Basic Unix Command

1 Navigation commands

`ls`	list the content of a directory
`cd dirname`	change directory, e.g., `cd /usr/local/doc`
`cd ..`	move one directory up
`cd`	(without arguments) move to the home directory
`pwd`	prints the working directory (where you are!)
`mkdir dirname`	create an empty directory named "dirname"
`rmdir dirname`	delete an empty directory named "dirname"
`rm filename1 filename2`	delete files "filename1" and "filename2"
`rm -r dirname`	delete a directory and its content
`mv filename dirname`	move the file to a different directory
`mv oldfilename newfilename`	rename a file, from "oldfilename" to "newfilename"
`cp filename dirname`	copy the file to a directory
`cp -r dirname newpath`	copy a directory to a new path

2 Looking into files

`cat filename`	shows the content of a file
`cat filename1 filename2 > outputfile`	concatenate the content of two file and write it into a new file "outputfile"
`less filename`	shows the content of the file starting from the top (use arrows to navigate into the file and hit "`q`" to exit from the program)
`head filename`	shows the content of the first 10 lines of the file
`head -n 20 filename`	shows the content of the first 20 lines of the file
`tail filename`	shows the content of the last 10 lines of the file
`tail -n 20 filename`	shows the content of the last 20 lines of the file

3 File processing

`grep "something" filename`	extract lines that contain the string "something" (case sensitive; add "-i" parameter for case insensitive)
`grep -v "something" filename`	extract lines that do NOT contain the string "something" (case sensitive; add "-i" parameter for case insensitive)
`wc filename`	print characters, words and lines in the file
`wc -c filename`	prints characters in the file
`wc -w filename`	prints words in the file
`wc -l filename`	prints lines in the file
`cut -f 3,7 filename`	prints 3rd and 7th columns of a tab-separeted file
`cut -d, -f 3,7 filename`	prints 3rd and 7th columns of a comma-separeted file

4 Command line tips

`history`	Shows all the command executed in the shell.
← `(left arrow key)`	Move cursor to the left without deleting characters.
→ `(right arrow key)`	Move cursor to the without without deleting characters.
↑ `(up arrow key)`	Select the previous command in history (can be repeated several time).
↓ `(down arrow key)`	Select the next command in history (can be repeated several times).
`[tab]`	Autocomplete partial command/path.
`[CTRL+R] something`	Search "something" in command history.
`man command`	Print manual about "command". Hit "q" to exit and use arrow keys to navigate. "man man" prints the manual about "man" itself. If man pages are not available, you can try also "command –help" or "command -h".

5 Saving the output

To save the output of a command into a file, use the syntax:

```
command > output.txt
```

The "command" output is saved into "output.txt" file for further analysis, e.g.,

```
grep "ACTG" file > list.txt
wc -l list.txt > lines.txt
```

6 Pipelines

Several commands can be combined together: the output of a command can be redirected as input of another program. The paradigm is called "pipeline". In linux terminals you can use the character "|" (usually called "pipe key" and located in the top left keyboard corner).

Example 1. If we want to count the number of fasta sequences that have the label "ecoli" in the name, you an use:

```
cat file.fasta | grep ">" | grep "ecoli" | wc -l
```

Example 2. If we want the lines from 23th to 52th (30 lines) we can write:

```
cat file.txt | head -n 52 | tail -n 30
```

Note that in `grep`, `wc`, and `head` commands we do not write the input file name: input comes from the previous command.