

BIOL 343

Applied Bioinformatics I

Python environments

Dr. Nic Wheeler

Managing Python projects with environments

Problems when using Python:

1. System pollution

OS uses Python with specific library versions

2. Discrepant dependencies

Not all projects will be able to use the same version of each library

3. Conflicting packages

Not all packages work with each other

4. Irreproducibility

Analyses performed on your computer should be able to be easily reproduced on a different computer



Virtual environments

Managing Python projects with environments

```
$ which python3  
/opt/homebrew/bin/python3
```

```
$ conda activate base  
$ which python3  
/Users/njwheeler/micromamba/bin/python3  
$ python3 --version  
Python 3.12.2  
$ python3 -c 'import numpy; print(numpy.version.version)'  
Traceback (most recent call last):  
  File "<string>", line 1, in <module>  
ModuleNotFoundError: No module named 'numpy'
```

```
$ conda activate tracking  
$ which python3  
/Users/njwheeler/micromamba/envs/tracking/bin/python3  
$ python3 --version  
Python 3.11.5  
$ python3 -c 'import numpy; print(numpy.version.version)'  
1.26.4
```

Managing Python projects with environments

Many package/environment managers:

1. Native virtual environments (venv)
2. Virtualenv
3. Anaconda/Miniconda/Mamba/Micromamba

Advantages to the conda family:

1. Not just for Python (can manage R and R libraries as well as other command line tools)
2. Very popular within the bioinformatics community (i.e., the bioconda channel)
3. Strong support base
4. Mamba is installed on BOSE

Conda envs in BIOL 343

- Each student will use their own conda env
- Start with the **biol343** foundation (find in the GitHub repo)
- Can install your own packages

Live code activity...

1. Create new biol343 env
2. Activate env
3. Add bioconda and conda-forge channels
4. Install a new package (clustalw)
5. Test the new package
6. Demonstrate that the file system perseveres