

gemControl

November 13, 2025

What is gemControl?

- Plugin inherited from MUSE that performs low level GEM analysis.
- Decoding of gem data, common mode + pedestal determination, clustering is done here.
- Plotting of raw data, pedestal+common mode subtracted data at both the GEM level (i.e. axis) and APV level done here.
- Finding clusters and fill a branch with the cluster information. No plotting of histograms with cluster information (done separately in gemDiagnostics plugin)

gemControl Recipe

```
[defineHistograms]
    gemControl=defineHistograms

[startup]
    gemControl=startup
    gemControl=initPedestal
    gemControl=setPedestalArrays

[execute]
    gemControl=processraw
    gemControl=determineCommonMode
    gemControl=determinePedestal
    gemControl=determineRMS

[postprocess]
    gemControl=dumpPedestalInfo
    gemControl=applyPedestal
    gemControl=applyRMS

[execute2]
    gemControl=determineCommonMode
    gemControl=readCommonMode
    gemControl=processcorr
    gemControl=processclusters

[finalize]
    gemControl=finalize
```

Functionality described was separated into different functions.

Processraw - sample averaged ADC vs strip for each GEM + sample average for each APV card

Processcorr - same as above but pedestal+cmode subtracted.

Processclusters - finds clusters and stores them to the output

Idea was in principle, one can have different recipes doing different things eg. one just for clustering

So far haven't included any detailed sample dependent code as of yet

gemControl Configuration

[gemControl]

[config]

```
gemnum=initNumberOfGems
apv=associateApvToGem
cmode=selectCmode
sample=initNumberOfSamples
allwords=initAllWords
format=setDataFormat
activeGEM=setActiveGEM
```

[run:0]

```
gemnum=1,4
cmode=2
allwords=0
format=2
sample=6
activeGEM=1,1,0,0
```

```
apv:0=0,0,"X",0
apv:1=0,1,"X",0
apv:2=0,2,"X",0
apv:3=0,3,"X",0
apv:4=0,4,"X",0
apv:5=0,5,"X",0
apv:6=0,6,"X",0
apv:7=0,7,"X",0
apv:8=0,8,"Y",0
apv:9=0,9,"Y",0
apv:10=0,10,"Y",0
apv:11=0,11,"Y",0
apv:12=0,12,"Y",0
```

Tried to make gemControl flexible enough to be able to decode different data structures, handle single vs multi sample, turn on/off GEMs.

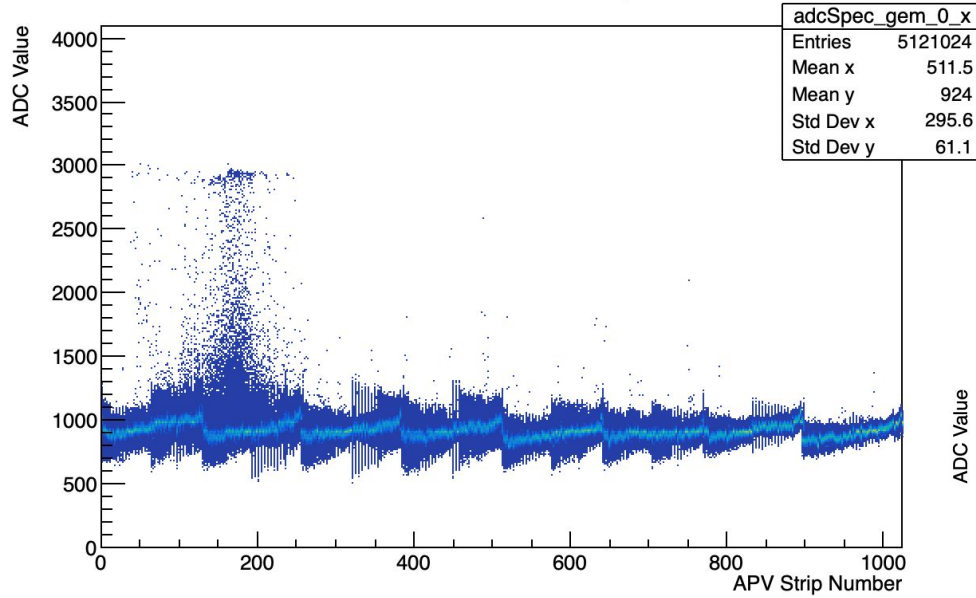
Format = 2 (Current data structure)

Sample = 6 (6 Sample Readout)

activeGEM = 1, 1, 0, 0 (Left arm only)

Apv (GEM id, APV id, Axis, Channel to start)

GEM Left Bottom GEM X spectrum

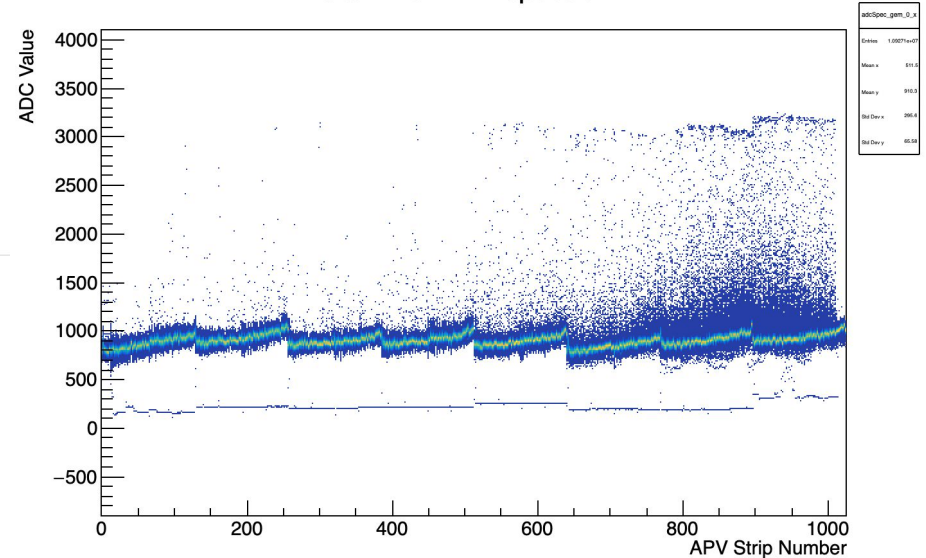


Single sample, single GEM configuration
Data format saves header, APV data +
trailer

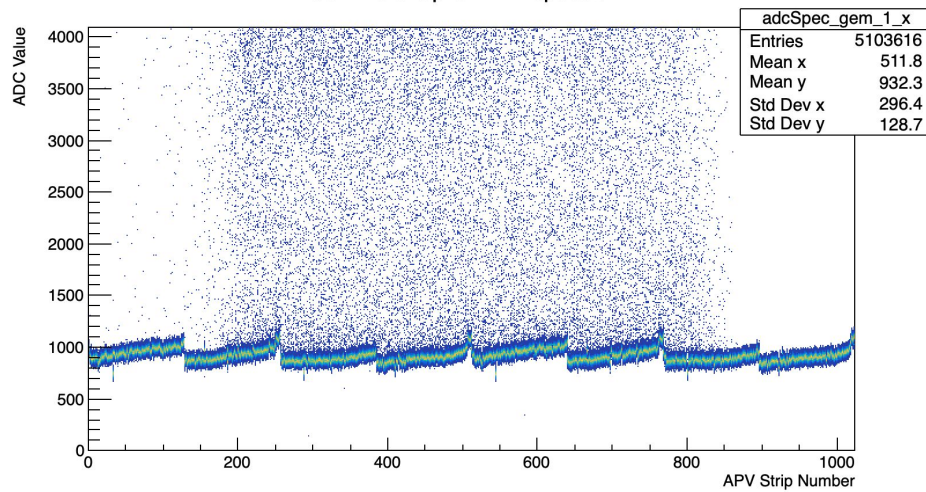
TRIUMF Test Lab
Two GEM, single sample run.

Data structure is different here.

GGLB GEM X spectrum



Gem Left Top GEM X spectrum



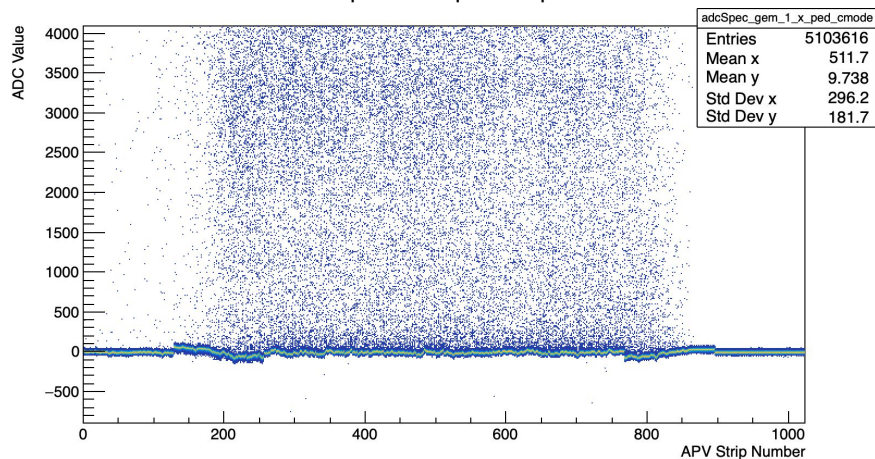
gemControl output from simulation.

Simulation has current data structure, 4 GEM, 6 sample running

Here is the sample averaged ADC

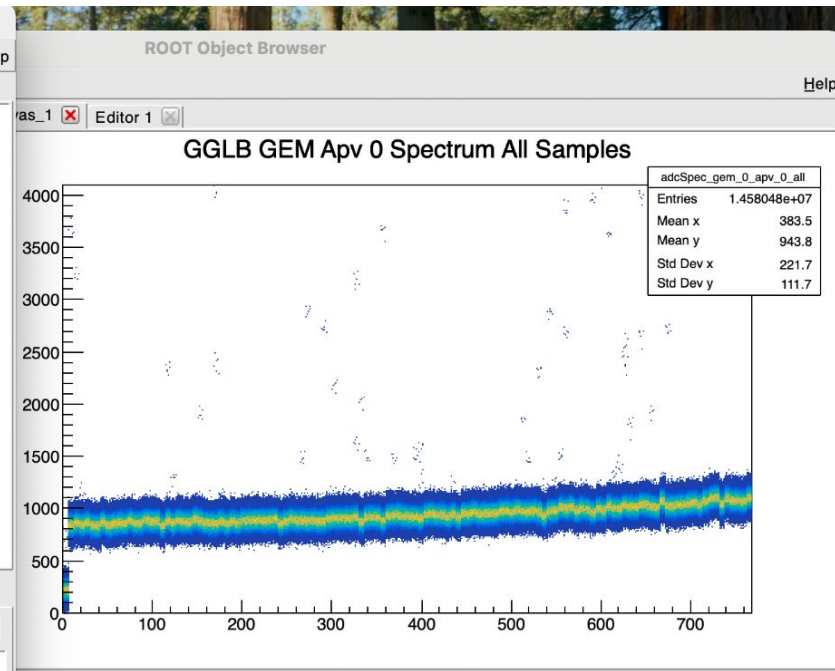
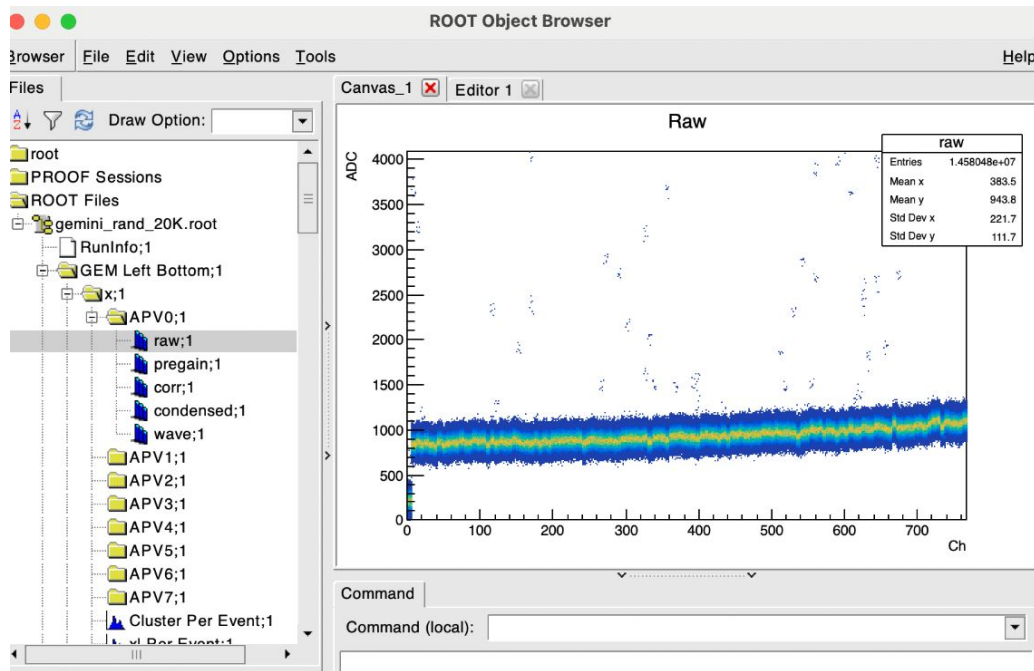
Same outputs for other GEMs.

Gem Left Top GEM X spectrum ped+cmode



To validate the decoding, cross checked with GEMini output

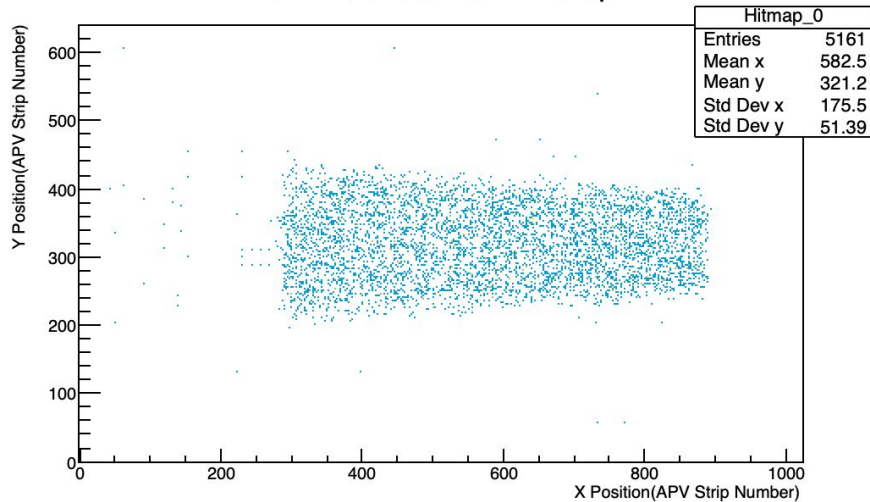
Cross Checks with GEMini



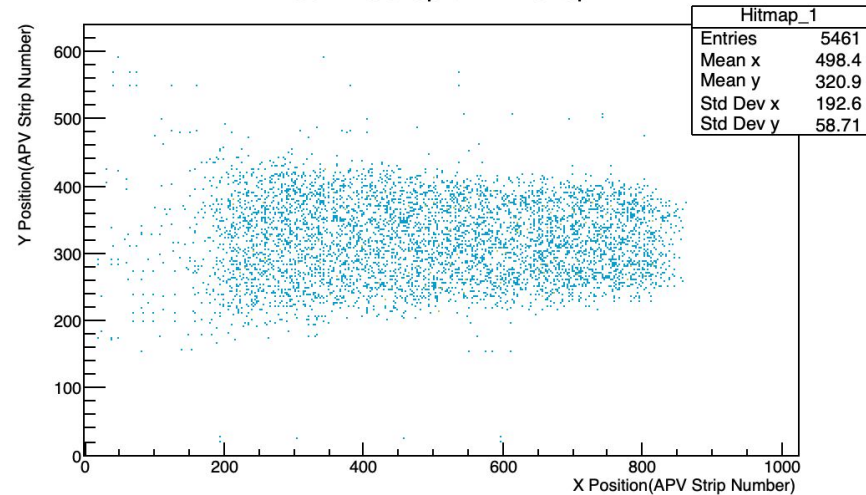
Left is from GEMini, right is from gemControl - All data for all samples for an APV card.

Clustering

GEM Left Bottom GEM Hitmap

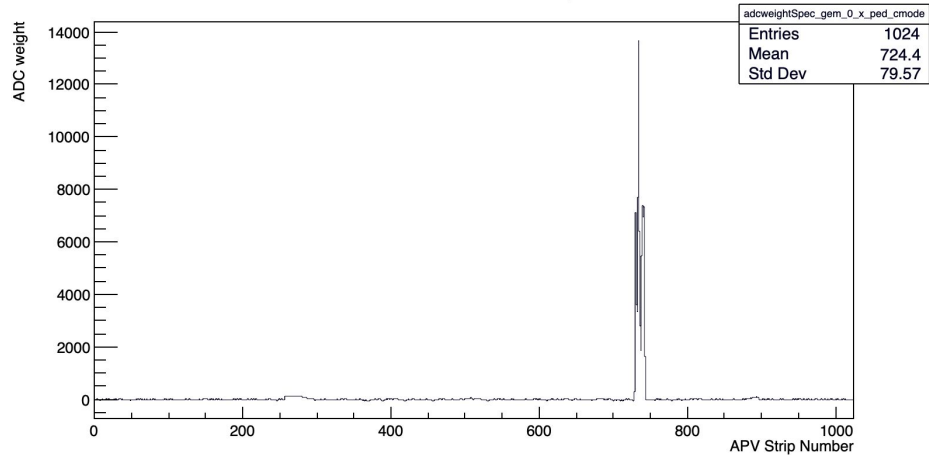


Gem Left Top GEM Hitmap

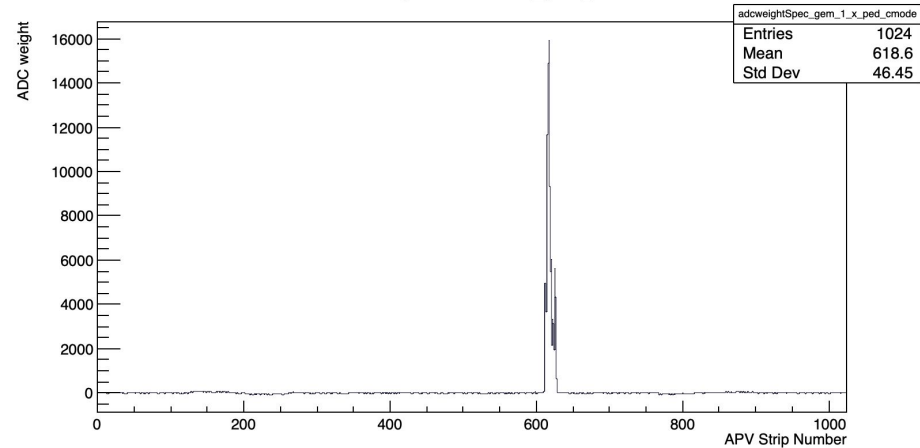


Can use clusters and fill histograms. Example here is from the simulation where we plot the cluster position for each cluster for each axis (all combinations of x+y clusters). This is now in the gemDiagnostics plugin. Can use gemControl or GEMini output for that plugin.

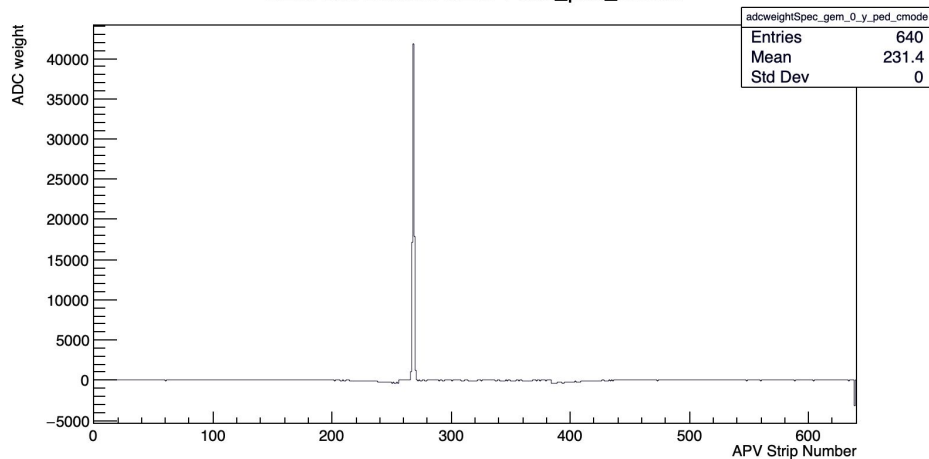
GEM Left Bottom GEM X adc_ped_cmode



Gem Left Top GEM X adc_ped_cmode



GEM Left Bottom GEM Y adc_ped_cmode



Gem Left Top GEM Y adc_ped_cmode

