Human analysis on cell size

yejg

2018/1/29

### 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)

### The function will be used in the follow

## Analysis based on cell size

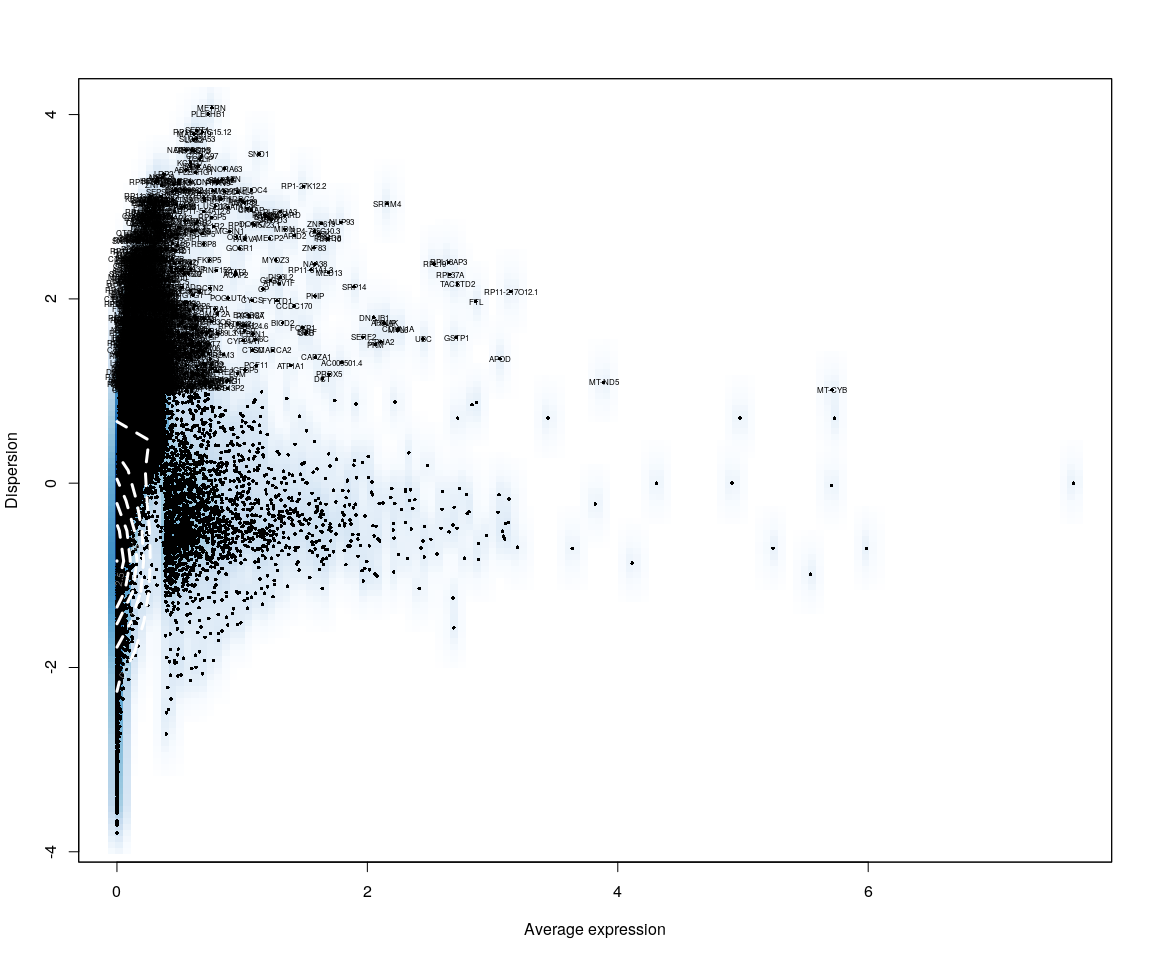
According to the previous analysis on sample group,remove the group **hc001** and cell size **2um** ### Read data ### Data QA

human.only.pro <- Load\_data(data\_dir = "../data/human.txt")  
important.genes <- c("ITGB4", "ABCB5", "KRT19", "ACTB", "KRT12", "KRT5", "GAPDH",   
 "KRT3", "PAX6", "WNT7A", "KRT14", "TP63", "KRT10")  
human.only.pro <- human.only.pro[, colnames(human.only.pro)[unlist(lapply(colnames(human.only.pro),   
 function(x) return(str\_split(x, "\_")[[1]][2]))) %in% c("10um", "20um", "6um")]]  
human.only.pro <- human.only.pro[, colnames(human.only.pro)[!unlist(lapply(colnames(human.only.pro),   
 function(x) return(str\_split(x, "\_")[[1]][1]))) %in% c("hc001", "shoutiao")]]  
# human.only.pro<-Simplify\_Select(human.only.pro)

### Create Seurat object and not caculate DESeq,but not set **min.cells** and **min.genes**

# only select the cells contain 10 genes expressed at least,select the genes  
# must be expressed in two cells at least  
human.all.DESeq <- DESeq\_SeuratObj(X = human.only.pro, DESq = FALSE, min.cells = 10,   
 min.genes = 2)

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



all.sample.group <- unlist(lapply(human.all.DESeq@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][1])))  
all.sample.size <- unlist(lapply(human.all.DESeq@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
# reset ident  
human.all.DESeq <- SetIdent(human.all.DESeq, cells.use = human.all.DESeq@cell.names,   
 ident.use = all.sample.size)

## Figure Explore

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

# human.Lognorm<-data.frame(as.matrix(LogNormalize(human.only.pro,display.progress  
# = FALSE)))  
# human.imp.lognorm<-data.frame(t(human.Lognorm[important.genes,]))  
human.imp.lognorm <- data.frame(FetchData(human.all.DESeq, vars.all = important.genes[important.genes %in%   
 rownames(human.all.DESeq@raw.data)]))  
human.imp.lognorm$cell.size <- unlist(lapply(rownames(human.imp.lognorm), function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
human.imp.lognorm.melt <- melt(human.imp.lognorm)

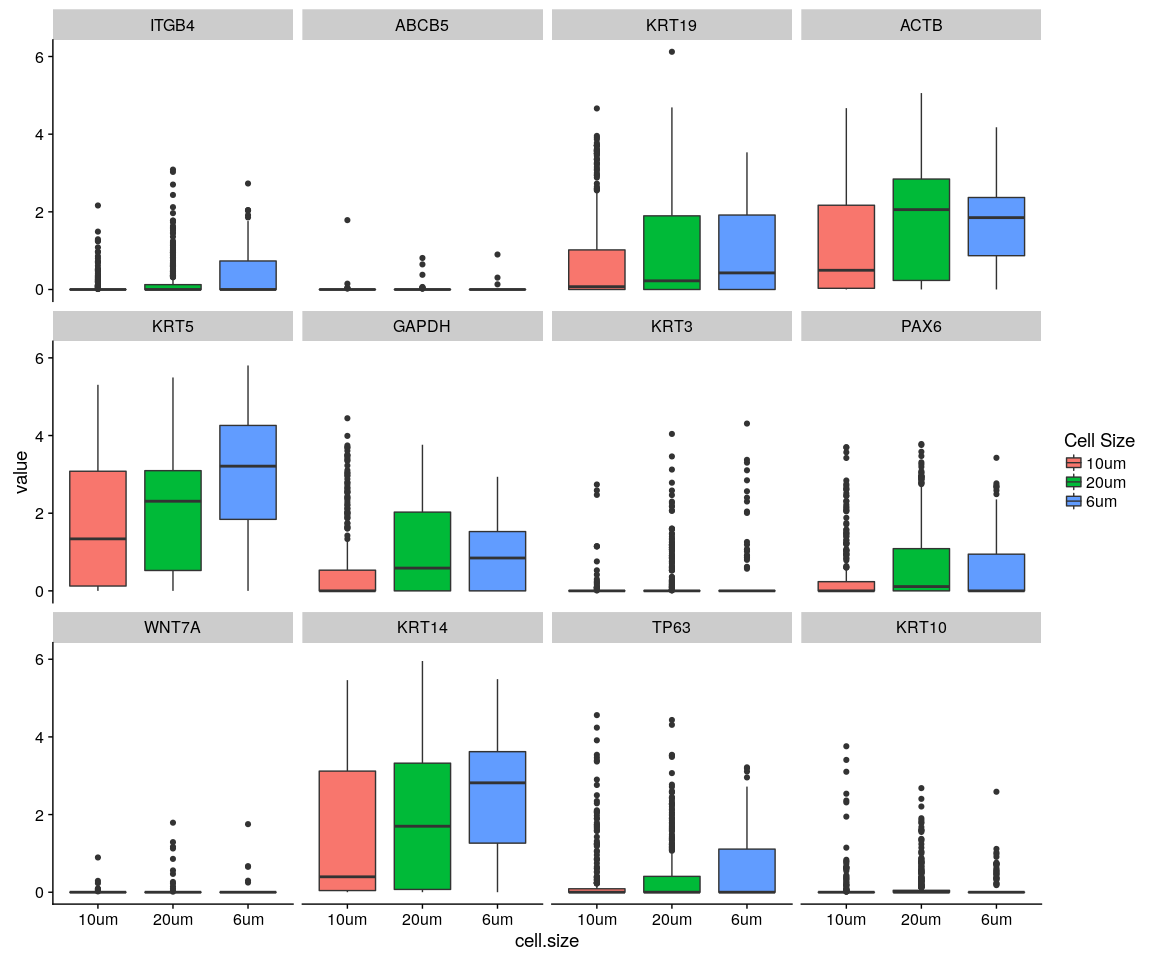
### Figure Explore.1

#### Violin

p <- ggplot(data = human.imp.lognorm.melt, aes(y = value, x = cell.size, fill = cell.size))  
p + geom\_violin(trim = FALSE, scale = "width") + facet\_wrap(~variable) + geom\_jitter() +   
 guides(fill = guide\_legend(title = "Cell Size"))

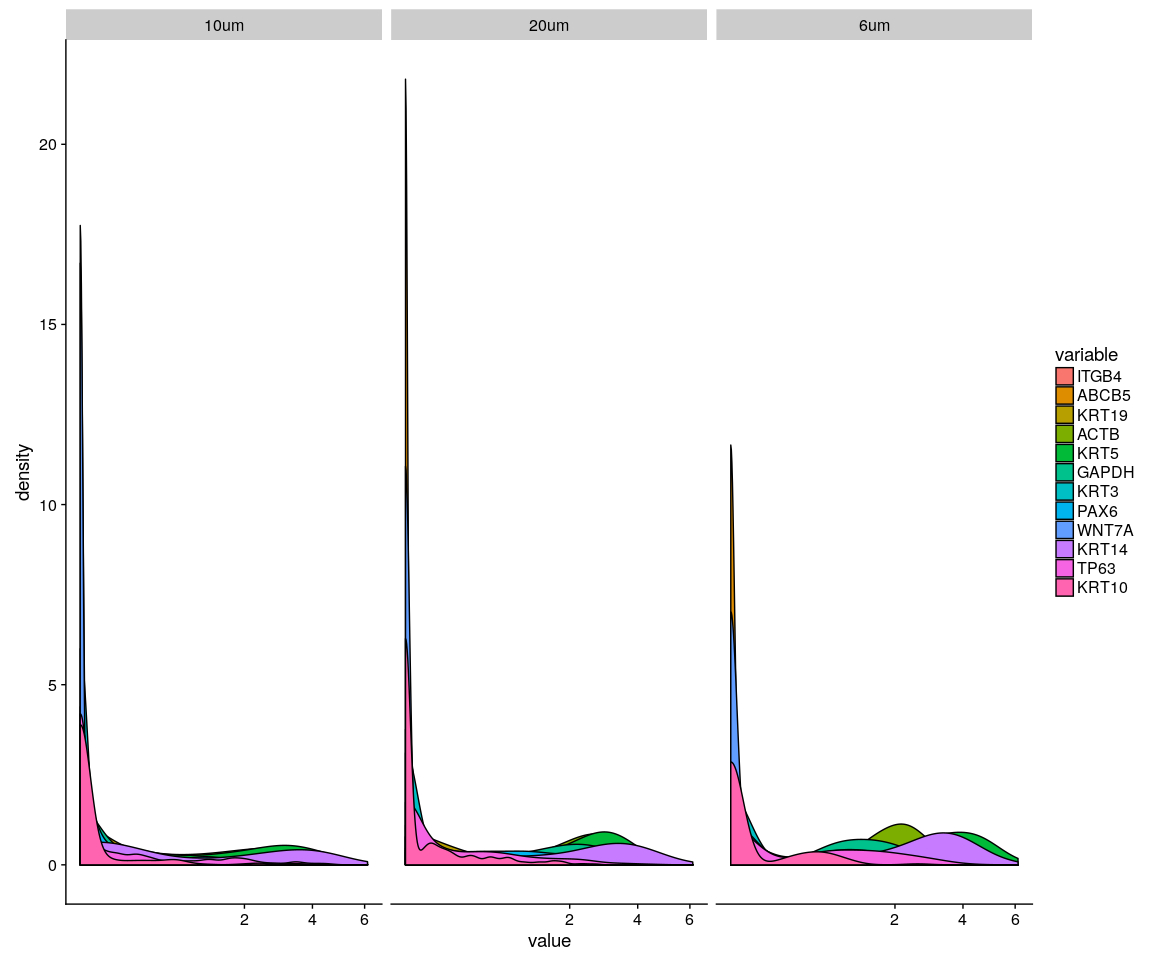
#### Boxplot

p <- ggplot(data = human.imp.lognorm.melt, aes(y = value, x = cell.size, fill = cell.size))  
p + geom\_boxplot() + guides(fill = guide\_legend(title = "Cell Size")) + facet\_wrap(~variable)

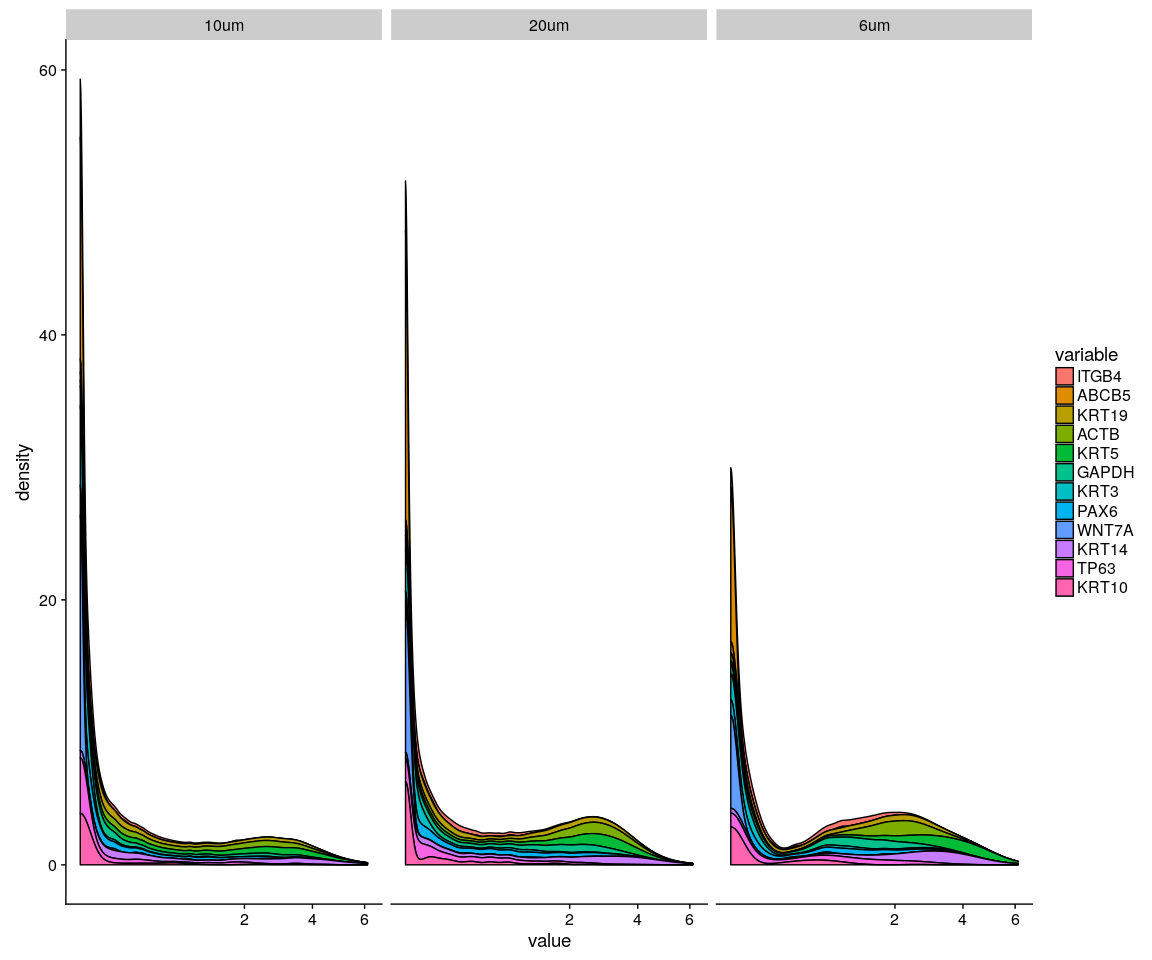


#### Density,histogram

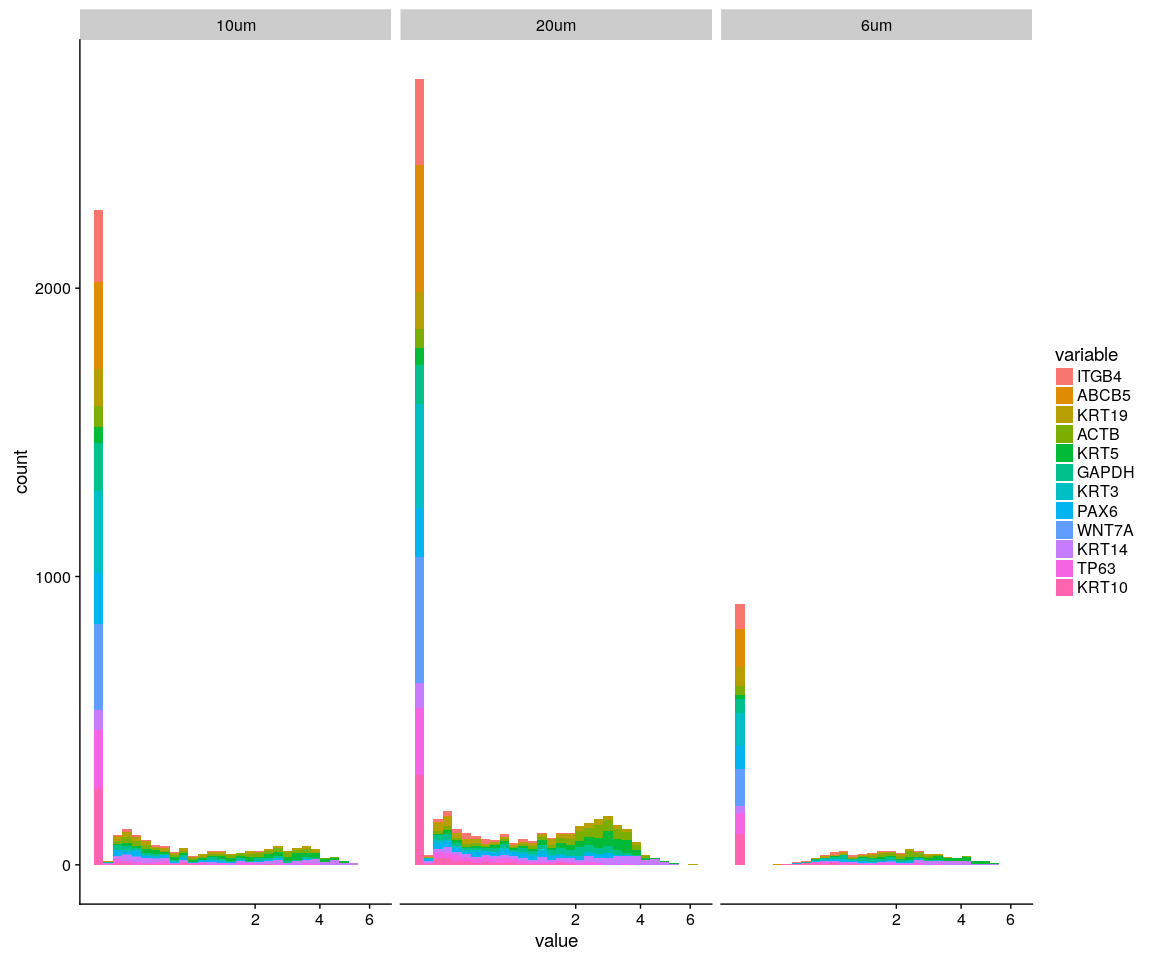
ggplot(data = human.imp.lognorm.melt, aes(x = value, fill = variable)) + geom\_density(kernel = "gaussian") +   
 scale\_x\_sqrt() + facet\_wrap(~cell.size)



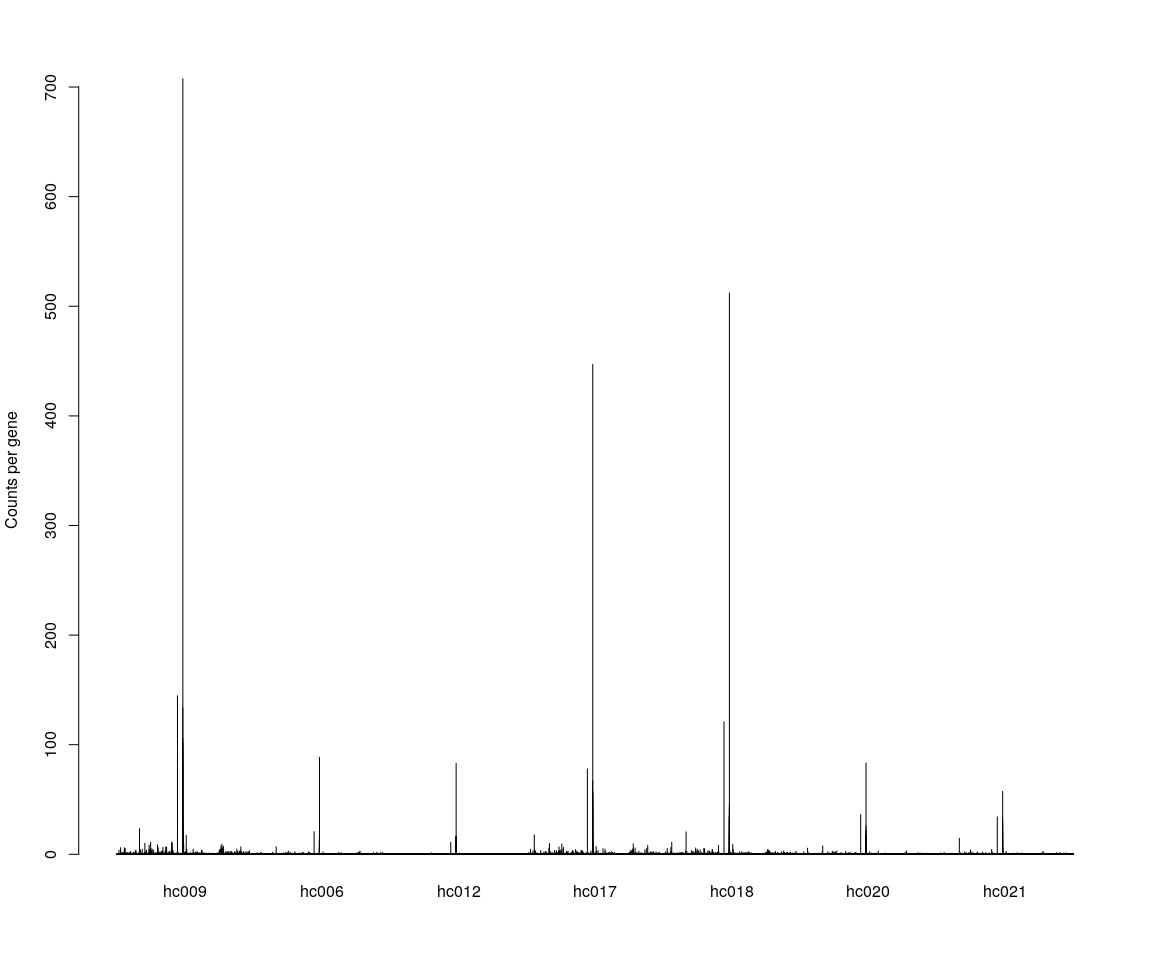
ggplot(data = human.imp.lognorm.melt, aes(x = value, fill = variable)) + geom\_density(kernel = "gaussian",   
 position = "stack") + scale\_x\_sqrt() + facet\_wrap(~cell.size)



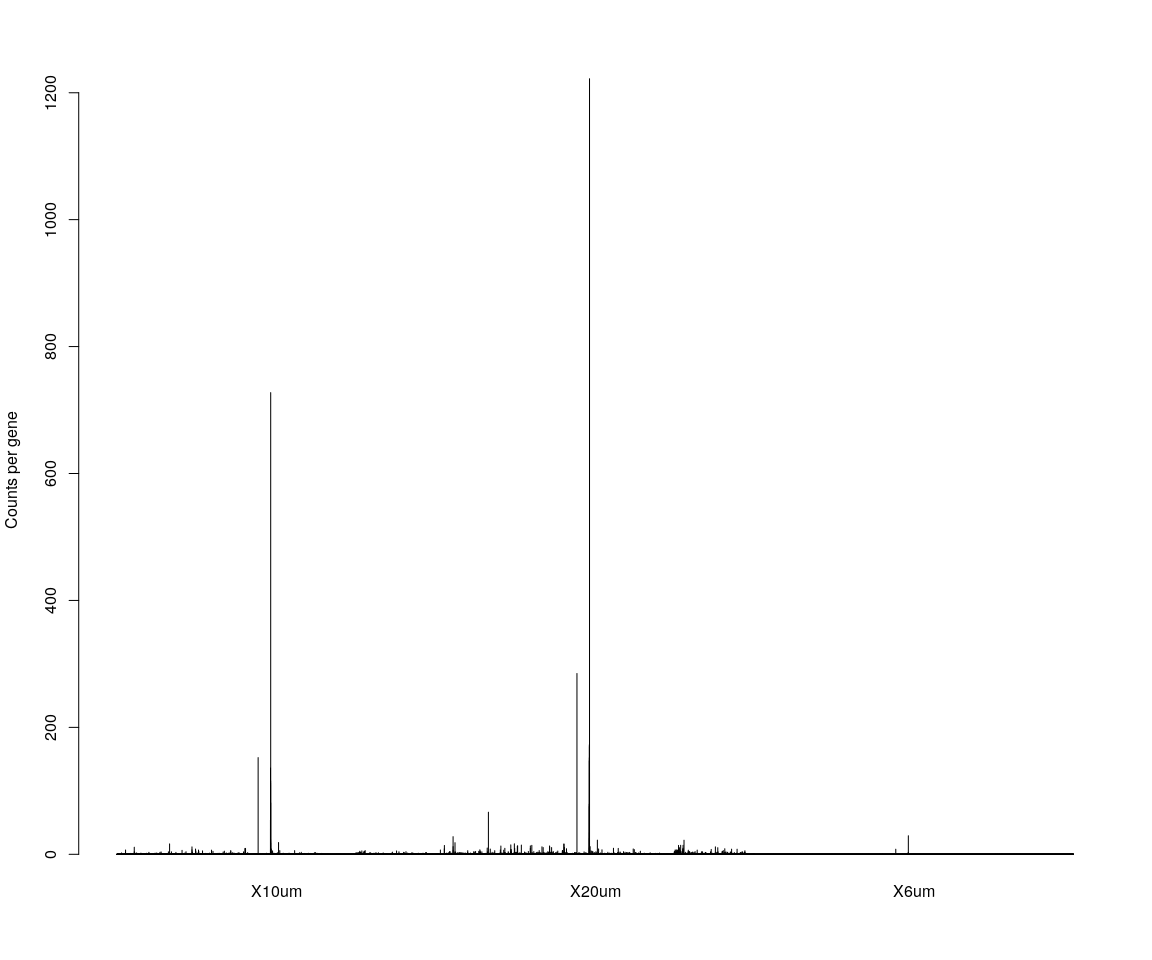
ggplot(data = human.imp.lognorm.melt, aes(x = value, fill = variable)) + geom\_histogram() +   
 scale\_x\_sqrt() + facet\_wrap(~cell.size)



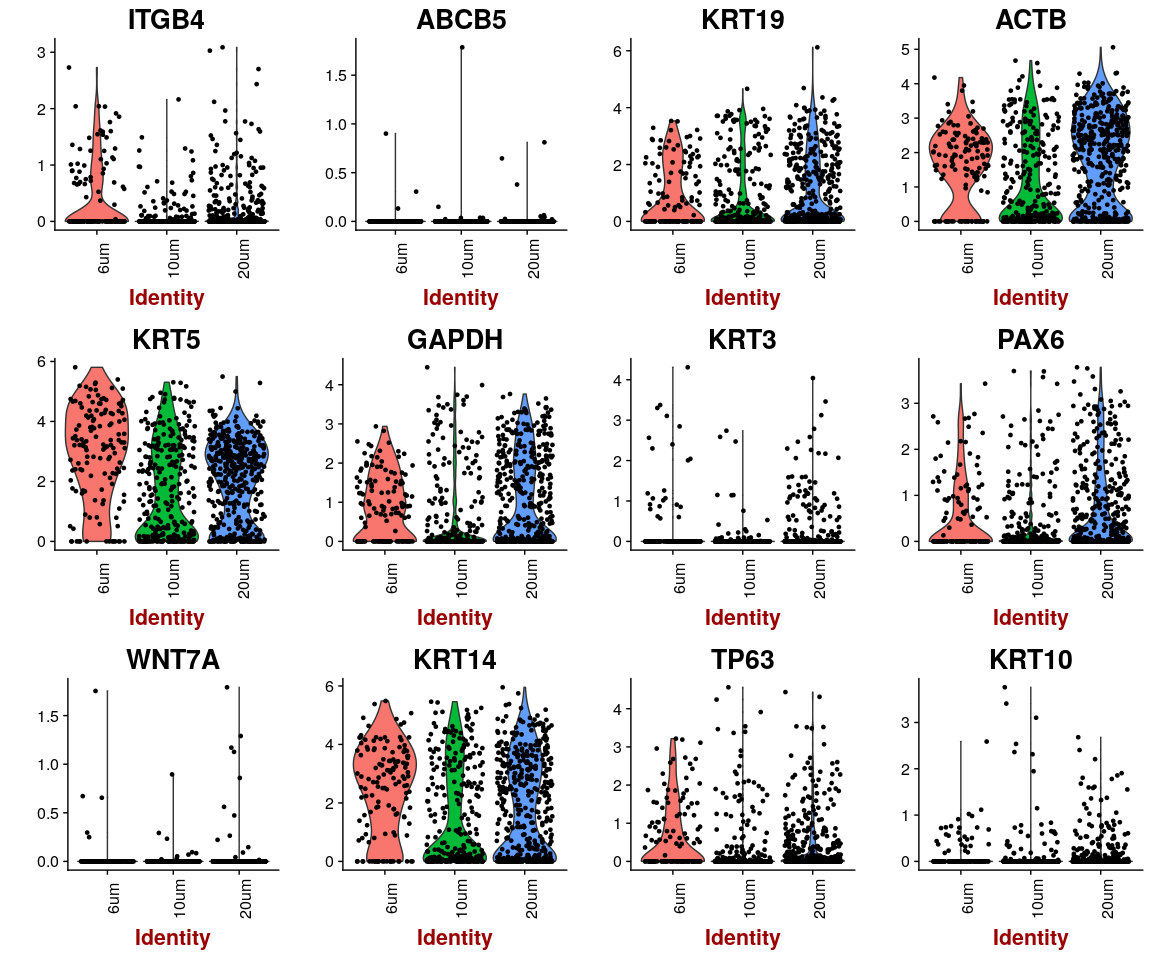
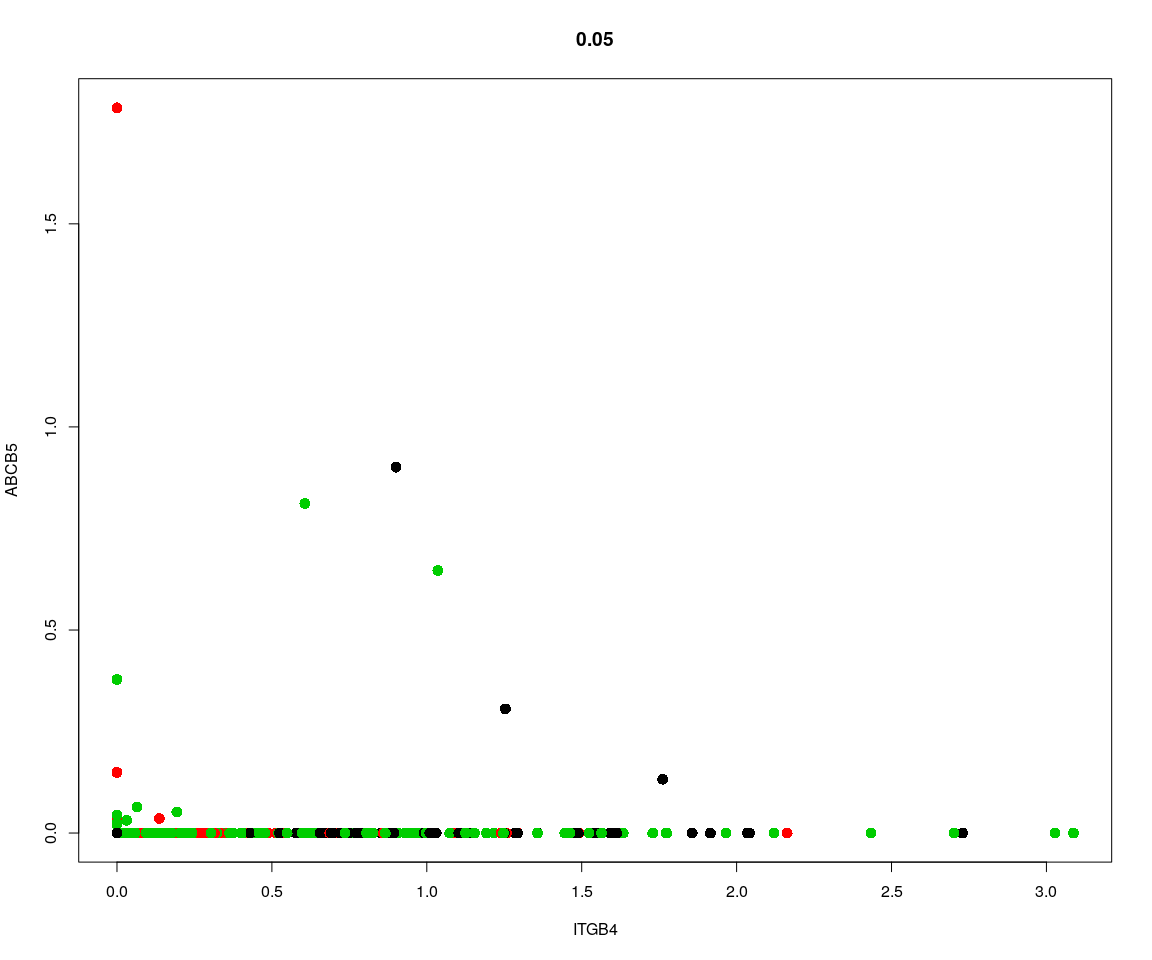
Group\_Bar(human.all.DESeq@raw.data, group = all.sample.group)



Group\_Bar(human.all.DESeq@raw.data, group = all.sample.size)



# We are interested in the gene ITGB4  
GenePlot(human.all.DESeq, gene1 = "ITGB4", gene2 = important.genes[2])  
# VlnPlot(human.all.DESeq,features.plot = 'ITGB4',y.lab.rot = 90) # Violinn  
# plot of gene ITGB in all sample  
VlnPlot(human.all.DESeq, features.plot = important.genes[important.genes %in%   
 rownames(human.all.DESeq@raw.data)], y.lab.rot = 90) # Violinn plot of gene ITGB in all sample



## Dimensionality reduction

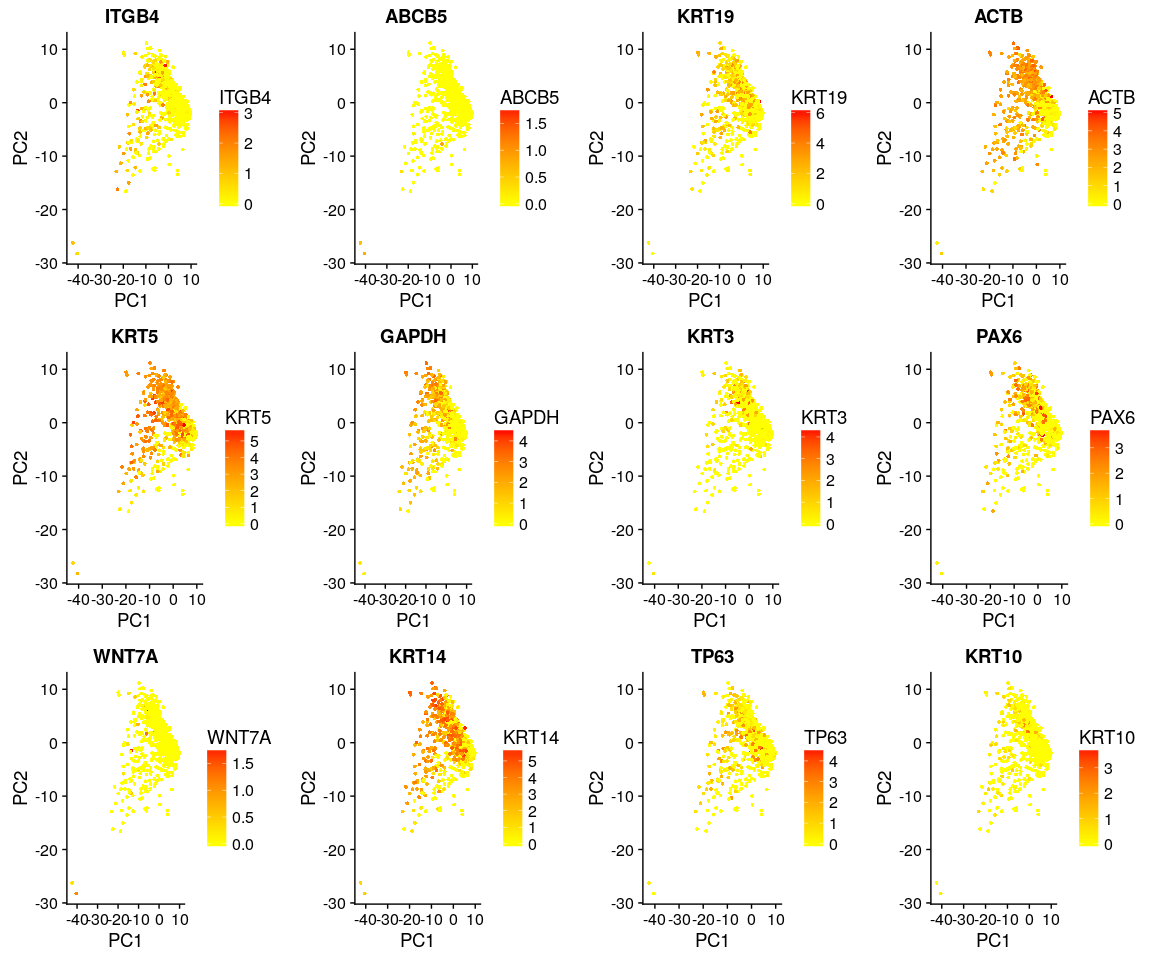
### **PCA** and **tSNE**

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

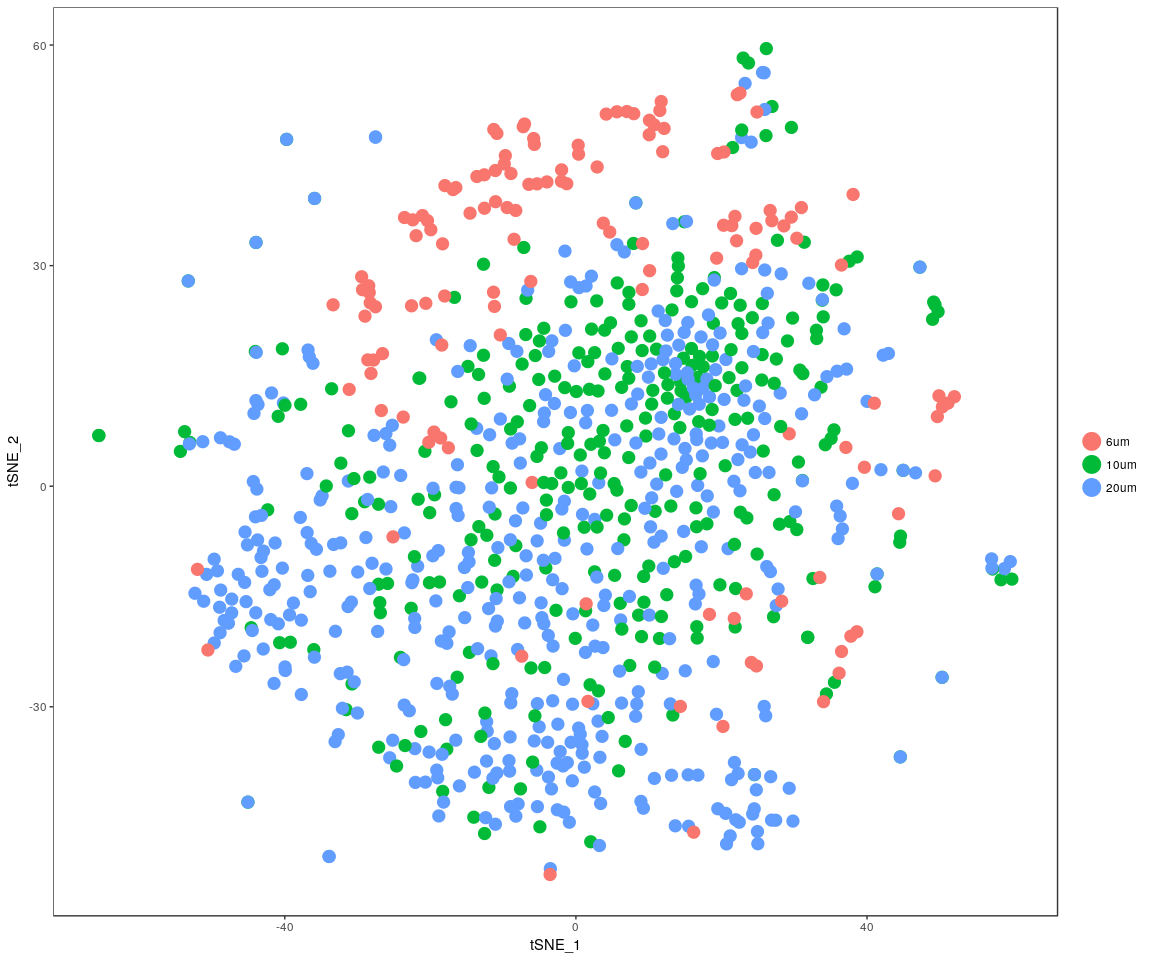
all.pbmc <- PCA.TSNE(object = human.all.DESeq, pcs.compute = FALSE, num.pcs = 28)

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

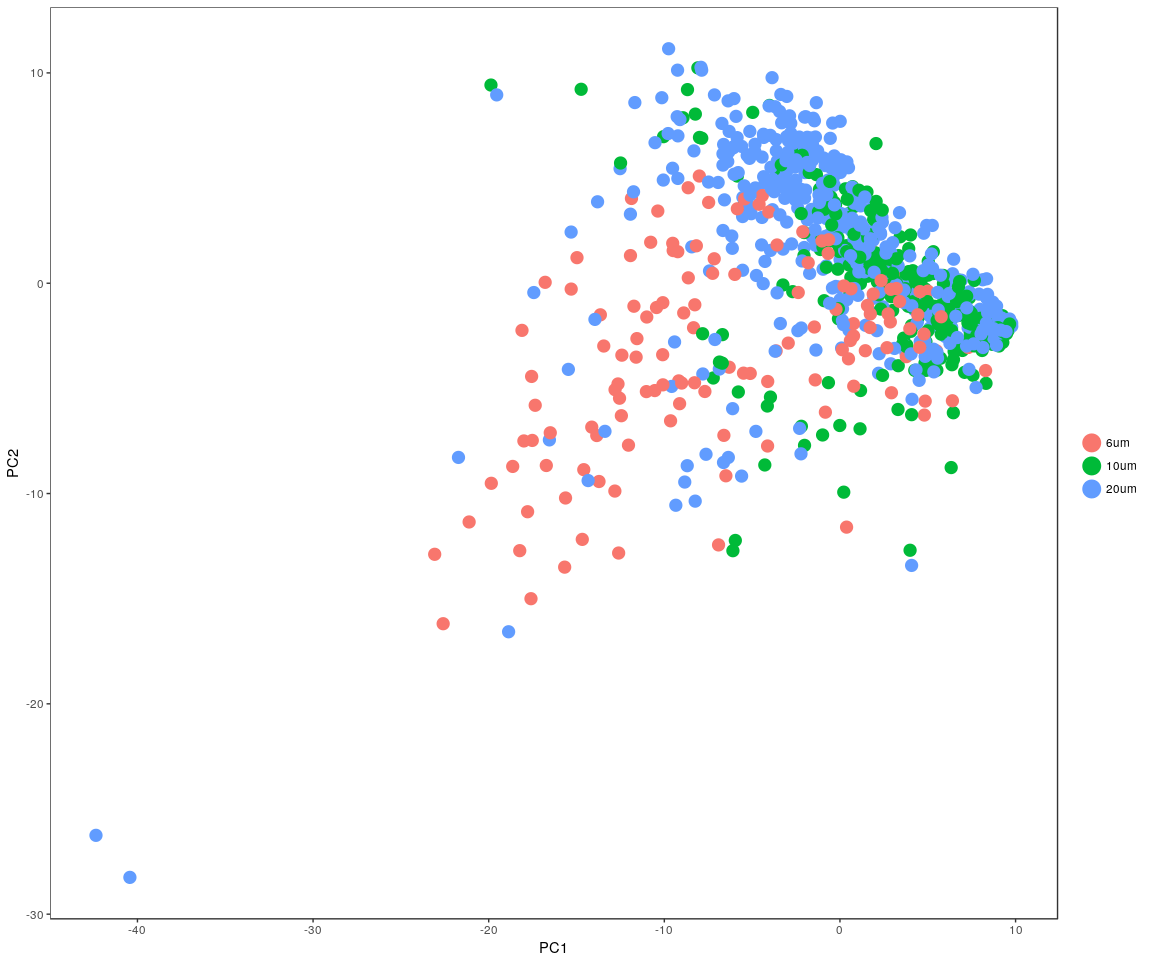
# FeaturePlot(object = all.pbmc,features.plot ='ITGB4',pt.size = 4,no.legend  
# = FALSE) # ITGB4 gene in part dataset  
FeaturePlot(object = all.pbmc, features.plot = important.genes[important.genes %in%   
 rownames(human.all.DESeq@raw.data)], pt.size = 1, no.legend = FALSE, reduction.use = "pca") # ITGB4 gene in part dataset



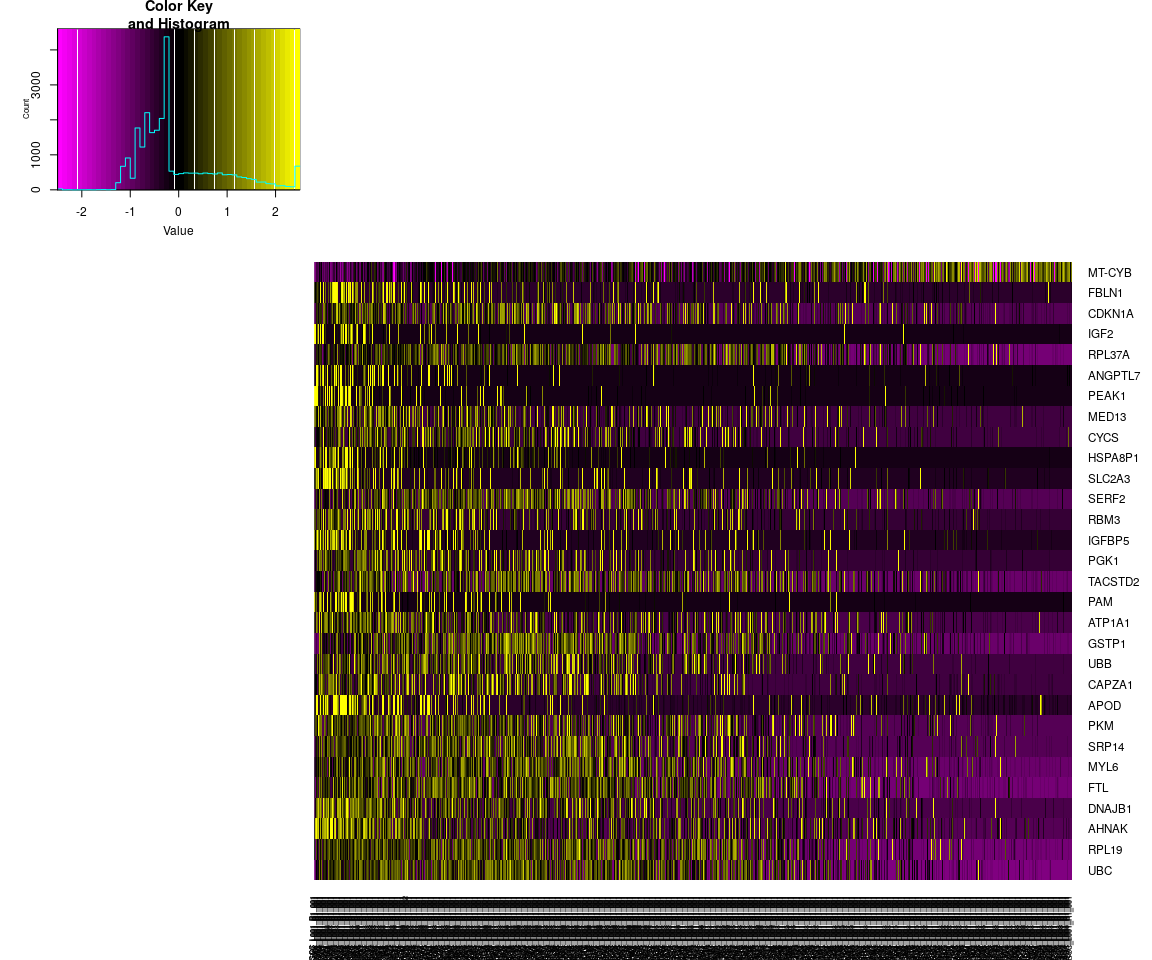
DimPlot(all.pbmc, reduction.use = "tsne", pt.size = 4) # grour by sample



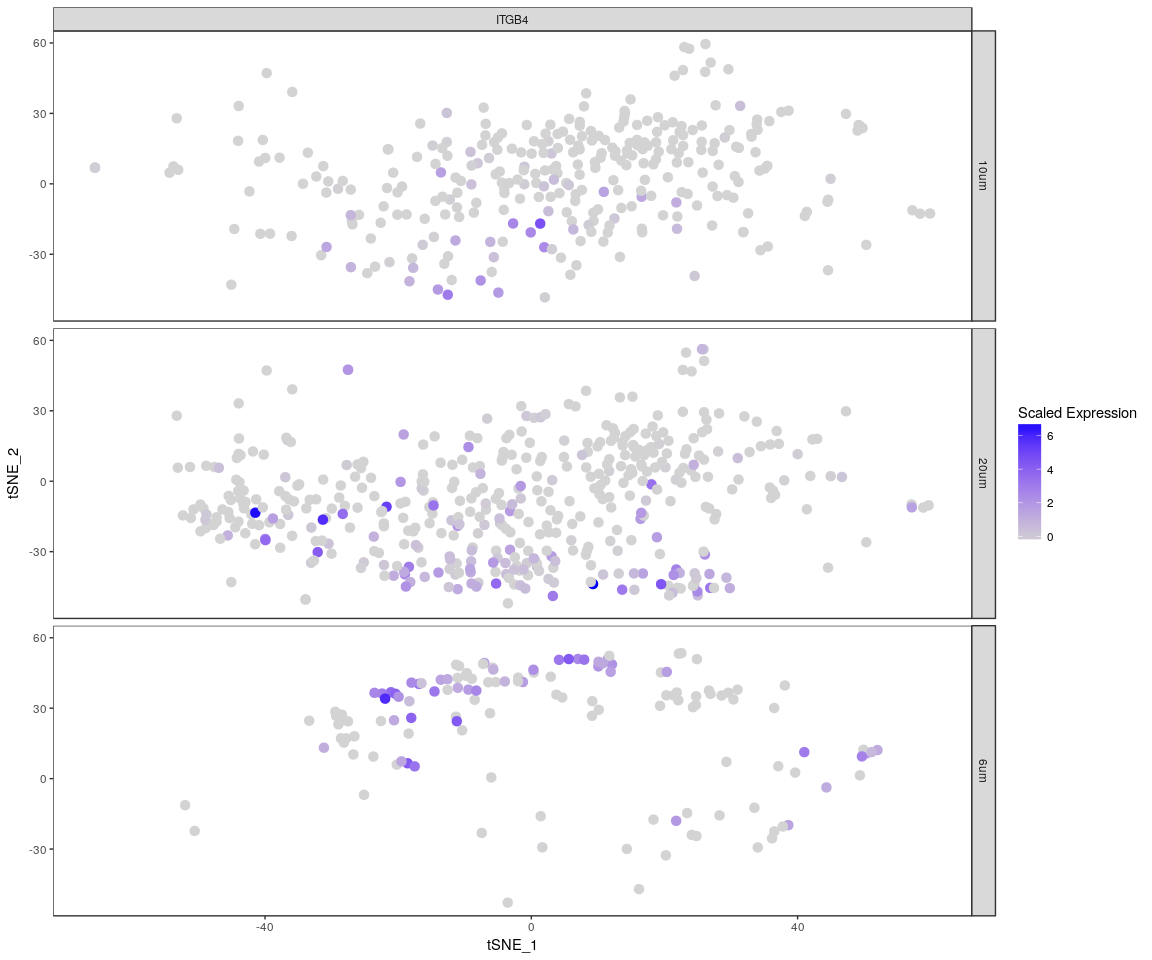
DimPlot(all.pbmc, reduction.use = "pca", pt.size = 4) # grour by sample



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



FeatureHeatmap(all.pbmc, features.plot = "ITGB4", pt.size = 3, plot.horiz = TRUE,   
 cols.use = c("lightgrey", "blue"))

 The Faetureplot of ITGB4, ABCB5, KRT19, ACTB, KRT12, KRT5, GAPDH, KRT3, PAX6, WNT7A, KRT14, TP63, 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. About the heatmap,we only show the gene **ITGB4** And the FeatureHeatmap and Heamap also comfirm this phenomeno.We try the other four variable genes,which has the similar result as gene *ITGB4* 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

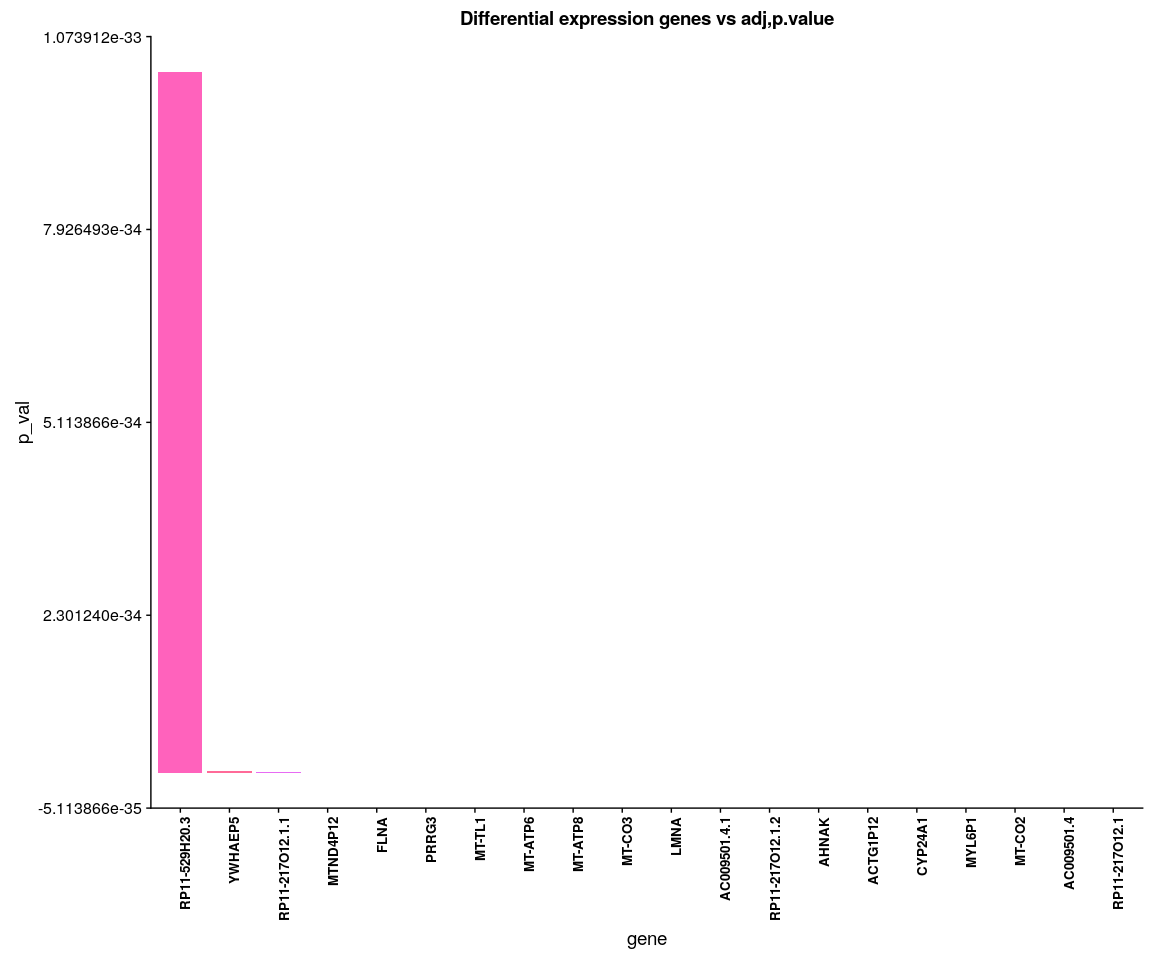
Next,we will have analysis on gene differential expression.Find maker genes across sample.We use the method: \*\*wilcox test\*\*

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

## p\_val avg\_logFC pct.1 pct.2 p\_val\_adj cluster  
## RP11-217O12.1 1.090404e-152 2.9265260 0.993 0.967 2.058682e-148 6um  
## AC009501.4 2.570580e-105 3.1408602 0.822 0.157 4.853255e-101 6um  
## MT-CO2 1.507064e-70 -2.2077454 0.852 0.979 2.845337e-66 6um  
## MYL6P1 2.564892e-67 0.2545159 0.044 0.400 4.842517e-63 6um  
## CYP24A1 1.094166e-66 2.3676155 0.711 0.096 2.065785e-62 6um  
## ACTG1P12 1.202945e-59 0.5510963 0.030 0.149 2.271161e-55 6um  
## gene  
## RP11-217O12.1 RP11-217O12.1  
## AC009501.4 AC009501.4  
## MT-CO2 MT-CO2  
## MYL6P1 MYL6P1  
## CYP24A1 CYP24A1  
## ACTG1P12 ACTG1P12

We check whether the important genes are still in the marker genes we found from the DESeq analysis. the genes:ITGB4, KRT19, ACTB, KRT5, GAPDH, KRT3, PAX6, KRT14 are still in the marker genes.

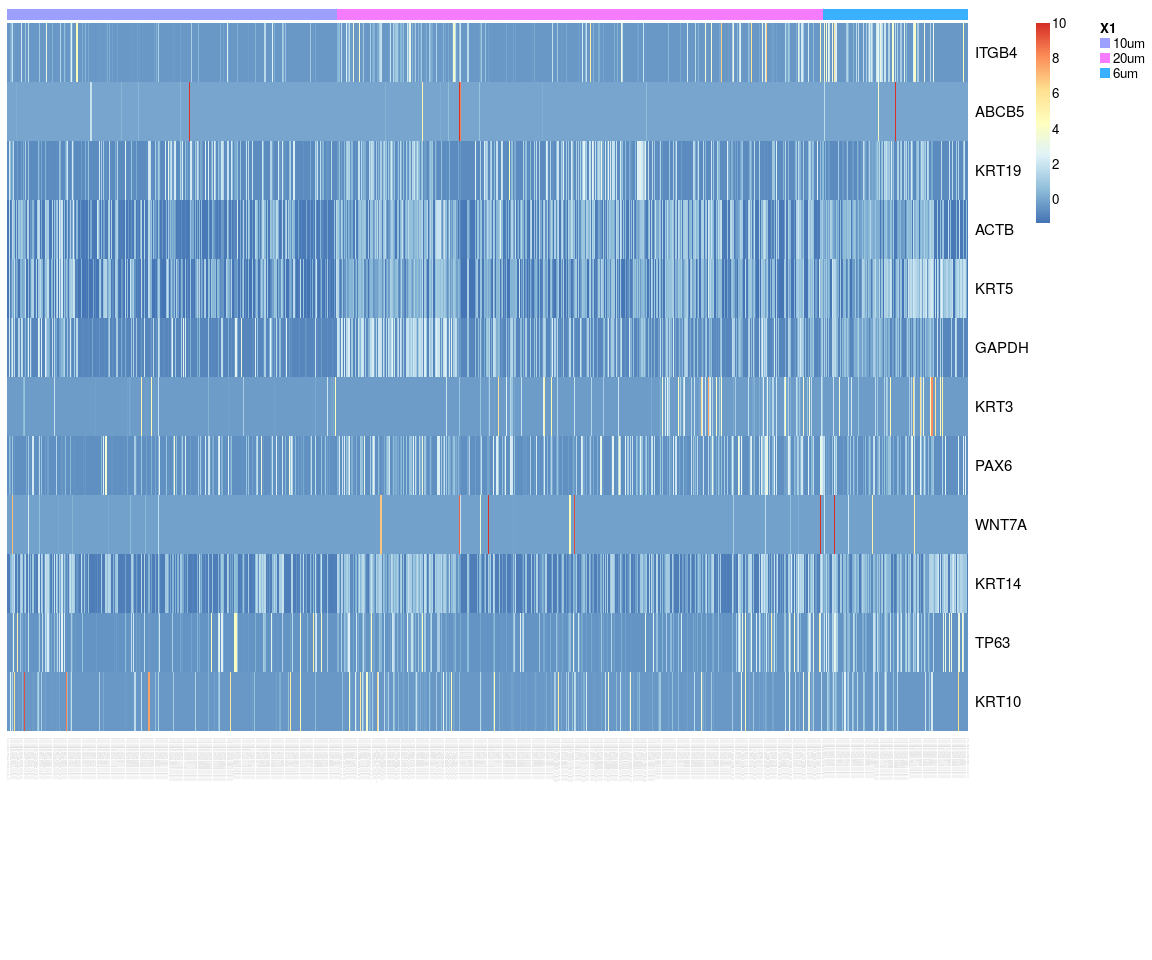
### Bar plot of gene’s p.val



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

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

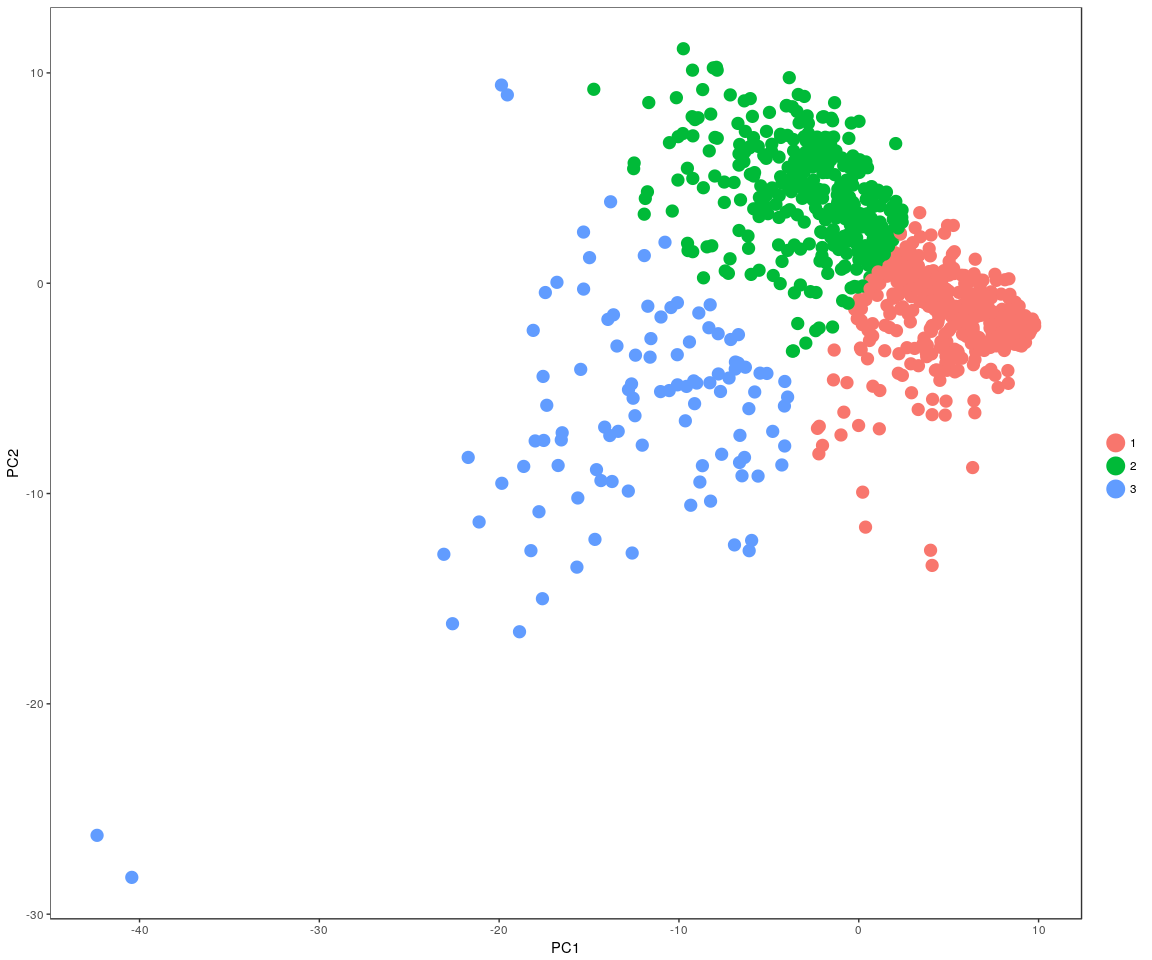
NMF::aheatmap(human.heatmap[[2]], Rowv = NA, Colv = NA, annCol = human.heatmap[[1]],   
 scale = "none")



We have find all marker genes across sample,there are 1938 significant genes(adjust p-value <0.05) in all marker genes.

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

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



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

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

