homework 3

The main script is named homework.sh in the same github folder.

This is the line to signal that the script is a bash script:

```
!/bin/bash
```

These two lines are used to loaded the needed modules that have the software that we need:

```
module load jje/kent/2014.02.19
```

```
module load jje/jjeutils/0.1a
```

These three lines create the needed directories for either fasta or gtf:

```
mkdir /bio/khoih/ee282/
```

```
mkdir /bio/khoih/ee282/fasta/ mkdir /bio/khoih/ee282/gtf/
```

These 5 lines are used to change to the correct directory by using cd and download the needed files from flybase using wget:

```
cd /bio/khoih/ee282/fasta/
```

```
wget ftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.24_FB2018_05/fasta/*chromosome* (ftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.24_FB2018_05/fasta/*chromosome*) .
```

```
wget ftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.24_FB2018_05/fasta/*sum* (ftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.24_FB2018_05/fasta/*sum*) .
```

```
cd /bio/khoih/ee282/gtf/
```

```
wget ftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.24_FB2018_05/gtf/* (ftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.24_FB2018_05/gtf/*)
```

This line is just to echo status so we know that the script is done upto this point:

```
echo “download done”
```

These lines are use to change to correct directory of downloaded fasta using cd. The grep command is the cut out the md5sum line of chromosome fasta and output to new md5sum file:

homework question 1:

```
cd /bio/khoih/ee282/fasta/
```

```
grep -e “chromosome” ./md5sum.txt > ./chromosomemd5.txt
```

if then so that script only run if file intergrity (checked with md5sum) is ok. File intergrity status is also output to status report (“filestat.txt”):

```
if md5sum -c ./chromosomemd5.txt;
```

```
then
```

```
echo “gz is ok”> ./filestat.txt
```

get nucleotides total count, total count of N, and total number of sequence

using faSize to acquire all the needed count

```
faSize <(zcat ./*fasta.gz) > ./qccount.txt
```

The three grep commands do the same thing as faSize but they output the count for nucleotide, N , and sequence count seperately: zcat *.fasta.gz is to unzip and loaded into grep with < . grep -v “^>” is used to cut all line started with >. Then, awk is used to calculate nucleotide count by third column (total count of byte) - first column (total count of line)

count all nucleotide with grep

```
grep -v “^>” <(zcat ./*.fasta.gz) | wc | awk ‘{print $3-$1}’ > ./totalnucleotidecount.txt
```

count all N with grep . The grep command is the same as above. tr -cd N is used to delete everthing that is not N. wc -c is used to count all the N.

```
grep -v “^>” <(zcat ./*.fasta.gz) | tr -cd N | wc -c > ./Ncount.txt
```

sequence count with grep. Grep is used as described above with each line representing a sequence.

```
grep -c “^>” <(zcat ./*.fasta.gz) > ./sequencecount.txt
```

else is used to set a different action when the fasta file is corrupted.

```
else
```

```
echo “gz is corrupted” > ./filestat.txt
```

```
fi
```

echo is used to print status report

```
echo “question 1 done”
```

homework question 2:

cd to change to correct directory:

```
cd /bio/khoih/ee282/gtf/
```

if then so that script only run if file intergrity is ok by using md5sum:

```
if md5sum -c ./md5sum.txt;
```

```
then
```

```
echo “gz is ok”> ./filestat.txt
```

feature count for all chromosome. zcat is to unzip the gtf.gz file. Then awk is use to print out the third column for all chromosome. the third column contain features. Then the file is sort and count the total occurence of each features for all chromosome.

```
zcat ./*.gtf.gz | awk ‘{print $3}’ | sort | uniq -c > ./featurecount.txt
```

gene count by chromosome. The script is similar to above with two difference. The $1 ==“X” condition is used to ensure that awk only print third column when the chromosome is X. Then, after counting all features, grep only cut the count of gene to output text file. The remaining line is the same with $1 == selecting different chromosome:

```
zcat ./*.gtf.gz | awk ‘$1 == “X” {print $3}’ | sort | uniq -c | grep -e “gene” > ./Xcount.txt
```

```
zcat ./*.gtf.gz | awk ‘$1 == “Y” {print $3}’ | sort | uniq -c | grep -e “gene” > ./Ycount.txt
```

```
zcat ./*.gtf.gz | awk ‘$1 == “2L” {print $3}’ | sort | uniq -c | grep -e “gene” > ./2Lcount.txt
```

```
zcat ./*.gtf.gz | awk ‘$1 == “3L” {print $3}’ | sort | uniq -c | grep -e “gene” > ./3Lcount.txt
```

```
zcat ./*.gtf.gz | awk ‘$1 == “2R” {print $3}’ | sort | uniq -c | grep -e “gene” > ./2Rcount.txt
```

```
zcat ./*.gtf.gz | awk ‘$1 == “3R” {print $3}’ | sort | uniq -c | grep -e “gene” > ./3Rcount.txt
```

```
zcat ./*.gtf.gz | awk '$1 == "4" {print $3}' | sort | uniq -c | grep -e "gene" > ./4count.txt
```

else is used to output status if the gtf file is corrupted:

```
else
```

```
echo "gz is corrupted" > ./filestat.txt
```

```
fi
```

echo the status of job to notify that the script is done to this point.

```
echo "question 2 done"
```