Ana/Mini/Conda

or “How to completely wipe and reinstall your complete scientific Python stack in under 5 minutes” (if you have to)

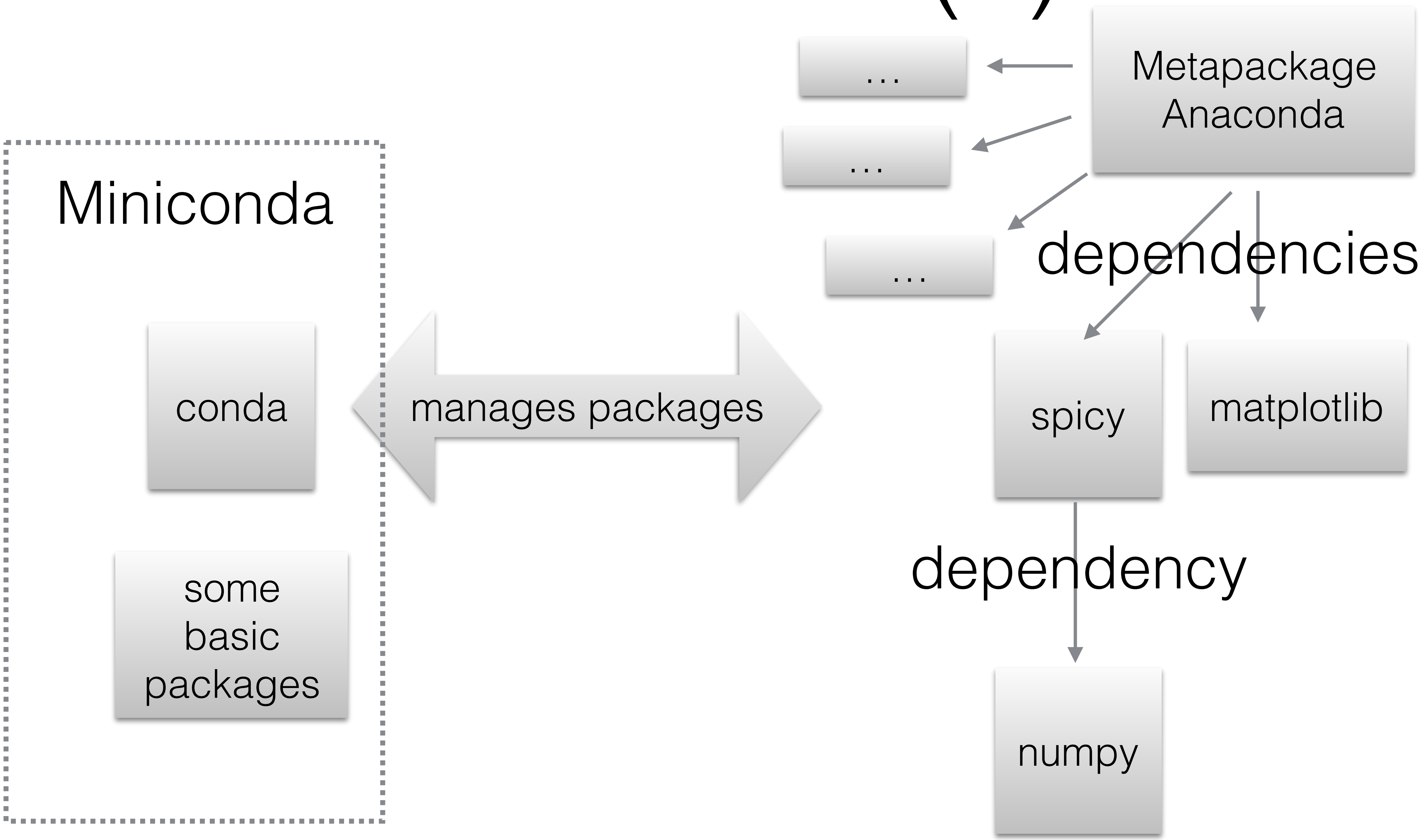
Material at

https://github.com/michaelaye/planetdata3

Conda vs Anaconda vs Miniconda

- How many here uses any conda-based system?
- How many really understand the difference between the 3 xxx-condas?

Conda vs Anaconda vs Miniconda (2)



Conda vs Anaconda vs Miniconda (3)

- So, in summary:
  - conda is the executable that manages packages (not only Python, e.g. HDF binaries etc.)
  - “miniconda” is a minimum set of packages for proper operation of conda, installed into a “root” environment
  - “anaconda” is a meta-package with a huge list of scientific packages (dependencies)
  - Hence: after installing miniconda and executing “conda install anaconda”, you have the same python env as somebody that DL-ed the Anaconda distribution.

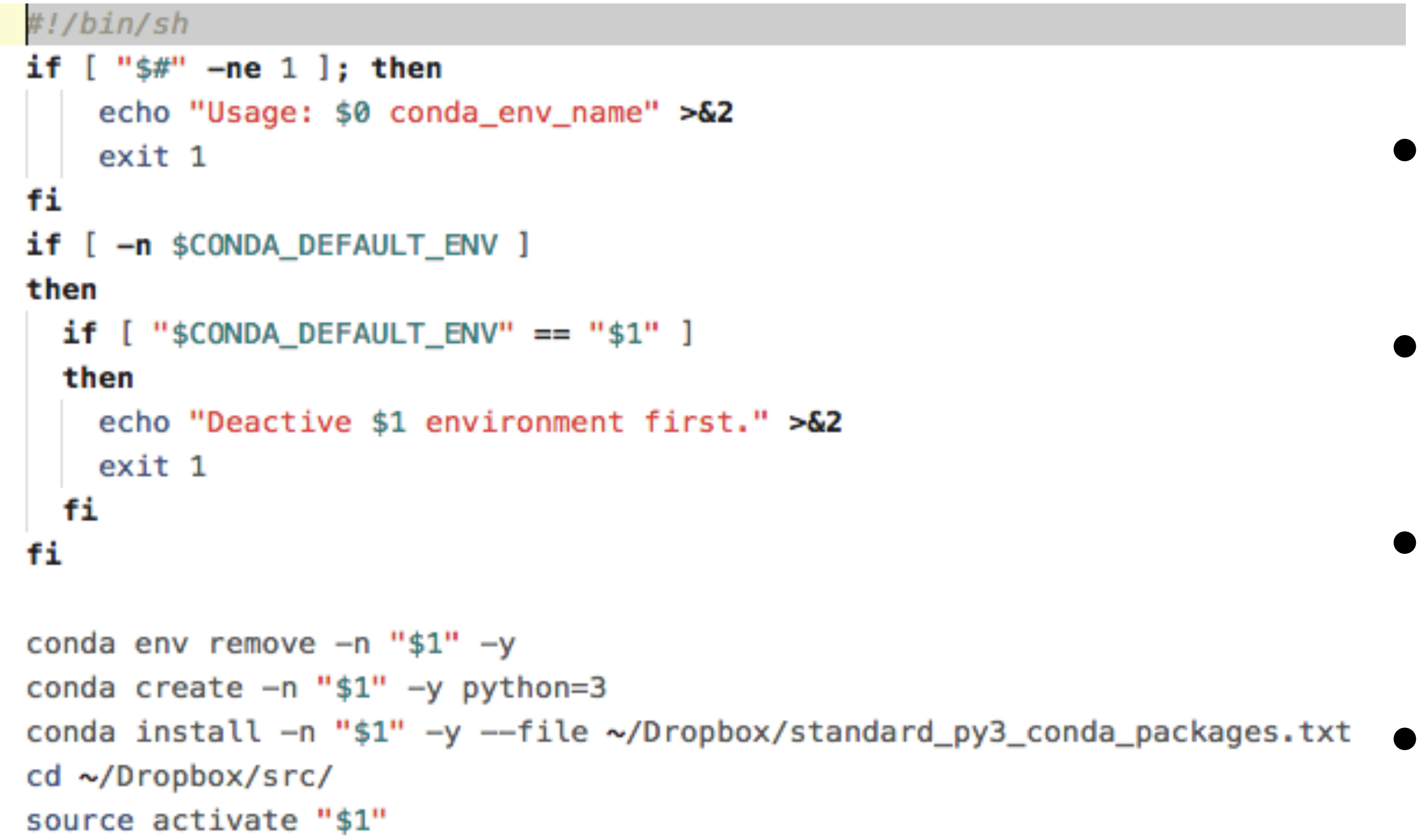
conda environments

- Advise: Don’t use the conda root environment.
    - Eventually some of your installs will mess up conda itself
- Always create a new default environment:
    - conda create -n ‘stable’ python=3 && source activate stable
- What if pkg not on conda?
    - pip install pkg_name
- What if pkg not even on Pypi server?
    - Github clone && python setup.py install (or “develop”)
- I use this mix for years with success

How to reinstall env in 5 min

- Even a “stable” env is rotting at some point
- The trick is to have:
    - File with a list of your conda packages
    - File with a list of your pip packages
    - File with a list of your own package folders and GH installs
    - a (couple of) bash script(s)

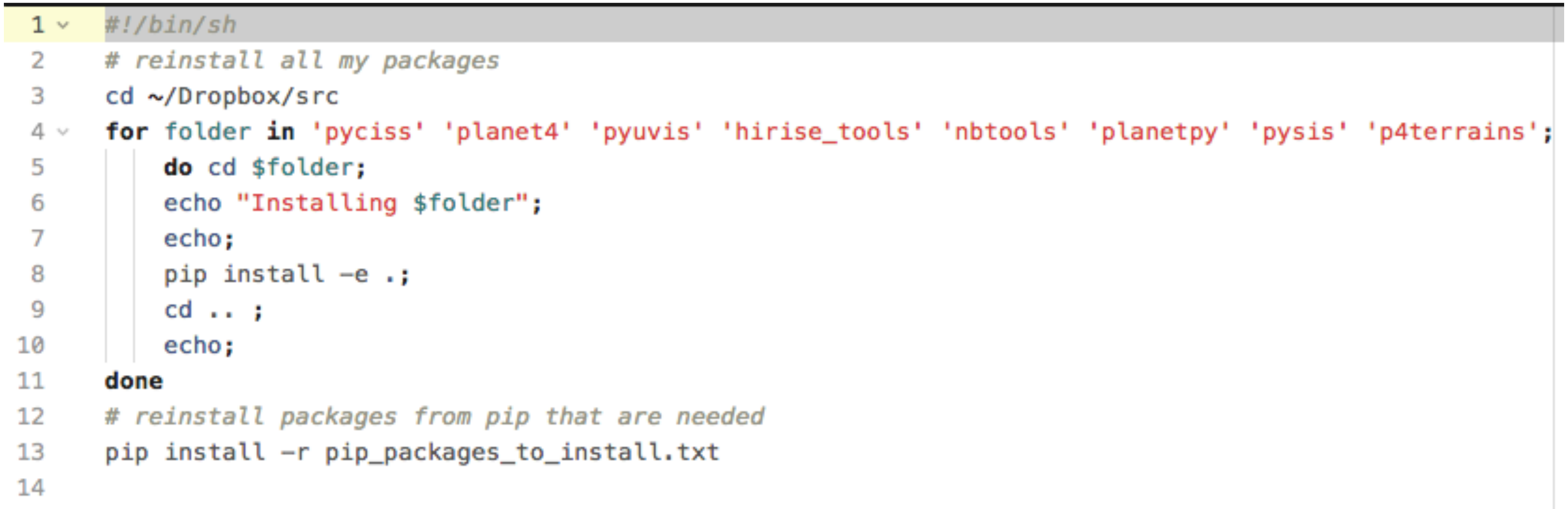
How to reinstall env in 5 min(2)



```sh
#!/bin/sh
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 conda_env_name" >&2
    exit 1
fi
if [ -n $CONDA_DEFAULT_ENV ]
then
  if [ "$CONDA_DEFAULT_ENV" == "$1" ]
  then
    echo "Deactive $1 environment first." >&2
    exit 1
  fi
fi

conda env remove -n "$1" -y
conda create -n "$1" -y python=3
conda install -n "$1" -y --file ~/Dropbox/standard_py3_conda_packages.txt
cd ~/Dropbox/src/
source activate "$1"
```

- source deactivate
- ./reinstall_env.sh stable
- source activate stable
- ./install_my_libs.sh



```sh
#!/bin/sh
# reinstall all my packages
cd ~/Dropbox/src
for folder in 'pyciss' 'planet4' 'pyuvis' 'hirise_tools' 'nbtools' 'planetpy' 'pysis' 'p4terrains';
    do cd $folder;
    echo "Installing $folder";
    echo;
    pip install -e .;
    cd .. ;
    echo;
done
# reinstall packages from pip that are needed
pip install -r pip_packages_to_install.txt

```

conda channels

- Basically, channels are different repositories of packages
- Sometimes you see “conda install -c conda-forge pkg_name (or any other channel name)
    - I would not do that in “stable”, better to create a new env for testing first.
    - Mixing packages from channels often works, but sometimes not
        - e.g. GDAL will break when mixing “defaults” and “conda-forge” channel !!