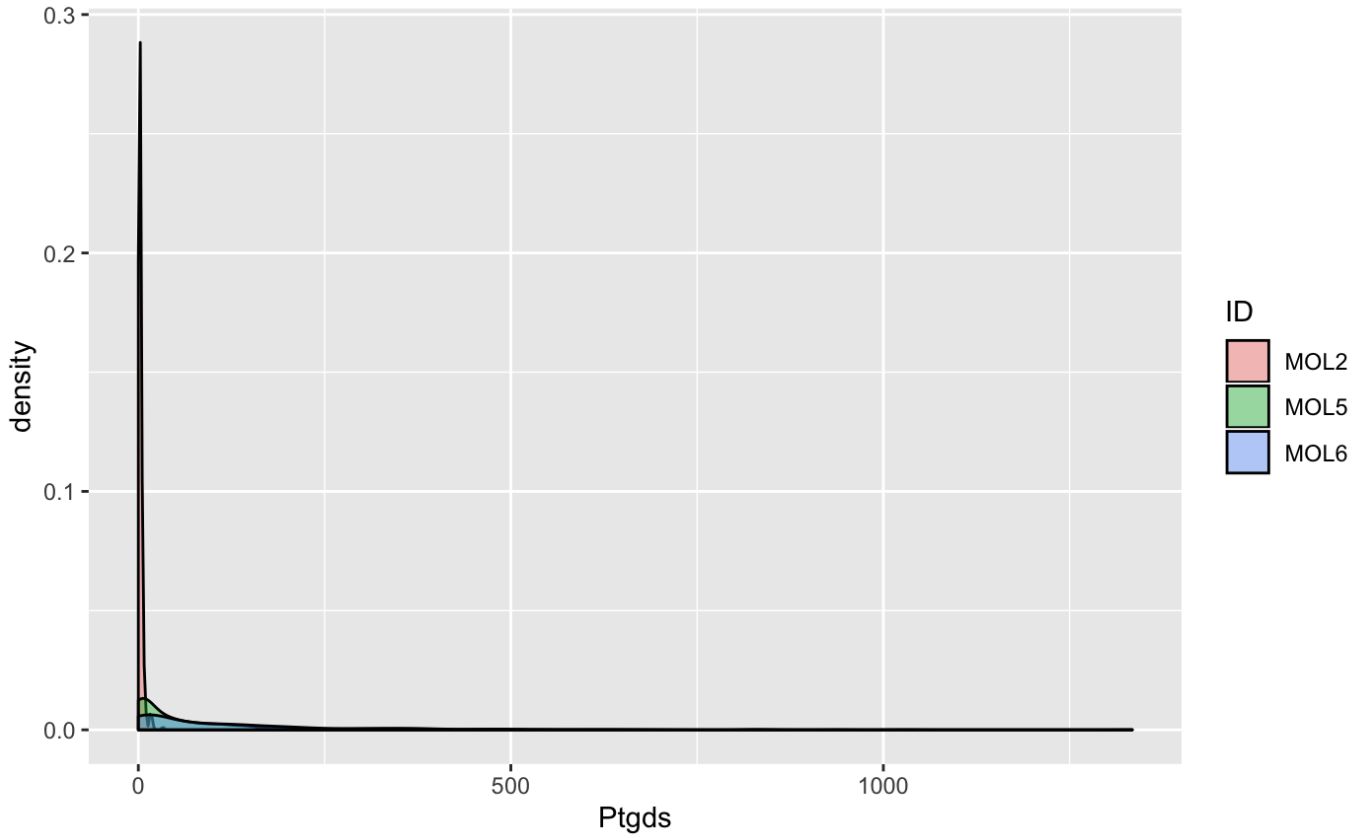
Hide

```
ggplot(ExpressionCombosorted, aes(x=log(Klk6+1),y=log(Ptgds+1), color=ID)) +
  geom_point(alpha=0.4)
```



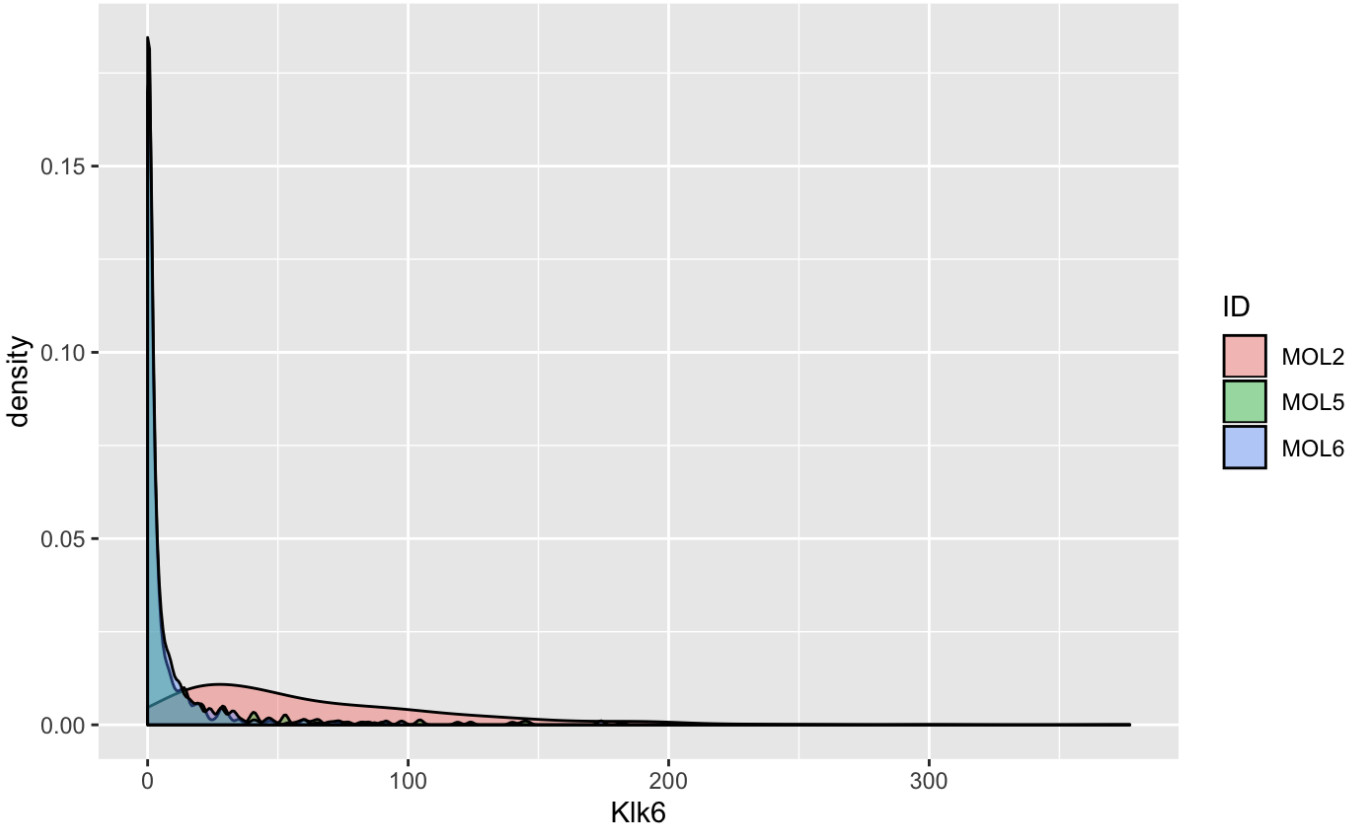
Hide

```
ggplot(ExpressionCombosorted, aes(x=Ptgds, fill=ID)) +
  geom_density(alpha=0.4)
```



Hide

```
ggplot(ExpressionCombosorted, aes(x=Klk6, fill=ID)) +
  geom_density(alpha=0.4)
```


density
0.15
0.10
0.05
0.00
0
100
200
300
Klk6
ID
MOL2
MOL5
MOL6