

Conda Package Management

Not just for python any more

Once upon a time we installed software this way:

```
10 Download source files
20 ./config --PREFIX=/somewhere/bin
30 make && make install
40 Find missing dependencies
50 go to 10 until all dependencies met
```

The Goddesses saw our plight, and sent us* the demigod

APT

*“Us” meaning if you are a sysadmin or *sudoer*. Otherwise, you are back at the previous slide.

Now we install things this way*:

```
apt-get install <mypackage>
```

apt installs all dependencies and makes sure everything is in place.

Thank you Goddesses!

*If you are a sysadmin or *sudoer*. Otherwise, you are back at the previous slide -1.

We used to install python packages like this:

10 Download package

20 python setup.py install

30 find missing dependencies

40 go to 10 until all dependencies met

The Goddesses saw our plight, and sent us the demigod*

PIP

*but only for Python

Now we install things this way:

```
pip install <my python package>
```

Pip installs all dependencies and makes sure everything is in place.

Thank you Goddesses!

The Goddesses also sent us*:

```
install.packages( "my_package" )
```

Thank you again Goddesses!

*but only for R

Enter the world of bioinformatics...

In bioinformatics, and other Data Science fields, we use toolchains comprised of software written in many languages:

- Python
- R
- Bash
- Java
- C
- C++
- Fortran
- ...

The Goddess **Continuum** saw our plight, and sent to earth the demigod

ANACONDA

and her faithful sidekick

MINICONDA

<https://www.continuum.io/anaconda-overview>



Now we can install *anything* in our toolchain with

```
conda install <anything>
```

Conda installs all dependencies, downloads source from anyplace it needs to, and builds packages as needed.

Thank you Goddesses!

Like seriously, thank you, thank you, thank you.

Caveat:

- This only works for packages for which people have build “conda recipes.”

However...

You can use conda for 95% of what you need

- Conda plays super nice with pip
 - Pip is installed along with conda
 - If you want to install a package and there isn't a conda recipe, but there is a pip recipe, then you can install with pip (but use `.../conda/bin/pip`)
 - Conda will manage the pip installed packages and update when you run
`conda update ...`
- There are TONS of conda recipes in a multitude of “channels”
- In particular:
 - **Continuum Analytics** is a data science company, and the *anaconda* channel has every kind of data science tool you need such as R and Python packages for statistics and visualization.
 - The *bioconda* channel contains all of the major bioinformatics tools, including everything from bioconductor.

Shareable and Reproducible Science

- Conda allows you to create custom “environments” that include exactly (and only) the specific versions of software you want.
- This completely avoids P2 vs P3 conflicts
- Environments are **shareable** and **reproducible**
- Conda can export a list of all installed packages (including those installed with pip) that can completely reproduce an entire application environment. For example,
- .../BioCoders/Applications/conda gets FUBARed?
`conda env create -f conda_root.yml`
- Back in action!

Is Conda just the latest fad?

“Galaxy has adopted a new standard for tool dependencies: Conda packages!”

“Not only do Conda packages make tool dependencies more reliable and stable, they are also easier to test and faster to develop than the traditional Tool Shed package recipes.”

https://docs.galaxyproject.org/en/master/admin/conda_faq.html

Ariel did a nice job of summarizing the advantages of Conda: it is cross-platform, requires no sudo access, distributes binaries, and supports virtual environments. For all these reasons, IMO it is the best packager out there these days. To add to the discussion:

-I use the bioconda channel (-c bioconda) for installation of my bioinformatics software these days

-Using conda environments, it's straightforward to identify minimum requirements for a project. That said, the size of software packages is minimal compared to data needs, so I don't worry much about package bloat.

- With conda environments, it's simple to include particular software versions, so I don't worry about someone polluting the main store: I can always create an environment and installed pinned versions in that environment.

Best,

J.

(Jeremy Goecks, known scientist thief)

Final Caveat:

- If you are going to use conda, you need to let it be your python path, or problems can arise.
- <https://youtu.be/Y3lQSxNdr3c>