I want to update Public Super Macro to take in specific run numbers for specific regions

Go is in the src directory. It runs the workers which are passed in as a list.

Go([\

u1,\

ReadCIS(),\

CleanCIS(),\

readbchfromcool,\

readcalfromcool,\

CISFlagProcedure\_modified(dbCheck=database\_check),\

getscans,\

# cis\_recalibrate\_procedure,\

hist\_percent\_public,\

hist\_percent\_regions,\

calib\_const\_regions,\

flag\_plots,\

cool\_plots,\

rms\_plots,\

calibration\_distributions,\

tune\_rms\_chi2,\

tune\_maxpoint\_likelycalib,\

detectortimestability,\

pr,\

s],

memdebug=mdebug)

* U1,\
  + Use(run=runs, runType=’CIS’)
  + I think here i just need to modify which runs are used?
  + If one date is specified, runs is just the date
  + If two dates are specified runs is a tuple (date1, date2)
  + If ldates is specified runs is the list of runs
  + Use.py is located in workers directory
  + I THINK If I run Use 8 times (4 partitionx2 gain) that this will work. Currently Use takes in an optional region argument but it is not used in public super macro.
  + So I say Use(run=runs, runType=’CIS’, region=LBA) etc
  + Only issue is this does not specify gain….. Hmmmm
  + Really ugly workaround is for region specify all for example ‘EBA\_m01\_c00\_lowgain’ ‘EBA\_m01\_c01\_lowgain’ etc but that might crash

Use.py

def \_\_init\_\_(self, run, run2='', type='readout', region=None,

useDateProg=True, verbose=0, runType='all',

keepOnlyActive=True, filter=' ',

amp='15000', getLast = False, updateSpecial = True,

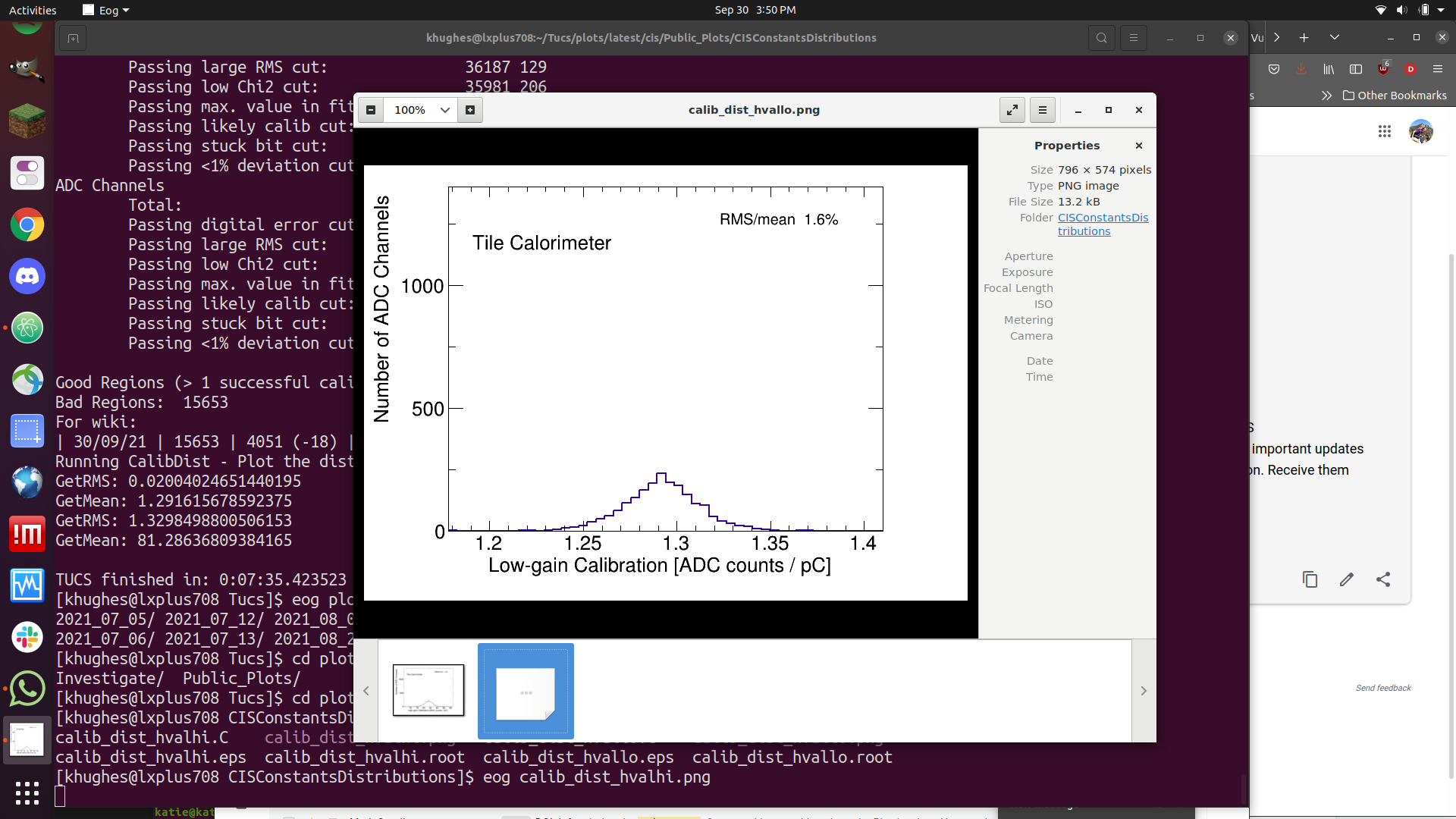
updateC10 = True, specialFile='./specialcells.txt',#akamensh

allowC10Errors = False, cscomment='', TWOInput= False,

preprocessing=False)

So adding in two Use calls seems to work without error. Only issue is the plots look really wrong.

The CIS constant distribution looks like this:



Is this an issue or not? I think no, normal plots have the peak at about 1000 and this is about ¼ of that.

Problem: I think the runs are not being repeated

It is also incredibly slow.

I tried running:

Go([\

Use(run=Run\_List['EBA\_lowgain'], runType='CIS', region=GainToList('EBA\_lowgain')),\

Use(run=Run\_List['EBA\_highgain'], runType='CIS', region=GainToList('EBA\_highgain')),\

Use(run=Run\_List['EBC\_lowgain'], runType='CIS', region=GainToList('EBC\_lowgain')),\

Use(run=Run\_List['EBC\_highgain'], runType='CIS', region=GainToList('EBC\_highgain')),\

ReadCIS(),\

CleanCIS(),\

readbchfromcool,\

…..

It took forever to run.

Other issue is GainToList is a very long string.

‘EBA’

Use.py creates a global Use\_run\_list and Use\_date\_list which are referenced in ReadBadChFromCool

Additionally i think i can specify gains???

Ex in ProcessRegion many take in a region and look if ‘gain’ in region.GetHash()

macros/cis/Test\_Super\_Macro.py --date "28 days" --validruns 'results/newSept2021/valid\_runs.txt' --gcals

ReadCIS uses a global run\_list

And goes through run in run\_list

In ReadCIS: the runs it has selected and printed out in ProcessStop are:

tileCalibCIS\_400063\_CIS.0.root

tileCalibCIS\_400103\_CIS.0.root

tileCalibCIS\_400323\_CIS.0.root

tileCalibCIS\_400656\_CIS.0.root

tileCalibCIS\_400841\_CIS.0.root

tileCalibCIS\_400899\_CIS.0.root

tileCalibCIS\_401325\_CIS.0.root

tileCalibCIS\_401520\_CIS.0.root

tileCalibCIS\_401633\_CIS.0.root

tileCalibCIS\_401687\_CIS.0.root

tileCalibCIS\_401696\_CIS.0.root

tileCalibCIS\_401898\_CIS.0.root

tileCalibCIS\_402124\_CIS.0.root

This is all of them.

EBA\_lowgain [401633, 401687, 401696, 401898, 402124]

EBA\_highgain [400063, 400103, 400323, 400656, 400841, 400899, 401325, 401520, 401633, 401687, 401696, 401898, 402124]

Current idea:

Use all runs (cumulative\_runs) because doing many use calls on different regions takes too long

ReadCIS and CleanCIS on these runs

Then in individual plots, I filter by partition using the list of valid runs ?

Read calib from cool is taking forever lol

Output looks like this sorta

Calib:78.7998275756836 | Cum. RMS/MEAN 0.0034795978353328615

Calib:77.92973327636719 | Cum. RMS/MEAN 0.004223772961756242

TILECAL\_LBC\_m39\_c38\_highgain 78.63418851579938 0.3321329593226766 0.004223772961756242

Run Number: 401898 | 1.3091614246368408

Run Number: 401520 | 1.3088306188583374

Run Number: 400656 | 1.3078948259353638

Run Number: 402124 | 1.3093687295913696

Run Number: 401632 | 1.3049342632293701

Run Number: 400841 | 1.307617425918579

Example:

Run Number: 400558 | 80.4042739868164

Run Number: 400323 | 80.41374206542969

Run Number: 401898 | 80.49010467529297

Run Number: 400656 | 78.57536315917969

Run Number: 400841 | 80.39734649658203

Run Number: 402124 | 80.46157836914062

Run Number: 400063 | 80.40805053710938

Run Number: 401632 | 80.50738525390625

Run Number: 400899 | 80.5368423461914

Run Number: 400103 | 80.3990707397461

Run Number: 401633 | 79.60287475585938

Run Number: 401001 | 80.49490356445312

Run Number: 401687 | 79.87660217285156

Run Number: 401325 | 80.48422241210938

Run Number: 401696 | 80.50096130371094

Calib:80.4042739868164 | Cum. RMS/MEAN 0.00021627503416051677

Calib:80.41374206542969 | Cum. RMS/MEAN 0.0003297817908078129

Calib:80.49010467529297 | Cum. RMS/MEAN 0.00060949836617017

Calib:80.39734649658203 | Cum. RMS/MEAN 0.0006391328375569375

Calib:80.46157836914062 | Cum. RMS/MEAN 0.0007615559737316418

Calib:80.40805053710938 | Cum. RMS/MEAN 0.0007953311155871461

Calib:80.50738525390625 | Cum. RMS/MEAN 0.0009797946801689533

Calib:80.3990707397461 | Cum. RMS/MEAN 0.0009996627192358973

Calib:79.60287475585938 | Cum. RMS/MEAN 0.0027391908769577597

Calib:80.49490356445312 | Cum. RMS/MEAN 0.0027898308841977186

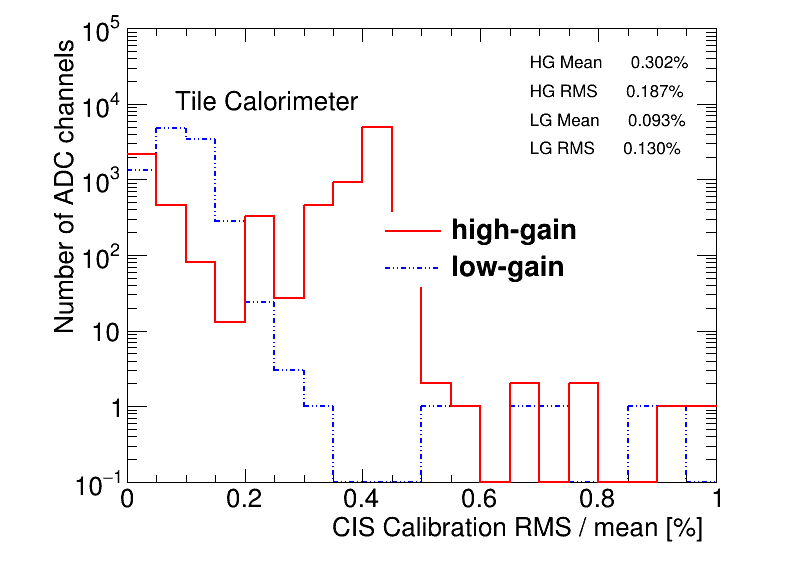
Calib:79.87660217285156 | Cum. RMS/MEAN 0.0032187273394510703

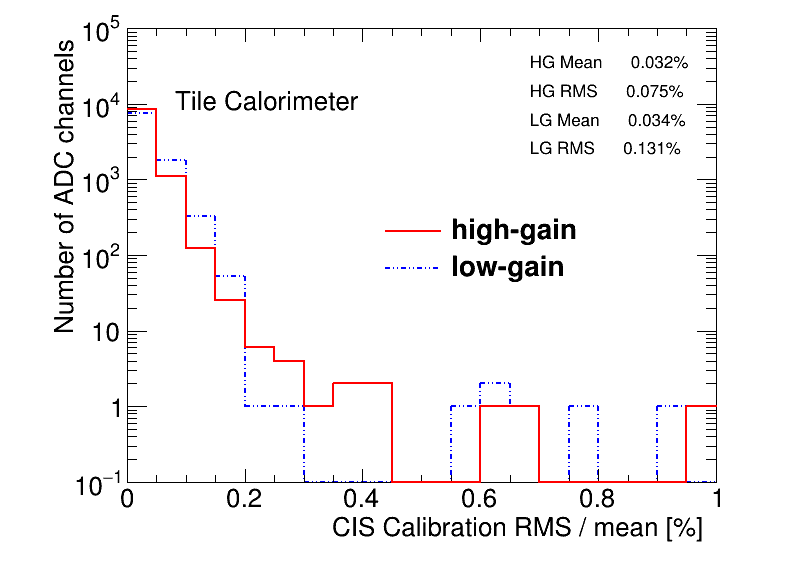
Calib:80.48422241210938 | Cum. RMS/MEAN 0.0032561531336224156

Calib:80.50096130371094 | Cum. RMS/MEAN 0.0033022857479513843

TILECAL\_EBC\_m15\_c41\_highgain 80.34162433330829 0.2653110010031481 0.0033022857479513843

SUCCESSS IVE DONE IT

Before and after: 



What other plots do I want to make?   
This is the list of ones we usually do:

CIS Constant distributions

macros/cis/Public\_Super\_Macro.py --gcals --date 2020-06-21 2020-09-09 --datelabel 'June 21 - September 9, 2020'  
--gcals calls worker CalibDist(datelabel=dfadskfjasd;kf)

TECHNICALLY u can do this with plotcalibfromcool at the very end. Would be nice to do it from here tho.

Can also do it by dumping what is in the final sqlite file. Might be easier to do it that way.

Detector Stability

macros/cis/Public\_Super\_Macro.py --date '06/21/20' '09/09/20' --mean --lowmem --datelabel 'June 21 - September 9, 2020'

--mean triggers flag Detector\_Stability = True

Which calls detectortimestability = DetectorTimeStability(datelabel, startdate, enddate)

DONE in the same way as RMSPlots :D

CIS Constant stability (one year vs another)

RMS/Mean DONE

TUCS Flags