Linux Software Installation
Session 2

Qi Sun
Bioinformatics Facility

Installation as non-root user

- **Change installation directory;**
  - Default procedure normally gives “permission denied” error.
- **Sometimes not possible;**
  - Installation path is hard coded;
  - Service daemons, e.g. mysql;

- **Ask system admin;**
- **Docker**

Python

Two versions of Python

Python2 is the legacy but still very popular

Python3 is the current version.

They are not compatible, like two different software sharing the same name.

Running software

python 2

python myscript.py

python 3

python3 myscript.py

Installing software

python 2

pip install myPackage

python 3

pip3 install myPackage

Check shebang line, to make sure it points to right python

A typical Python software directory

```
ls pyGenomeTracks-2.0

bin   lib   lib64
```

```
ls pyGenomeTracks-2.0

bin  lib  lib64
```

To run the software:

```
export PATH=/programs/pyGenomeTracks-2.0/bin:$PATH

export PYTHONPATH=/programs/pyGenomeTracks-
2.0/lib64/python2.7/site-packages:/programs/pyGenomeTracks-
2.0/lib/python2.7/site-packages/
```

Other library files

If there is a problem, modify PYTHONPATH or files in $HOME/.local/lib

1. **$PYTHONPATH** directories

2. **$HOME/.local/lib**

3. Python **sys.path** directories

Defined by you;

Defined by Python, but you can change files in directory;

Defined by Python;

Check which python module is being used

For example:

>>> import numpy

>>> print numpy.__file__
/usr/lib64/python2.7/site-packages/numpy/__init__.pyc

>>> print numpy.__version__
1.14.3

$PYTHONPATH frequently causes problem. E.g. python2 and python3 share the same $PYTHONPATH

* run these commands in “python” prompt

PIP – A tool for installing/managing Python packages

- PIP default to use PYPI repository;

- PIP vs PIP3

  PIP -> python2

  PIP3 -> python3

Two ways to run “pip install” as non-root users

```
pip install deepTools --user
```

Installed in
$HOME/.local/bin
$HOME/.local/lib & lib64

* Suitable for personal installation

```
pip install deepTools \
--install-option="--prefix=mydir" \
--ignore-installed
```

Installed in
mydir/bin
mydir/lib & lib64

* Suitable for installation for a group

PIP would also install/update all required modules

e.g. **requirements.txt** file included in deepTools package

```
numpy>=1.9.0
scipy>=0.17.0
matplotlib>=2.1.2
pysam>=0.14.0
py2bit>=0.2.0
numpydoc>=0.5
pyBigWig>=0.2.1
plotly>=1.9.0
```

How PIP handle required modules?

Default behavior:

Missing modules -> Install

Existing modules

- **Version acceptable: skip**
- **Version not acceptable: uninstall and install right version**

Existing modules

- Version acceptable: skip
- Version not acceptable: uninstall and install right version

- For root user, there is a risk of downgrade a module to an old version;
- For non-root user, you will get an error message of permission denied;

PIP parameters to change default behavior

--user

- Will not un-install existing modules;
- Skip modules meet requirement;
- Install required module that do not exist;

Parameters to change default behavior

--ignore-installed (-I)

- Install all required modules, present or not;
- Together with **--install-option="--prefix=mydir"**

PIP parameters to change default behavior

--upgrade : Upgrade package and all required modules to latest version

Uninstall python module

```
pip uninstall numpy
```

Or, simply delete the ~/.local/lib/ directory or sub-directories

Install python package not present in PYPI

-- follow instructions by author

- Download files;
- Run command:

python setup.py install --prefix=/home/qisun/myPython

- Set PYTHONPATH before running code;

PERL

Two versions of PERL on BioHPC

- **5.22: /usr/local/bin/perl (Default) :**
- **5.16: /usr/bin/perl :**

*** /usr/bin/perl is distributed with the system.**

What you need to know about PERL versions

- **/usr/bin/perl** : **default PERL on most Linux system;**
- **/usr/local/bin/perl**: **default PERL on BioHPC**

We only install modules with /usr/local/bin/perl

When running PERL scripts:

```
perl  myscript.pl          ## you are fine

./myscript.pl              ## check shebang line
```

Shebang lines that work with BioHPC PERL

Use default PERL

#/usr/bin/env perl

#/usr/local/bin/perl

More developers start using this shebang line

Use /usr/bin/perl

#!/usr/bin/perl

* This might give you error message complaining missing modules

PERL

Check whether a PERL module is present

```
perl -e "use Bio::Seq"
```

or

```
perldoc Bio::Seq
```

Change the shebang line of all *.pl files

Run this command to change files within a directory

```
sed -i 's/\/usr\/bin\/perl/\/usr\/local\/bin\/perl/' *.pl
```

Install PERL modules from CPAN

https://cbsu.tc.cornell.edu/lab/doc/Install_PERL_modules.pdf

1. Configure cpan - specify the directory to install;

2. Run cpan:

```
install XML::Simple
or
force install XML::Simple
```

Specify paths of PERL modules installed by you

export PERL5LIB=$HOME/perl/lib/perl5

PERL would search these paths to find a module:

1. **$PERL5LIB defined directories;**

2. **@INC defined directories;**
   Use this command to check:
   perl "-e print @INC“

R

Different versions of R on BioHPC

```
ls /programs |grep "^R-"
R-2.13.0
R-2.15.0
R-2.15.2
R-3.0.1
R-3.0.1a
R-3.0.2
R-3.1.0
R-3.2.2
R-3.2.2p
R-3.2.2s
R-3.2.5
R-3.2.5s
R-3.3.2
R-3.3.2s
R-3.4.1
R-3.4.1s
R-3.4.2
R-3.4.2s
```

R Command: R

Default: R-3.4.2

To use another version of R:
export PATH=/programs/R-3.4.1/bin:$PATH

- This is also applicable to using Rscript command to run R script.

On BioHPC, two separate installations for each R version

R-3.4.2 (Default): parallel BLAS library

R-3.4.2s: regular BLAS library

- Parallel BLAS (Basic Linear Algebra Subprograms) reduces computing time for linear algebra calls by a factor of 3 or more;
- Parallel BLAS could cause '**illegal operand**' errors for some packages;

Install R packages from CRAN

```
# install R package
install.packages("GD")

# load R package
library(GD)
```

* R packages are only installed to the specific version of R you are using.

Check version and path of R packages

Check version: **packageVersion("edgeR")**

Find package location: **find.package("edgeR")**

Check search path: **.libPaths()**

Install R packages from GitHub

```
llibrary(devtools)

install_github("rqtl/qtl2geno")
```

C / C++

Versions of the GCC compiler

Default gcc: 4.8.5

Other versions: 5.5.0, 7.3.0 (/usr/local/gcc-*)

To use a different version of GCC

```
export PATH=/usr/local/gcc-7.3.0/bin:$PATH

export LD_LIBRARY_PATH=/usr/local/gcc-7.3.0/lib:/usr/local/gcc-7.3.0/lib64
```

General installation procedure

1. **Configure** - customize/verify compilation instruction;

2. **Compile** - from source code to binary code;

3. **Install** - put the executable/library in right locations;

Pre-built Binary

vs

Compilation From Source Code

Why compilation?

1. Pre-built binary code not provided;

2. Compilation would optimize CPU/GPU usage
(e.g. software developed for GPU)

* Sofware package manager, e.g. conda cannot do compilation, only install pre-built code;

Configure – Part 1

```
./configure
```

1. **Specify installation directory;**

   To change: ./configure --prefix=/home/xxxxx/bin

2. **Verify the compiler, libraries;**

   Deal with it if libraries missing or not in right version
   (e.g. set LD_LIBRARY_PATH)

Configure – Part 2

Another system for configuration:

cmake

or

cmake3

Most software requires cmake v3. The command on BioHPC is cmake3

Specify installation directory

```
cmake3 -DCMAKE_INSTALL_PREFIX:PATH=/home/xxxx/bin
```

Configure – Part 3

Manually edit the makefile

Install

```
make install
```

To run the software:

- Modiy $PATH or use full path of the executable;
- Modify $LD_LIBRARY_PATH if custom libraries installed;

Environment variable for executables path

$PATH

Environment variables for custom library path

PYTHON	**$PYTHONPATH**
PERL	**$PERL5LIB**
C	**$LD_LIBRARY_PATH**

Set environment variable:
export PATH=mydirectory:**$**PATH

Check environment variable:
echo **$**PATH