

Building Packages with Conda

CodeJam 8 - Lightning Talk

Alper Yegenoglu

`a.yegenoglu@fz-juelich.de`

Institute of Neuroscience and Medicine (INM-6) and Institute for Advanced Simulation (IAS-6),
JARA Brain Inst I,
Juelich Research Center

September 15, 2017

- **Deployment:** Building packages and releasing them, or making them publicly available using conda
- Example: (Electrophysiological Data Analysis Toolkit) Elephant

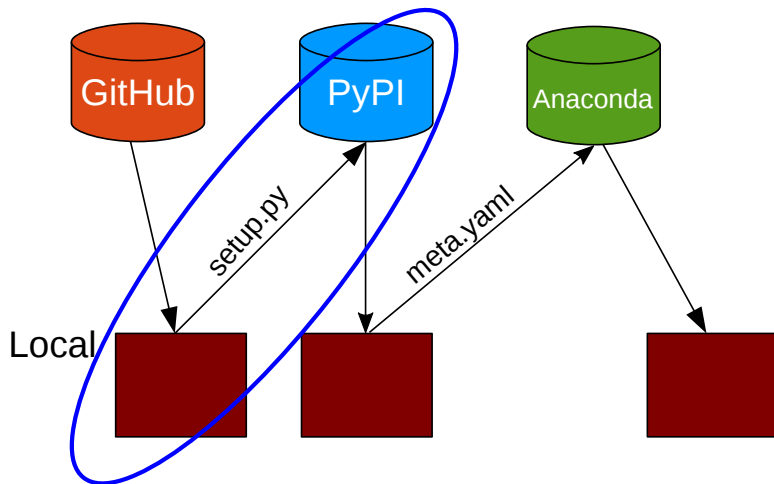


- Here Python only

Background information

- Python Package Index (**PyPI**): Software repository for Python
Default source for packages and their dependencies (pip)
- **setup.py**: Python script to maintain the package, lists all dependencies, docs (html)
- **Anaconda**: Distribution of Python and other languages, uses conda
- **conda**: Package manager and environment management system
- **Anaconda-cloud**: Repository for conda packages
- **meta.yaml**: Script to build packages for conda, contains metadata

Workflow



setup.py

```
from setuptools import setup
import os

long_description = open("README.rst").read()
install_requires = ['neo>0.3.3', 'numpy>=1.8.2', 'quantities>=0.10.1',
                   'scipy>=0.14.0', 'six>=1.10.0']
extras_require = {'pandas': ['pandas>=0.14.1'], 'docs': ['numpydoc>=0.5',
                                                         'sphinx>=1.2.2'], 'tests': ['nose>=1.3.3']}

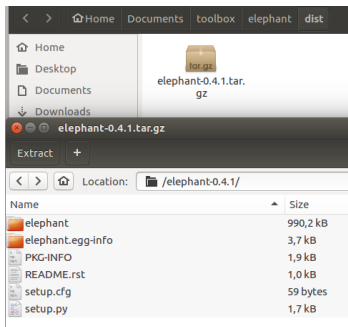
setup(
    name="elephant",
    version='0.4.1',
    packages=['elephant', 'elephant.test'],
    package_data = {'elephant': [os.path.join('current_source_density_src', i)
                                   for i in ['test_data.mat', 'LICENSE',
                                             'README.md', '*.py']]},

    install_requires=install_requires,
    extras_require=extras_require,

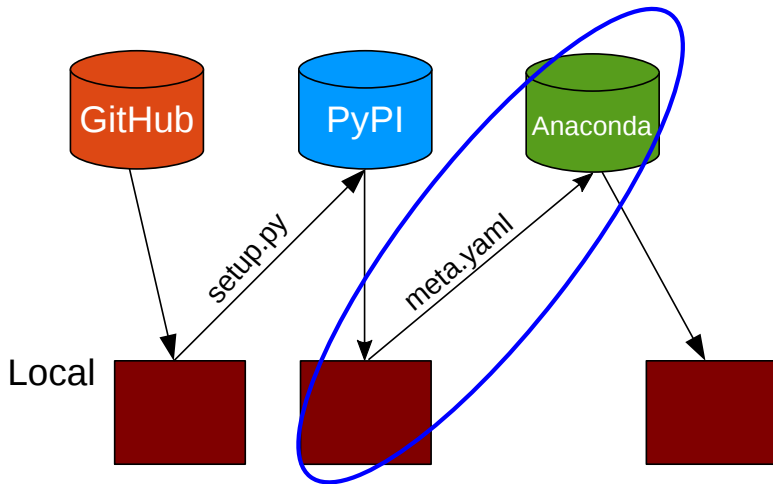
    author="Elephant authors and contributors",
    author_email="andrew.davison@unic.cnrs-gif.fr",
    description="Elephant is a package for analysis of electrophysiology data in Python",
    long_description=long_description,
    license="BSD",
    url='http://neuralensemble.org/elephant',
    classifiers=[
        'Development Status :: 4 - Beta',
        'Intended Audience :: Science/Research',
        'License :: OSI Approved :: BSD License',
        'Natural Language :: English',
        'Operating System :: OS Independent',
        'Programming Language :: Python :: 2',
```

sdist

- Build a source package:
- `python setup.py sdist`
- A folder `dist` is created



- Upload the package to PyPI:
- `python setup.py sdist upload --repository pypi`
- Elephant on PyPI



Package via conda

- Make sure to have conda installed
- Install conda-build: `conda install conda-build`
- Use conda skeleton to generate a conda script/recipe
- `conda skeleton pypi elephant`
- Creates a directory (*elephant*) and scripts: `meta.yaml`, `build.sh`, `bld.bat`
- Create the package: `conda build elephant`
- If specified in the `meta.yaml`, conda will try to import the package and run tests
- Packages are stored as tarballs e.g. under `/home/user/miniconda/conda-bld/linux-64/elephant`
- Alternative approach via GitHub

meta.yaml

```
package:
  name: elephant
  version: "0.4.1"

source:
  fn: elephant-0.4.1.tar.gz
  url: https://pypi.python.org/packages/ff/2b/cf614d7e039f9ffcc9fded53c47a268503e569
  md5: 0e6214c96cae6ce777e4b3cf29bbdaa9

requirements:
  build:
    - python
    - setuptools
    - neo >0.3.3

  run:
    - python
    - neo >0.3.3

test:
  # Python imports
  imports:
    - elephant
    - elephant.test
```

Upload to anaconda cloud

- Anaconda client command line interface to access your Anaconda Cloud account, upload packages, etc.
- `conda install anaconda-client`
- `anaconda upload elephant`
- You will be asked for user name and password to login and upload the package to your anaconda cloud
- Upload to <https://anaconda.org/USERNAME/PACKAGENAME>
- Via `anaconda convert --platform all /home/user/miniconda2/conda-bld/linux-64/packagename` it is possible to convert the package for other systems

References

- ▶ Github: <https://github.com/>
- ▶ PyPI: <https://pypi.python.org/pypi>
- ▶ Conda installation: <https://conda.io/miniconda.html>
- ▶ Anaconda cloud: <https://anaconda.org>
- ▶ <https://docs.python.org/3/distutils/packageindex.html>
- ▶ https://conda.io/docs/build_tutorials.html
- ▶ <https://docs.continuum.io/anaconda-cloud/using>

Thank you!

Additional Slides

conda steps

- 1 Reads the metadata
- 2 Downloads the source into a cache
- 3 Extracts the source into the source directory
- 4 Applies any patches
- 5 If source was necessary to fill any metadata values, the metadata is re-evaluated
- 6 Creates a build environment and installs the build dependencies there
- 7 Runs the actual build script. The current working directory is the source directory with environment variables set. The build script installs into the build environment
- 8 Does some necessary post-processing steps: shebang, rpath, etc
- 9 Packages up all the files in the build environment that are new from step 5 into a conda package along with the necessary conda package metadata
- 10 Tests the new conda package: deletes the build environment and creates a test environment with the package and its dependencies, and runs the test scripts. This step is not run if there are no tests in the recipe

```
[distutils]
```

```
index-servers =  
    pypi  
    other
```

```
[pypi]
```

```
repository: <repository-url>  
username: <username>  
password: <password>
```

```
[other]
```

```
repository: https://example.com/pypi  
username: <username>  
password: <password>
```

git meta.yaml

```
package:  
  name: elephant  
  version: "0.4.1.1"  
  
source:  
  git_url: https://github.com/INM-6/elephant.git  
  git_tag: master
```


conda and C++

- ▶ Example: `https://github.com/emka/conda-recipes/tree/master/nest-simulator`
- ▶ `https://github.com/nest/nest-simulator/issues/210`