

Hoffman2 Happy Hour: Anaconda for HPC

Charles Peterson

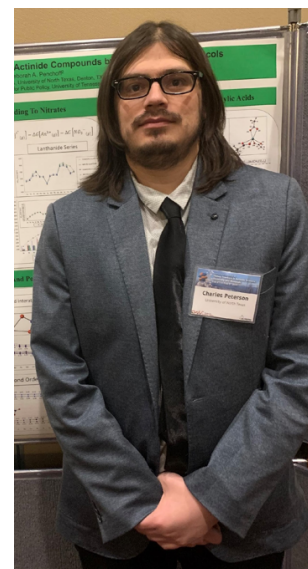
Overview

Welcome to Hoffman2 Happy Hour!

The H²HH are designed to be short interactive talks that focus on a certain aspect of HPC.



- In this H²HH we will go over using Anaconda on Hoffman2
- This information can be applied to other HPC resources



Any suggestions for upcoming workshops, email me at cpeterson@oarc.ucla.edu

Files for this Presentation

This presentation can be found on our UCLA OARC's github repo.

- https://github.com/ucla-oarc-hpc/H2HH_anaconda

View slides:

- PDF format: [H2HH_anaconda.pdf](#)
- html format:
 - The html slides can be viewed at:
 - https://ucla-oarc-hpc.github.io/H2HH_anaconda

Note

This presentation was build with [Quarto](#) and RStudio.

- Quarto file: [H2HH_anaconda.qmd](#)

What is Anaconda

- Anaconda is a very popular Python and R distribution.
- Great option for simplifying package management and pipelines.
- Easily install popular Python and R packages.



Why use Anaconda

- Easy install many python and R packages with simple `conda` commands
- Create isolated python/R environments for different projects
 - Different python/R setups and switch between them
- Checks and solve for possible version conflicts when installing packages
- Share conda env on different systems.
 - Version control!

Starting Anaconda

On Hoffman2, Anaconda is installed and can be used by loading modules

- See available anaconda versions

```
1 module av anaconda
```

- Load anaconda in your environment

```
1 module load anaconda3/2020.11
```

- Loading the anaconda module will setup anaconda in your environment and ready to be used!

Important

By using anaconda, you do **NOT** need to load any other python/R modules. The python/R builds will be available via anaconda.

Using other python build might cause conflicts with your anaconda python (or R)

Anaconda environment

- Anaconda environments (conda env) is a virtual environment
 - install and update packages that you can control
- These conda env's will reside in your personal workspace
 - By default `$HOME/.conda`
- Conda is also a package manager
 - Many packages/software can be install via Conda's repository
 - <https://anaconda.org/anaconda/repo>
- Can also install python packages within your conda env outside of Conda's repo
 - PyPI's `pip` for python
 - R's CRAN software repository

Creating conda env

Creating a new conda environment

```
1 conda create [options]
2 conda create -n myconda
```

- The `-n` option will name your new conda env

You can install packages and software while creating the env

```
1 conda create -n myconda python
```

Install multiple conda packages with `conda create` command

```
1 conda create -n myconda python=3.9 pandas scipy r-base
```

This will install

- python version 3.9
- scipy
- R

Conda envs

See list of all your environments that you have created

```
1 conda env list
```

Start (activate) your conda environment

```
1 conda activate myconda
```

Activating the conda env gives access to the software within the env

This version of **python** and **R** is installed locally in your conda env and is different from the builds of python on Hoffman2.

You can see location of python in your env and check the version.

```
1 which python
2 python -V
```



Important

You do **NOT** need to load the python module if you installed python via anaconda.

Tips for running on HPC

You maybe familiar with using Anaconda on your local machine.

- Running on HPC may be different.

Warning

Do not use conda's default `base` env

When conda is install, it creates an conda env named `base` that you may see when running `conda env list`

- Located in the central anaconda installation path can **CANNOT** be modified by users

Tips for running on HPC



Warning

Do not run `conda init` on H2.

You may see messages or online tips about running the `conda init` command.

This initializes conda but is **NOT NEEDED** to run on Hoffman2

While this does setup conda, it will change `~/.bashrc` and may cause conflicts using different versions/envs.

Loading the anaconda module will already setup conda.

Installing packages

Once your conda env is activate, you can install more packages with

- `conda install`

```
1 conda create -n myconda
2 conda activate myconda
3 conda install python=3.9 pandas scipy tensorflow -c conda-forge
```

Note

The `-c` option in conda is for the “conda channel”. The conda channels are different locations where packages are stored. Examples are ‘conda-forge’, ‘bioconda’, ‘defaults’, etc. Conda will search though the available channels for the request packages to install.

Installing packages

You can use `pip` when you are in a conda env

```
1 conda activate myconda
2 pip3 install scipy
```

Tip

When using pip/pip3 in a conda env, you do **NOT** need to have `--user`. Using just `pip` will install the package inside the conda env. If you use `--user`, it will install the package outside of the conda env, inside of `~/.local` and may cause conflicts with other python builds or conda env's you have.

Tips

By default, when you install a conda env, it will install it at
`~/ .conda`

You can change this location, esp if you are low in space at
`$HOME`

```
1 conda create -p $SCRATCH/mypython python=3.9
2 conda activate $SCRATCH/mypython
```

- Some detailed information on using Anaconda on Hoffman2 can be found on our website
- <https://www.hoffman2.idre.ucla.edu/Using-H2/Software/Software.html#anaconda>

Job examples

Once you create you env, you don't need to create it again.

```
1  qrush -l h_data=5G
2  module load anaconda3/2020.11
3  conda activate myconda
4  python test.py
```

Job script example

```
1  #!/bin/bash
2  #$ -cwd
3  #$ -j y
4  #$ -l h_rt=1:00:00,h_data=5G
5  #$ -pe shared 1
6
7  # load the anaconda module
8  . /u/local/Modules/default/init/modules.sh
9  module load anaconda3/2020.11
10 # Activate the 'myconda' conda env
11 conda activate myconda
12
```

Searching for anaconda packages

Find software that is available on Anaconda's package repo

- <https://anaconda.org/anaconda/repo>

Here, you can search for software and other packages. It will also explain what conda commands you need to install them inside your conda env.

Using yml files

You can create a conda file from a `environment.yml` file

```
1 name: myconda
2 dependencies:
3   - numpy
4   - pandas
5   - python=3.9
```

You can create a conda env with all these packages by running:

```
1 conda env create -f environment.yml
```

Using yml files

An `.yaml` file can be created from an existing conda env so you can create the same conda env.

```
1 conda activate myconda
2 conda env export > environment.yml
```

This file can be shared with others to reproduce any conda env.

- Keep the same versions of packages when running anaconda on different HPC resources

Installing Anaconda

While Hoffman2 already has Anaconda install, you may need to install Anaconda on a separate machine or another HPC resource.

Visit <https://repo.anaconda.com/archive/> for all the versions of Anaconda that are available.

In this example, anaconda is install at [\\$HOME/apps/anaconda/2021.11](#)

```
1 #Download anaconda script for Linux
2 wget https://repo.anaconda.com/archive/Anaconda3-2021.11-Linux-x86_64.sh
3 #Run Anaconda installer
4 bash Anaconda3-2021.11-Linux-x86_64.sh -p $HOME/apps/anaconda/2021.11 -b
```

Now create and activate a conda env with this new anaconda build

```
1 # Setup Anaconda
2 source $HOME/apps/anaconda/2021.11/etc/profile.d/conda.sh
3 # Create new conda env
4 conda create -n myconda python=3.9
5 conda activate myconda
```

Installing Anaconda



Tip

Don't run `conda init`

Instead, `source /CONDA/PATH/etc/profile.d/conda.sh`

This will setup Anaconda without changing the `~/.bashrc` file



Tip

`Miniconda` is a good alternative to Anaconda.

It is a Minimal installer for conda that is smaller than Anaconda.

Thank you!

Questions? Comments?

Charles Peterson

- cpeterson@oarc.ucla.edu
- Look at for more Hoffman2 workshops at <https://idre.ucla.edu/calendar>
 - Search for [HPC](#)

