Julie Garcia
March 26, 2018

HW 3 (Due April 9, 2018 6pm)
Advanced Genomics and Genetics Analyses

In this assignment, you will be working with genome-wide 50 bp paired end RNA-Seq data that was generated from a human subject. Due to the size of this data, we will only be working with the first 2 lanes of a flow cell – the first of which includes sequences generated from thyroid tissue and the second of which includes sequences generated from testes tissue. You will be using tools below, so make sure you have installed them all:
RSEM: https://github.com/bli25ucb/RSEM_tutorial &
http://deweylab.biostat.wisc.edu/rsem/rsem-calculate-expression.html
https://deweylab.github.io/RSEM/README.html#built
Bowtie2: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
Samtools: http://www.htslib.org/doc/samtools.html
Integrative Genomics Viewer (IGV) from the Broad Institute, and either R or some language of choice to parse the necessary information asked of you.

1.) Go to the course website and download the following FASTQ formatted read sequence files:
s_1_1_sequence.txt.gz
s_1_2_sequence.txt.gz (first mate pair)
s_2_1_sequence.txt.gz
s_2_2_sequence.txt.gz(second mate pair)

Then obtain the fna (FASTA) and gff (gene annotation) files from the github site above using wget:
ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/vertebrate_mammalian/Homo_sapiens/all_assembly_versions/GCF_000001405.31_GRCh38.p5/GCF_000001405.31_GRCh38.p5_genomic.fna.gz

ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/vertebrate_mammalian/Homo_sapiens/all_assembly_versions/GCF_000001405.31_GRCh38.p5/GCF_000001405.31_GRCh38.p5_genomic.gff.gz

Alternatively, both files should be on the course website.

To obtain the primary assembly, run the following RSEM python script:

```
rsem-refseq-extract-primary-assembly
GCF_000001405.31_GRCh38.p5_genomic.fna
GCF_000001405.31_GRCh38.p5_genomic.primary_assembly.fna
```

Now create the RSEM reference using the commands below (modify for your environment). This may take ~1 hour to run so plan accordingly.

```
rsem-prepare-reference --gff3
GCF_000001405.31_GRCh38.p5_genomic.gff --bowtie2 --bowtie2-
path /bowtie2-2.2.3 --trusted-sources BestRefSeq
GCF_000001405.31_GRCh38.p5_genomic.primary_assembly.fna
human_refseq
```

2.) Using rsem-calculate-expression and the bowtie2 aligner, align the two tissue PE sequences separately. Note that you can run this on multiple processors if you have them (-p argument), if not, a single processor will take 8-12 hrs to run on a single processor, so plan accordingly. Be sure to use the following arguments: `--output-genome-bam --bowtie2-sensitivity-level very_fast --append-names`

```
rsem-calculate expression -p 2 --paired-end -bowtie2 -
bowtie2-path /usr/local/bin --bowtie2-sensitivity-level
very_fast --append-names --output-genome-bam
s_1_1_sequence.txt s_1_2_sequence.txt human_refseq s1

rsem-calculate expression -p 2 --paired-end -bowtie2 -
bowtie2-path /usr/local/bin --bowtie2-sensitivity-level
very_fast --append-names --output-genome-bam
s_2_1_sequence.txt s_2_2_sequence.txt human_refseq s2
```

3.) Look at the output results written to screen and provide the alignment rate for the thyroid read mapping.

```
81836199 reads; of these:
  81836199 (100.00%) were paired; of these:
    24329272 (29.73%) aligned concordantly 0 times
    25128352 (30.71%) aligned concordantly exactly 1 time
    32378575 (39.57%) aligned concordantly >1 times
70.27% overall alignment rate
```

```
I should say that the first time I ran this with -p set to
4, this step came up with a different number and then the
steps to calculate expression failed. This tells me that
there may be a bug in the code that splits and rejoins the
results from the separate threads.
```

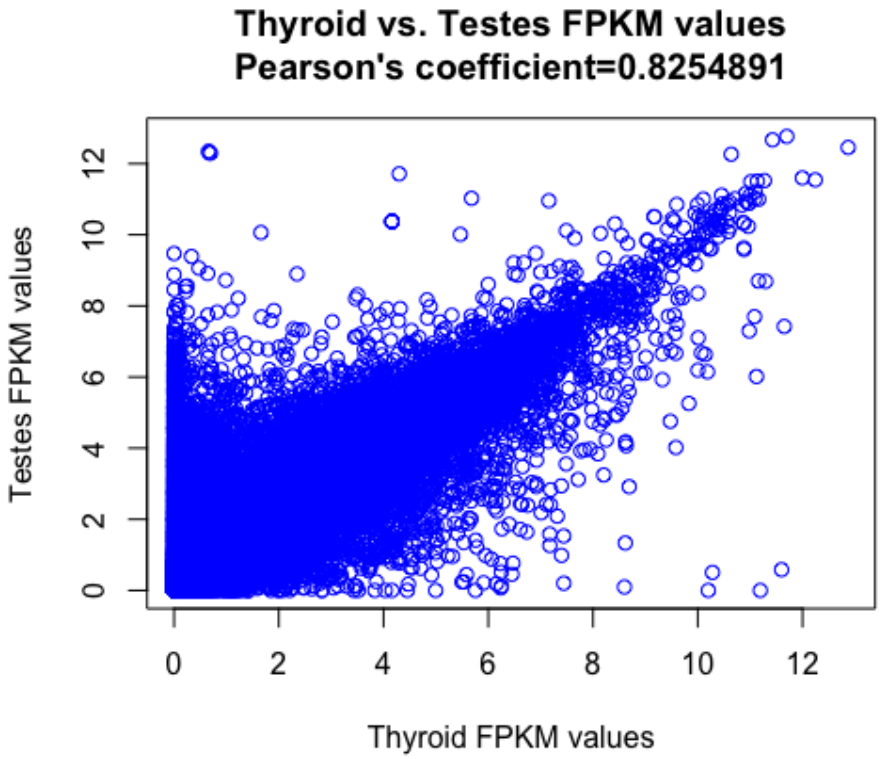
4.) Parse the gene results file from the testes and thyroid output to extract the TPM and FPKM values (two columns). Add one to each value and log them (base 2). Then plot a scatter plot and provide the reported Pearson's correlation coefficient between thyroid and testes tissue using FPKM, then TPM values. You should provide a plot for each quantification method.

```
#get thyroid and testes expression datasets
data.thyroid =
read.table("//Volumes/Drive/Bioinformatics/s1.genes.results"
, header=T)
dim(data.thyroid)
colnames(data.thyroid)
data.thyroid[1:4,]

data.testes =
read.table("//Volumes/Drive/Bioinformatics/s2.genes.results"
, header=T)
dim(data.testes)
colnames(data.testes)
data.testes[1:4,]

#subset for just TPM and FPKM
data.thyroid.stats <- data.thyroid[,6:7]
data.thyroid.stats[1:4,]
data.testes.stats <- data.testes[,6:7]
data.testes.stats[1:4,]

#normalize TPM and FPKM
normalize <- function(x) {
  return(log2(x+1))
}
data.thyroid.stats$TPM <- lapply(data.thyroid.stats$TPM,
function(x) sapply(x, normalize))
data.thyroid.stats$FPKM <- lapply(data.thyroid.stats$FPKM,
function(x) sapply(x, normalize))
data.testes.stats$TPM <- lapply(data.testes.stats$TPM,
function(x) sapply(x, normalize))
data.testes.stats$FPKM <- lapply(data.testes.stats$FPKM,
function(x) sapply(x, normalize))

# compare
data.thyroid[1:4,6:7]
data.thyroid.stats[1:4,]
data.testes[1:4,6:7]
data.thyroid.stats[1:4,]

#Pearson's correlation and scatter plot
help("cor")
data.FPKM.cor <- cor(as.numeric(data.thyroid.stats$FPKM),
y=as.numeric(data.testes.stats$FPKM),
use="pairwise.complete.obs")
```

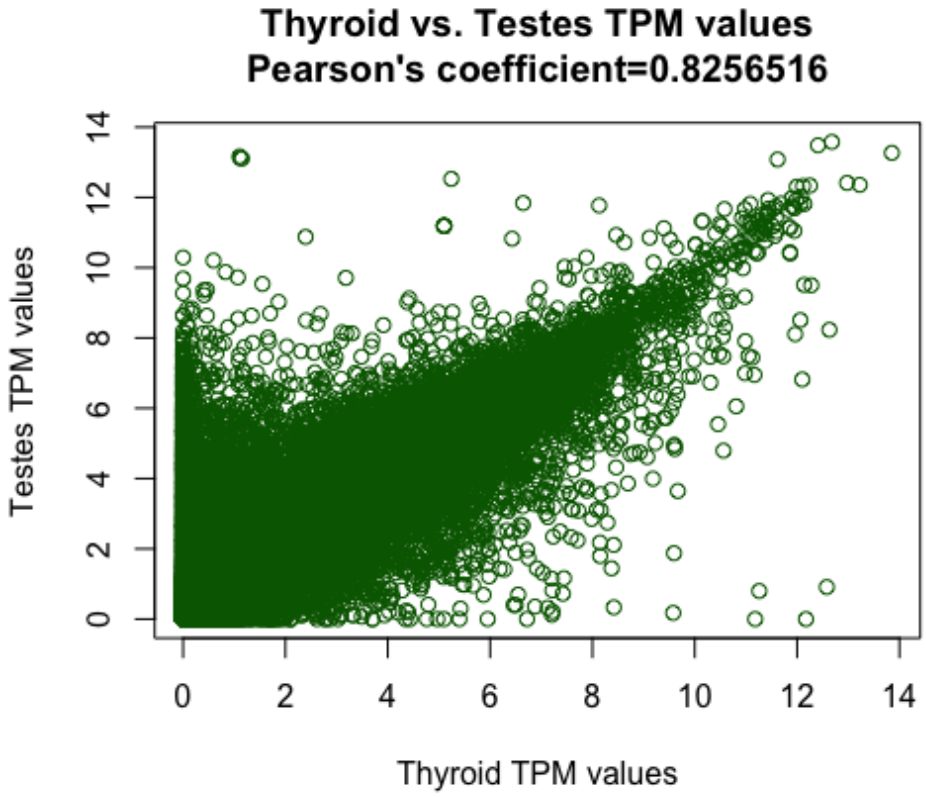
```
data.TPM.cor <- cor(as.numeric(data.thyroid.stats$TPM),
y=as.numeric(data.testes.stats$TPM),
use="pairwise.complete.obs")
data.FPKM.cor
data.TPM.cor
```

Pearson's coefficient for FPKM was 0.8254891.
Pearson's coefficient for TPM was 0.8256516.

```
plot(data.thyroid.stats$FPKM, data.testes.stats$FPKM,
     xlab="Thyroid FPKM values", ylab="Testes FPKM values",
     main="Thyroid vs. Testes FPKM values\nPearson's
coefficient=0.8254891",
     col="blue")
```




```
plot(data.thyroid.stats$TPM, data.testes.stats$TPM,
     xlab="Thyroid TPM values", ylab="Testes TPM values",
     main="Thyroid vs. Testes TPM values\nPearson's
coefficient=0.8256516",
     col="darkgreen")
```



5.) Calculate a fold change between the testes and thyroid tissue samples using TPM values and provide the top 10 genes that are over-expressed in testes tissue. Then do the same for the top 10 genes over-expressed in thyroid tissue.

```
# calculate fold change between testes and thyroid
foldchange <- as.numeric(data.thyroid.stats$TPM) -
as.numeric(data.testes.stats$TPM)
data <- cbind(data.thyroid[,1:2], foldchange)
data.sorted <- data[order(foldchange),]
dim(data.sorted)
genes.testes.over <- data.sorted[1:10,]$gene_id
genes.thyroid.over <- data.sorted[26100:26109,]$gene_id
genes.testes.over
```

```
gene38055_PRM2         gene38056_PRM1      gene7916_TNP1
gene7918_LOC101928327 gene48961_GAGE2D    gene3008_TSACC
gene2853_LELP1         gene38999_CMTM2     gene28007_LDHC
gene939_HMGB4
```

```
genes.thyroid.over
```

```
gene22565_SLC26A7      gene20358_SLC26A4-AS1
```

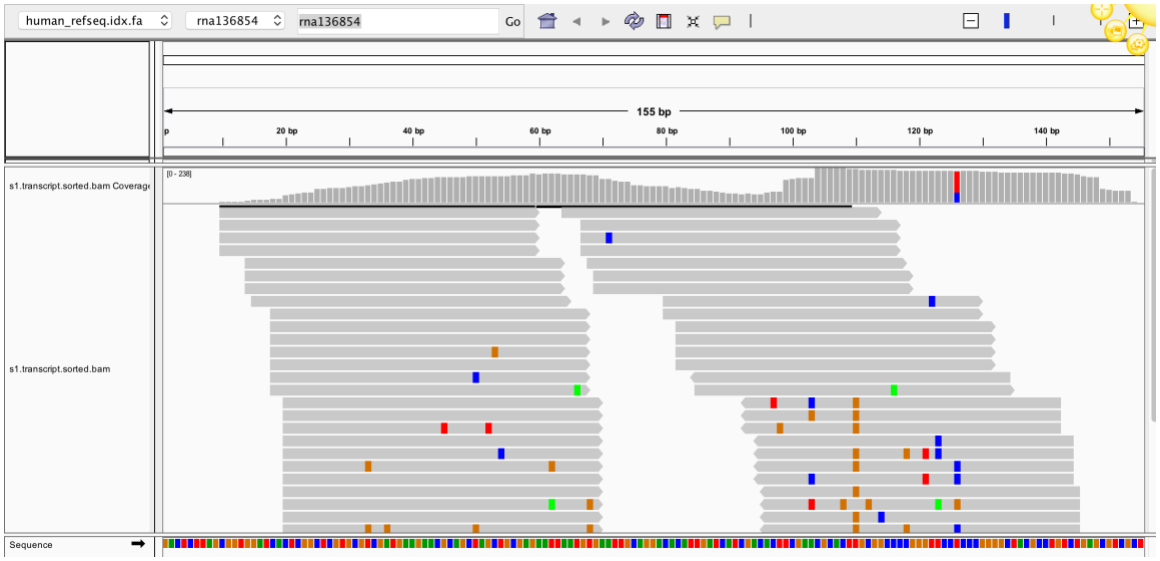
```
gene34555_NKX2-1gene37210_CRABP1
gene35236_TSHR          gene18348_IYD          gene4826_TPO
gene27923_PTH           gene23042_TG           gene50866_RNA5-
8S5
```

6.) Sort the testes and thyroid transcript bam files and index each with Samtools. Note that you can use multiple threads for sorting with the -@ argument.

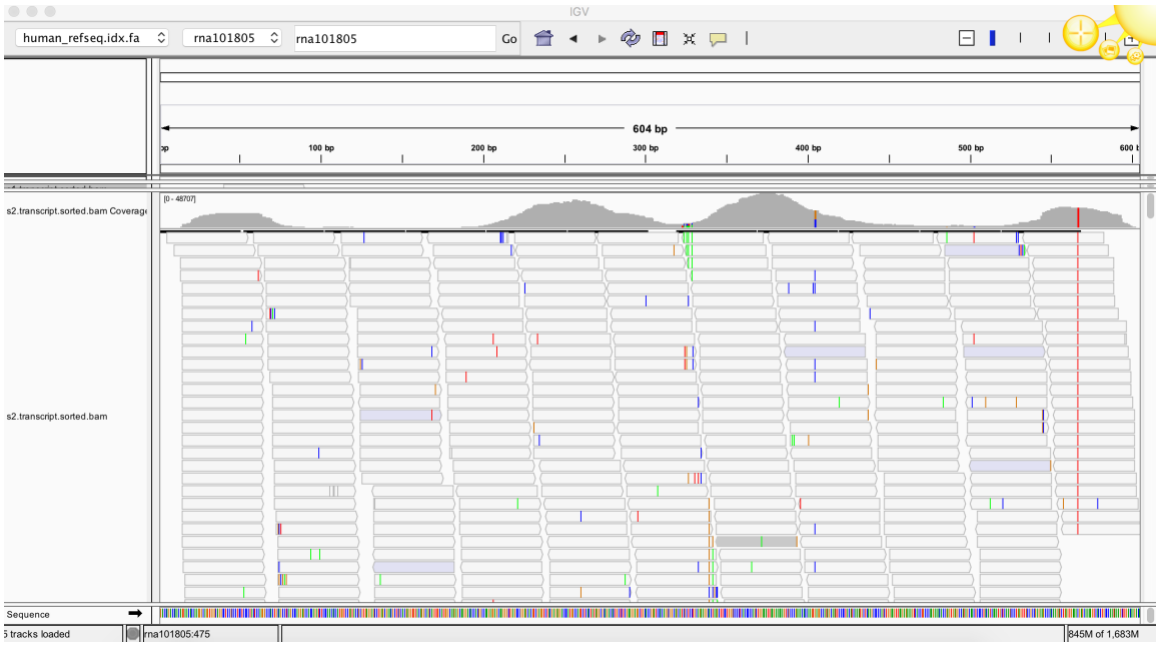
```
samtools sort -@ 2 -o s1.transcript.sorted.bam
s1.transcript.bam
samtools index s1.transcript.sorted.bam
samtools sort -@ 2 -o s2.transcript.sorted.bam
s2.transcript.bam
samtools index s2.transcript.sorted.bam
```

7.) Using Integrated Genomic Viewer (IGV), load the bam files for the testes and thyroid tissues as well as the human_refseq.idx.fa file that was created in a previous step. Then go to the top over-expressed gene in <u>thyroid tissue</u> and include this image in your output. You can use the GTF file you created previously to map the RefSeq transcript/gene ID to the gene symbol. Be sure to capture the entire gene in your image and show an expanded view of the reads. Also, color by read strand.

```
The top overexpressed gene in the thyroid sample was
gene50866_RNA5-8S5. The record in the GTF file gave the
transcript id of rna136854. The image below is of
human_refseq.idx.fa loaded at this location with
s1.transcript.sorted.bam. (I could not zoom to gene name, it
was looking for a transcript id so I used that).
```



The top overexpressed gene in the testes sample was gene38055_PRM2. The record in the GTF file gave the gene name PRM2. There were several different transcript ids for this gene, and since the human_refseq.idx.fa is organized by transcripts, I had to choose one to zoom to. I chose rna101805 shown in the image below.



8.) What do the blue, red, green, and orange vertical tick marks represent within each read bar (you might have to zoom in to see them)? What do the red and blue read pairs indicate? What does the grey histogram indicate above each track?

When you zoom in you can see that the vertical tickmarks represent mismatches in the reads with the reference genome. Blue = C, Red = T, Green = A, and Orange = G as shown below. The gray areas represent matches with the reference genome.

The blue and read read pairs indicate a locus at which the reads differ from the reference in more than 20% of quality reads. The red and blue bar below shows that at this locus the reads contained 67% T (red), but 33% C (Blue) which is signifcant. The gray histogram at the top shows the read depth at each locus. When the histogram is gray, it means there were not a significant number of variants there.