

Биомолекула <https://biomolecula.ru/articles/ia-b-v-bioinformatiki-poshiol-pust-menia-nauchat> по мотивам статьи (<https://www.nature.com/articles/nbt.2740>)

<https://zenodo.org/record/1462325#.XWaiT-gzY2z> – книжка линукс для биоинформатиков

Unix tutorial

Listing files and directories

ls (short for list)

```
$ ls
```

To list all files in your home directory including those whose names begin with a dot

```
$ ls -a
```

with additional information:

```
$ ls -la
```

Making Directories

mkdir (make directory)

```
$ mkdir bioinformatics
```

```
$ ls
```

Changing to a different directory

cd (change directory)

```
$ cd bioinformatics
```

Type ls to see the contents (which should be empty)

Exercise

Make another directory inside the **bioinformatics** directory called **seminar1**. Navigate through

The directories . and ..

Still in the **seminar1** directory, type

```
$ ls -a
```

As you can see, in the **seminar1** directory (and in all other directories), there are two special directories called (.) and (..)

In UNIX, (.) means the current directory, so typing

```
$ cd .
```

NOTE: there is a space between cd and the dot

means stay where you are (the **seminar1** directory).

(..) means the parent of the current directory, so typing

```
$ cd ..
```

Pathnames

pwd (print working directory)

Pathnames enable you to work out where you are in relation to the whole file-system

```
$ pwd
```

The full pathname will look something like this -

```
/Users/mariap
```

which means that mariap (your home directory) is in the directory Users (the directory containing all user directories), which is located in the root directory (the root directory is indicated by the leading slash '/').

Exercise

Use the commands ls, pwd and cd to explore the file system. (Remember, if you get lost, type cd by itself to return to your home-directory)

~ (your home directory)

```
$ ls ~/bioinformtics
```

```
$ ls ~
```

```
$ ls ~/.
```

Copying Files

cp (copy)

cp *file1 file2*

\$ cd ~/bioinformatics

\$ touch dna.txt

\$ cp dna.txt protein.txt

Moving files

mv (move)

mv *file1 file2* moves (or renames) **file1** to **file2**

\$ mkdir rna

\$ cp dna.txt rna.txt

\$ mv rna.txt rna/.

\$ cp dna.txt dna1.txt

\$ mv dna1.txt rna.txt

Type ls and ls/rna to see if it has worked.

Removing files and directories

rm (remove), rmdir (remove directory)

\$ rm rna.txt

\$ ls (to check if it has deleted the file)

\$ rmdir rna

Displaying the contents of a file on the screen

clear (clear screen)

\$ clear

DOWNLOAD A FILE TO PLAY WITH

16SRNA.txt

cat, less, more, head, tail

The command cat can be used to display the contents of a file on the screen. Type:

```
$ cat 16SRNA.txt
```

The command less writes the contents of a file onto the screen a page at a time. Type

```
$ less 16SRNA.txt
```

Press the [space-bar] if you want to see another page, type [q] if you want to quit reading. As you can see, less is used in preference to cat for long files.

```
$ head 16SRNA.txt
```

```
$ head -5 16SRNA.txt
```

```
$ tail 16SRNA.txt
```

Searching the contents of a file

Взять файл geneticCode.txt

Simple searching using less

Using less, you can search through a text file for a keyword (pattern). For example, to search through **science.txt** for the word 'science', type

```
$ less geneticCode.txt
```

then, still in less (i.e. don't press [q] to quit), type a forward slash [/] followed by the word to search

```
/CCC
```

```
/Glycine
```

As you can see, less finds and highlights the keyword. Type [n] to search for the next occurrence of the word.

grep (globally search a regular expression and print)

grep is one of many standard UNIX utilities. It searches files for specified words or patterns. First clear the screen, then type

```
$ grep Glycine geneticCode.txt
```

To ignore upper/lower case distinctions, use the -i option, i.e. type

```
$ grep -i glycine geneticCode.txt
```

Some of the other options of grep are:

-v display those lines that do NOT match

-n precede each matching line with the line number

-c print only the total count of matched lines

```
$ grep -ivc glycine geneticCode.txt
```

wc (word count)

A handy little utility is the wc command, short for word count. To do a word count on **science.txt**, type

```
$ wc -w geneticCode.txt
```

To find out how many lines the file has, type

```
$ wc -l geneticCode.txt
```

Redirection

cat

Standard input – from the keyboard, standard output – to the terminal screen.

If you run the cat command without specifying a file to read, it reads the standard input (the keyboard), and on receiving the 'end of file' (^D), copies it to the standard output (the screen).

```
$ cat
```

Then type a few words on the keyboard and press the [Return] key.

Finally ^D to end the input.

If you run the cat command without specifying a file to read, it reads the standard input (the keyboard), and on receiving the 'end of file' (^D), copies it to the standard output (the screen).

In UNIX, we can redirect both the input and the output of commands.

Redirecting the Output

We use the > symbol to redirect the output of a command. Create a file called **nucleotides** containing a list of nucleotides, type

```
$ cat > nucleotides
```

Type:

adenine

cytosine

guanine

thymine

^D (Control D to stop)

```
$ cat nucleotides1
```

>> - appends standard output to a file

Exercise: create file **nucleotides2** with 4 of them, then append one more (uracil) to the file

We will now use the cat command to join (concatenate) **nucleotides1** and **nucleotides2** into a new file called **nucleotides3**. Type

```
$ cat nucleotides1 nucleotides2 > nucleotides3
```

```
$ cat nucleotides3
```

Redirecting the Input

< - to redirect the input of a command.

sort - alphabetically or numerically sorts a list

```
$ sort
```

Then type in the names of some nucleotides. Press [Return] after each one.

adenine

cytosine

guanine

thymine

^D (control d to stop)

```
$ sort < nucleotides1
```

output the sorted list to a file:

```
$ sort < nucleotides1 > sorted_list
```

```
$ cat sorted_list
```

Pipes

```
$ cat nucleotides1 | sort
```

Exercise

Using pipes, print all lines of **nucleotides1** and **nucleotides2** containing the letter 'u', sort the result, and save to the **list**.

(answer: `$ cat nucleotides1 nucleotides2 | grep u | sort > list`)

remove this files: `$ rm list nucleotides1 nucleotides2`

Wildcards

* and ?

* - none or more character(s) in a file (or directory) name

? - exactly one character

```
$ ls *.txt
```

```
$ ls ?eneticCode.tx?
```

Getting Help

man (manual)

```
$ man cat
```

apropos - when you are not sure of the exact name of a command

```
$ apropos copy
```

File system security (access rights)

```
$ ls -l
```

first column - access rights

the left group of 3 gives the file permissions for the user that owns the file (or directory)

(sburns in the above example)

the middle group gives the permissions for the group of people to whom the file (or directory) belongs (sburns in the above example);

the rightmost group gives the permissions for all others.

Access rights on files.

r (or -), indicates read permission (or otherwise), that is, the presence or absence of permission to read and copy the file

w (or -), indicates write permission (or otherwise), that is, the permission (or otherwise) to change a file

x (or -), indicates execution permission (or otherwise), that is, the permission to execute a file, where appropriate

Access rights on directories

r allows users to list files in the directory;

w means that users may delete files from the directory or move files into it;

x means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files.

Changing access rights

chmod (changing a file mode)

Processes and Jobs

ps (process) - information about your processes

\$ ps

& at the end of the command line - background process

Backgrounding a current foreground process

\$ sleep 100

^Z Then to put it in the background, type

\$ bg

jobs - see list of running processes with status

\$ jobs

to restart process

\$ fg %jobnumber

fg with no job number foregrounds the last suspended process

kill process

\$ kill %jobnumber

or

\$ kill (PID_number)

if refuses

\$ kill -9 (PID_number)

Other useful UNIX commands

df - reports the space left on the file system

du - the amount disk space used by each subdirectory
the depth of the listing with the -d switch

```
$ du -d 1
```

zip

```
$ zip geneticCode geneticCode.txt
```

```
$ unzip geneticCode.txt.zip
```

history - show command history list

file - classifies the named files according to the type of data they contain

```
$ file *
```

Pattern Search

```
$ cd /usr/share/common-licenses/
```

Поиск встречаемости паттерна

```
$ grep "GNU" GPL-3
```

Поиск паттерна не обращая внимание на размер символа

```
$ grep -i "license" GPL-3
```

Найдем строки не содержащие BSD

```
$ grep -v "the" BSD
```

В дополнение к предыдущему выведем еще номер строки

```
$ grep -vn "the" BSD
```

Регулярные выражения

[]: Matches any one of a set characters

[] with hyphen: Matches any one of a range characters

^: The pattern following it must occur at the beginning of each line

^ with [] : The pattern must not contain any character in the set specified

\$: The pattern preceding it must occur at the end of each line

. (dot): Matches any one character

\ (backslash): Ignores the special meaning of the character following it

*: zero or more occurrences of the previous character

(dot).*: Nothing or any numbers of characters.

Строки где слова начинаются с GNU и что угодно может идти после

grep "^GNU" GPL-3

Найти любую строку, которая заканчивается на "and"

\$ grep "and\$" GPL-3

Найти "сpt" с любыми двумя символами перед этим

\$ grep "..sept" GPL-3

Выражения в скобках для поиска с использованием регулярных выражений

Найдет too или two

\$ grep "t[wo]o" GPL-3

Начинается с "с" и имеет "ode" после

\$ grep "[^с]ode" GPL-3

Найдем все слова когда строка начинается с заглавных букв от А до Z

\$ grep "^[A-Z]" GPL-3

Аналогичное можно сделать, используя POSIX стандарт

\$ grep "^[[:upper:]]" GPL-3

Ищем слова в скобках с возможностью пробела между ними

\$ grep "([A-Za-z]*)" GPL-3

Найти любую строку, которая начинается с заглавной буквы и заканчивается точкой

\$ grep "^[A-Z].*\\.\$" GPL-3

Найти либо GPL либо General Public License

Внимание -E означает группировку (расширенные регулярные выражения)

\$ grep -E "(GPL|General Public License)" GPL-3

Найти либо copy либо copyright либо right

\$ grep -E "(copy)?right" GPL-3

Любые три символа из AEIOUaeiou идут рядом

\$ grep -E "[AEIOUaeiou]{3}" GPL-3

