Conda & Reproducibility

Wim R.M. Cardoen
Email: $(prefix)[at]gmail[dot]com
where
prefix='wcardoen'

November 2, 2022

I Reproducible science using conda

Conda environments can be used to make python software installations reproducible, i.e. to generate python environments which bear identical version numbers for their python packages.

In a first step, we will create an environment using the latest version of miniconda3[1] installed on a machine running the Ubuntu 18.04 OS. Subsequently, the settings of this newly created conda environment will be exported into yaml file. In the final step, we will load the settings in a new conda environment on a CHPC node running the Rocky8 OS.

I.1 Creation of the reproducible environment

We assume that miniconda3 has been installed and has been set up in such a way that it allows to support more than one conda environment (the default conda environment bears the name `base`).

The most common way to support multiple conda environments is to invoke `conda init` post the miniconda3 installation. Unfortunately, this procedure modifies a user's existing startup shell (`.bashrc/.tcshrc`) on a **permanent** basis: it appends a block of shell code in the startup shell and forces the latest miniconda3 installation to become the default. At best, this approach may serve an individual user, but it is not apt for CHPC's cluster environment.

Therefore, we strongly recommend to use CHPC's lua module template which does not modify the startup shell, supports multiple conda environments and allows the user the flexibility to maintain different miniconda distributions next to one another.

In the following coding block, an environment with the name `genscience` is created. The command `conda activate genscience` allows one to enter the `genscience` environment. Subsequent an array of Python packages (`numpy,..., statsmodels`) as well as `texlive-core` will be installed in the `genscience` environment. The command `conda deactivate` forces one to leave the `genscience` environment and return to the `base` environment.

```
# Create a simple env: genscience
module load py39_4.12.0
# Create a simple env: genscience
conda create -y -n genscience
# Activate the genscience environment
conda activate genscience
# Install an array of packages
conda install -y -c anaconda numpy scipy matplotlib
conda install -y -c anaconda pandas scikit-learn scikit-learn-intelex
conda install -y -c anaconda jupyter
conda install -y -c anaconda xarray
conda install -y -c bokeh bokeh
```

[1] can be exchanged interchangeably for anaconda3 for every instance in the document

```
conda install -y -c conda-forge dask
conda install -y -c conda-forge gdal
conda install -y -c conda-forge jupyterlab voila
conda install -y -c conda-forge jupyterlab-latex
conda install -y -c conda-forge scikit-image
conda install -y -c conda-forge statsmodels
conda install -y -c conda-forge texlive-core
# Deactivate the genscience environment
conda deactivate
```

The command `conda list` displays the environments that are accessible to the miniconda3 installation. The symbol ('*') prepend the directory where the currently activated environment is stored.

```
# Conda List:
# ----------
conda list

# Find all the Conda envs:
# ------------------------
base                  *  /home/sleipnir/software/pkg/mini3/py39_4.12.0
genscience               /home/sleipnir/software/pkg/mini3/py39_4.12.0/envs/
   genscience


sleipnir@ragnarok:~$ conda activate genscience
(genscience) sleipnir@ragnarok:~$ conda env list
# conda environments:
#
base                     /home/sleipnir/software/pkg/mini3/py39_4.12.0
genscience            *  /home/sleipnir/software/pkg/mini3/py39_4.12.0/envs/
   genscience
conda deactivate
```

I.2 Export of a conda environment

The information of the `genscience` environment can be easily stored in a file. The following command stores the details of the `genscience` environment in the yaml file `genscience.py39.yml`.

```
# Export the genscience env into a YAML file:
# -------------------------------------------
conda env export -v -n genscience -f genscience.py39.yml
```

I.3 Reproduce an environement based on a YAML file

In what follows we will recreate our environment on a new machine. After loading an existing anaconda distribution we are able to generate a Python environment based on the previously exported yaml file.

```
module use ~/EigenModules
module load myanaconda/2020.11

[u0253283@kingspeak5:~]$ conda env list
# conda environments:
#
base                  *  /uufs/chpc.utah.edu/common/home/u0253283/software/pkg/
   anaconda3/2020.11
```

```
conda env create -n genscience -f $HOME/genscience.py39.yml

(genscience) [u0253283@kingspeak5:~]$ python3
Python 3.10.4 (main, Mar 31 2022, 08:41:55) [GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> np.__version__
'1.22.3'
>>> import statsmodels
>>> statsmodels.__version__
'0.13.2'
```