Bioconductor Lab 6 raport

Celem jest analiza eksperymentu RNA-seq – porównanie wyników ekspresji różnicowej uzyskanych za pomocą różnych pakietów - limma, edgeR, Deseq2

1. Zczytanie macierzy zliczeń z pliku i przygotowanie tebli porównań

```
library(limma)
library(ggplot2)
library(edgeR)
library(DESeq2)


count_matrix = read.table('C:\\Users\\marce\\hakowanie\\programming_R\\bioconductor\\lab6\\RNAcounts_basic.txt')
count_matrix


count_metrices <- lapply(seq(3, 13, by = 2), function(i) {

  cols <- c(colnames(count_matrix)[1],  colnames(count_matrix)[i],
            colnames(count_matrix)[2],  colnames(count_matrix)[i+1])

  df <- data.frame( count_matrix[, 1], count_matrix[, i],
              count_matrix[, 2], count_matrix[, i + 1])

  rownames(df) <- rownames(count_matrix)

  df <- setNames(df, cols)

  return(df)
})
```

Tworzona tutaj lista count_metrices zawiera wyodrębnione fragmenty macierzy zliczeń w taki sposób jak należy porównywać ze sobą próki. Na przykład poniżej pierwsze porównanie:

```
> data.frame(count_metrices[1])
          A2780a a4PTXa A2780b a4PTXb
DDX11L1        0      0      0      0
WASH7P        20     29     54      5
MIR6859-3      0      0      0      0
MIR6859-2      0      0      0      0
MIR6859-4      0      0      0      0
MIR6859-1      0      0      0      0
MIR1302-11     0      0      0      0
MIR1302-9      0      0      0      0
MIR1302-2      0      0      0      0
MIR1302-10     0      0      0      0
```

A2780a i b to próbki kontrolne natomiast próbki PTX są traktowane.
Mamy w sumie 6 takich jak powyżej tabel, gdzie kontrole w każdej są te same a próbki PTX wzrastają z 4 na 8…16..itd. Cyfry te oznaczają dawki leku PTX.

2. Swtorzenie projektu eksperymentu:

```
design <- factor(c('C', 'T', 'C', 'T'))
design <- model.matrix(~design)
design
colnames(design) <- c("Intercept","T")
```

```
> design
  Intercept T
1         1 0
2         1 1
3         1 0
4         1 1
attr(,"assign")
[1] 0 1
attr(,"contrasts")
attr(,"contrasts")$design
[1] "contr.treatment"
```

3. Badanie ekspresji różnicowej

```
results = list()

it = 0
for (matrix in count_metrices){
  it <- it + 1
  matrix <- data.frame(matrix)

  ### LIMMA ###
  dge <- DGEList(counts=matrix,  group = factor(c('C', 'T', 'C', 'T')))

  #flitorwanie
  keep <- filterByExpr(dge)
  #cat("Number of deleted genes:", sum(!keep), "\n")

  dge <-dge[keep, , keep.lib.sizes=FALSE]

  #normalizacja
  dge <- calcNormFactors(dge)

  #ekspresja róznicowa
  v <- voom(dge, design, plot=FALSE)
  fit <- lmFit(v, design)
  fit <- eBayes(fit)
  t_limma <- topTable(fit, coef=ncol(design), number=Inf)
  t_limma <- t_limma[t_limma$adj.P.Val<0.01,]

  ### EDGE R ###
  y <- estimateDisp(dge,design)
  et <-exactTest(y)
  t_edge <- topTags(et, n=Inf)
  t_edge <- t_edge$table
  adj.P.Val <- p.adjust(t_edge[,3], method='BH')
  t_edge <- data.frame(t_edge, adj.P.Val)
  t_edge <- t_edge[t_edge$adj.P.Val<0.01,]

  samples <- data.frame(sample=colnames(matrix), condition=factor(c('C', 'T', 'C', 'T')))
  samples
  dds <- DESeqDataSetFromMatrix(countData = matrix,
                                colData = samples,
                                design = ~ condition)

  keep <- rowSums(counts(dds)) >= 10
  dds <- dds[keep,]
  dds <- DESeq(dds)
  t_deseq <- results(dds)
  t_deseq <- t_deseq[!is.na(t_deseq$padj) & t_deseq$padj < 0.01, ]
  t_deseq <- data.frame(t_deseq)


  results[[paste('limma', it, sep = "_")]] <- t_limma
  results[[paste('edger', it, sep = "_")]] <- t_edge
  results[[paste('deseq', it, sep = "_")]] <- t_deseq

}
```

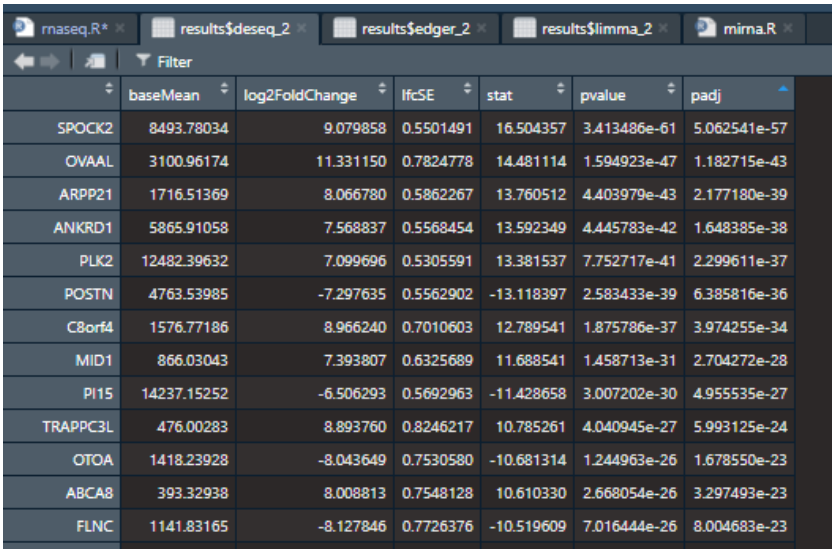
Powyższa pętla dokonuje wszystkich niezbędnych kroków (filtrowani, analiza, ekspresja różnicowa) w analizie i dodaje tabele z genami o ekspresji różnicowej na listę results. Każda tabela zostaje przefiltrowana tylko do genów o adj p value < 0.01.
W efekcie posiadamy na liście wyniki nazwane według konwencji limma_1..limma_6 i analogicznie dla pozostałych pakietów

4. Interpretacja wyników

Do wyżej sworzonej listy możemy łatwo się odwoływać aby zbadać wyniki poszczególnych porównań.

a. Diagramy Venna dla różnych porównań względem wszystkich metod:

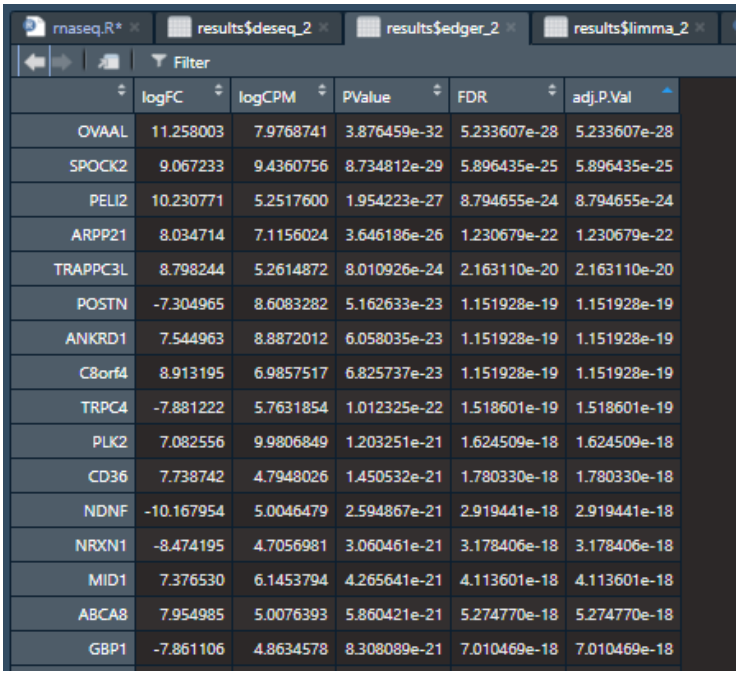
Przykładowe fragmenty stworzonych tabel z genami (dla próbek A8PTX):



rnaseq.R* | results$deseq_2 | results$edger_2 | results$limma_2 | mirna.R

Filter

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
SPOCK2	8493.78034	9.079858	0.5501491	16.504357	3.413486e-61	5.062541e-57
OVAAL	3100.96174	11.331150	0.7824778	14.481114	1.594923e-47	1.182715e-43
ARPP21	1716.51369	8.066780	0.5862267	13.760512	4.403979e-43	2.177180e-39
ANKRD1	5865.91058	7.568837	0.5568454	13.592349	4.445783e-42	1.648385e-38
PLK2	12482.39632	7.099696	0.5305591	13.381537	7.752717e-41	2.299611e-37
POSTN	4763.53985	-7.297635	0.5562902	-13.118397	2.583433e-39	6.385816e-36
C8orf4	1576.77186	8.966240	0.7010603	12.789541	1.875786e-37	3.974255e-34
MID1	866.03043	7.393807	0.6325689	11.688541	1.458713e-31	2.704272e-28
PI15	14237.15252	-6.506293	0.5692963	-11.428658	3.007202e-30	4.955535e-27
TRAPPC3L	476.00283	8.893760	0.8246217	10.785261	4.040945e-27	5.993125e-24
OTOA	1418.23928	-8.043649	0.7530580	-10.681314	1.244963e-26	1.678550e-23
ABCA8	393.32938	8.008813	0.7548128	10.610330	2.668054e-26	3.297493e-23
FLNC	1141.83165	-8.127846	0.7726376	-10.519609	7.016444e-26	8.004683e-23



rnaseq.R* | results$deseq_2 | results$edger_2 | results$limma_2

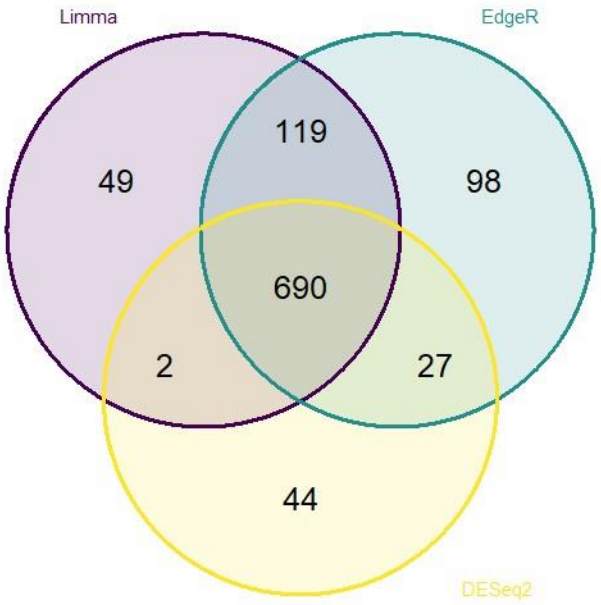
Filter

	logFC	logCPM	PValue	FDR	adj.P.Val
OVAAL	11.258003	7.9768741	3.876459e-32	5.233607e-28	5.233607e-28
SPOCK2	9.067233	9.4360756	8.734812e-29	5.896435e-25	5.896435e-25
PELI2	10.230771	5.2517600	1.954223e-27	8.794655e-24	8.794655e-24
ARPP21	8.034714	7.1156024	3.646186e-26	1.230679e-22	1.230679e-22
TRAPPC3L	8.798244	5.2614872	8.010926e-24	2.163110e-20	2.163110e-20
POSTN	-7.304965	8.6083282	5.162633e-23	1.151928e-19	1.151928e-19
ANKRD1	7.544963	8.8872012	6.058035e-23	1.151928e-19	1.151928e-19
C8orf4	8.913195	6.9857517	6.825737e-23	1.151928e-19	1.151928e-19
TRPC4	-7.881222	5.7631854	1.012325e-22	1.518601e-19	1.518601e-19
PLK2	7.082556	9.9806849	1.203251e-21	1.624509e-18	1.624509e-18
CD36	7.738742	4.7948026	1.450532e-21	1.780330e-18	1.780330e-18
NDNF	-10.167954	5.0046479	2.594867e-21	2.919441e-18	2.919441e-18
NRXN1	-8.474195	4.7056981	3.060461e-21	3.178406e-18	3.178406e-18
MID1	7.376530	6.1453794	4.265641e-21	4.113601e-18	4.113601e-18
ABCA8	7.954985	5.0076393	5.860421e-21	5.274770e-18	5.274770e-18
GBP1	-7.861106	4.8634578	8.308089e-21	7.010469e-18	7.010469e-18

	logFC	AveExpr	t	P.Value	adj.P.Val	B
SPOCK2	8.982184	5.886477049	16.637327	1.915018e-07	0.002056107	6.884311
PLK2	7.102319	7.416519260	14.522131	5.452321e-07	0.002056107	6.388085
ANKRD1	7.530046	6.083300688	14.493751	5.534670e-07	0.002056107	6.343231
ARPP21	8.003501	4.092153312	13.943123	7.444206e-07	0.002056107	5.828845
POSTN	-7.307966	5.950353438	-13.573823	9.137575e-07	0.002056107	5.806392
OVAAL	11.524699	3.221952318	13.649191	8.759593e-07	0.002056107	4.982204
PI15	-6.421818	7.864744153	-11.564847	3.075932e-06	0.003822789	4.972593
GNAI1	5.323445	6.097031150	11.083934	4.231015e-06	0.003822789	4.816498
DCLK1	5.192463	5.005692220	10.799123	5.140425e-06	0.003822789	4.622026
C8orf4	8.889377	3.428475638	11.445699	3.325025e-06	0.003822789	4.567143
MID1	7.331075	3.408684941	11.116287	4.139638e-06	0.003822789	4.544324
PELI2	10.269531	1.121773662	12.113043	2.169306e-06	0.003822789	4.360843
PCGF5	5.537547	4.143429232	10.375806	6.924593e-06	0.003822789	4.310482
TRPC4	-7.813260	2.787926520	-11.090114	4.213387e-06	0.003822789	4.207575

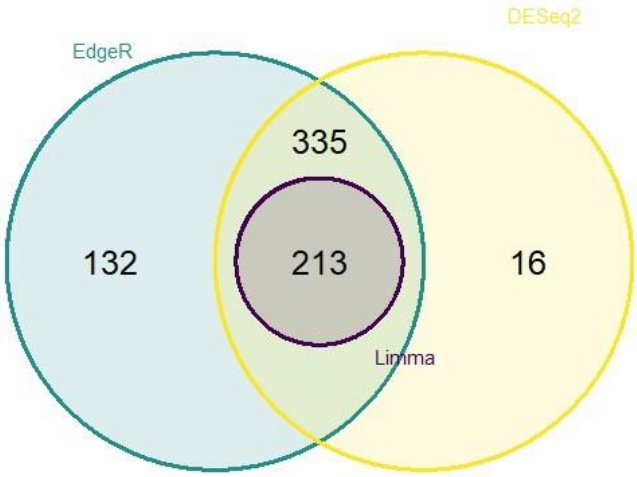
Diagramy venna dla wszystkich porównań w zależności od metody:

Porównanie 4PTX:



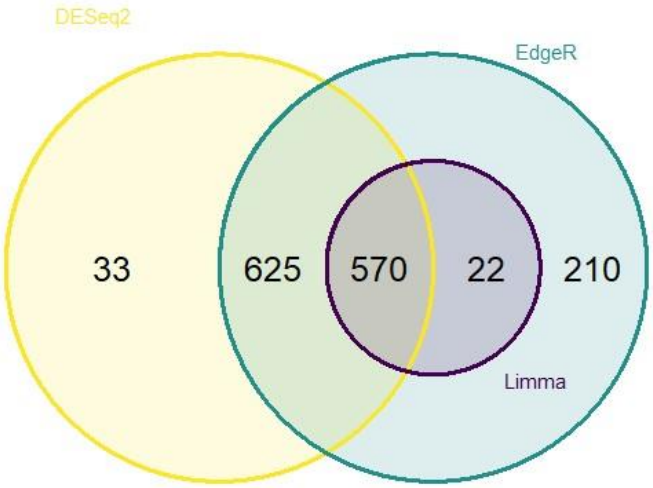
Wiele genów bo aż 690 jest wspólnych dla wszystkich metod. W dodatku widzimy, że edgeR i Limma mają ze sobą również wiele – 119 wspólnych genów – dużo więcej niż Deseq z pozostałymi pakietami. W dodatki widać że ogólnie edgeR wykrył największą liczbę genów.

Porównanie 8PTX:



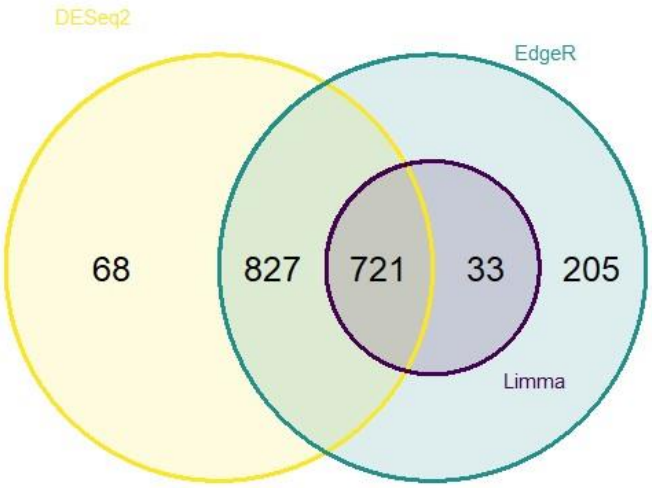
Tutaj ciekawy wynik – genów dla limmy jest dość mało i wszystkie zostały wykryte zarówno przez Deseq i EdgeR. Edge I deseq bardzo duża zgodność. EdgeR ponowanie największa liczba genów.

Porównanie 16PTX:



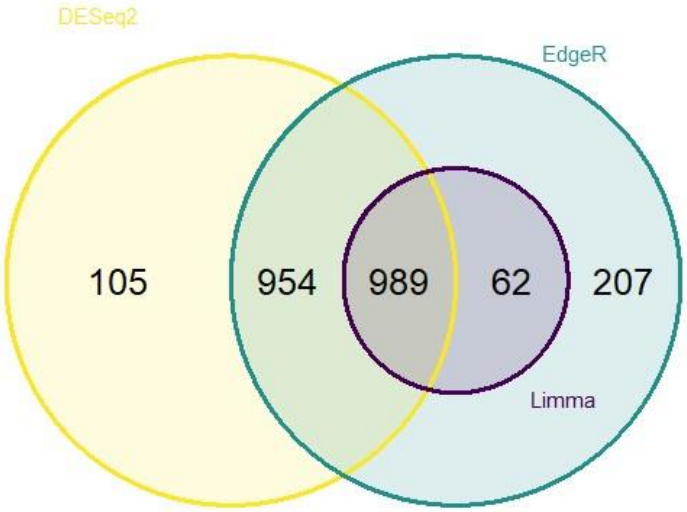
Duża zgodność między wszystkimi metodami – 570 genów. W ogólności widać już trend, że Limma jest często bardziej zgodna z EdgeR (poza wcześniejsztm porównaniem.)

Porównanie 32PTX:



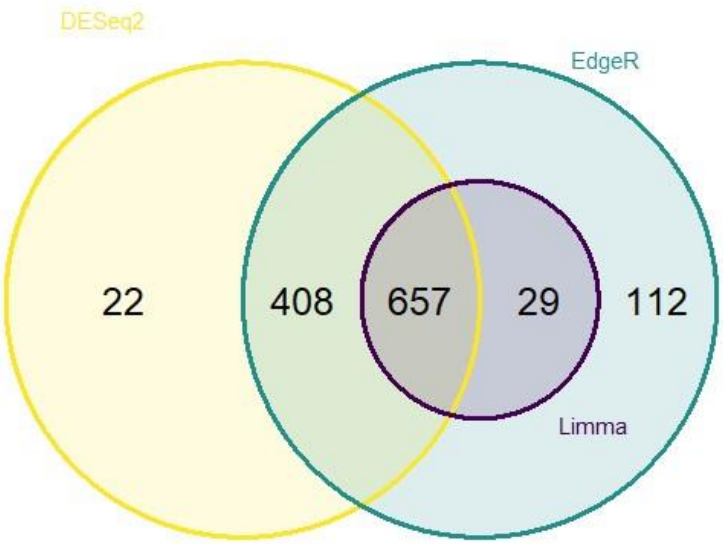
W ogólności wykryto trochę więcej genów niż w pozostałych porównaniach. Reszta obserwacji zostaje podtrzymana.

Porównanie 64PTX:



Jeszcze więcej genów i brak nowych obserwacji. Ten schemat jest obserwowany już 3 raz z rzędu.

Porównanie 128PTX:

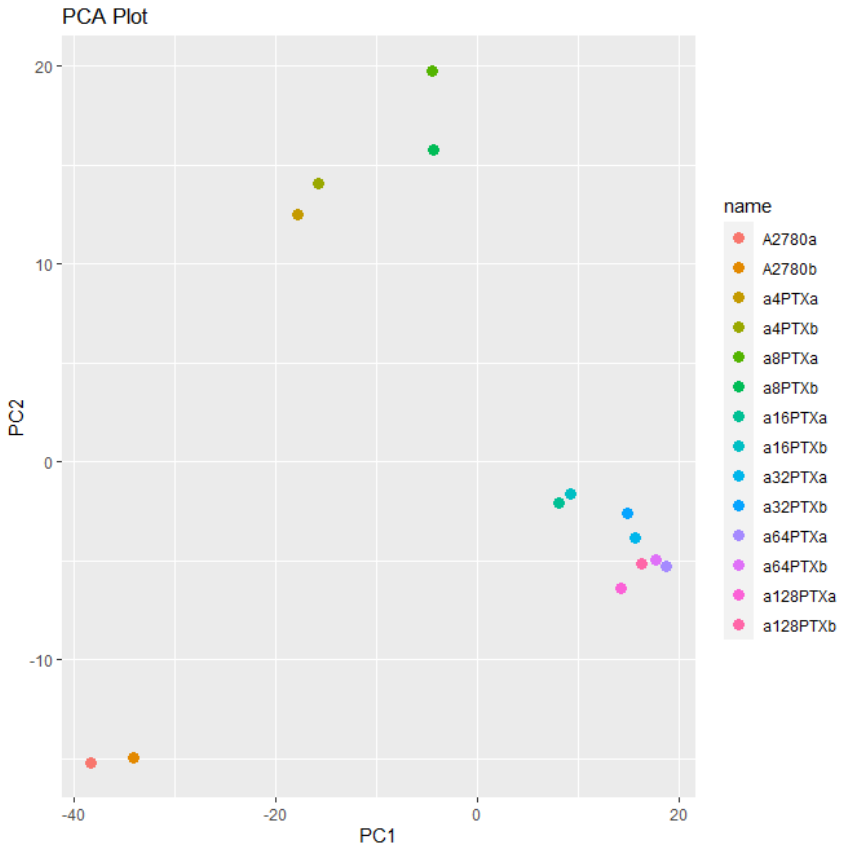


Kod do generowania diagramów:

```
library(VennDiagram)
library(tidyverse)


venn.diagram(
  x =  list(rownames(results[16]$limma_6), rownames(results[17]$edger_6), rownames(results[18]$deseq_6)),
  category.names = c("Limma" , "EdgeR" , "DESeq2"),
  filename = './venn.png',
  output = TRUE ,
  imagetype="png" ,
  height = 480 ,
  width = 480 ,
  resolution = 300,
  compression = "lzw",
  lwd = 1,
  col=c("#440154ff", '#21908dff', '#fde725ff'),
  fill = c(alpha("#440154ff",0.3), alpha('#21908dff',0.3), alpha('#fde725ff',0.3)),
  cex = 0.5,
  fontfamily = "sans",
  cat.cex = 0.3,
  cat.default.pos = "outer",
  cat.pos = c(-27, 27, 135),
  cat.dist = c(0.055, 0.055, 0.085),
  cat.fontfamily = "sans",
  cat.col = c("#440154ff", '#21908dff', '#fde725ff'),
  rotation = 1
)
```

b. Principal component analysis:

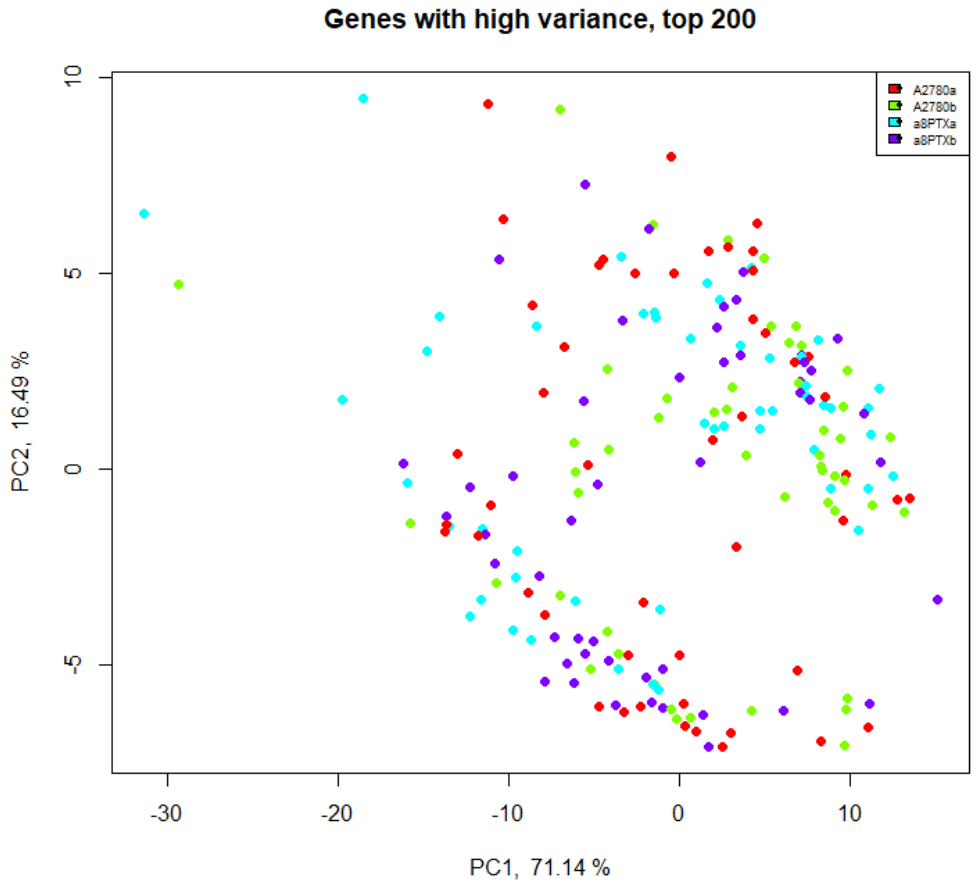
```
rld = rlog(dds)
ret = plotPCA(rld, ntop = 200, returnData=TRUE)


legend_order <- c("A2780a", "A2780b", "a4PTXa", "a4PTXb", "a8PTXa", "a8PTXb", "a16PTXa", "a16PTXb", "a32PTXa", "a32PTXb", "a64PTXa", "a64PTXb", "a128PTXa", "a128PTXb")

ggplot(ret, aes(x = PC1, y = PC2, color = name)) +
  geom_point(size = 3) +
  scale_color_discrete(limits = legend_order) +
  labs(title = "PCA Plot", x = "PC1", y = "PC2")
```



Widać na wykresie, że mamy 3 klastry wyznaczone na podstawie 2 składowych głównych w naszych danych (do analizy użyto całej tabeli zliczeń dla wszystkich próbek). Można zauważyć, że próbki kontrolne tworzą osobny klaster mocno oddalony od próbek traktowanych. W dodatku próbki traktowane tworzą dwa osobne klastry – jeden z nich o małych dawkach leku PTX (4 i 8) a drugi skupia pozostałe próbki traktowane większymi dawkami.

To jest klastrowanie próbek, natomiast można też poklastrować geny z danych próbek, poniżej klastrowanie genów (200 o największej wariancji) z porównania 8PTX:

```r
### PCA GEN0W ###
legend_order <- c("A2780a", "A2780b", "a8PTXa", "a8PTXb")

samples <- data.frame(sample=colnames(data.frame(count_metrices[2])), condition=factor(c('C', 'T', 'C', 'T')))
samples
dds <- DESeqDataSetFromMatrix(countData = data.frame(count_metrices[2]),
                              colData = samples,
                              design = ~ condition)

keep <- rowSums(counts(dds)) >= 10
dds <- dds[keep,]
dds <- DESeq(dds)

data_rld <- assay(rld)

v_log = matrix(nrow = length(data_rld[,1]), ncol = 1)
rownames(v_log) <- c(rownames(data_rld))

for (i in 1:length(data_rld[,1]))
{
  v_log[i] = var(data_rld[i,])
}

v_log_sort <- as.matrix(v_log[order(v_log,decreasing=TRUE),])

genes_with_var_log <- rownames(v_log_sort)
genes_with_var_log <- genes_with_var_log[1:200]

d1_normT_var_log <- data_rld[genes_with_var_log,]
pca <- prcomp(d1_normT_var_log)
project.pca.proportionvariances <- ((pca$sdev^2) / (sum(pca$sdev^2)))*100


auto_colors <- rainbow(length(legend_order))

# Plot using automatic colors
plot(x = pca$x[, 1], y = pca$x[, 2], col = auto_colors, pch = 16, main = "Genes with high variance, top 200",
     xlab = paste("PC1, ", round(project.pca.proportionvariances[1], 2), "%"),
     ylab = paste("PC2, ", round(project.pca.proportionvariances[2], 2), "%"))

# Add legend with automatic colors
legend(x = "topright", legend_order, fill = auto_colors, pch = 16, cex = 0.6, bg = "white")
```



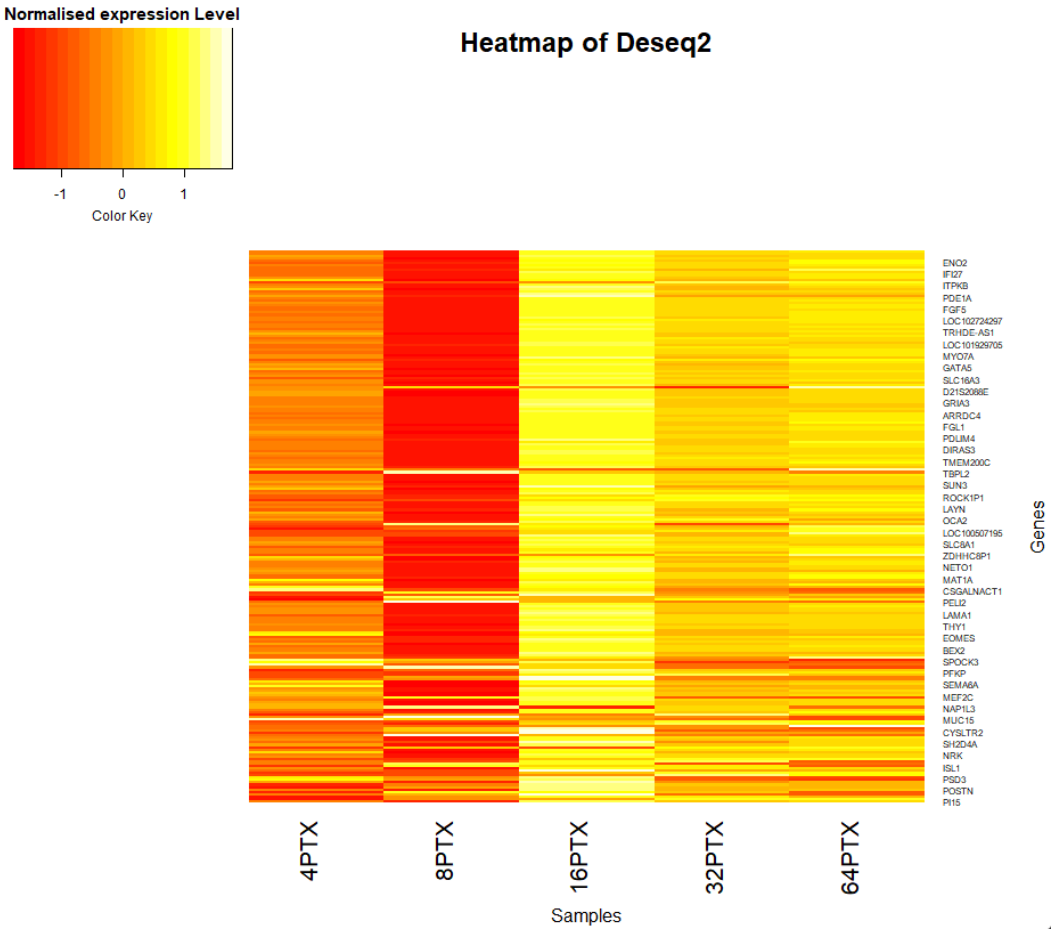
Niestety ciężko tu wysnuć jakieś wnioski, raczej nie ma podziału na klastry.

c. Mapa ciepła

Mapa ciepła dla kilku próbek PTX z zastosowania metody Deseq2.

Liczba wspólnych genów ulegających różnicowej ekspresji (adj. Pval < 0.01) w próbkach 4,8,16,32 i 64 PTX to 235.

```
res_des_1 <- results[3]$deseq_1
res_des_2 <- results[6]$deseq_2
res_des_3 <- results[9]$deseq_3
res_des_4 <- results[12]$deseq_4
res_des_5 <- results[15]$deseq_5

row_names_1 <- rownames(res_des_1)
row_names_2 <- rownames(res_des_2)
row_names_3 <- rownames(res_des_3)
row_names_4 <- rownames(res_des_4)
row_names_5 <- rownames(res_des_5)

common_row_names <- Reduce(intersect, list(row_names_1, row_names_2, row_names_3, row_names_4, row_names_5))

res_des_1 <- res_des_1[common_row_names, ]
res_des_2 <- res_des_2[common_row_names, ]
res_des_3 <- res_des_3[common_row_names, ]
res_des_4 <- res_des_4[common_row_names, ]
res_des_5 <- res_des_5[common_row_names, ]

common_matrix <- cbind(res_des_1[, 'baseMean'], res_des_2[, 'baseMean'], res_des_3[, 'baseMean'], res_des_4[, 'baseMean'], res_des_5[, 'baseMean'])

rownames(common_matrix) <- common_row_names
colnames(common_matrix)<- (c('4PTX', '8PTX', '16PTX', '32PTX', '64PTX'))

normalized_matrix <- scale(common_matrix, center = FALSE, scale = apply(common_matrix, 2, max) - apply(common_matrix, 2, min))


heatmap.2(common_matrix, col = heat.colors(20), scale = "row",
          main = "Heatmap of Deseq2",
          xlab = "Samples", ylab = "Genes",
          margins = c(7, 7),
          trace = "none",  # Turn off trace lines
          density.info = "none",  # Turn off density plot
          dendrogram = "none",  # Turn off dendrogram
          key = TRUE, keysize = 1.5, key.title = "Normalised expression Level",
          key.xlab = "Color Key",
          Colv = FALSE,)
```



Powyższą heatmapę należałoby interpretować tak, że przy dawce 4PTX zmiana ekpresji jest najmniejsza, przy dawce 8PTX następuje bardzo duża regulacja negatywna ekspresji genów, przy dawce 16PTX mamy olbrzymi przeskok do bardzo silnej nadekpresji, później przy dawce 32PTX znowu przeskok do raczej delikatnej regulacji dodatniej, i przy dawce 64PTX znowu silniejsza regulacja dodatnia.
Jest to dość dziwna zależnośc.
*oczywiście po prawej stronie wykresu nie mieszczą się wszystkie 235 genów, bo wykres jest za mały, więc nie jest to zbyt informatywne, ale można podejrzeć przynajmniej część nazw genów z tego zbioru.

d. Correlation plot:
Za pomocą poniższego kodu dla wyników DESEQ2:

```
res_des_1 <- results[3]$deseq_1
res_des_2 <- results[6]$deseq_2
res_des_3 <- results[9]$deseq_3
res_des_4 <- results[12]$deseq_4
res_des_5 <- results[15]$deseq_5
res_des_6 <- results[18]$deseq_6

row_names_1 <- rownames(res_des_1)
row_names_2 <- rownames(res_des_2)
row_names_3 <- rownames(res_des_3)
row_names_4 <- rownames(res_des_4)
row_names_5 <- rownames(res_des_5)
row_names_6 <- rownames(res_des_6)


## pod cor

common_row_names <- Reduce(intersect, list(row_names_5, row_names_6))
res_des_1 <- res_des_5[common_row_names, ]
res_des_2 <- res_des_6[common_row_names, ]
res_des_1 <- na.omit(res_des_1)
res_des_2 <- na.omit(res_des_2)

negative_des1 <- res_des_1$log2FoldChange < 0
negative_des2 <- res_des_2$log2FoldChange < 0
common_negative_ids <- res_des_1$log2FoldChange[negative_des1 & negative_des2]
length(common_negative_ids)

positive_des1 <- res_des_1$log2FoldChange > 0
positive_des2 <- res_des_2$log2FoldChange > 0
common_positive_ids <- res_des_1$log2FoldChange[positive_des1 & positive_des2]
length(common_positive_ids)
```

Stworzono tabelę w excelu:

A	B	C	D	E	F	G
sample	A/4PTX	A/8PTX	A/16PTX	A/32PTX	A/64PTX	A/128PTX
A/4PTX	1	118	114	103	105	94
A/8PTX	311	1	98	81	79	65
A/16PTX	279	264	1	379	402	385
A/32PTX	262	297	379	1	657	361
A/64PTX	250	279	364	714	1	442
A/128PTX	249	281	352	402	417	1

Niestety mam problemy ze stworzeniem wykresu corrplot.