

Linux 'sed' command in Perl programming

Author : Muniba Faiza

Categories : [Bioinformatics Programming](#)

Date : June 3, 2018



When it comes to handling large data files to process, it becomes very difficult to write programs for the beginners, especially when you have to execute another script within one. For this purpose, Linux operating system offers several advantages for bioinformatics programming such as 'awk' and 'sed' one-liners, they prove to be of great help!

The *sed* command is a stream editor which parses and performs basic text transformations on a file or an input stream from a pipeline. *sed* allows restricting the command to certain lines or characters. It reads text line-by-line, removes trailing newline and stores them into an empty data buffer known as pattern space, on which the specified commands are executed. It is an amazing utility but the documentation is a bit difficult.

Syntax:

```
sed OPTIONS.. [SCRIPT] [INPUTFILE]
```

In bioinformatics, sometimes we need to edit large fasta files, which is quite difficult and tedious when done manually. The replacement and deletion commands of *sed* are most widely used, which can edit more than five thousands of sequences in a few seconds.

sed substitution command:

```
sed 's/regexp/replacement/g' input > output
```

or to modify the input file without getting output in another file, -i option is used:

```
sed -i input 's/regexp/replacement/g'
```

It has a vast range of applications in bioinformatics programming such as:

- one can easily edit or modify the fasta headers in a file consisting of a large number of sequences,
- can search for a particular expression present within the sequences,
- can delete text in between the lines, or in terms of bioinformatics, it can easily delete a specific number of residues from specific lines,
- can handle large data files, and so on.

In Perl, *sed* command can be easily executed using `exec()`, `system()`, `qx//`, or backticks (```) depending on the [need of the program](#). The similarity between the *sed* and Perl is that they use similar identifiers and/ characters. For example,

Functions	Character in sed	Character in perl
matches the last line of the file/ or \$ each input file.	\$	\$
matches the regular expression "regexp". If regexp itself includes any "/" characters, each must be escaped by a backslash ("\")	/regexp/	/regexp/
the I modifier allows case- insensitive pattern matching	/regexp/I	/regexp/i
treats the regexp being matched against as multiple lines	/regexp/M	/regexp/m

If you want to insert five blank spaces at the beginning of each line, then the following can be used:

```
$ sed 's/^/    /'
```

Again, here '^' is used for recognizing the first character in a line in Perl and *sed*.

Further reading:

<https://www.computerhope.com/unix/used.htm>

<http://www.grymoire.com/Unix/Sed.html#uh-47>

Sharing is caring. Spread the love!

- [Print](#)
- [Email](#)
- [LinkedIn](#)
- [Twitter](#)
- [Facebook](#)
- [Google](#)

