Monkey analysis:Positive and Negative(ITGB4)

yejg

2018/2/1

### Load the packages

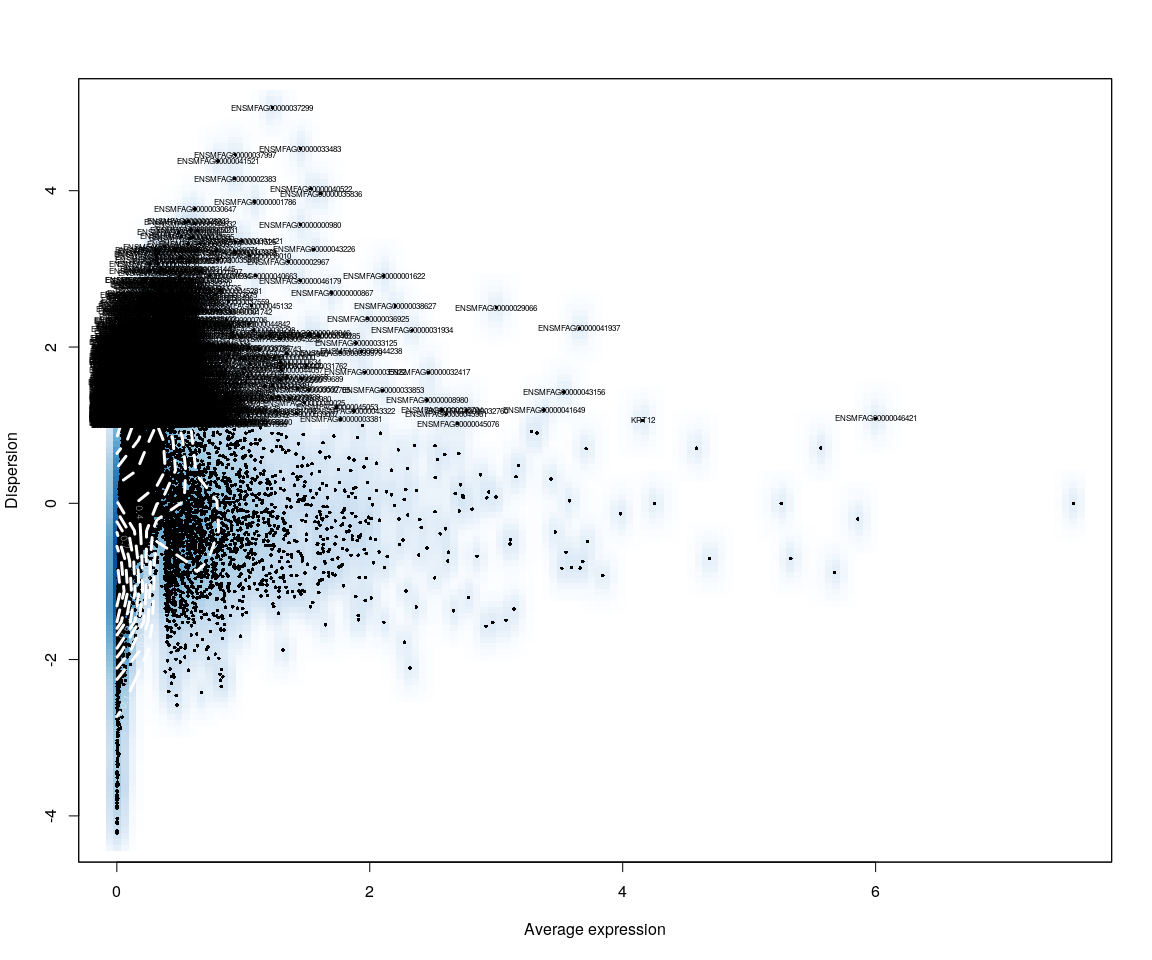
library(Seurat)  
library(data.table)  
library(NMF)  
library(rsvd)  
library(Rtsne)  
library(ggplot2)  
library(cowplot)  
library(sva)  
library(igraph)  
library(cccd)  
library(KernSmooth)  
library(beeswarm)  
library(stringr)  
library(formatR)  
source("../tools.R")  
library(DESeq2)

### Read data

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

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

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



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

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

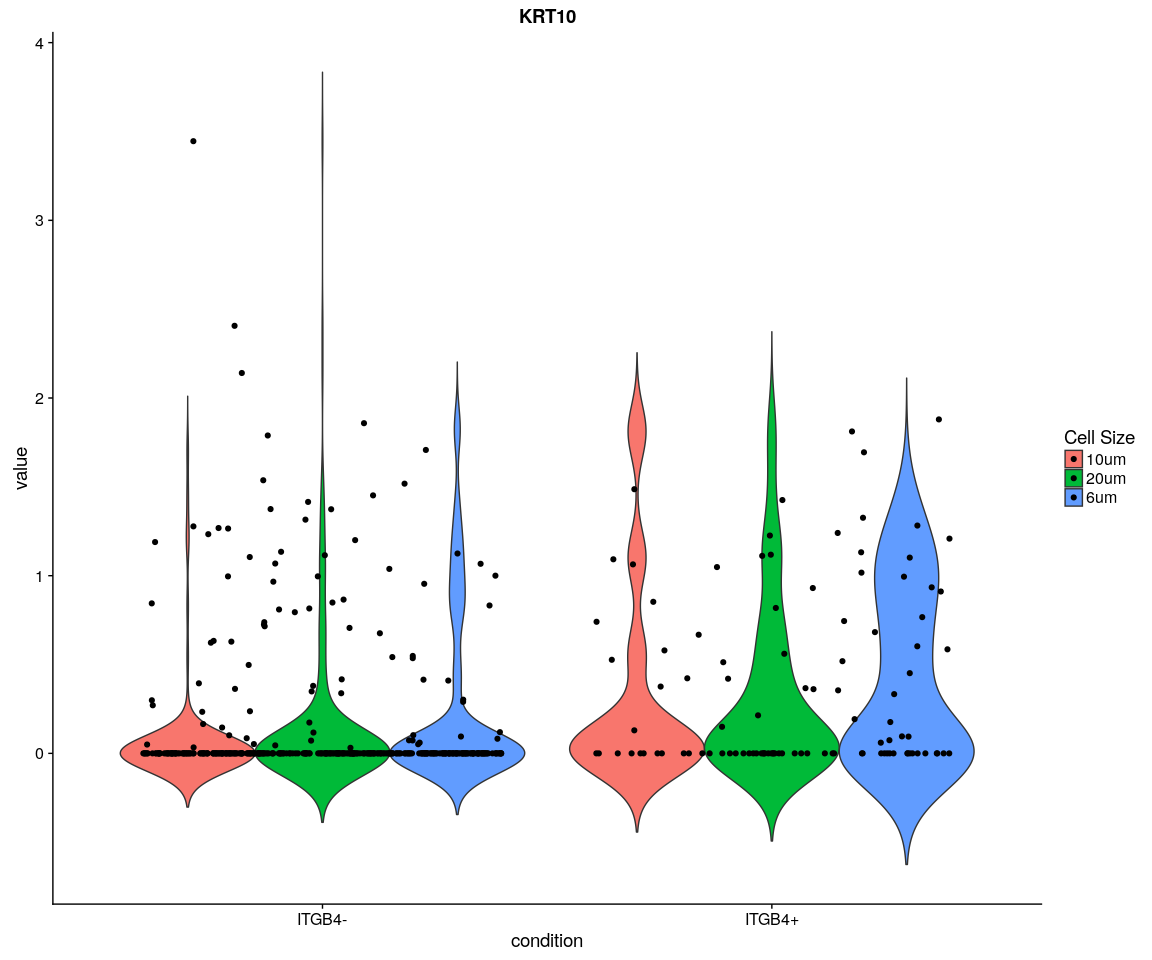
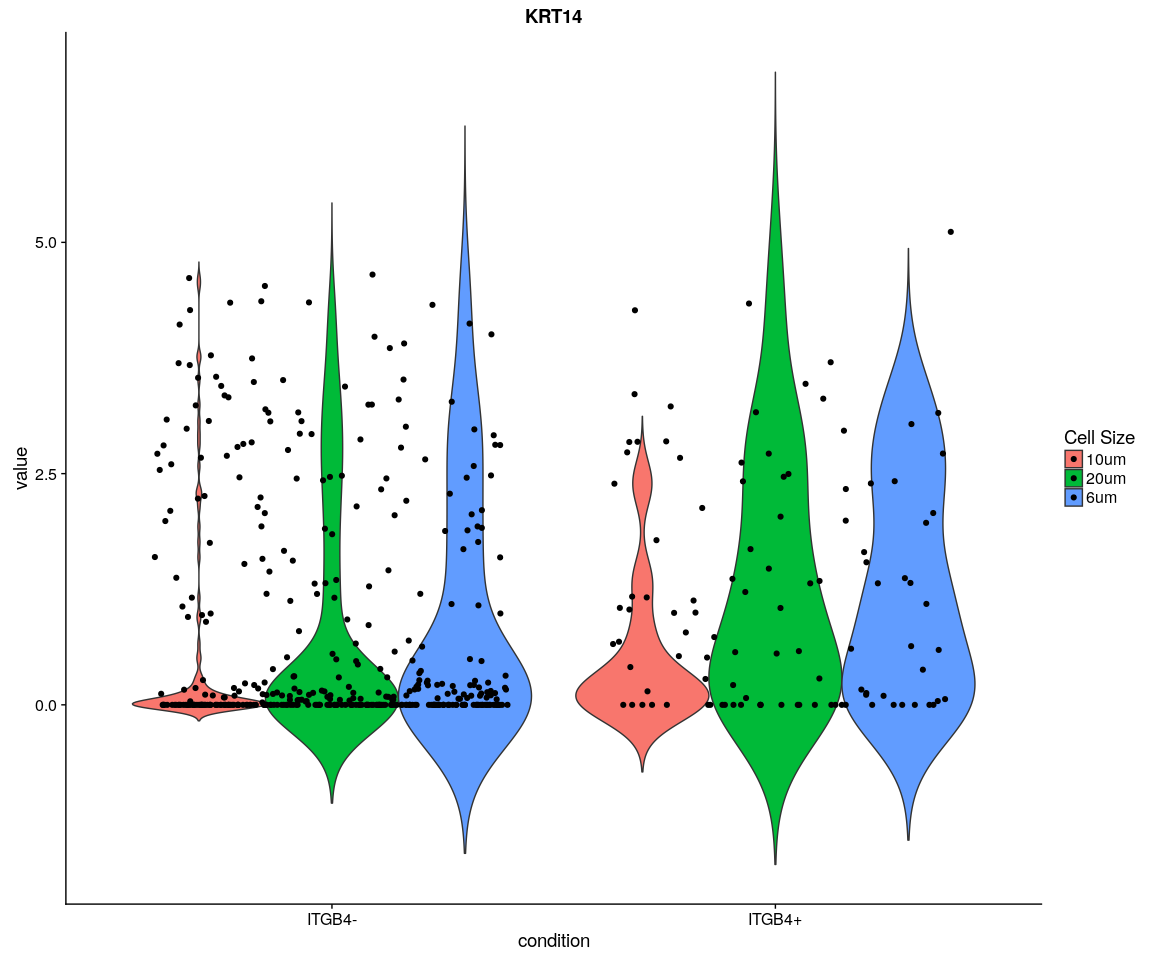
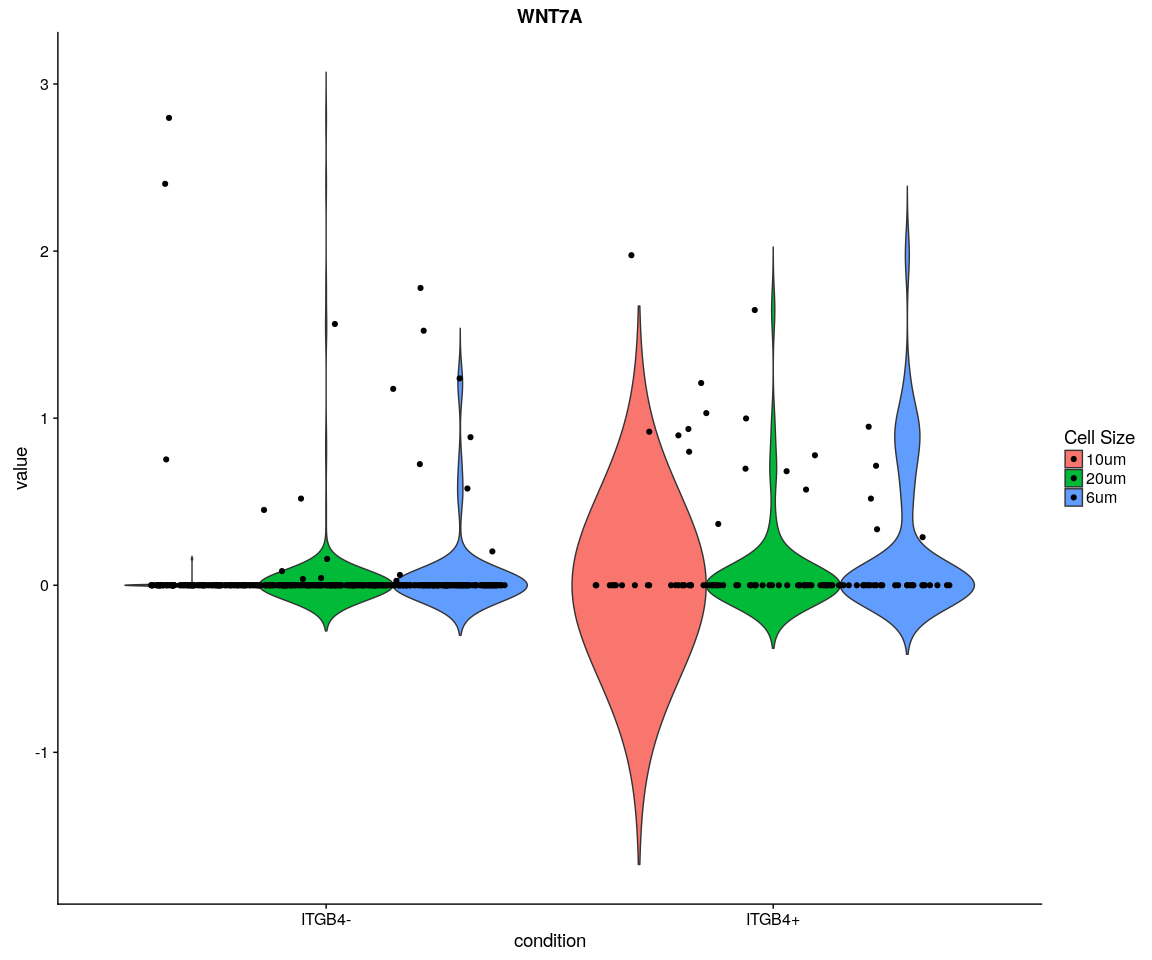
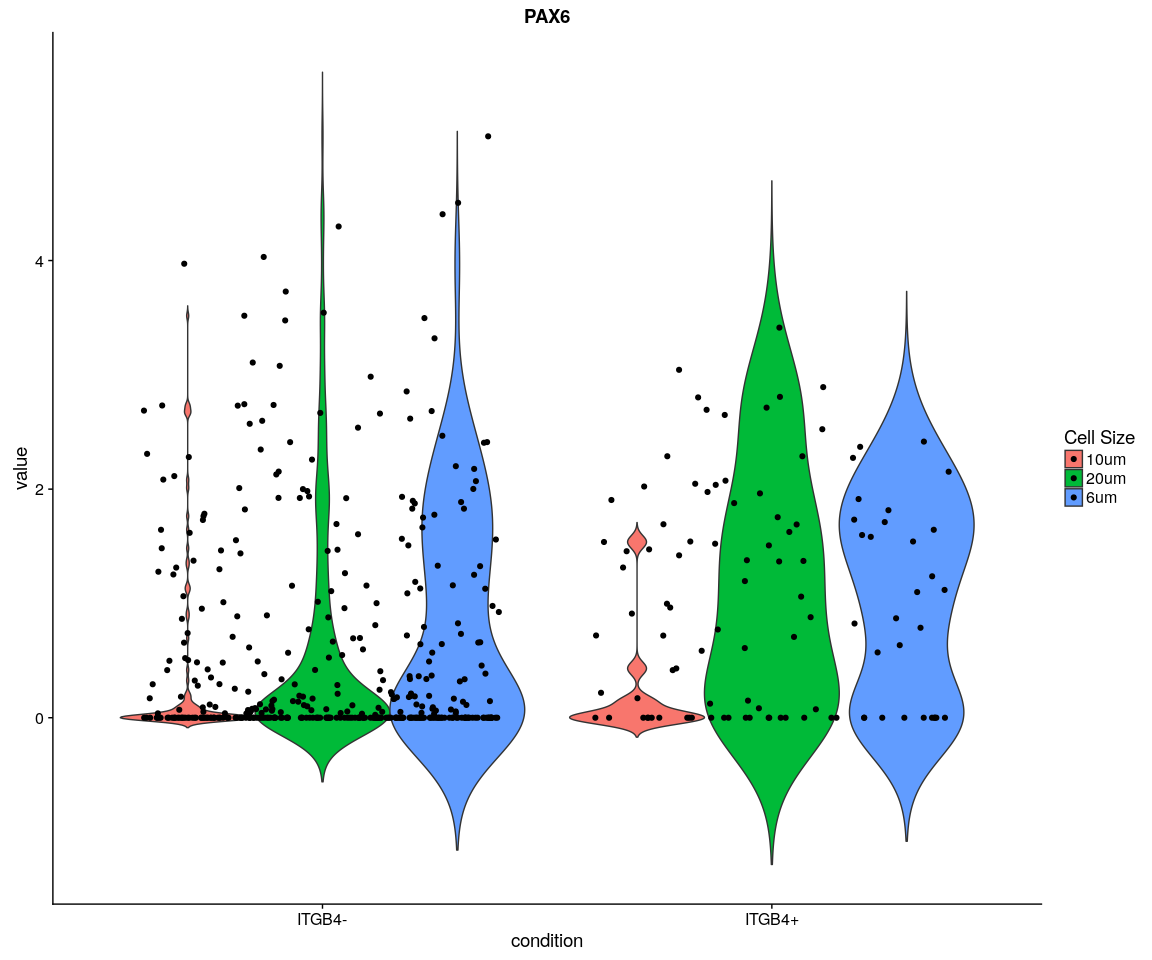
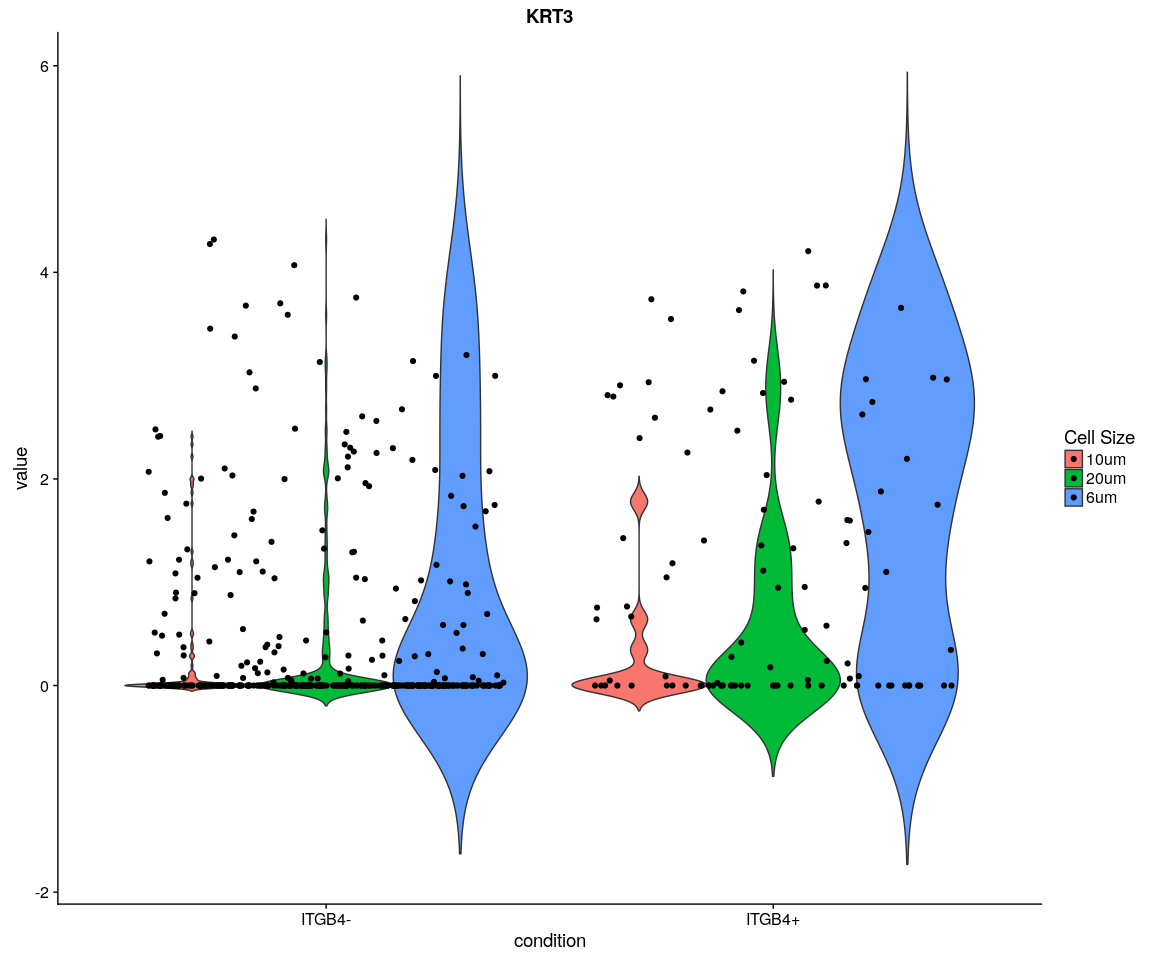
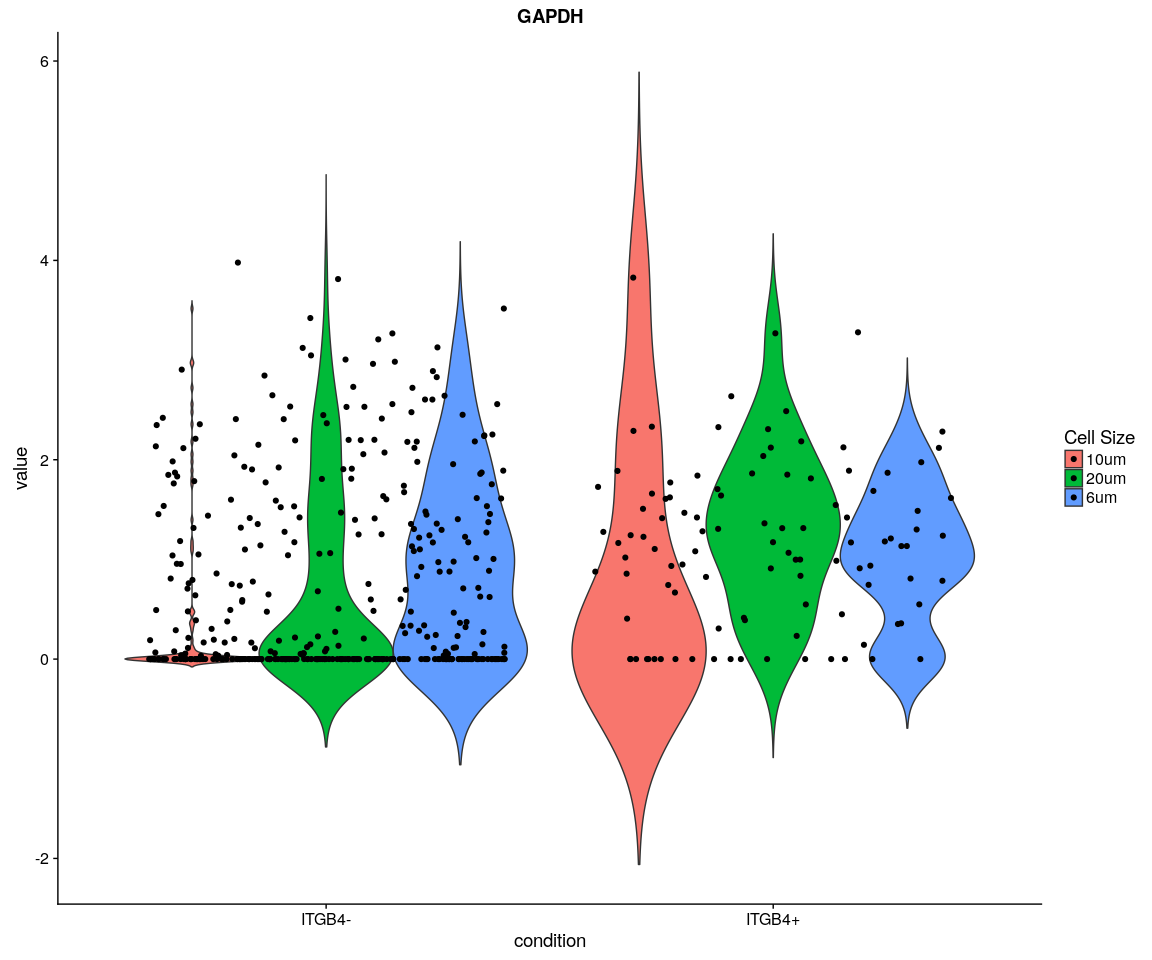
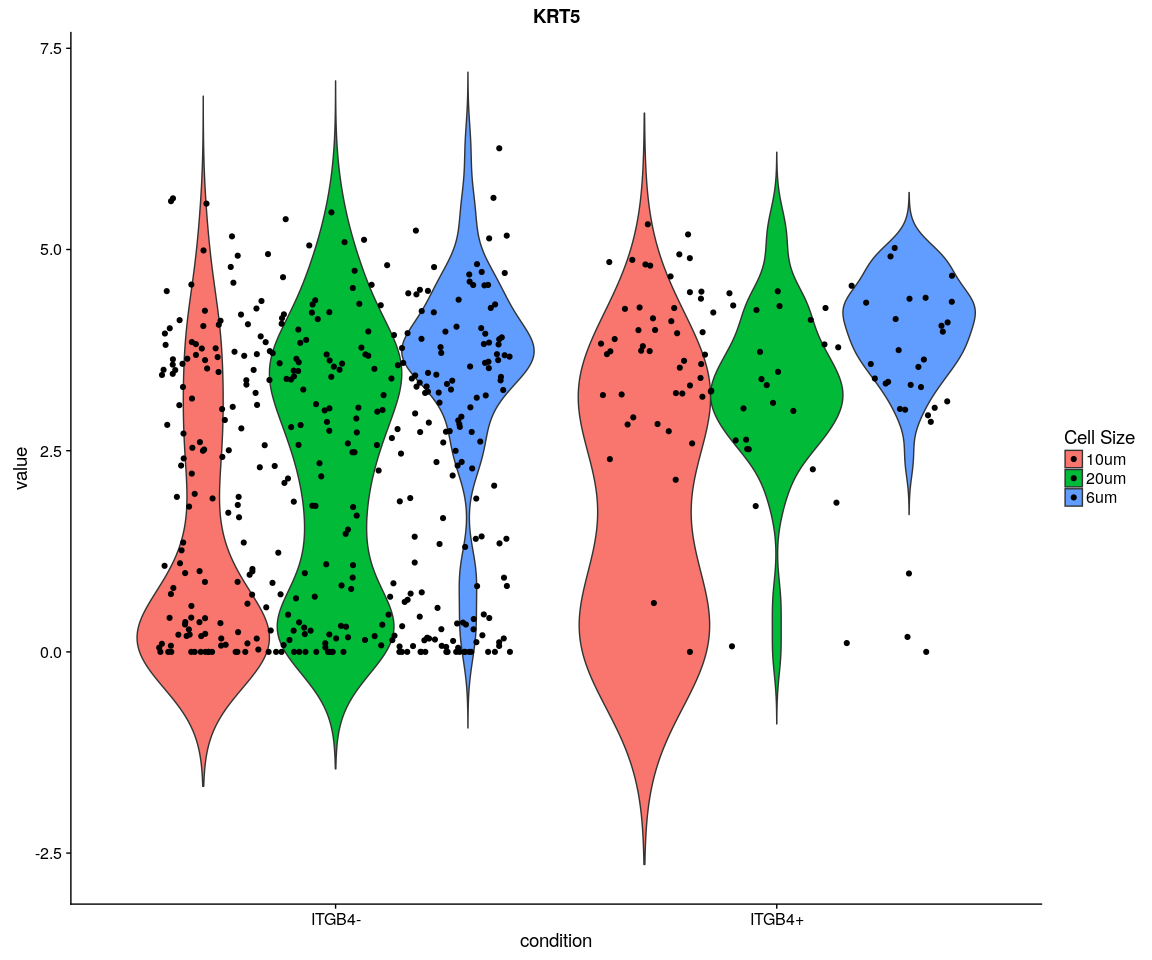
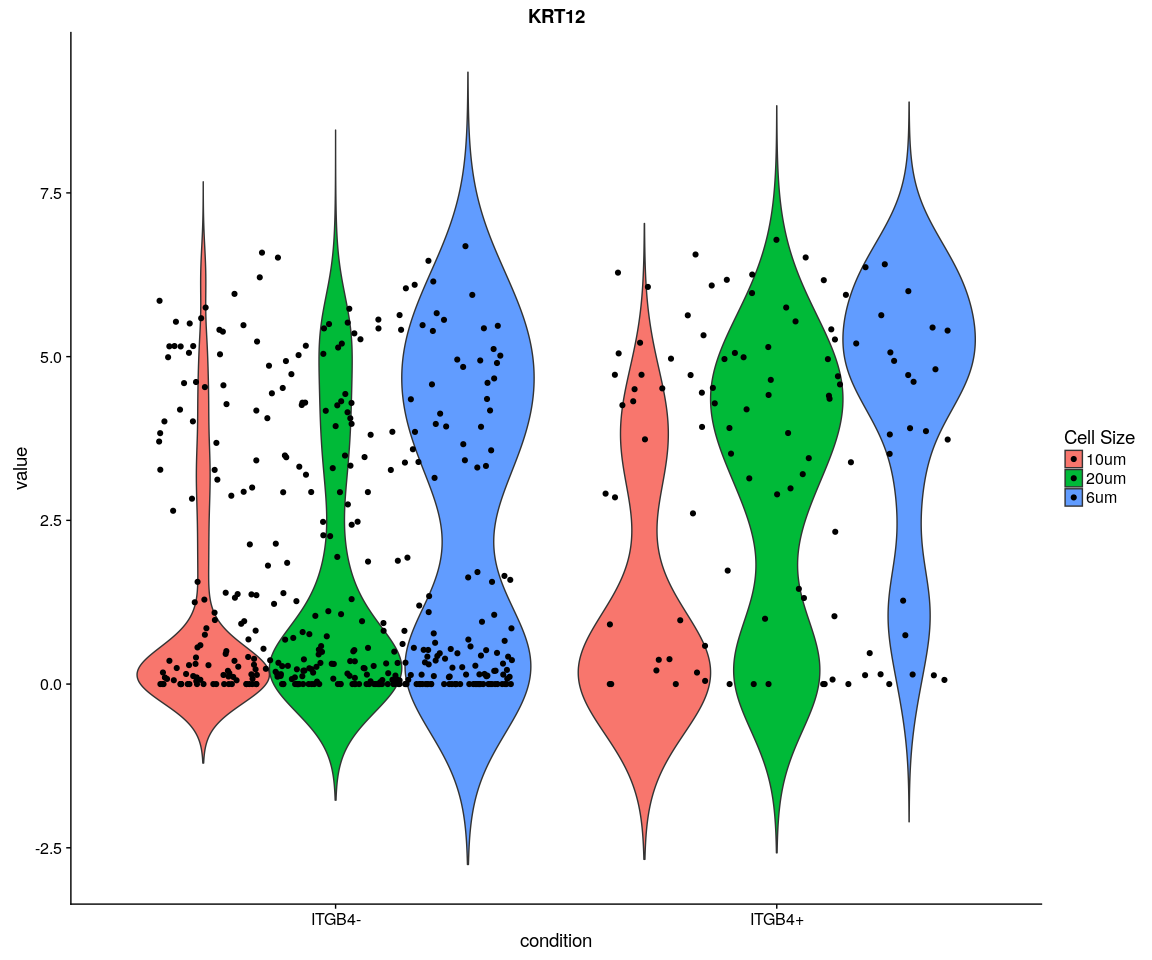
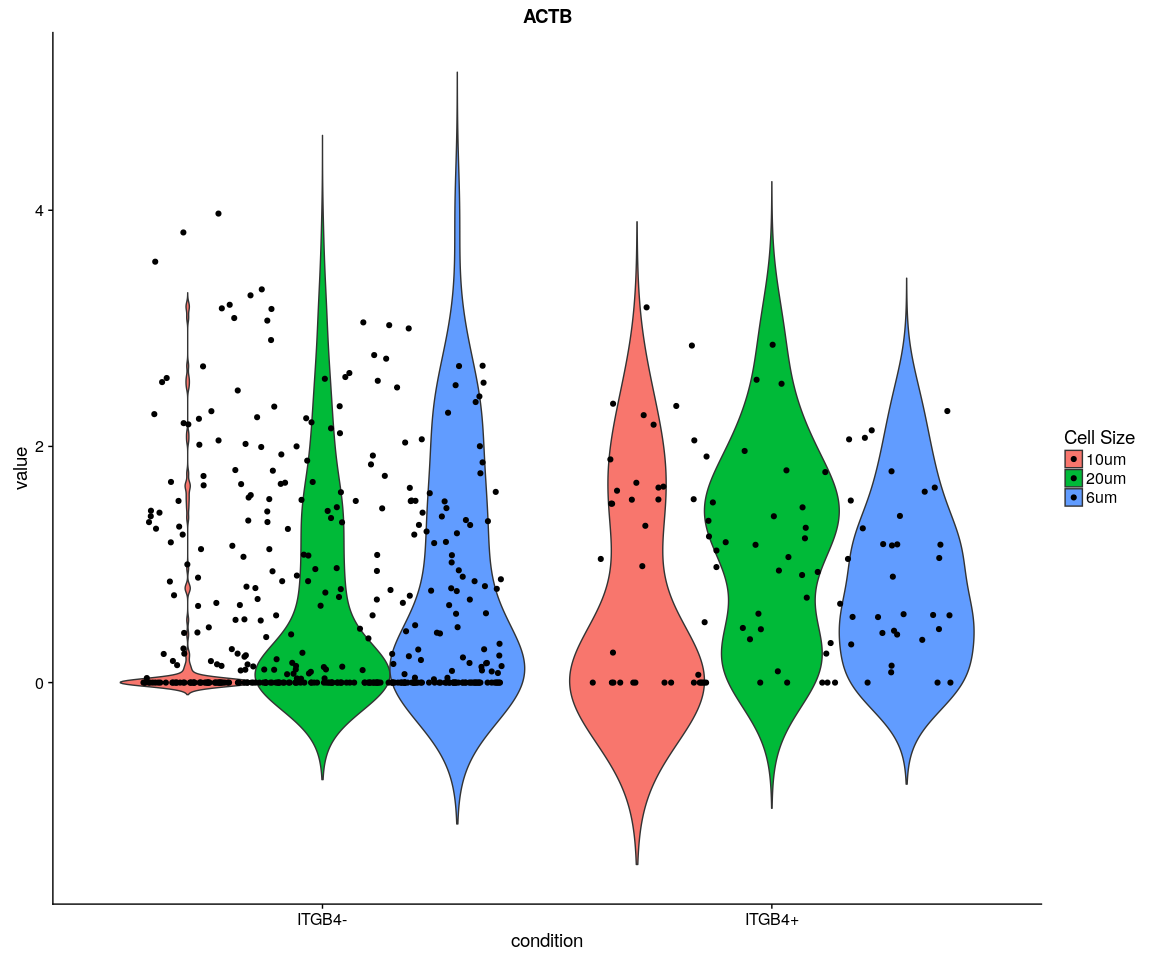
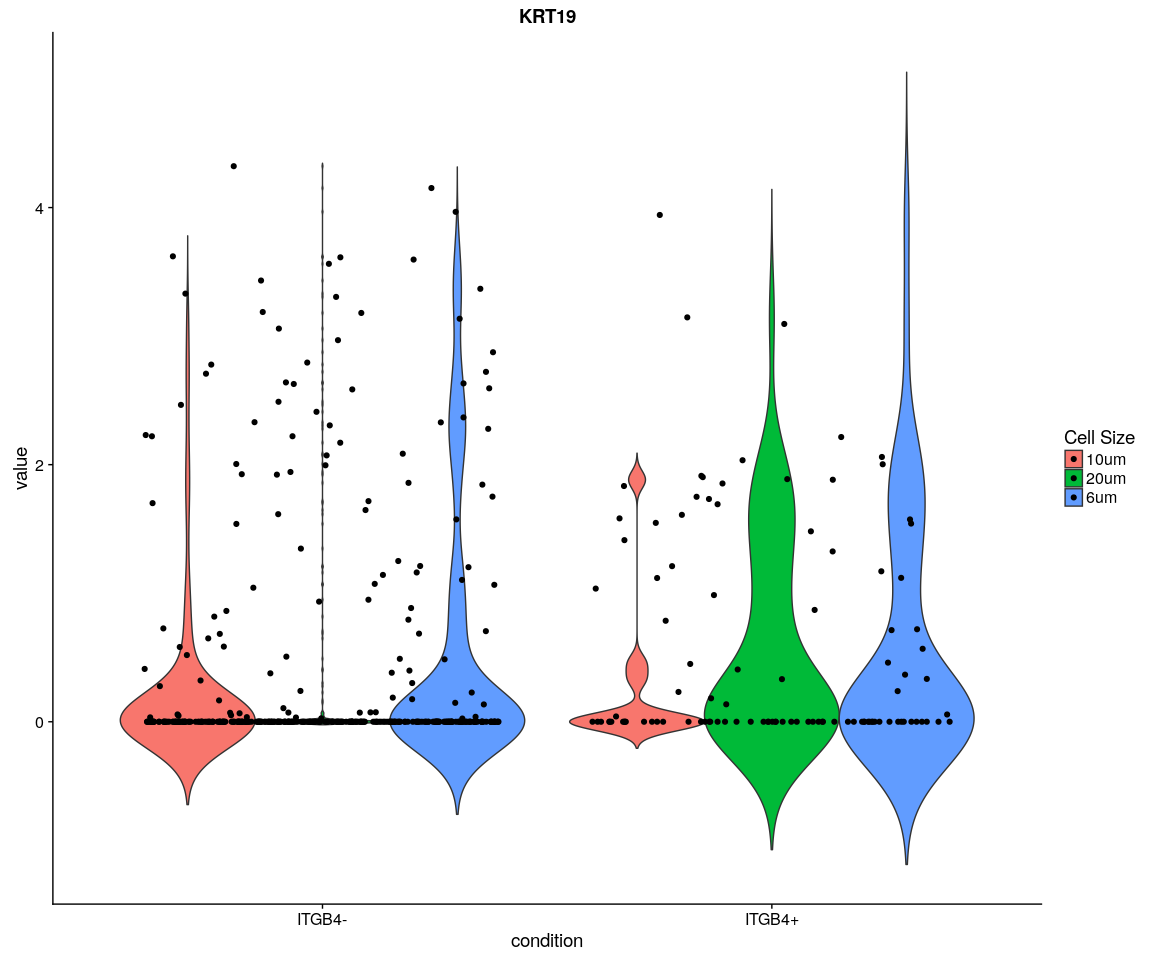
## Figure Explore.1

# 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

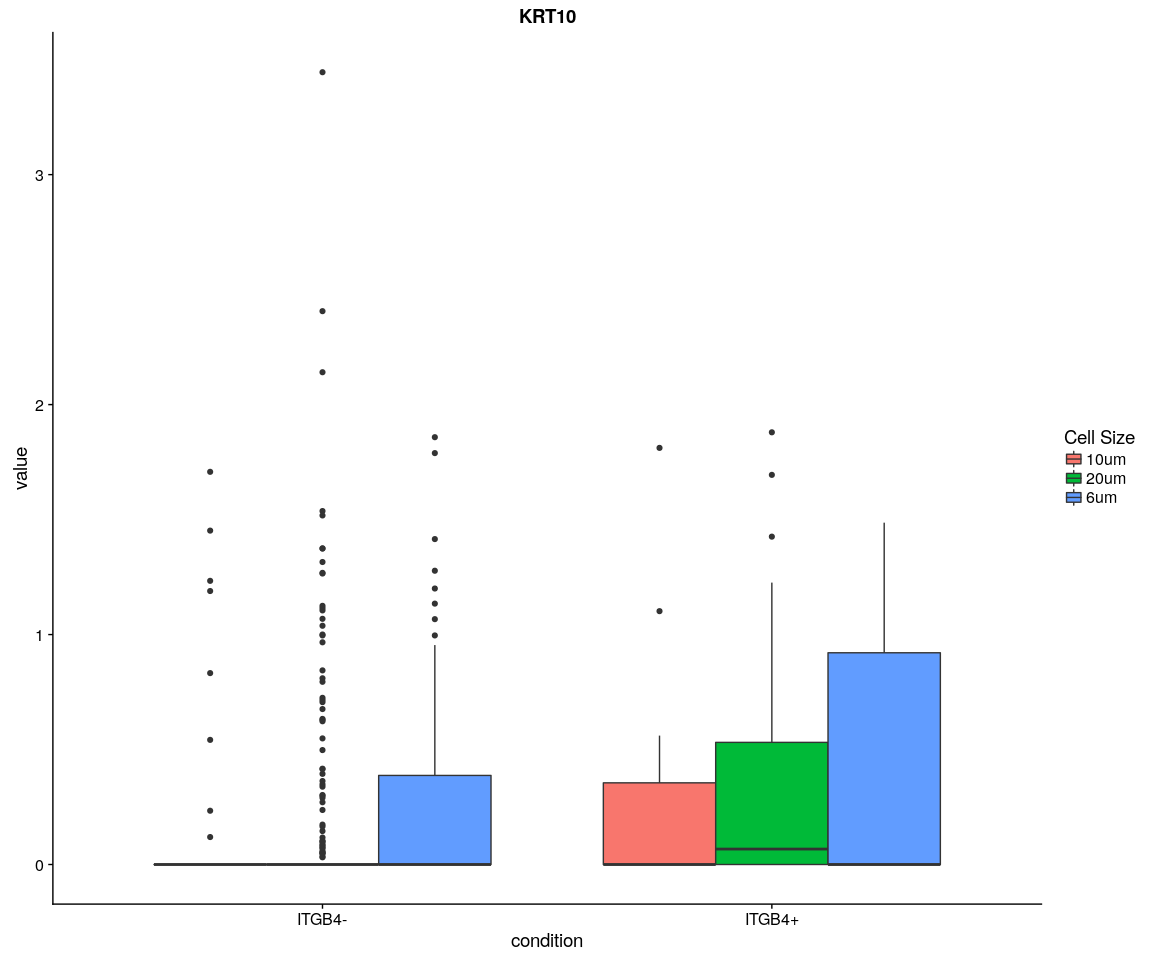
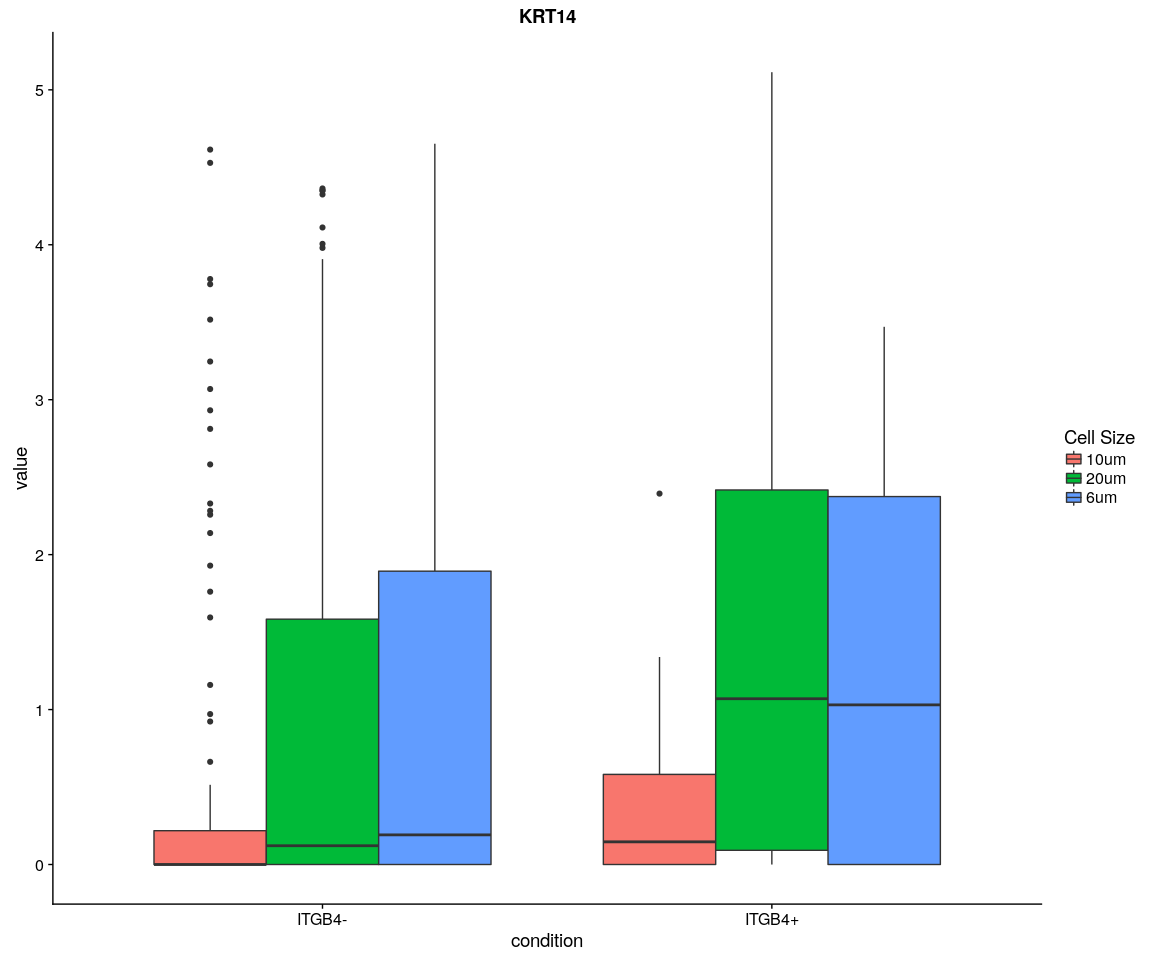
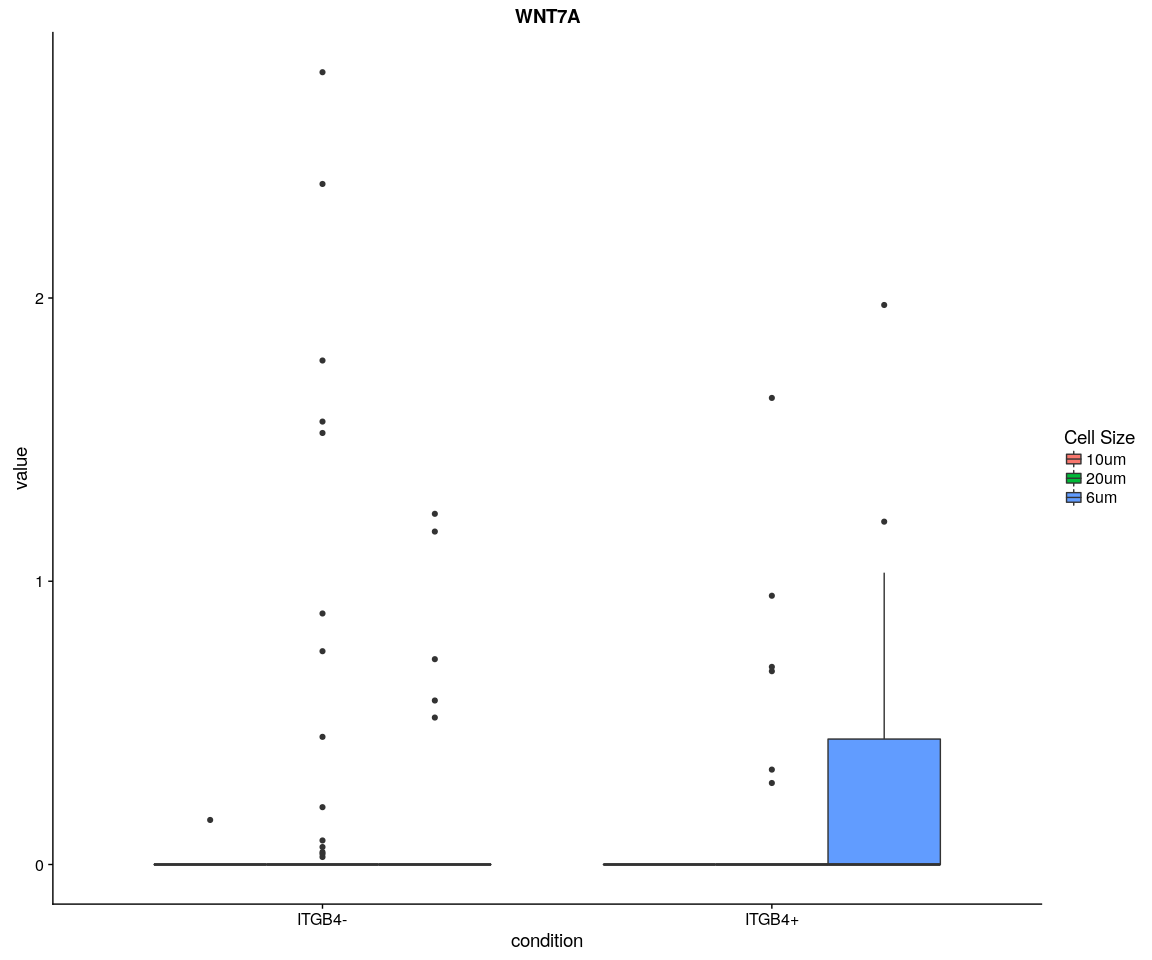
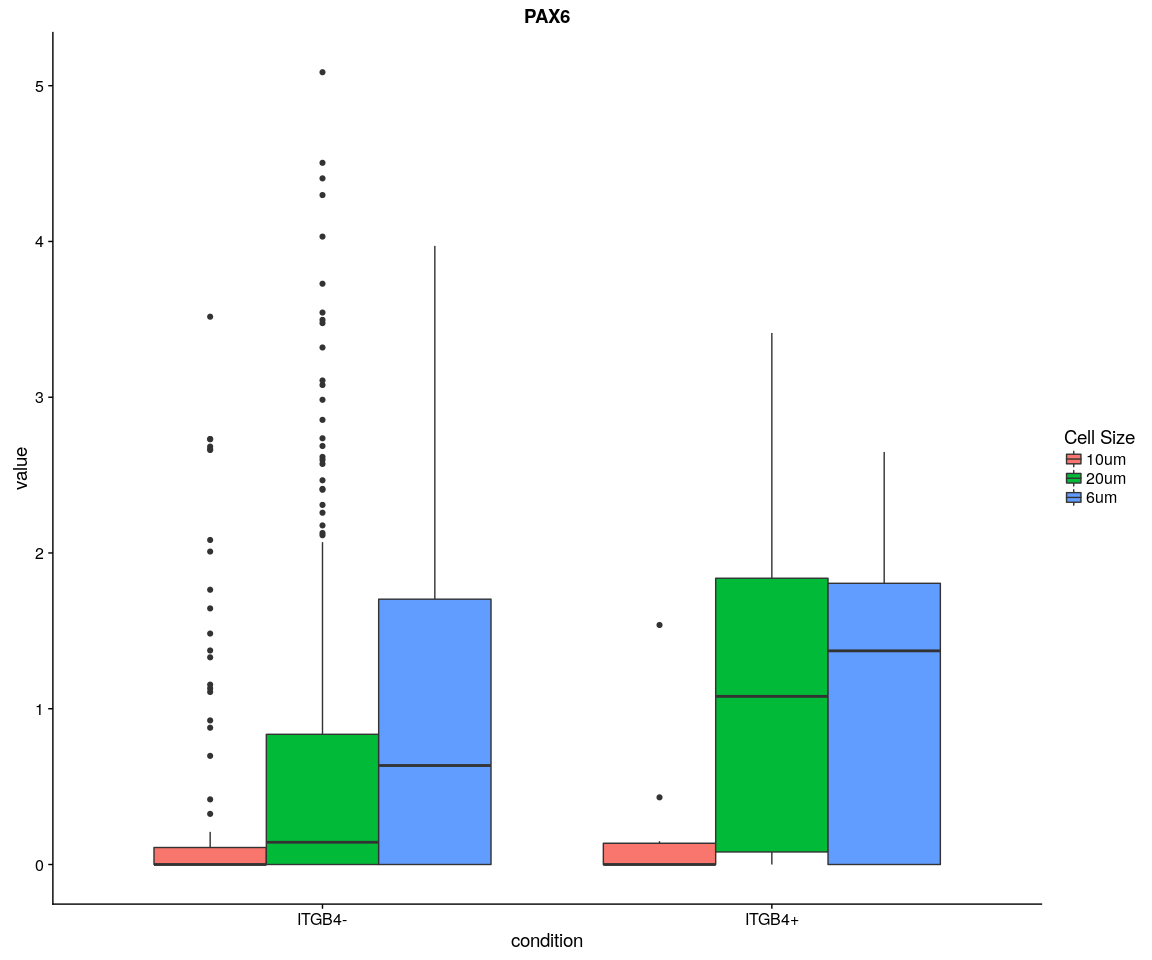
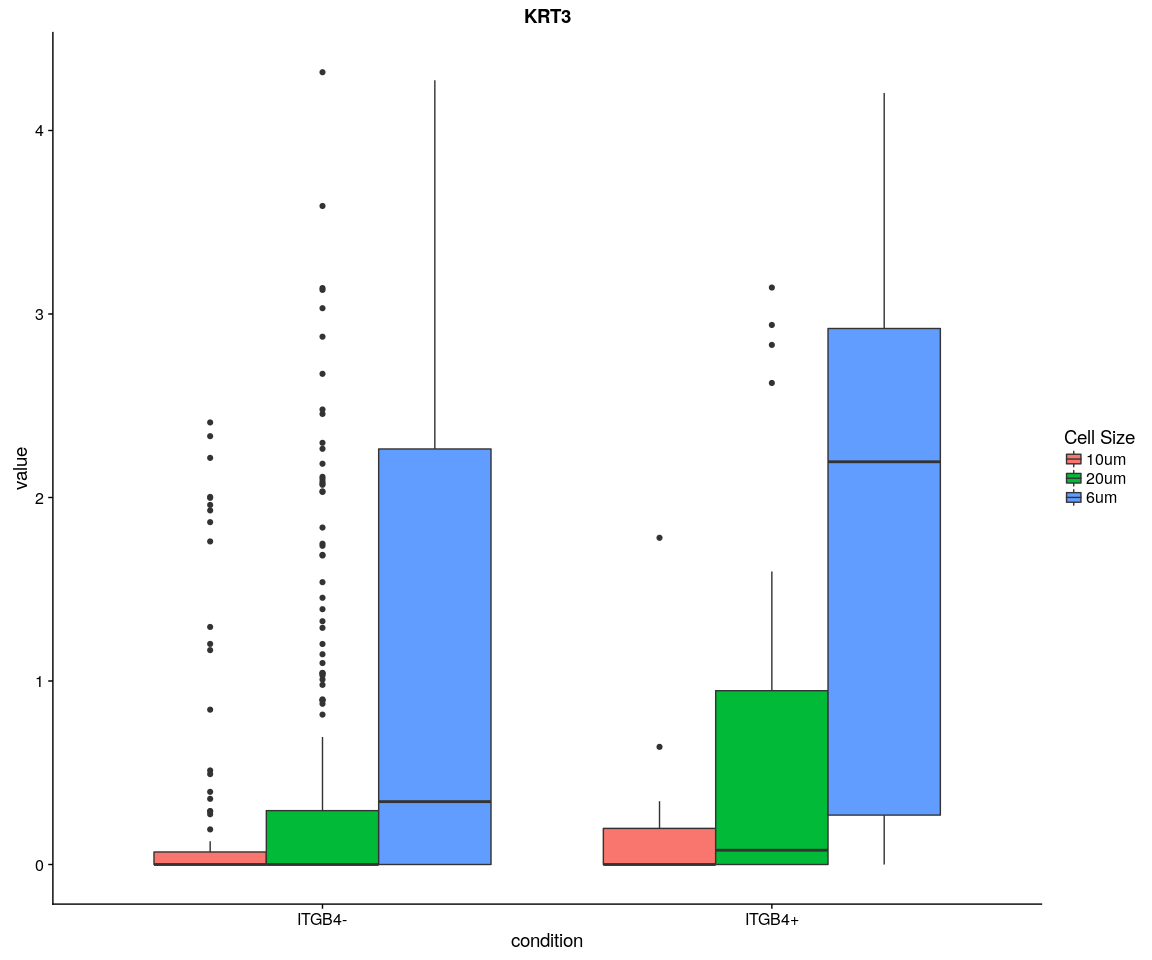
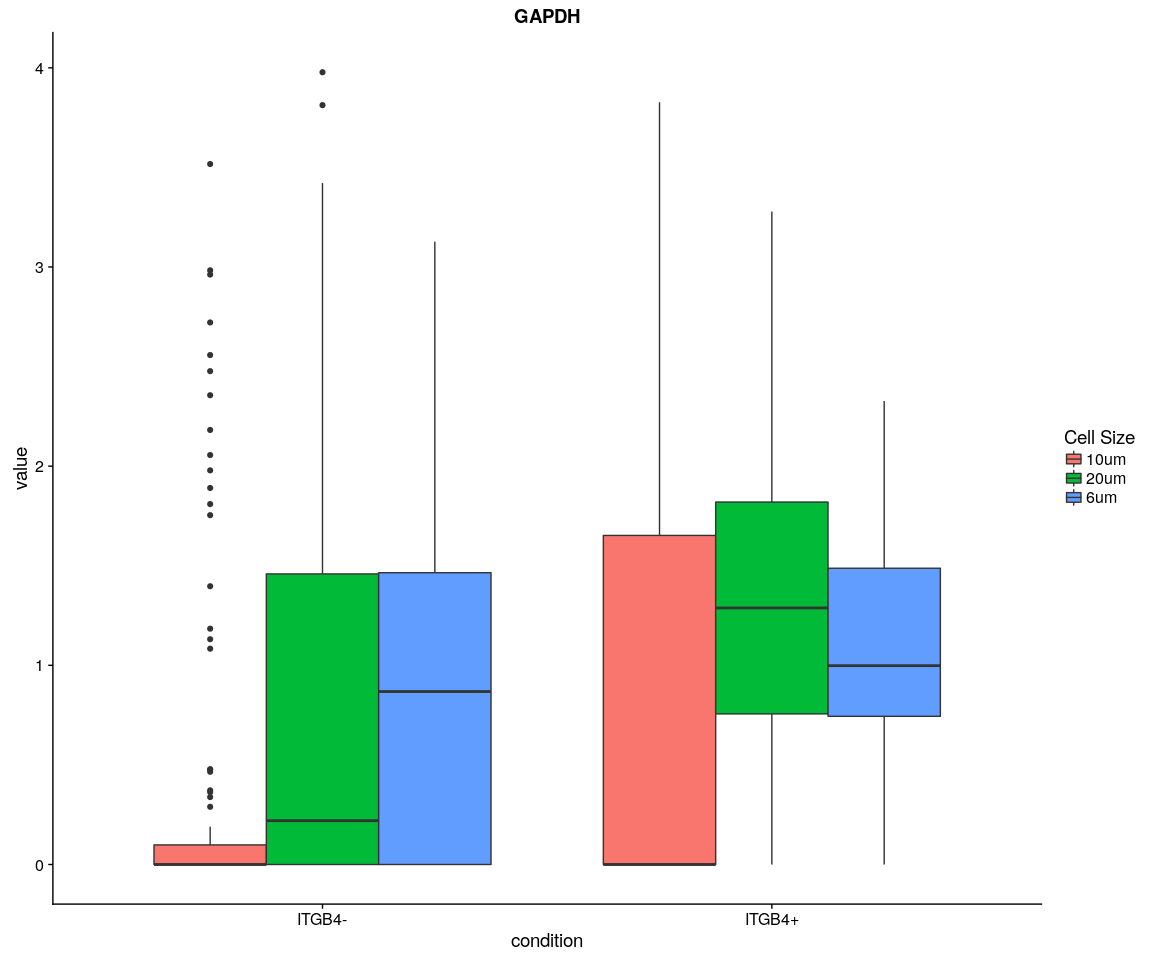
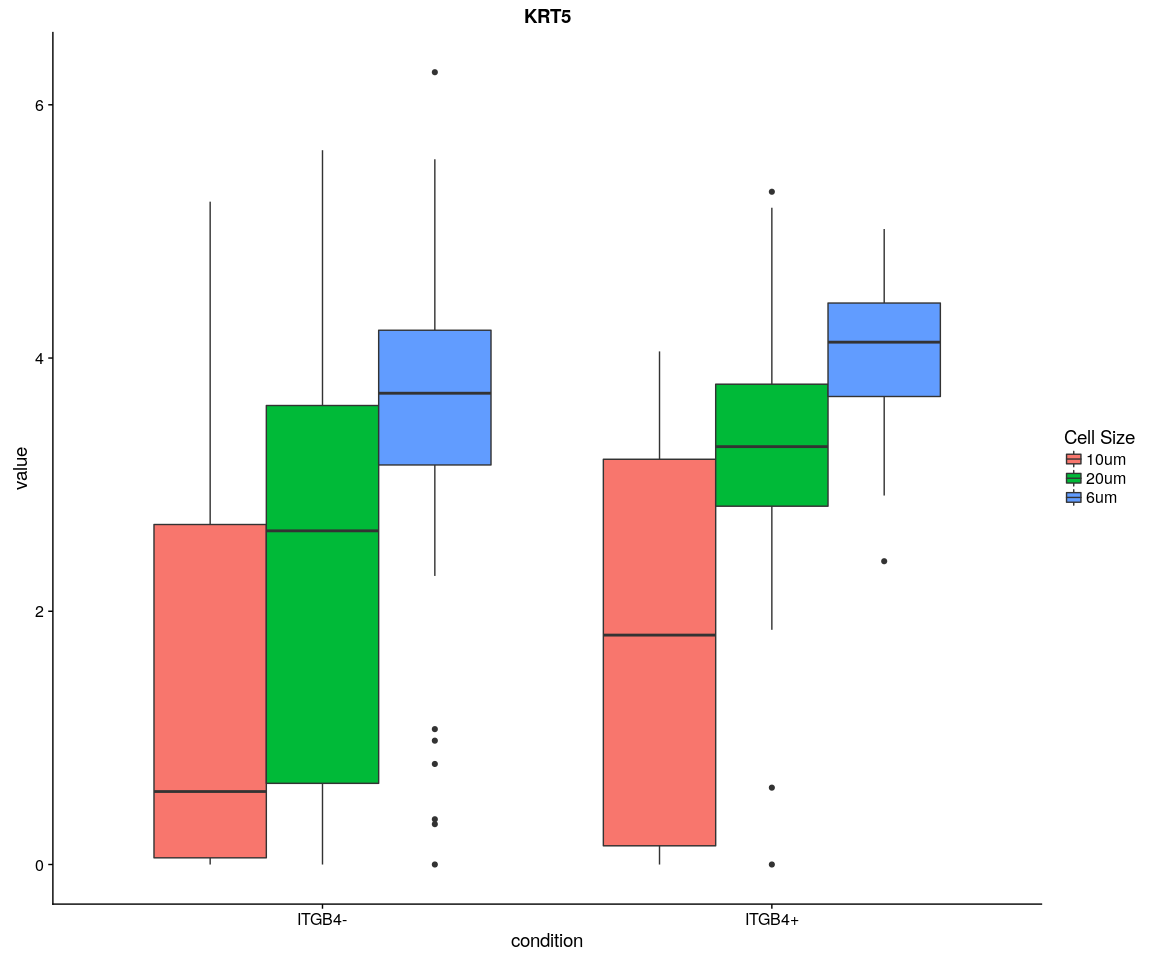
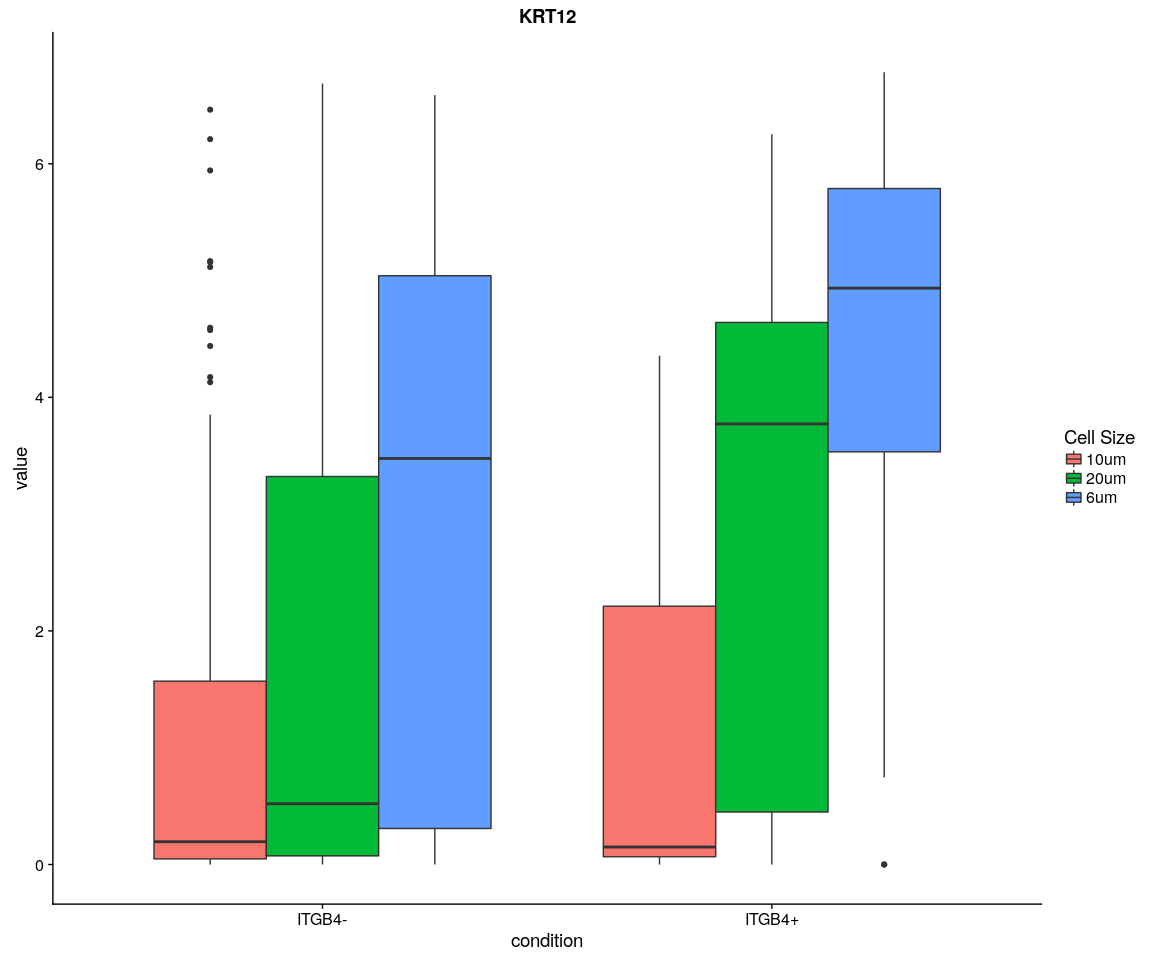
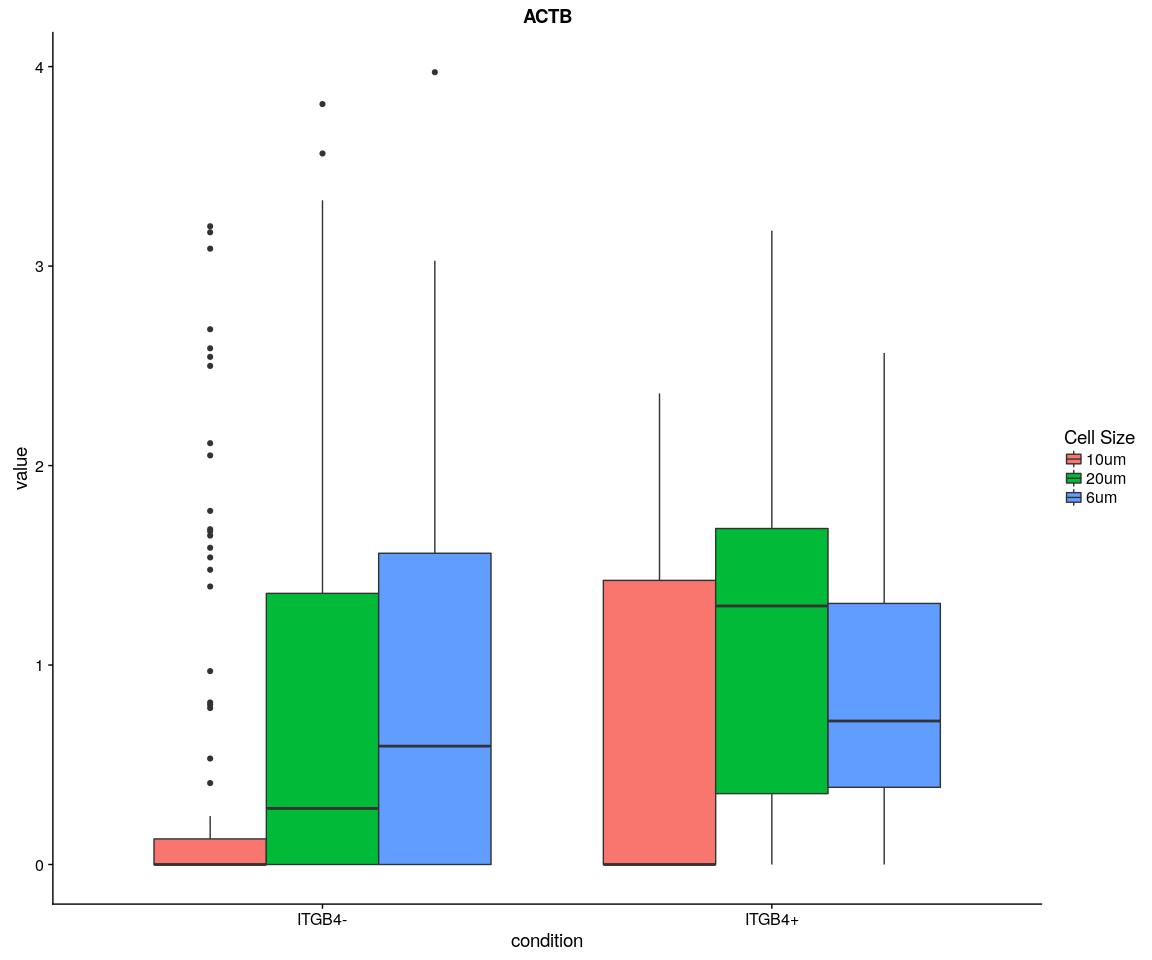
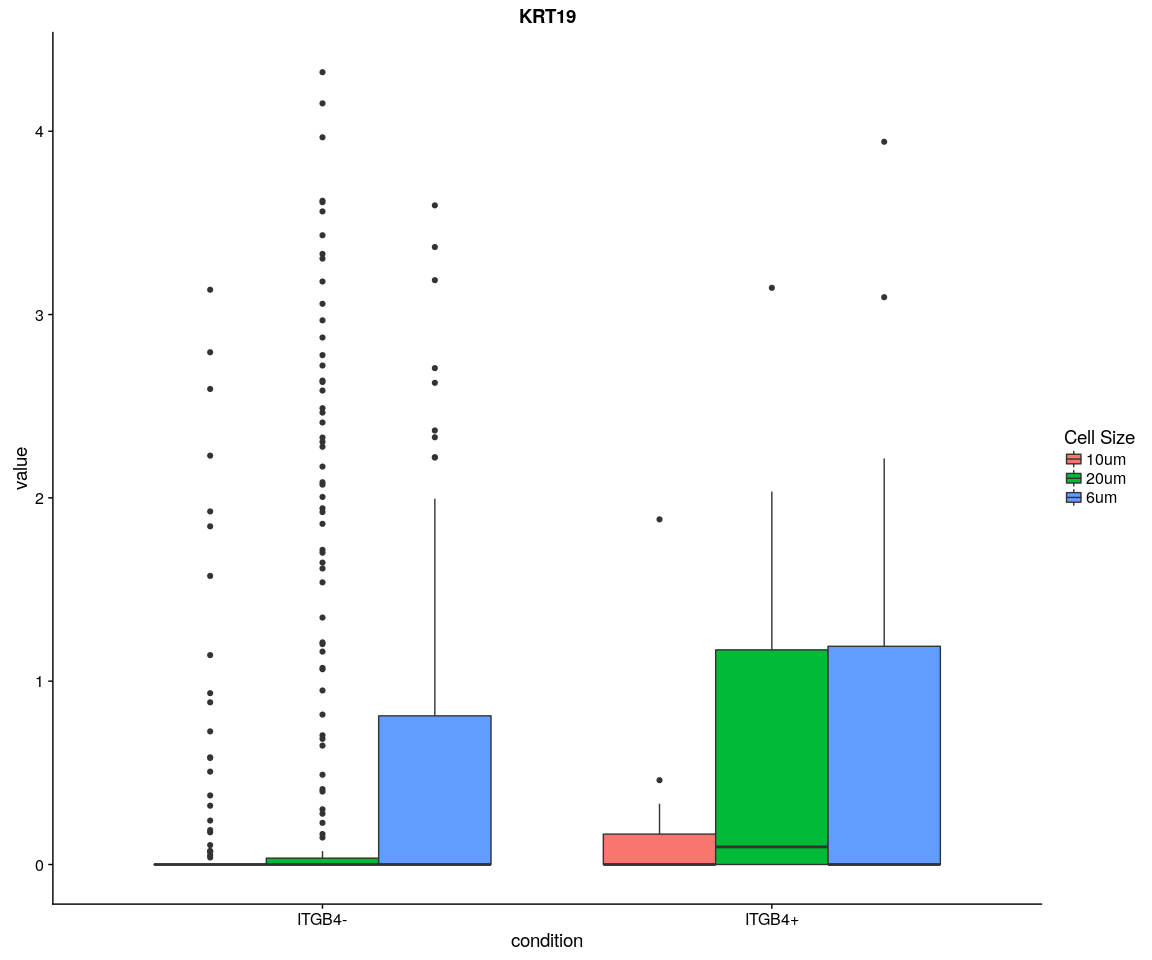
#### Violin

# 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))  
}



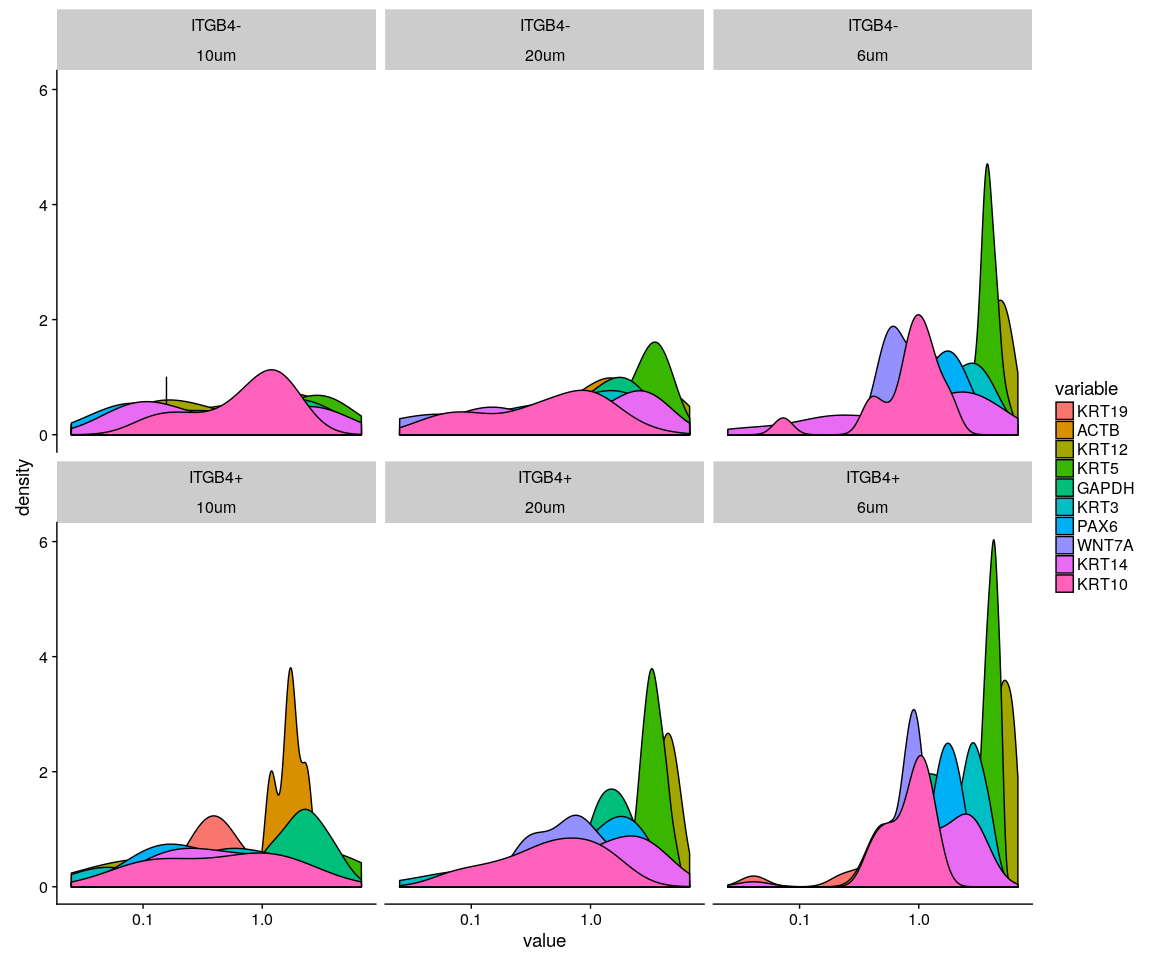
#### Boxplot

# 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))  
}

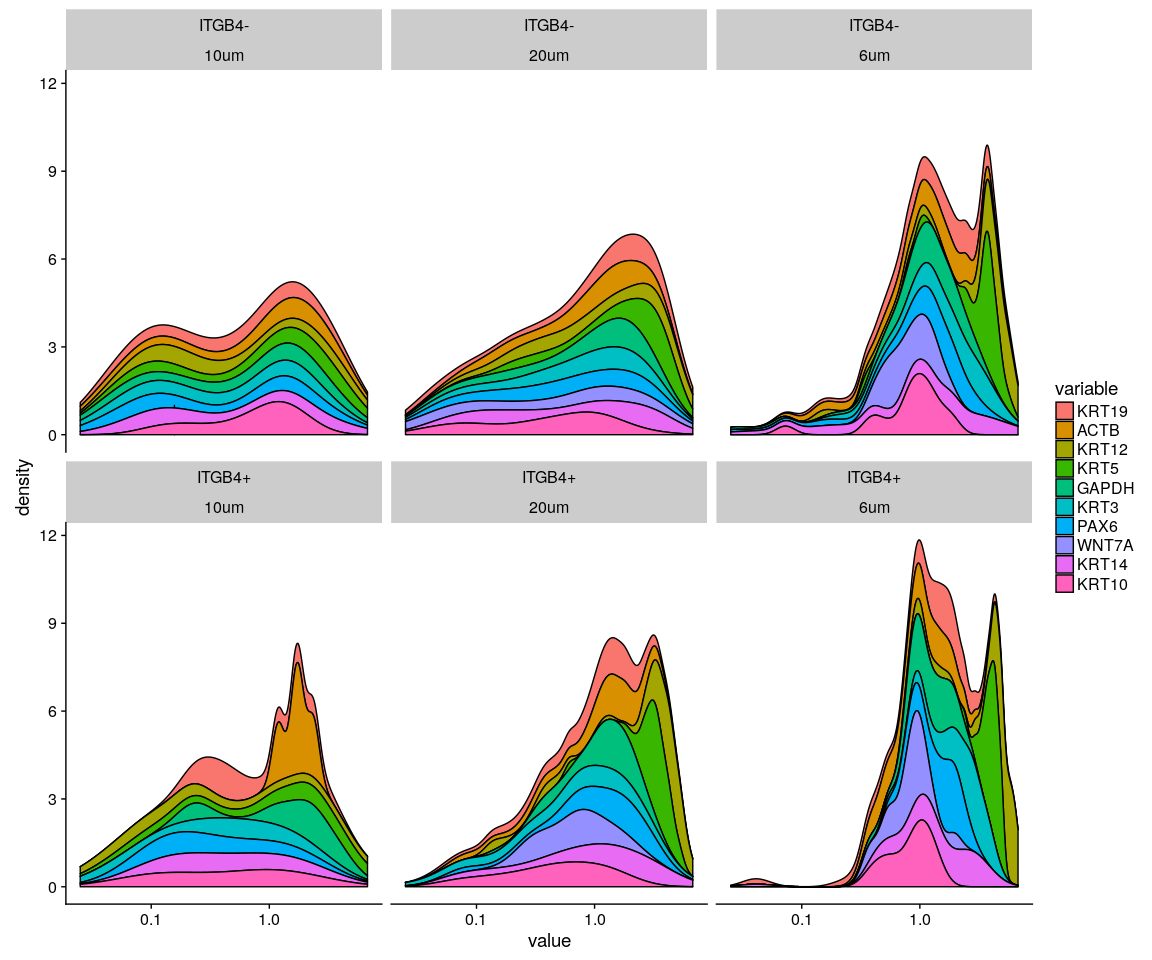


#### Density,histogram

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

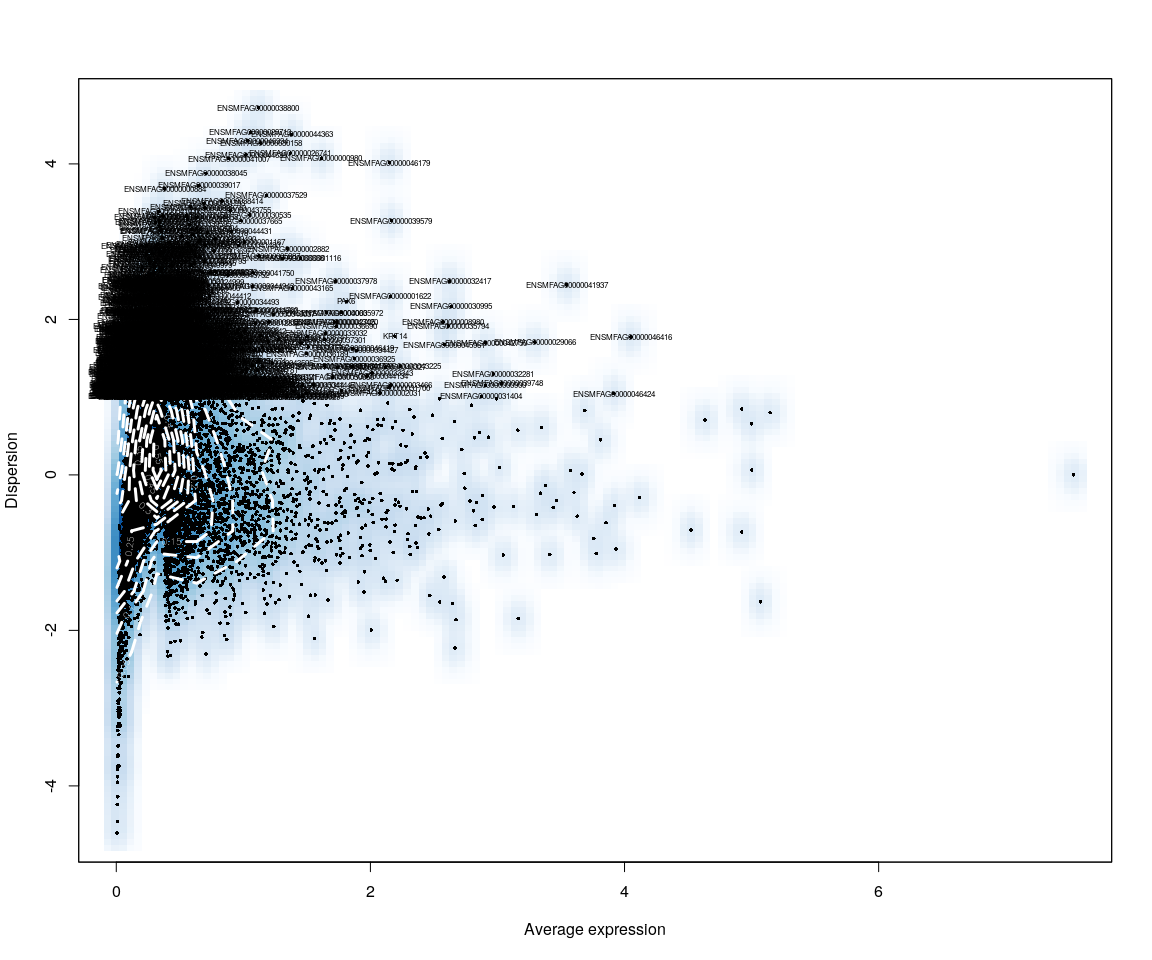
ITGB4 <- as.integer(monkey.only.pro["ITGB4", ])  
Positive.idx <- which(ITGB4 > 0)  
Negative.idx <- which(ITGB4 == 0)  
Positive.data <- monkey.only.pro[, Positive.idx, drop = FALSE]  
Negative.data <- monkey.only.pro[, Negative.idx, drop = FALSE]

### 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%



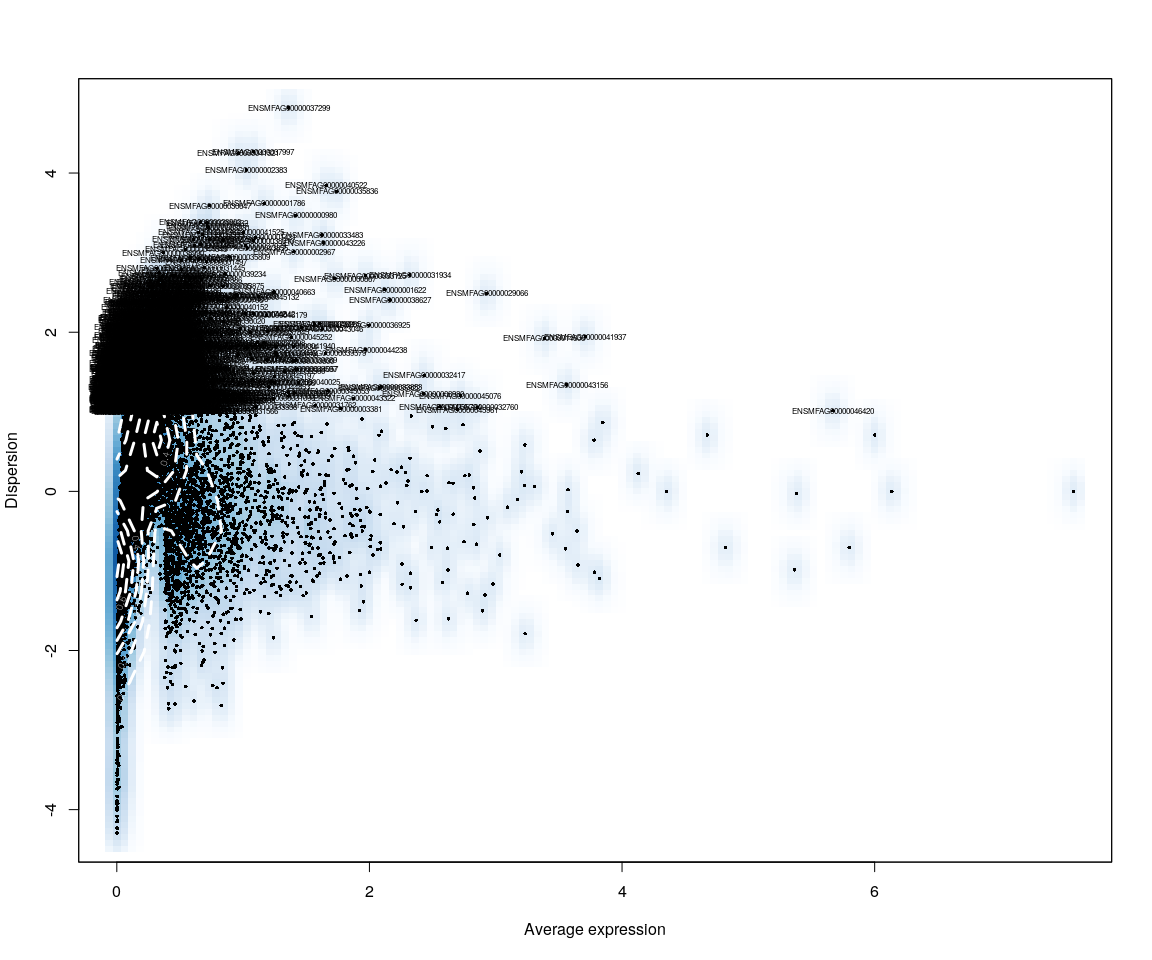
Positive.sample.group <- unlist(lapply(Positive.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][1])))  
Positive.sample.cellsize <- unlist(lapply(Positive.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
  
Positive.pbmc <- SetIdent(Positive.pbmc, cells.use =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

Pmonkey.imp.lognorm <- data.frame(FetchData(Positive.pbmc, vars.all = important.genes[important.genes %in%   
 rownames(Positive.pbmc@raw.data)]))  
Pmonkey.imp.lognorm$cell.size <- unlist(lapply(rownames(Pmonkey.imp.lognorm),   
 function(x) return(str\_split(x, "\_")[[1]][2])))  
Pmonkey.imp.lognorm$cell.sample <- unlist(lapply(rownames(Pmonkey.imp.lognorm),   
 function(x) return(str\_split(x, "\_")[[1]][1])))  
Pmonkey.imp.lognorm.melt <- melt(Pmonkey.imp.lognorm)

#### 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)

Nmonkey.imp.lognorm <- data.frame(FetchData(Negative.pbmc, vars.all = important.genes[important.genes %in%   
 rownames(Negative.pbmc@raw.data)]))  
Nmonkey.imp.lognorm$cell.size <- unlist(lapply(rownames(Nmonkey.imp.lognorm),   
 function(x) return(str\_split(x, "\_")[[1]][2])))  
Nmonkey.imp.lognorm$cell.sample <- unlist(lapply(rownames(Nmonkey.imp.lognorm),   
 function(x) return(str\_split(x, "\_")[[1]][1])))  
  
Nmonkey.imp.lognorm.melt <- melt(Nmonkey.imp.lognorm)

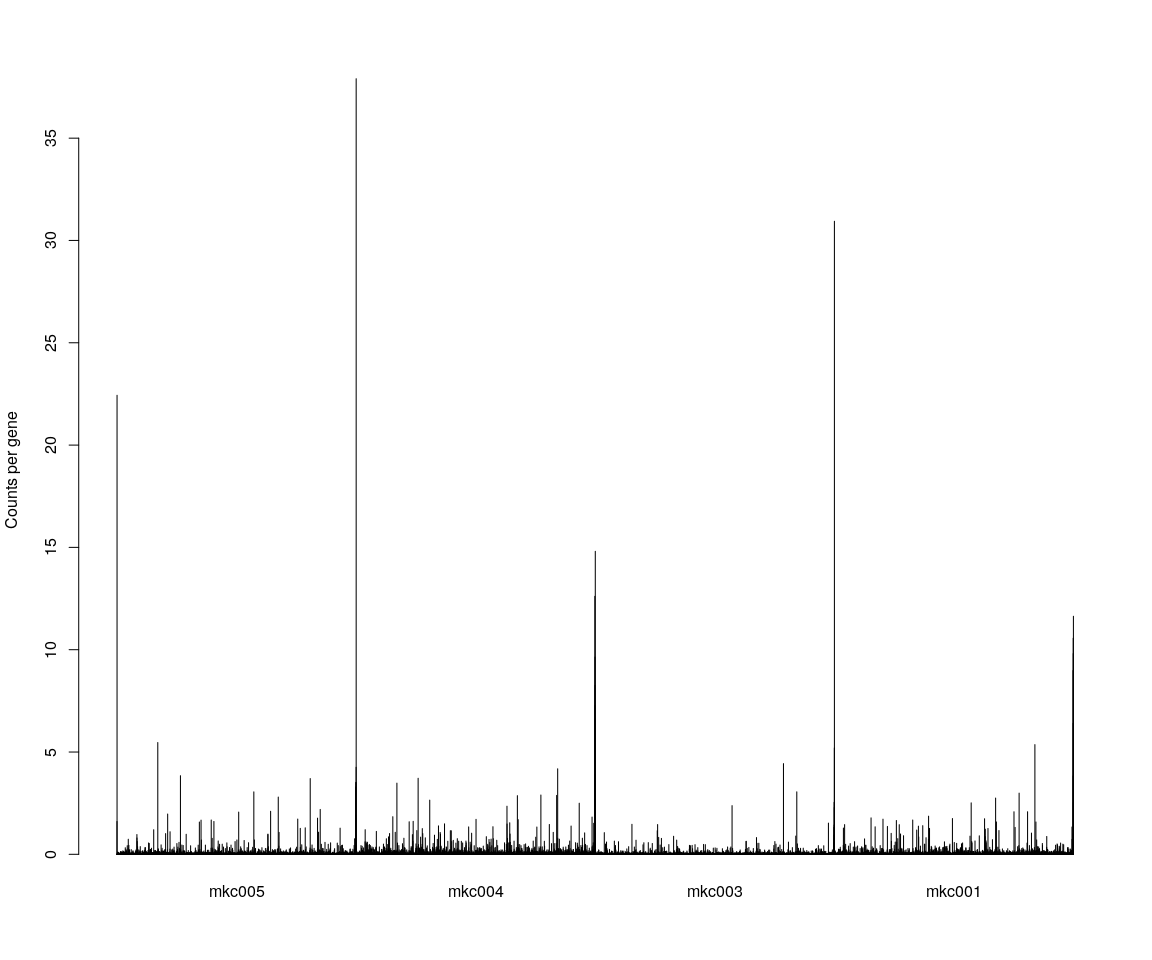
Accordind to the Dispersion vs Avearge expression of Positive and Negative data on ITGB4,they tell us that the although they have similar shape and trend,dispersion of Positive data is more significant than Negative in some genes.

## Step 1: analysis on Positive data

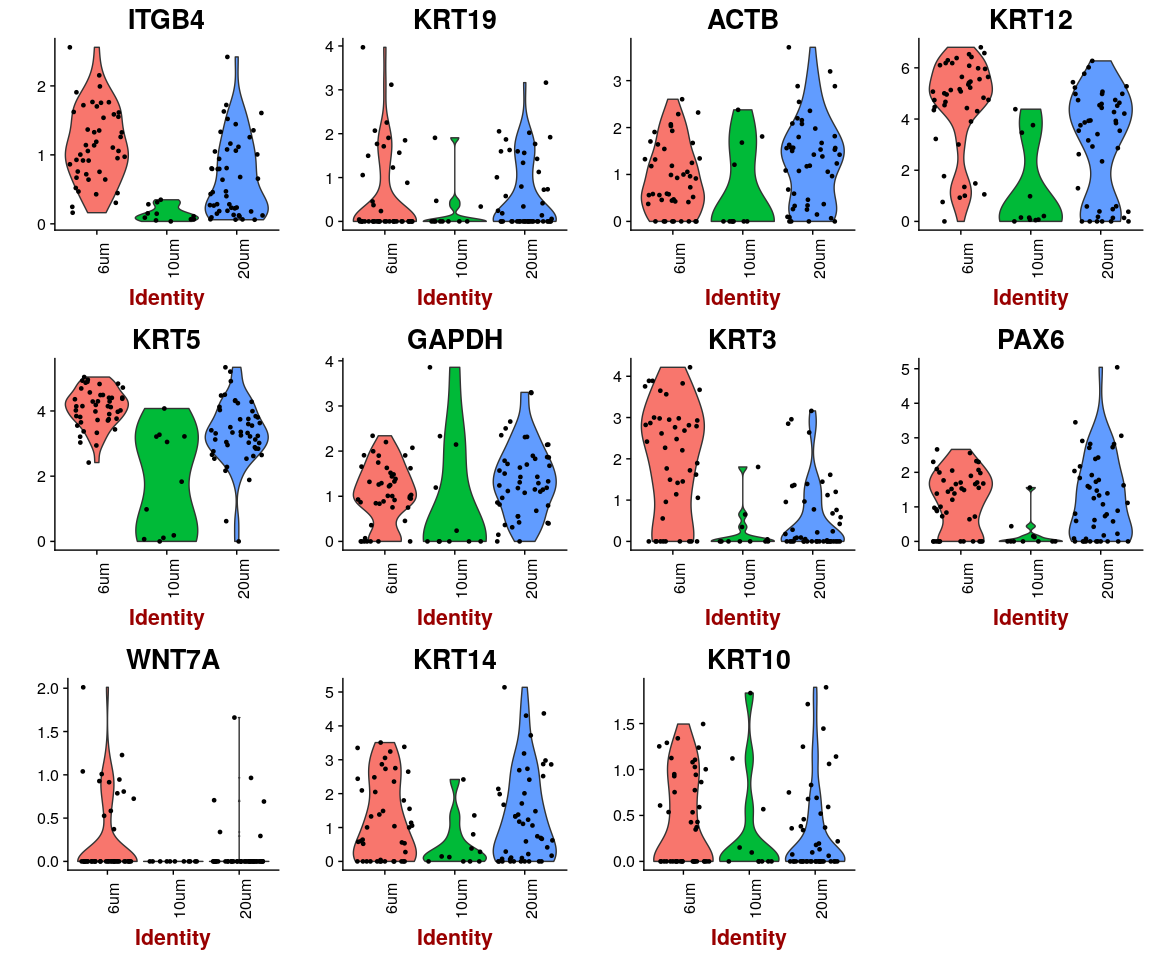
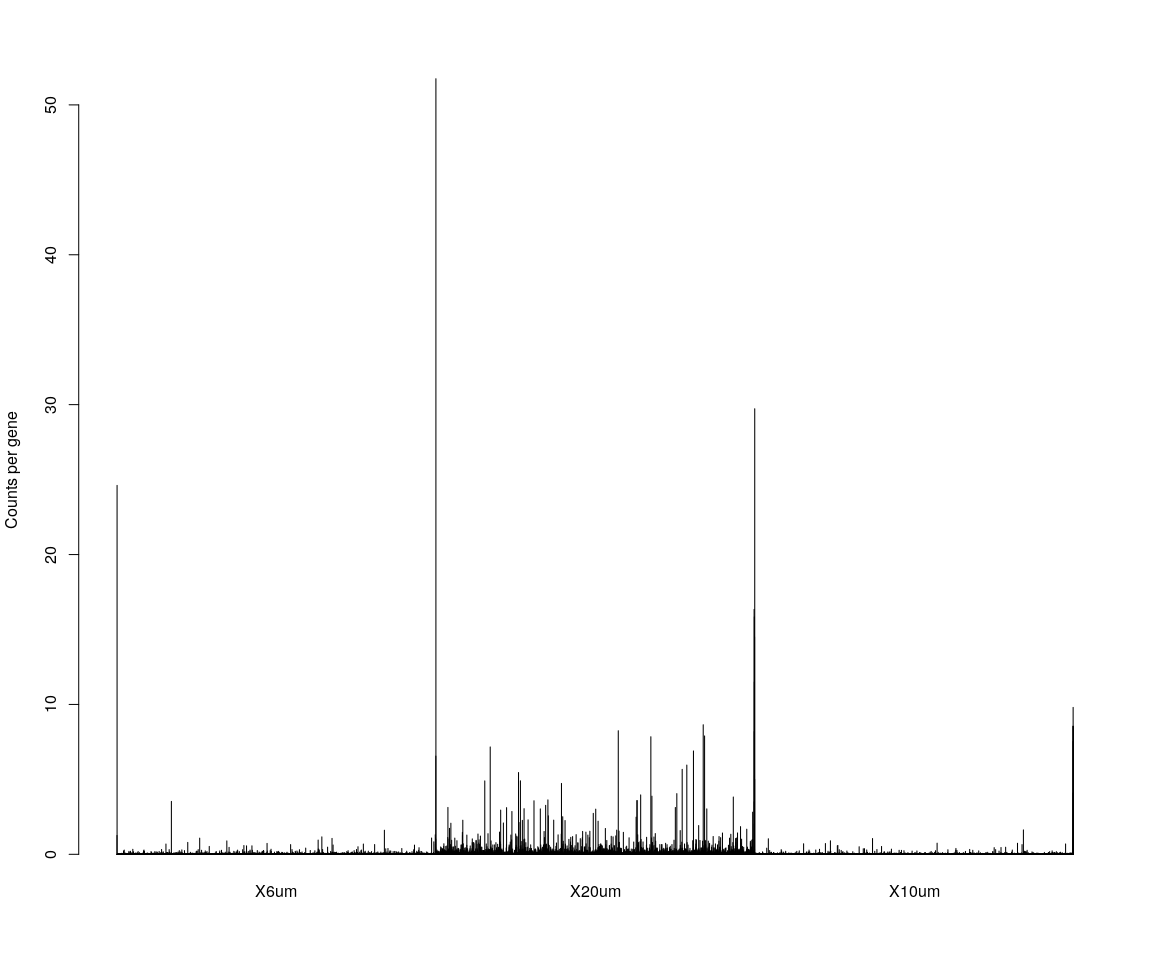
## Figure Explore

### 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**

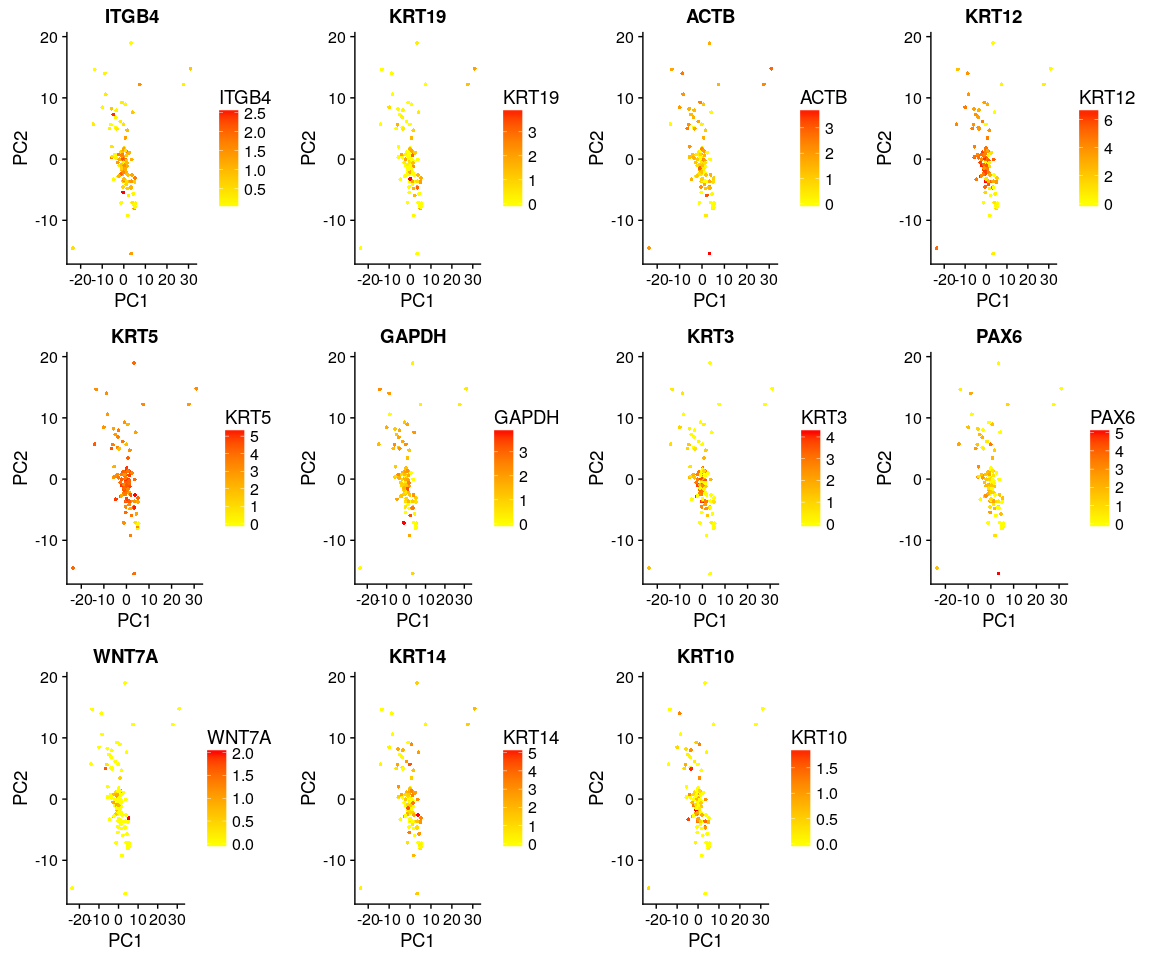
Here,do the dimensionality reduction using the PCA, tSNE method

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

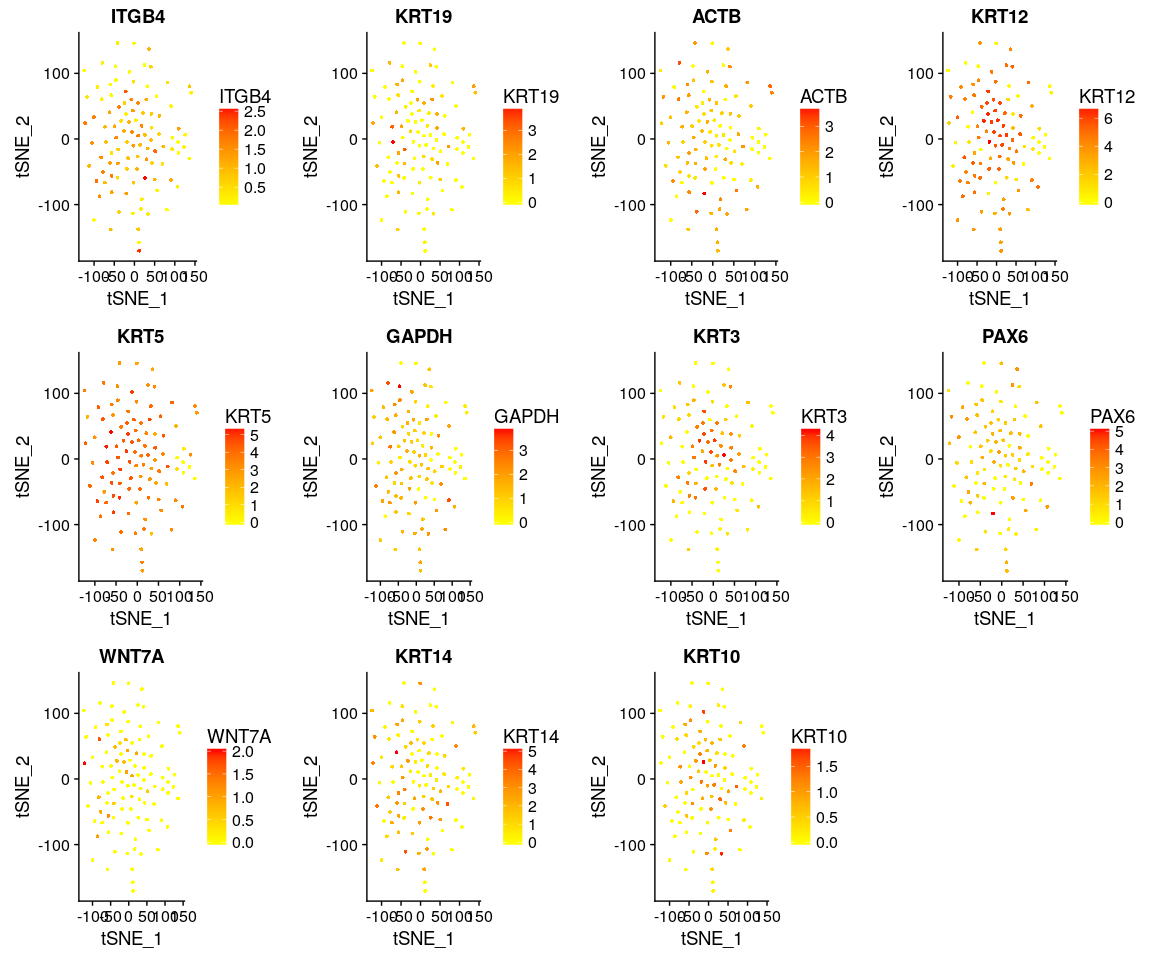
Positive.pbmc <- PCA.TSNE(object = Positive.pbmc, pcs.compute = FALSE, num.pcs = 28)

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

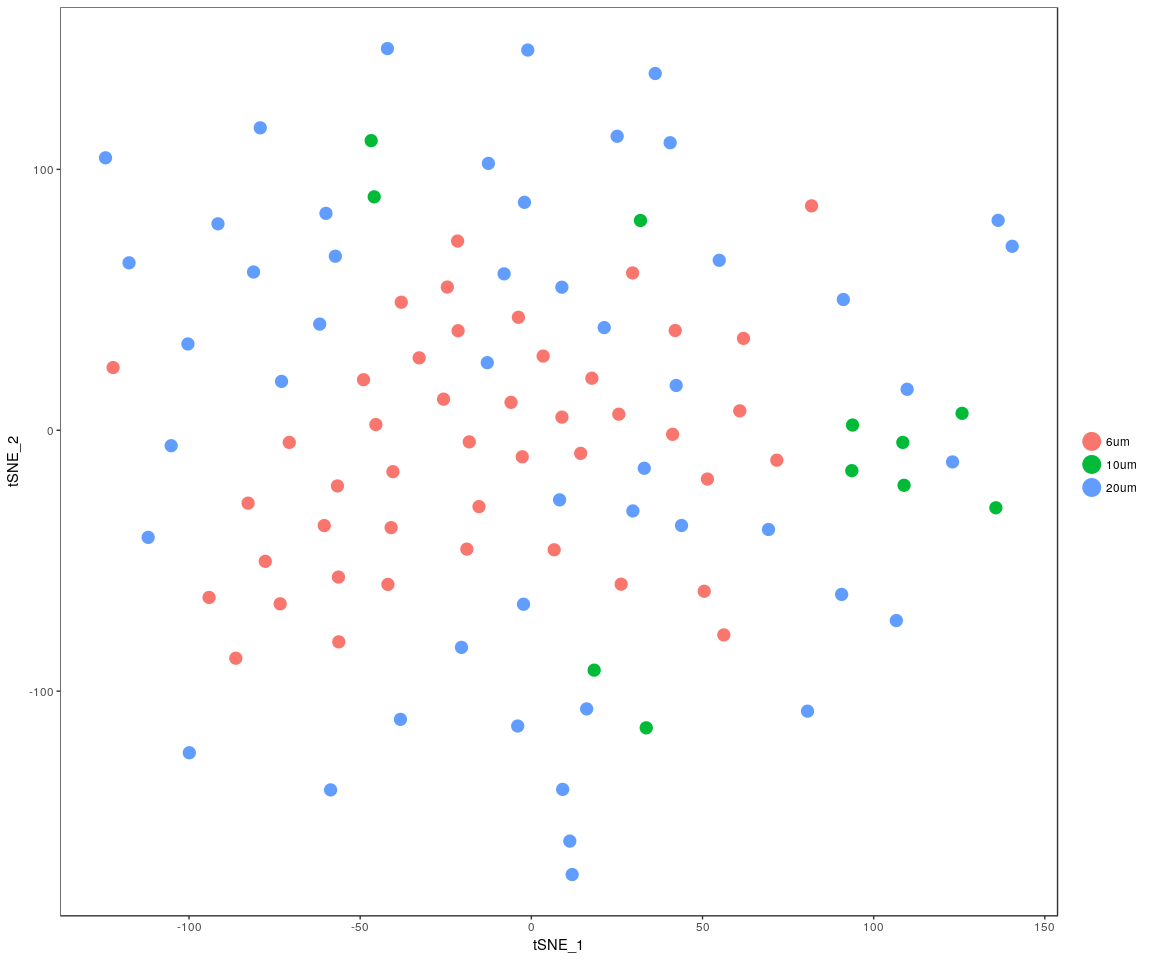
FeaturePlot(object = Positive.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Positive.pbmc@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "pca") # ITGB4 gene in part dataset



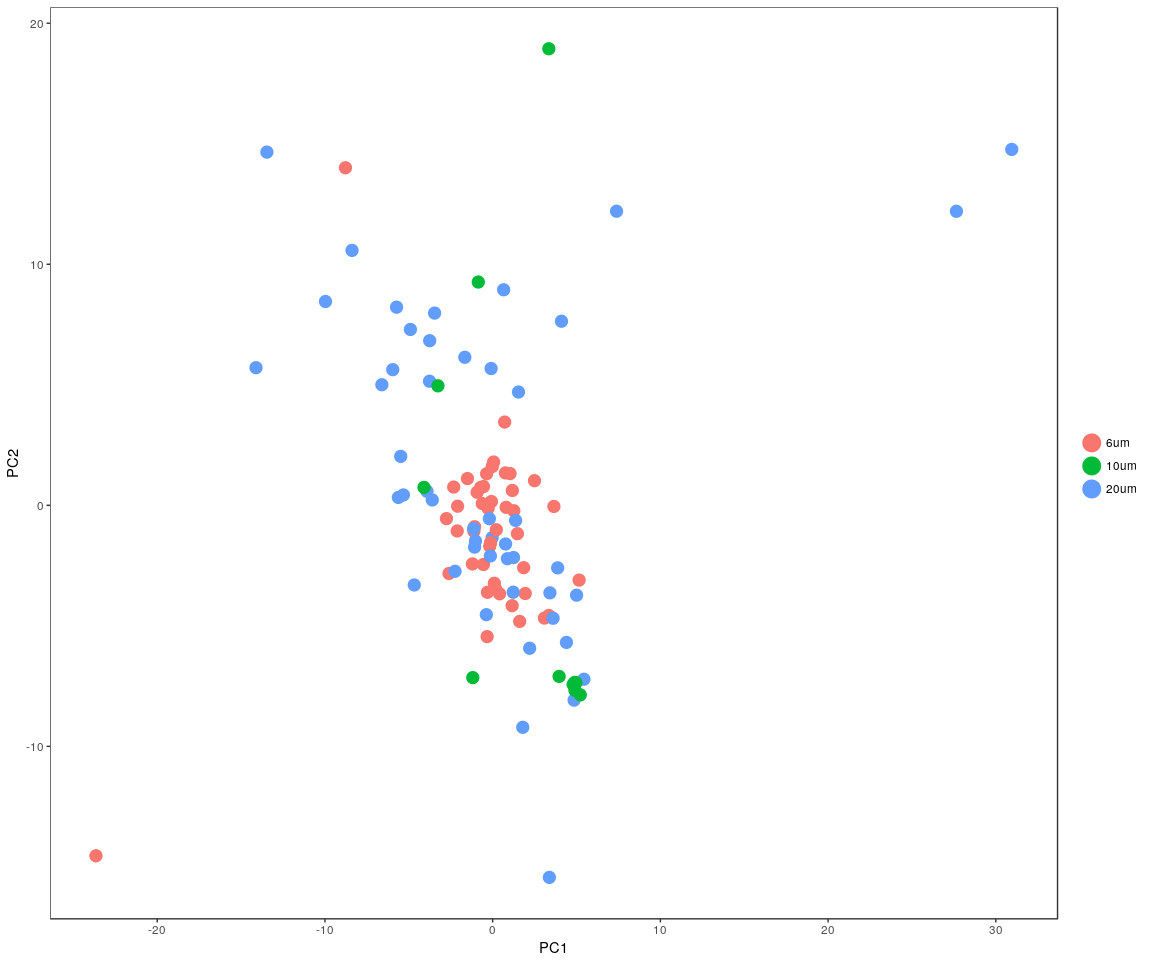
FeaturePlot(object = Positive.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Positive.pbmc@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "tsne") # ITGB4 gene in part dataset



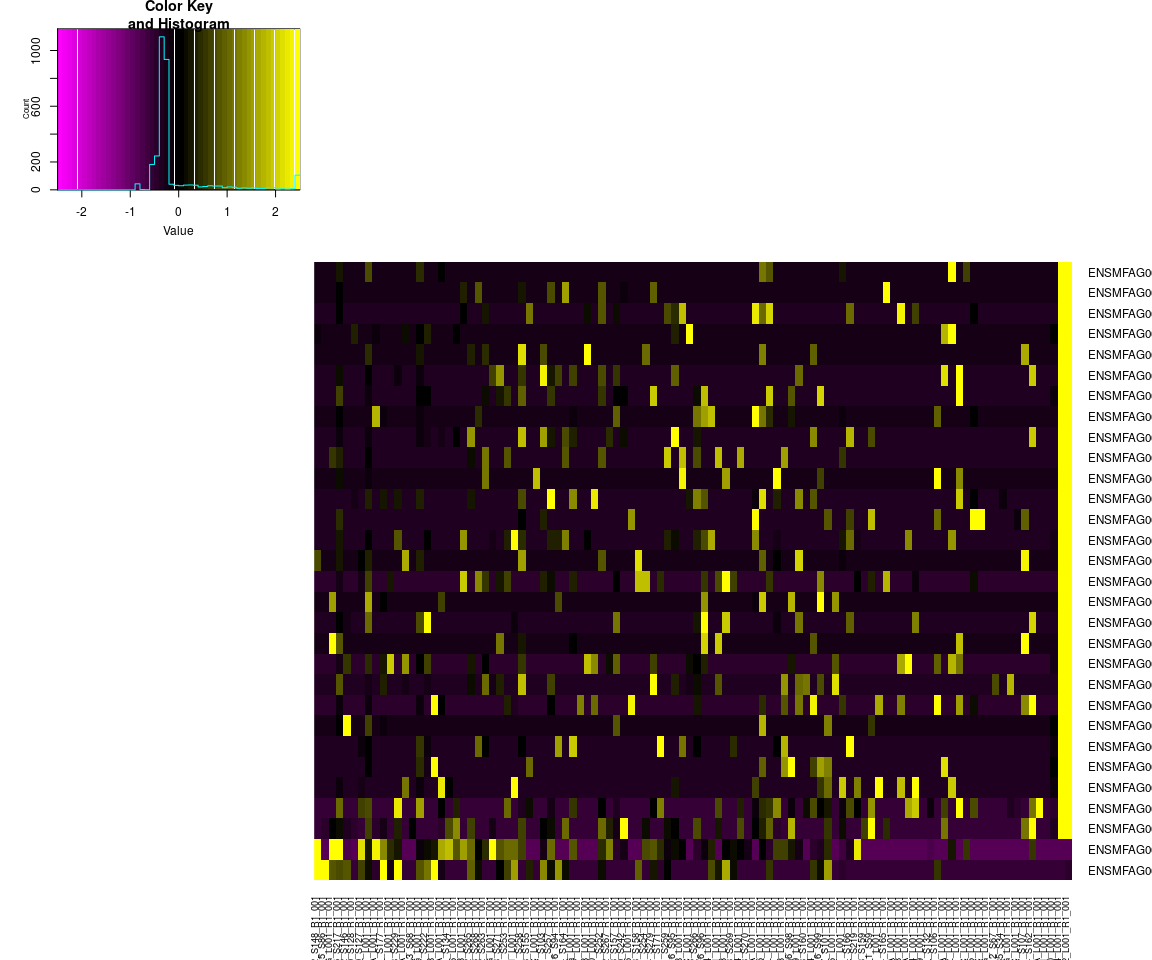
DimPlot(Positive.pbmc, reduction.use = "tsne", pt.size = 4) # grour by sample



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



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



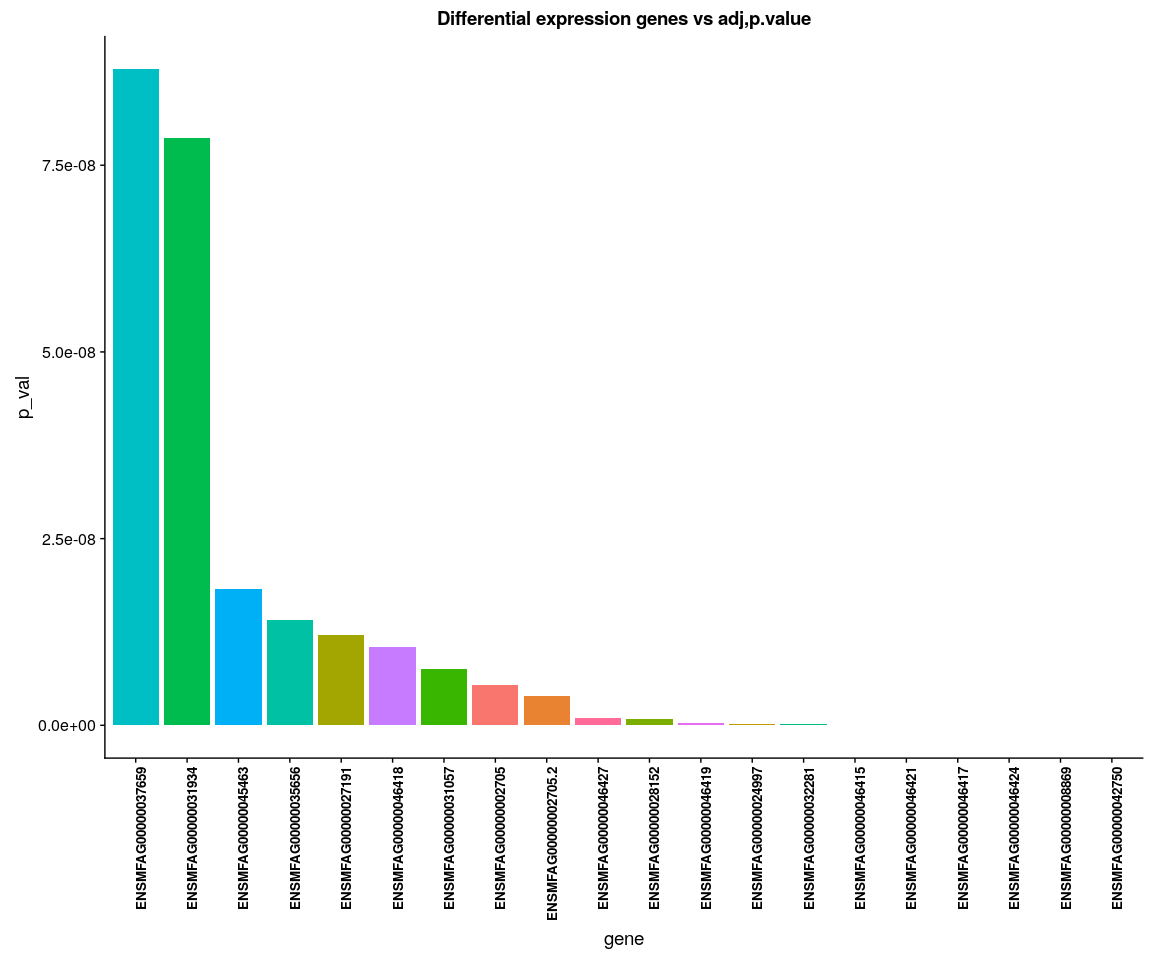
## Differential expression

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  
## ENSMFAG00000042750 3.230780e-18 -2.153475 0.822 0.983 2.649240e-14 6um  
## ENSMFAG00000008869 1.549249e-17 2.504637 0.956 0.305 1.270384e-13 6um  
## ENSMFAG00000046424 1.021197e-15 -1.872760 1.000 1.000 8.373819e-12 6um  
## ENSMFAG00000046417 2.536596e-15 -1.649695 0.956 0.983 2.080008e-11 6um  
## ENSMFAG00000046421 1.107121e-14 -1.773717 1.000 1.000 9.078396e-11 6um  
## ENSMFAG00000046415 1.329215e-14 -2.043827 1.000 1.000 1.089956e-10 6um  
## gene  
## ENSMFAG00000042750 ENSMFAG00000042750  
## ENSMFAG00000008869 ENSMFAG00000008869  
## ENSMFAG00000046424 ENSMFAG00000046424  
## ENSMFAG00000046417 ENSMFAG00000046417  
## ENSMFAG00000046421 ENSMFAG00000046421  
## ENSMFAG00000046415 ENSMFAG00000046415

### Bar plot of gene’s p.val

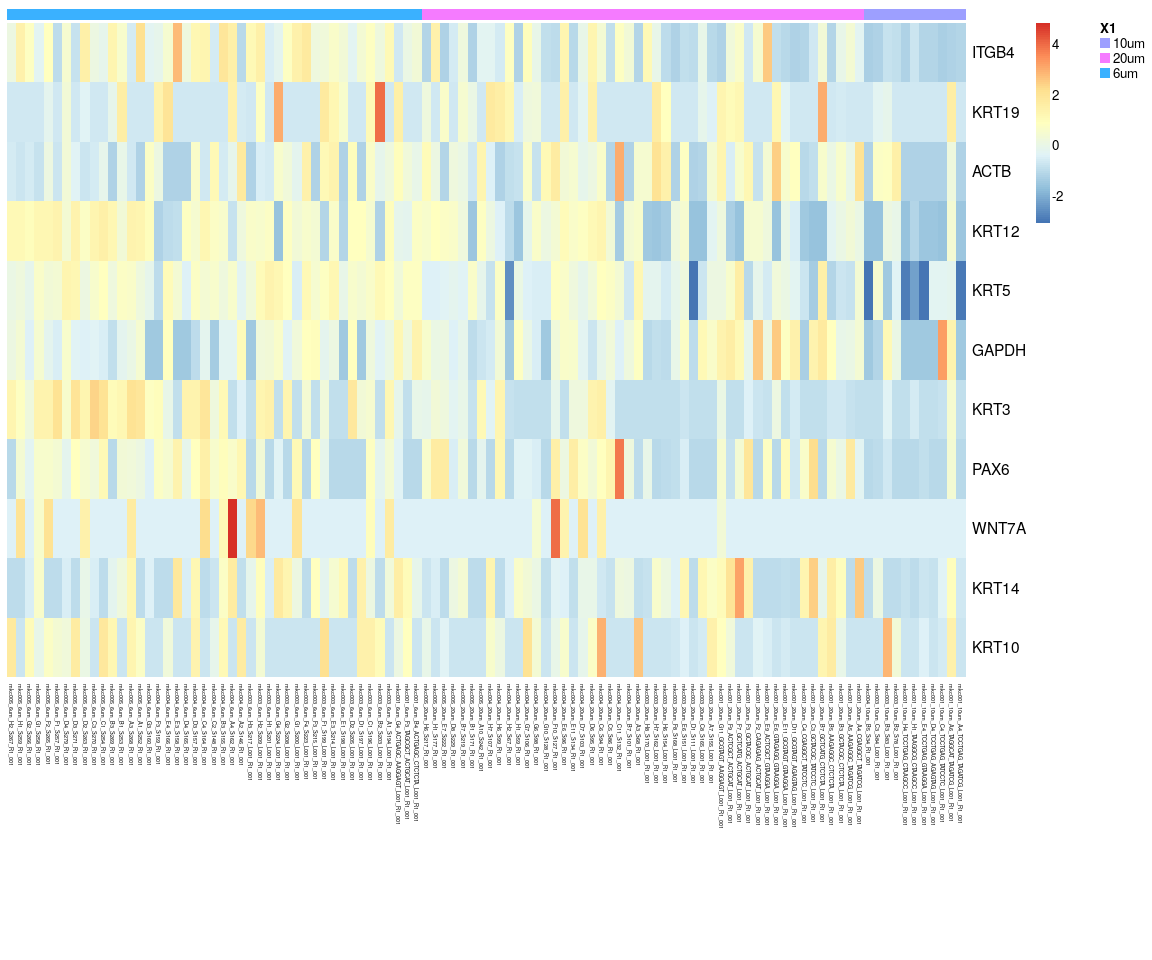


### Poistive Heatmap for important genes

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

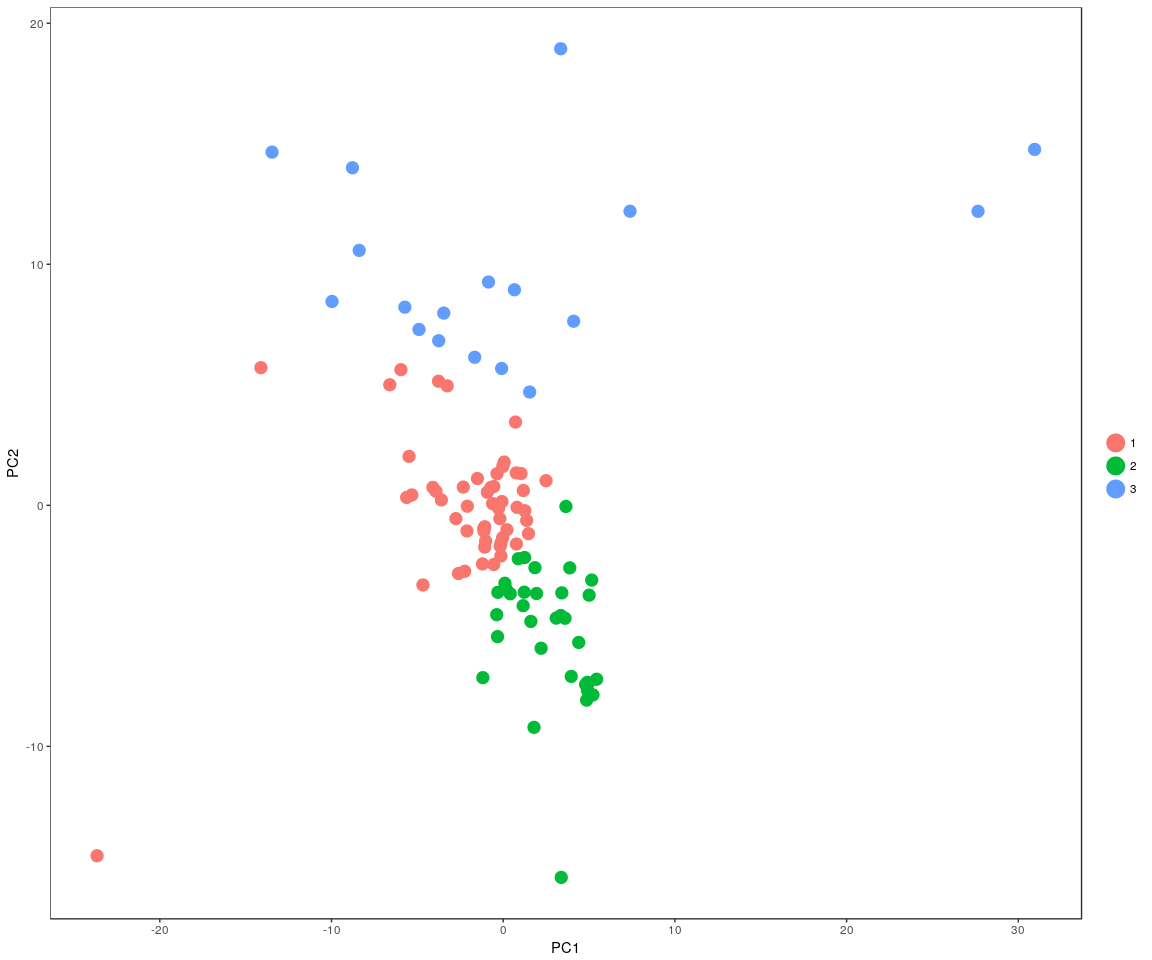
NMF::aheatmap(Positive.heatmap[[2]], Rowv = NA, Colv = NA, annCol = Positive.heatmap[[1]],   
 scale = "none")



We have find all marker genes across sample,there are 242 significant genes(adjust p-value <0.05) in all marker genes.

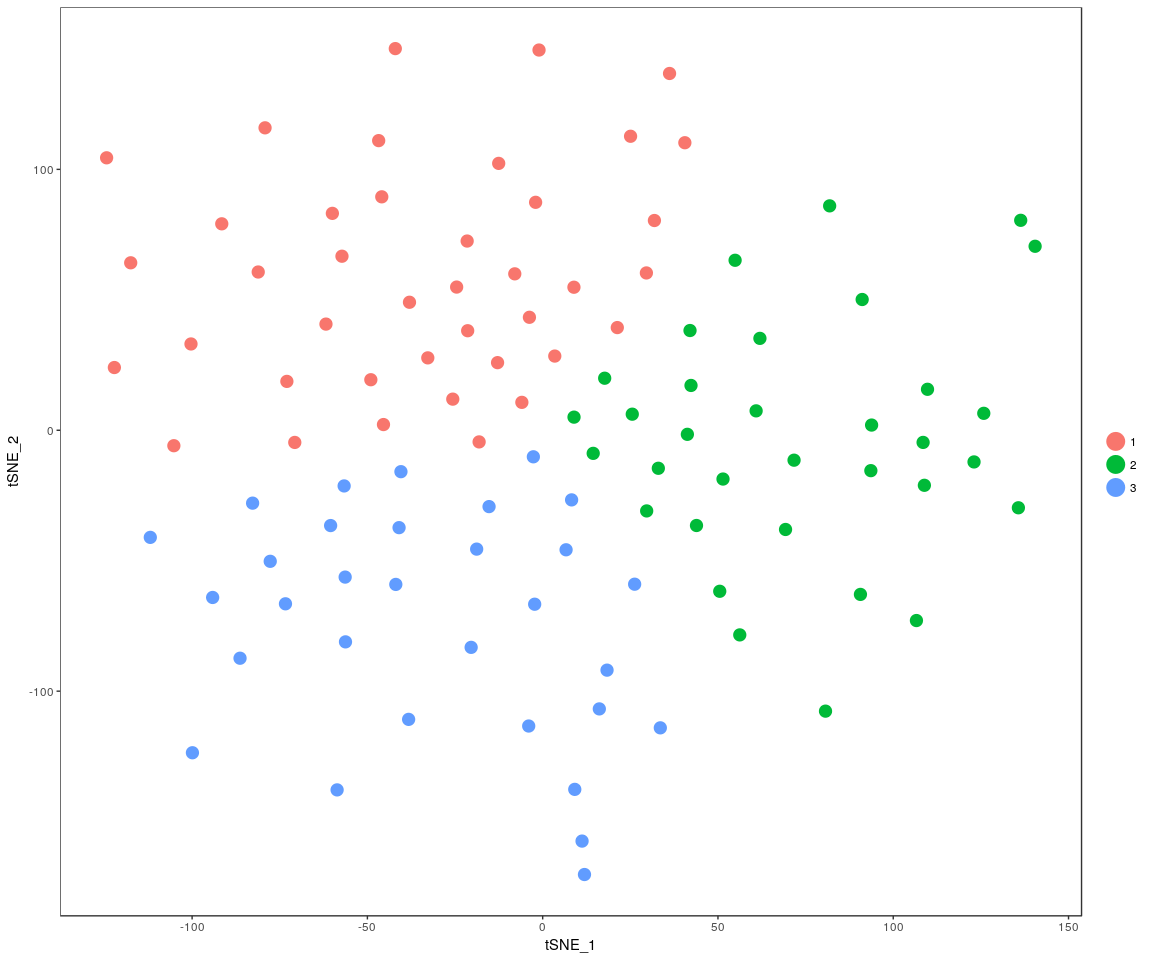
### Next,Spectral k-means clustering on single cells based on PCA

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

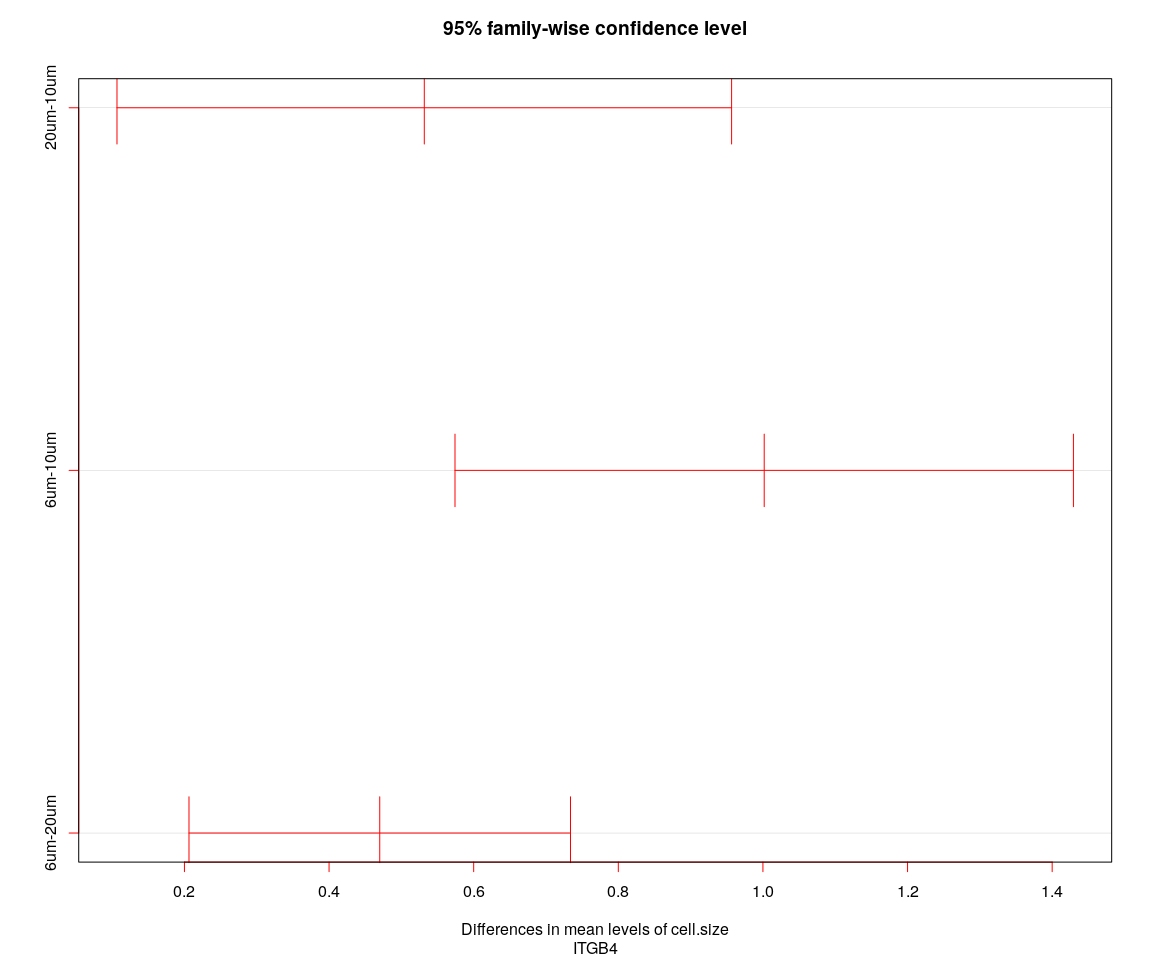
Positive.pbmc <- KClustDimension(Positive.pbmc, reduction.use = "tsne", k.use = 3)  
clusters.tsne <- Positive.pbmc@meta.data$kdimension.ident  
DimPlot(Positive.pbmc, pt.size = 4, group.by = "kdimension.ident", reduction.use = "tsne")



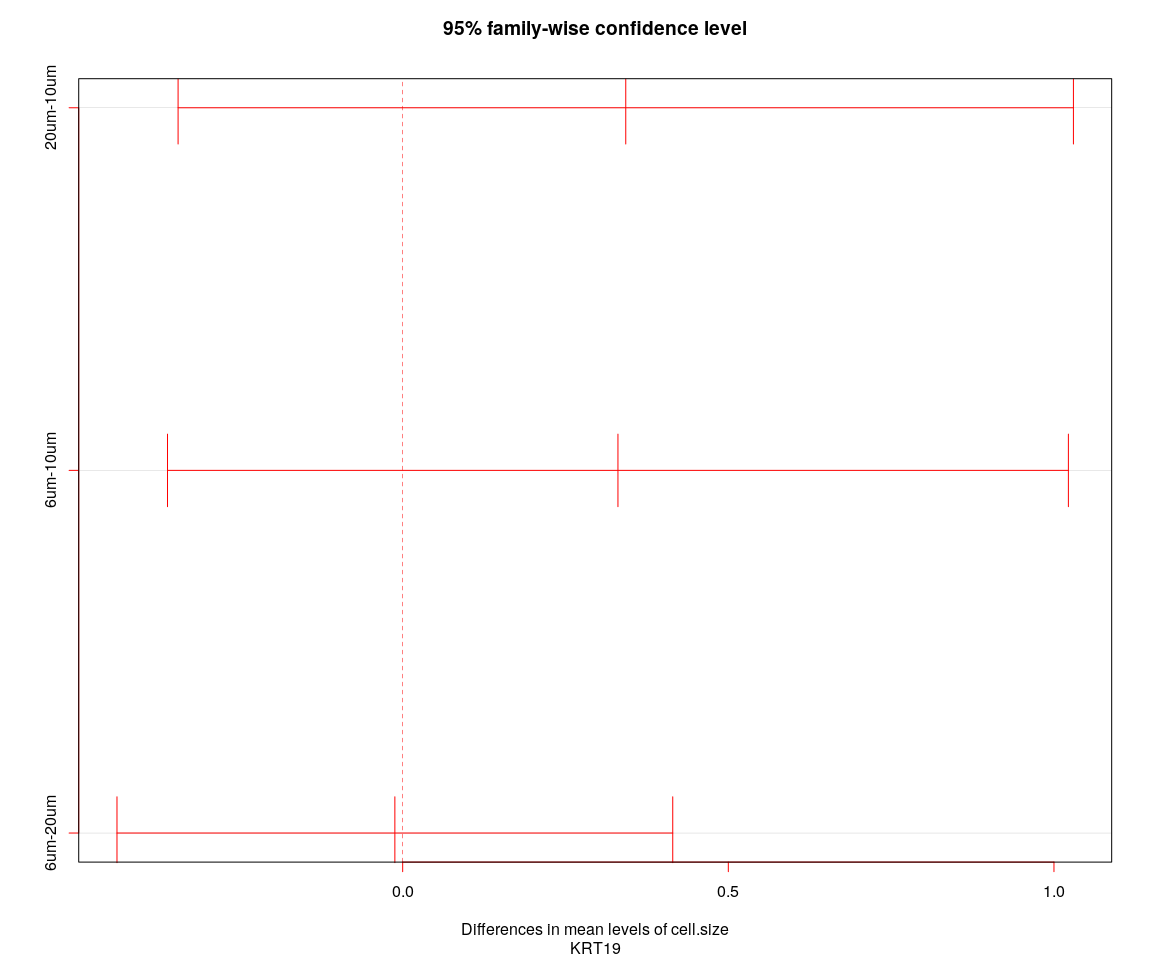
### Positive: Anova analsis:test the important genes’mean between cell size,cell groups ,whether is equal

#### anavo analysis

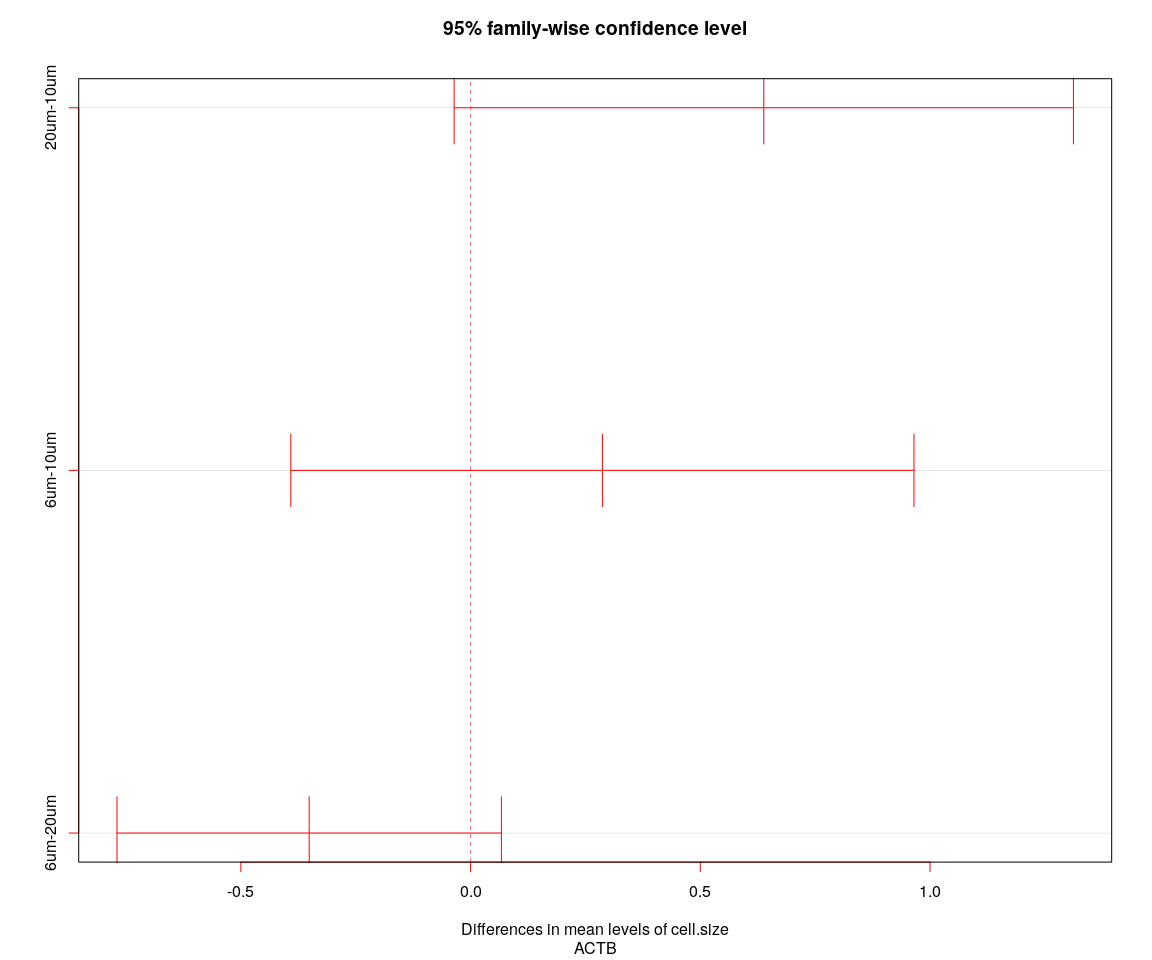
for (gene in unique(Pmonkey.imp.lognorm.melt$variable)) {  
 aov.gene <- Anova.gene(data = Pmonkey.imp.lognorm.melt, gene = gene, tuk.which = "cell.size",   
 inter = FALSE, plot.aov = FALSE)  
}



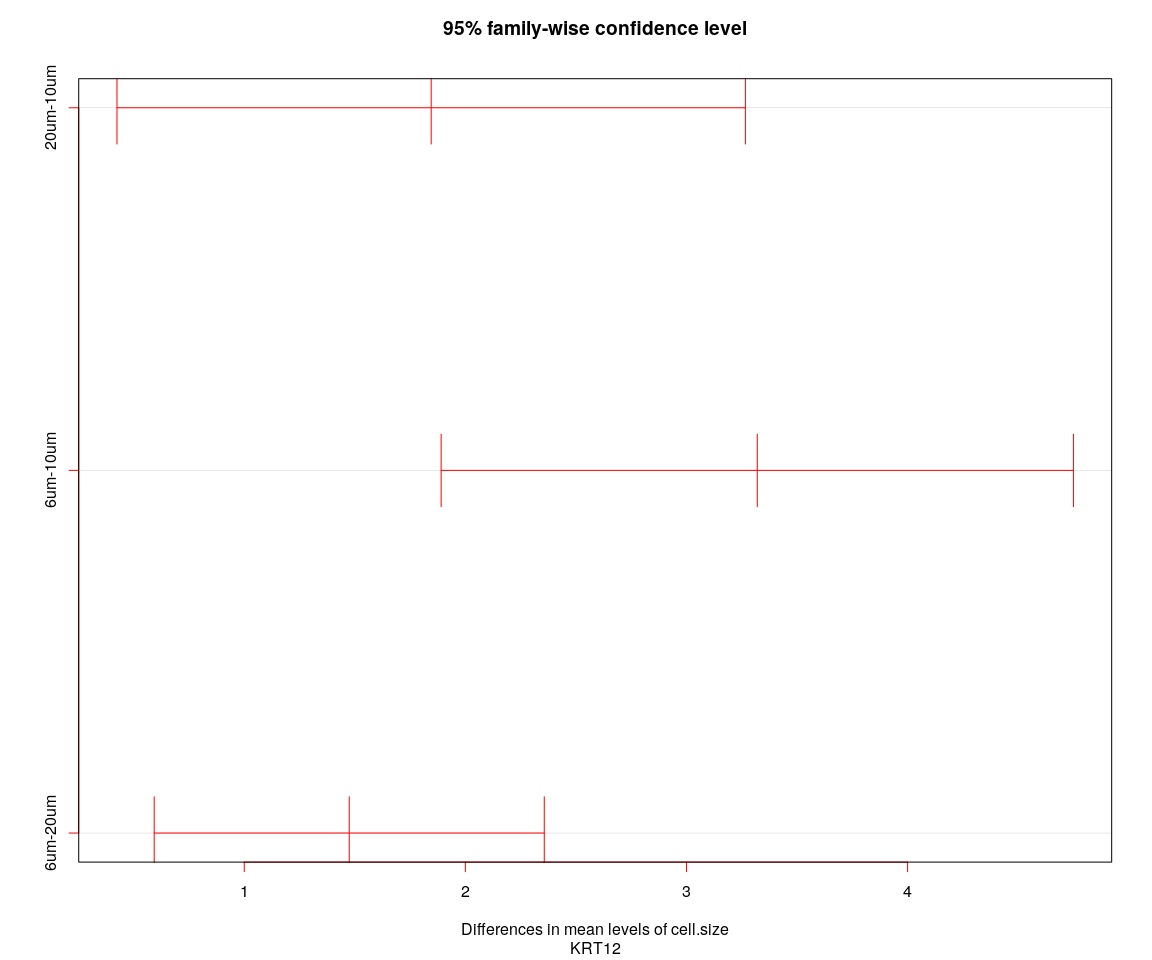
## [1] "----Gene:ITGB4----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.5316900 0.1066983 0.9566817 0.0101576  
## 6um-10um 1.0017753 0.5741506 1.4294000 0.0000007  
## 6um-20um 0.4700853 0.2062785 0.7338921 0.0001485



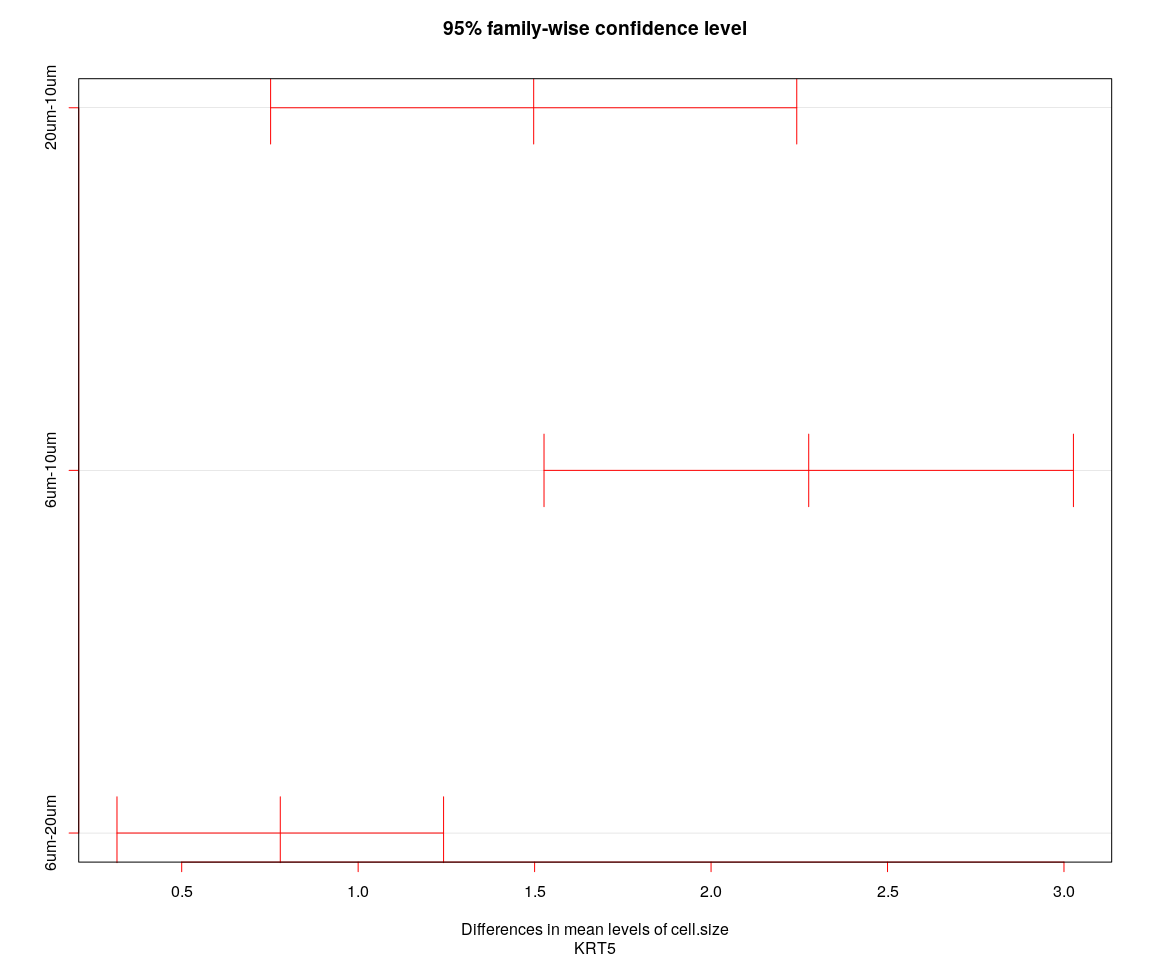
## [1] "----Gene:KRT19----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.34252575 -0.3448319 1.0298834 0.4643598  
## 6um-10um 0.33053302 -0.3610831 1.0221492 0.4935398  
## 6um-20um -0.01199273 -0.4386590 0.4146735 0.9975361



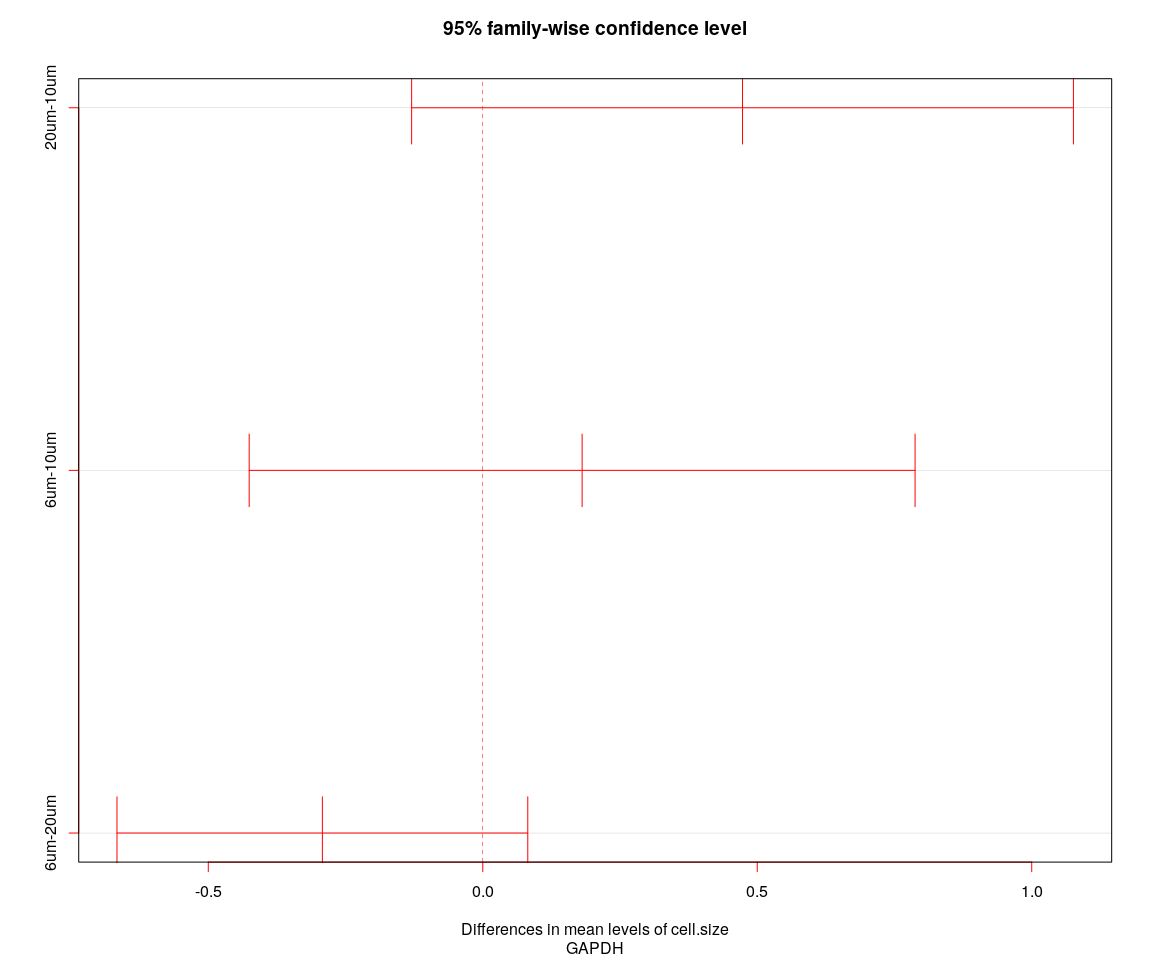
## [1] "----Gene:ACTB----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.6382092 -0.03581204 1.31223053 0.0674219  
## 6um-10um 0.2869532 -0.39124393 0.96515043 0.5742652  
## 6um-20um -0.3512560 -0.76964394 0.06713195 0.1180789



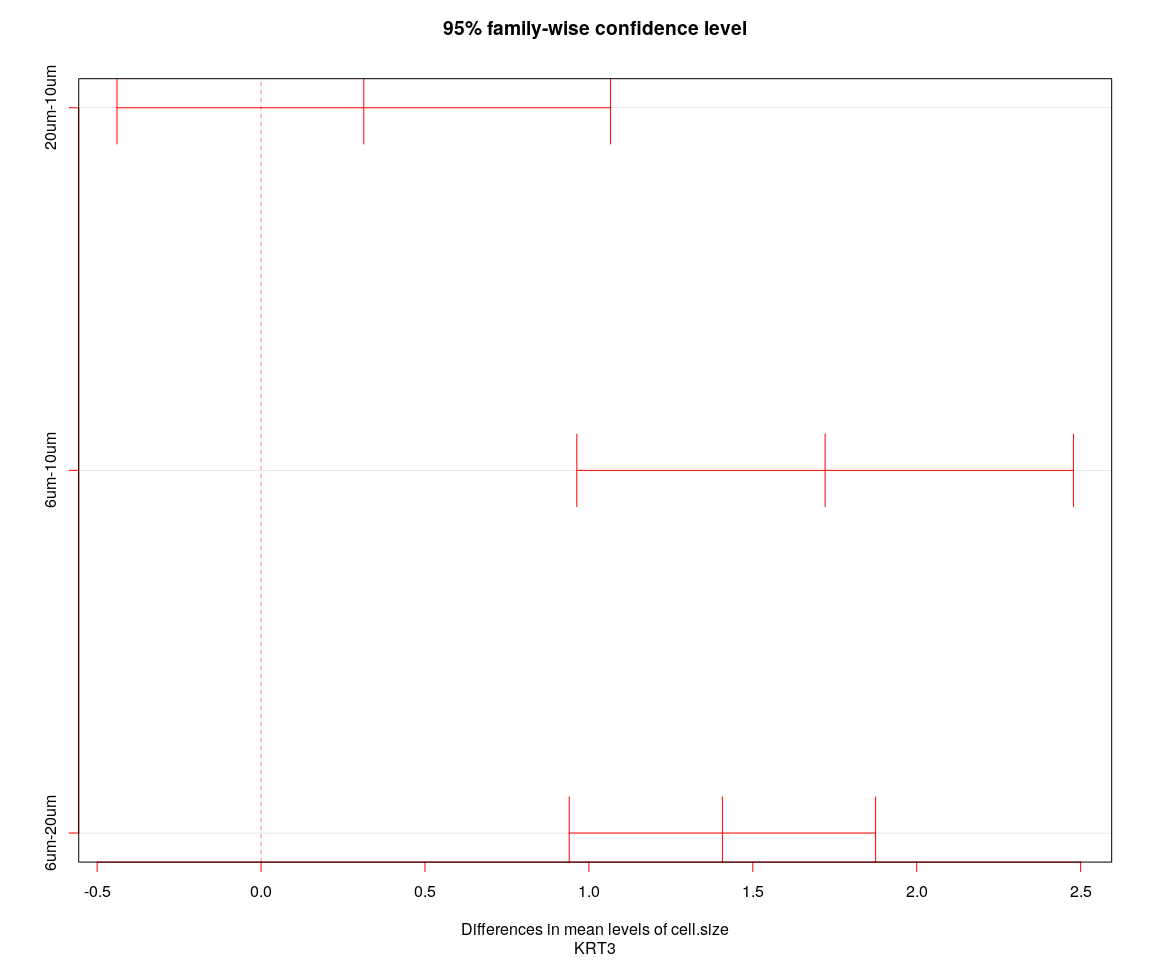
## [1] "----Gene:KRT12----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 1.845442 0.4244039 3.266481 0.0072732  
## 6um-10um 3.320201 1.8903589 4.750044 0.0000008  
## 6um-20um 1.474759 0.5926721 2.356846 0.0003878



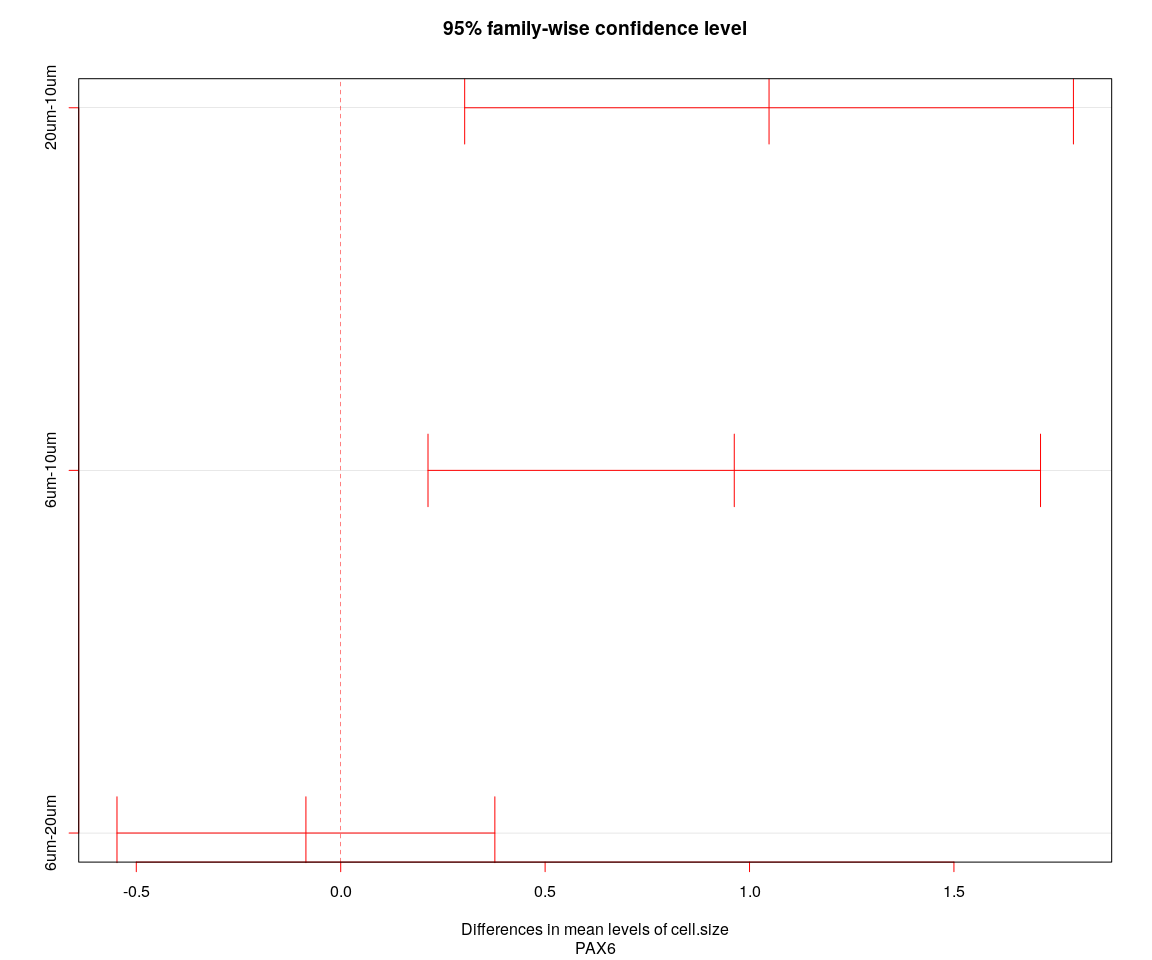
## [1] "----Gene:KRT5----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 1.4973842 0.7519524 2.242816 0.0000183  
## 6um-10um 2.2767268 1.5266766 3.026777 0.0000000  
## 6um-20um 0.7793426 0.3166277 1.242057 0.0003488



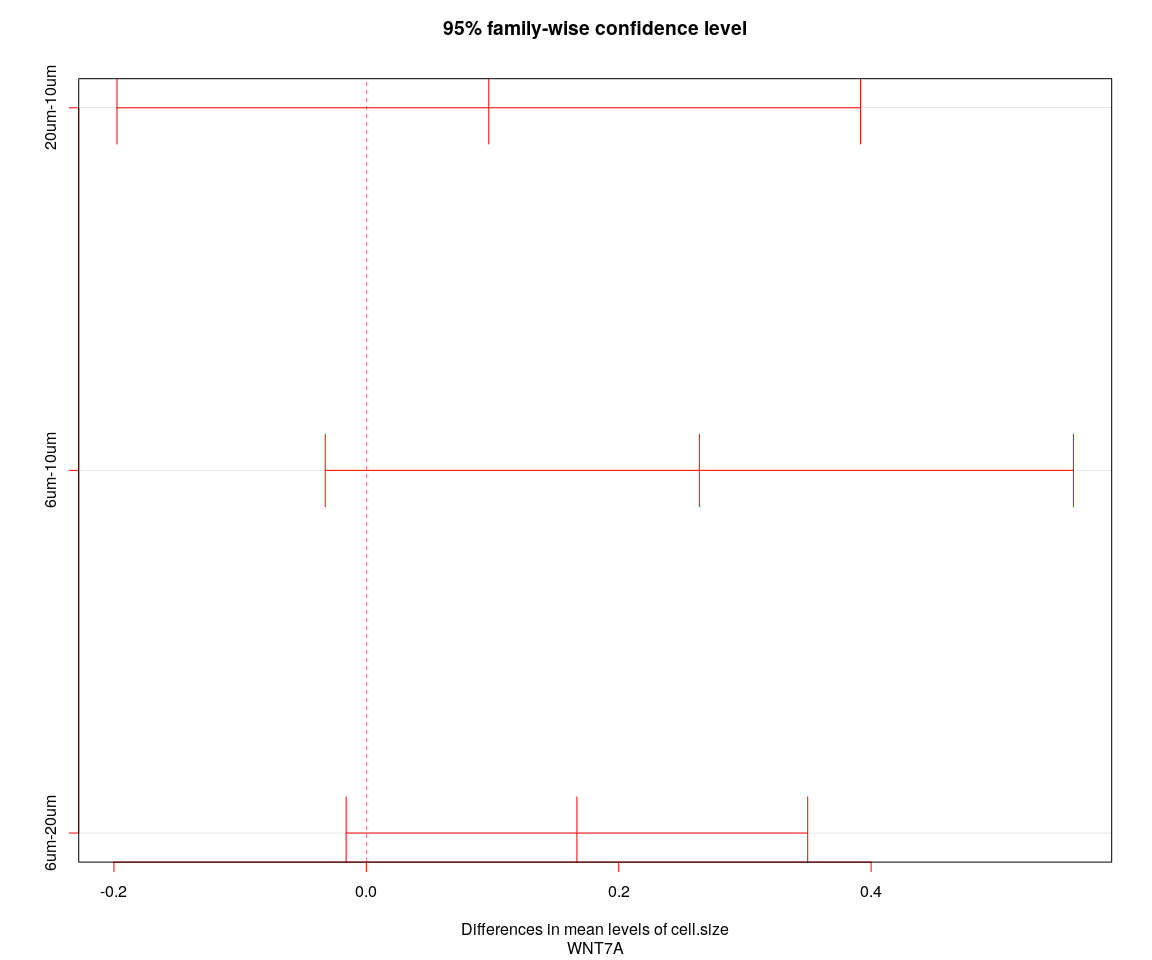
## [1] "----Gene:GAPDH----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.4731777 -0.1295920 1.07594737 0.1534204  
## 6um-10um 0.1810084 -0.4254958 0.78751249 0.7580244  
## 6um-20um -0.2921693 -0.6663290 0.08199032 0.1563974



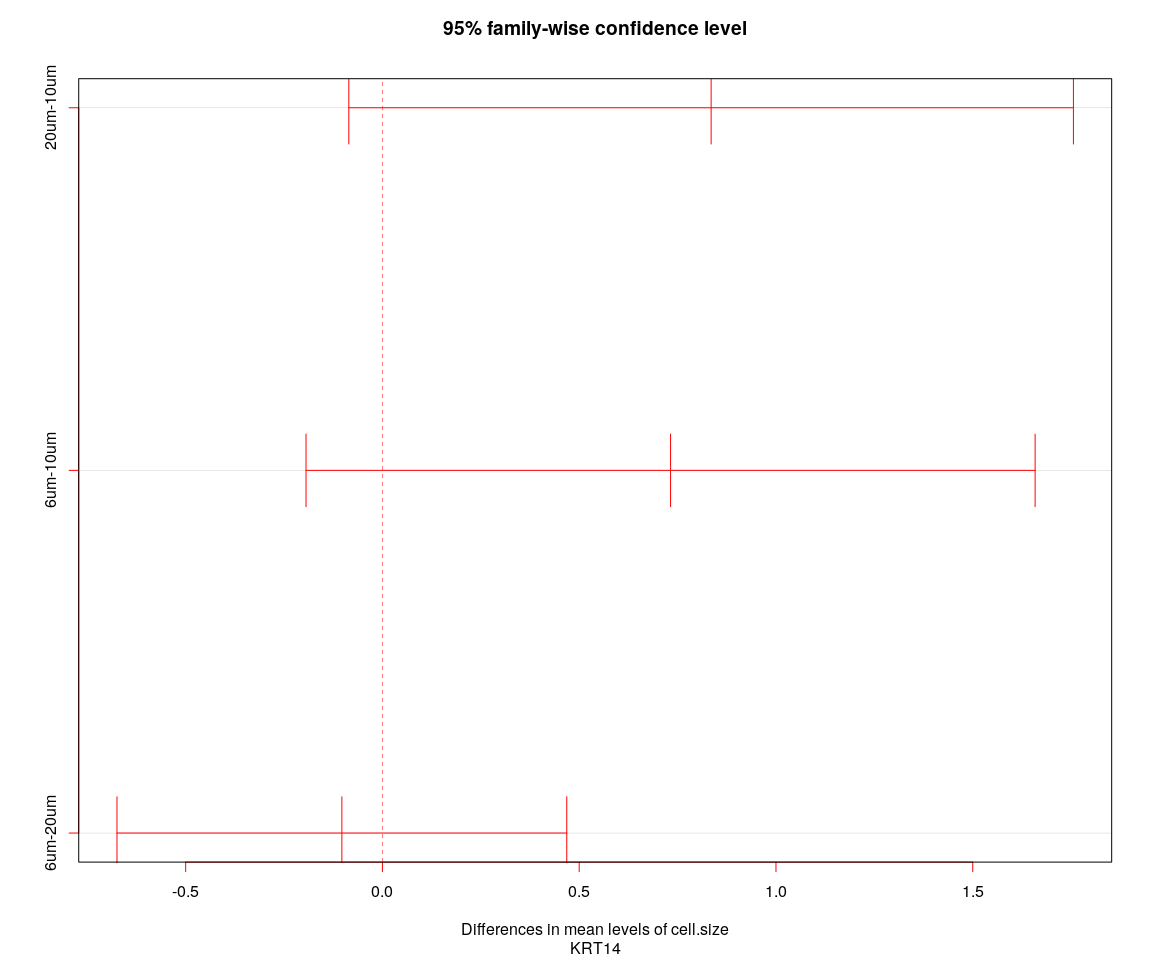
## [1] "----Gene:KRT3----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.3132777 -0.4395025 1.066058 0.5846602  
## 6um-10um 1.7206145 0.9631705 2.478059 0.0000014  
## 6um-20um 1.4073368 0.9400605 1.874613 0.0000000



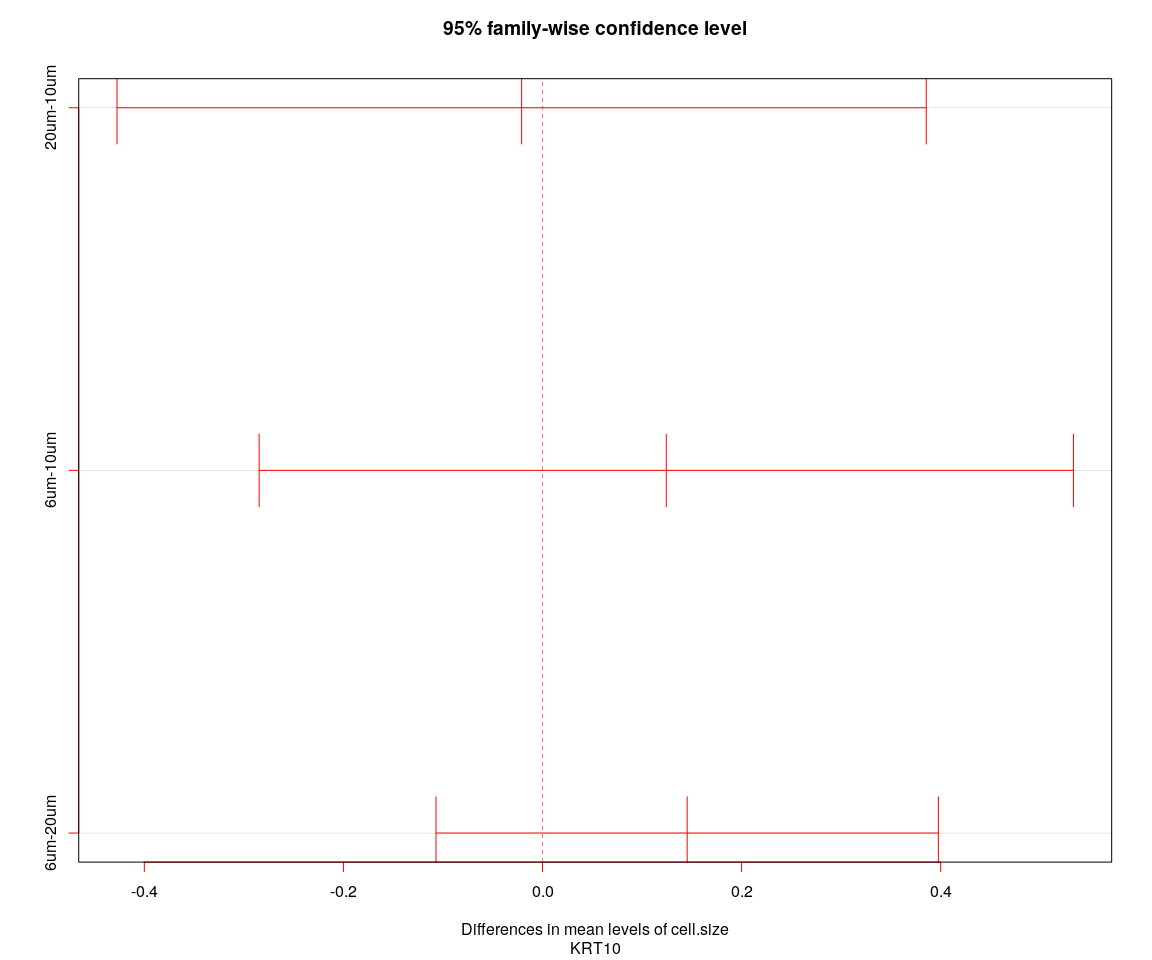
## [1] "----Gene:PAX6----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 1.04776841 0.3032485 1.7922884 0.0032750  
## 6um-10um 0.96268528 0.2135527 1.7118179 0.0080089  
## 6um-20um -0.08508312 -0.5472320 0.3770657 0.8997146



## [1] "----Gene:WNT7A----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.09701824 -0.19758351 0.3916200 0.7138799  
## 6um-10um 0.26387354 -0.03255342 0.5603005 0.0913157  
## 6um-20um 0.16685530 -0.01601404 0.3497246 0.0812238



## [1] "----Gene:KRT14----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.8352295 -0.08539373 1.7558528 0.0834903  
## 6um-10um 0.7321730 -0.19415394 1.6585000 0.1496005  
## 6um-20um -0.1030565 -0.67451867 0.4684057 0.9035664



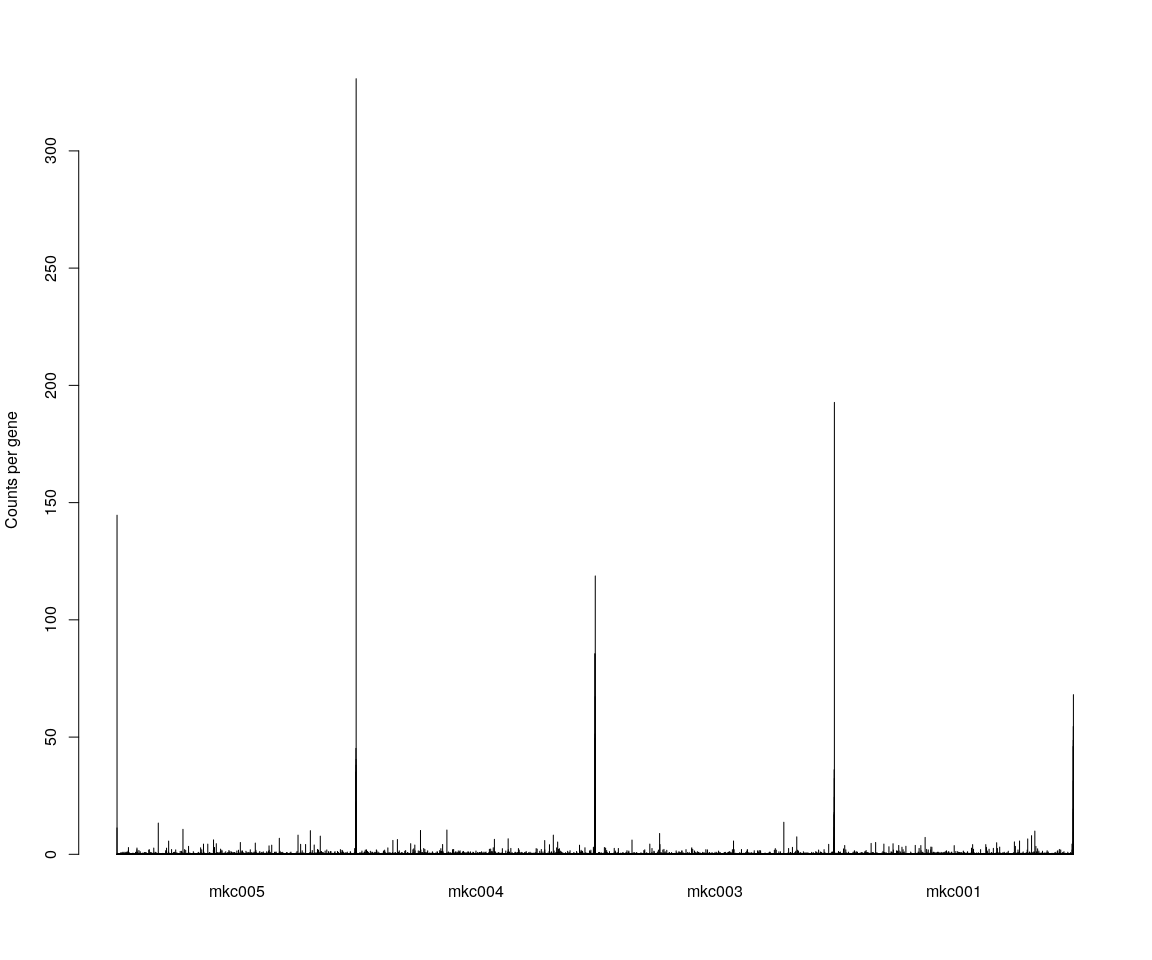
## [1] "----Gene:KRT10----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um -0.02099816 -0.4274695 0.3854732 0.9917024  
## 6um-10um 0.12432786 -0.2846618 0.5333175 0.7502241  
## 6um-20um 0.14532602 -0.1069846 0.3976366 0.3601675

## Step 1: analysis on Negative data

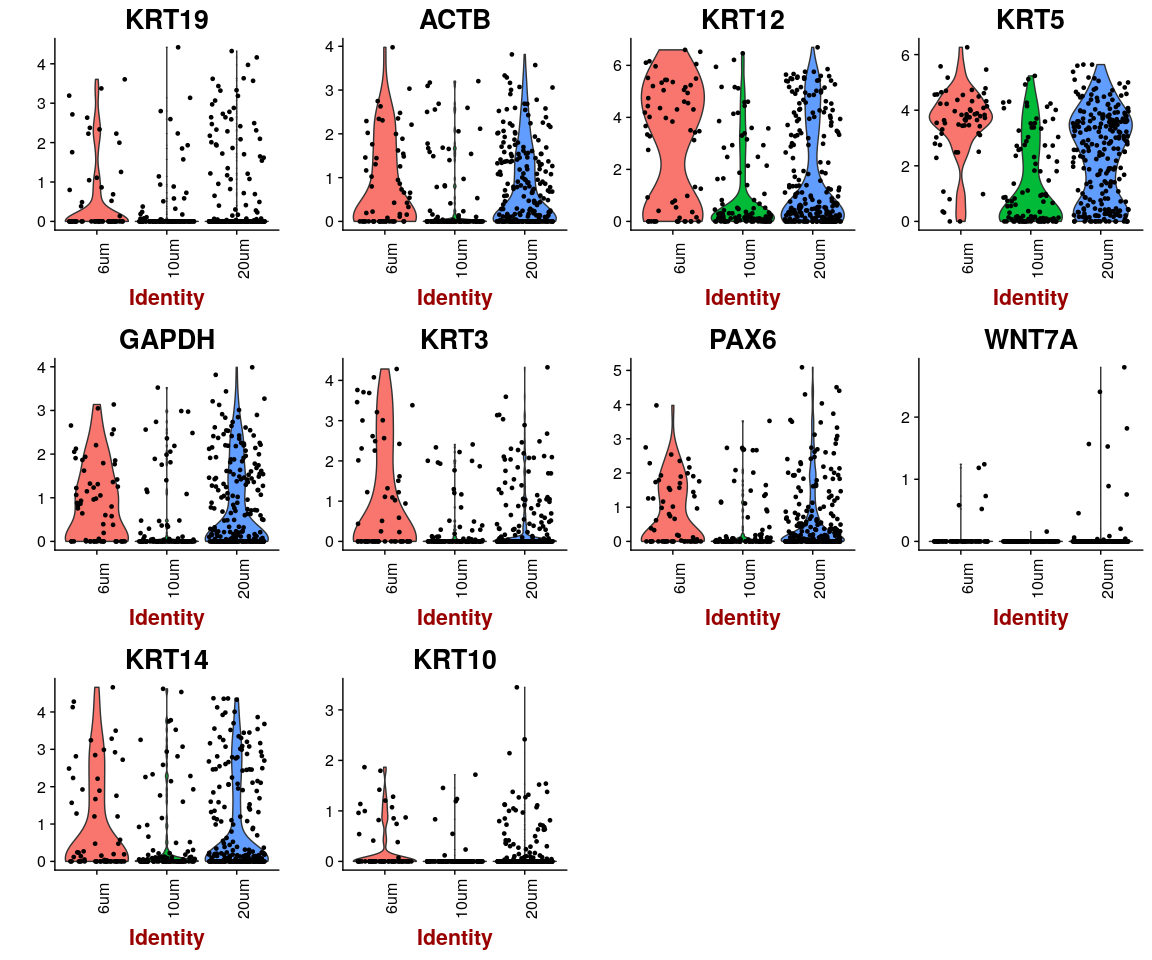
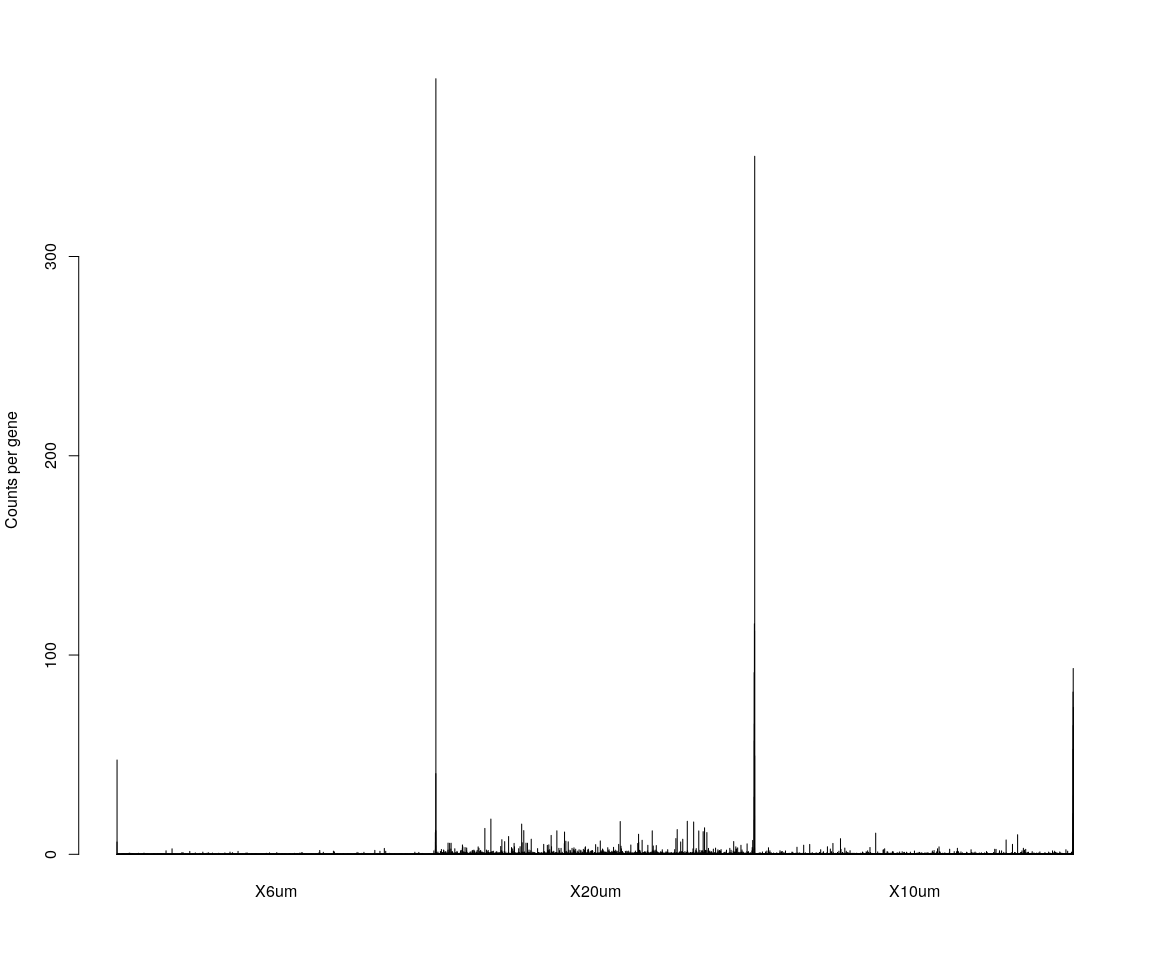
## Figure Explore

### 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**

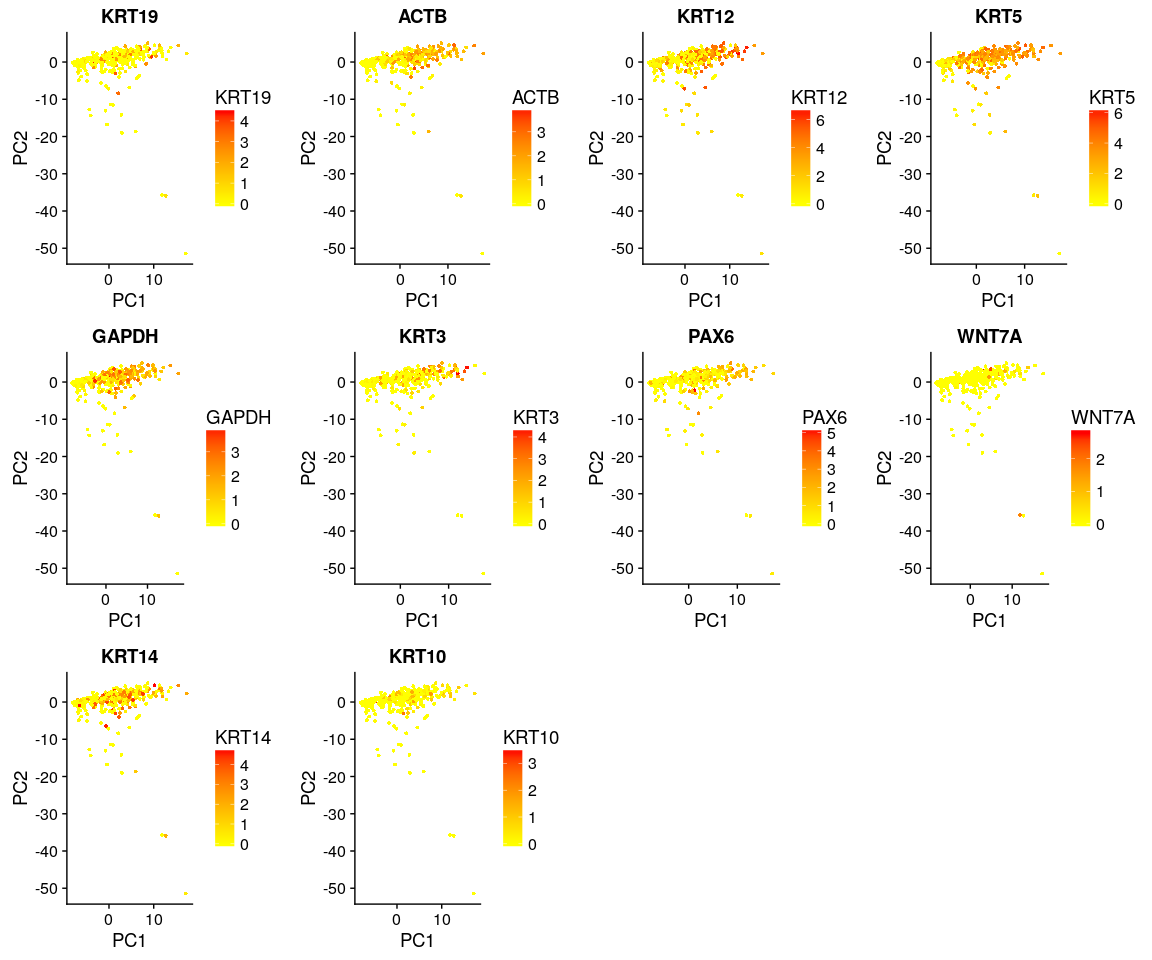
Here,do the dimensionality reduction using the PCA, tSNE method

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

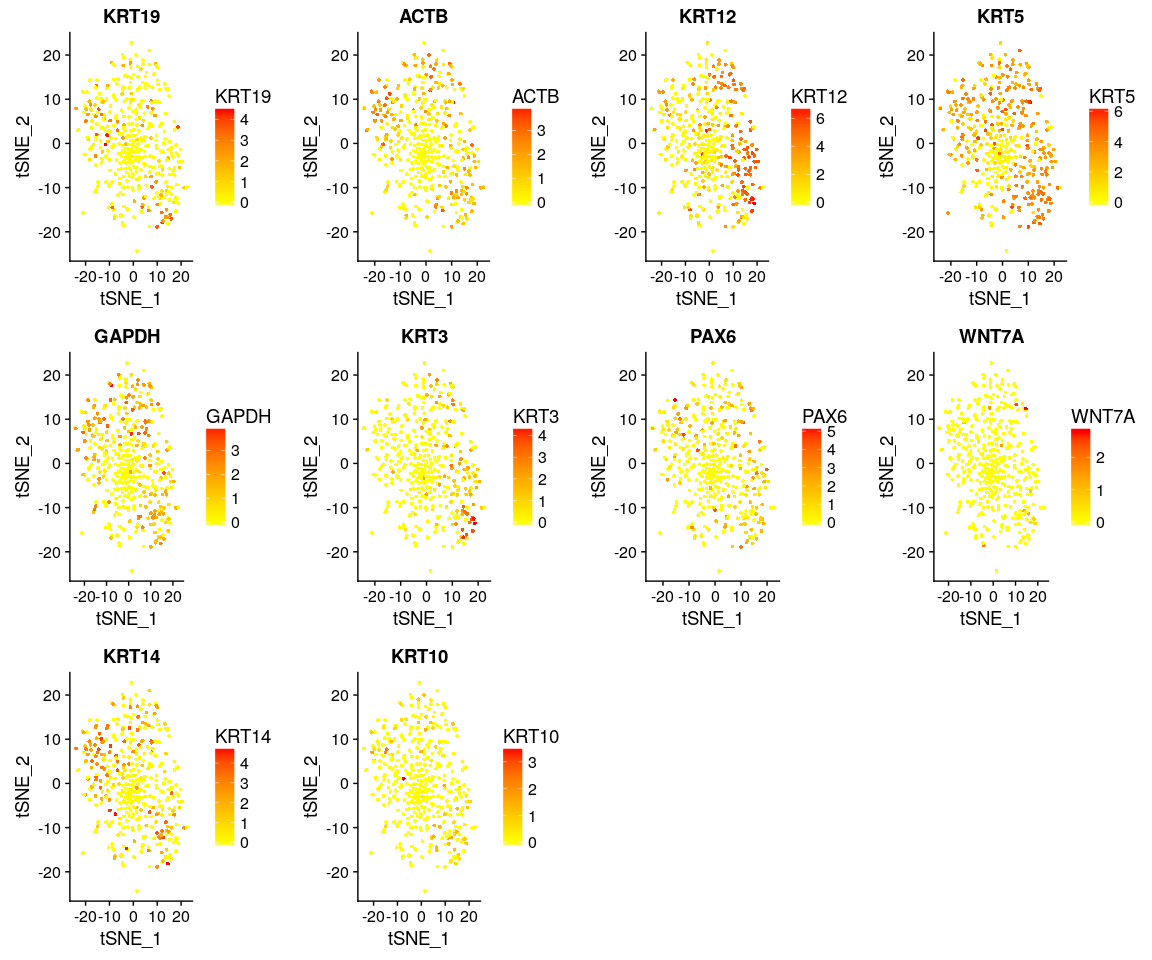
Negative.pbmc <- PCA.TSNE(object = Negative.pbmc, pcs.compute = FALSE, num.pcs = 28)

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

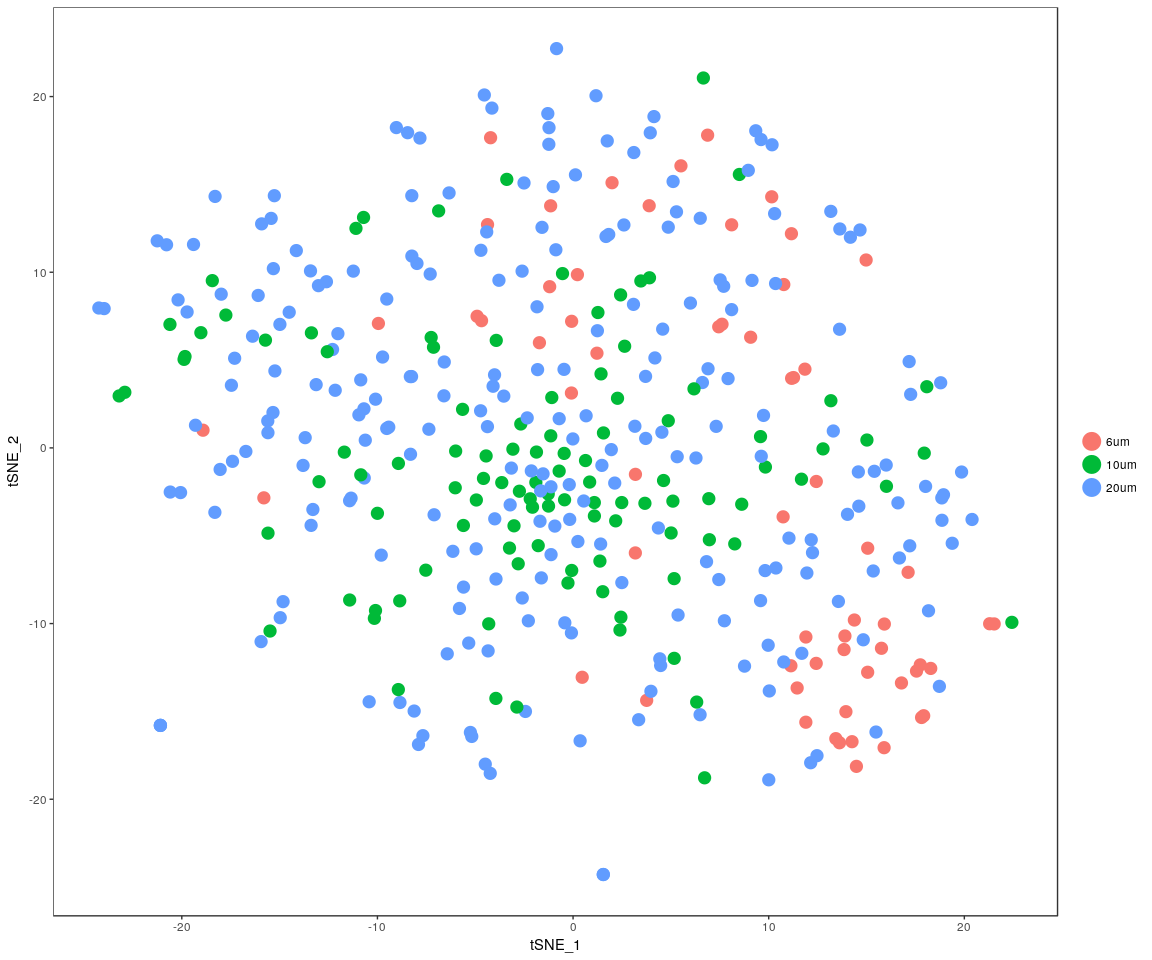
FeaturePlot(object = Negative.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Negative.pbmc@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "pca") # ITGB4 gene in part dataset



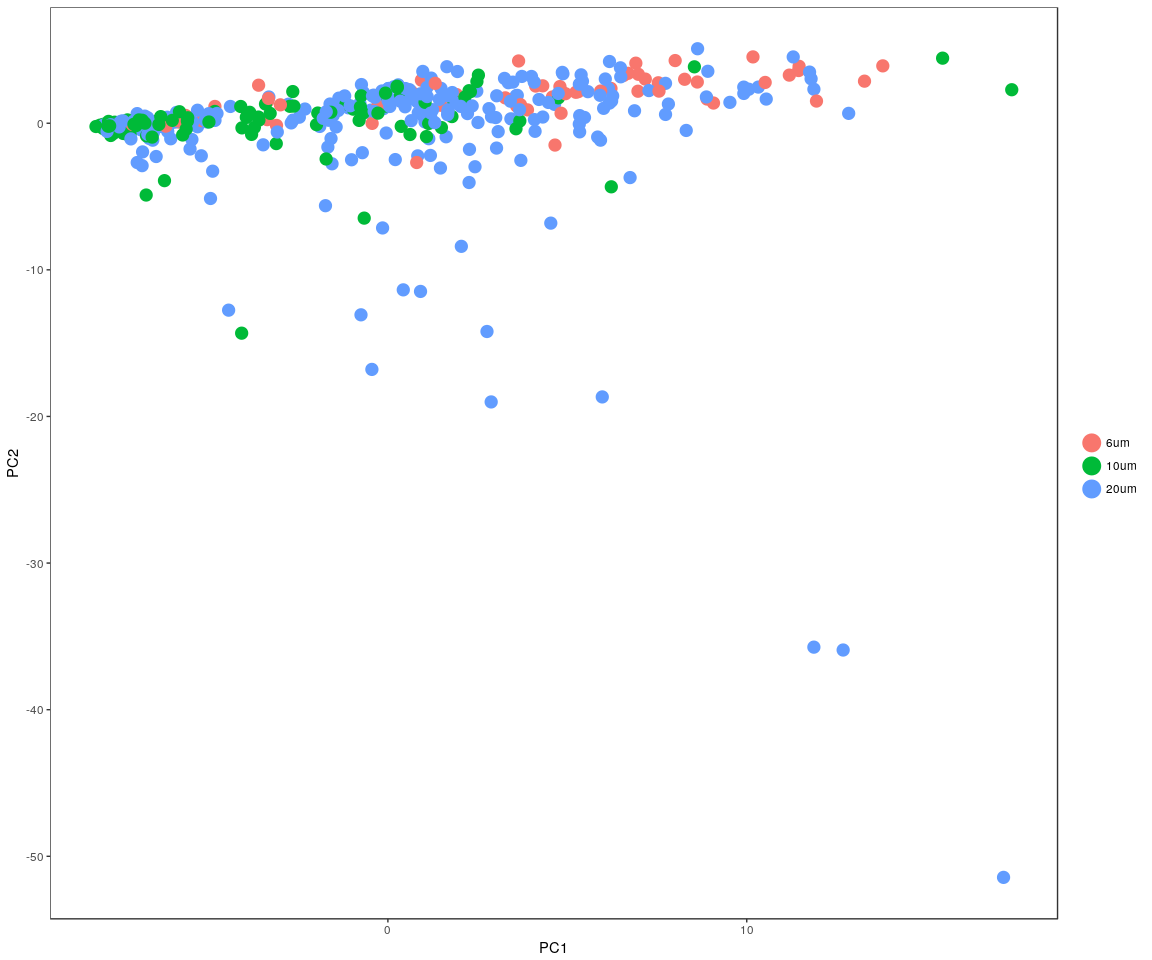
FeaturePlot(object = Negative.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(Negative.pbmc@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "tsne") # ITGB4 gene in part dataset



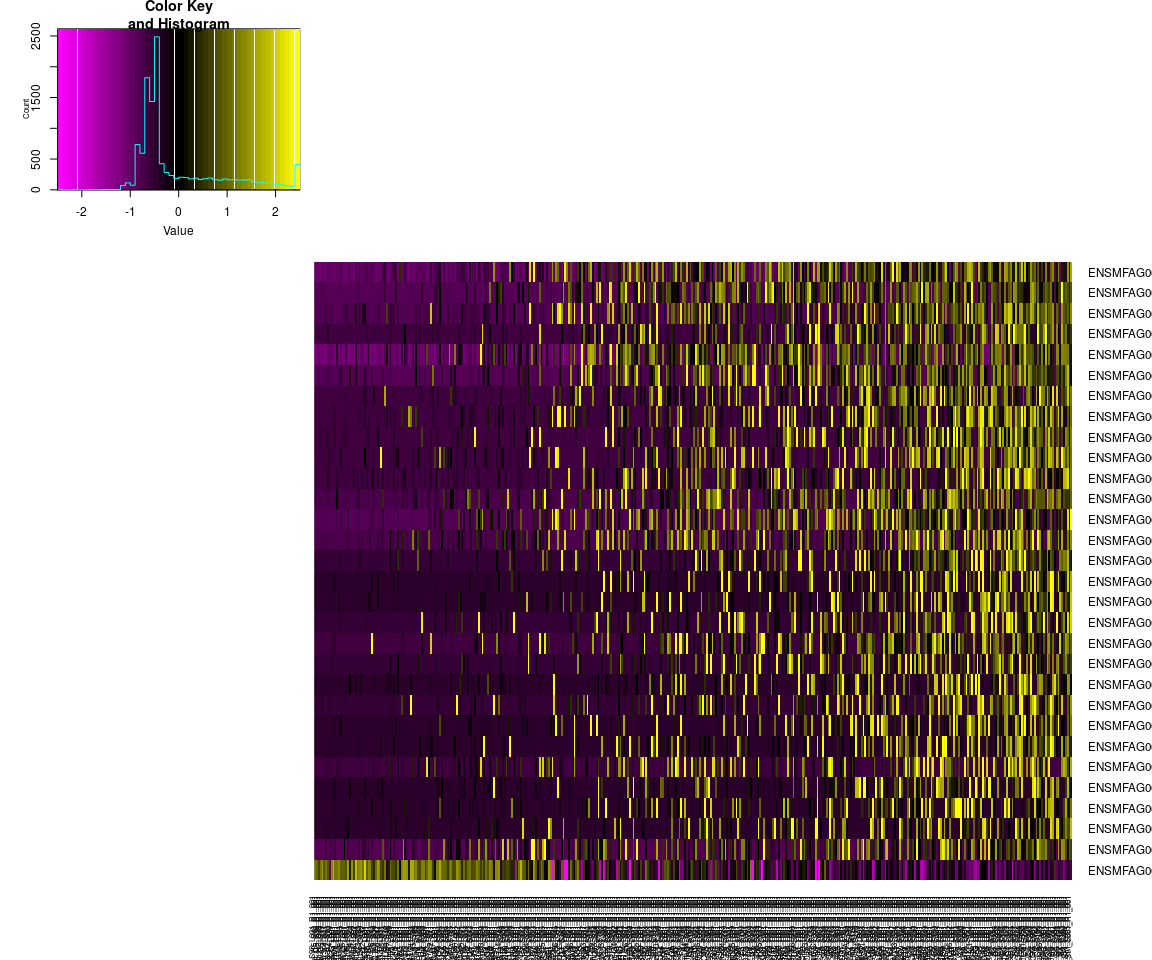
DimPlot(Negative.pbmc, reduction.use = "tsne", pt.size = 4) # grour by sample



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



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



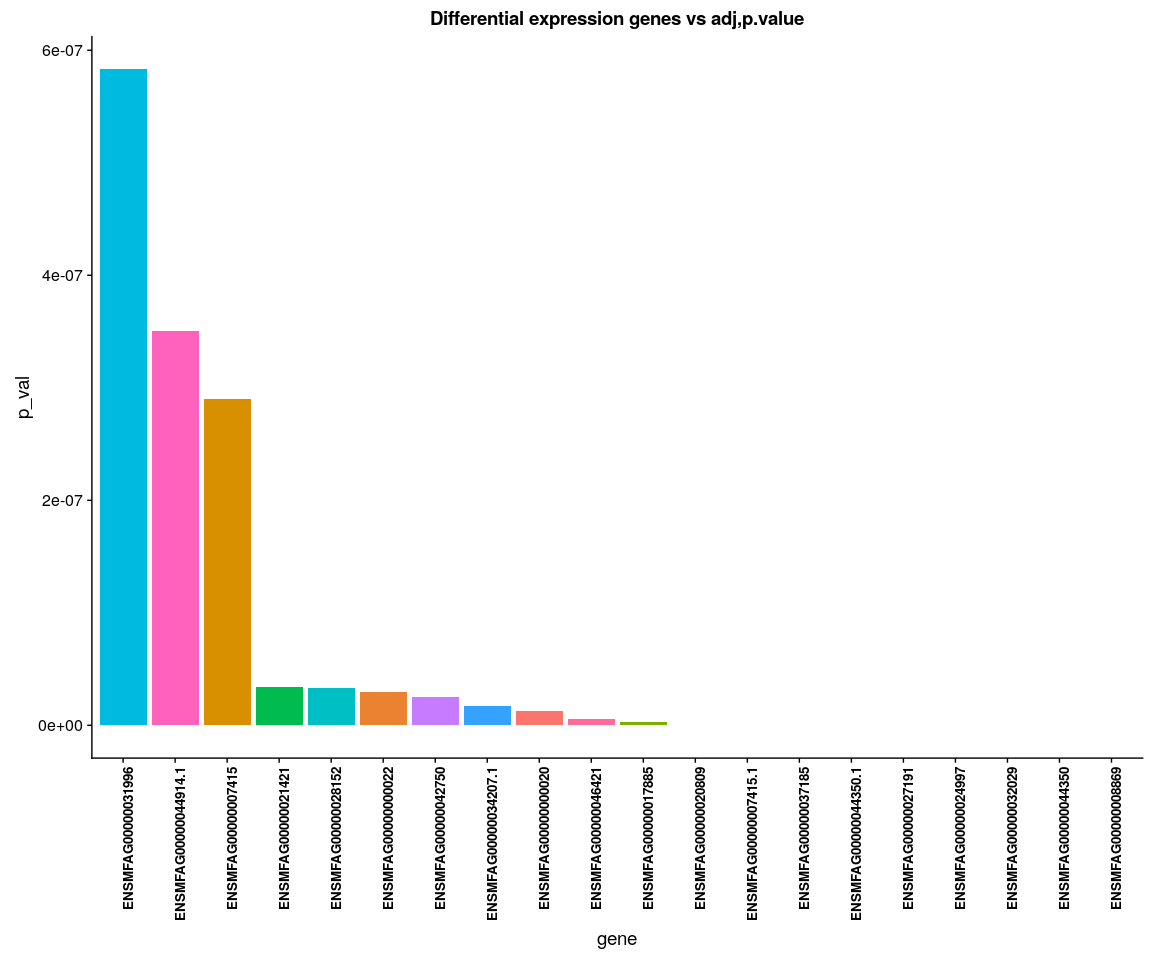
## Differential expression

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  
## ENSMFAG00000008869 7.299596e-18 2.1825795 0.581 0.166 7.497415e-14  
## ENSMFAG00000044350 1.690167e-17 0.9621157 0.194 0.177 1.735971e-13  
## ENSMFAG00000032029 1.742303e-17 0.2748837 0.210 0.389 1.789519e-13  
## ENSMFAG00000024997 1.137949e-15 1.6368161 0.645 0.374 1.168787e-11  
## ENSMFAG00000027191 4.312751e-15 -0.3668832 0.306 0.834 4.429626e-11  
## ENSMFAG00000017885 2.892364e-13 1.2963948 0.661 0.294 2.970748e-09  
## cluster gene  
## ENSMFAG00000008869 6um ENSMFAG00000008869  
## ENSMFAG00000044350 6um ENSMFAG00000044350  
## ENSMFAG00000032029 6um ENSMFAG00000032029  
## ENSMFAG00000024997 6um ENSMFAG00000024997  
## ENSMFAG00000027191 6um ENSMFAG00000027191  
## ENSMFAG00000017885 6um ENSMFAG00000017885

### Bar plot of gene’s p.val

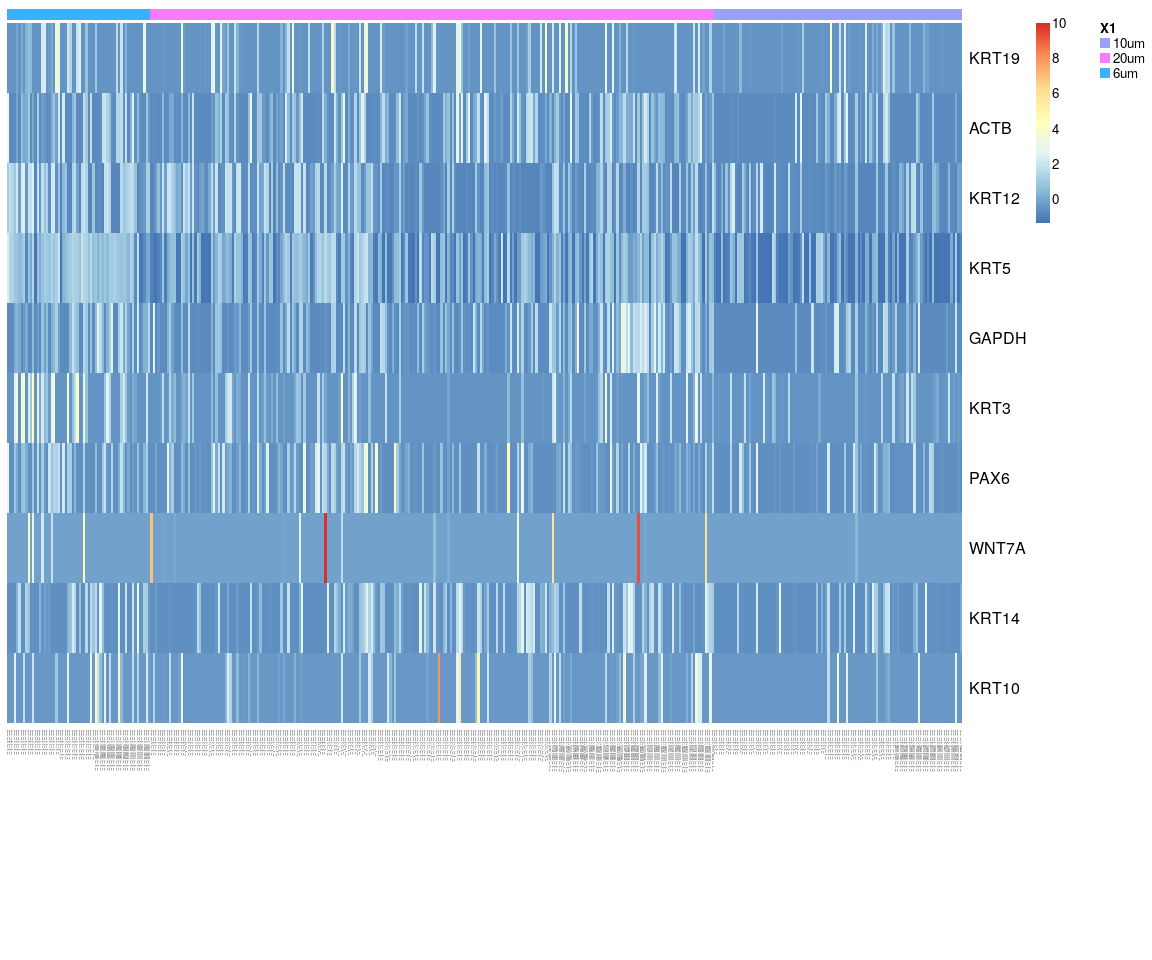


### Poistive Heatmap for important genes

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

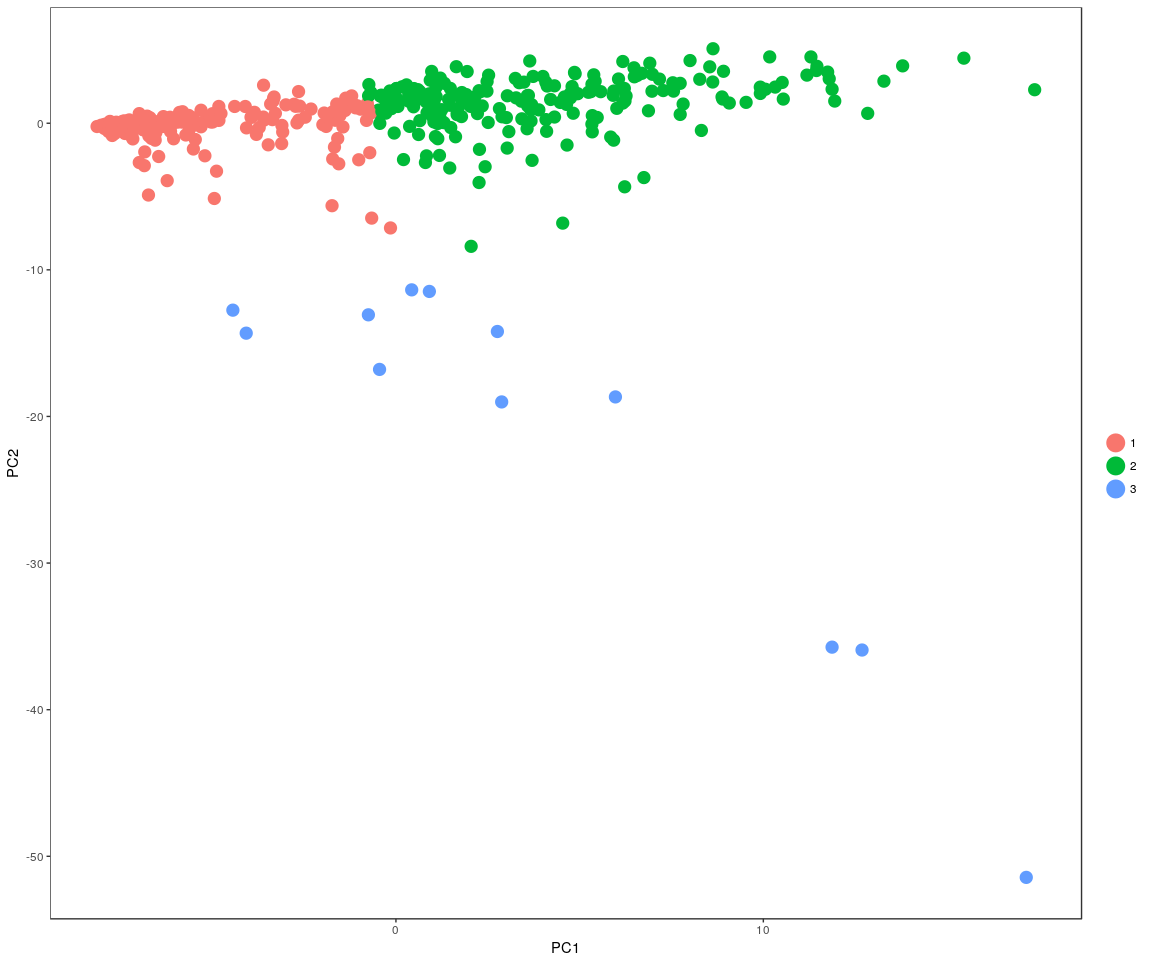
NMF::aheatmap(Negative.heatmap[[2]], Rowv = NA, Colv = NA, annCol = Negative.heatmap[[1]],   
 scale = "none")



We have find all marker genes across sample,there are 342 significant genes(adjust p-value <0.05) in all marker genes.

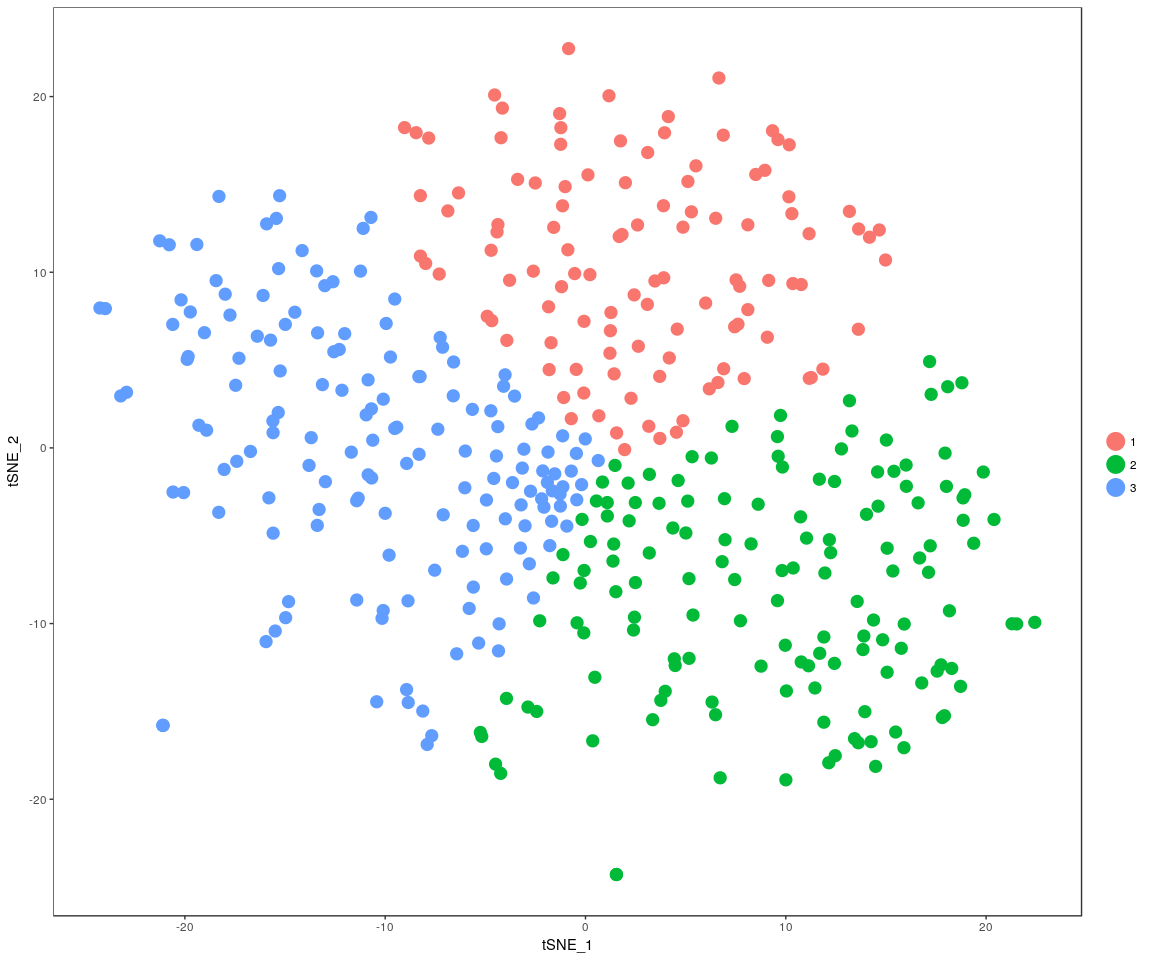
### Next,Spectral k-means clustering on single cells based on PCA

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

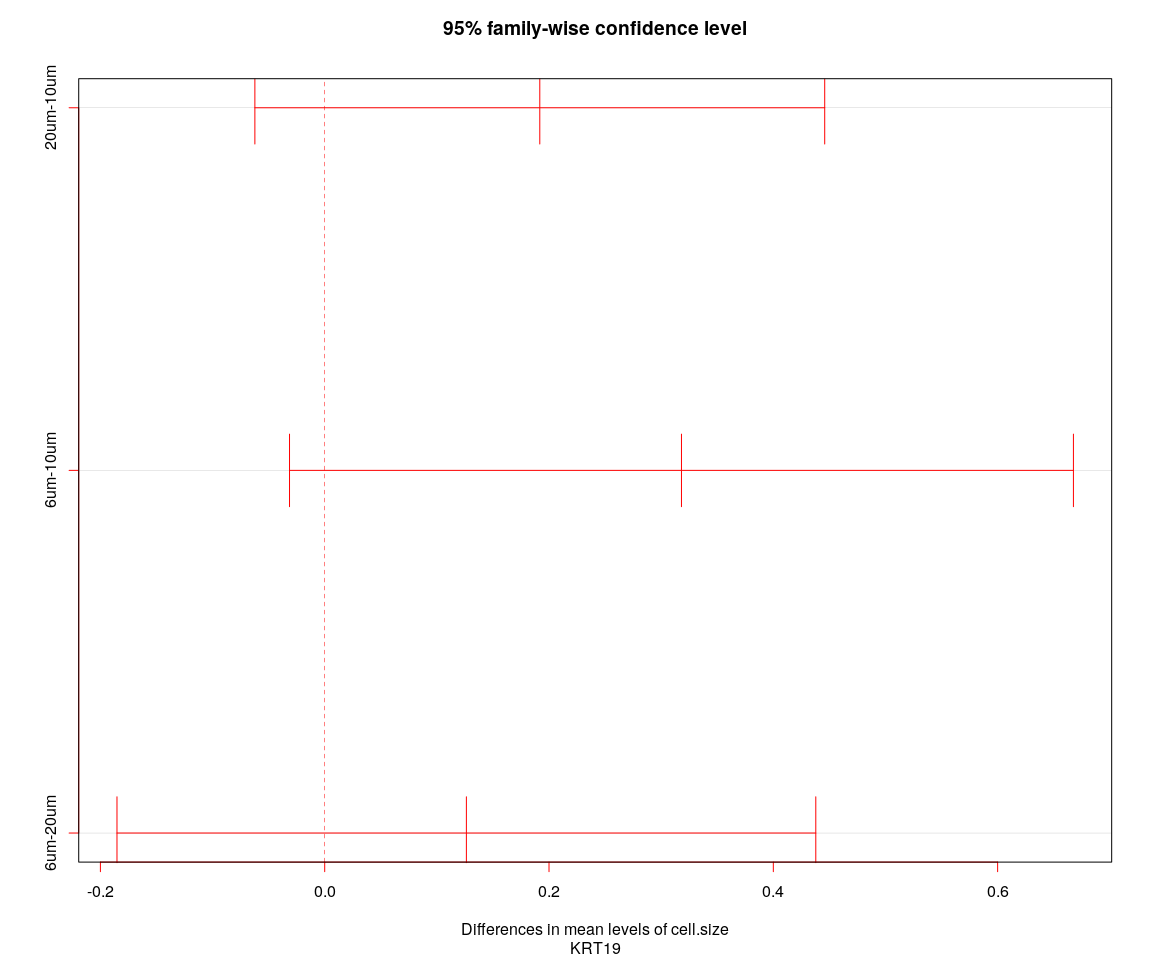
Negative.pbmc <- KClustDimension(Negative.pbmc, reduction.use = "tsne", k.use = 3)  
clusters.tsne <- Negative.pbmc@meta.data$kdimension.ident  
DimPlot(Negative.pbmc, pt.size = 4, group.by = "kdimension.ident", reduction.use = "tsne")



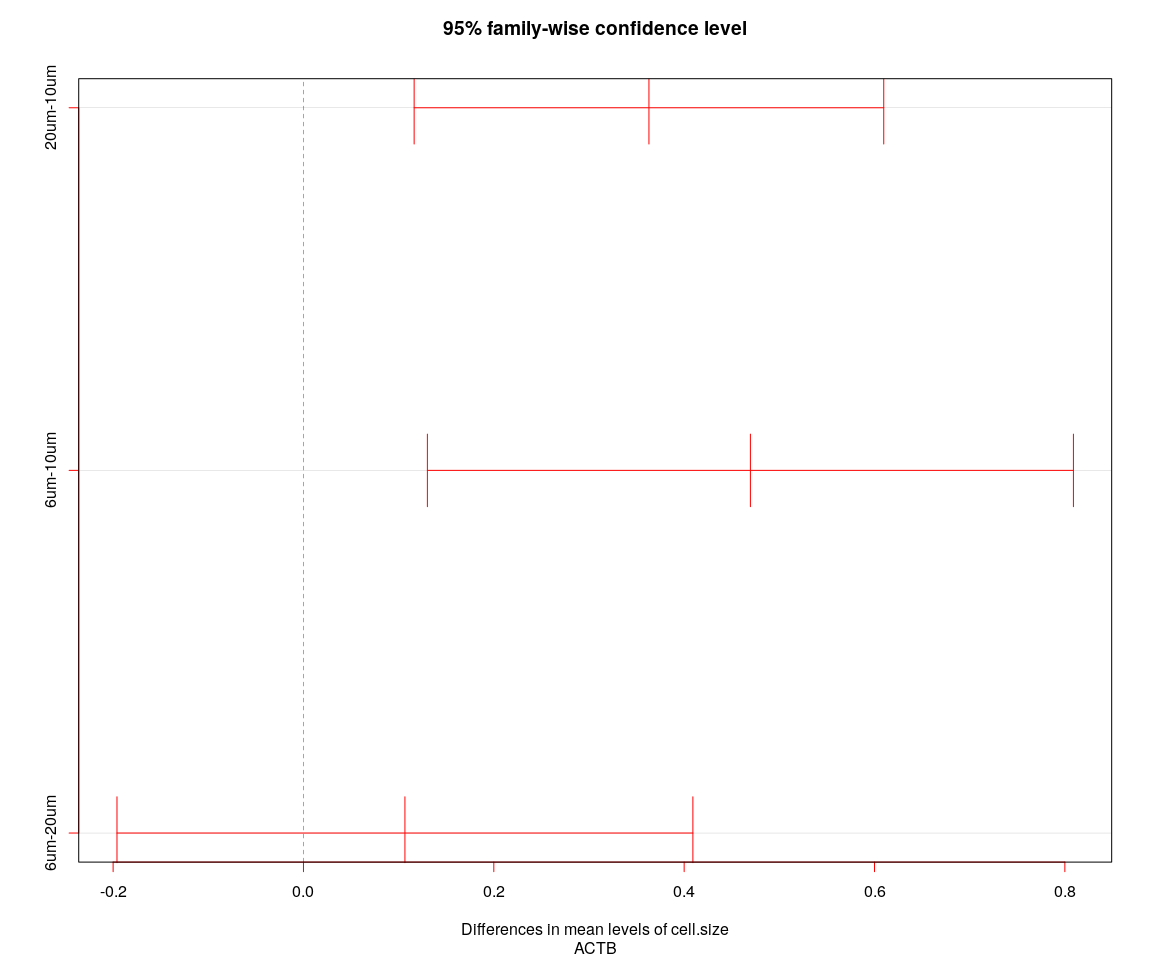
### Negative: Anova analsis:test the important genes’mean between cell size,cell groups ,whether is equal

#### anavo analysis

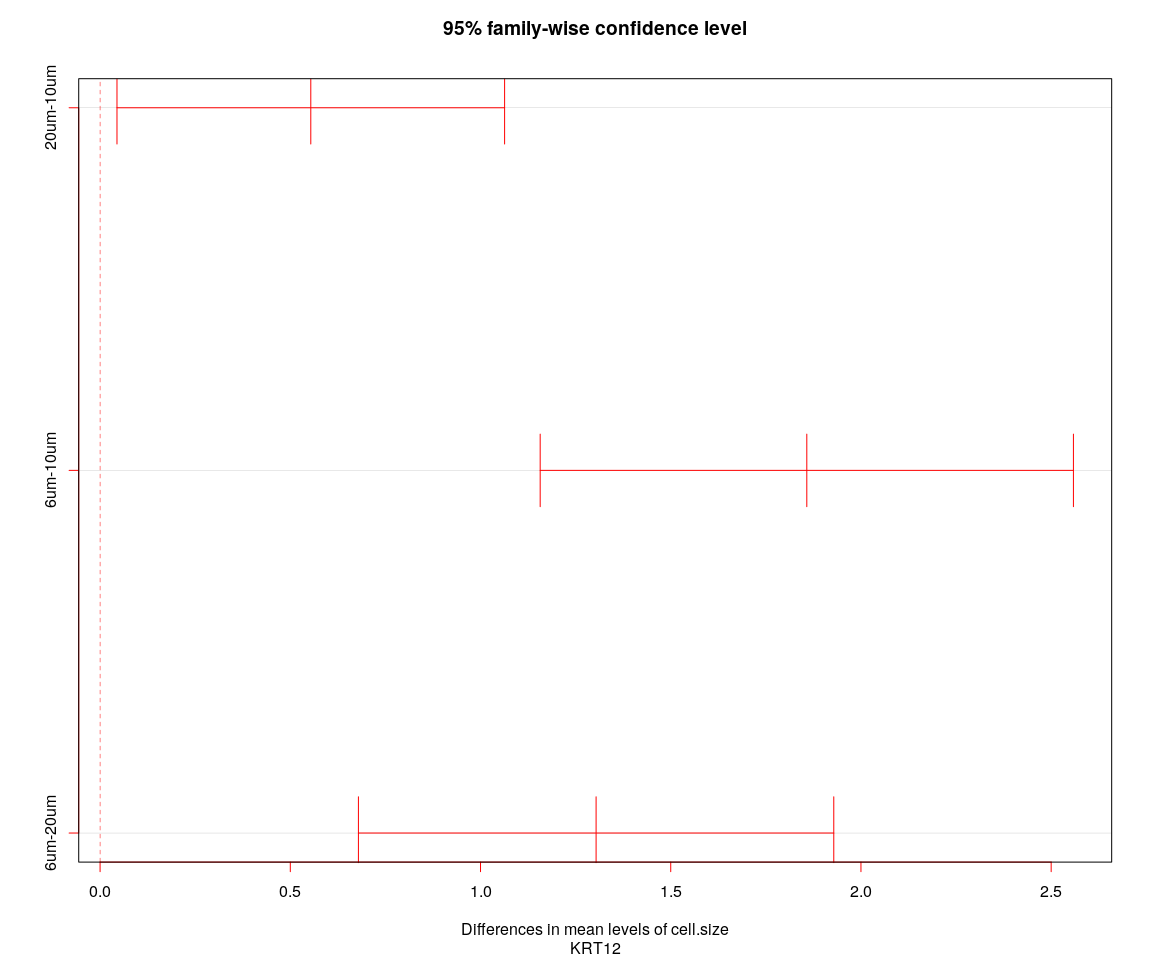
for (gene in unique(Nmonkey.imp.lognorm.melt$variable)) {  
 aov.gene <- Anova.gene(data = Nmonkey.imp.lognorm.melt, gene = gene, tuk.which = "cell.size",   
 inter = FALSE, plot.aov = FALSE)  
}



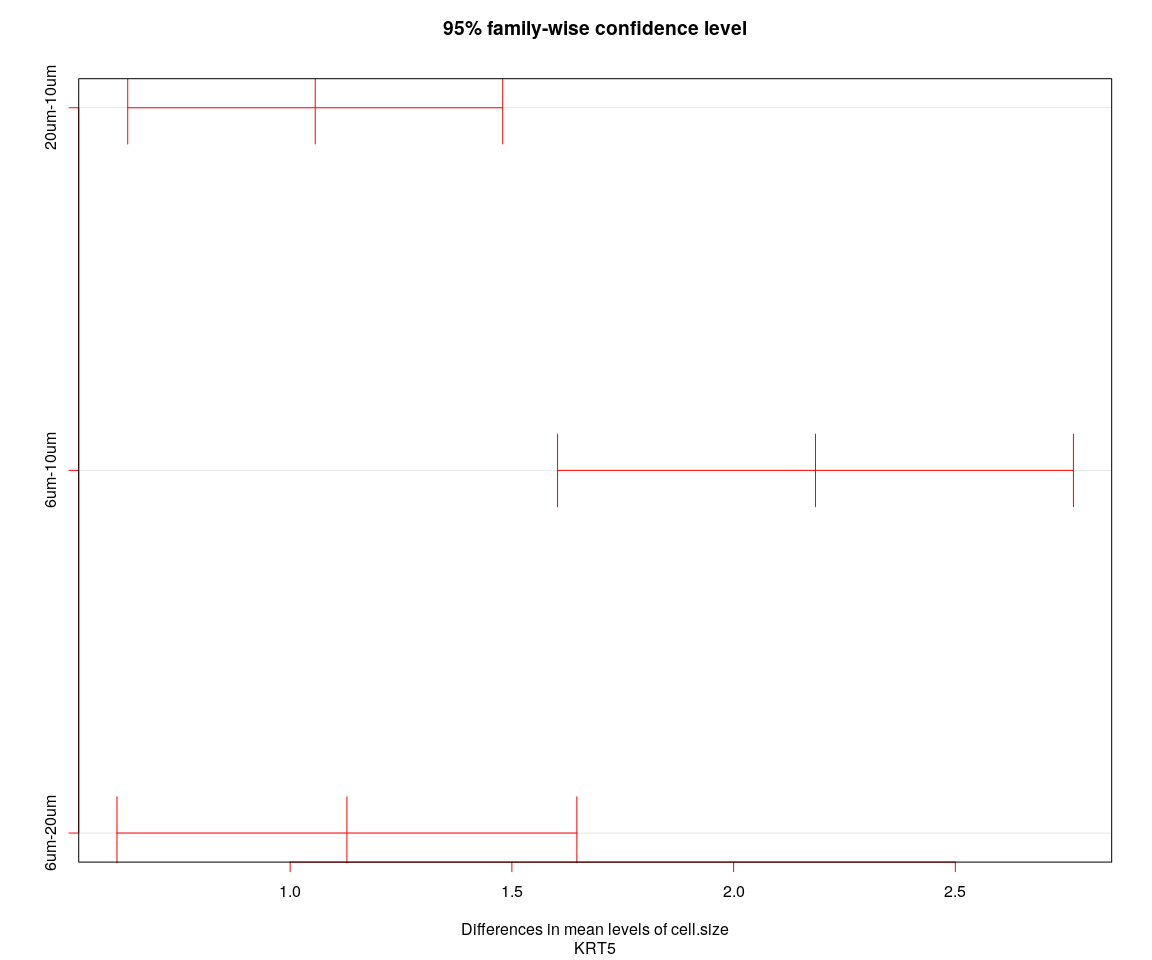
## [1] "----Gene:KRT19----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.1917770 -0.06226501 0.4458191 0.1790336  
## 6um-10um 0.3181030 -0.03137693 0.6675829 0.0829641  
## 6um-20um 0.1263259 -0.18521673 0.4378686 0.6065112



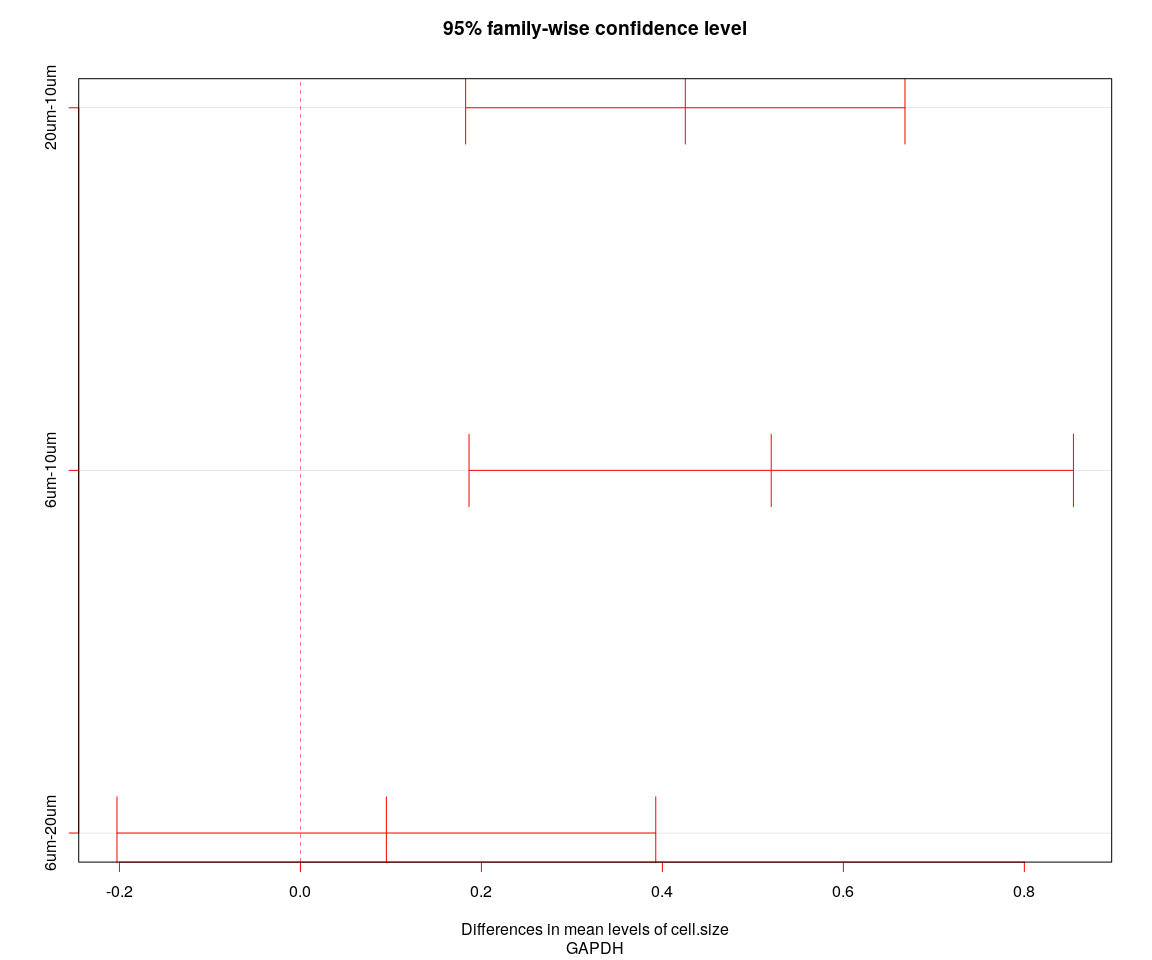
## [1] "----Gene:ACTB----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.3629990 0.1162827 0.6097152 0.0017204  
## 6um-10um 0.4696086 0.1302066 0.8090106 0.0035157  
## 6um-20um 0.1066096 -0.1959491 0.4091684 0.6853133



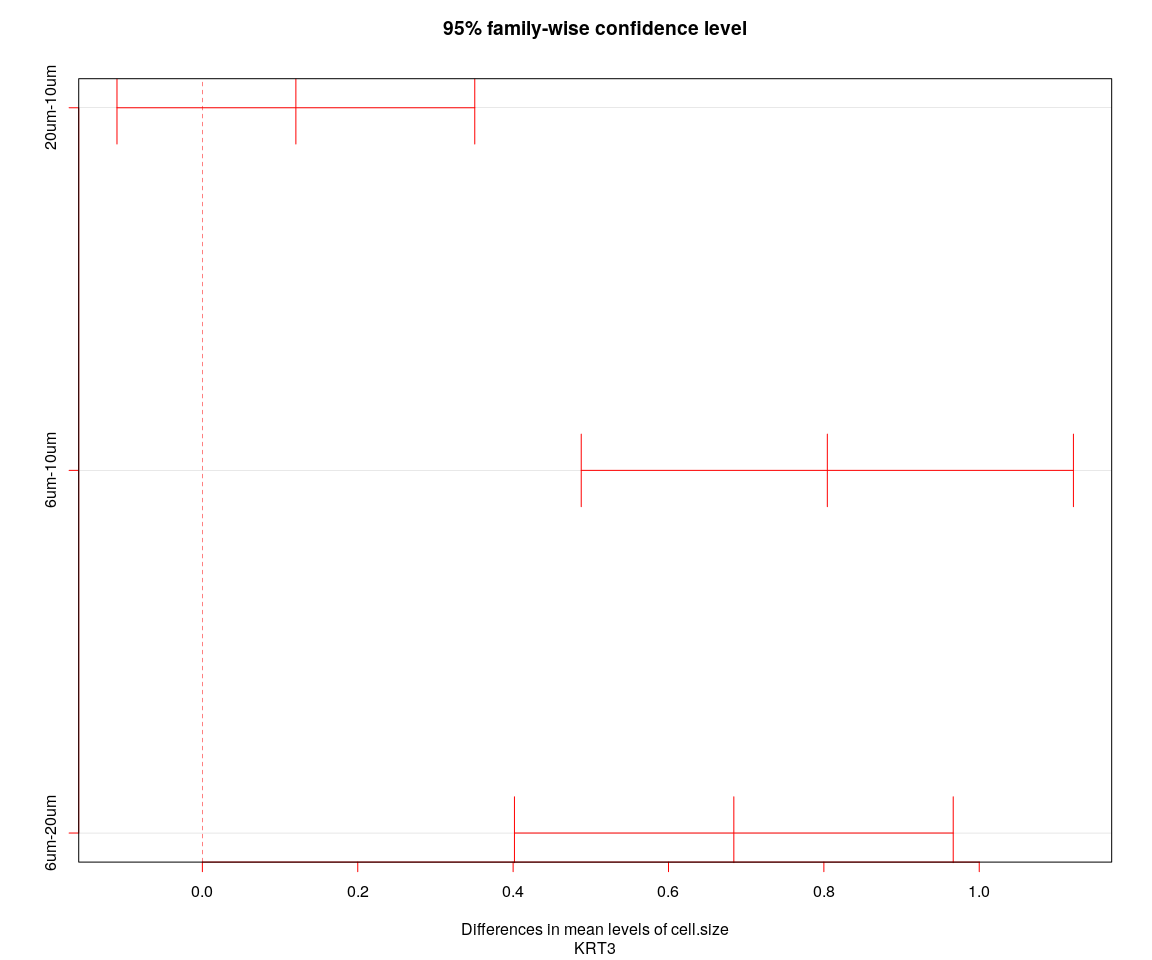
## [1] "----Gene:KRT12----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.5538611 0.04441033 1.063312 0.0293087  
## 6um-10um 1.8576686 1.15682865 2.558509 0.0000000  
## 6um-20um 1.3038075 0.67904613 1.928569 0.0000040



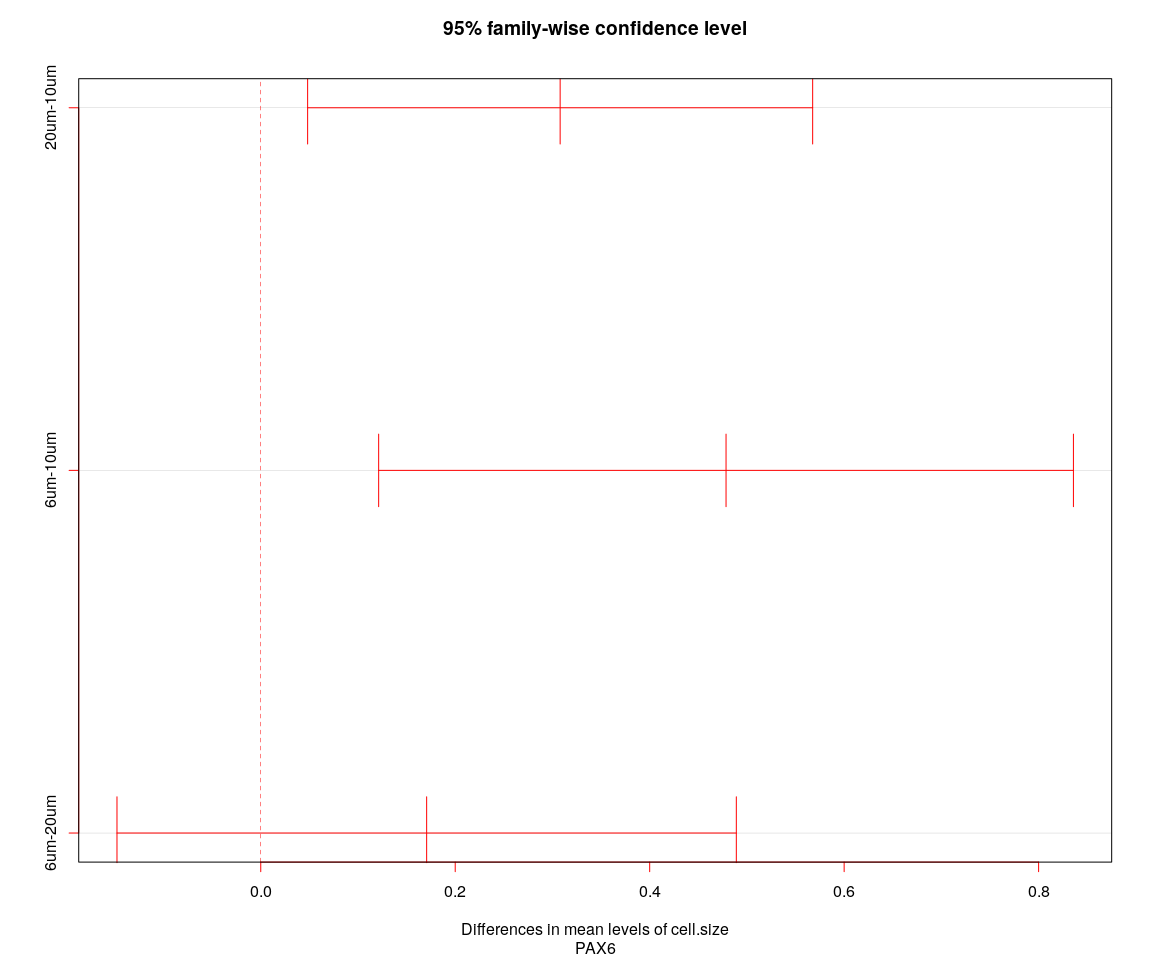
## [1] "----Gene:KRT5----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 1.056597 0.6338064 1.479388 0.0e+00  
## 6um-10um 2.184640 1.6030164 2.766264 0.0e+00  
## 6um-20um 1.128043 0.6095564 1.646530 1.4e-06



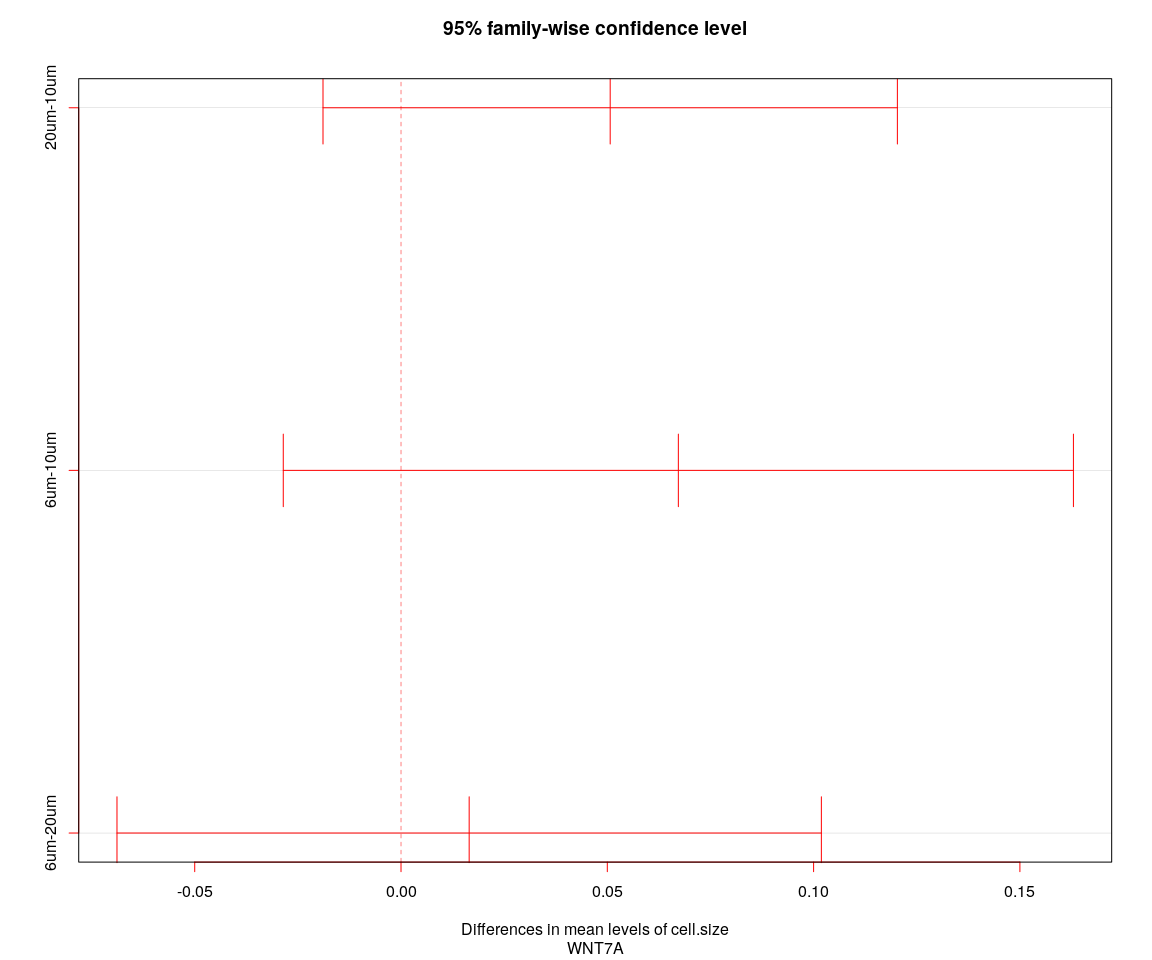
## [1] "----Gene:GAPDH----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.42531130 0.1825606 0.6680620 0.0001350  
## 6um-10um 0.52027776 0.1863311 0.8542245 0.0008174  
## 6um-20um 0.09496646 -0.2027292 0.3926621 0.7335525



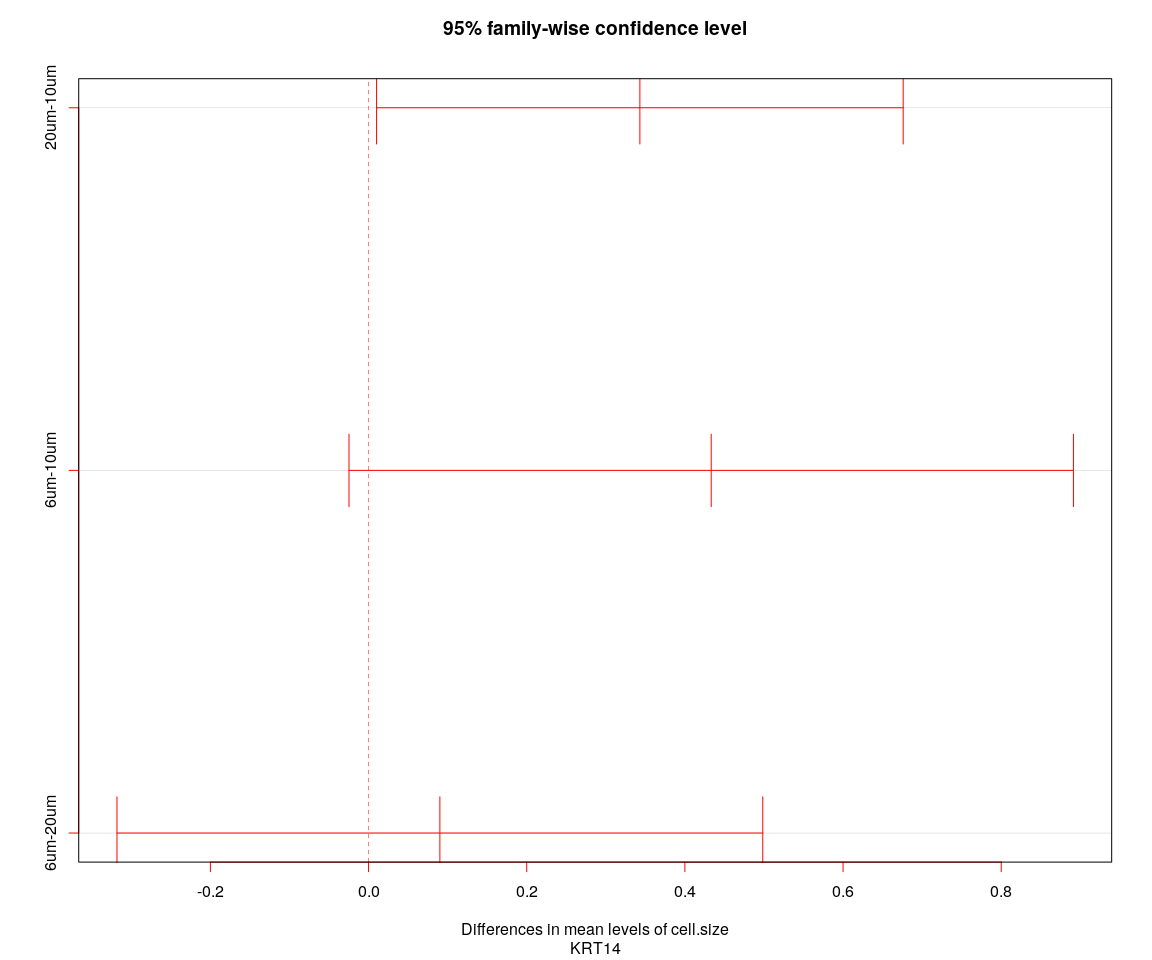
## [1] "----Gene:KRT3----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.1203449 -0.1099322 0.3506220 0.4365194  
## 6um-10um 0.8044506 0.4876635 1.1212376 0.0000000  
## 6um-20um 0.6841057 0.4017070 0.9665044 0.0000001



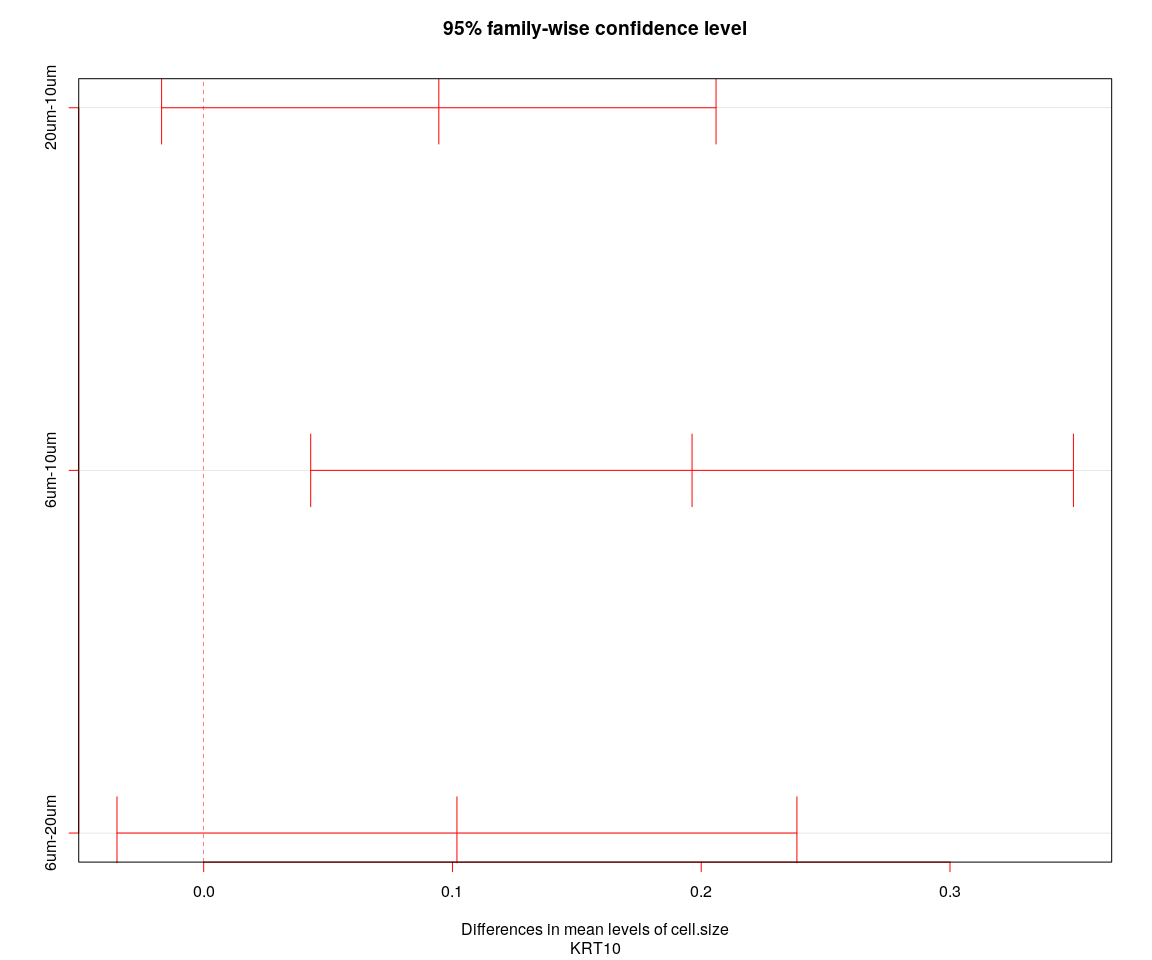
## [1] "----Gene:PAX6----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.3079159 0.04821216 0.5676196 0.0152570  
## 6um-10um 0.4785357 0.12126719 0.8358043 0.0049661  
## 6um-20um 0.1706199 -0.14786597 0.4891057 0.4186097



## [1] "----Gene:WNT7A----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.05069779 -0.01890849 0.1203041 0.2014035  
## 6um-10um 0.06721700 -0.02853879 0.1629728 0.2254950  
## 6um-20um 0.01651921 -0.06884195 0.1018804 0.8921052



## [1] "----Gene:KRT14----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.3430892 0.01015709 0.6760213 0.0416714  
## 6um-10um 0.4332863 -0.02472091 0.8912935 0.0681572  
## 6um-20um 0.0901971 -0.31809185 0.4984860 0.8617855



## [1] "----Gene:KRT10----"  
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = value ~ cell.size + cell.sample, data = genes.data)  
##   
## $cell.size  
## diff lwr upr p adj  
## 20um-10um 0.09451596 -0.01692186 0.2059538 0.1146828  
## 6um-10um 0.19631802 0.04301554 0.3496205 0.0077421  
## 6um-20um 0.10180207 -0.03485891 0.2384630 0.1872221