HW3: Transcriptome Analysis

Homework 3

Names: Mikkel Corfitzen, Katja Johansen, Stefano Pellegrini, Rikke Stausholm

Group: 5

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

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

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

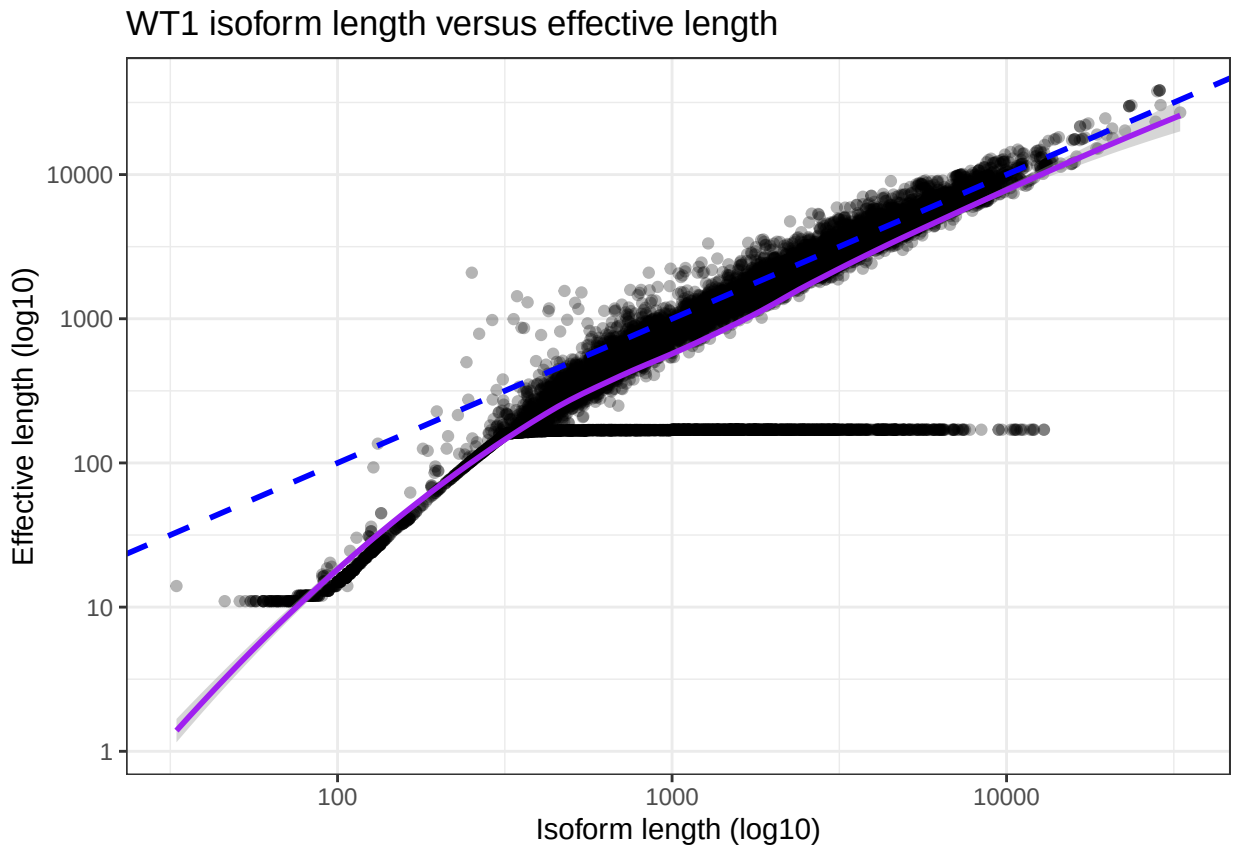
Read the "quant.sf" file from the WT1 Salmon result folder into R with "read_tsv()". Plot the isoform length versus the effective length, add a geom smooth and a dashed line along the diagonal. Scale both axis using log10 via ggplot. Comment on the comparison on the differences between the trend line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

```
quant_wt1 <- read_tsv("salmon_result_part1/WT1/quant.sf")

quant_wt1 %>% ggplot(aes(x=Length, y=EffectiveLength)) + geom_point(alpha=0.3) +
  geom_smooth(method="loess", color="purple") +
  geom_abline(intercept = 0, slope = 1, color="blue", linetype = "dashed", size=1) +

  scale_x_continuous(trans="log10") +
  scale_y_continuous(trans="log10") +

  labs(title = "WT1 isoform length versus effective length") +
  xlab("Isoform length (log10)") +
  ylab("Effective length (log10)") + theme_bw()
```



Salmon and other advanced methods for quantifying transcript abundance use modern algorithms called pseudo aligment. These algorithms are used for aligning reads directly to the transcripts and quantifying their expression using RNA-Seq data. This data has specific biases and these algorithms also aim to correct for them. Recalling that the length of an isoform is used for the within sample isoforms normalization, the most common bias in RNA-Seq data is the coverage problem. After sonification, we obtain random fragments of RNA and due to size selection, the very short fragments located at the ends of the isoforms are removed. This results in a reduction of the isoforms coverage, and this affects more shorter isoforms than longer ones. This is the main reason why Salmon computes the effective length and uses it for the isoforms normalization and for the calculation of the TPM (Transcript Per Million). The effective length takes into account all factors that can effect the probability of sampling fragments from a certain transcript. These include the fragment length distribution of a given transcript and the sequence-specific and gc-fragment bias [1].

We can describe the effective length of an isoform i as $\tilde{l}_i = l_i - \mu_{l_i}$, where l_i is the isoform length and μ_{l_i} is the average length of all fragments mapping to that isoform [1]. Then, more avanced statistical methods are used to incorporate the fragment bias correction into the effective length, such that it also takes into account the likelihood of sampling each possible fragment that the transcript can produce [1]. So, the effective length accounts for the bias correction, and for the fact that the range of fragment sizes that can be sampled is limited near the ends of a transcript. Therefore, it doesn't depend only to the isoform length but also to the distribution of the length of the reads mapping to that isoform and to the additional fragment bias.

This explain the differences between the trend line and the diagonal line. In fact, we can see that the effective length of the shorter transcripts is smaller than the effective lenghts of the longer ones, and that is because the effective length correction takes into account that the shorter isoforms are more affected by the coverage problem than the longer ones.

Question 1.2

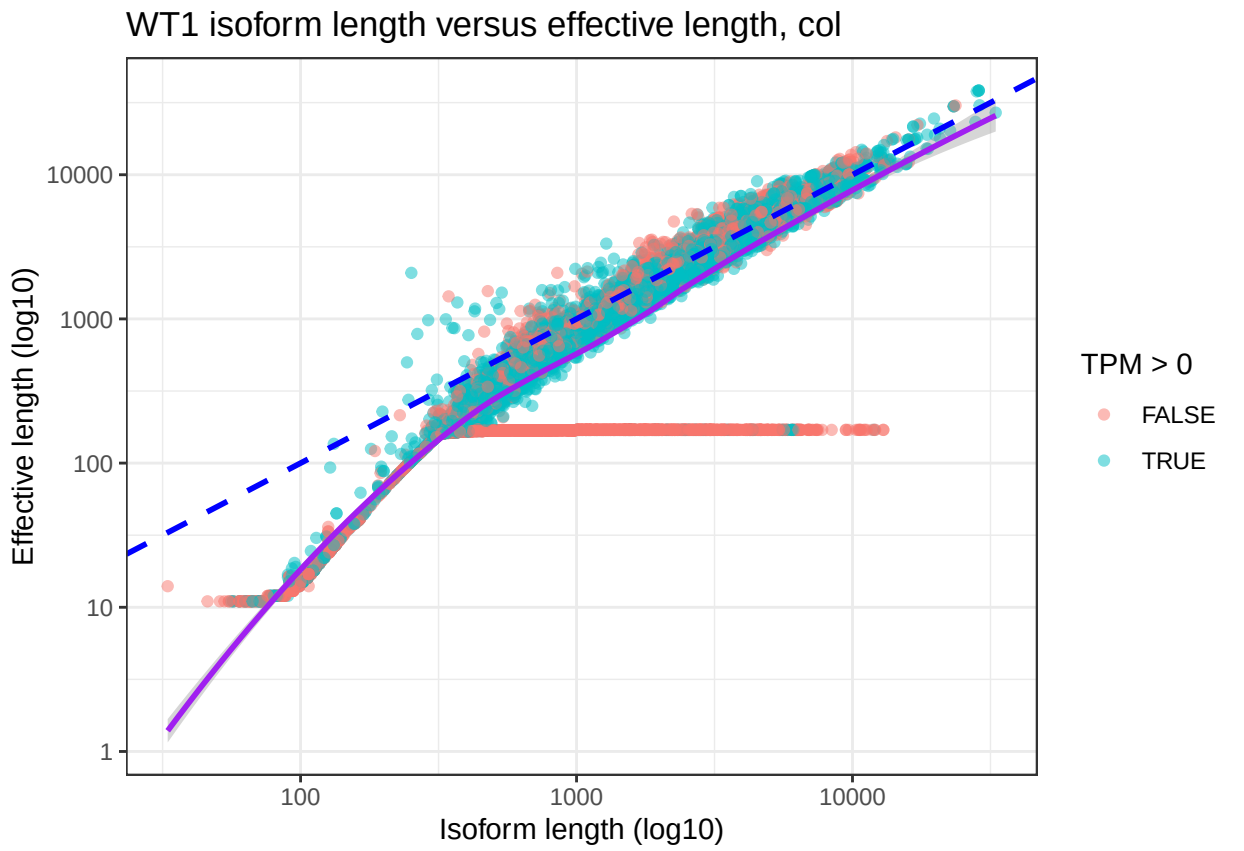
Analyze and comment on the strange outliers in the plot from Question 1.1. **Use max 100 words.**

```
quant_wt1 %>% ggplot(aes(x=Length, y=EffectiveLength, color=TPM>0)) + geom_point(alpha=0.5) +
  geom_smooth(method="loess", color="purple") +
  geom_abline(intercept = 0, slope = 1, color="blue", linetype = "dashed", size=1) +

  scale_x_continuous(trans="log10") +
  scale_y_continuous(trans="log10") +

  labs(title = "WT1 isoform length versus effective length, col") +
  xlab("Isoform length (log10)") +
  ylab("Effective length (log10)") + theme_bw()
```

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



The outliers in the plot from question 1.1. are data-points with an log10 effective length of approximatively 250 bases. We think these points corresponds to isoforms of different length, which have no reads mapped to them. Our hypothesis is that the algorithm assign a default effective length value to the transcript with zero coverage. Since there are other transcripts with zero coverage that have a different effective length, we think that other variables are taken into account. The other variables could be the fragments bias mentioned in answer 1.1.

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.**

```
salmonQuant <- importIsoformExpression(parentDir = "salmon_result_part1")

log2TPM <- as.matrix(log2(salmonQuant$abundance[2:ncol(salmonQuant$abundance)]+1))
row.names(log2TPM) <- salmonQuant$abundance$isoform_id

knitr::kable(head(log2TPM, 4),
             format = "latex",
             booktabs = TRUE,                    # Format
             align = "lccccc",                   # Columns aligment (left, center, right)
             caption = "Log2 abundance matrix.") %>%
  kable_styling(latex_options = "hold_position") %>%
  row_spec(0, bold=TRUE ) %>%                    # Bold header
  column_spec(1, bold=TRUE)                      # Bold row names
```

Table 1: Log2 abundance matrix.

	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

```
dim(as_tibble(log2TPM))
```

```
## [1] 10000     6
```

The advantage of using a pseudocount of 1 is that by adding 1 to all TPMs before applying the log2, we avoid computing log2 of zero, which is undefined.

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.

```
as_tibble(log2TPM) %>%
  add_column(Isoform_ID = salmonQuant$abundance$isoform_id, .before = "WT1") %>% # Add ID
  mutate(Variance = apply(log2TPM, 1, var)) %>%
  arrange(desc(Variance)) %>%
  head(100) -> top100log2TPM

knitr::kable(head(top100log2TPM, 5),
             format = "latex",
             booktabs = TRUE,
             align = "lrrrrrrr",
             caption = "Top 5 most variable isoforms from log2 abundance matrix.") %>%
  kable_styling(latex_options = c("hold_position", "scale_down")) %>%
  row_spec(0, bold=TRUE)
```

Table 2: Top 5 most variable isoforms from log2 abundance matrix.

Isoform_ID	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

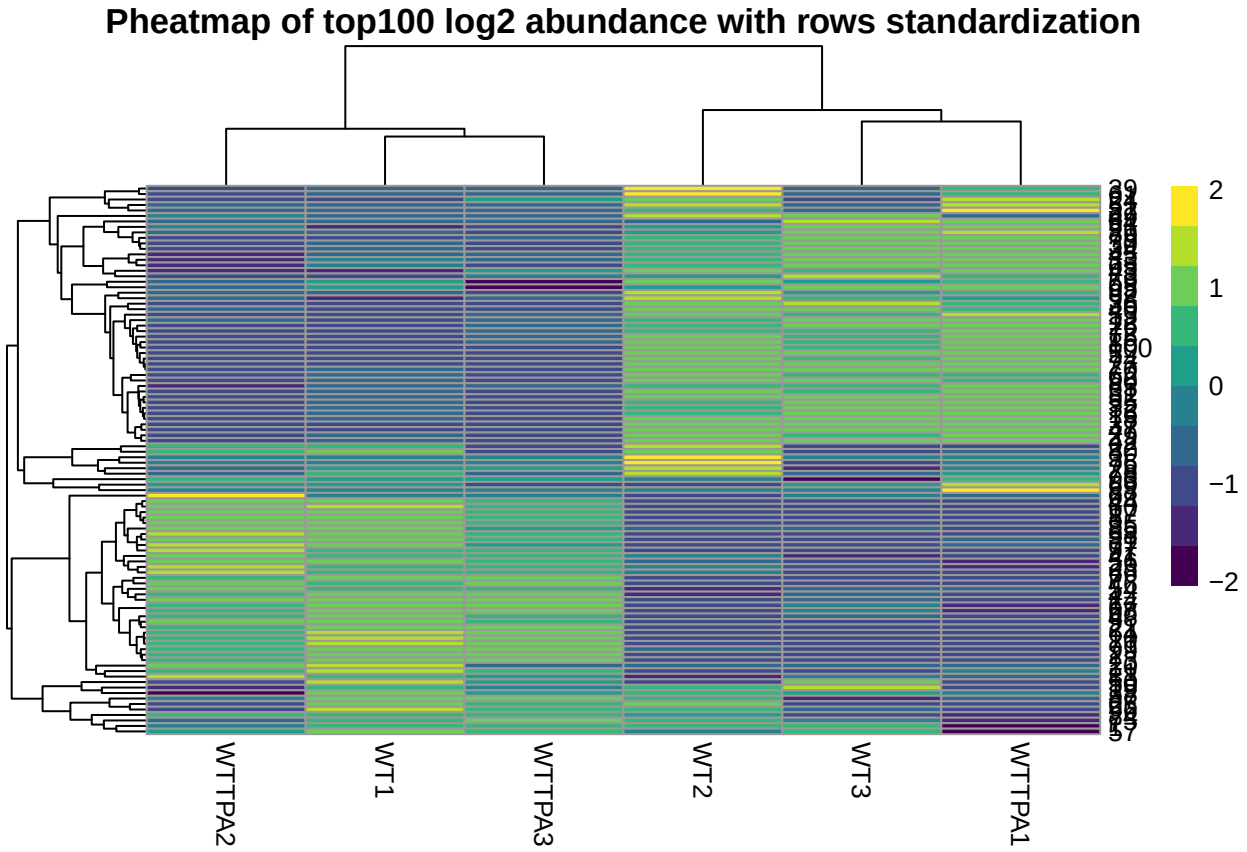
Use the pheatmap R package to make one (and just 1) visually appealing heatmap of the isoforms from 1.4 and comment on the result. Columns should be samples and rows isoforms. Furthermore, discuss pros and cons of the argument `scale = "row"` vs `scale = "none"`. **Use max 100 words.**

```
library("viridis")
```

```
## Loading required package: viridisLite
```

```
# Removing ID and Variance column from the sorted by variance abundance
top100log2TPM_heat <- top100log2TPM[c(-1,-ncol(top100log2TPM))]

# Pheatmap
pheatmap(top100log2TPM_heat, scale = "row", color=viridis(10),
         main = "Pheatmap of top100 log2 abundance with rows standardization")
```



The pheatmap shows two clusters of samples and two clusters of isoforms. We expect all the WT samples in one cluster and the WTTPA samples in another cluster but we note that WT1-sample is clustered together with the treated WTTPA2/3 samples. We observe a contrast in isoform expression pattern across the clusters of samples, which is almost opposite.

After using both scale="row" (normalized) and scale="none" (non normalized) we found a difference between the two pheatmaps. In the non normalized pheatmap the absolute values are presented which can be used to see the total expression of the different isoforms. In the normalized pheatmap the data are centered and scaled such that each row will have a mean of 0 and a standard deviation of 1. In this way, the colors provide us information about the relative expression of an isoform between different samples.

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?

```
# Import Salmon quantification data
salmonQuant <- importIsoformExpression(
    parentDir = ("salmon_result_part2/"),
    addIsofomIdAsColumn = TRUE
)
```

```
# Make design matrix (indicate which biological replicates belong to which condition)
myDesign <- data.frame(
    sampleID = colnames(salmonQuant$abundance)[-1],
    condition = sapply(colnames(salmonQuant$abundance)[-1], function(x) gsub('\\d+','', x))
)

# Create switchAnalyzeRlist
mySwitchList <- importRdata(
                  isoformCountMatrix = salmonQuant$counts, # Import counts from salmon
                  isoformRepExpression = salmonQuant$abundance, # Import abundance from salmon
                  designMatrix         = myDesign,
                  isoformExonAnnoation = ("salmon_result_part2/subset.gtf"), # Isoform annotation
                  addAnnotatedORFs=FALSE)
```

```
# Summary
summary(mySwitchList)
```

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

addAnnotatedORFs=FALSE is an argument used to specify that the coding sequence (CDS) and the open reading frame (ORF) annotated in the GTF file should not be added to the switchAnalyzeRlist. We enable it here because we did not supply fasta file containing the transcripts sequence.

Question 2.2

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

Because the GTF file contains the transcript structure of the isoforms (in genomic coordinates) as well as information about which isoforms originate from the same gene, while the data obtained from Salmon contains the quantification of the fragments that map to each isoform. So it is important that the annotation used in GTF and Salmon files match.

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.

```
myCompleteSwitchList <- readRDS("hw3switchList.Rdata")

# Top 10 switching genes
top10Genes <- extractTopSwitches(
              myCompleteSwitchList,
              filterForConsequences = TRUE,
              n=10)

knitr::kable(top10Genes,
             format = "latex",
             booktabs = TRUE, # Format
             align = "lllccccc", # Columns aligment (left, center, right)
             digits = 70,
             caption = "Top 10 switching genes with predicted consequences.") %>%
  kable_styling(latex_options="scale_down") %>% row_spec(0, bold=T )
```

Table 3: Top 10 switching genes with predicted consequences.

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

```
# SwitchPlot of the top 10 genes
switchPlotTopSwitches(switchAnalyzeRlist = myCompleteSwitchList,
                      n = 10,
                      filterForConsequences = TRUE,
                      fileType = "pdf",
                      pathToOutput = "."
)
```

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.**

We chose the gene Snx14 (Figure 1) because we see a large change in expression between the TCONS_00135386 (top) and TCONS_00135388 (2° from top), in the WT compared to the mutant. TCONS_00135386 (top) is highly expressed in the WT, and is reduced in the mutant. While TCONS_00135388 (2° from top) is increased in the mutant. The TCONS_00135388 (2° from top) includes a signal peptide sequence in the N-terminal which targets the protein to the secretory pathway, which results in the protein being translocated to the ER, or membrane bound to the organelles in the secretory pathway as a membrane protein [2]. Effectively the TCONS_00135388 (2° from top) isoform is likely translocated to a different subcellular compartment.

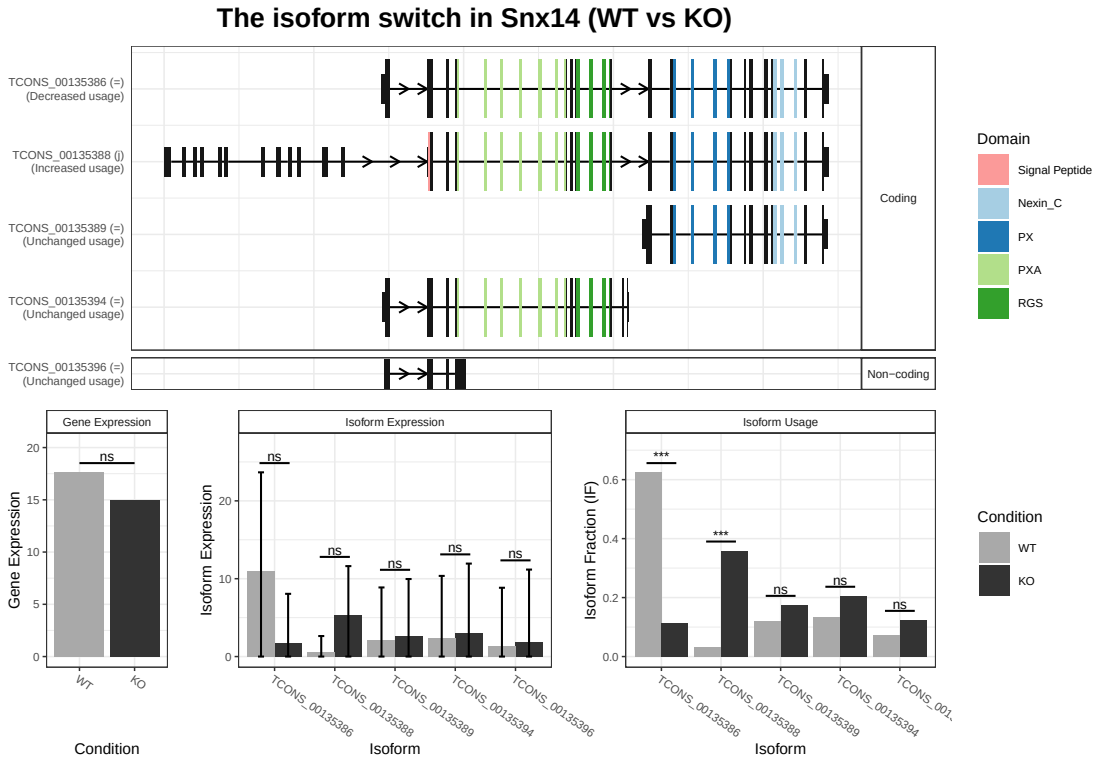
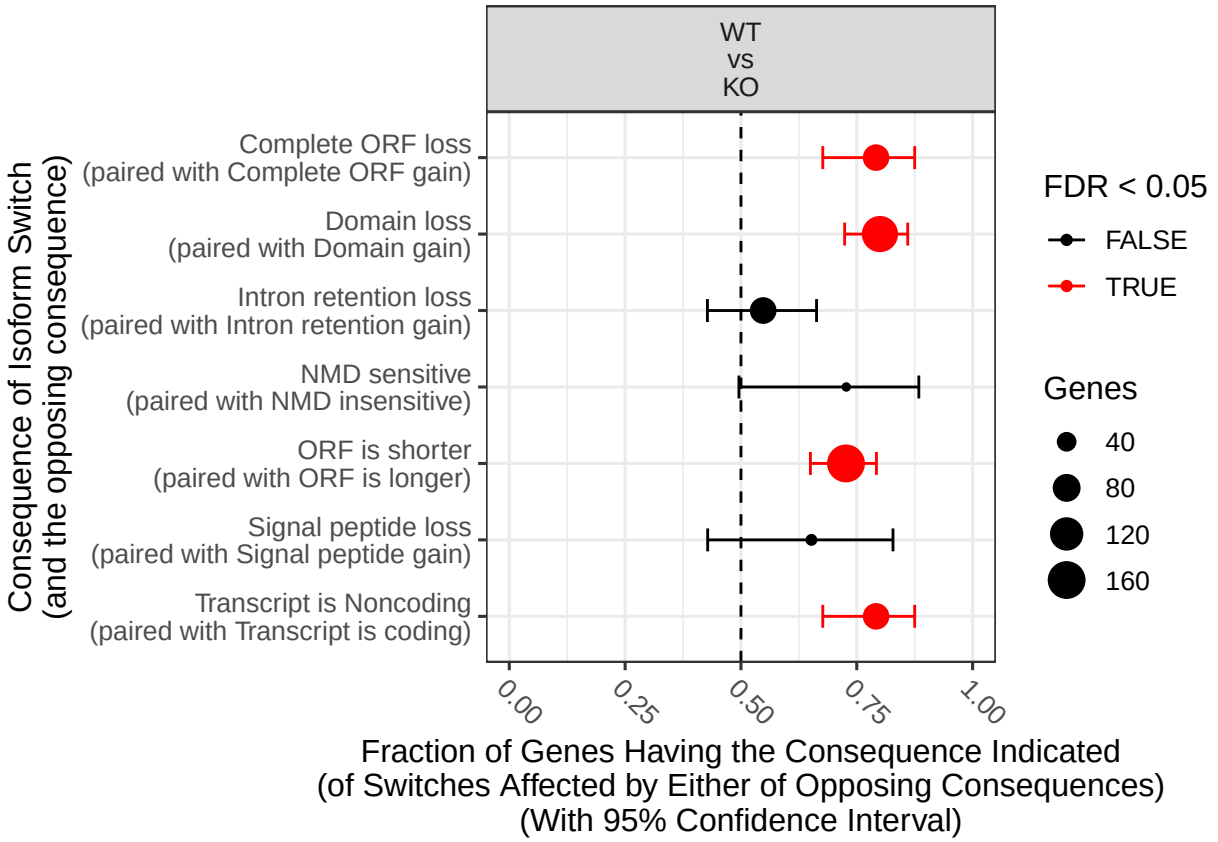


Figure 1: SwitchPlot of Snx14 gene.

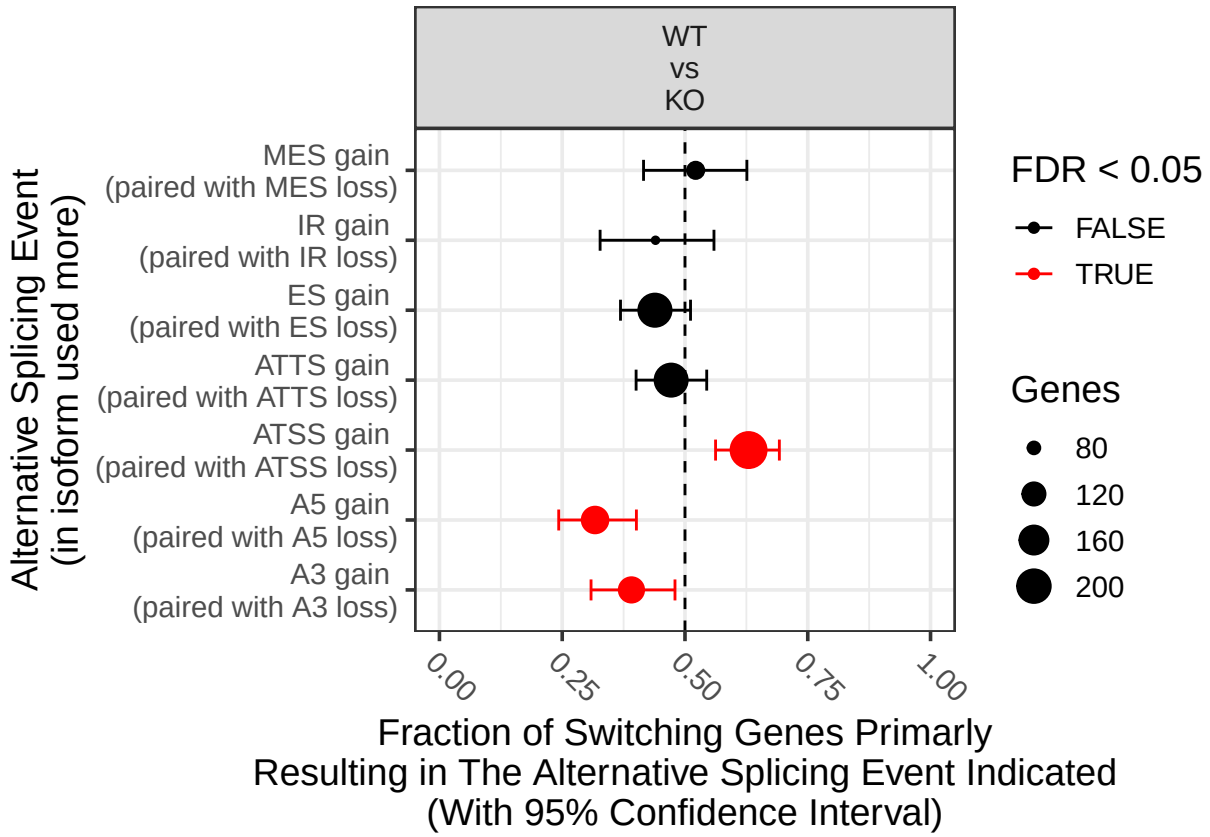
Question 2.6

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.**

```
# Consequence Enrichment Analysis
extractConsequenceEnrichment(
    myCompleteSwitchList,
    consequencesToAnalyze='all',
    analysisOppositeConsequence = TRUE,
    returnResult = FALSE
)
```




```
# Splicing Enrichment Analysis
extractSplicingEnrichment(
    myCompleteSwitchList,
    returnResult = FALSE
)
```



From the global enrichment of switch consequences we observe a significant increase in complete ORF loss, a significant increase in Domain loss, a significant reduction in ORF length and a significant increase in the amounts of non-coding mRNAs. All these observations together show a pattern of disruption of the normal splicing in the cells. This disruption results in abnormal transcripts and an altered transcriptome.

From the splicing enrichment analysis we observe a significant increase in alternative transcription start site, and a significant loss of 3' and 5' acceptor splice site compared to WT. These underlying reasons are what causes the altered transcriptome. The loss of 3' and 5' acceptor splice site compared to the WT indicates X to be a splicing factor involved in recognizing 3' and 5' acceptor splice sites. Thus the data indicates that X are involved in defining the intron/exon border and helps regulate normal splicing function.

Reference

[1] Patro, R., Duggal, G., Love, M. et al. Salmon provides fast and bias-aware quantification of transcript expression. Nat Methods 14, 417–419 (2017).
[1] Kelemen, Olga et al. Function of Alternative Splicing. Gene 514.1, 1–30 (2013).