Conda and Bioconda
Conda

Package, dependency and environment management for any language

“Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.”

from: https://conda.io/docs/
My Python environment has become so degraded that my laptop has been declared a Superfund site.
which python
Why you need Python environments and how to manage them with Conda
Conda can be used to install:

• Python (with disclaimer)
• Python packages
• R
• Tools and packages for bioinformatics and data science

✓ Specific versions
1. The installer first sets up Conda.

2. Then Conda creates the root environment.

3. Python is being installed as a package.

4. Later you can add as many additional environments as you want (and you can name it whatever you like).

5. Different environments can contain different Python versions and different sets of packages.
Channels are like storages...

- **Channel 1**: First, it is looking for the package in this channel (this has the highest priority).
- **Channel 2**: It was not in the first channel, so it moves to the storage with the 2nd highest priority.
- **Channel 3**: YAY! Conda has found the package! Now it is added to your environment.
- **Channel 4**: (Not filled yet, a placeholder)

**What happens if Conda did not find the package?**

By default, Conda looks for packages in the official storages of Continuum. Why? Because by default these have the highest priorities.

However, you have the power to:

- **Add new channels (storages)** that contain the packages you need!

So let's say you want to install this package: 🌟

- → Conda does not find it in the first 3 channels
- → You need to add a 4th one!

Illustration by Krisztina Szerovay
Questions?
1. This is one decision-making point

2. These depend on your system

3.

4. This is the other decision-making point

---

**INSTALLING CONDA**

**CHOOSING THE INSTALLER:**

In our case it is **MINICONDA**

---

**WINDOWS**

- 32 BIT
- 64 BIT

**LINUX**

- 32 BIT
- 64 BIT

**MACOSX**

- 64 BIT

---

The combination of your operating system & the bit count is referred to as "platform."

---

For example:

In this case, you use a Miniconda installer that sets up a Conda *for a 64-bit Windows*

*Containing a 3.x version Python as part of the root environment.

---

Illustration by Krisztina Szerovay

---

[https://conda.io/docs/user-guide/install/index.html](https://conda.io/docs/user-guide/install/index.html)
General conda commands

• conda --help
• conda list #all packages available in a given environment
• conda env list #lists all environments
Create and activate new environment

- conda create --name test
- source activate test

- conda install -c bioconda plink
- conda install -c bioconda minimap
- conda install -c bioconda bioawk
- conda install --channel r r-essentials=1.4
- conda install --channel r r=3.3.1

- source deactivate
Looking for packages to install

Newest version:
   Google `conda name_of_the_package`

Which versions of the package are available?
   `conda search -f samtools`
Power of bioconda

• conda search --channel bioconda
• conda search --channel bioconda | wc -l
Installing perl modules via conda

- conda create --name perly_env
- source activate perly_env
- conda install -c anaconda perl # Installing perl
- conda install -c bioconda perl-app-cpanminus # Installing CPAN for managing perl modules
- If -c bioconda does not work, try:
  - conda install -c conda-forg perl-app-cpanminus
- cpanm Bio::Perl # Getting any module you like

Commands courtesy of Samarth Rangavittal
https://hcc-docs.unl.edu/display/HCCDOC/Installing+Perl+modules
Questions?

twitter.com/biomonika