

Working with Conda environments

Most of the times, when working on shared workstations or servers or clusters, you are not given administrative permissions. That is, you are not allowed to freely roam around and manage directories or install softwares. This is basically for your (and everyone's) safety, so that for example important files are not damaged by mistake.

Though, you will need for sure to install new softwares for your analyses. So how can you be able to get programmes without asking every time your administrator? Conda comes into help!!

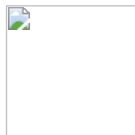


Table of contents

- [Conda is a package and environment management system](#)
- [Create a new Conda environment](#)
- [Install a software through Conda](#)
- [List all Conda environment and installed packages](#)
- [Export a Conda environment to a YAML file](#)
- [Remove a Conda environment](#)
- [Extension: what about Mamba?](#)

Command cheat sheet

Command	Description
conda create	Create a new Conda environment
conda activate	Activate a Conda environment
conda deactivate	Deactivate a Conda environment
conda install	Install a software and required dependencies through Conda
conda env list	List all the available Conda environments
conda list	List all the available packages in a Conda environment
conda env export	Export a Conda environment to a YAML file
conda remove	Remove a Conda environment and all the included packages

Conda is a package and environment management system

[Conda](#) is a software that helps you install other softwares by also taking care of managing all the requeried dependencies.

Besides helping in installing without administrative privileges, Conda is also really useful to resolve conflicts between programming language and package versions. So, for example, if you need two softwares, one requiring python v2.7 and one requiring python v3.1, with Conda you can install both of them with litte or no effort.

Conda accomplishes this by **operating within environments**.

You can think of a Conda environment as... a natural environment, surprisingly. Like the Amazon, the Sahara and the Mediterranean environments—which are section of the natural world with their own species assemblage, temperature and rain conditions—each Conda environments are sections (directories) of your computer with their own packages and versions of programming languages. The only difference with natural environments is that Conda environments are independent one to the others and do not exchange information (or data).

[↑ Table of contents ↑](#)

Create a new Conda environment

To start working with Conda and get a new software installed, you first need to **create the environment** in which the software will be hosted:

```
# create a new Conda environment
$ conda create -n my_env_name
```

The newly created environment is now empty and just waits to be filled up with installations.

If you have a configuration file ([YAML file](#), which is indicated by the `.yaml` extension) of another Conda environment, just type:

```
# create a new Conda environment
$ conda env create -f environment.yaml
```

This is the easiest way to clone already-existing Conda environments: you just need the relative YAML file (see [here](#) to know how to get a YAML file from a Conda environment).

[↑ Table of contents ↑](#)

Install a software through Conda

Before installing a software into a Conda environment, make sure the environment is active:

```
# activate a new Conda environment
$ conda activate my_env_name
```

Once the environment has been activated, the terminal will display its name in brackets before the prompt symbol.

To **install a new software**, just check in the [Anaconda website](#) (the repository where tons of data science packages are provided) if the package is available as a Conda distribution (very advanced tip: just google your software name followed by "conda" to check ☺). Then follow the instruction on the website.

```
# to install IQTREE2 with Conda, first of all let's create an environment and activate it
$ conda create -n iqtrees_env
...
$ conda activate iqtrees_env
...
# then just follow the instructions on the Anaconda website of the iqtrees package. Thus, let's type the following. Note tha
(iqtrees_env) $ conda install -c bioconda iqtrees
...
```

If you want to quit the working Conda environment, just type:

```
# activate a Conda environment
$ conda deactivate
```

[↑ Table of contents ↑](#)

List all Conda environments or installed packages

After having worked for some times with bioinformatics, you may end up with tons of different conda environments. Thus, it is likely you won't remember all of them and want to check what you have already installed. To **list all the Conda environments** you have created, just type:

```
$ conda env list
# conda environments:
#
base          * /home/filonico/anaconda3
iqtrees_env    /home/filonico/anaconda3/envs/mamba_env
iqtrees_env    /home/filonico/anaconda3/envs/iqtrees_env
```

Note that the active environemnt is marked with an asterisk *. Also, the `base` environment is created by default during the Conda installation and contains a series of commonly-used bioinformatic packages.

To **get the list of installed packages** in a Conda environment, type:

```
# if the environment is already active
(iqtrees_env) $ conda list
# packages in environment at /home/filonico/anaconda3/envs/iqtrees_env:
#
# Name                          Version          Build      Channel
_ libgcc_mutex                  0.1              main
_ openmp_mutex                  5.1              1_gnu
iqtrees                         2.1.4_beta       hdcc8f71_0 bioconda
libgcc-ng                      11.2.0           h1234567_1
libgomp                        11.2.0           h1234567_1
libstdc++-ng                   11.2.0           h1234567_1
zlib                           1.2.13           h5ee18b_0

# if the environment is not active
$ conda list -n my_env_name
...
```

[↑ Table of contents ↑](#)

Export a Conda environment to a YAML file

Sometimes, it is extremely useful to be able to share a Conda environment with someone else, so they are able to recreate the environment with the same packages. To this purpose, you need to **export your Conda environment** and create a [YAML file](#):

```
# be sure that the environment you want to export is active
(my_env_name) $ conda env export > my_env_name.yml
```

See [here](#) to know how to create an environment from a starting YAML file.

[↑ Table of contents ↑](#)

Remove a Conda environment

If you wish to **remove a Conda environment** that you no longer use or that turned into a messy collection of packages, type:

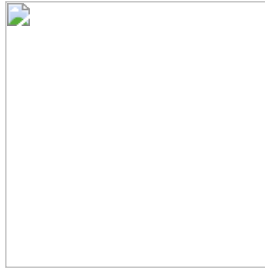
```
$ conda remove -n my_env_name --all
```

[↑ Table of contents ↑](#)

Extension: what about Mamba?

All right, Conda is tremendously helpful for bioinformaticians to manage software installations and the relative dependencies. However, sometimes even Conda struggles with installing certain packages and you may spend hours in trying to fix your environment.

To overcome such difficulties, you can try to use **another environment and package manager**, [Mamba](#) (there must be something going on with herpetology here...). You can easily install it through the normal Conda pipeline, but then you will rely on Mamba to install softwares and dependencies. Apparently, it is faster and more efficient in solving environments than Conda.



Mamba has the same commands as Conda, but has indeed a far **more fancy graphical output!** So once installed, you can for example install softwares by typing:

```
(iqtree_env) $ mamba install -c bioconda iqtree
```

I would like to thank [for-giobbe](#) for sharing this package manager with me!

[↑ Table of contents ↑](#)