

DATA ANALYSIS PACKAGE

MATTHEW BURIGANA
FEBRUARY 21ST, 2019

DATAANALYSIS PACKAGE

- Four python scripts used to plot the various types of data we collect
- plateReader generates plots for combi cell data
- cellReader takes data exported from Swagelok cells
- phaseDiagram takes ternary phase data, with optional capacities of each phase, and plots a ternary phase diagram
- pxrd plots PXRD data either exported from Rietica with a fit or unfitted data converted to a .csv

ANACONDA3

THE COMPILER FOR RUNNING SCRIPTS

- Data science platform for python
- Includes plotting and math libraries used during analysis
- Provides the terminal used to interact with scripts without using an IDE
- <https://www.anaconda.com/distribution/#download-section>

GENERAL NOTES

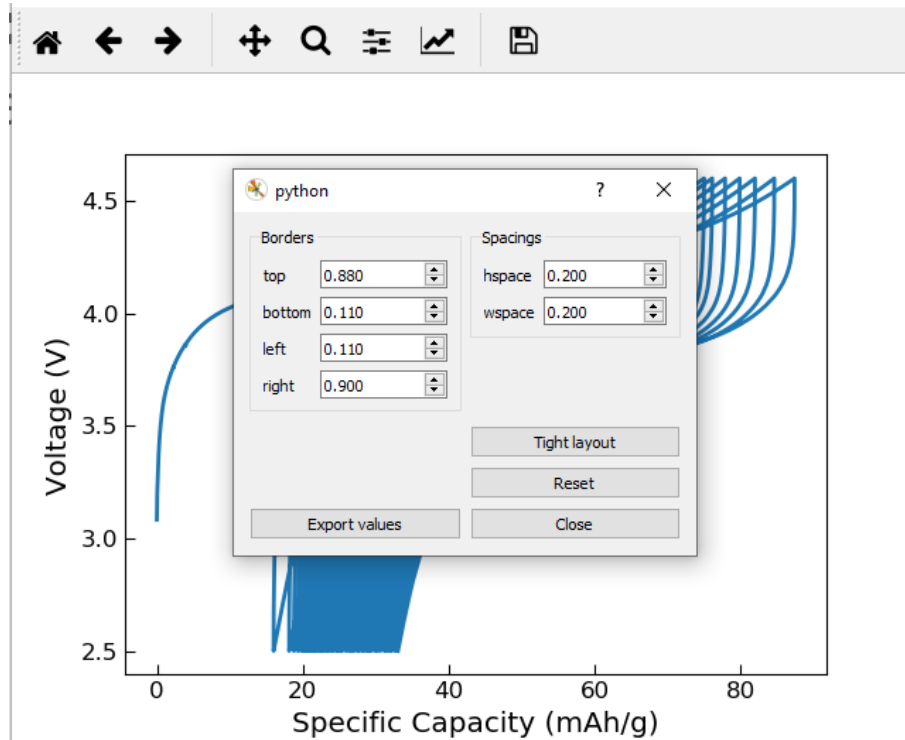
- Make sure the DataAnalysis folder is copied into the Anaconda3 folder
- Use the python.exe application found in the Anaconda3 folder to use any of the scripts and their respective methods
- Importing any script to the terminal follows the command pattern:
“from DataAnalysis.*scriptName* import *”
- All method names are case sensitive, file names are not
- plateReader and phaseDiagram do not support multiple loaded files, cellReader and pxrd do
- Methods with parameters set equal to a value have defaults
 - plotVoltageWindow(vMin, vMax, cycle = 1, fileNum = 1)

USING THE SCRIPTS

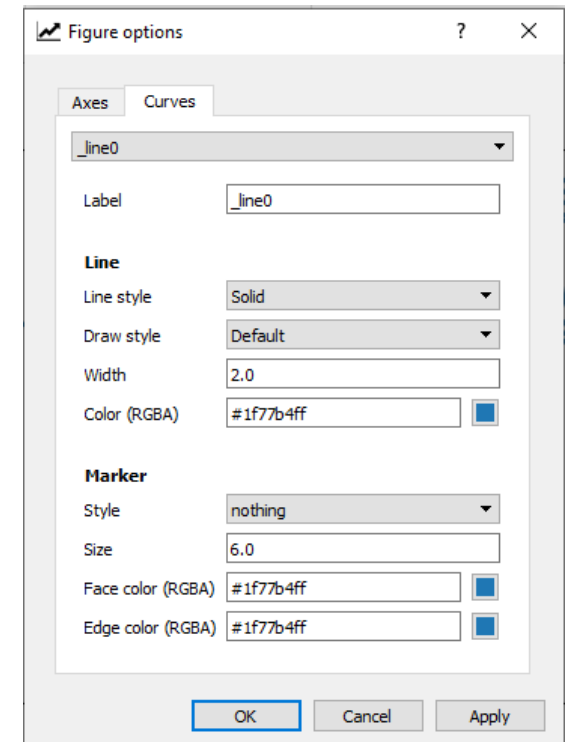
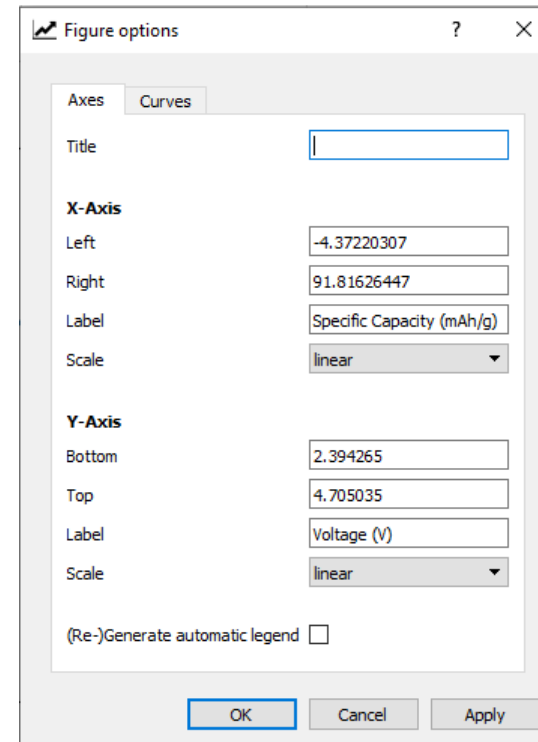
- Enter “from DataAnalysis.*scriptName* import *”
- Every script will now prompt you to enter the file name of the file you want to analyze
- Enter the filename without the extension
 - Ex. exampleFile.csv only needs “exampleFile” entered
- Enter the method name with the desired parameters
 - Ex. plotVoltageWindow(3.5, 4.2, 3, 2)
- Save the pop-up plot after editing any of the image parameters

EDITING PLOTS

- Configure Subplots: 



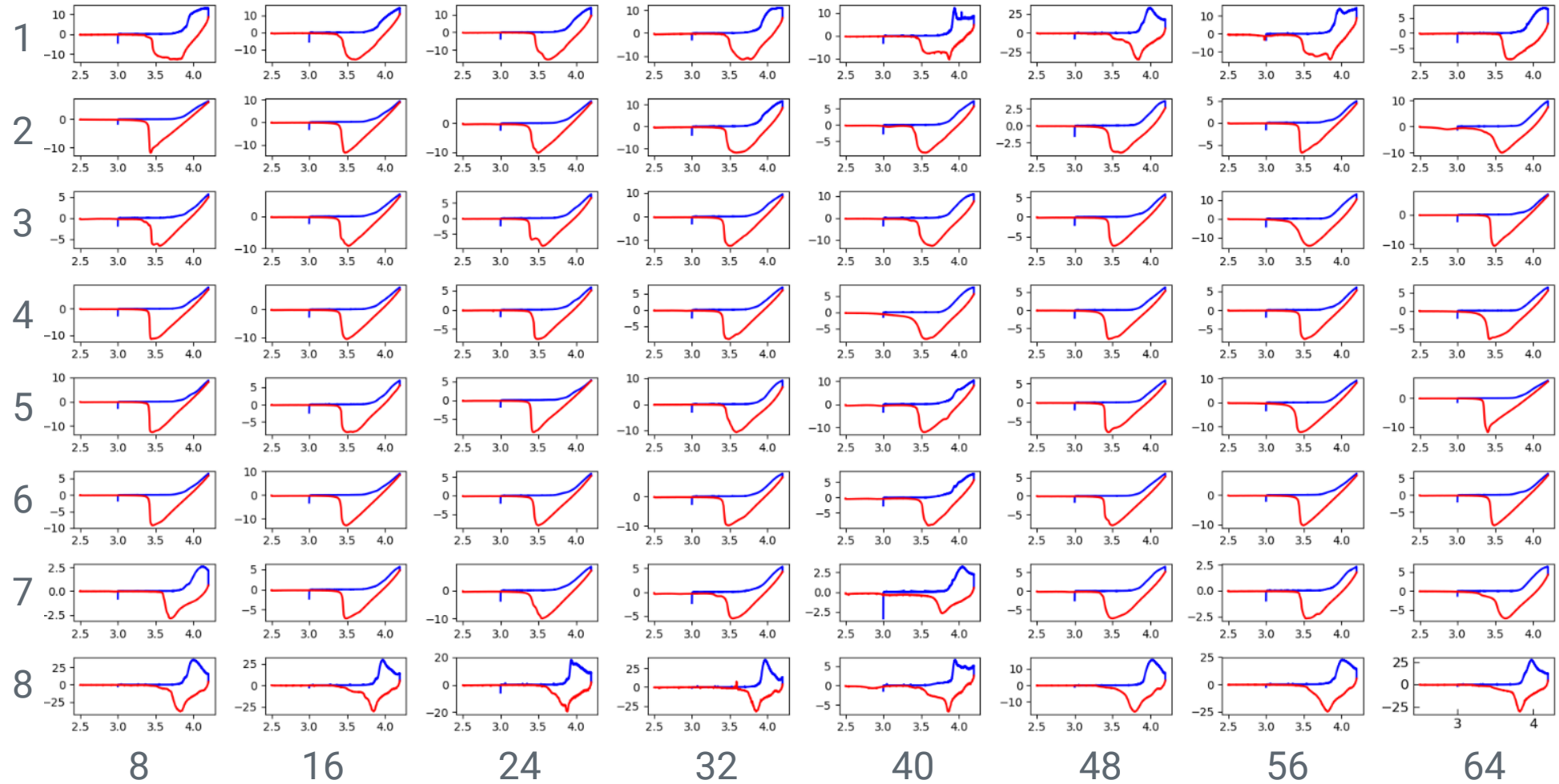
- Edit axis, curve, and image parameters: 



PLATEREADER

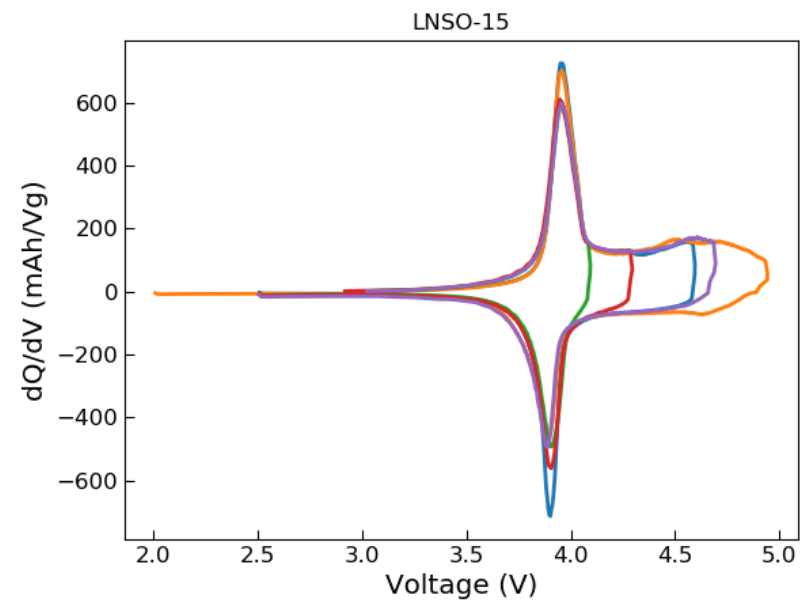
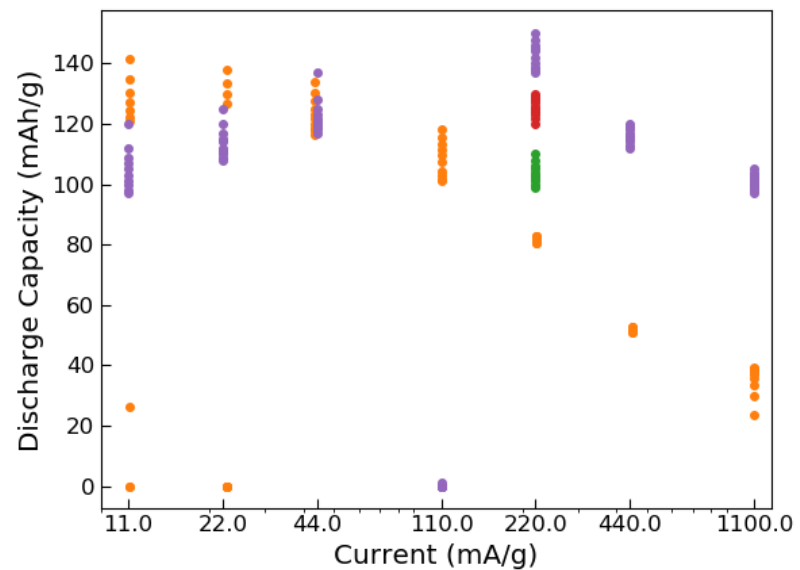
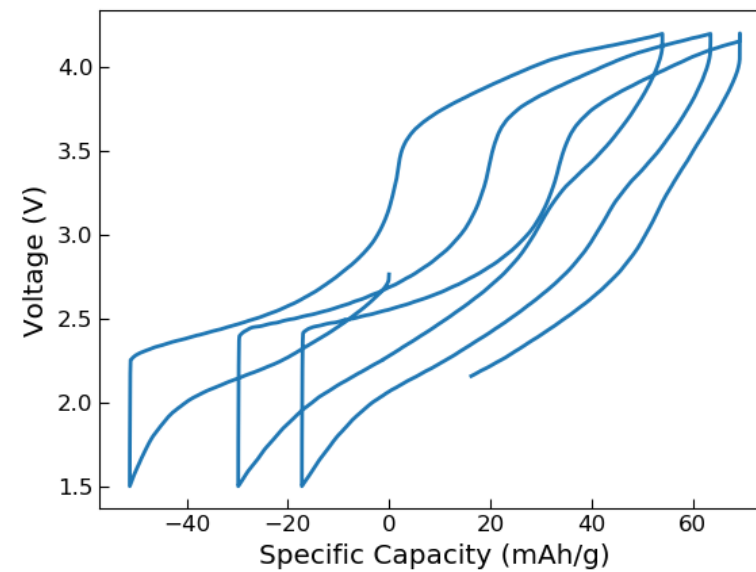
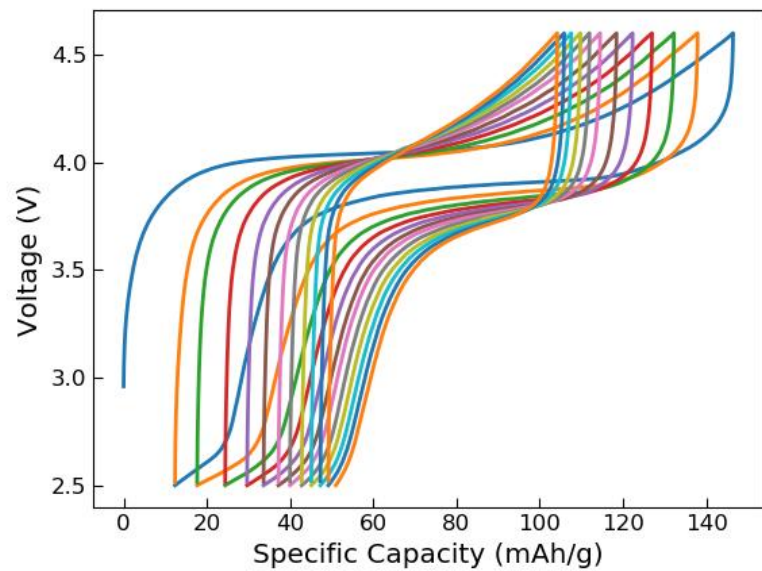
- Reads combi plate data for 64 channels
- Has an optional mass file that can be loaded to remove the need for entering the mass of each channel individually
 - Example of mass file format is in the Shared>>Code folder, called “massFile.csv”
 - Masses in the same location as masses in the exported file format
 - Skip the mass file by clicking enter without typing anything
- Can export file data to a .csv file
 - Exports masses of each channel, channel charge and discharge capacities, channel resistances, the average resistance, and standard deviation of the resistances
- For 64 channel plots, select “Configure Subplots”, then “Tight Layout” to spread out the plots

NUMBERING FOR 64 CHANNEL PLOTS



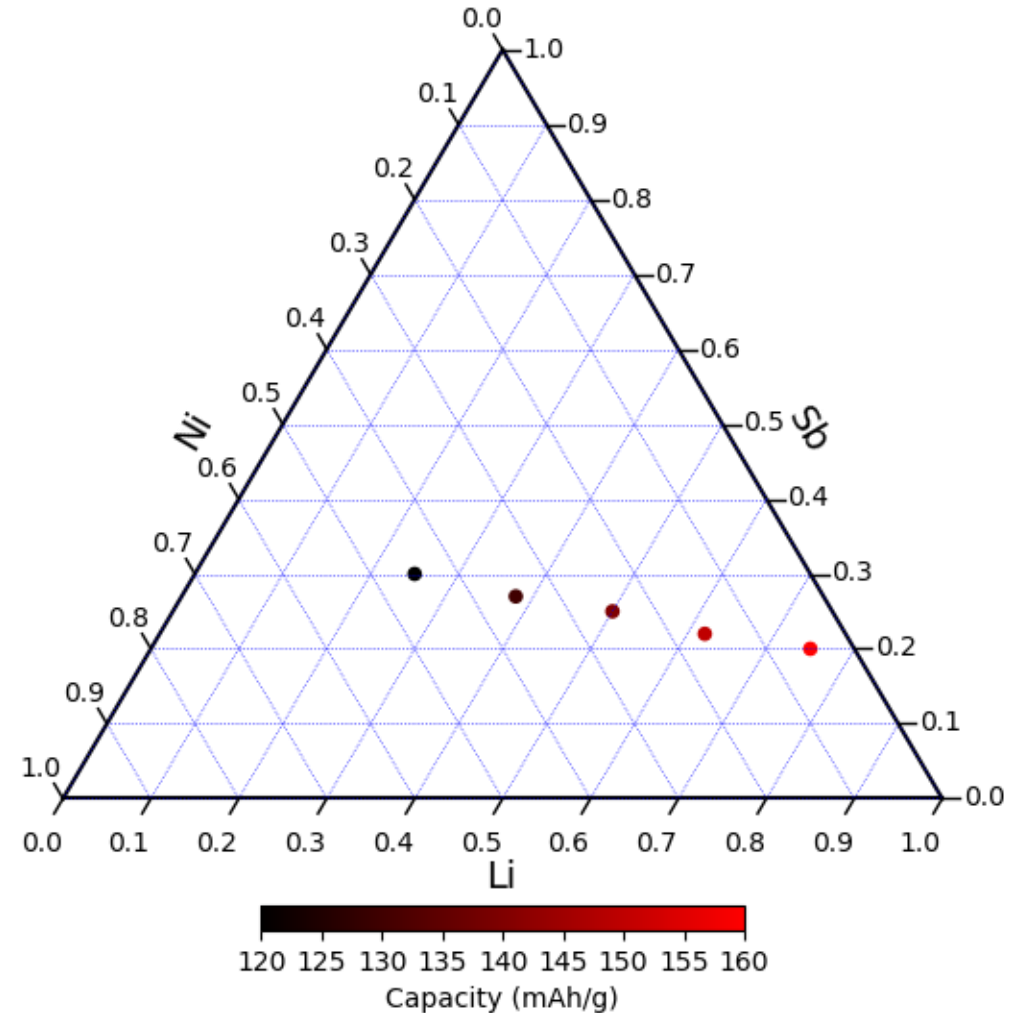
CELLREADER

- Exported as a “General Report” of type CSV from Neware
- Calculates masses from the discharge capacity divided by the specific discharge capacity, which is based on the mass entered before starting the cell
 - Cells that do not reach the first discharge cycle will give an error
- Plots overlaying data sets will plot 10 sets before restarting the color scheme
- Cells starting on a discharge will have the discharge considered the first cycle
- `plot(fileNum = 1)` calls three other methods to generate plots for the input file.



PHASE DIAGRAM

- Plots a ternary phase diagram
- Files formatted in the same way as phaseDiagramFile.csv
- Labels come from top row titles, with the left, middle and right corresponding to the bottom, right, and left labels
- Requires the Python-Ternary library added to Anaconda3
- Requires all three columns to contain data
- Note: capacities are optional, just delete the last column in the CSV file



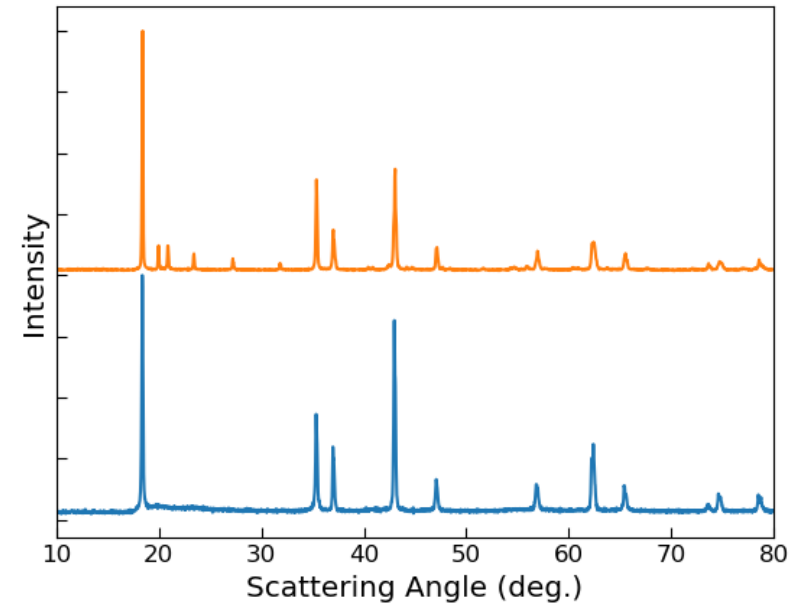
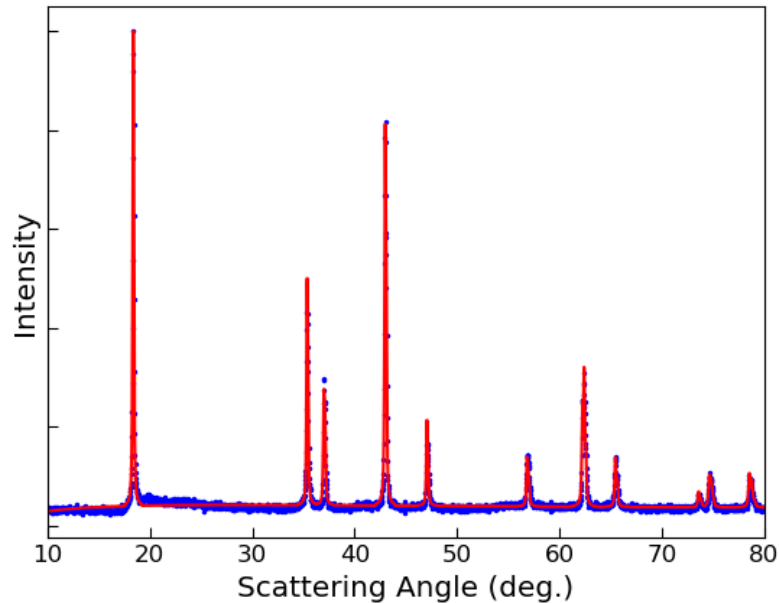
PYTHON-TERNARY LIBRARY

REQUIRED FOR PHASEDIAGRAM

- If it is already installed, a folder called “ternary” can be found in the file path “Ananconda3\Lib\site-packages”
- Added through the Anaconda Prompt
- Follow commands listed in DataAnalysis Package Instructions to complete installation
- Citation for python-ternary
 - Marc Harper et al. (2015). python-ternary: Ternary Plots in Python. Zenodo. 10.5281/zenodo.34938

PXRD

- Plots .asc or .xy files converted to .csv
- Plots Rietica plots exported as .txt then converted to .csv with the fit
- Plots multiple files as a stack plot, loaded with or without the fit



CONVERTING PXRD FILES TO CSV

- From the refined plot in Rietica, go to File>>Export to Excel/Sigma plot file
- Save the file as .txt
- The following procedure also applies to .asc or .xy files saved from X'pert highscore
- Open in Excel
 - Select the file description as “Delimited”, then click “Next>”
 - Under Delimiters, select “Space”
 - Ensure the data appears in separate columns in the File Preview
 - Click “Finish”
 - If the first column is empty, move all the data over one column to fill it
- Save the file as a .csv file
- Rietica exports are expected to have a header, files without a fit are not

CHANGING THE FILE PATH

- Open any script in a text editor (I recommend Notepad)
- Right click on a file in the desired folder, select “Properties”
- Copy the Location of the file
- Paste the Location into the “filepath” variable in any script
 - Found on line 6 of every script
 - Keep the file path enclosed by the apostrophes
- Add a second backslash to any location where there is one and add two at the end
 - filepath = 'C:\Users\McCalla Lab\Desktop\Data' should become
filepath = 'C:\\Users\\McCalla Lab\\Desktop\\Data\\'
- Save the script and close it

SYNOLOGY BACKUP

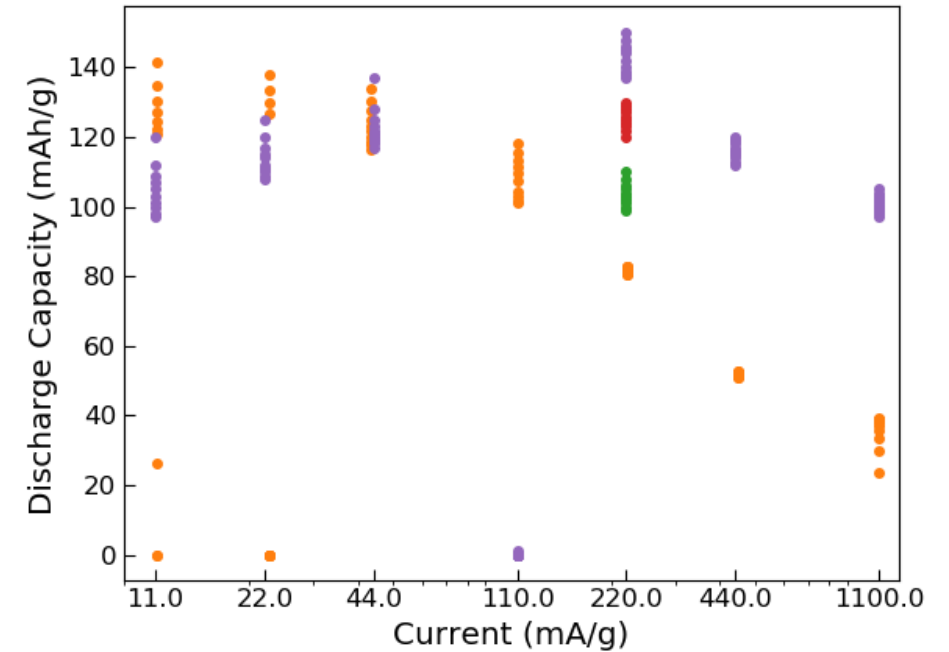
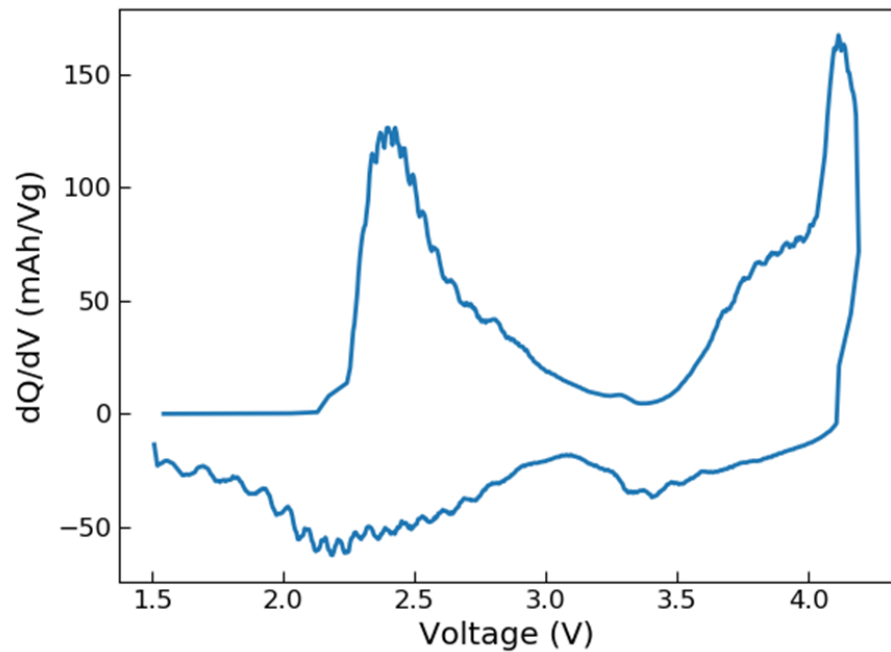
- Shared>>Code
 - Example files for each script are in “Code”
 - plateFile.csv: example file for plateReader
 - massFile.csv: example file containing correctly formatted masses used by plateReader
 - plateReader export.csv: example export from plateReader
 - cellFile.csv: example file for cellReader
 - blank.csv: example filler file for cellReader
 - phaseDiagramFile.csv: example file for formatting used by phaseDiagram
 - xrdFile.csv: example .asc or .xy file converted to .csv for pxrd
 - xrdFit.csv: example Rietica plot export (.txt) converted to .csv for pxrd
- Shared>>Code>>DataAnalysis
 - Contains newest version of each script

GITHUB

- Used for version control
- Contains history of all new versions of scripts since January 2019
- <https://github.com/matthewburigana/DataAnalysis>

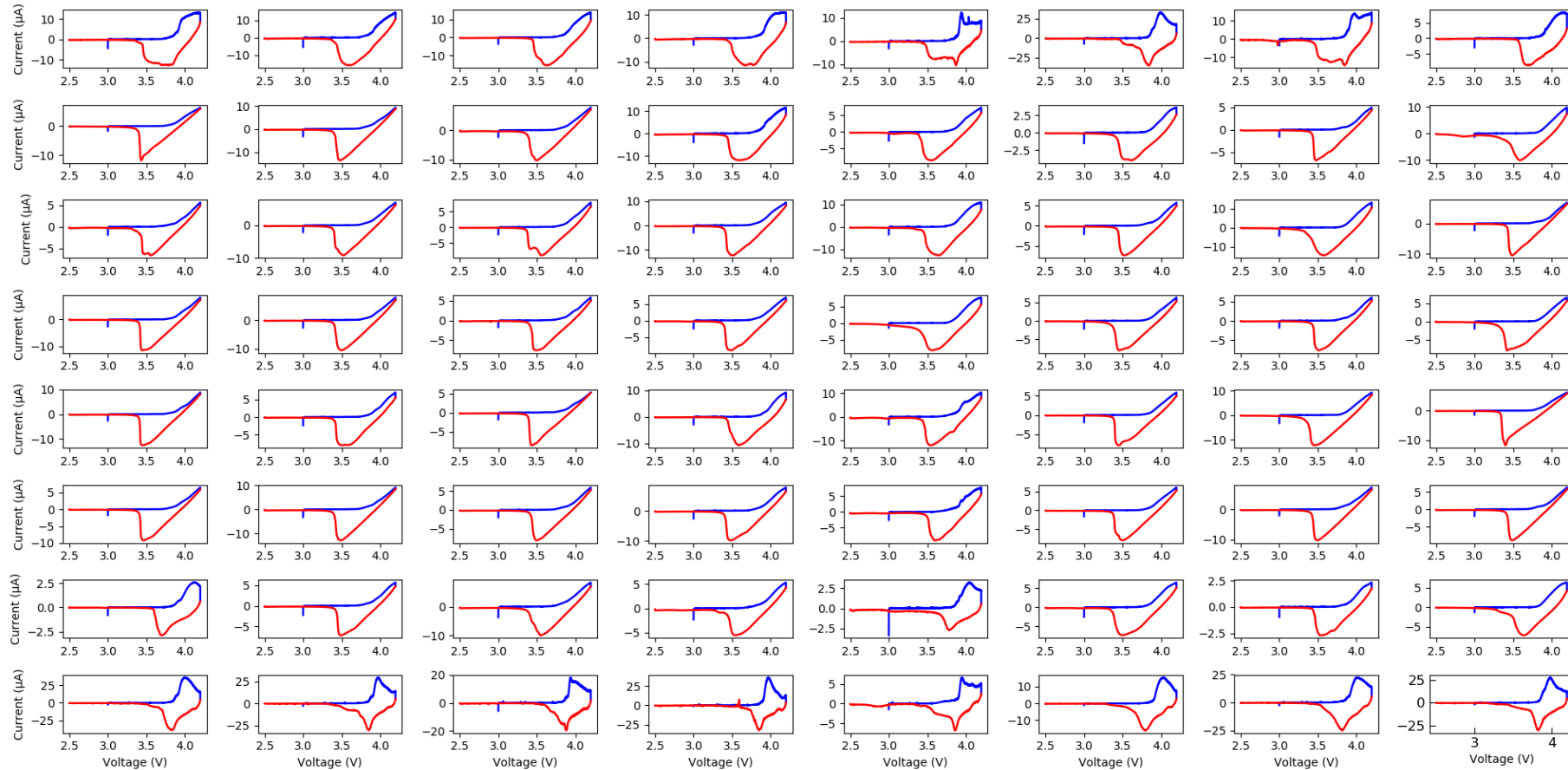
FUTURE CHANGES

- Remove 0 points from capacityVsCycle() and capacityVsRate()
- Smooth dQ/dV Curve
 - Make boxcar a variable
- Make plateReader tight layout automatic



FUTURE CHANGES

- Allow plateReader to hold multiple files
- Single axis label on 64 channel plateReader plots



SUGGESTIONS?