Cycling Analyzer and Plotting – Guidelines

Scope and summary

Cycling Analyzer and Plotting (CAP) is an *in-house* Python script made in such a way that can be used easily by experimentalists (***no coding knowledge needed***), and whose main objective is the **automatic data evaluation of cycling data** which were obtained either through Maccor, or EC-LAB. In particular, it analyzes one single file at a time associated with a cycle life experiment, a rate-capability test, or a combination of those (first rate-capability and then cycle life). Here a rate-capability test is defined as an experiment in which different currents are applied, each current for a given number of cycles, while a cycle life test is defined as a test in which the charge/discharge current is kept constant, and many cycles are performed to assess the cell cycle life. The code can also be applied to results whose cycling protocol is not finished yet.

In short, the users should export the data in a readable format (see “Data exporting” section) through Maccor or EC-LAB, fill an excel in which the information about the experiment should be reported (see “Code’s Inputs” section), and run an executable (see “Running & Results” section). These simple tasks will allow obtaining a series of results, including well-organized CSV files and images containing the voltage profiles, dQ/dV and dV/dQ curves, coulombic efficiency, and many other analyses.

Data Exporting

At first, data should be exported in a readable (.txt) format. For this, simple procedures exist in both Maccor and EC-LAB (Figure 1).

Maccor

For data exporting in Maccor, Maccor Export should be used. In particular, you should select all the properties selected by default plus “MD/State” (last selection on the bottom – Figure 1 A). Afterward, you should select the path in which the readable data should be exported (bottom right of Figure 1A), select a tab-delimited “Text Output” and that all cycles should be exported. Finally, you should select the original Maccor files you want to export (more than one file can be selected here) and click “Convert”. Remember to select the files to be exported as the last step; if not, Maccor export could bug. This will generate a .txt file which can be used as input for the CAP code.

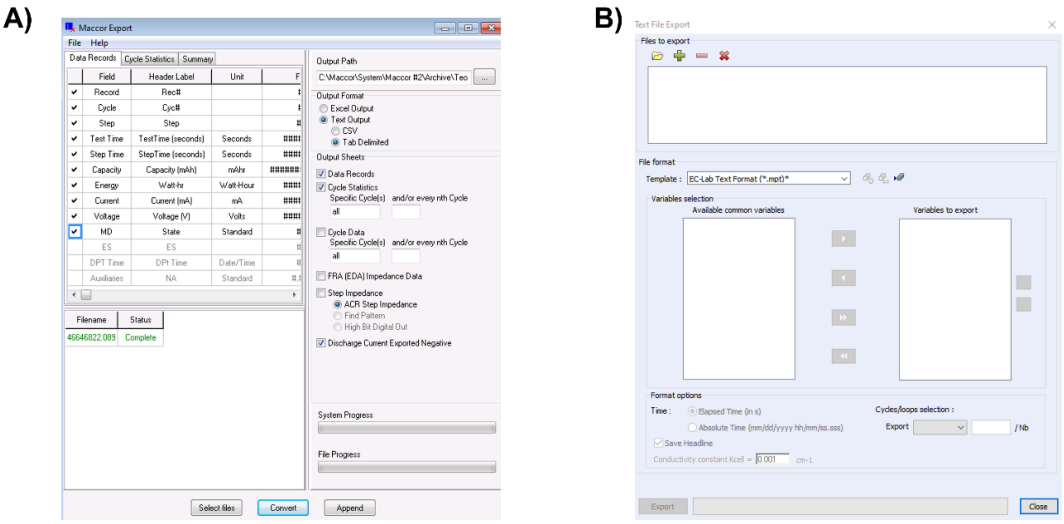

Figure 1. Procedure to export data as readable text format in **A)** Maccor, and **B)** EC-LAB.

EC-Lab

For data exporting in EC-Lab, a built-in function already exists. For using it, it is enough to go to "Export→Export as text". This will open the window showed in Figure 1B. Here it is enough to add the .mpr (native EC-Lab format) files that should be converted (more than one can be selected here) and click on export. This will generate (in the same location of each mpr file) an associated mpt file, a readable data format which can be used as input for the CAP code.

Code's Inputs

CAP needs two inputs: a readable text file associated to a cycling experiment (See Data Exporting section) and a devoted Excel file, which should be filled to report key information on the cycling experiment to be analyzed.

Concerning the Excel file, a schematic of the requested parameters (of which some are mandatory and some optional, as a function of the specific experiment you carried out) is reported in Figure 2. The different parameters can be divided in 5 different sections, associated to 5 different colors:

1) Software used and file location (Yellow).
2) Rate-capability test parameters (Salmon).
3) Cycle life test parameters (red).
4) Parameters used for the smoothing (light brown).
5) Normalization mass (brown).

Parameters' name	Parameters' value
Software	Maccor
Path	path
Filename	filename.txt
C-rate list	C_50, C_10, C_5, C_2, C
Cycles per formation	1
Cycles per C-rate	3
C-rates	C
Cycles number per C-rate	300
Cycle number to be predicted	100
Points per linear interpolation	100
Savgol filter - windows length	100
Savgol filter - polynomial order	2
Mass (mg)	5.55

Figure 2. Panorama of the different parameters to be reported in the Excel file associated with the CAP code.

In general, each parameter is associated to a devoted and complete explanation in the Excel file (Inputs.xlsx), which however is also summarized below.

Software used and file location: Here you should indicate if you used Maccor or EC-Lab to cycle your cell, the path to the folder containing the data file to be analyzed, and the full name of the latter (including the data extension).

Rate-capability test parameters: Here you should indicate the different C-rates you used for your testing, the number of cycles for the formation procedure (if any), and the number of cycles for the remaining C-rates tested.

The different C-rates you used should be separated by a comma, and they will be used to name the different voltage profiles, dQ/dV curves etc., including for the formation (considered to be the first C-rate indicated by default). Therefore, it is advised to use a unique name for each C-rate tested.

The “cycles per formation” indicates how many cycles have been used for the formation protocol, which is associated to the first C-rate indicated in the “C-rate list”. If no formation, please report 0 in here.

The “cycles per C-rate” indicates how many cycles were performed for each C-rate tested (except the formation, if any). The assumption here is that each C-rate is tested by using the same number of charge/discharge cycles.

Cycle life test parameters: Here you should indicate the C-rate you used for your cycle life test, the maximal number of cycles, and (in future) the number of cycles for capacity predictions (beyond the ones tested experimentally).

The name of the C-rate is used to name your results, so report the one fitting you the best.

The “Cycle number per C-rate” indicates the (maximal) number of cycles to be performed in the cycle life test.

The “Cycle number to be predicted” allows predicting the capacity evolution beyond the ones experimentally measured. However, the model for doing that is not ready yet, so please, if you try it, do not expect particularly trustable results.

Rate-capability and cycle life test parameters: Please report the information on your full protocol, even if the experiment did not finish yet. If this is the case, the code will automatically take into account that, and analyze all the results available except the last charge/discharge cycle.

Parameters used for the smoothing: Here you should indicate the parameters used for smoothing the voltage profiles before the dQ/dV and dV/dQ calculations. For this, the original initial profile is divided in n equally distanced regions, where n is defined by the “Points per linear interpolation”. Each of these points is connected through linear interpolation. Therefore, the dQ/dV and dV/dQ curves are calculated and smoothed through a Savgol filter. The latter requires two parameters: the windows length, and the polynomial order. Playing with those, you can get smoother or rough dQ/dV and dV/dQ curves.

Normalization mass: Here report the mass (in mg) to be used for normalizing capacity, if you want to do so (both not normalized and normalized results will be outputted)

Accounting for formation procedures using more than one C-rates

The CAP code was developed assuming a formation using a single C-rate (coming back in time I would change this, but that's how it stands now). However, you can circumvent this limitation by:

1) Combining the formation option (for the first C-rate) and the rate-capability part (for the subsequent – 2nd, 3rd, *etc.* - C-rates, but knowing that here it is assumed that all those C-rates use the same number of cycles) in case you did perform a cycle life test;
2) You can report the number of cycles per formation equal to the total cycles you did, but assuming they all use the same C-rate (you can use an effective name for this). This option has the disadvantage that the naming for the formation cycles will not be correct (you will be able to use one single naming for them all), but at least you can visualize and print in an easy-to-plot format the entirety of your results (remember that this code is for visualization + saving time purposes anyway, for the final plotting you can still give the name you want to the different cycles).

Running & Results

Once performed the steps above, it is enough to run the executable (CAP.exe) in the same folder in which the Excel with the inputs is located, and wait for the analysis to be performed (Figure 3). Please, remember to save the Excel with the updated inputs before to run the executable or the code. A small message will be printed when this is done, and the analysis time will change as a function of the number of data to be analyzed, but typically it is in the order of 30 seconds.

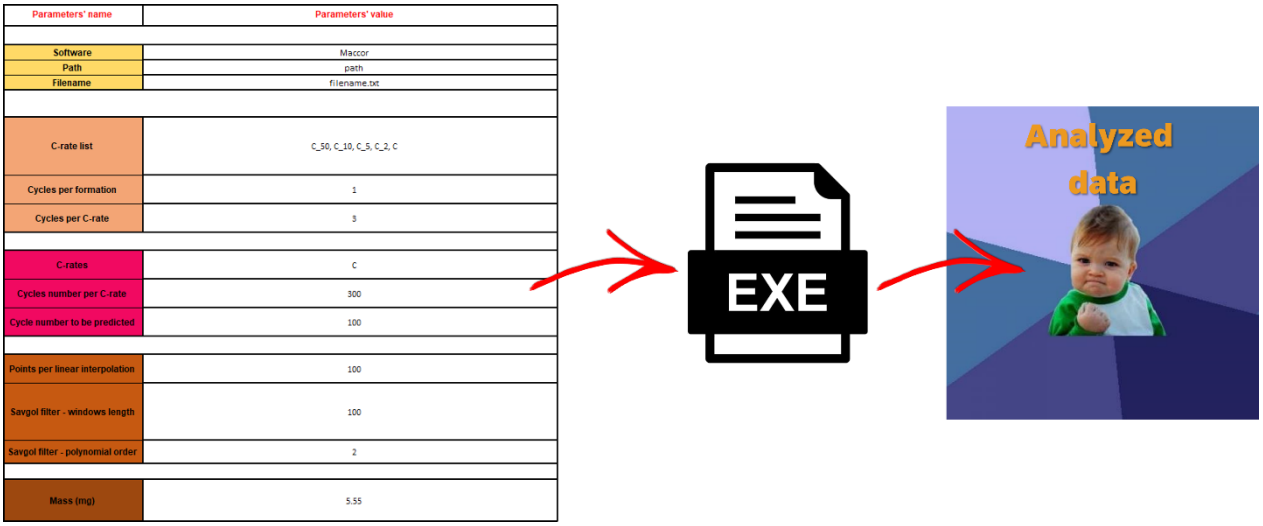

Figure 3. Schematics of the CAP running procedure.

CAP outputs several results, going from the most “classical” data analysis (Figure 4), like the capacity evolution as a function of the cycle number, Coulombic efficiency, voltage profiles, and dQ/dV and dV/dQ curves, to most uncommon ones (Figure 5), like the evolution of the average voltage as a function of the cycle number, the capacity variance (with respect to the cell voltage) evolution during cycling, and the contribution of the constant voltage step (if any) on the (dis)charge capacity as a function of the cycle number. The examples reported in Figure 4 and 5 refers to a cycle life experiment, but similar analyses are performed for rate-capability tests as well. In addition to the plotting, the data are also reported in well-organized CSV files, as in the example reported in Figure 6.

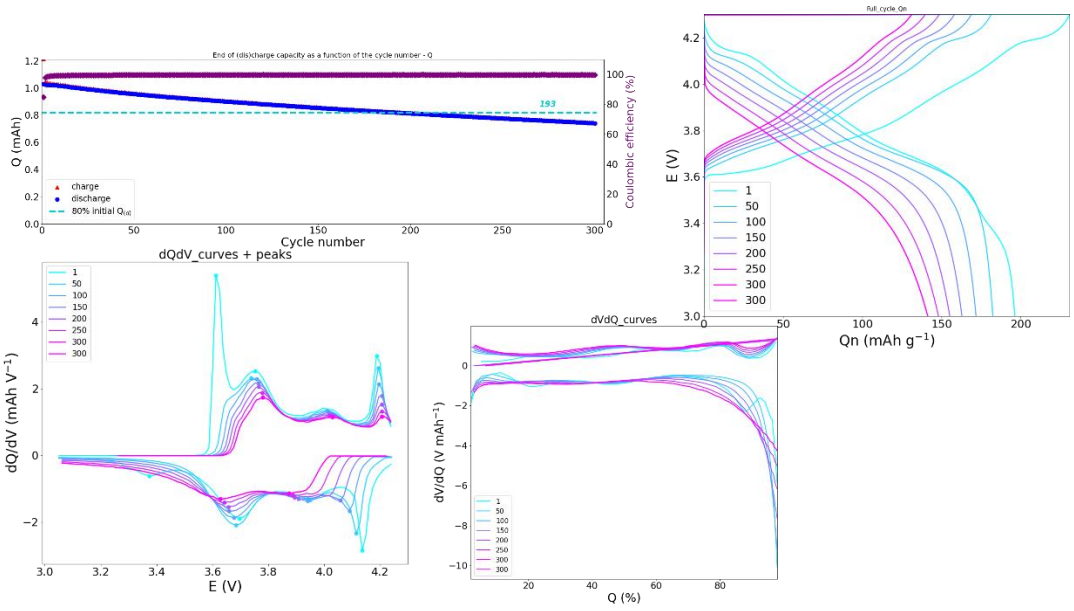

Figure 4. Example of CAP results for “classical” data analyses, like voltage profiles and dQ/dV curves.

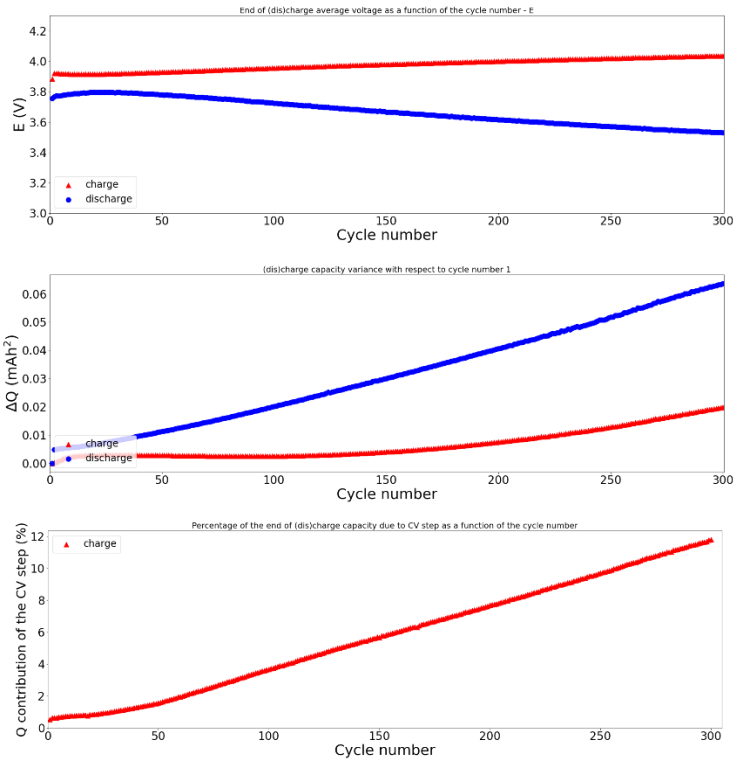

Figure 5. Example of CAP results for uncommon data analyses, like average voltage and capacity variance evolution as a function of the cycle number.

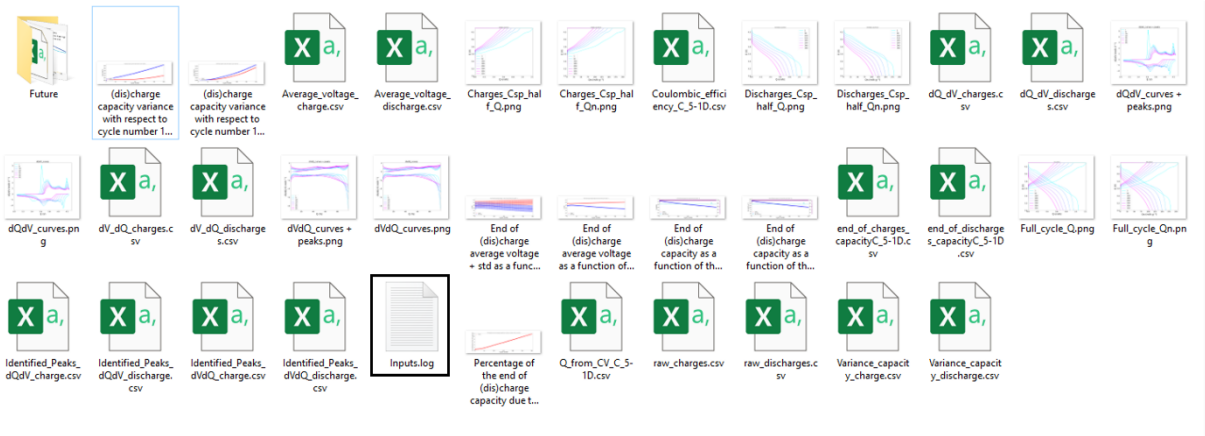

Figure 6. Example of results printed by the CAP for a cycle life experiment. Both the images and CSV files are reported in a devoted folder that is created by the CAP code.

Lastly, once the code is run, the information reported in the input Excel is reported in a log file (highlighted in black in Figure 6). This file has a double scope: First, it allows users to verify the inputs used, to verify if any mistake has been done. Second, if the code is run a

second time (the same file but recovered after a few days/weeks/months of cycling, *i.e.*, containing some extra data) the code automatically recovers the information. Therefore, in this case, it is enough to indicate in the Excel file the path for the file and the filename, while all the other information will be read in the log file, saving time to the user and minimizing the risks of mistakes.

Bonus – Running the Source Code and the Interactive Plottings

Together with the executable, the source code (Jupyter Notebook) is also available, allowing a higher degree of flexibility. In addition, through the source code is also possible to visualize, plot, and save interactive plotting (voltage profiles, dQ/dV, and dV/dQ curves comparing the cycles selected by the user).

In case you want to use this functionality but you have no clue about Python, learning how to install Jupyter notebook and the libraries used in the code (pieces of code already done and made public in the Python community) is extremely easy.

The easiest path is probably installing Anaconda (containing Jupyter Notebook, among other things):

https://www.anaconda.com/products/distribution

After that, you can launch the Anaconda Navigator (just search it after the installation) and from there you can run Jupyter Notebook.

The libraries to be installed are the ones reported in the very top of the code (“import Xxx” or “from Xxx”). It is very easy to install them (you can also google, library by library, how to install them, but typically it works for all in the same way):

https://youtu.be/Yr_ihLKq_yY

If you want to understand a bit better and learn more about Python (that can really be handy, so I definitely advise it!), you can look at the following videos series:

https://youtu.be/7uE6hypji0o

Multi cells data comparison

Data obtained for each cell analyzed through the CAP code can then be compared through a second code (“CAP_multi cells_cycle_life”), which also works in the same format as the CAP code (executable or code + Excel file for the model inputs). This code at the moment works also for comparing cycle life data.

The only input required by this code is the path in which you have placed the results of different cells (obtained previously through the CAP code), to be entered in the “Inputs_CAP_multi.xlsx” file. The folder containing the previous data should be organized as reported in Figure 7.

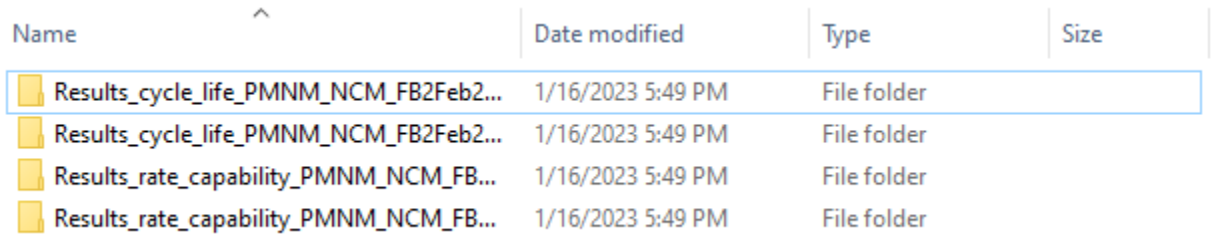

Figure 7. Example of folder structure for running the multi cells comparison code.

Running the model will lead to a message to inform you the analysis is been done (similarly to the CAP model) and the comparison of the data will be reported in a folder called “Comparison”. An example of results is reported in Figure 8.

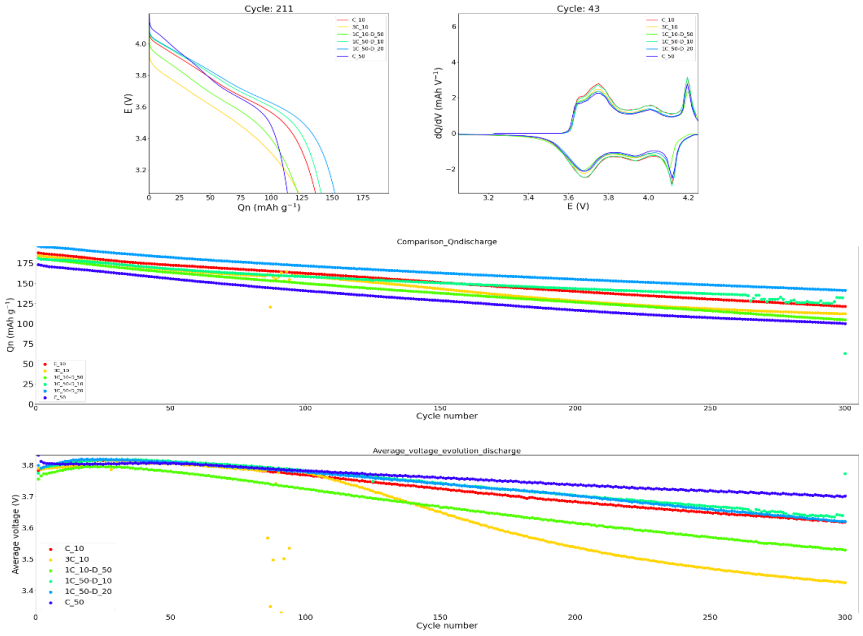

Figure 8. Example of results obtained through the cells data comparison code.

Contact for problems/doubts

If you have any problem using the codes, feel free to contact me on my personal email: teo.lombardo3@gmail.com