Organization	Course	Exercise	Semester	Professor
Seoul National University	Introduction to Bioinformatics	Bash introduction	Spring 2022	Asst. Prof. M. Steinegg (martin.steinegger@sn

Exercise02: Bash introduction #2

The goal of this exercise is to improve skills working with basic bash commands and awk in UNIX environment.

> Commands to learn awk, uniq, sort, file redirection with '>'

To submit your result, follow these steps:

- Step 1. Clone this template repository to your working directory and execute "setup.sh"
- Step 2. Fill in the command used in the command0X.sh in the "command" directory. The commands should generate the result of step 3. The result can either be printed to the terminal or written to a file.
- Step 3. Save the result to ./result/result0X_X.txt or ./result/result0X_X.csv for each command.
- Step 4. Add edited files to git and commit

```
git add .
git commit -m "COMMIT MESSAGE"
```

- Step 5. Submit your answers by pushing the cloned repository.

```
git push origin main
```

General instruction

- Use redirection to print your results on the result files.
- The default order of 'sort' can differ by the envrionment setting, but you don't have to care if you use the provided docker. You may lose your points if the result is sorted in a different order.
- Running main.sh should be enough to reproduce all the results.

command01.sh

1. Download the GTF file of Drosophila melanogaster and save it as **d_melanogaster.genes.gtf.gz** in the "data" directory. (No result file)

   Link: ftp://ftp.ensembl.org/pub/release-103/gtf/drosophila_melanogaster/Drosophila_melanogaster.BDGP6.32.103.gtf.gz

2. Extract the gzip-compressed GTF file. (No result file)

command02.sh

The first few lines of GTF file begin with "#". These lines are called header lines.

1. Use a command to extract only the header line from a GTF file and store the result to **result02_1.txt** in the "result" directory.
   - Don't use 'head' command here.
2. Count the number of lines in the GTF file except for header lines and save the number to **result02_2.txt**.
3. Count the number of header lines in the GTF file and save the number to **result02_3.txt**.

command03.sh

1. Extract unique chromosome names in "d_melanogaster.genes.gtf" and save it to **result03_1.txt**.

   > You can find the structure of GTF file from this link or from the lecture slides.

   - Example (Please follow this format, sorted by the default order)

   ```
   name1
   name2
   ...
   ```

2. Count genes in each chromosome and find **the chromosomes which have 100 or more genes**. Sort the name of chromosomes with 100 or more genes by the default order and write the **chromosome names as column 1** and the **count of genes as column 2** to **result03_2.csv**.

> Columns in CSV file, which means Comma-Separated Values, should be separated with comma, ",". When counting the number of genes in a chromosome, count the lines of which the feature type is "gene".

- Example (Please follow this format)

```
1C,200
A,100
B,300
C,400
...
```

command04.sh

- TIPS

```
 http://reasoniamhere.com/2013/09/16/awk-gtf-how-to-analyze-a-transcriptome-like-a-pro-part-1/
 http://reasoniamhere.com/2013/09/17/awk-gtf-how-to-analyze-a-transcriptome-like-a-pro-part-2/
 http://reasoniamhere.com/2013/09/18/awk-gtf-how-to-analyze-a-transcriptome-like-a-pro-part-3/
```

1. Extract the distinct genomic feature types (e.g., gene, exon, transcript ...) from the GTF file. Sort the values by the default order and save them to **result04_1.txt**.

- Example (Tab seperated)

```
CDS   1234  # There should be one tab between 'CDS' and '1234'.
Selenocysteine 1234
exon  1234
...
```

2. Find the line in which the feature type is "gene" and the gene name is "Raf". Save the line to **result04_2.txt**.

3. The "Raf" gene has multiple transcripts. Find all transcripts and store the attribute "transcript_name" (e.g., transcript_name "Raf-RE";) to **result04_3.txt**.

    - Example (please follow this format)

```
Raf-XX
Raf-AA
...
```

4. Count the number of exons of each transcript from "Raf" gene and save the count to **result04_4.csv** (remember CSV files are comma separated). Write the **transcript names (value of transcript_name) as column 1** and the **count of exons as column 2** like this:

```
Raf-XX,5
Raf-AA,3
...
```

5. Calculate the total exon length of each transcript from "Raf" gene and save the result to **result04_5.csv**. Write the **transcript names as column 1** and the **length of exons as column 2** like this:

```
Raf-XX,3300
Raf-AA,2500
...
```

> The position of GTF is 1-based, which means the 100nt-length region from 1st position to 100th position in chromosome 1 is represented as "chr1 1 100". Please consider this when calculating the length from position indices.

command05.sh

1. Download old version of *D. melanogaster* GTF file and unzip it as "d_melanogaster_old.genes.gtf" in data directory. (No result file)

   Link: ftp://ftp.ensembl.org/pub//release-89/gtf/drosophila_melanogaster/Drosophila_melanogaster.BDGP6.89.gtf.gz

2. Count the genes in the positive (+) and negative (-) strand of old *D. melanogaster* genome and save the counts to **result05_2.csv**.

   - Example (please follow this format)

```
x,y
```

3. The gtf file is keeping updated to cover new discoveries. Let's compare the changes in the number of genomic features (third column in gtf files like exon, gene, CDS, etc).

   3-1. Count the distinct genomic feature (exon, gene, and others) types of d_melanogaster_old.genes.gtf. Sort the values by default order and write them in **result05_3_1.txt** following the format of the example of command04-1.

   3-2. Check join.awk in the command directory. It is from the lecture slide (21th of Linux 2 slides). You have to make a small change in the code to follow the printing format of the example of command05-3-3. Your changes in join.awk should be pushed.

   3-3. Join the result05_3_1.txt and result05_3_2.txt using your join.sh and print the result in **result05_3_3.txt**

   - Example (Tab seperated, write the values from the old version in the 3rd column)

```
CDS   1234  12345
Selenocysteine 1234 2345
exon  1234  2345
...
```

If you can't execute a shell file due to "Permission denied" error, please try this command.

```
chmod +x ./<SOME_SHELL_FILENAME>.sh
./<SOME_SHELL_FILENAME>.sh
```