Hw2

Home work 2 - to be done as groups

Names: Adrian Ramon Santonja, Jonas B H Andersen, Ismael Rodriguez Palomo, Steen J. Oestergaard, Luiza Czerwinska.

Group: 3

For deadlines etc, see Absalon.

Question 1: Dicer dissected

The human DICER1 gene encodes an important ribonuclease, involved in miRNA and siRNA processing. Several mRNAs representing this gene have been mapped to the human genome (March 2006 assembly). We will look closer at one of them with the accession number AK002007.

a) What are the first five genomic nucleotides that are read by RNA polymerae II from this transcript?

Answer: The first 5 genomic nucleotides seen from the UCSC-genome browser is: *AAAGG*

This is seen on the following screenshot (Fig 1.) with the first exon starting on the right and running to the left. The sequence CCTTT is reserve complemented and gives AAAGG.

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

The first 5 nucleotides from the GenBank sequence is GAAGCAA. This is seen in the screenshot on figure 2.

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

The discrepancy is hard to explain, but we have a theory. Looking at the GenBank entry we can see that the sequences are found by oligo-capping. In this method a cDNA library is constructed by removal of the 5'-Cap and insertion of a small synthetic oligo. This sequence could also show up in the sequencing and be shown in the genbank, but removed when aligned to the genome in the UCSC browser.

Source: 1. Suzuki, Y., Yoshitomo-Nakagawa, K., Maruyama, K., Suyama, A. & Sugano, S. Construction and characterization of a full length-enriched and a 5???-end-enriched cDNA library. Gene 200, 149???156 (1997).

Question 2: ERA and ERB

Our collaborators designed a ChIP study using so-called tilling arrays (an outdated technique these days, but the top of the pop at the time: see http://en.wikipedia.org/wiki/Tiling_array): one for estrogen receptor alpha (ERA), one for estrogen receptor beta (ERB). All the sites are stored in BED files respectively for two ERs. These are now available in the homework directory, and are both mapped on hg18 genome. The

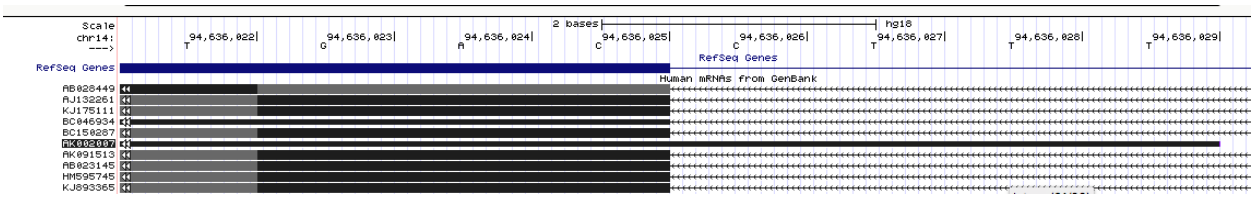


Figure 1: UCSC screenshot, showing the five first bases of transcription of AK002007

```
                     Endoribonuclease Dicer (EC 3.1.26.-)
                     /codon_start=1
                     /protein_id="BAG51002.1"
                     /translation="MVVSIFDPPVNWLPPGYVVNQDKSNTDKWEKDEMTKDCMLANGK
                     LDEDYEEEDEEEESLMWRAPKEEADYEDDFLEYDQEHIRFIDNMLMGSGAFVKKISLS
                     PFSTTDSAYEWKMPKKSSLGSMPFSSDFEDFDYSSWDAMCYLDPSKAVEEDDFVVGFW
                     NPSEENCGVDTGKQSISYDLHTEQCIADKSIADCVEALLGCYLTSCGERAAQLFLCSL
                     GLKVLPVIKRTDREKALCPTRENFNSQQKNLSVSCAAASVASSRSSVLKDSEYGCLKI
                     PPRCMFDHPDADKTLNHLISGFENFEKKINYRFKNKAYLLQAFTHASYHYNTITDCYQ
                     RLEFLGDAILDYLITKHLYEDPRQHSPGVLTDLRSALVNNTIFASLAVKYDYHKYFKA
                     VSPELFHVIDDFVQFQLEKNEMQGMDSELRRSEEDEEKEEDIEVPKAMGDIFESLAGA
                     IYMDSGMSLETVWQVYYPMMRPLIEKFSANVPRSPVRELLEMEPETAKFSPAERTYDG
                     KVRVTVEVVGKGKFKGVGRSYRIAKSAAARRALRSLKANQPQVPNS"
ORIGIN
        1 gaagcaaaaa ggtcagcaac tgtaatctgt atcgccttgg aaaaaagaag ggactaccca
       61 gccgcatggt ggtgtcaata tttgatcccc ctgtgaattg gcttcctcct ggttatgtag
      121 taaatcaaga caaaagcaac acagataaat gggaaaaaga tgaaatgaca aaagactgca
      181 tgctggcgaa tggcaaactg gatgaggatt acgaggagga ggatgaggag gaggagagcc
      241 tgatgtggag ggctccgaag gaagaggctg actatgaaga tgatttcctg gagtatgatc
      301 aggaacacat cagatttata gataatatgt taatggggtc aggagctttt gtaaagaaaa
      361 tctctctttc tccttttca accactgatt ctgcatatga atggaaaatg cccaaaaaat
      421 cctccttagg tagtatgcca ttttcatcag attttgagga ttttgactac agctcttggg
      481 atgcaatgtg ctatctggat cctagcaaag ctgttgaaga agatgacttt gtggtggggt
      541 tctggaatcc atcagaagaa aactgtggtg ttgacacggg aaagcagtcc atttcttacg
      601 acttgcacac tgagcagtgt attgctgaca aaagcatagc ggactgtgtg gaagccctgc
      661 tgggctgcta tttaaccagc tgtggggaga gggctgctca gcttttcctc tgttcactgg
      721 ggctgaaggt gctcccggta attaaaagga ctgatcggga aaaggccctg tgccctactc
      781 gggagaattt caacagccaa caaaagaacc tttcagtgag ctgtgctgct gcttctgtgg
      841 ccagttcacg ctcttctgta ttgaaagact cggaatatgg ttgtttgaag attccaccaa
      901 gatgtatgtt tgatcatcca gatgcagata aaacactgaa tcaccttata tcggggtttg
      961 aaaattttga aaagaaaatc aactacagat tcaagaataa ggcttacctt ctccaggctt
     1021 ttacacatgc ctcctaccac tacaatacta tcactgattg ttaccagcgc ttagaattcc
     1081 tgggagatgc gattttggac tacctcataa ccaagcacct ttatgaagac ccgcggcagc
     1141 actccccggg ggtcctgaca gacctgcggt ctgccctggt caacaacacc atctttgcat
     1201 cgctggctgt aaagtacgac taccacaagt acttcaaagc tgtctctcct gagctcttcc
     1261 atgtcattga tgactttgtg cagtttcagc ttgagaagaa tgaaatgcaa ggaatggatt
     1321 ctgagcttag gagatctgag gaggatgaag agaaagaaga ggatattgaa gttccaaagg
     1381 ccatgggggа tatttttgag tcgcttgctg gtgccattta catggatagt gggatgtcac
     1441 tggagacagt ctggcaggtg tactatccca tgatgcggcc actaatagaa aagttttctg
     1501 caaatgtacc ccgttcccct gtgcgagaat tgcttgaaat ggaaccagaa actgccaaat
     1561 ttagcccggc tgagagaact tacgacggga aggtcagagt cactgtggaa gtagtaggaa
     1621 aggggaaatt taaaggtgtt ggtcgaagtt acaggattgc caaatctgca gcagcaagaa
     1681 gagccctccg aagcctcaaa gctaatcaac ctcaggttcc caatagctga aaccgctttt
     1741 taaaattcaa aacaagaaac
//
```

Figure 2: Screenshot from the GenBank databse of AK002007. The 7 nucleotides which are different between the UCSC browser and the GenBank entry are highlighted in blue.

current situation is that we know to some degree what ERA does, but not what ERB does (there are some evidence that they share some functions, but not all). So, we need bigger experiments and better statistics.

a) Using BEDtools within Linux: What is the genome coverage (% of base pair covered at each chromosome) for ERB and ERA sites? If you need a file with chromosome sizes for hg18, it included in the assignment: hg18_chrom_sizes.txt. Plot the fractions for all chromosomes as a single barplot in R. Briefly comment the results. Is there anything particularly surprising? Try to explain the outcome (biological and/or experimental setup explanations)?

We start by sorting initial bed files to be used for further analysis

```
sort -k1,1 -k2,2n  ERa_hg18.bed  > sorted_ERa_hg18.bed # Sort ERa
sort -k1,1 -k2,2n  ERb_hg18.bed  > sorted_ERb_hg18.bed # Sort ERb
```

We anlyse the hg18 genome, for the fraction of nuclotides that are covered by either atleast 1 or no ERa or ERb sites at each chromosome and the wholde genome.

```
nice bedtools genomecov -i sorted_ERa_hg18.bed -g hg18_chrom_sizes.txt -max 1 >
  ERa_genomecoverage.bed
nice bedtools genomecov -i sorted_ERb_hg18.bed -g hg18_chrom_sizes.txt -max 1 >
  ERb_genomecoverage.bed
```

Files are transferred to R, first using FileZilla to extract files to computer to be used in R.

```
ERa_genome_coverage <- as.data.frame(read.table("ERa_genomecoverage.bed",header = FALSE,
                                    sep="\t",stringsAsFactors=FALSE, quote="")) #Read data
colnames(ERa_genome_coverage) <- c("Chromosome","Depth_of_Coverage","Nr_of_Bases",
                                   "Size_chr","Fraction_of_Coverage") #add column names
ERa_genome_coverage
```

```
##    Chromosome Depth_of_Coverage Nr_of_Bases   Size_chr
## 1        chr1                 0   247150807  247249719
## 2        chr1                 1       98912  247249719
## 3       chr21                 0    46928331   46944323
## 4       chr21                 1       15992   46944323
## 5       chr22                 0    49676075   49691432
## 6       chr22                 1       15357   49691432
## 7        chr3                 0   199422735  199501827
## 8        chr3                 1       79092  199501827
## 9        chr6                 0   170836223  170899992
## 10       chr6                 1       63769  170899992
## 11       chrX                 0   154889288  154913754
## 12       chrX                 1       24466  154913754
## 13       chr2                 0   242951149  242951149
## 14       chr4                 0   191273063  191273063
## 15       chr5                 0   180857866  180857866
## 16       chr7                 0   158821424  158821424
## 17       chr8                 0   146274826  146274826
## 18       chr9                 0   140273252  140273252
## 19      chr10                 0   135374737  135374737
## 20      chr11                 0   134452384  134452384
## 21      chr12                 0   132349534  132349534
## 22      chr13                 0   114142980  114142980
## 23      chr14                 0   106368585  106368585
## 24      chr15                 0   100338915  100338915
## 25      chr16                 0    88827254   88827254
## 26      chr17                 0    78774742   78774742
```

```
## 27      chr18                 0    76117153   76117153
## 28      chr19                 0    63811651   63811651
## 29      chr20                 0    62435964   62435964
## 30       chrY                 0    57772954   57772954
## 31       chrM                 0       16571      16571
## 32     genome                 0  3080138463 3080436051
## 33     genome                 1      297588 3080436051
##    Fraction_of_Coverage
## 1           9.99600e-01
## 2           4.00049e-04
## 3           9.99659e-01
## 4           3.40659e-04
## 5           9.99691e-01
## 6           3.09047e-04
## 7           9.99604e-01
## 8           3.96447e-04
## 9           9.99627e-01
## 10          3.73136e-04
## 11          9.99842e-01
## 12          1.57933e-04
## 13          1.00000e+00
## 14          1.00000e+00
## 15          1.00000e+00
## 16          1.00000e+00
## 17          1.00000e+00
## 18          1.00000e+00
## 19          1.00000e+00
## 20          1.00000e+00
## 21          1.00000e+00
## 22          1.00000e+00
## 23          1.00000e+00
## 24          1.00000e+00
## 25          1.00000e+00
## 26          1.00000e+00
## 27          1.00000e+00
## 28          1.00000e+00
## 29          1.00000e+00
## 30          1.00000e+00
## 31          1.00000e+00
## 32          9.99903e-01
## 33          9.66058e-05
```

```
ERb_genome_coverage <- as.data.frame(read.table("ERb_genomecoverage.bed",header = FALSE,
                                    sep="\t",stringsAsFactors=FALSE, quote="")) #Read data
colnames(ERb_genome_coverage) <- c("Chromosome","Depth_of_Coverage","Nr_of_Bases",
                                   "Size_chr","Fraction_of_Coverage") #add column names
ERa_genome_coverage
```

```
##    Chromosome Depth_of_Coverage Nr_of_Bases   Size_chr
## 1        chr1                 0   247150807  247249719
## 2        chr1                 1       98912  247249719
## 3       chr21                 0    46928331   46944323
## 4       chr21                 1       15992   46944323
## 5       chr22                 0    49676075   49691432
## 6       chr22                 1       15357   49691432
```

```
## 7        chr3                0   199422735  199501827
## 8        chr3                1       79092  199501827
## 9        chr6                0   170836223  170899992
## 10       chr6                1       63769  170899992
## 11       chrX                0   154889288  154913754
## 12       chrX                1       24466  154913754
## 13       chr2                0   242951149  242951149
## 14       chr4                0   191273063  191273063
## 15       chr5                0   180857866  180857866
## 16       chr7                0   158821424  158821424
## 17       chr8                0   146274826  146274826
## 18       chr9                0   140273252  140273252
## 19      chr10                0   135374737  135374737
## 20      chr11                0   134452384  134452384
## 21      chr12                0   132349534  132349534
## 22      chr13                0   114142980  114142980
## 23      chr14                0   106368585  106368585
## 24      chr15                0   100338915  100338915
## 25      chr16                0    88827254   88827254
## 26      chr17                0    78774742   78774742
## 27      chr18                0    76117153   76117153
## 28      chr19                0    63811651   63811651
## 29      chr20                0    62435964   62435964
## 30       chrY                0    57772954   57772954
## 31       chrM                0       16571      16571
## 32     genome                0  3080138463 3080436051
## 33     genome                1      297588 3080436051
##    Fraction_of_Coverage
## 1           9.99600e-01
## 2           4.00049e-04
## 3           9.99659e-01
## 4           3.40659e-04
## 5           9.99691e-01
## 6           3.09047e-04
## 7           9.99604e-01
## 8           3.96447e-04
## 9           9.99627e-01
## 10          3.73136e-04
## 11          9.99842e-01
## 12          1.57933e-04
## 13          1.00000e+00
## 14          1.00000e+00
## 15          1.00000e+00
## 16          1.00000e+00
## 17          1.00000e+00
## 18          1.00000e+00
## 19          1.00000e+00
## 20          1.00000e+00
## 21          1.00000e+00
## 22          1.00000e+00
## 23          1.00000e+00
## 24          1.00000e+00
## 25          1.00000e+00
## 26          1.00000e+00
```

```
## 27          1.00000e+00
## 28          1.00000e+00
## 29          1.00000e+00
## 30          1.00000e+00
## 31          1.00000e+00
## 32          9.99903e-01
## 33          9.66058e-05
```

A fraction of 9.66058e-05 of the genomes basepairs are covered by at least one ERa site For ERb sites this fraction is 7.46787e-05.

We next plot the fraction of coverage for each chromosome of respectively ERa and ERb

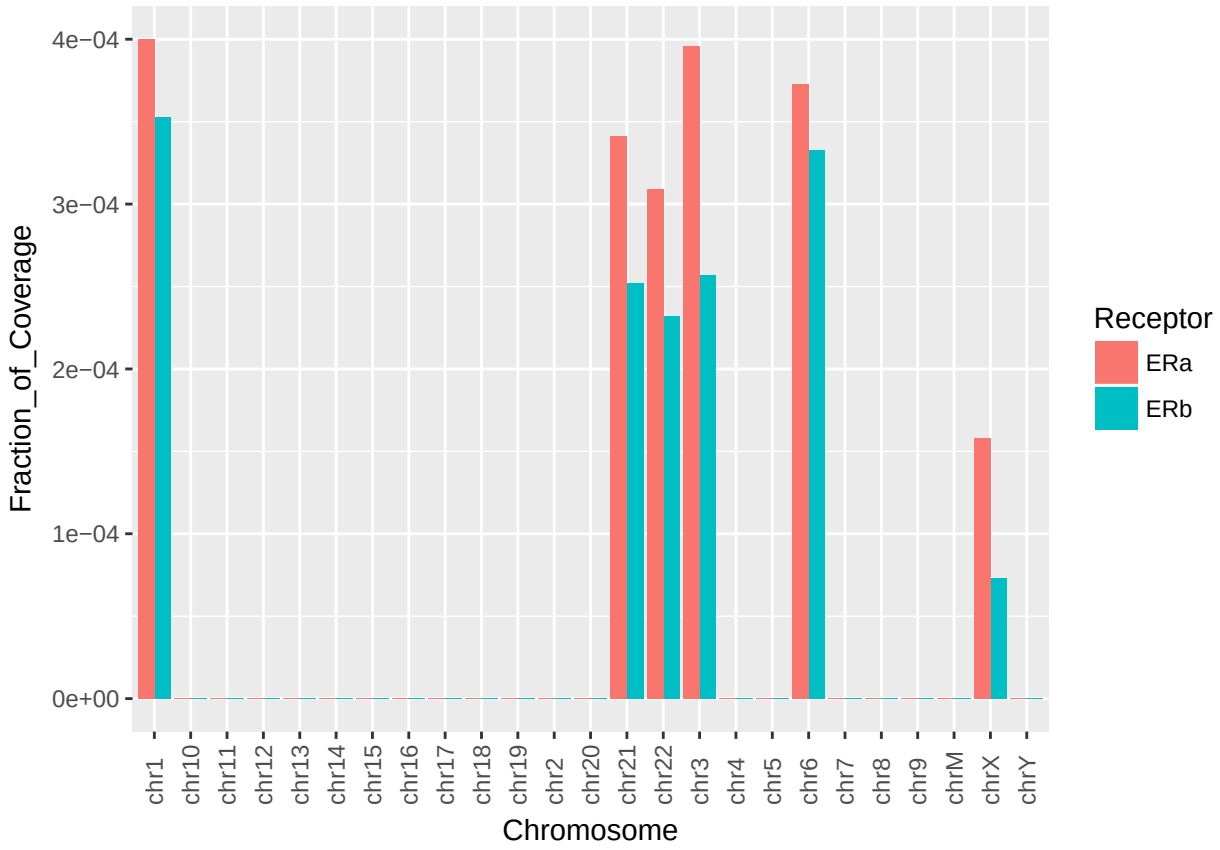
```
# both tibbles are modified to include the fraction of coverage at each chromosome
#(even if it is 0), filter away the whole genome coverage and duplicates of chromosomes,
#and adding a column with the respective receptor.

Simple_ERa <- ERa_genome_coverage %>% mutate( Fraction_of_Coverage = 1 - Fraction_of_Coverage) %>%
  filter(Depth_of_Coverage == "0" , str_detect(Chromosome, "chr")) %>% mutate(Receptor="ERa")

Simple_ERb <- ERb_genome_coverage %>% mutate( Fraction_of_Coverage = 1 - Fraction_of_Coverage) %>%
  filter(Depth_of_Coverage == "0" , str_detect(Chromosome, "chr")) %>% mutate(Receptor="ERb")

Merged_Chromosome_Coverage <- rbind(Simple_ERa,Simple_ERb)
# Merge ERa and ERb to plot both receptors coverage in same graph.

ggplot(Merged_Chromosome_Coverage) + geom_bar(aes(x=Chromosome, y=Fraction_of_Coverage,
                      fill=Receptor), stat = "identity", position="dodge") +
  theme(axis.text.x = element_text(angle = 90, vjust=0.5, hjust=0))
```




```
# plot the fraction of coverage at each chromosome for both ERa and ERb
```

A large portion of the chromosomes do not show any DNA which the estrogen receptors bind to, not even noise. This is most likely due to an experimental decision to not use probes which correspond to regions in these chromosomes in the tilling array. If these probes were utilized we would have been able to observe noise. It is not possible to have a probe for each gene in the genome, and most often experiments rely on multiple random probes for each gene they wish to investigate.

The graph indicate that ERa sites has a higher fraction of exons covering the chromosomes, meaning that a larger fraction of the DNA binds to ERa compared to ERb. However, since we only have one replicate for each receptor, the experiment is highly susceptible to noise. Difference in concentrations of ERa and ERb, incomplete immunoprecipitation, incomplete removal of DNA fragments or incomplete hybridization are all possible sources that would disrupt the experiment.

b) Again, using BEDtools in Linux: How many ERA sites do/do not overlap ERB sites, and vice versa? Show the Linux commands and then a Venn diagram summarizing the results. The Venn diagram can be made in R using one of many venn diagram packages, but you can also make it in any drawing program.

We start by counting the ERa or ERb sites that overlaps either ERb or ERa sites respectively.

```
bedtools intersect -a ERa_hg18.bed -b ERb_hg18.bed -c > ERa_count.bed
# each ERa site is counted for the number of ERb sites it overlaps (1 or 0)

bedtools intersect -a ERb_hg18.bed -b ERa_hg18.bed -c > ERb_count.bed
# each ERb site is counted for the number of ERa sites it overlaps (1 or 0)

#seperate the sites that overlaps from those that does not
awk '{ if ($4 == 1) { print } }' ERa_count.bed > ERa_overlap_ERb
```

```
awk '{ if ($4 == 0) { print } }' ERa_count.bed > ERa_NOT_overlap_ERb
awk '{ if ($4 == 1) { print } }' ERb_count.bed > ERb_overlap_ERa
awk '{ if ($4 == 0) { print } }' ERb_count.bed > ERb_NOT_overlap_ERa
```

To find the number of sites that overlaps or does not overlap, we simply count the number of lines in the seperated files.

```
#count lines in each dataset
wc -l ERa_overlap_ERb ERa_NOT_overlap_ERb ERb_overlap_ERa ERb_NOT_overlap_ERa
```

```
##   345 ERa_overlap_ERb
##   236 ERa_NOT_overlap_ERb
##   345 ERb_overlap_ERa
##   140 ERb_NOT_overlap_ERa
##  1066 total
```

We use R to make the Venn-diagrams.

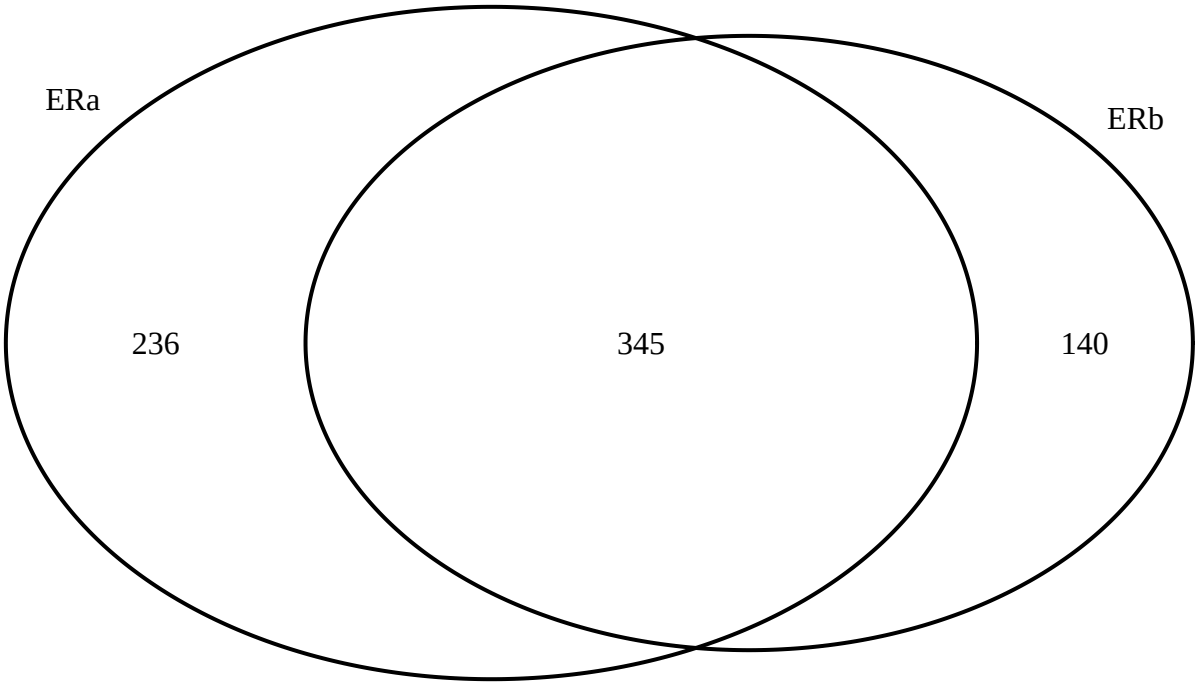
```
#install.packages("VennDiagram")
library(VennDiagram) #Install package
```

```
## Loading required package: grid
```

```
## Loading required package: futile.logger
```

```
#Number of total sites for each receptor and number of overlapping sites
ERa_sites <- 345 + 236
ERb_sites <- 345 + 140
Overlapping_Sites <- 345

#Draw Venn Diagram
draw.pairwise.venn(area1=ERa_sites,area2 = ERb_sites,cross.area = Overlapping_Sites,
                   category = c("ERa", "ERb"))
```




```
## (polygon[GRID.polygon.65], polygon[GRID.polygon.66], polygon[GRID.polygon.67], polygon[GRID.polygon.
```

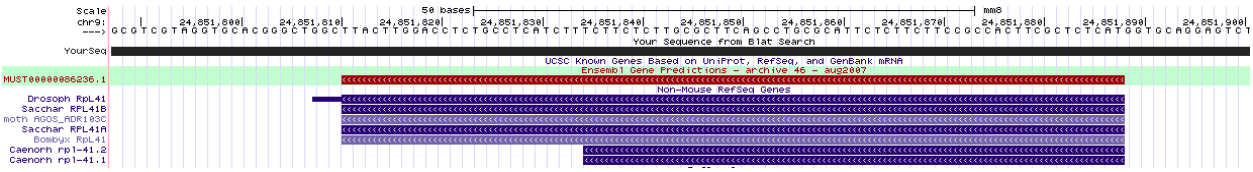


Figure 3: USCS screenshot from the mouse genomic region.

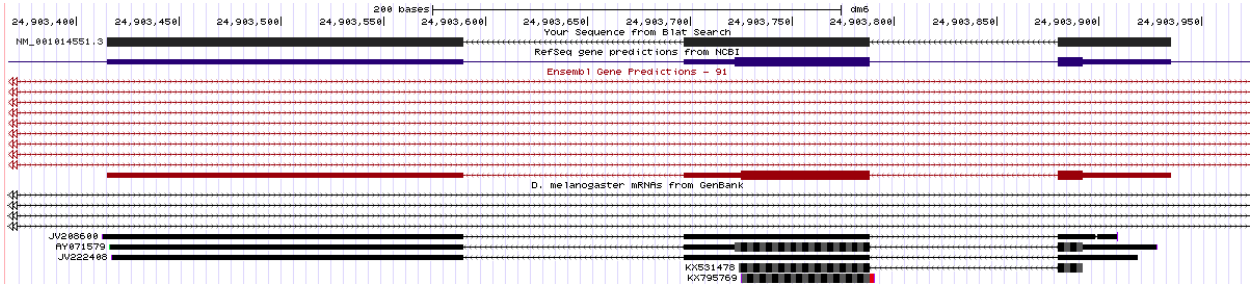


Figure 4: USCS screenshot from the fly BLAT result.

The Venn-diagram indicate that larger fraction of ERb site overlab ERa sites, compared to ERa sites overlaping ERb sites. The large overlap of homologous binding sites, could indicate similar function for the two receptors.

Question 3: Ribosomal Gene (*)

Your group just got this email from a frustrated fellow student:

My supervisor has found something he thinks is a new ribosomal protein gene in mouse. It is at chr9:24,851,809-24,851,889, assembly mm8. His arguments for this are a) It has high conservation in other species because ribosomal protein genes from other species map to this mouse region b) And they are all called Rpl41 in the other species (if you turn on the other Refseq you see this clearly in fly and other species).

But, I found out that if you take the fly refseq sequence mentioned above (from Genbank) and BLAT this to the fly genome, you actually get something that looks quite different from the one in the mouse genome. How can this be? Is the mouse gene likely to be real? If not, why? (Maximum 20 lines, plus possibly genome browser pictures)

Answer:

This is the genomic region in mouse in which we can see that there is a conserved gene in fly and some other species, all of them ribosomal proteins (RpL41) (Fig 3).

We can see that the mRNA from the gene we are looking for does not seem to be spliced, but when we take the sequence from the fly and BLAT it against its own genome, we get that the mRNA from gene we find (RpL41) is spliced here, while was not in the possible gene from mouse (Fig. 4).

In order to find out a reason for that, we took the mouse genomic sequence and BLAT against the mouse genome (mm8). The hits we got are in fig 5.

We observe that the same phenomenon of pieces of RPL41 exons being inserted into the genome, is happening in multiple places in different chromosomes. Now focusing on the sixth hit, we see that the span (574) is significantly higher than in the others (78). If we go to the genome browser for that region, we can see that there is an actual RefSeq gene at that location, a ribosomal protein - The full Rpl41 (fig. 6).

We also took a extended sequence, 200 bp from both sides, and performed the BLAT alignment again, getting a longer alignment for the same ribosomal gene (fig. 7).

QUERY	SCORE	START	END	QSIZE	IDENTITY	CHRO	STRAND	START	END	SPAN
YourSeq	78	1	78	78	100.0%	9	-	24851811	24851888	78
YourSeq	78	1	78	78	100.0%	13	-	112714412	112714489	78
YourSeq	78	1	78	78	100.0%	10	-	43144839	43144916	78
YourSeq	78	1	78	78	100.0%	14	+	104567548	104567625	78
YourSeq	76	1	78	78	98.8%	16	-	3932208	3932285	78
YourSeq	76	1	78	78	100.0%	10	-	127951336	127951909	574
YourSeq	74	1	78	78	97.5%	17	-	12680975	12681052	78
YourSeq	74	1	78	78	97.5%	1	-	51407807	51407884	78
YourSeq	74	1	78	78	97.5%	11	+	12548094	12548171	78
YourSeq	72	1	78	78	96.2%	2	-	113896864	113896941	78
YourSeq	72	1	78	78	92.0%	15	-	28003397	28003471	75
YourSeq	70	1	78	78	94.9%	18	-	10274594	10274671	78
YourSeq	68	1	78	78	94.9%	13	-	55192053	55192145	93
YourSeq	68	1	78	78	93.6%	2	+	150618774	150618851	78
YourSeq	66	10	78	78	98.6%	17	-	6779921	6779993	73
YourSeq	66	1	78	78	92.4%	16	-	38488929	38489006	78
YourSeq	66	11	78	78	98.6%	12	-	81699245	81699312	68
YourSeq	64	1	78	78	91.1%	1	-	147698430	147698507	78
YourSeq	64	10	78	78	97.2%	17	+	7797103	7797175	73
YourSeq	64	1	78	78	91.1%	16	+	96214019	96214096	78
YourSeq	61	1	78	78	88.8%	4	+	131819713	131819788	76
YourSeq	59	1	78	78	82.5%	11	-	97117712	97117785	74
YourSeq	58	1	78	78	82.9%	11	-	20174315	20174390	76
YourSeq	55	1	63	78	93.7%	4	+	134649234	134649296	63
YourSeq	49	18	76	78	91.6%	11	-	6176859	6176917	59
YourSeq	45	30	78	78	96.0%	6	+	70898452	70898500	49
YourSeq	42	28	75	78	93.8%	8	-	10717096	10717143	48
YourSeq	42	19	78	78	85.0%	6	-	107061992	107062051	60
YourSeq	42	26	75	78	87.8%	2	+	42065104	42065152	49
YourSeq	38	26	67	78	95.3%	7	+	57411777	57411818	42
YourSeq	35	1	37	78	97.3%	18	-	79300084	79300120	37
YourSeq	35	39	77	78	94.9%	11	+	43750972	43751010	39
YourSeq	28	45	78	78	84.9%	4	-	116541235	116541267	33
YourSeq	27	52	78	78	100.0%	6	-	28096491	28096517	27
YourSeq	25	1	25	78	100.0%	7	+	34131784	34131808	25
YourSeq	24	44	67	78	100.0%	7	-	75772459	75772482	24
YourSeq	21	47	67	78	100.0%	12	+	32829422	32829442	21
YourSeq	20	59	78	78	100.0%	6	-	100360374	100360393	20

Figure 5: List of hits from BLAT.

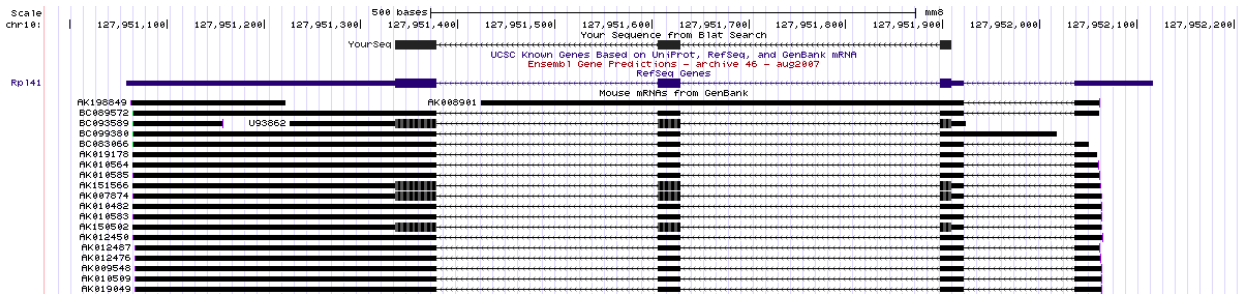


Figure 6: Genome browser showing the sixth hit.

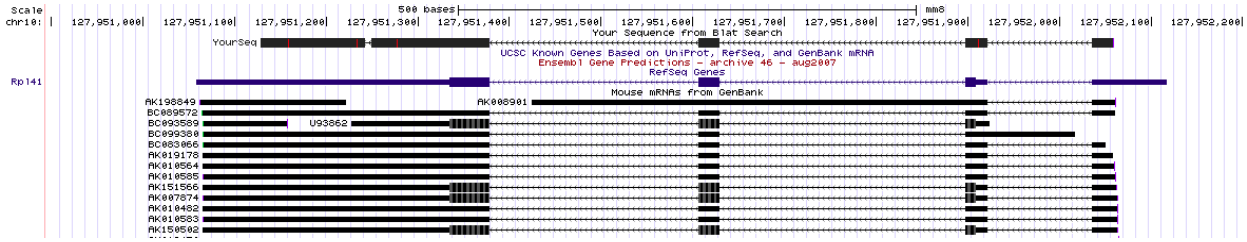


Figure 7: USCS screenshot from the mouse chromosomic region.

All in all, the explanation we have for what we see, is that the original sequence in mouse is likely to be a processed retrotransposon. This can happen when retro-transcription of the spliced mRNA of the actual ribosomic protein gene occurs and is followed by an insertion into the genome. The fact that we found many identical sequences in the genome also supports this explanation.

From searching in literature we see that this is a very common phenomenon for ribosomic protein genes [1]. It has been long thought that this was junk DNA, but now it is being investigated whether these sequences could have some function [2]. In our own research, we observe a high conservation in these sequences and also some ESTs overlapping with the original sequence. Usually pseudogenes are not coding any protein but they might perform some function as RNA altering protein expression of the original gene.

References:

1. Zhang Z, Harrison P, Gerstein M. Identification and Analysis of Over 2000 Ribosomal Protein Pseudogenes in the Human Genome. Genome Research. 2002;12(10):1466-1482. doi:10.1101/gr.331902.

2. Pink RC, Wicks K, Caley DP, Punch EK, Jacobs L, Francisco Carter DR. Pseudogenes: Pseudofunctional or key regulators in health and disease? RNA. 2011;17(5):792-798. doi:10.1261/rna.2658311.