UNIX INTRODUCTION

Janos Binder - janos.binder@embl.de

Biology in 2014

- Omics Era
- Several Go of data / experiment
- Personalized Medicine

Why using Unix in Biomedical Research?

Free

Fast

Ethics

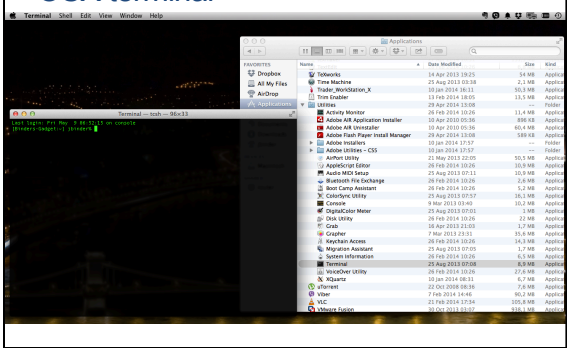
Open

Easy & documented!

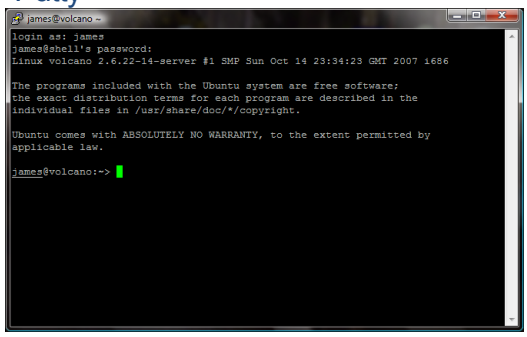
What do you do with Unix in Biology?

- File manipulation
  - Sequences
  - Interaction networks
  - Research articles
  - Spectrum
  - …
- Analysis
  - Calling programs (R, Perl, Python, Java, etc…)
  - Generating knowledge from raw information
  - Statistics
  - Filtering
  - …

OSX terminal



Putty



Why do we use terminal

- To learn how to work on a server, because:
- Common collaborations/projects
- More computing power
- Safer (backups)

What do you do with Unix in Biology?

- File manipulation
  - Sequences
  - Interaction networks
  - Research articles
  - Spectrum
  - …

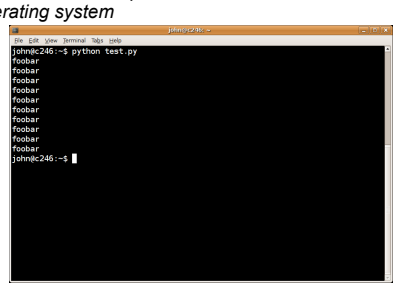
Part 1

- Analysis
  - Calling programs (R, Perl, Python, Java, etc…)
  - Generating knowledge from raw information
  - Statistics
  - Filtering
  - …

Part 2

The Shell – What the (s)hell is this?

- *Piece of software that provides an interface for users of an operating system*



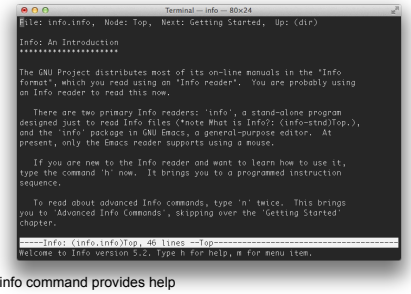
The Shell – Tips

- Previous command: ↑
- Auto-Complete: TAB
- Emergency Stop: Ctrl C / Ctrl Z
- Going forward/back: Alt ← / Alt →

STEP 0

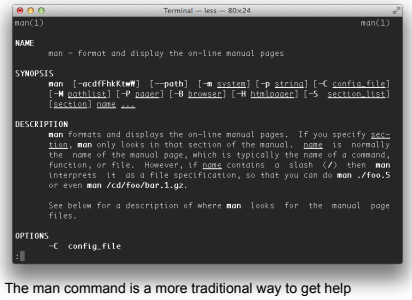
Getting help

info [command]



The info command provides help

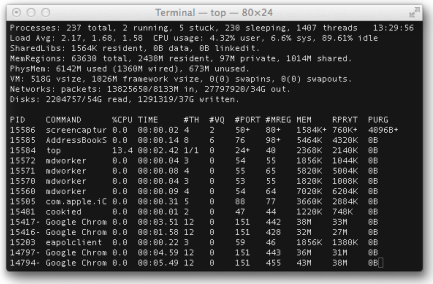
man [command]



The man command is a more traditional way to get help

What's going on:

top

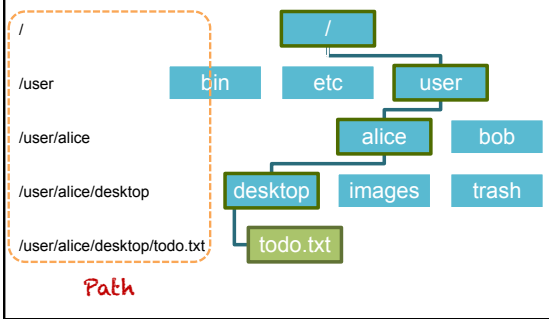


Display the list of running processes
To kill a program type 'k' on the keyboard followed by the PID

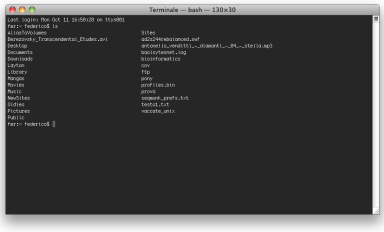
STEP 1

Browsing around

Everything's a file

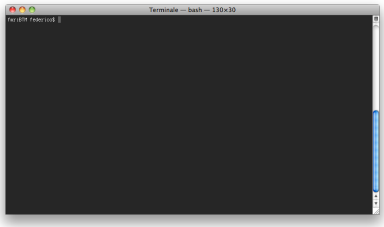


ls [directory] (list)



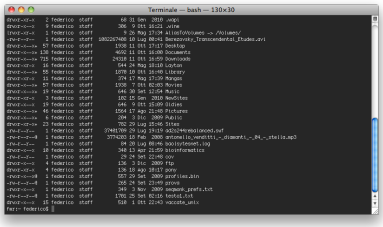
The ls command lists the contents of your current working directory

clear (clear screen)

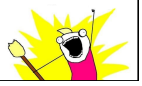


The clear command clears the screen

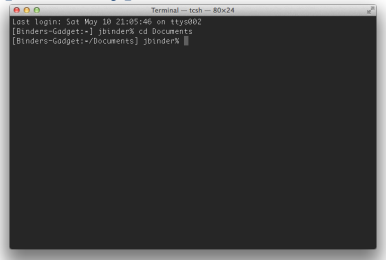
ls -al [directory] (long list)



The -al parameters show all files and their attributes
Try also –ltrh, it shows all files ordered by modification date

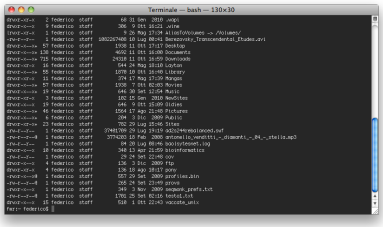


cd [directory]



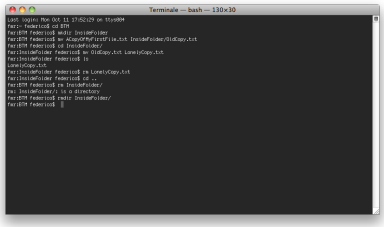
The cd command changes the current working directory to 'directory'
One level up: ..
This directory: .
Your home: ~

mkdir directory (make directory)



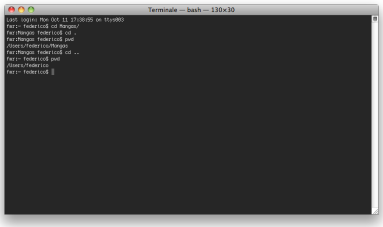
The mkdir command creates a new directory

rmdir directory (remove directory)



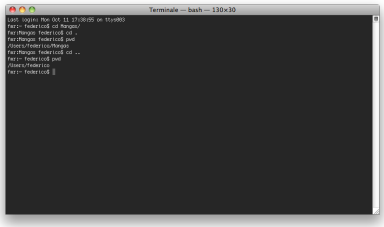
The rmdir command removes (a) directory

The directories . and ..



. (a full stop) refers to the current directory,
.. (two full stops) refers to the parent directory

pwd (print working dir)

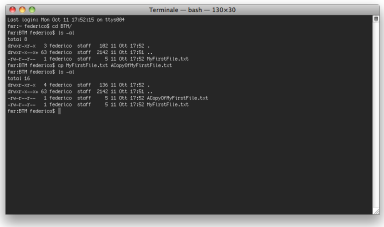


The pwd command prints the absolute pathname of the current working directory

STEP 2

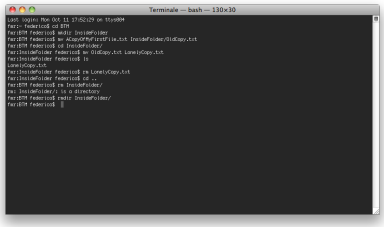
Handling files

cp file1 file2 (copy)



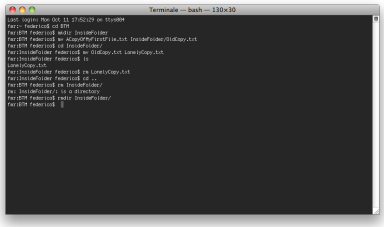
The cp command makes a copy of file1, naming it file2

mv file1 file2 (move)



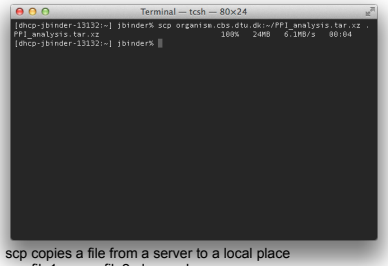
The mv command moves file1 into file2
(if file2 is in the same directory, file1 is renamed)

rm file (remove)



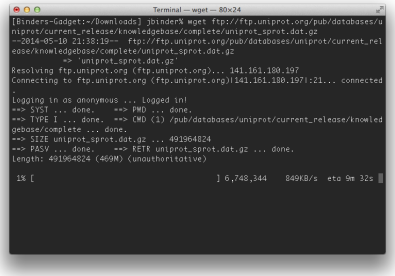
The rm command removes (a) file

scp server:file1 file2 (secure copy)



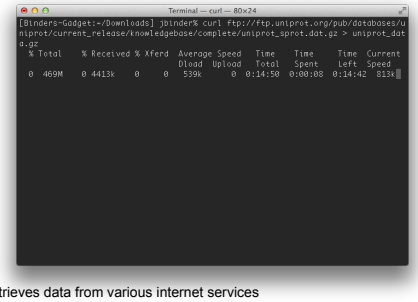
scp copies a file from a server to a local place
scp file1 server:file2 also works

wget url



wget can download files

curl url > filename

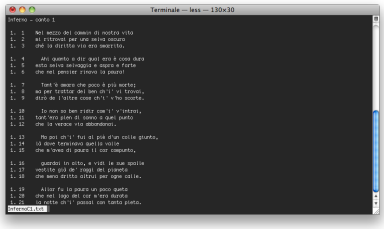


curl retrieves data from various internet services

STEP 3

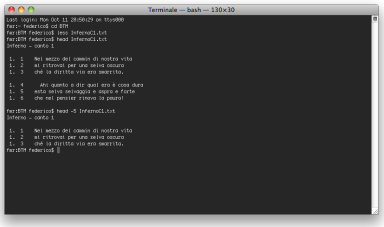
Files content

less file1



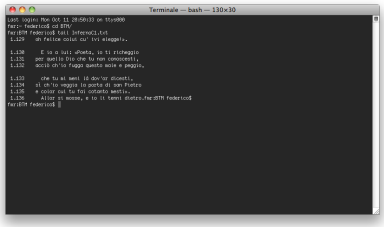
The less command shows the contents of a file
on the screen, one page at a time

head [-n] file



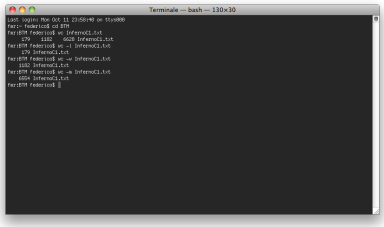
The head command shows
the first 10 (or n) lines of a file

tail [-n] file



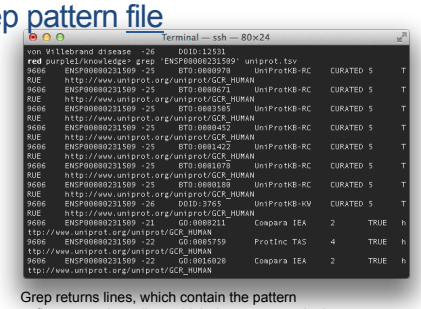
The tail command shows
the last 10 (or n) lines of a file

wc [-lmw] file (word count)



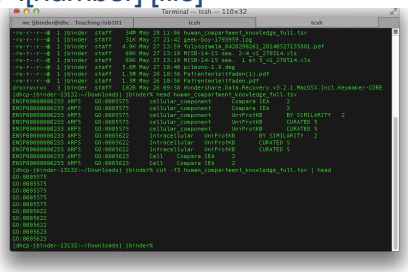
The wc command returns
lines/words/characters number of a file

grep pattern file



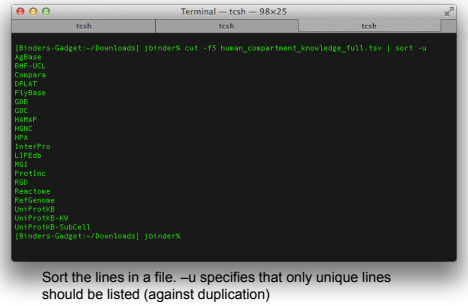
Grep returns lines, which contain the pattern
-v flag return those line, which does not contain the pattern

cut -f[number] [file]



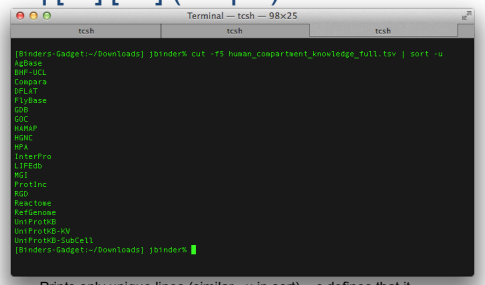
Cut returns only the specified columns.

sort –u [file]



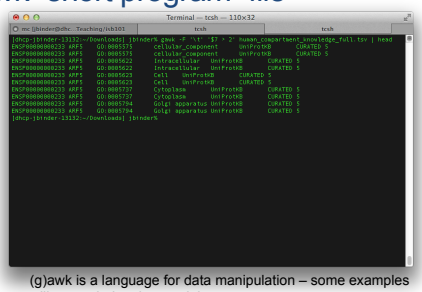
Sort the lines in a file. –u specifies that only unique lines
should be listed (against duplication)

uniq [–c] [file] (unique)



Prints only unique lines (similar –u in sort). –c defines that it
counts also the occurence

gawk “short program” file



(g)awk is a language for data manipulation – some examples
will come during the tutorial

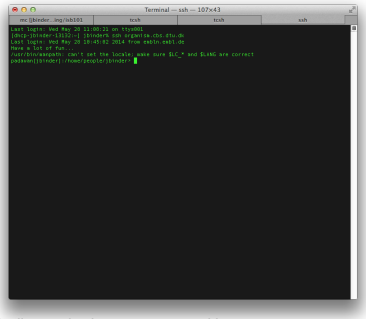
Take home message

- Unix/Linux is the most widely used operating system in bioinformatics
- Unix/Linux is used to handle files and to run analysis on very large experiments, by calling external programs
  - You can manipulate your files through command lines
  - You can launch programs by the command line (next talk)
- Google is the best way to find information about particular command lines

STEP 4

Setting up programs

ssh

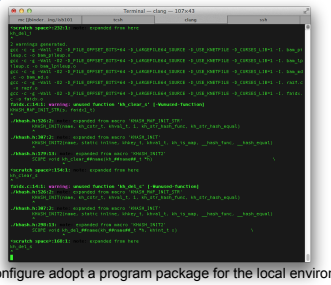


ssh allows to log in to a remote machine

chmod a+x file

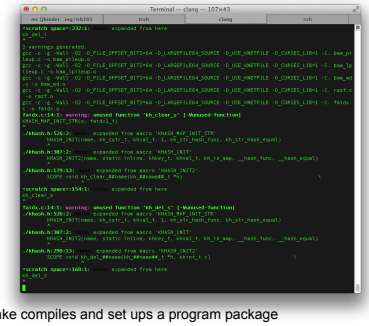
chmod changes permission of a file. Here it becomes executable

configure



configure adopt a program package for the local environment

make



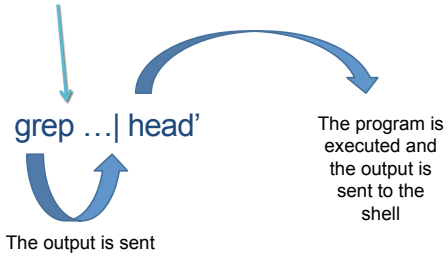
make compiles and set ups a program package

STEP 4

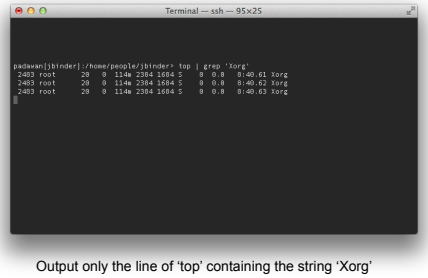
Pipelines and streams

CHAINING PROGRAMS

Pipes – Chaining programs

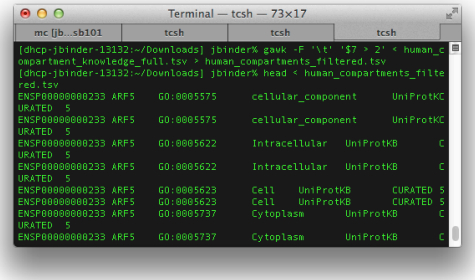


Checking whether Xorg is running: top | grep ‘Xorg’



Output only the line of ‘top’ containing the string ‘Xorg’

Redirecting standard input & output



< reads in a file, > write the results into a file

Introduction to bash scripting

- Repeating steps is time-consuming and error-prone
- Reproducible research ->
- An automated solution is needed
- UNIX – shell script

First script – Hello World

- The first line defines what kind of program will interpret this script
- echo – this command prints out text
- " denotes that it is a text – work also without it, but error prone

```
#!/bin/bash

echo "Hello World!"
```

Variables

- Storing often used texts, numbers
- Storing results of a calculation
- Remember to the $

```
#!/bin/bash

HW="Hello World!"

echo $HW
echo "It is boring to print" $HW
```

Conditionals

- Let you decide whether to perform an action or not, which based on the evaluation of the expression

```
#!/bin/bash

FILE="3_basic_conditionals.sh"

if [ -f $FILE ]; then
    echo "FILE:" $FILE "exists!"
fi
```

- An if-then-else construct:

```
if [ -f $FILE ]; then
    echo "FILE:" $FILE "exists!"
else
    echo "FILE:" $FILE "does not exists!"
fi
```

Streams and pipes - revisited

- Storing the names of the files in a directory

```
ls -1 > files_in_the_directory.txt
less files_in_the_directory.txt
```

- But it shows also lists `files_in_the_directory.txt`
- Piping the other way

```
grep -v files_in_the_directory.txt <
files_in_the_directory.txt | less
```

- A more clever solution:

```
ls -1 | grep -v files_in_the_directory.txt >
files_in_the_directory.txt
less files_in_the_directory.txt
```

Loops

- To do stuff the "same" stuff more than once

```
#!/bin/sh

echo "We count to ten!"

for (( i = 1 ; i <= 10 ; i++ )) ; do
    echo $i
done
```

```
#!/bin/sh

(( i = 1 ))
while (( i <= 10 )) ; do
    echo $i
    (( i++ ))
done
```

How do I execute a series of commands?
Bash scripting

- You can create scripts (e.g, "*my_series_of_command.sh*") executing a series of Unix commands

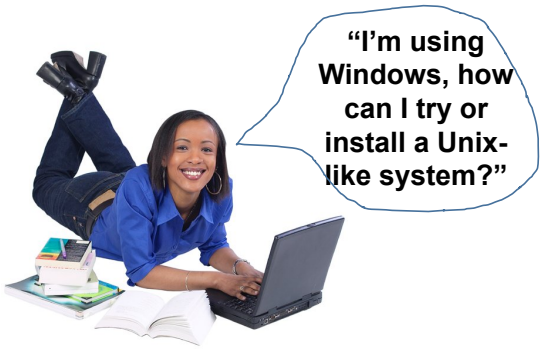
```
#!/bin/bash

for i in $(seq 1 5)
do
      echo "i is equal to: $i"
      ls -l
done
```

You can execute the scripts via: sh my_series_of_command.sh
or chmod a+x my_series_of_command.sh; ./my_series_of_command.sh

Good news:

- The recent Mac OS are Unix based.
- So you were using it without realizing it
- Mac OS has a shell, most of the commands (used in biology) are working directly!



Try Wubi!

- It will install Ubuntu on your Windows machine
- You install it like a normal program inside Windows
- When you will boot your computer, you will have the choice to run Ubuntu

Take home message

- Command lines are helpful to run programs or handle computers with no visual interface
- You have much more commands in the real life, helping you for a variety of tasks → Google is your friend
- Install Linux on your PC: Ubuntu

Acknowledgements

- Samuel Croset and others at EMBL for the Unix introduction slides.

Important parts

- Understand the file/directory structure of Unix (+ special characters . .. ~ /)
- Knowledge of basic commands
- Pipes, redirection (| < >)
- Basic manipulation and filtering of text files
- Writing simple scripts (conditionals, loops, printing out things to the console)