Single Cell Testing - Data Processing Guide

After completing our single cell testing, we have a lot of data to go through. Hopefully this page will help guide people in going through the data. If you have any questions, contact Micah Black. The tests chosen were guided by the Individual Cell Testing Evaluation page.

Going through the data started with a simple python script that would originally go through a chosen test file, the type of testing would manually be entered, and the data displayed using MatPlotLib. Getting the results for every single scenario has not yet been completed. This worked well for going through individual tests and validating that our instrument scripts worked, but not for data processing at scale.

Multiple testing scenarios were completed, and all the data logged. The scenarios and the instructions for going through the data can be found below. One thing to note - with some of the tests, the current measurements from the B2902A SMU were garbage data for the first 241 measurements. Also note that there were a few cells that were not tested, so there are not files for those number cells.

I would love to have a graph generated for every single data point for every single cell. The Visualization Status is for this - do we have a graph of every cell for this measurement, as well as a graph of the distribution of the results (plot the result from every cell on a graph).

Open Circuit Voltage

Priority	1
File Name	cell#-OcvTestScenario.csv
Results Status	Not Completed
Visualization Status	Not Completed

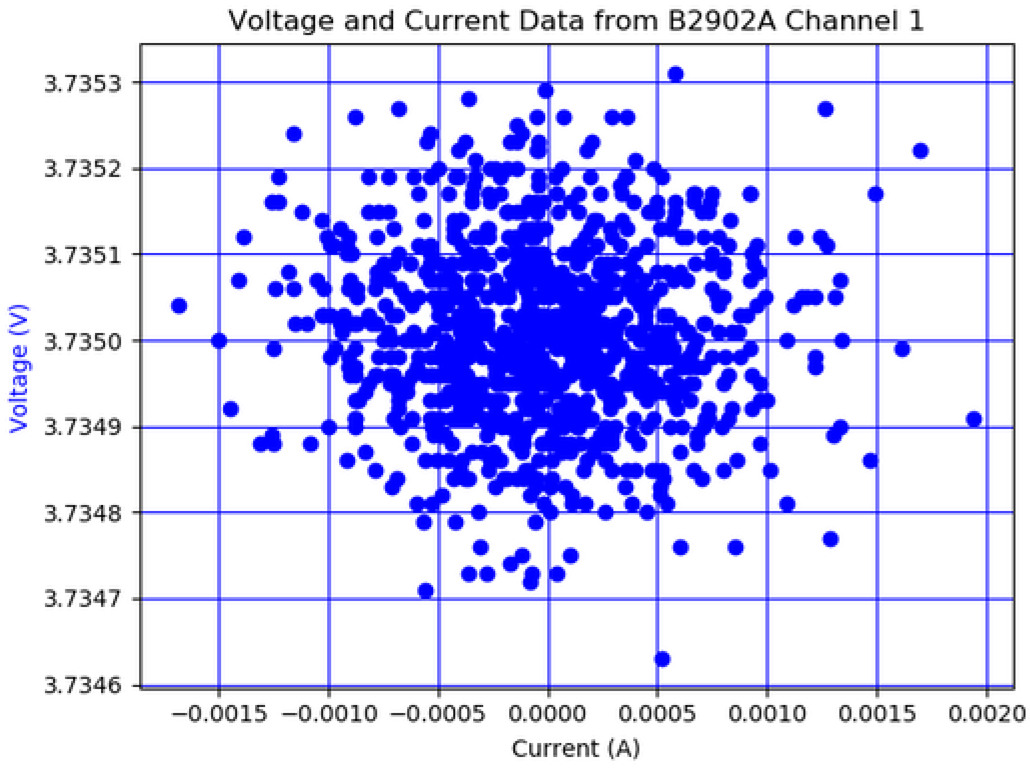
These cells have been sitting for almost a year since being manufactured before we tested them. All cells have what is called a self-discharge current, where the cell gradually discharges itself. To determine the self discharge current cells are usually left sitting and their voltage periodically monitored. Any cells that diverge in voltage quicker than other cells have a high self discharge current.

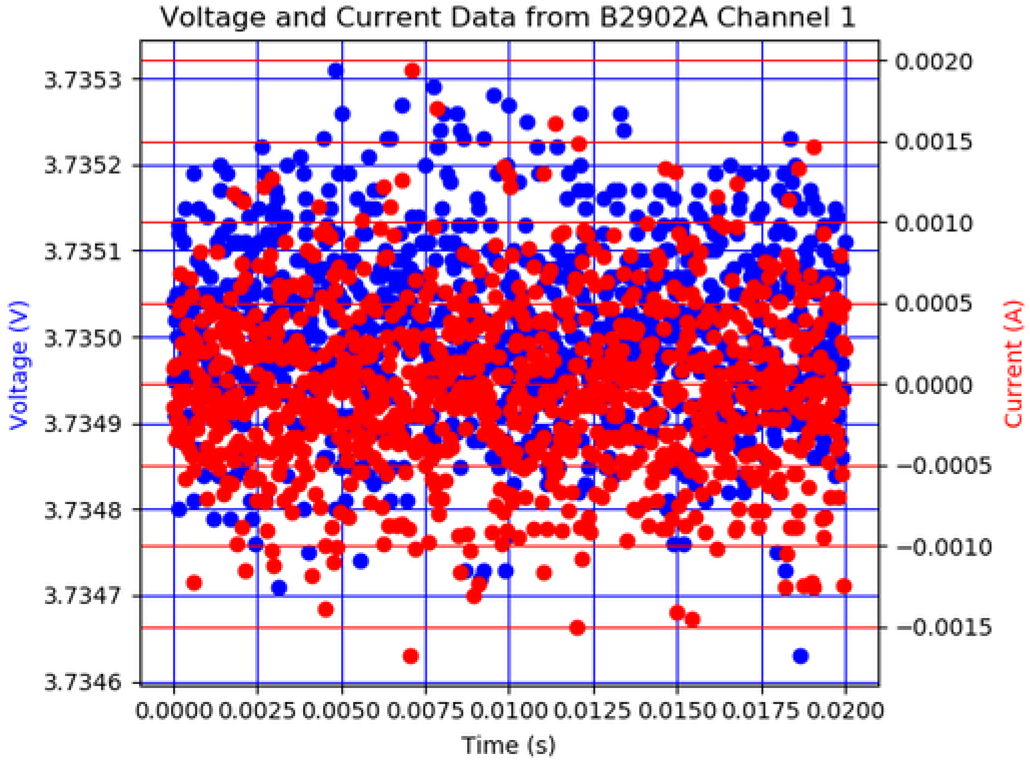
For manufacturers, this is not the best way to test cells as it is a long process. Keysight has come up with a novel way of measuring the self discharge current directly, similar to the way that water drips out of a small hole in the bucket. If you can push current in to the cell and the cell voltage does not increase or decrease, than you have found the self discharge current. This requires extremely precise equipment and there are lots of factors that come in to play much more evidently here that in other style of measurements, for instance the dependence of open circuit voltage on temperature. See here for more info: http://literature.cdn.keysight.com/litweb/pdf/5992-2292EN.pdf

For us though, the cells have already been sitting for a long time so the voltage should have already diverged and a simple OCV test should suffice as an indication of the cell's self-discharge current, provided we make the assumption that all cells when manufactured started at the same OCV and SoC.

This test measures the voltage of the cell while applying 0A of current to it.

Due to some measurement error and inaccuracies in controlling super low currents, the current measured by the SMU is jumps around a bit. We need to remove the measurements that current or voltage was an outlier, and then create somewhat of an average of the rest of the measurement with more weight being given to the voltage measurements taken when the current was closer to 0 - there is probably a more technical term for this, but you get the point. The distribution of sample data is below. Graphs were made in MatPlotLib in Python.





DC Internal Resistance Pulses

Priority	4

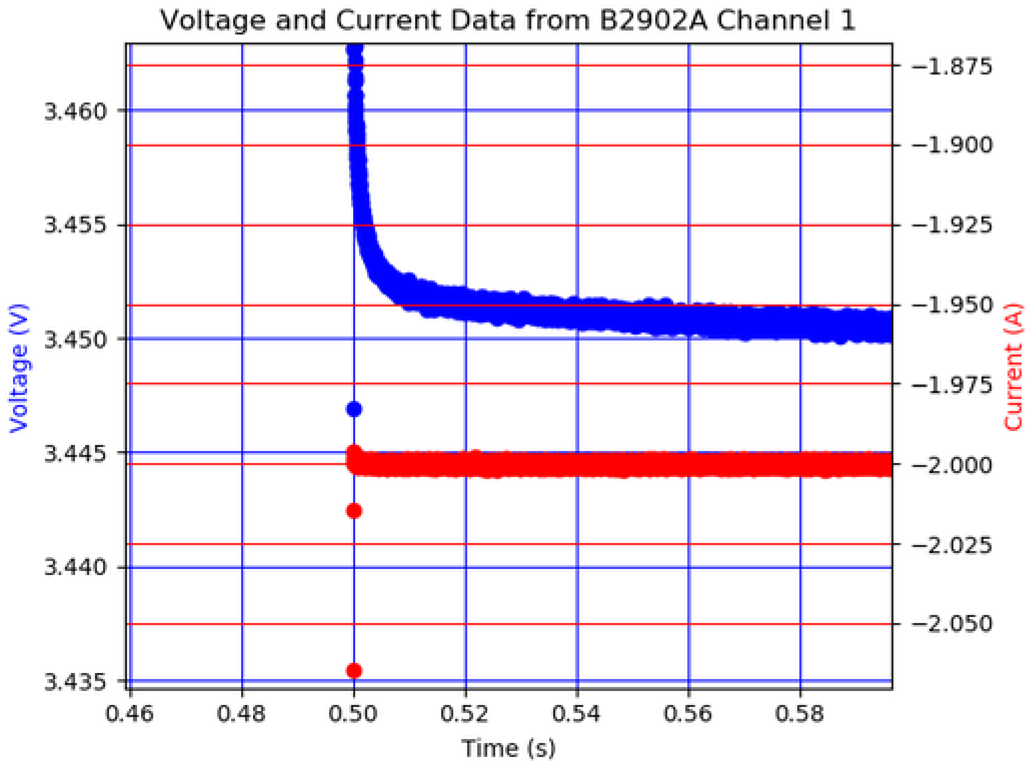
File Name	cell#-DcCurrentPulseScenario.csv
Results Status	Not Completed
Visualization Status	Not Completed

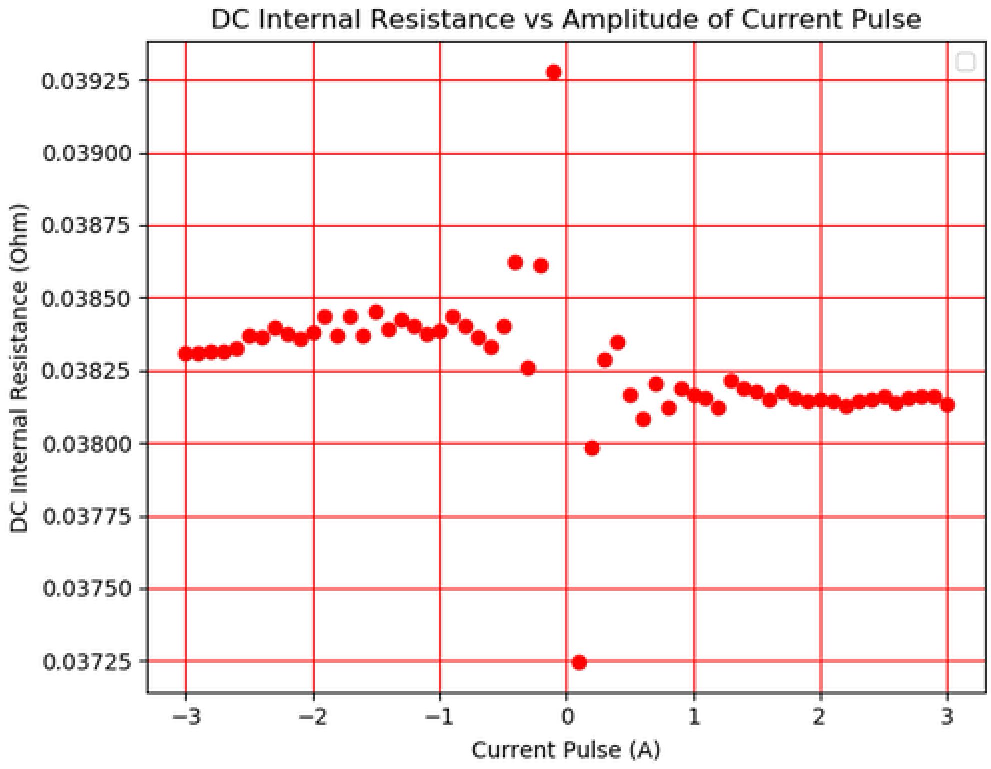
The DC Internal Resistance Pulse test is very common as it is super easy to do (maybe not so accurately) with a multimeter and a power resistor. However, there are many complications with this test.

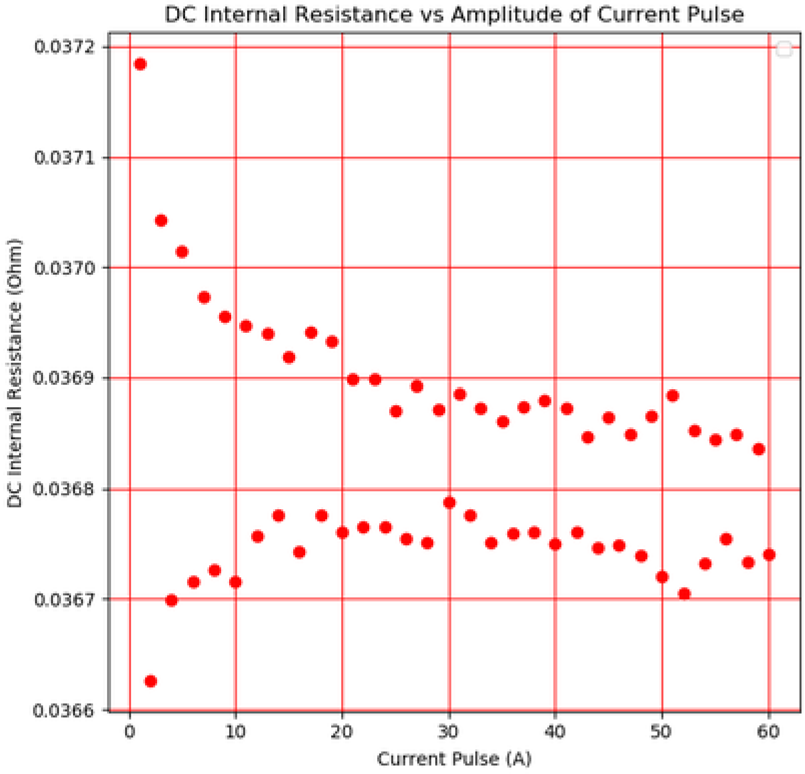
The voltage settling time is an interesting thing here. It gives a measurement in a sense of the mobility of the ions in the electrolyte. As the current is applied, the voltage does not drop immediately, it requires some time to approach the linear area.

Also notice that there is still a small slope during the linear region. This is the effect of discharging a cell - the voltage drops. We want to find the slope of the line and trace is back to when the pulse of current was first applied to the cell. This will tell up the exact internal resistance of the cell, and not include any effects of the discharge voltage drop.

The measured internal resistance (shown in the second graph) also seems to change depending on the number of current pulses that you do (but not on the amplitude of the current pulse) - compare second and third graphs. We did pulses at a bunch of different currents, but we should try and see where the measurements converge to as seen in the third graph. Note that the charge and discharge currents seem to converge to different values, and then should be calculated separately.







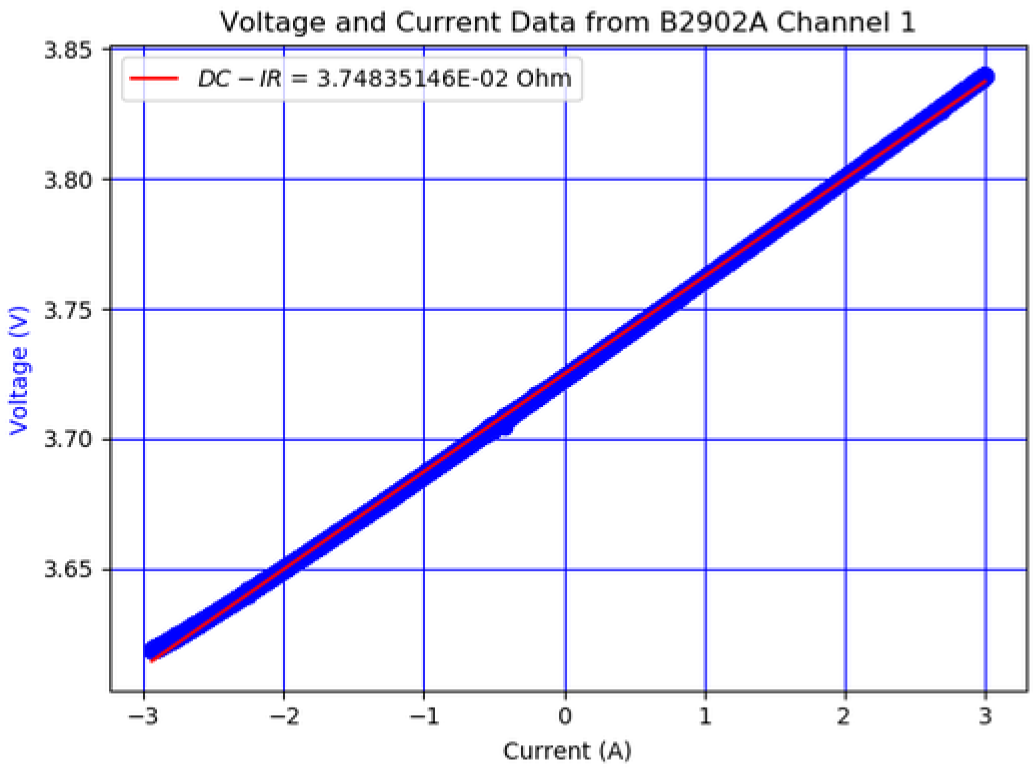
Note on this graph (the third graph), the bottom axis should read pulse number, and not amplitude of current. These pulses were all done at the same current.

DC Internal Resistance Slope

Priority	3
File Name	cell#-DcLinearSweepScenario.csv
Results Status	Not Completed
Visualization Status	Not Completed

This measurement is similar to determining the resistivity of a sample of material, but applied to a battery cells. We applied a changing current from -3A all the way to 3A and measured the voltage of the cell. This gives a fairly linear response, but would maybe be better split up in to charge and discharge sections, or possibly a quadratic would fit the data better.

The slope of the line on a voltage vs current curve gives you the internal resistance of the cell. It is essentially a DC pulse measurement, but average over all amplitudes of pulses.



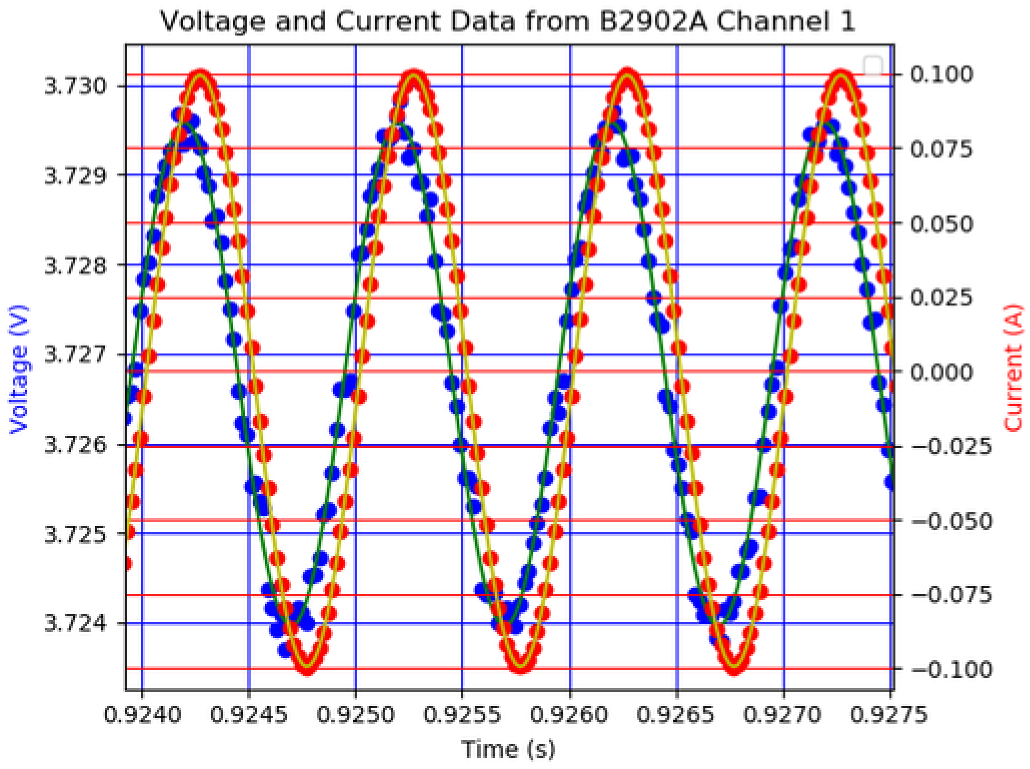
AC Internal Impedance (1kHz)

Priority	2
File Name	cell#-AcInternalScenario.csv
Results Status	Completed
Visualization Status	Not Completed

The AC Internal Impedance test at 1kHz is an industry standard test for measuring the internal resistance of cells, and is usually the value that manufacturers put an upper limit on. (LG says <40mOhm no the MJ1 datasheet I believe). The AC test produces different results than the DC methods as the AC method also takes into account the inductance and capacitance of the cell, and some of the other complexities of the equivalent circuit model. Essentially, the voltage never has time to fully settle out.

The test happens by applying a 1kHz sinusoidal current (or voltage) to the cells, and measuring the response. In our case, we applied a 100mA sinusoidal current profile to the cell, and measured the voltage response.

The internal impedance is calculated by dividing the amplitude of the voltage by the amplitude of the current. We used MatPlotLib and python to fit our data to sinusoidal profiles and extract the parameters, then got our results.



Capacity Ration

Priority	6
File Name	cell#-TenSecondsDischargeScenario.csv
Results Status	Not Completed
Visualization Status	Not Completed

I think that this test is super cool and has the possibility to give some super interesting results.

The term capacity ration is the capacity per percent state of charge.

Basically the idea here is to take a small chunk of the charge-discharge curve, remove any transients from the start of the test, then find the capacity drained from the cell during this small chunk of time.

Then, if we make the assumption that all cells will have a relatively similar state of charge vs OCV graph (or we can take the internal resistance in to account as well), and then match the slope of the discharge curve to an area on the full discharge of a single cell (which can be approximated as linear in small regions).

We can compare the capacity obtained from the reference cell to the capacity obtained from the 10 second discharge during this region, and get a capacity estimate for the cell.

Weight (Possible Indication of Capacity)

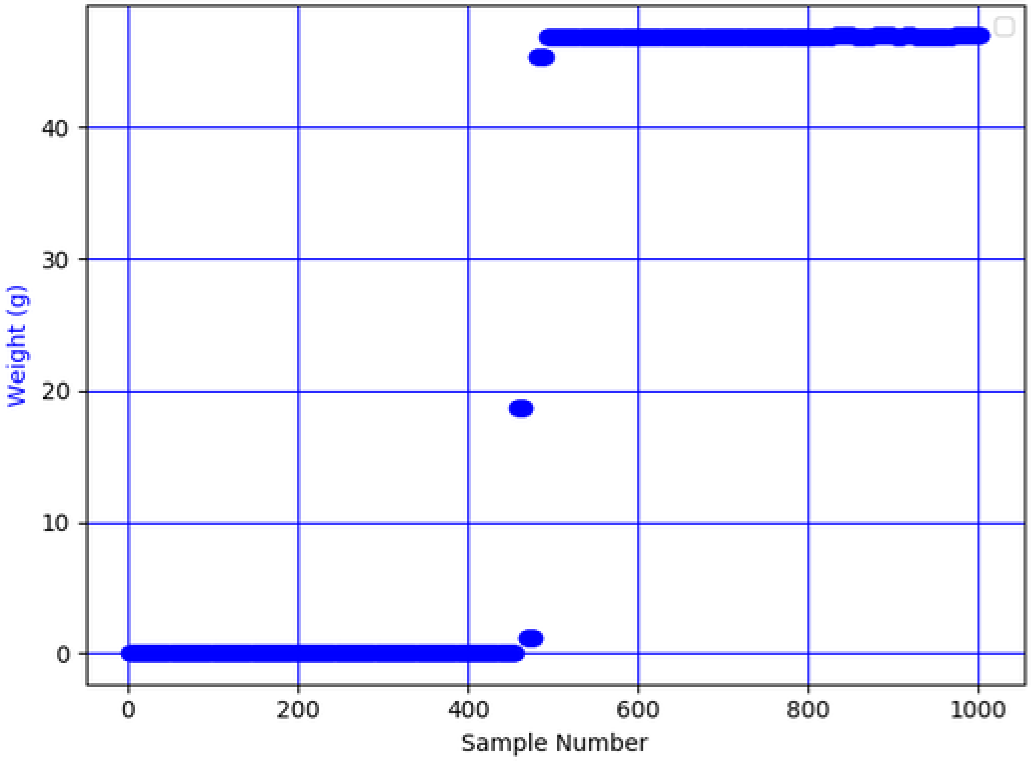
Priority	5

File Name	cell#-weight.csv
Results Status	Not Completed
Visualization Status	Not Completed

Some of the papers linked on the Individual Cell Testing Evaluation page found a correlation between weight and capacity, while others did not. For us, this data is just another source of outlier identification.

The cells are all super close in weight per the manufacturer's spec. We are expecting a maximum change in weight of under 0.5g over all 1400 cells.

Getting a number for the weight requires a bit of data processing, and understanding of how the tests were conducted.



Sample data for the weight measurements is shown in the graph above. The data will start with either the old cell on the scale, or no cell on the scale. If the old cell was on the scale, it is removed, and then the new cell is put on the scale. There will be measurement errors, so zoom in on the data in MatPlotLib to see what you're dealing with, remove outliers, and average it.

All testing result data can be found here in the Single-Cell-Testing-Results zip file: https://drive.google.com/drive/u/2/folders/19MzeWnzwiVnc4nDDM7SIhcqjXieO3bOU

Warning - We have a lot of data - The zip file is 500MB, and extracted it is over 6GB.

Here is the python file used to generate the AC-IR values. It is a bit of a mess. It should be run from the same file as the data is in as of right now, but can definitely be improved.

AC_Internal_Script.py

ac-irs.csv