Homework 3 - first submission

Group 7: Nynne Nymann, Peter Horskjr, Carlotta Porcelli, Max Tomlinson & Ke Zhai

30/5/2017

Part 1

Question 1: Calculating the mean signal of these two genes across the patient groups (HIV and non-HIV) shows that IRX3 always has higher signals than RXR. Can we then conclude that IRX3 is higher expressed in all these samples?

When taking these data into consideration, it is not possible for us to conclude that IRX3 is higher expressed in all samples. First of all the data should be normalized before we can analyse it. As we discussed in the genomic browser exercise earlier this course, IRX3 consists of extremely large CpG islands, and the increase in expression could be explained by the fact that the probe is more prone to bind to this sequence, making the comparison of expression levels invalid.

Question 2: Do a suitable statistical test for each row to find the differentially expressed genes (show the R code only ??? we will use the result in the next few questions)

```
rm(list=ls())
hiv <- read.table("normalized_data.txt", header=F)
colnames(hiv) <- c("HIV_1","HIV_2","HIV_3","HIV_4","HIV_5",
                   "Healthy_1","Healthy_2","Healthy_3","Healthy_4","Healthy_5")
```

To begin, data is checked to choose the most appropriate statistical test. In this case, data of gene expressions exhibit normality (not shown). Hence, we choose to perform a student t-test. Our hypotheses are: H0: The difference of the mean in gene expression in both HIV and healthy patients is the same HA: The difference of the mean in gene expression in both HIV and healthy patients is not the same

```
p_val <- sapply(1: nrow(hiv), function(x) t.test(hiv[x, 1:5], hiv[x, 6:10])$p.value)
```

Question 3: How many false positives would you expect for this experiment if you use a threshold of 0.05? How many genes do you actually get with a p-value less than 0.05?

We would expect the following number of false positives:

```
length(p_val)*0.05
```

```
## [1] 1114.15
```

Whereas, what resulted from our test:

```
sum(p_val<0.05)
```

```
## [1] 1911
```

Thus, we get a difference of 797 FPs when comparing expected versus real.

Question 4: The function p.adjust(p-values) can be used to correct for multiple testing. How many genes do you get with a p-value <0.2 when you use the Bonferroni correction? How many do you get with a FDR (Use the BH method) than 0.2. How many of these genes(FDR<0.2) would you expect to be false positives?

By using the p.adjust function in R, we obtain the following adjusted p-values:

```
p_val_bonf <- p.adjust(p_val,method="bonferroni")
numb_bonf <- sum(p_val_bonf<0.2)
numb_bonf
```

```
## [1] 0
```

```
p_val_FDR <- p.adjust(p_val,method="BH")
numb_FDR <- sum(p_val_FDR<0.2)
numb_FDR
```

```
## [1] 12
```

Thus, 0 and 12 genes have a p-value below 0.2 with Bonferroni and FDR adjustment, respectively. As regards to the fdr adjustment, we would expect 12x0.2 =~ 2 out of the 12 genes to be false positives.

Question 5: She also want to see how big the changes between the conditions are. So calculate the log2 foldchange for each gene. Report the fold changes for the genes with a FDR<0.2. Are there most up (Up in HIV) or down regulated genes in this subset? Comment on the size of the log2FCs

```
#calculating the fold-change by apply function:
mean_hiv <- sapply(1:nrow(hiv), function(x) mean(as.numeric(hiv[x,1:5])))
mean_con <- sapply(1:nrow(hiv), function(x) mean(as.numeric(hiv[x,6:10])))
foldchange_hiv <- log2(mean_hiv)-log2(mean_con)
foldchange_hiv[p_val_FDR<0.2]
```

```
##  [1] 0.03680802 0.33677927 0.06387247 0.15993496 0.60119418 0.33087497
##  [7] 0.08063546 0.09060348 0.08200741 0.17300186 0.09105046 0.16164736
```

Based on the log2-value: all the 12 genes have a positive value and hence upregulated in HIV.

Part 2

Question 2: For each condition calculate the average number of lines in the 3 'assembled transcripts'

The lines for transcripts WTD_1, WTD_2, WTD_3 are: 991, 1014, 991 and the average for the WildType condition is: (991 + 1014 + 991)/3 =~ 998

The lines for transcripts KD_1, KD_2, KD_3 are: 1004, 1015, 1064 and the average for the KnockDown condition is: (1004 + 1015 + 1064)/3 =~ 1027

Question 3: How many lines does the combined GTF file have? How does that compare to the individual transcriptomes? What does this suggests?

Merging the 'assembled transcripts' from the cufflinks and using the imported UCSC genes as Reference Annotation, the GTF file has 1207 lines. Cuffmerge gives rise to a GTF file containing 1207 lines. Cufflink

gives an output in the form of a transcriptome for each replica in each conditions (in this case for WT and KD). We then use cuffmerge to create a common set of transcripts - a combined transcriptome. Although most transcripts are shared between the individual transcriptomes, a minor pool of transcripts are only found in some of the sample which is why we see a difference in number of lines. The combined transcriptome has been uploaded on the genome browser UCSC.

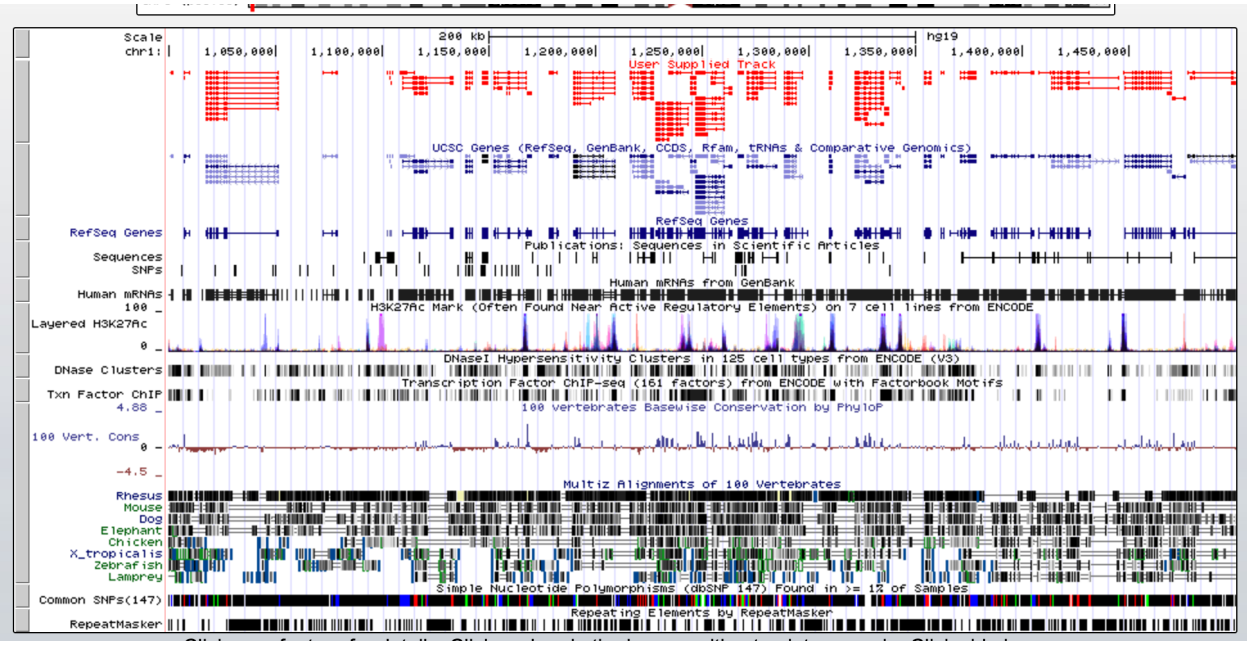

Figure 1: Combined transcriptome loaded on the UCSC genome browser. The User track is shown in red.

Question 4: Use 'Cuffdiff' to make the differential expression analysis between the two conditions (WT and KD). Why would be want to use bias correction?

In RNA-sequencing, there is a positional bias towards fragments are preferentially located towards either the start or end of the transcripts. Based on this, it is appropriate to use the function 'bias correction' in order to improve the expression analysis and for proper comparison of the samples analyzed. Reference: bias

Question 5: Specify what CuffDiff have tested in the 'Splicing differential expression testing'

The CuffDiff of the 'Splicing differential expression testing' results in a file listing, for each primary transcript, the amount of isoform switching detected among its isoforms. Only primary transcripts from which two or more isoforms are spliced are listed in this file'. reference: cuffdiff documentation

In our scenario it is shown how much differential splicing exists between isoforms of Wild Type and Knock Down.

Part 3

Question 1, 2 & 3: Read the supplied 3 CuffDiff result files and change the column name 'test_id' to 'transcript_id'

```
gene_diff_exp <- read.table("cuffdiff_gene_differential_expression.txt",
                            header = T)
colnames(gene_diff_exp)[which(names(gene_diff_exp) == "test_id")] <- "transcript_id"

gene_diff_exp_subset <- subset(gene_diff_exp,
  gene_diff_exp$value_1>0.0 | gene_diff_exp$value_2 > 0.0 )
gene_diff_exp_significant <- subset(gene_diff_exp, gene_diff_exp$significant =='yes')
nrow(gene_diff_exp_subset)
```

```
## [1] 26
```

```
nrow(gene_diff_exp_significant)
```

```
## [1] 12
```

```
transcript_diff_exp <- read.table("cuffdiff_transcript_differential_expression.txt",
                                  header = T)
colnames(transcript_diff_exp)[which(
  names(transcript_diff_exp) == "test_id")] <- "transcript_id"

transcript_diff_exp_subset <- subset(transcript_diff_exp,
  transcript_diff_exp$value_1>0.0 | transcript_diff_exp$value_2 > 0.0 )
transcript_diff_exp_significant <-
  subset(transcript_diff_exp, transcript_diff_exp$significant =='yes')
nrow(transcript_diff_exp_subset)
```

```
## [1] 98
```

```
nrow(transcript_diff_exp_significant)
```

```
## [1] 11
```

```
splicing_diff_exp <- read.table("cuffdiff_splicing_differential_expression.txt",
                                header = T)
```

The number of genes expressed is 26 and the number of expressed transcripts is 98. The number of significantly differentially expressed genes is 12 and the number of significantly differentially expressed transcripts is 11.

Question 4: How many rows does this new data.frame contain? How many columns?

```
transcript_diff_sub2 <- transcript_diff_exp_subset[
  c("transcript_id", "gene_id", "value_1","value_2")]
gene_diff_exp_sub2 <- gene_diff_exp_subset[
  c("gene_id", "gene", "value_1","value_2")]
transcript_gene_merged <-merge(gene_diff_exp_sub2, transcript_diff_sub2,
          by="gene_id", suffixes = c("gene","transcript"))
nrow(transcript_gene_merged)
```

```
## [1] 98
```

```
ncol(transcript_gene_merged)
```

```
## [1] 7
```

The number of rows of the merged file is 98 and the number of columns is 7.

Question 5: Calculate the Isoform Fraction values (IF values) and the corresponding dIF values?

```
hold1 <- vector('numeric')
hold2 <- vector('numeric')
for (i in 1:length(unique(transcript_diff_exp_subset$gene_id))){
  hold <- transcript_diff_exp_subset[transcript_diff_exp_subset$gene_id ==
          unique(transcript_diff_exp_subset$gene_id)[i],]
  hold1 <- c(hold1, hold$value_1 / sum(hold$value_1))
  hold2 <-c(hold2, hold$value_2 / sum(hold$value_2))
  hold3 = hold1 - hold2
  df_ifs_values <- rbind(hold1, hold2, hold3)
}
rownames(df_ifs_values) = c('IF1','IF2','DIF')
colnames(df_ifs_values ) = transcript_diff_exp_subset$transcript_id
df_ifs_values <- t(df_ifs_values)
```

We obtained several values corresponding to NAs. These NAs stem from the situation were neither the gene nor any transcripts are expressed in one of the two conditions, making in invalid to calculate the dIF when the gene expression is equal to zero. On this basis we cannot conclude that an isoform switch has occurred, but rather the gene expression is turned off. The NAs are removed to allow us to proceed the calculations.

Quesiton 6: What is the average (mean) and median dIF value? Compare the two values and discuss what it enables you to say about the distribution of dIF values.

```
dif_values <- na.omit(df_ifs_values[,3])
mean(dif_values)
```

```
## [1] -2.83513e-18
```

```
median(dif_values)
```

```
## [1] -0.0001255014
```

As in this case, where the mean and median is almost similar and both almost equal to zero, indicates that the majority of the genes in the data frame do not change in level of expression and hence a similar pool of genes are up and down-regulated. Illustrating the data in a histogram (not shown) shows a tendency to be normally distributed.

Question 7: Use R to subset the merged data.frame to only contain genes with potential isoform switching by identifying genes with dIF > +/- 0.25. Add the p_value from the 'Splicing differential expression testing' to the data.frame

```
transcript_gene_merged["DIF"] <- df_ifs_values[,3]
dif_value_checker <- abs(df_ifs_values[,3]) > 0.25
dif_value_checker[is.na(dif_value_checker)] <- FALSE

final_data_frame <- transcript_gene_merged[dif_value_checker,]
matching_on_gene_ids <- match(final_data_frame$gene_id, splicing_diff_exp$gene_id)
final_data_frame["p_value"] <- splicing_diff_exp$p_value[matching_on_gene_ids]
```

Question 8: Report the switch in the gene with the lowest p_value

```
gene_lowest_p_value <- final_data_frame[which.min(final_data_frame$p_value),]
gene_lowest_p_value[c('transcript_id', 'gene','DIF', 'p_value')]
```

```
##     transcript_id       gene       DIF p_value
## 11 TCONS_00000021 uc001aeo.3 0.3426093 0.00075
```

Question 9: Analyzing the gene with a switch

The gene uc001aeo.3 is from the assembly hg19/Feb. 2009 is a glycolipid transfer protein domain containing 1 (GLTPD1), mRNA. It mediates the intracellular transfer of ceramide-1- phosphate between organelle membranes and the cell membrane. ref:.

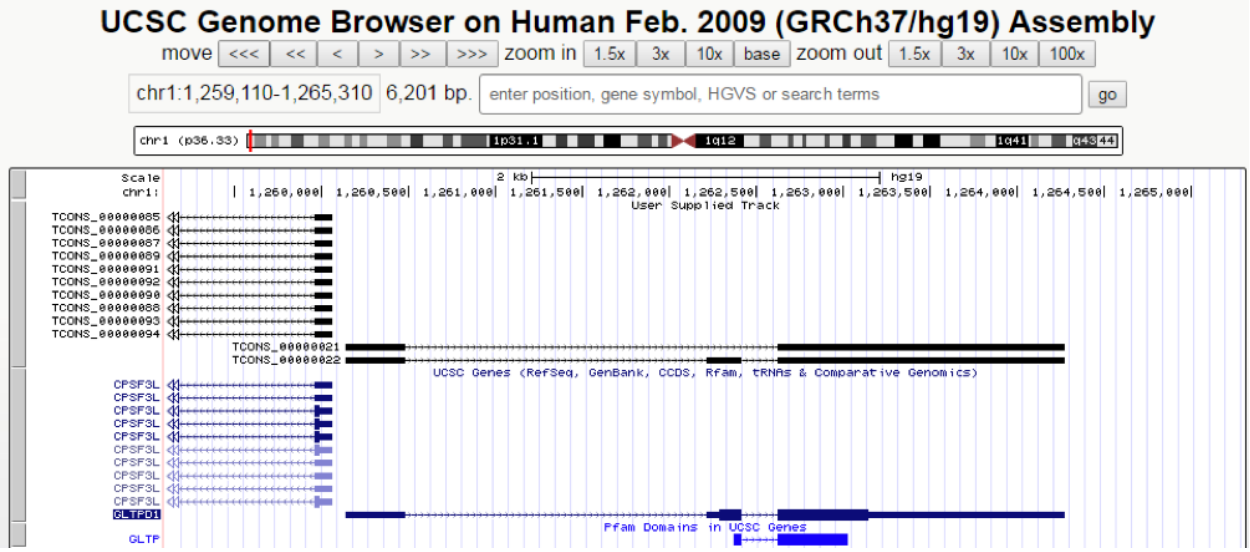

Figure 2: gene with a switch

As seen in the genome browser, our transcript of interest (TCONS_00000021) seems to lack exon 2 when compared to the GLTPD1 gene (uc001aeo.3), and the lack exon 2 is evident with a missing start codon which is present in the known gene. Turning on Pfam-track does predict protein domains, and we see that a potential protein domain has been predicted in the GLTPD1 gene. Since TCONS_00000021 isoform lacks the start codon and the predicted protein domain in exon 2, we suggest the probability of the isoform having a different function or activity when compared to that of the original gene.