Homework4

Carlotta Porcelli - exam_ID 62

6/13/2017

PART 1

Question 1A: Merge the ChIP peaks that overlap over 1bp or more. How many merged regions are produced compared to how many peaks you started with?

In order to run the *bedtools merge*, the *txnCHIP.bed* file needs to be sorted. This is done by chromosome and then by start position.

```
sort -k1,1 -k2,2n txnChIP.bed > sorted_txnCHIP.bed # presort of .bed file
```

```
wc -l txnChIP.bed
```

The length of the *txnCHIP.bed* file is: 4380444.

Merging the *txnCHIP.bed* file is done not only merging the intervals but also reporting the number of intervals that were integrated into the new file using the **-c 1** (applying *option* on first column) and **-o count** parameters.

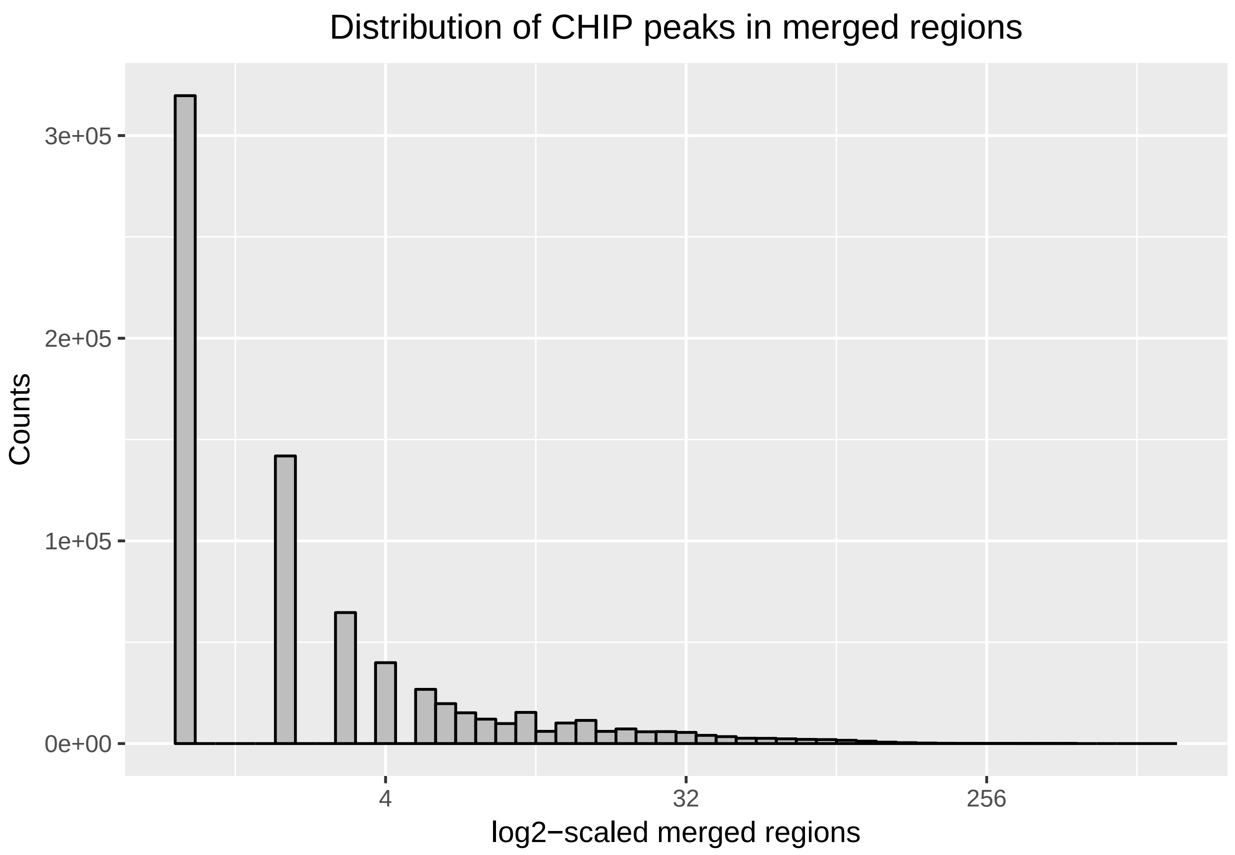
```
bedtools merge -i sorted_txnCHIP.bed -c 1 -o count > count_merged_sorted_txnCHIP.bed
```

```
wc -l count_merged_sorted_txnCHIP.bed
```

The merge tool produced 746610 regions, with a difference of 3633834 regions compared to the initial peaks number.

Question 1B: Plot the distribution of ChIP peaks in each merged region. Briefly comment your plot.

```
library("ggplot2")
CHIP_peaks <- read.table("count_merged_sorted_txnCHIP.bed", header = F)
CHIP_distribution <- ggplot(data=CHIP_peaks, aes(V4)) +
            geom_bar(stat="bin", color='black', fill='grey', binwidth = 0.2) +
            scale_x_continuous(trans='log2') +
            xlab("log2-scaled merged regions") +
            ylab("Counts") +
            ggtitle("Distribution of CHIP peaks in merged regions") +
            theme(plot.title = element_text(hjust = 0.5))
CHIP_distribution
```



The plot shows that more than 3e+05 of the merged regions are composed of only one site. The number of merged regions decrease drastically with the increase of overlapping regions.

Question 1C: Using R, produce a new BED file that contains the 10 merged regions having the highest number of ChIP peaks.

```r
top_10 <- as.data.frame(CHIP_peaks[order(-CHIP_peaks$V4),][0:10,])
write.table(top_10, file='top_10_CHIP_peaks.bed', quote = F, row.names = F, col.names = F)
```

Question 1D: Upload this into the UCSC browser. Look at each merged region and try to interpret it, also taking the peaks it contains into account. What do the merged regions typically overlap? Is there a particular factor that is responsible for the clusters?

Once the top_10 CHIP peaks BED file has been uploaded on the UCSC browser, the track has been mapped to the hg19 assembly. [Screenshots of the 10 merged regions with the highest number of ChIP peaks]

All of the merged regions overlap the RNA polymerase II subunit **POLR2A**. The **POLR2A** forms the largest subunit of RNA polymerase II, enzyme responsible for synthesizing messenger RNA in eukaryotes. In addition, this subunit forms the DNA binding domain of the polymerase, a groove in which the DNA template is transcribed into RNA.Reference:. The transcription factors binding sites are clustered because of the presence of **POLR2A** and its repeatedly presence.

Question 1E: What could we have done to improve the analysis?

To improve the analysis the merge function could be set to combine regions that overlap for more than one basepair. This constraint would produce less overlapping regions.

PART 2

Question 2.1: Provide one line of code which will make symbolic links only to the 6 fastq files.

```
ln -s /home/bohta/HW4/part2/*.fastq /home/qbp693/
```

Question 2.2: Check how well the sequencing run went - Use the 'wc' function to calculate the number of reads in all fastq files and comment on the results.

```
grep '^+$' -c *.fastq # counts the number of lines starting and ending with '+' sign
```

The count of reads for each file is: WT1_R1.fastq:45626717, WT1_R2.fastq:45626717, WT2_R1.fastq:41428670, WT2_R2.fastq:41428670, WT3_R1.fastq:19326183, WT3_R2.fastq:19326183 It is noticeable that there is no difference in number of reads between the strands R1 and R2 of the same library. This is because the two files come from the same cDNA sequence and they have probably been trimmed before removing the orphan reads.

Question 2.3:

A) Report the command for running Kallisto on the WT1 RNA-seq data.

```
nice /home/bohta/bin/kallisto quant --index /home/bohta/HW4/part2/kallistoIndex
    --fr-stranded -t 6 --plaintext --bias -o kallisto_out/ WT1_R1.fastq WT1_R2.fastq
```

B) Report the number of pseudo-aligned reads.

The number of pseudo-aligned reads is 23720001.

C*) Report the estimated average fragment length. Based on this result, what is then the distance between the 3' ends of the two reads in an average read pair? The estimated average fragment length is 178.486.

```
cat WT1_R1.fastq | awk '{if(NR%4==2) print length($1)}' > input_readslength_R1.txt
cat WT1_R2.fastq | awk '{if(NR%4==2) print length($1)}' > input_readslength_R2.txt
```

```
read_len_R1 <- read.table('input_readslength_R1.txt', header=F) # R1 reads length
read_len_R2 <- read.table('input_readslength_R2.txt', header=F) # R2 reads length
r1_mean <- mean(read_len_R1$V1) # average length of R1 reads
r2_mean <- mean(read_len_R2$V1) # average length of R2 reads
avg_fragment_length <- 178.486
abs(avg_fragment_length - (r1_mean+r2_mean)) # computes the absolute distance between 3' ends
```

The resulting fragment from the computation is a transcript compatible with the mapped reads, this means that the distance between the two 3' ends of a read pair can be computed as the absolute difference between the average_fragment_length and the average length of the reads in R1 and R2. This results in an average distance between the 3' ends of an average read pair of almost 18 bases.

Question 2.4:

A*) Report the command for running Salmon on the WT1 RNA-seq data.

```
nice /home/bohta/bin/salmon quant --index /home/bohta/HW4/part2/salmonIndex -p 6
  --libType A --seqBias --gcBias -1 WT1_R1.fastq -2 WT1_R2.fastq -o salmon_out/
```

B) Report the most likely library type as identify by Salmon.

The automatically detected most likely library is ISF.

C) Report mapping rate.

The mapping rate = 60.9567%.

Question 2.5: Which tool aligned more reads? Comment on the result.

Salmon aligned 27812554 reads, with almost 61% of mapped reads whereas Kallisto mapped almost 52% . Kallisto is based on pseudoalignments to *compatible transcripts*, Salmon is based on lightweight alignments, chains of maximal and super maximal exact matches. Both of the tools mapped above 50% which can be considerated as a good result but surely Salmon performed better.

Question 2.6:

A) Compare the estimated 'effective length' of the isoform 'TCONS_00000020' from Kallisto and Salmon to each other and the reference length.

```
kallisto_quant <- read.table('abundance.tsv', header=T)
salmon_quant <- read.table('quant.sf', header=T)
e_len_kal <- kallisto_quant[kallisto_quant$target_id=='TCONS_00000020',]
e_len_salmon <- salmon_quant[salmon_quant$Name == 'TCONS_00000020',]
e_len_kal
```

```
##          target_id length eff_length est_counts tpm
## 41 TCONS_00000020   4456    4828.72          0   0
```

```
e_len_salmon
```

```
##               Name Length EffectiveLength TPM NumReads
## 41 TCONS_00000020   4456         3842.17   0        0
```

B) What could explain the difference in the effective length? Which estimate do you trust more?

The effective lengths will be affected by the estimated empirical fragment length distribution, the method of calculating effective lengths and whether or not bias correction is used. The most trustworthy is the Salmon computation because it corrects not only the for sequence-specific biases but also for the fragment-level GC biases. Moreover it is unlikely that the effective length is larger than the actual length as Kallisto shows.

PART 3

Question 3.1: Load the data into R. How many isoforms are quantified in the count data?

```
part3_data <- load('part3.Rdata')
nrow(countDF)
```

```
## [1] 5787
```

The number of isoforms quantified in the count data is: 5787.

Question 3.2: Report the R code for how to calculate RPKM values.

```
library_size <- colSums(countDF) # number of reads mapped
transcript_lenghts <- c(annotationDF$length) # vector of lengths of transcripts
# calculation of RPKM values
Rpkm_values <-  as.data.frame(t(t(countDF) * 1000000 / library_size) *
                                as.vector(1000 / transcript_lenghts))
head(Rpkm_values, 2)
```

```
##                K_WT1_Counts K_WT2_Counts K_WT3_Counts S_WT1_Counts
## TCONS_00003947     5.812518     3.087735    7.9140098     5.577944
## TCONS_00003950     2.403414     3.902034    0.8945008     1.156377
##                S_WT2_Counts S_WT3_Counts
## TCONS_00003947     2.249663   7.43761890
## TCONS_00003950     3.452771   0.03223709
```

Question 3.3: Make a one-liner (without the use of ';') that outputs the mean RPKM value of each sample. The restrictions from the previous question no longer apply.

```
mean_values <- cbind(rowMeans(Rpkm_values[,0:3]), rowMeans(Rpkm_values[,4:6]))
colnames(mean_values)<- c('K samples means', 'S samples means')
head(mean_values, 3)
```

```
##                K samples means S samples means
## TCONS_00003947        5.604754        5.088409
## TCONS_00003950        2.399983        1.547128
## TCONS_00000007       24.766350       22.186157
```

Question 3.4: Make histograms of the distribution of the logRpkm expression values.

```
library("ggplot2")
trans_log <- data.frame(t(logRpkmDF))
# trans_log$tool_label <-
#
# logRpkm_plot <- ggplot(data=logRpkmDF, aes(logRpkmDF[])) +
#             geom_histogram(stat="bin", color='black', fill='grey', binwidth = 0.2) +
#             xlab("log10-transformed filtered RPKM") +
#             ylab("Frequency") +
#             ggtitle("Distribution of CHIP peaks in merged regions") +
#             theme(plot.title = element_text(hjust = 0.5))
#     #         scale_fill_manual(values =
#     # c("K_WT1_RPKM", 'K_WT2_RPKM', 'K_WT3_RPKM' = "darkblue", "S_WT1_RPKM", 'S_WT1_RPKM', 'S_WT1_RPK
#     #         facet_wrap(~KnockDownTarget)
# logRpkm_distribution
```

Question 3.5: For each tool use logRpkm to calculate all pairwise replicate Pearson correlations of the replicate expression values and report the numbers in a table (one table per tool).

Question 3.6:

A) Construct a function which calculates the CV and use the apply() function to calculate the CV based on logRpkm values for Kallisto and Salmon (separately). 1p

B) Plot the distribution of CV values as density lines (in one single plot) using color to indicate the tool and log transform the x-axis (using log10). 1p

C) Comment on the CV plots using max 75 words. 2p

Question 3.7: Based on all the results you have collected here (all of part 2 and all of part 3), discuss in max 100 words which tool would you choose to continue with if you wanted to make a differential expression analysis.

PART 4

```
set.seed(2017)
library(ggplot2)
library(GGally)
```

Question 1: Read both the expression matrix ("ExpressionMatrix.tab") and study design ("StudyDesign.tab") into R. Report the number of samples and the number of genes.

```
expression_m <- read.table('ExpressionMatrix.tab', header=T) # data import
study_design <- read.table('StudyDesign.tab', header=T) # data import
```

```
samples_number <- ncol(expression_m)
samples_number
```

```
## [1] 75
```

```
gene_number <- nrow(expression_m)
gene_number
```

```
## [1] 10000
```

In the data set there are 75 samples and 10000 genes.

Question 2: Perform PCA on the samples and report the amount of variance contained in the first 5 principle components.

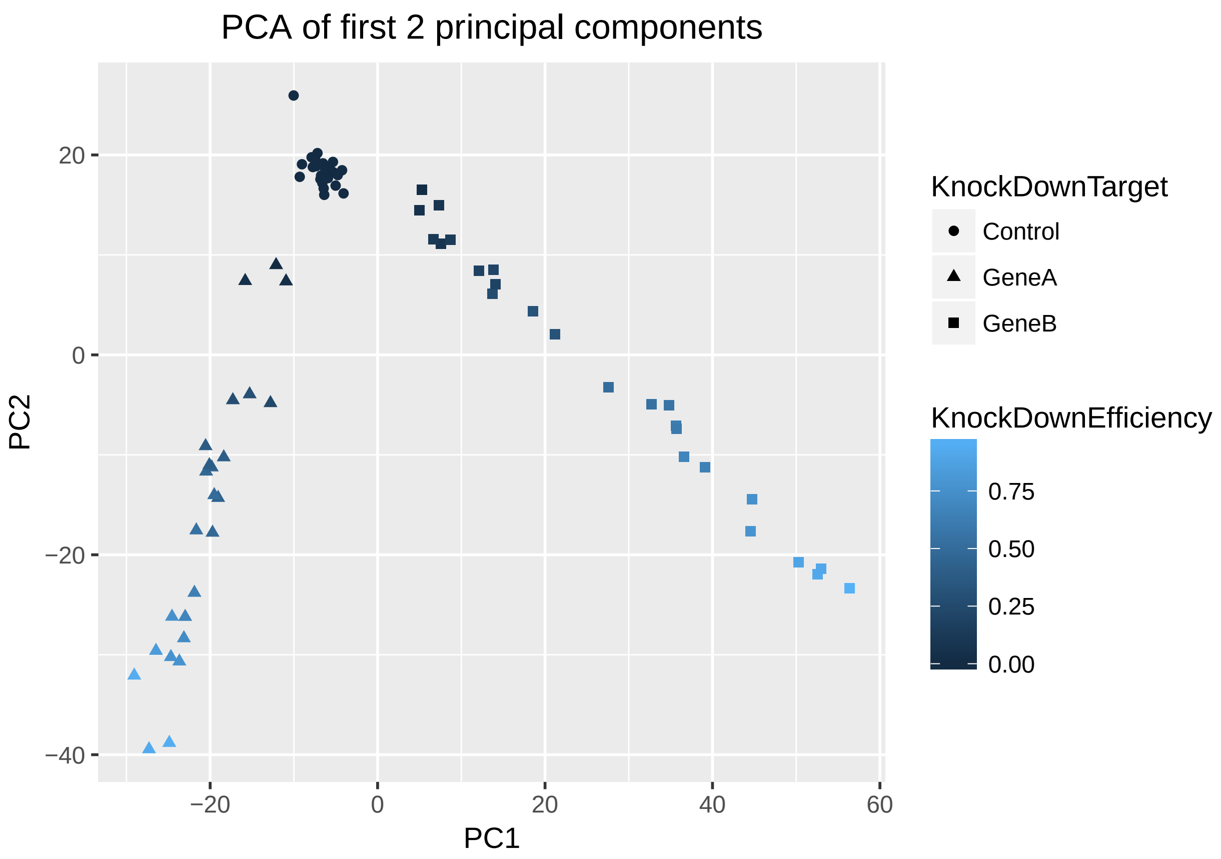
```
expression_m_transposed <- t(expression_m) # transposition of the matrix
pca_genes <- prcomp(expression_m_transposed, center = T, scale. = T) # pca
summary(pca_genes)
```

```
## Importance of components:
##                             PC1      PC2      PC3      PC4      PC5
## Standard deviation     22.56522 18.11428 14.48096 13.69861 12.35100
## Proportion of Variance  0.05092  0.03281  0.02097  0.01877  0.01525
## Cumulative Proportion   0.05092  0.08373  0.10470  0.12347  0.13872
##                             PC6      PC7      PC8      PC9     PC10
## Standard deviation     12.20201 12.00511 11.95412 11.93596 11.89816
## Proportion of Variance  0.01489  0.01441  0.01429  0.01425  0.01416
## Cumulative Proportion   0.15361  0.16802  0.18231  0.19656  0.21072
##                            PC11     PC12     PC13     PC14     PC15
## Standard deviation     11.85152 11.80436 11.78726 11.77023 11.71214
## Proportion of Variance  0.01405  0.01393  0.01389  0.01385  0.01372
## Cumulative Proportion   0.22476  0.23870  0.25259  0.26644  0.28016
##                            PC16     PC17     PC18     PC19    PC20
## Standard deviation     11.67953 11.64034 11.62490 11.59931 11.5779
## Proportion of Variance  0.01364  0.01355  0.01351  0.01345  0.0134
## Cumulative Proportion   0.29380  0.30735  0.32087  0.33432  0.3477
##                            PC21     PC22     PC23     PC24     PC25
## Standard deviation     11.56378 11.54035 11.52656 11.51621 11.47243
## Proportion of Variance  0.01337  0.01332  0.01329  0.01326  0.01316
## Cumulative Proportion   0.36110  0.37442  0.38770  0.40096  0.41413
##                           PC26     PC27     PC28     PC29     PC30
## Standard deviation     11.4468 11.41536 11.40649 11.38546 11.36297
## Proportion of Variance  0.0131  0.01303  0.01301  0.01296  0.01291
## Cumulative Proportion   0.4272  0.44026  0.45327  0.46623  0.47914
##                            PC31     PC32     PC33     PC34     PC35
## Standard deviation     11.33459 11.31782 11.30687 11.28550 11.24722
## Proportion of Variance  0.01285  0.01281  0.01278  0.01274  0.01265
## Cumulative Proportion   0.49199  0.50480  0.51759  0.53032  0.54297
##                            PC36     PC37     PC38    PC39     PC40
## Standard deviation     11.22168 11.20169 11.18680 11.1790 11.16086
## Proportion of Variance  0.01259  0.01255  0.01251  0.0125  0.01246
## Cumulative Proportion   0.55556  0.56811  0.58063  0.5931  0.60558
##                            PC41     PC42     PC43     PC44     PC45
## Standard deviation     11.14527 11.12300 11.10767 11.08247 11.06800
## Proportion of Variance  0.01242  0.01237  0.01234  0.01228  0.01225
## Cumulative Proportion   0.61800  0.63037  0.64271  0.65499  0.66724
##                            PC46     PC47     PC48     PC49     PC50
## Standard deviation     11.06574 11.02358 11.00717 10.97090 10.94381
## Proportion of Variance  0.01225  0.01215  0.01212  0.01204  0.01198
## Cumulative Proportion   0.67949  0.69164  0.70376  0.71579  0.72777
##                            PC51     PC52     PC53     PC54     PC55
## Standard deviation     10.92636 10.91780 10.89834 10.85267 10.83593
## Proportion of Variance  0.01194  0.01192  0.01188  0.01178  0.01174
## Cumulative Proportion   0.73971  0.75163  0.76351  0.77528  0.78703
##                            PC56     PC57     PC58     PC59     PC60
## Standard deviation     10.80048 10.78061 10.75729 10.74325 10.72811
## Proportion of Variance  0.01167  0.01162  0.01157  0.01154  0.01151
## Cumulative Proportion   0.79869  0.81031  0.82188  0.83343  0.84494
##                            PC61     PC62     PC63     PC64     PC65
## Standard deviation     10.71760 10.69939 10.63827 10.63368 10.59586
## Proportion of Variance  0.01149  0.01145  0.01132  0.01131  0.01123
## Cumulative Proportion   0.85642  0.86787  0.87919  0.89049  0.90172
##                           PC66     PC67     PC68     PC69     PC70
## Standard deviation     10.5824 10.52937 10.49488 10.48517 10.47995
## Proportion of Variance  0.0112  0.01109  0.01101  0.01099  0.01098
## Cumulative Proportion   0.9129  0.92401  0.93502  0.94602  0.95700
##                            PC71     PC72     PC73     PC74      PC75
## Standard deviation     10.44465 10.42134 10.33458 10.27205 3.444e-14
## Proportion of Variance  0.01091  0.01086  0.01068  0.01055 0.000e+00
## Cumulative Proportion   0.96791  0.97877  0.98945  1.00000 1.000e+00
```

The expression values have been scaled using center and scale arguments set to TRUE in order to normalize the data. The amount of variance in the first 5 principal components is: PC1:0.05092, PC2:0.03281, PC3:0.02097, PC4:0.01877, PC5:0.01525.

Question 3: Make a plot of PC1 vs PC2, where knock down efficiency is indicated by color and knock down target is indicated by shape. Comment on the plot.

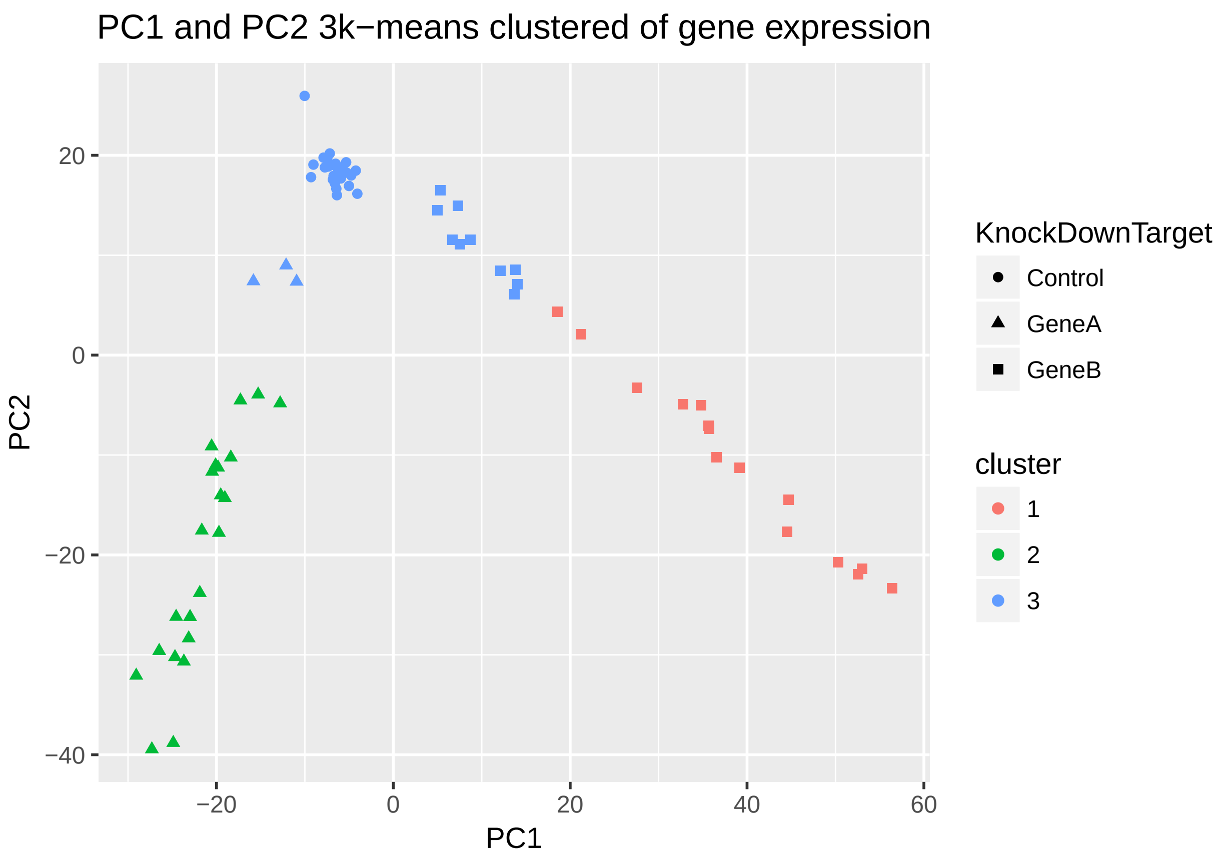
```r
library(ggplot2)
library(GGally)
gene_plot <- data.frame(pca_genes$x, study_design)
ggplot(data=gene_plot, aes(PC1, PC2, color=KnockDownEfficiency, shape = KnockDownTarget)) +
  geom_point(size=1.5)+
  ggtitle('PCA of first 2 principal components') +
  theme(plot.title = element_text(hjust = 0.5))
```



The plot shows a clearly separated cluster of control genes with knock down efficiency equals to zero. The majority of Gene A samples seem to have a knock down efficiency between 0.50 and 1 on the other end there are three outliers closer to the control cluster with efficiency closer to zero. The samples from the Gene B are well clustered on the right side of the control targets and they span the entire range of efficiency values.

Question 4: Perform a K-means clustering of the data, using k=3 and 10 random starting points. Visualize the clustering by making a plot of PC1 vs PC2, where the clusters are indicated by color. Briefly comment on how the clustering corresponds to the known knockdown targets.

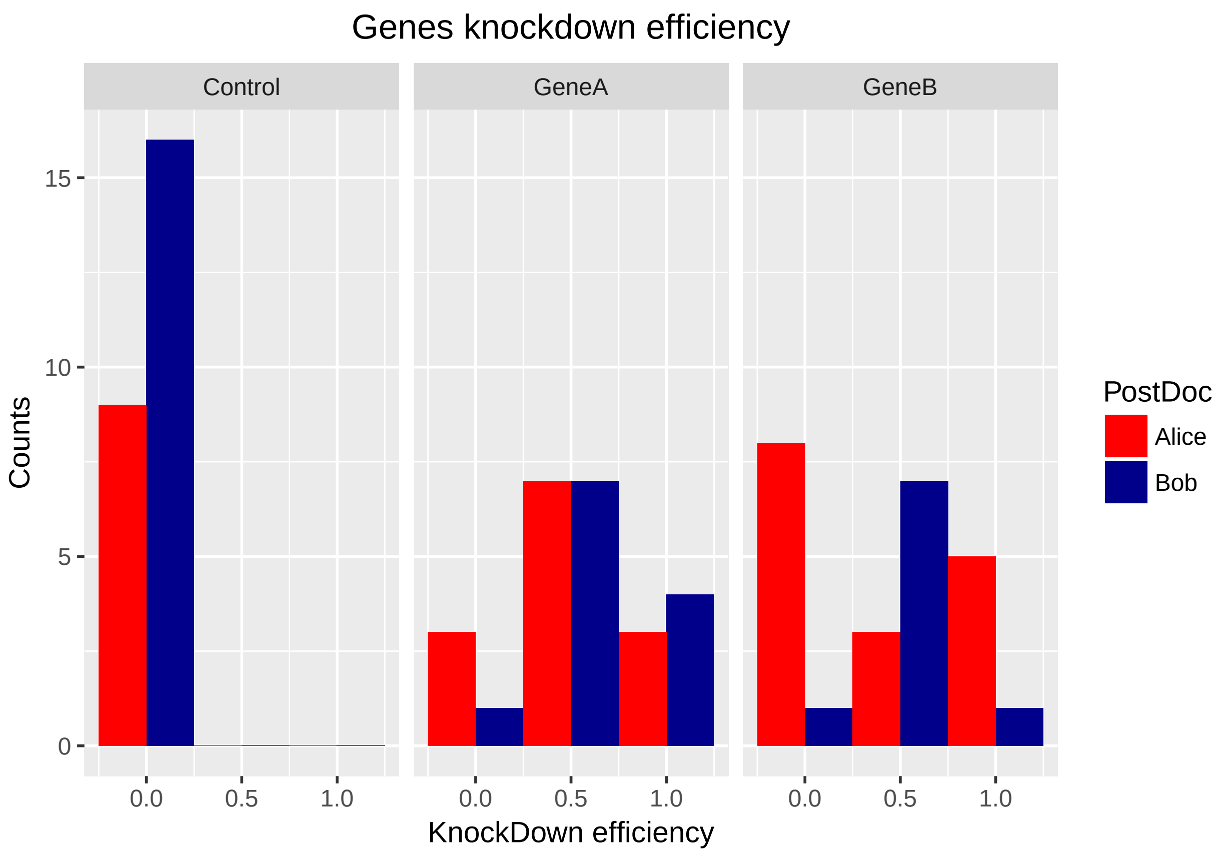
```r
k3_means <- kmeans(expression_m_transposed, centers=3, nstart = 10) # 3-means with 10 starting points
gene_plot$cluster <- factor(k3_means$cluster) # adding clusters to data frame
# plotting results
k_means_plot <- ggplot(data=gene_plot, aes(x=PC1, y=PC2, color=cluster, shape = KnockDownTarget)) +
  geom_point(size=1.5) +
  ggtitle('PC1 and PC2 3k-means clustered of gene expression') +
  theme(plot.title = element_text(hjust = 0.5))
k_means_plot
```



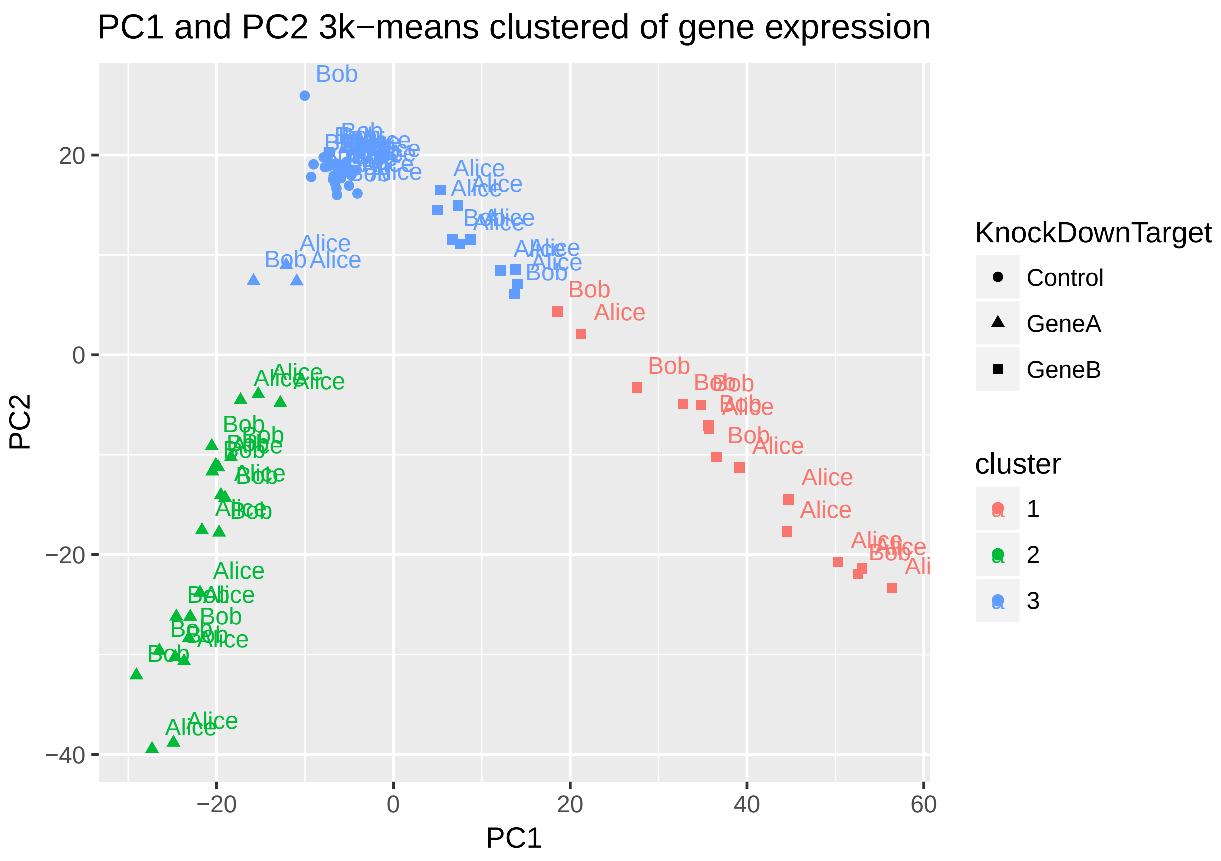
The plot shows that the middle cluster contains datapoints with knock down efficiency closer to zero. This cluster should contain only control samples so it is evident that something in the experiment didn't work as expected. The cluster on the right side groups all the samples of Gene B and the other cluster on the left contains all the samples from Gene A with higher knock down efficiencies.

Question 5: Using a maximum number of 3 plots, investigate whether there is any truth to the postdoc allegations: Can you see a difference in samples prepared by Alice and Bob? Is there any indication Bob has ruined a sample? Discuss you results using a maximum of 75 words.

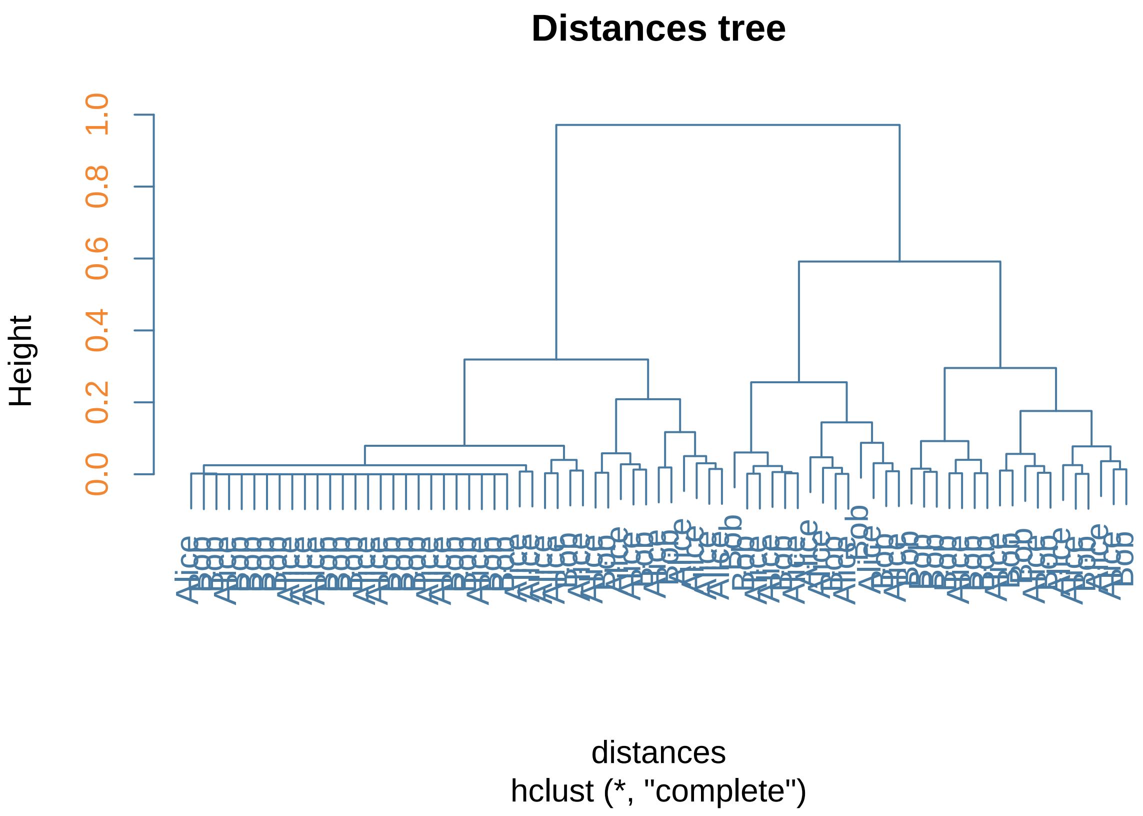
```
KD_effiency <- ggplot(data=study_design,
            aes(KnockDownEfficiency, fill=PostDoc)) +
            geom_bar(stat="bin", binwidth = 0.5, position=position_dodge()) +
            scale_fill_manual(values = c("Bob" = "darkblue", "Alice" = "red"))+
            xlab("KnockDown efficiency") +
            ylab("Counts") +
            ggtitle("Genes knockdown efficiency") +
            theme(plot.title = element_text(hjust = 0.5))+
            facet_wrap(~KnockDownTarget)
KD_effiency
```



```
k_means_plot + geom_text(aes(label=PostDoc), size=3, hjust=-0.25, vjust=-0.8)
```



```r
distances <- dist(study_design$KnockDownEfficiency)
distance_tree <- hclust(distances)
plot(distance_tree, label=study_design$PostDoc,
     col = "#487AA1", col.axis = "#F38630",
     main='Distances tree')
```



Question 6: Based on all you observations in Question 1-5, discuss whether the experiment is still useful, or whether the postdocs have ruined it. Use a maximum of 100 words. 3p

PART 5

Question 1: Read the DE analysis ('DifferentialExpression.tab) results into R, and show the first few lines of the file.

```
differential_expression <- read.table('DifferentialExpression.tab', header=T)
head(differential_expression, 3)
```

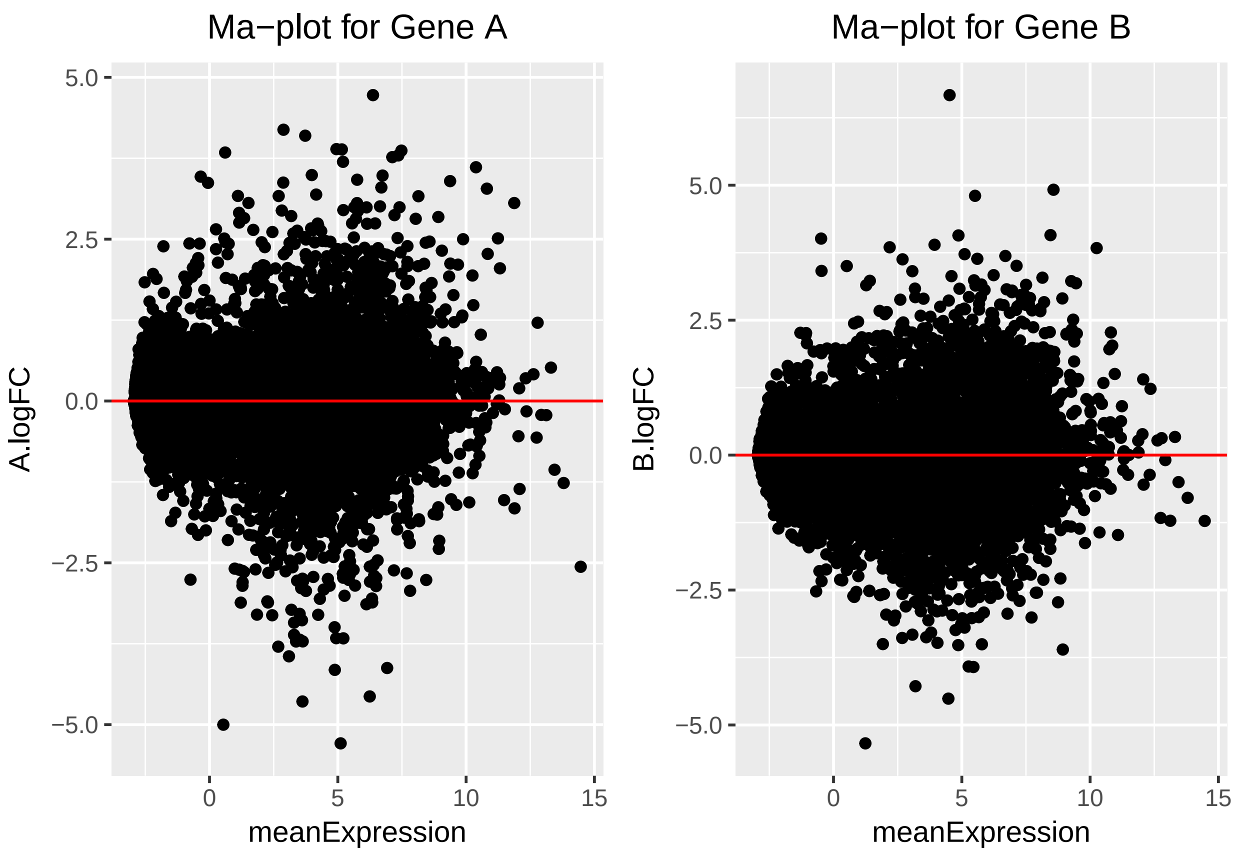
```
##    gene meanExpression     A.logFC   A.pvalue     A.FDR      B.logFC
## 1 Gene1     -2.5530584  0.6113569 0.0624194 0.3605973 -0.06796082
## 2 Gene2     10.8615216  0.1950991 0.7051621 0.9458926  2.02880013
## 3 Gene3      0.4656058 -0.4424548 0.3237347 0.7569917 -0.49773125
##        B.pvalue      B.FDR
## 1 8.250010e-01 0.94580106
## 2 7.077611e-05 0.00132045
## 3 2.411661e-01 0.59217398
```

Question 2: As part of a single plot, produce two MA-plots (one for each knock-down). To mitigate overplotting, points on the plot should be transparent

```r
library(ggplot2)
library(GGally)
library(gridExtra)
ma_plot <- ggplot(data = differential_expression) +
  geom_point(aes(meanExpression, A.logFC), size=1.5) +
  geom_hline(yintercept = 0, col='red')+
  ggtitle("Ma-plot for Gene A") +
  theme(plot.title = element_text(hjust = 0.5))

mb_plot <- ggplot(data = differential_expression) +
  geom_point(aes(meanExpression, B.logFC), size=1.5) +
  geom_hline(yintercept = 0, col='red')+
  ggtitle("Ma-plot for Gene B") +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(ma_plot, mb_plot, nrow=1, ncol=2)
```



Question 3: Explain how an MA-plot can be used to investigate whether expression data has been properly normalized (max 75 words).

Question 4: Report the number of upregulated and downregulated genes that are significantly DE after correction for multiple testing (at alpha=0.05). Which knockdown target has the highest total number of DE genes?

Question 5: Plot the two sets of logFCs (log2 fold changes) against each other. Color points based on whether they are significantly DE after multiple testing correction (at alpha=0.05) in either one or both of the knockdowns. 3p

Question 6: Calculate the Pearson correlation between the logFCs of the two knockdowns, and test whether this correlation is significant. 2p

Question 7: Caroline asks you to construct a 2-by-2 contingency table showing whether genes are significantly DE (after correcting for multiple testing, alpha=0.05) in the two knockdowns. Perform a Fisher???s Exact test for any association between the DE genes in the two knockdowns. 3p

Question 8: Based on Question 4-7, do you think the geneA and geneB transcription factors regulate the same genes? Use a maximum of 100 words.