Homework 3, group 5

Ann C. Printz
Jesus V. Torresano Lominchar
Peter C. Holm
Sesilie B. Weiss
William P. Bullock

Part 1: Microarrays

1) Your colleague has made two microarray expression experiments: one for HIV-positive patients (5) and one for healthy controls (5). She is interested in two genes in particular (RXRA and IRX3). 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? Explain why/why not?

Comparing expression levels of two genes with a microarray experiment is very difficult. The relationship between abundance of gene product and signal on the microarray is influenced by many different factors. For instance, probes are almost never comparable as they will have different binding affinity and melting temperatures (due to for example different lengths and GC-contents). If probes are not unique for the individual genes, cross-hybridization to other genes would lead to a false increase in signal. It is possible to partly normalize some of these differences, but even then, the relationship between gene product and signal is not linear. Therefore, we conclude that the conclusion can't be made.

2) The actual normalized data is in the file normalized_data.txt. The five first columns are from HIV patients and the last five from healthy controls. Each row contains the normalized expression values from the probes(s) corresponding to one gene. 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)

```r
# Load packages
library(magrittr)

# Read the normalized_data.txt into R
HIV_study <- read.table(file = "data_for_hw3/data_for_part_1/normalized_data.txt")

# Change colnames
colnames(HIV_study)  <- c(paste0("HIV", 1:5), paste0("normal", 1:5))

# Run t.test on each row, and add p-vals to a new column
HIV_study <- HIV_study %>%
  apply(1, function(x) t.test(x[1:5], x[6:10])$p.value) %>%
  cbind(HIV_study, p.val = .)
```

We are assuming that expression of each gene is normally distributed within each condition, and therefore use a t.test. For each gene, the null hypothesis is that the difference in mean expression value between HIV-positive and HIV-negative is equal to zero. The alternative hypothesis is that there is a difference in mean expression value between HIV-positive and HIV-negative.

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?

```
# Number of false positives expected with a threshold of 0.05
0.05 * nrow(HIV_study)
```

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

```
# Number of genes with p-val less than 0.05
sum(HIV_study$p.val < 0.05)
```

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

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?

```
# Count number of genes with a p.val less than 0.2 when using bonferroni correction
HIV_study$p.val %>%
  p.adjust("bonferroni") %>%
  is_less_than(0.2) %>%
  sum()
```

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

```
# Count number of genes with a q.val less than 0.2 when using FDR correction
HIV_study$p.val %>%
  p.adjust("BH") %>%
  is_less_than(0.2) %>%
  sum()
```

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

```
# Number of genes expected to be false positives
12 * 0.2
```

```
## [1] 2.4
```

```
# So the number of false postives expected are ~2
```

5) She also want to see how big the changes between the conditions are. So calculate the log2 foldchange for each gene.

```
# Add a fold.change column to the data set.
HIV_study <- HIV_study %>%
  apply(1, function(x) log2(mean(x[1:5])) - log2(mean(x[6:10]))) %>%
  cbind(HIV_study, fold.change = .)
```

6) 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

```r
# Genes that have a FDR < 0.2
genes.subset <- HIV_study$p.val %>%
  p.adjust("BH") %>%
  is_less_than(0.2)

# Foldchanges for these genes
HIV_study$fold.change[genes.subset]
```

```
##  [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
```

All fold-change values are positive, meaning that all genes are more upregulated in HIV than in normal patients. However, since none of the values are larger than one, the fold changes are less than doubled. The highest value is 0.6011, which corresponds to a fold change of 1.5169717, which in a biological context isn't that impressive.

Part 2: Identification of isoform switching based on RNA-seq data of HOXA1-knockdown experiment

1) Upload all 6 files to the BINF galaxy server (http://galaxy.bio.ku.dk/). Remember to use hg19 as reference genome. Import a GTF file containing all knownGenes (UCSC genes) in the region chr1 position 1,000,000 to 1,500,000. This should be used as a reference transcriptome in this exercise.

```r
# Uploading files
"Get Data" -> "Upload File" -> Select files for upload -> "Genome (set all)" -> "Start"

# Get GTF file containing knownGenes
"Get Data" -> "UCSC main" -> set regions to chr1:1,000,000-1,500,000 ->
set output-format to GTF  -> "Get Output" -> "Send query to Galaxy"
```

2) Use cufflinks to assemble the transcriptomes for the individual samples. Use the imported knownGenes as reference guide annotation and leave the rest at default parameters.

Q: For each condition calculate the average number of lines in the 3 assembled transcripts

```r
n.lines <- list(
  kd = c(1004, 1015, 1064),
  wt = c(991, 1014, 991)
)

lapply(X = n.lines, FUN = mean)
```

```
## $kd
## [1] 1027.667
##
## $wt
## [1] 998.6667
```

3) Use Cuffmerge to obtain a single combined transcriptome from all the samples, and use the imported UCSC genes as Reference Annotation. Additional information about the run can be found by pressing the button when the dataset is expanded.

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

The combined GTF file contains 1207 lines. As this is larger than the average number of lines in each conditions, this means, that there are ~180 transcripts that are specific for KD-samples and vice-versa for WT-samples (roughly).

Q: Download the dataset by pressing on the download button to export the combined transcriptome. You should then load it to the UCSC genome browser (http://genome.ucsc.edu/) using My Data -> Custom Tracks tabs and take a screenshot (with default tracks) of the (whole) region with the data and include it in your answers. Use squished for knownGenes and your User Track.

The screenshot can be seen below:

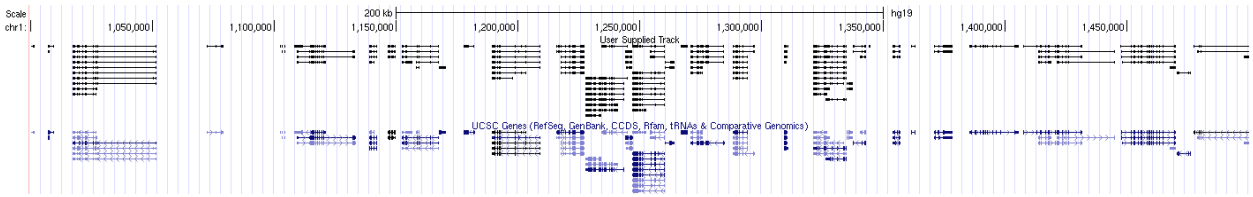

Figure 1: The combined transcriptome seen as a custom track on the genome browser

4) Use Cuffdiff to make the differential expression analysis between the two conditions (WT and KD) (by highlighting the appropriate files). Turn on multi-read correct with default parameters.

Q: While we are not using it here, why would be want to use bias correction (google it and include reference for answer).

In the preparation of a cDNA library most protocols include a fragmentation step. The inherent randomness of these fragmentation procedures are generally believed to produce fragments that are uniformly random. However, recent data suggests that the fragmentation step include both a positional aswell as a sequence bias - that is some places and specific sequences in RNA-molecules are more prone to fragment than others. This so called fragmentation bias, if not corrected for, can make in more difficult to identify significant differential expression for some transcripts.

Reference: Roberts, Adam, et al. "Improving RNA-Seq expression estimates by correcting for fragment bias." *Genome biology* 12.3 (2011): R22.

5) After the CuffDiff run is done, remove all results except the result of the differential expression analysis on gene, transcript and splicing: gene differential expression testing, transcript differential expression testing and Splicing differential expression testing.

Q: Specify what CuffDiff have tested in the Splicing differential expression testing. Google or deduce it, include deduction and/or refrence.

From the "Tool help" tab on galaxy, information regard the splicing differential expression (SDE) can be found. For each primary transcript with atleast 2 isoform, the function test if the mix of different isoforms are significantly changed between conditions. Whereas the transcript gene expresion testing look individually at each transcript, the SDE groups all transcripts from same transcription start-site and tests if isoform weights are different between conditions.

6) To enable us to check your workflow you need to save html file showing your work flow. Please make sure to delete all non-used data from your history (for example you should remove all Cufflinks output not used (by pressing the x)).

The html file is generated as follows: Make sure that all datasets in your history are expanded (by clicking them once). In the history options menu (the small wheel at the top of your history) select Show structure. This should open a site showing your galaxy workflow, including your settings. When this site have loaded right click anywhere on the page, where the mouse icon does NOT change to a hand (no mouse-over effect - it should look like your standard curser). Choose save as. Save the html file to your computer. Check that the html file work by opening it in a browser the result should look identical to your galaxy except none of the buttons can be used.

Upload the resulting HTML file AND the associated folder together with the answers and the 3 CuffDiff result files files to the homework (zip the html and the folder into one).

Part 3 - Post analysis in R

Read the supplied 3 CuffDiff result files (gene differential expression testing, transcript differential expression testing and splicing differential expression testing) into R as three data.frames (one for each). Note that the files provided are different from what you are expected to get if you solved part 2, so the results should not be compared.

```
# Load data into R.
GDE <- read.table(
  file = "data_for_hw3/data_for_part_3/cuffdiff_gene_differential_expression.txt", h = T)
TDE <- read.table(
  file = "data_for_hw3/data_for_part_3/cuffdiff_transcript_differential_expression.txt", h = T)
SDE <- read.table(
  file = "data_for_hw3/data_for_part_3/cuffdiff_splicing_differential_expression.txt", h = T)
```

1) The unique transcript id in the transcript data.frame is test_id (which is also a column in the gene data.frame). Change the column name (test_id) to transcript_id to enable us to differentiate between them.

```
# Change "test_id" to "transcript_id" in the TDE.output data frame.
names(TDE)[which(names(TDE) == "test_id")] <- "transcript_id"
```

2) Make two new data.frames that only contains genes/transcripts that are expressed in at least 1 condition. For each data.frame make this in one line of R code without using the semicolon (;). Use these data.frames in the rest of the assignment.

```
# Get the genes and transcripts that are ex
GDE <- subset(GDE, value_1 > 0 | value_2 > 0)
TDE <- subset(TDE, value_1 > 0 | value_2 > 0)
```

3) How many genes and how many transcripts were expressed? How many genes and how many transcripts were significantly differentially expressed between conditions?

```r
# Number of genes
nrow(GDE)
```

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

```r
# Number of transcripts
nrow(TDE)
```

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

```r
# Number of significantly differentially expressed genes
GDE %>% subset(significant == "yes") %>% nrow()
```

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

```r
# Number of significantly differentially expressed transcripts
TDE %>% subset(significant == "yes") %>% nrow()
```

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

4) Make two new data.frames (one for gene, one for transcripts) where the transcript data.frame only contains the transcript_id, gene_id, value_1 and value_2 columns and the gene data.frame only have the gene_id, gene, value_1, value_2 columns. Use the merge() function to combine these two data.frames, based on gene ids. Use the suffix parameter to make the resulting column names easily understandable. How many rows does this new data.frame contain? How many columns?

```r
# Select the required rows
a <- TDE[ , c("transcript_id", "gene_id", "value_1", "value_2")]
b <- GDE[ , c("gene_id", "gene", "value_1", "value_2")]

# Merge the data.frames
TDE_GDE <- merge(a, b, by = "gene_id", suffixes = c("_transcript", "_gene"))

# Number of rows
nrow(TDE_GDE)
```

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

```r
# Number of columns
ncol(TDE_GDE)
```

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

5) For all transcripts calculate the Isoform Fraction values (IF values) and the corresponding dIF values. Do any of these calculation results in NAs? Explain why you could get NAs and discuss whether this should be corrected (by for example setting it to 0 or 100).

```r
library(tidyverse)

TDE_GDE <- TDE_GDE %>%
  mutate(
    IF1 = value_1_transcript / value_1_gene,
    IF2 = value_2_transcript / value_2_gene,
```

```
    DIF = IF2 - IF1
  )
```

Some of the IF and DIF-calculations return NA. When none of the transcripts from a given gene is expressed under a condition, then the formula `value_transcript / sum(value_transcript)` returns NA as the denominator becomes zero. Situations where one gene is transcribed under one condition and not under the other are interesting, but doesn't really represent differential expression. Furthermore, it is difficult to correct for this in a satisfying manner by simply arbitrarily setting the value to either 0 or 100, so we instead leave it unchanged.

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.

```
mean(TDE_GDE$DIF, na.rm = T)
```

```
## [1] 7.468228e-09
```

```
median(TDE_GDE$DIF, na.rm = T)
```

```
## [1] 0.0001255016
```

Both the mean and the median are very close to zero. This shows that the average transcript is not differentially expressed, and that outliers are evenly distributed between the two conditions.

7) Use R to subset the merged data.frame to only contain genes with potential isoform switching by identifying genes with dIF $>$ +/- 0.25 (0.25 is an arbitrary (but large) value). Furthermore add the p_value from the Splicing differential expression testing to the data.frame using the match() function.

```
TDE_GDE.subset <- subset(TDE_GDE, abs(DIF) > 0.25)

TDE_GDE.subset$p_value <- SDE[match(TDE_GDE.subset$gene_id, SDE$gene_id), "p_value"]
```

8) For the switch in the gene with the lowest p_value report

A) the transcript ids, B) the gene name (not the gene_id), if the gene have multiple names just report the first, C) the dIF values and D) the pvalue. Include the R code to extract exactly this data.

```
subset(TDE_GDE.subset, p_value == min(p_value))[c(2, 5, 10, 11)] %>%
  knitr::kable()
```

	transcript_id	gene	DIF	p_value
11	TCONS_00000021	uc001aeo.3	-0.3426094	0.00075
12	TCONS_00000022	uc001aeo.3	0.3426083	0.00075

9) Analyze the gene with a switch: What does the gene do (a few sentences with refrences). Take a look at the gene in the genome browser (upload and use the supplied GTF file instead of the one you made in part 2). Make sure to compare it to the knownGenes annotation you provided as guide. What is the difference between the transcripts involved in the isoform switches? Compare the alternatively spliced regions to the Pfam in UCSC Gene data track. Rearrange the tracks so the order is: 1) your track, 2) UCSC genes and 3) Pfam Domains. Report you findings including screenshots. Include an explanation of why it is (potentially) interesting to compare to the pfam track.

The gene uc001aeo.3 encodes a protein called GTLPD1 (Glycolipid Transfer Protein Domain-Containing Protein 1). The protein mediates intracellular transfer of ceramide-1-phosphate between organelle membranes and the cell membrane, and is required for normal structure of the golgi-stacks.

Looking at the gene in the UCSC genome browser (figure 2), the pfam-track shows that the isoform switch skips a protein-coding part of the region. This could potentially be interesting, as this could modulate the biological activity of the protein. One could hypothesize that one protein-isoform has altered binding affinities towards different glycolipids and could thereby alter the lipid-composition of the membranes.

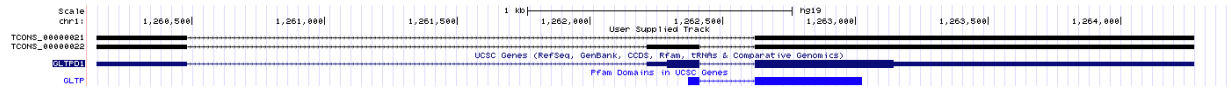

Figure 2:

"Non-vesicular trafficking by a ceramide-1-phosphate transfer protein regulates eicosanoids." Simanshu D.K., Kamlekar R.K., Wijesinghe D.S., Zou X., Zhai X., Mishra S.K., Molotkovsky J.G., Malinina L., Hinchcliffe E.H., Chalfant C.E., Brown R.E., Patel D.J. Nature 500:463-467(2013))