Assignment 2 group 6

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

May 15, 2017

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 sense DNA genomic sequences (substituting Ts by Us). As AK002007 is transcribed on the reversed 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)?

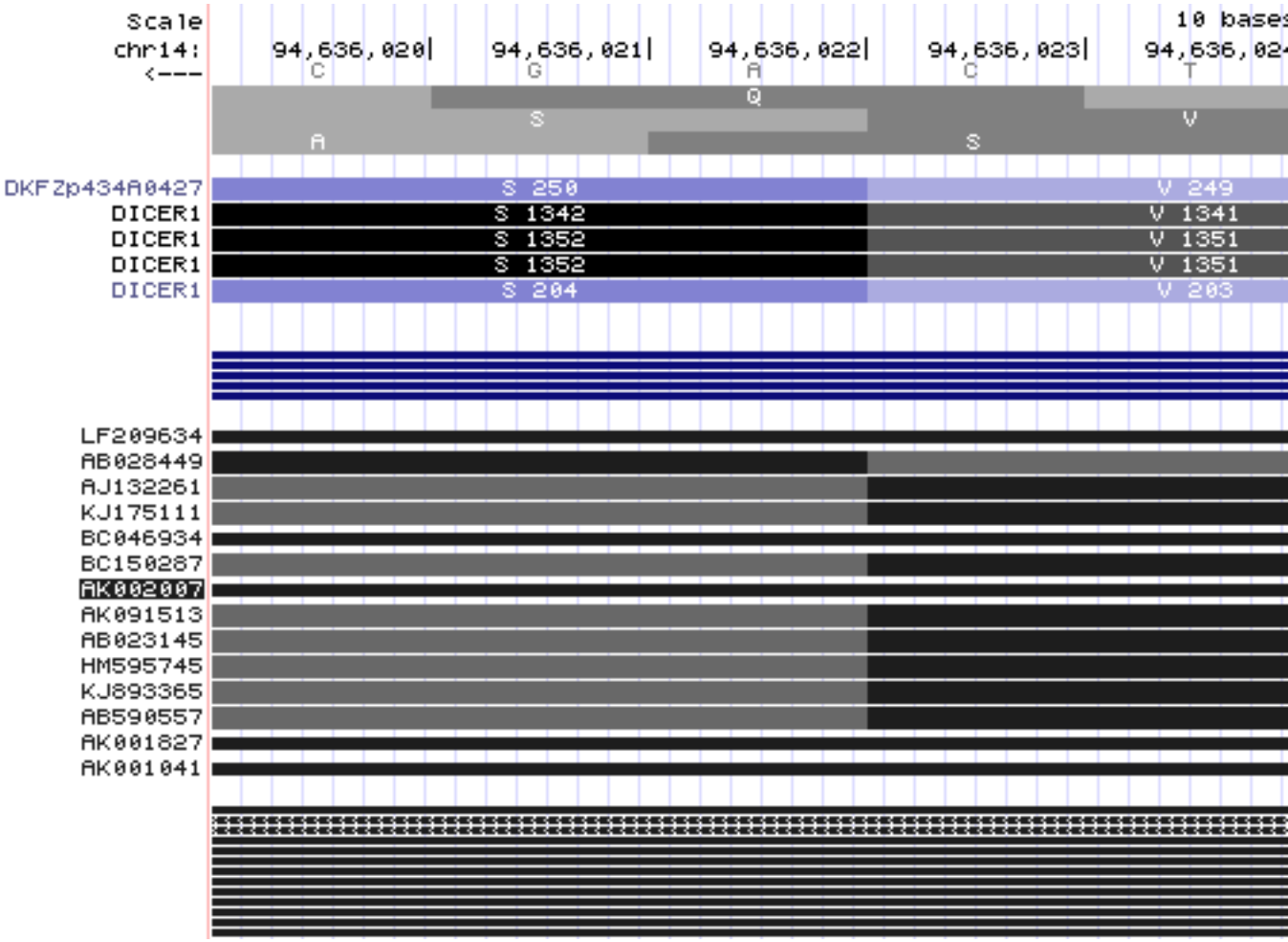
TODO Include capture from UCSCBrowser and Isabella's illustration showing the mess

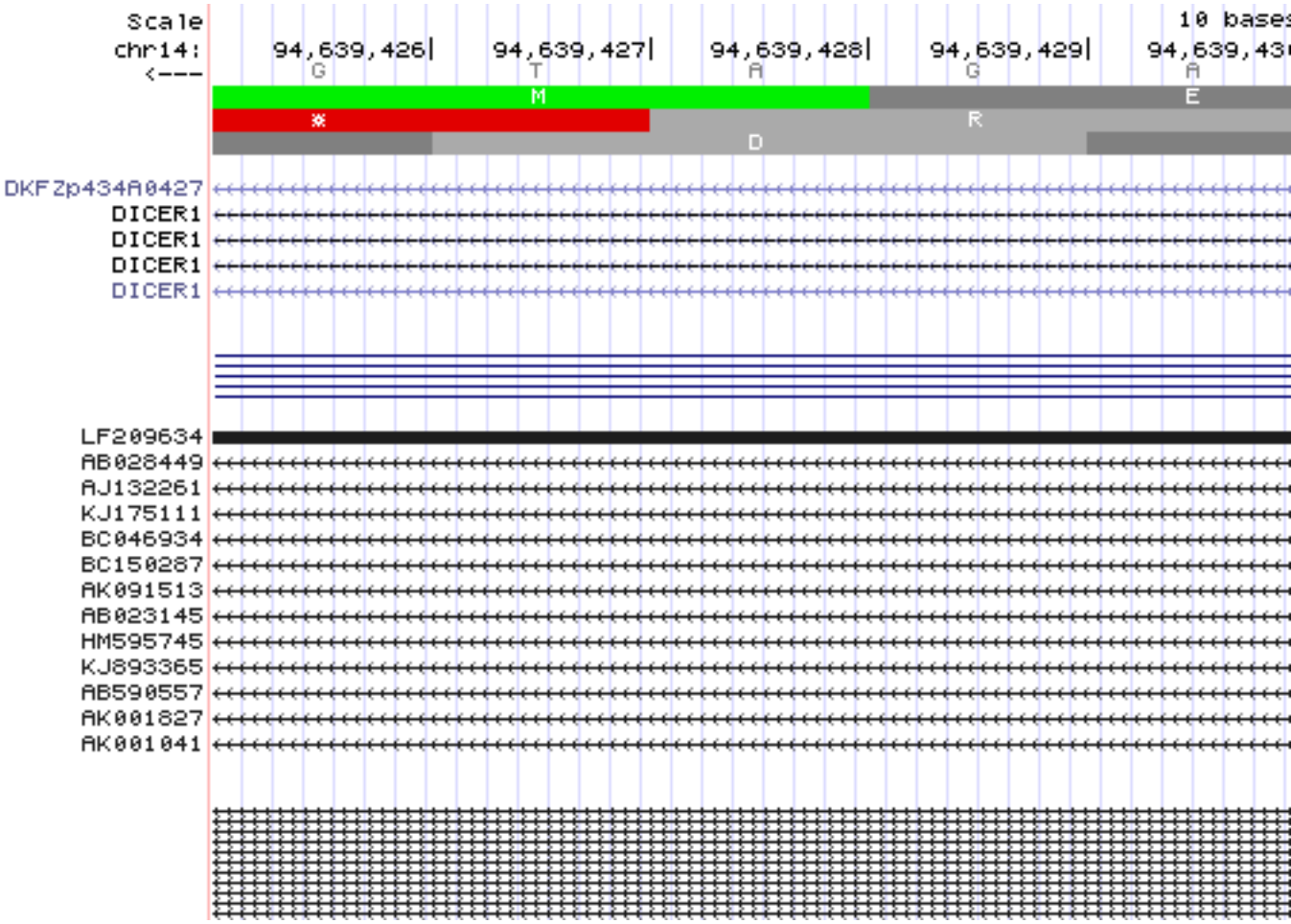
What is going on?

The sequencing process generated a truncated version of the mature mRNA starting at one of the last exons of the DICER1 gene. This truncated molecule bears the last 11 nucleotides of exon 21/27 ("gaagcaaaaag") and continues to hold the nucleotides in exon 22/27. Nevertheless, the aligner has mismapped these first nucleotides. Instead, the first 7 letters have been ignored and the remaining 4 letters have been mapped to the end of the intronic region between exons 21 and 22.

How could that happen?

This could be due to the aligner penalizing opening the intron gap just to align the leading 11 letters. This is only allowed because unfortunately the last 4 letters of exon 21 are identical to the last 4 letters in the intron, therefore providing to the aligner with the freedom to choose where to put these 4 bases


Scale
10 bases
chr14:
94,636,020
94,636,021
94,636,022
94,636,023
94,636,024
<---
C
G
A
C
T
Q
S
V
A
S
DKFZp434A0427
S 250
V 249
DICER1
S 1342
V 1341
DICER1
S 1352
V 1351
DICER1
S 1352
V 1351
DICER1
S 204
V 203
LF209634
AB028449
AJ132261
KJ175111
BC046934
BC150287
AK002007
AK091513
AB023145
HM595745
KJ893365
AB590557
AK001827
AK001041



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
```

R

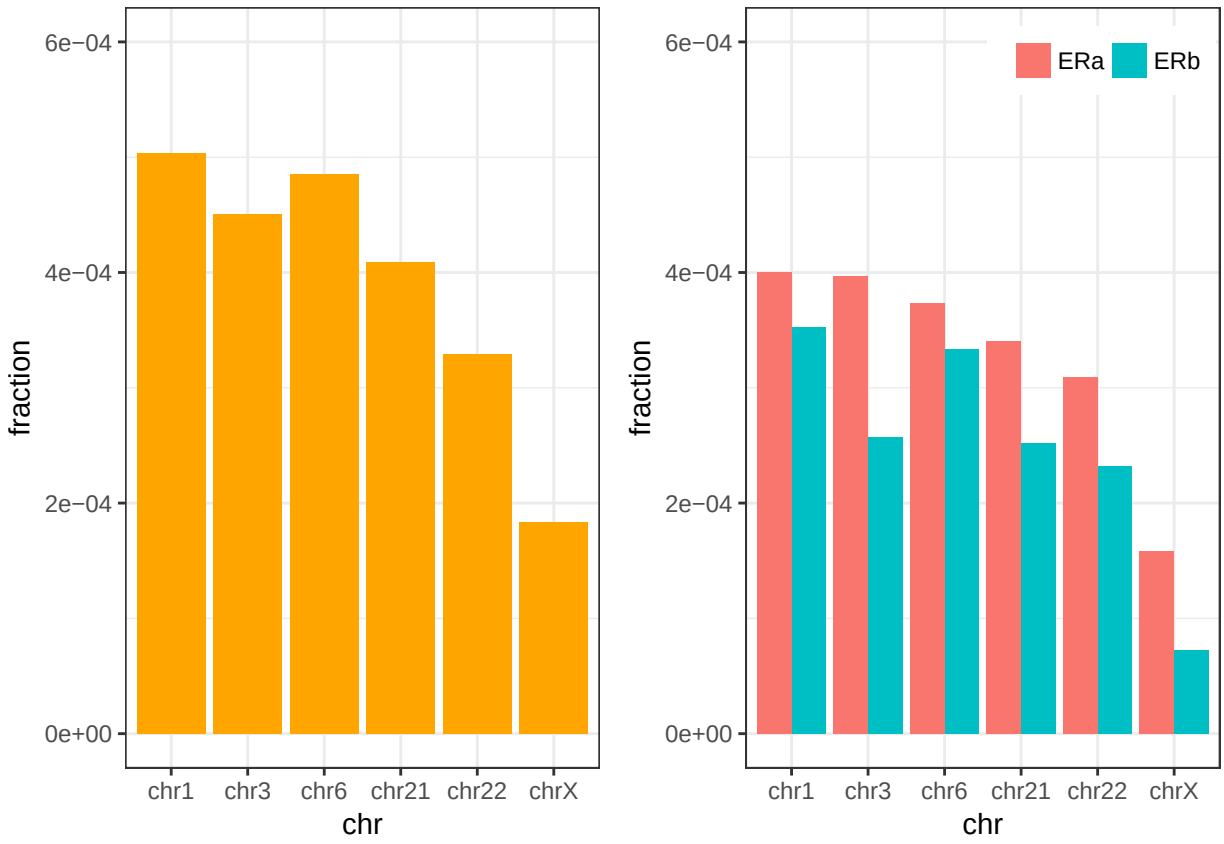
```
df <- read.table(file = "coverage.txt",
                 col.names = c("chr", "bit", "start", "end", "fraction"))
```

```
##      chr bit     start       end    fraction
## 1   chr1   0 247125142 247249719 0.999496000
## 2   chr1   1    124577 247249719 0.000503851
## 13 chr2   0 242951149 242951149 1.000000000
## 7   chr3   0 199411936 199501827 0.999549000
## 8   chr3   1     89891 199501827 0.000450577
```

```
## 14 chr4    0 191273063 191273063 1.000000000
## Saving 6.5 x 3 in image
```

Figure 1. A *Barplot showing the distribution of the exon counts. Even though most of the genes contain less than 60 exons, as many as 150 may be found in some of them.* **B** *Detail for genes with max. 20 exons. The mode can be visualized at 3-5 exons per gene (max found at 4). The number of exons per gene decreases steadily beyond it.*

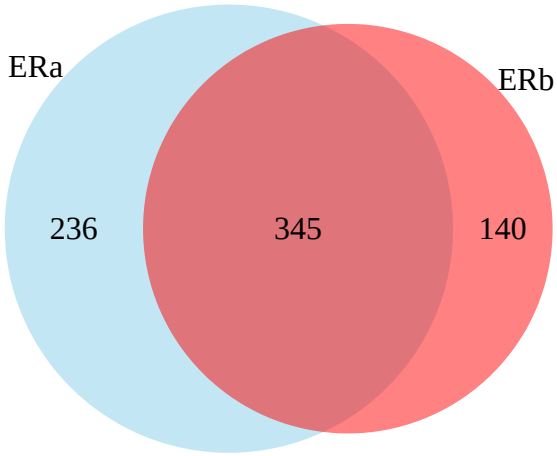
```
bedtools genomecov -i ERa_hg18.bed -g hg18_chrom_sizes.txt > ERa.txt
bedtools genomecov -i ERb_hg18.bed -g hg18_chrom_sizes.txt > ERb.txt
```



(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.163], polygon[GRID.polygon.164], polygon[GRID.polygon.165], polygon[GRID.polyg
```

8. Appendix

```
## 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 dplyr_0.5.0
## [4] reshape2_1.4.2      cowplot_0.7.0       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         png_0.1-7            DBI_0.6-1
## [13] lambda.r_1.1.9       htmltools_0.3.6      yaml_2.1.14
## [16] lazyeval_0.2.0       rprojroot_1.2        digest_0.6.12
## [19] assertthat_0.2.0     tibble_1.3.0         futile.options_1.0.0
## [22] evaluate_0.10        rmarkdown_1.5        labeling_0.3
## [25] stringi_1.1.5        compiler_3.4.0       scales_0.4.1
```

```
## [28] backports_1.0.5
```