ProtoDUNE HD Offline Data Processing

Barnali Chowdhury, Wenqiang Gu, Jay Jo, Michael Kirby, Lino Gerlach
Argonne National Laboratory & Brookhaven National Laboratory

The Second Wire-Cell Reconstruction Summit

April 10th, 2024

Offline Data Processing for ProtoDUNE HD

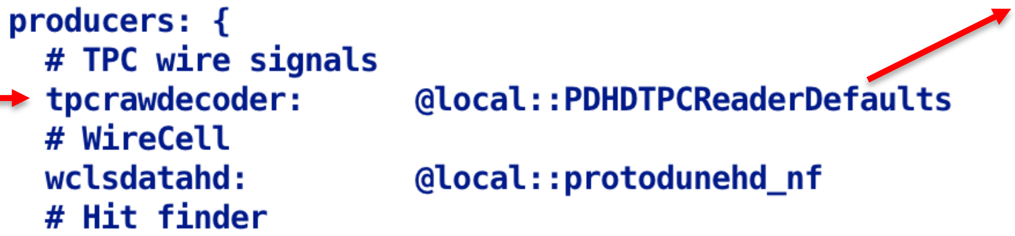
- The ProtoDUNE SP reco strategy runs "Dataprep" module which writes full (i.e. no ROI filtering) processed waveforms as recob:Wire to the event data store
  - "larwirecell" provides recob:Wire to WireCell
  - which are used as input to the WCT (wirecell) module. WCT does deconvolution and ROI finding (and more) writing another recob::Wire container to be used in hit finding.

- What changed between ProtoDUNE SP and now?
  - DAQ started taking data in HDF5 format
  - We want to move away from using DataPrep

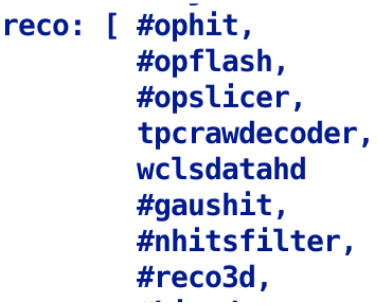
- Goal is
  - to replace DataPrep with WireCell
  - to move Dataprep functionality into Wire-Cell. This means communication with the tpc decoder and all ADC mitigations, noise removal, and pedestal finding etc.
  - to develop WCT Noise Filter for HD and integrate it with tpc raw decoder.
  - to develop PDHD reconstruction chain and put into production.

Offline Data Processing for ProtoDUNE HD

- Introducing HD TPC decoder as the first stage of reconstruction chain
- Fhicl file aimed to run officially for upcoming PDHD data reconstruction
    - **/exp/dune/app/users/barnali/ProtoDUNE2/srcs/dunesw/fcl/protodunhd/reco/ testing_for_standard_reco_protoduneHD.fcl**



```
producers: {
  # TPC wire signals
  tpcrawdecoder:       @local::PDHDTPCReaderDefaults
  # WireCell
  wclsdatahd:          @local::protodunehd_nf
  # Hit finder
```



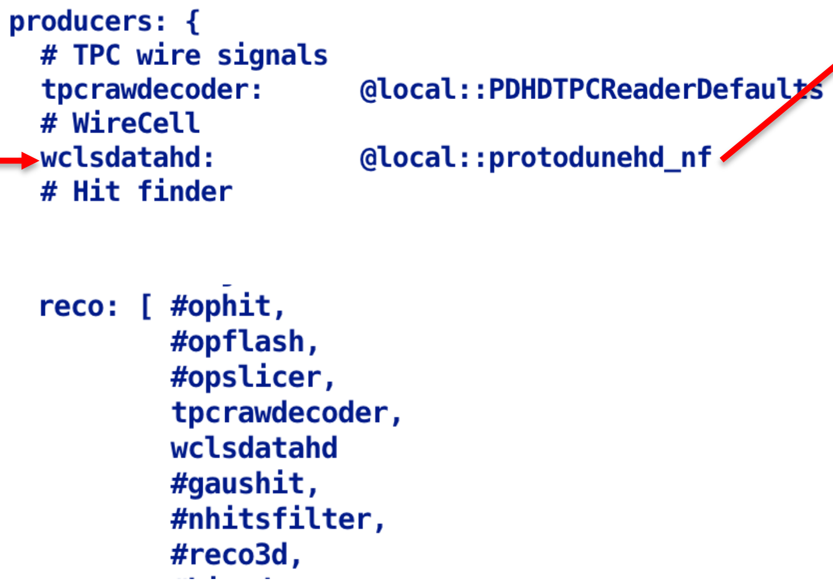
```
reco: [ #ophit,
        #opflash,
        #opslicer,
        tpcrawdecoder,
        wclsdatahd
        #gaushit,
        #nhitsfilter,
        #reco3d,
```

```
PDHDTPCReaderDefaults:
{
  module_type: "PDHDTPCReader"
  InputLabel:  ”tpcrawdecoder:daq"
  OutputInstance: "daq"
  APAList:     [ 1, 2, 3, 4 ]
  DecoderToolParams:
@local::PDHDDataInterfaceWIB3Defaults
}
```

The snippet above locates the input raw::RawDigit collection in the art::Event by raw_input_label: "tpcrawdecoder:daq”.

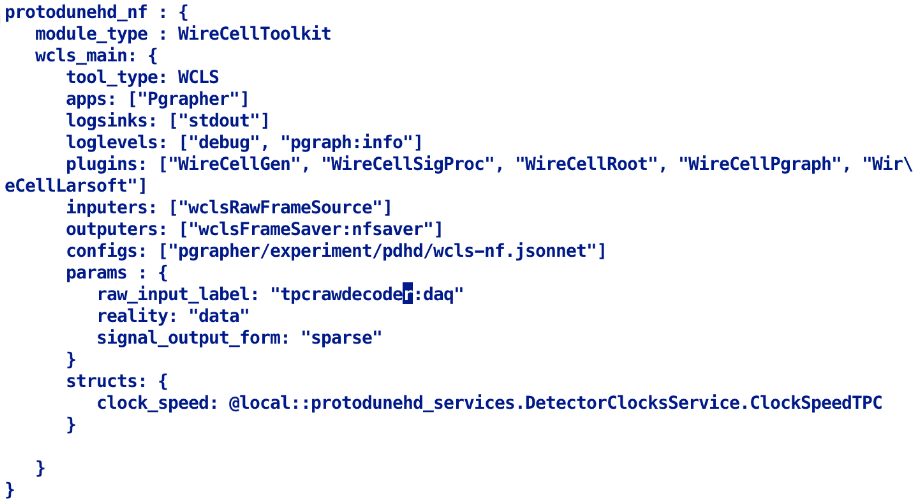
Offline Data Processing for ProtoDUNE HD

- Wirecell configuration for reading “RawDigits”
- Configuration ”protodunehd_nf” is added in https://github.com/DUNE/dunereco/blob/develop/dunereco/DUNEWireCell/wirecell_dune.fcl#L143
  - It assumes a data product of RawDigit with label tpcrawdecoer:daq.
  - Only configured for noise filtering (nf)



```
producers: {
  # TPC wire signals
  tpcrawdecoder:      @local::PDHDTPCReaderDefaults
  # WireCell
  wclsdatahd:         @local::protodunehd_nf
  # Hit finder


  reco: [ #ophit,
          #opflash,
          #opslicer,
          tpcrawdecoder,
          wclsdatahd
          #gaushit,
          #nhitsfilter,
          #reco3d,
```



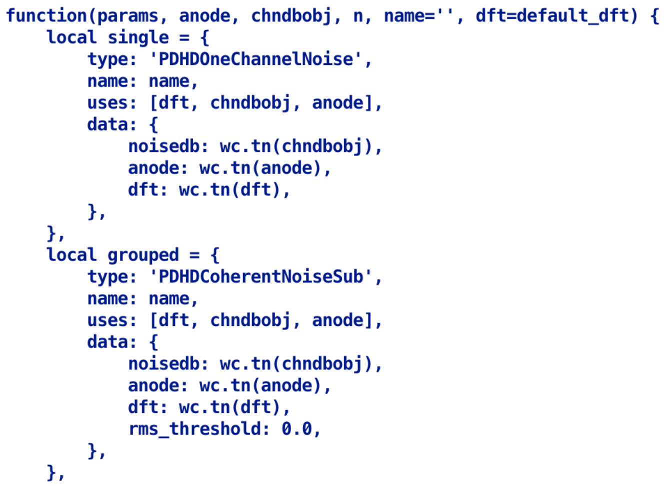
```
protodunehd_nf : {
   module_type : WireCellToolkit
   wcls_main: {
      tool_type: WCLS
      apps: ["Pgrapher"]
      logsinks: ["stdout"]
      loglevels: ["debug", "pgraph:info"]
      plugins: ["WireCellGen", "WireCellSigProc", "WireCellRoot", "WireCellPgraph", "Wir\
eCellLarsoft"]
      inputers: ["wclsRawFrameSource"]
      outputers: ["wclsFrameSaver:nfsaver"]
      configs: ["pgrapher/experiment/pdhd/wcls-nf.jsonnet"]
      params : {
         raw_input_label: "tpcrawdecoder:daq"
         reality: "data"
         signal_output_form: "sparse"
      }
      structs: {
         clock_speed: @local::protodunehd_services.DetectorClocksService.ClockSpeedTPC
      }

   }
}
```

Offline Data Processing for ProtoDUNE HD

- We also modify nf.jsonnet

  /exp/dune/app/users/barnali/ProtoDUNE2_HD/srcs/dunereco/dunereco/DUNEWireCell/pdhd/nf.jsonnet



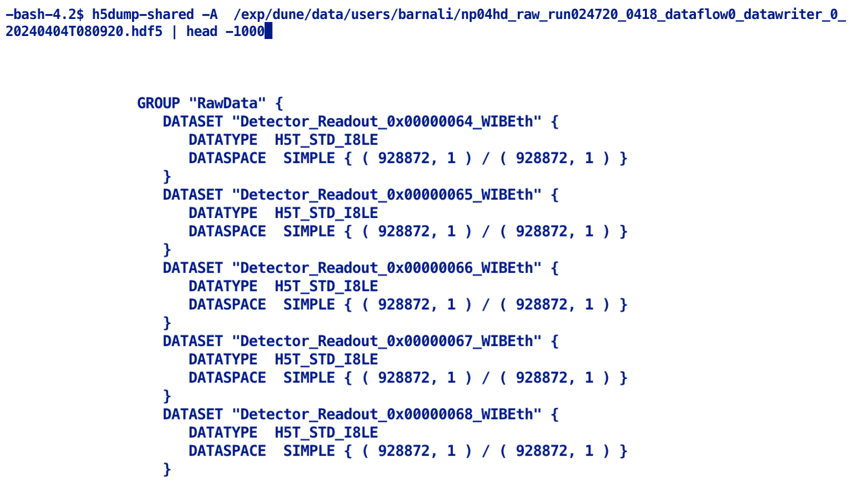
```
function(params, anode, chndbobj, n, name='', dft=default_dft) {
    local single = {
        type: 'PDHDOneChannelNoise',
        name: name,
        uses: [dft, chndbobj, anode],
        data: {
            noisedb: wc.tn(chndbobj),
            anode: wc.tn(anode),
            dft: wc.tn(dft),
        },
    },
    local grouped = {
        type: 'PDHDCoherentNoiseSub',
        name: name,
        uses: [dft, chndbobj, anode],
        data: {
            noisedb: wc.tn(chndbobj),
            anode: wc.tn(anode),
            dft: wc.tn(dft),
            rms_threshold: 0.0,
        },
    },
```

- np04_hd Data Selection : Run no. 24720, 24726 (suggested by Roger), and 19337
  - 24720 and 24726 are both mid-filling (taken in last couple of weeks) and noise runs
  - 19337 is a pulsar run (taken in February 2023)

np04_hd Data Selection

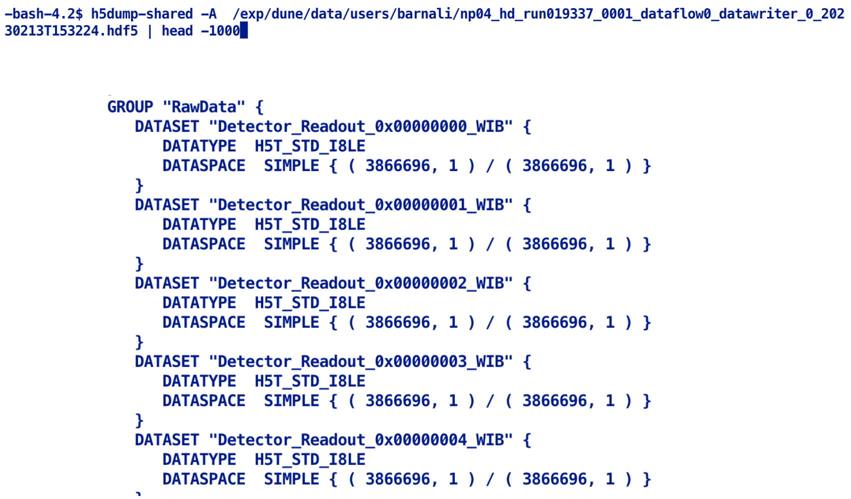
Relatively old : Run 19337

Fairly recent data: Run 24720/24726



```
-bash-4.2$ h5dump-shared -A  /exp/dune/data/users/barnali/np04hd_raw_run024720_0418_dataflow0_datawriter_0_
20240404T080920.hdf5 | head -1000

GROUP "RawData" {
   DATASET "Detector_Readout_0x00000064_WIBEth" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 928872, 1 ) / ( 928872, 1 ) }
   }
   DATASET "Detector_Readout_0x00000065_WIBEth" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 928872, 1 ) / ( 928872, 1 ) }
   }
   DATASET "Detector_Readout_0x00000066_WIBEth" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 928872, 1 ) / ( 928872, 1 ) }
   }
   DATASET "Detector_Readout_0x00000067_WIBEth" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 928872, 1 ) / ( 928872, 1 ) }
   }
   DATASET "Detector_Readout_0x00000068_WIBEth" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 928872, 1 ) / ( 928872, 1 ) }
   }
```



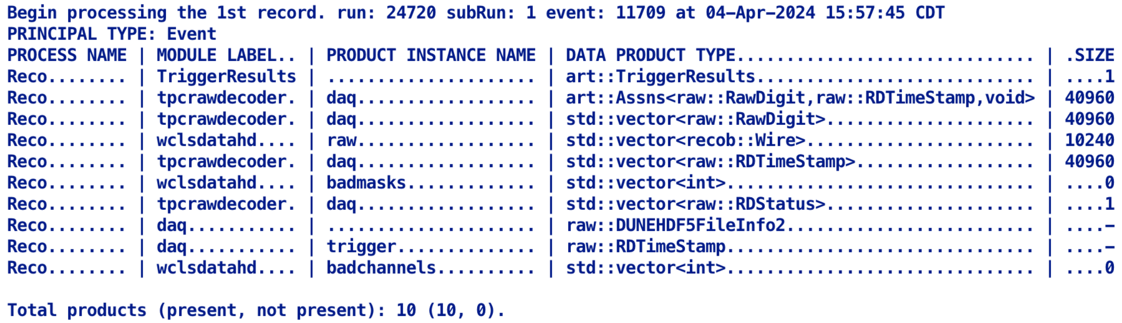
```
-bash-4.2$ h5dump-shared -A  /exp/dune/data/users/barnali/np04_hd_run019337_0001_dataflow0_datawriter_0_202
30213T153224.hdf5 | head -1000

GROUP "RawData" {
   DATASET "Detector_Readout_0x00000000_WIB" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 3866696, 1 ) / ( 3866696, 1 ) }
   }
   DATASET "Detector_Readout_0x00000001_WIB" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 3866696, 1 ) / ( 3866696, 1 ) }
   }
   DATASET "Detector_Readout_0x00000002_WIB" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 3866696, 1 ) / ( 3866696, 1 ) }
   }
   DATASET "Detector_Readout_0x00000003_WIB" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 3866696, 1 ) / ( 3866696, 1 ) }
   }
   DATASET "Detector_Readout_0x00000004_WIB" {
      DATATYPE  H5T_STD_I8LE
      DATASPACE  SIMPLE { ( 3866696, 1 ) / ( 3866696, 1 ) }
```

- Run successful test with
  - ○ lar -n 1 -c testing_for_standard_reco_protoduneHD.fcl –s /exp/dune/data/users/barnali/np04hd_raw_run024720_0418_dataflow0_datawriter_0_20240404T080920.hdf5
  - ○ lar -n 1 -c testing_for_standard_reco_protoduneHD.fcl -s /exp/dune/data/users/barnali/np04_hd_run019337_0001_dataflow0_datawriter_0_20230213T153224.hdf5

Testing np04_hd Data

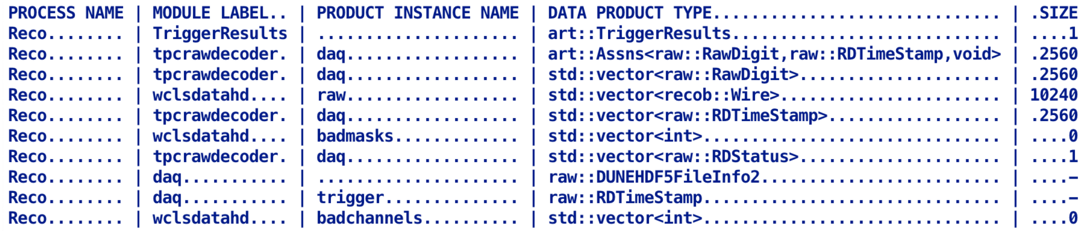
- lar -n1 -c eventdump.fcl np04hd_raw_run024720_0418_dataflow0_datawriter_0_20240404T080920_reco.root



```
Begin processing the 1st record. run: 24720 subRun: 1 event: 11709 at 04-Apr-2024 15:57:45 CDT
PRINCIPAL TYPE: Event
PROCESS NAME | MODULE LABEL.. | PRODUCT INSTANCE NAME | DATA PRODUCT TYPE............................. | .SIZE
Reco........ | TriggerResults | ..................... | art::TriggerResults........................... | ....1
Reco........ | tpcrawdecoder. | daq.................. | art::Assns<raw::RawDigit,raw::RDTimeStamp,void> | 40960
Reco........ | tpcrawdecoder. | daq.................. | std::vector<raw::RawDigit>.................... | 40960
Reco........ | wclsdatahd.... | raw.................. | std::vector<recob::Wire>...................... | 10240
Reco........ | tpcrawdecoder. | daq.................. | std::vector<raw::RDTimeStamp>................. | 40960
Reco........ | wclsdatahd.... | badmasks............. | std::vector<int>.............................. | ....0
Reco........ | tpcrawdecoder. | daq.................. | std::vector<raw::RDStatus>.................... | ....1
Reco........ | daq........... | ..................... | raw::DUNEHDF5FileInfo2........................ | ....-
Reco........ | daq........... | trigger.............. | raw::RDTimeStamp.............................. | ....-
Reco........ | wclsdatahd.... | badchannels.......... | std::vector<int>.............................. | ....0

Total products (present, not present): 10 (10, 0).
```

- lar -n1 -c eventdump.fcl np04_hd_run019337_0001_dataflow0_datawriter_0_20230213T153224_reco.root



```
PROCESS NAME | MODULE LABEL.. | PRODUCT INSTANCE NAME | DATA PRODUCT TYPE............................. | .SIZE
Reco........ | TriggerResults | ..................... | art::TriggerResults........................... | ....1
Reco........ | tpcrawdecoder. | daq.................. | art::Assns<raw::RawDigit,raw::RDTimeStamp,void> | .2560
Reco........ | tpcrawdecoder. | daq.................. | std::vector<raw::RawDigit>.................... | .2560
Reco........ | wclsdatahd.... | raw.................. | std::vector<recob::Wire>...................... | 10240
Reco........ | tpcrawdecoder. | daq.................. | std::vector<raw::RDTimeStamp>................. | .2560
Reco........ | wclsdatahd.... | badmasks............. | std::vector<int>.............................. | ....0
Reco........ | tpcrawdecoder. | daq.................. | std::vector<raw::RDStatus>.................... | ....1
Reco........ | daq........... | ..................... | raw::DUNEHDF5FileInfo2........................ | ....-
Reco........ | daq........... | trigger.............. | raw::RDTimeStamp.............................. | ....-
Reco........ | wclsdatahd.... | badchannels.......... | std::vector<int>.............................. | ....0
```

Implementation of HD Noise Filter

- There are two types of noise filters:
  - Single-channel : RC undershoot correction (aka “tail removal”) and and the pedestal removal.
  - Group-channel : Coherent noise removal
  - channel calibration for electronics response is not implemented

- First one is implemented in
  - WireCell::Waveform::ChannelMaskMap PDHD::OneChannelNoise::apply(int ch, signal_t& signal) const
  - https://github.com/WireCell/wire-cell-toolkit/blob/master/sigproc/src/ProtoduneHD.cxx#L726

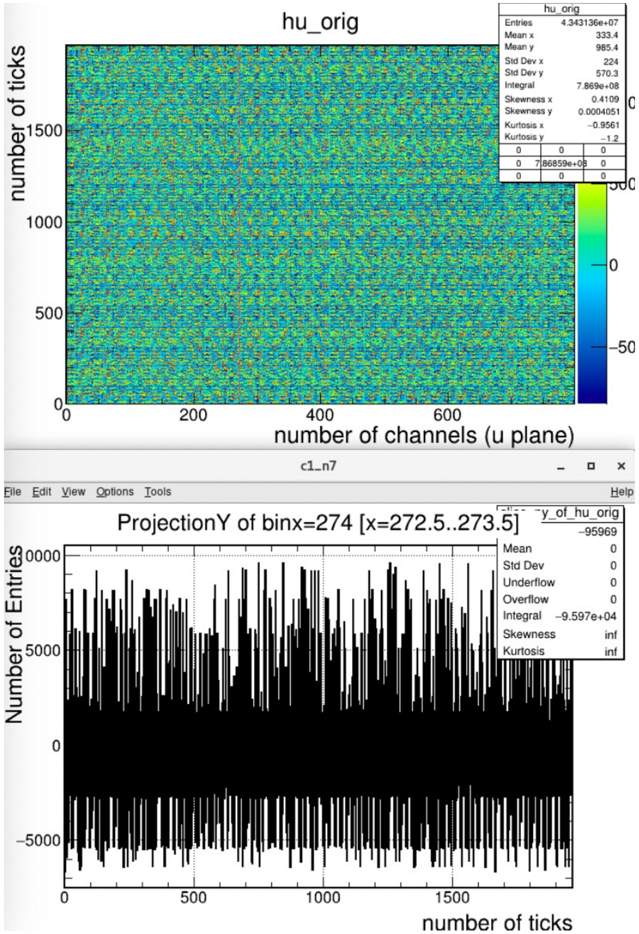
- Coherent noise removal is implemented in
  - WireCell::Waveform::ChannelMaskMap PDHD::CoherentNoiseSub::apply(channel_signals_t& chansig) const
  - https://github.com/WireCell/wire-cell-toolkit/blob/master/sigproc/src/ProtoduneHD.cxx#L809C1-L809C99

- There could be some other noise removal components we need to work on when protoDUNE HD data arrives

First look at ProtoDUNE HD noise data with WireCell noise filter

Run 24726 : Effect of Noise Filter

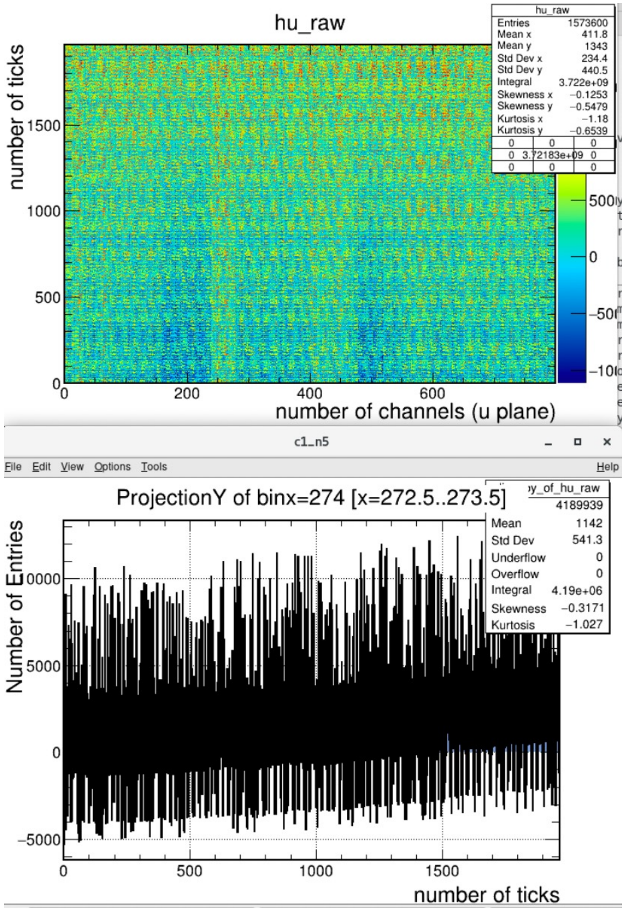
Before NF



```
hu_orig
Entries      4.343136e+07
Mean x              333.4
Mean y              985.4
Std Dev x             224
Std Dev y           570.3
Integral        7.869e+08
Skewness x         0.4109
Skewness y      0.0004051
Kurtosis x        −0.9561
Kurtosis y           −1.2
0        0               0
0   7.86859e+08          0
0        0               0
```

```
y_of_hu_orig
                   −95969
Mean                    0
Std Dev                 0
Underflow               0
Overflow                0
Integral       −9.597e+04
Skewness              inf
Kurtosis              inf
```

After NF

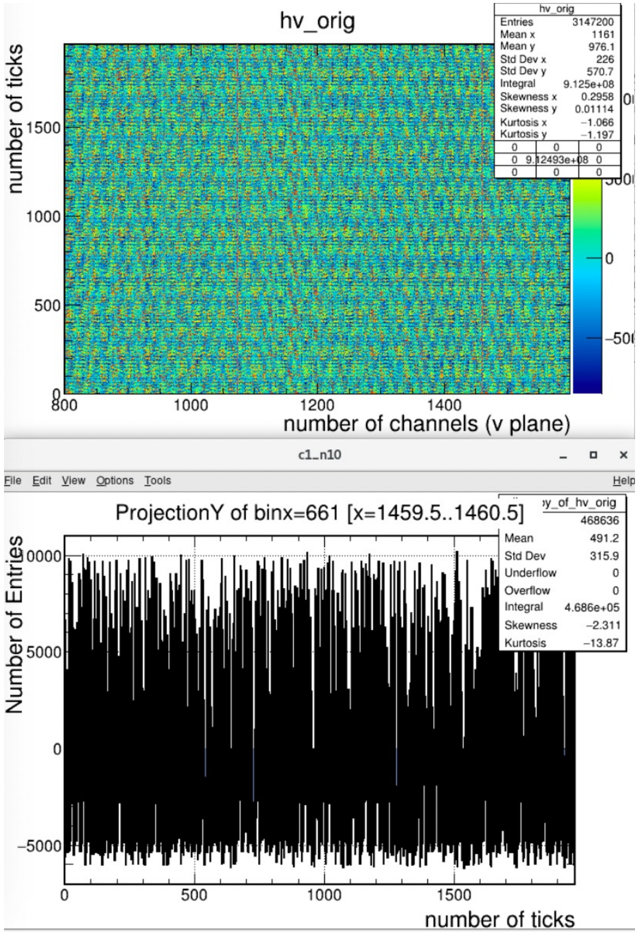


```
hu_raw
Entries           1573600
Mean x              411.8
Mean y               1343
Std Dev x           234.4
Std Dev y           440.5
Integral        3.722e+09
Skewness x        −0.1253
Skewness y        −0.5479
Kurtosis x          −1.18
Kurtosis y        −0.6539
0        0               0
0   3.72183e+09          0
0        0               0
```

```
y_of_hu_raw
                  4189939
Mean                 1142
Std Dev             541.3
Underflow               0
Overflow                0
Integral         4.19e+06
Skewness          −0.3171
Kurtosis           −1.027
```

Run 24726 : Effect of Noise Filter

Before NF

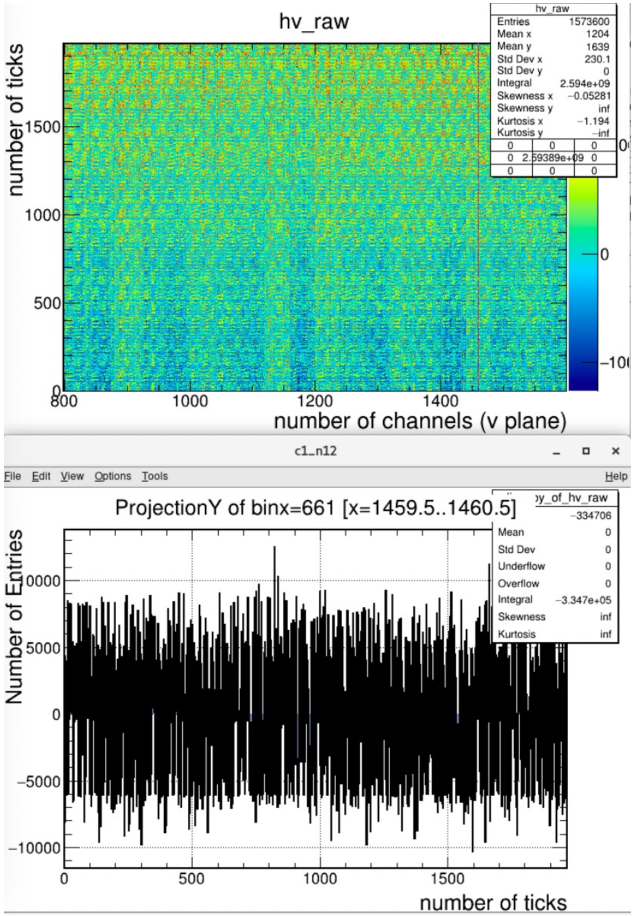


hv_orig	
Entries	3147200
Mean x	1161
Mean y	976.1
Std Dev x	226
Std Dev y	570.7
Integral	9.125e+08
Skewness x	0.2958
Skewness y	0.01114
Kurtosis x	−1.066
Kurtosis y	−1.197

0	0	0
0	9.12493e+08	0
0	0	0

›y_of_hv_orig	
	468636
Mean	491.2
Std Dev	315.9
Underflow	0
Overflow	0
Integral	4.686e+05
Skewness	−2.311
Kurtosis	−13.87

After NF



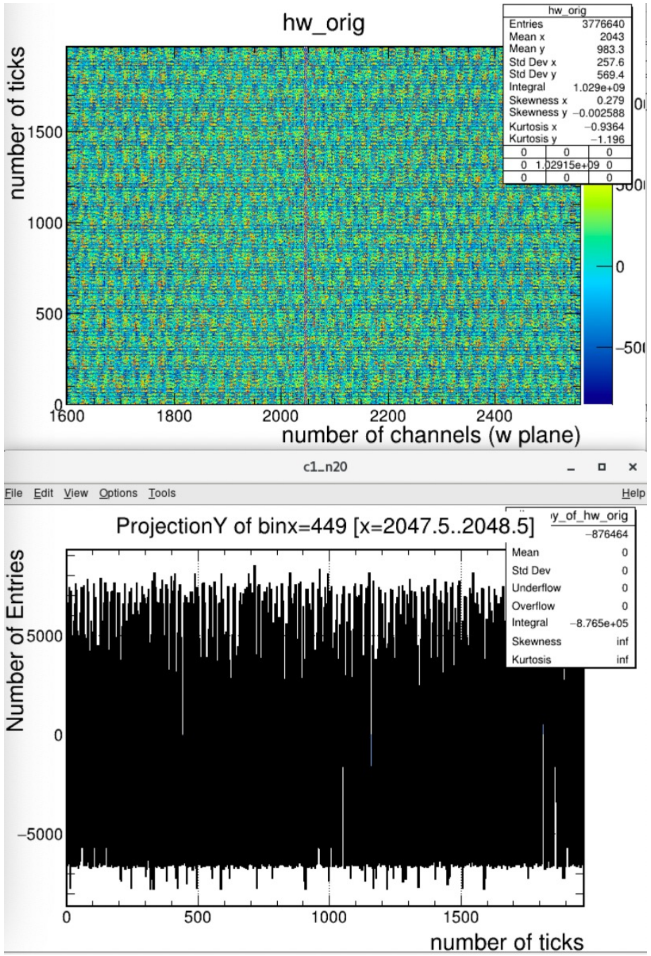
hv_raw	
Entries	1573600
Mean x	1204
Mean y	1639
Std Dev x	230.1
Std Dev y	0
Integral	2.594e+09
Skewness x	−0.05281
Skewness y	inf
Kurtosis x	−1.194
Kurtosis y	−inf

0	0	0
0	2.59389e+09	0
0	0	0

›y_of_hv_raw	
	−334706
Mean	0
Std Dev	0
Underflow	0
Overflow	0
Integral	−3.347e+05
Skewness	inf
Kurtosis	inf

Run 24726 : Effect of Noise Filter

Before NF

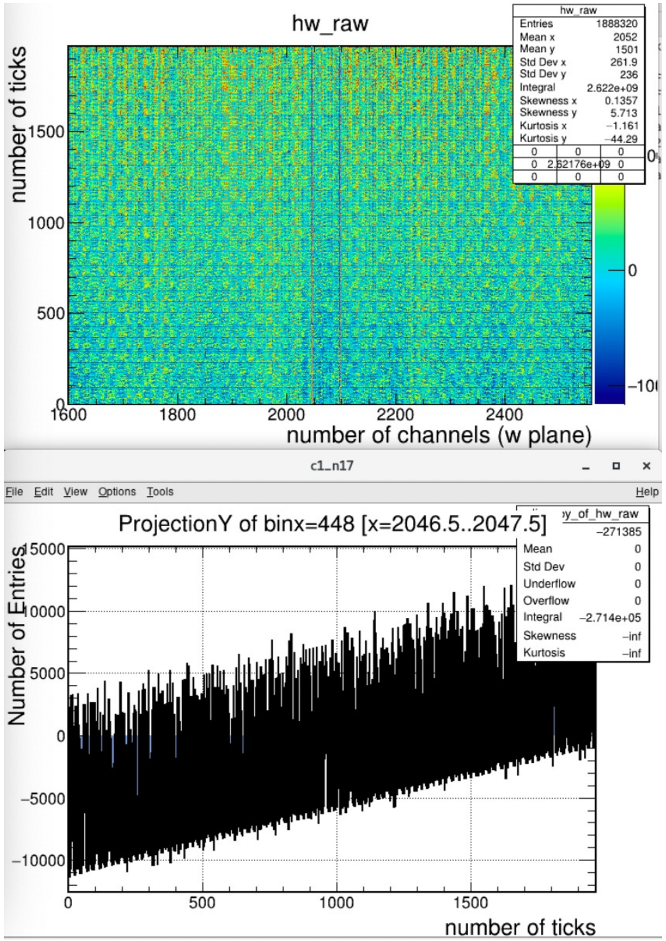


hw_orig	
Entries	3776640
Mean x	2043
Mean y	983.3
Std Dev x	257.6
Std Dev y	569.4
Integral	1.029e+09
Skewness x	0.279
Skewness y	−0.002588
Kurtosis x	−0.9364
Kurtosis y	−1.196

0	0	0
0	1.02915e+09	0
0	0	0

y_of_hw_orig	
	−876464
Mean	0
Std Dev	0
Underflow	0
Overflow	0
Integral	−8.765e+05
Skewness	inf
Kurtosis	inf

After NF



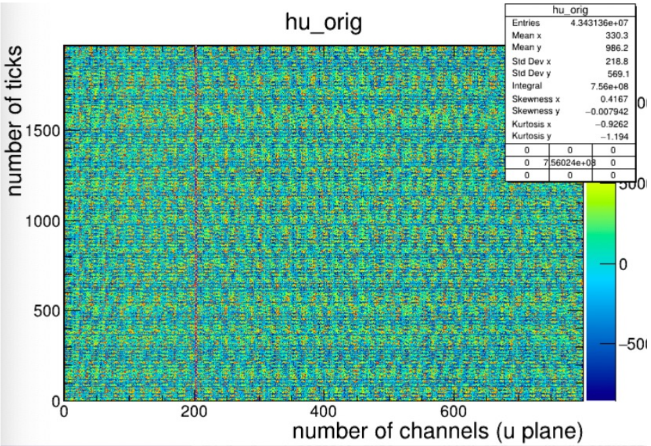
hw_raw	
Entries	1888320
Mean x	2052
Mean y	1501
Std Dev x	261.9
Std Dev y	236
Integral	2.622e+09
Skewness x	0.1357
Skewness y	5.713
Kurtosis x	−1.161
Kurtosis y	−44.29

0	0	0
0	2.62176e+09	0
0	0	0

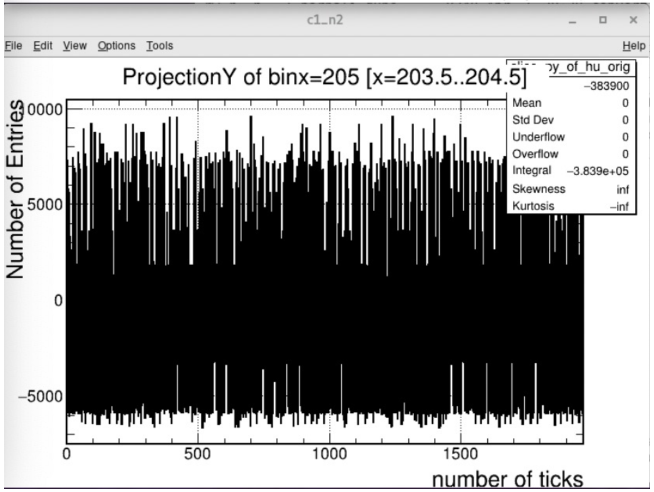
y_of_hw_raw	
	−271385
Mean	0
Std Dev	0
Underflow	0
Overflow	0
Integral	−2.714e+05
Skewness	−inf
Kurtosis	−inf

Run 24720 : Effect of Noise Filter

Before NF

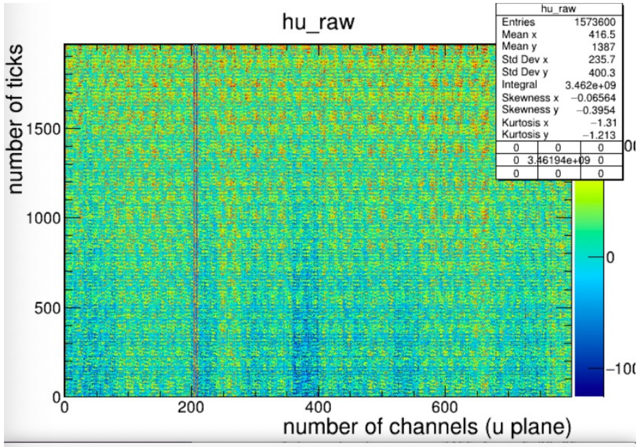


```
hu_orig
Entries     4.343136e+07
Mean x             330.3
Mean y             986.2
Std Dev x          218.8
Std Dev y          569.1
Integral        7.56e+08
Skewness x        0.4167
Skewness y     −0.007942
Kurtosis x       −0.9262
Kurtosis y        −1.194
   0        0         0
   0   7.56024e+08    0
   0        0         0
```

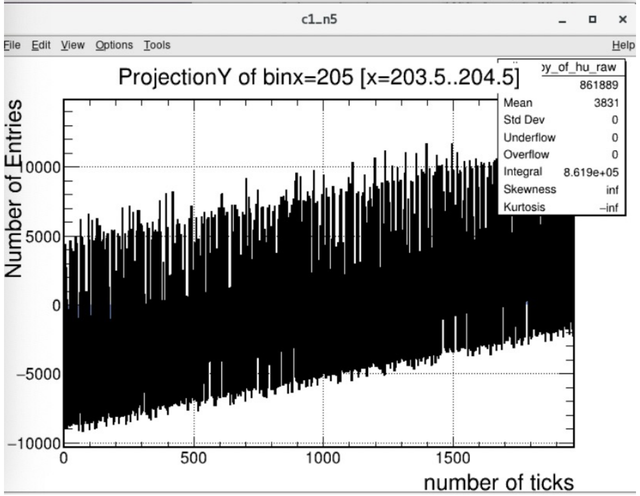


```
py_of_hu_orig
                 −383900
Mean                   0
Std Dev                0
Underflow              0
Overflow               0
Integral      −3.839e+05
Skewness             inf
Kurtosis            −inf
```

After NF



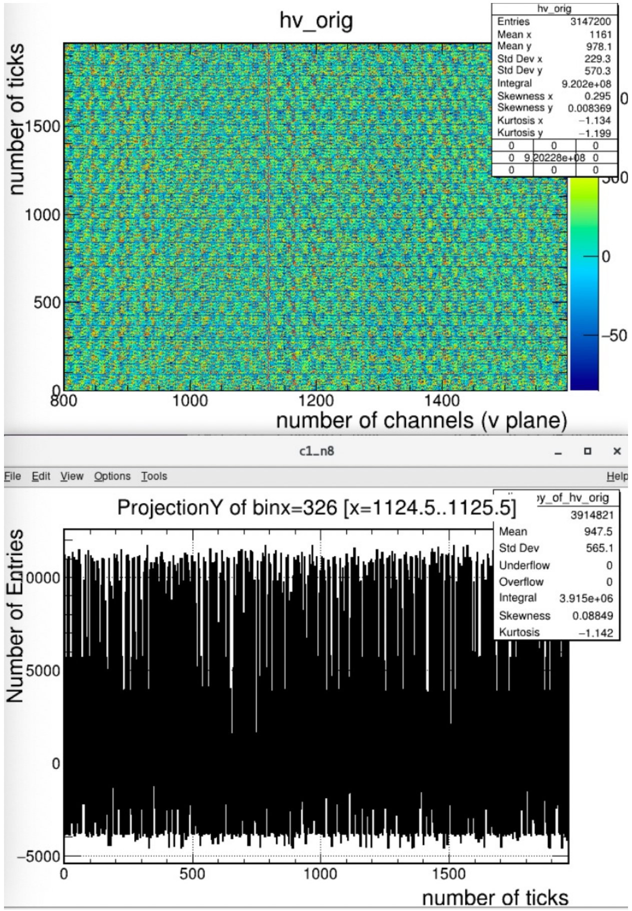
```
hu_raw
Entries          1573600
Mean x             416.5
Mean y              1387
Std Dev x          235.7
Std Dev y          400.3
Integral       3.462e+09
Skewness x      −0.06564
Skewness y       −0.3954
Kurtosis x         −1.31
Kurtosis y        −1.213
   0        0         0
   0   3.46194e+09    0
   0        0         0
```



```
py_of_hu_raw
                  861889
Mean                3831
Std Dev                0
Underflow              0
Overflow               0
Integral       8.619e+05
Skewness             inf
Kurtosis            −inf
```

Run 24720 : Effect of Noise Filter

Before NF

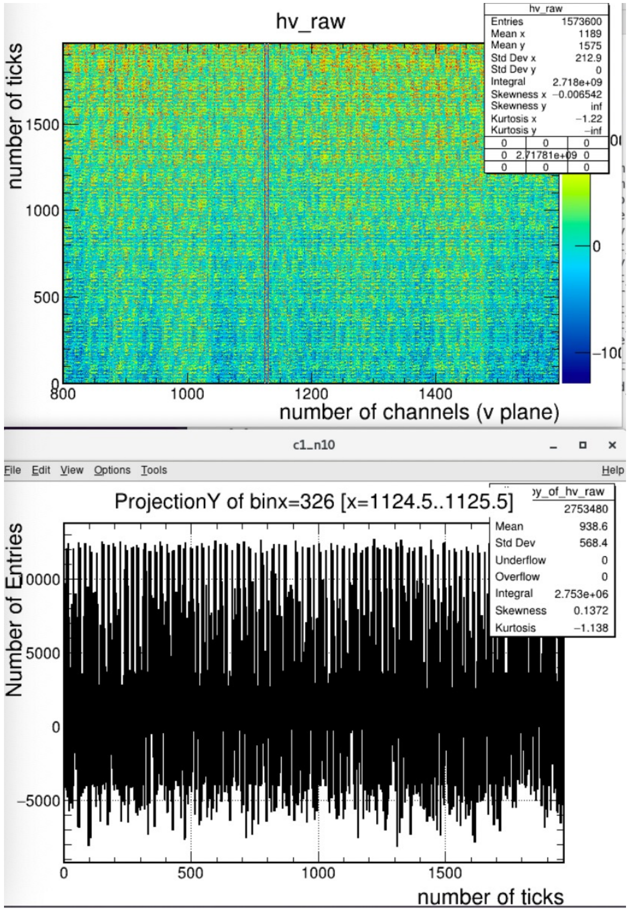


hv_orig	
Entries	3147200
Mean x	1161
Mean y	978.1
Std Dev x	229.3
Std Dev y	570.3
Integral	9.202e+08
Skewness x	0.295
Skewness y	0.008369
Kurtosis x	−1.134
Kurtosis y	−1.199

0	0	0
0	9.20228e+08	0
0	0	0

›y_of_hv_orig	
	3914821
Mean	947.5
Std Dev	565.1
Underflow	0
Overflow	0
Integral	3.915e+06
Skewness	0.08849
Kurtosis	−1.142

After NF



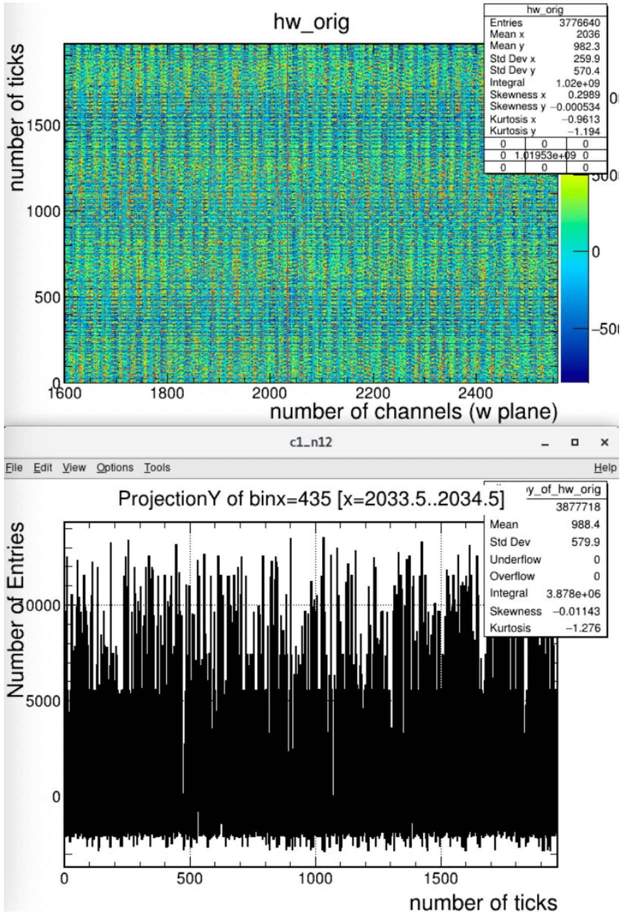
hv_raw	
Entries	1573600
Mean x	1189
Mean y	1575
Std Dev x	212.9
Std Dev y	0
Integral	2.718e+09
Skewness x	−0.006542
Skewness y	inf
Kurtosis x	−1.22
Kurtosis y	−inf

0	0	0
0	2.71781e+09	0
0	0	0

›y_of_hv_raw	
	2753480
Mean	938.6
Std Dev	568.4
Underflow	0
Overflow	0
Integral	2.753e+06
Skewness	0.1372
Kurtosis	−1.138

Run 24720 : Effect of Noise Filter

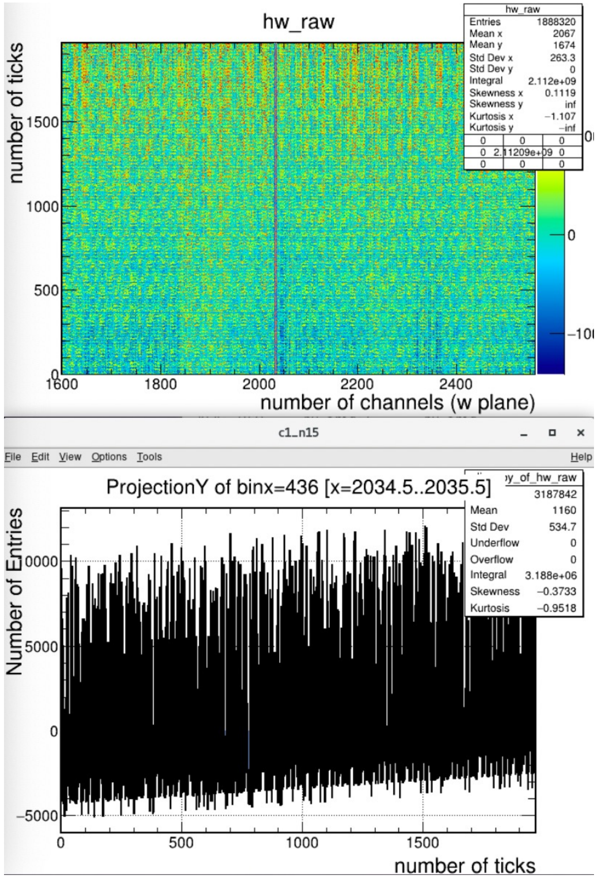
Before NF



```
hw_orig
Entries          3776640
Mean x              2036
Mean y             982.3
Std Dev x          259.9
Std Dev y          570.4
Integral        1.02e+09
Skewness x        0.2989
Skewness y   −0.000534
Kurtosis x       −0.9613
Kurtosis y        −1.194
   0        0         0
   0  1.01953e+09     0
   0        0         0
```

```
y_of_hw_orig
                 3877718
Mean               988.4
Std Dev            579.9
Underflow              0
Overflow               0
Integral       3.878e+06
Skewness        −0.01143
Kurtosis          −1.276
```

After NF

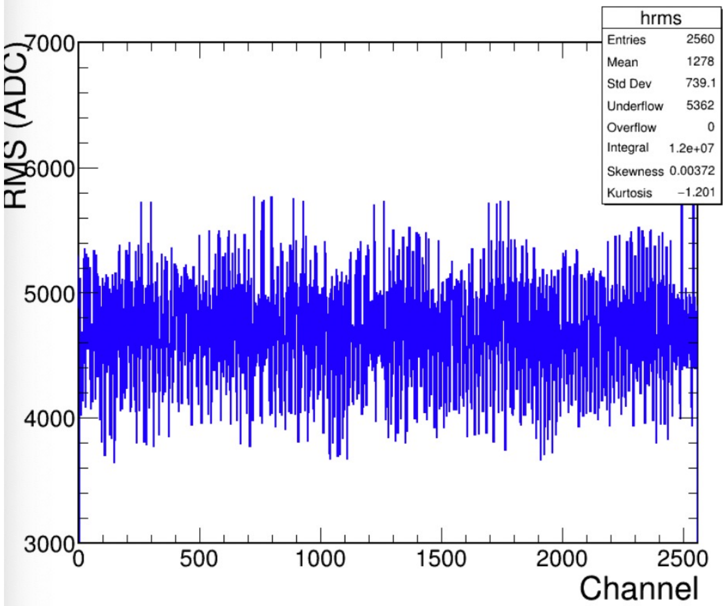


```
hw_raw
Entries          1888320
Mean x              2067
Mean y              1674
Std Dev x          263.3
Std Dev y              0
Integral       2.112e+09
Skewness x        0.1119
Skewness y           inf
Kurtosis x        −1.107
Kurtosis y          −inf
   0        0         0
   0  2.11209e+09     0
   0        0         0
```

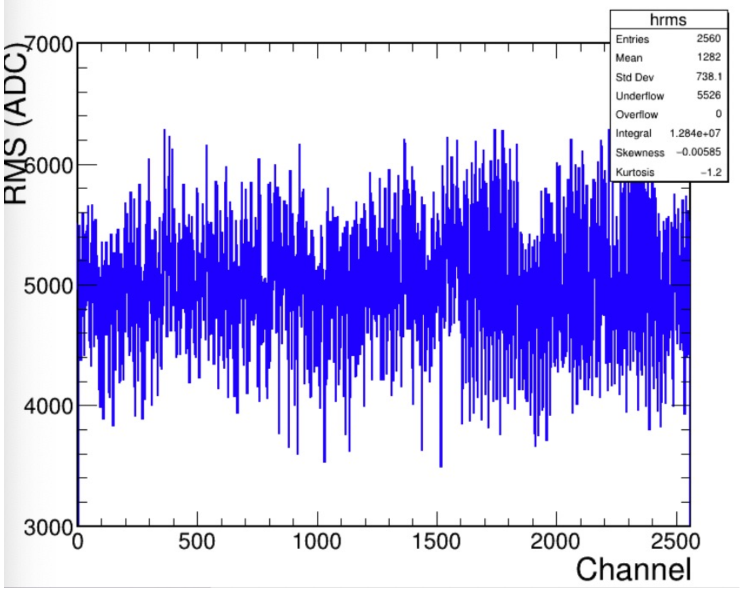
```
y_of_hw_raw
                 3187842
Mean                1160
Std Dev            534.7
Underflow              0
Overflow               0
Integral       3.188e+06
Skewness         −0.3733
Kurtosis         −0.9518
```

Run 24720 : Effect of Noise Filter

Before NF

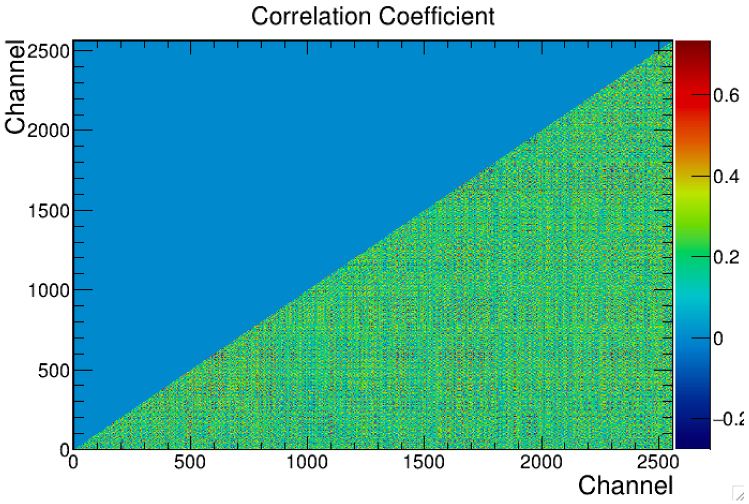


After NF

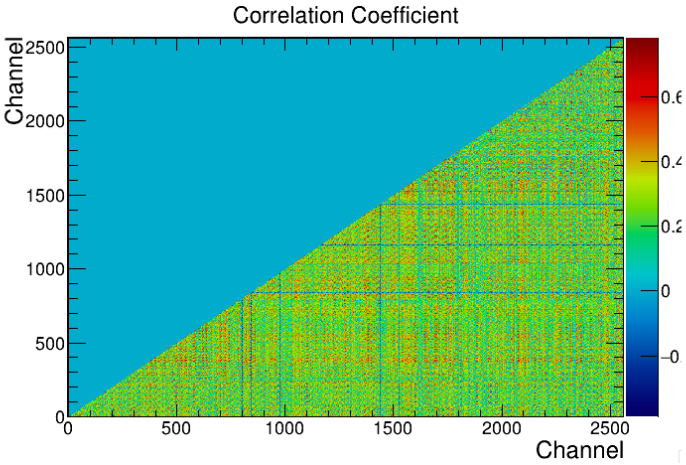


Run 24720 : Linear Correlation Between Channels

Before NF



After NF

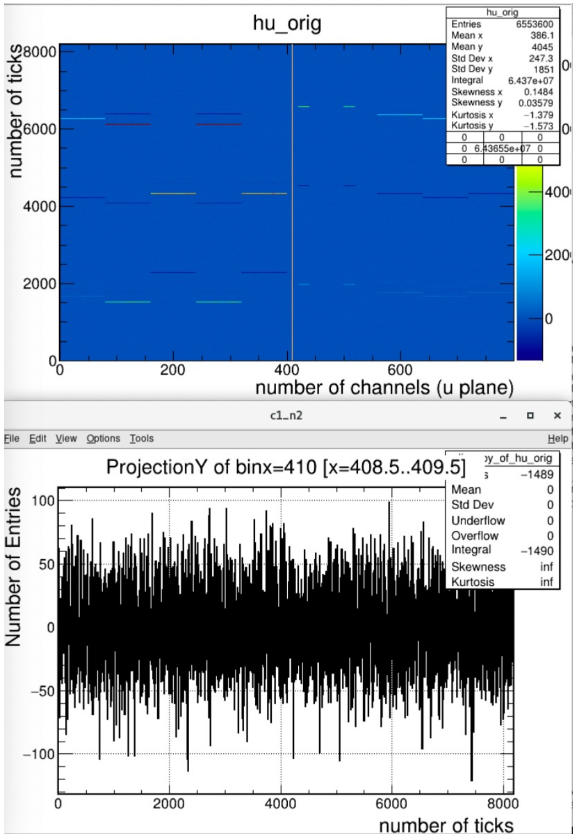


Reason for observing unexpected behavior for Run no. 24720 & 24726

- Chatting with Tom
  - Hi Barnali -- Kurt sent me a message saying that they changed the data and file format (again!) But that the software will not be finalized until April 17. I've been holding off copying the DAQ versions for the offline until it is tagged and released, as Kurt says further changes are possible. He said some files have "snuck through" with the new format.

Run 19337 : Effect of Noise Filter

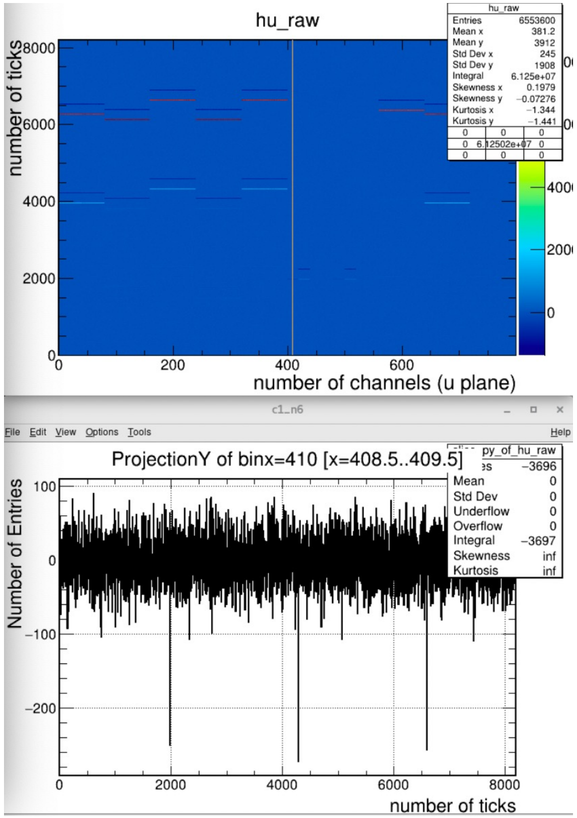
Before NF



```
hu_orig
Entries          6553600
Mean x             386.1
Mean y              4045
Std Dev x          247.3
Std Dev y           1851
Integral       6.437e+07
Skewness x        0.1484
Skewness y       0.03579
Kurtosis x        −1.379
Kurtosis y        −1.573
   0        0        0
   0  6.43655e+07    0
   0        0        0
```

```
py_of_hu_orig
s                  −1489
Mean                   0
Std Dev                0
Underflow              0
Overflow               0
Integral           −1490
Skewness             inf
Kurtosis             inf
```

After NF



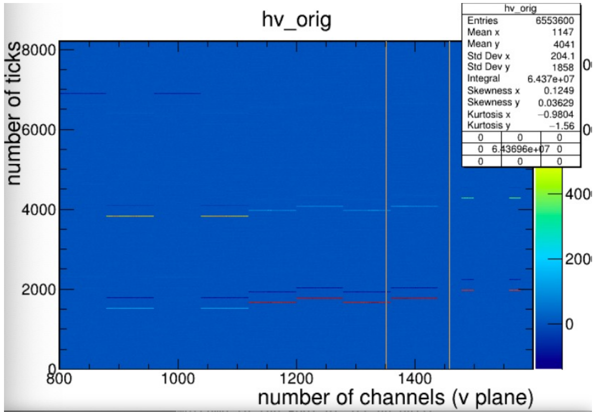
```
hu_raw
Entries          6553600
Mean x             381.2
Mean y              3912
Std Dev x            245
Std Dev y           1908
Integral       6.125e+07
Skewness x        0.1979
Skewness y      −0.07276
Kurtosis x        −1.344
Kurtosis y        −1.441
   0        0        0
   0  6.12502e+07    0
   0        0        0
```

```
py_of_hu_raw
es                 −3696
Mean                   0
Std Dev                0
Underflow              0
Overflow               0
Integral           −3697
Skewness             inf
Kurtosis             inf
```

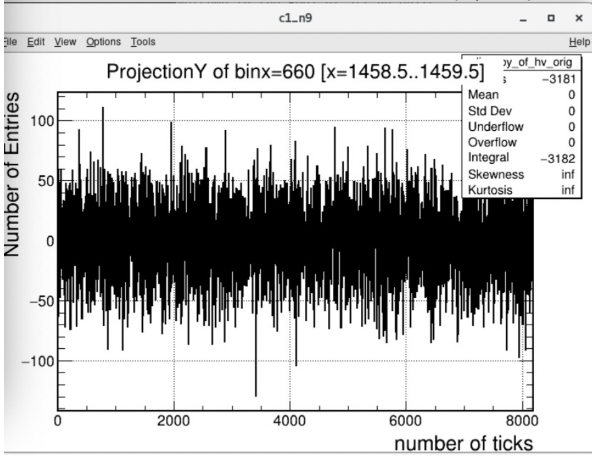
- This is pulsar data, but selected channel is seeing only noise
- The negative peak after noise filter is interesting --- need further investigation

Run 19337 : Effect of Noise Filter

Before NF

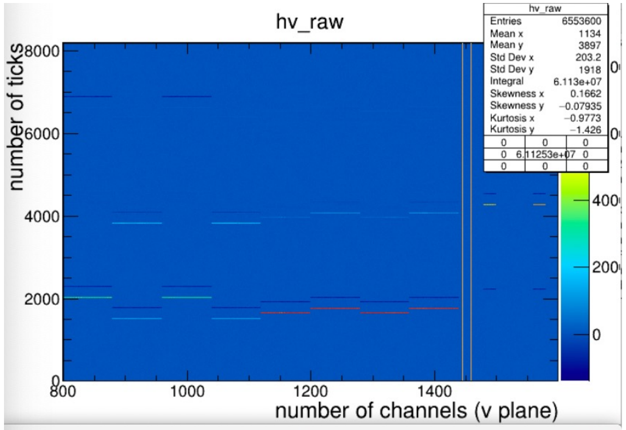


```
hv_orig
Entries        6553600
Mean x            1147
Mean y            4041
Std Dev x        204.1
Std Dev y         1858
Integral     6.437e+07
Skewness x      0.1249
Skewness y     0.03629
Kurtosis x     −0.9804
Kurtosis y       −1.56
  0       0        0
  0  6.43696e+07   0
  0       0        0
```

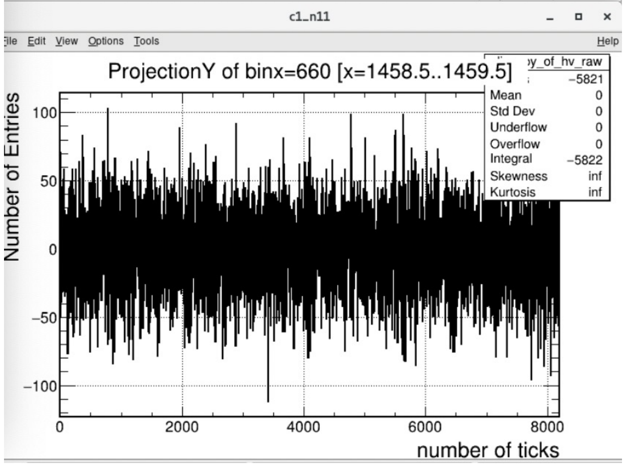


```
y_of_hv_orig
s           −3181
Mean            0
Std Dev         0
Underflow       0
Overflow        0
Integral    −3182
Skewness      inf
Kurtosis      inf
```

After NF



```
hv_raw
Entries        6553600
Mean x            1134
Mean y            3897
Std Dev x        203.2
Std Dev y         1918
Integral     6.113e+07
Skewness x      0.1662
Skewness y    −0.07935
Kurtosis x     −0.9773
Kurtosis y      −1.426
  0       0        0
  0  6.11253e+07   0
  0       0        0
```

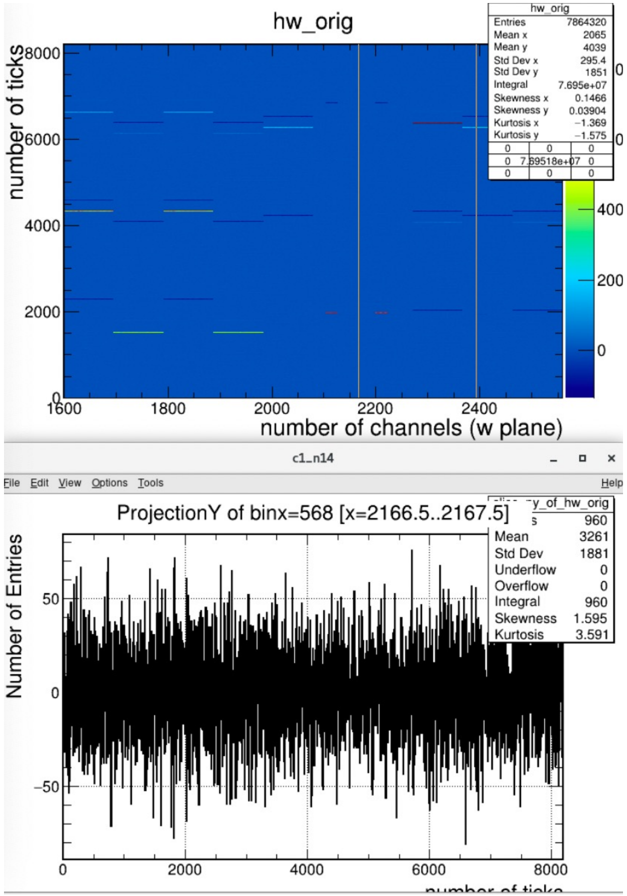


```
y_of_hv_raw
            −5821
Mean            0
Std Dev         0
Underflow       0
Overflow        0
Integral    −5822
Skewness      inf
Kurtosis      inf
```

- ➢ The diagrams are in v plane

Run 19337 : Effect of Noise Filter

Before NF

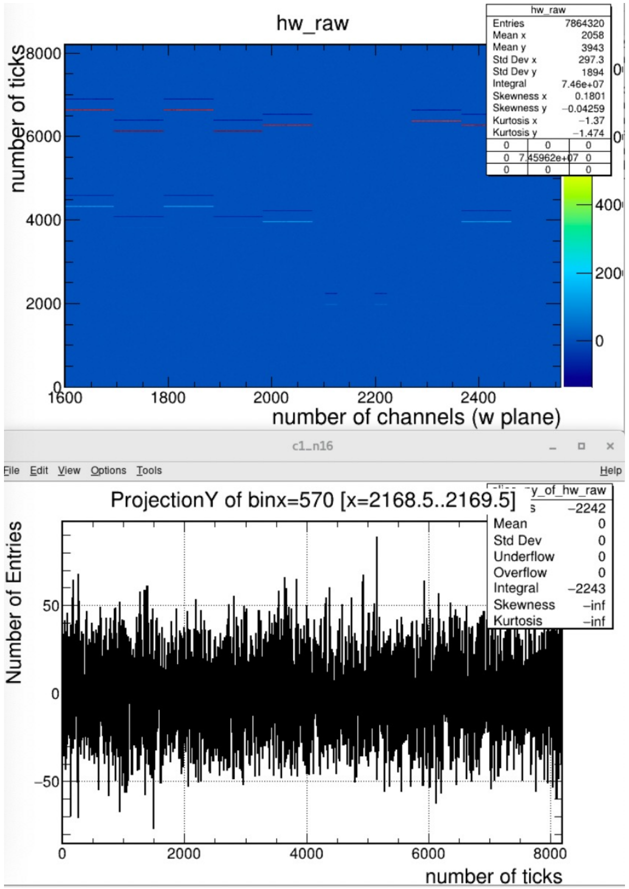


hw_orig	
Entries	7864320
Mean x	2065
Mean y	4039
Std Dev x	295.4
Std Dev y	1851
Integral	7.695e+07
Skewness x	0.1466
Skewness y	0.03904
Kurtosis x	−1.369
Kurtosis y	−1.575

0	0	0
0	7.69518e+07	0
0	0	0

...y_of_hw_orig	
...s	960
Mean	3261
Std Dev	1881
Underflow	0
Overflow	0
Integral	960
Skewness	1.595
Kurtosis	3.591

After NF



hw_raw	
Entries	7864320
Mean x	2058
Mean y	3943
Std Dev x	297.3
Std Dev y	1894
Integral	7.46e+07
Skewness x	0.1801
Skewness y	−0.04259
Kurtosis x	−1.37
Kurtosis y	−1.474

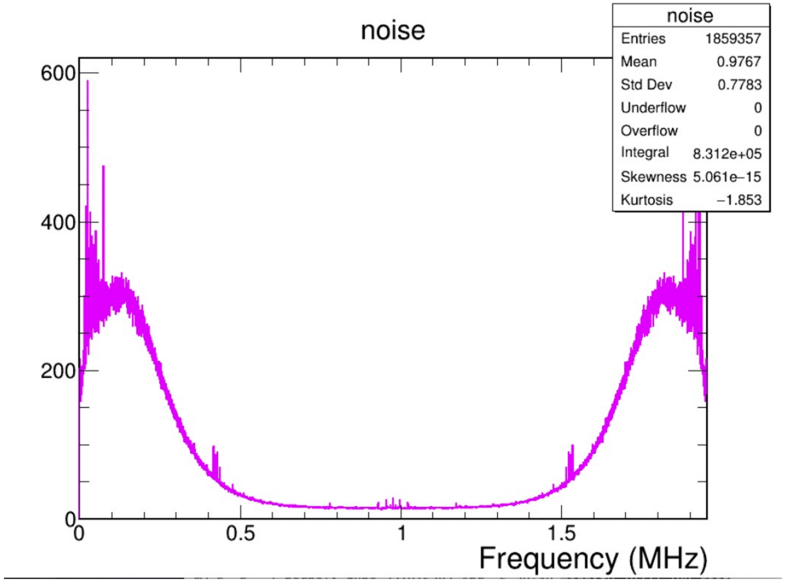
0	0	0
0	7.45962e+07	0
0	0	0

...y_of_hw_raw	
...s	−2242
Mean	0
Std Dev	0
Underflow	0
Overflow	0
Integral	−2243
Skewness	−inf
Kurtosis	−inf

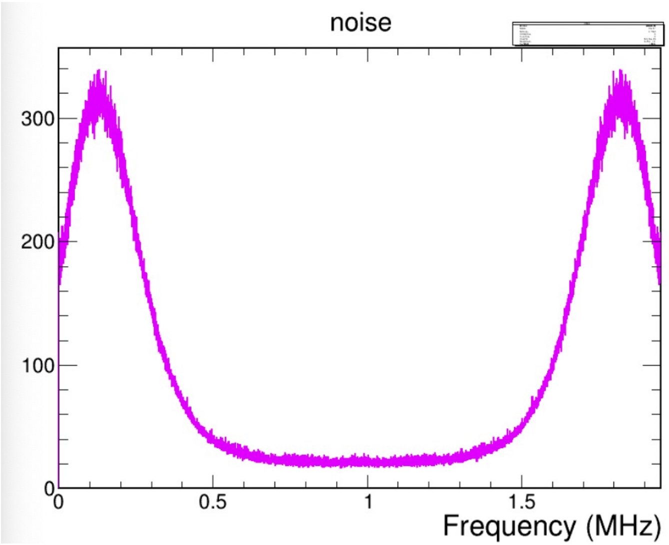
- The diagrams are in w plane

Run 19337 : Noise in Frequency Spectrum (Effect of Noise Filter)

Before NF



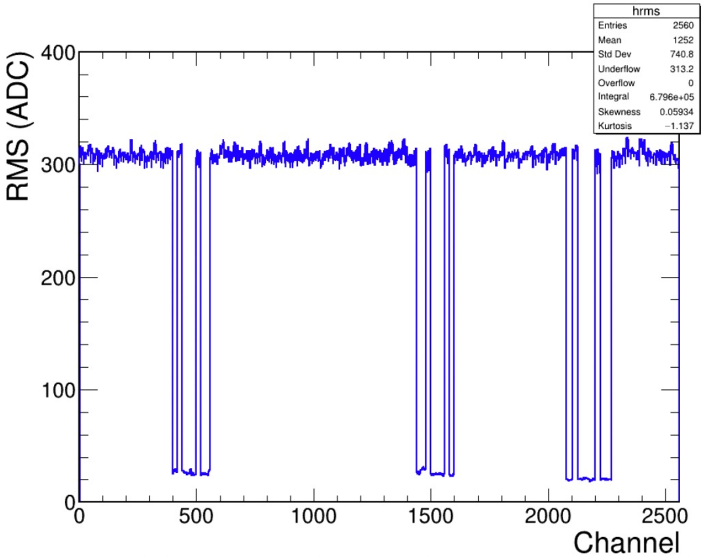
After NF



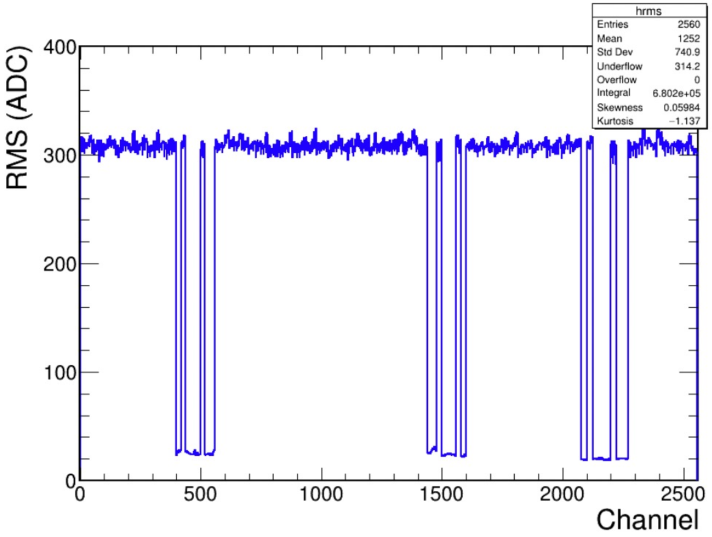
- The diagrams are in frequency spectrum
  - Frequency = a little less than 2 MHz
  - T = approximately 500 ns

Run 19337 : ADC RMS plot (Effect of Noise Filter)

Before NF



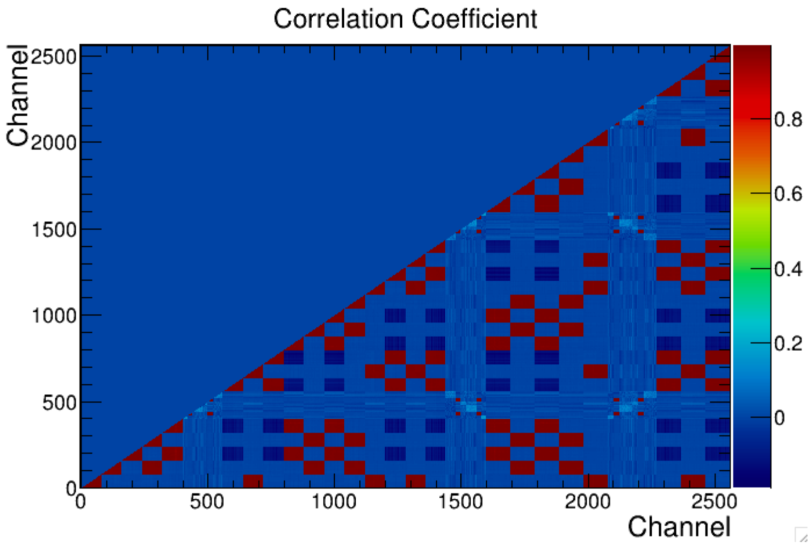
After NF



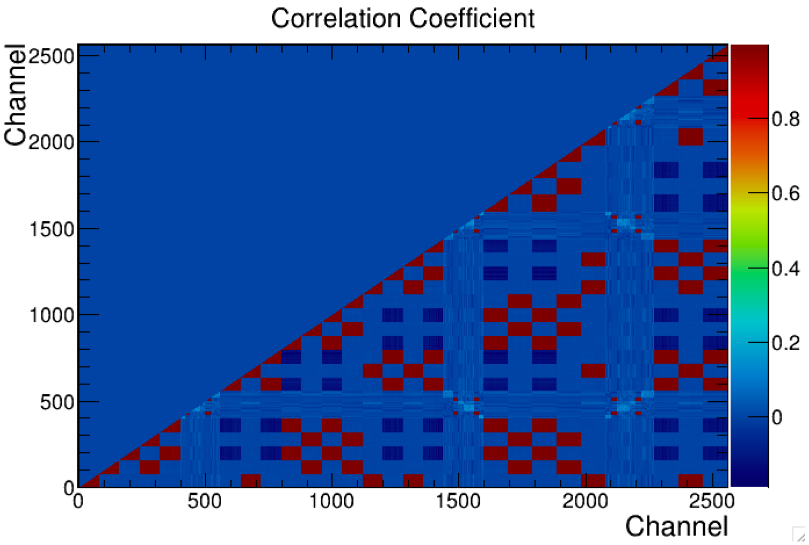
- Calculate ADC RMS before and after NF
- u, v, and w planes combined
- NF should reduce RMS in most cases
- Several dead channels cause RMS (ADC) values drop

Run 19337: Linear Correlation Between Channels

Before NF



After NF



- Calculate pairwise linear correlation for all channels
- Noise removal reduces linear correlation

Summary and Outlook

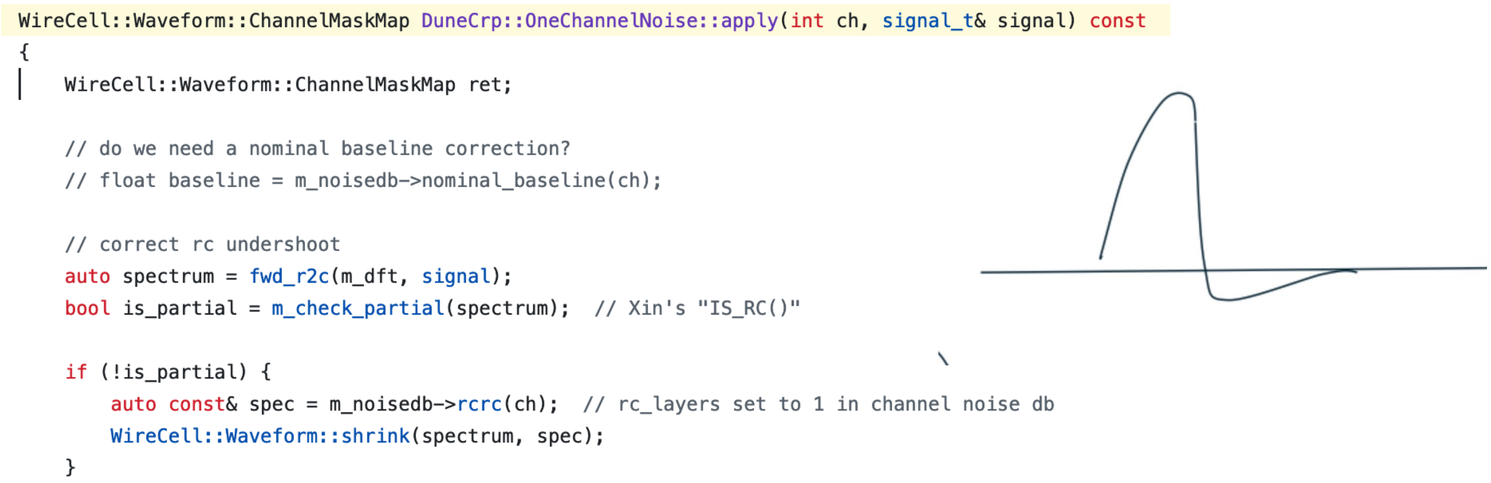
- Summary
  - Code for PDHD noise filter is developed
  - Consider good noise run to study from old runs ??
  - Or wait until April 17th for DUNE DAQ, DUNE data format and DUNE file format to stop evolving

- Outlook
  - Conduct analysis for other HD runs
  - Investigate effect of WireCell Noise Filter (NF)
  - Look for new / unknown types of noise
  - Find dead / noisy channels
  - Understand coherence introduced by NF
  - Investigate correlation / coherence across wire plane

Thank You !

Data Preparation and Noise Suppression with WireCell

- Correct RC undershoot first



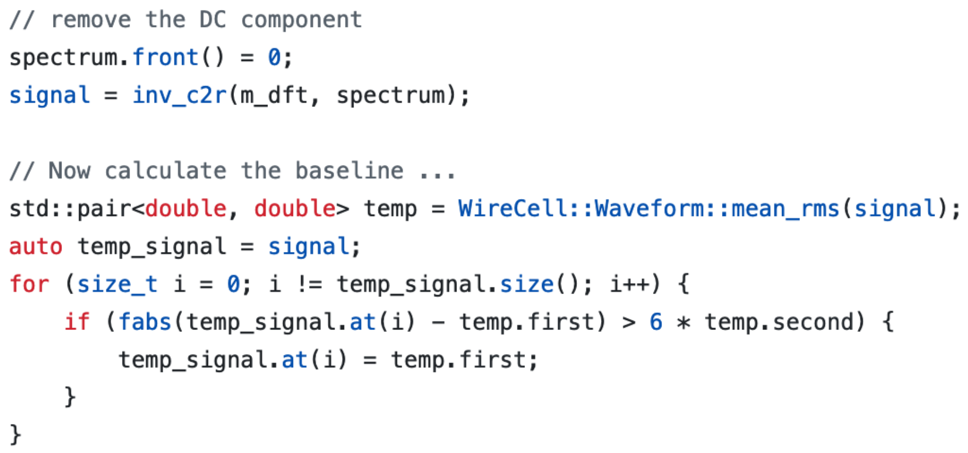
```
WireCell::Waveform::ChannelMaskMap DuneCrp::OneChannelNoise::apply(int ch, signal_t& signal) const
{
    WireCell::Waveform::ChannelMaskMap ret;

    // do we need a nominal baseline correction?
    // float baseline = m_noisedb->nominal_baseline(ch);

    // correct rc undershoot
    auto spectrum = fwd_r2c(m_dft, signal);
    bool is_partial = m_check_partial(spectrum);  // Xin's "IS_RC()"

    if (!is_partial) {
        auto const& spec = m_noisedb->rcrc(ch);  // rc_layers set to 1 in channel noise db
        WireCell::Waveform::shrink(spectrum, spec);
    }
```

- Remove the DC component or the baseline

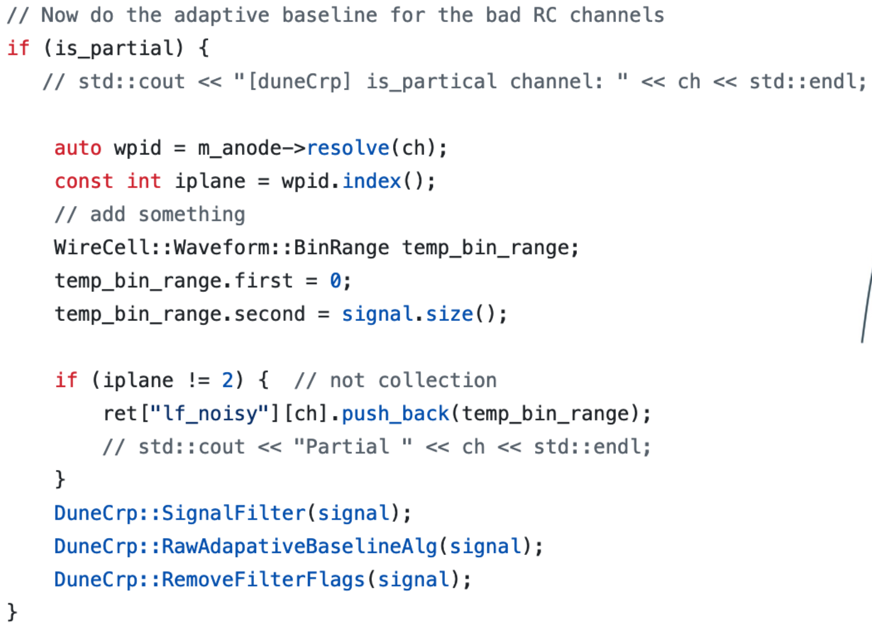


```
// remove the DC component
spectrum.front() = 0;
signal = inv_c2r(m_dft, spectrum);

// Now calculate the baseline ...
std::pair<double, double> temp = WireCell::Waveform::mean_rms(signal);
auto temp_signal = signal;
for (size_t i = 0; i != temp_signal.size(); i++) {
    if (fabs(temp_signal.at(i) - temp.first) > 6 * temp.second) {
        temp_signal.at(i) = temp.first;
    }
}
```

Data Preparation and Noise Suppression with WireCell

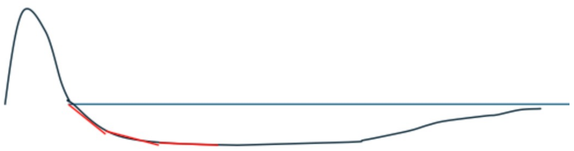
- Remove signal before the read-out window



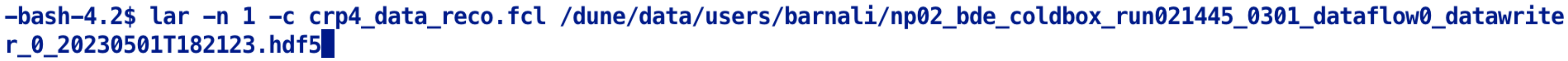
```
// Now do the adaptive baseline for the bad RC channels
if (is_partial) {
   // std::cout << "[duneCrp] is_partical channel: " << ch << std::endl;

    auto wpid = m_anode->resolve(ch);
    const int iplane = wpid.index();
    // add something
    WireCell::Waveform::BinRange temp_bin_range;
    temp_bin_range.first = 0;
    temp_bin_range.second = signal.size();

    if (iplane != 2) {  // not collection
        ret["lf_noisy"][ch].push_back(temp_bin_range);
        // std::cout << "Partial " << ch << std::endl;
    }
    DuneCrp::SignalFilter(signal);
    DuneCrp::RawAdapativeBaselineAlg(signal);
    DuneCrp::RemoveFilterFlags(signal);
}
```



- Run only “OneChannelNoise”



```
-bash-4.2$ lar -n 1 -c crp4_data_reco.fcl /dune/data/users/barnali/np02_bde_coldbox_run021445_0301_dataflow0_datawrite
r_0_20230501T182123.hdf5
```