Deep analysis on human

yejg

2018/1/31

### 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)

## Step 1: All data: Analysis based on sample group

### 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")]]  
  
table(unlist(lapply(colnames(human.only.pro), function(x) return(str\_split(x,   
 "\_")[[1]][2]))))

##   
## 10um 20um 6um   
## 326 560 159

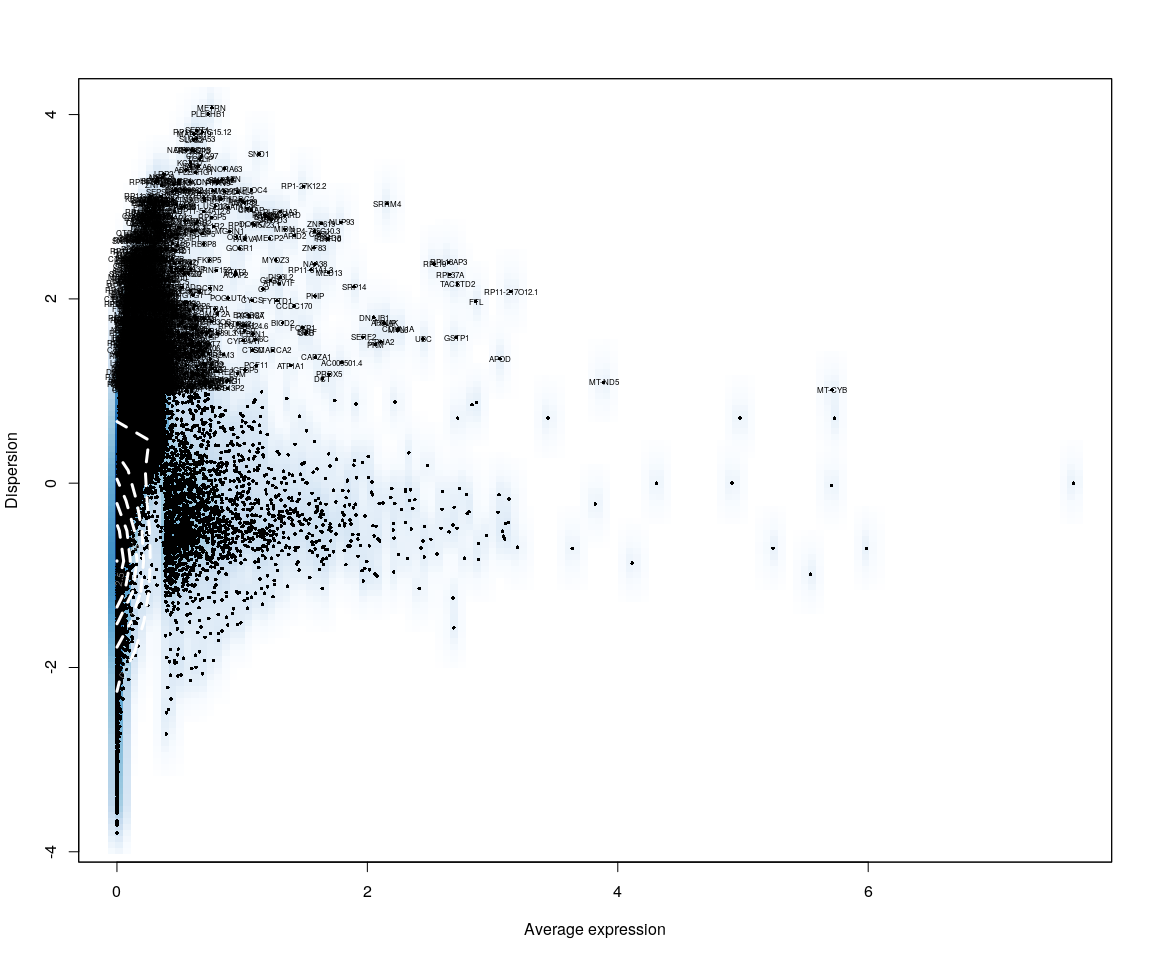
table(unlist(lapply(colnames(human.only.pro), function(x) return(str\_split(x,   
 "\_")[[1]][1]))))

##   
## hc006 hc009 hc012 hc017 hc018 hc020 hc021   
## 92 187 66 170 188 184 158

### Create Seurat object and not caculate DESeq,but set **min.cells=10** and **min.genes=2**

# only select the cells contain 10 genes expressed at least,select the genes  
# must be expressed in two cells at least  
human.all.pbmc <- 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.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][1])))  
all.sample.size <- unlist(lapply(human.all.pbmc@cell.names, function(x) return(str\_split(x,   
 "\_")[[1]][2])))  
# reset ident  
human.all.pbmc <- SetIdent(human.all.pbmc, cells.use = human.all.pbmc@cell.names,   
 ident.use = all.sample.size)  
table(all.sample.group, all.sample.size)

## all.sample.size  
## all.sample.group 10um 20um 6um  
## hc006 0 21 0  
## hc009 65 91 0  
## hc012 21 16 13  
## hc017 64 72 34  
## hc018 60 95 33  
## hc020 58 70 24  
## hc021 40 87 31

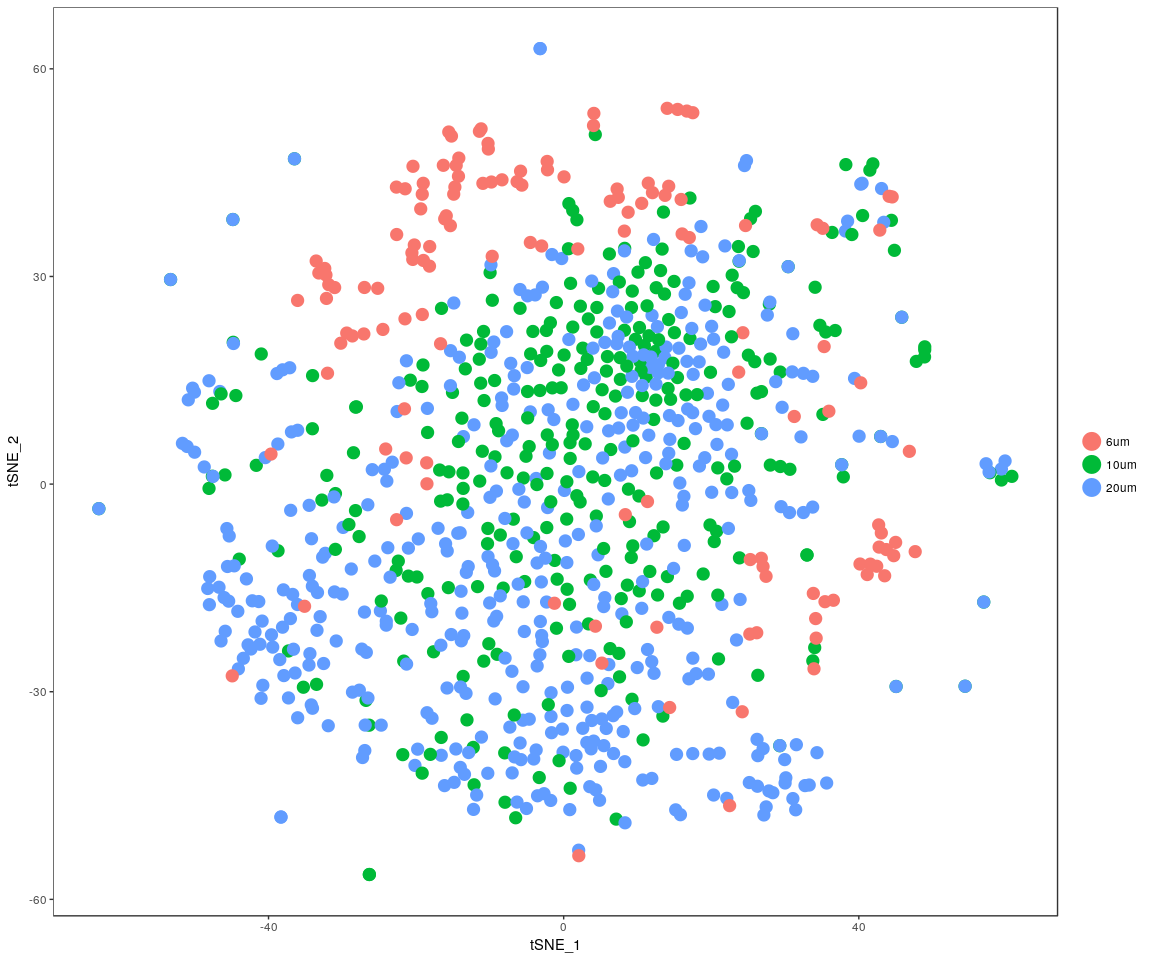
## Dimensionality reduction

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

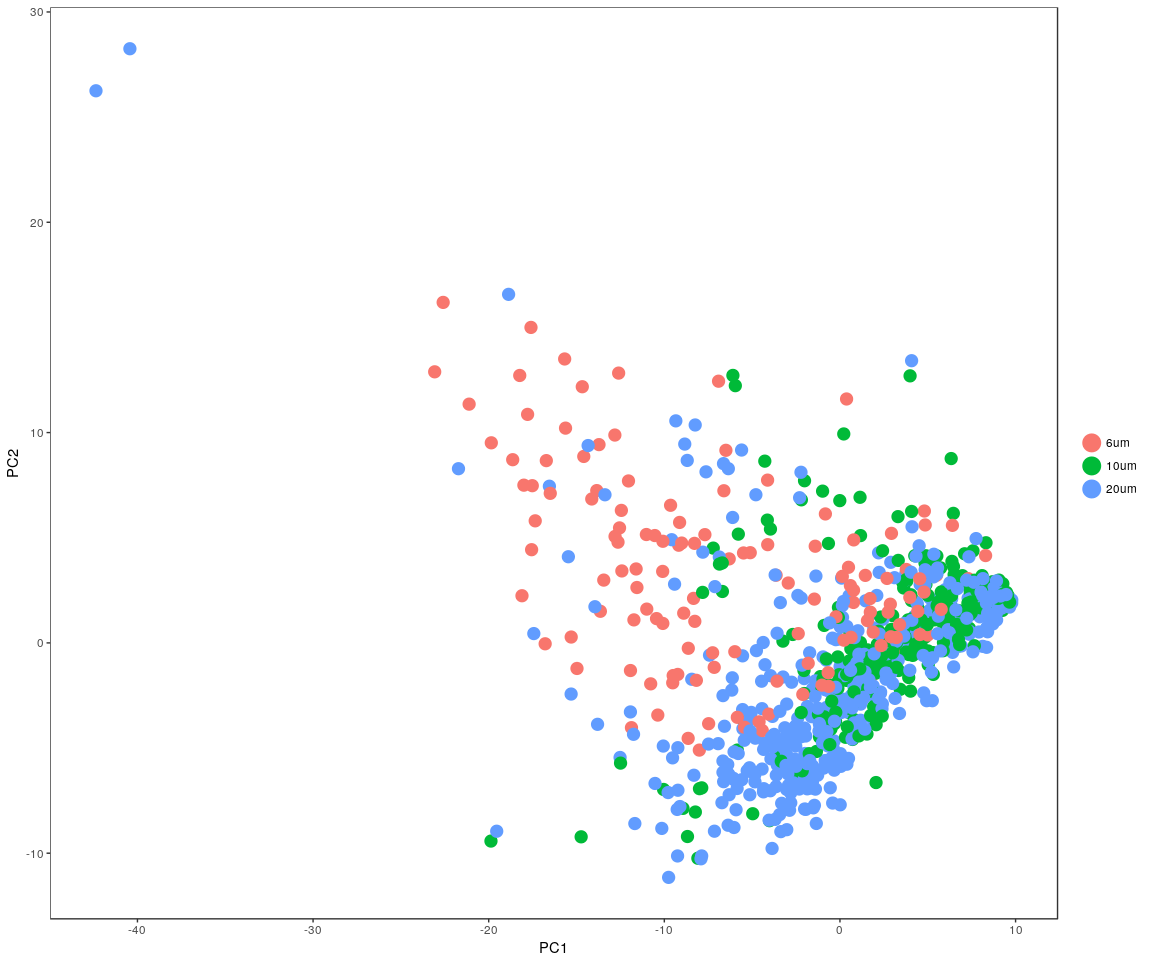
Here,do the dimensionality reduction using the PCA, tSNE method

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

DimPlot(all.pbmc, reduction.use = "tsne", pt.size = 4)



DimPlot(all.pbmc, reduction.use = "pca", pt.size = 4)

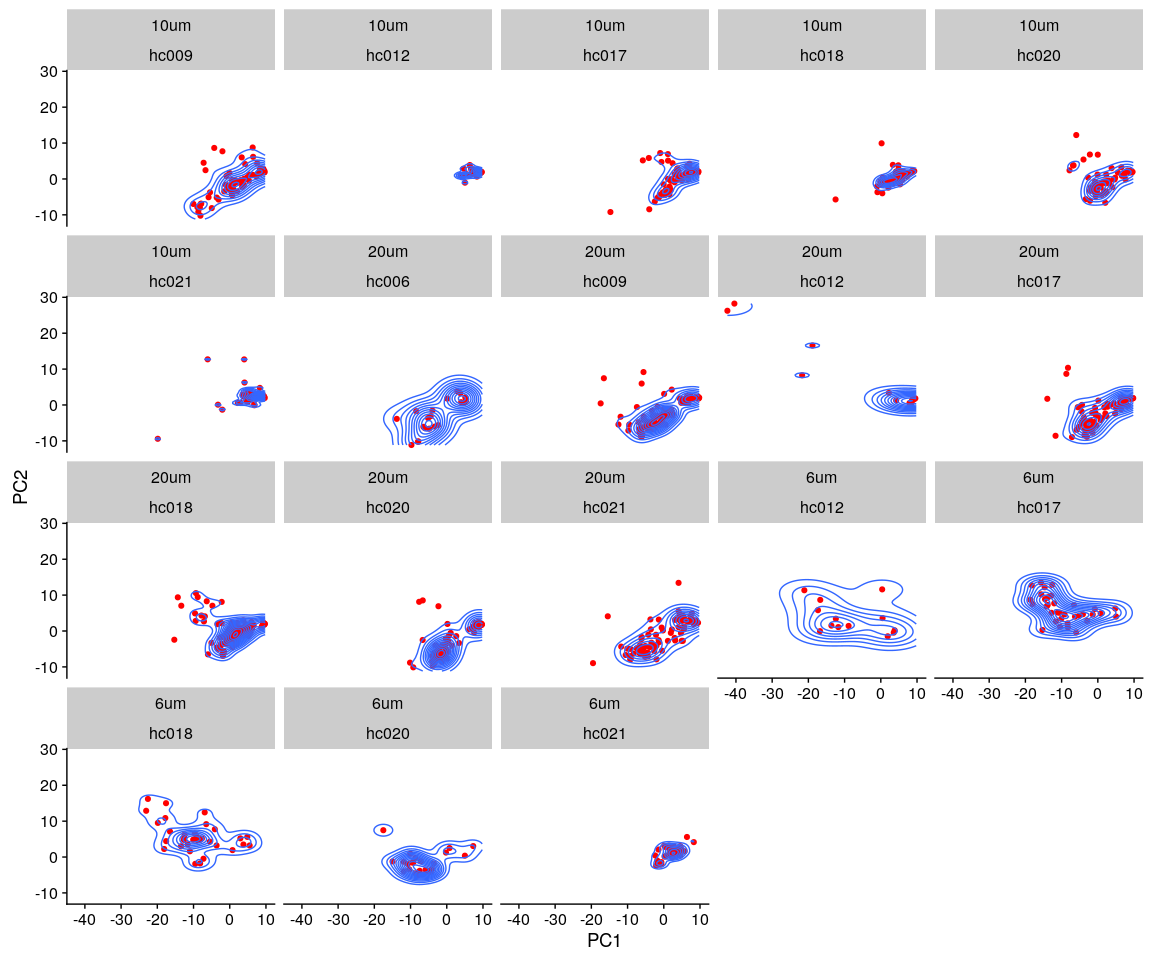


### In the sample cell size,compare the sample distribution

pca <- data.frame(all.pbmc@dr$pca@cell.embeddings)  
tsne <- data.frame(all.pbmc@dr$tsne@cell.embeddings)  
pca$cell.size <- all.sample.size  
pca$cell.group <- all.sample.group  
tsne$cell.size <- all.sample.size  
tsne$cell.group <- all.sample.group

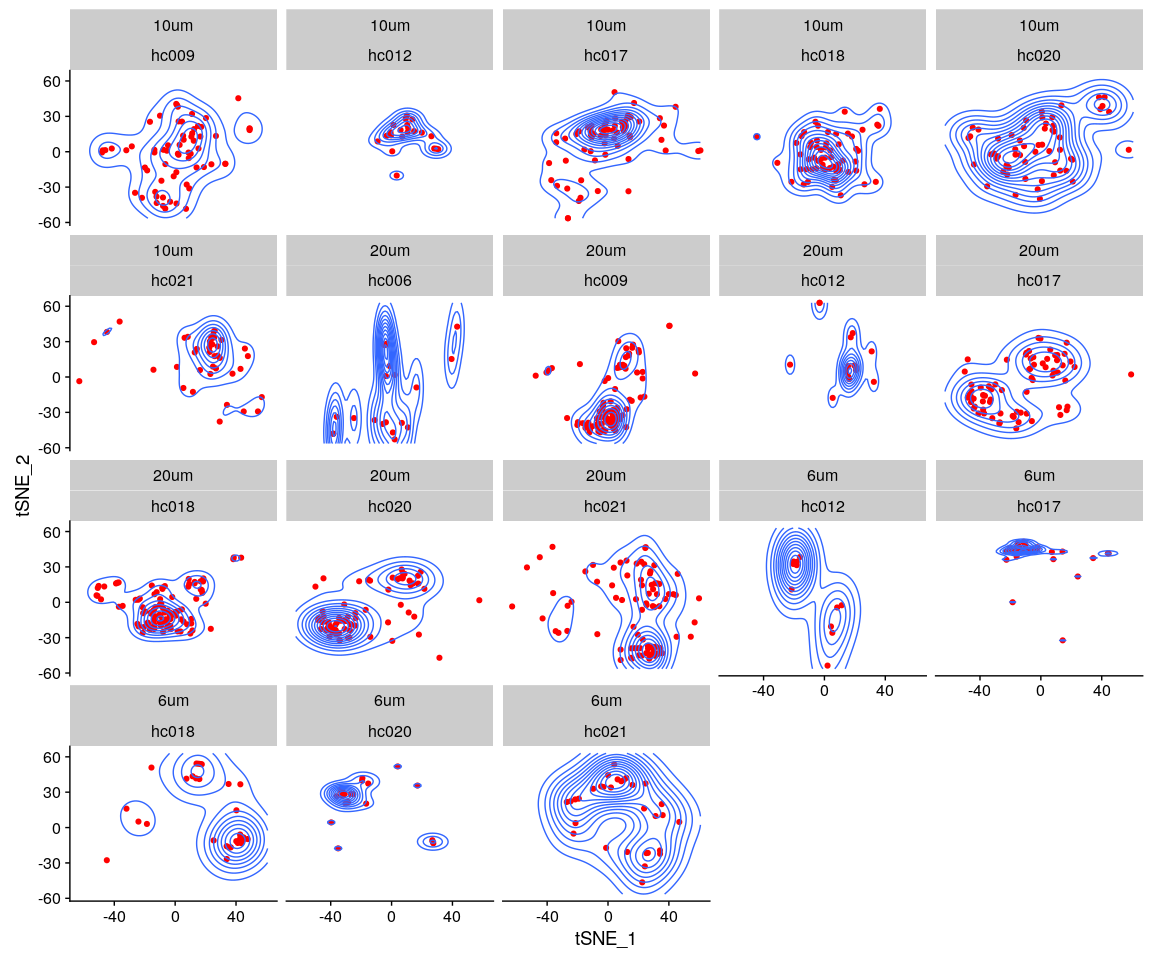
#### PCA method

ggplot(data = pca) + geom\_point(aes(x = PC1, y = PC2), color = "red") + geom\_density2d(aes(x = PC1,   
 y = PC2), contour = TRUE) + facet\_wrap(~cell.size + cell.group)



#### tSNE method

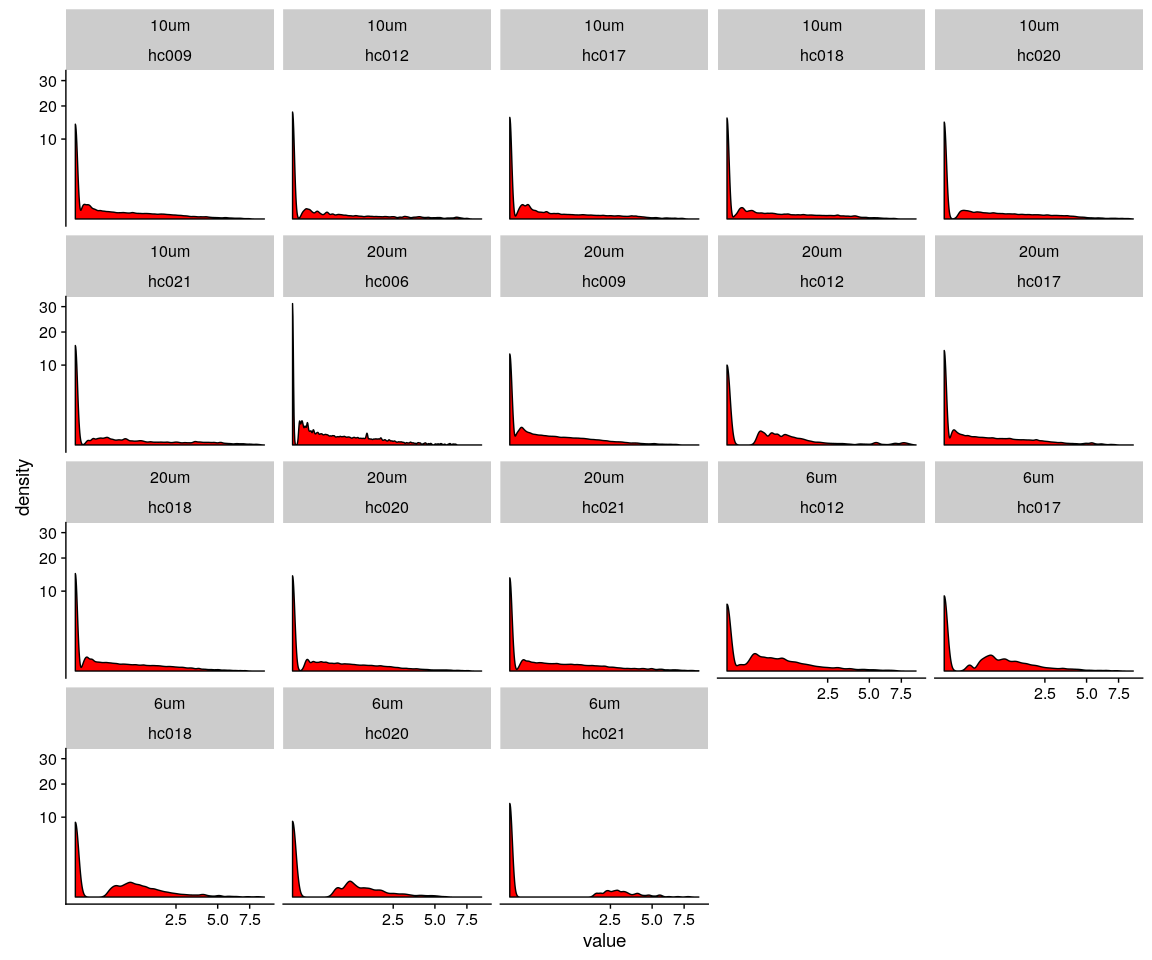
ggplot(data = tsne) + geom\_point(aes(x = tSNE\_1, y = tSNE\_2), color = "red") +   
 geom\_density2d(aes(x = tSNE\_1, y = tSNE\_2), contour = TRUE) + facet\_wrap(~cell.size +   
 cell.group)



#### density method

human.lognorm <- data.frame(FetchData(all.pbmc, vars.all = all.pbmc@var.genes))  
human.lognorm$cell.size <- all.sample.size  
human.lognorm$cell.group <- all.sample.group  
human.lognorm.melt <- melt(human.lognorm)

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



#### the Kolmogorov–Smirnov test (K–S test or KS test) is a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test)

##### The null distribution of this statistic is calculated under the null hypothesis that the sample is drawn from the reference distribution (in the one-sample case) or that the samples are drawn from the same distribution (in the two-sample case). In each case, the distributions considered under the null hypothesis are continuous distributions but are otherwise unrestricted

size.group.table <- as.data.frame(as.matrix(table(all.sample.group, all.sample.size)))  
size.group.table <- size.group.table[size.group.table$Freq > 0, ] # only test cell size,group both not null

### 6um:under the cell size 6um test

KS.test.comb(all.pbmc, genes = important.genes, cell\_size = "6um")

## 6um\_p.value  
## hc012\_hc017 1.881488e-03  
## hc012\_hc018 2.297485e-01  
## hc012\_hc020 1.253244e-01  
## hc012\_hc021 7.279066e-12  
## hc017\_hc018 8.406973e-03  
## hc017\_hc020 6.587685e-03  
## hc017\_hc021 4.131246e-08  
## hc018\_hc020 3.483077e-01  
## hc018\_hc021 6.453171e-12  
## hc020\_hc021 7.876061e-08

### 10um:under the cell size 10um test

KS.test.comb(all.pbmc, genes = important.genes, cell\_size = "10um")

## 10um\_p.value  
## hc009\_hc012 2.664535e-15  
## hc009\_hc017 2.166676e-04  
## hc009\_hc018 5.494429e-06  
## hc009\_hc020 7.269474e-11  
## hc009\_hc021 6.328271e-15  
## hc012\_hc017 6.888093e-08  
## hc012\_hc018 4.651964e-05  
## hc012\_hc020 2.147998e-04  
## hc012\_hc021 7.149712e-02  
## hc017\_hc018 1.070482e-01  
## hc017\_hc020 3.263727e-03  
## hc017\_hc021 3.374847e-06  
## hc018\_hc020 3.391620e-01  
## hc018\_hc021 1.783187e-03  
## hc020\_hc021 5.018920e-03

### 20um:under the cell size 20um test

KS.test.comb(all.pbmc, genes = important.genes, cell\_size = "20um")

## 20um\_p.value  
## hc006\_hc009 5.957277e-01  
## hc006\_hc012 0.000000e+00  
## hc006\_hc017 9.062330e-06  
## hc006\_hc018 2.958155e-03  
## hc006\_hc020 1.354329e-07  
## hc006\_hc021 3.110368e-04  
## hc009\_hc012 0.000000e+00  
## hc009\_hc017 2.220446e-15  
## hc009\_hc018 1.143267e-08  
## hc009\_hc020 0.000000e+00  
## hc009\_hc021 4.984624e-11  
## hc012\_hc017 6.781020e-12  
## hc012\_hc018 1.354472e-14  
## hc012\_hc020 6.990112e-08  
## hc012\_hc021 1.396661e-12  
## hc017\_hc018 1.408464e-02  
## hc017\_hc020 1.442314e-01  
## hc017\_hc021 3.619428e-04  
## hc018\_hc020 3.673977e-03  
## hc018\_hc021 1.554877e-01  
## hc020\_hc021 7.500422e-02