Total Differential Capacity Plot Analysis Using Data Science Methods

Nicole L. Thompson

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Chemical Engineering

University of Washington

2018

Committee:

Vincent C. Holmberg, Chair

David A. Beck

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

Total Differential Capacity Plot Analysis Using Data Science Methods

Nicole L. Thompson

Chair of the Supervisory Committee:
Assistant Professor Vincent C. Holmberg
Department of Chemical Engineering

Differential capacity plots can be a powerful tool for uncovering battery performance characteristics buried within large datasets of charge-discharge curves. Due to the difficulty in analyzing these datasets in their entirety, arbitrarily chosen subsets of cycles are typically reported in the literature and used to draw qualitative conclusions describing the electrochemical changes that drive the peak shifts. Herein, open-source software we developed to quantitatively analyze entire cycling datasets is discussed. Peak features, such as peak locations and areas, are extracted by individually fitting each of the differential capacity plots. We implemented a database that allows users to perform this analysis, save model fits, and return to their data at a later point. Further, we demonstrate the ability to differentiate between two battery chemistries using peak characteristics and a support vector classifier, with an accuracy of 77%. This work provides the framework for more in-depth, quantitative analyses of differential capacity data.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Vincent Holmberg, for his support and guidance through this project and my entire time in graduate school. Without his support, this thesis would not have been possible. I am also extremely thankful to my group members, specifically Grant Williamson, Elena Pandres, Soohyung Lee, Sabiha Rousidan, and Brittany Bishop for their encouragement, suggestions, and assistance throughout my work. Also, I appreciate the amount of time, effort, and care my colleagues Theodore Cohen, Sarah Alamdari and Robert Masse dedicated to this project during the initial stages.

I also extend my sincere gratitude to Dave Beck, for giving me valuable feedback on my thesis, for serving as my committee member, and for being an irreplaceable data science teacher and mentor. Without his willingness to teach, guide, and encourage, this project would have never become a reality.

Also, I am thankful for the numerous friends, colleagues, and dogs who supported me throughout my time in graduate school. Special thanks to Caitlin Parke, for her advice, humor, and most of all, her immense kindness to me throughout graduate school. Thanks to Sabiha Rousidan who I spent many late nights writing with, for her motivational attitude and for being a truly amazing, strong person and friend. Also, to my incredible fiancé Earl Lara for his constant love and support in everything I do. Last but not least, I would like to thank my entire family, specifically my parents, for their support, encouragement, and love through my entire life.

Chapter 1. INTRODUCTION & BACKGROUND

1.1 Differential Capacity Analysis

Differential capacity plots are an extremely useful tool for the analysis of battery charge storage and degradation mechanisms. In order to examine long-term performance, researchers will often cycle battery electrodes thousands of times to simulate the real-world use case and even small-scale research testing typically includes tens to hundreds of cycles. In these experiments, a current-voltage data point is typically generated once every ten seconds, for weeks or even months, resulting in massive datasets. These cycling experiments result in charge-discharge curves of voltage and capacity, with different slopes corresponding to various electrochemical events occurring at different voltages. However, these plots can be cumbersome to read or draw definitive conclusions from, because the different electrochemical events are signified in these plots by plateaus where the capacity changes a large amount for a small change in voltage. If multiple plateaus from different electrochemical events overlap, electrochemical events can be easily missed. For this reason, researchers studying degradation or charge storage mechanisms in electrochemical cells often use differential capacity analysis.[1]-[6] In total differential capacity plots (also referred to as dQ/dV plots), the plateaus that were apparent in the original charge-discharge curves become peaks which correlate with the various electrochemical events.[6]

The reactions that correspond to each peak can be discovered by performing various characterization techniques, including *in situ* X-ray diffraction (XRD), Raman, and nuclear magnetic resonance (NMR) spectroscopy.[7] Differential capacity plots are often used to elucidate battery degradation mechanisms, state of health, and/or specific sources of capacity fade, because

the specific chemical reactions can be correlated to the capacity and voltage at which each event occurs.[6], [8]-[10]

1.2 Limits of Differential Capacity Plot Analysis

While differential capacity analysis is a powerful tool for understanding the underlying kinetics and thermodynamics of a system,[1] it is often underutilized by experimentalists mainly due to the vast quantity of data generated by cycling experiments. Cycling data for anything more than a few cycles becomes difficult to analyze by manual plotting due to the time constraints associated with the amount of data that must be analyzed. Therefore, researchers typically either reduce the number of cycles that they analyze, resort to using purely qualitative claims, such as discussing peak shifts and changes over cycles, or both.

The typical strategy for analyzing differential capacity plots involves selecting a subset of cycles to represent the data as a whole. For example, a researcher may choose to report and analyze cycles 1, 2, 5, and 10 for an electrode that underwent 100 total charge/discharge cycles. This set of cycles captures the solid electrolyte interphase layer formation and “normal cycling”. However, this method draws conclusions on an incomplete dataset, and it also prevents researchers from completely sharing their data with other investigators, which could be hindering the field as a whole.

There exist no widely accepted standards on how to choose which subsets of cycles should be used or reported, or even on how the final differential capacity plots reported in literature should be obtained. Smoothing the data, for example, is a common practice to distinguish features in differential capacity plots from noise (this is also done with differential voltage plots) and is often done by obscure procedures in the battery cycling instrumentation itself. However, there is no

standard method utilized by researchers, so methods range from using a five-point moving average,[9] to simply removing data,[5] to statistical based modeling to smooth the data.[4] The lack of standards for analysis and data reporting naturally raises questions on how any reported differential capacity data in literature was obtained. An automated, widely used tool could address some of these concerns.

One other problem with differential capacity analysis methods that slows the progress of the battery degradation field as a whole is that researchers are forced to conduct extensive literature searches while attempting to assign electrochemical events to peaks in their data. There does not yet exist an open, queryable database of differential capacity plots and their peak assignments. A database such as this would alleviate a tremendous amount of time, effort, and money that researchers spend either combing through literature to find peak assignments, or needlessly conducting experiments to identify a peak that was previously identified by another research group, unbeknownst to them.

1.3 Peak Deconvolution of Differential Capacity Plots

Some efforts have been made towards quantifying battery cycling data through curve fitting. For example, Weng *et al.* [6] processed charge cycling data by fitting the raw data with a third order polynomial in a piecewise manner with a moving window (similar to the Savitzky-Golay filter). The derivative of the middle of each window was then recorded as the differential capacity at that voltage and the differential capacity curve was smoothed by averaging. Following this data processing, Wang *et al.* fit the data with a fifth order polynomial, as there were two peaks in the differential capacity plots, and then used a support vector regression to fit the data to inform battery management systems. However, while this method results in good line fits, it provides little to no

information on the relationship between multiple electrochemical events occurring within a cell, as peak deconvolution would.

Christophersen *et al.*[4] fit voltage profiles with Gaussian distributions, with the intent of obtaining a smoothed differential capacity plot when the model was differentiated. In addition to using Gaussian fitting as a smoothing method, peak deconvolution of differential capacity plots can be used to elucidate the electrochemical reactions occurring within a cell. Calculating the areas of deconvolved peaks associated with various reactions and comparing to the total area of the dQ/dV curve gives insight into which reactions are responsible for what fractions of the total capacity of the cell. Aihara *et al.*[3] deconvoluted differential capacity plots using Gaussian peak fitting, and the ratio of the two peak areas was used to inform which reactions were occurring in the cell. However, this type of analysis was only done on a subset of cycles, limiting the amount of information gathered via this method.

Torai *et al.*[2] successfully fit peaks in charge cycles of lithium iron phosphate ($LiFePO_4$) differential capacity plots using a Pseudo-Voigt distribution function, with the purpose of estimating state-of-health (SOH). In this paper, the Pseudo-Voigt function was deformed to introduce an asymmetry parameter and an area parameter, which the authors used to represent the reversibility of the reaction and the total capacity in the phase transition, respectively. The change of the fit parameters with cycling was found to correlate well with the generally accepted notion that capacity fade in $LiFePO_4$/graphite batteries is due to the loss of activity in the graphite and the loss of cyclable Li^+ by a side reaction. The SOH estimates based on the developed fit were very close to the SOH values determined from the experimental data.[2]

Despite many researchers using curve fitting and peak deconvolution as a method to draw conclusions from experimental battery cycling data,[1]-[4], [6] there does not yet exist a standardized

tool for researchers to use that automatically deconvolutes peaks in every cycle of differential capacity data. This means researchers conduct this type of analysis on only one or a few cycles in the dataset, limiting the amount of information obtained.

In the presented work, these drawbacks associated with differential capacity analysis are addressed via a software tool which takes raw cycling data, calculates the differential capacity, cleans and smooths the dQ/dV plots, and performs automatic peak locating and deconvolution. This type of software could allow researchers to draw more definitive, quantitative conclusions from their cycling data. Further, the tool uses a queryable database, setting the stage for more extensive data sharing within the electrochemistry community.

Chapter 2. METHODS

2.1 PYTHON PACKAGES USED

Python was the main coding language used, with a multitude of packages for various parts of the software. Of them, the most important Python packages are listed and discussed here. A complete list of dependencies of the software are found in the *requirements.txt* file in the package, located on GitHub.[11]

The Python package PeakUtils[12] was paramount for peak detection in each set of data, with the indexes function – which returns the locations of peaks in a set of one-dimensional data – being particularly important. LMFIT,[13] a Python package for non-linear least-squares minimization and curve fitting, was used to generate and fit models based on the peak locations for each cycle. Additionally, scikit-learn[14] was used for the support vector machine (SVM) classification discussed further in Section 2.6. Dash was used for the visualization platform discussed further in Section 3.3. Dash[15] is a Python framework for building interactive web applications, combining

Plotly[16] for the plotting aspects, React[17] for the interactive elements, and Flask[18] for the web application components. Pandas,[19] a Python package for data structures and data analysis tools, was used repeatedly throughout the data processing and plotting steps.

2.2 Data Cleaning and Smoothing

In order to analyze peaks, the raw cycling data had to be converted into clean differential capacity curves, in a generalized manner that would be applicable not only to every cycle in a dataset, but also to every dataset that would be analyzed. The package's ability in this regard is discussed below.

Additionally, the program supports two data file formats: files collected from Arbin cyclers and files collected from MACCOR cyclers. The Arbin files are expected to be in Microsoft Excel file format, while the MACCOR files are expected to be in text file format in order to be loaded properly. The data is first loaded using either Pandas' function to read Excel files or Pandas' function to read comma separated values (CSV) files, depending on the data format specified by the user. Additionally, the two data formats have slightly different column headers, so those column variables are also specified and stored in a variable used throughout the data processing. In order to add more data formats, this function would simply be expanded to incorporate other file formats and their column headers, and the loading method would be specified for the new file format.

Once the raw data is loaded, differential capacity for the i^{th} row is calculated using Equation 2.1, saved in a new column and then this data is saved in the database, discussed further in Section 2.5.

$$\left(\frac{dQ}{dV}\right)_i = \frac{Q_i - Q_{i-1}}{V_i - V_{i-1}} \tag{2.1}$$

Because each cycle within the entire dataset spans the same voltage range, it was necessary to separate the data into individual cycles before cleaning. This is done by utilizing the cycle number label the cycler collects for each data point and creating a dictionary of separate data frames corresponding to individual cycles, which is then passed to the cleaning portion of the program.

Then, the program iterates over the cycles and each is cleaned of repeated voltage data points to both reduce computational time and to avoid large jumps in the differential capacity profiles, as are seen in the raw data shown in Figure 2.1 (a-b). These jumps result from when the denominator of Equation 2.1 is near zero, and is a common problem associated with plotting dQ/dV.[9] This step of the data cleaning was accomplished by creating a column in the data where the voltage was rounded to the third decimal point, then deleting rows which had duplicated rounded voltages. Rows with undefined values (NaN) for differential capacity were also removed from the dataset. Then the differential capacity at each point was recalculated using Equation 2.1.

The ability to remove certain user-specified voltage ranges of the data has been explored but is not finalized in the application. The user may be interested in focusing on specific peaks or removing large jumps in the differential capacity profiles near either end of the voltage range that was bypassed by the automatic data cleaning. Difficulties associated with implementation of this feature are discussed in Chapter 5.

After the data was cleaned of data points that would hinder the performance of the peak-finding code, there was still some noise in the differential capacity data, as shown in Figure 2.1 (c-d). In order to accurately identify real peaks instead of noise, the data for each cycle was smoothed using a Savitzky-Golay filter, which is a moving polynomial of specified order fit over a specified number of data points, called the window length. These parameters could be altered to become

user-specified variables, but for now, the window length is set to nine data points and a third order polynomial is used. This appeared to be the best combination of window length and polynomial that worked for the majority of data explored. The result of this smoothing was stored in a column of the cleaned data. The cleaned, smoothed cycles were saved individually in the database and then recombined, yielding the final processed dataset which was also added to the database. This cleaning procedure is summarized visually in Figure 2.1, with (a-b) depicting the raw data, (c-d) depicting the data after cleaning, and (e-f) depicting the smoothed data after the application of the Savitzky-Golay filter.

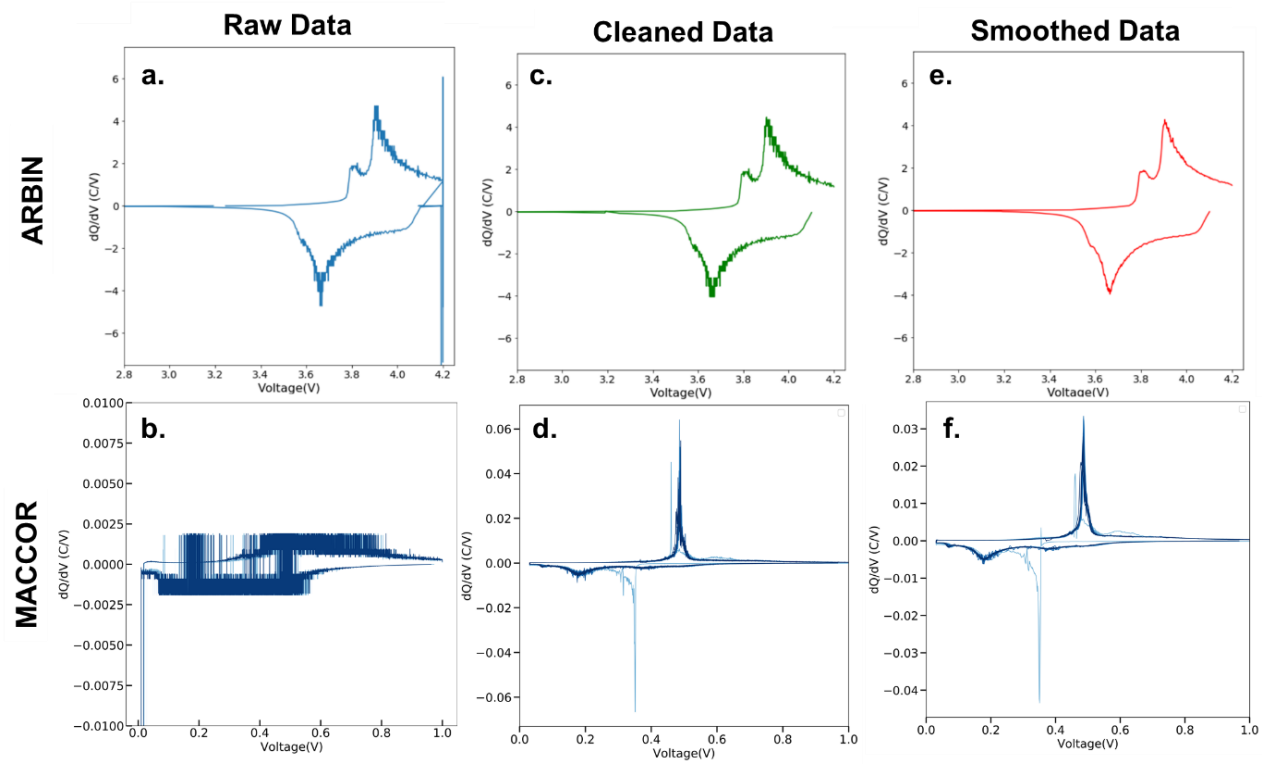

Figure 2.1. Data cleaning procedure for cycling data collected via both Arbin (a, c, e) and MACCOR (b, d, f) cyclers showing: (a-b) the raw cycling data converted to differential capacity plots without any cleaning, (c-d) the clean data obtained by eliminating where dV is close to zero, and (e-f) the smooth data obtained via the Savitzky-Golay filter.

2.3 Peak Tracking

Peaks corresponding to various electrochemical events in the differential capacity plots were identified and tracked using the PeakUtils Python package. The smoothed differential capacity data was used as a one-dimensional dataset, and then a user-specified peak threshold variable was used as an argument to find the indices of the peaks in the dataset. The peak threshold can be specified through the app, but the default value of 0.7 is used for the initial processing of new data until the user specifies a new value and reprocesses the data. The peak threshold indicates that only peaks with an amplitude greater than the normalized peak threshold will be detected. Thus, if the peak threshold is set to 0.7, and the maximum amplitude of the data is 10 A·h/V, only peaks with an amplitude greater than 7 A·h/V will be detected. Another handle on the peak detection is the minimum distance between the detected peaks. The value of this function argument that seemed to work best was the length of the cycle with the most data points, divided by 50.

One interesting problem associated with peak tracking was associated with the case when peaks either appeared or disappeared after multiple charge/discharge cycles. This caused an issue with peak tracking, as peaks were systematically labeled based on the order they were identified in the data. For example, if a peak was found around 4 V and called “Peak 1” for the first 9 cycles, and then a second peak appeared around 2 V, the peak around 2 V would then be labeled “Peak 1” and the peak around 4 V would be labeled “Peak 2” for that cycle (see Figure 2.2). This led to difficulties in accurately displaying the peak tracking data. In order to remedy this mislabeling, all peak locations were first gathered for every cycle, and then were rearranged on the basis of which peak locations were closest to one another.

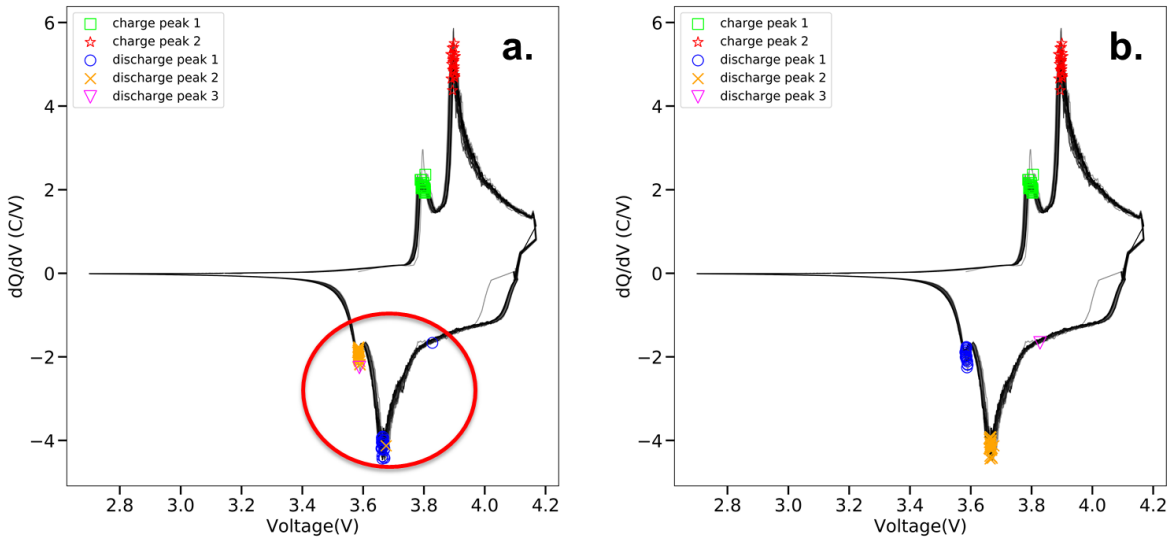

Figure 2.2. Example dataset depicting the (a) mislabeling of peak assignments and (b) the result of automated correction. The red circle in (a) highlights the location of the mislabeling, where discharge peak 1 was errantly labeled as “discharge peak 2” and discharge peak 2 was errantly labeled as “discharge peak 3”.

2.4 Model Fitting

In order to track peak areas, each cycle underwent a model fitting process which utilized the peak locations identified with the procedure discussed in the previous section. If there were no peaks found in a certain cycle number, a Gaussian model was fit to the data. Else, a Gaussian model was added to Pseudo-Voigt distributions set at each peak location. The purpose of the Gaussian is to act as a baseline, capturing the width of the differential capacity profiles and the area that is not clearly due to one peak. In electrochemistry, the Gaussian baseline is expected to loosely represent the solid-solution transitions, whereas the peaks are expected to represent the two-phase transitions in the electrode. The Pseudo-Voigt distribution was identified as a highly generalizable function able to fit the large variety of peak shapes that occur in various differential capacity curves. Previously, a variation of this function has been used to fit XRD profiles[20] and, more recently, differential capacity data.[2] Unlike a Voigt distribution, which is the result of the

convolution of a Gaussian and a Lorentzian curve, the Pseudo-Voigt distribution is simply the linear combination of the two. This distribution is often used in fitting experimental spectral data,[21] and is presented below in Equations 2.2 and 2.3.

$$f(x; A, \mu, \sigma, \alpha) = \frac{(1-\alpha)A}{\sigma_g\sqrt{2\pi}} e^{[-(x-\mu)^2/2\,\sigma_g^2]} + \frac{\alpha A}{\pi}\left[\frac{\sigma}{(x-\mu)^2+\sigma^2}\right] \quad (2.2)$$

$$\sigma_g = \sigma/\sqrt{2\ln 2} \quad (2.3)$$

Equation 2.2 is a function of peak amplitude (A), centering position (μ), peak width (σ), and the fraction parameter (α), which determines the relative weight of the Lorentzian and Gaussian components.

For every cycle in the dataset, the LMFIT package was used to generate an initial model with Pseudo-Voigt distributions assigned at peak locations along with a Gaussian baseline. Then, this unfit model is optimized to create the best fit, individualized for every cycle, as shown in Figure 2.3.

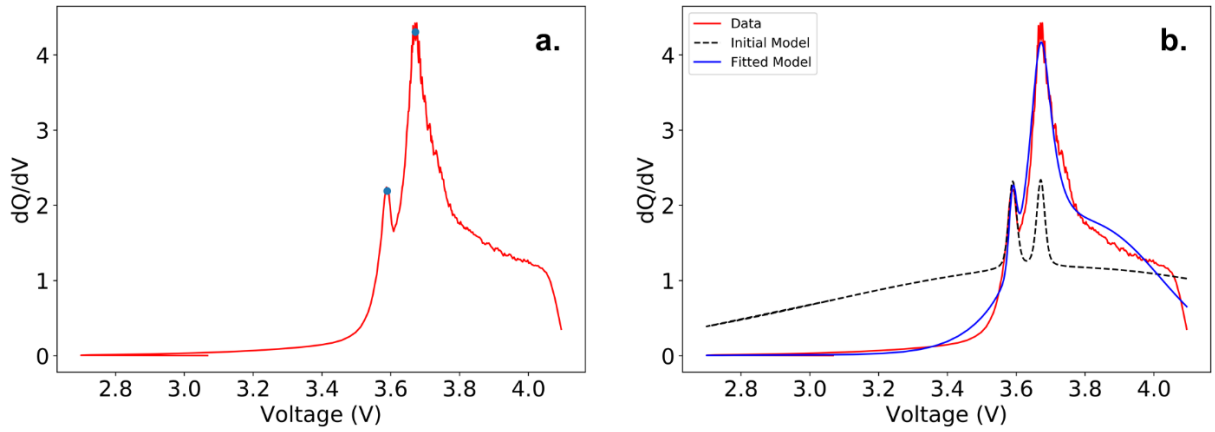

Figure 2.3. Example charge cycle depicting (a) peak locations found by PeakUtils, and (b) initial and final fitted model of a linear combination of one Gaussian and two Pseudo-Voigt distributions.

In the fitting of various differential capacity curves, all parameters are permitted to vary to achieve the optimized fit, except the centering position of the Pseudo-Voigt peaks which are determined from the peak-finding code discussed previously. The initial values for each part of the model were set as functions of the maximum, minimum, and/or average values of the data, such that the initial model was on the same scale as the data. This model fit can be evaluated visually by the user through the web application discussed in Section 3.3. Examples of model fits for various differential capacity plots are presented in Figure A.1 and Figure A.2 in the Appendix.

2.5 Database Backend

The data was saved in an SQLite[22] database multiple times throughout the processing procedure, including the raw data, the individual clean cycles, the full cleaned set, the model parameters for each cycle, and the descriptors for each cycle, including sorted peak locations, areas, and other peak parameters. This database also houses usernames and passwords, enabling the application to have login abilities. Implementation of a database was found to not only be an efficient method of storing the data but was also necessary for integration of the data processing with the application. Because of the database, users are able to upload their raw data, run it through the peak finding and processing code, and then visualize and plot their data on the same screen as the upload. Even further, users are able to process their data, and return to it at a later point, through the same application. This becomes extremely useful as the cleaning and model fitting process can be fairly time consuming for datasets with many cycles (100 cycles could take anywhere from 5 minutes to more than 20 minutes to process). Additionally, this concept of a web-based database to which anyone can contribute sets the stage for further collaboration and data-sharing within the electrochemistry community, which could help to drive the field forward as a whole.

2.6 CLASSIFICATION

The ability to classify battery chemistries based on their differential capacity curves was explored. This initial study was done using data from the Center for Advanced Life Cycle Engineering (CALCE)[23] for two different cathodes: lithium iron phosphate ($LiFePO_4$) and lithium cobalt oxide ($LiCoO_2$).

Each dataset underwent the cleaning, smoothing, and model fitting procedure described previously, and was labeled as either $LiFePO_4$ or $LiCoO_2$. For the purpose of classification, each dataset was separated into cycles and each cycle was treated as an individual instance of one of the two battery chemistries. A total of 866 individual cycles of $LiCoO_2$ data and 2060 cycles of $LiFePO_4$ data were obtained and processed to gather descriptors. If a cycle did not have a value for a descriptor, the value for that descriptor was set to zero. Then this complete, labeled set was split by a random 20-80 test-train split.

Classification between the two battery chemistries was done utilizing the descriptors gathered via the model fit for each cycle. The Least Absolute Shrinkage Selection Operator (LASSO) method was used for feature set reduction, to identify which descriptors generated via the model fitting were most important for classification. Once LASSO reduced the number of features required for classification, a SVM algorithm with a linear kernel was used to classify the data. This machine learning algorithm was trained on the training dataset, and then tested using the test portion of the dataset. Results from this classification are presented in Section 3.2.

Chapter 3. RESULTS AND DISCUSSION

3.1 PEAK TRACKING

3.1.1 *Location and Height Tracking*

The typical research article utilizing differential capacity analysis depicts their results using a scheme similar to Figure 3.1 (a), where cycles from a given subset are overlaid and trends are highlighted by supporting arrows. However, this qualitative display may be missing subtle trends in the changes in differential capacity peaks. Additionally, the subset of cycles chosen is completely arbitrary with no accepted standard of which cycles to report within the discipline. A quantitative analysis addresses these weaknesses associated with differential capacity plot analysis. By automating the process of identifying peaks and peak trends, the ability to analyze every cycle in a dataset becomes feasible. This allows researchers to provide quantitative data supporting claims of how peaks (and their associated electrochemical events) are growing in, fading out, or shifting to higher or lower voltages over all cycles. These types of quantitative data will support a better understanding of the drivers for degradation in a given cell.

Using this tool, peak locations and heights in the differential capacity plots of various sets of cycling data were able to be identified and tracked successfully; an example is shown in Figure 3.1 (b-c).

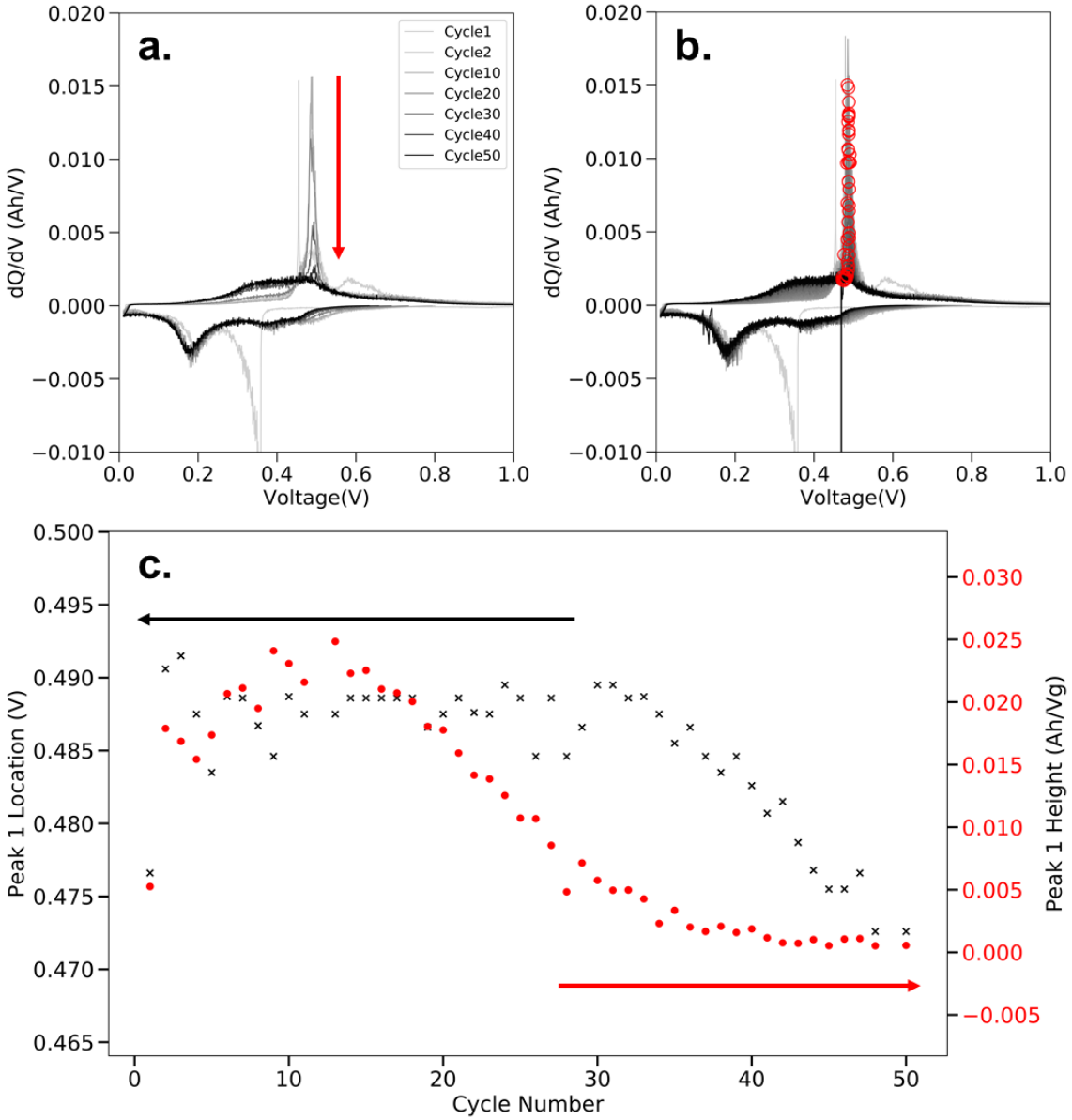

Figure 3.1. Example dataset showing the peak location and height tracking abilities of the tool. (a) The typical strategy for reporting differential capacity plot analysis, with an arbitrary subset of cycles and supporting arrows to show peak shifts. (b) The peak location and height found by the tool. (c) Peak location and height vs. cycle number as found by the tool.

3.1.2 *Peak Area Tracking*

In addition to the ability to track peak locations and heights, the model fitting function of the program allows the user to track the areas of each peak. This is important because the peak areas have implications related to the amount of charge being exchanged in the electrochemical event associated with the peak in the differential capacity plot. While the model could be improved to more accurately portray peak areas, initial results indicate the feasibility of using this tool to track peak areas, as shown in Figure 3.2. Further, the ratio of these peak areas can be tracked over cycles, which could be useful for electrochemists in analyzing where charge is or is not being exchanged efficiently. In the example shown, cycling data for a graphite cell is processed, with an example model fit shown in Figure 3.2 (a) and peak locations shown in (b). From this model, peak areas were obtained by integrating the respectful Pseudo-Voigt distributions, and plotted for every cycle number, as shown in (c). While this method could be improved upon, the data shows a clear trend in peak areas for peak one and peak two, located around 0.15 V and 0.18 V, respectfully. Additionally, the area ratios between peak one and peak two were examined, and again a clear downward trend was found in early cycles, followed by a leveling-out at a ratio of around two-to-one after forty cycles.

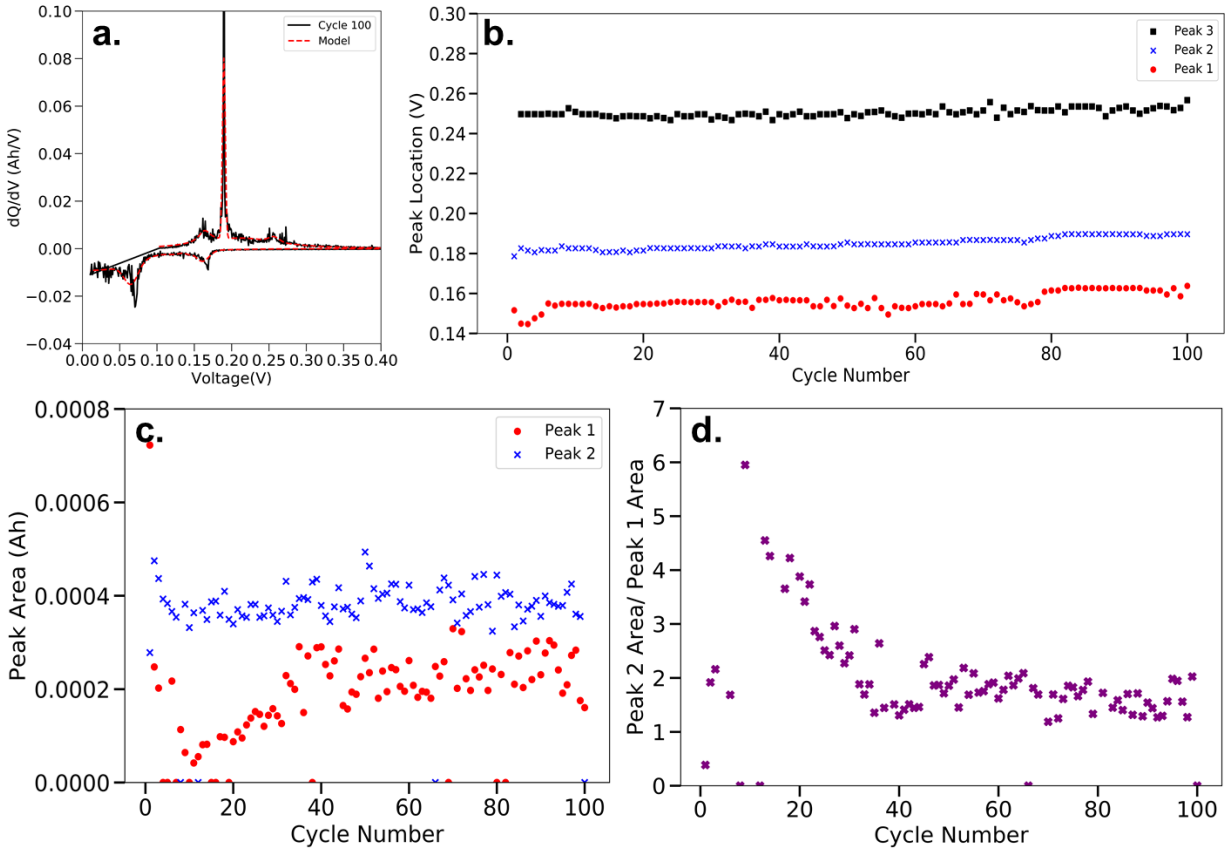

Figure 3.2. Example dataset showing the differential capacity modeling and peak area tracking abilities of the tool. (a) Model of one cycle in the data set demonstrating the fit with the experimental data. (b) The peak location for the three main peaks as found by the tool, demonstrating peaks are consistently found in accurate locations. (c) Areas of the peak located around 0.15 V and the peak located around 0.18 V found by the tool for every cycle number, and (d) the area ratio of those two peaks for every cycle number.

3.2 Classification using SVM

It was desired to examine the ability of a machine learning algorithm to accurately classify battery chemistry by the differential capacity plot. To this end, a simple classification problem was examined, with the goal to distinguish between $LiFePO_4$ and $LiCoO_2$ with data obtained from the CALCE website.[23] Example total differential capacity curves of each cathode chemistry are

shown below in Figure 3.3. The machine learning algorithm was implemented utilizing the general method discussed in Section 2.6.

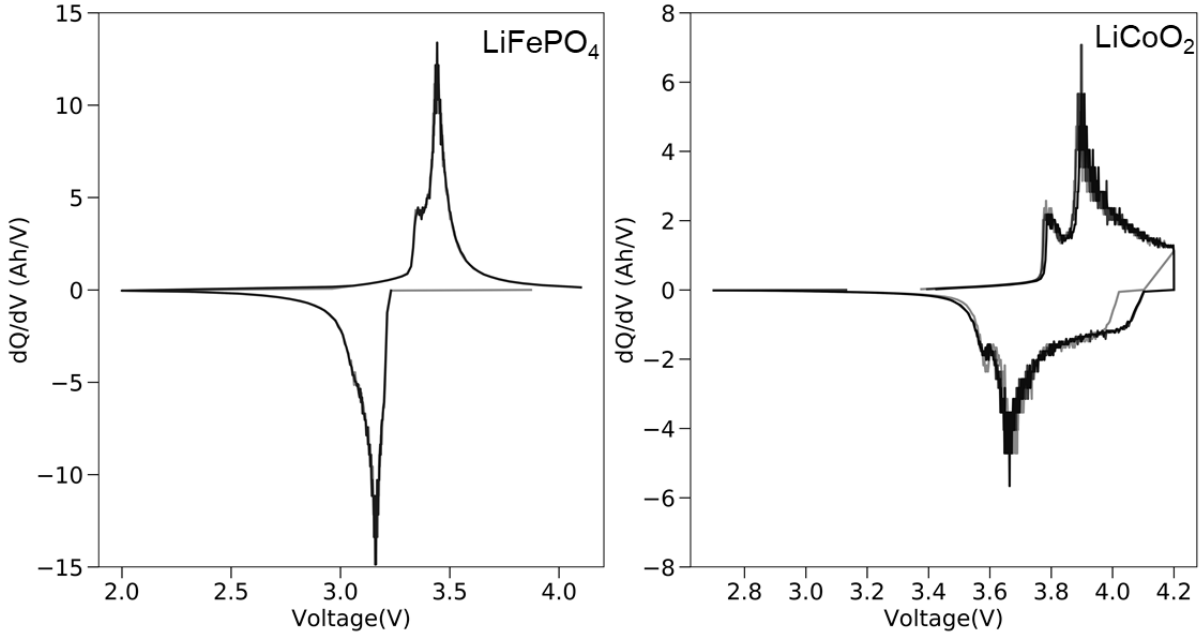

Figure 3.3. Example differential capacity plot profiles of the two cathode chemistries classified with the SVM algorithm, (a) lithium iron phosphate, and (b) lithium cobalt oxide.

Initial results are indicative that the classification of battery chemistry based on peak descriptors is promising. Once the cycles were processed, a total of 142 descriptors were obtained including peak areas, locations, widths, and heights along with the number of peaks for both the charge and discharge portions of the cycle. Initially, LASSO was utilized to reduce the total feature set to only three features, all of which were descriptive of peak heights. The final feature set was comprised of the first discharge peak height, the second charge peak height, and the fourth discharge peak height. The values of these descriptors for the two battery chemistries examined are depicted in Figure 3.4. Using this feature set and a linear SVM model, a 77% model accuracy was obtained. Test set error was 23% and the training set error was 20%.

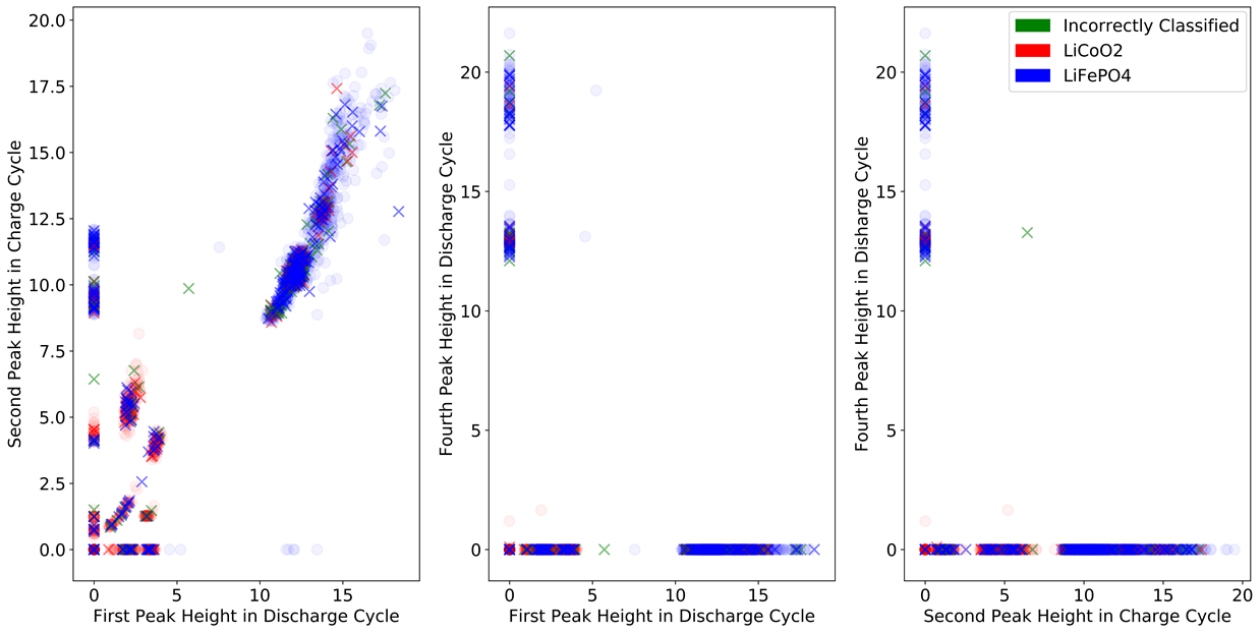

Figure 3.4. Differential capacity plot descriptors chosen by LASSO to classify between $LiFePO_4$ and $LiCoO_2$. The three descriptors chosen were all peak heights (Ah/V), including that of the first discharge peak, the second charge peak, and the fourth discharge peak. The "x" markers represent the test set and the circle markers represent the train set. Classification accuracy with a SVM algorithm was 77%.

While peak heights seem to generate a reliable feature set for classifying battery chemistry, it was desired to explore other features as well, since intuitively peak heights may change depending on the scale of the axis and a number of other experimental factors. For this reason, the peak heights were removed from the feature set, LASSO was performed again, and a new SVM model was built.

With the peak heights removed, the features LASSO selected were all peak locations, including that of the first, second, third, and fifth charge peak, and the location of the second discharge peak. Using these features, an SVM model with a test set error of 26% and a training set

error of 23% was obtained, indicating that peak locations are also viable descriptors for classification between two battery chemistries.

Interestingly, the number of peaks in the charge or discharge portion of the cycling profile was not identified as an important feature by LASSO. This is likely because the program found similar numbers of peaks for the two battery chemistries, or at least did not identify a reliable disparity in the number of peaks between the two battery chemistries. Additionally, it is important to note that the highest peak number chosen as a descriptor does not necessarily mean each differential capacity curve had that number of peaks, but the lack of a value for any given peak descriptor would have also been considered, as all undefined values were replaced with zeros prior to feature selection or model training.

3.3 Visualization and Accessibility

The package's processing, peak tracking, and model fitting is tied together with the Dash-based web application, which can either be run locally or accessed online at dqdvanalyzer.com. The application layout is depicted in Figure A.5 in the Appendix. The application uses Dash's basic authorization package, and thus requires a login username and password. These username-password pairs are stored in the database and can be modified at any time. Once logged in, users are able to browse previously uploaded data via a dropdown menu populated by the database's master table, where the username matches the user currently logged in. Alternatively, users may upload their own data by specifying the data collection mechanism (Arbin or MACCOR) and using Dash's file upload component. Once the data is uploaded, it automatically undergoes the data cleaning, processing, and saving to the database. This processing is done with an initial peak threshold of 0.7. Once the file is successfully processed, a message appears saying the file now

exists in the database, indicating the user may select it from the dropdown menu to explore and visualize.

The user can view individual cycles using a slider bar and the model fit for each cycle can be explored by selecting the “show model” toggle button. Below the figure displaying the raw data and model fit to the cleaned data, various descriptors over all the cycles can be plotted, including peak area, peak location, peak width and peak height. Additionally, the app provides a section to evaluate the Gaussian baseline fit aspect of the model, which is useful for determining if the correct number of peaks were located or if the Gaussian baseline is dominating an area which should be covered by Pseudo-Voigt peaks. In this section, the user has the ability to alter the peak threshold of just one sample cycle and see the effects on the model fit. If the new peak threshold seems to produce a more accurate model, then the user can update the model in the database for all the cycles in that dataset with the new peak threshold. Near the bottom of the web application page, users can see the data tables associated with the raw data, clean data, and cycle descriptors. The cycle descriptors data table can be downloaded as a comma-separated value (CSV) file for researchers to use as they see fit.

To run the application locally, the package can be downloaded from GitHub.[11] This application is also hosted using Google Compute Engine and can be accessed without any Python requirements at *dqdvanalyzer.com*.[24]

Chapter 4. CONCLUSIONS

A software tool for automated quantitative analysis of differential capacity plots has been developed and its applications have been demonstrated. Peaks in differential capacity plots can be identified and tracked over all the cycles comprising a dataset. This provides quantitative results

researchers can use to either replace or support qualitative claims regarding peak shifts and changes in total differential capacity plots as an electrode is cycled.

Additionally, a model comprised of a Gaussian function and multiple Pseudo-Voight functions located at identified peak locations can be fit to the data and used to determine peak areas for each cycle. These peak areas correspond to the amount of charge transferred during the reaction associated with the differential capacity plot feature. This quantitative method to track these peaks provides researchers with a tool to apply to their own data, to track where charge is being exchanged efficiently and which reactions in a cell may be performing inefficiently or are related to degradation.

The implemented database provides the framework to share differential capacity data amongst researchers in electrochemistry. This could reduce the need for researchers to conduct extensive literature reviews, and instead they could simply query the database to find a dataset of interest.

Even further, this tool provides an efficient way to collect cycle profile descriptors to use in classification of battery chemistries. Initial results indicate a classification accuracy between two battery chemistries of 77% using features selected by LASSO and an SVM classifier model. Despite being a trivial example, the ability to classify battery chemistry by differential capacity plots in the future could provide a method for researchers working with new electrode chemistries to identify if previously studied materials have had similar cycling profiles, assuming an open, shared database existed.

Chapter 5. FUTURE WORK

For this tool to be implemented practically and become a consistently useful tool for electrochemists, efforts must be made to enhance the user experience and the model's reliability.

The first task would be to look for pieces of the code that are especially slow and improve upon them. While the automated process is much quicker than examining the data manually, processing 100 cycles could take around 20 minutes, depending on the complexity of the model fits.

Also, the robustness of the data parsing upon upload could be improved such that differential capacity curve data in any format could be uploaded and processed. Currently, the code expects the files that are uploaded to be in a particular format, with particular column headers. These file formats are specific for data collected via Arbin or MACCOR instrumentation. The rigidity of the file upload process means that files that fall outside of these expectations will not be processed. The code could be rewritten to determine which file type is being uploaded based on the file extension (.csv, .txt, .xlsx), and it may also be possible to edit the code to determine the relevant column headers, without being given the datatype (for example, if a column header contains "voltage" or "[V]", use this as the voltage column). Further, in a few cases, a file contained non-ASCII characters, which prevented the contents from being parsed until the problem characters were manually removed from the file. This could likely be solved with a setting to ignore any non-ASCII characters upon upload. With methods like those proposed above, it would likely become possible to upload and process files that are collected via a different battery cycler.

Some further user input for the model building section of the code could also prove useful. While users at this point are able to adjust the threshold at which peaks are identified, it could be beneficial to allow users to adjust other parameters as well. The smoothing of the data seems to affect the model build, so being able to adjust the window length and polynomial order used by the Savitzky-Golay filter would be useful. Alternatively, different methods of smoothing could be explored, such as the method presented by Christophersen *et al.*[4] The model could further be

improved by utilizing methods used by other peak deconvolution software, such as QSoas, which is an open-source tool for model fitting.[25]

Additionally, in some cases peaks will be found where there was not a peak, but just noise that was remaining after the filter, leading to an overall better model fit but at the expense of the ability to track peak areas accurately. A user input for peak locations had been explored initially, but there have been challenges associated with allowing that peak location to vary over all cycles in the dataset, while still capturing the peak the user identified. However, if implemented, this sort of user feedback would be extremely valuable for determining the best model fits.

Further, the application layout and abilities could be expanded upon. One thing that is often desired is the ability to plot differential capacity normalized by the weight of active material. This is currently not supported by the application, but would be a simple modification, where the user would input the weight and the program would divide the ordinate of the differential capacity plots by that weight. Another feature that the app currently does not support is allowing users to create, modify, and download plots of their data. The substitute for this at the moment is the ability of users to download the processed data, including the peak descriptors, but it would also be useful to be able to create professional plots of the data in the application itself. One example would be to generate differential capacity plots depicting a user-chosen subset of cycles, as is traditionally reported in literature.

One other issue that must be addressed before opening the application to multiple users is upgrading the login and security. At this point, there is a set number of user and password combinations stored in the database, which cannot be edited or added to by users, leaving the code owner responsible for designating and managing login credentials. Additionally, the security of the application could be reconsidered to ensure users' data is completely protected. The current

code ensures users are only able to access their data by cross-referencing the login username with the upload username (added to a dataset when it is added into the database), and then populates the dropdown menu with the file names that pass that cross-reference. More security checks could be added to prevent users from seeing files that were not uploaded by them, such as performing that cross-reference multiple times through the data displaying code. There should also be checks to ensure usernames are unique. With these changes in place, this software has the potential to become a robust, standardized tool for performing differential capacity analysis on battery cycling data.

REFERENCES

[1] Marzocca, L. M., & Atwater, T. B. (n.d.). Differential Capacity-Based Modeling for In-Use Battery Diagnostics, Prognostics, and Quality Assurance, 4.

[2] Torai, S., Nakagomi, M., Yoshitake, S., Yamaguchi, S., & Oyama, N. (2016). State-of-health estimation of LiFePO4/graphite batteries based on a model using differential capacity. *J. Power Sources*, *306*, 62–69. https://doi.org/10.1016/j.jpowsour.2015.11.070

[3] Aihara, Y., Ito, S., Omoda, R., Yamada, T., Fujiki, S., Watanabe, T., Park, Y., Doo, S. (2016). The Electrochemical Characteristics and Applicability of an Amorphous Sulfide-Based Solid Ion Conductor for the Next-Generation Solid-State Lithium Secondary Batteries. *Frontiers in Energy Research, 4*. https://doi.org/10.3389/fenrg.2016.00018

[4] Christophersen, J. P., & Shaw, S. R. (2010). Using radial basis functions to approximate battery differential capacity and differential voltage. *J. Power Sources*, *195*(4), 1225–1234. https://doi.org/10.1016/j.jpowsour.2009.08.094

[5] Christophersen, J. P., Bloom, I., Thomas, E. V., Gering, K. L., Henriksen, G. L., , V. S., & Howell, D. (2006). *Advanced Technology Development Program for Lithium-Ion Batteries: Gen 2 Performance Evaluation Final Report* (No. INL/EXT-05-00913, 911596). https://doi.org/10.2172/911596

[6] Weng, C., Cui, Y., Sun, J., & Peng, H. (2013). On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression. *J. Power Sources*, *235*, 36–44. https://doi.org/10.1016/j.jpowsour.2013.02.012

[7] Aurbach, D., Markovsky, B., Weissman, I., Levi, E., & Ein-Eli, Y. (1999). On the correlation between surface chemistry and performance of graphite negative electrodes for Li ion

batteries. *Electrochimica Acta*, *45*(1), 67–86. https://doi.org/10.1016/S0013-4686(99)00194-2

[8] Honkura, K., Takahashi, K., & Horiba, T. (2011). Capacity-fading prediction of lithium-ion batteries based on discharge curves analysis. *J. Power Sources*, *196*(23), 10141–10147. https://doi.org/10.1016/j.jpowsour.2011.08.020

[9] Bloom, I., Jansen, A. N., Abraham, D. P., Knuth, J., Jones, S. A., Battaglia, V. S., & Henriksen, G. L. (2005). Differential voltage analyses of high-power, lithium-ion cells. *J. Power Sources*, *139*(1–2), 295–303. https://doi.org/10.1016/j.jpowsour.2004.07.021

[10] Schmerling, M., Schwenzel, J., & Busse, M. (2018). Investigation of the degradation mechanisms of silicon thin film anodes for lithium-ion batteries. *Thin Solid Films*, *655*, 77–82. https://doi.org/10.1016/j.tsf.2018.03.037

[11] Thompson, N. (2018). Chachies. GitHub Repository. https://github.com/nicolet5/chachies

[12] PeakUtils — PeakUtils 1.3.0 documentation. (n.d.). Retrieved December 4, 2018, from https://peakutils.readthedocs.io/en/latest/

[13] Non-Linear Least-Squares Minimization and Curve-Fitting for Python — Non-Linear Least-Squares Minimization and Curve-Fitting for Python. (n.d.). Retrieved December 4, 2018, from https://lmfit.github.io/lmfit-py/

[14] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *J. Machine Learning Research*, *12*(Oct), 2825–2830.

[15] Dash by Plotly - Plotly. (n.d.). Retrieved December 4, 2018, from https://plot.ly/products/dash/

[16] Plotly Technologies Inc. (2015). Collaborative Data Science. Retrieved December 4, 2018, from https://plot.ly/python/

[17] React – A JavaScript library for building user interfaces. (n.d.). Retrieved December 4, 2018, from https://reactjs.org/index.html

[18] Flask (A Python Microframework). (n.d.). Retrieved December 4, 2018, from http://flask.pocoo.org/

[19] McKinney, W. (2010). Data Structures for Statistical Computing in Python (pp. 51–56). Presented at the Proceedings of the 9th Python in Science Conference. Retrieved from http://conference.scipy.org/proceedings/scipy2010/mckinney.html

[20] Sánchez-Bajo, F., & Cumbrera, F. L. (1997). The Use of the Pseudo-Voigt Function in the Variance Method of X-ray Line-Broadening Analysis. *J. Appl. Crystallogr.*, *30*(4), 427–430. https://doi.org/10.1107/S0021889896015464

[21] Wertheim, G. K., Butler, M. A., West, K. W., & Buchanan, D. N. E. (1974). Determination of the Gaussian and Lorentzian content of experimental line shapes. *Rev. Sci. Instrum.*, *45*(11), 1369–1371. https://doi.org/10.1063/1.1686503

[22] SQLite Home Page. (n.d.). Retrieved November 29, 2018, from https://www.sqlite.org/index.html

[23] CALCE Battery Group. (n.d.). Retrieved November 29, 2018, from https://web.calce.umd.edu/batteries/data.htm

[24] Quantitative dQ/dV Analysis and Visualization. (n.d.). Retrieved November 29, 2018, from http://dqdvanalyzer.com/

[25] Fourmond, V. (2016). QSoas: A Versatile Software for Data Analysis. *Anal. Chem.*, *88*(10), 5050–5052. https://doi.org/10.1021/acs.analchem.6b00224

APPENDIX

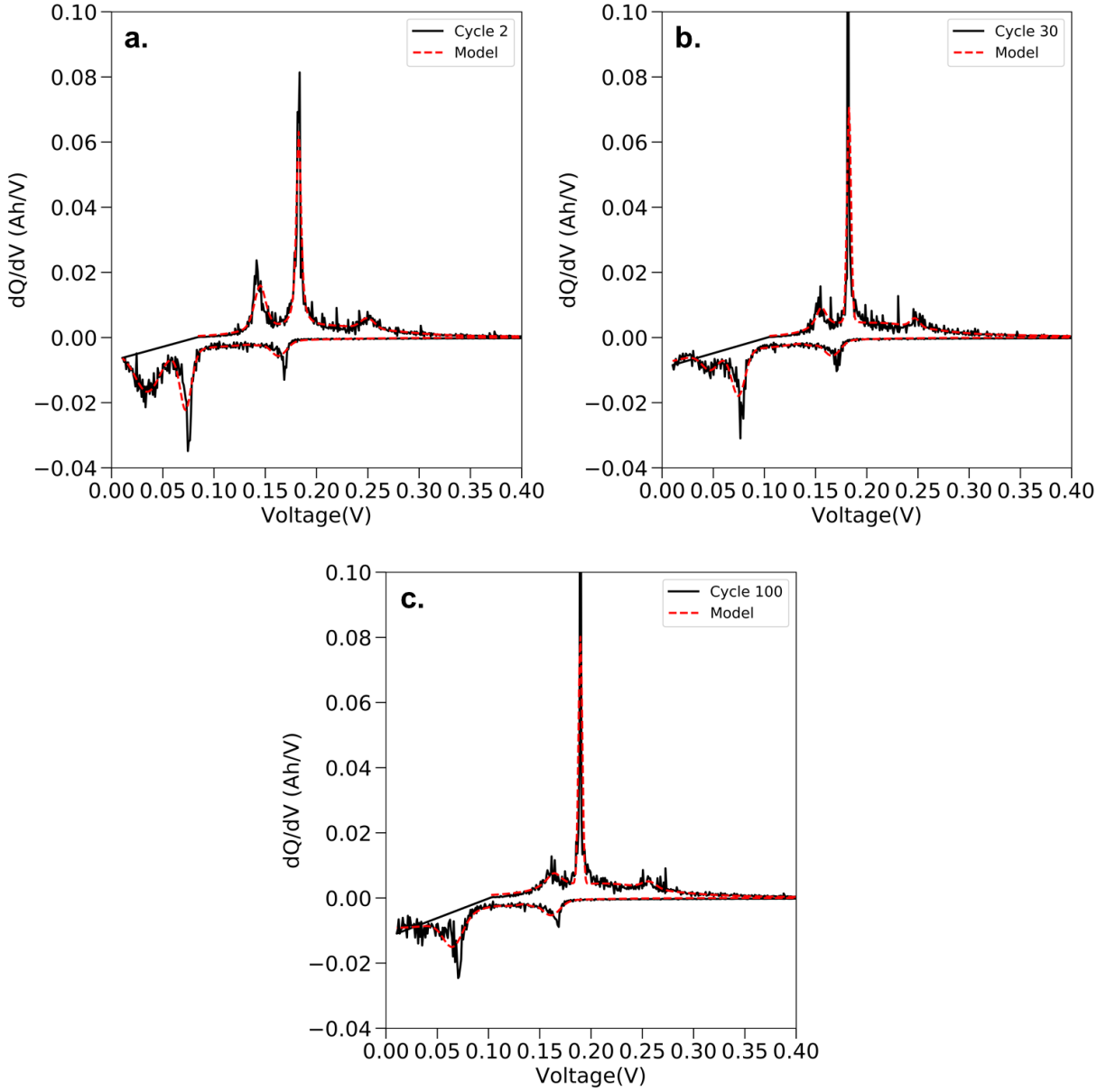

Figure A.1. Example model fits for one set of differential capacity plots, (a) cycle 2, (b) cycle 30, and (c) cycle 100.

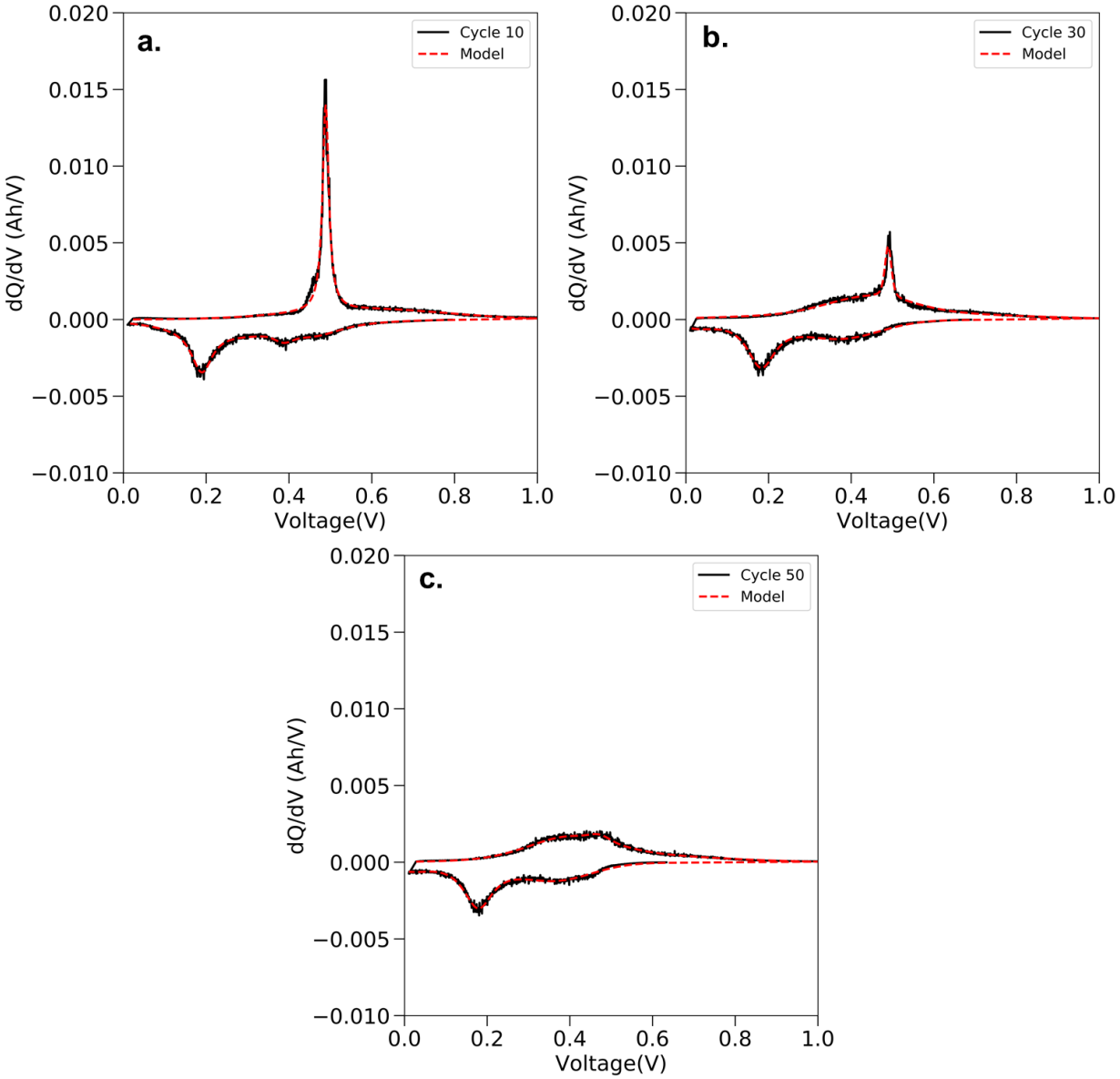

Figure A.2. Example model fits for one set of differential capacity plots, (a) cycle 10, (b) cycle 30, and (c) cycle 50.

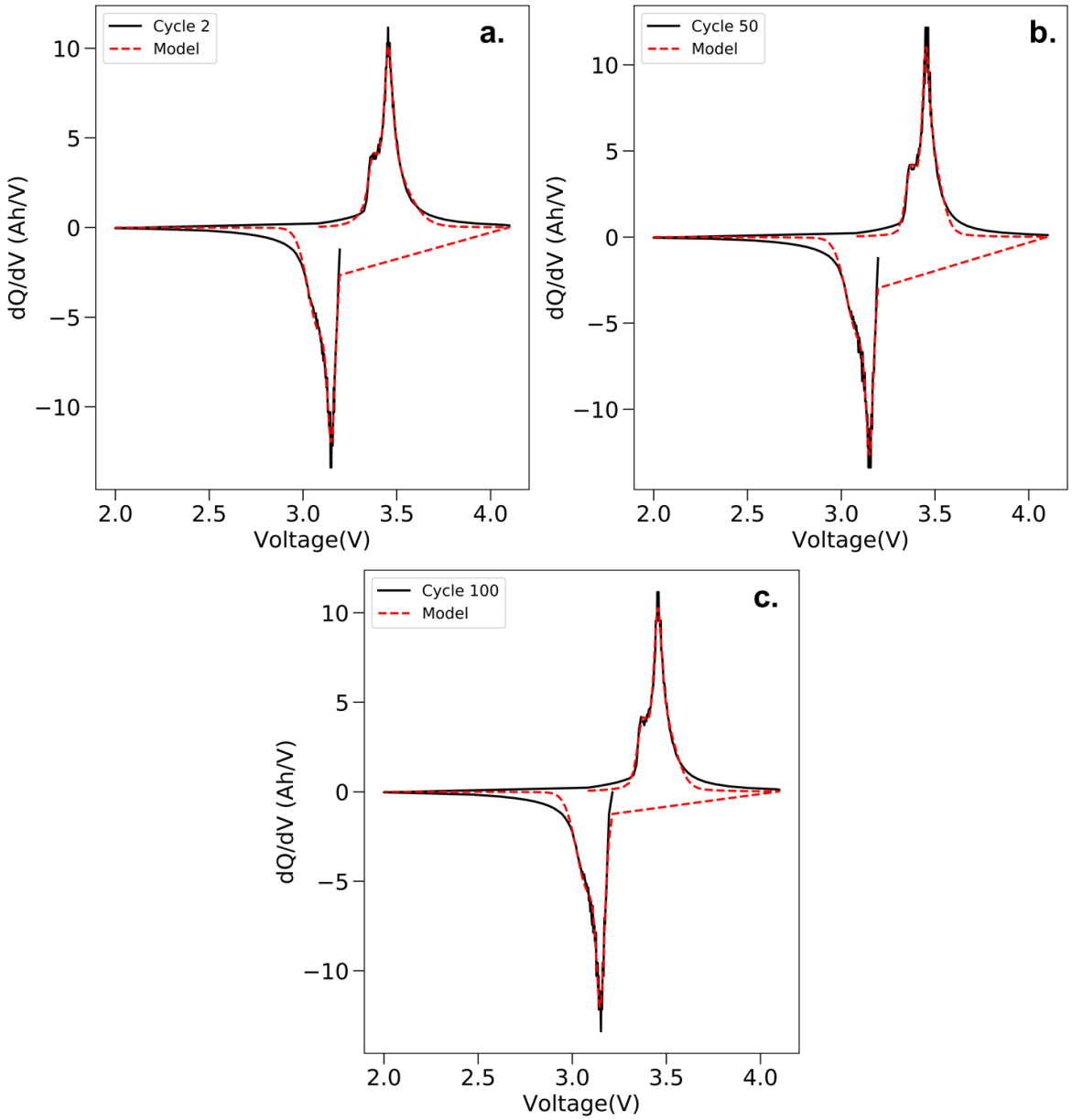

Figure A.3. Example model fits for one $LiFePO_4$ set of differential capacity plots, (a) cycle 2, (b) cycle 50, and (c) cycle 100.

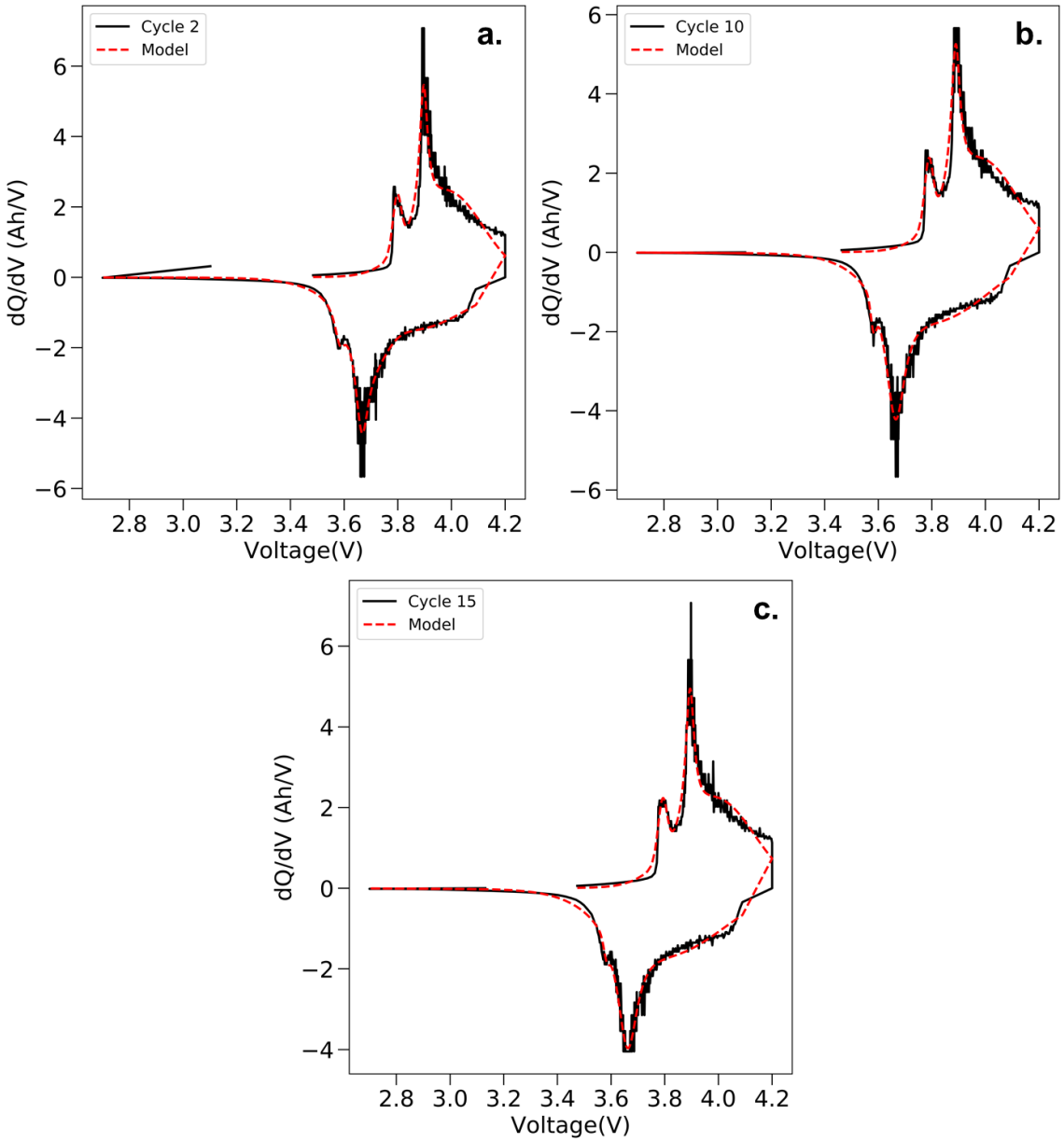

Figure A.4. Example model fits for one $LiCoO_2$ set of differential capacity plots, (a) cycle 2, (b) cycle 10, and (c) cycle 15.

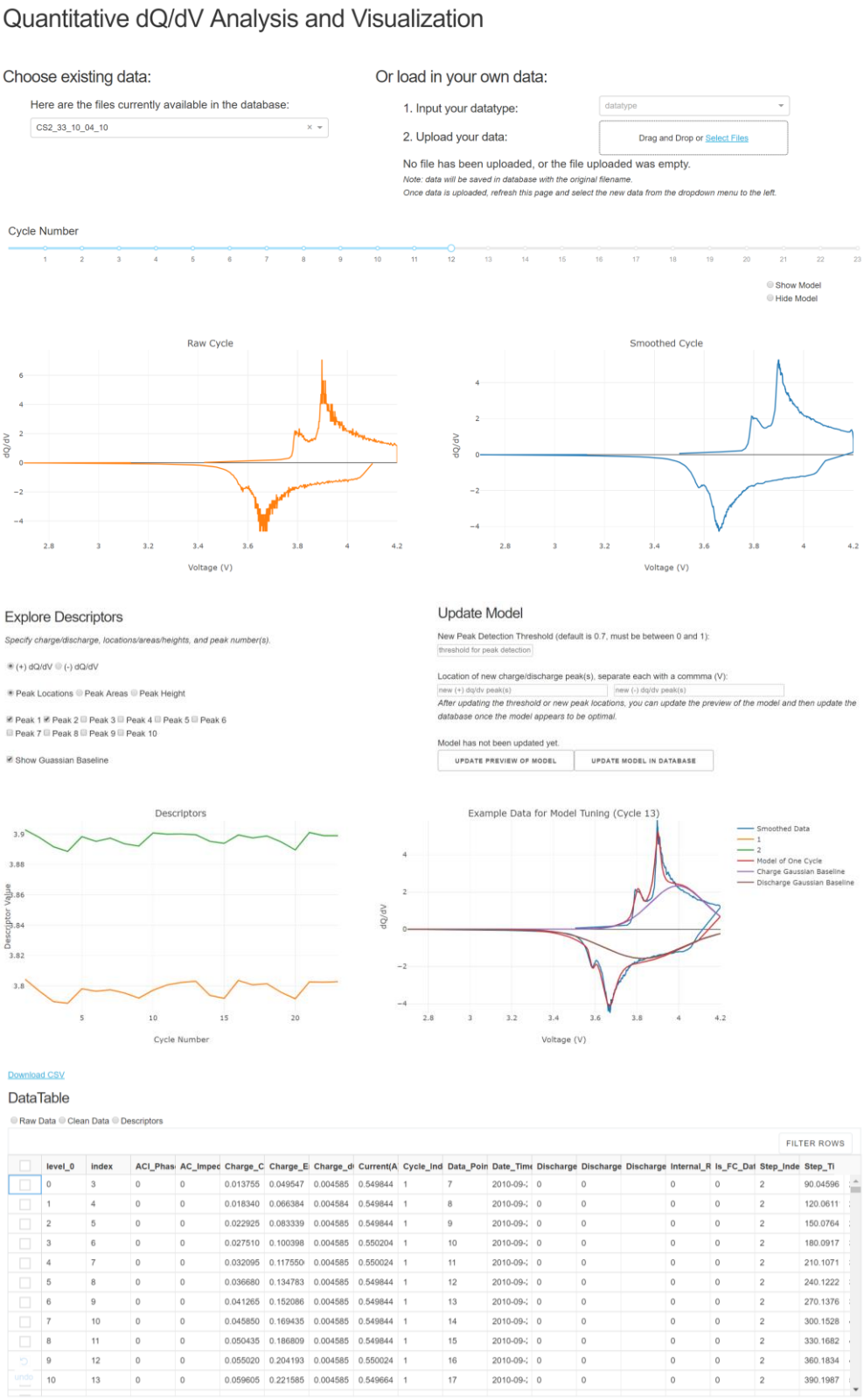

Figure A.5. Dash application layout, with an example CALCE dataset loaded and processed.