Update: CHESS2 Readout

Derek Hamersly

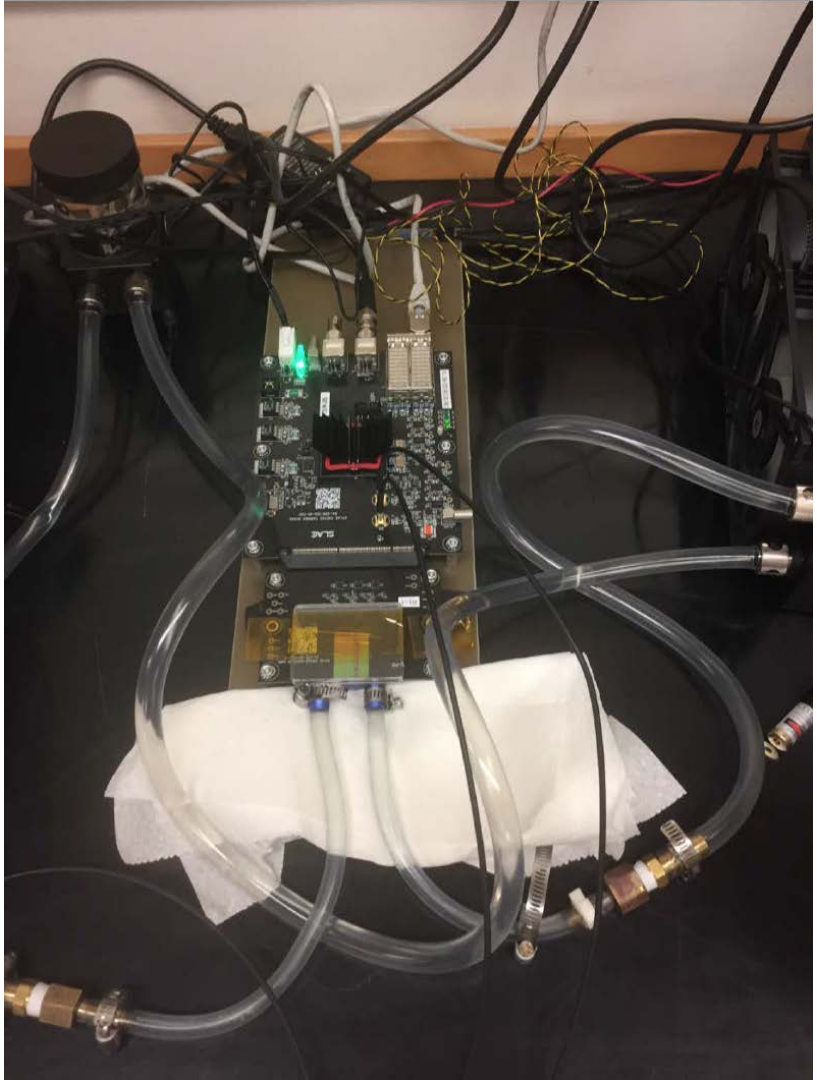
Aug 2nd, 2018

Review

Water cooled chip

Data acquisition software:

- Hitmap_Plotter -- live hitmap
- Frame_data -- decodes frames
- Chess_control -- easier to configure
- Chess2 chip



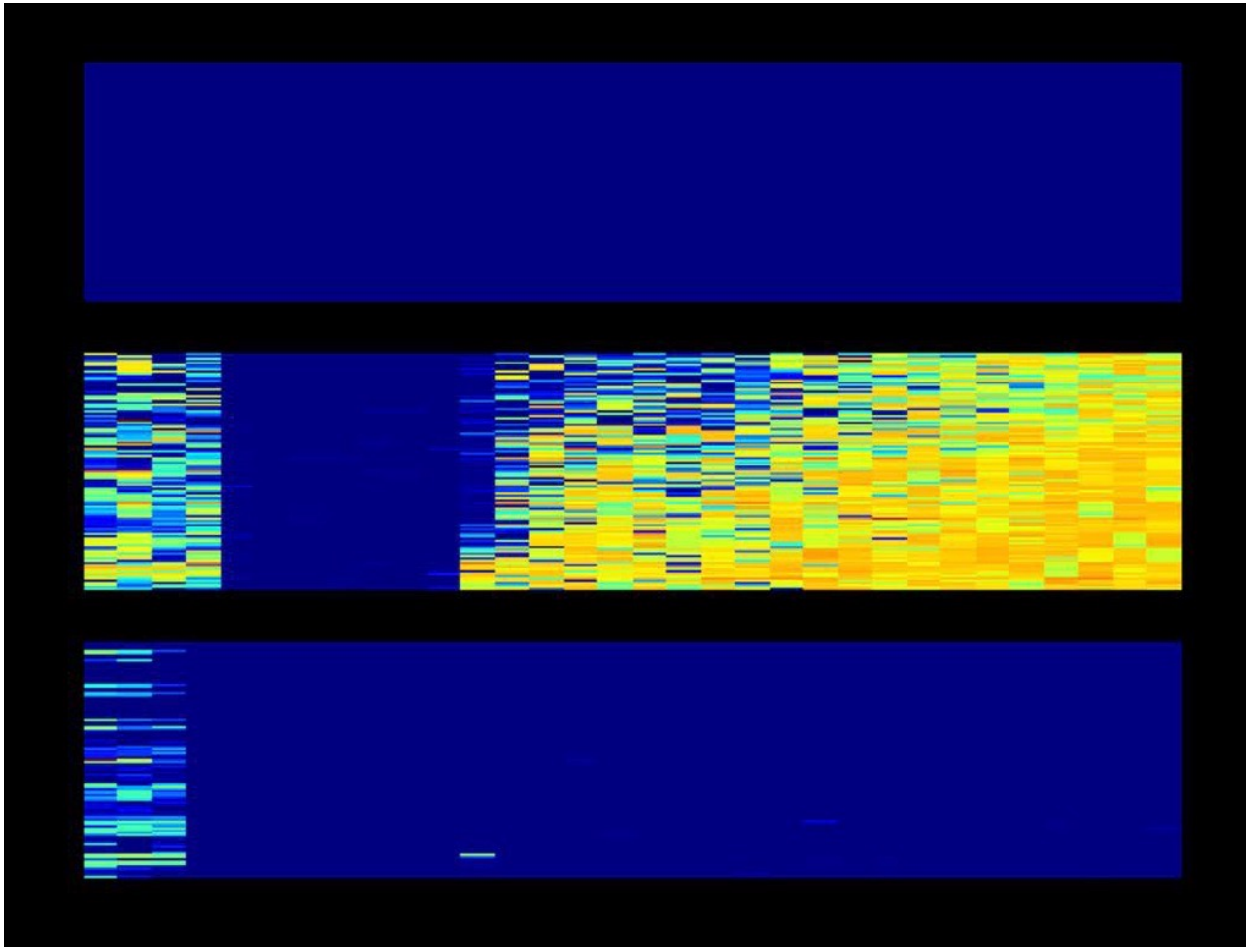
Hitmap GUI—all pixels enabled (low threshold)

(no activity)

matrix 0

matrix 1

matrix 2



Scan Test

- Scanning various front end parameters such as bias currents.
- **Purpose:** determine optimal working values for all parameters
- **Current method:**
  - Scan parameters through allowed range (0-32, log scale)
    - Make histograms for each value, for each parameter
    - For now, simply pick best looking (“S” curve) plot by eye, and update parameter in configuration file
- **To do:**
  - Scan multiple regions, understand why there are dead zones
  - Need better algorithm for finding optimal working values

Scan Test Main Loop

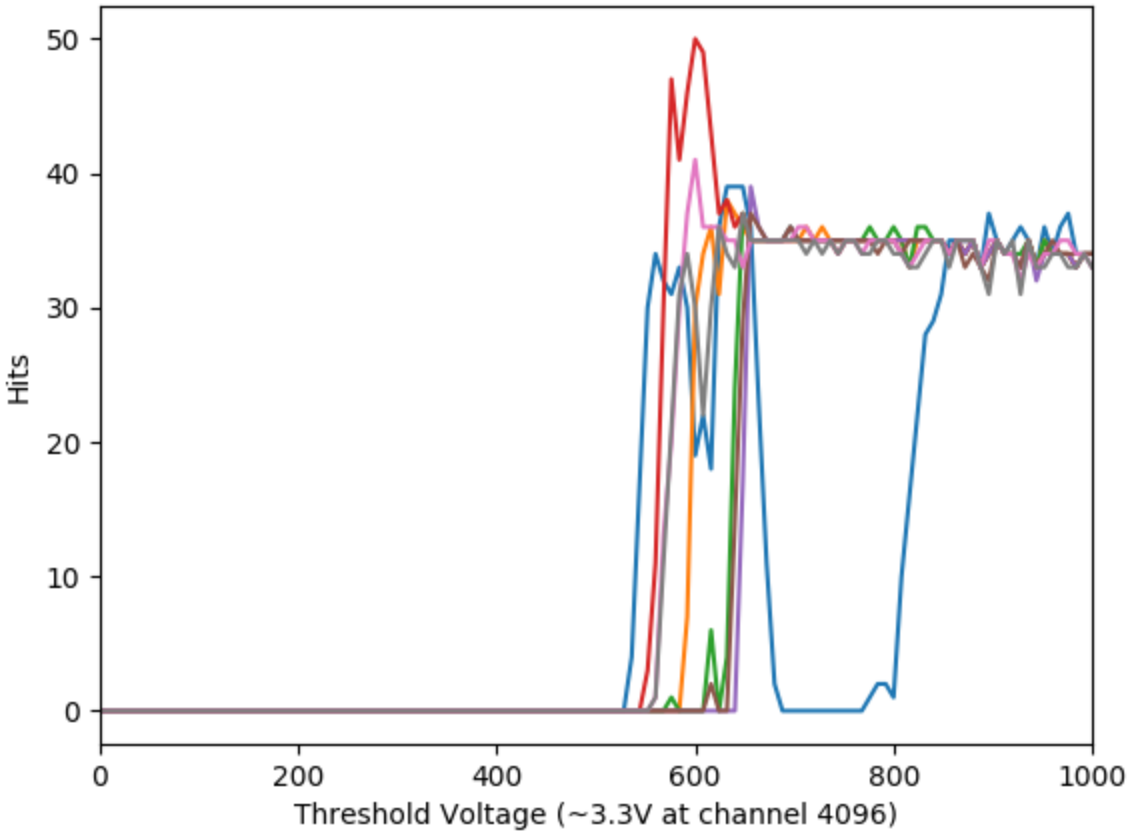
```
hist_fig = Hist_Plotter(self.shape,x_list,x_label,fig_title)
hist_fig.show()

for x in x_list:
        if self.is_th_scan:
                system.feb.dac.dacPIXTHRaw.set(x)
        else:
                system.feb.dac.dacBLRaw.set(x)
                system.feb.dac.dacBLRRaw.set(x+144)
        eventReader.hitmap_reset()
        system.feb.sysReg.timingMode.set(0x0) #enable data stream
        print("taking data")
        trig_count = 0
        while trig_count < self.ntrigs:
                time.sleep(self.sleeptime/1000.0)
                system.feb.sysReg.softTrig()
                trig_count += 1
        system.feb.sysReg.timingMode.set(0x3) #stop taking data
        eventReader.hitmap_plot()
        eval("hist_fig.add_data(eventReader.plotter.data"+str(self.matrix)+"[self
topleft[1]+self.shape[1]])")
        hist_fig.plot()
stop_time = datetime.now()
self.save_fig(hist_fig,val)
plot_config_msg = self.get_plot_config_msg(system,val,val_fields,start_time,stop_
self.save_fig_config(val,plot_config_msg)
hist_fig.close()
del hist_fig
```

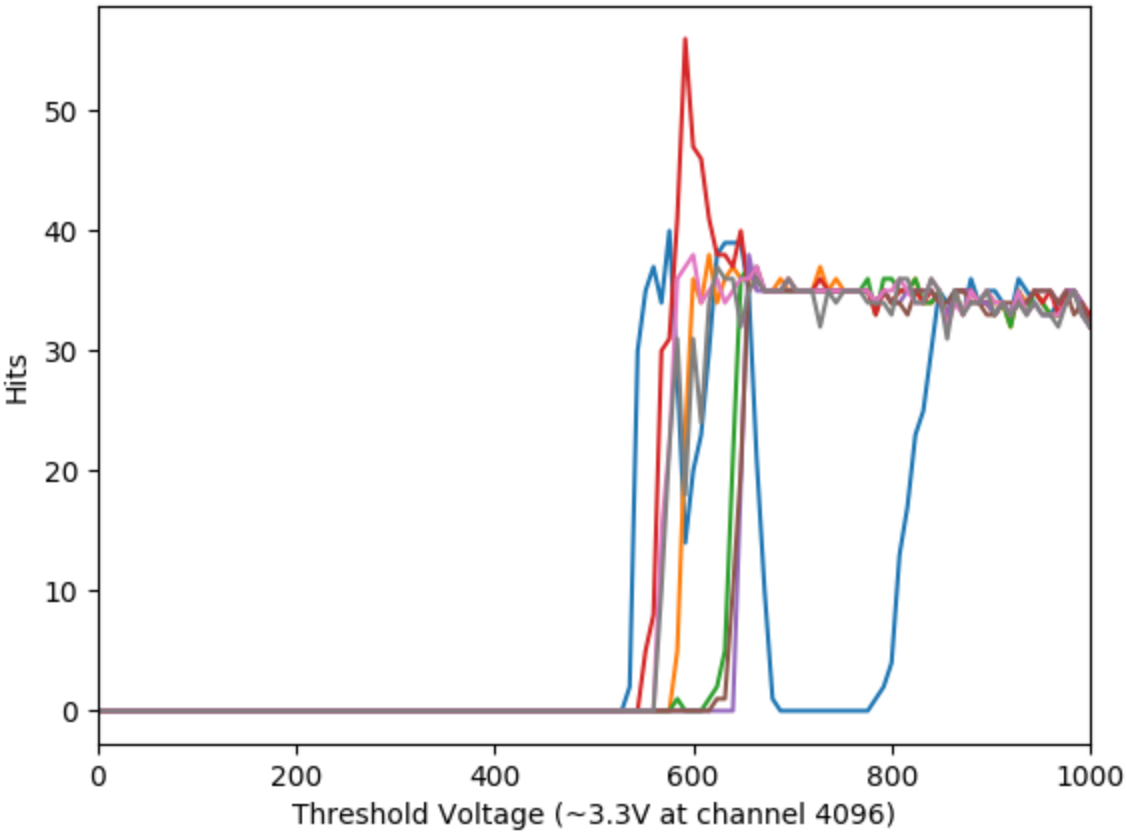
Example of scan for some parameter

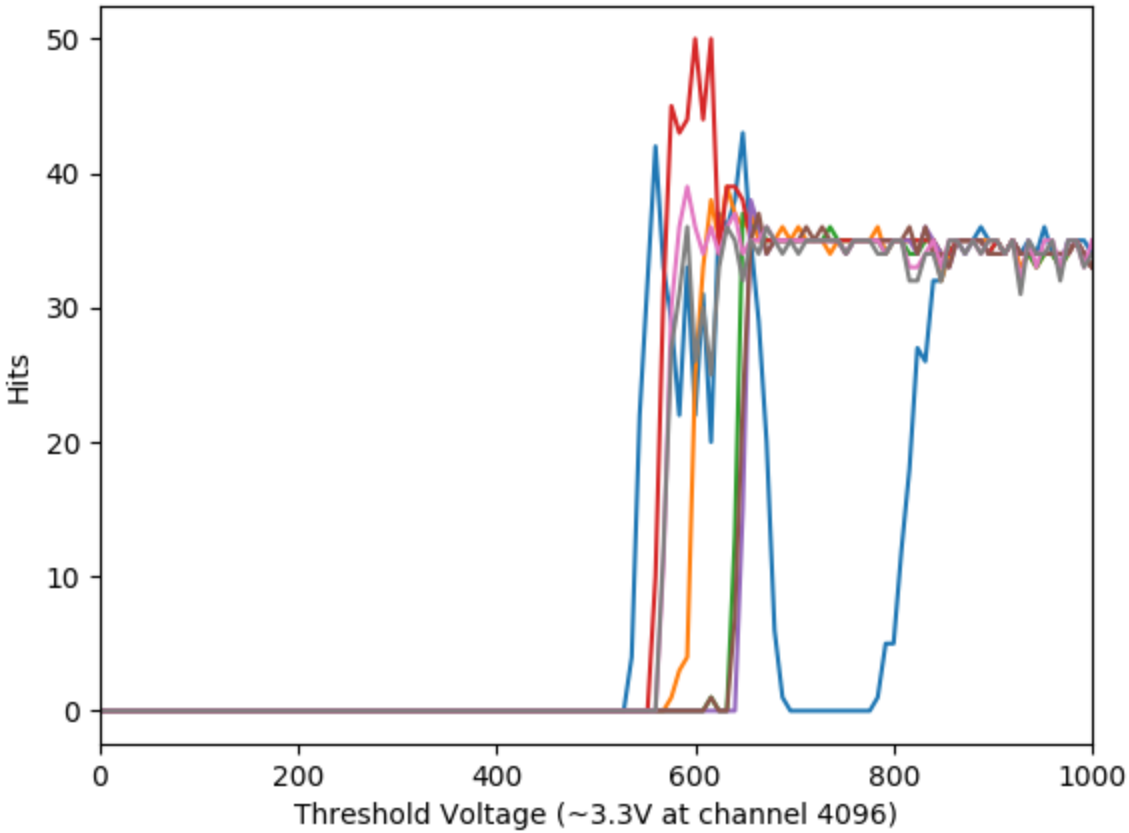
Create histogram showing 8 pixels (different colors) over 125 threshold bins

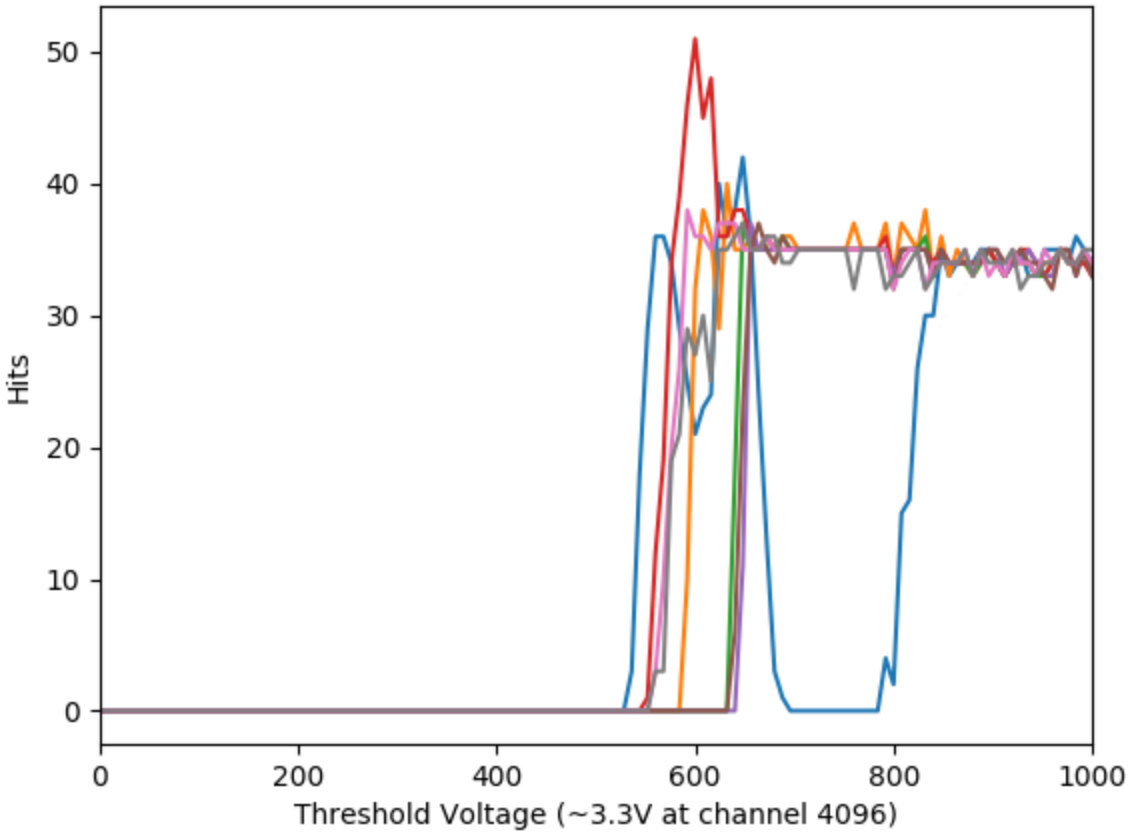
Following slides show scan of “VPFB attenuation” parameter over a log scale

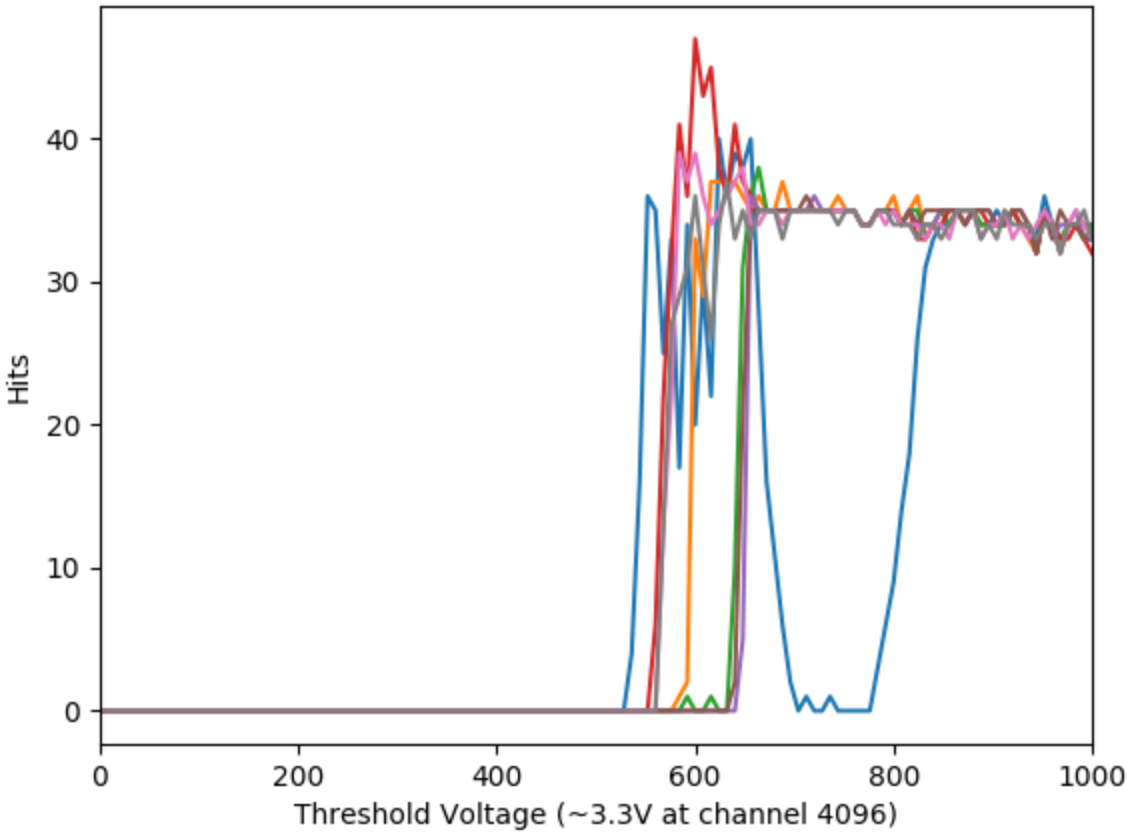
each plot takes ~20 sec to create

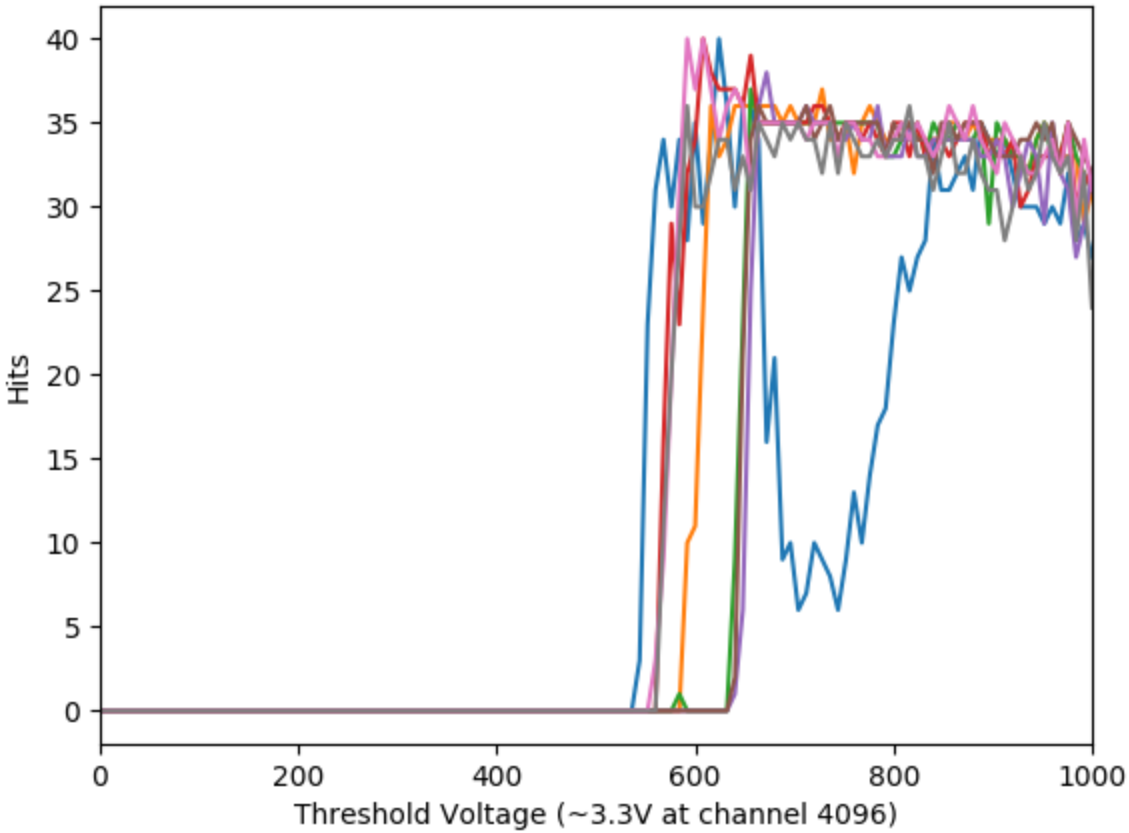


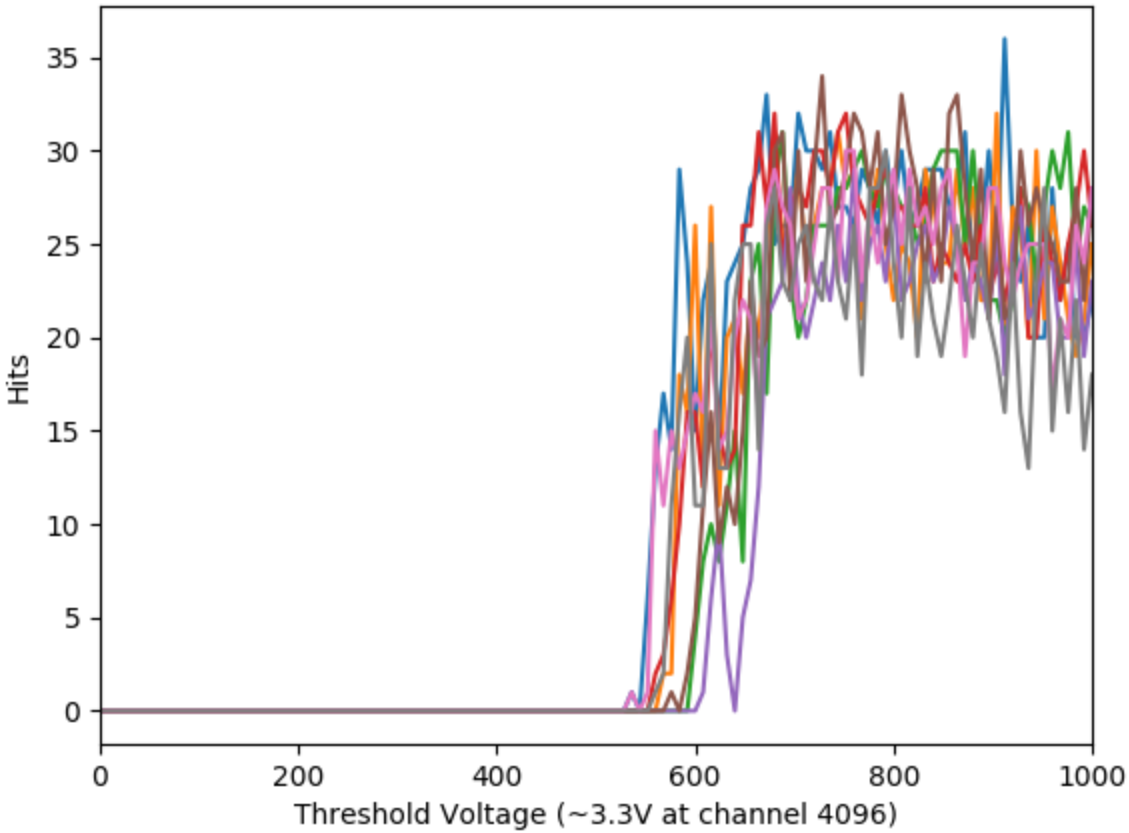

Hits
50
40
30
20
10
0
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)

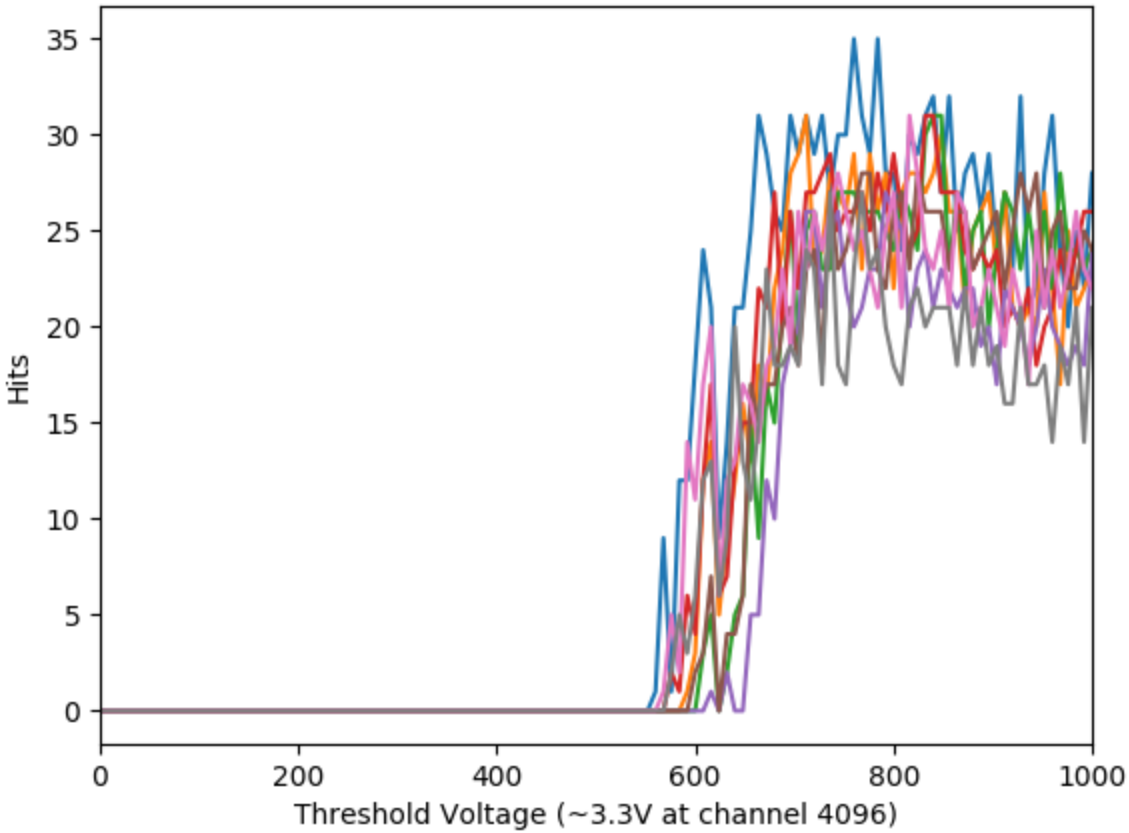

50
40
30
20
10
0
Hits
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)

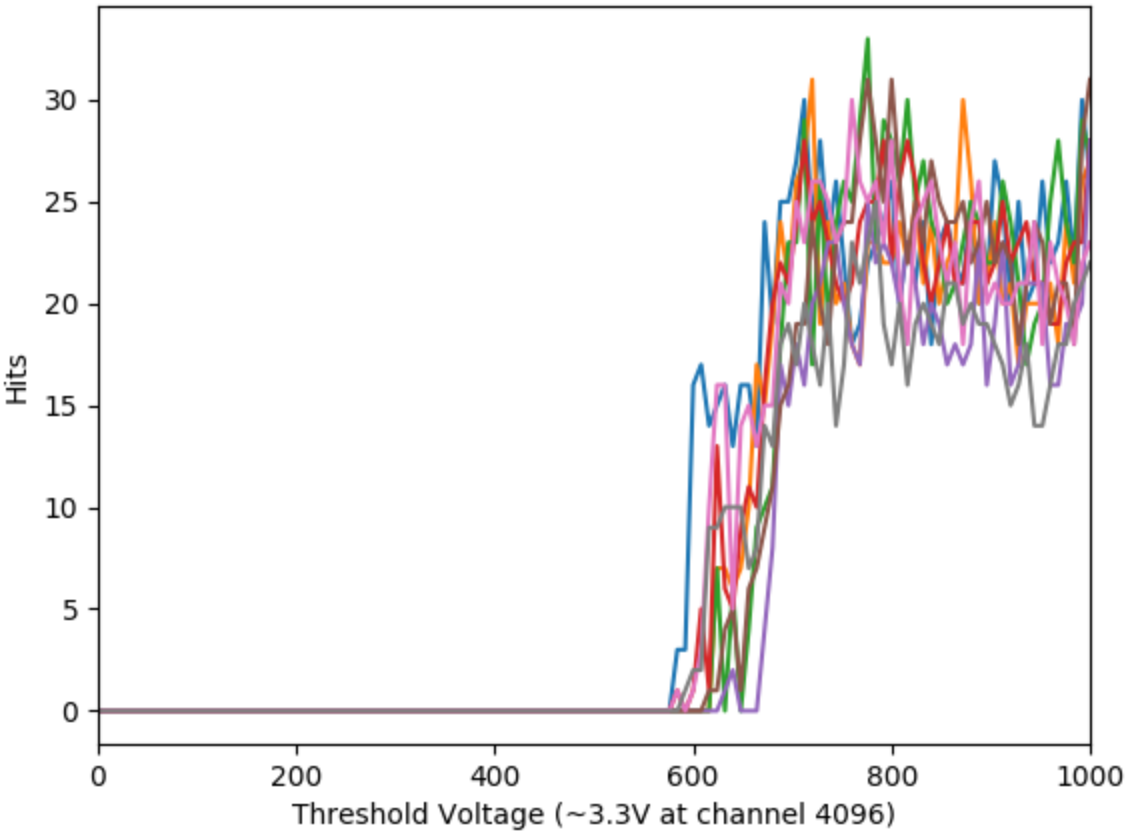

Hits
50
40
30
20
10
0
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)

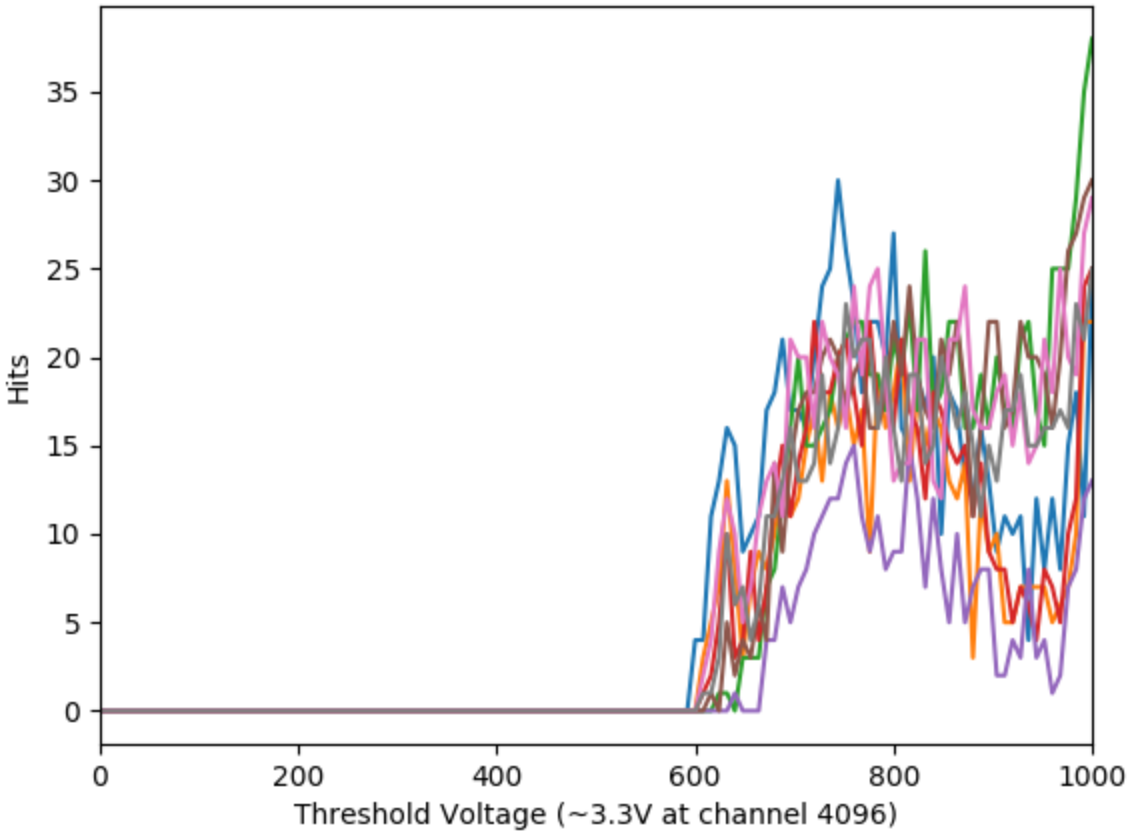

Hits
0
10
20
30
40
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)

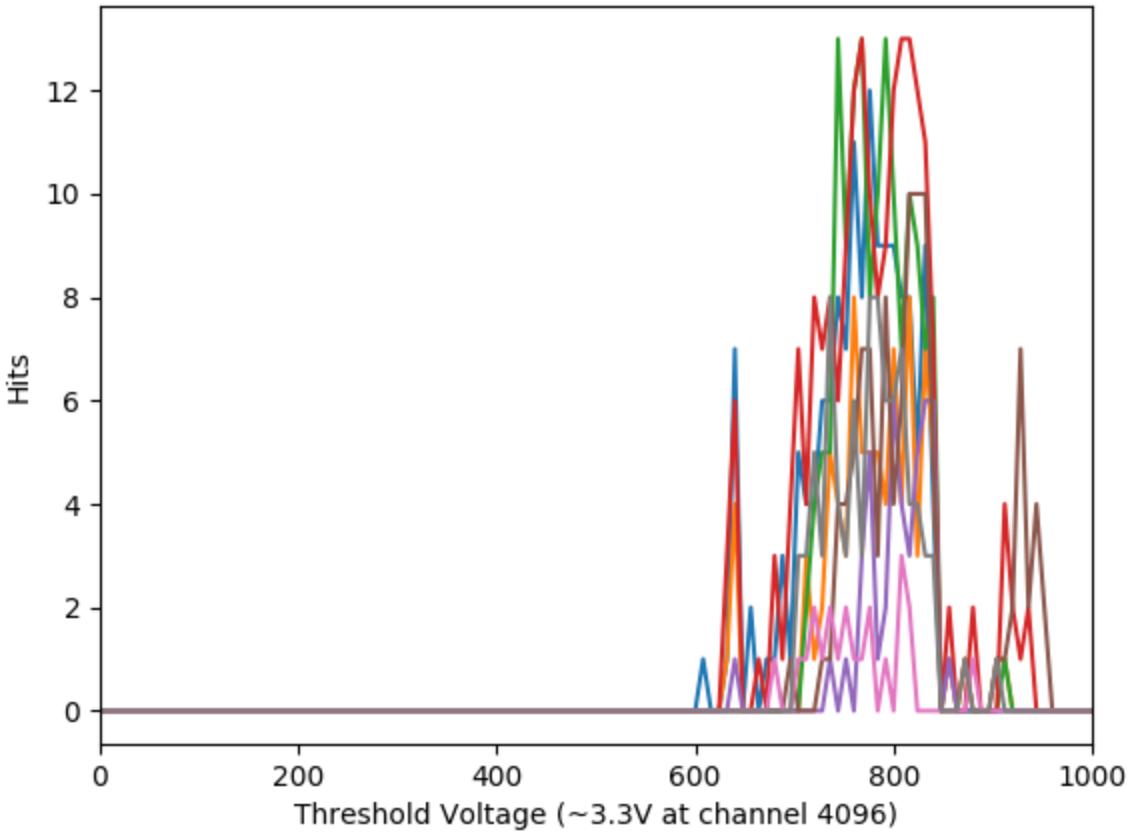

Hits
40
35
30
25
20
15
10
5
0
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)


Hits
35
30
25
20
15
10
5
0
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)


Hits
35
30
25
20
15
10
5
0
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)


Hits
30
25
20
15
10
5
0
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)


Hits
35
30
25
20
15
10
5
0
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)


Hits
0
2
4
6
8
10
12
0
200
400
600
800
1000
Threshold Voltage (~3.3V at channel 4096)

In Progress: Saving Data to csv files

Maintain fast plotting: accumulate data during runtime, but only write csv files after plots are completed

Proposed File Tree:

<This Scan (datetime)>/

config.txt

Plots/

All plots (png)

Data/

All data for plots (csv)

To Explore

- Readout trigger behavior
  - More data arrives after multiple readout triggers
- Histograms
  - No “S” curve
  - Threshold offset?
- Dead zones
  - Matrix 0 seems completely dead
  - Matrix 1 has a large dead zone
- Regional differences in pixel noise

Github

Currently pushing to public mirror repo: thederber/Feb-software

Available for cloning