Jupyter Notebook Set up and Anaconda Installation

MSU Data Science and Computational Seismology Lab

Min Chen (chenmi22@msu.edu)
Tong Zhou (zhouton3@msu.edu)
Last updated on May 26, 2019

The goal of this tutorial is to set up Python program for your research projects. Jupyter Notebook, an interactive environment for python coding, will be used for this purpose. There are three ways you can run Jupyter Notebook with Python: 1) installing python on your personal computer, 2) run Jupyter Notebook remotely on the iCER High Performance Computing Center (HPCC) machine, and 3) connecting to the engineering JupyterHub server.

Option 1: Install python on your personal computer

1. Check your system architecture and python environment. Usually recent computers use x86-64 architecture. Type python in your command line to check your python version.
2. Go to the Anaconda Download webpage:
   https://www.anaconda.com/distribution/#download-section
3. Choose your operating system (Windows, OS X, or Linux. If using a ubuntu-subsystem in windows10, choose Linux). Click on "Python 3.7 version" and "64-Bit Graphical Installer (662 MB)". Install Anaconda on your computer by double clicking on the package you downloaded and follow the instructions.
4. Open "Anaconda-Navigator", click on "Environments" tab and search the relevant packages that are not installed, which includes obspy, obspyck, obspandas, and geopandas. Select them and then click on the "Apply" button. You will also need to install the dependencies of geopandas in the way as installing the packages mentioned above. Required dependencies: numpy, pandas (version 0.19.1 or later), shapely (interface to GEOS), fiona (interface to GDAL), pyproj (interface to PROJ), six. Further, optional dependencies are: rtree (optional; spatial index to improve performance and required for overlay operations; interface to libspatialindex), psycopg2 (optional; for PostGIS connection), geopy (optional; for geocoding). For plotting, these additional packages may be used: matplotlib, descartes, mapclassify.
5. Open the command line program on your computer
   - On Windows, type CMD in the run box in the start menu.
   - On Mac, type "terminal" and hit enter in the Finder window, or double click to open the terminal application in the Utility folder.
   - On Linux, open up a console application.
6. Type "ipython notebook" or "jupyter notebook" in the command line and hit enter.

If everything goes correctly, a browser window should open up with the Jupyter interface running. If things don't work, don't worry, we will help you get started.

Option 2: Run Jupyter notebook on the iCER HPCC machine

MSU iCER HPCC machine has Linux x86-64 architecture and python 2.7 environment by default. To activate the anaconda on the iCER HPCC, use this command:

```
module load Anaconda2/4.2.0
```

Or you can download the latest anaconda through the official website https://www.anaconda.com/distribution/#download-section. Choose the right version (Linux, x86-64, python2.7) Log into your account, use scp to copy the local installer to your home folder, and then install the anaconda as instructed in Option 1.

```
scp local_file usrname@hpcc.msu.edu:hpcc_remote_folder
```

After you successfully installed (or activated) anaconda, you may then install all the required packages. You may create a virtual environment first, and then install Jupyter Notebook, numpy, obspy, ...

```
conda create -n environment_name python=3.7
conda install jupyter notebook
conda install numpy
conda install scipy
conda install matplotlib
conda install obspy
...
```

Next step is to set up the port forwarding.

Open a new terminal window, use this command to log in:

```
ssh -L localhost:portnumber1:localhost:portnumber2 usrname@hpcc.msu.edu
```

It established a link that forwards the output of the remote computer from portnumber2 to the local computer on portnumber1. Note: use different portnumber2 between 30000 to 60000 will reduce the chance that you got interference with others. Then use this command to start a Jupyter Notebook

```
jupyter notebook --no-browser --port=portnumber2
```

Then copy and paste the output URL to your local browser, and change the portnumber2 to portnumber1, you should be all set.

Option 3: Connect to the engineering JupyterHub server

You can request an engineering computing account. If this is your first time using your Engineering account you will need to activate the account by going to the following website:

https://www.egr.msu.edu/decs/myaccount/?page=activate

Enter your MSU NetID. The initial password will be your APID with an @ at the end (example: A12345678@) and then you have to set a new password that meets the requirements listed on the page. Verify the password. Then agree to the terms and Activate.
Once your account is activated you can access the classroom Jupyterhub server using the following instructions:

1. Open up a web browser and go to the URL https://jupyterhub.egr.msu.edu.
2. Type your engineering login name, which is your MSU NetID.
3. Enter your engineering password.

If everything is working properly you will see the main "Files" windows in the Jupyter interface.

Instructions for getting IPython notebook files into Jupyter

IPython notebooks, also known as Jupyter notebooks, are files that end with the .ipynb extension. We will give you two example files to work with, you can edit them for your own learning or research purpose. Please download the ipynb files from the our GitHub repository by using the following command line:

```
git clone https://github.com/msu-seismo/python-data-analysis-viz.git
```

or visit the website to browse the files in the repository:

https://github.com/msu-seismo/python-data-analysis-viz.git

You can load the ipynb files in the directory downloaded from our GitHub repository into Jupyter using the "upload" button on the main "Files" tab in the Jupyter Notebook web interface. Clicking on this button will cause a file browser window to open. Just navigate to your ipynb file, select it and hit the open button. Once you see your filename in the Jupyter Notebook window you can just click on that name to start using that file.

Resources

We recommend you to browse through Lion Krischer's GitHub repository:

https://github.com/krischer/mess_2019

Actually, only this following ipynb file is relevant to our tutorial on Tuesday:

1_monday/00_Python_Crash_Course.ipynb

Please download this Jupyter Notebook by clicking on "Raw" button, right clicking your mouse, and "save as" ipython file. This notebook provide some basics of python coding, from simple a + b to complex signal processing.

Alternatively, you can clone this entire GitHub repository to either your own computer or your iCER HPCC account by using the following command:

```
git clone https://github.com/krischer/mess_2019.git
```

If you really want to learn coding, instead of GUI you may want to get used to the command lines. Here are some ideas of setting up your command-line environment on your personal computer.

Check the operating system you use.

Linux:

If you use Linux, you might have learned how to use command lines. You can skip the rest of this section.

MacOS:

If you never used command lines in a terminal, hit command+space, search terminal and then open one.

MacOS also use bash. You can find plenty of bash scripting tutorials on YouTube. You may find the following three videos useful.

Bash Scripting Basics Part 1
Bash Scripting Basics Part 2
Bash Scripting Basics Part 3

Install the xcode command line tools. You may find this link helpful:

http://osxdaily.com/ 2014/02/12/install-command-line-tools-mac-os-x/.

Install homebrew: It is a useful software package manager on MacOS.

http://osxdaily.com/2018/03/07/how-install-homebrew-mac-os/

Windows10:

Install the ubuntu-subsystem on your computer. It is fast and safe, with nearly all the features of the ubuntu-linux available. This link may be helpful:

https://itsfoss.com/install-bash-on-windows/

After you successfully installed ubuntu-subsystem, please open command line tools by finding the 'ubuntu' in the startup menu, or open a cmd window and type bash.

You can also find relevant resources for learning linux through this web page:

seismic-training/linux-101 of Dr. Shawn Wei's Lab