

Authors: Andy Bean & Carrie Harris
DIFUSE Project - Astro-Imaging Module
Dartmouth College
May 2023

How to Colab on Local Runtime

Summary

Install Anaconda Navigator on your Computer

Create a virtual environment with the correct library versions for your Colab

Connect to a local runtime instead of a hosted runtime using Colab and bash commands

What is this guide for?

Colab allows you to run your code on a “hosted” environment. This is a Google virtual machine which has a set of packages pre-downloaded onto it that are difficult to downgrade. If Colab updates its packages and breaks your Colab file, you might want this guide!

Instead of using Google’s hosted environment that might no longer work, we can create a local virtual environment with exactly the right packages and connect to that instead. Using a local runtime also allows you to interact with files directly on your computer (so make sure you trust the code you are running since it can touch your personal files!).

Step 1: Anaconda Navigator

Anaconda Navigator is a package manager for several coding environments. Most relevant to us, it allows us to easily create a “virtual environment” that has exactly the package versions we want for our code.

Download and install Anaconda Navigator at: <https://www.anaconda.com/download>

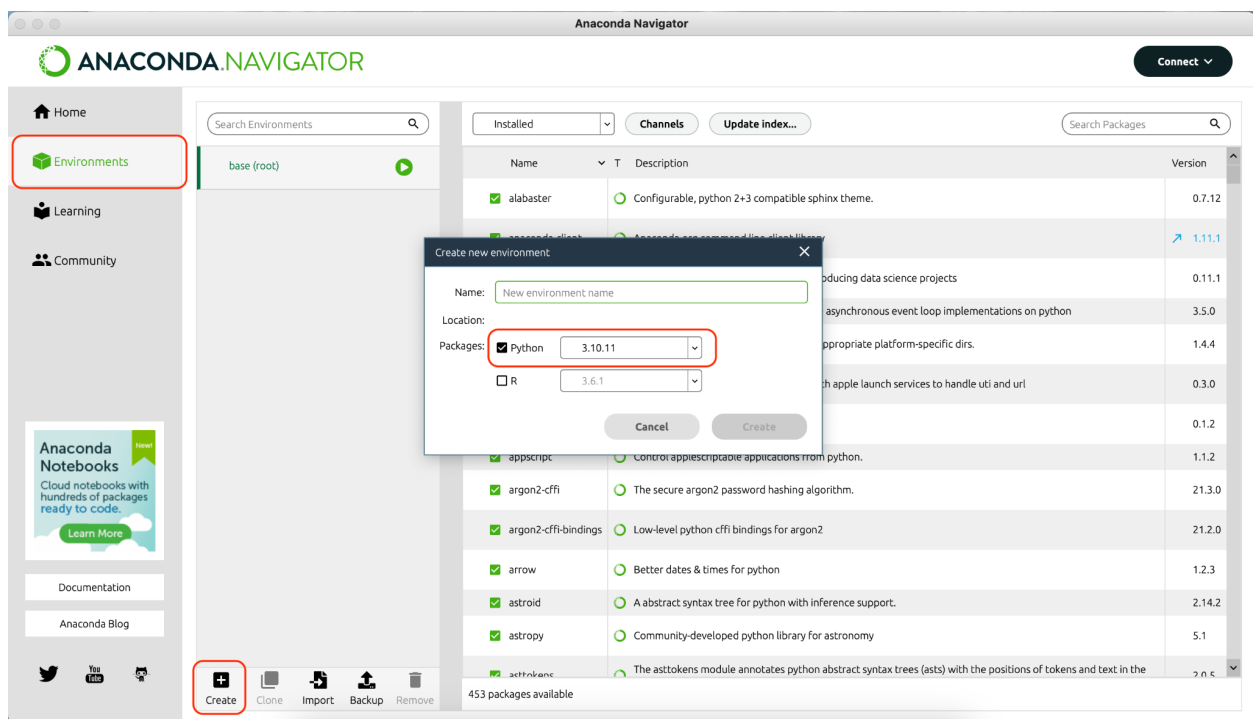
Step 2: The Virtual Environment

Create the Environment with the Right Language Version

Once Anaconda is installed, we can create the proper environment to run the Colab on. Go to “Environments” on the left, then “Create” at the bottom, and fill out the pop-up that appears with your proper version of Python

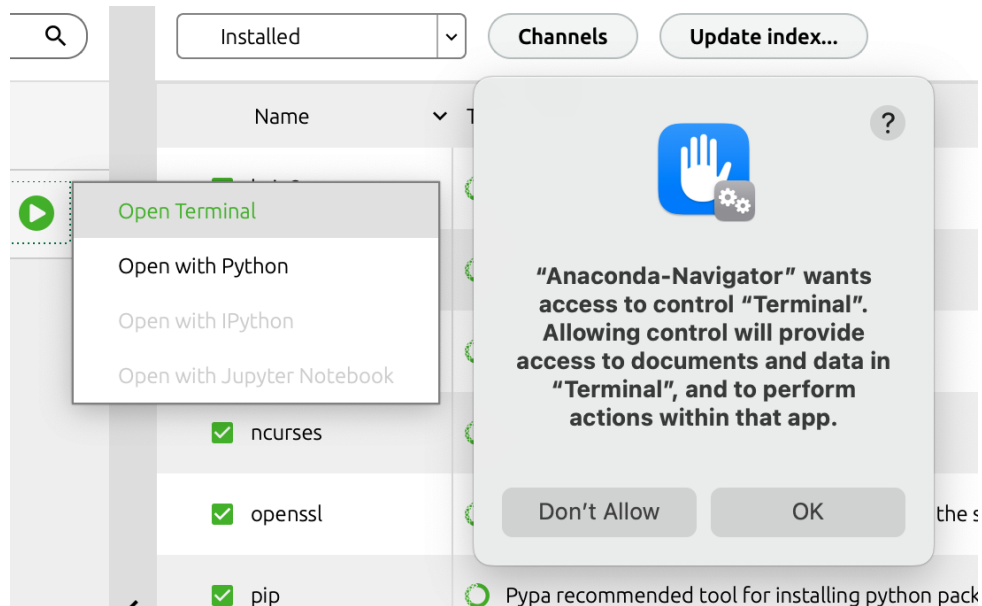
For the Astro-Imaging Colab, we want Python version 3.10.11

You can name the environment whatever you want, but a name with no spaces and no special characters is easier to use. E.g. “astroImagingEnv”



Put Packages on the Environment

Pressing “play” next to your new environment will open up a menu asking you how you want to run the environment. Select “Open Terminal” and allow Anaconda to access the computer terminal



In the terminal window that opens, we are going to install the packages we need. The command format for this is generally:

```
conda install package
```

The list of libraries we need for the Astro-Imaging Colab is below:

```
pandas v1.5.3
matplotlib v3.7.1
ipywidgets v7.7.2
scikit-learn v1.2.2
statsmodels v0.13.5
seaborn v0.12.2
appnope v0.1.2
matplotlib-inline v0.1.6
```

A few other libraries are necessary to run these libraries (e.g. numpy, scipy), and those will be automatically installed by installing these main ones. Those other necessary libraries are called “dependencies.”

So, type the following commands into your terminal and press enter to execute each command, one at a time:

```
conda install pandas
conda install matplotlib
conda install -c esri ipywidgets
conda install scikit-learn
conda install statsmodels
conda install seaborn
conda install appnope
conda install matplotlib-inline
```

After running each command, the console may say that several other packages are also needed. If the console asks you to confirm with [y]/ n, type “y” and hit enter. This is just to confirm that you agree to the packages being installed. Let the console take its time to install your packages!

If this method is a little daunting, you can also install packages through the Anaconda UI. Change your packages list to show “not installed” and then use the search bar to find a package. If that package is the right version, click the check box next to it and then press apply in the bottom right to install it. **HOWEVER, we need an older version of ipywidgets, so that one has to be installed with the console command,** or by adding the channel “esri” to the channels options to find and select ipywidgets version 7.7.2 instead of version 8.

The screenshot shows the Anaconda UI interface. On the left, there's a sidebar with 'base (root)' and 'astroimagingEnv'. The main area is titled 'Not installed' and shows a list of packages. A search bar at the top right contains 'matplotlib'. The list of packages includes 'basemap', 'basemap-data', 'basemap-data-hires', 'cmyt', 'descartes', 'ipyml', 'matplotlib', 'matplotlib-base', 'matplotlib-inline', 'mpl-scatter-density', and 'mpld3'. The 'matplotlib' row is highlighted, and its version '3.7.1' is circled in red. The 'matplotlib' checkbox is also checked.

| Name | Description | Version |
|--|--|---------|
| <input type="checkbox"/> basemap | Plot on map projections using matplotlib | 1.3.6 |
| <input type="checkbox"/> basemap-data | Plot on map projections (with coastlines and political boundaries) using matplotlib. | 1.3.6 |
| <input type="checkbox"/> basemap-data-hires | Plot on map projections (with coastlines and political boundaries) using matplotlib. | 1.3.6 |
| <input type="checkbox"/> cmyt | A collection of matplotlib colormaps from the yt project | 1.1.3 |
| <input type="checkbox"/> descartes | Use geometric objects as matplotlib paths and patches. | 1.1.0 |
| <input type="checkbox"/> ipyml | Matplotlib jupyter extension | 0.8.7 |
| <input checked="" type="checkbox"/> matplotlib | Publication quality figures in python | 3.7.1 |
| <input type="checkbox"/> matplotlib-base | Publication quality figures in python | 3.7.1 |
| <input type="checkbox"/> matplotlib-inline | Inline matplotlib backend for jupyter | 0.1.6 |
| <input type="checkbox"/> mpl-scatter-density | Matplotlib helpers to make density scatter plots | 0.7 |
| <input type="checkbox"/> mpld3 | D3 viewer for matplotlib. | 0.5.7 |

Step 3: Connect to Local Runtime

This next step requires “jupyter,” but that was automatically installed with anaconda, so we don’t need to do anything! (If jupyter is giving you trouble, you may need to uninstall and reinstall it from the home page of Anaconda navigator)

Run these commands in your console to obtain a link to connect with your runtime (you can copy and paste them from this document):

```
jupyter notebook \
  --NotebookApp.allow_origin='https://colab.research.google.com' \
  --port=8888 \
  --NotebookApp.port_retries=0
```

Once you obtain a link, **copy it to your clipboard!** You’re looking for something like the link highlighted below:

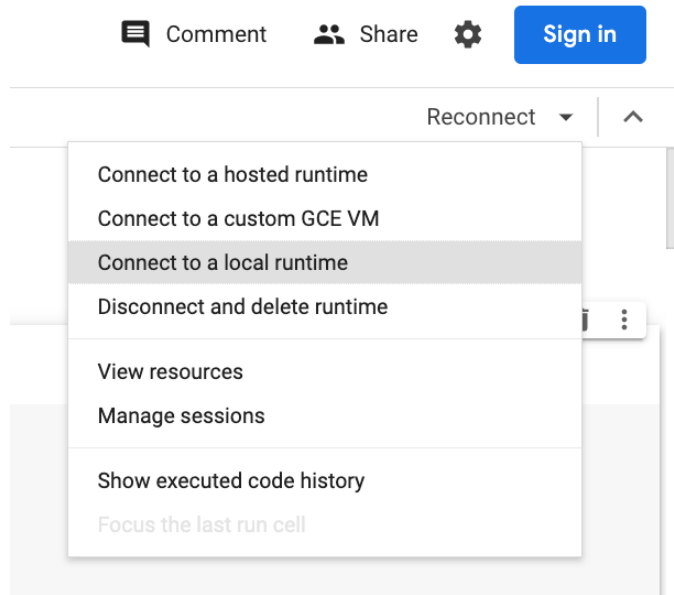
```
https://jupyter-notebook.readthedocs.io/en/latest/migrate_to_notebook7.html

Please note that updating to Notebook 7 might break some of your extensions.

[I 15:59:58.853 NotebookApp] Serving notebooks from local directory: /Users/abea
n
[I 15:59:58.853 NotebookApp] Jupyter Notebook 6.5.4 is running at:
[I 15:59:58.853 NotebookApp] http://localhost:8888/?token=36c7d15803edc8f6e5dba8
93705e69f08faa37b0edc3d44d
[I 15:59:58.853 NotebookApp] or http://127.0.0.1:8888/?token=36c7d15803edc8f6e5
dba893705e69f08faa37b0edc3d44d
[I 15:59:58.853 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[C 15:59:58.861 NotebookApp]

To access the notebook, open this file in a browser:
file:///Users/abea/Library/Jupyter/runtime/nbserver-15582-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=36c7d15803edc8f6e5dba893705e69f08faa37b0edc
3d44d
or http://127.0.0.1:8888/?token=36c7d15803edc8f6e5dba893705e69f08faa37b0edc
3d44d
```

Then, go to your Colab file. In the top right, press the arrow next to connect and select “Connect to a local runtime.” In the box that appears, paste your link and click the button to confirm! **You’ve got a local runtime!** You can now interact with your colab in the colab workspace with the right packages for things to work. Your file will save output and markdown blocks as normal.



Notes:

- You'll still need to upload your files to the session storage as per the normal "how to Colab" instructions, or otherwise put it on the top level of your computer (immediately under your "user" folder). You could also give your code the more specific file path to access your data file
- Some packages specific to google colab can no longer be used, so those library calls and related comments will need to be commented out. This module has a backup code version to use with local runtime that already has these removed.
- To exit out, you can quit your process from the terminal or the "home page" that appears when Jupyter is launched
- If you need to quit and connect again, you do not need to reinstall any packages; just skip to step 3 to launch your runtime!