Installing and updating software

Jeroen F. J. Laros

Leiden Genome Technology Center

Department of Human Genetics
Center for Human and Clinical Genetics

Updating

First of all, run the update command to make sure we download the latest versions.

```
1   $ sudo apt-get update
```

Listing 1 : Retrieve version information.

Updating

First of all, run the update command to make sure we download the latest versions.

```
1    $ sudo apt−get update
```

Listing 1 : Retrieve version information.

Once we know what can be upgraded, we can choose to upgrade all packages at once.

```
1    $ sudo apt−get upgrade
```

Listing 2 : Upgrade to the newest version.

Small packages

In many cases, the package name is equal to the command we want to use:

First we install a program called "**sl**".

```
1   $ sudo apt−get install sl
2   $ sl
```

Listing 3 : Insatllation of "sl".

Searching for packages

If we want to search for all packages that have anything to do
with "alignment":

```
1    $ apt-cache search alignment
2    bwa          - Burrows-Wheeler Aligner
3    samtools     - processing sequence alignments in SAM
4                   and BAM formats
5    seaview      - Multiple sequence alignment editor
6    sigma-align  - Simple greedy multiple alignment of
7                   non-coding DNA sequences
8    sim4         - tool for aligning cDNA and genomic DNA
9    wise         - comparison of biopolymers, commonly
10                  DNA and protein sequences
11   ...
```

Listing 4 : Searching packages.

Searching package content

Suppose we know a command, but we do not know which package to install:

```
1   $ apt−cache search fastaclip
```

Listing 5 : Failing installation.

This will return nothing (unlike our "**sl**" example).

Searching package content

Get a list of packages that provide the file "**exonerate**".

```
1   $ apt−file update
2   $ apt−file search fastaclip
3   exonerate : /usr/bin/fastaclip
```

Listing 6 : Search package content.

Searching package content

Get a list of packages that provide the file "**exonerate**".

```
1   $ apt−file update
2   $ apt−file search fastaclip
3   exonerate : / usr / bin / fastaclip
```

Listing 6 : Search package content.

Now we can install the package.

```
1   $ sudo apt−get install exonerate
```

Listing 7 : Install the right package.

Why manual installation?

Reasons for manual installation:

- There is no package available.

- We want the latest (development) version.

- We do not have permission to install something system wide.

Why manual installation?

Reasons for manual installation:

- There is no package available.
- We want the latest (development) version.
- We do not have permission to install something system wide.

So, we essentially do the same as we do for Windows.

- Download an *archive* from the internet.
- Extract the content.

Archives

Commonly used archiving programs:

- **tar** in combination with **gzip** or **bzip2**.
- **zip**.

Zip is commonly used for Windows and is easy to use for Linux.

```
1   $ unzip archivename.zip
```

Listing 8 : Unpack a zip archive.

Archives

Commonly used archiving programs:

- **tar** in combination with **gzip** or **bzip2**.
- **zip**.

Zip is commonly used for Windows and is easy to use for Linux.

```
1   $ unzip archivename.zip
```

Listing 8 : Unpack a zip archive.

```
1   $ zip −r archivename.zip directoryname
```

Listing 9 : Create a zip archive.

The option **−r** stands for *recursive*.

Archives

Tar is an old program used to pack lots of files into one archive to put it on tape (hence the name Tape ARchive).

Archives

Tar is an old program used to pack lots of files into one archive
to put it on tape (hence the name Tape ARchive).

A Tar archive is *not* compressed, but seperate compression util-
ities are used.

- Gzip is fast.
- Bzip2 is slow, but compresses better than Gzip.

Archives

Tar is an old program used to pack lots of files into one archive to put it on tape (hence the name Tape ARchive).

A Tar archive is *not* compressed, but seperate compression utilities are used.

- Gzip is fast.
- Bzip2 is slow, but compresses better than Gzip.

Method	Extension	Alternative
Gzip	`.tar.gz`	`.tgz`
Bzip2	`.tar.bz2`	

Table 1 : Extensions of compressed tar archives.

Archives

Commonly used options for **tar**:

Option	Description
-x	Extract.
-c	Create.
-z	Use the **gzip** compression utility.
-j	Use the **bzip2** compression utility.
-v	Be verbose.
-f <filename>	The name of the archive.

Table 2 : Commonly used Tar options.

Example extraction

Extraction of a **tar.gz** file (**tar** compressed with **gzip**):

```
1   $ tar −xzvf myarchive.tar.gz
```

Listing 10 : Extract a compressed archive.

Example extraction

Extraction of a **tar.gz** file (**tar** compressed with **gzip**):

```
1   $ tar −xzvf myarchive.tar.gz
```

Listing 10 : Extract a compressed archive.

Creation of a **tar.gz** file (**tar** compressed with **gzip**):

```
1   $ tar −czvf myarchive.tar.gz directoryname
```

Listing 11 : Create a compressed archive.

Example extraction

Extraction of a **tar.bz2** file (**tar** compressed with **bzip2**):

```
1    $ tar −xjvf myarchive.tar.bz2
```

Listing 12 : Extract a compressed archive.

Example extraction

Extraction of a **tar.bz2** file (**tar** compressed with **bzip2**):

```
1   $ tar −xjvf myarchive.tar.bz2
```

Listing 12 : Extract a compressed archive.

Creation of a **tar.bz2** file (**tar** compressed with **bzip2**):

```
1   $ tar −cjvf myarchive.tar.bz2 directoryname
```

Listing 13 : Create a compressed archive.

Installing software: Manual installation

If there are no *executables* available, we have to make them ourselves.

- Go to the Bowtie website.
- Click on the link in the "latest release" section.
- Click on the link that ends with **.src.zip**.
- Choose "save file".
- Open a terminal.

Installing software: Manual installation

If there are no *executables* available, we have to make them ourselves.

- Go to the Bowtie website.
- Click on the link in the "latest release" section.
- Click on the link that ends with **.src.zip**.
- Choose "save file".
- Open a terminal.

```
1   $ unzip bowtie−<version>−src.zip
2   $ cd bowtie−<version>
3   $ make
4   $ ./bowtie −h
```

Listing 14 : Compiling bowtie from source.

About this part of the course

- We are not going to cover the "easy" part (browsing the internet, sending mail, playing music, . . .).

- We focus on the *command line*, since this is the most powerful interface to the tools we need.

- We are going to connect to other machines (servers that have more memory or computing power).

Michiel van Galen
Jeroen Laros

https://humgenprojects.lumc.nl/trac/humgenprojects/wiki/NGS-intro