HW3: Transcriptome Analysis

##Homework 3 - to be done as groups **Names**: Mahdi Robbani, Ciara Pugh, Oline Jensen, Michael Widdowson, Niels Lorenzen

Group: 7

##Homework instructions

For deadlines etc., see Absalon.

You have to supply both the answer (whatever it is: numbers, a table, plots or combinations thereof), as well as the R or Linux code you used to make the plots. This should be done using this R markdown template: we want both the R markdown file and a resulting PDF. For PDF output, you may have to install some extra programs - RStudio will tell you.

Note that:

1. If the R code gives different results than your results, you will get severe point reductions or even 0 points for the exercise
2. Some questions may request you to use R options we have not covered explicitly in the course: this is part of the challenge
3. While this is a group work, we expect that everyone in the group will have understood the group solution: similar or harder question might show up in the individual homework. So, if something is hard, it means you need to spend more time on it
4. The results should be presented on a level of detail that someone else could replicate the analysis.

For statistical tests, you have to:

1. Motivate the choice of test
2. State exactly what the null hypothesis is (depends on test!)
3. Comment the outcome: do you reject the null hypothesis or not, and what does this mean for the actual question we wanted to answer (interpretation)?

When we state "use tidyverse" it means that you should:

1. Only use tidyverse functions.
2. As far as possible, make a combined data analysis using pipes (%>%) so that intermediaries are kept at the necessary minimum.

Please use knitr::kable() to produce nicely formatted tables when you are asked provide a table.

Please note you need IsoformSwitchAnalyzeR v > 1.5.11 if not you need to update first (see the announcement on Absalon for instructions on how to). To show that you have the right version please include and run the following R code in your Rmarkdown:

```
paste(
    'The IsoformSwitchAnalyzeR version is okay:',
    packageVersion("IsoformSwitchAnalyzeR") > "1.5.11",
    sep=' '
)
```

```
## [1] "The IsoformSwitchAnalyzeR version is okay: TRUE"
```

##Intro As you already know how to quantify RNA-seq data (see the quantification exercise during RNA seq lecture) this HW is about post analysis of such quantifications.

##Part1: Data analysis and clustering Use the supplied Salmon quantification subset stored in the "salmon_result_part1.zip" file. These files contain the Salmon quantification of 6 samples - 3 biological replicates of non-treated (WT) and 3 biological replicates of where the cells were treated with a cancer promoting drug called TPA (WTTPA). Salmon was run with the "–seqBias" option.

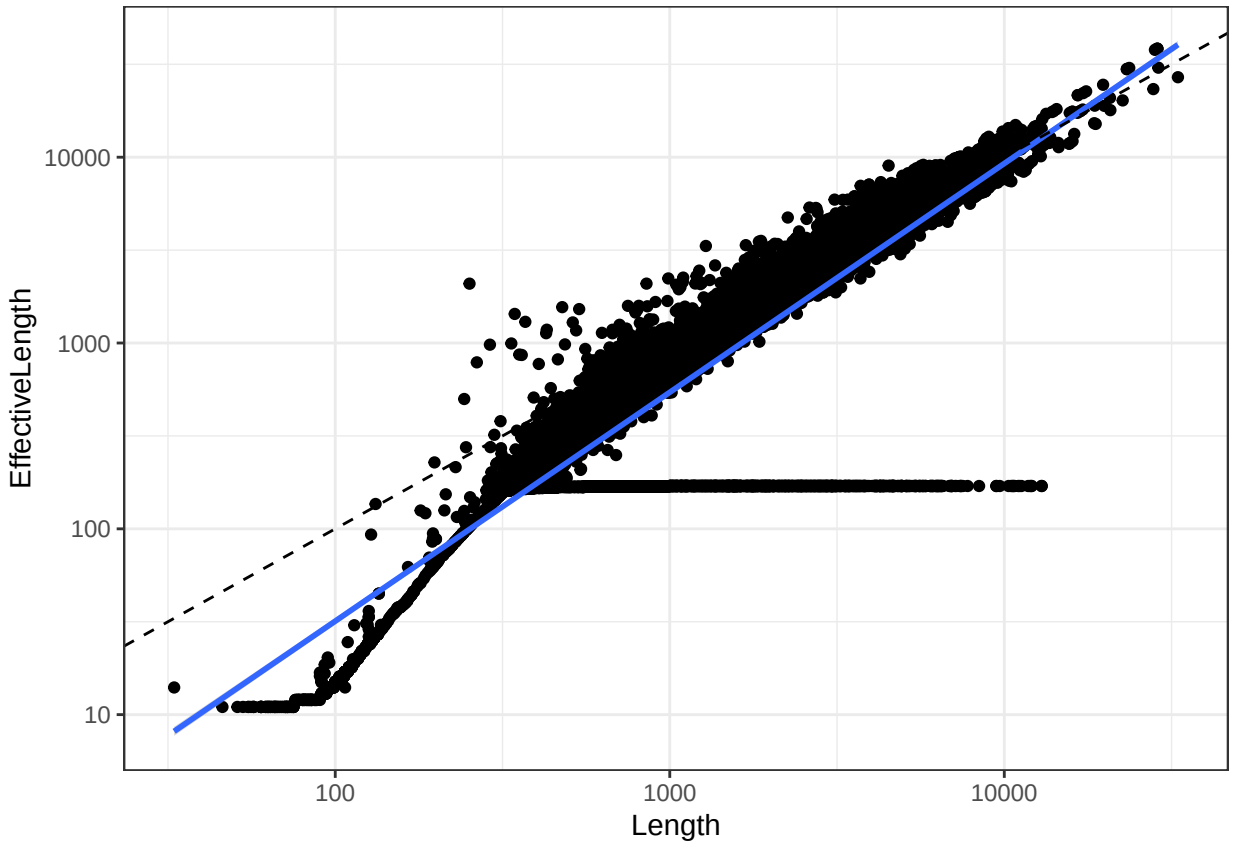
####Question 1.1 Read the "quant.sf" file from the WT1 Salon result folder into R with "read_tsv()". Plot the isoform length versus the effective length, add a geom smooth and a dashed line along the diagonal. Scale both axis using log10 via ggplot. Comment on the comparison on the differences between the trend line and the diagonal line with respect to what is expected.

```
WT1 <- read_tsv('salmon_result_part1/WT1/quant.sf')
```

```
## Parsed with column specification:
## cols(
##   Name = col_character(),
##   Length = col_double(),
##   EffectiveLength = col_double(),
##   TPM = col_double(),
##   NumReads = col_double()
## )
```

```
WT1 %>% ggplot(mapping = aes(x = Length, y = EffectiveLength)) + geom_point() + theme_bw() + geom_smooth
```

```
## `geom_smooth()` using formula 'y ~ x'
```

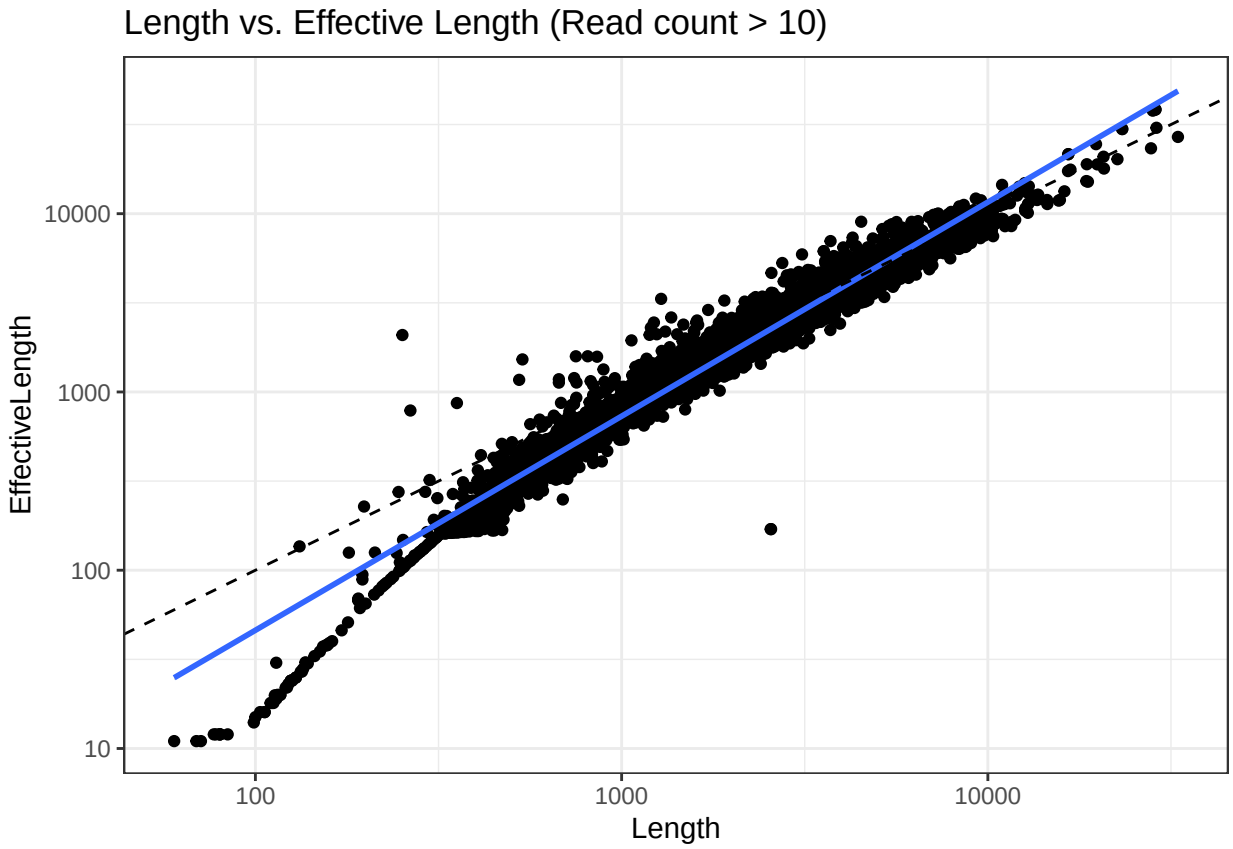


The trend line is pulled downwards from the diagonal due to a series of strange outliers whose gene lengths apparently form a sigmoid against their effective lengths. Futhermore, a tilt down to the left below the diagonal might be explained by the fact that the effective length of short genes is reduced proportionally more in relation to their gene lengths as compared to long genes.

####Question 1.2 Analyze and comment on the strange outliers in the plot from Question 1.1. **Use max 100 words.**

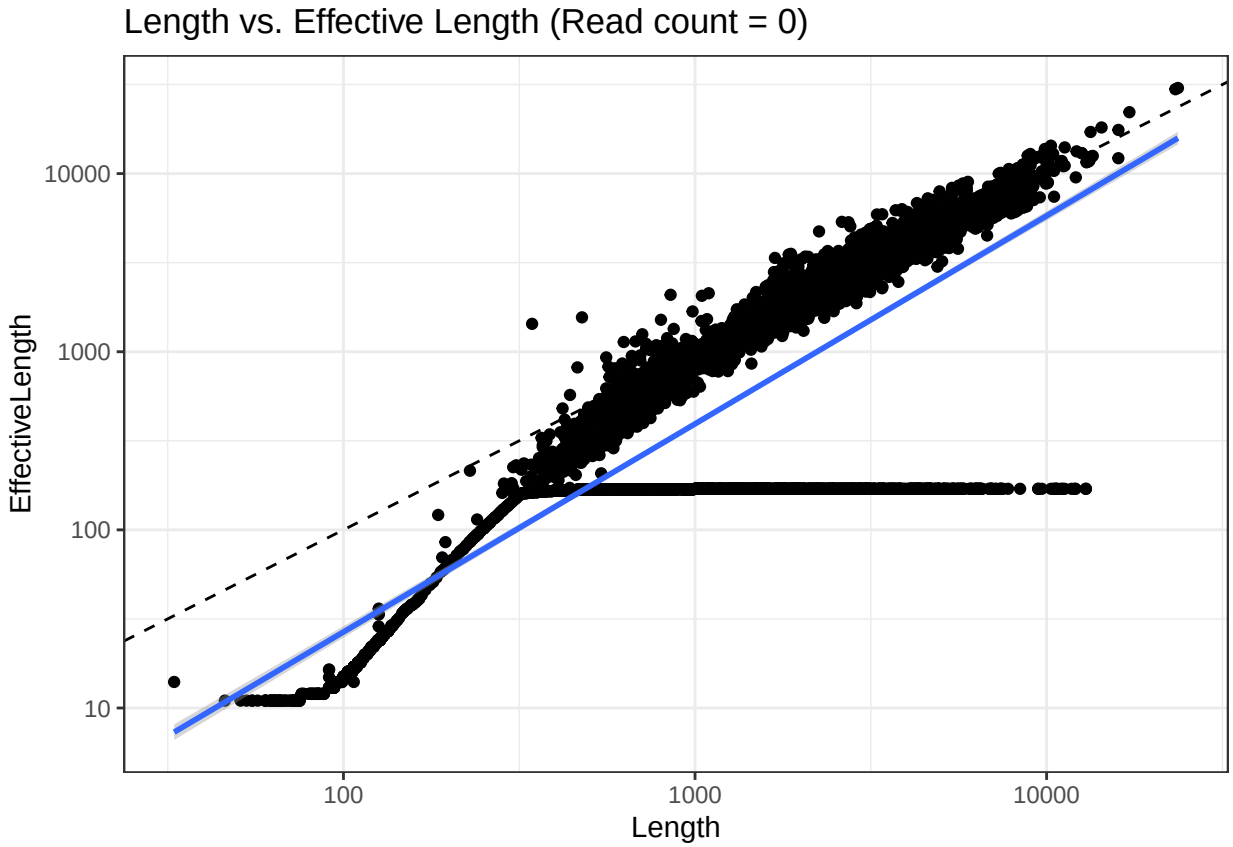
```
WT1 %>% filter(NumReads > 10) %>%
  ggplot(mapping = aes(x = Length, y = EffectiveLength)) + geom_point() +
  theme_bw() + geom_smooth(method = 'lm') + geom_abline(slope = 1, lty = 'dashed') +
  scale_y_log10() + scale_x_log10() + ggtitle('Length vs. Effective Length (Read count > 10)')
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
WT1 %>% filter(NumReads == 0) %>%
  ggplot(mapping = aes(x = Length, y = EffectiveLength)) + geom_point() +
  theme_bw() + geom_smooth(method = 'lm') + geom_abline(slope = 1, lty = 'dashed') +
  scale_y_log10() + scale_x_log10() + ggtitle('Length vs. Effective Length (Read count = 0)')
```

```
## `geom_smooth()` using formula 'y ~ x'
```



As seen when plotting only the lengths of transcripts with 0 read counts this includes most of the strange outliers and when cutting off read counts below 10 the strange outlies are almost gone. This suggests that it could be an algorithmic effect on effective length estimation at low read estimates. This could also be an issue with the data as there are relatively low read counts in the sample (around 1.5M) when at least >5M are usually expected to examine gene expression (according to illumina).

####Question 1.3 Use IsoformSwitchAnalyzeR's `importIsoformExpression()` to import all the data into R. Convert the abundances imported by `importIsoformExpression()` into a log2 transformed abundance matrix (using a pseudocount of 1) where columns are samples and isoform ids are stored as rownames. Report the first 4 rows as a table and discuss the advantage of a pseudocount of 1. **Use max 100 words.**

```
SQ <- importIsoformExpression(parentDir = 'salmon_result_part1/', addIsofomIdAsColumn = FALSE)
```

```
## Step 1 of 3: Identifying which algorithm was used...
##     The quantification algorithm used was: Salmon
##     Found 6 quantification file(s) of interest
## Step 2 of 3: Reading data...
## reading in files with read_tsv
## 1 2 3 4 5 6
## Step 3 of 3: Normalizing abundance values (not counts) via edgeR...
## Done
```

```
AbundanceMatrix <- log2(SQ$abundance + 1)
AbundanceMatrix[1:4,]
```

```
##                      WT1       WT2       WT3    WTTPA1    WTTPA2 WTTPA3
```

```
## TCONS_00000001 0.2973299 0.0000000 0.0000000 0.3822156 0.0000000       0
## TCONS_00000002 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000       0
## TCONS_00000003 0.0000000 0.2984888 0.2253968 1.0124265 0.0000000       0
## TCONS_00003946 0.0392366 0.0000000 0.1913649 0.0000000 0.0564598       0
```

Without adding a pseudocount of 1 genes with zero TPM would get a value of minus infinity when taking their log. This would mess up the calcutation of summary statistics such as the mean and variance.

####Question 1.4 **Use tidyverse** to extract the 100 most variable isoforms (aka those with highest variance) from the log2-transformed expression matrix. Provide a table with top five most variable isoforms.

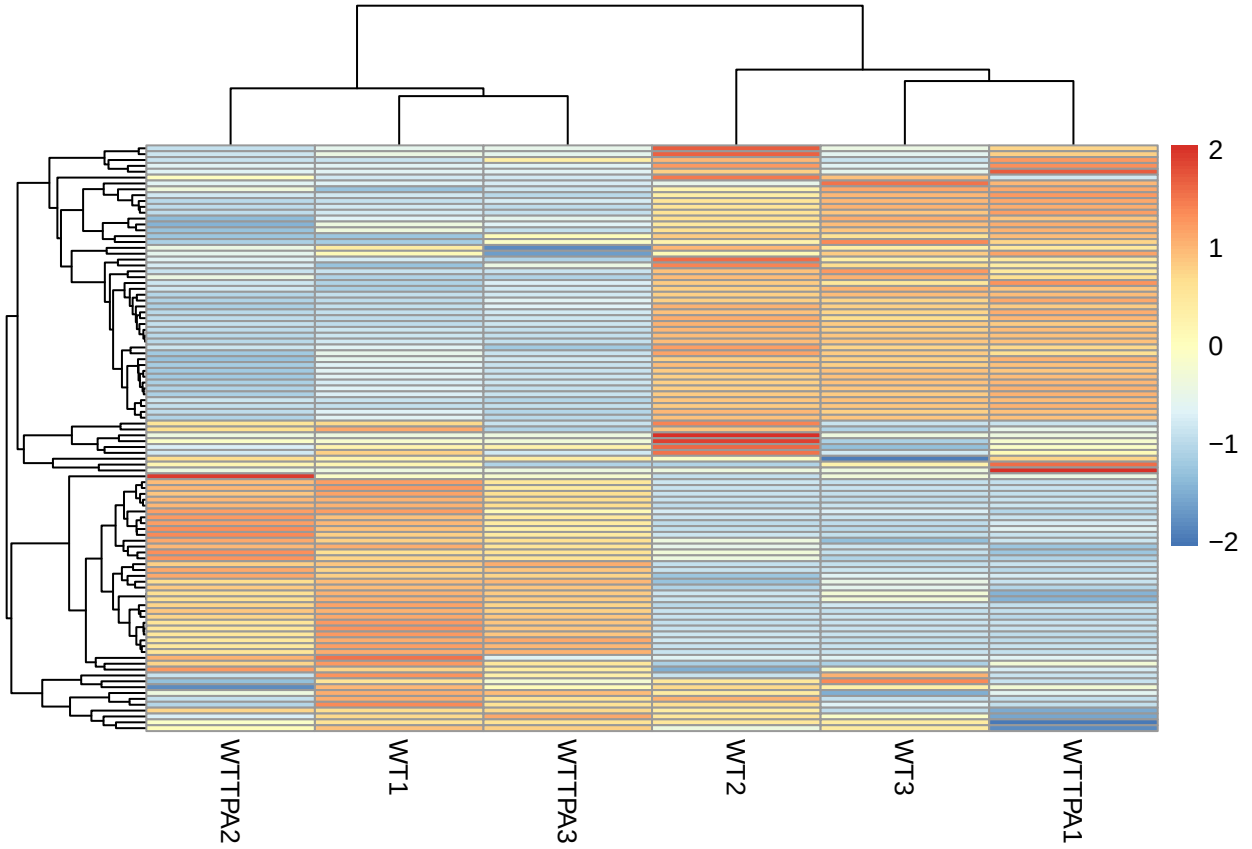
```
VarSortedAbundanceMatrix <- AbundanceMatrix %>% rownames_to_column() %>%
  mutate(variance = apply(AbundanceMatrix,1,var)) %>%
  arrange(desc(variance)) %>% head(100)  %>%
  remove_rownames() %>% column_to_rownames(var='rowname')

VarSortedAbundanceMatrix %>% head(5)
```

```
##                       WT1      WT2      WT3   WTTPA1   WTTPA2   WTTPA3  variance
## TCONS_00010929 8.663593 8.329549 8.082337 0.000000 6.265013 8.663799 11.470264
## TCONS_00006168 5.982148 0.000000 0.000000 0.000000 3.876122 5.435162  8.273979
## TCONS_00006650 0.000000 6.205570 0.000000 0.000000 0.000000 0.000000  6.418182
## TCONS_00001502 8.066044 2.439869 4.496400 3.436107 7.262710 8.104670  6.199817
## TCONS_00003104 1.538157 6.325017 5.913584 5.883973 1.664267 1.907422  5.682774
```

####Question 1.5 Use the pheatmap R package to make one (and just 1) visually appealing heatmap of the isoforms from 1.4 and comment on the result. Columns should be samples and rows isoforms. Furthermore, discuss pros and cons of the argument `scale = "row"` vs `scale = "none"`. **Use max 100 words.**

```
pheatmap(VarSortedAbundanceMatrix[1:6], show_rownames = F, scale = 'row')
```



The results show that the WT1 data clusters with the TPA treated samples whereas WTTPA1 clusters with the other non-treated samples. Generally the clustering seems to be robust with good separation of what isoforms are putatively upregulated in some of the samples. It seems that there might be a large-scale switch in isoform expression upon treatment although this needs statistical verification. When scaling by row one loses information about the absolute expression levels of the isoforms so that highly expressed isoforms cannot be differentiated from lowly expresssed isoforms. On the other hand when scaling by row the relative expression of each isoform between samples becomes comparable regardless of whether the gene is highly or lowly expressed.

##Part2: Isoform switch analysis with IsoformSwitchAnalyzeR Use the supplied Salmon quantification subset stored in the "salmon_result_part2.zip" file (Different from what you used in part 1!). These files contain the Salmon quantification of 6 samples - 3 biological replicates of non-treated (WT) and 3 biological replicates of a knock out (KO) of a suspected splice factor - let us call it the X factor for dramatic effect. Salmon was run with the –seqBias option.

Your job is to analyze the changes to the transcriptome using IsoformSwitchAnalyzeR to elucidate the effect of the knock out in relation to the hypothesis that factor X is a splice factor.

####Question 2.1 Use the `importIsoformExpression` and `importRdata(...,addAnnotatedORFs=FALSE)` functions to create a switchAnalyzeRList object from the Salmon output supplied in the "salmon_result_part2.zip" folder. Use the GTF file also included in the zip file. Report the summary statistics of the resulting switchAnalyzeRList. What does the `addAnnotatedORFs=FALSE` argument do and why do you think it is enabled here?

```
SQ2 <- importIsoformExpression(parentDir = 'salmon_result_part2/', addIsofomIdAsColumn = FALSE)
```

```
## Step 1 of 3: Identifying which algorithm was used...
##     The quantification algorithm used was: Salmon
##     Found 6 quantification file(s) of interest
## Step 2 of 3: Reading data...
## reading in files with read_tsv
## 1 2 3 4 5 6
## Step 3 of 3: Normalizing abundance values (not counts) via edgeR...
## Done
```

```
SwitchList <- importRdata(isoformCountMatrix = SQ2$counts, isoformRepExpression = SQ2$abundance,
                          designMatrix = data.frame(sampleID = colnames(SQ2$abundance),
                                                    condition = c(rep('KO',3),rep('WT',3))),
                          addAnnotatedORFs=FALSE,
                          isoformExonAnnoation = 'salmon_result_part2/subset.gtf')
```

```
## Step 1 of 6: Checking data...
##     Using row.names as 'isoform_id' for 'isoformCountMatrix'. If not suitable you must add them manua
##     Using row.names as 'isoform_id' for 'isoformRepExpression'. If not suitable you must add them ma
## Step 2 of 6: Obtaining annotation...
##     importing GTF (this may take a while)
## Step 3 of 6: Calculating gene expression and isoform fraction...
##      2433 ( 24.33%) isoforms were removed since they were not expressed in any samples.
## Step 4 of 6: Merging gene and isoform expression...
##   |                                                                              |
## Step 5 of 6: Making comparisons...
##   |                                                                              |
```

```
## Step 6 of 6: Making switchAnalyzeRlist object...
## Done
```

```
SwitchList
```

```
## This switchAnalyzeRlist list contains:
##  7567 isoforms from 3304 genes
##  1 comparison from 2 conditions (in total 6 samples)
```

Setting the addAnnotatedORFs to FALSE means that we do not use the annoted ORFs from the GTF file imported in isoformExonAnnoation. This allows the later analysis to find ORFs de novo.

####Question 2.2 Why is it essential the annotation stored in the GTF file is the exact annotation quantified with Salmon (in the context of IsoformSwitchAnalyzeR functionalities)? **Use max 100 words.**

IsoformSwitchAnalyzeR needs the correct gene annotations for each isoform as they were quantified with Salmon so that each isoform can be mapped to the right gene in order to be able to quantify what genes have a change in isoform expression.

####Question 2.3 Load the supplied "switchList.Rdata" object into R with the `readRDS()` function. This is the result of running the whole IsoformSwitchAnalyzeR workflow on the full dataset. Make a table with the Top 10 switching genes with predicted consequences when sorting on q-values.

```
SwitchListAnalyzed <- readRDS('hw3switchList.Rdata')
topSwitches <- extractTopSwitches(SwitchListAnalyzed, filterForConsequences = TRUE,
                                  sortByQvals = TRUE)
head(topSwitches, 10)
```

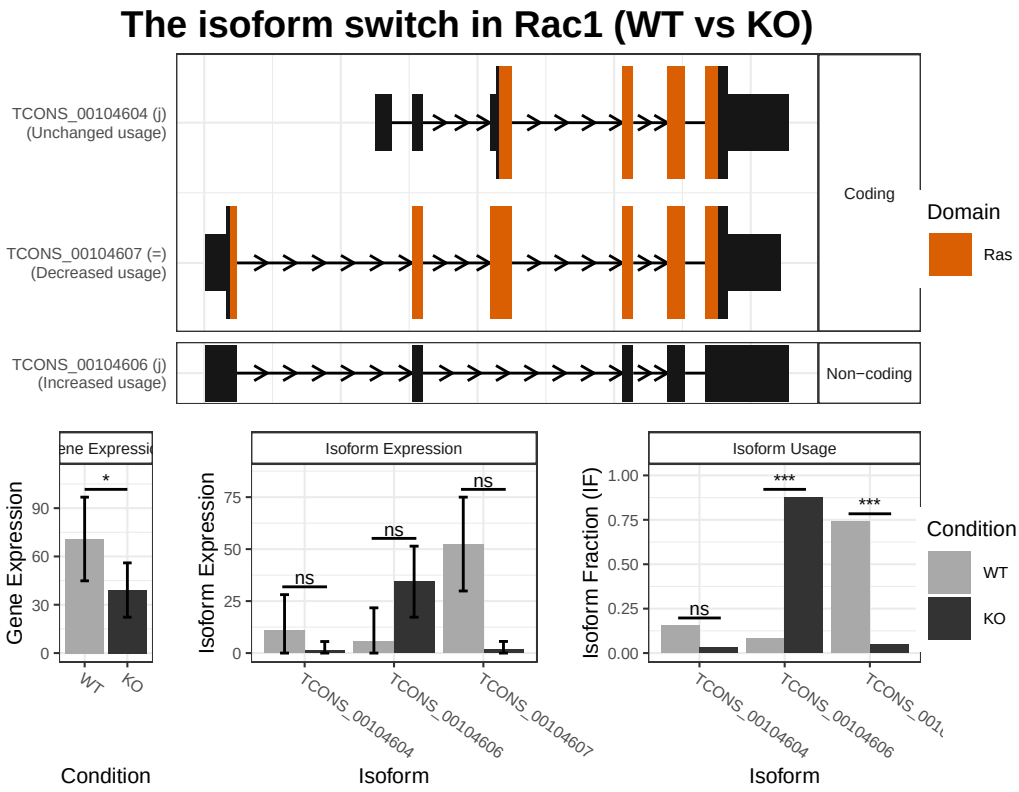
```
##             gene_ref           gene_id     gene_name condition_1 condition_2
## 1  geneComp_00100550       XLOC_047302 5830418K08Rik          WT          KO
## 2  geneComp_00076087       XLOC_023295        Ablim1          WT          KO
## 3  geneComp_00068215       XLOC_015573           Tef          WT          KO
## 4  geneComp_00068223       XLOC_015581         Xrcc6          WT          KO
## 5  geneComp_00101368 XLOC_048111:Snx14         Snx14          WT          KO
## 6  geneComp_00066816       XLOC_014190         Slmap          WT          KO
## 7  geneComp_00089842       XLOC_036766          Rac1          WT          KO
## 8  geneComp_00081221       XLOC_028310         Fbxw7          WT          KO
## 9  geneComp_00058485       XLOC_006025          Pld2          WT          KO
## 10 geneComp_00080160       XLOC_027267         Rrbp1          WT          KO
##    gene_switch_q_value switchConsequencesGene Rank
## 1         3.175544e-64                    TRUE    1
## 2         1.155042e-15                    TRUE    2
## 3         4.686282e-15                    TRUE    3
## 4         9.951012e-13                    TRUE    4
## 5         4.031854e-12                    TRUE    5
## 6         6.992658e-11                    TRUE    6
## 7         8.587909e-10                    TRUE    7
## 8         7.331074e-09                    TRUE    8
## 9         1.277562e-08                    TRUE    9
## 10        1.922603e-08                    TRUE   10
```

####Question 2.4 Show code for how to produce switchPlot for these 10 genes and save them to your own computer. The plots should not be included in the report (only the code for how to produce it)!

```
pdf(file = 'topgenes.pdf', onefile = TRUE, height=6, width = 9)
for(i in topSwitches$gene_id){
switchPlot(SwitchListAnalyzed, gene = i, condition1 = 'KO', condition2 = 'WT')
}
dev.off()
```

####Question 2.5 Which of the top 10 genes with switches do you think is the most important? Include/produce the switchPlot for that particular gene and discuss the reason why you chose that gene, including references when needed. **Use max 100 words.**

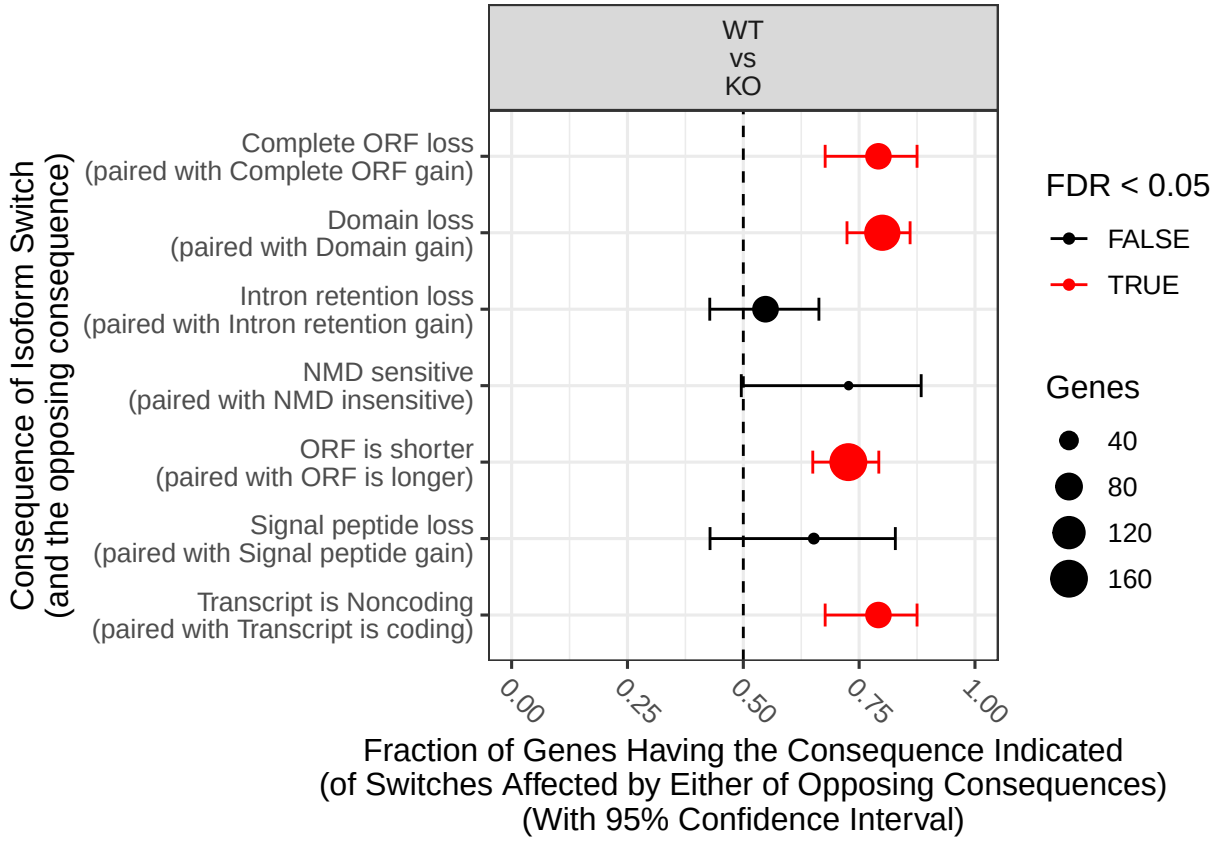
```
switchPlot(SwitchListAnalyzed, gene = 'Rac1', condition1 = 'KO', condition2 = 'WT')
```



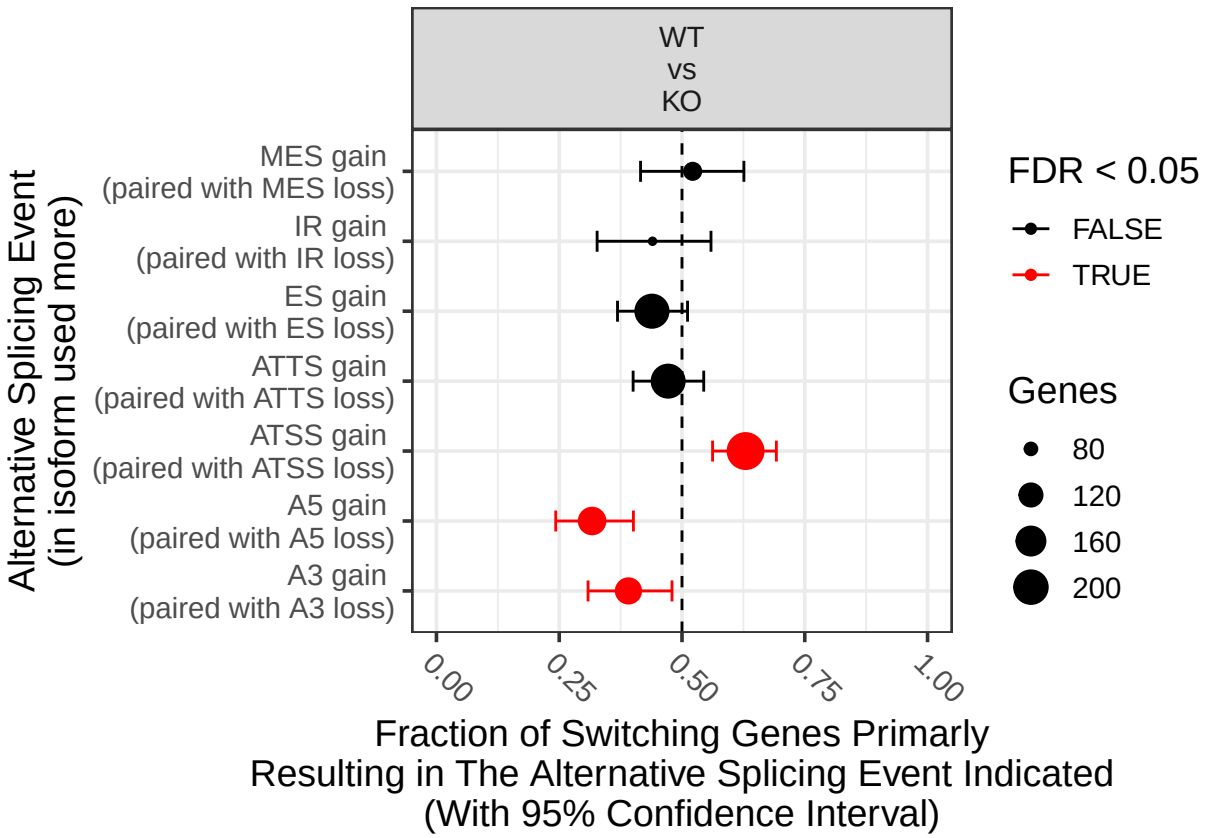
Rac1 is a GTPase involved in cytoskeletal regulation and cellular migration. The switchplot shows that there is a general decrease in expression of the gene and a very significant switch to a non-coding RNA demonstrating how isoform level resolution can improve information of gene expression. The effect alone of reducing Rac1 expression (assmuing no function of the ncRNA) includes deficent lammelopodia formation and cell migration aswell as a decrease in celullar adhesion and increase in anoikis (lack-of-adhesion mediated cell-death)(Fukun Guo et al. 2006).

####Question 2.6 Plot the global enrichment of switch consequences and alternative splicing and comment on it. What are the general patterns and what does that mean for the transcriptome? How does that relate to the original hypothesis about Factor X? **Use max 100 words.**

```
extractConsequenceEnrichment(
    SwitchListAnalyzed,
    consequencesToAnalyze='all',
    analysisOppositeConsequence = TRUE,
    returnResult = FALSE
)
```




```
extractSplicingEnrichment(
    SwitchListAnalyzed,
    returnResult = FALSE
)
```



Generally the consequence enrichment seems to be skewed to the side of more frequent loss than gain of ORFs, domains and coding transcripts aswell a shortening of ORFs. These are consequences with likely functional effects. The splice enrichment shows a more frequent gain of alternative transcription start sites and a general loss of 3' acceptor and 5' donor sites. The loss of splice sites generally alludes to less splicing activity which supports the hypothesis that the x-factor is indeed important for splicing, the gain of alternative start sites indicates that x-factor could also play a role in transcription initiation.