Why are we going to talk to you about environments/containers?

Phil Ashton, PGI Advanced Training - KEMRI Wellcome, April 2022

There are two* universally painful parts of being a bioinformatician:

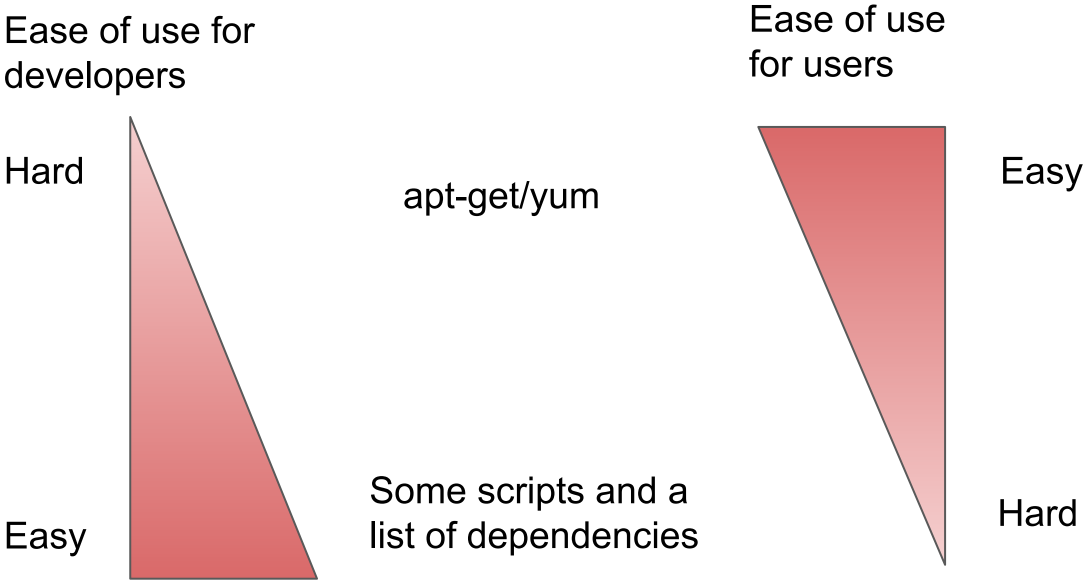
- Software installation
- Reproducibility

Environments/containers can help solve both these problems

*at least two

Installation

Traditionally (i.e. before 2016) there was a trade off between the simplicity of making software available and the simplicity of installing software



Ease of use for developers

Hard

Easy

apt-get/yum

Some scripts and a list of dependencies

Ease of use for users

Easy

Hard

Virtual environments and containers provide a “happy medium” - relatively straightforward for both users and developers

Reproducibility

“It works on my machine”

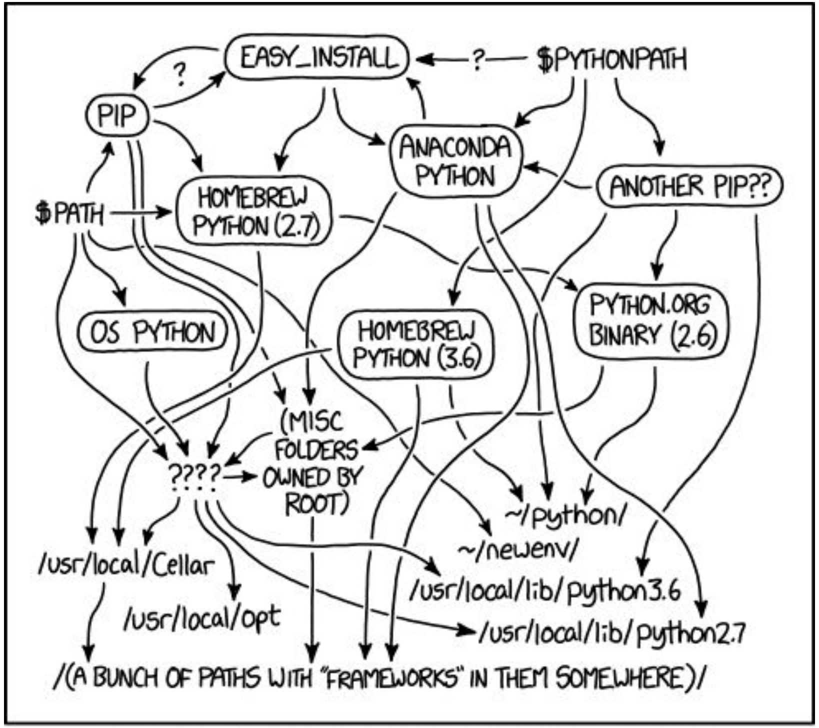
“We build our computer systems the way we build our cities: over time, without a plan,
on top of ruins” - Ellen Ullman

The solution is to make our machines as similar as possible, and to start from fresh each time

- environments and containers help us do this

Conda and Conda environments

(Python) package management is (was?) a mess



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

https://xkcd.com/1987/

Introduction to Conda

What is the problem?

- We want to avoid “dependency hell”:
  - I want to run “software A”
  - Software A depends on Software B, C, and D.
  - Software B depends on Software E and F
  - Software C depends on G and H
  - Software D depends on I and J
  - …

What is the solution?

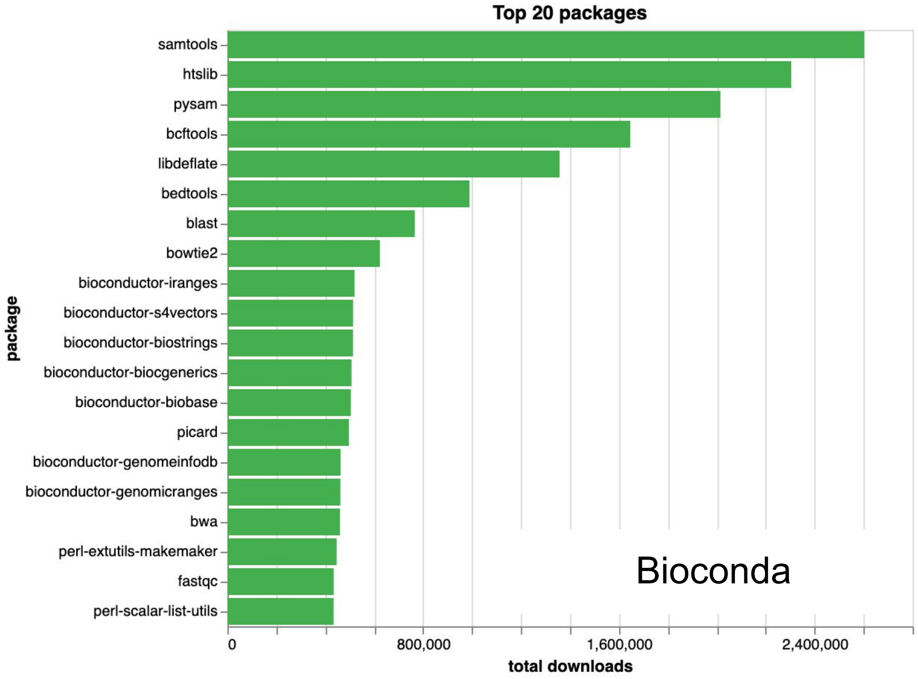
- “Package management”

Enter Anaconda

- Anaconda is a distribution of Python & R aimed at scientists that aims to simplify package management and deployment.
- When you install Anaconda you get:
    - Python + common scientific packages
    - Conda (language agnostic)
        - Package manager
        - Environment manager
- Miniconda is:
    - Python
    - Conda

conda

- Thousands of packages available to install via conda
- Different “channels” that are the location where packages are hosted
- Channels include
  - Bioconda - for bioinformatics software
  - Conda-forge - miscellaneous community provided packages
- Channels are searched in order



Mamba

- Conda became a victim of its own success - as the number of packages increases, the time taken to “solve the dependencies” has increased.
  - What is the current state of the software environment on the computer?
  - What is the desired state of the software environment (for the new software)?
  - Make an upgrade plan that allows installation of new software, without breaking existing software.
- Two things slow down conda:
  - Increasing numbers of packages installed on your computer
  - Increasing number of packages available for installation.

Mamba

- Mamba is a “drop in” replacement for conda that is much quicker:
    - `mamba install`
    - `mamba create`
- Uses the libsolv dependency solver and optimised code in C++

Conda environments

- Conda is also an environment manager
- Environments (also known as virtual environments) are a way to have separate and independent installations of software on the same computer.
- They work by modifying the $PATH variable (very simple explanation).
- Advantages of environments:
  - Software installs quickly (if there isn't much inside your environment)
  - It's possible to have software that requires e.g. different versions of the same dependencies.
- Default environment is called `base`

What do you want your base environment to look like?

vs

Limitations of Conda

- Can be differences in dependency resolution solutions between platforms
- Conda environments are not inherently cross-platform

I.e. doesn’t *necessarily* solve the “works on my machine” problem.

Outline of practical

1. Install miniconda
2. Use conda to install mamba
3. Use mamba to install a package into the base environment
4. Use mamba to create a new environment and install software into it
5. Install software into an existing environment
6. Install software from a yaml file
7. Export the specification for an existing environment to a yaml file, and create a new environment from that file

Acknowledgements & further reading

Anna Price, Cardiff/CLIMB - https://www.youtube.com/watch?v=qORviM_ELdk

Open Software Packaging for Science | by QuantStack | Medium
Python Virtual Environments: A Primer