

Python Package Install: miniconda

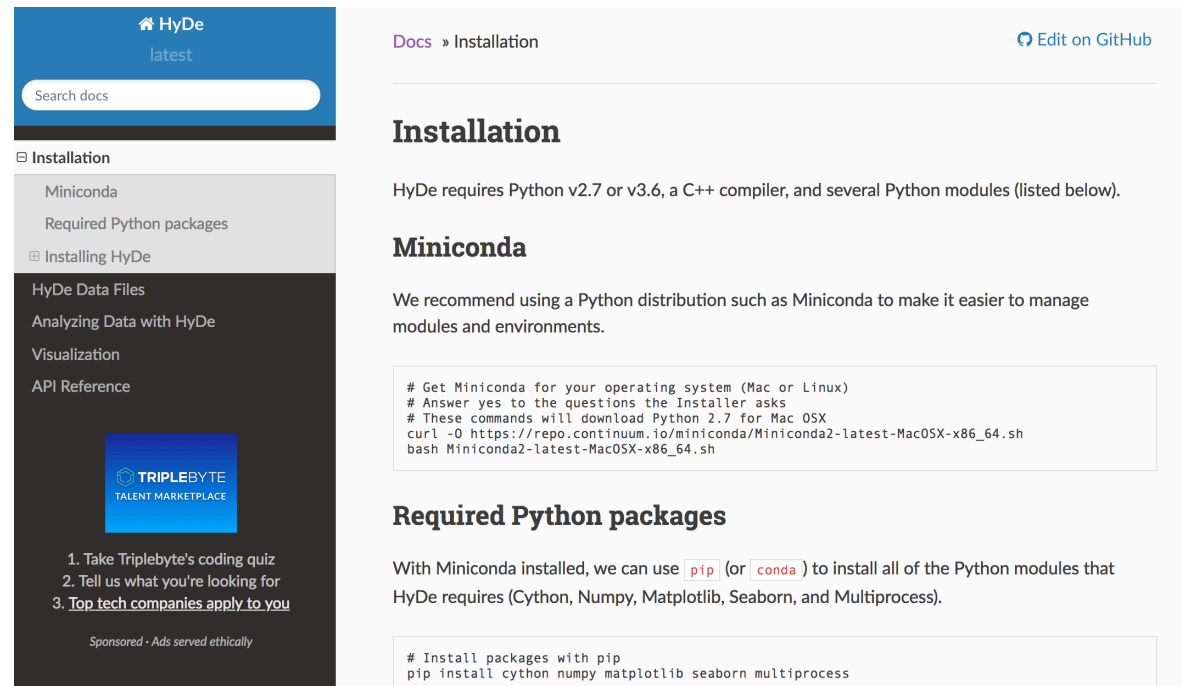
Neranan V. Perera

Computational Biology Core



Python Package Install

- Example:
 - HyDe install
 - <https://hybridization-detection.readthedocs.io/installation.html>



The screenshot displays the HyDe documentation website. The left sidebar contains a navigation menu with the following items: Installation (expanded), Miniconda, Required Python packages, Installing HyDe, HyDe Data Files, Analyzing Data with HyDe, Visualization, and API Reference. Below the menu is a Triplebyte logo and a list of three items: 1. Take Triplebyte's coding quiz, 2. Tell us what you're looking for, and 3. Top tech companies apply to you. The main content area is titled 'Installation' and includes a search bar, a 'Docs » Installation' breadcrumb, and an 'Edit on GitHub' link. The text states that HyDe requires Python v2.7 or v3.6, a C++ compiler, and several Python modules. A section titled 'Miniconda' recommends using a Python distribution for easier management. A code block provides terminal commands to install Miniconda for Mac OS X. Another section titled 'Required Python packages' explains that pip or conda can be used to install the necessary modules, with a code block showing the pip install command for cython, numpy, matplotlib, seaborn, and multiprocessing.

HyDe
latest

Search docs

Installation

Miniconda

Required Python packages

Installing HyDe

HyDe Data Files

Analyzing Data with HyDe

Visualization

API Reference

TRIPLEBYTE
TALENT MARKETPLACE

1. Take Triplebyte's coding quiz
2. Tell us what you're looking for
3. [Top tech companies apply to you](#)

Sponsored - Ads served ethically

Docs » Installation [Edit on GitHub](#)

Installation

HyDe requires Python v2.7 or v3.6, a C++ compiler, and several Python modules (listed below).

Miniconda

We recommend using a Python distribution such as Miniconda to make it easier to manage modules and environments.

```
# Get Miniconda for your operating system (Mac or Linux)
# Answer yes to the questions the Installer asks
# These commands will download Python 2.7 for Mac OSX
curl -O https://repo.continuum.io/miniconda/Miniconda2-latest-MacOSX-x86_64.sh
bash Miniconda2-latest-MacOSX-x86_64.sh
```

Required Python packages

With Miniconda installed, we can use `pip` (or `conda`) to install all of the Python modules that HyDe requires (Cython, Numpy, Matplotlib, Seaborn, and Multiprocess).

```
# Install packages with pip
pip install cython numpy matplotlib seaborn multiprocessing
```



Python Package Install

Start an interactive session using srun




```
$ srun --mem=1G --partition=general --qos=general --pty bash
```



Python Package Install

- Installing Miniconda <https://conda.io/docs/user-guide/install/linux.html>
 - <https://conda.io/miniconda.html>

Miniconda

	 Windows	 Mac OS X	 Linux
Python 3.7	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)
Python 2.7	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)

- `wget https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.sh`



Python Package Install

Run the downloaded bash script using:

```
sh Miniconda2-latest-Linux-x86_64.sh
```

This will install miniconda in your home directory

```
/home/FCAM/$USER/miniconda2
```

Once installed it will add the miniconda path to your **.bashrc** file

```
# added by Miniconda2 installer
```

```
export PATH="/home/FCAM/$USER/miniconda2/bin:$PATH"
```

For this change to become active, you have to open a new terminal.



Python Package Install

- Open a new terminal / interactive session
 - conda install will be active
- To test the conda is active type
 - > `conda list`

```
# Name Version Build Channel
asn1crypto 0.24.0 py27_0
ca-certificates 2018.03.07 py27_0
certifi 2018.8.24 py27_1
cffi 1.11.5 py27he75722e_1
chardet 3.0.4 py27_1
conda 4.5.11 py27_0
conda-env 2.6.0 1
cryptography 2.3.1 py27hc365091_0
enum34 1.1.6 py27_1
futures 3.2.0 py27_0
idna 2.7 py27_0
ipaddress 1.0.22 py27_0
libedit 3.1.20170329 h6b74fdf_2
libffi 3.2.1 hd88cf55_4
libgcc-ng 8.2.0 hdf63c60_1
libstdcxx-ng 8.2.0 hdf63c60_1
```

- Update the `conda`
 - > `conda update conda`



Python Package Install

- Install required packages for the program using conda
 - > `conda install numpy`

```
Solving environment: done
## Package Plan ##

environment location: /home/FCAM/nperera/miniconda2

added / updated specs:
- numpy

The following packages will be downloaded:
```

package	build	
mkl-2019.0	118	204.4 MB
numpy-1.15.3	py27h1d66e8a_0	36 KB
mkl_random-1.0.1	py27h4414c95_1	361 KB
numpy-base-1.15.3	py27h81de0dd_0	4.1 MB
blas-1.0	mkl	6 KB
mkl_fft-1.0.6	py27h7dd41cf_0	148 KB
intel-openmp-2019.0	118	721 KB
libgfortran-ng-7.3.0	hdf63c60_0	1.3 MB
Total:		211.1 MB

```
The following NEW packages will be INSTALLED:
```

blas:	1.0-mkl
intel-openmp:	2019.0-118
libgfortran-ng:	7.3.0-hdf63c60_0
mkl:	2019.0-118
mkl_fft:	1.0.6-py27h7dd41cf_0
mkl_random:	1.0.1-py27h4414c95_1
numpy:	1.15.3-py27h1d66e8a_0
numpy-base:	1.15.3-py27h81de0dd_0

```
[Proceed ([y]/n)? y
```



Python Package Install

Solving environment: failed

PackagesNotFoundError: The following packages are not available from current channels:

- multiprocessing



Current channels:

- <https://repo.anaconda.com/pkgs/main/linux-64>
- <https://repo.anaconda.com/pkgs/main/noarch>
- <https://repo.anaconda.com/pkgs/free/linux-64>
- <https://repo.anaconda.com/pkgs/free/noarch>
- <https://repo.anaconda.com/pkgs/r/linux-64>
- <https://repo.anaconda.com/pkgs/r/noarch>
- <https://repo.anaconda.com/pkgs/pro/linux-64>
- <https://repo.anaconda.com/pkgs/pro/noarch>

To search for alternate channels that may provide the conda package you are looking for, navigate to

<https://anaconda.org>

and use the search bar at the top of the page.

⚡ Favorites	▼ Downloads	⚡ Package (owner / package)	Platforms
0	44012	 cvxgrp / multiprocess 0.70.4	linux-64 osx-64 win-64 conda
0	42762	 conda-forge / multiprocess 0.70.6.1 better multiprocessing and multithreading in python	linux-64 osx-64 win-32 win-64 conda
0	2378	 omnia / multiprocess 0.70.4 better multiprocessing and multithreading in python	linux-64 osx-64 win-32 win-64 conda

> conda install -c conda-forge multiprocessing



Python Package Install

- Installing HyDe
 - > `git clone https://github.com/pblischak/HyDe.git`
 - > `cd HyDe`
 - > `python setup.py install`
 - Installed `/home/FCAM/$USER/miniconda2/lib/python2.7/site-packages/phyde-0.4.2-py2.7-linux-x86_64.egg`
- Some times there might be version dependences
 - `conda search -f cython`
- To install a specific version
 - `conda install cython==0.25`



Xanadu Cluster Information

Understanding the Xanadu HPC Resource

1. What is a cluster
2. How to obtain an account
3. How to reset a password
4. How to access the Xanadu cluster
5. HPC resources and limits
6. Working with Slurm (Running interactive jobs and submission scripts)
 1. Resource Limitations
 2. Different ways of Running Jobs
 3. How to submit, Monitor and Cancel a Job
7. How to load software
8. File system
9. How to Transfer Data Between Clusters

What is a cluster

A desktop or a laptop, in most cases, is inadequate for analysis of large scale datasets (e.g. Genomics) or to run simulations (e.g. protein docking). They lack both the processing power and the memory to execute these analyses. This limitation can be overcome by combining machines (computers) in a predefined architecture/configuration so that they act as a single unit with enhanced computational power and shared resources. This is the basic concept of a high performance cluster. A cluster consists of a set of connected computers that work together so that they can be viewed as a single system. Each computer unit in a cluster is referred to as 'node'.

The components of a cluster are usually connected through fast local area networks ("LAN"), with each node running its own instance of an operating system. The benefits of clusters include low cost, elasticity and the ability to run jobs anytime, anywhere.

Install Python and R packages in local (home) directories

This short tutorial will provide instruction for installing and loading Python and R packages in your home directory without administrative access. We will use `alfpy`, `remeta` and `ggplot2` in this tutorial.

Install python package `alfpy` via `pip`

`Pip` is a package management system used to install/manage software packages from [Python Package Index](#). `Pip` is simple and user friendly. Usually, as users, we don't need to worry about prerequisites when we install a package via `pip`. Since `pip` has already been globally installed in BBC/Xanadu, and `alfpy` can be found in Python Package Index, we can use the following command from the home directory (the first location you arrive at after login to BBC or Xanadu).

```
pip install --user alfpy
```

note that flag `--user` is required because `pip` by default will install packages in root packages directory where most of users do not have permission to make changes.

Sometimes the latest version of a package is not compatible with the latest Python version in BBC/Xanadu or some features you need only exist in a certain version. Try to install an older version if you encounter this issue. To install `alfpy 1.0.4`, we first need to make sure current version of the package is uninstalled (note that flag `--user` is not required to uninstall).

```
pip uninstall alfpy
```

Then begin installation by entering:

<https://bioinformatics.uconn.edu/resources-and-events/tutorials/>

