Channel classification instruction v5

Cheng-Wei Shih, Chia-Ming Kuo

National Central University, Taiwan

2021/7/15

New feature

- New plot in “summary_plot_time.pdf”

- Entry & gaussian_width of each channel are stored in root file.

Procedures

- Create a folder, make sure it has
  - run.sh
  - summary_plot.c
  - calibration_ana_code_multi.c
  - check_chip_prototypeMaximam_new.c
  - All “.dat” files (not .root file), 10 runs is enough.

Example

```
[5202011@chip01 INTT_multi_run_v2_test3]$ ls
NCU_fphx_raw_module_268_20210610-1132_0.dat  NCU_fphx_raw_module_268_20210610-1209_0.dat  NCU_fphx_raw_module_268_20210610-1246_0.dat
NCU_fphx_raw_module_268_20210610-1137_0.dat  NCU_fphx_raw_module_268_20210610-1215_0.dat  calibration_ana_code_multi.c
NCU_fphx_raw_module_268_20210610-1143_0.dat  NCU_fphx_raw_module_268_20210610-1220_0.dat  check_chip_prototypeMaximam_new.c
NCU_fphx_raw_module_268_20210610-1148_0.dat  NCU_fphx_raw_module_268_20210610-1225_0.dat  run.sh
NCU_fphx_raw_module_268_20210610-1153_0.dat  NCU_fphx_raw_module_268_20210610-1231_0.dat  summary_plot.c
NCU_fphx_raw_module_268_20210610-1159_0.dat  NCU_fphx_raw_module_268_20210610-1236_0.dat
NCU_fphx_raw_module_268_20210610-1204_0.dat  NCU_fphx_raw_module_268_20210610-1241_0.dat
[5202011@chip01 INTT_multi_run_v2_test3]$
```

Root version : 6.01

Procedures

- Before run the code, please modify “run.sh”

```
folder_direction="/home/5202011/INTT_cal/INTT_cal_test/INTT_multi_run_v2_test"
number_of_file=15
merge_file_name="aaa_test_summary"
module_ID=2

rm multi_run_status.txt
rm $merge_file_name.root
echo 1
sleep 15


ls *.dat > dat_file.txt
sleep 15

let number_for_final=number_of_file-1



for seed in $(seq 0 $number_for_final)
do
        cp check_chip_prototypeMaximam_new.c check_chip_prototypeMaximam_new_copy.c
        sed -i "s/data_index/${seed}/g" check_chip_prototypeMaximam_new_copy.c
        root -l -b -q check_chip_prototypeMaximam_new_copy.c\($module_ID\)
        rm check_chip_prototypeMaximam_new_copy.c
        sleep 15
done

ls *.root > total_file.txt
sleep 15

for seed in $(seq 0 $number_for_final)
do
    cp calibration_ana_code_multi.c calibration_ana_code_multi_copy.c
    sed -i "s/data_index/${seed}/g" calibration_ana_code_multi_copy.c
    root -l -b -q calibration_ana_code_multi_copy.c\(\"$folder_direction\",$module_ID,true,false,0,true,false,false,true\)
    rm calibration_ana_code_multi_copy.c
    sleep 15
done

sleep 15
hadd $merge_file_name.root */*_summary.root

sleep 15
root -l -b -q summary_plot.c\($number_of_file,\"$folder_direction\",\"$merge_file_name\"\)
```

Number of root files

Folder direction

Name of final merged root file

Module ID

Variables descriptions are in next slide

Merge all output files

Procedures

```
#Variable of calibration_ana_code_multi.c :
#Variable 1 : TSting,  folder direction
#Variable 2 : int,   port_ID,
#Variable 3 : bool,  output the adc-ampl plot for each channel (should be true)
#Variable 4 : bool,  original unbond channel check             (should be false)
#Variable 5 : int,   overall ampl noise level check            (0 can be good)
#Variable 6 : bool,  output offset ampl distribution plot for each channel (should be true)
#Variable 7 : bool,  cout unbonded channel status @ without bias run. (should be false)
#Variable 8 : bool,  cout wider gaus width channel @ with bias run.   (should be false)
#Variable 9 : bool,  output multi_run_status.txt               (should be true)
```

Variable 3 & 6 : if “false” -> no plots created, can be faster

- After modification, please run “run.sh” → “. run.sh”
  - It takes ~ 15 mins to finish 20 root files.

Procedures

- Two files and a lot folders will be created after the run
  - Two files : aaa_test_summary.root & multi_run_status.txt
  - One “.root file” will be created in each folder of each run to record the status of the problematic channels.

- 3 final summary plots and un-functional channels status will be created and printed, examples are shown in next slide.
  - 3 final summary plots :
    - Summary_noise.pdf
    - Summary_entry.pdf
    - Summary_plots_time.pdf

Final overall results

Channels failed in noise criteria

```
Noise channel found, chip : 2   channel : 0     failed times : 13/20    ratio : 0.65
Noise channel found, chip : 7   channel : 12    failed times : 20/20    ratio : 1
Noise channel found, chip : 19  channel : 36    failed times : 20/20    ratio : 1
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Noise channel, chip : 2 channel : 0
 failed in file index : 2       file name : fphx_raw_20210428-1320_0.root, gaus width : 46.9425
 failed in file index : 3       file name : fphx_raw_20210428-1321_0.root, gaus width : 78.376
 failed in file index : 4       file name : fphx_raw_20210428-1322_0.root, gaus width : 4.47558
 failed in file index : 5       file name : fphx_raw_20210428-1324_0.root, gaus width : 4.59855
 failed in file index : 6       file name : fphx_raw_20210428-1331_0.root, gaus width : 75.7756
 failed in file index : 7       file name : fphx_raw_20210428-1332_0.root, gaus width : 32.8639
 failed in file index : 8       file name : fphx_raw_20210428-1333_0.root, gaus width : 38.3274
 failed in file index : 11      file name : fphx_raw_20210428-1654_0.root, gaus width : 147.086
 failed in file index : 12      file name : fphx_raw_20210428-1656_0.root, gaus width : 127.588
 failed in file index : 14      file name : fphx_raw_20210428-1659_0.root, gaus width : 4.8683
 failed in file index : 15      file name : fphx_raw_20210428-1700_0.root, gaus width : 4.15262
 failed in file index : 17      file name : fphx_raw_20210428-1703_0.root, gaus width : 4.41939
 failed in file index : 19      file name : fphx_raw_20210428-1706_0.root, gaus width : 33.9221
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

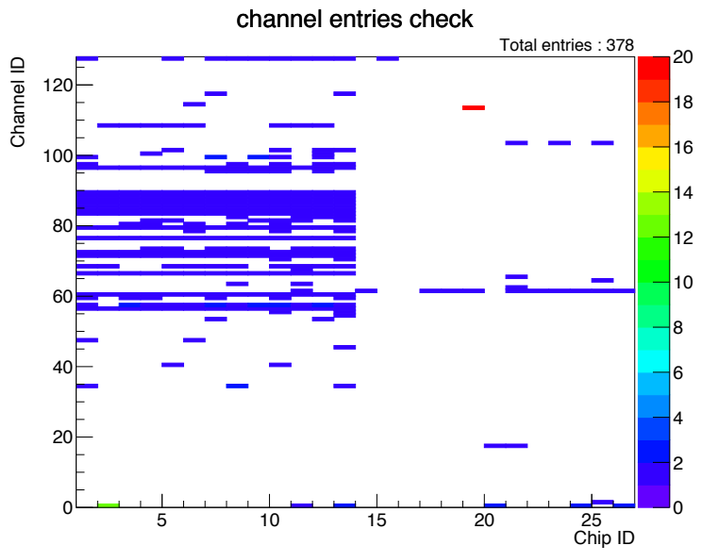
Channels failed in entry criteria

```
Weird entries found, chip : 2   channel : 0     failed times : 12/20    ratio : 0.6
Weird entries found, chip : 19  channel : 113   failed times : 20/20    ratio : 1
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Bad entries channel, chip : 2 channel : 0
 bad in file index : 2  file name : fphx_raw_20210428-1320_0.root, entries : 4410
 bad in file index : 3  file name : fphx_raw_20210428-1321_0.root, entries : 5820
 bad in file index : 5  file name : fphx_raw_20210428-1324_0.root, entries : 645
 bad in file index : 6  file name : fphx_raw_20210428-1331_0.root, entries : 7557
 bad in file index : 7  file name : fphx_raw_20210428-1332_0.root, entries : 2104
 bad in file index : 8  file name : fphx_raw_20210428-1333_0.root, entries : 4062
 bad in file index : 11 file name : fphx_raw_20210428-1654_0.root, entries : 20615
 bad in file index : 12 file name : fphx_raw_20210428-1656_0.root, entries : 17784
 bad in file index : 14 file name : fphx_raw_20210428-1659_0.root, entries : 569
 bad in file index : 15 file name : fphx_raw_20210428-1700_0.root, entries : 422
 bad in file index : 17 file name : fphx_raw_20210428-1703_0.root, entries : 467
 bad in file index : 19 file name : fphx_raw_20210428-1706_0.root, entries : 2125
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

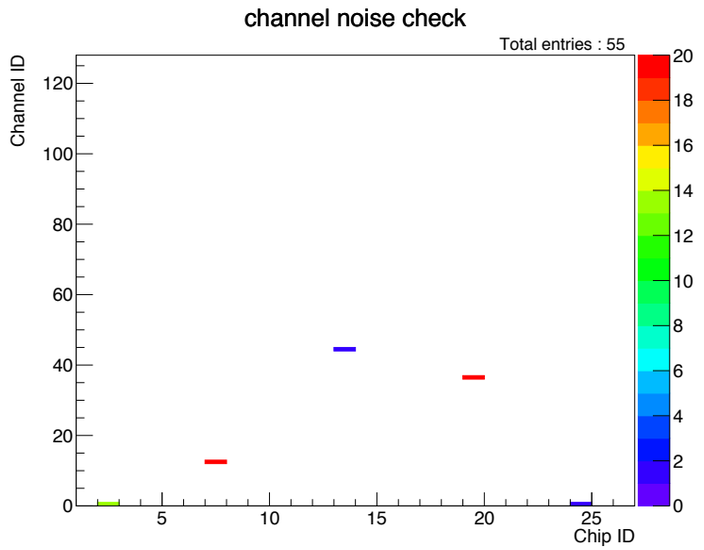
```
======================== Final summary ========================
# of noise channels : 3
# of bad entry chan : 2
# of double counting: 1

Total bad channels : 4
Overall good channel ratio : 3324 / 3328 = 99.880%
======================== Final summary ========================
```

Final result

Summary_entry.pdf



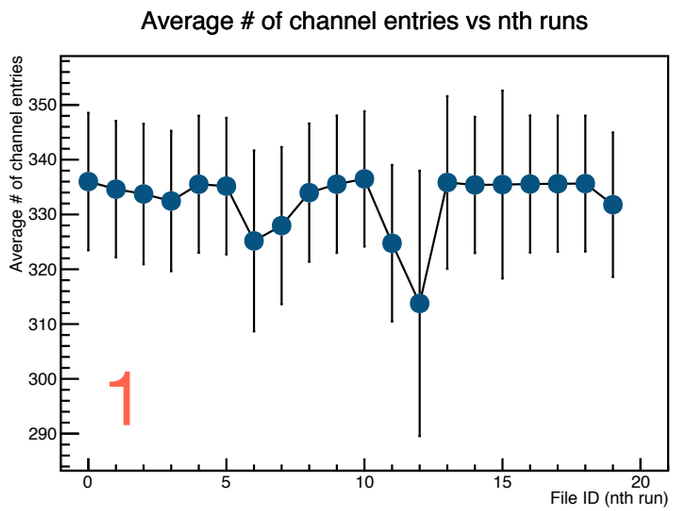
Summary_noise.pdf

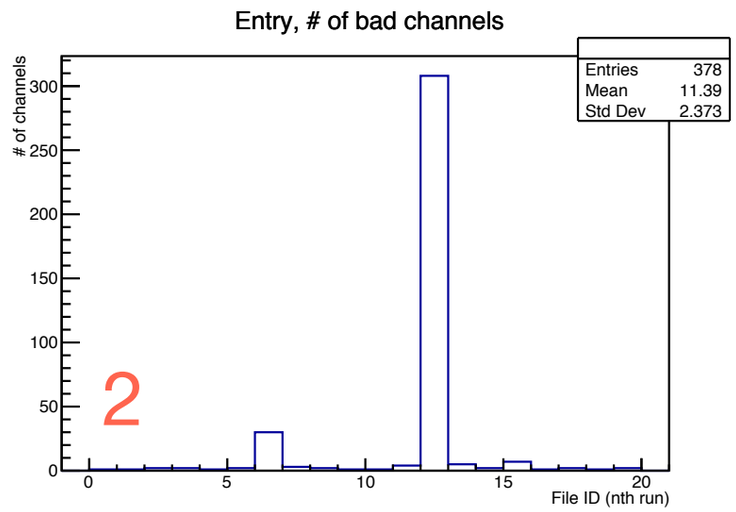


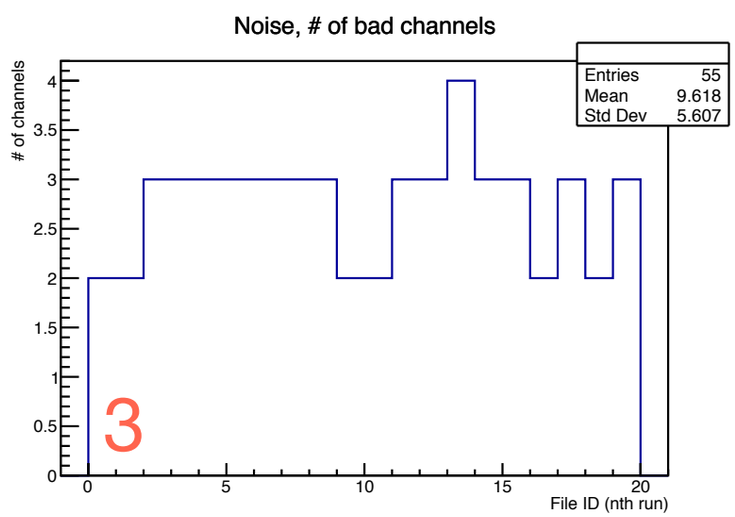
Principle : the less entries in the plot, the better performance it is

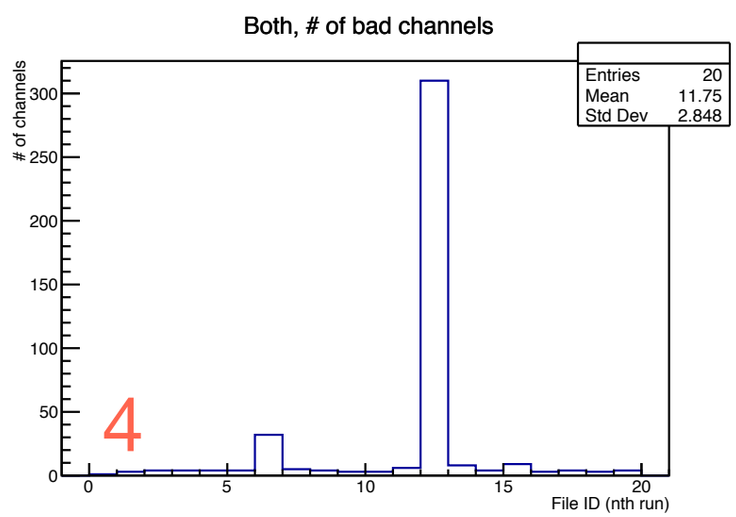
Final overall results : Summary_plots_time.pdf

- Summary_plots_time.pdf contents 4 plots
- Plot 1 : Average # of channel entries vs file ID
- Plot 2 : # of channels fail in entry criterion vs file ID.
- Plot 3 : # of channels fail in noise criterion vs file ID.
- Plot 4 : # of bad channels criterion vs file ID. (No double counting)



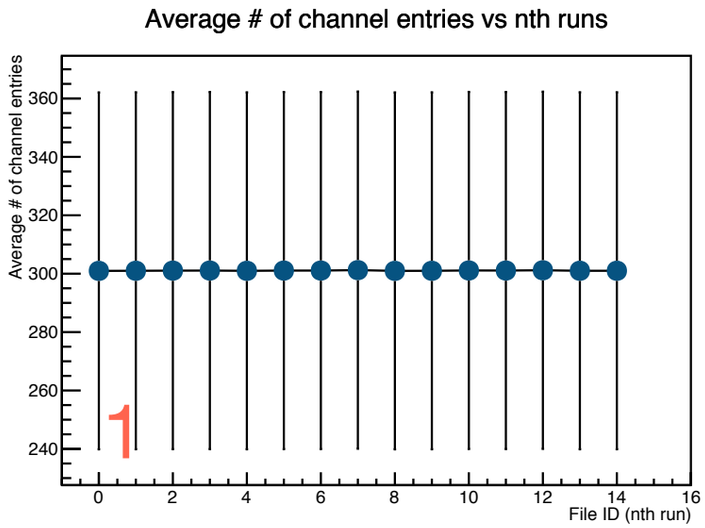


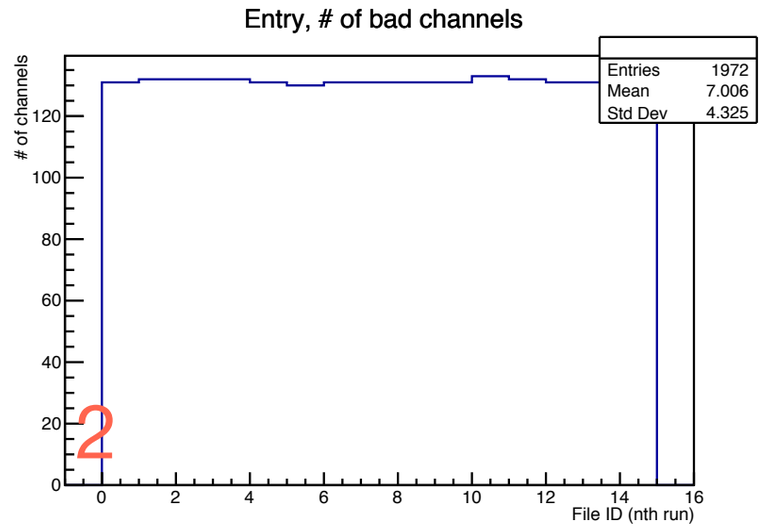


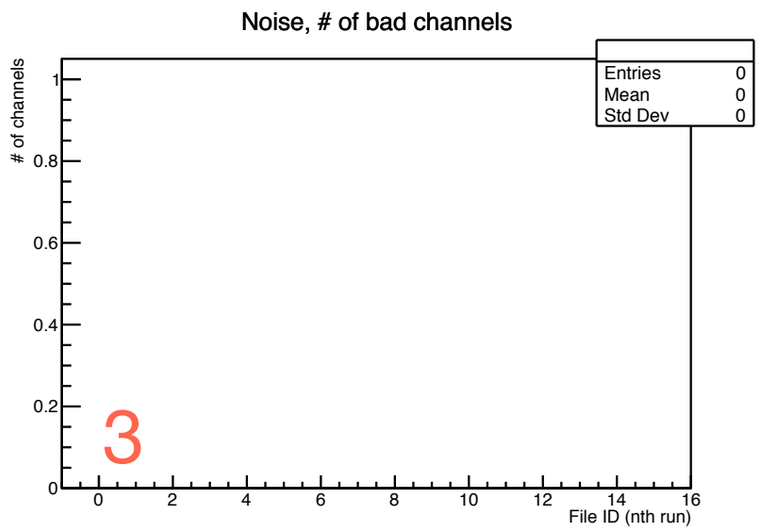


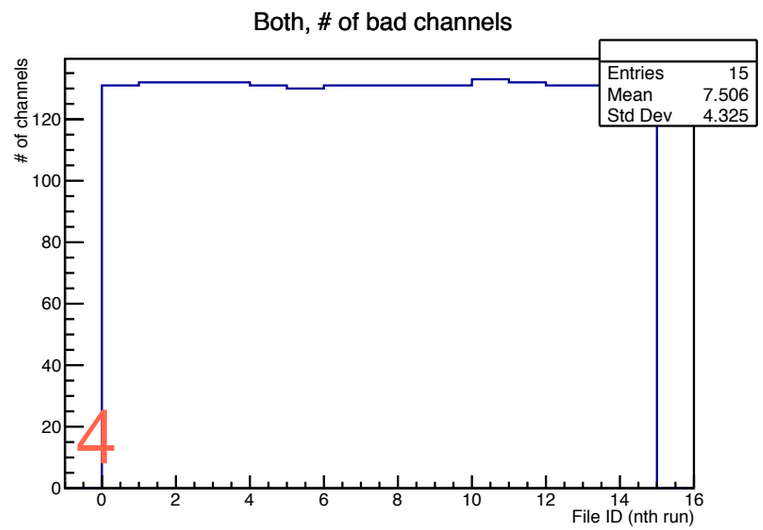
Final overall results : Summary_plots_time.pdf

- If the Testbench system is stable, the plot 1 should be a horizontal line.





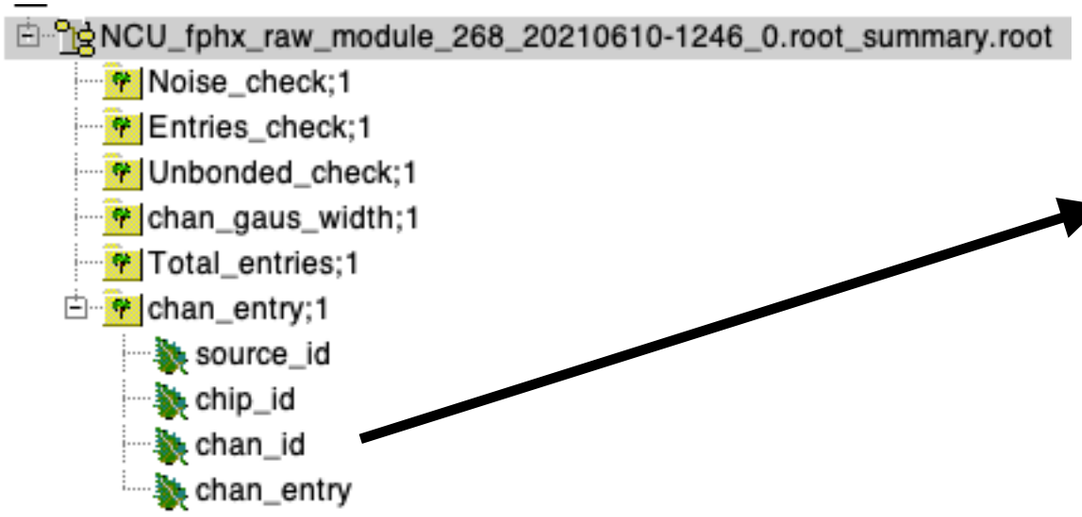




Root file of introduction

- Each “.dat file” corresponds to one output root file.
- Example :
  “NCU_fphx_raw_module_268_20210610-1246_0.dat”
  corresponds to
  “NCU_fphx_raw_module_268_20210610-1246_0.root_summary.root”
  in folder
  “folder_NCU_fphx_raw_module_268_20210610-1246_0.root”

- Tree “chan_gaus_width” : the gaus_width of each channel (entry : 3328)
- Tree “chan_entry” : # of events of each channel (entry : 3328)



source_id = file_id

Row	source_id	chip_id	chan_id	chan_entr
0	14	1	0	336
1	14	1	1	299
2	14	1	2	320

Criteria of bad channel

- # of input files > 10 files.

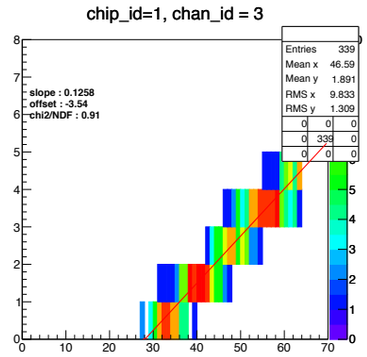
- Single channel is tested by 2 criteria :
  - Entry :
    - If channel with entry < 280 OR entry > 400.
    - If it happens more than 2 times in all calibration runs -> Bad.
  - Noise :
    - If channel with fit gaussian width > 4.
    - If it happens more than 2 times in all calibration runs -> Bad.

- If channel fails in both criteria, it is counted by 1 only.
  - No double counting.

Plot descriptions

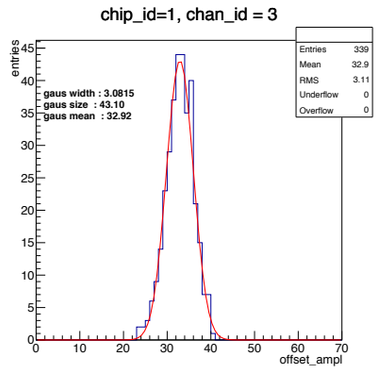
There are a lot of plots in each folder created by “. run.sh”. Here I introduce some plots I frequently check

- chipX_detail_ampladc.pdf
  - Ampl - ADC response for single channel
  - 128 pages for 128 channels

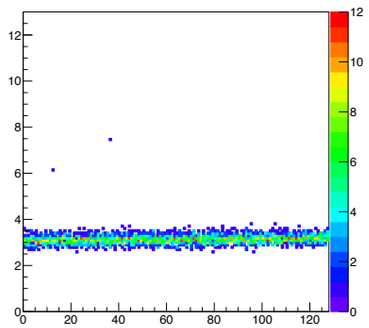


X axis : ampl,
Y axis : ADC

- chipX_detail_amploffset.pdf
  - Ampl width distribution after offset, single channel.
  - 128 pages for 128 channels



- ampl_adc_width_detial_TH2.pdf
  - Overall gaus width status of half-ladder.
  - Entries of plot : 128*26 = 3328

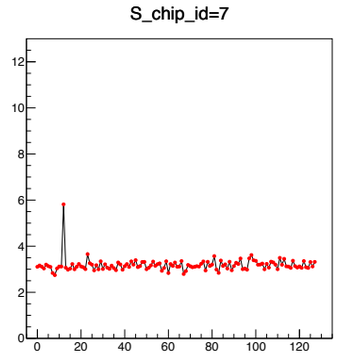


X axis : Channel,
Y axis : Gaussian width

Plot descriptions

There are a lot of plots in each folder created by “. run.sh”. Here I introduce some plots I frequently check

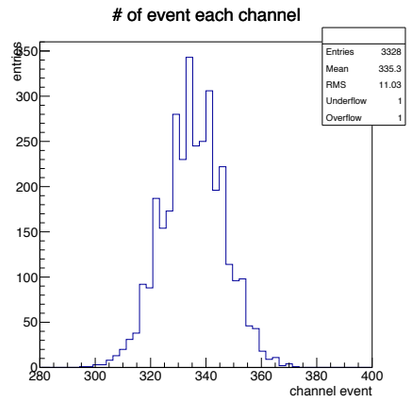
- ampl_adc_width_detial.pdf
  - Overall gaus width status of each chip
  - 26 pages in total



# of data points : 128

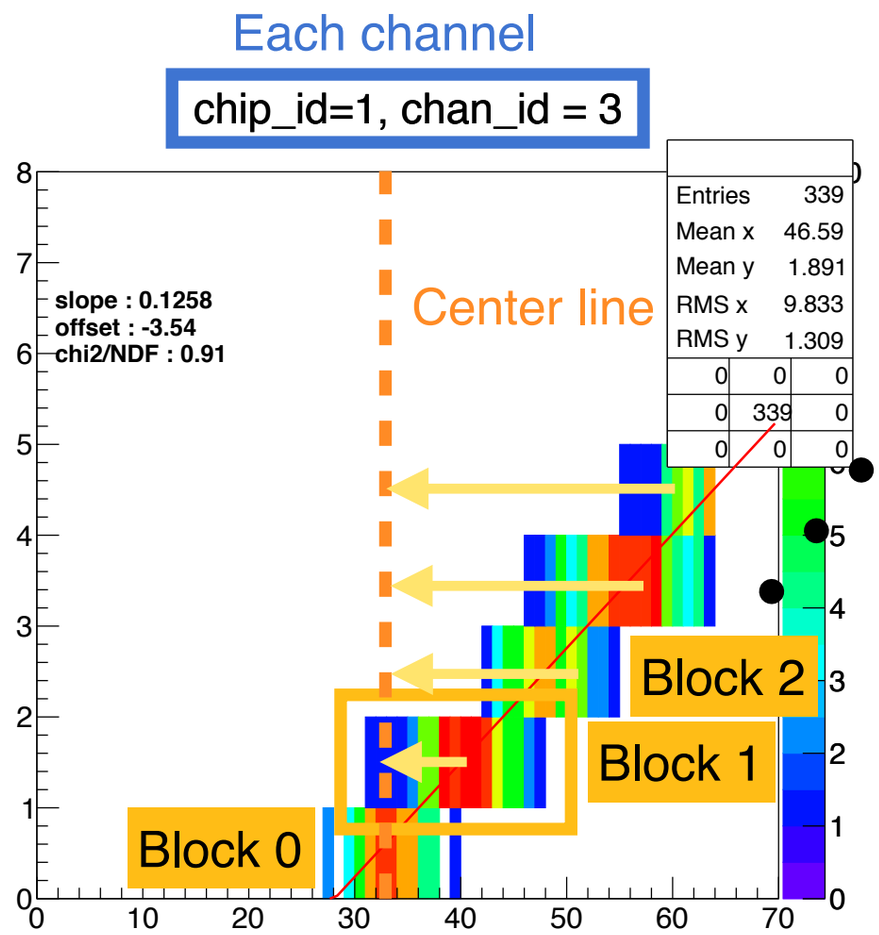
X axis : Channel,
Y axis : Gaussian width

- channel_entries.pdf
  - Distribution of # of events of each channel
  - Entries : 26*128 = 3328

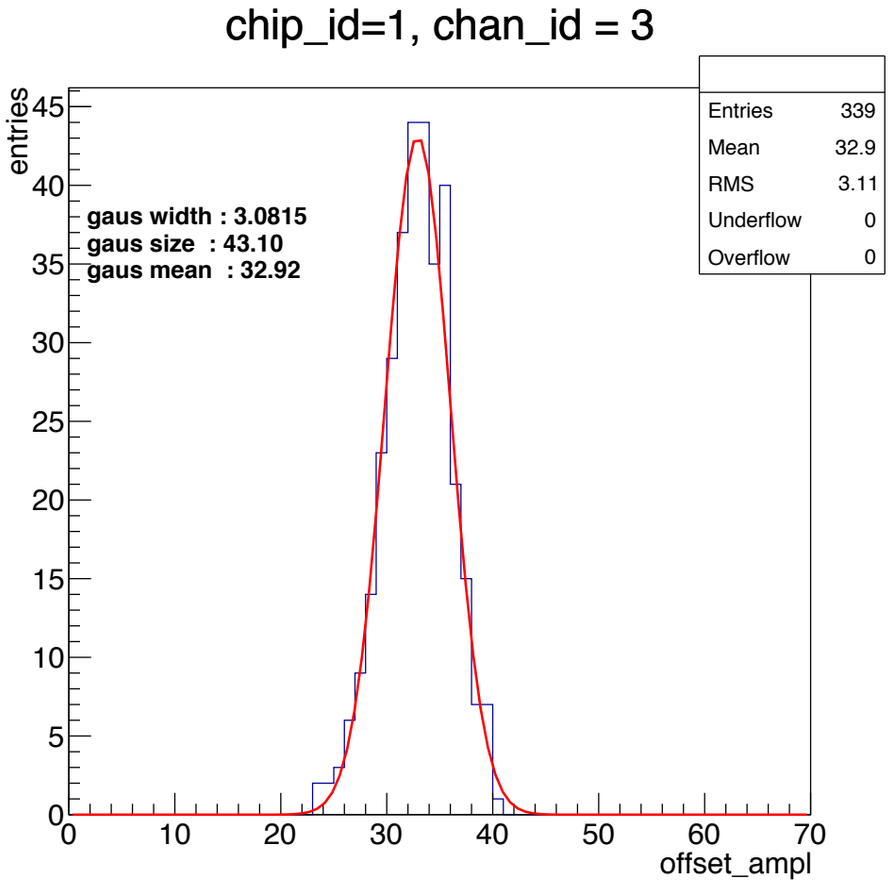


Algorithm introduction

- For each channel of each chip :
  - Center line : mean of events in “Block 0” (ADC==0)
  - Center of the rest blocks are panned to center line.
    - Amount of movement : Mean of each block - center line
  - Each event is filled in TH1F, and fit with gaussian.



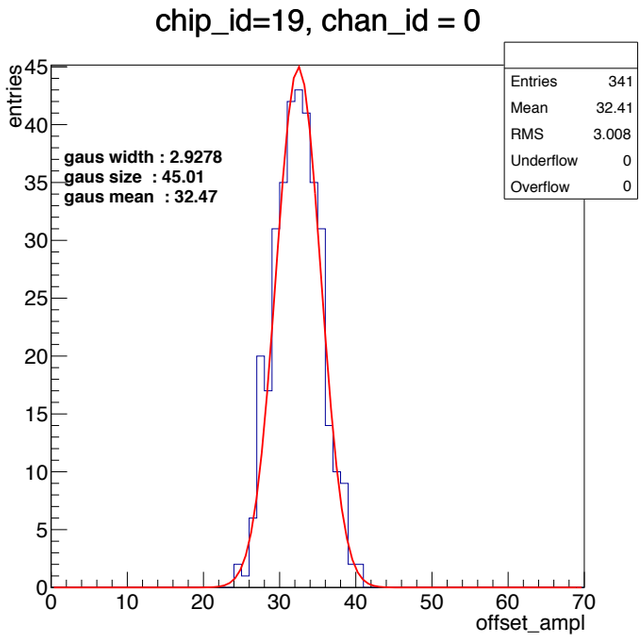
X axis : amp,
Y axis : ADC

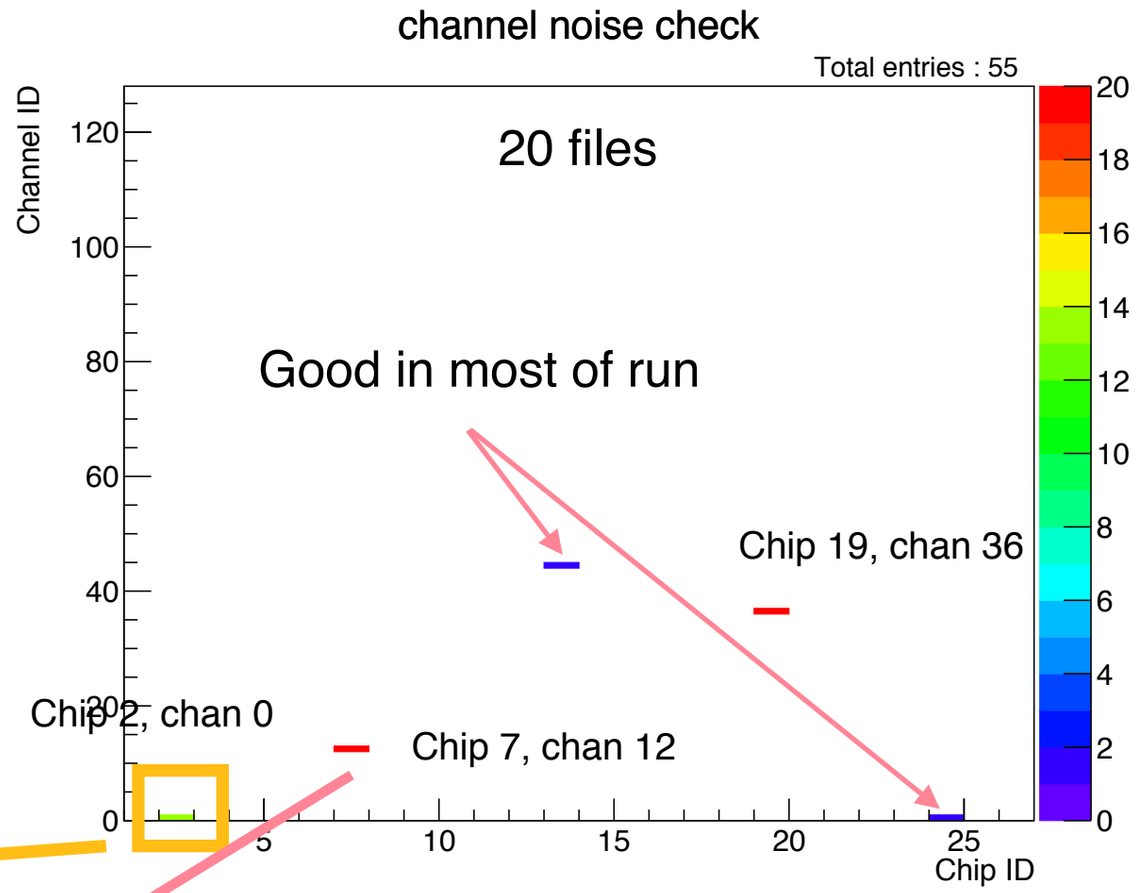


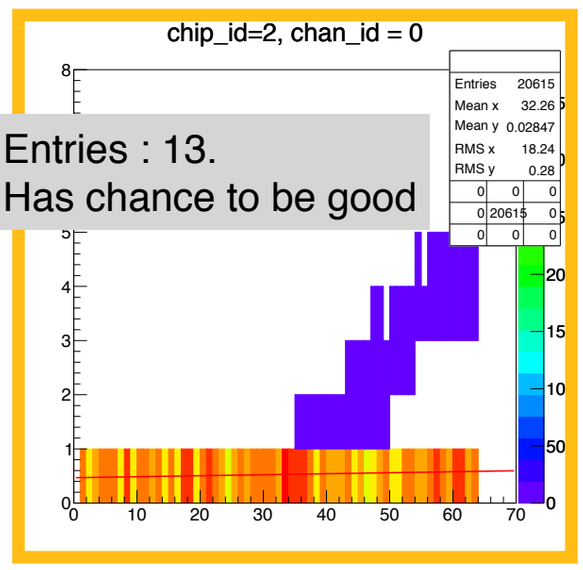
Algorithm introduction

- Update of my algorithm, 2 criteria : noise and entries

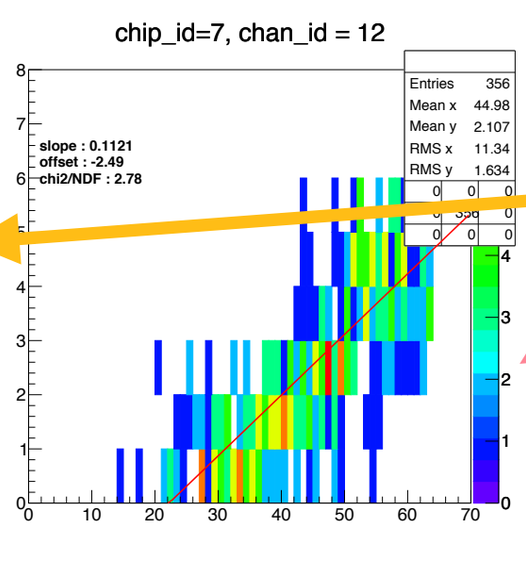
Gaus width > 4 will be shown in right plot







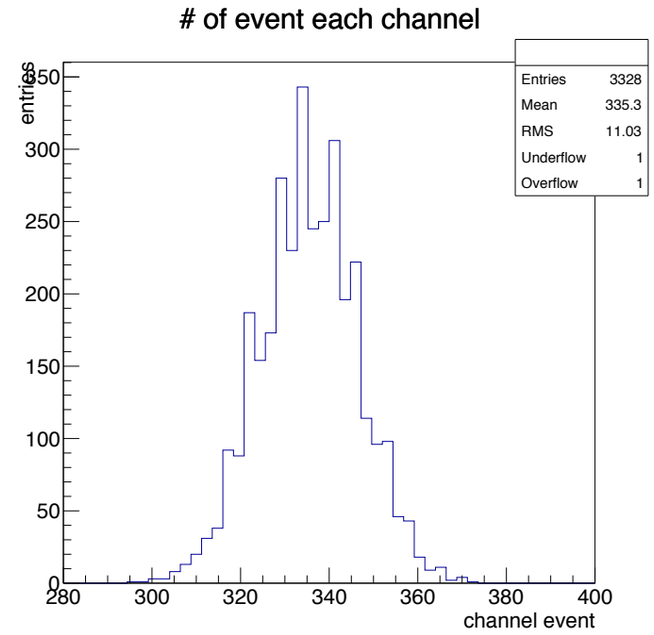
Entries : 13.
Has chance to be good



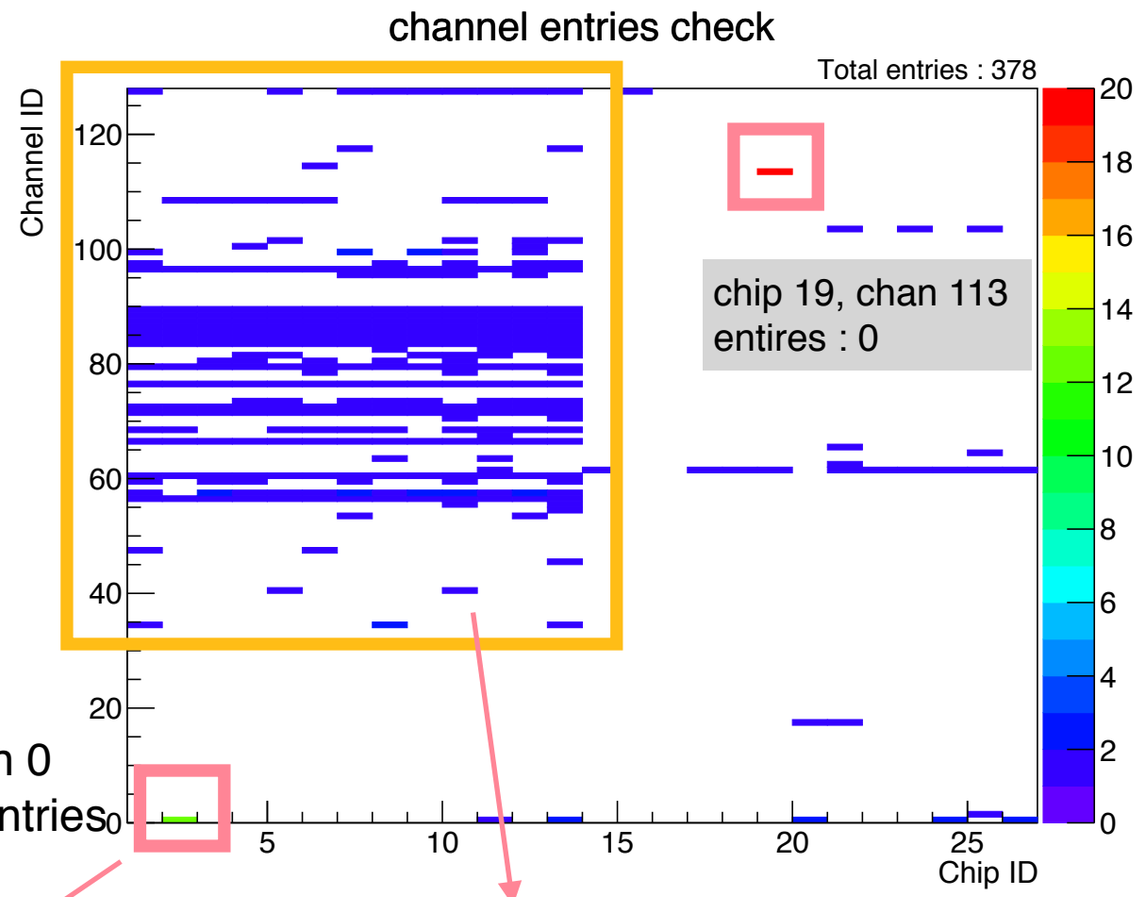
The less entries in plot, better performance it is.

Algorithm introduction

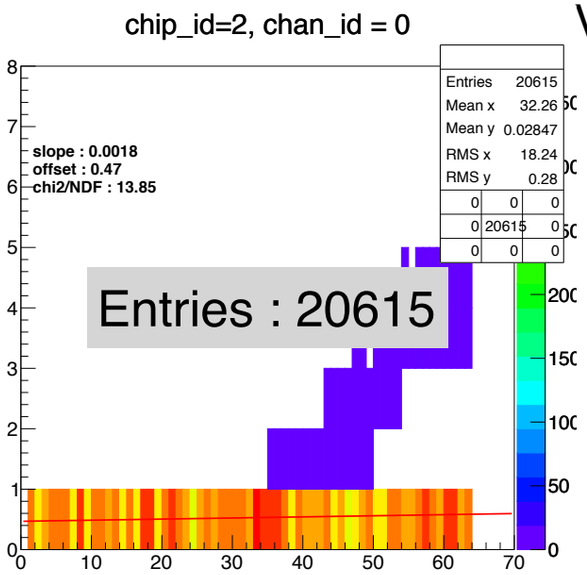
of event of each channel ~ 330

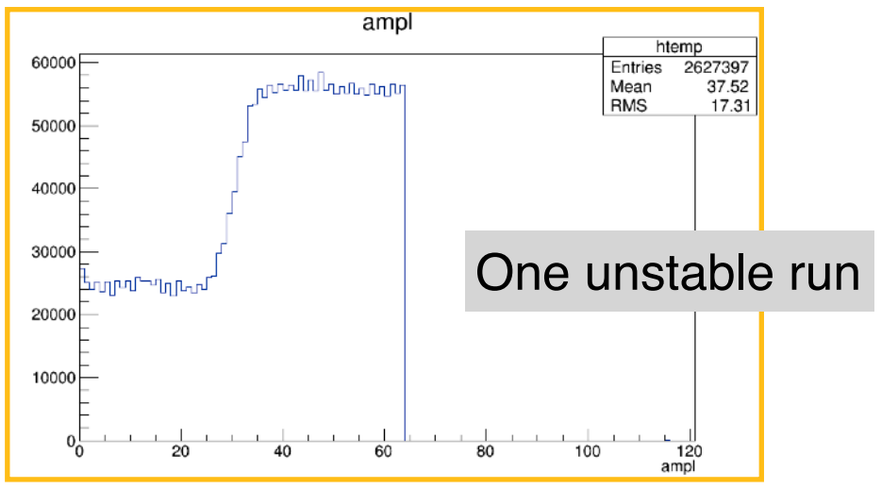


# of event > 400 or < 280 will be filled in the plot



chip 2, chan 0
Very high entries

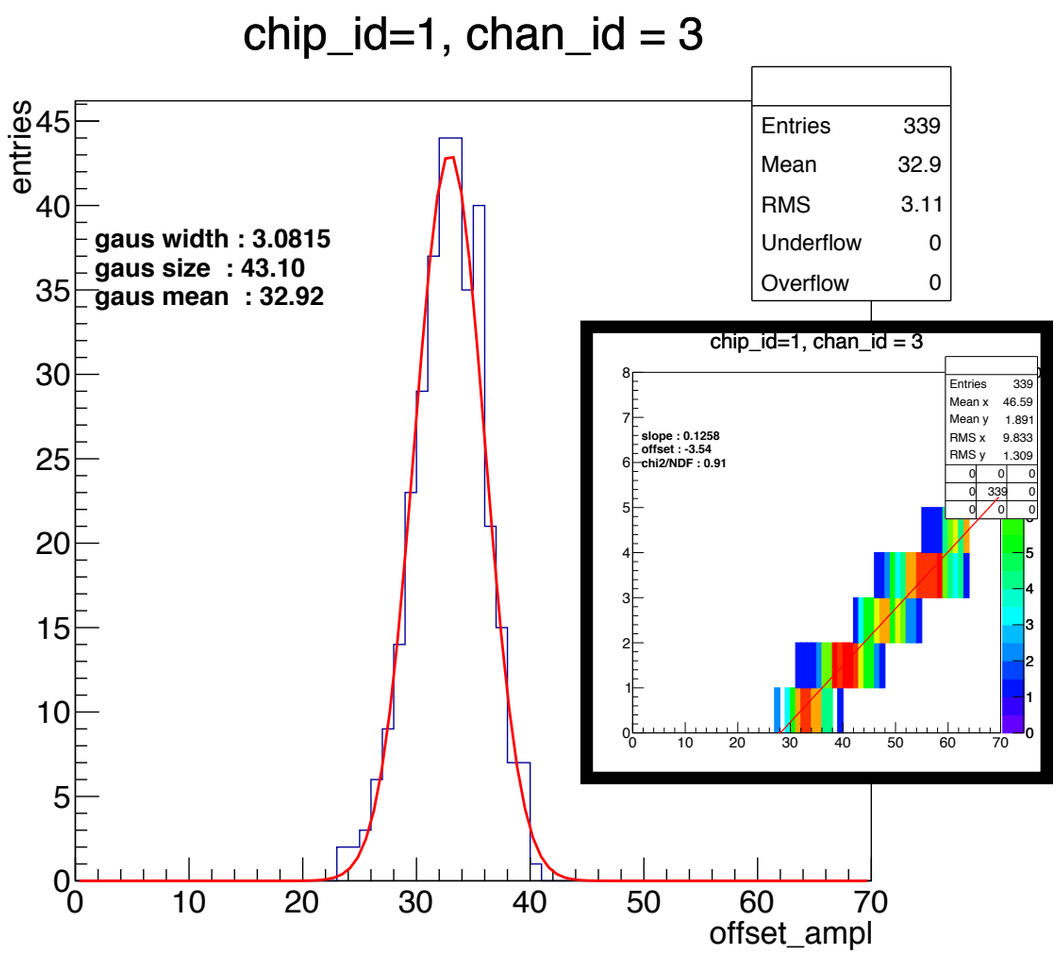




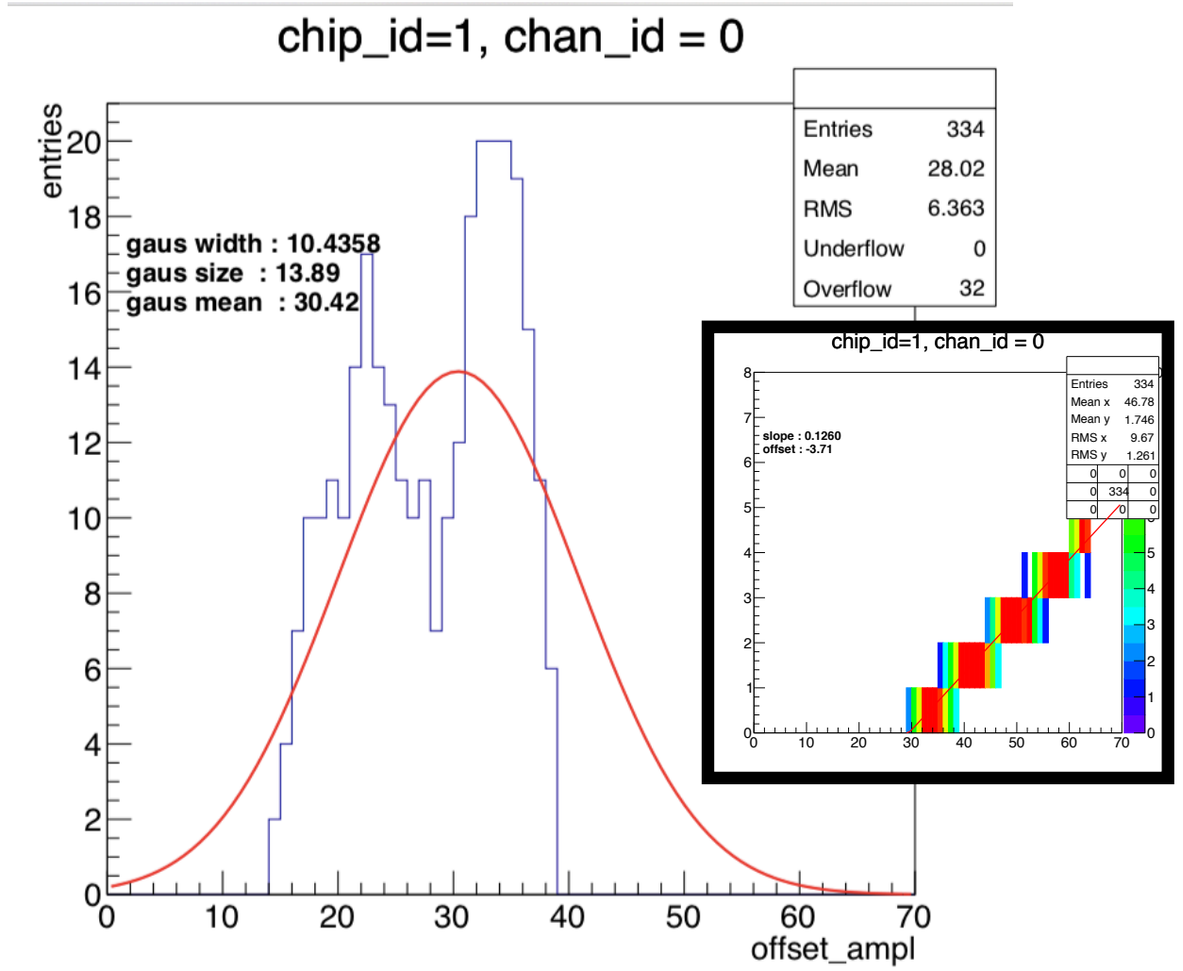
Back up

Attention !

- When you check “chipX_detail_amploffset.pdf”, if the distribution is not reasonable (For example : double or triple peaks, left plot). Please let me know.
- It is a bug, and has been solved already.



Correct one



Bug case,
The amtpl-adc distribution is good, but it has a bug when filling the event in TH1F