Noise filter in PDVD data

Xuyang Ning

Brookhaven National Laboratory

electronic mapping

- mapping files:
  - https://github.com/DUNE/duneprototypes/blob/develop/duneprototypes/Protodune/vd/ChannelMap/PD2VDTPCChannelMap_v2.txt

```
3390 10 4 10 0 0 48 0 794 6 3 0
3391 10 4 10 0 0 49 0 795 6 3 1
3392 10 4 10 0 0 50 0 796 6 3 2
3393 10 4 10 0 0 51 0 797 6 3 3
3394 10 4 10 0 0 52 0 798 6 3 4
3395 10 4 10 0 0 53 0 799 6 3 5
3396 10 4 10 0 0 54 0 800 6 3 6
3397 10 4 10 0 0 55 0 801 6 3 7
3398 10 4 10 0 0 56 0 802 6 3 8
3399 10 4 10 0 0 57 0 803 6 3 9
3400 10 4 10 0 0 58 0 804 6 3 10
3401 10 4 10 0 0 59 0 805 6 3 11
3402 10 4 10 0 0 60 0 806 6 3 12
3403 10 4 10 0 0 61 0 807 6 3 13
3404 10 4 10 0 0 62 0 808 6 3 14
3405 10 4 10 0 0 63 0 809 6 3 15
3406 10 4 10 0 0 0 0 810 6 0 0
```

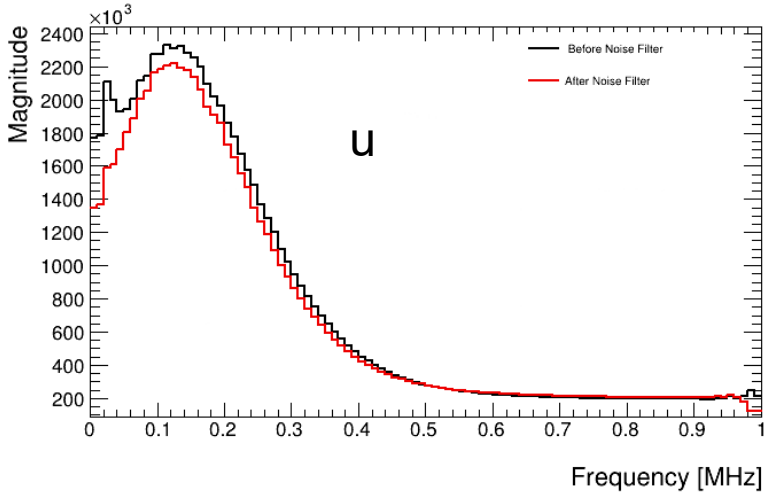
- meaning of the mapping (maybe):
  - https://github.com/DUNE/dunecore/blob/develop/dunecore/ChannelMap/TPCChannelMapSP.cxx#L2

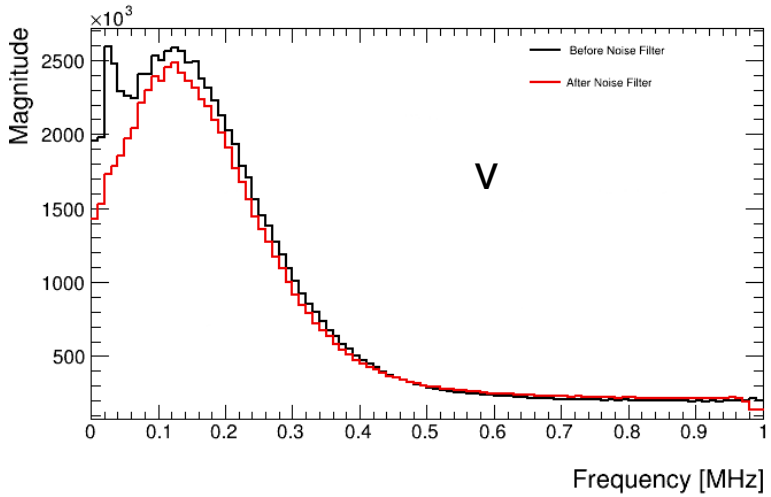
```
while (std::getline(inFile, line)) {
  std::stringstream linestream(line);

  TPCChanInfo_t chanInfo;
  linestream >>
    chanInfo.offlchan >>
    chanInfo.detid >>
    chanInfo.detelement >>
    chanInfo.crate >>
    chanInfo.slot >>
    chanInfo.stream >>
    chanInfo.streamchan >>
    chanInfo.plane >>
    chanInfo.chan_in_plane >>
    chanInfo.femb >>
    chanInfo.asic >>
    chanInfo.asicchan;

  chanInfo.valid = true;
  ++fNChans;
```

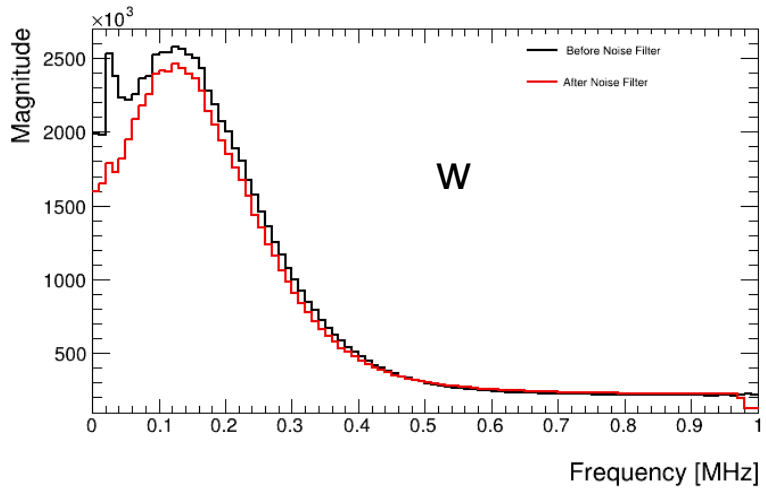
- Group channel for pdvd:
  - separate top and bottom, group offline channels according to femb, crate, stream;
  - streamchan would be 0-63
  - All channels will be divided in 64 in a group.

Last Time:CNR with new channel groups

CRP1

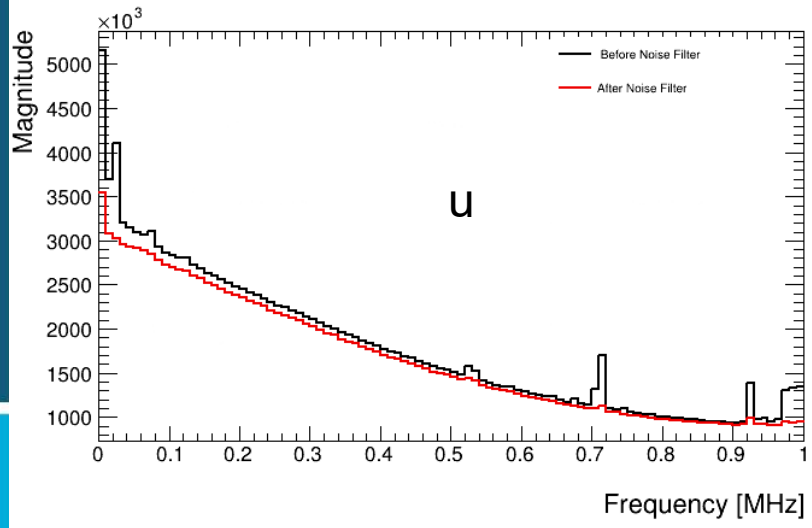


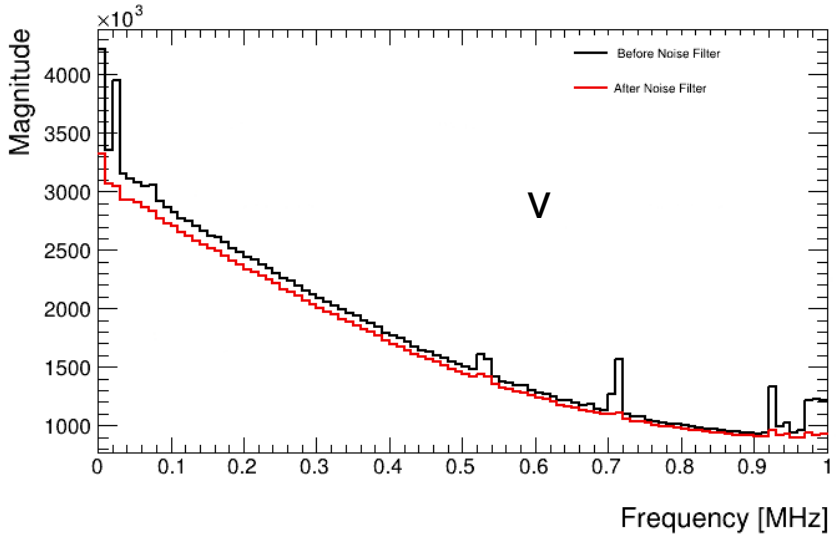


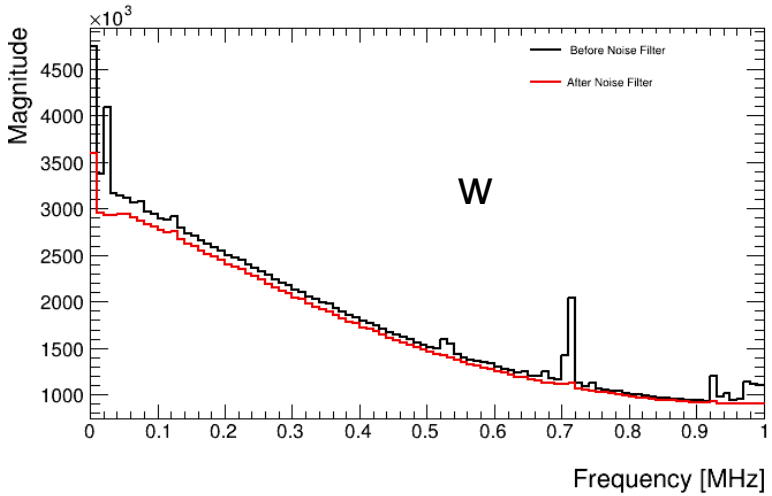


CRP3

noise







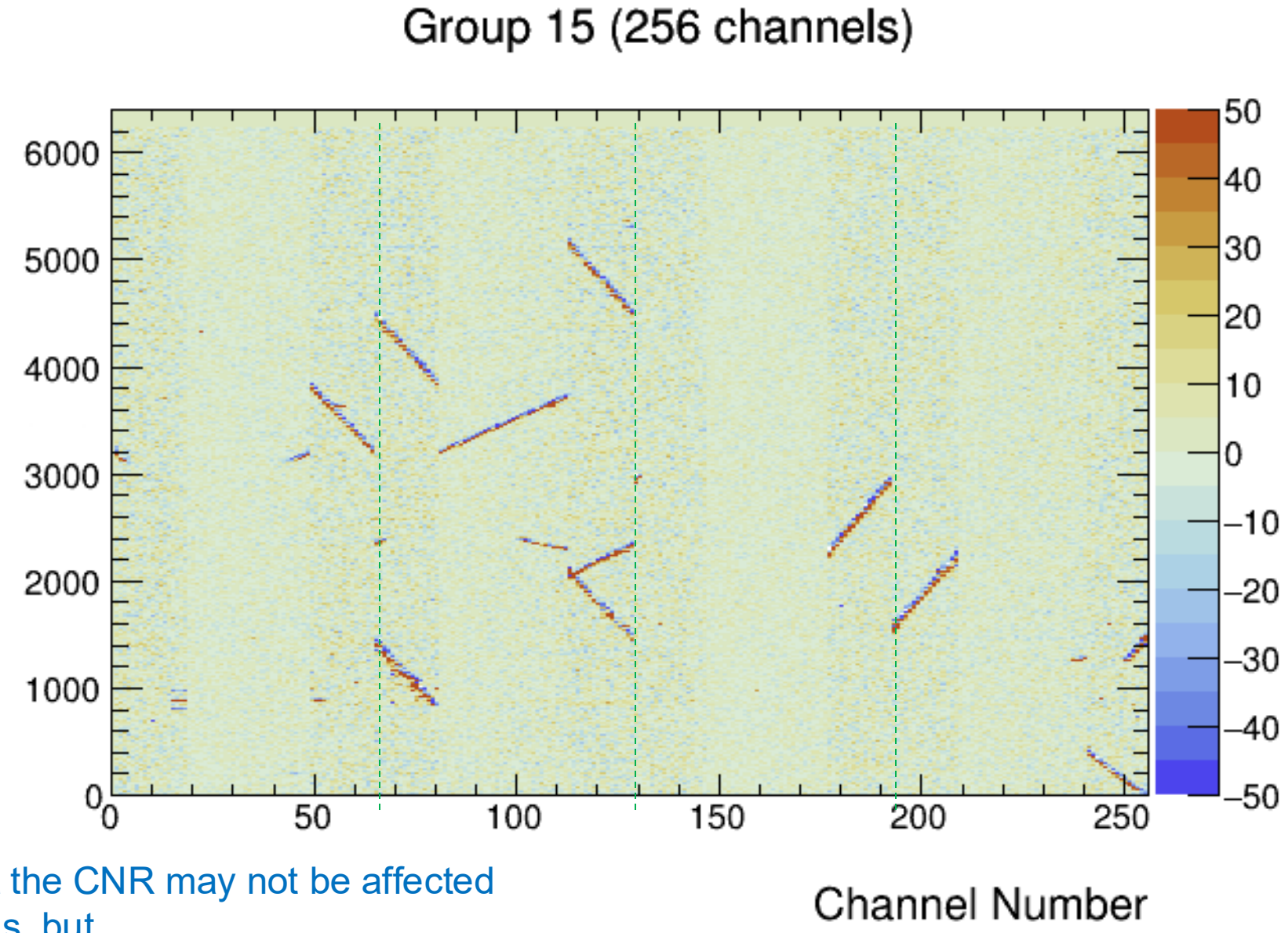
FEMB channel groups

The map is up to date.
The way we group is right.

- Each **crate** has FEMB labeled 1-24
- For a certain FEMB, each **stream** is from one COLDATA which is half-FEMB

- The difference we see here is because in they connect to 2 (or 3) type of planes.

Thanks Roger for explaination!

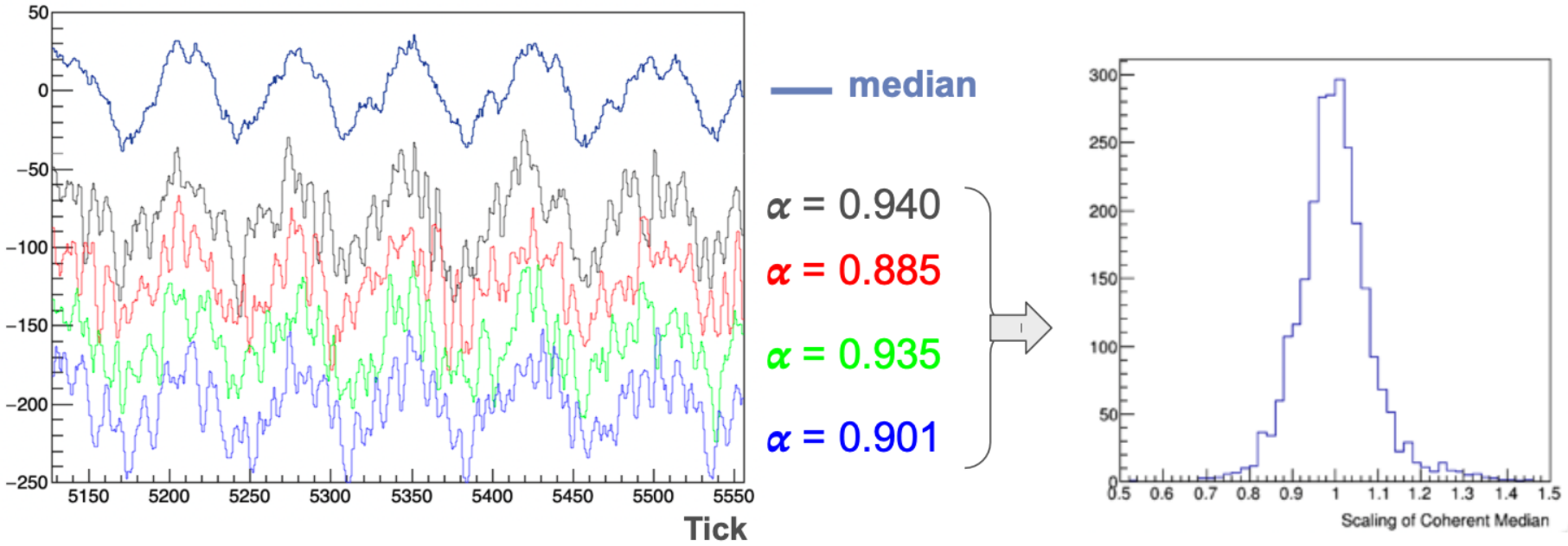


- At the beginning I thought the CNR may not be affected by the type of the channels, but…

CNR check

Plots from ProtoDUNE-HD

- Estimate the coherent noise waveform from grouped channels using **median**
- Scale the median waveform using linear regression (coherent noise ~ α * median)



- Subtract scaled median waveform from raw waveform
  - scaling factor α is derived from correlation coefficient between each channel and median waveform

Check α

Print out channel vs α

- clearly separation between different planes
- The group should be further separated according to plane

```
CoherentNoiseSub: ch=1444 response offset:0
1444 0.346258
1445 0.286447
1446 0.333597
1447 0.350074
1448 0.269872
1449 0.430576
1450 0.357251
1451 0.331853
1452 0.23778
1453 0.353605
1454 0.295621
1455 0.334175
1456 0.3435
1457 0.28909
1458 0.358006
1459 0.423131
2536 1.5
2537 1.5
2538 1.29208
2539 1.39928
2540 1.41983
2541 1.4511
2542 1.44343
2543 1.5
2544 1.5
2545 1.27468
2546 1.5
2547 1.38323
2548 1.33642
2549 1.41885
2550 1.24282
2551 1.27781
2552 1.34751
2553 1.45005
2554 1.20061
2555 1.20178
2556 1.03249
2557 1.05506
2558 1.01898
2559 1.07985
2560 1.15816
2561 1.13004
2562 1.10292
2563 1.03953
2564 1.00428
2565 1.08725
2566 1.04245
2567 1.13275
2568 0.946949
2569 0.984264
2570 1.06998
2571 1.01997
2572 1.08273
2573 0.969389
2574 1.10983
2575 1.16239
2576 1.08926
2577 1.20763
2578 1.08277
2579 1.09437
2580 1.1869
2581 1.28313
2582 1.16956
2583 1.43838
```

v plane

w plane

Bottom

```
CoherentNoiseSub: ch=6270 response offset:0
print out median: 6270
6270 1.20706
6271 1.22052
6272 1.09959
6273 1.15258
6274 1.22296
6275 1.14853
6276 1.10197
6277 1.22427
6278 1.11951
6279 1.16381
6280 1.22844
6281 1.07814
6282 1.13979
6283 1.12221
6284 1.17159
6285 1.17881
6286 1.06083
6287 1.09635
6288 1.08232
6289 1.07814
6290 1.07594
6291 1.02204
6292 1.09174
6293 0.817238
6294 1.02888
6295 0.979488
6296 0.96391
6297 0.959162
6298 1.10757
6299 1.04157
6300 1.03154
6301 1.22232
7318 1.00737
7319 0.801635
7320 0.838025
7321 0.878999
7322 0.854322
7323 0.849425
7324 0.847336
7325 0.89639
7326 0.919648
7327 0.868018
7328 0.867186
7329 0.889062
7330 0.930678
7331 0.879412
7332 0.891266
7333 0.901661
7334 0.952311
7335 0.883321
7336 0.902862
7337 0.832206
7338 0.904336
7339 0.86345
7340 0.883646
7341 0.960692
7342 0.935713
7343 0.860368
7344 0.911474
7345 0.836316
7346 1.02239
7347 0.919437
7348 0.919417
7349 1.0528
```

U plane

V plane

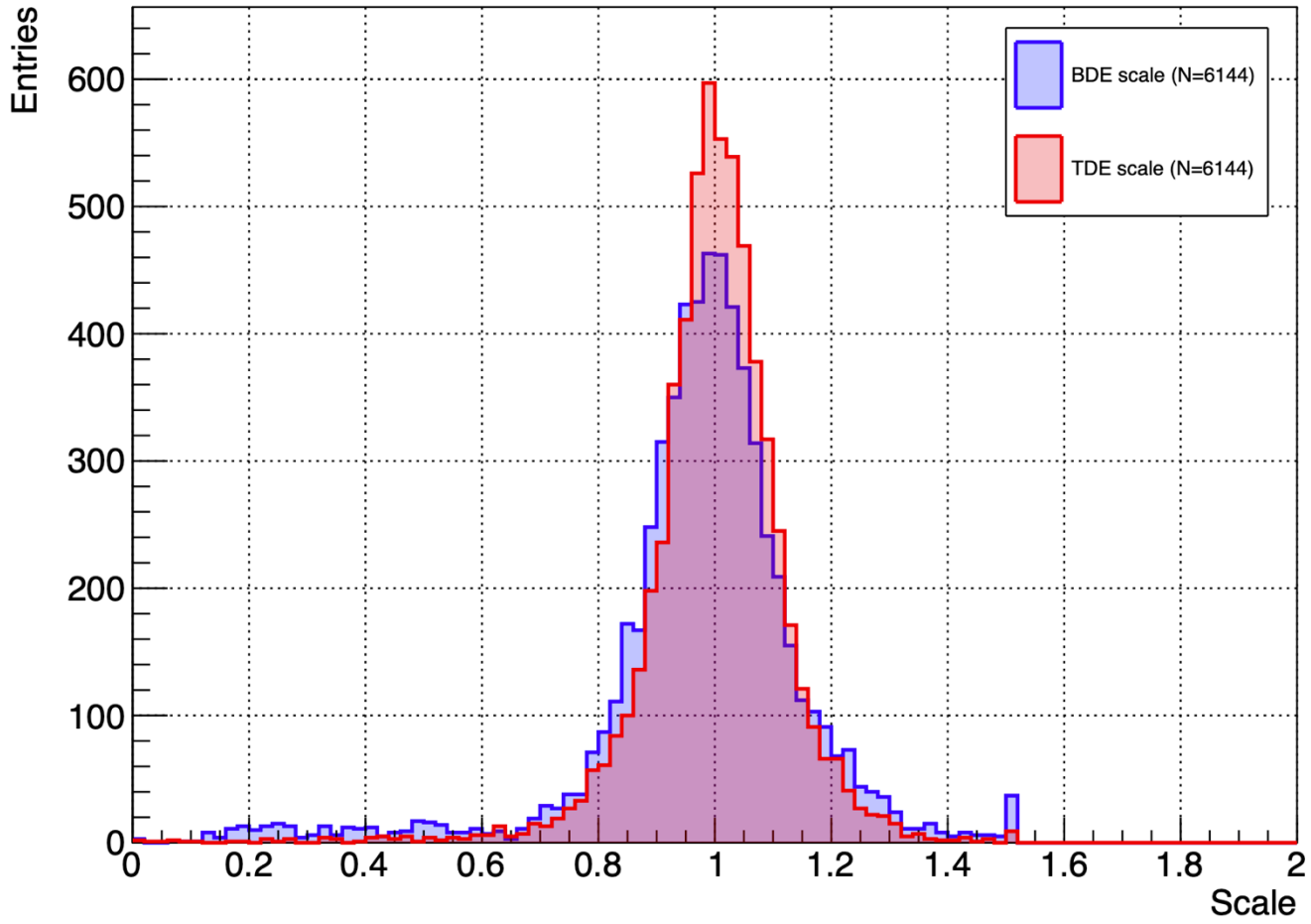
Top

New groups:

- Group channel for pdvd:
  - separate top and bottom, group offline channels according to femb, crate, stream, plane;
  - All channels will be divided in groups of different numbers.
    - 3;4;16;30;32;48

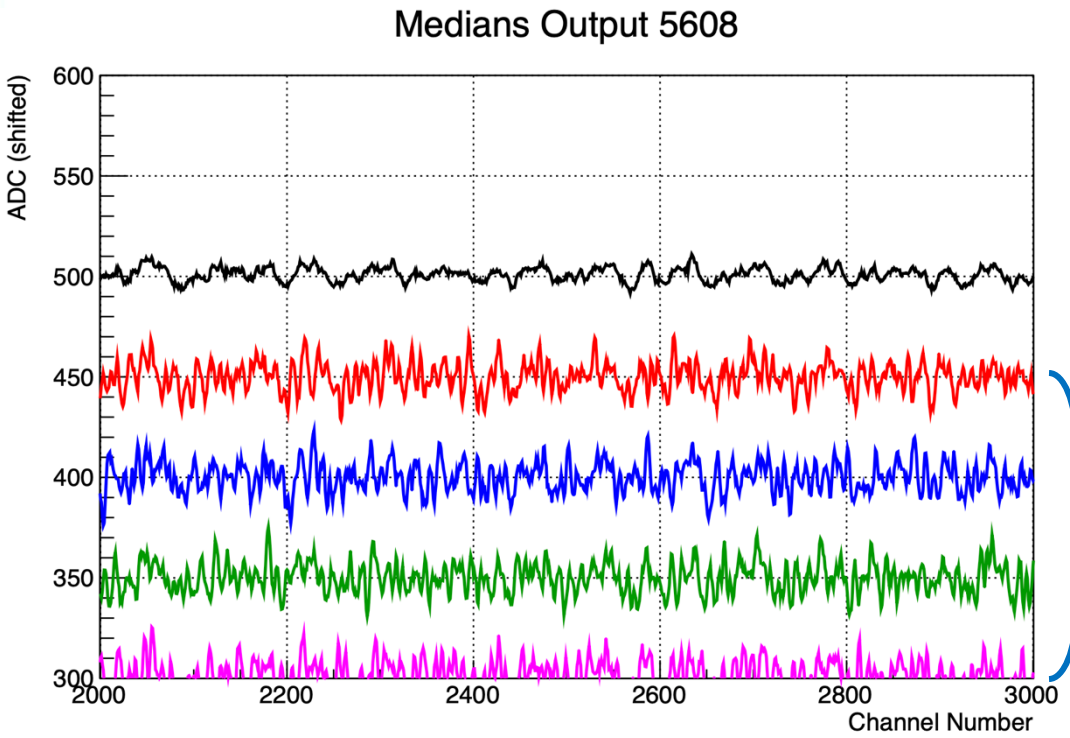
```
groups:[
[4492, 4493, 4494, 4495, 4496, 4497, 4498, 4499, 4484, 4485, 4486, 4487, 4488, 4489, 4490, 4491],
[5537, 5540, 5543, 5546, 5549, 5552, 5555, 5558, 5559, 5557, 5556, 5554, 5553, 5551, 5550, 5548, 5547, 5545, 5544, 5542, 5541, 5539, 5538, 5536, 5535, 5533, 5532, 5530, 5529, 5527, 5526, 5524, 5523,
[4476, 4477, 4478, 4479, 4480, 4481, 4482, 4483, 4468, 4469, 4470, 4471, 4472, 4473, 4474, 4475],
[5489, 5492, 5495, 5498, 5501, 5504, 5507, 5510, 5511, 5509, 5508, 5506, 5505, 5503, 5502, 5500, 5499, 5497, 5496, 5494, 5493, 5491, 5490, 5488, 5487, 5485, 5484, 5482, 5481, 5479, 5478, 5476, 5475,
[1435, 1434, 1433, 1432, 1431, 1430, 1429, 1428, 1443, 1442, 1441, 1440, 1439, 1438, 1437, 1436],
[2510, 2507, 2504, 2501, 2498, 2495, 2492, 2489, 2488, 2490, 2491, 2493, 2494, 2496, 2497, 2499, 2500, 2502, 2503, 2505, 2506, 2508, 2509, 2511, 2512, 2514, 2515, 2517, 2518, 2520, 2521, 2523, 2524,
[1451, 1450, 1449, 1448, 1447, 1446, 1445, 1444, 1459, 1458, 1457, 1456, 1455, 1454, 1453, 1452],
[2558, 2555, 2552, 2549, 2546, 2543, 2540, 2537, 2536, 2538, 2539, 2541, 2542, 2544, 2545, 2547, 2548, 2550, 2551, 2553, 2554, 2556, 2557, 2559, 2560, 2562, 2563, 2565, 2566, 2568, 2569, 2571, 2572,
[4460, 4461, 4462, 4463, 4464, 4465, 4466, 4467, 4452, 4453, 4454, 4455, 4456, 4457, 4458, 4459],
[5441, 5444, 5447, 5450, 5453, 5456, 5459, 5462, 5463, 5461, 5460, 5458, 5457, 5455, 5454, 5452, 5451, 5449, 5448, 5446, 5445, 5443, 5442, 5440, 5439, 5437, 5436, 5434, 5433, 5431, 5430, 5428, 5427,
[4444, 4445, 4446, 4447, 4448, 4449, 4450, 4451, 4436, 4437, 4438, 4439, 4440, 4441, 4442, 4443],
[5393, 5396, 5399, 5402, 5405, 5408, 5411, 5414, 5415, 5413, 5412, 5410, 5409, 5407, 5406, 5404, 5403, 5401, 5400, 5398, 5397, 5395, 5394, 5392, 5391, 5389, 5388, 5386, 5385, 5383, 5382, 5380, 5379,
[1467, 1466, 1465, 1464, 1463, 1462, 1461, 1460, 1475, 1474, 1473, 1472, 1471, 1470, 1469, 1468],
[2606, 2603, 2600, 2597, 2594, 2591, 2588, 2585, 2584, 2586, 2587, 2589, 2590, 2592, 2593, 2595, 2596, 2598, 2599, 2601, 2602, 2604, 2605, 2607, 2608, 2610, 2611, 2613, 2614, 2616, 2617, 2619, 2620,
[1483, 1482, 1481, 1480, 1479, 1478, 1477, 1476, 1491, 1490, 1489, 1488, 1487, 1486, 1485, 1484],
[2654, 2651, 2648, 2645, 2642, 2639, 2636, 2633, 2632, 2634, 2635, 2637, 2638, 2640, 2641, 2643, 2644, 2646, 2647, 2649, 2650, 2652, 2653, 2655, 2656, 2658, 2659, 2661, 2662, 2664, 2665, 2667, 2668,
[4428, 4429, 4430, 4431, 4432, 4433, 4434, 4435, 4420, 4421, 4422, 4423, 4424, 4425, 4426, 4427],
[5345, 5348, 5351, 5354, 5357, 5360, 5363, 5366, 5367, 5365, 5364, 5362, 5361, 5359, 5358, 5356, 5355, 5353, 5352, 5350, 5349, 5347, 5346, 5344, 5343, 5341, 5340, 5338, 5337, 5335, 5334, 5332, 5331,
[4412, 4413, 4414, 4415, 4416, 4417, 4418, 4419, 4404, 4405, 4406, 4407, 4408, 4409, 4410, 4411],
[5297, 5300, 5303, 5306, 5309, 5312, 5315, 5318, 5319, 5317, 5316, 5314, 5313, 5311, 5310, 5308, 5307, 5305, 5304, 5302, 5301, 5299, 5298, 5296, 5295, 5293, 5292, 5290, 5289, 5287, 5286, 5284, 5283,
[1499, 1498, 1497, 1496, 1495, 1494, 1493, 1492, 1507, 1506, 1505, 1504, 1503, 1502, 1501, 1500],
[2702, 2699, 2696, 2693, 2690, 2687, 2684, 2681, 2680, 2682, 2683, 2685, 2686, 2688, 2689, 2691, 2692, 2694, 2695, 2697, 2698, 2700, 2701, 2703, 2704, 2706, 2707, 2709, 2710, 2712, 2713, 2715, 2716,
[1515, 1514, 1513, 1512, 1511, 1510, 1509, 1508, 1523, 1522, 1521, 1520, 1519, 1518, 1517, 1516],
[2750, 2747, 2744, 2741, 2738, 2735, 2732, 2729, 2728, 2730, 2731, 2733, 2734, 2736, 2737, 2739, 2740, 2742, 2743, 2745, 2746, 2748, 2749, 2751, 2752, 2754, 2755, 2757, 2758, 2760, 2761, 2763, 2764,
[3547, 3546, 3545, 3544, 3543, 3542, 3541, 3540, 3539, 3538, 3537, 3536, 3535, 3534, 3533, 3532, 3531, 3530, 3529, 3528, 3527, 3526, 3525, 3524, 3523, 3522, 3521, 3520, 3519, 3518],
[4390, 4391, 4392, 4393, 4394, 4395, 4396, 4397, 4398, 4399, 4400, 4401, 4402, 4403, 4374, 4375, 4376, 4377, 4378, 4379, 4380, 4381, 4382, 4383, 4384, 4385, 4386, 4387, 4388, 4389],
[5268, 5270, 5271, 5269],
[3517, 3516, 3515, 3514, 3513, 3512, 3511, 3510, 3509, 3508, 3507, 3506, 3505, 3504, 3503, 3502, 3501, 3500, 3499, 3498, 3497, 3496, 3495, 3494, 3493, 3492, 3491, 3490, 3489, 3488, 3487, 3486],
[4358, 4359, 4360, 4361, 4362, 4363, 4364, 4365, 4366, 4367, 4368, 4369, 4370, 4371, 4372, 4373, 4342, 4343, 4344, 4345, 4346, 4347, 4348, 4349, 4350, 4351, 4352, 4353, 4354, 4355, 4356, 4357],
[476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505],
[1537, 1536, 1535, 1534, 1533, 1532, 1531, 1530, 1529, 1528, 1527, 1526, 1525, 1524, 1553, 1552, 1551, 1550, 1549, 1548, 1547, 1546, 1545, 1544, 1543, 1542, 1541, 1540, 1539, 1538],
[2779, 2777, 2776, 2778],
[506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537],
[1569, 1568, 1567, 1566, 1565, 1564, 1563, 1562, 1561, 1560, 1559, 1558, 1557, 1556, 1555, 1554, 1585, 1584, 1583, 1582, 1581, 1580, 1579, 1578, 1577, 1576, 1575, 1574, 1573, 1572, 1571, 1570],
```

α distribution in new groups:



Raw waveform and median

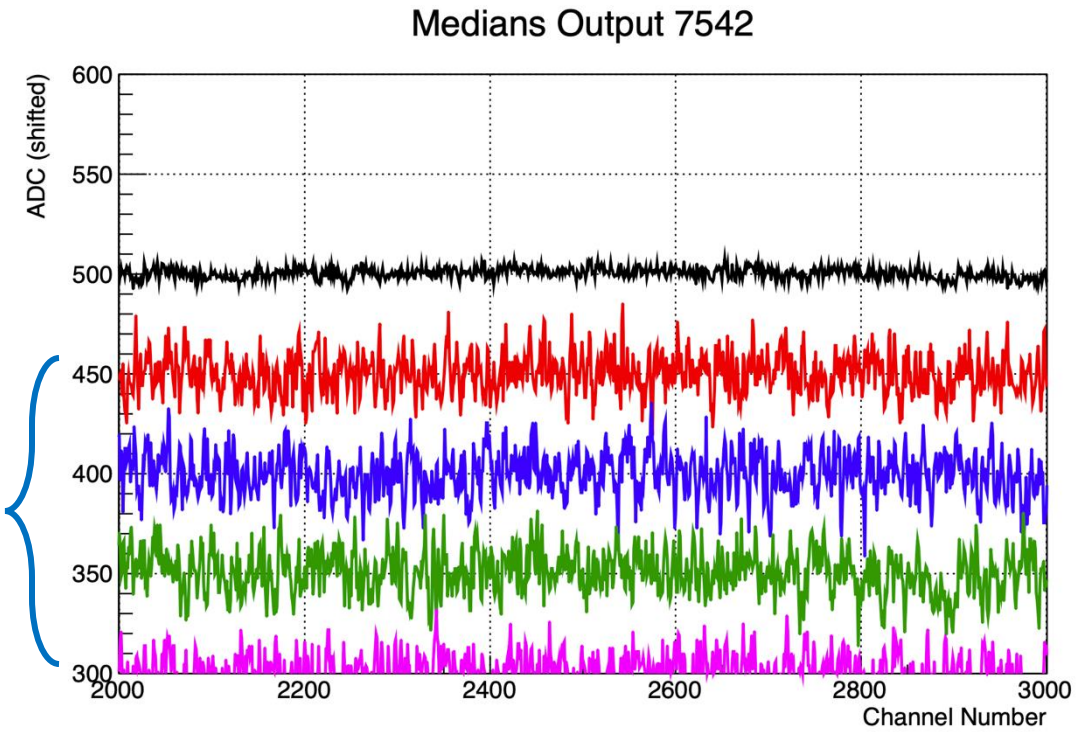
Bottom



median

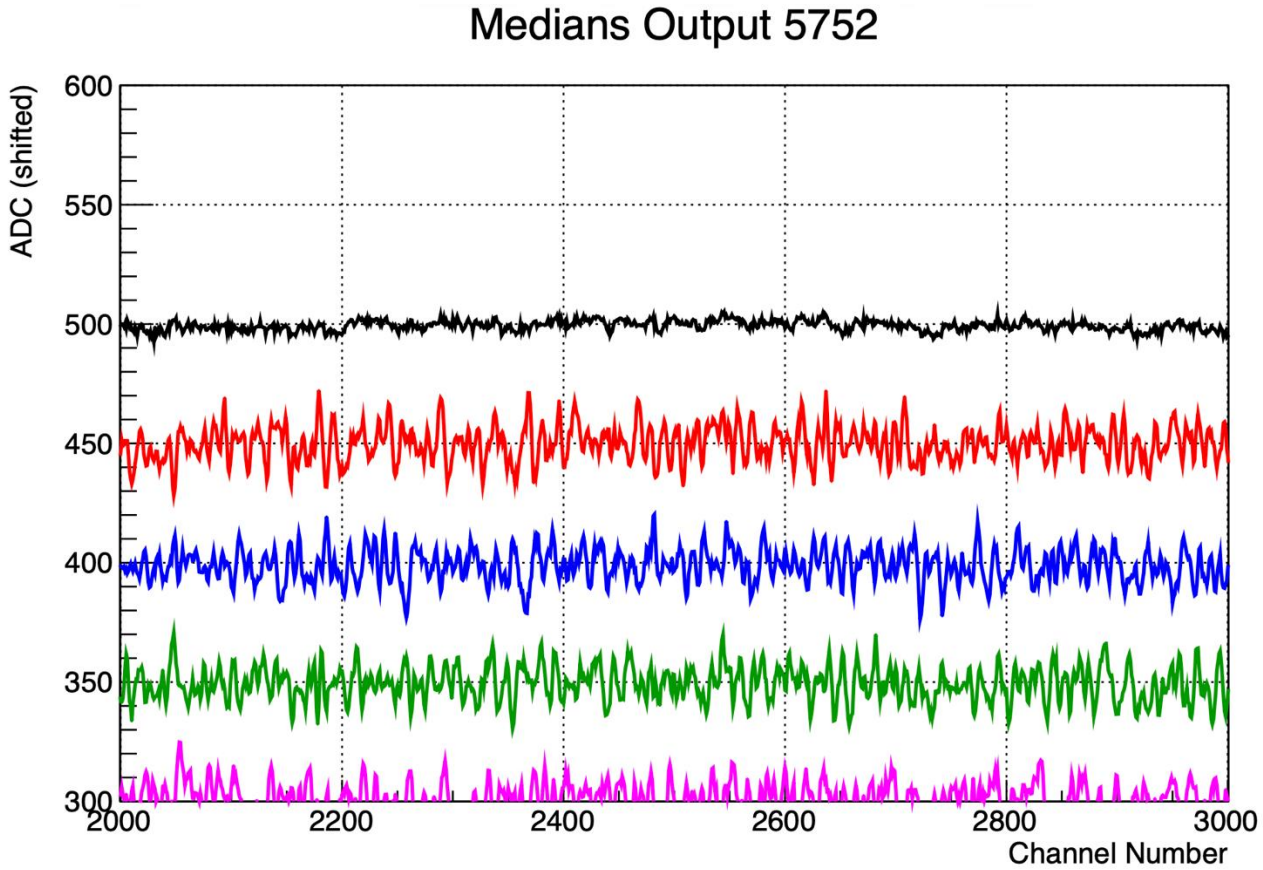
Raw waveforms

Top



A problematic case:

```
CoherentNoiseSub: ch=5752 response offset:0
5752 0.580068
5753 0.976333
5754 0.916009
5755 0.993078
5756 0.906328
5757 0.871398
5758 0.608601
5759 0.756006
5760 0.987007
5761 0.951296
5762 0.853426
5763 1.18283
5764 1.02673
5765 0.850676
5766 0.962293
5767 1.31905
5768 0.988111
5769 1.16056
5770 1.5
5771 1.49694
5772 1.14595
5773 0.963998
5774 0.97006
5775 0.860401
5776 0.773995
5777 0.872471
5778 0.854162
5779 1.04718
5780 1.25411
5781 1.01969
5782 0.816197
5783 0.468119
5784 0.713335
5785 0.806933
5786 0.472585
5787 0.71571
5788 0.946
5789 0.407932
5790 0.852226
5791 0.917154
5792 0.880963
5793 0.767942
5794 0.605548
5795 0.721979
5796 0.713439
5797 1.5
5798 0.725952
5799 1.01552
```



Other problem; partial check:

- This partial check by default is on
- Almost all top channels, some bottom channels are found partial
- Then apply adaptative baseline.

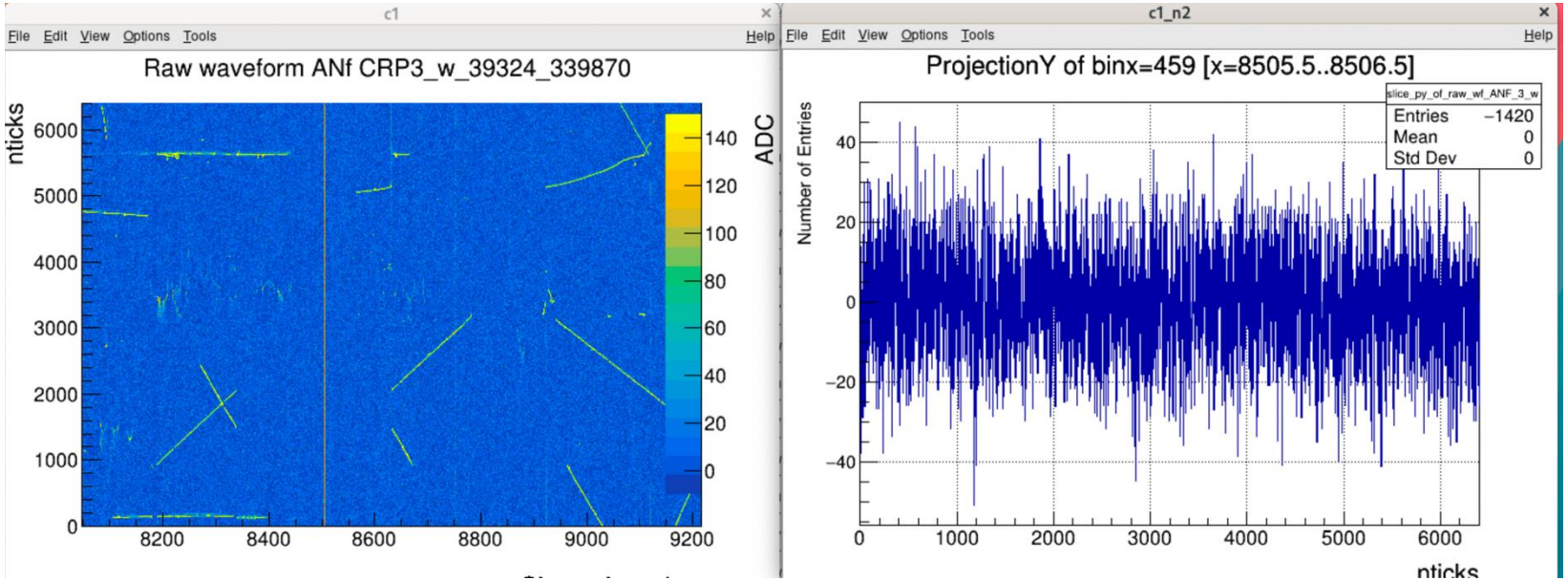
```
bool Diagnostics::Partial::operator()(const WireCell::Waveform::compseq_t& spec) con
{
    const double mag0 = std::abs(spec[0 + 1]);
    double sum = mag0;
    for (int ind = 1; ind <= nfreqs && ind < (int) spec.size(); ++ind) {
        const double magi = std::abs(spec[ind + 1]);
        if (mag0 <= magi) {
            return false;
        }
        sum += magi;
    }
    return sum / (nfreqs + 1) > maxpower;
}
```

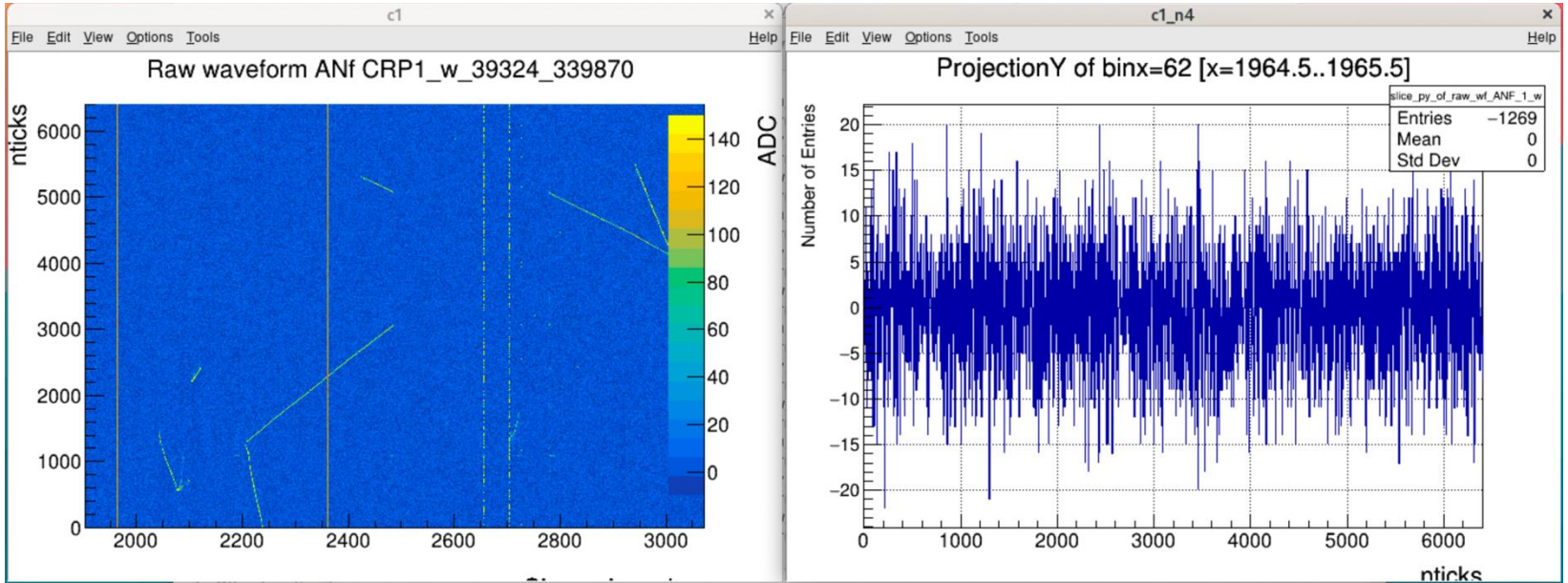
- Modify this for Top?
- maxpower 6000 may need to be changed?
- 512-tick window need to be changed.

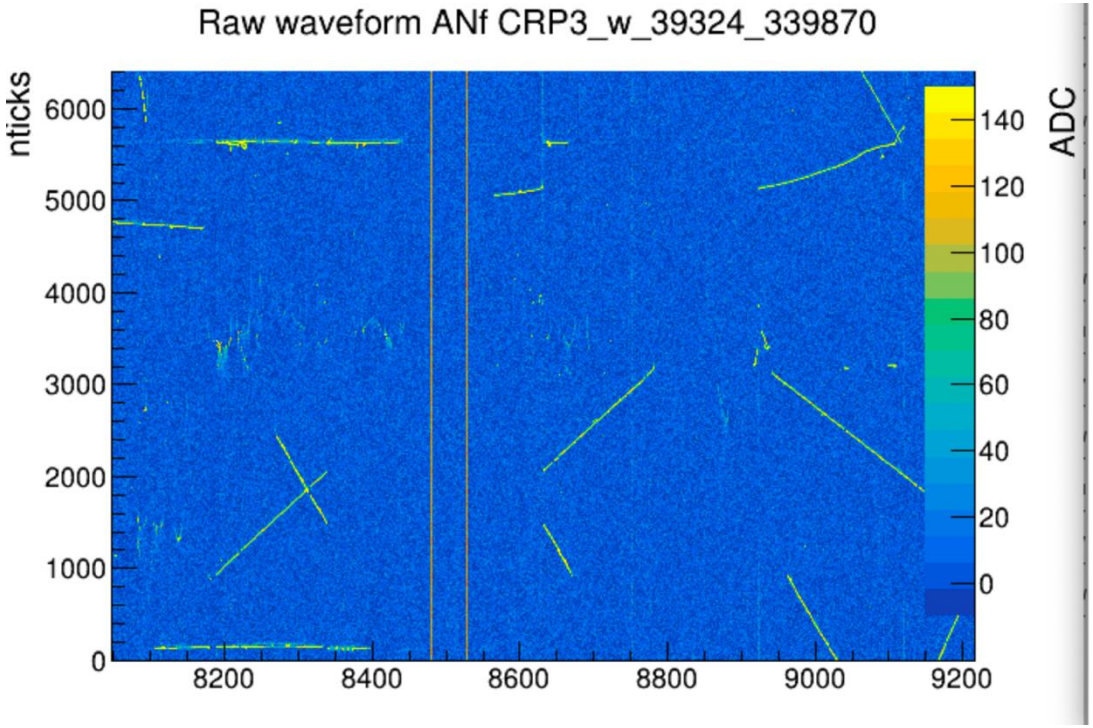
Algorithm:

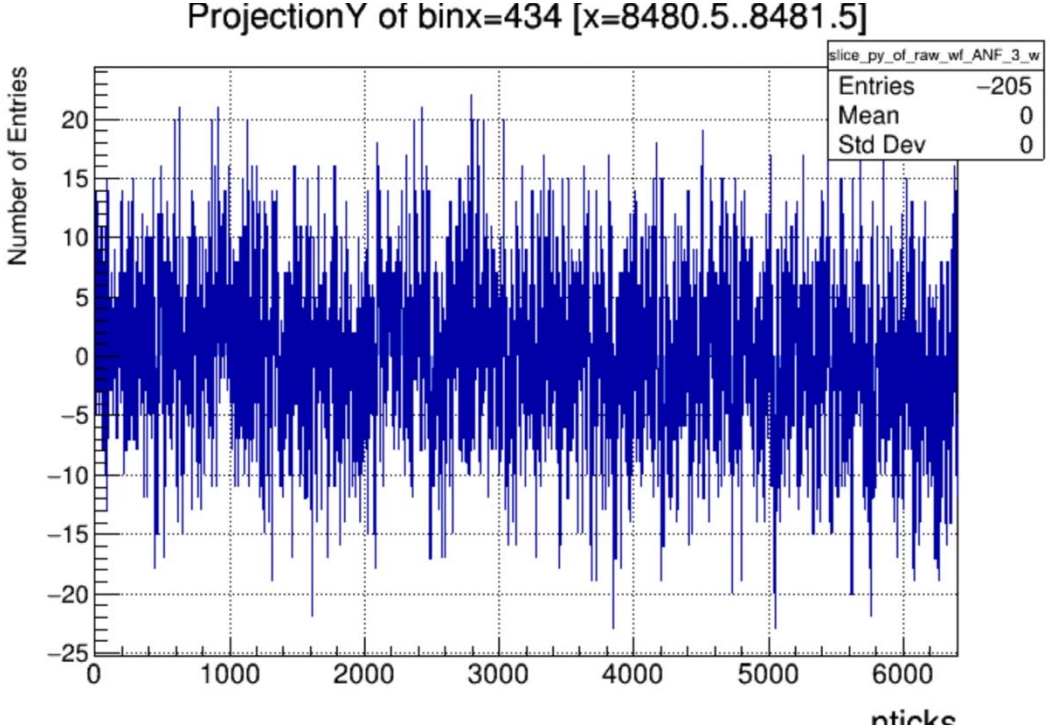
1. Checks if the DC component (lowest frequency) dominates over the next `nfreqs` lowest frequencies
2. Verifies all these low frequencies have magnitudes less than the DC component
3. Calculates the average power of these low frequencies
4. Returns `true` if the average exceeds `maxpower` (default 6000.0)

- `RawAdapativeBaselineAlg()`: Calculates and subtracts an adaptive baseline using a sliding window approach (512-tick windows), interpolating baseline values between valid points


c1
File Edit View Options Tools Help
Raw waveform ANf CRP3_w_39324_339870
nticks
ADC
c1_n2
File Edit View Options Tools Help
ProjectionY of binx=459 [x=8505.5..8506.5]
slice_py_of_raw_wf_ANF_3_w
Entries −1420
Mean 0
Std Dev 0
Number of Entries
nticks


c1
File Edit View Options Tools Help
Raw waveform ANf CRP1_w_39324_339870
nticks
ADC
c1_n4
File Edit View Options Tools Help
ProjectionY of binx=62 [x=1964.5..1965.5]
slice_py_of_raw_wf_ANF_1_w
Entries −1269
Mean 0
Std Dev 0
Number of Entries
nticks


Raw waveform ANf CRP3_w_39324_339870
nticks
6000
5000
4000
3000
2000
1000
0
8200
8400
8600
8800
9000
9200
ADC
140
120
100
80
60
40
20
0


ProjectionY of binx=434 [x=8480.5..8481.5]
slice_py_of_raw_wf_ANF_3_w
Entries −205
Mean 0
Std Dev 0
Number of Entries
20
15
10
5
0
−5
−10
−15
−20
−25
0
1000
2000
3000
4000
5000
6000
nticks