CellLine_Br

Vagmita Pabuwal

5/22/2021

R Markdown

This R script is to analyze general information about cell line data without any experimental effect. This can be used to understand differential count and clustering of cell lines and genes if any experimental condition would have been applied on the cell lines This Analysis primarily was done to understand the effect of different therapeutic agent on different Breast cancer cell lines.There are 30 cell lines with 6 different condition, no replicates. I have taken only the TPM values of expression without any therapeutic agent given. Aim was to see based on TPM data how differnt are the cell lines expression. With the aim top 100 genes were plotted as a heatmap and differential count was calculated for one pair. We can calculate pairwise in similar way for all other cell lines. Just to show few statistical analysis performance, I performed PCA and cluster analysis to see if the cell lines could be differentiates, One cell line definitely stood out which was HCC1806 (squamous cell breast carcinoma, acantholytic variant) Overall we can further analyze data based on the data to see effect of drug on the cell lines and comparing the variant data to check which genes are most affected.

#Setting the directory and loading required libraries libraries can also be loaded or installed as per requirement while doing our analysis

```
setwd("/Users/vagmi/Documents/")
library(edgeR)
library(RColorBrewer)
library(scatterplot3d)
library(dplyr)
library(DESeq2)
library(ggplot2)
library(plyr)
library(gplots)
library(pheatmap)
library(stats)
library(ggplot2)
library(ggfortify)
library(factoextra)
```

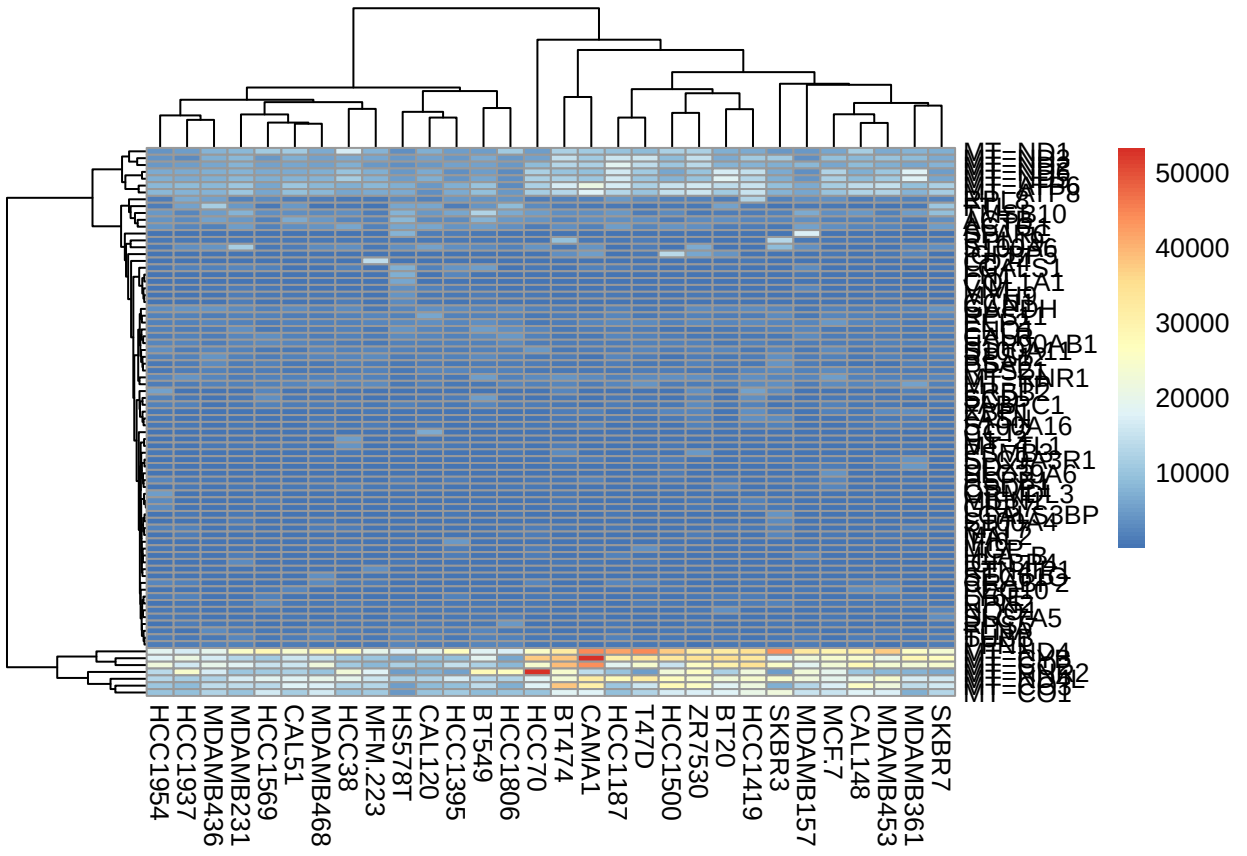
Reading the data and cleaning a bit There were 2 duplicate genes which were removed Wanted to see if there were lot of mitochondrial gene, but they were not as may so kept all in the data.

```
Data_Br <- read.csv("/Users/vagmi/Documents/EMTAB_BreastCL.csv", header=TRUE, stringsAsFactors = TRUE,
Data_Br_unique <- Data_Br[!duplicated(Data_Br$Gene.Name), ]

rownames(Data_Br_unique) <- Data_Br_unique[,1]
Data_Br_unique <- Data_Br_unique[,-1]
count_Br_unique <- Data_Br_unique[rowSums(Data_Br_unique >20) >=1,]
```

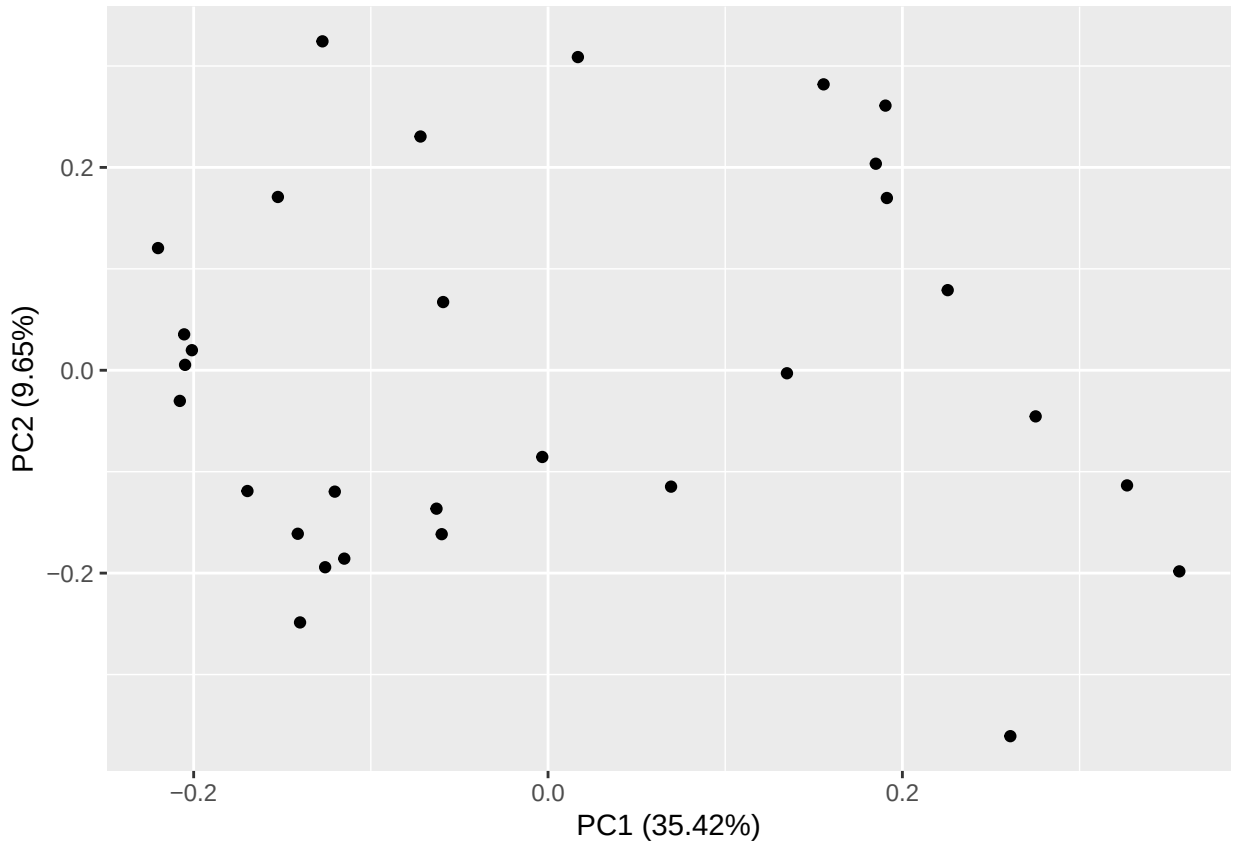
```
mito_gene <- count_Br_unique[grep("^MT-", rownames(count_Br_unique)),]
Exp_design <- read.csv("Exp_design.csv", header=T)
Exp_design <- Exp_design[,-1]
colData <- read.csv("Exp_design.csv", header=T, stringsAsFactors = TRUE)
colData <- colData[,-1]
designFormula <- "~group"
```

plot Heatmap and PCA based on variance #Transpose the matrix for PCA ##Transforming to log2 scale #COmputing PCA plot PCA using ggplot2

```
V <- apply(round(Data_Br_unique), 1, var)
selectedGenes <- names(V[order(V, decreasing = T)][1:80])

pheatmap(as.matrix(Data_Br_unique[selectedGenes,]))
```

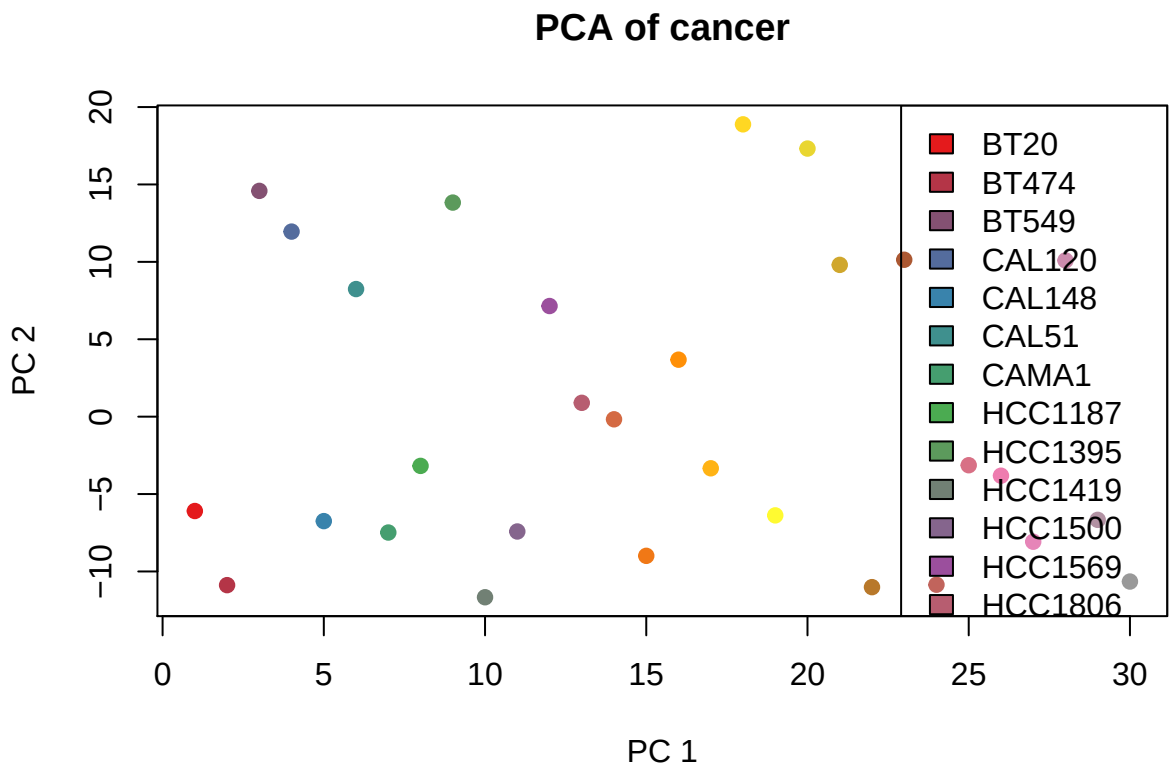



```
Matrix <- t(Data_Br_unique[selectedGenes,])

Matrix <- log2(Matrix+1)

pcaResults <- prcomp(Matrix)

autoplot(pcaResults, data = colData)
```



Another way of plotting PCA

```
colNames1 = colnames(Data_Br_unique)
colorInterpolation = colorRampPalette(brewer.pal(9,'Set1'))
col1 = colorInterpolation(length(sort(unique(colNames1))))
names(col1) = sort(unique(colNames1))
cols1 = as.character(col1[colNames1])



plot(pcaResults$x[,'PC1'],pcaResults$y[,'PC2'],main='PCA of cancer',col=cols1,pch=19,xlab='PC 1',ylab='
legend('topright',legend=unique(colNames1),fill=col1)
```



DESeq prereq### ###Differential count but primarily to find genes based on pval

```
condition <- factor(c('invasive ductal carcinoma', 'breast adenocarcinoma', 'breast carcinoma', 'metapla
','squamous cell breast carcinoma, acantholytic variant', 'breast ductal adenocarcinoma'))

dds <- DESeqDataSetFromMatrix(countData = round(Data_Br_unique), colData = colData, design =~ condition
dds <- DESeq(dds)
```

Compare 2 conditions and get gene based on pvalue for pairwise set

```
DEresults_BACvsBC = results(dds, contrast = c("condition", 'breast adenocarcinoma','breast carcinoma'))

DEresults_BACvsBC <- DEresults_BACvsBC[order(DEresults_BACvsBC$pvalue),]
print (DEresults_BACvsBC)
```

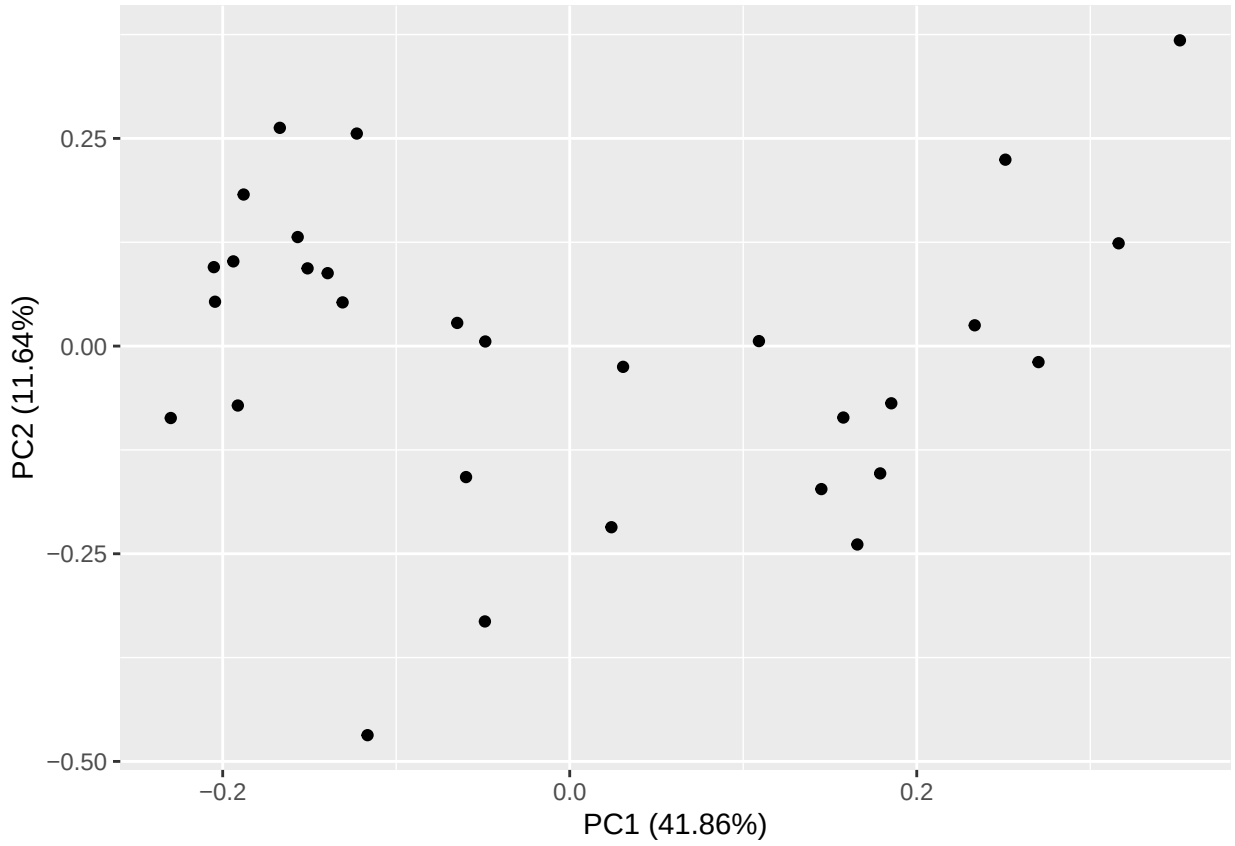
```
## log2 fold change (MLE): condition breast adenocarcinoma vs breast carcinoma
## Wald test p-value: condition breast adenocarcinoma vs breast carcinoma
## DataFrame with 11772 rows and 6 columns
##             baseMean log2FoldChange     lfcSE      stat      pvalue        padj
##            <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
## TAGLN        74.5208       -5.36985  0.951656  -5.64264 1.67466e-08 0.000189153
## GOLIM4       73.4796       -2.07533  0.390148  -5.31934 1.04144e-07 0.000386341
## COL1A1      261.0953       -6.51272  1.230060  -5.29464 1.19251e-07 0.000386341
## CLIP3        19.9382       -4.59137  0.871316  -5.26947 1.36819e-07 0.000386341
## LOXL3        13.2533       -4.35974  0.864299  -5.04425 4.55308e-07 0.001028541
```

```
## ...              ...          ...       ...       ...          ...          ...
## ORMDL3     216.23845      0.457976  0.726202  0.630646           NA           NA
## CSPG4        7.21135     -2.874618  1.383035 -2.078486           NA           NA
## C2orf88      4.61004     -4.603717  1.131447 -4.068878           NA           NA
## CD24       638.72631     -2.949352  1.128562 -2.613371           NA           NA
## AC008764.8   7.38216     -3.803898  0.925853 -4.108533           NA           NA
```

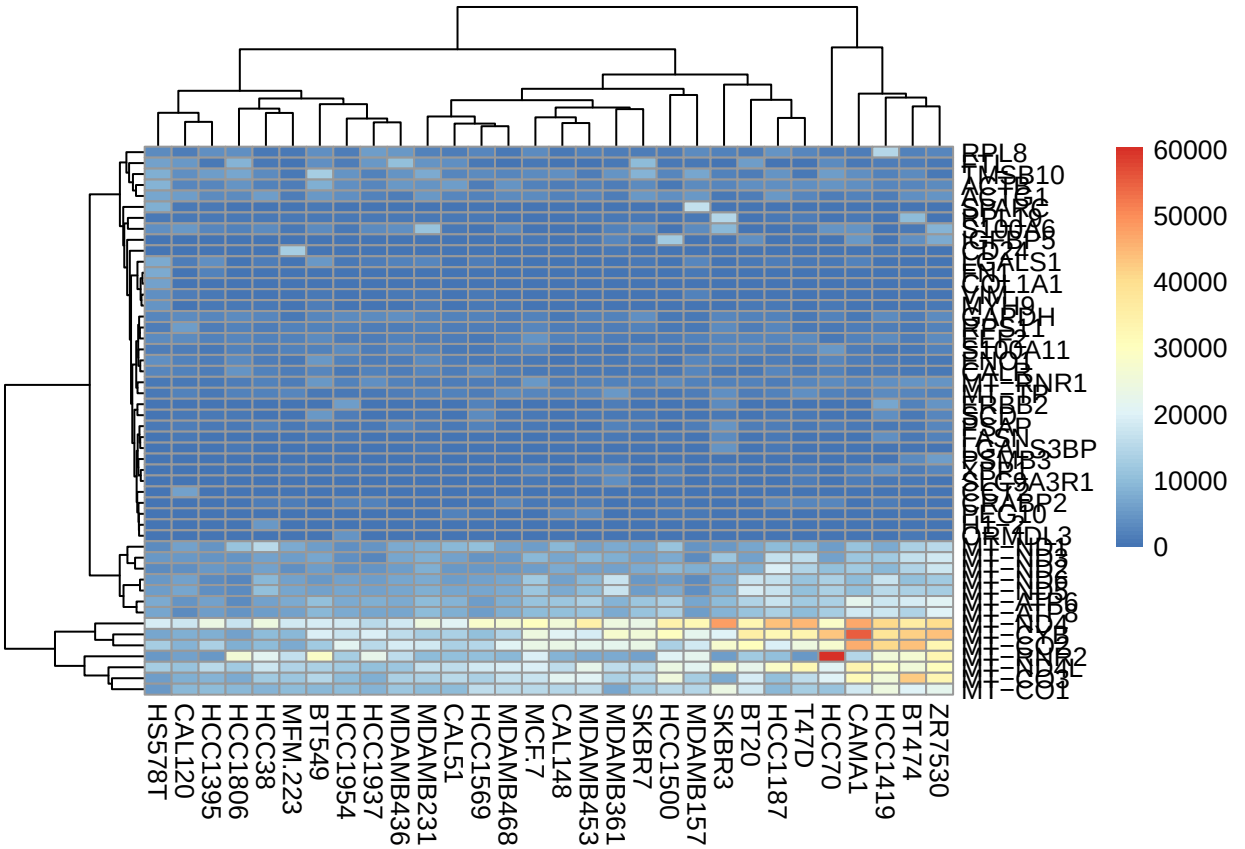
```
write.csv(DEresults_BACvsBC, file="DEresults_BACvsBC.csv")
```

PCA & Plot heatmap for normalized DESEq result

```
countsNormalized <- DESeq2::counts(dds, normalized=TRUE)
selectedGenes_dds <- names(sort(apply(countsNormalized, 1, var), decreasing=TRUE)[1:50])
normMatrix <- t(countsNormalized[selectedGenes_dds,])
normMatrix <- log2(normMatrix+1)
pcaResult_Norm<- prcomp(normMatrix)

autoplot(pcaResult_Norm, data = colData)
```



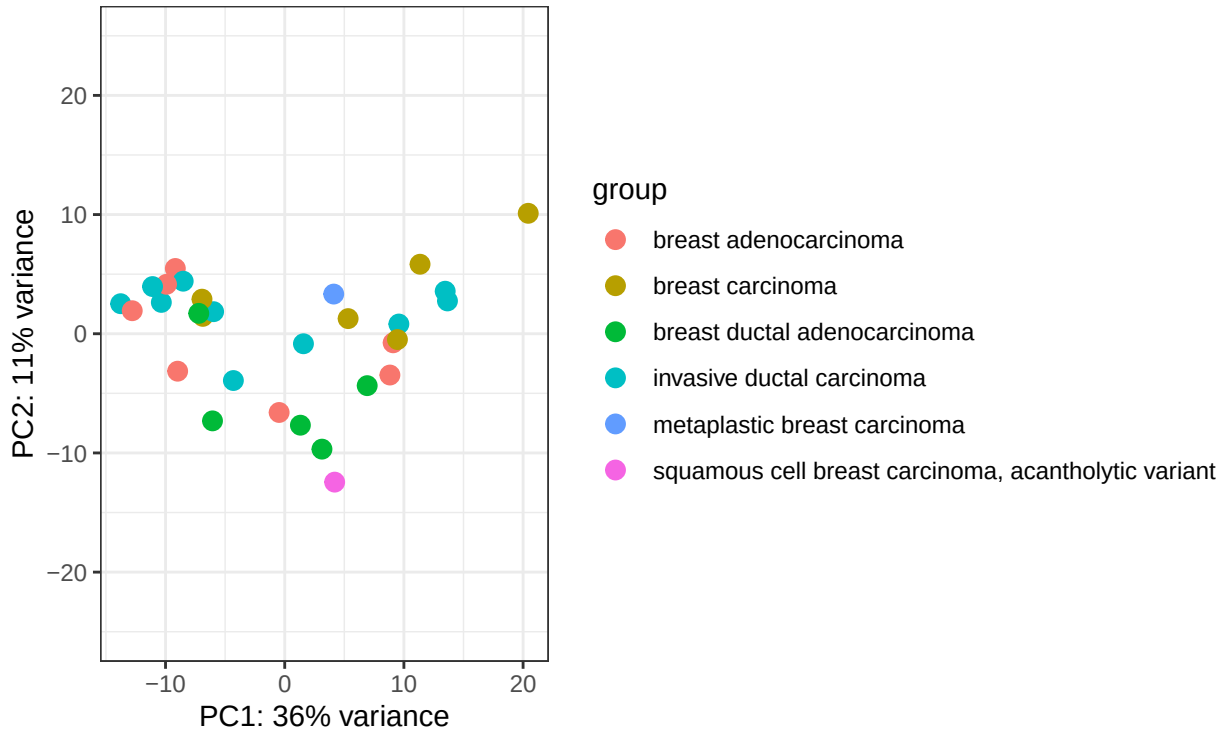

```
pheatmap(as.matrix(countsNormalized[selectedGenes_dds,]))
```




```
rld_Br <- rlog(dds)
```

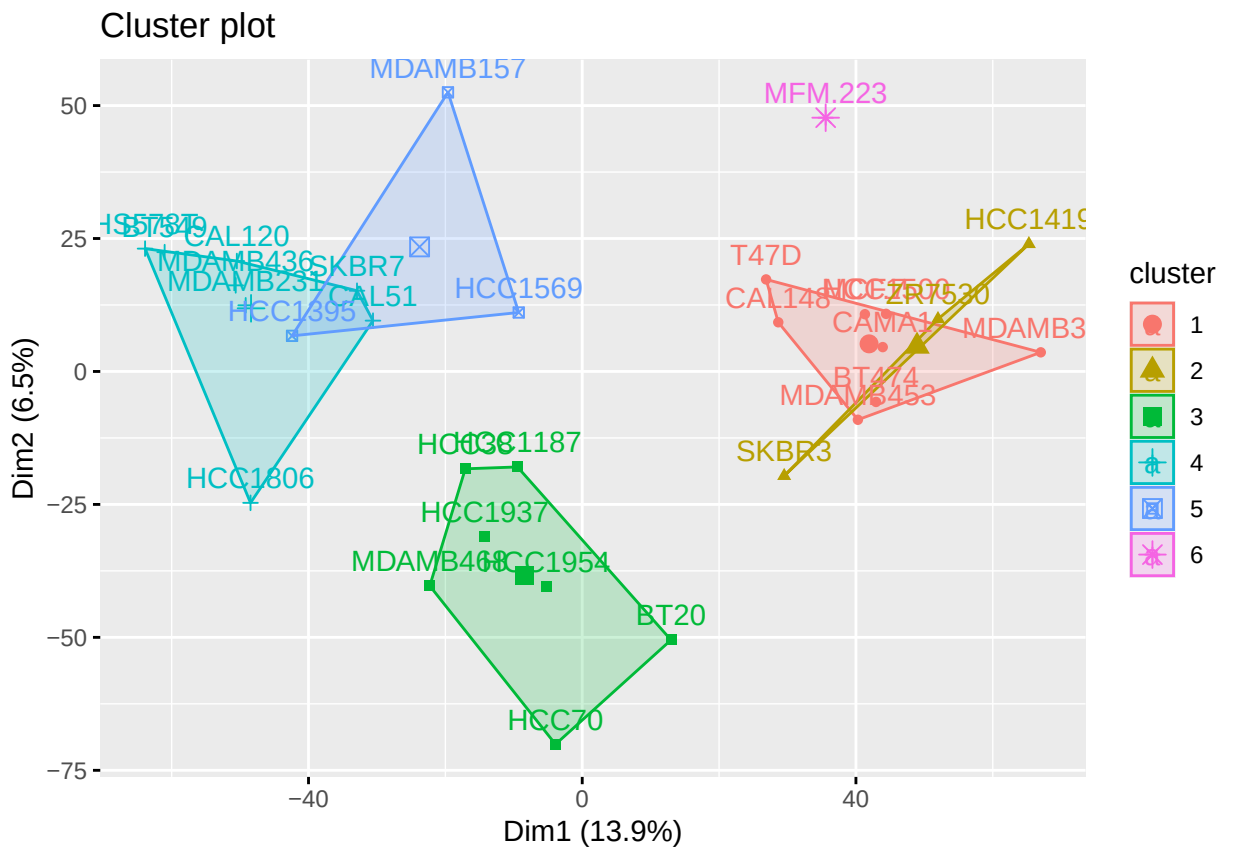
```
## rlog() may take a few minutes with 30 or more samples,
## vst() is a much faster transformation
```

```
DESeq2::plotPCA(rld_Br, ntop=100, intgroup = 'condition') + ylim(-25,25) + theme_bw()
```



CLuster using factoextra based on normalized count from DESeq

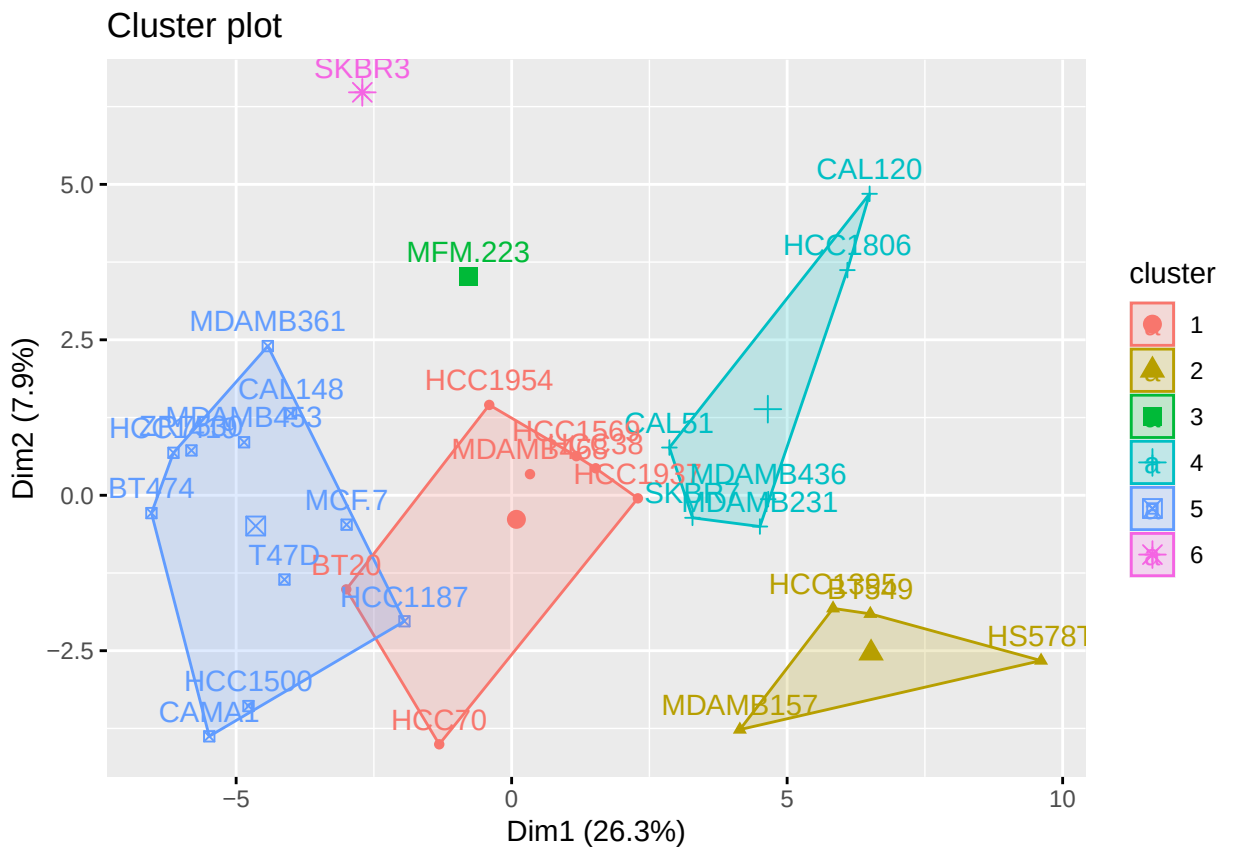
```
countsNormalized_log <- log2(countsNormalized + 1)
scale_countsNormalized_log_t <- scale(t(countsNormalized_log))
km_countsNormalized_log_t <- kmeans(scale_countsNormalized_log_t, 6, nstart=25)
fviz_cluster(km_countsNormalized_log_t, scale_countsNormalized_log_t, ellipse = TRUE)
```




```
cluster_Br <- km_countsNormalized_log_t$cluster
```

cluster based on variance for top 100 genes

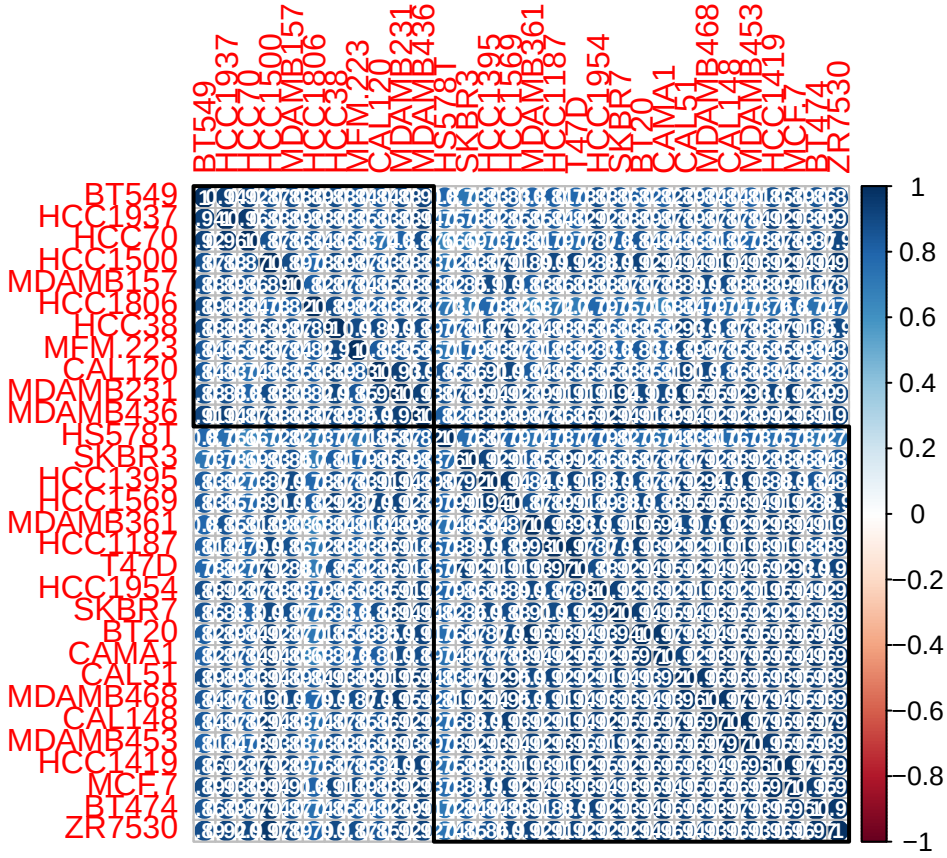
```
scale_Matrix <- scale(Matrix)
km_Matrix <- kmeans(scale_Matrix, 6, nstart=25)
fviz_cluster(km_Matrix, scale_Matrix, ellipse = TRUE)
```



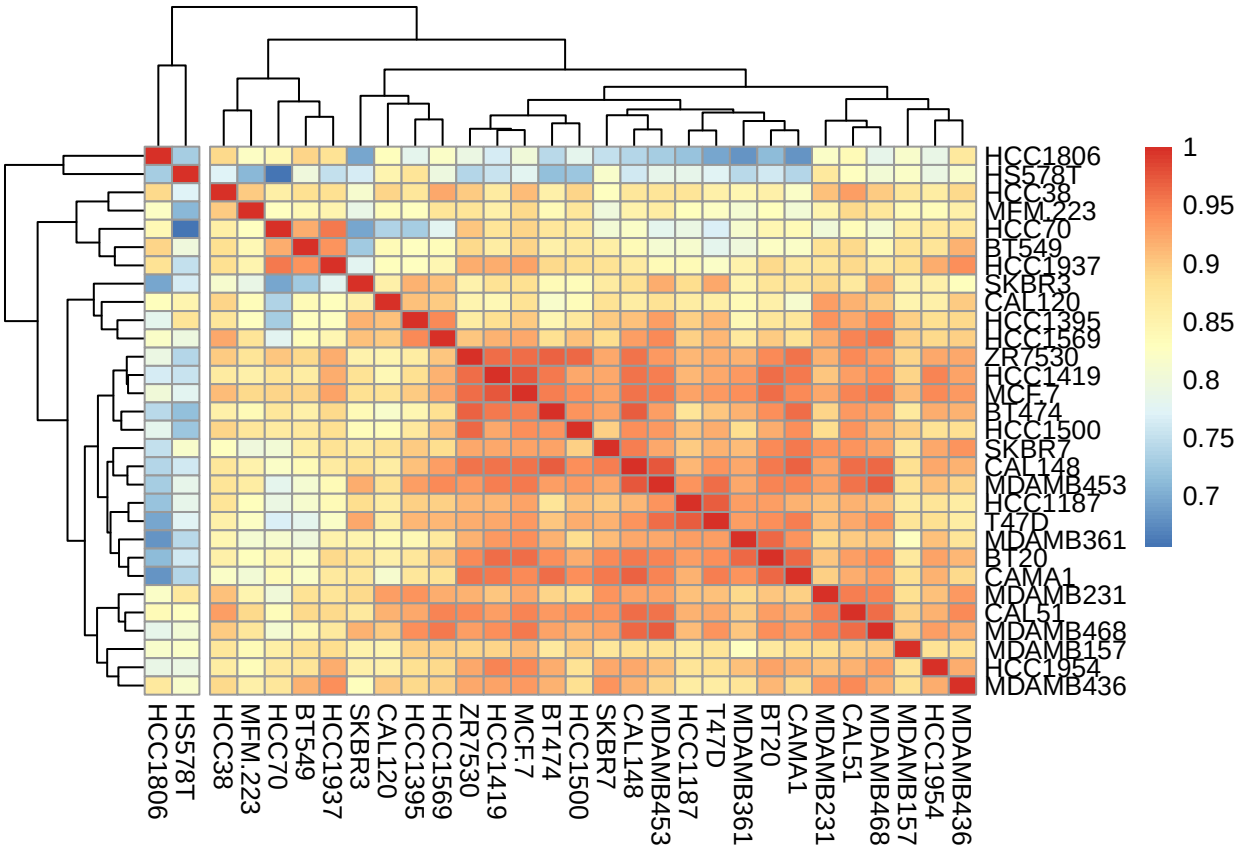

```
cluster_Br_Var <- km_Matrix$cluster
```

check the correlation between the cell lines

```
correlationMatrix <- cor(countsNormalized)


corrplot::corrplot(correlationMatrix, order = 'hclust',
                   addrect = 2, addCoef.col = 'white',
                   number.cex = 0.7)
```

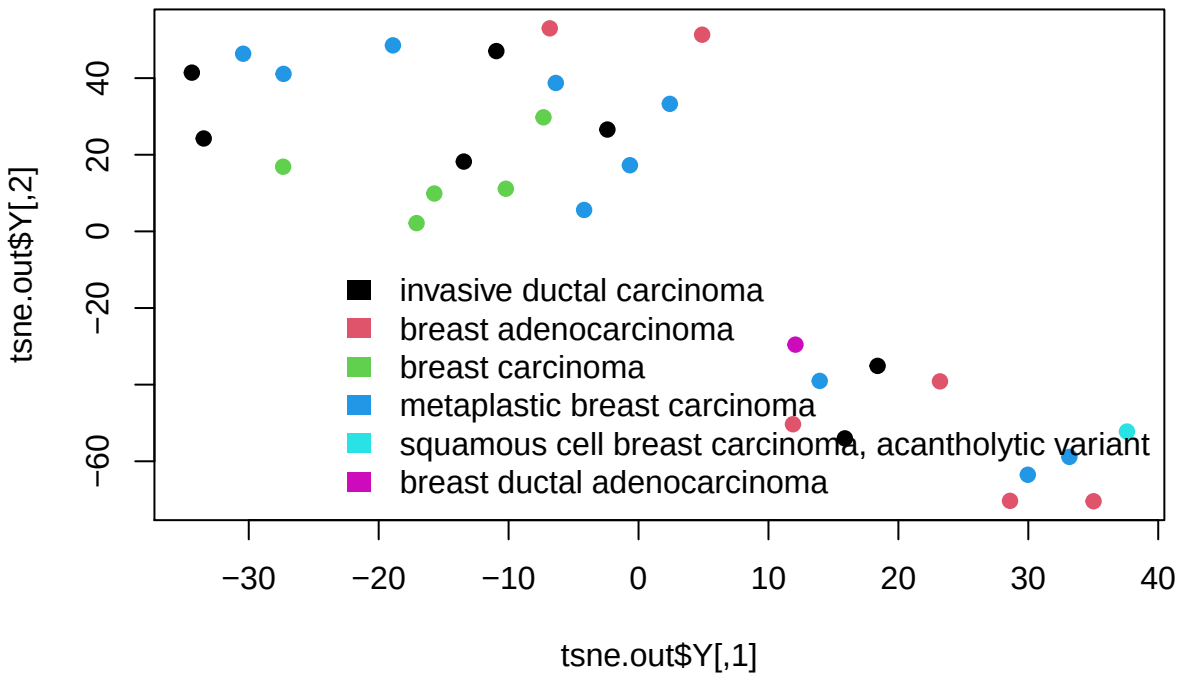



```
pheatmap(correlationMatrix,cutree_cols = 2)
```



TSNE but data is not as huge

```
library(Rtsne)
set.seed(46)
tsne.out <- Rtsne(Matrix,perplexity = 5)

plot(tsne.out$Y,col=as.factor(colData$condition),
     pch=19)

legend("bottomright",
       legend=unique(colData$condition),
       fill =palette("default"),
       border=NA,box.col=NA)
```


tsne.out$Y[,2]
40
20
0
−20
−60
−30
−20
−10
0
10
20
30
40
tsne.out$Y[,1]
invasive ductal carcinoma
breast adenocarcinoma
breast carcinoma
metaplastic breast carcinoma
squamous cell breast carcinoma, acantholytic variant
breast ductal adenocarcinoma