

# Easy installations and virtual environments with conda

author: Maxime Borry date: 24/01/2020

## Follow the presentation

[maxibor.github.io/conda-presentation](https://maxibor.github.io/conda-presentation)

## What does it mean to install a program ?

-> Copy/move the executable of the program in the PATH

What is the PATH ?

```
borry@mpi-sdag1:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
games:/usr/local/games:/opt/dell/srvadmin/bin
```

- /usr/local/sbin
- /usr/local/bin
- /usr/sbin
- /usr/bin
- ...

When can't "install" programmes on a computer, it's because you don't have write access to these directories

## What's in my path ?

For example, let's have a look at /usr/bin !

```
borry@mpi-sdag1:~$ ls -1 /usr/bin
...
head
headerdoc2html
heap
heap32
hexdump
hidutil
hiutil
host
...
```

## Great, but what about Conda ?

I have a confession: this is not my real path

```
borry@mpi-sdag1:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
games:/usr/local/games:/opt/dell/srvadmin/bin
```

Here is my real path

```
borry@mpi-sdag1:~$ echo $PATH
/projects1/clusterhomes/borry/miniconda3/bin:/usr/local/
sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
games:/usr/local/games:/opt/dell/srvadmin/bin
```

The location where executables will be installed by Conda

```
/projects1/clusterhomes/borry/miniconda3/bin
```

## Conda

“Conda is an open source, **cross-platform**, **language-agnostic package and environment management system** distributed by Continuum Analytics.”

- Free
- **No admin rights required**
- Tons of bioinformatic packages available
- Easy to install

## Install conda

Two distributions:

- Anaconda (fully featured, heavy)
- **Miniconda** (bare minimum)

Installation for Mac

```
borry@maxime:~$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-x86_64.sh -O ~/miniconda.sh
borry@maxime:~$ bash ~/miniconda.sh
```

Installation for Linux

```
borry@mpi-sdag1:~$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda.sh
borry@maxime:~$ bash ~/miniconda.sh
```

## Using conda

### Install a package

When you install a package, conda automatically handles the installation of all its dependencies

Jupyter/IPython notebook

```
(base)borry@mpi-sdag1:~$ conda install jupyter
```

### Install a package from a specific channel

BWA

```
(base)borry@mpi-sdag1:~$ conda install -c bioconda bwa
```

Bioconda: Channel for bioinformatics packages

Bioconda: sustainable and comprehensive software distribution for the life sciences

## Everything with conda is an environment

An environment is an isolated sandbox that allows a fine control on program's versions and dependencies

By default, you're in the base environment

```
(base)borry@mpi-sdag1:~$ conda env list
# conda environments:
#
base          *  /projects1/clusterhomes/borry/miniconda3
```

But can create new environments...

```
(base)borry@mpi-sdag1:~$ conda create -n myEnvName
```

## Everything with conda is an environment(2)

Change your current environment (activate)

```
(base)borry@mpi-sdag1:~$ conda activate myEnvName
```

Note that your prompt now includes the name of the active environment

Install a package in this environment, for example, RStudio

```
(myEnvName) borry@mpi-sdag1:~$ conda install -c r rstudio
```

And go back to the base environment

```
(myEnvName) borry@mpi-sdag1:~$ conda deactivate
```

The packages installed in one environment are not accessible from outside !

## Let's create an environment for MetaPhlan2 and Krona

Krona plots

```
(base)borry@mpi-sdag1:~$ conda create -n metaphlan
(base)borry@mpi-sdag1:~$ conda activate metaphlan
(metaphlan) borry@mpi-sdag1:~$ conda install -c bioconda metaphlan2
# Now you can use metaphlan !
(metaphlan) conda install -c bioconda krona
```

Now let's share our environment !

```
(metaphlan) borry@mpi-sdag1:~$ conda env export > metaphlan_env.yml
```

And recreate it on another machine from the environment file

```
(base)borry@maxime:~$ conda env create -f metaphlan_env.yml
```

## Other useful conda commands

List installed packages and versions in an environment

```
(metaphlan) borry@mpi-sdag1:~$ conda list
```

Uninstall a package (here Krona)

```
(metaphlan) borry@mpi-sdag1:~$ conda remove krona
```

Delete an environment

```
(base)borry@mpi-sdag1:~$ conda env remove -n metaphlan
```

Conda documentation: [conda.io/docs](https://conda.io/docs)