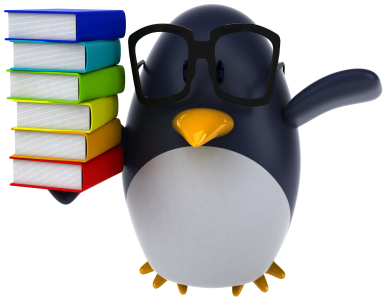
Monday– part 1

Introduction to Linux

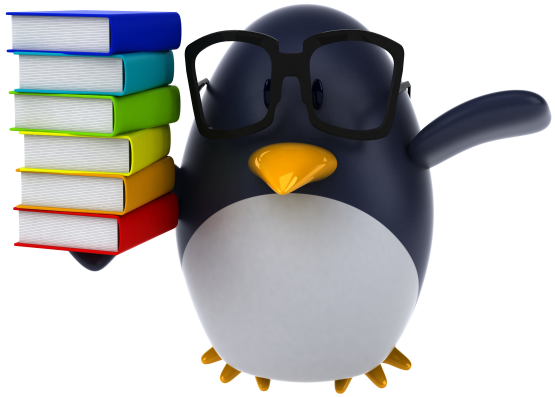
Jan Kosiński
Maciej Osowiecki

Faculty of Biology, Adam Mickiewicz University, Poznań
ideas4biology Ltd.



What is Linux used for?

- Machines running Linux can be clustered together to work on a single scientific task
- Linux can be used on your home computer (instead of / along with Windows)
- Most servers, web apps, websites or databases run on Linux



Why Linux?

- Majority of bioinformatics tools and software are developed only for Linux
- Scripting and system tools
- Command line tools (incredible flexibility, multiple jobs at once, hundreds of jobs in parallel)
- It's free (open source)

Loman N, Watson M, *So you want to be a computational biologist?*, Nature Biotechnology 31, 996–998 (2013)

Installing Linux on your computer

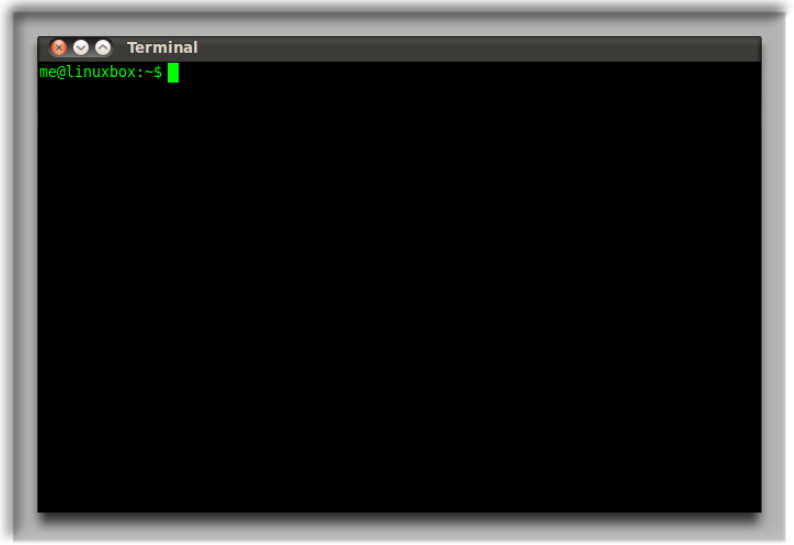
- find a distribution that suits you (e.g. Ubuntu, Mint, Mate, Debian, Manjaro)
- head to downloads section on your distribution's website (e.g. https://www.ubuntu.com/download/)
- download the .ISO file
- get a pendrive with 4GB+ of free space or a blank DVD
- download USB image burning software (Rufus or Universal USB Installer) and burn the image onto your pendrive. Step by step instructions for using Rufus can be found on ubuntu page after you start downloading the .iso file. For DVD's use any DVD burning software available for your system.restart your computer
- restart your computer
- press a key when "press any key to boot from CD" prompt is displayed
- If you boot straight into Windows instead of linux you have to change the default boot device in BIOS/UEFI. Search for "boot to bios/uefi windows 7/8/10" and put the USB/CD-ROM device on top of your boot list.
- 64-bit ubuntu should work with Microsoft Secure Boot technology. With other distributions try disabling secure boot in bios/uefi.
- in case you can't see your Windows installation on GRUB system list after installing linux, try running boot-repair from your linux installation. https://help.ubuntu.com/community/Boot-Repair
- if you experience problems with video output on default video driver read how to install proprietary NVIDIA/INTEL/AMD drivers for your distribution.
- any other problems can be usually resolved by adding "ubuntu" to your problem description in your search engine of choice

OR

google :D

Getting Started

- SHELL - command line interface
- Terminal - lets you interact with the shell



Getting started

- **man** *command*: display the on-line manual pages
- *command* — **help**, -**h**: display help

Getting started

- **pwd** - return working directory name
- **cd** - change directory
- **mkdir** *directoryName* - make directory
- **ls** - list directory contents
- **touch** *file* - make empty file
- **rm** *file* - remove file
- **cp** *file directory* - copy files
- **mv** *file directory* - move files (change name)

Getting started

Exercise 1

Create catalogue structure shown below. Find out how to create level3 directory using *mkdir* directly (single command, don't use *mkdir level1*, *cd level1*, etc.)

Exercise 2

Create an empty file called xyz.txt at level3. Copy xyz.txt to level2_1. Rename to abc.txt. Remove level2_3 directory.

Getting started

- **echo** *xyz* - write *xyz* to the standard output
- **cat** *file* - print files
- **more** *file* - print files
- **less** *file* - print files
- **wc** *file* - word, line, character, and byte count
- **head** *file* - display first lines of a file
- **tail** *file* - display the last part of a file

Getting started

- **ssh** - remote login program
- **scp** - remote file copy program
- **wget** *file* - download files from the Web

Getting started

Exercise 3

Download the Little Prince book from
http://bioinformatics-school.pl/littleprince.txt

Find the difference between *cat*, *more*, *less*, *head* and *tail* commands. Number the lines using *more*.

Count lines in book (don't use *more*).

Getting started

- **sort** *file* - sort lines of text files
- **uniq** *file* - filter out repeated lines in a file

Getting started

- **&** - run process in background
- **ps** - process status
- **bg** *pid* - resume suspended job and keep it running in the background
- **fg** *pid* - resume suspended job and keep it running in the foreground
- **jobs** - show suspended jobs
- **kill** -9 *pid* - terminate a process

Getting started

Exercise 4

Run *sleep 3600* command in background. Show all running processes using *ps.* Terminate the process.

Run *sleep 3600* command in foreground. Use Ctrl+Z to stop it. Show all stoped processes using jobs. Run process in background. Terminate the process.

Getting started

- **gunzip** *file* - decompression tool
- **gzip** *file* - compression tool
- **tar** -zxvf *file* - decompression of tar.gz files

Getting started

- **cut** - cut out selected portions of each line of a file
- **grep** - file pattern searcher
- **sed** s/expression1/expression2/g - replace expression1 with expression2

Text editor

- **nano** *file*

User groups and access rights

- **u** - user, **g** - group, **o** - others, **a** - all
- **r** - read, **w** - write, **x** - execute
- **chmod …** *file*
- … - e.g. g=rwx, a+x, u+w

Redirections

- **>** and **>>** - capture output from a command and send as input to another file
- **<** and **<<**
- **2>** - redirect errors

Pipelines

- Pipeline - sequence of processes chained together, so that the output of each process is used as input to the next one.
- e.g. sort *file* **|** uniq
- grep *„sth” file* **|** wc -l

Bash

```
#!/bin/bash

# variable "number"
number=3

# clear screen
clear

# "seq" is a simple application to generate ranges of numbers
# "$1" and "$2" are first and second arguments passed to our script
# we want to "do" something "for" every "x" in the number range
# $() tells the script to use the result of a command run inside of
the parentheses

for x in $(seq $1 $2);
do

   # Double parentheses are used for arithmetic operations
   # we want to check if dividing $i by $number gives us a reminder
equal to 0
   if [ $(( $x % $number )) -eq 0 ];

   then
       echo "${x} : true"
   else
       echo "$x"
   fi
done
```

Getting started

Exercise 5

Create a simple bash script using nano.

Script should be able to:

download and decompress the bed file from http://bioinformatics-school.pl/pssb_sample.bed.gz

replace „chr7” with „7” (sed)

create a file having two values in separate lines (number of features on + an - strand, use at least one redirection, one pipeline and grep). Values should be sorted in ascending order.

Getting started

Exercise 6*

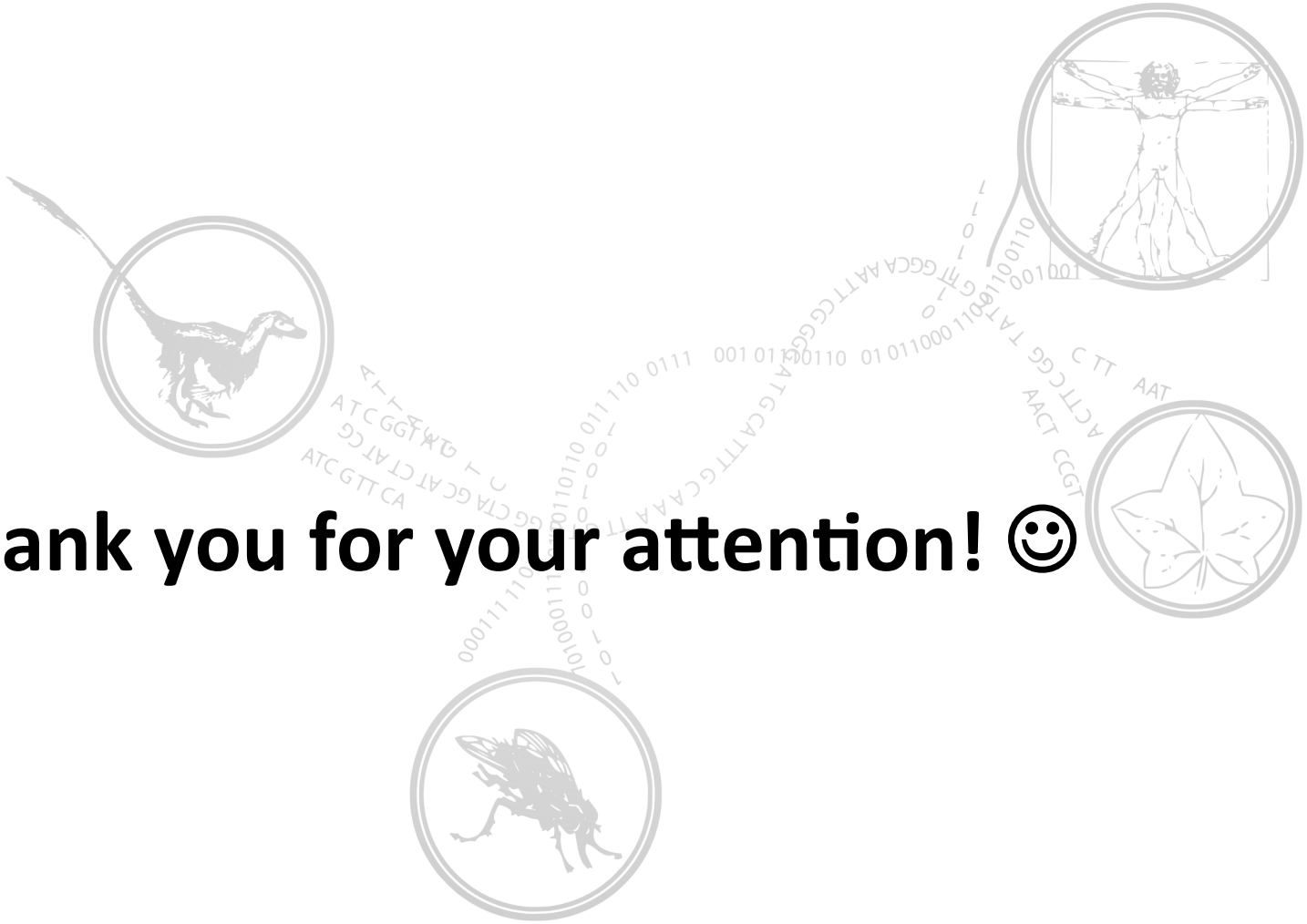
Sort **pssb_sample.bed** by start coordinates in descending order. Print only unique features from + strand. Count the lines.

Do it using single command.

Installing software

- Package managers: sudo apt-get install „*whatever*"
- Anaconda Installer (https://docs.continuum.io/anaconda/install/linux)
- Build the program from its source code (download sections of desired software):
  - tar -zxvf *file.tar.gz*
  - cd to your decompressed package
  - ./configure (configuration)
  - make (compilation)
  - make install (installation)

Looking for help



Thank you for your attention! ☺