Assignment 6
Xiaojun Gao

Notes:
1. For all of the tabs when typing in terminal, since copy and paste would not work I am just manually typing out the commands again. (As well as for some | and “)
2. There is a command that I did not add for last time since it is only mentioned in a private post. This command might lead to slightly different results.

Part 1:
First, use the following command for each of the samples from last time:

```
stringtie -p 8 -G /opt/ccb/data/grch38/CHESS_v3.0_chr17.gtf -o
sample1.gtf -l sample1 sample1.bam
```

Next, merge the results from the previous step into one file, where merge list.txt contains the names of the result files:

```
stringtie --merge -p 8 -G /opt/ccb/data/grch38/
CHESS_v3.0_chr17.gtf -o stringtie_merged.gtf mergelist.txt
```

```
1a: grep "  transcript  " stringtie_merged.gtf | wc
```
Answer: There are 10023 transcripts in total.

```
1b: grep "  transcript  " stringtie_merged.gtf | cut -f 2 -d '"' |
sort -u | wc
```
Answer: There are 2941 distinct genes in total.

```
1c: grep "    transcript      " CHESS_v3.0_chr17.gtf | wc
```
Answer: There are 8324 transcripts in total.

```
1d: grep "    transcript      " CHESS_v3.0_chr17.gtf | grep
"protein_coding" | wc
```
Answer: 5770 of the transcripts are protein encoding.

```
1e: grep "protein_coding" CHESS_v3.0_chr17.gtf | cut -f 4 -d '"' |
sort -u | wc
```
Answer: 1323 distinct genes are protein encoding.

Part2:
Use the following command to compare the merged result with the original guide file:

```
gffcompare -r /opt/ccb/data/grch38/CHESS_v3.0_chr17.gtf -o merged
stringtie_merged.gtf
```

2a: How many of your transcripts exactly match all the introns of a known gene from the CHESS annotation?

```
Command: grep = merged.annotated.gtf | wc
```

Answer: 8150

2b: How many novel transcripts (i.e., they match a protein-coding gene, but they do not match any of the intron chains in the annotated transcripts) did you find in protein-coding gene loci?

Command:

```
grep protein_coding CHESS_v3.0_chr17.gtf > part2b_file2
```

The above step is just to make the script easier.

```
python part2b.py
```

Answer: 1320

2c: How many of your novel transcripts occur at entirely novel locations (code "u" from gffcompare)?

Command: `grep u merged.stringtie_merged.gtf.tmap | wc`

Since I did not ignore the first line when counting, I would need to minus 1 manually since there is a u within the first line. Moreover, gripping directly for u is valid because after observing all u are in the third field and not contained in any gene names or transcript names.

Answer: 58

Part3:

Use the following command to re-estimate the .bam files using the merged file as a guide:

```
stringtie -e -B -p 8 -G stringtie_merged.gtf -o
sample5_reestimate.gtf sample5.bam
```

3a: Among all the transcripts you assembled, and among all 11 samples, which one has the highest TPM? Report the transcript record (just the 'transcript' line) for this one as well as the sample in which you found it.

Here I am just trying to find the max TPM across all samples and see which transcript it correspond to. Since all that have TPM must be a transcript, we can just grep all the transcripts by grepping all lines that have TPM and write into file 3a.

```
grep TPM *_reestimate.gtf > 3a
```

To run the script: `python part3a.py`

The largest TPM value is 58949.156250.

To find out which file it is in:

```
grep 58949.156250 *_reestimate.gtf
```

The line found is:

```
chr17   StringTie       transcript      81509971        81512851
1000    -       .       gene_id "MSTRG.1681"; transcript_id
"CHS.23541.3"; ref_gene_name "ACTG1"; cov "5827.455078"; FPKM
"18322.001953"; TPM “58949.156250";
     which is in sample9_reestimate.gtf
```

3b: Looking across all 11 samples, how many distinct transcripts have a TPM above 0?

Since all that have TPM must be a transcript, we can just grep all the transcripts by grepping all lines that have TPM and write into file 3b.

```
grep TPM *_reestimate.gtf > 3b
```

Use python script part3b.py to figure the results of distinct transcripts having TPM above 0 by summing up all TPM per transcript so that I can identify any transcript that have at least one TPM above zero.

To run the script: python part3b.py

Result: 6768

3c: How many distinct genes have a TPM above 0?

Commands: The method is similar to 3b, only that this time we are looking for genes instead of transcripts.

```
grep TPM *_reestimate.gtf > 3c
python part3c.py
```

Result: 1747

3d: For every transcript, find its maximum TPM in all 11 samples. Report how many distinct transcripts have a maximum TPM greater than 50.

According to the TA, we should be finding the maximum TPM for a single transcript among each of the 11 samples. That is, we decide in which file does transcript have the largest TPM. We would then see how many distinct transcripts have the TPM above 50.

Commands:

```
grep TPM *_reestimate.gtf > 3d
python part3d.py
```

Result: 3442

3e: This one takes a bit more work. Sample SRR47952 is a control sample, and SRR47954 is a sample that was treated with a cancer drug, diarylpropionitrile (DPN). What you are doing in this exercise is just the beginning of an analysis to determine what genes were affected by the drug treatment.

For these two samples, SRR47952 and SRR47954, compute the total expression in TPM for each gene. This requires you to sum up all of the transcript TPM values for each gene. There will be nearly 3000 genes in your output, but we only want you to report the top 10 most-highly expressed genes, along with their total TPM values, for each sample. You will notice that the lists for SRR47952 and SRR47954 are different—think about whether you can attribute those differences to the drug treatment.

These correspond to my sample 1 & sample 2.

```
grep TPM sample1_reestimate.gtf > 3e_sample1
grep TPM sample2_reestimate.gtf > 3e_sample2
python part3e.py
```

Reported below are the highest TPM values and their associated reference gene name:

```
SRR47952
MSTRG.1681      59534.415283
MSTRG.361       51832.181947
MSTRG.915       26549.283813
MSTRG.782       23456.210815
MSTRG.142       22541.542236
MSTRG.281       18753.890015
MSTRG.1177      15532.078575
MSTRG.1443      12800.50445
MSTRG.28        12726.647949
MSTRG.555       10903.323433
```

```
SRR47954
MSTRG.1681      55431.926277
MSTRG.361       49412.500763
MSTRG.915       26607.427351
MSTRG.782       22595.666534
MSTRG.142       20002.772339
MSTRG.281       19543.814311
MSTRG.1177      14477.737433
MSTRG.1443      13474.539143
MSTRG.28        13327.57373
MSTRG.555       11850.421127
```

The above gene ids (the left column) correspond to the following reference genes, listed in order as above:
(Note that the order of gene ids corresponding to reference genes is the same for both samples!)
1) ACTG1: gene encoding for actin and affect function of muscles.
2) UBB: gene encoding ubiquitin.
3) RPL27: gene encoding a ribosomal protein that is a component of the 60S subunit.
4) RPL19: gene encoding a ribosomal protein that is a component of the 60S subunit.
5) PFN1: gene encoding a member of the profilin family of small actin-binding proteins. The encoded protein plays an important

role in actin dynamics by regulating actin polymerization in response to extracellular signals.
6) RPL26: gene encoding a ribosomal protein that is a component of the 60S subunit.
7) NME2: synthesis of nucleoside triphosphates other than ATP. Moreover, it acts as a transcriptional activator of the MYC gene which serves as a "master regulator" of cellular metabolism and proliferation. It would inhibit cell proliferation.
8) KCNJ16: encoding an integral membrane protein that acts as inward-rectifier type potassium channel.
9) YWHAE: expression was associated with tumor size, lymph node metastasis, and poor patient survival in patients with breast cancer.
10) SNORD42B: non-coding RNA (ncRNA) molecule which functions in the modification of other small nuclear RNAs (snRNAs).

The red genes are likely targets that would show different results if the drug is targeted towards cancer.

I did a comparison for the results of the change: (subtracting sample 2 result from sample 1)

MSTRG.1681	4102.489006
MSTRG.361	2419.681184
MSTRG.915	-58.143538
MSTRG.782	860.544281
MSTRG.142	2538.769897
MSTRG.281	-789.924296
MSTRG.1177	1054.341142
MSTRG.1443	-674.034693
MSTRG.28	-600.925781
MSTRG.555	-947.097694

I would say that according to the above result, there is a difference in the expression levels of the above genes between the 2 samples, despite being not significant. NME2 expression level is increased in sample 2 which means cell proliferation is increased, helping with healing cancer. YWHAE expression level is decreased, which means less cell proliferation in sample 2. Overall, sample 2 (i.e. SRR47954) seems to be the treated sample and sample 1 (i.e. SRR47952) is the original.

With that said, the change in TPM of genes likely to account for effective cancer treatment is minimal and not distinguishable at all compared to change of other genes. I am even not able to tell which of the samples are before and after treatment. I would conclude that I do not see this drug as helpful to cancer treatment based on these 2 samples.