Assignment 2 group 6

Adham Khaled, Juan Manuel Medina, Antonio Ortega, Isabella Skandorff & Andreas Vincent

May 15, 2017

```
library("ggplot2")
library("cowplot")
library("reshape2")
library("dplyr")
library("gridExtra")
library("VennDiagram")
theme_set(theme_bw())
```

Part 1

(a) What are the first five genomic nucleotides from the first exon of this transcript?

AAAGG

The DICER1 mRNA molecule should have the same sequence as the DNA genomic sequences in the sense strand (substituting Ts by Us). As AK002007 is transcribed on the reverse strand and the default genomic sequence presented by the browser is the antisense one, we have to reverse it. Therefore, the first five nucleotides of the exon in the AK002007 cDNA are AAAGG.

(b) Look at the raw mRNA sequence of AK002007, from the database it actually comes from. What are the first five nucleotides?

When we check the raw mRNA sequence of AK002007, it can be seen that the first five nucleotides are GAAGC.

(c) How do you explain the discrepancy (maximum 5 lines)?

AK002007 is possibly a truncated version of the DICER1 mRNA found in Genbank and refseq. The first 11 nt of AK002007 are found in the last 11 nt of the previous exons in the other mrna sequences. The first 4 nt of these are postulated to be part of the 5' UTR of AK002007 by the aligner, confirming a misalignment. Otherwise, those 11 nt would be aligned as a separate exon in alignment with the other mrna sequences.

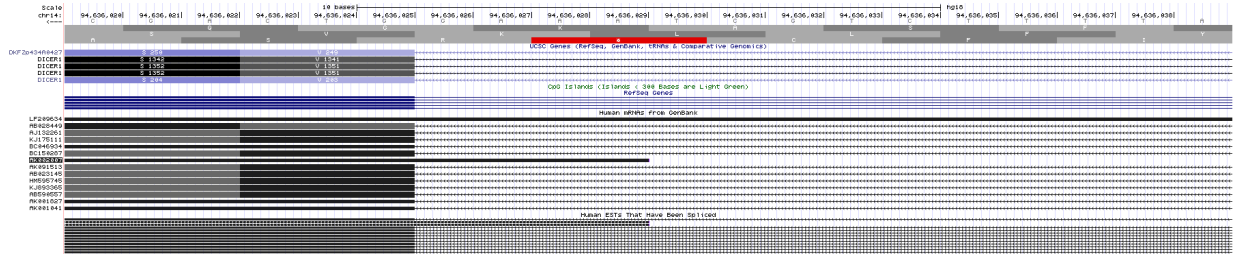


Figure 1: Screenshot of the start of AK002007. The first 7 letters are discarded by the aligner, while the following 4 are aligned to the end of the intron.

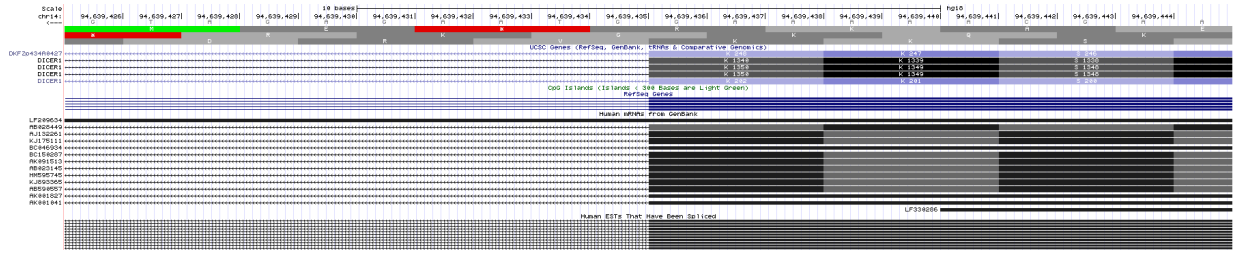


Figure 2: Screenshoot of the upstream exon. As we can see, the last letters of the exon match the letters aligned to the intron.

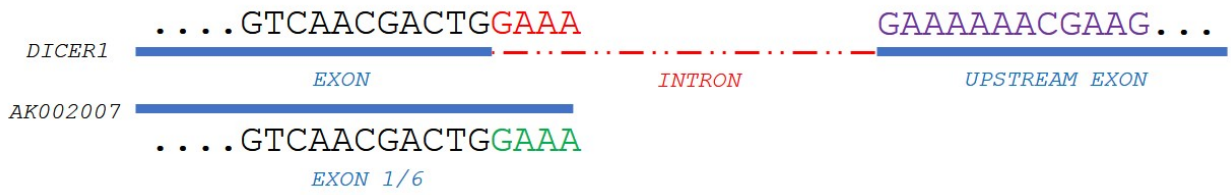


Figure 3: Shows the mRNA sequence of AK002007 and the general DICER1. These mRNA sequences are read from the right to the left, because they are transcribed from the minus strand. Remarkably, the final 4 letters of the upstream exon and the intron are the exact same GAAA

Part 2

(a)

Bedtools commands to preprocess data before R

```
cat ERa_hg18.bed ERb_hg18.bed > full.bed
bedtools sort -i full.bed > full.sorted.bed
bedtools merge -i full.sorted.bed > merged.bed
# Calculates the percent coverage by chromosome
bedtools genomecov -i merged.bed -g hg18_chrom_sizes.txt > coverage.txt
bedtools genomecov -i ERa_hg18.bed -g hg18_chrom_sizes.txt > ERa.txt
bedtools genomecov -i ERb_hg18.bed -g hg18_chrom_sizes.txt > ERb.txt
```

R

```
df <- rbind(cbind(read.table(file = "coverage.txt"), protein = "Merged"),
            cbind(read.table(file = "ERa.txt"), protein = "ERa"),
            cbind(read.table(file = "ERb.txt"), protein = "ERb"))
colnames(df) <- c("chr", "bit", "start", "end", "fraction", "protein")
df$chr <- factor(as.character(df$chr),
                 levels = c(paste("chr", c(as.character(1:22), "X", "Y", "M"), sep = ""), "genome"))
sorted.index <- order(as.numeric(substring(df$chr, 4)))
df <- df[sorted.index,]
ggplot(data = df[(df$bit == 1 & df$chr != "genome"),],
       mapping = aes(x = chr, y = fraction, fill = protein)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous(limits = c(0, 6e-4)) +
   theme(legend.title = element_blank(),
         legend.position = c(0.75, 0.87),
         legend.key.size = unit(0.5, "cm"),
```

```
        legend.direction="horizontal")
```

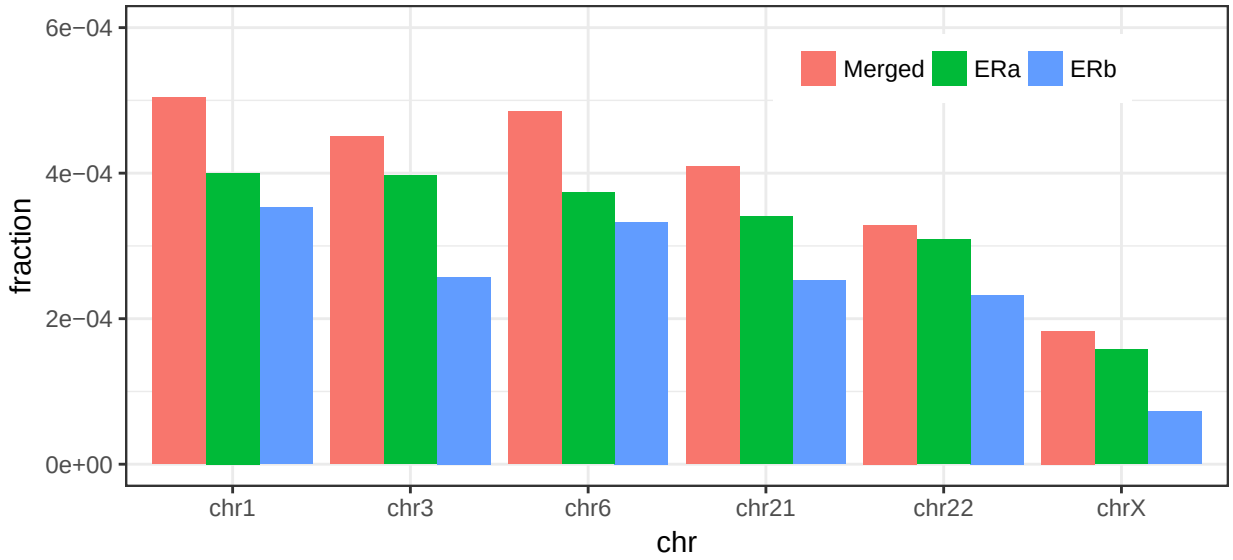
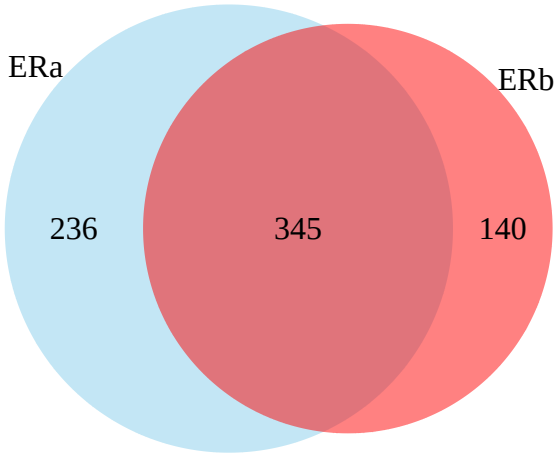


Figure 4. *holi*

(b)

Bedtools commands to preprocess data before R

```
# Calculates number of interval overlaps between ERa and ERb
# Reports multiple overlaps as a single occurrence
bedtools intersect -a ERa_hg18.bed -b ERb_hg18.bed -c > AtoBoverlap.bed
bedtools intersect -a ERb_hg18.bed -b ERa_hg18.bed -c > BtoAoverlap.bed
```




```
## (polygon[GRID.polygon.60], polygon[GRID.polygon.61], polygon[GRID.polygon.62], polygon[GRID.polygon.
```

8. Appendix

```
sessionInfo()
```

```
## R version 3.4.0 (2017-04-21)
```

```
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.2 LTS
##
## Matrix products: default
## BLAS: /usr/lib/openblas-base/libblas.so.3
## LAPACK: /usr/lib/libopenblasp-r0.2.18.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=es_ES.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=es_ES.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] VennDiagram_1.6.17  futile.logger_1.4.3 gridExtra_2.2.1
## [4] dplyr_0.5.0         reshape2_1.4.2      cowplot_0.7.0
## [7] ggplot2_2.2.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.10         knitr_1.15.1         magrittr_1.5
##  [4] munsell_0.4.3        colorspace_1.3-2     R6_2.2.1
##  [7] stringr_1.2.0        plyr_1.8.4           tools_3.4.0
## [10] gtable_0.2.0         DBI_0.6-1            lambda.r_1.1.9
## [13] htmltools_0.3.6      yaml_2.1.14          lazyeval_0.2.0
## [16] rprojroot_1.2        digest_0.6.12        assertthat_0.2.0
## [19] tibble_1.3.0         futile.options_1.0.0 evaluate_0.10
## [22] rmarkdown_1.5        labeling_0.3         stringi_1.1.5
## [25] compiler_3.4.0       scales_0.4.1         backports_1.0.5
```