Biopython prep course:
Introduction to the command line

$ [ whereis my brain?
sh: 2: [: missing ]

root@tecmint:~# touch girls\ boo**
touch: cannot touch `girls boo**': Permission denied

root@tecmint:~# nice man woman
No manual entry for woman
Linux is an operating system

An operating system translates commands that you type or mouse movements into instructions to the components of the computer.
Operating systems come with two types of user interfaces

Graphical User Interface (GUI)
< windows, icons, buttons, menus...
Manipulated by a mouse

Command line interface (shell)
Window without decorations
You type commands that tell computer what to do
Results are displayed as text
Linux command line

= Bash

Program that reads commands typed by user and gives them to OS to carry out

User has to use specific syntax

Commands are typed in a terminal (program that gives access to command line)

Available in Linux and in Mac

Most commands that we will do in Linux will be exactly the same on Mac

But behavior of some commands can differ in details
Open terminal in Linux Mint 17
The prompt in Linux Mint 17

Prompt:
supplied automatically by terminal, you don’t have to type it

Components:
username@machinename

current working folder: ~ means home folder of the user

$ : normal privileges

# : superuser (administrator aka root) privileges
Each user has a home folder

Command line: /home/bits
Each user has a home folder

Command line:  /home/bits
Organization of files in Linux
The disk on your computer that runs Linux is the File system.

Command line: `/`
Home folder is a folder of the File system

Command line: /home
bits folder is a folder of the home folder

Command line: /home/bits

Here the bits user is allowed to write: it’s his folder

In all other folders only the superuser may write
/bin and /usr/bin contain most of the commands/programs
/bin: essential programs that the system needs to operate
/usr/bin: commands that be used by the users
Open firefox by typing firefox in the terminal

Terminal sends command ‘firefox’ to OS
OS opens firefox web browser
Open firefox by clicking the firefox icon

Mouse click is translated into command ‘firefox’
OS opens firefox web browser
Why command line?

If you need to repeat a command 1000x

GUI: 1000 clicks

CL: write program (= script) that repeats command 1000x

Some tools are only available via command line (no GUI)
Navigation in the terminal

**UP- arrow key:** you go **back to** the **previous command**

Linux remembers the last 500 commands and stores them in a history.

To view the history type **history**

![Command history example]

Type **!number** of command you want to repeat.

**DOWN-arrow key:** you go **to** the **next command** from the command history.

**LEFT and RIGHT arrow keys:** move the cursor along the command line.
Copy and paste in the terminal

Copy: select the text you want to copy with your mouse
the text will be highlighted

```
bits@training-laptop ~ $ help
```

Paste: Shift key + Insert key

```
bits@training-laptop ~ $ help
help: command not found
```

Ctrl + C and Ctrl + V for copy and paste do not work in a terminal!

**Ctrl + C will terminate a running command**

**Mac users:** Command-C and Command-V work in the Mac terminal!
How commands work

<table>
<thead>
<tr>
<th>command</th>
<th>–options</th>
<th>arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ls</td>
<td>-l</td>
<td></td>
</tr>
<tr>
<td>cd</td>
<td></td>
<td>/usr/bin</td>
</tr>
<tr>
<td>man</td>
<td></td>
<td>su</td>
</tr>
<tr>
<td>less</td>
<td></td>
<td>seq.fasta</td>
</tr>
<tr>
<td>head</td>
<td>-n</td>
<td>4 sprot.fasta</td>
</tr>
</tbody>
</table>

Description of options and arguments are given in the manual or help file

? Define prompt, command, options and arguments ?

bits@training-laptop ~ $ cd NGS
bits@training-laptop NGS $ ls -l
bits@training-laptop NGS $ rm –r Variant
General commands in the terminal

**Leaving the terminal:** `exit`

**Changing to superuser:** `su`

Linux will ask for the superuser’s password: bitstrain

```
bits@training-laptop ~ $ su
Password: training-laptop
```

**Viewing the manual of a command:** `man command you want to view manual of`

*How to view the manual of the superuser command?*

**Viewing the help-file of a command:** `command -h`

`command --help`

**Printing text to the screen:** `echo text`
Manual gives description of how to use a command

**NAME**

```
 su - change user ID or become superuser
```

**SYNOPSIS**

```
 su [options] [username]
```

**DESCRIPTION**

The `su` command is used to become another user during a login session. Invoked without a `username`, `su` defaults to becoming the superuser. The optional argument - may be used to provide an environment similar to what the user would expect had the user logged in directly.

Additional arguments may be provided after the username, in which case they are supplied to the user's login shell. In particular, an argument of `-c` will cause the next argument to be treated as a command by most command interpreters. The command will be executed by the shell specified in `/etc/passwd` for the target user.

You can use the `--` argument to separate `su` options from the arguments supplied to the shell.
Naming of folders in Linux command line

~ home folder
/
root folder (File system)
.
./ current folder
../ mother folder (up one branch in the File system tree)

http://linuxcommand.org/lts0020.php#file
Commands to navigate through the file system

Name of current working folder:  

```
training-laptop bits # pwd
/home/bits
```

Change current working folder:  

```
training-laptop bits # cd Documents
training-laptop Documents # cd ..
training-laptop bits # cd Documents
training-laptop Documents # cd
training-laptop ~ #
```

- `cd [folder]`  
  - go to specified folder (in current folder)  
  - go up in tree to mother folder  
  - go to home folder

```
bits@training-laptop ~ $ cd /usr/bin
bits@training-laptop /usr/bin $
```

- `cd Documents = cd ./Documents`  
  - you may omit the ./ in most cases

- `cd = cd ~`  
  - both can be used

http://linuxcommand.org/lts0020.php#file
Commands to navigate through the file system

List content of current working folder: `ls [-options]`

```
bits@training-laptop /usr/bin $ cd /home
bits@training-laptop /home $ ls
bits
bits@training-laptop /home $ cd bits
bits@training-laptop ~ $ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
bits@training-laptop ~ $ ls -l
```
```diff
 total 32
  drwxr-xr-x 2 bits bits 4096 Sep 24 16:46 Desktop
  drwxr-xr-x 3 bits bits 4096 Oct 15 13:04 Documents
  drwxr-xr-x 2 bits bits 4096 Oct 11 11:12 Downloads
  drwxr-xr-x 2 bits bits 4096 Sep 24 16:46 Music
  drwxr-xr-x 2 bits bits 4096 Sep 24 16:46 Pictures
  drwxr-xr-x 2 bits bits 4096 Sep 24 16:46 Public
  drwxr-xr-x 2 bits bits 4096 Sep 24 16:46 Templates
  drwxr-xr-x 2 bits bits 4096 Sep 24 16:46 Videos
```
Commands to navigate through the file system

List content of current working folder: `ll = ls -l` including hidden files/folders
Manipulating folders and files

Create a folder (= directory): `mkdir [-options] folder`

Create a file: `touch file`

Remove a folder and all its contents: `rm -r folder`

Remove a file: `rm [-options] file`

Copy a folder: `cp -R source destination`

Copy a file: `cp [-options] source destination`

Moving or renaming files/folders: `mv [-options] source path_to_destination`

Locating files: `find where_to_look -name what_to_look_for`

Dumping content of file on screen: `cat file`

? How to search for a file sequence.fasta anywhere in the file system?

? How to search for a file sequence.fasta in your home folder?
Manipulating multiple files

* represents all files in a folder

? How to list all text files in a folder?

? How to remove all fasta files in a folder?

? How to remove all files in a folder that have blast in their name?

? How to find all txt files whose name starts with sequence?

! can do a lot, among others represents NOT

? How to remove all files in a folder except sequence.fasta?

? How to remove all files in a folder except the zip files?
Tips for making your life in command line easier

Tab completion of names of files and folders

These are the folders located in my home folder:

```
bits@training-laptop ~ $ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

To go to the Downloads folder:

I can type: `cd Downloads`

But I can also type: `cd Dow + Tab`

The command line will complete the name of the folder automatically

Making references to files and folders

If you have to use the name of a file / folder repeatedly you can make a shortcut for it:

```
folder=/home/bits/Documents/
cd ${folder}
```

Wiki exercise: Navigating the Linux file system
Commands to view files

View content of a text file: `less file`

```
training-laptop db # less sprot.fasta
```

View the first 10 lines of a file: `head file`

```
training-laptop db # head sprot.fasta
>sp|P24928|RPB1_HUMAN DNA-directed RNA polymerase II subunit RPB1 OS=Homo sapiens GN=POLR2A PE=1 SV=2 MHGGGPSPGSACPLRTIKRVQFGVLSPEDELKRMSVTEGGIKYPETTEGGRPKLGGLMDP RQGVIERGRCQTCAGNMTECPGHFGHIELAKPVFHVGVFLVKTMVLRVCVFCCSKLLVD SNNPKIKIDILAKSKQPKKRLTHYDLCKGKNICEEGEGMNDKFGVEQPEGDELTKEKG HGGGCRQYQPRIRRSSGLELYAEWKHVNESSQEKKILLSPERVHEIFKRISDEECVFVLMGEP RYARPEWMIHVTPLVPVPLSVRPAVVMQGSARNQDDLTHKLADIVKINNLQRNENQNGAAA HAVIAEDKLLQFHVATMVNDLNPGLPRAMQKGRLKSLKQLRGKKEVRGNGLMKRVVD FSARTVITPDNLIDQVQGVPRIAANMTFAEIVTPFNIDRLQELVRGNSQYPAGKYII RDNGDRLDLRFHPKPSDLHQTGYKVERHMCDGDIVIFNRQPTLHKSMMMGRHVRILPWS TFRLNLSVTPYPNADFDGDEMNHLHPQSLERAIQELAMVPRMIVTPQSNRPVVMGIVQD
```

View the first X lines of a file: `head -n X file`

```
training-laptop db # head -n 4 sprot.fasta
>sp|P24928|RPB1_HUMAN DNA-directed RNA polymerase II subunit RPB1 OS=Homo sapiens GN=POLR2A PE=1 SV=2 MHGGGPSPGSACPLRTIKRVQFGVLSPEDELKRMSVTEGGIKYPETTEGGRPKLGGLMDP RQGVIERGRCQTCAGNMTECPGHFGHIELAKPVFHVGVFLVKTMVLRVCVFCCSKLLVD SNNPKIKIDILAKSKQPKKRLTHYDLCKGKNICEEGEGMNDKFGVEQPEGDELTKEKG
```

http://linuxcommand.org/lts0030.php
Less opens the file as text in the terminal.
Navigating through a file opened with less

Pg Up               scroll back one page
Pg Dn               scroll down one page
G                   go to the end of file
g                   go to the start of file
UP arrow             navigate forward by one line
DOWN arrow           navigate backward by one line
/                    search for a pattern that you specify after /
                      takes you to the first occurrence from the position where you are
n                   goes to the next occurrence of the last search
h                   show help page
q                   exit the less page
File permissions

<table>
<thead>
<tr>
<th>Owner</th>
<th>Group</th>
<th>World</th>
</tr>
</thead>
<tbody>
<tr>
<td>rWX</td>
<td>rWX</td>
<td>rWX</td>
</tr>
</tbody>
</table>

Permissions:
- **r** (read)
- **w** (write)
- **x** (execute)

Owner and group permissions:
- **owner:** root
- **group:** root

HTTP link: [http://linuxcommand.org/lts0070.php](http://linuxcommand.org/lts0070.php)
Changing file permissions

`chmod 3 numbers filename`

<table>
<thead>
<tr>
<th>0</th>
<th>000</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001</td>
<td>--x</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>-w-</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>-wx</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>r--</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>r-x</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>rw-</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>rwx</td>
</tr>
</tbody>
</table>

First  Second  Third number

<table>
<thead>
<tr>
<th>Owner</th>
<th>Group</th>
<th>World</th>
</tr>
</thead>
<tbody>
<tr>
<td>rWX</td>
<td>rWX</td>
<td>rWX</td>
</tr>
</tbody>
</table>

? How to make a file readable by everyone and writeable by the owner?
Using the output generated by a command

command1 | command2: use the output generated by command1 in command2
command  >  filename: writing the output of the command to a file
command  <  filename: file contains input for command
command  >>  filename: appending the output of the command to the file

http://linuxcommand.org/lts0060.php
Downloading from the internet

`wget [-options] url`

url can start with http / https / ftp

`wget -P folder url`

saves the file in the specified folder

Wiki exercise: Manipulating files
Compressing files

Compression to .gz format:  

``gzip  file``

Compression to .bz2 format:  

``bzip2  file``

Uses Burrows-Wheeler compression

Makes smallest files but slow

Format not always compatible with bioinformatics tools

Compression to .zip format:  

``zip  zip_file  file_to_be_zipped``

Compression of one or multiple files/folders:

``tar  -options  compressed_file  files_to_be_compressed``

can perform tar + gzip or tar + bzip2 compression

``tar  --zcvf`` performs tar + gzip and generates a .tar.gz file

``tar  --jcvf`` performs tar + bzip2 and generates a .tbz2 file

``tar  --cvf`` performs tar and generates a .tar file

http://linuxcommand.org/lts0060.php
Decompressing files

Decompression from .gz format: **gunzip** `file`

Decompression from .bz2 format: **bunzip2** `file`

Decompression from .zip format: **unzip** `file`

Decompression from tar format:

```
  tar -options file
```

can perform tar + gzip or tar + bzip2 decompression

```
  tar -zxvf  decompressses .tar.gz
  tar -jxvf  decompresses .tbz2
  tar -xvf   decompresses .tar
```
Installing tools in Linux

**apt-get install:** fetches code from repositories
does everything for you (installing dependencies...)
OK for commonly used tools

**from binaries:** decompress

**from source:** decompress
configure (install all other tools it needs)
compile (source -> binary)

**Python scripts:** **pip install** tool

Installing tools in Linux

download source code: e.g. via wget
wget https://cutadapt.googlecode.com/files/cutadapt-1.3.tar.gz

decompress the file: e.g. via tar
tar –xvfz cutadapt-1.3.tar.gz

read documentation on how to install/compile

configure making sure your computer has everything to compile and install the tool
./configure

compile the source code to create all binaries e.g. via make
make

install these binaries so that they can be invoked from anywhere
make install
Error messages during configuration/compiling

If he can’t find a tool you’ll receive an error, saying you need to install a certain tool

```
checking for _FILE_OFFSET_BITS value needed for large files... 6
4
checking for dlopen... no
checking for dlopen in -ldl... yes
checking for fileno()... yes
checking for the %z format string in strftime()... yes
checking whether NLS is requested... yes
checking for intltool-update... no
checking for intltool-merge... no
checking for intltool-extract... no
configure: error: The intltool scripts were not found. Please install intltool.
```

Install missing tools with the `apt-get install` command and run configure again

```
sudo apt-get install intltool

./configure
```
Tips for making your life in command line easier

Linking tools
To avoid that you have to type the full path to a tool each time you want to use it

    e.g. /usr/bin/db/NGS/FASTQC/fastqc

You can make a link to that file in /usr/local/bin using ln

    e.g. ln -s /usr/bin/db/NGS/FastQC/fastqc /usr/local/bin/fastqc

Once you have this link you can start the tool by its name e.g. Fastqc

Destroying links

    e.g. unlink /usr/local/bin/fastqc
All tools can be run from command line

Every tool can be started from command line by typing the command with parameters and attributes when you have made a link:

```
blastp -query seq.fasta -db mydb.fasta -out output.xml
```

Going to the correct folder and preceding it with ./ when you have not made a link:

```
./blastp -query seq.fasta -db mydb.fasta -out output.xml
```

Java binaries (.jar) via special command:

```
java -jar program.jar
```

If the tool works via a GUI you can also start them by double clicking to open the GUI.

In most cases you cannot run downloaded tools as a safety measure as you need to set the permission for this file to 'executable'.
Finding tools

```
joachim@joachim-VirtualBox ~ $ bowtie --version
bowtie version 0.12.7
64-bit
Built on allspice
Thu May  5 12:19:01 UTC 2011
Compiler: gcc version 4.6.1 20110503 (prerelease) (Ubuntu 4.6.0-6ubuntu2)
Options: -O3  -Wl,--hash-style=both -g -02 -g -02
Sizeof {int, long, long long, void*, size_t, off_t}: {4, 8, 8, 8, 8, 8, 8}
joachim@joachim-VirtualBox ~ $ which bowtie
/usr/bin/bowtie
```
Writing a bash script

Text file containing bash commands

1. Create in a text editor e.g. gedit, Geany...
   First line: `#!/usr/bin/env bash`
   Tells the system to use bash to interpret the script

2. Save as a .sh file

3. Make executable via `chmod 755`

4. Run via `bash name_script.sh`

http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html
The shebang line

Simple text files become scripts when adding a shebang line as first line
Says which program should read and execute this text file

<table>
<thead>
<tr>
<th>Shebang</th>
<th>File extension</th>
<th>Run the script</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bash</td>
<td>test.sh</td>
<td>bash test.sh</td>
</tr>
<tr>
<td>Perl</td>
<td>test.pl</td>
<td>perl test.pl</td>
</tr>
<tr>
<td>Python</td>
<td>test.py</td>
<td>python test.py</td>
</tr>
</tbody>
</table>