Coding tasks for 26.11.2020

Topic: Basic file filtering with awk

I. Introduction:

AWK is a commonly used tool for filtering and extracting specific data. But not only that, to be clear AWK is a computer language on its own and was released in 1977.

Small awk-commands are a easy method to work with files on the command-line or to perform some fast filtering before using the files elsewhere. Of course it can also be used to write longer scripts for all sorts of more complex filtering or data extraction.

These tasks serve to practice the basics and show only the easiest ways to use awk. Using awk as a command-line tool looks like this:

```
awk ' condition { action }' file
```

```
awk ' $2==0 { print $1 }' coding_task_sample.sam
```

There are hints for every task and some useful links at the end. All tasks should be solved on the coding_task_sample.sam-file.

II. Tasks:

1. How many records are in the sample-file?
Hint: NR contains number of line, print the one at the END.

2. Print the first line of the sample file.
Hint: $0 contains the full line, print the one at NR = 1.

3. Print every second line.
Hint: Use modulo on number of line (NR%2).

4. Get the bitwise flag of "SRR9077711.2552".
Hint: Check if first column ($1) contains the string, then print the bitwise flag ($2).

5. What is the mean length of all segment sequences?
Hint: You can sum up columns (var += $10), get length of the string first (length($10)).

6. What is the maximum length of all segment sequences?
Hint: If clauses can be used in awk ({if ($2 > x) x=$2}).

7. Find all mappings that contain the sequence "AGACACCG".
Hint: Regex can also be used in awk ($1~/Regex/).

8. Get all mappings to chr9 or chr7 with a segment sequence length of 71-73 and edit-distance of 0.
Hint: In a sam-file the edit-distance is an optional field (NM:i:<edit_dis>).

9. Write out read, chromosome and CIGAR-string for all mappings into a csv-file.
Hint: At the BEGIN set the seperator ({OFS=","}) and then get the searched cols ({print $1,$...}).

III. AWK commands as scripts:

As mentioned before awk-commands can be written in a script to reuse them. Structure of an awk-script:

```
#!/bin/awk
BEGIN {
        action1
    }
    {
        action2
    }
    END {
        output
    }
```

These scripts are used like this:

```
awk -f awk_script input_file1 input_file2
```

IV. Useful links for the tasks:

SAM file format: https://en.wikipedia.org/wiki/SAM_(file_format)

Small regex-cheat-sheet: https://tinyurl.com/y62lmdl3

Small awk cheat sheet: https://www.shortcutfoo.com/app/dojos/awk/cheatsheet

Full guide for awk: https://www.gnu.org/software/gawk/manual/gawk.html#Getting-Started