PLEASE

PLEASE: The Python Low-energy Electron Analysis SuitE - Enabling rapid analysis and visualization of LEEM and LEED Data.

Author: Maxwell Grady

Affiliation: University of New Hampshire Department of Physics

Current Version: 1.0.0

The PLEASE software package is written in python and depends on a number of third party libraries from the standard scientific python stack for functionality. All libraries used are freely available and relatively straightforward to install.

Dependencies:

- Numpy >= 1.12.0
- scipy >= 0.19.0
- pillow >= 4.0.0
- PyQt >= 5.6.0 ** Note, for Riverbank Computing Ltd. provides two versions of their Qt python bindings, PyQt4 and PyQt5. The PyQt5 APIs are NOT backwards compatible to PyQt4, thus it is imperative to use PyQt5 in order to run PLEASE.
- Pyyaml >= 3.12.0
- Pendulum >= 1.1.0
- Pyqtgraph >= 0.10.0 ** Note, there are a few minor API changes

between version 0.9 (provided by Anaconda) and 0.10 (installable via pip). Thus it is imperative to use pyqtgraph version 0.10 or higher.

Python Version:

Python versions 2.7 and 3.5+ are supported. All other python versions are not officially supported. The Anaconda Python distribution, provided free of charge by Continuum Analytics, is suggested for ease of use and community support. All python modules required by PLEASE are available via a combination of Anaconda's package manager, conda, and the standard python package manager, pip.

Pre-Setup Notes:

Many computer operating systems come pre-installed with the python programing language and some of the system utilities may rely on this installation in the background. Thus, it is **highly** suggested that you do not alter the pre-existing python installation on your machine in order to run PLEASE. Detailed here will be a guide for how to setup a 'virtual environment' which isolates PLEASE and its dependencies from your system python installation.

This guide will demonstrate how to install the Anaconda Python Distribution, provided free of charge from Continuum Analytics, how to create an isolated python environment for running PLEASE, and how to install all of the required dependencies.

Obtaining Anaconda Python

Continuum Analytics provides two options for using the Anaconda python distribution. You can install the full Anaconda python distirbution,

which contains a python interpreter along with a large number of commonly used scientific packages. Or you can use Miniconda, which provides you access to the package manager, Conda, and leaves it to the user to decide which python version and which packages to install. Installing miniconda will require the smallest amount of disk space and smallest amount of downloaded data. Thus I suggest using Miniconda.

1. The Anaconda Python Distribution can be downloaded from https://www.continuum.io/downloads
2. Miniconda can be downloaded from https://conda.io/miniconda.html
3. I recommend using the Python 3 version, however, this code will also work with python 2.7
4. Follow the instructions provided by Continuum Analytics for installation and setup of Anaconda
5. When the installation is finished, you should have access to the Anaconda Python distribution from your Command Line / Terminal.
6. Alongside the python distribution installed with Anaconda is a package and environment manager called 'conda' You may need to restart your terminal before the 'conda' command is recognized.

Setting up a Python Environment for PLEASE

1. First we will create a python environment for PLEASE and its dependencies to isolate this from other python installations and environments.

2. Execute the following line in the Terminal to create a conda environment called PLEASE:

```
conda create -n PLEASE
```

Note if you downloaded miniconda instead of the full Anaconda distribution, you will also want to specify the python version:

```
conda create -n PLEASE python=3.5
```

3. Activate the PLEASE environment with the following line:
    - OS X + Linux:

```
source activate PLEASE
```

    - Windows:

```
activate PLEASE
```

4. Now any packages installed will be isolated to this python environment
5. At any time the environment can be deactivated/reactivated by executing the following commands:
    - OS X and Linux:

```
source deactivate
```

    - Windows:

```
deactivate
```

- OS X and Linux:

```
source activate PLEASE
```

- Windows:

```
activate PLEASE
```

Installing dependencies

1. We will use two tools to install the required packages for PLEASE.

   a. conda - the Anaconda package and environment manager

   b. pip - the standard python package manager

2. Make sure you have activated the PLEASE environment as shown above.

3. Numpy, Scipy, Pillow, and Pyyaml will be installed with conda

4. Execute the following:

```
conda install numpy scipy pillow pyyaml
```

5. Type Y and press enter when prompted to install the packages

6. The rest of the dependencies will be installed with pip

7. Execute the following:

```
pip install pyqt5 pendulum pyqtgraph
```

8. Type Y and press enter if prompted to continue the installation

9. All required packages should now be installed

Downloading PLEASE source

1. Note: The PLEASE source code is rather small, however, included with PLEASE are two test data sets along with the corresponding Config files to facilitate loading the data. The inclusion of this data makes the total download ~600Mb. Keep this in mind when downloading the source.

2. The PLEASE source code is hosted at https://www.github.com/mgrady3/PLEASE

3. The source can be downloaded from the website or via the commandline if you have git installed

   a. download from the web by clicking the green "Clone or download". Select Download zip. Extract the Zip to your Desktop.

   b. using git: from the command line navigate to your Desktop directory and then execute:

```
git clone https://www.github.com/mgrady3/PLEASE
```

Source structure

Within the main source tree for PLEASE, the outermost directory contains information files such as the README, the LICENSE, etc. Subdirectories for media files, test data, and source code are contained

in the top level directory. All python and other language source files are located in the /source/ directory.

Executing PLEASE

1. To start PLEASE, first activate the PLEASE environment:
    - OS X + Linux:

      ```
      source activate PLEASE
      ```

    - Windows:

      ```
      activate PLEASE
      `
      ```

2. The file that needs to be executed to start the application is called main.py. This file resides in /PLEASE/source/.
3. Execute the main python file to start the application:

   ```
   python /path/to/PLEASE/source/main.py
   ```

4. If all dependencies are installed, the application should launch and the main Graphical User Interface should be visible.
5. If the application does not start, error messages should be displayed in your Terminal. The most common sources of error will be missing dependencies. Ensure that you have setup a python environment and installed all the required dependencies as listed above. Also ensure that the correct python environment is active when trying to

start PLEASE. When your PLEASE environment is active, you should see something similar to (PLEASE) at the start of the command prompt which indicates that the PLEASE environment is active.

6. **Windows Users**: If you see a message along the lines of **"ImportError: Failed to load DLL ..."** This a most likely a mesasge indicating that the Qt/PyQt installation is not working properly. This tends to happen when PyQt5 is installed via pip and either the path is not set correctly or the Qt executables are not placed in the correct directories. As a workaround try this:

```
pip uninstall pyqt5
```

```
conda install pyqt=5
```

```
python /path/to/PLEASE/source/main.py
```

Make sure you execute the commands above after activating your PLEASE environment.

Optional: Creating a launch script

There are a number of ways to arrive at an executable file that will run the PLEASE software.

Option 1. Making main.py executeable:

1. You will need to know the exact path to the python executable that your conda (or other) environment uses. For example, if you created an environment called PLEASE, the environment files should be

installed in a path similar to $HOME/Anaconda/envs/PLEASE. where $HOME is your user home folder. On OS X this is /Users/YourUserName/

2. The python executable is located in the /bin/ directory of the above mentioned path: $HOME/Anaconda/envs/PLEASE/bin/python

3. Edit the main.py file and add the following line as the topmost line of the file:

```
#!/Users/YourUserName/Anaconda/envs/PLEASE/bin/python
```

4. Now use the terminal to set the main.py file as executable:

```
sudo chmod u+x /path/to/PLEASE/source/main.py
```

5. You should now be able to start PLEASE by double clicking the main.py file

6. **Note:** You must leave main.py in the source folder, so this method is less convenient.

Option 2. Create a separate bash script that will execute main.py and make this file executable. **(Recommended)**

1. The instructions here will differ depending on your OS. Shown here are instructions of OS X

2. Create a file in your text editor of choice with the following lines

```
source activate PLEASE
python /path/to/PLEASE/source/main.py
```

3. Save this file to any location you like and name it PLEASE-Start.command

4. Use your terminal to set the file as executable:

```
sudo chmod u+x /path/to/PLEASE-Start.command
```

5. You should now be able to run PLEASE by double clicking the PLEASE-Start file

Option 3. Using a 'Freezing' package to create an executable (.app, .exe, etc...) There are a number of python packages that allow you to package your project source code alongside a python interpreter and all the required libraries and bundle all of this into an executable such as a .app file on OS X or a .exe on Windows. Some of the more popular options for doing this are Py2exe (Windows), Py2app (OS X), pyinstaller (OS X, Linux, Windows), and CxFreeze (OS X, Linux, Windows). A full explanation of how to create an executable bundle using these tools is beyond the scope of this document, but in principle it should be possible on all operating systems.