Capstone project

Hien Le

21/12/2021

Title

Evidence about the suppression of glomerular filtration rate in castration resistant prostate cancer

Abstract

An increase in the mortality rate and number of patients with prostate cancer (PC), particularly the advanced and metastatic cancer, have raised the challenges in prostate cancer therapy. Over the past several decades, androgen receptor has been the pimary therapeutic option for patients with advanced PC. However, castration-resistant prostate cancers (CRPC) developed as a poor prognosis stage for the androgen deprivation therapy and induced cancer mortality. Due to CRPC being incurable, clinical researches and laboratory investigations have been focused on an investigation and understanding of CRPC progression. In this study, we have performed RNA Illumina HiSeq 2000 to explore the difference in transcription levels between PC and CRPC samples. Over forty clinical samples have been collected from 13 CRPC and 30 PC patients to analysis upon over ten thousand transcripts. A sustained androgen receptor (AR) signal is still regarded as the main cause of CRPC. Additionally, Glomerular Filtration Rate (GFR) has been proposed as a novel target that causes CRPC, and this has led to the development of novel agents targeting Ras, PI3K/Akt, p53, and cell cycle signaling. These mechanisms could raise the investigation for cancer proliferation and survival in CRPC more than PC patients. Taken together, our findings reveal a key molecular determination of differences between PC and CRPC at the level of transcriptiome. Furthermore, the finding of GFR could provide an insight into the novel cancer therapy for CRPC treatment.

Keywords: prostate cancer, castration resistant prostate cancer, growth factor receptor, cell cycle, cell death.

Introduction

Prostate cancer (PC) is the second leading cause of death in males worldwide, but the molecular mechanism bebind the PC invasion and migration is very limited. This has allowed the development of novel therapies with specific targets. In several centuries, due to the emergence of androgen-independent prostate tumor growth, however, PCa secures as androgen-independent, highly metastatic advanced disease as castration-resistant prostate cancer (CRPC), defined as disease progression despite serum testosterone levels of < 20 ng/dl. The recent discovery that AR signaling involves systemic castration through intratumoral production of androgens led to the development of novel anti-androgen therapies consisting of known clinical drugs as abiraterone acetate and enzalutamide. Although these agents effectively inhibit disease symptoms and prolong patients'life, CRPC still induces the limitation of AR deprivation therapy in cancer treatment. An increased understanding of the mechanisms that underlie the pathogenesis of CRPC is therefore needed to investigate and develop novel therapeutic approaches for this disease. Recently, RNA sequencing is revolutionizing the study of the transcriptome. A highly sensitive and accurate tool for measuring expression across the transcriptome, it is providing the revolution with visibility into previously undetected changes occurring in disease status, in response to therapeutic, under different environmental conditions, and across a broad range of other study designs. It is well reported to detect both known and novel features in a single assay, enabling the detection of transcript isoforms, genes fusion, single nucleotide variants, and other feature without the limitation of prior knowledge. Here, we hypothesized that there is still a significant amount of unexplored transcriptome which are differences between CRPC and PC. To investigate the first comprehensive views of protein-coding genes in these cancer models, we examined whole transcriptome sequencing of 13 CRPCs and 30 untreated PCa. In addition to identify potentially regulated pathways, annotation data conduction and enrichments were used to conduct pathway visualization. The findings provide the new sign in the differences of transcripts levels between CRPCs and PCs samples that address several aspects of challenges in PCa treatments, molecular markers, mechanism, and explanation of cell phenomena.

Results

Comprehensive transcriptome analysis reveals alterations in key regulator pathways

After filtering the low-expressed genes, a comprehensive view of prostate cancer (PC) and castration-resistant prostate cancer (CRPC) was presented. As shown in the below figure, herarchical clustering presents that CRPC_531 and PC_6864 samples were computed together.

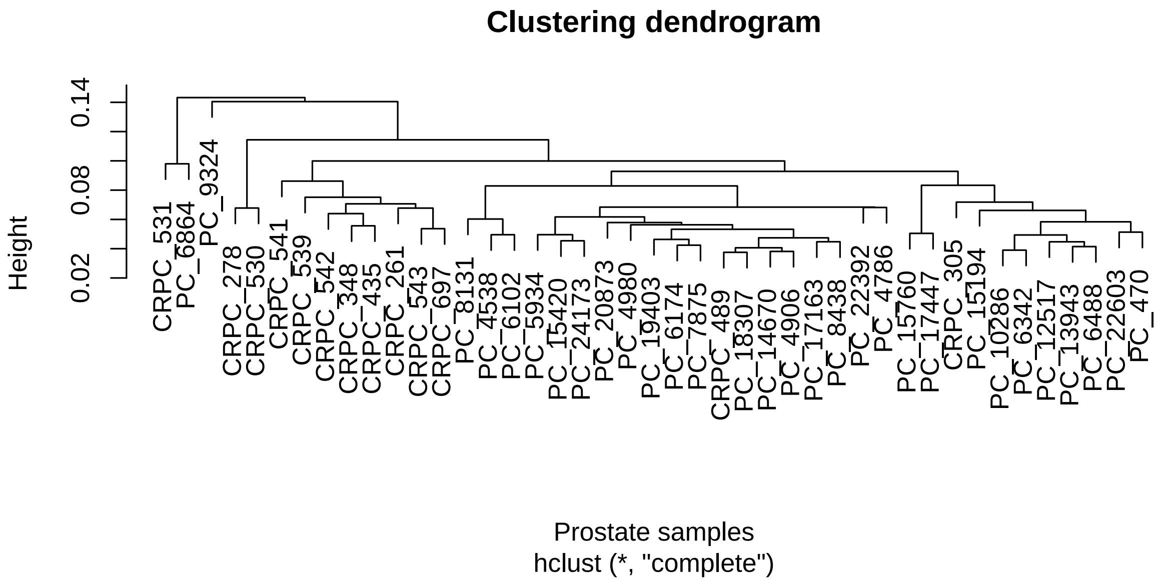


Figure 1: Hierarchical clustering of annotated genes

Notably, principal component analysis (PCA) of gene-expression profile presented CRPC and PC samples as green and red in a visualization graph, respectively (Figure 2). It is separated the distribution of the samples from the same group on the overall correlation. Differentially expressed genes were identified using the t-test with the threshold for significant difference $p < 0.01$. In detail, as shown in a volcano plot, 573 genes are noticed the log2-ratio above 1 and $p < 0.01$ as red whereas 1099 genes are filtered with only the log2-ration above 1 as blue. It is noted that 1174 genes are satisfied with the adjusted p values less 0.01 as green.

To clarify the differential expression of statistical genes, heatmap was performed for the visualization as shown in Figure 3. We identified genes significantly differentially expressed in the two groups if they passed the threshold of adjusted p value < 0.01 and fold change of >1.

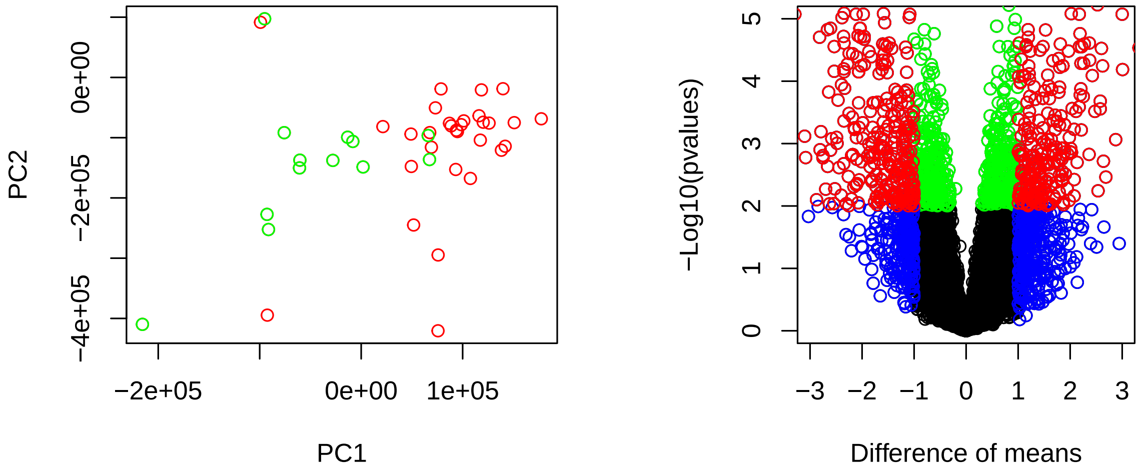


Figure 2: Analysis of expression level characterization

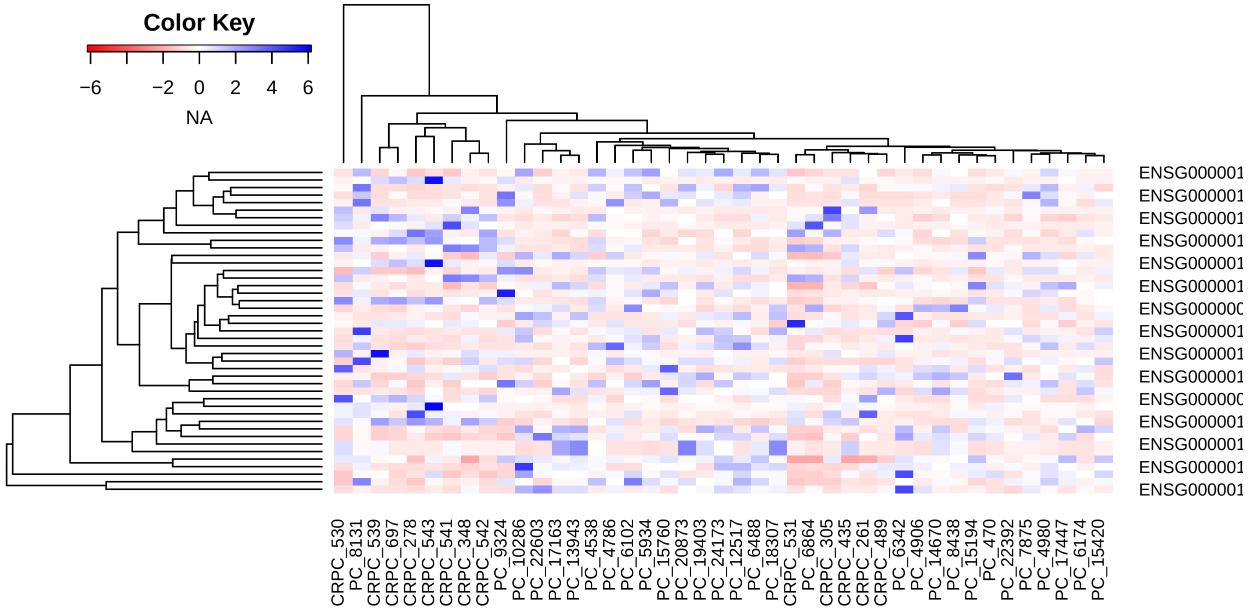


Figure 3: Heat map of annotated genes

Regulated pathway analysis

The differential expressed transcripts were categorized through an annotation tool where enrichment analysis of biological processes was performed to determine the potential KEGG pathway regulated by the different cancer stages. We could observe that androgen receptor (AR), cell cycle, olfactory transduction, and cell proliferation in prostate were identified as potential regulated signaling in a comparison between two groups. Altered processes were indicated by regulatory pathways in which EGR family, CDK family, ATF family, DUSP1, DUSP5, and PTEN play a crucial role (Supplementary table 1). These genes were over-expressed in PCa, but not in CRPCs. To highlight the pathway level changes in PC in a comparison between PC and CRPC, we constructed pathway models to observe expression changes into gene levels CRPCs in a comparison with PCs. AR expression level was detected as the lower expression in CRPCs than PCs. AR is required as the mandatory control for the development of normal prostate and associated with the expansion of prostate tumors. The combination between natural ligands and AR leads to transport dimerization into nucleus to regulate with the promoter or enhancer regions of DNA; which mediates various transcription factors and growth pathways such as cell proliferation, anti-apoptotic response, and PSA level. In this study, it is observed the increase in PSA in CRPCs in a comparison with PCs. Notably, Glomerular Filtration Rate (GFR) pathway was identified as a potential in a comparison between two groups. Several important pathways were regulated by GFR as shown in the below figure, including Ras, PI3K/Akt, and p53. In detail, we noticed a significant combined expression of the proliferation markers in ETV5, MMP, Ras, PSA in CRPCs, suggesting a high cellular invasion, proliferation, and cell survival. In addition, p53 signaling pathway was listed as the potential target in a comparison between two groups with the notification of a high expression level of MDM2 and a low expression of p53 in CRPCs. The high proliferation rate was also reflected in the expressions of b-catenin, cyclinD1, BCL2 were also identified in PC. The low expression of tumor suppressors as CDKN1B, NKX3.1, and PTEN were listed in CRPCs. Notably, loss of tumor suppressor phosphatase and tensin homolog (PTEN) was observed in CRPCs samples. In cell cycle, we noted a strong combined lower expression of the cell cycle markers and inhibitors such as CDK family, cyclin E, Chk, MCM family, Bub amily, and Ocr family in PC samples, suggesting a lower cell progression in PCs than CRPCs. For a deeper insight, we also analyzed the key genes regulated in cell progression and DNA damage process upon the both groups. The top up-regulated genes consisting CDK1, CDK2, CDK4, CDK6, and CDK7 which were detected as lower expression in PCs are the families of protein kinase involved in regulating transcription, mRNA processing, and the differentiation of cell progression. Furthermore, MCM2-7 were noticed as lower expression in PCs than CRPCs, which plays a central role in the replication as replicative DNA helicase, and form a hexameric ring-shaped complex around DNA replication. BUB1 and BUB3, responsible for the establishment of the mitotic spindle assembly checkpoint and chromosome congression, have lower expression in PCs than CRPCs.

```
##          pathway_identifier      pValue
## hsa00983           hsa00983 0.008192434
## hsa05215           hsa05215 0.009639375
## hsa04740           hsa04740 0.009860331
## hsa04270           hsa04270 0.013023206
## hsa04621           hsa04621 0.024965405
## hsa04916           hsa04916 0.028417269
## hsa03060           hsa03060 0.032480635
## hsa04910           hsa04910 0.035481977
## hsa04110           hsa04110 0.039368282
## hsa04976           hsa04976 0.048910978
##                                 pathway_name
## hsa00983     Drug metabolism - other enzymes
## hsa05215                     Prostate cancer
## hsa04740              Olfactory transduction
## hsa04270  Vascular smooth muscle contraction
## hsa04621 NOD-like receptor signaling pathway
## hsa04916                       Melanogenesis
## hsa03060                      Protein export
## hsa04910           Insulin signaling pathway
## hsa04110                          Cell cycle
## hsa04976                      Bile secretion
```

Discussion

Analysis 8: The findings in this analysis is same line with what I expect to see based on the previous publications.

We have performed to investigate the similarity and correlation between CRPCs and PC groups. Interestingly, the hierarchical clustering indicated that some CRPCs and PCs could be computed together. It could be explained that a similar or close stage of cancer could be detected in a deeper view between these samples. CRPCs have been investigated for several centuries after the failure of androgen inhibition therapies cause most of the death from prostate cancer, which raises the requirements to fully understand CRPC until molecular and transcriptions. In here, the decrease in AR expression level in CRPCs was observed when comparing with PC group. Several previous studies have reported that almost prostate tumor promotes in an AR state, thus androgen deprivation therapy is executed as the mandatory administration and results in improved clinical outcomes. However, some high-risk PCa moderately develops to CRPC due to AR variants, point mutations, amplifications, and changes in it cofactors. This finding is consistent with the line that CRPCs have the high expression of AR. Furthermore, it is well known that the decrease in PTEN, a tumor suppressor, is one of the most frequent alterations in PCa, which is present around of CRPC patients around 40% of CRPC patients prior received AA post-docetacxel. The activity of PTEN involves a number of cellular processes consisting of cell growth, migration, invasion, and cellular architecture. Preclinical studies have investigated that evidence about PTEN function in the activation of PI3K/Akt pathway which regulates AR signaling and worse prognosis. Likewise, PTEN loss and subsequence protein kinase B activation confer radian and chemotherapy resistance. Over expression of AKt active form stimulates cell growth and invasivenes, and cell survival mechanism. This provides the evidence about the therapy development in CRPC patients, which could include AR-targeted therapies and PI3K/Akt inhibitors. In addition, CDKN1B inhibitor downregulation was observed in CRPCs as coding for p21 protein, which may regulate at both transcriptional and transnational levels. There are multiple pathways that influence the down-expression of p21 and thereby, increase cell proliferation and further simulates the release of G1 phase in cell cycle. The tumor suppressor function of p21/CDKN1B was reported in in vivo and in vitro test. In in vivo models, mice inhibited the expression of p21 was indicated to be more susceptible to hematopoietic, epithelial, and endothelial tumor progression than normal mice. The down-expression of p53 tumor suppressor gene, which could contribute to uncontrolled cell growth and proliferation have related to prostate tumors. Presence of a p53 protein can be considered as a potential biomarker of disease recurrence after patients received radical prostatectomy. Other authors have found evidence between p53 protein and features of aggressive disease such as metastasis PCa and a decrease in survival. These studies were conducted on prostate tissues as there is no currently available assay to determine and evaluate this biomarker in blood due to the instability of p53 protein. However, this question related to p53 remain inclusive for advanced PCa therapy. There have few hypotheses about TP53 mutation involved in metastasis PCa progression in clinical samples of mCRPC patients. Other molecular markers which have a role in controlling cell cycle have been explored. The well-known deputation is p27 which inhibits the cells cycle and DNA damage. Immunohistochemical studies on samples from radical prostatectomy have found an inverse correlation between p27 protein expression and cancer prognosis.

The higher moderate and mobility have been reported in CRPCs model. During PC progression, the androgen axis engages the growth factor network to an active cross-talk conferring a survival and invasion advantage of prostate cancer cells. The current finding dissects the regulation between AR and GFR in controlling tumor progression. In here, the decrease or decline in the GFR implies the progression of underlying kidney disease or the occurrence of a superimposed insult to the kidneys. This is most commonly due to problems such as dehydration and volume loss. An improvement in the GFR may indicate that the kidneys are recovering some of their function. The finding of downexpression level of GFR in this study is in the same line about the function GFR in prostate. Of late, MAPK signaling pathway, PI3K-Akt signaling pathway, and p53 signaling have shared the regulation with GFR expression. Taken together, these findings give the overall view about the differences in transcription levels between CRPCs and PCs, which might support the future therapy treatment for CRPCs.

Reference

1.Trancriptome sequencing reveal PCAT5 as a novel ERG-regulated long noncoding RNA in prostate cancer, cancer research, 2015. 2. Loss of phosphatase and tensin homolog or phosphoinositol-3 kinase activation and response to trastuzumab or lapatinib in human epidermal growth factor receptor 2-overexpressing locally advanced breast cancers, clinical oncology 2011. 3. Molecular Characterization of Prostate Cancers in the Precision Medicine Era, cancer, 2021. 4. Expression of p53 and its mechanism in prostate cancer, oncology letters, 2018. 5. Role of Androgen Receptor in Prostate Cancer: A Review, World journal of men's health, 2018. 6. Current management of castrate-resistant prostate cancer, current oncology, 2010.

Material and methods

Patient samples and sequencing

Fresh-frozen tissue specimens were obtained from prostate cancer patients during surgery. There includes 43 samples from two main groups with 13 castration-resistant prostate cancer (CRPC) patients and 30 untreated prostate cancer (PC) patients. RNA isolated from the samples were subjected to Illumina HiSeq 2000. Briefly, the sequencing reads were aligned to human genome, and expression estimates for all expressed transcripts were measured.

Data preparation

The data was obtained by performing RNA-seq experiment over 43 prostate tissue samples. After loading data, 21,990 transcripts are computed. Among over 20,000 genes, it is found that 601 genes are not expressed in our data set. To reduced the effect of noise, the lowly expressed transcripts were filtered. There is no exact definition for what makes a gene lowly expresses. Here, median visualization will present the overall view for determination of the low expressed genes. The median of gen counts across samples is below 500. There, it is suggested that gens that have counts smaller than or equal to the value of the lower quartile (25%-12.84675) to be considered as the low expressed genes. 5498 genes are cut off due to the low expression for reducing the noise in further analysis. The median of filtered genes are higher after filtering out the low expressed genes.

To compare the difference of gene expression between 43 samples, the normalization is performed to achieve the difference expression values of genes in the total intensity. DESeq2 package (verson 1.24.0) is used to observe the normalization value with DESeqDataSetFromMatrix function. To address the difference of gene expression after and before filter and normalization, the boxplot is applied to observe. While taking into accessing gene analysis, Ensemble IDs of genes are instead by gene symbols using the information on package "biomaRt" databases version 2.40.5. The non-mapping Ensemble IDs were removed and the mapping Ensemble IDs were replaced by gene symbol for creating a new data matrix for further analysis.

Herarchical clustering analysis

To create herarchical clustering, correlation values between two samples were calculated to access the distance between them. The hierarchical clustering of samples is drawn for the data set following Pearson and Spearman correlation. The Pearson's correlation coefficient is a widely used statistical score to measure the co-variables. It is particularly used in the case of hierarchical clustering applied to the RNA-seq data.

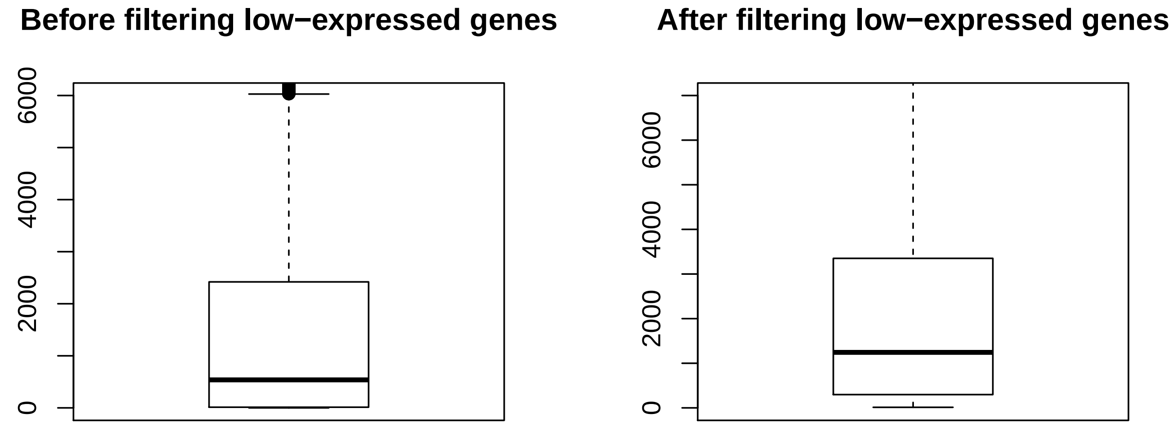
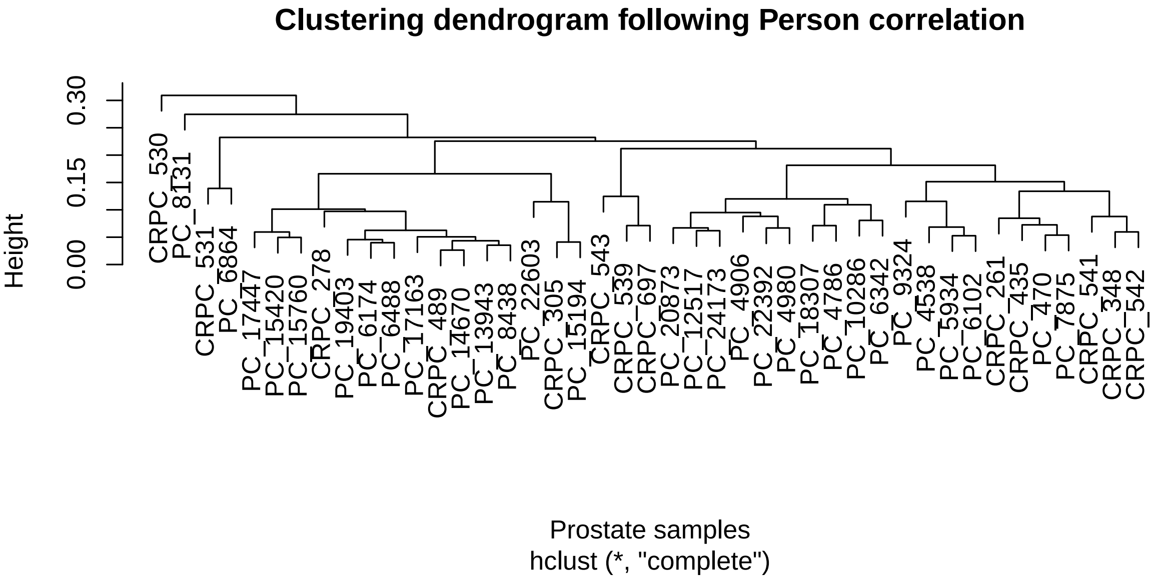
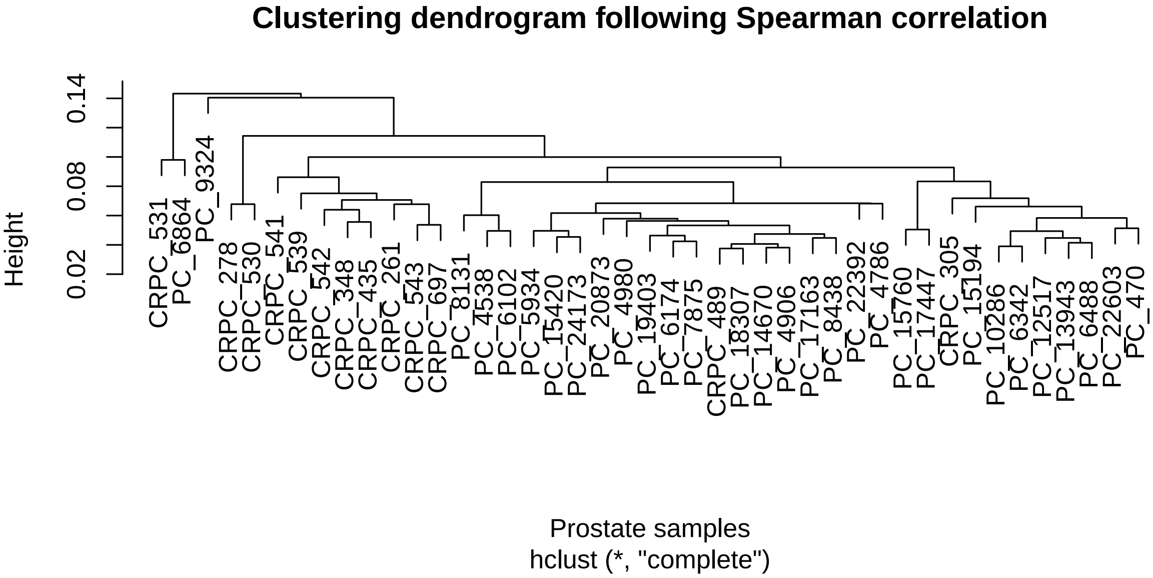


Figure 4: Box plot of samples group together after normalization and filter

Meanwhile, Spearman's correlation coefficient can be also particularly useful to assess the similarity of two given expression profile. Thus, this rank-based measure is more robust to outlines than Pearson's correlation coefficient, it could be less sensitive.





Clustering validation method would be to choose the optional number of clusters by minimizing the within-cluster sum of squares and maximizing the between-cluster sum of square. The Pearson correlation evaluates the linear relationship between two continuous variables while the Spearman correlation coefficient is based on the ranked values for each variable rather than the raw data. The purpose of this research is to find the difference between two groups. In addition, the cluster sums of squares between samples in Spearman correlation are often less than in Pearson correlation. Therefore, Spearman correlation is indicated as the better cluster model for further analysis in this study.

Principal component analysis

To reduce the dimensional of samples, principal component analysis (PCA) is used to visualize relationships and clusters in the data and identify essential characteristics of samples. The visualization of CRPC and PC was indicated as green and red, respectively. In here, more input features often make a predictive modeling task more chaleenging to model, more generally referred to as the curse of dimensionality.

Statistical analysis

Due to find differential expressed genes between the two sample groups, the data set was analysed using t-test and Wilcoxon test of “genefilter” packages verson 1.66.0 with the threshold for significant difference adjusted $p < 0.01$. A t-test is a type of statistical test that is used to compare the means of two group whereas Wilcoxon test is indicated as a non-parametric statistical hypothesis test purposed either to test the location of a set of samples or to compare the locations of two groups. As shown in histogram of adjusted p-values in t.test and Wilcoxon test, the flat distribution along the bottom is all a null p-values which are uniformly distributed between 0 and 1. This is a part of a definition of a p-value: under the null, it has a 5% chance of being less than 0.5, a 10% chance of being less than 0.1,etc. This describes a uniform distribution. Then p-adjustment was performed to reduced a false positive in t.test. We could see the difference of p-values via histogram distribution.

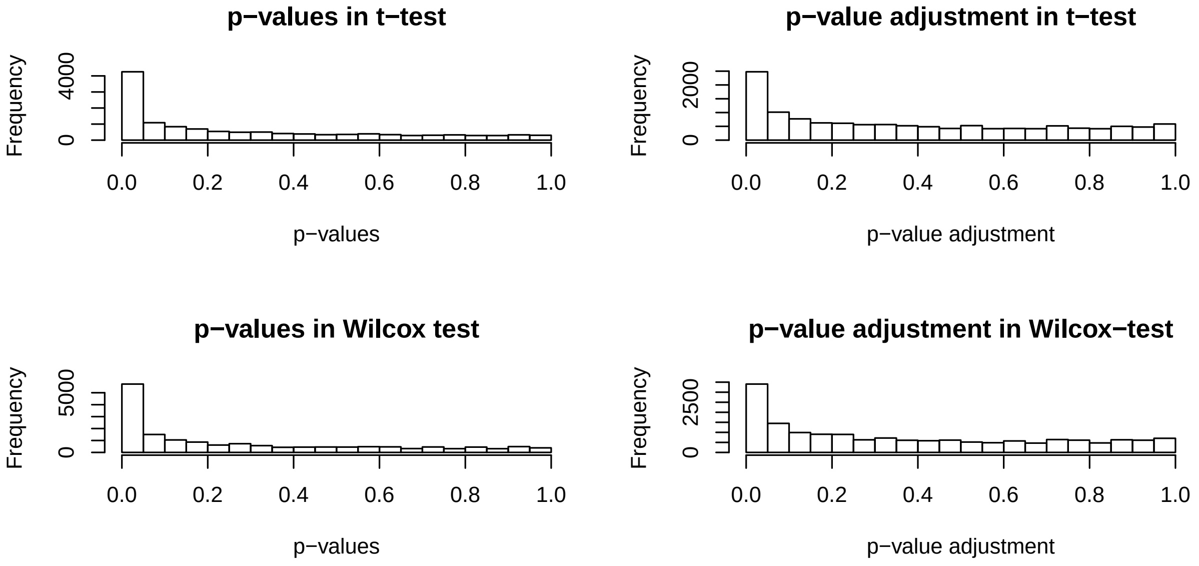


Figure 5: Histograme of p-values upon t-test and Wilcoxon test

Notably, I tested three threshold of p-value 0.05, 0.01, and 0.001. I have considered the threshold of p-value 0.01 could be the best option based on visualization of heatmap and the results of pathway analysis. t-test and Wilcoxon test indicate 1174 and 1424 transcripts satisfied the threshold adjusted $p < 0.01$, respectively, with the sharing transcripts about 863 indications. Due to the flat distribution along the bottom in adjusted p-values in t-test and the purpose of t-test, the results of adjusted $p < 0.01$ in t-test was used for further analysis.

To observe the distribution of p-values in t-test via the volcano plot, I used the threshold of adjustment p values less 0.01 and the different expression over 1. The different among adjusted p value was presented by colors with green corresponding for adjusted $p < 0.1$, blue corresponding for log2(different expression) $>$ 1, red corresponding for adjusted $p < 0.01$ and log2(different expression) >1, and black corresponding for adjusted p values unsatisfied the above requirements.

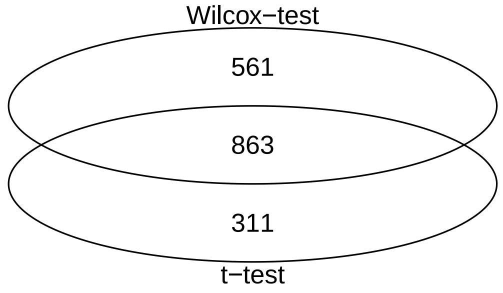


Figure 6: Comparision of statistical expressed genes upon t-test and Wilcoxon test

Heatmap

To take into account the comprehensive analysis, heatmap is performed to visualize the representation of statistical data set as colors. In heat maps the data is displayed in a grid where each row represented a gene and each column represent a sample. In here, heatmap is used to present the selected statistical genes upon the clustering model in the above analysis. We indicated the expression level of genes via color with blue corresponding as a high expression, red corresponding as a low expression, and white as unchanged expression level.

Pathway analysis

To indicate novel pathway related to the data set, bioconductor annotation data package "KEGG.db" verson 3.2.3 was accessed with two main index "KEGGEXTID2PATHID" and "KEGGPATHID2NAME" for pathway identifiers and pathway name, respectively, based on gene ID of the data set. Based on the filter data set and the list of statistical gene analysis from the above analysis, we could find N corresponding to the total number of genes in all KEGG pathways, n corresponding to the number of genes in each KEGG pathway, M corresponding to number of genes in our statistical gene list, and k corresponding to the number of differential expression genes in in each list. In here, a one-tailed test is selected to test in which the critical areas of a distribution is one-sided so that it is either greater or less than a certain values, but not bot, The statistical annotation pathway was calculated as one-tailed analysis with threshold $p < 0.05$. Thus, we could have the list of annotation pathway. The list includes the information of p-value and pathway identifier. To see the name of pathway, we use the index "KEGGPATHID2NAME" to assess the name list. The index "KEGGPATHID2NAME" contains pathway identifiers for assessing and subsequently exporting the pathway name.

Pathway view

To fully characterize the wealth of expressed transcripts in different stages of prostate cancer, pathway analysis was presented regarding bioconductor annotation data package "pathview" in R with the fold change of transcription levels was indicated by color. For visualization of the pathway view, we need the change of different expression genes between PC and CRPC samples, pathway identifier, format of input, and species.

R scripts

Analysis 1: Data preparation

```
### Data preparation
setwd("/student_data/BBT.BI.202_2021/students/leh/project_work")

# To load data
count_matrix = readRDS("RNA_expressions.RDS")

# To calculate the number of genes that are not expressed in the data set
df_count_matrix <- data.frame(rowSums(count_matrix))
colSums(df_count_matrix == 0) #601 genes

# To choose a threshold by defining a gene to be lowly expressed.
row_medians <- apply(count_matrix, 1, median)
par(pty = "s")
boxplot(row_medians, ylim = c(0,6000)) # To observe the summary expression of total genes

# To define genes that have counted smaller than or equal to the value of the lower quartile.
threshold <- quantile(row_medians, 0.25, na.rm=FALSE, names = TRUE)
threshold #12.84675
df_row_medians <- data.frame(apply(count_matrix, 1, median))
colSums(df_row_medians <= threshold) #5498 genes would be cut off

# To make the new data matrix by filtering with threshold
df_count_matrix <- data.frame(count_matrix, fix.empty.names=F)
valid_rows <- df_row_medians > threshold
count_matrix_filtered <- df_count_matrix[valid_rows,]
par(las = 2)
boxplot(apply(count_matrix_filtered, 1, median), ylim = c(0,7000))

### Normalization
library(DESeq2)
column_names = colnames(count_matrix_filtered)
sample_types = matrix(sub(pattern="_.*",
                          replacement="",
                          column_names)
                      )
rownames(sample_types) = column_names
colnames(sample_types) = c("Type")

# To create the data set expected by DESeq2
dataset <- DESeqDataSetFromMatrix(countData=round(count_matrix_filtered),
                                  colData=sample_types,
                                  design=~1)
dataset <- estimateSizeFactors(dataset)
count_matrix_filtered_mor_normed <- counts(dataset,
                                           normalized=TRUE)
```

```
### To convert Ensembl IDs to gene symbol
library('biomaRt') # To load biomaRt packages
ensembl_gene_ids <- rownames(count_matrix_filtered_mor_normed)
to_convert = ensembl_gene_ids
mart <- useDataset("hsapiens_gene_ensembl", useMart("ensembl"))
mappings <- getBM(
                  filters= "ensembl_gene_id",
                  attributes = c("ensembl_gene_id","hgnc_symbol"),
                  values=to_convert,
                  mart= mart)
mappings

# To select on unique gene
# Filter the map to remove duplicate
mappings_filter = mappings[!duplicated(mappings[ , c("ensembl_gene_id")]),]

#Find item from rownames
temp_data = rownames(count_matrix_filtered_mor_normed) %in% mappings_filter$ensembl_gene_id

#Filter count_matrix_filtered_mor_normed with temp_data
count_matrix_filtered_mor_normed_filtered = count_matrix_filtered_mor_normed[temp_data,]

#This new matrix only contains row names from mappings_filter$ensembl_gene_i
dim(count_matrix_filtered_mor_normed_filtered)

#Create an empty hgnc_symbols_vec
hgnc_symbols_vec = c()

#Get all row names of count_matrix_filtered_mor_normed_filtered. Store in 'rowNameEnIDs'
rowNameEnIDs = rownames(count_matrix_filtered_mor_normed_filtered)
for( i in 1:length(rowNameEnIDs)){
  # Find rowNames[i] in mappings_filter$ensembl_gene_id
  hgnc_symbols_vec = c(hgnc_symbols_vec, mappings_filter[
    mappings_filter$ensembl_gene_id %in% rowNameEnIDs[i], 'hgnc_symbol'])
}
length(hgnc_symbols_vec)

#Copy count_matrix_filtered_mor_normed_filtered to a new df
count_matrix_hgnc_symbols = count_matrix_filtered_mor_normed_filtered

#Now, change the row names from EnsemId to hgnc_symbol
rownames(count_matrix_hgnc_symbols) = hgnc_symbols_vec
head(count_matrix_hgnc_symbols)
dim(count_matrix_hgnc_symbols)#16486 x 43
```

```
# To calculate correlation distance
correlation_dist = function(x, method="pearson"){
  corr_distance = as.dist((1 - cor(x, y = NULL, method = method))/2)
  return(corr_distance) ## the class of the return value should be 'dist'
}

# To draw clustering following pearson method
```

```
a = correlation_dist(count_matrix_hgnc_symbols[,1:43], method = "pearson")
hc_pearson = hclust(a, method = "complete")
par(las = 2)
plot(hc_pearson,
     main = "Clustering dendrogram following Person correlation",
     xlab = "Prostate samples")

# To draw clustering following spearman method

b = correlation_dist(count_matrix_hgnc_symbols[, 1:43], method ='spearman')
hc_spearman = hclust(b, method = "complete")
par(las = 2)
plot(hc_spearman,
     main = "Clustering dendrogram following Spearman correlation",
     xlab = "Prostate samples")

# To check the correlation distance between random samples regarding each correlation method
round(correlation_dist(count_matrix_hgnc_symbols[,  1:3], method ='pearson'), 2)
round(correlation_dist(count_matrix_hgnc_symbols[,  1:3], method ='spearman'), 2)
```

Analysis 2: Hierarchical clustering

Analysis 3: Dimensionality reduction

```
### Principal component analysis (PCA)
pca_result <- prcomp(x=t(count_matrix_hgnc_symbols),
                     retx = TRUE,
                     center = TRUE,
                     scale. = TRUE)
str(pca_result)

points2 <-t(t(pca_result$rotation[,1:2])%*%count_matrix_hgnc_symbols)
dim(points2)

# To create the PCA 2D plot
points <- pca_result$rotation
CRPC_points <- points2[1:13,]
PC_points <- points2[14:43,]
par(las = 2)
plot(CRPC_points,
     xlab="BPH Samples",
     ylab="Expression level",
     main="Scatterplot of the CRPC data",
     col = "red")
plot(PC_points,
     xlab="PC Samples",
     ylab="Expression level",
     main="Scatterplot of the PC data",
     col = "green")
text(PC_points,
     labels=row.names(PC_points),
     data=PC_points,
     cex=0.9,
     font=2
     )

plot(points2, col="red") # PC samples
points(points2[1:13,], col="green") # CRPC samples
text(PC_points, labels=row.names(PC_points),data=PC_points, cex=0.9, font=2)

# Another way for visualization
library(ggplot2)
pca <- prcomp(x=t(count_matrix_hgnc_symbols), retx = TRUE, center = TRUE, scale. = TRUE)
pca.data <- data.frame(Sample=row.names(pca$x), X=pca$x[,1], Y=pca$x[,2])
pca.var <- pca$sdev^2
pca.var.per <- round(pca.var/sum(pca.var)*100,1)
par(pty="s")
ggplot(data=pca.data,aes(x=X,y=Y, label=Sample)) +
  geom_text() +
  xlab(paste("PC1 - ", pca.var.per[1], "%", sep="")) +
  ylab(paste("PC2 - ", pca.var.per[2], "%", sep="")) +
  theme_bw() +
  ggtitle("PCA analysis of Prostate cancer gene expression data")
```

Analysis 4: Differential expression analysis

```
### Principal component analysis (PCA)
library(genefilter)
cols = as.vector(colnames(count_matrix_filtered_mor_normed))
cols
count_matrix_cols = sub("_[0-9]*", "", cols)
count_matrix_cols
count_matrix.fac = factor(count_matrix_cols)
count_matrix.fac

# Prepare data for ttest
log2_count_matrix = log2(count_matrix_filtered_mor_normed)

# t-test values
ttest_results = rowttests(log2_count_matrix, count_matrix.fac)
ttest_results = ttest_results[ttest_results$p.value != "NaN",]
statcrit_ttest = which(ttest_results$p.value < 0.01)
length(statcrit_ttest)
par(las = 2)
hist(ttest_results$p.value, xlab = "p-values", main = "Histograme of p-values")

# t-test adjust
ttest_results$p.value.adj = p.adjust(ttest_results$p.value, method = "BH")
statcrit_adjust = which(ttest_results$p.value.adj < 0.01)
length(statcrit_adjust)
par(las = 2)
hist(ttest_results$p.value.adj,
     xlab = "p-values adjust",
```

```r
     main = "Histograme of p-values adjust"
     )

# To see the difference between t-test and t-test adjust on p-values
plot(ttest_results$p.value,
     ttest_results$p.value.adj,
     xlab = "p-value",
     ylab = "p-value adjust",
     main = "Correlation between ttest and ttest-adjust"
     )
abline(h = 0.05, v = 0.05, col = "red")

# To make the table which contains the gene are statisfied all condition of t-test and t-test adjust
ttest_table = ttest_results[ttest_results$p.value < 0.01,]
ttest_adjust_table = ttest_results[ttest_results$p.value.adj < 0.01,]

gene_cutoff = count_matrix_filtered_mor_normed[
  rownames(ttest_results) %in% rownames(ttest_adjust_table),
  ]
gene_cutoff_pvalues = ttest_results[
  rownames(ttest_results) %in% rownames(ttest_adjust_table),
  ]

# To order p-value
gene_cutoff_pvalues_order = gene_cutoff_pvalues[order(gene_cutoff_pvalues$p.value.adj, decreasing = FALS

### A VOLCADO plot (week3, ex3)
#Genes that satisfy none of the criteria are colored black
#Genes that satisfy the p-value criterion are colored green
#Genes that satisfy the effect criterion are colored blue
#Genes that satisfy both of the criteria are colored red

totcrit <- which((ttest_results$p.value.adj < 0.01) & (abs(ttest_results$dm) > 1))
statcrit <- which(ttest_results$p.value.adj < 0.01)
effcrit<- which(abs(ttest_results[,2]) > 1)
par(pty="s")
plot(ttest_results[,2],
     -log10(ttest_results[,4]),
     xlab = "Difference of means",
     ylab="-Log10(pvalues)",
     xlim = c(-3,3),
     ylim = c(0, 5)
     )
points(ttest_results[statcrit, 2], -log10(ttest_results[statcrit, 4]), col="green")
points(ttest_results[effcrit, 2], -log10(ttest_results[effcrit, 4]), col="blue")
points(ttest_results[totcrit, 2], -log10(ttest_results[totcrit, 4]), col="red")

### Gene symbol
gene_cutoff_pvalues_order = gene_cutoff_pvalues[
  order(gene_cutoff_pvalues$p.value,
        decreasing = FALSE),
  ]
ensembl_gene_ids_pvalues <- rownames(gene_cutoff_pvalues_order)
```

```
to_convert_pvalues = ensembl_gene_ids_pvalues
mart <- useDataset("hsapiens_gene_ensembl", useMart("ensembl"))
mappings_pvalues <- getBM(
                  filters= "ensembl_gene_id",
                  attributes = c("ensembl_gene_id","hgnc_symbol"),
                  values=to_convert_pvalues,
                  mart= mart)
mappings_pvalues

mappings_pvalues_filter = mappings_pvalues[!duplicated(mappings_pvalues[ , c("ensembl_gene_id")]),]
mappings_pvalues_filter = mappings_pvalues_filter[mappings_pvalues_filter$hgnc_symbol != "",]
temp_data_pvalues = rownames(gene_cutoff_pvalues_order) %in% mappings_pvalues_filter$ensembl_gene_id
gene_cutoff_pvalues_order_filtered = gene_cutoff_pvalues_order [temp_data_pvalues,]

hgnc_symbols_vec_pvalues = c()
rowNameEnIDs_pvalues = rownames(gene_cutoff_pvalues_order_filtered)
for( i in 1:length(rowNameEnIDs_pvalues)){
  hgnc_symbols_vec_pvalues = c(hgnc_symbols_vec_pvalues, mappings_pvalues_filter[
      mappings_pvalues_filter$ensembl_gene_id %in% rowNameEnIDs_pvalues[i], 'hgnc_symbol'])
}
length(hgnc_symbols_vec_pvalues)
gene_cutoff_hgnc_symbols = gene_cutoff_pvalues_order_filtered
gene_cutoff_hgnc_symbols$gene_symbol = hgnc_symbols_vec_pvalues
gene_cutoff_hgnc_symbols

### Wilcox test
my_wilcox = function(v, group1, group2) {
    test_result = wilcox.test(x=v[group1], y=v[group2], exact=FALSE)
    p.value = test_result$p.value
    names(p.value) = "p.value"
    return(c(test_result$statistic, p.value))
}
wilcox_results <- apply(count_matrix_filtered_mor_normed,
                        1,
                        my_wilcox,
                        group1=1:13,
                        group2=14:43
                        )
wilcox_results_df = as.data.frame(t(wilcox_results))
#p-values in Wilcoxon test less than 0.01
statcrit_wilcox = which(wilcox_results_df$p.value < 0.01)
length(statcrit_wilcox)
# Adjustment p-value in Wilcox rank-sum test
wilcox_results_df$p.value.adj.BH <- p.adjust(wilcox_results_df$p.value, method = "BH")
statcrit_wilcox_adjust <- which(wilcox_results_df$p.value.adj.BH < 0.01)
length(statcrit_wilcox_adjust)

### Gene symbol for wilcox_results
wilcox_results_df
wilcox_adjust_table = wilcox_results_df[wilcox_results_df$p.value.adj < 0.01,]
gene_cutoff_wilcox = wilcox_results_df[
  rownames(wilcox_results_df) %in% rownames(wilcox_adjust_table),
  ]
```

```r
gene_cutoff_wilcox_order = gene_cutoff_wilcox[
  order(gene_cutoff_wilcox$p.value,
        decreasing = TRUE),
  ]
ensembl_gene_ids_wilcox <- rownames(gene_cutoff_wilcox_order)
to_convert_pvalues = ensembl_gene_ids_wilcox
mart <- useDataset("hsapiens_gene_ensembl", useMart("ensembl"))
mappings_wilcox <- getBM(
                  filters= "ensembl_gene_id",
                  attributes = c("ensembl_gene_id","hgnc_symbol"),
                  values=to_convert_pvalues,
                  mart= mart)
mappings_wilcox

# To select on unique gene
mappings_wilcox_filter = mappings_wilcox[
  !duplicated(mappings_wilcox[ , c("ensembl_gene_id")]),
  ]
mappings_wilcox_filter = mappings_wilcox_filter[
  mappings_wilcox_filter$hgnc_symbol != "",
  ]
temp_data_wilcox = rownames(gene_cutoff_wilcox_order) %in%
  mappings_wilcox_filter$ensembl_gene_id

gene_cutoff_wilcox_order_filtered = gene_cutoff_wilcox_order[temp_data_wilcox,]
dim(gene_cutoff_wilcox_order_filtered)

hgnc_symbols_vec_wilcox = c()
rowNameEnIDs_wilcox = rownames(gene_cutoff_wilcox_order_filtered)
for( i in 1:length(rowNameEnIDs_wilcox)){
  hgnc_symbols_vec_wilcox = c(hgnc_symbols_vec_wilcox, mappings_wilcox_filter[
      mappings_wilcox_filter$ensembl_gene_id %in% rowNameEnIDs_wilcox[i], 'hgnc_symbol'])
}
length(hgnc_symbols_vec_wilcox)
gene_cutoff_hgnc_symbols_wilcox = gene_cutoff_wilcox_order_filtered
gene_cutoff_hgnc_symbols_wilcox$gene_symbol = hgnc_symbols_vec_wilcox
head(gene_cutoff_hgnc_symbols_wilcox)
dim(gene_cutoff_hgnc_symbols_wilcox)
gene_cutoff_hgnc_symbols_wilcox = gene_cutoff_hgnc_symbols_wilcox[
        order(gene_cutoff_hgnc_symbols_wilcox$p.value.adj.BH, decreasing = FALSE),]

# Venn plot
ttest <- row.names(ttest_results)[ttest_results$p.value.adj < 0.01]
wilcox <- row.names(wilcox_results_df)[wilcox_results_df$p.value.adj.BH < 0.01]
venn_data <- list(ttest, wilcox)
names(venn_data) = c("t-test", "Wilcox-test")
library(gplots)
venn(venn_data)

# To create an exel file
write.table(gene_cutoff_hgnc_symbols,
            file = "ttest_result.tsv",
            quote=FALSE,
            sep='\t'
            )
write.table(gene_cutoff_hgnc_symbols_wilcox,
            file = "wilcox_result.tsv",
            quote=FALSE,
            sep='\t'
            )

# To creat ensemble ID list name
write.csv(row.names(ttest), file = "ensembleID_ttest.csv")
write.csv(row.names(wilcox), file = "ensembleID_wilcox.csv")
```

Analysis 5: Heatmap of differntially expressed genes

```
#Find row names from ttest_results which have adjusted t-test p-value are below 0.01
row_names_ttest_result_filter = row.names(ttest_results)[ttest_results$p.value.adj < 0.01]
row_names_ttest_result_filter = row.names(ttest_results)[abs(ttest_results$dm) > 1]
row_names_ttest_result_filter

#Filter in count_matrix
count_matrix_de_ttest = count_matrix_filtered_mor_normed[
  row.names(count_matrix_filtered_mor_normed) %in% row_names_ttest_result_filter, ]
count_matrix_de_ttest

#Plot a heat map of count_matrix_de_test
mycol = colorpanel(75, "red", "white", "blue")
heatmap.2(count_matrix_de_ttest,
          Rowv = as.dendrogram(hc_spearman),
          col = mycol, density.info = "none",
          trace = "none",
          dendrogram = "both",
          key.xlab = "NA",
          key.ylab = "NA",
          scale = "row",
          labRow = FALSE
          )
```

Analysis 6

```
### KEGG pathway
gene_cutoff_pvalues = ttest_results[
  rownames(ttest_results) %in% rownames(ttest_adjust_table),
  ]

library(KEGG.db)
map_ke <- as.list(KEGGEXTID2PATHID)
# To get entrezgene_id for KEGG identifiers
library(biomaRt)
ensembl_gene_ids = rownames(count_matrix_filtered_mor_normed)
```

```
# mapping
to_convert_de_ttest = ensembl_gene_ids
mart <- useDataset("hsapiens_gene_ensembl", useMart("ensembl"))
mappings_ensembl_entrez <- getBM(
                  filters= "ensembl_gene_id",
                  attributes = c("ensembl_gene_id", "entrezgene_id"),
                  values=to_convert_de_ttest,
                  mart= mart)
dim(mappings_ensembl_entrez)

# Annotation data
kegg_entrez_pathway = map_ke[!is.na(match(
  mappings_ensembl_entrez$entrezgene_id, names(map_ke)
  )
  )
  ]
ensembl2kegg_pathway = list()
for (i in 1:nrow(mappings_ensembl_entrez)) {
  temp = kegg_entrez_pathway[as.character(mappings_ensembl_entrez[i, ]$entrezgene_id)]
  if (!is.na(names(temp))){
    ensembl2kegg_pathway[mappings_ensembl_entrez[i, ]$ensembl_gene_id] = temp
  }
}

ensembl2kegg_pathway_table = table(unlist(ensembl2kegg_pathway))
N = sum(!is.na(unlist(ensembl2kegg_pathway)))
N

# Calculate M
gene_list = gene_cutoff_pvalues
names(ensembl2kegg_pathway) %in% rownames(gene_list)
ensembl2kegg_pathway_de = ensembl2kegg_pathway[
  names(ensembl2kegg_pathway) %in% rownames(gene_list)]
unlist(ensembl2kegg_pathway_de)
M = sum(!is.na(unlist(ensembl2kegg_pathway_de)))
M

ns = c()
ks = c()

ensembl2kegg_pathway_table = table(unlist(ensembl2kegg_pathway))
ensembl2kegg_pathway_de_table = table(unlist(ensembl2kegg_pathway_de))

#ns
ensembl2kegg_pathway_table_filter = ensembl2kegg_pathway_table[
  names(ensembl2kegg_pathway_table) %in% names(ensembl2kegg_pathway_de_table)]

for( i in 1:length(ensembl2kegg_pathway_table_filter)){
  ns[i] = ensembl2kegg_pathway_table_filter[[i]]
}
ns

#ks
```

```
ensembl2kegg_pathway_de_table = table(unlist(ensembl2kegg_pathway_de))
ensembl2kegg_pathway_de_table
for( i in 1:length(ensembl2kegg_pathway_de_table)){
  ks[i] = ensembl2kegg_pathway_de_table[[i]]
}
ks

#pValue
pValues <- c()
for(i in 1:length(ks)){
  pValues[i] = sum(dhyper(ks[i]:ns[i], M, N-M, ns[i]))
}
pValues
names(pValues) = names(ensembl2kegg_pathway_de_table)
sum(pValues < 0.05)
df_pValues = data.frame(pathway_identifier = names(ensembl2kegg_pathway_de_table),
                        pValue = pValues
                        )
df_pValues_sort = df_pValues[order(df_pValues$pValue),]

# Pathway name
df_pValues_sort
map_ke_name <- as.list(KEGGPATHID2NAME)
pValue_sort_rownames = rownames(df_pValues_sort)
map_ke_names_array = names(map_ke_name)
pathway_names = c()
for( i in 1: length(pValue_sort_rownames)){
  temp = substring(pValue_sort_rownames[i], 4)
  name = map_ke_name[[which(map_ke_names_array %in% temp)]]
  pathway_names = c(pathway_names, name)
}
pathway_names
df_pValues_sort$pathway_name = pathway_names
df_pValues_sort

# pathway view
CRPC_samples = count_matrix_filtered_mor_normed[,1:13]
PC_samples = count_matrix_filtered_mor_normed[,14:43]
count_matrix_mean_diff = apply(PC_samples,1,mean) - apply(CRPC_samples,1,mean)
library(pathview)
count_matrix_mean_diff_matrix = as.matrix(count_matrix_mean_diff)
pv.out = pathview(gene.data = count_matrix_mean_diff_matrix,
                  gene.idtype = "ENSEMBL",
                  pathway.id = "04621",
                  out.suffix="NOD_signaling_p01",
                  limit=list(gene=2, cpd=1),
                  species = "hsa",
                  kegg.native = TRUE,
                  )
write.table(df_pValues_sort,
            file = "annotation_pathway_analysis.tsv",
            quote=FALSE,
            sep='\t'
            )
```