Mouse analysis on ITGB4:Negative and Positive

yejg

2018/1/26

### Load the packages

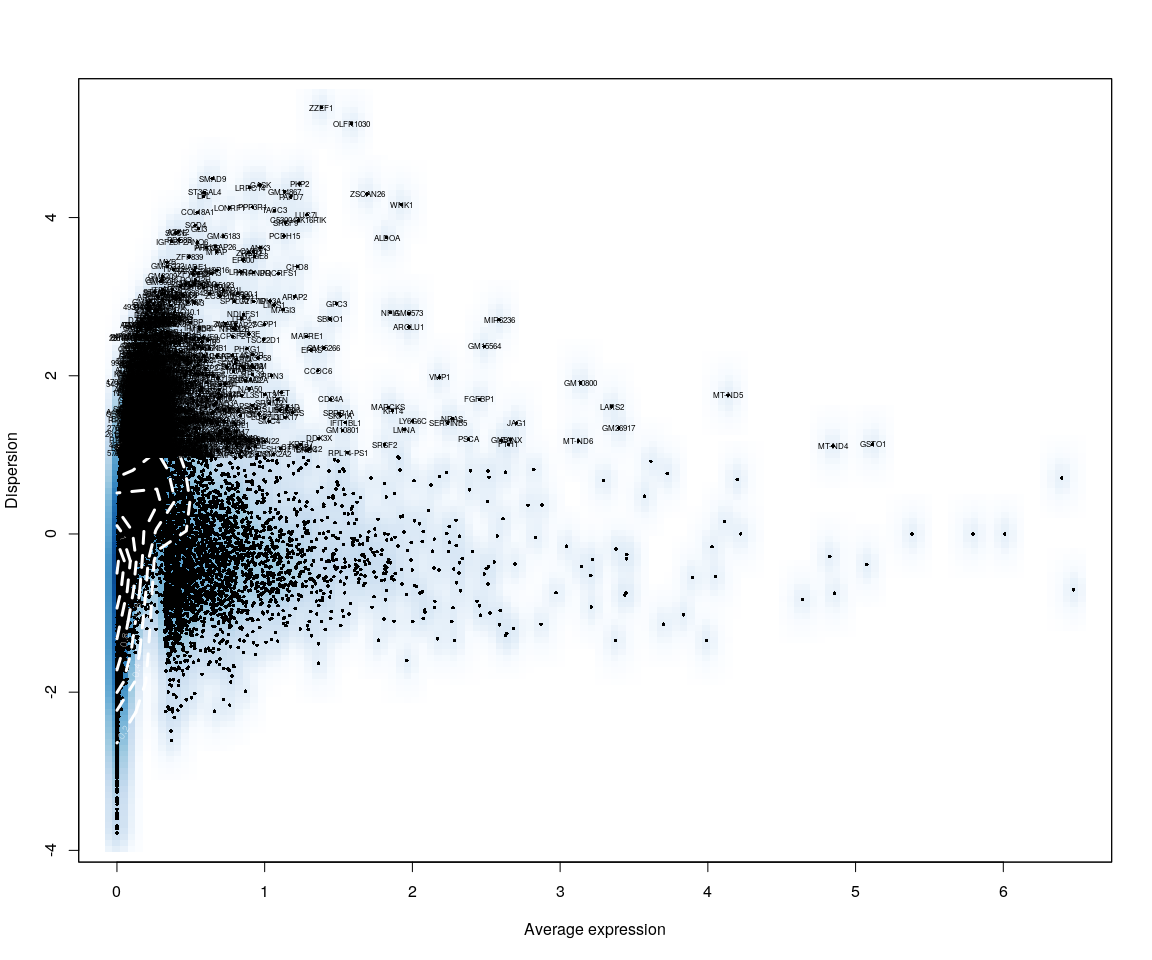
library(Seurat)  
library(data.table)  
library(NMF)  
library(rsvd)  
library(Rtsne)  
library(ggplot2)  
library(cowplot)  
library(sva)  
library(igraph)  
library(cccd)  
library(KernSmooth)  
library(beeswarm)  
library(stringr)  
library(formatR)  
source("../tools.R")  
library(DESeq2)

### Read data

mouse.only.pro <- Load\_data(data\_dir = "../data/mouse.txt")  
rownames(mouse.only.pro) <- unlist(lapply(rownames(mouse.only.pro), str\_to\_upper))  
important.genes <- c("ITGB4", "ABCB5", "KRT19", "ACTB", "KRT12", "KRT5", "GAPDH",   
 "KRT3", "PAX6", "WNT7A", "KRT14", "TRP63", "KRT10")

mouse.all.pbmc <- DESeq\_SeuratObj(X = mouse.only.pro, DESq = FALSE, min.cells = 10,   
 min.genes = 2)

## [1] "Scaling data matrix"  
##   
 |   
 | | 0%  
 |   
 |=================================================================| 100%



all.sample.group <- unlist(lapply(mouse.all.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][1])))  
all.sample.size <- unlist(lapply(mouse.all.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
# reset ident  
mouse.all.pbmc <- SetIdent(mouse.all.pbmc, cells.use = mouse.all.pbmc@cell.names,   
 ident.use = all.sample.size)  
  
mouse.imp.lognorm <- data.frame(FetchData(mouse.all.pbmc, vars.all = important.genes[important.genes %in%   
 rownames(mouse.all.pbmc@raw.data)]))

library(ggplot2)  
library(reshape2)  
  
ITGB4 <- as.numeric(mouse.imp.lognorm[, "ITGB4"])  
Positive.idx <- which(ITGB4 > 0)  
Negative.idx <- which(ITGB4 == 0)  
Positive.data <- mouse.imp.lognorm[Positive.idx, , drop = FALSE]  
Negative.data <- mouse.imp.lognorm[Negative.idx, , drop = FALSE]  
Positive.data <- Positive.data[, -1] # remove ITGB4  
Negative.data <- Negative.data[, -1]

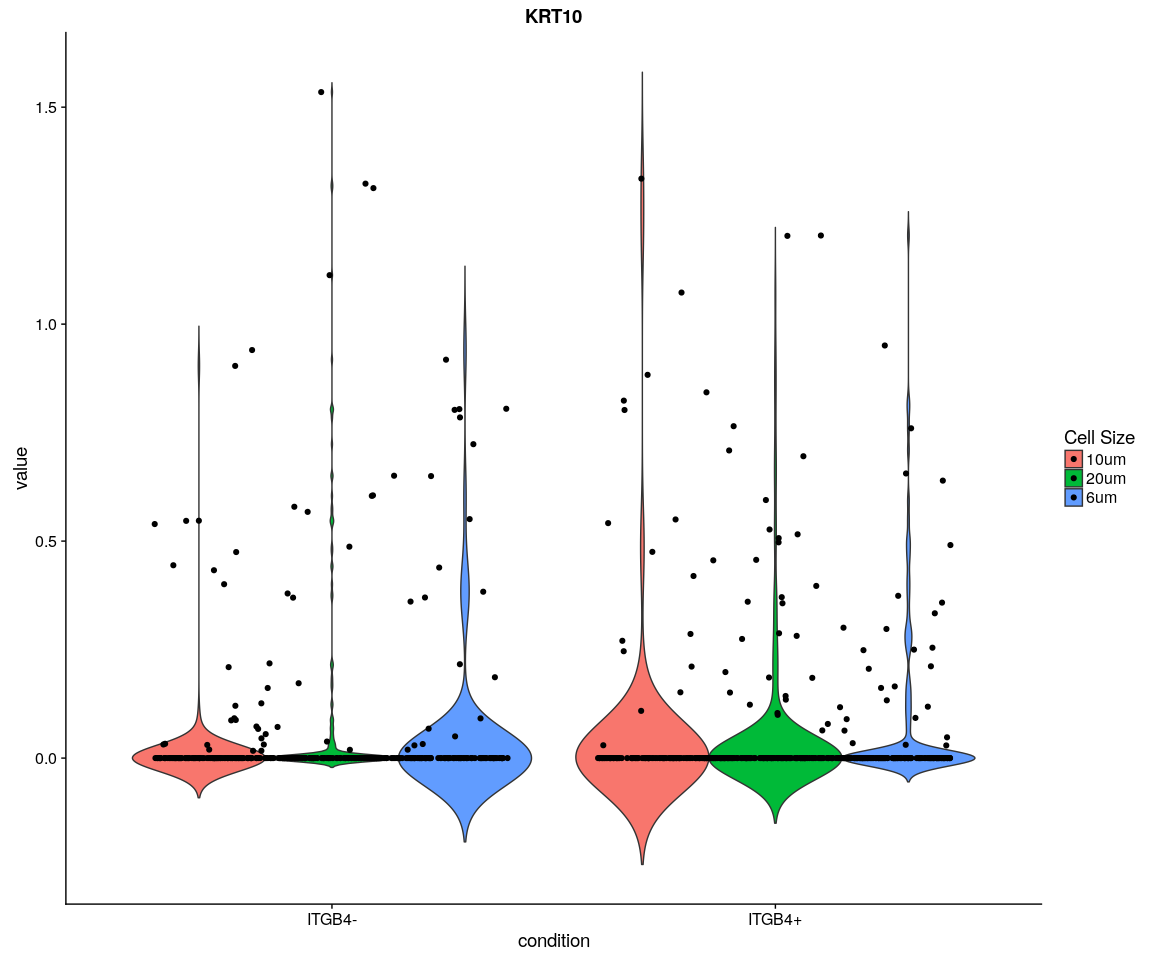
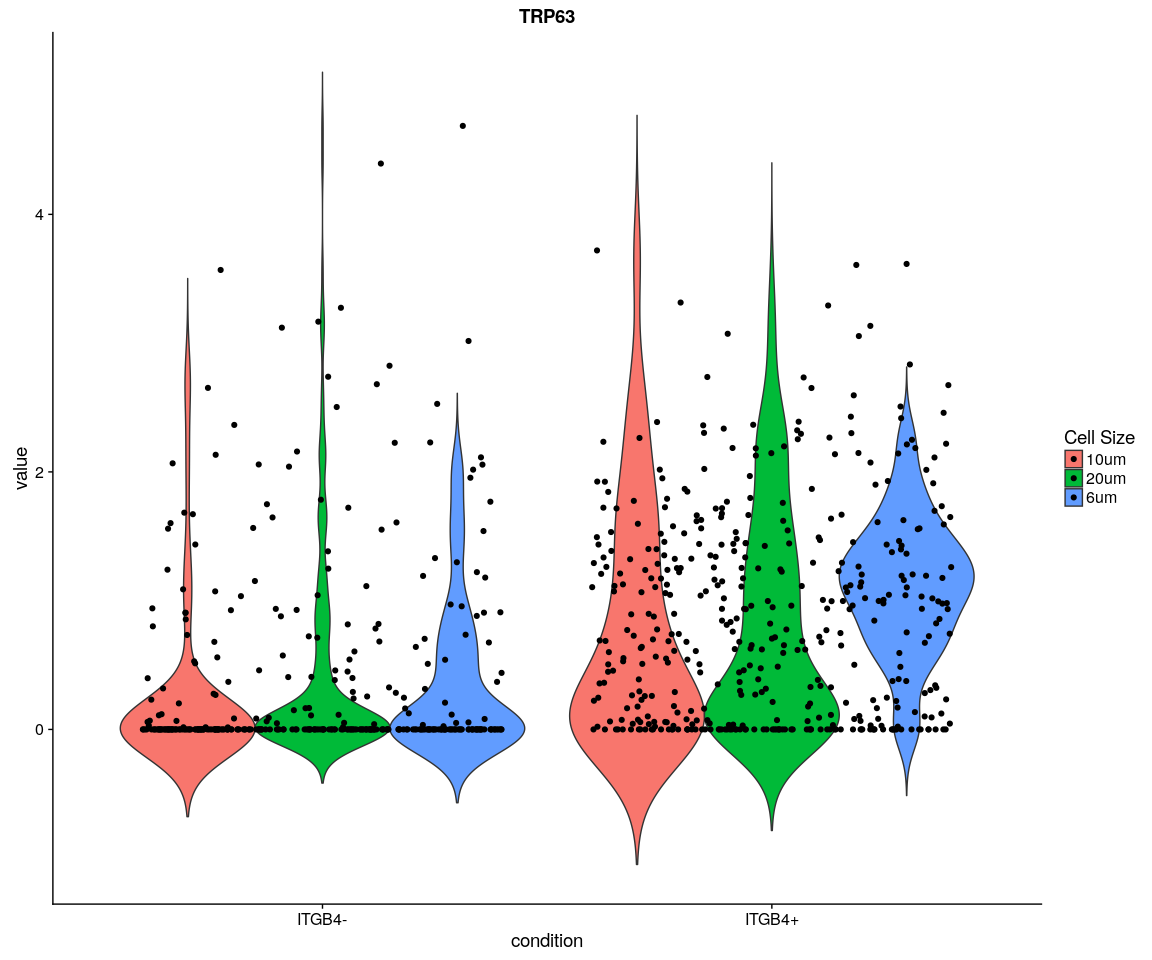
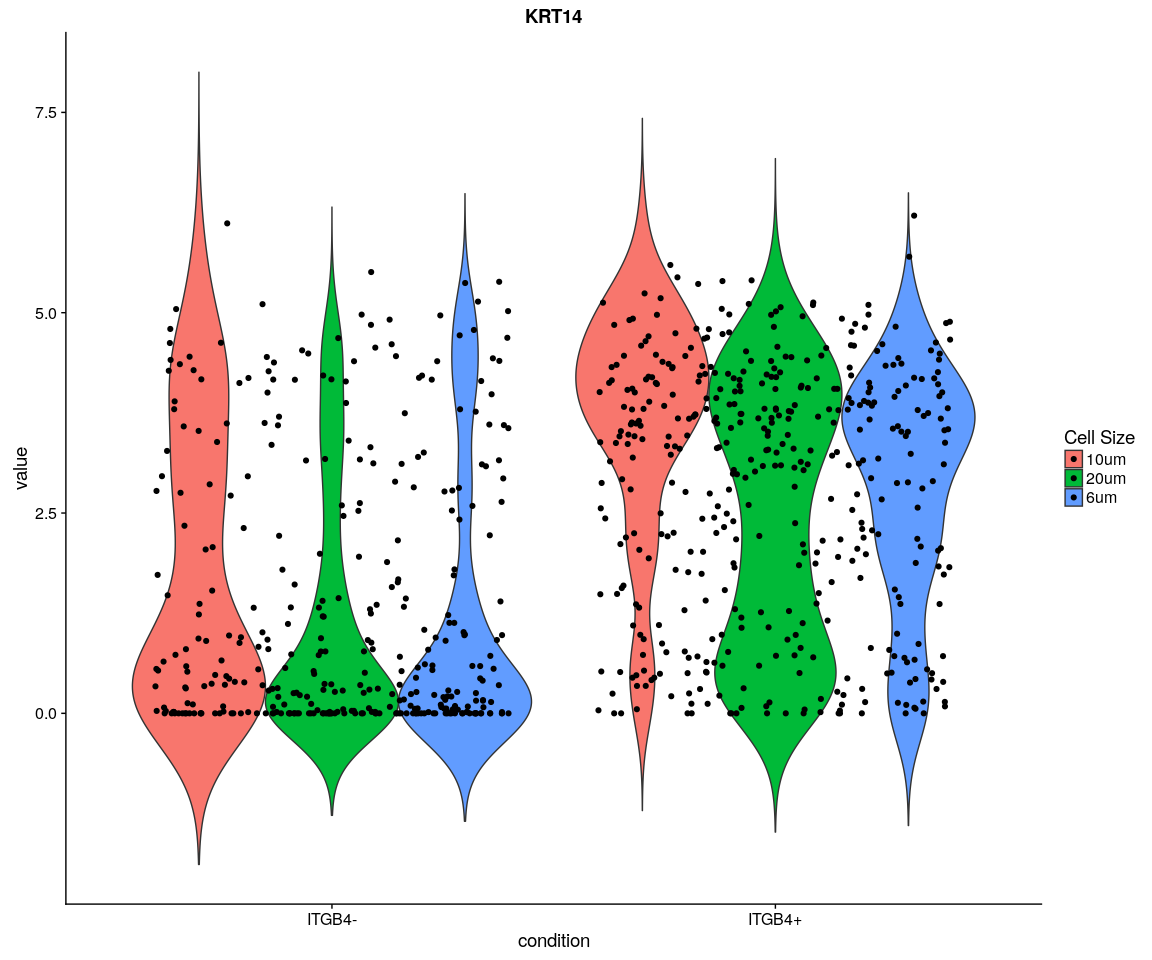
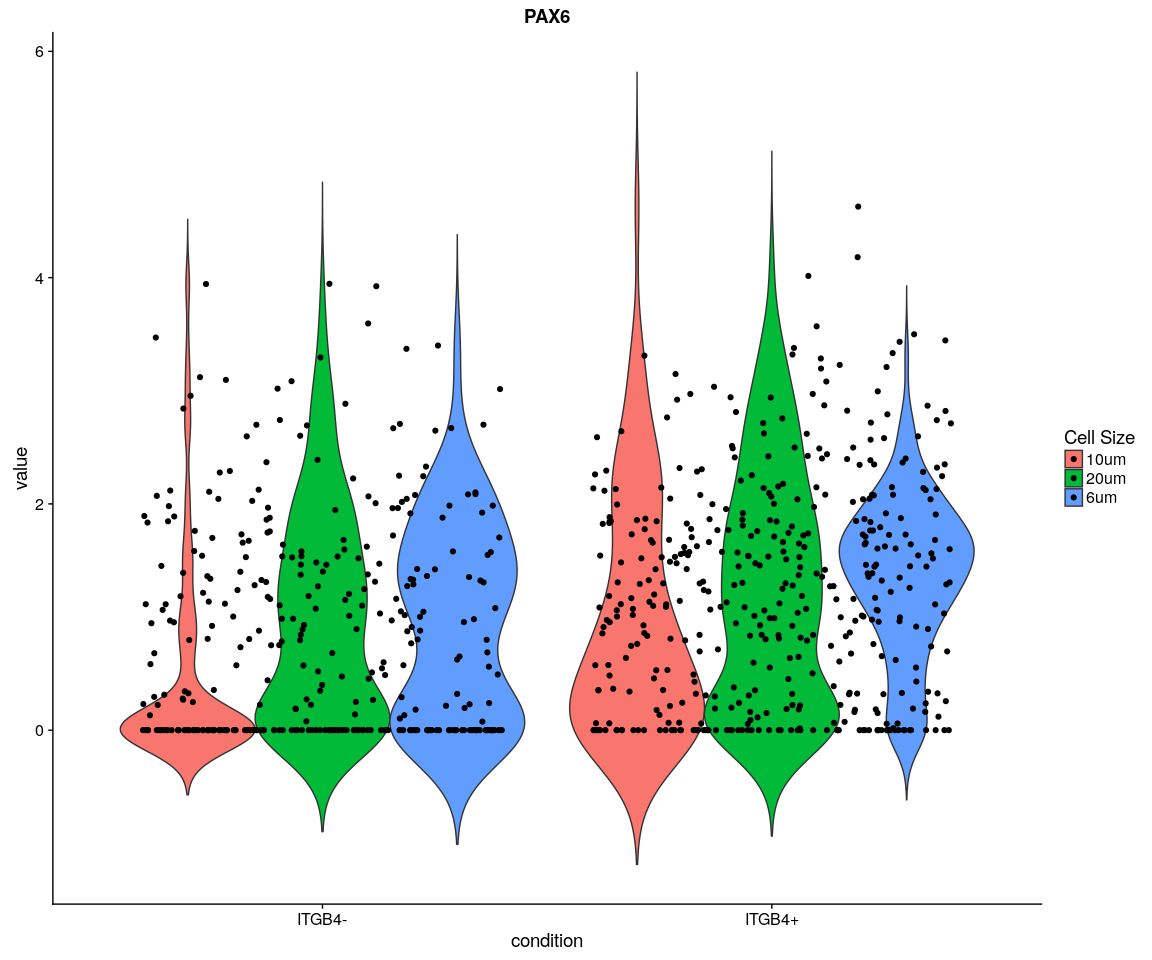
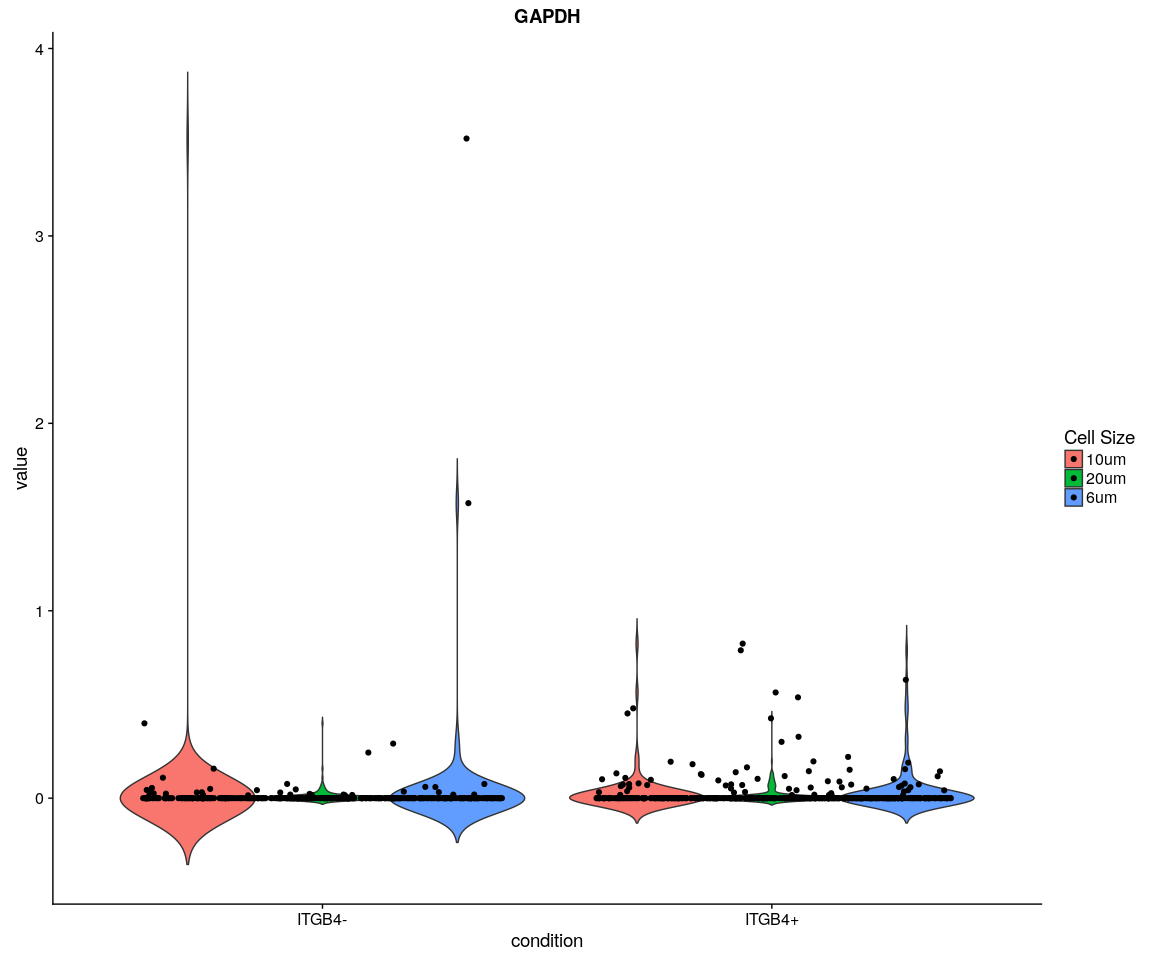
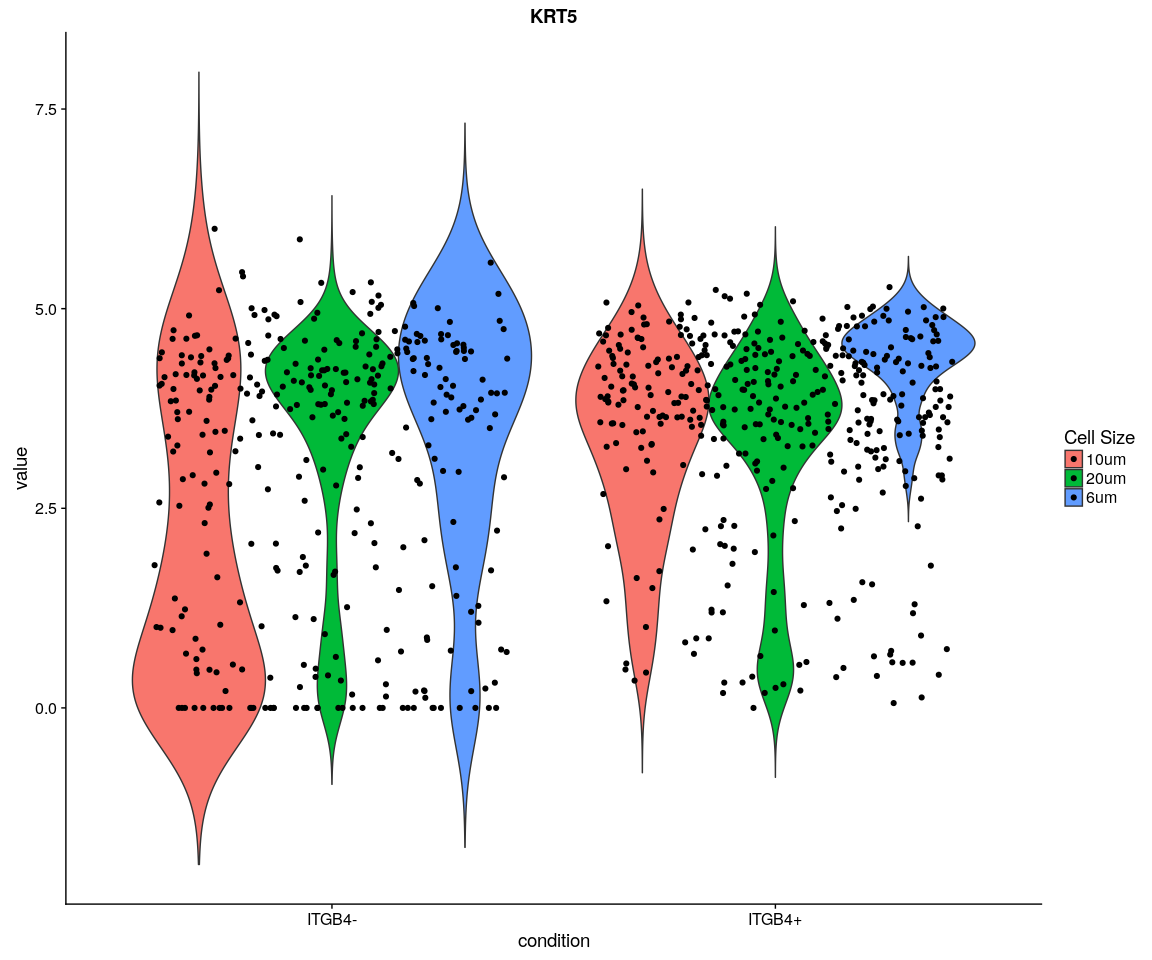
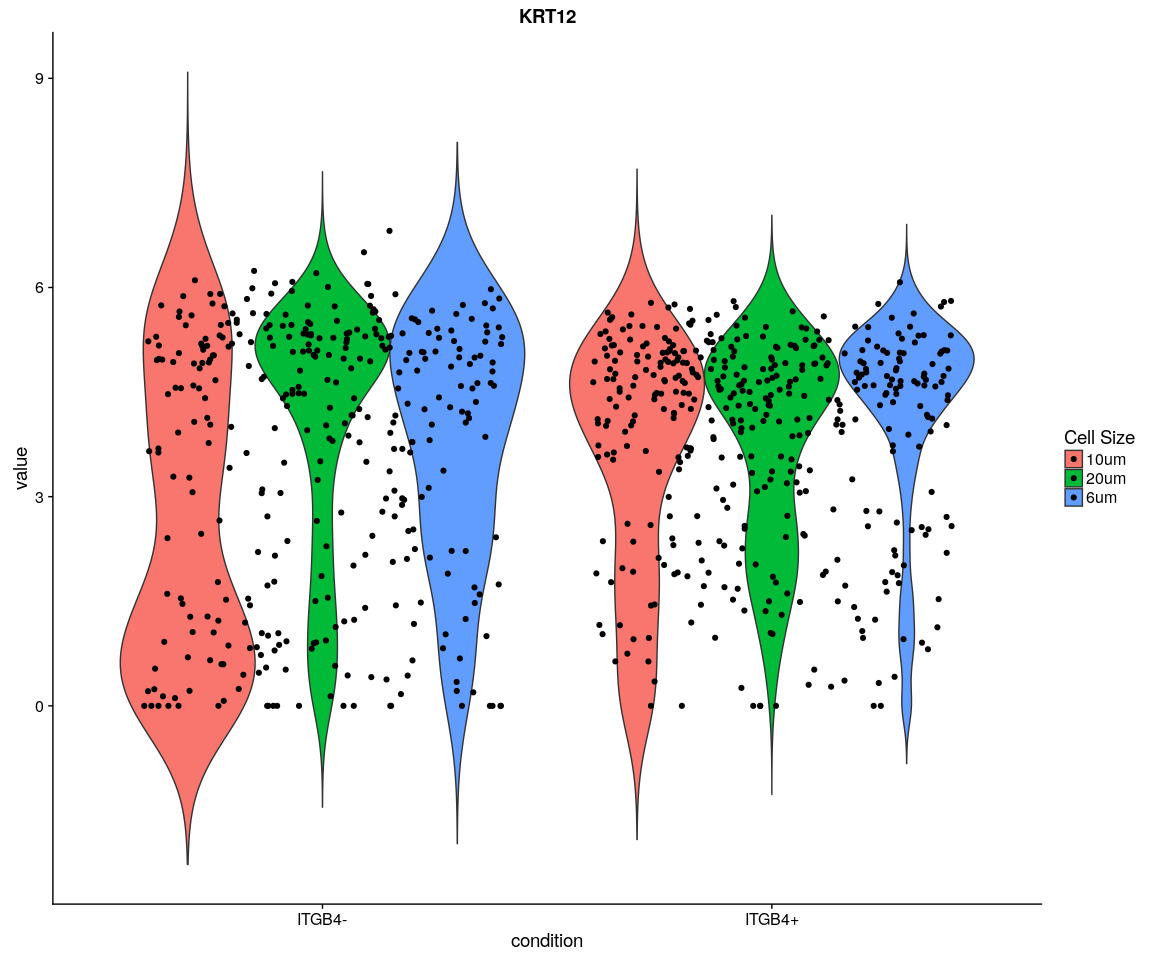
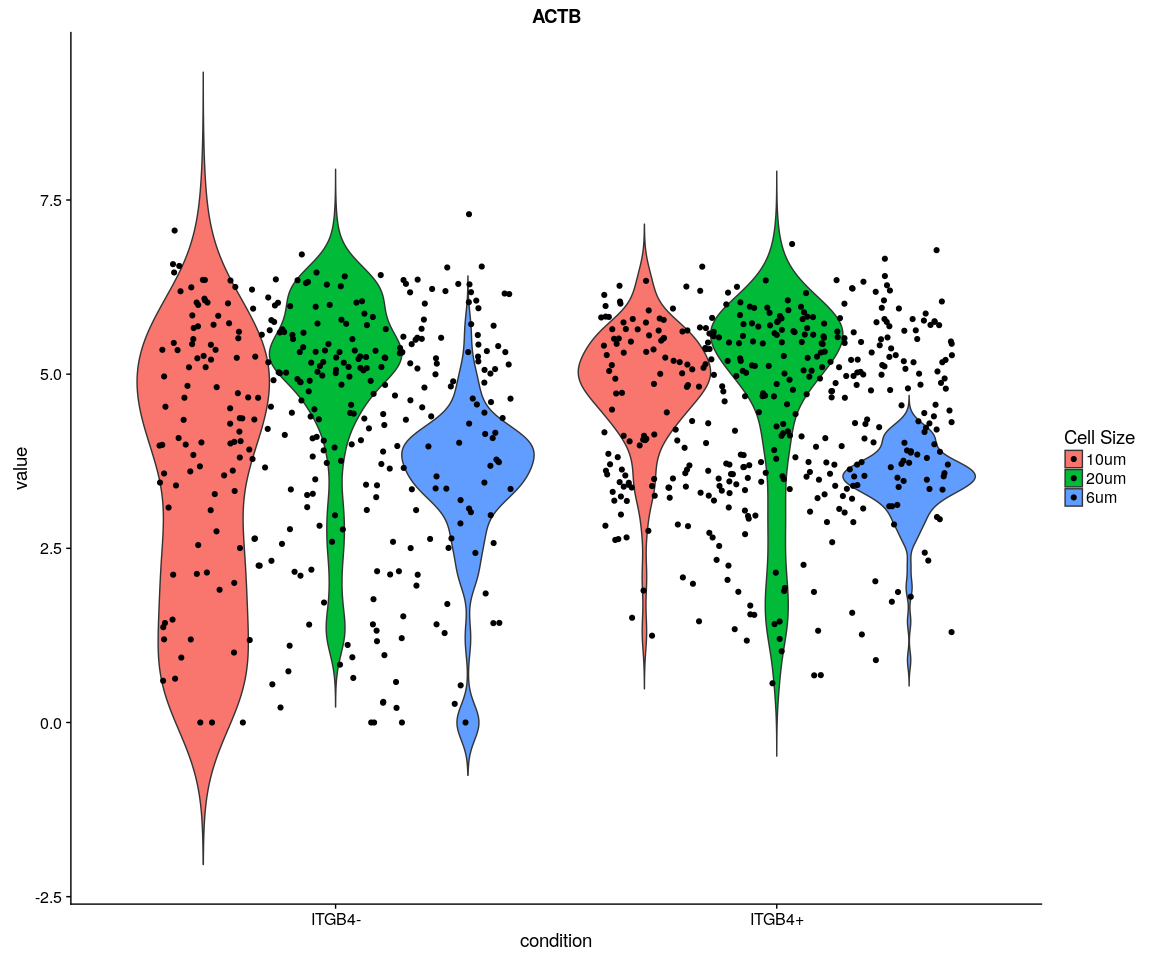
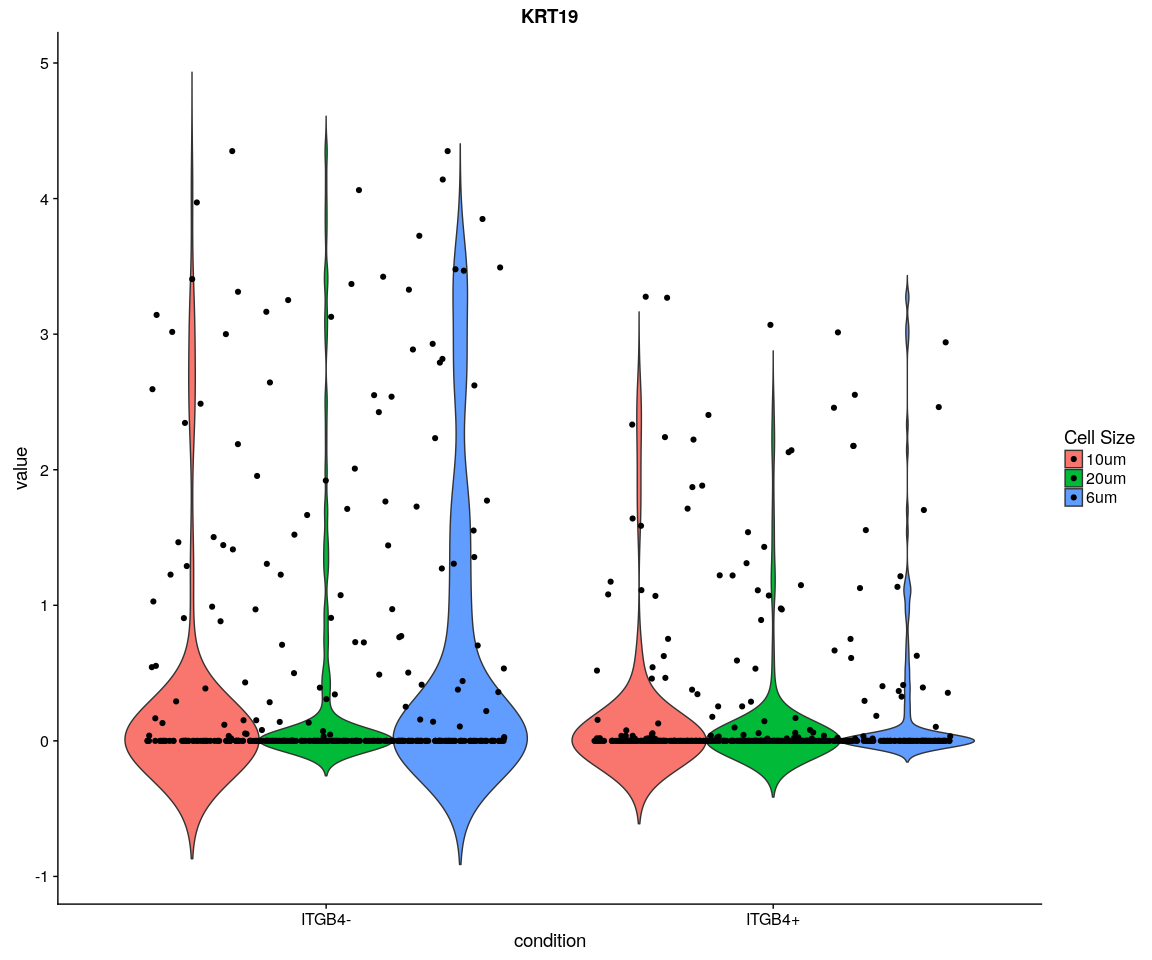
## Figure Explore.1

# Positive.t<-data.frame(as.matrix(LogNormalize(Positive.data,display.progress  
# = FALSE)))  
# Negative.t<-data.frame(as.matrix(LogNormalize(Negative.data,display.progress  
# = FALSE))) Positive.t<-data.frame(t(Positive.t[important.genes,]))  
# Negative.t<-data.frame(t(Negative.t[important.genes,]))  
plot.data <- rbind(Positive.data, Negative.data)  
plot.data$condition <- rep(c("ITGB4+", "ITGB4-"), times = c(dim(Positive.data)[1],   
 dim(Negative.data)[1]))  
cell.size <- c(unlist(lapply(rownames(Positive.data), function(x) return(str\_split(x,   
 "\_")[[1]][2]))), unlist(lapply(rownames(Negative.data), function(x) return(str\_split(x,   
 "\_")[[1]][2]))))  
  
  
plot.data$cell.size <- cell.size  
X <- melt(plot.data)

### Melt the data

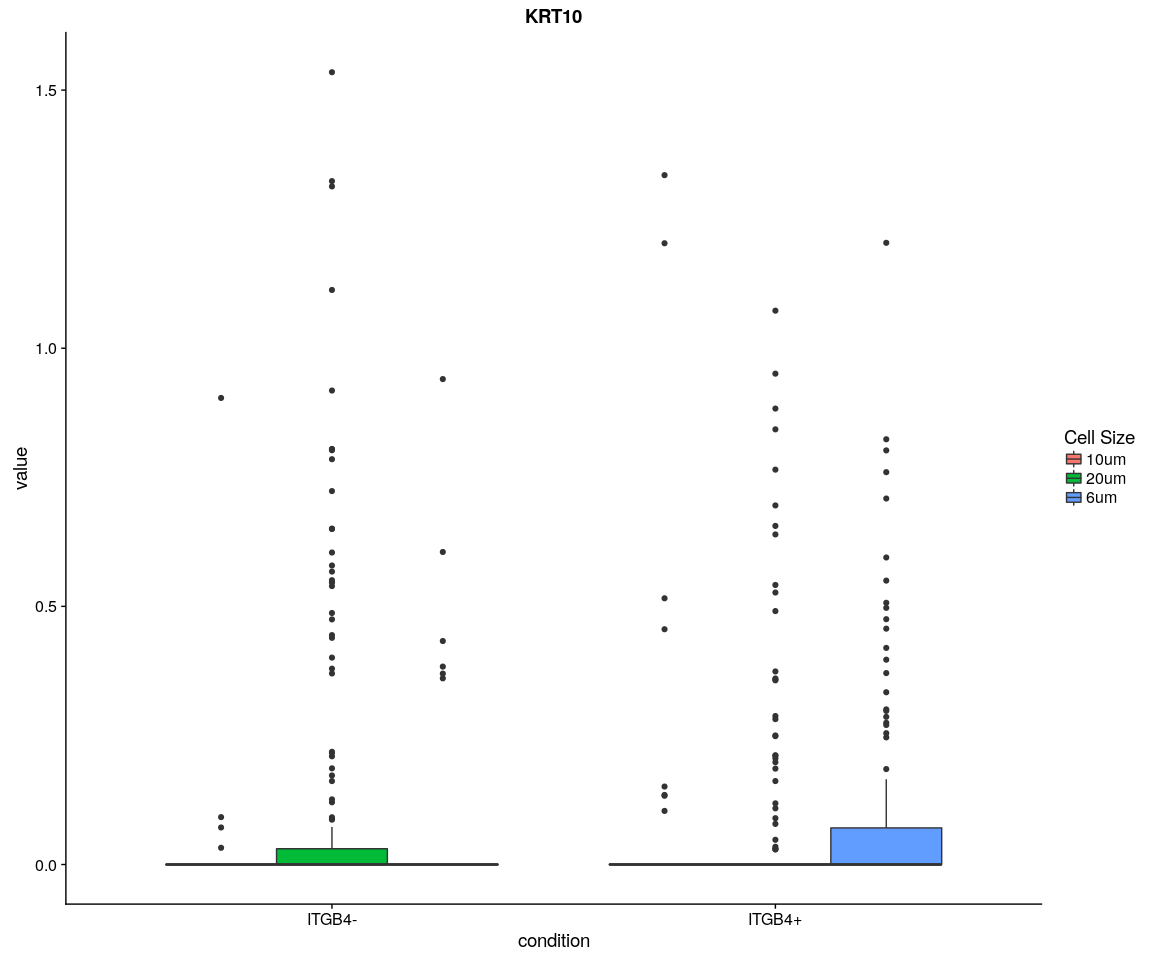
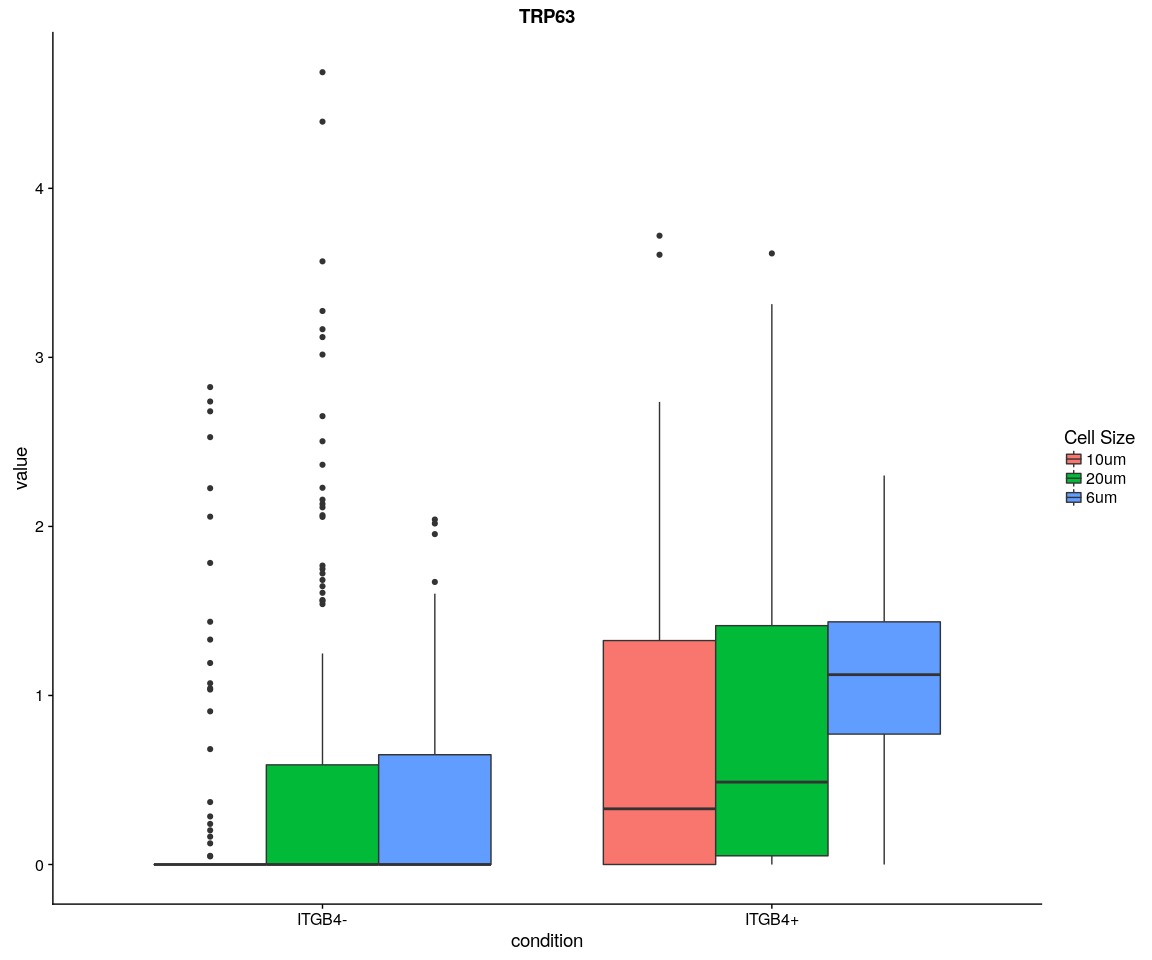
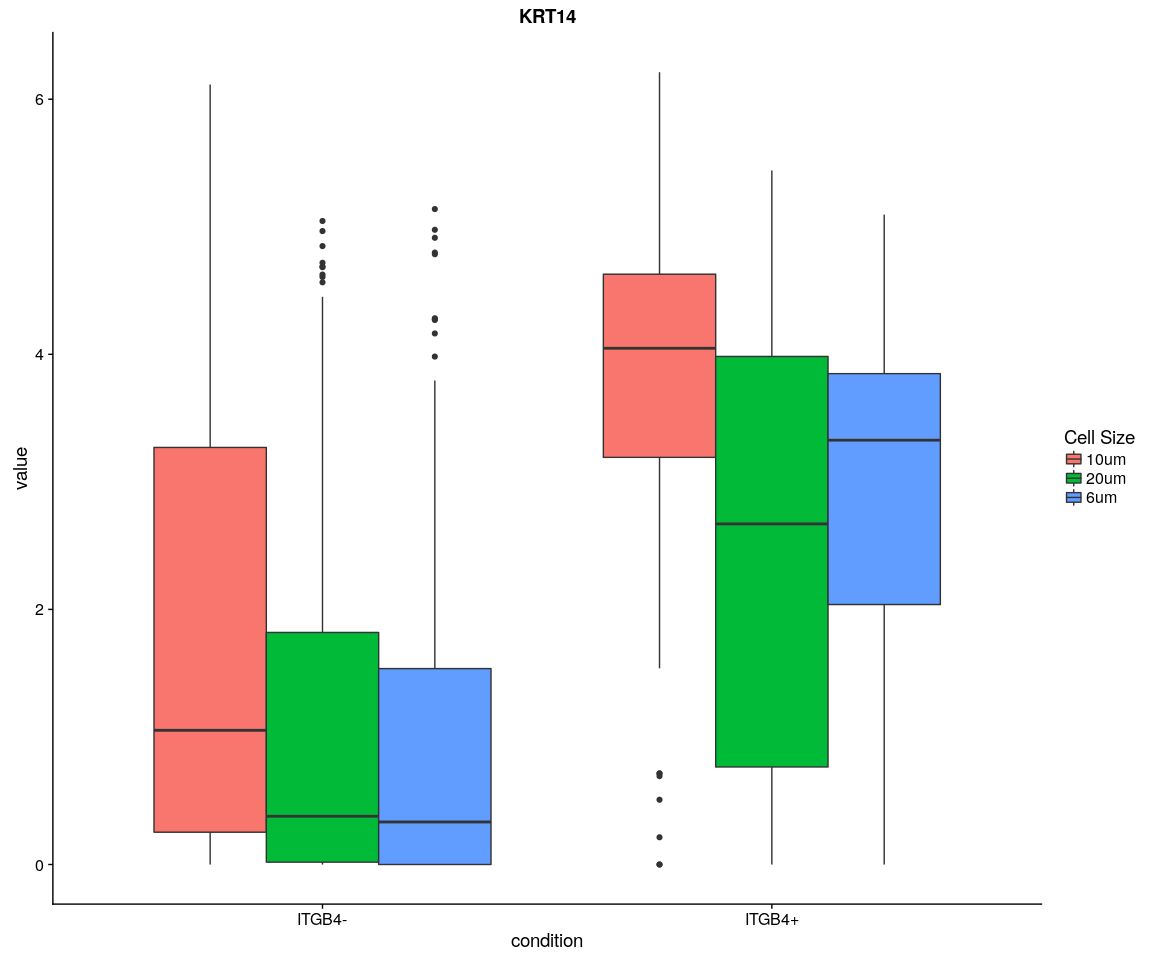
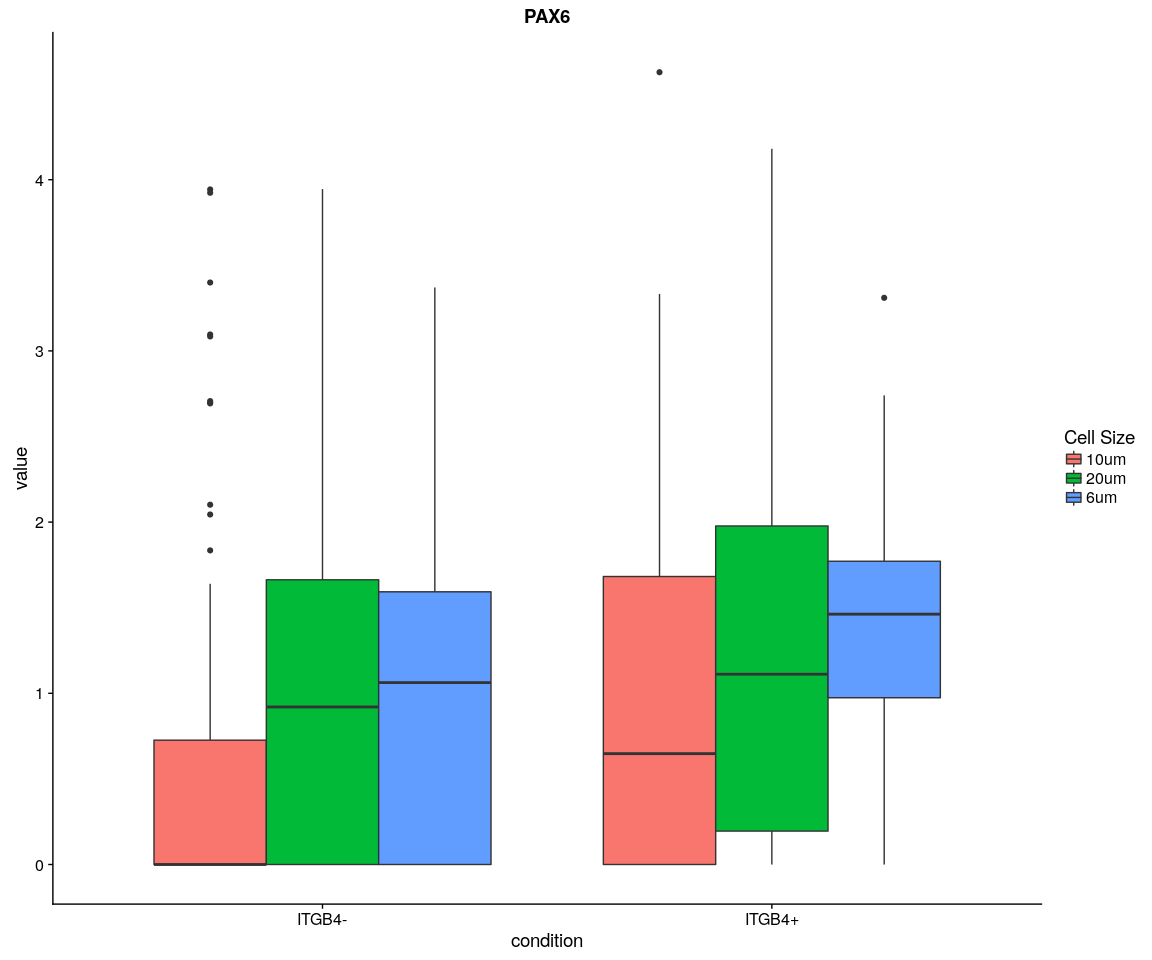
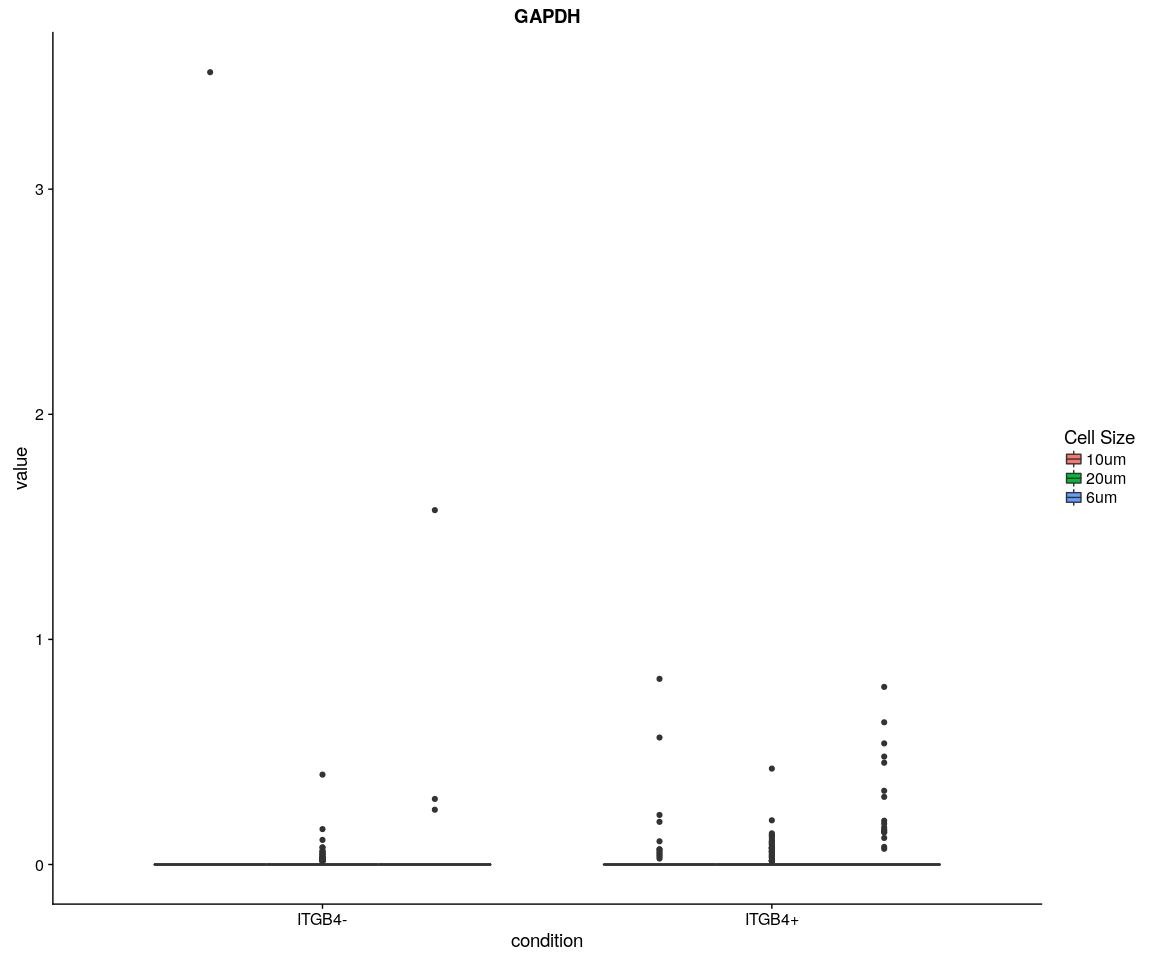
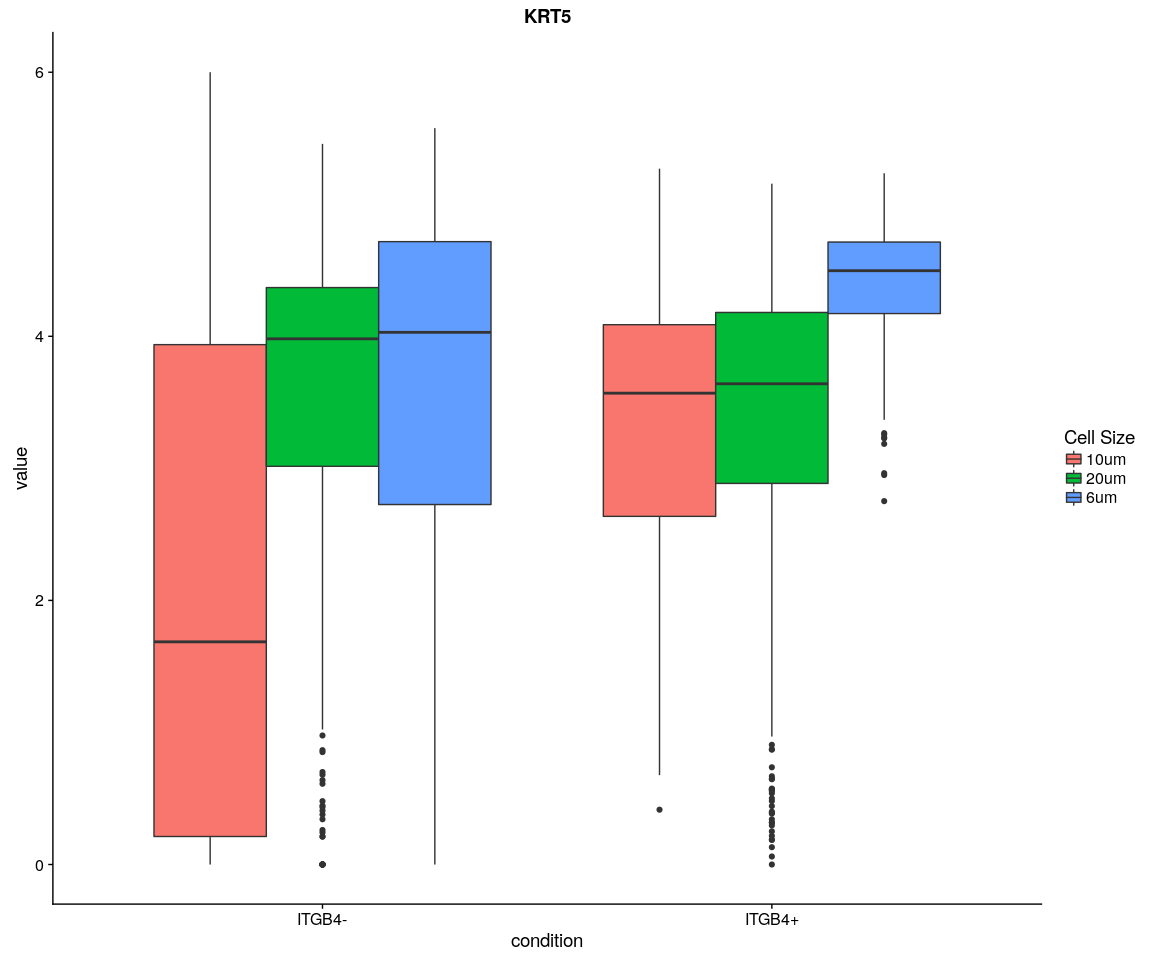
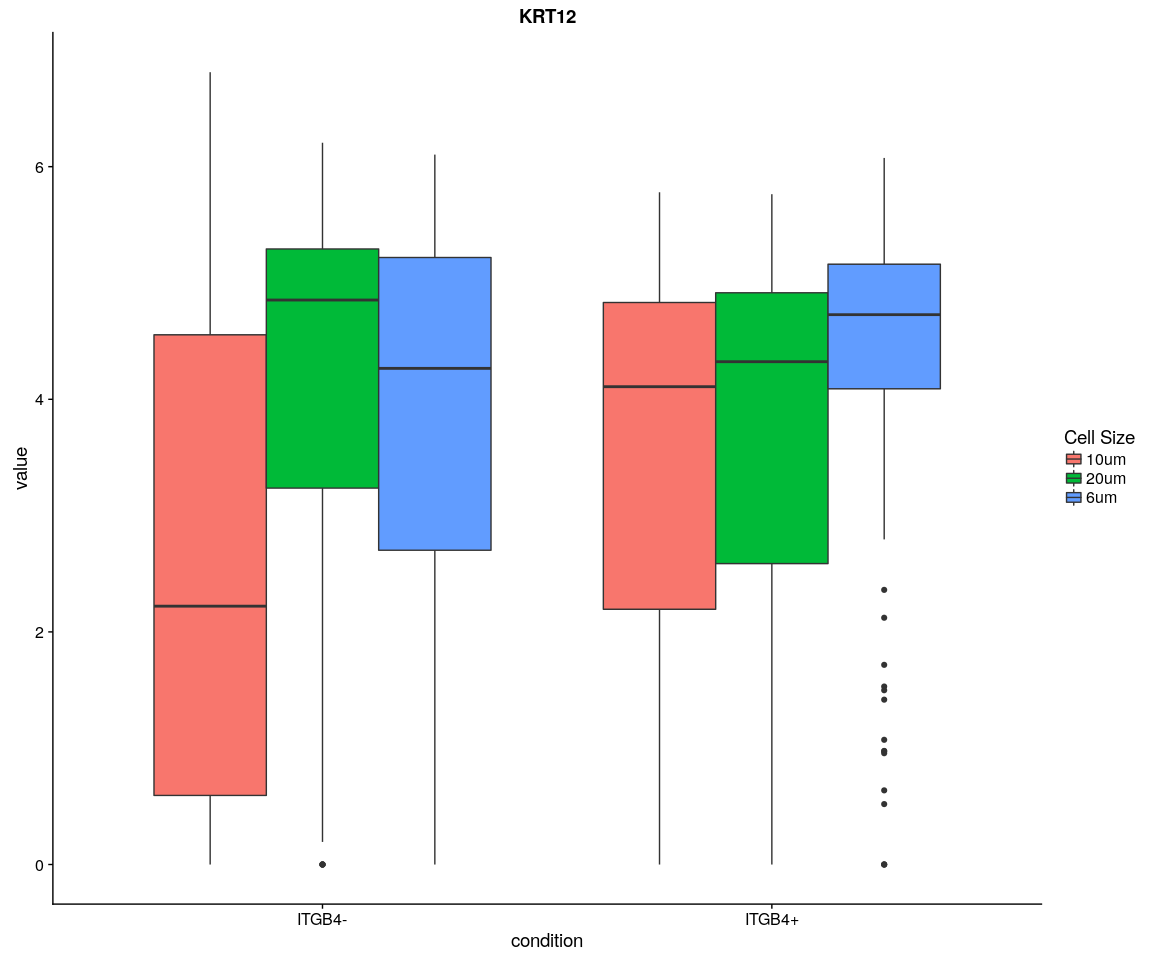
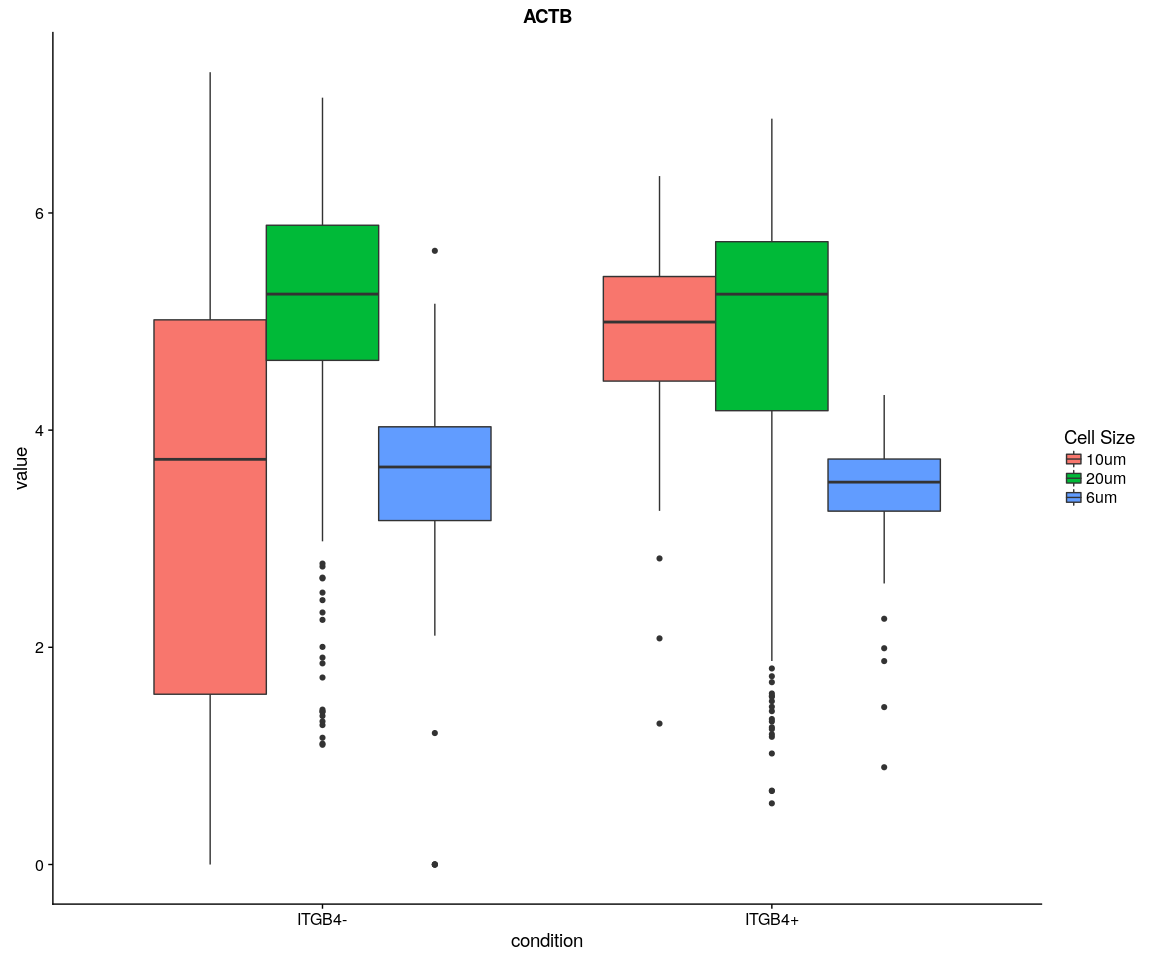
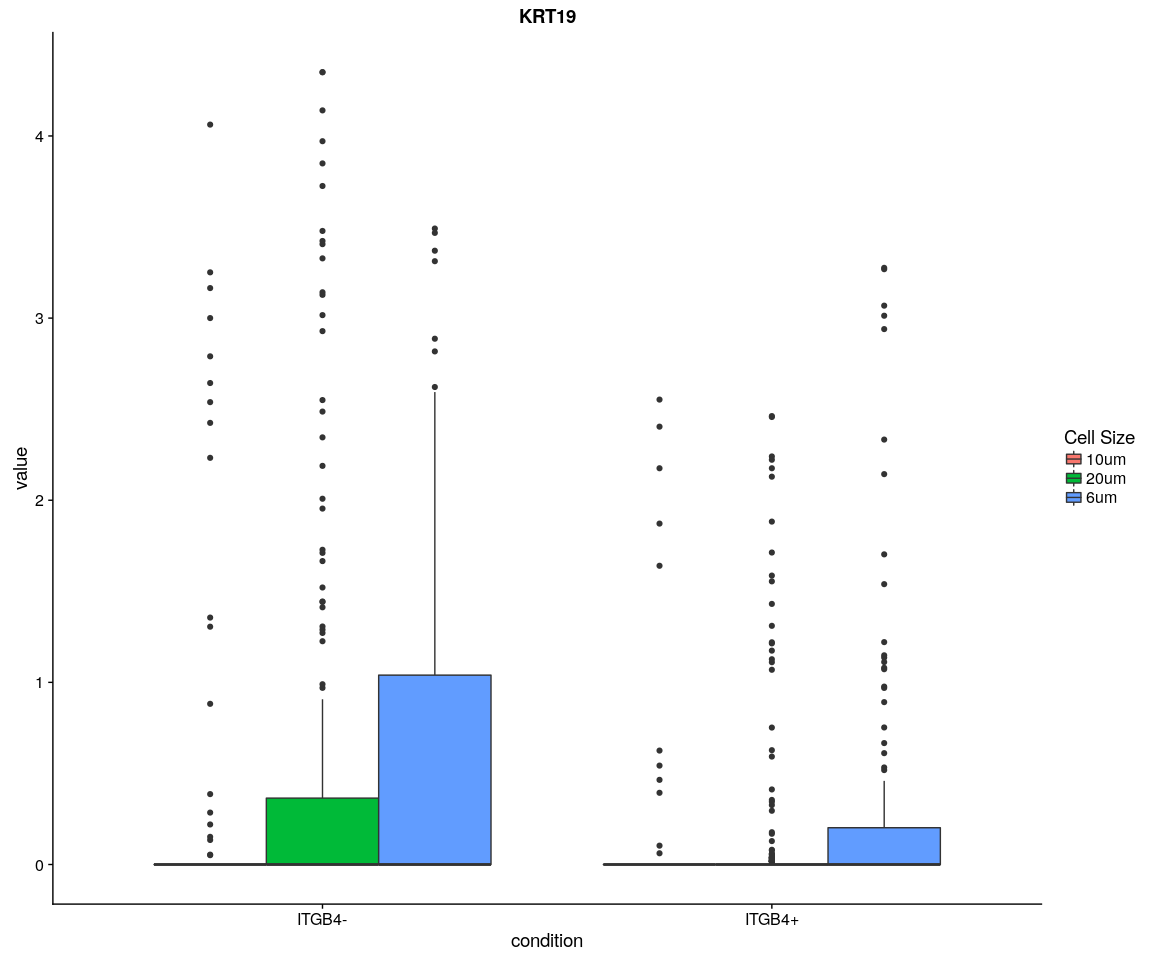
#### Violin

# p<-ggplot(data = X,aes(y=value,x=condition,fill=cell.size))  
# p+geom\_violin(trim = FALSE,scale =  
# 'width')+facet\_wrap(~variable+condition)+  
# geom\_jitter()+guides(fill=guide\_legend(title='Cell Size'))  
  
for (var in as.character(unique(X$variable))) {  
 p <- ggplot(data = X[X$variable == var, ], aes(y = value, x = condition,   
 fill = cell.size))  
 print(p + geom\_violin(trim = FALSE, scale = "width") + geom\_jitter() + guides(fill = guide\_legend(title = "Cell Size")) +   
 ggtitle(label = var))  
}



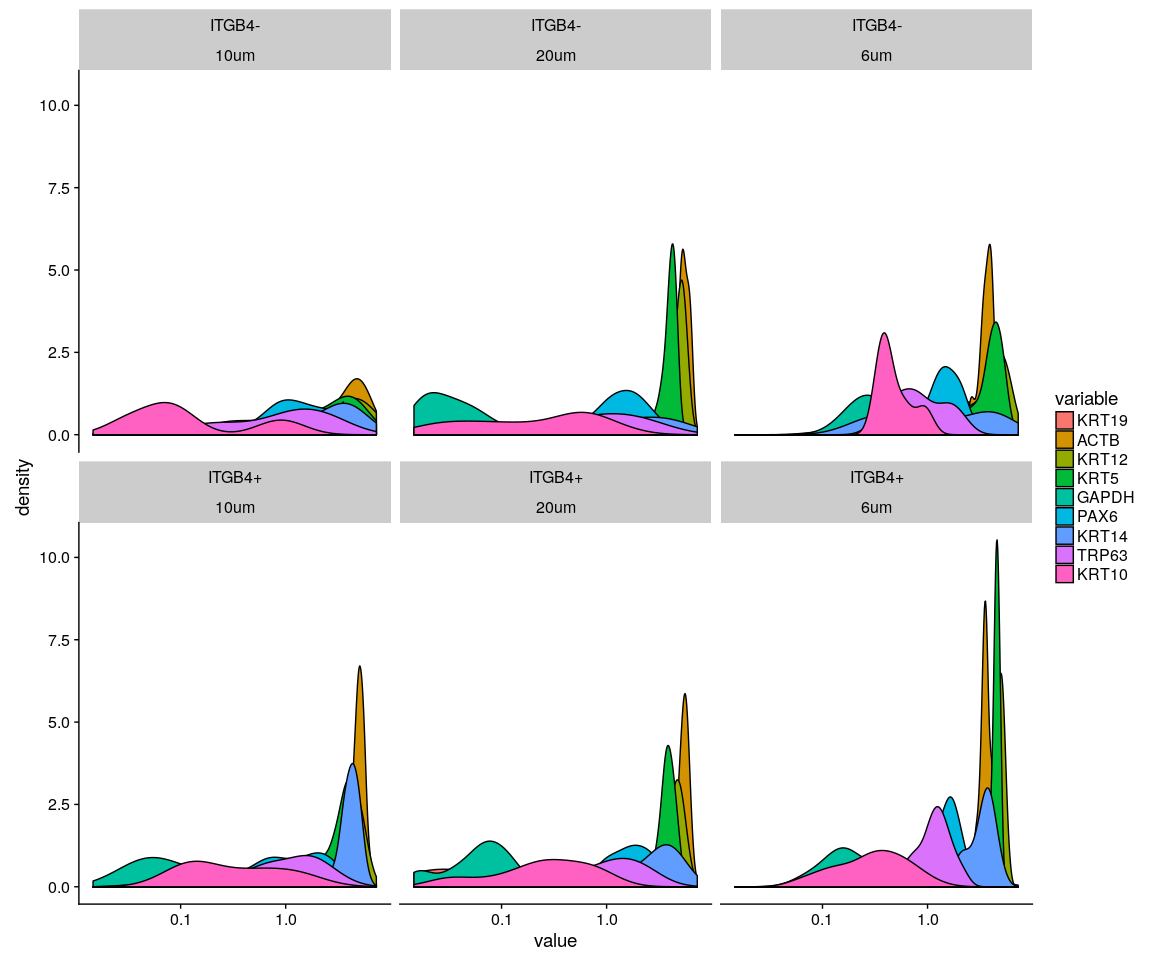
#### Boxplot

# p<-ggplot(data = X,aes(y=value,x=condition,fill=cell.size))  
# p+geom\_boxplot()+guides(fill=guide\_legend(title='Cell  
# Size'))+facet\_wrap(~variable+condition)  
for (var in as.character(unique(X$variable))) {  
 p <- ggplot(data = X[X$variable == var, ], aes(y = value, x = condition,   
 fill = cell.size))  
 print(p + geom\_boxplot() + guides(fill = guide\_legend(title = "Cell Size")) +   
 ggtitle(label = var))  
}

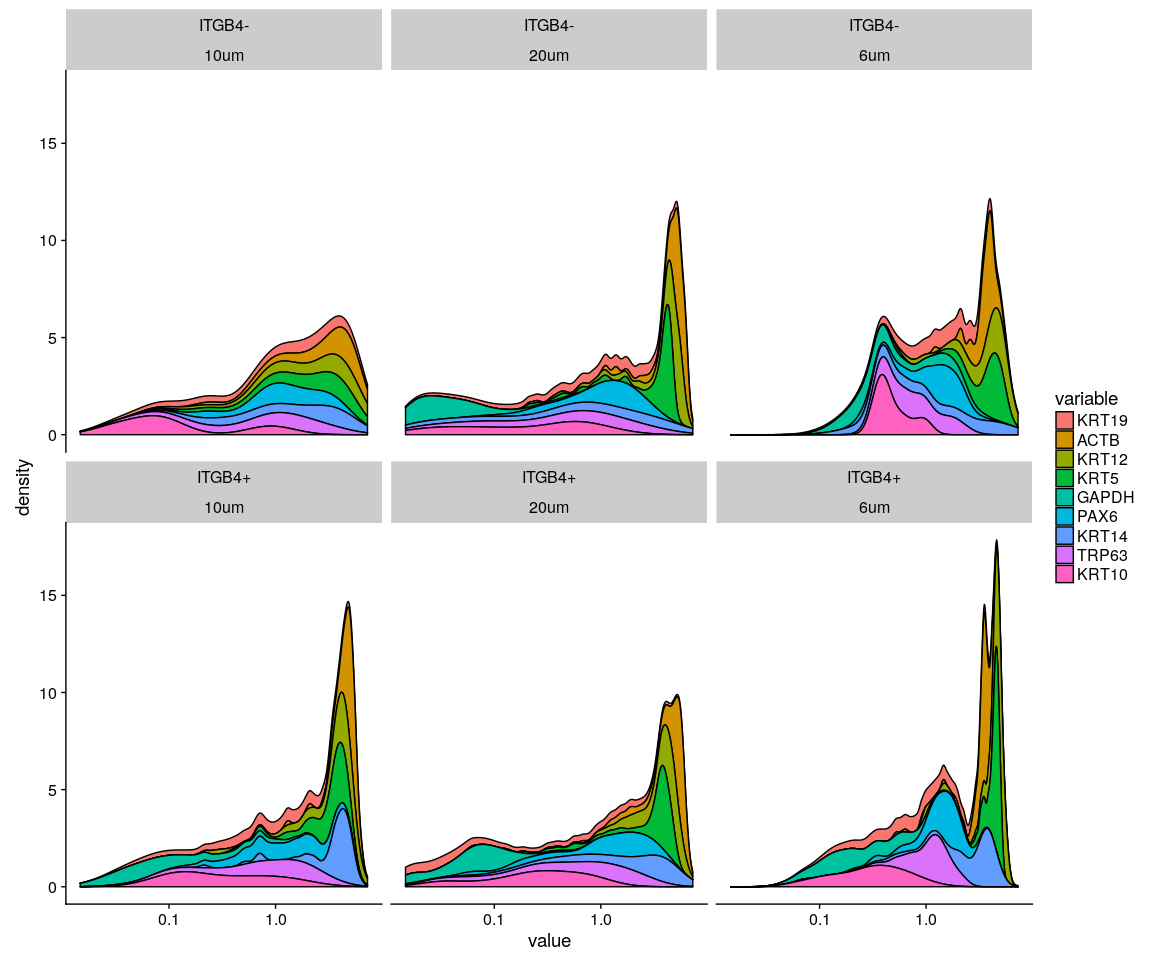


#### Density,histogram

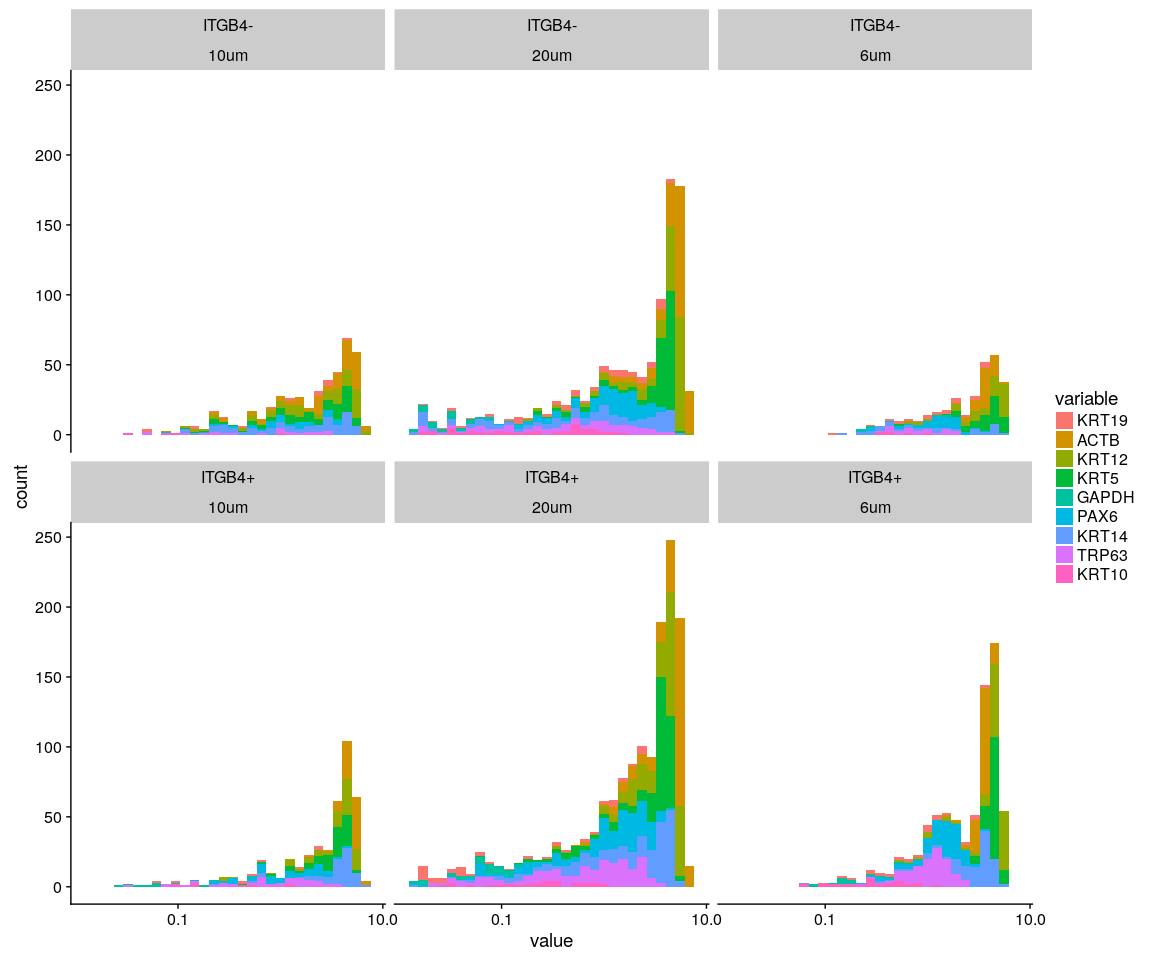
ggplot(data = X, aes(x = value, fill = variable)) + geom\_density(kernel = "gaussian") +   
 scale\_x\_log10() + facet\_wrap(~condition + cell.size)



ggplot(data = X, aes(x = value, fill = variable)) + geom\_density(kernel = "gaussian",   
 position = "stack") + scale\_x\_log10() + facet\_wrap(~condition + cell.size)



ggplot(data = X, aes(x = value, fill = variable)) + geom\_histogram() + scale\_x\_log10() +   
 facet\_wrap(~condition + cell.size)



### Split the data according to whether the gene ITGB4 is Negative or negative

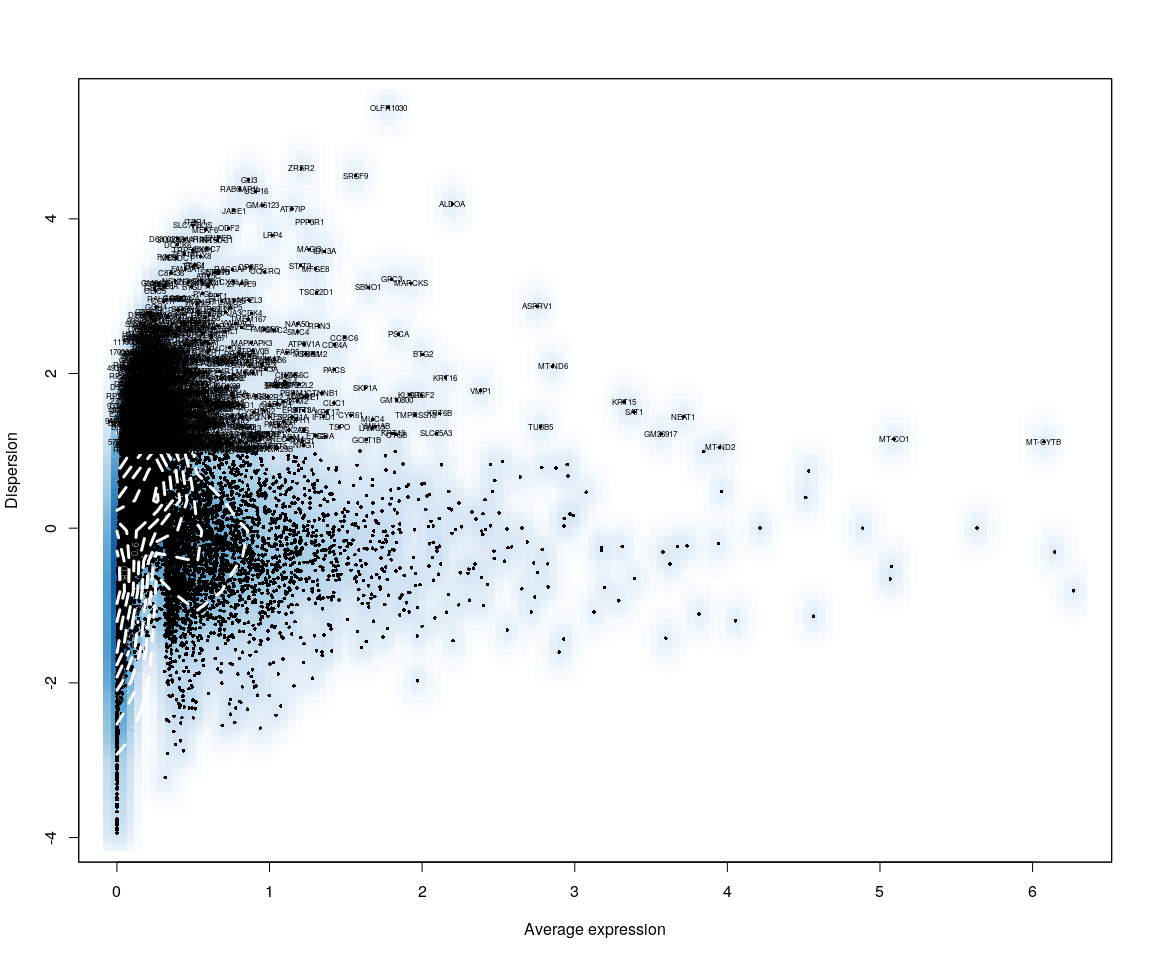
ITGB4 <- as.integer(mouse.only.pro["ITGB4", ])  
Positive.idx <- which(ITGB4 > 0)  
Negative.idx <- which(ITGB4 == 0)  
Positive.data <- mouse.only.pro[, Positive.idx, drop = FALSE]  
Negative.data <- mouse.only.pro[, Negative.idx, drop = FALSE]

### Create Seurat object and not caculate DESeq

#### Positive object

Positive.pbmc <- DESeq\_SeuratObj(X = Positive.data, min.cells = 10, min.genes = 2)

## [1] "Scaling data matrix"  
##   
 |   
 | | 0%  
 |   
 |=================================================================| 100%

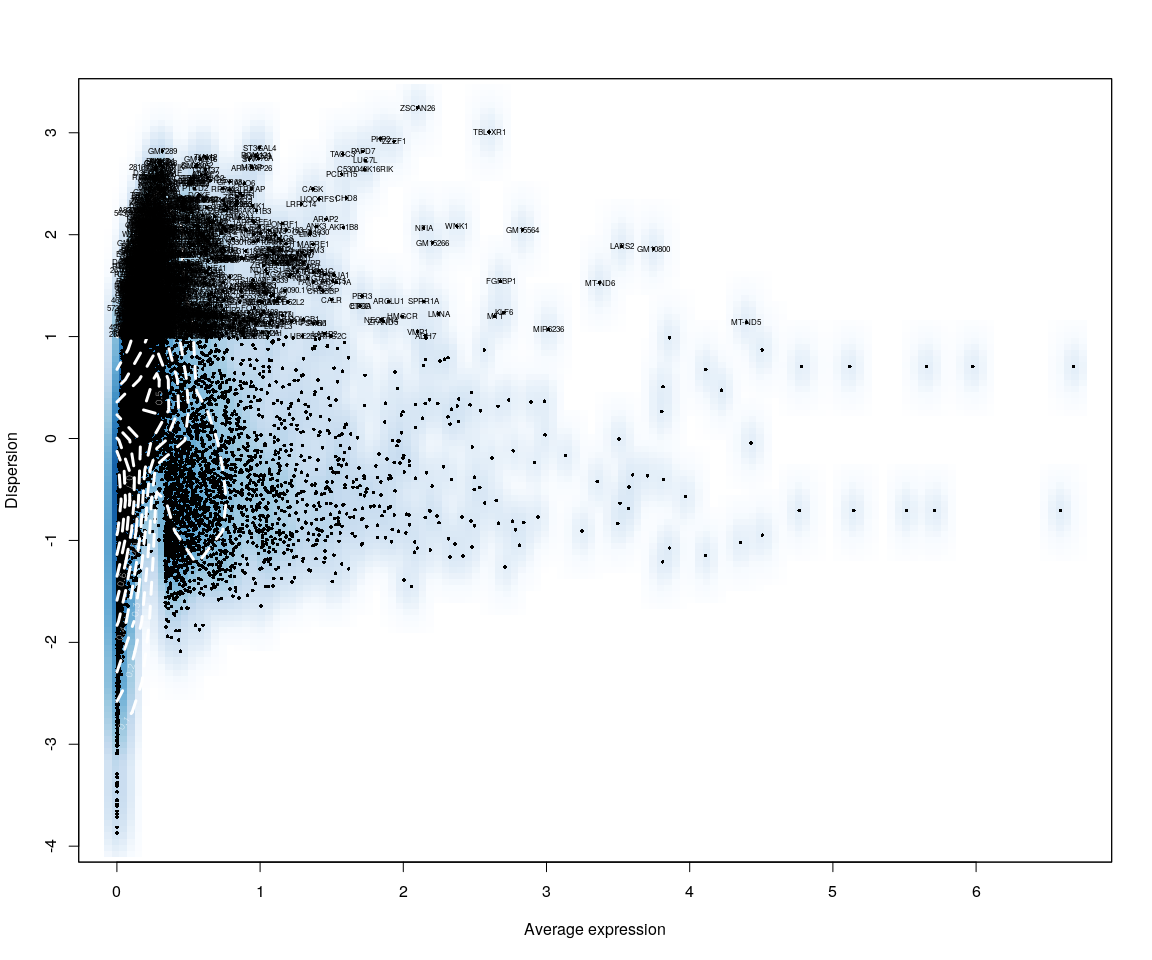


Positive.sample.group <- unlist(lapply(Positive.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][1])))  
Positive.sample.cellsize <- unlist(lapply(Positive.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
  
Positive.pbmc <- SetIdent(Positive.pbmc, cells.use = Positive.pbmc@cell.names,   
 ident.use = Positive.sample.cellsize)

#### Negative object

Negative.pbmc <- DESeq\_SeuratObj(X = Negative.data, min.cells = 10, min.genes = 2)

## [1] "Scaling data matrix"  
##   
 |   
 | | 0%  
 |   
 |=================================================================| 100%



Negative.sample.group <- unlist(lapply(Negative.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][1])))  
Negative.sample.cellsize <- unlist(lapply(Negative.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
  
Negative.pbmc <- SetIdent(Negative.pbmc, cells.use = Negative.pbmc@cell.names,   
 ident.use = Negative.sample.cellsize)

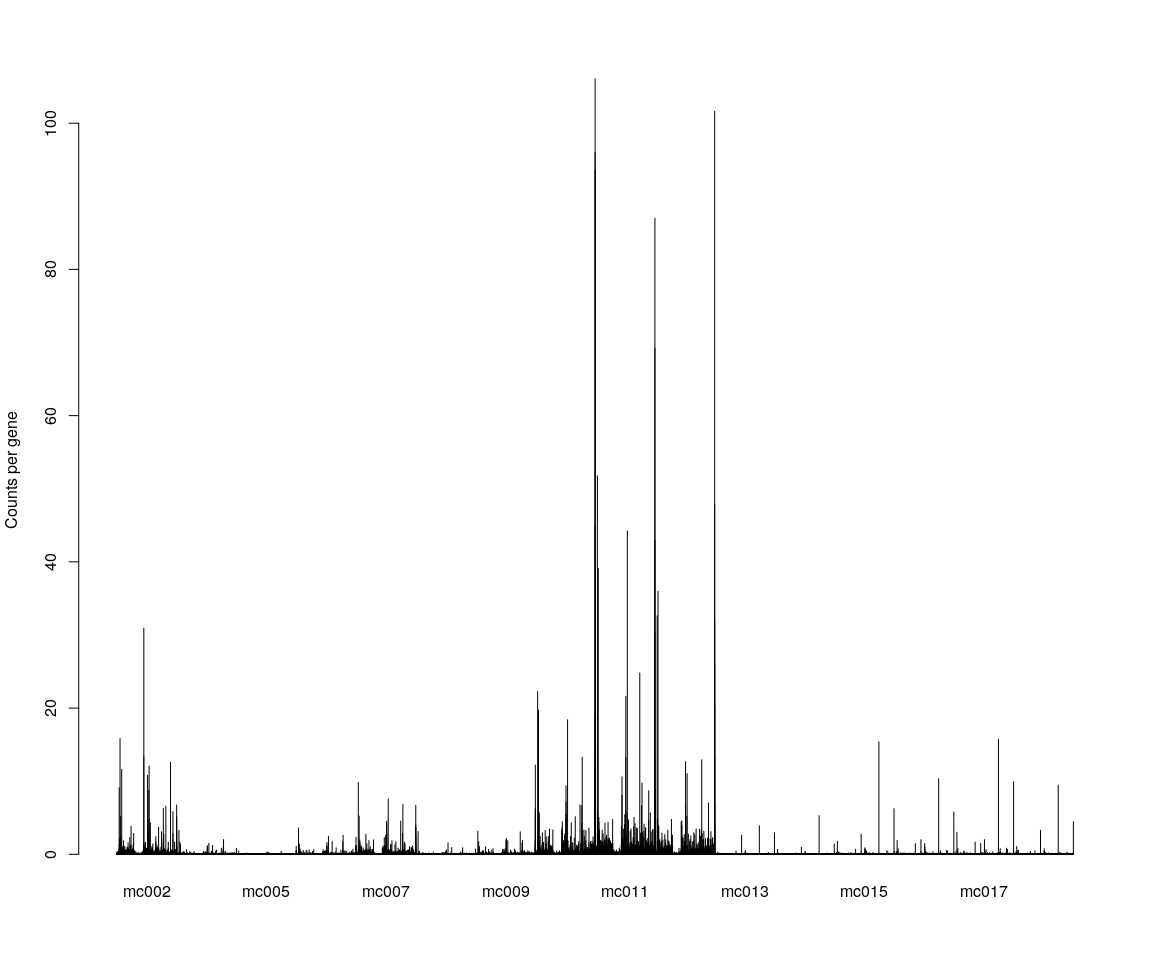
Accordind to the Dispersion vs Avearge expression of Positive and Negative data on ITGB4,they tell us that the although they have similar shape and trend,dispersion of Positive data is more significant than Negative in some genes.

## Step 1: analysis on Positive data

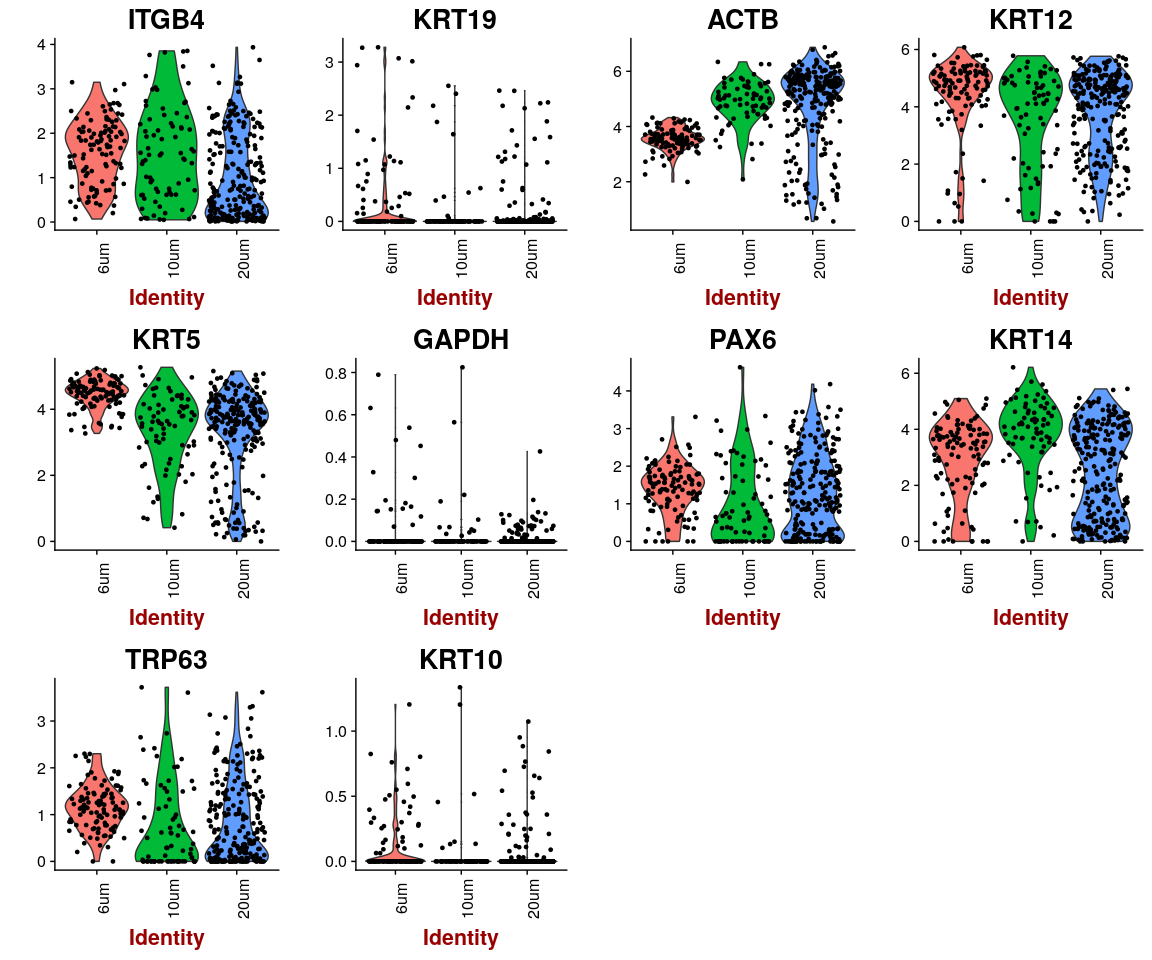
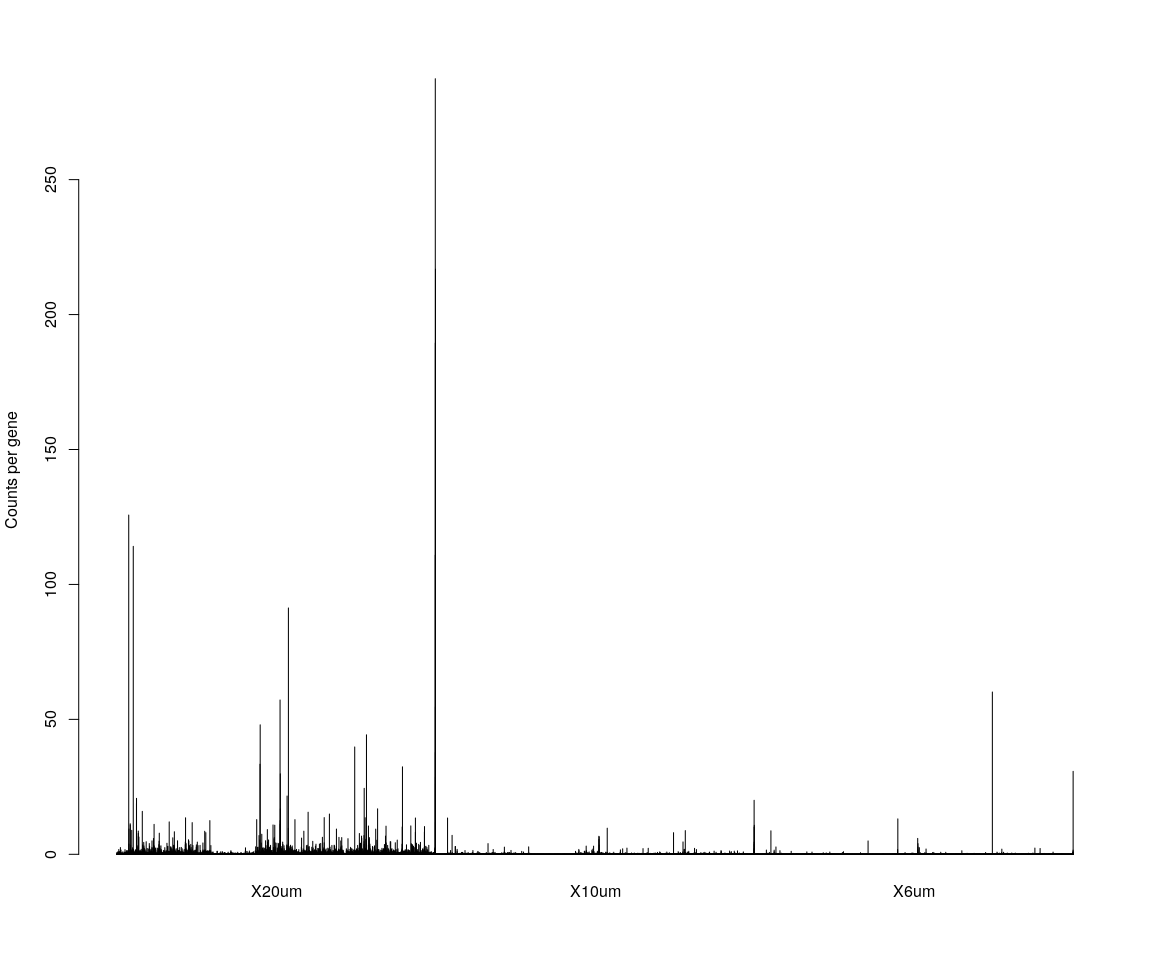
## Figure Explore

### First,use the plot,eg. Barplot,Violin…,we can explore some message from sample

Group\_Bar(Positive.pbmc@raw.data, group = Positive.sample.group)



Group\_Bar(Positive.pbmc@raw.data, group = Positive.sample.cellsize)  
  
VlnPlot(Positive.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Positive.pbmc@raw.data)], y.lab.rot = 90) # Violinn plot of gene ITGB in all sample



## Dimensionality reduction

### **PCA** and **tSNE**

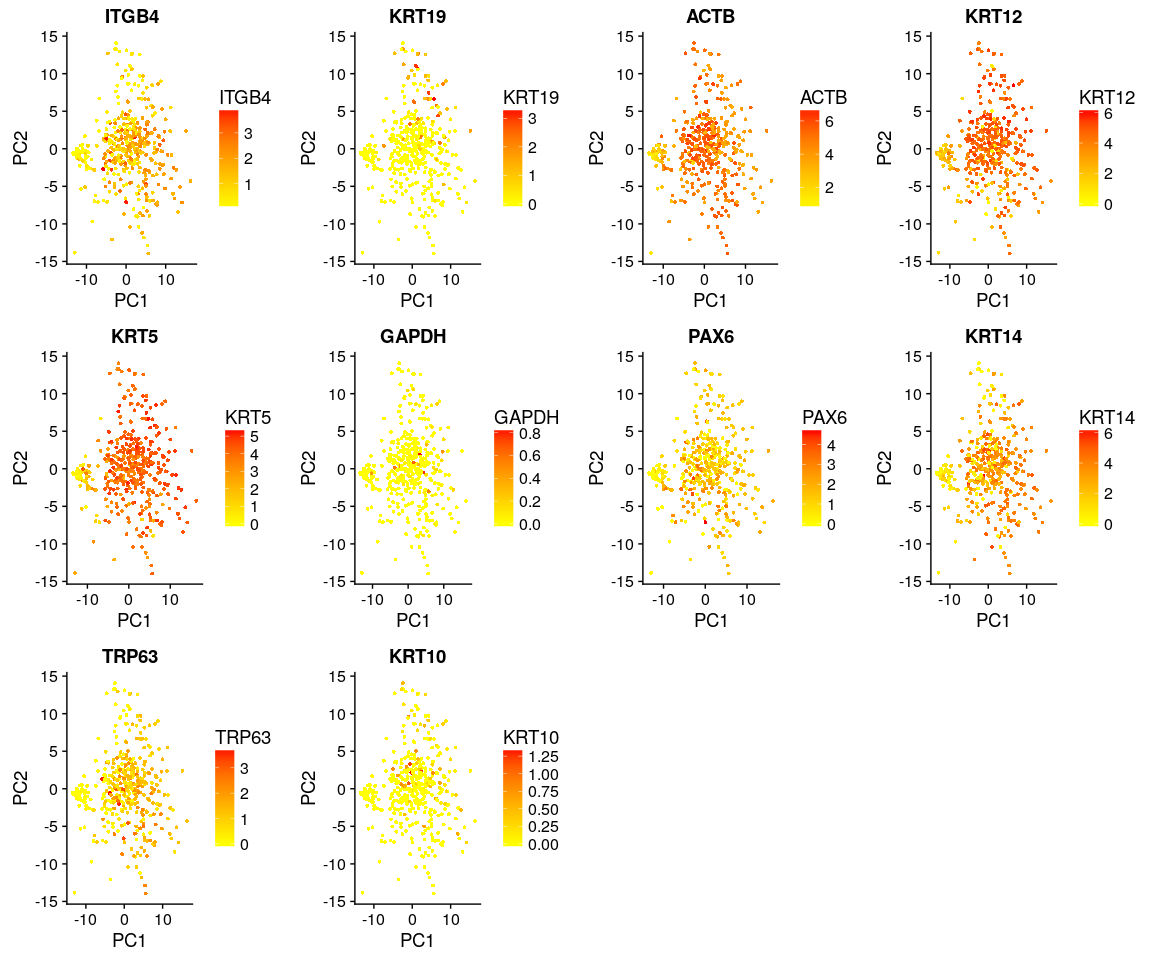
Here,do the dimensionality reduction using the PCA, tSNE method

*It will take a long time to caculate significant pcs.So,here we use the default value*

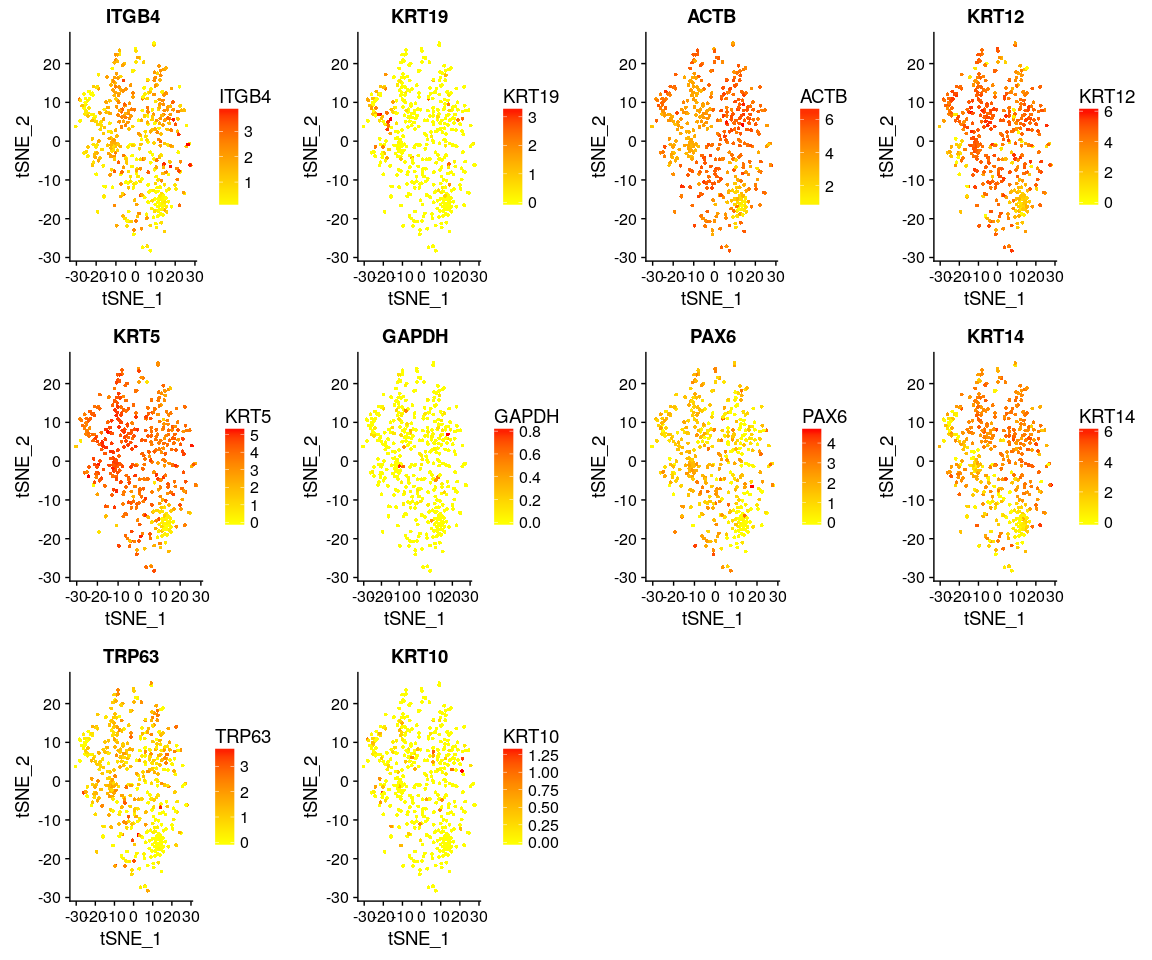
Positive.pbmc <- PCA.TSNE(object = Positive.pbmc, pcs.compute = FALSE, num.pcs = 28)

### After the PCA and tSNE,try plot: Featureplot of **ITGB4**,four var.genes,PCA plot,tSNE plot…

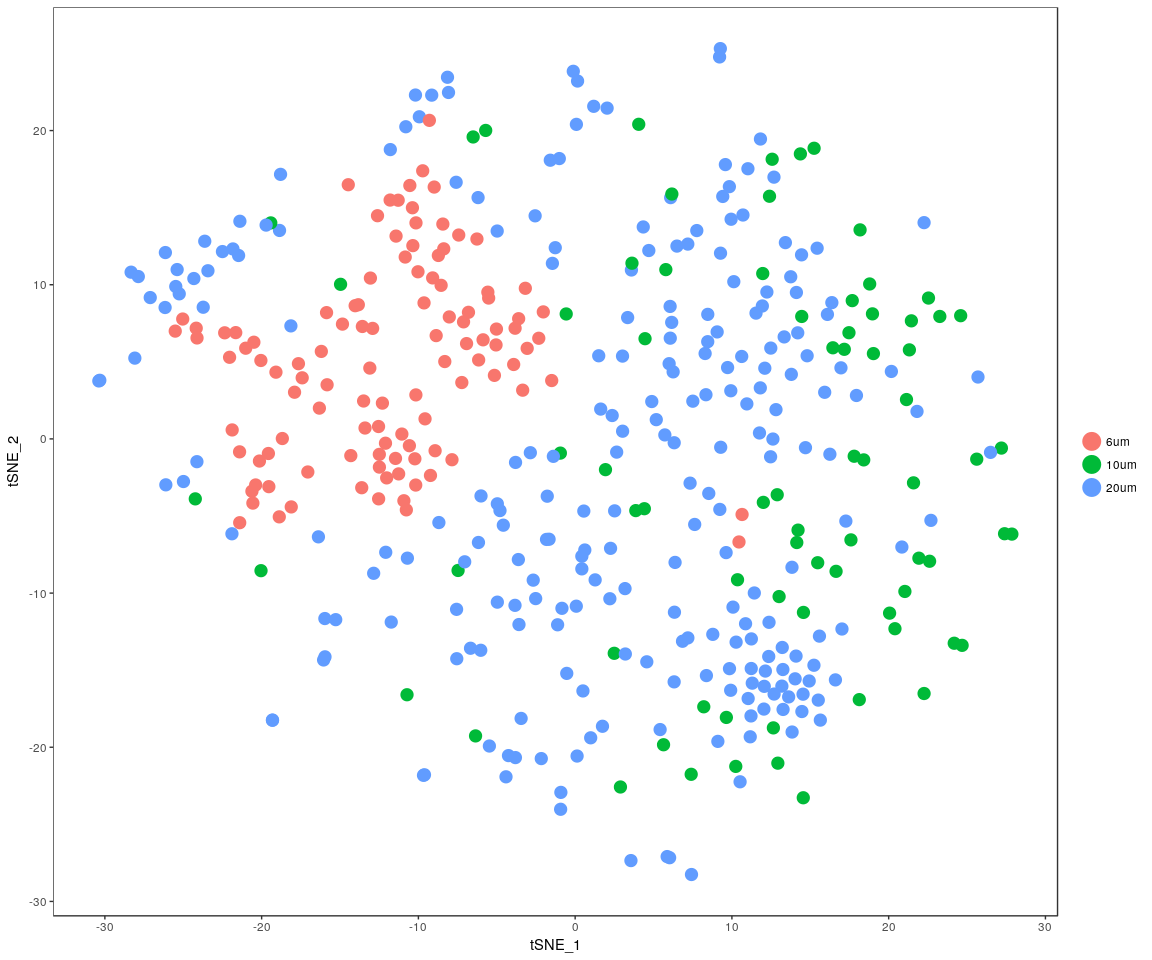
FeaturePlot(object = Positive.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Positive.pbmc@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "pca") # ITGB4 gene in part dataset



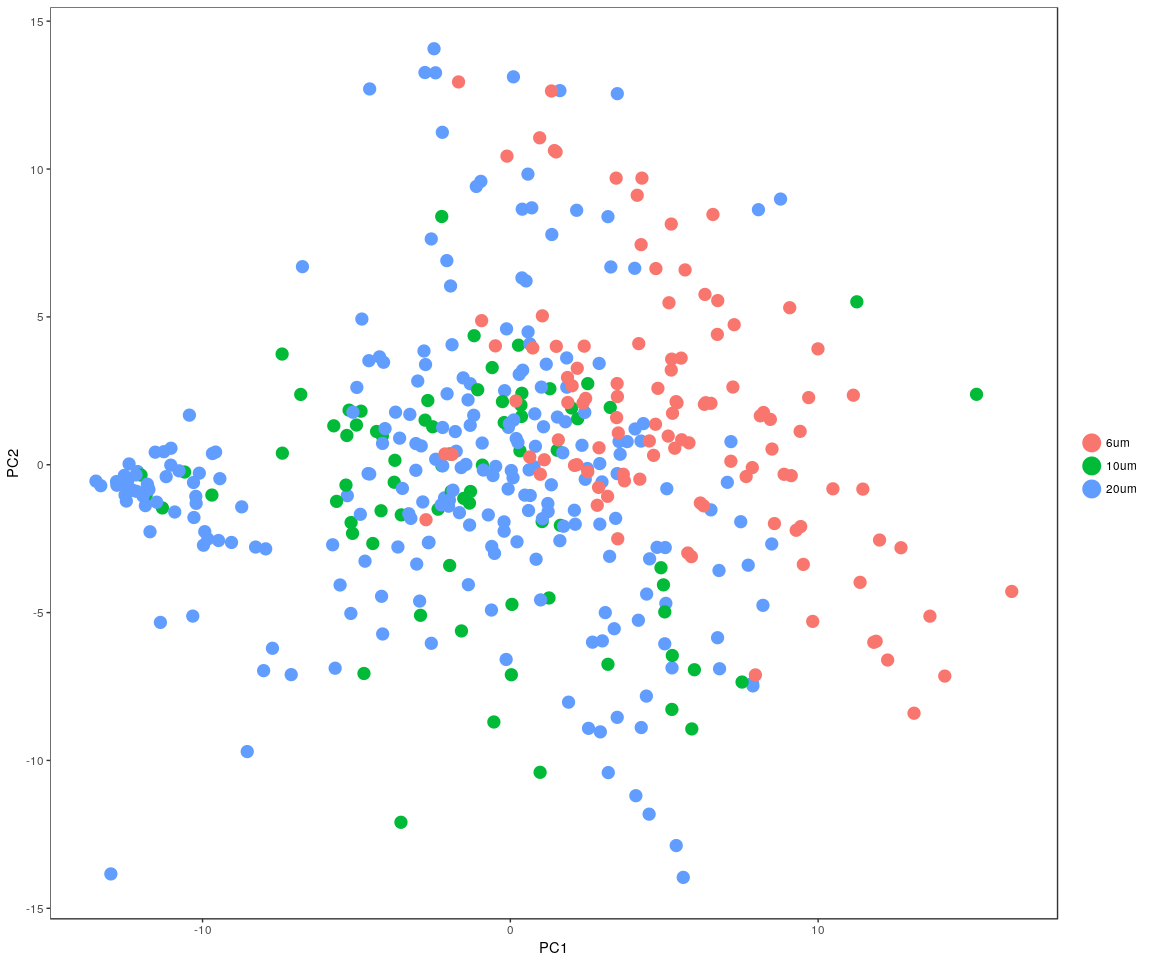
FeaturePlot(object = Positive.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Positive.pbmc@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "tsne") # ITGB4 gene in part dataset



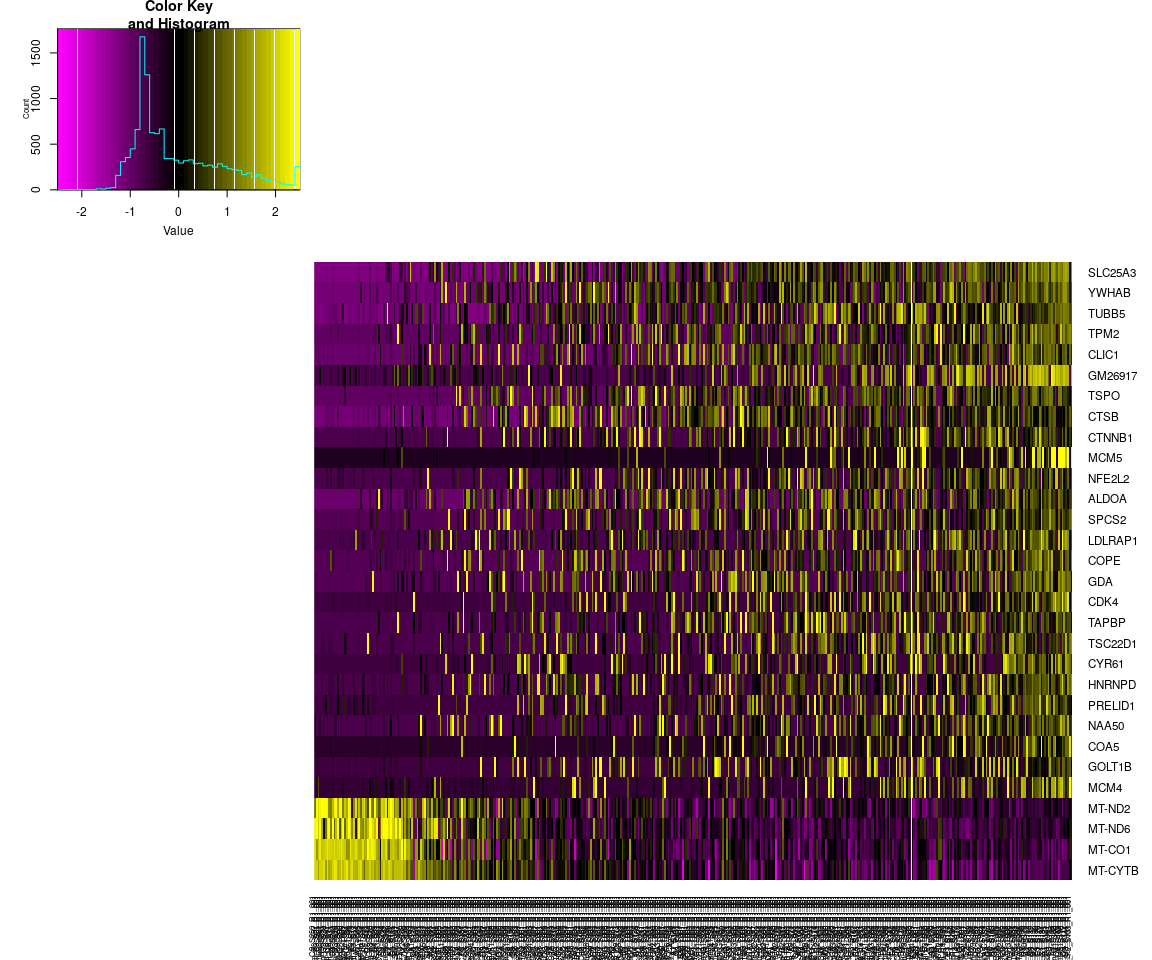
DimPlot(Positive.pbmc, reduction.use = "tsne", pt.size = 4) # grour by sample



DimPlot(Positive.pbmc, reduction.use = "pca", pt.size = 4) # grour by sample



DimHeatmap(Positive.pbmc, reduction.type = "pca", check.plot = FALSE)

 The Faetureplot of ITGB4, KRT19, ACTB, KRT12, KRT5, GAPDH, PAX6, KRT14, TRP63, KRT10based on **PCA** shows that,they only has high expression level in few samples,and expresss lowly in most sample.It means that may be these important genes express differently across sample.The plot also tell us the gene **KRT5,GAPDH,PAXX6,KRT14** have more higher expression level than the other important genes.It is consistent with the result of violin plot. But the *tSNE* and \* PCA \* plot show that, the sample can not be split apparently.The result may be is not good based on the PCA and tSNE method.

## Differential expression

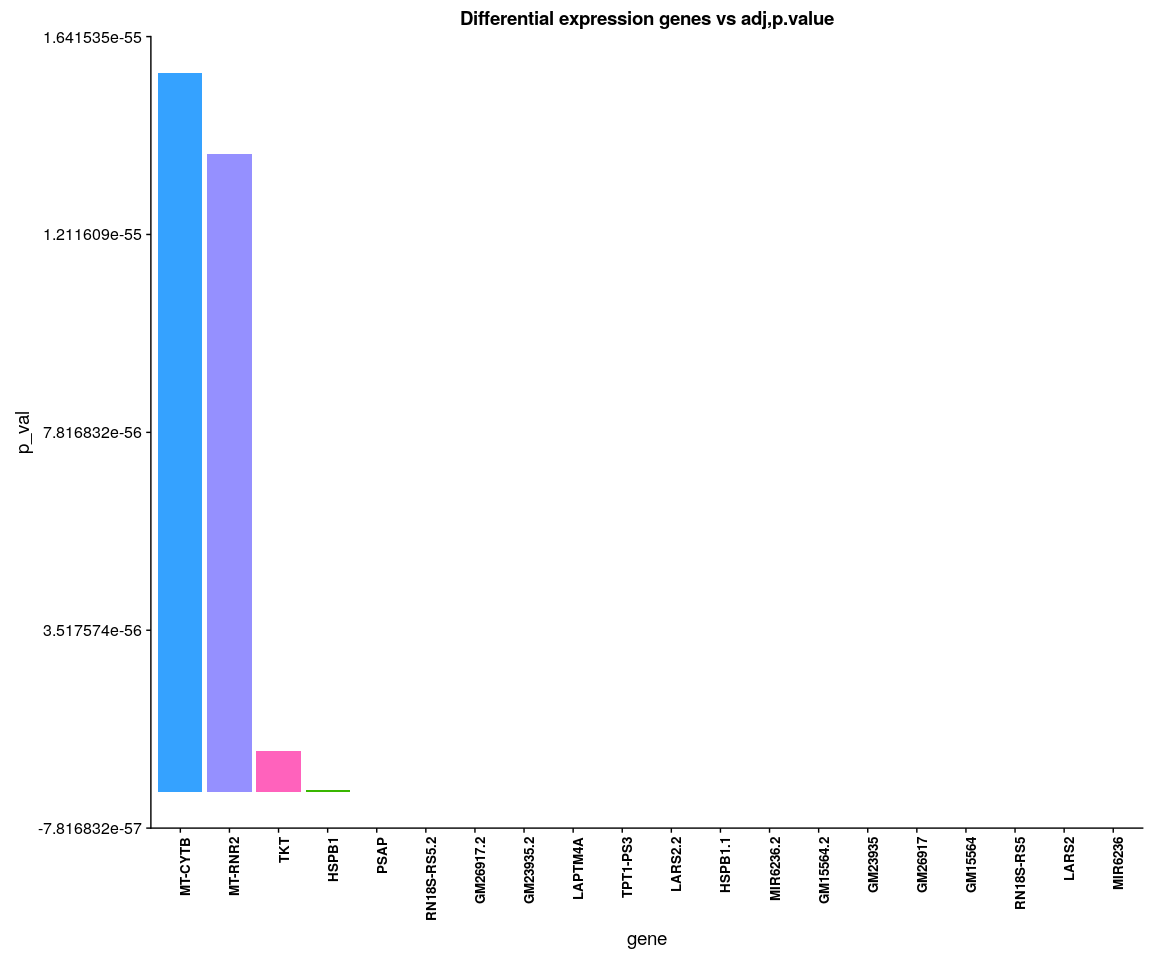
Next,we will have analysis on gene differential expression.Find maker genes across sample.We use the method: \*\*wilcox test\*\*

# Finds markers (differentially expressed genes) for each of the identity  
# classes in a dataset  
Positive.markers <- FindAllMarkers(Positive.pbmc, test.use = "bimod", print.bar = FALSE)  
head(Positive.markers)

## p\_val avg\_logFC pct.1 pct.2 p\_val\_adj cluster  
## MIR6236 5.919142e-158 3.057346 1.000 0.191 8.022805e-154 6um  
## LARS2 4.250212e-152 2.849736 1.000 0.959 5.760737e-148 6um  
## RN18S-RS5 4.850037e-146 2.928549 1.000 1.000 6.573740e-142 6um  
## GM15564 1.195421e-140 2.251240 1.000 0.856 1.620274e-136 6um  
## GM26917 3.212669e-119 3.815874 0.991 0.622 4.354452e-115 6um  
## GM23935 8.754999e-112 1.576575 1.000 0.381 1.186653e-107 6um  
## gene  
## MIR6236 MIR6236  
## LARS2 LARS2  
## RN18S-RS5 RN18S-RS5  
## GM15564 GM15564  
## GM26917 GM26917  
## GM23935 GM23935

We check whether the important genes are still in the marker genes we found from the DESeq analysis. the genes:ITGB4, KRT19, ACTB, KRT12, KRT5, KRT14 are still in the marker genes.

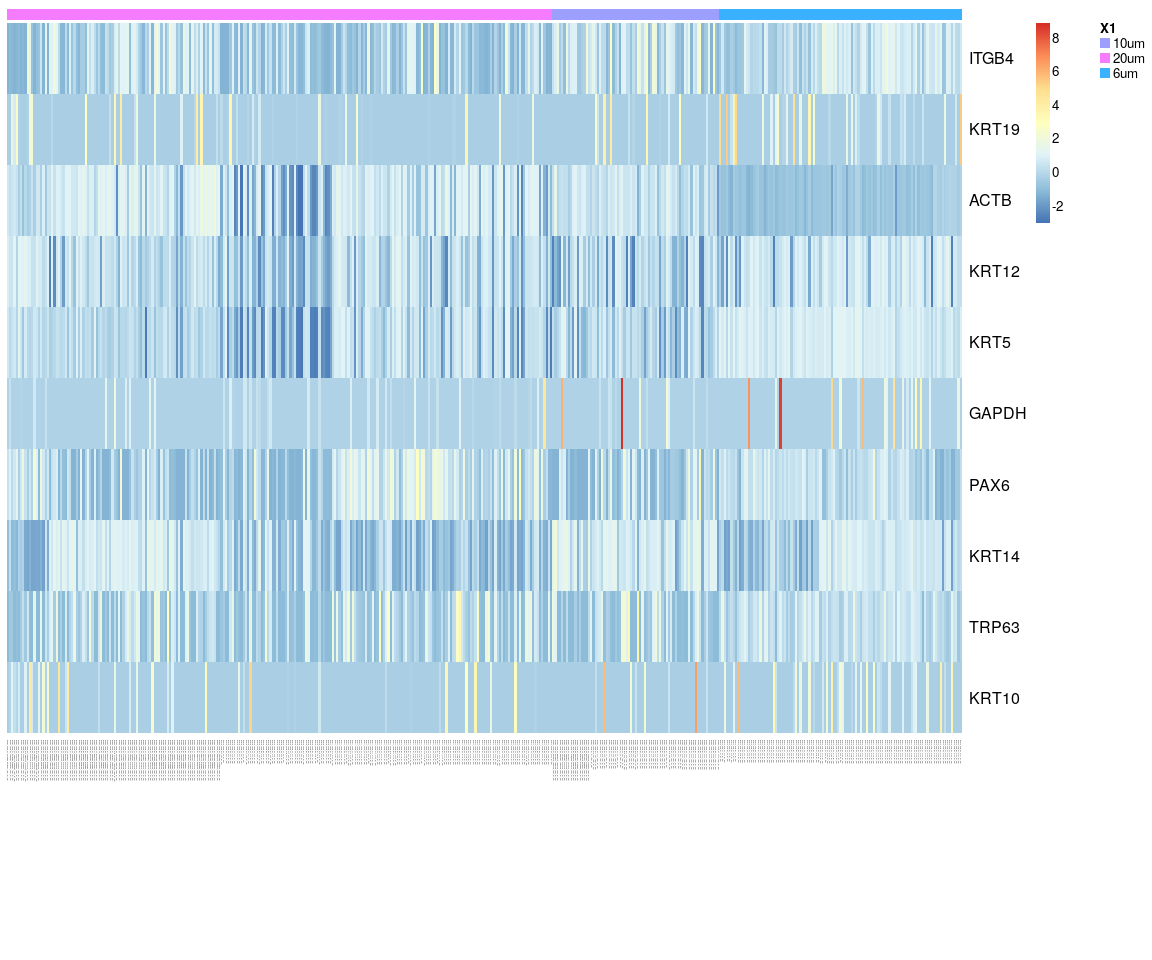
### Bar plot of gene’s p.val



Positive.heatmap <- Heatmap\_fun(genes = important.genes[important.genes %in%   
 rownames(Positive.pbmc@raw.data)], tpm.data = Positive.pbmc@scale.data,   
 condition = unique(as.character(Positive.pbmc@ident)), all.condition = as.character(Positive.pbmc@ident))

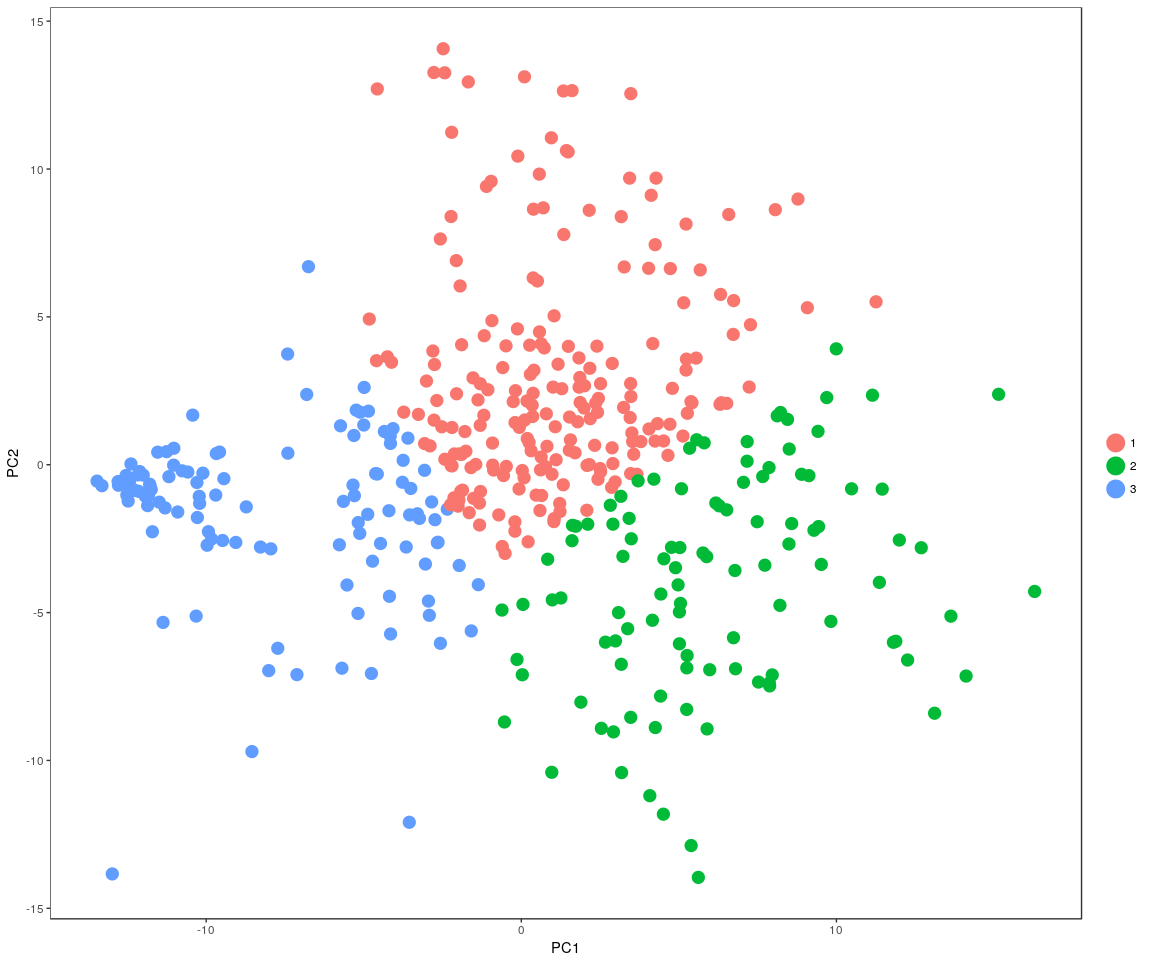
## There ara 3 conditions  
## Whether creat data accurate 0

NMF::aheatmap(Positive.heatmap[[2]], Rowv = NA, Colv = NA, annCol = Positive.heatmap[[1]],   
 scale = "none")

 We have find all marker genes across sample,there are 3438 significant genes(adjust p-value <0.05) in all marker genes.

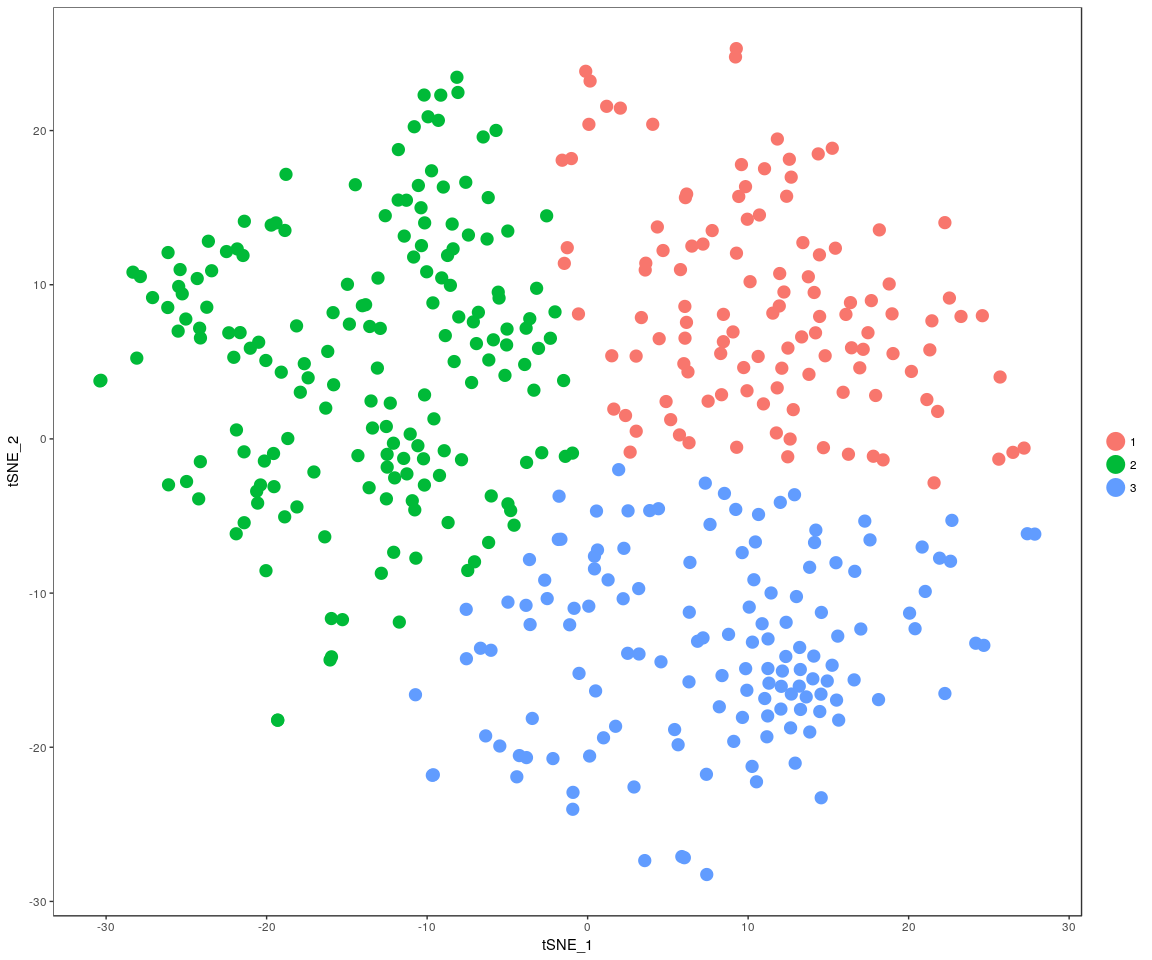
### Next,Spectral k-means clustering on single cells based on PCA

Positive.pbmc <- KClustDimension(Positive.pbmc, reduction.use = "pca", k.use = 3)  
clusters.pca <- Positive.pbmc@meta.data$kdimension.ident  
DimPlot(Positive.pbmc, pt.size = 4, group.by = "kdimension.ident")



### Spectral k-means clustering on single cells based on tSNE

Positive.pbmc <- KClustDimension(Positive.pbmc, reduction.use = "tsne", k.use = 3)  
clusters.tsne <- Positive.pbmc@meta.data$kdimension.ident  
DimPlot(Positive.pbmc, pt.size = 4, group.by = "kdimension.ident", reduction.use = "tsne")

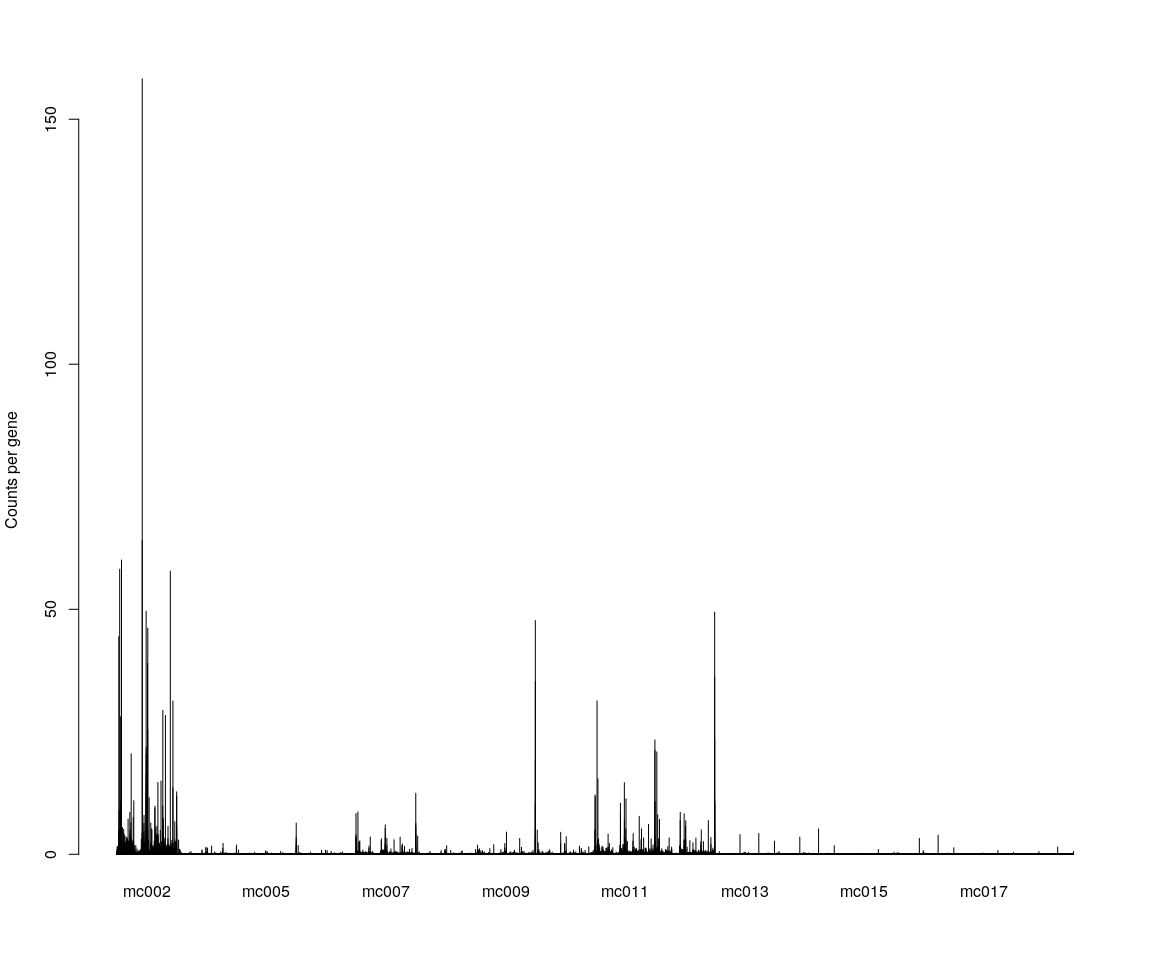


## Step 2: analysis on Negative data

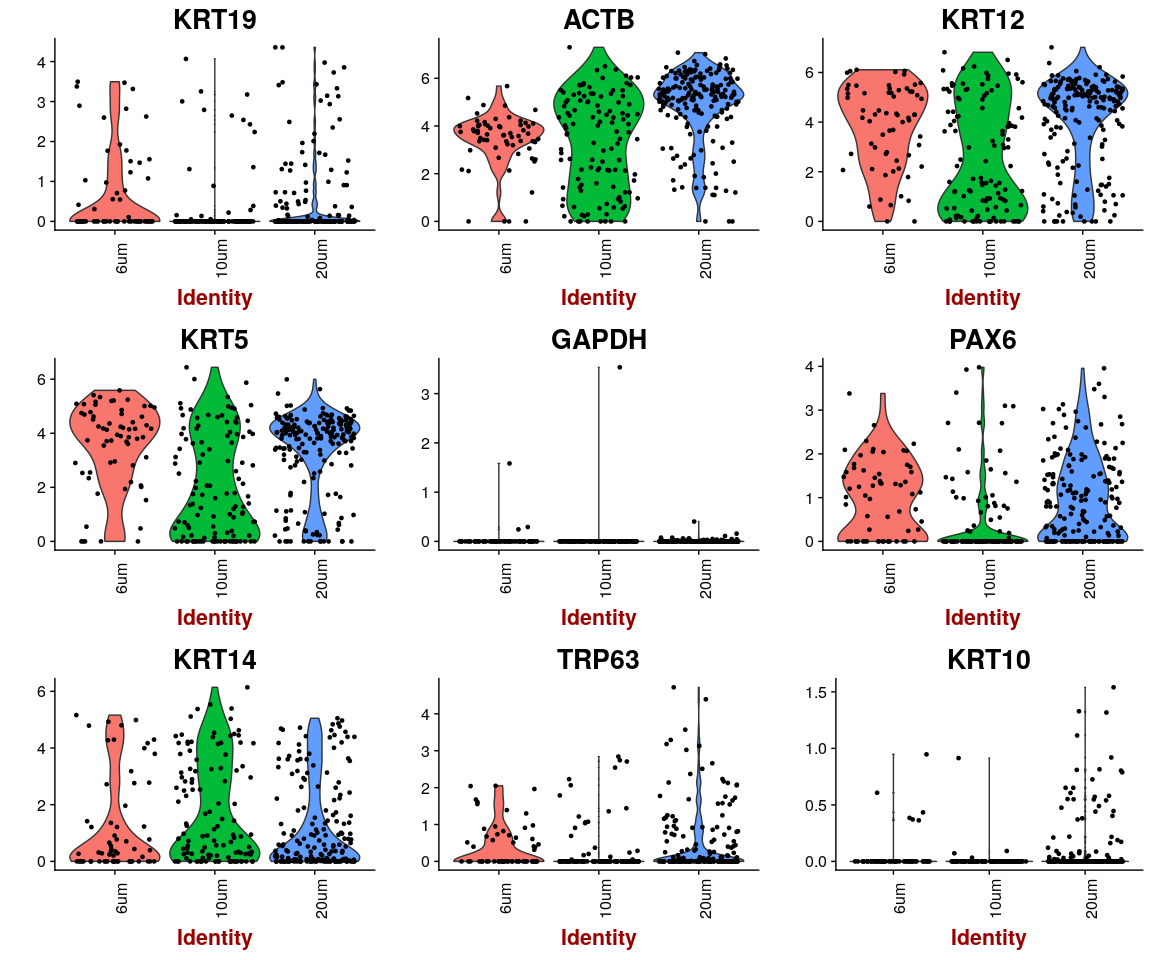
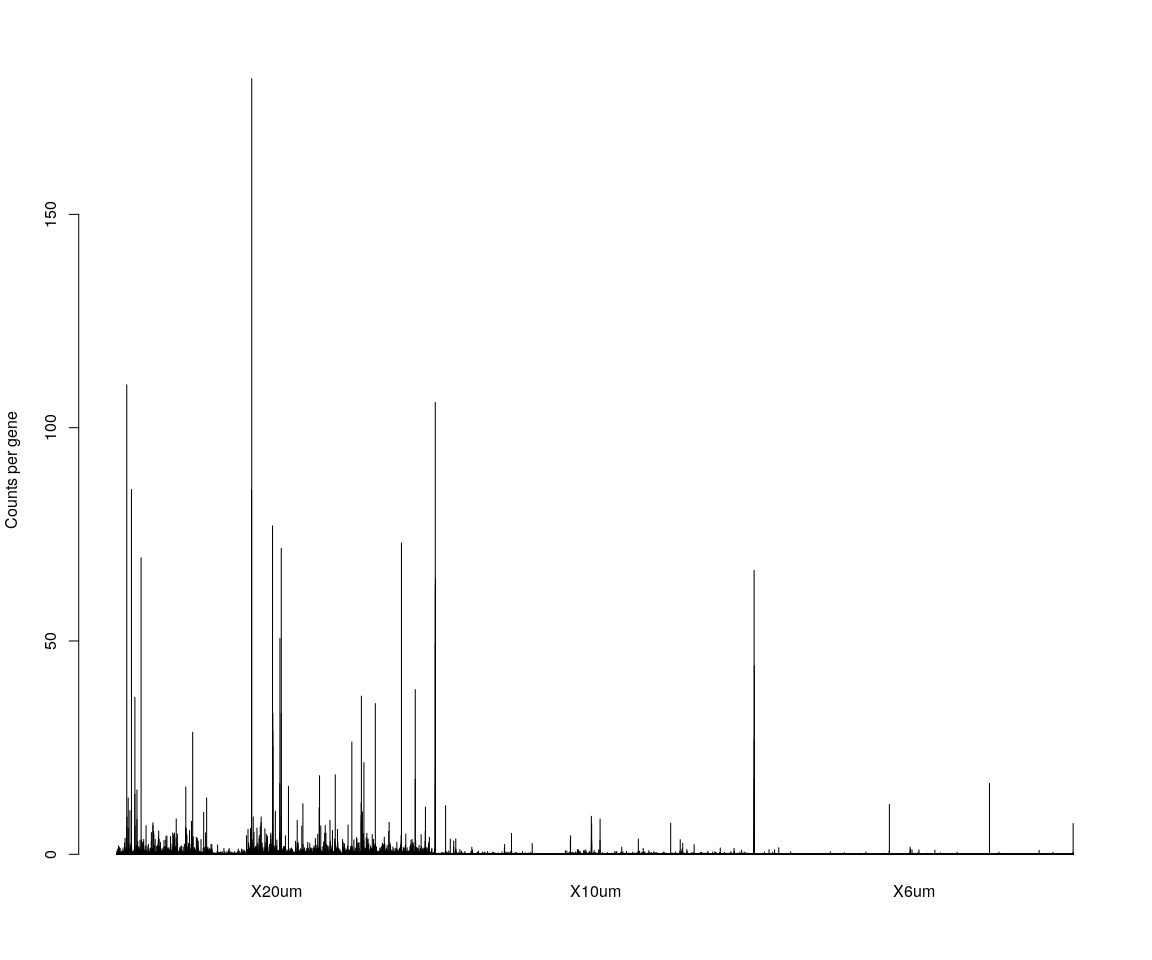
## Figure Explore

### First,use the plot,eg. Barplot,Violin…,we can explore some message from sample

Group\_Bar(Negative.pbmc@raw.data, group = Negative.sample.group)



Group\_Bar(Negative.pbmc@raw.data, group = Negative.sample.cellsize)  
  
VlnPlot(Negative.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Negative.pbmc@raw.data)], y.lab.rot = 90) # Violinn plot of gene ITGB in all sample



## Dimensionality reduction

### **PCA** and **tSNE**

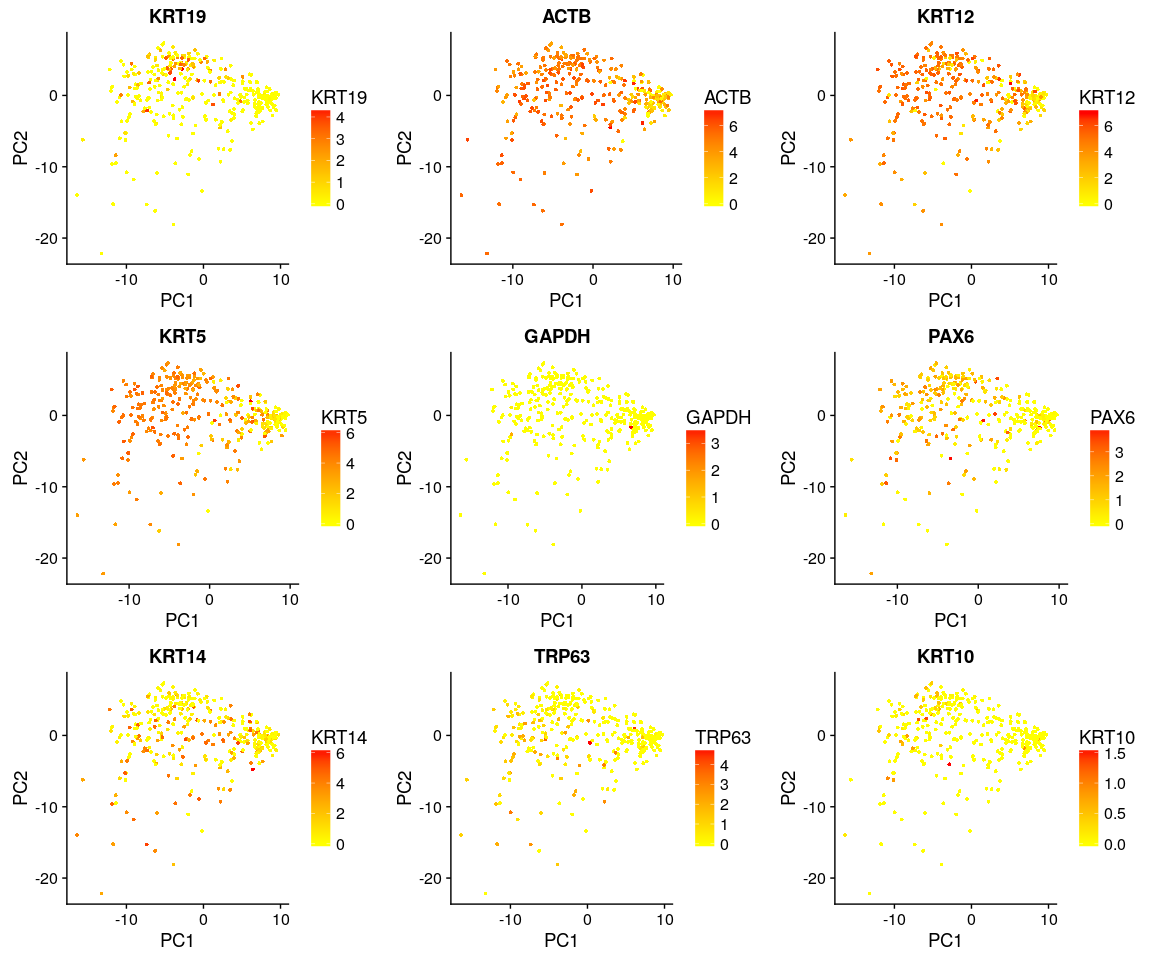
Here,do the dimensionality reduction using the PCA, tSNE method

*It will take a long time to caculate significant pcs.So,here we use the default value*

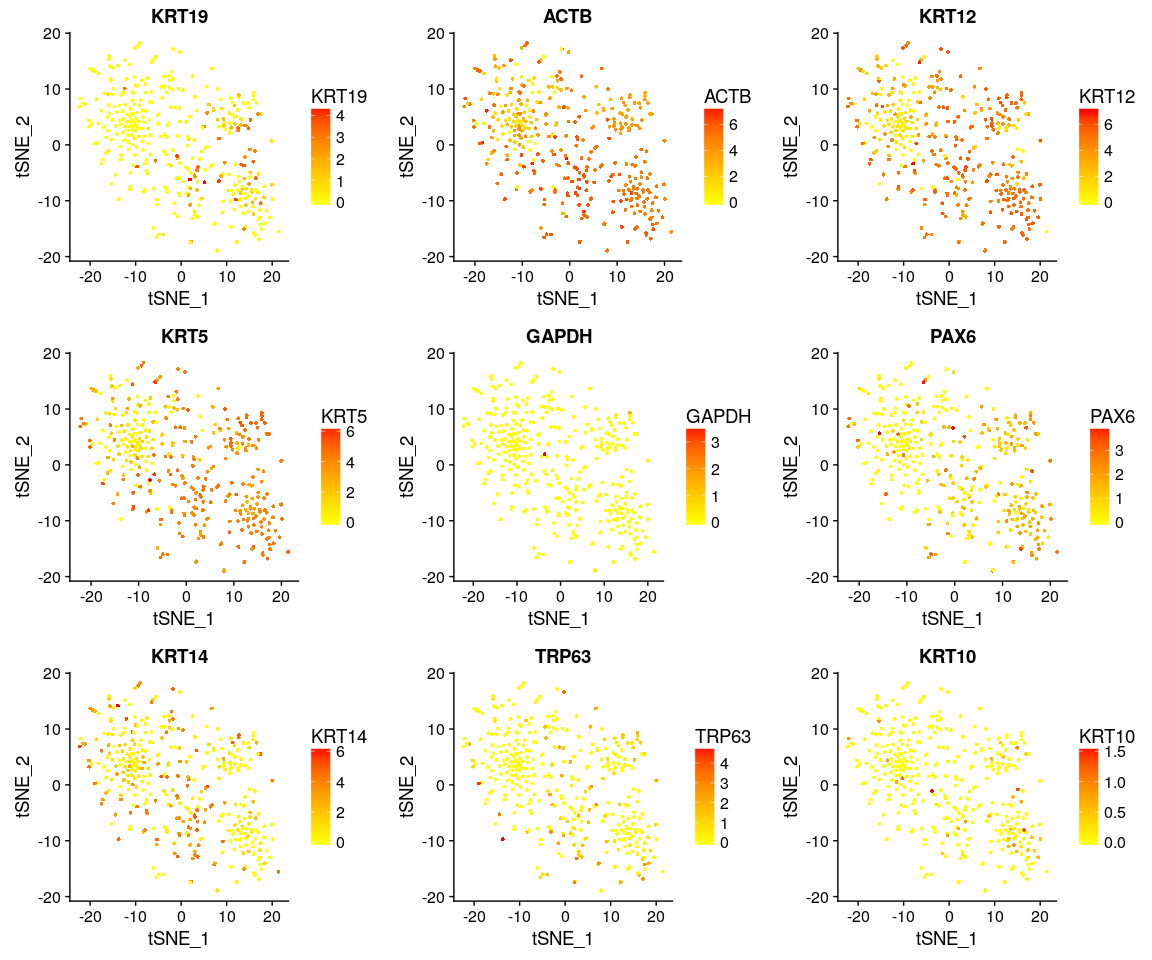
Negative.pbmc <- PCA.TSNE(object = Negative.pbmc, pcs.compute = FALSE, num.pcs = 28)

### After the PCA and tSNE,try plot: Featureplot of **ITGB4**,four var.genes,PCA plot,tSNE plot…

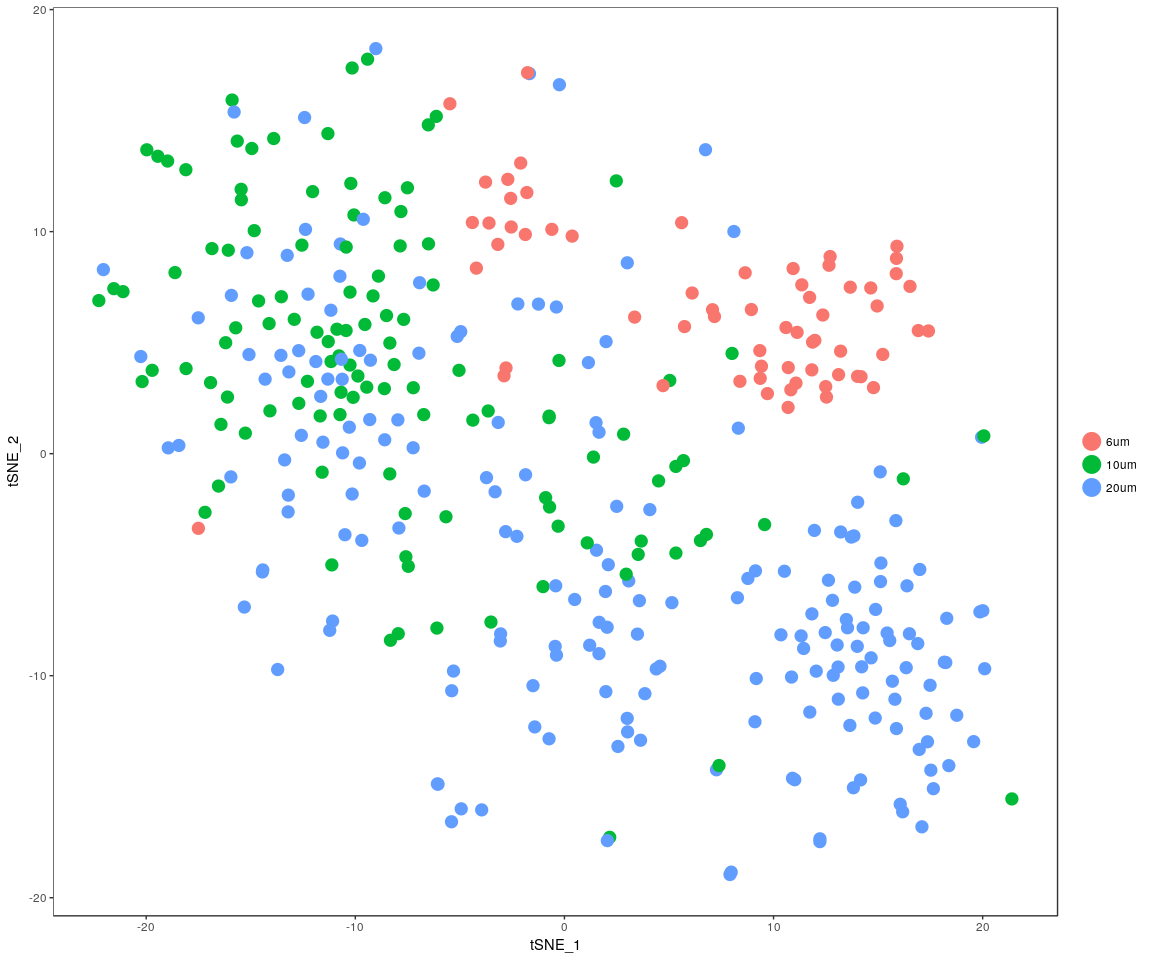
FeaturePlot(object = Negative.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Negative.pbmc@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "pca") # ITGB4 gene in part dataset



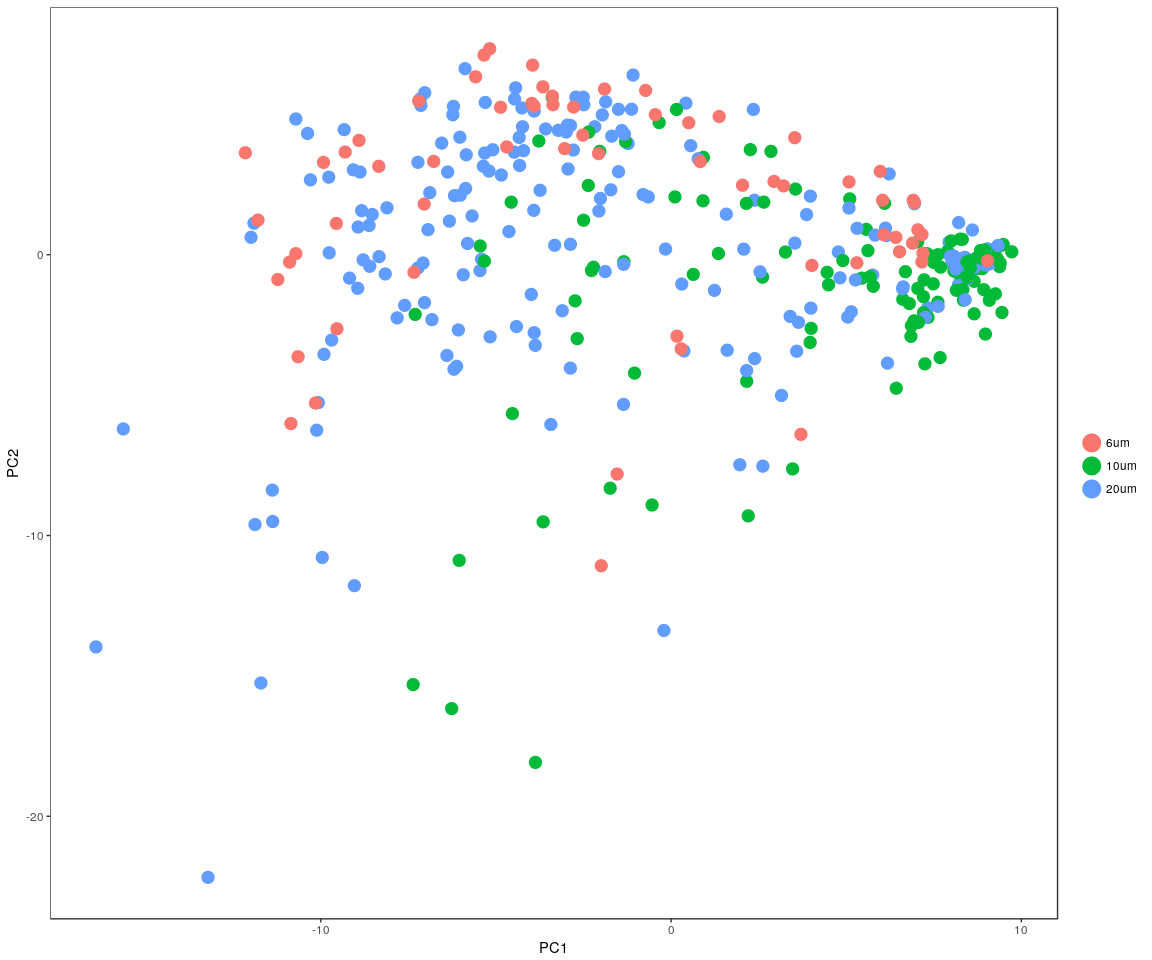
FeaturePlot(object = Negative.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Negative.pbmc@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "tsne") # ITGB4 gene in part dataset



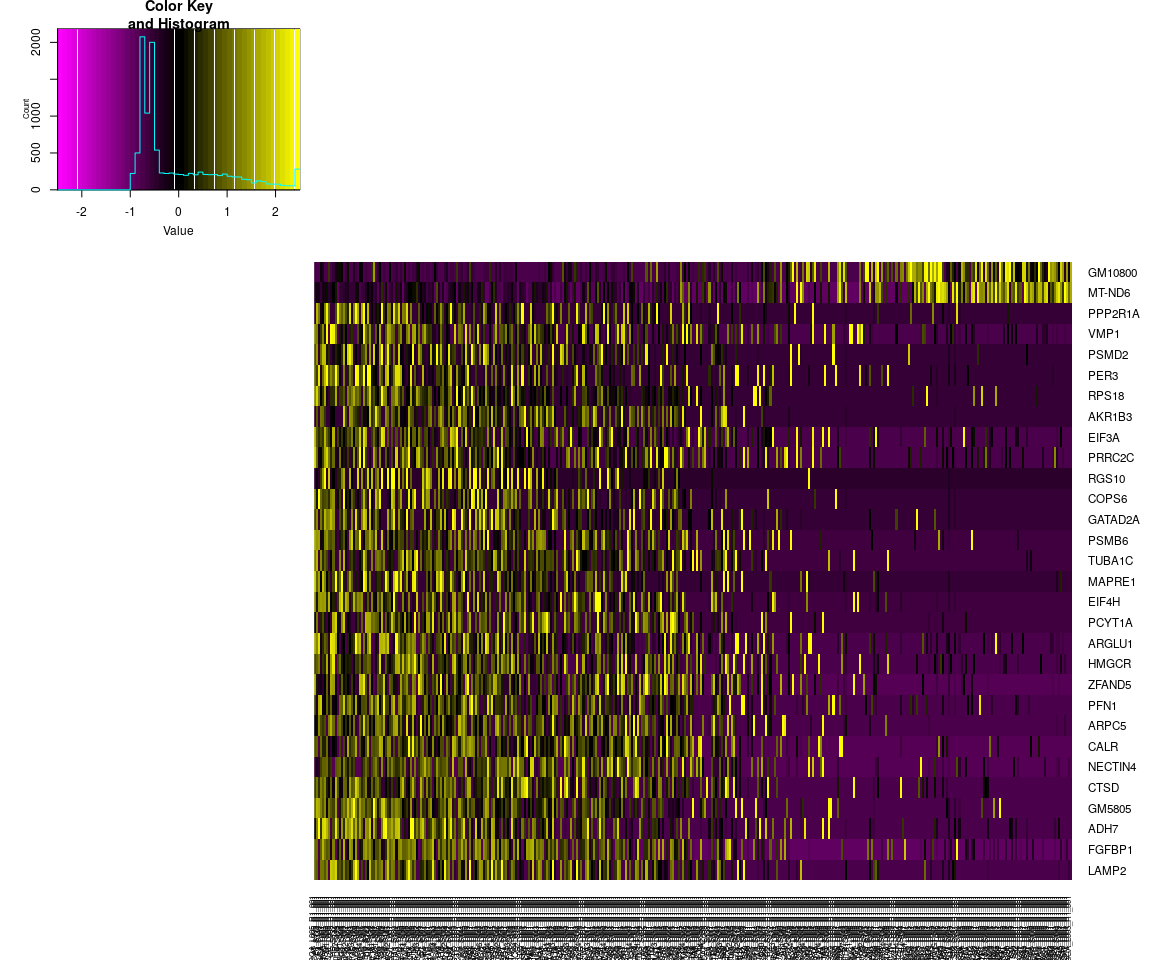
DimPlot(Negative.pbmc, reduction.use = "tsne", pt.size = 4) # grour by sample



DimPlot(Negative.pbmc, reduction.use = "pca", pt.size = 4) # grour by sample



DimHeatmap(Negative.pbmc, reduction.type = "pca", check.plot = FALSE)



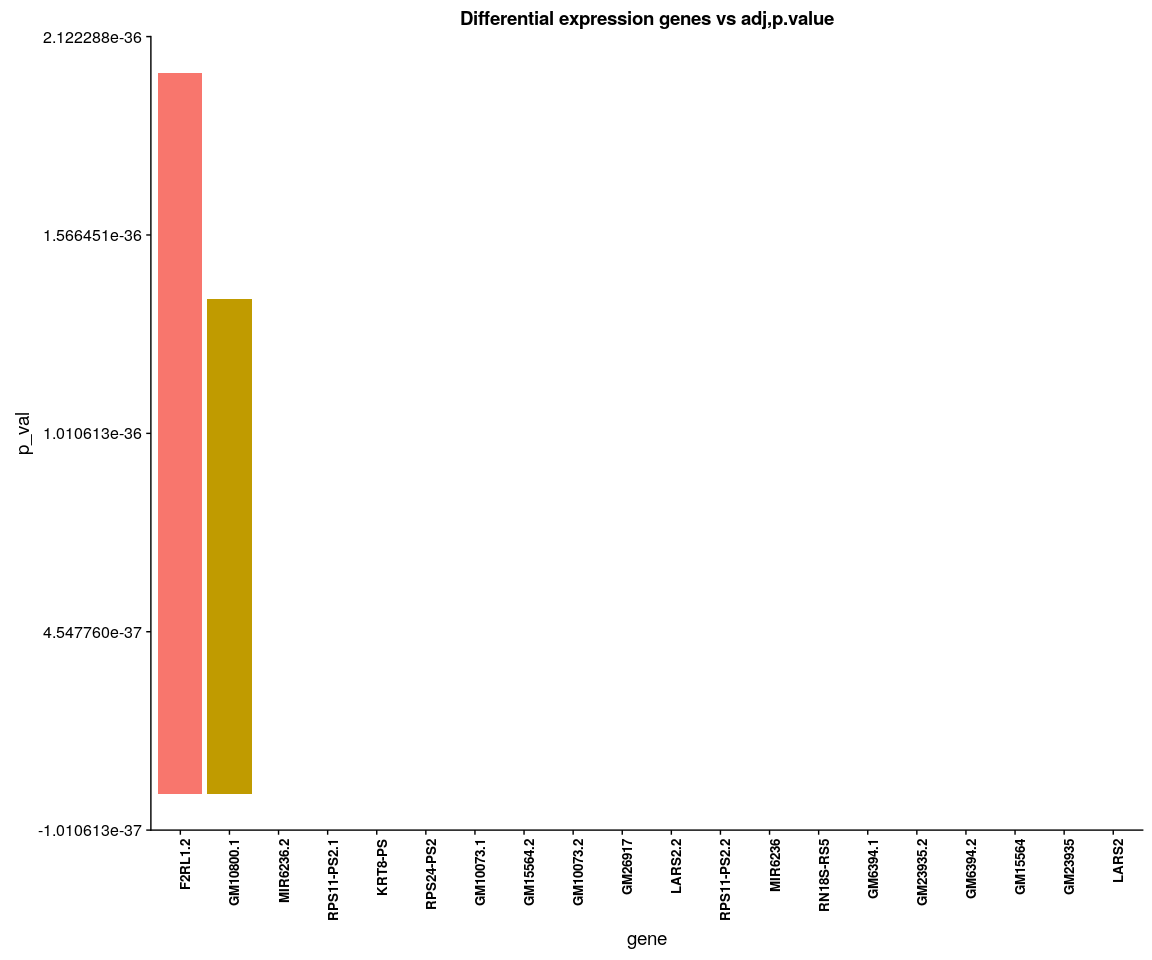
## Differential expression

Next,we will have analysis on gene differential expression.Find maker genes across sample.We use the method: \*\*wilcox test\*\*

# Finds markers (differentially expressed genes) for each of the identity  
# classes in a dataset  
Negative.markers <- FindAllMarkers(Negative.pbmc, test.use = "bimod", print.bar = FALSE)  
head(Negative.markers)

## p\_val avg\_logFC pct.1 pct.2 p\_val\_adj cluster  
## LARS2 2.749269e-111 4.376151 0.984 0.654 2.968111e-107 6um  
## GM23935 3.571136e-108 1.931945 0.781 0.269 3.855398e-104 6um  
## GM15564 1.915601e-87 3.928626 0.938 0.602 2.068082e-83 6um  
## RN18S-RS5 1.071081e-66 2.924028 1.000 0.919 1.156339e-62 6um  
## MIR6236 8.262896e-65 4.354583 0.969 0.110 8.920623e-61 6um  
## GM26917 3.809505e-59 3.290449 0.938 0.392 4.112742e-55 6um  
## gene  
## LARS2 LARS2  
## GM23935 GM23935  
## GM15564 GM15564  
## RN18S-RS5 RN18S-RS5  
## MIR6236 MIR6236  
## GM26917 GM26917

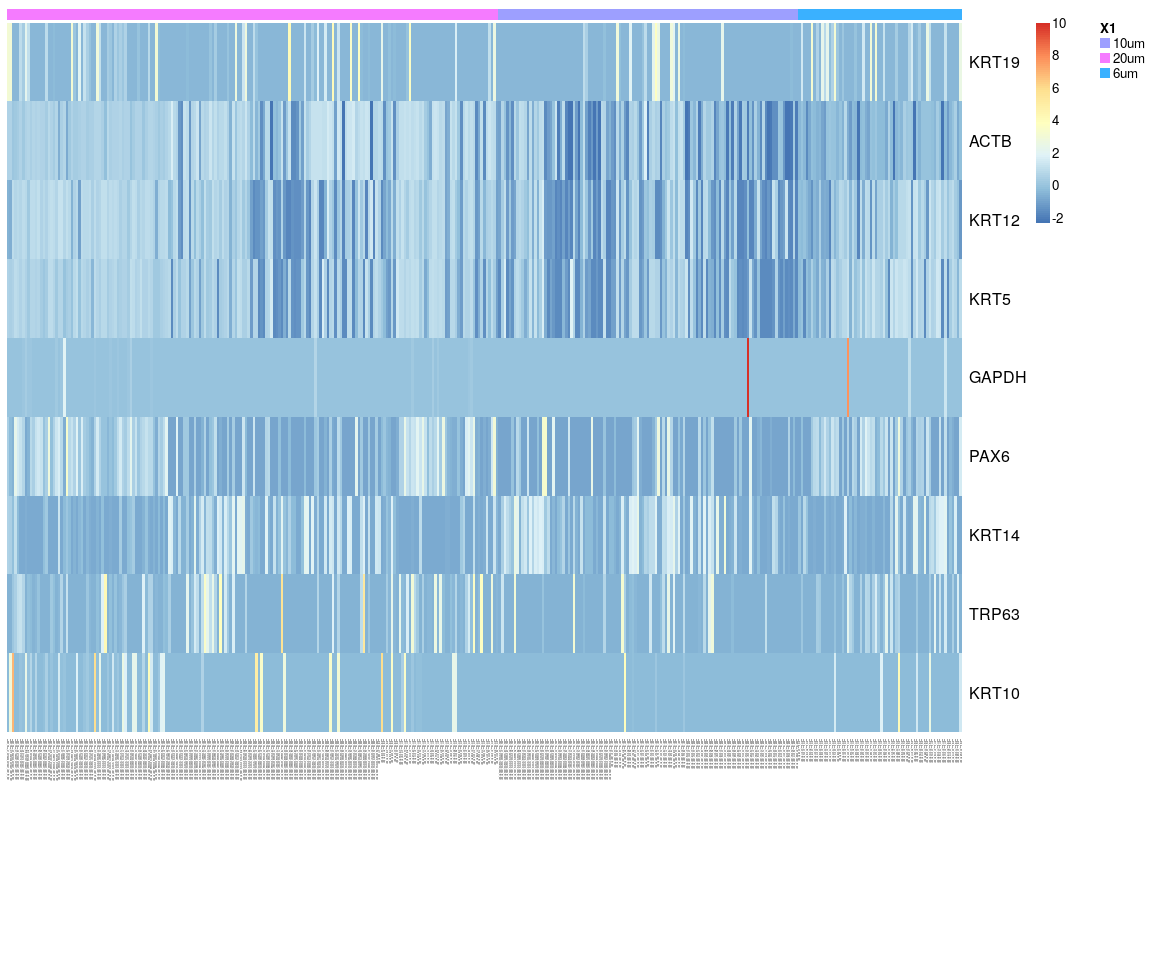
### Bar plot of gene’s p.val



Negative.heatmap <- Heatmap\_fun(genes = important.genes[important.genes %in%   
 rownames(Negative.pbmc@raw.data)], tpm.data = Negative.pbmc@scale.data,   
 condition = unique(as.character(Negative.pbmc@ident)), all.condition = as.character(Negative.pbmc@ident))

## There ara 3 conditions  
## Whether creat data accurate 0

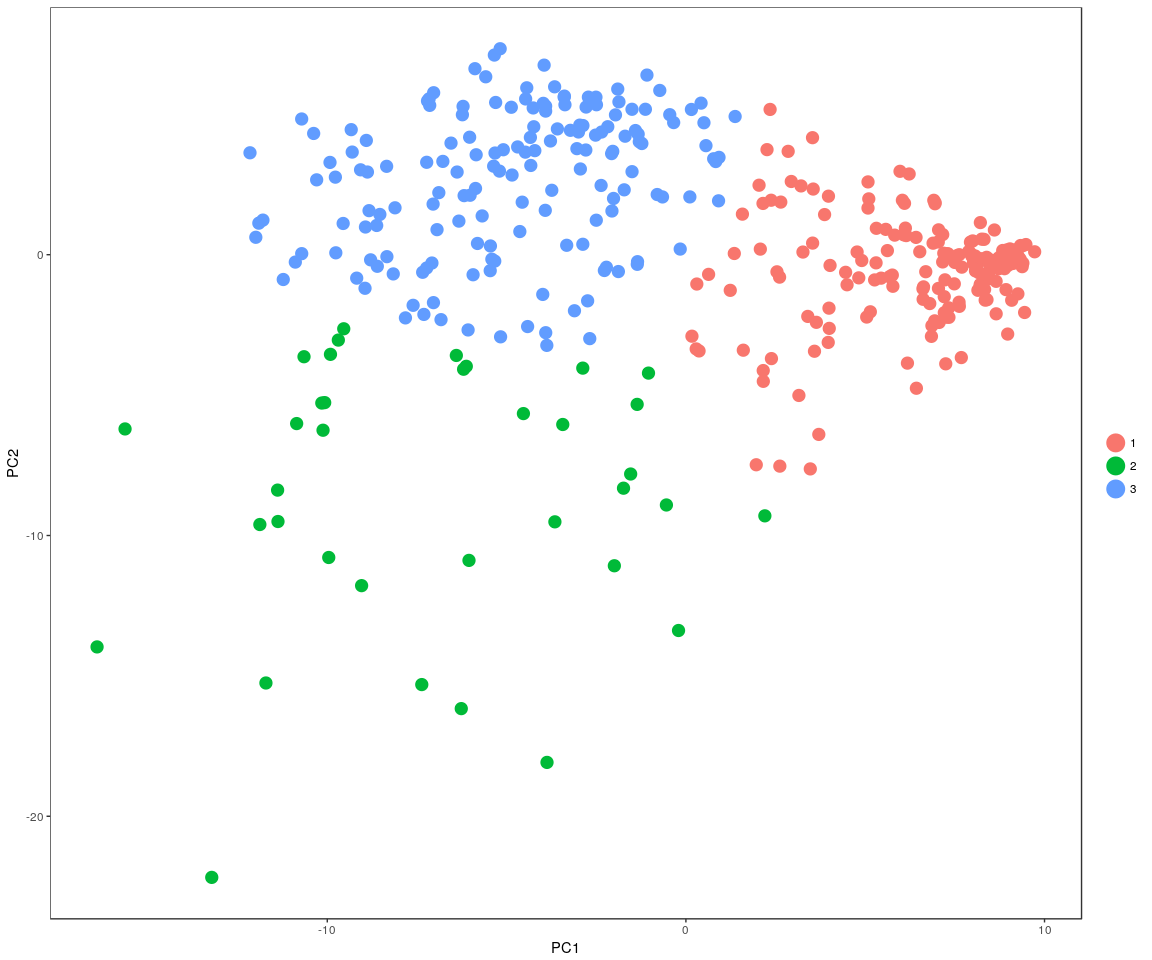
NMF::aheatmap(Negative.heatmap[[2]], Rowv = NA, Colv = NA, annCol = Negative.heatmap[[1]],   
 scale = "none")



We have find all marker genes across sample,there are 3765 significant genes(adjust p-value <0.05) in all marker genes.

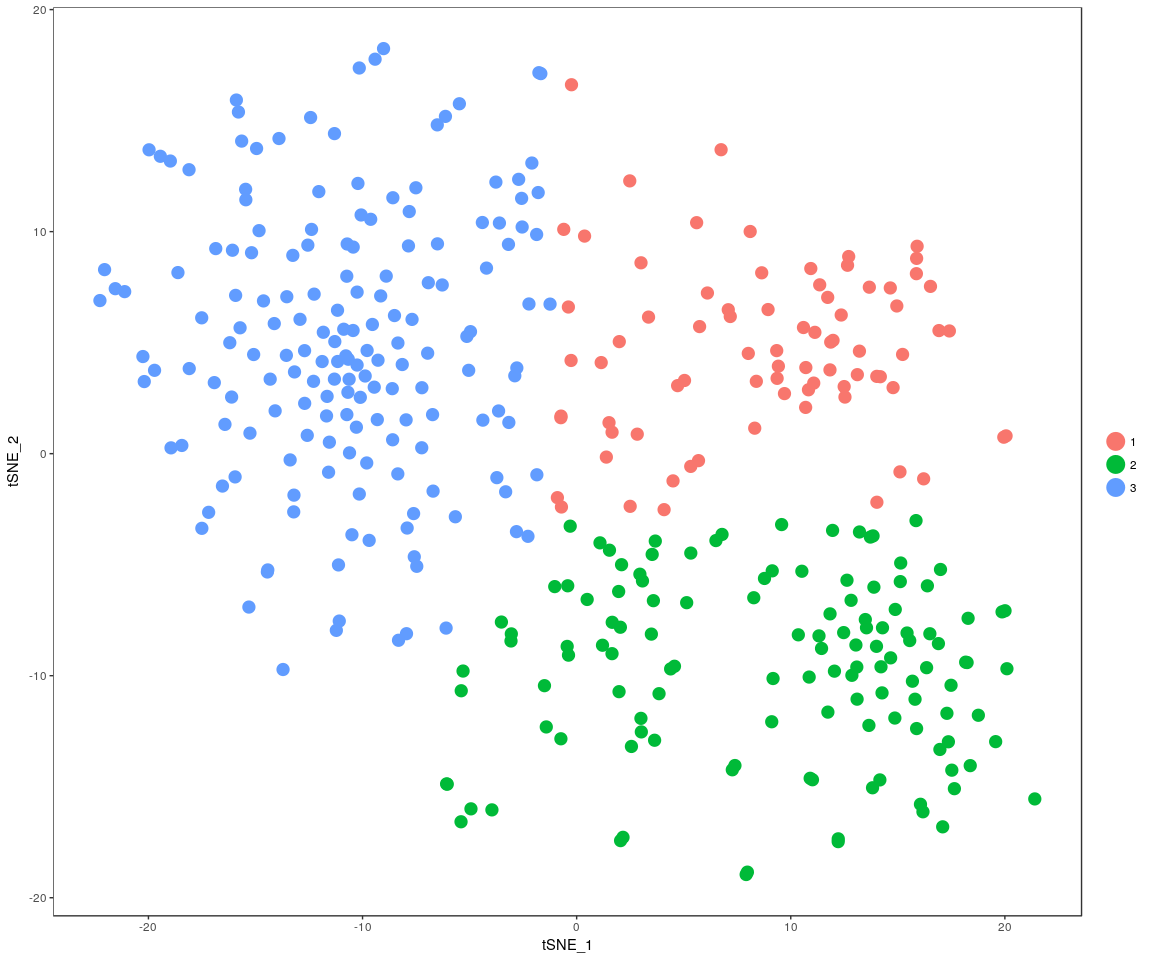
### Next,Spectral k-means clustering on single cells based on PCA

Negative.pbmc <- KClustDimension(Negative.pbmc, reduction.use = "pca", k.use = 3)  
clusters.pca <- Negative.pbmc@meta.data$kdimension.ident  
DimPlot(Negative.pbmc, pt.size = 4, group.by = "kdimension.ident")



### Spectral k-means clustering on single cells based on tSNE

Negative.pbmc <- KClustDimension(Negative.pbmc, reduction.use = "tsne", k.use = 3)  
clusters.tsne <- Negative.pbmc@meta.data$kdimension.ident  
DimPlot(Negative.pbmc, pt.size = 4, group.by = "kdimension.ident", reduction.use = "tsne")



## Differential expression use DESeq2 packages

When use the DESeq,it must require the gene count matrix satisify that: **every gene contains at least one zero, cannot compute log geometric means**. So have to take another method to handle data,but I do not know whether it is reasonable.Just try!!!

### Positive

condition.p <- unlist(lapply(Positive.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
Positive.xdds <- DESeq\_CT(count.data = Positive.pbmc@raw.data, condition.1 = condition.p)  
plotDispEsts(Positive.xdds, main = "Per-gene Dispersion")

## Do the DESeq test across all cells with sample group.And get all the significant genes between two groups(p.value < 0.05)

Positive.DESeqGenes <- DESeq\_result(Positive.xdds, condition = condition.p)

Positive.DESeqGenes.v <- as.vector(Positive.DESeqGenes)  
library(VennDiagram)  
grid.draw(venn.diagram(Positive.DESeqGenes.v[1:3], filename = NULL, fill = c("dodgerblue",   
 "goldenrod1", "darkorange1")))

### Negative

condition.n <- unlist(lapply(Negative.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
Negative.xdds <- DESeq\_CT(count.data = Negative.pbmc@raw.data, condition.1 = condition.n)  
plotDispEsts(Negative.xdds, main = "Per-gene Dispersion")

## Do the DESeq test across all cells with sample group.And get all the significant genes between two groups(p.value < 0.05)

Negative.DESeqGenes <- DESeq\_result(Negative.xdds, condition = condition.n)

Negative.DESeqGenes.v <- as.vector(Negative.DESeqGenes)  
library(VennDiagram)  
grid.draw(venn.diagram(Negative.DESeqGenes.v[1:3], filename = NULL, fill = c("dodgerblue",   
 "goldenrod1", "darkorange1")))