Homework answers

1. Create a directory named “hw2".

Answer: mkdir hw2

2. Move to “hw2" directory, and download this file (SRR25653406.fasta.tar.gz) using wget command:

https://github.com/UeenHuynh/MGMA_2024/raw/main/lecture2/2.%20basic%20Linux%20commands%20part%202/SRR25653406.fasta.tar.gz

Answer:

cd hw2

wget https://github.com/UeenHuynh/MGMA_2024/raw/main/lecture2/2.%20basic%20Linux%20commands%20part%202/SRR25653406.fasta.tar.gz

3. Decompressing this file SRR25653406.fasta.tar.gz using tar command.

Answer:

tar xvf SRR25653406.fasta.tar.gz

Additional information about FASTA format (The file has the extension .fasta, .faa, .fna, etc.)

FASTA format is a text-based format, that contains two lines:

- First line: is the comment (description) line.
  + Always start with the “>” sign (This information is useful for the exercise).
  + Give basic information about the sequence (nucleotide or amino acid).
- Second line: The actual sequence of the first line description, using a standard one-letter character string.

Example:

```
>M35309.1 E.coli 16S rRNA fragment
GGCATGAAGACACACTGCTAACTCCGAATACGCACAAGCCCGTAATGGAGCGACGGTGGGCCTTGTTCCC
GTGCCCCGATGTGGGGTGGAGGTGACTGTGGGTTGTGATATTCGGGGAGGCAAAAGAAGTAGCGAGTCTA
ACCTTGCTTACCACTTTGCCTAATACGGGAAACG
```

(https://www.ncbi.nlm.nih.gov/nuccore/M35309.1?report=fasta)

=> Let's call this single-sequence FASTA format (https://en.wikipedia.org/wiki/FASTA_format), so if a text file contains a single-sequence FASTA format, this file is called a single-sequence FASTA file.

If a text file contains two or more single-sequence FASTA format, this file is called a multiple-sequence FASTA file or multi-FASTA file.

An example of the multi-FASTA format :

(https://www.researchgate.net/figure/A-sample-of-the-Multi-FASTA-file_fig1_309134977)

Additional information about this file: SRR25653406.fasta (after decompressing using tar command).

- This is a multi-FASTA file. It is converted from fastq to fasta format file (Don't worry about fastq format in this exercise), and the original fastq format file is obtained from the sequencing machine.
- Each single-sequence FASTA in this file is information about a read - a piece of DNA sequence obtained from the sequencing machine:
    + First line: some information about the sequencing platform of this file, that's all, don't worry about the details.
    + Second line: DNA sequence of that read, a character string consisting of 4 types of letters: A, T, G, C representing 4 types of nucleotides. In addition, there is the letter N, which means 1 of 4 possible types of nucleotides, because the sequencing machine cannot identify which type of nucleotide it is.

A single-sequence FASTA in this file:

```
>SRR25653406.lite.3.8 M02606:41:000000000-J3RLF:1:1101:8131:2344 length=200
TTTGGTCCAGCAGGCTATTGCTGGGAATTGTTAGCTTTTCGGGATCCATTTTATTGGGCGGTGTGCTCATCTCTACCATTTCCAACATTATAGAGCGGCGTGTGG
ATGTGGTAAACACGGGGCGAATGACTTACCGCAACATAACTCAACACTATGTATTGATAGGCTTCAATGAACTAACTATCAATATGATACGCGAA
```

(The line containing the DNA sequence has 2 lines because the screen is not enough to display 1 line, so it is split into 2 lines).

4. How many lines does this file have? Using: wc.

Answer:

wc -l SRR25653406.fasta

```
khainguyen@khai:~/Documents$ wc -l SRR25653406.fasta
36354 SRR25653406.fasta
khainguyen@khai:~/Documents$ _
```

-> 36354 lines

5. How many reads does this file have? Using: grep for the search “^>” means the line starts with the character ">", and -c option.

Answers:

grep -c “^>” SRR25653406.fasta

```
khainguyen@khai:~/Documents$ grep -c "^>" SRR25653406.fasta
18177
khainguyen@khai:~/Documents$ _
```

-> 18177 reads

Explain:

To count the total number of lines containing information (the first line of a single-sequence FASTA) of all reads in the file, because each read has 1 line of information, so the number of lines containing information is equal to the number of reads.

And each line of information begins with the character > (the first line of a single-sequence FASTA), we use the grep command with the search string "^>", the character ^ is placed before the character >, which means Lines that begin with the character >. Use the -c option to count the number of lines that match the grep command. If you do not use the -c option, all results will be printed to the terminal.

6. How many reads do not have any N? Using: grep with -v, -c option; pipe.

Answer:

grep -v “^>” SRR25653406.fasta | grep -v -c “N"

```
khainguyen@khai:~/Documents$ grep -v "^>" SRR25653406.fasta | grep -v -c "N"
16757
```

-> 16757 reads without any N

Explain:

If not use option -v; with the search string "^>", the grep command will print lines starting with the character > (these are the information lines of all reads, the first line), because the character ^ means the starting line. top with...

If you use the -v option, the grep command will print lines that do not start with the > character, meaning it will print lines containing the sequence of all reads (second line).

Then use a pipe to pass this output to another grep command using the -v and -c options, now the second grep command has as input all lines containing the sequence of all reads. In this second grep command, the search string is "N", so with the -v option, it will print the sequence lines without the letter N, but here use the -c option to count the number of lines after executing the -v option, the result will be a number, not a sequence of lines printed to the terminal.

7. Create a file named "id_read.txt" containing only the first line (the line contains information) of all reads. Using: grep with “^v"; output redirection.

Answer:

grep “^>” SRR25653406.fasta > id_read.txt

You can check the results in 2 ways:

Way 1: Check the first 10 lines

```
khainguyen@khai:~/Documents$ cat id_read.txt | head
>SRR25653406.lite.3.1 M02606:41:000000000-J3RLF:1:1101:20298:1458 length=201
>SRR25653406.lite.3.2 M02606:41:000000000-J3RLF:1:1101:23187:1463 length=200
>SRR25653406.lite.3.3 M02606:41:000000000-J3RLF:1:1101:15837:1667 length=35
>SRR25653406.lite.3.4 M02606:41:000000000-J3RLF:1:1101:15601:1776 length=201
>SRR25653406.lite.3.5 M02606:41:000000000-J3RLF:1:1101:9791:1831 length=200
>SRR25653406.lite.3.6 M02606:41:000000000-J3RLF:1:1101:13988:1911 length=201
>SRR25653406.lite.3.7 M02606:41:000000000-J3RLF:1:1101:10459:2231 length=200
>SRR25653406.lite.3.8 M02606:41:000000000-J3RLF:1:1101:8131:2344 length=200
>SRR25653406.lite.3.9 M02606:41:000000000-J3RLF:1:1101:13210:2360 length=200
>SRR25653406.lite.3.10 M02606:41:000000000-J3RLF:1:1101:12039:2397 length=201
```

Way 2: Check all lines

```
khainguyen@khai:~/Documents$ cat id_read.txt | less
```

then ENTER.

```
khainguyen@khai: ~/Documents
>SRR25653406.lite.3.1 M02606:41:000000000-J3RLF:1:1101:20298:1458 length=201
>SRR25653406.lite.3.2 M02606:41:000000000-J3RLF:1:1101:23187:1463 length=200
>SRR25653406.lite.3.3 M02606:41:000000000-J3RLF:1:1101:15837:1667 length=35
>SRR25653406.lite.3.4 M02606:41:000000000-J3RLF:1:1101:15601:1776 length=201
>SRR25653406.lite.3.5 M02606:41:000000000-J3RLF:1:1101:9791:1831 length=200
>SRR25653406.lite.3.6 M02606:41:000000000-J3RLF:1:1101:13988:1911 length=201
>SRR25653406.lite.3.7 M02606:41:000000000-J3RLF:1:1101:10459:2231 length=200
>SRR25653406.lite.3.8 M02606:41:000000000-J3RLF:1:1101:8131:2344 length=200
>SRR25653406.lite.3.9 M02606:41:000000000-J3RLF:1:1101:13210:2360 length=200
>SRR25653406.lite.3.10 M02606:41:000000000-J3RLF:1:1101:12039:2397 length=201
:
```

Use the down and up arrow keys or the mouse wheel to view file contents.

Type “q" then ENTER to exit.

8. Create a file named "part_of_id_read.txt" containing a part of the first line (the line contains information) of all reads. Using: grep; pipe; cut; output redirection.

Two lines of 1 read:

```
>SRR25653406.lite.3.8 M02606:41:000000000-J3RLF:1:1101:8131:2344 length=200
TTTGGTCCAGCAGGCTATTGCTGGGAATTGTTAGCTTTTCGGGATCCATTTTATTGGGCGGTGTGCTCATCTCTACCATTTCCAACATTATAGAGCGGCGTGTGG
ATGTGGTAAACACGGGGCGAATGACTTACCGCAACATAACTCAACACTATGTATTGATAGGCTTCAATGAACTAACTATCAATATGATACGCGAA
```

(The line containing the DNA sequence has 2 lines because the screen is not enough to display 1 line, so it is split into 2 lines).

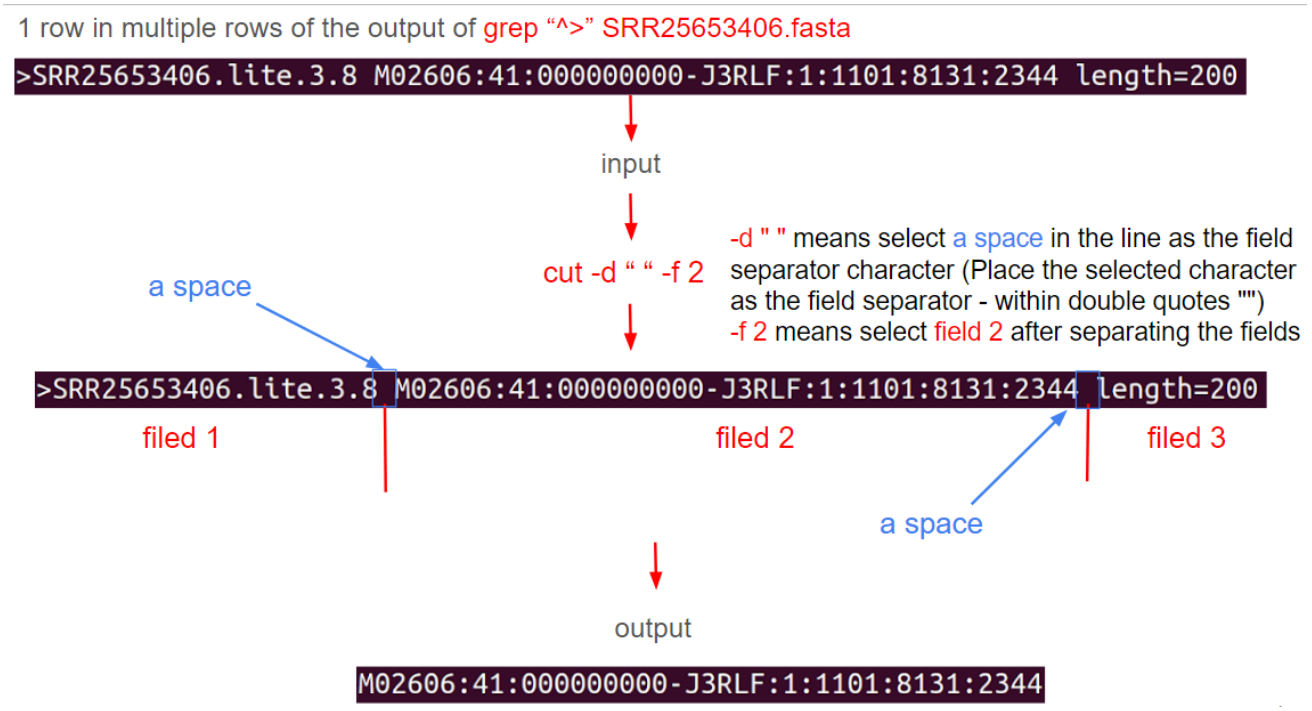
a part of the first line of 1 reads:

```
M02606:41:000000000-J3RLF:1:1101:8131:2344
```

Answer:

grep “^>” SRR25653406.fasta | cut -d “ “ -f 2 > part_of_id_read.txt

Explain:



Use the ways as in the answer to question 7 to check the file content. Here, I use way 2

```
khainguyen@khai:~/Documents$ cat part_of_id_read.txt | less
```

```
khainguyen@khai: ~/Documents
M02606:41:000000000-J3RLF:1:1101:20298:1458
M02606:41:000000000-J3RLF:1:1101:23187:1463
M02606:41:000000000-J3RLF:1:1101:15837:1667
M02606:41:000000000-J3RLF:1:1101:15601:1776
M02606:41:000000000-J3RLF:1:1101:9791:1831
M02606:41:000000000-J3RLF:1:1101:13988:1911
M02606:41:000000000-J3RLF:1:1101:10459:2231
M02606:41:000000000-J3RLF:1:1101:8131:2344
M02606:41:000000000-J3RLF:1:1101:13210:2360
M02606:41:000000000-J3RLF:1:1101:12039:2397
:
```