How to install CellProfiler from source with all plugins on Mac OS

1. Install Java 11 here

2. Install or update conda

   ○ Note: you can get the command to update conda by typing `conda update` on your command line. The command will generally look like:

   ```
   conda update --prefix /Users/USERNAME/opt/anaconda3 anaconda
   ```

   For beginners, we recommend you use Anaconda Navigator since it is more beginner-friendly. Download Anaconda from the website here, then open Anaconda Navigator

3. Download the environment file

   You can download the whole repo by cloning it with git or simply clicking the green **Code** button on the repo page and selecting **Download ZIP** (see below) and then extract the ZIP folder contents.

   Alternatively, you can copy and paste the contents of the .yml file into a text editor like Notepad. If you do this, make sure you save it as "CellProfiler_plugins_windows.yml" and as type "All Files" and **NOT** "Text file".
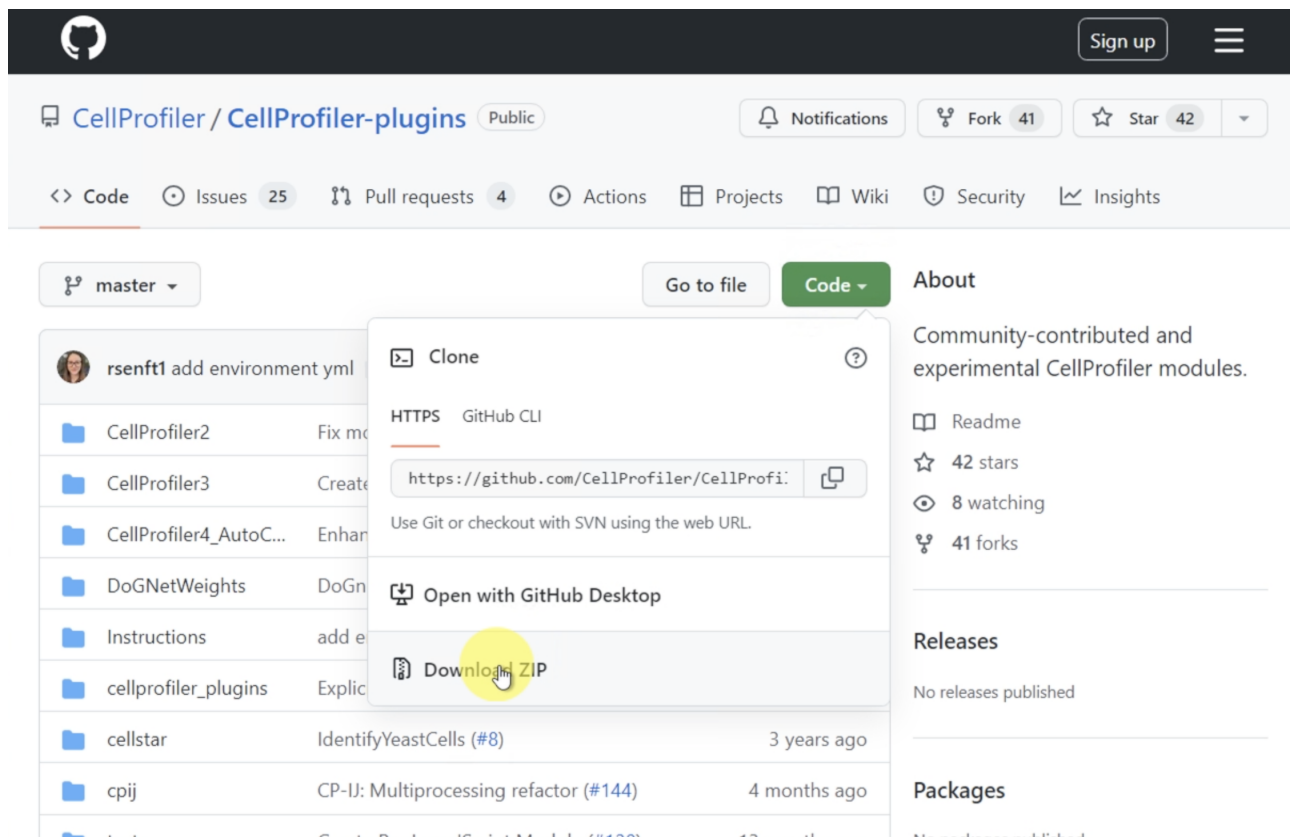
4. Try to create a new environment from the included .yml file

   Warning, this step may take a while

Open Anaconda Navigator and select the **Environments** tab on the left. We recommend you create the environment from the command line. To do this, Select the play button next to your base (root) environment and select **Open Terminal**:
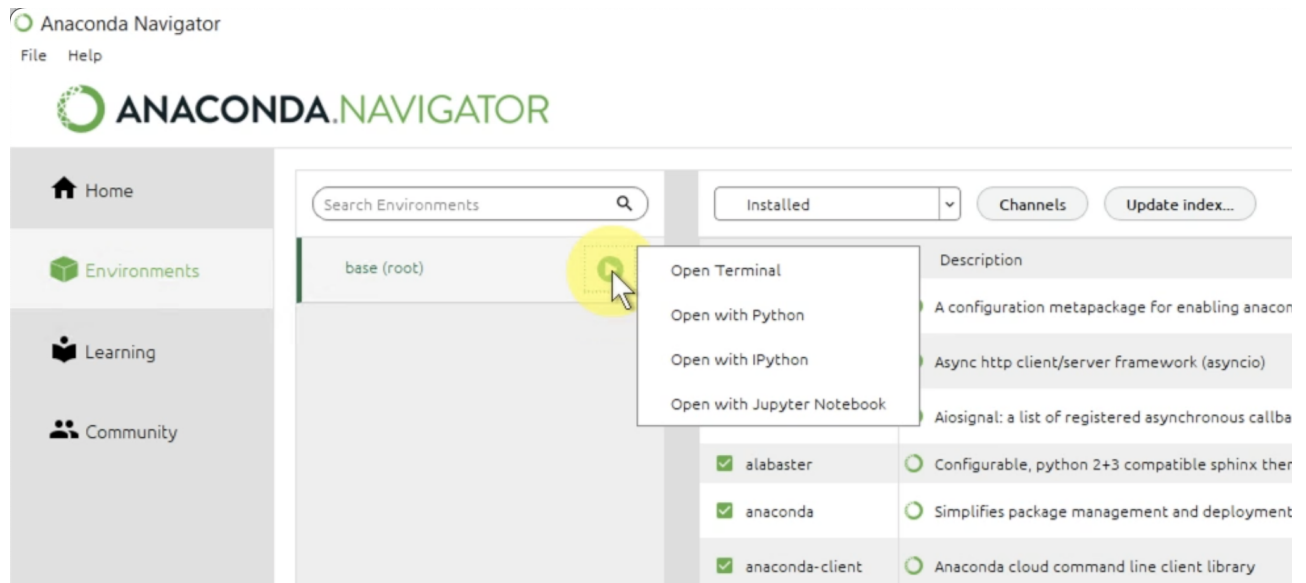
In the terminal, navigate to where your environment file is located with `cd PATH_TO_FOLDER` where `PATH_TO_FOLDER` is the path to the directory containing your yml file (e.g., `/Users/USER/Desktop`).

Then in the terminal window that pops up, enter the following command:

```
conda env create -f CellProfiler_plugins_mac.yml
```

5. Activate your environment

   In your terminal, enter `conda activate Cellprofiler_plugins` to activate your environment

6. Verify that cellprofiler is installed correctly by running it from the command line.

   In your terminal, type in `pythonw -m cellprofiler` and hit Enter. this will open CellProfiler or will give you an error message.

7. Install other packages for other plugins (just for RunStarDist)

   In terminal with your environment activated, enter:

```
pip install stardist csbdeep --no-deps
```

If you would like to use the omnipose models in cellpose, ensure you have cellpose 1.0.2 (you should by default if you've used our environment yml) and enter on the command line (in your activated environment):

```
pip install omnipose
```

8. Clone the CellProfiler-plugins Repo

   If you have not already downloaded the repo, download it from here.

   You can also use git or GitHub Desktop to clone the repo if you prefer.

9. Connect CellProfiler and the plugins repo

   With your environment active, type `pythonw -m cellprofiler` in terminal to open CellProfiler if it is not open already.

- In CellProfiler, go to **File** then **Preferences…**
- Scroll down and look for "CellProfiler Plugins Directory" on the left.
- Select the **Browse** button and choose the folder where you extracted the CellProfiler plugins files. It is probably called "CellProfiler-plugins-master" unless you have renamed it.
- Select **Save** at the bottom of the Preferences window
- Close CellProfiler and reopen it by typing `pythonw -m cellprofiler` on the command line

NOTE: You might get a warning like this:

```
W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not
load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not
found
2022-05-26 20:24:21.906286: I
tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart
dlerror if you do not have a GPU set up on your machine.
```

If you don't have a GPU, this is not a problem. If you do, your configuration is incorrect and you need to try reinstalling drivers and the correct version of CUDA for your system.

10. Verify that the installation worked

```
Add a module to your pipeline by hitting the **+** button in the pipeline
panel (bottom left)

In the "Add Modules" window that pops up, type "run" into the search bar.
You should be able to see plugins like RunCellpose and RunStarDist if the
installation was successful:
![](images/Install_environment_instructions_windows/2022-06-02T21-43-
56.png)
```

Common errors

1. My wheels are failing to build

- If you get a message like "ERROR: Failed building wheel for pyzmq" this usually means that you do not have pyzmq installed. Try to reinstall pyzmq.

2. Java virtual machine cannot be found

- If you're getting errors about Java, it means that java is not being configured properly on your system.
- Make sure you have installed The Java Development Kit 11 here. Note that newer versions may not work.
- Make sure you've added environment variables at the **System** level and not at the **User** level.

```
brew install java

# For the system Java wrappers to find this JDK, symlink it with
sudo ln -sfn /opt/homebrew/opt/openjdk/libexec/openjdk.jdk
/Library/Java/JavaVirtualMachines/openjdk.jdk

# Set version in zshrc
echo export JAVA_HOME=$(/usr/libexec/java_home -v 1.8) >> ~/.zshrc
source ~/.zshrc
```

3. Installing pyzmq failed

- You might get an error when trying to install pyzmq. Something like

```
ERROR: Command errored out with exit status 1:
```

And earlier in the traceback:

```
Fatal: Cython-generated file 'zmq\backend\cython\_device.c' not found.
                Cython >= 0.20 is required to compile pyzmq from a
development branch.
                Please install Cython or download a release package of
pyzmq.
```

To fix this, `conda install cython`