

Screwing up conda environments

Robert Haase

Don't try this
at home.

October 2022

Broken [conda] environments

- Common case of confusion in the Python ecosystem
- Happens to everyone – it's just a matter of time.
- Early symptom: You install a package and afterwards, it's not installed.

The image is a collage of three screenshots illustrating a broken conda environment.

Left Screenshot: A terminal window showing the MAMBA logo and the command `mamba install stackview -c conda-forge`. The output shows the installation progress for various packages, including `conda-forge/win-64`, `conda-forge/noarch`, `pkgs/msys2/win-64`, `pkgs/r/win-64`, `pkgs/r/noarch`, `pkgs/main/win-64`, `pkgs/msys2/noarch`, and `pkgs/main/noarch`. The status for each package is "Using cache" or "No change".

Middle Screenshot: A terminal window showing the output of `mamba install stackview -c conda-forge`. The output lists the installed packages and their versions, including `toolz`, `tornado`, `traitlets`, `typing_extensions`, `tzdata`, `ucrt`, `unicodedata2`, `vc`, `vs2015_runtime`, `wcwidth`, `wheel`, `widgetsnbextension`, `xorg-libxau`, `xorg-libxmdcp`, `xz`, `yaml`, `zeromq`, `zfp`, `zipp`, `zlib-ng`, and `zstd`. The summary shows 142 packages installed and a total download of 0 B.

Right Screenshot: A JupyterLab interface showing a Jupyter Notebook with a cell containing the code `import stackview`. The cell has been executed, but it has resulted in a `ModuleNotFoundError` with the message "No module named 'stackview'". The error traceback shows the error occurred in the first line of the cell.

Background

- The base-environment is special.
- You install packages into the base environment, if you do not run `conda activate my_env`
- You can run base-software from within other environments

Base

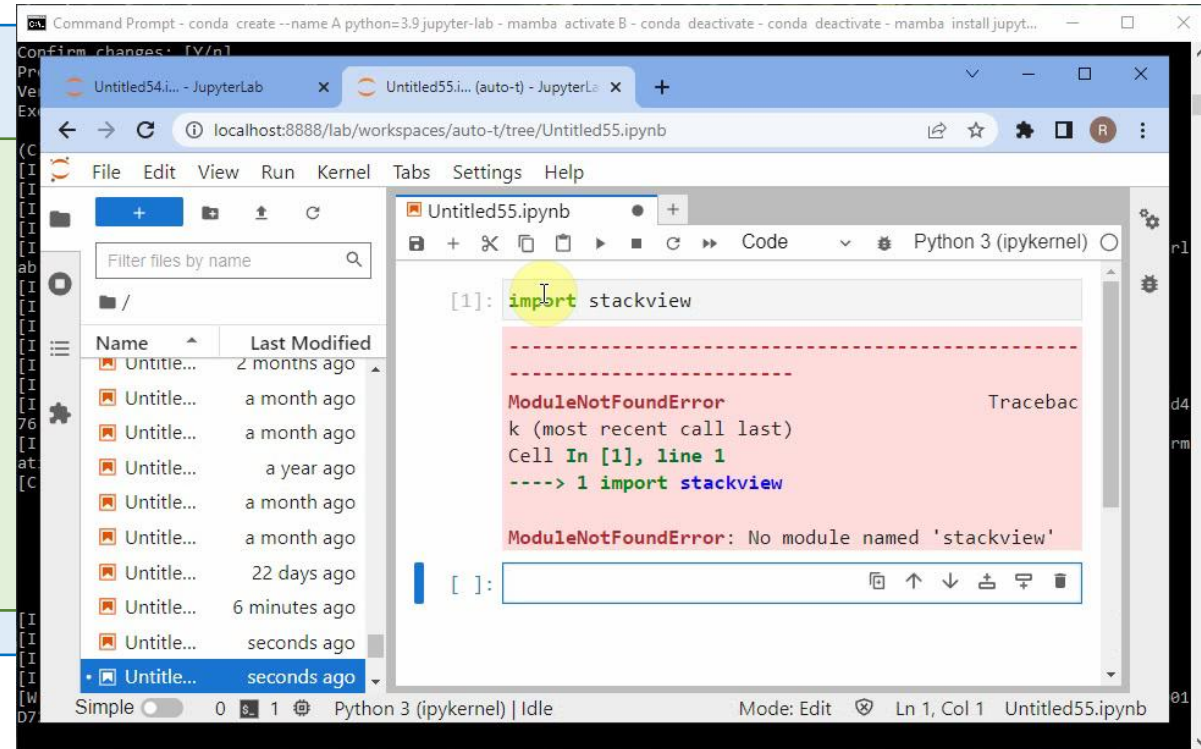
Installed packages:

Jupyter lab

my_env

Installed packages:

stackview



Broken [conda] environments

- The only cure: Uninstall and reinstall [Ana]conda.



Huge effort.

- Alternatively:
 - Create a new environment for every project.
 - `conda activate my_env`
 - Do not install project-specific stuff in the base-environment.



Little effort.

Documenting dependencies

- Maintain a document with the dependencies (and versions) you need in your project!
 - The conda way
 - The pip way

```
environment.yml - Notepad
File Edit Format View Help
name: cellpose-test
channels:
  - conda-forge
  - pytorch
dependencies:
  - python=3.8
  - pytorch=1.12
  - devbio-napari
  - cellpose-napari
Ln 3, C 100% Windows (CRLF) UTF-8
```

In case your environment is screwed up, you can rebuild it any time.

```
requirements.txt - Notepad
File Edit Format View Help
numpy>=1.21
bandas~=1.4.0
napari~=0.4.11
matplotlib~=3.5.0
scikit-image~=0.19
umap-learn~=0.5.1
hdbscan~=0.8.27
scikit-learn~=1.0
numba~=0.55.1
Ln 2, 100% Windows (CRLF) UTF-8
```

`conda env create -f environment.yml`

`pip install -r requirements.txt`

Documenting dependencies

- ... the complete way.

```
conda env export > environment.yml
```

```
environment.yml - Notepad
File Edit Format View Help
name: bio_39
channels:
- conda-forge
- defaults
dependencies:
- alabaster=0.7.12=py_0
- anyio=3.6.1=pyhd8ed1ab_1
- aom=3.5.0=h63175ca_0
- apoc-backend=0.10.0=pyhd8ed1a
- appdirs=1.4.4=pyh9f0ad1d_0
- argon2-cffi=21.3.0=pyhd8ed1ab
- argon2-cffi-bindings=21.2.0=p
- asciitree=0.3.3=py_2
- asttokens=2.0.8=pyhd8ed1ab_0
- attrs=22.1.0=pyh71513ae_1
- autopep8=1.7.0=pyhd8ed1ab_0
- babel=2.10.3=pyhd8ed1ab_0
- backcall=0.2.0=pyh9f0ad1d_0
- backports=1.0=py_2
Ln 391, Col 46 100% Wind
```

```
environment.yml - Notepad
File Edit Format View Help
- win_inet_pton=1.1.0=py39hcbf5309_4
- winpty=0.4.3=4
- wrapt=1.14.1=py39hb82d6ee_0
- x264=1!164.3095=h8ffe710_2
- x265=3.5=h2d74725_3
- xorg-libxau=1.0.9=hcd874cb_0
- xorg-libxdmcp=1.1.3=hcd874cb_0
- xz=5.2.6=h8d14728_0
- yaml=0.2.5=h8ffe710_2
- zarr=2.13.3=pyhd8ed1ab_0
- zeromq=4.3.4=h0e60522_1
- zfp=1.0.0=h0e60522_1
- zict=2.2.0=pyhd8ed1ab_0
- zipp=3.9.0=pyhd8ed1ab_0
- zlib-ng=2.0.6=h8ffe710_0
- zstd=1.5.2=h7755175_4
prefix: C:\Users\rober\mambaforge\envs\bio_39
Ln 391, Col 46 100% Windows (CRLF) UTF-8
```

Excellent way to document which dependencies were *actually* used...

It is *questionable* if re-creating an environment from this yml file works.