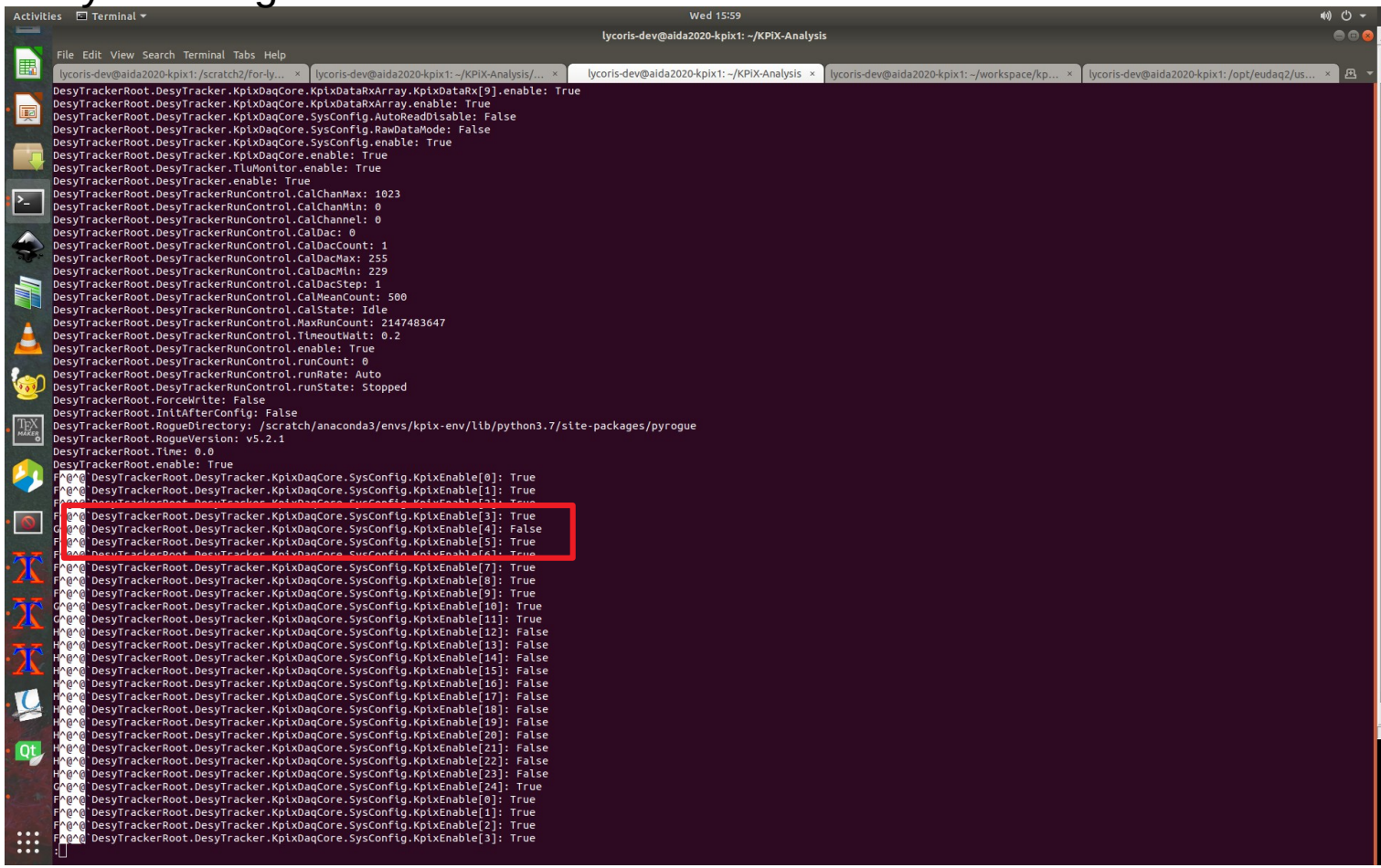
New DAQ version

- Data taking appears to be fine but Calibration randomly disables some KpiX even thought it was not set in the yml config file?



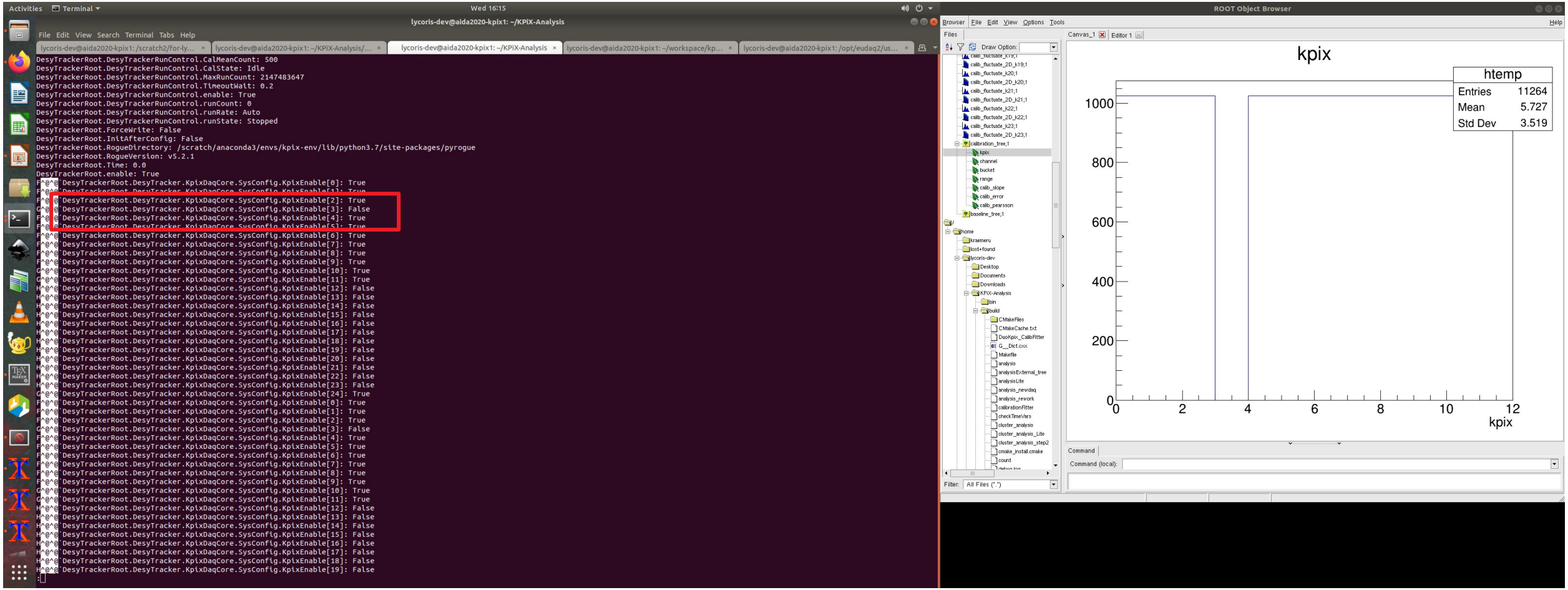
New DAQ version

- Data taking appears to be fine but Calibration randomly disables some KpiX even thought it was not set in the yml config file?
- I even made sure again that this is the case

```
KpixAsic[0]:
  enable: 'True'
KpixAsic[1]:
  enable: 'True'
KpixAsic[2]:
  enable: 'True'
KpixAsic[3]:
  enable: 'True'
KpixAsic[4]:
  enable: 'True'
KpixAsic[5]:
  enable: 'True'
KpixAsic[6]:
  enable: 'True'
KpixAsic[7]:
  enable: 'True'
KpixAsic[8]:
  enable: 'True'
KpixAsic[9]:
  enable: 'True'
KpixAsic[10]:
  enable: 'True'
KpixAsic[11]:
  enable: 'True'
KpixAsic[24]:
  enable: 'True'
```

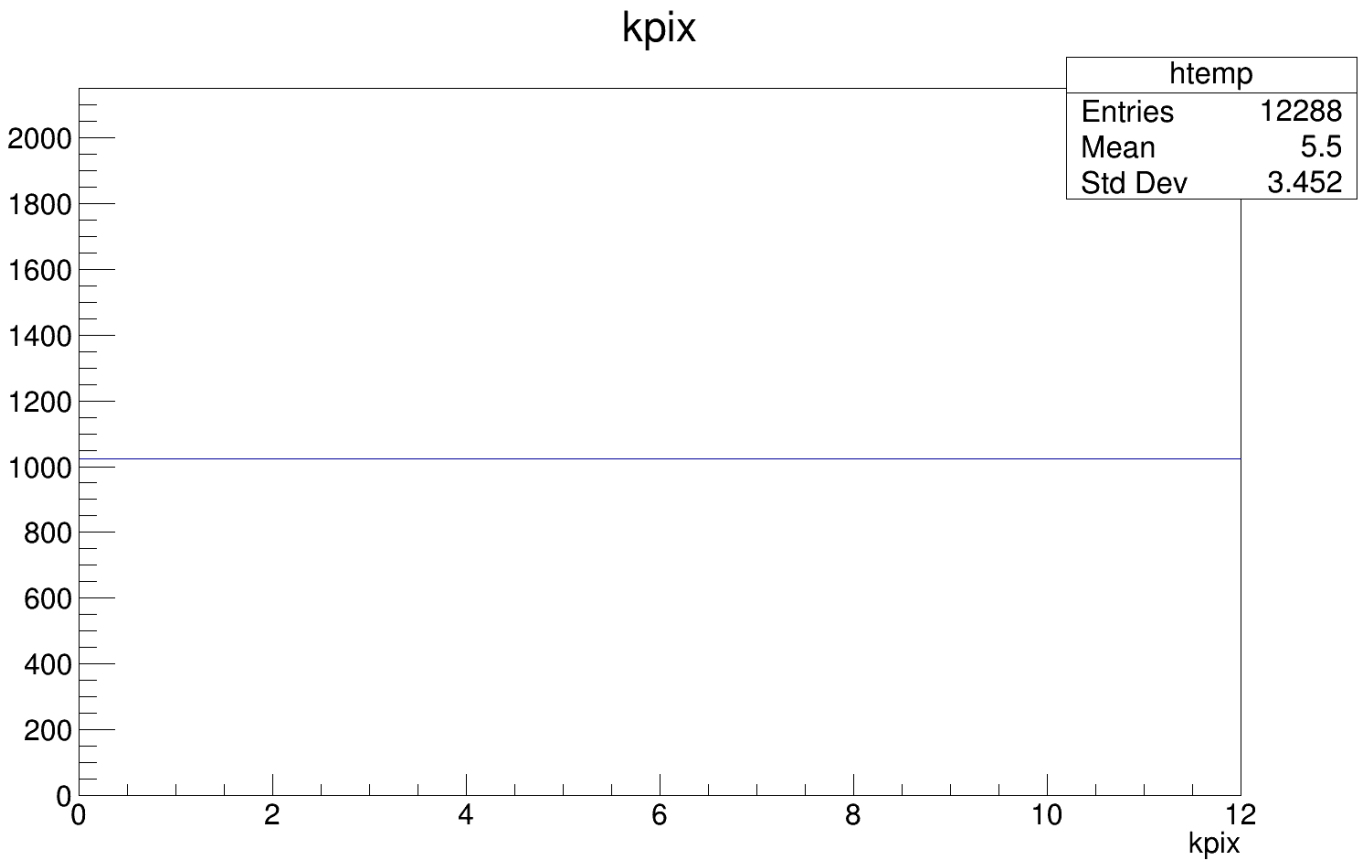
New DAQ version

- In a repeat with larger step size (5 instead of 1) it then disabled a different KpiX?



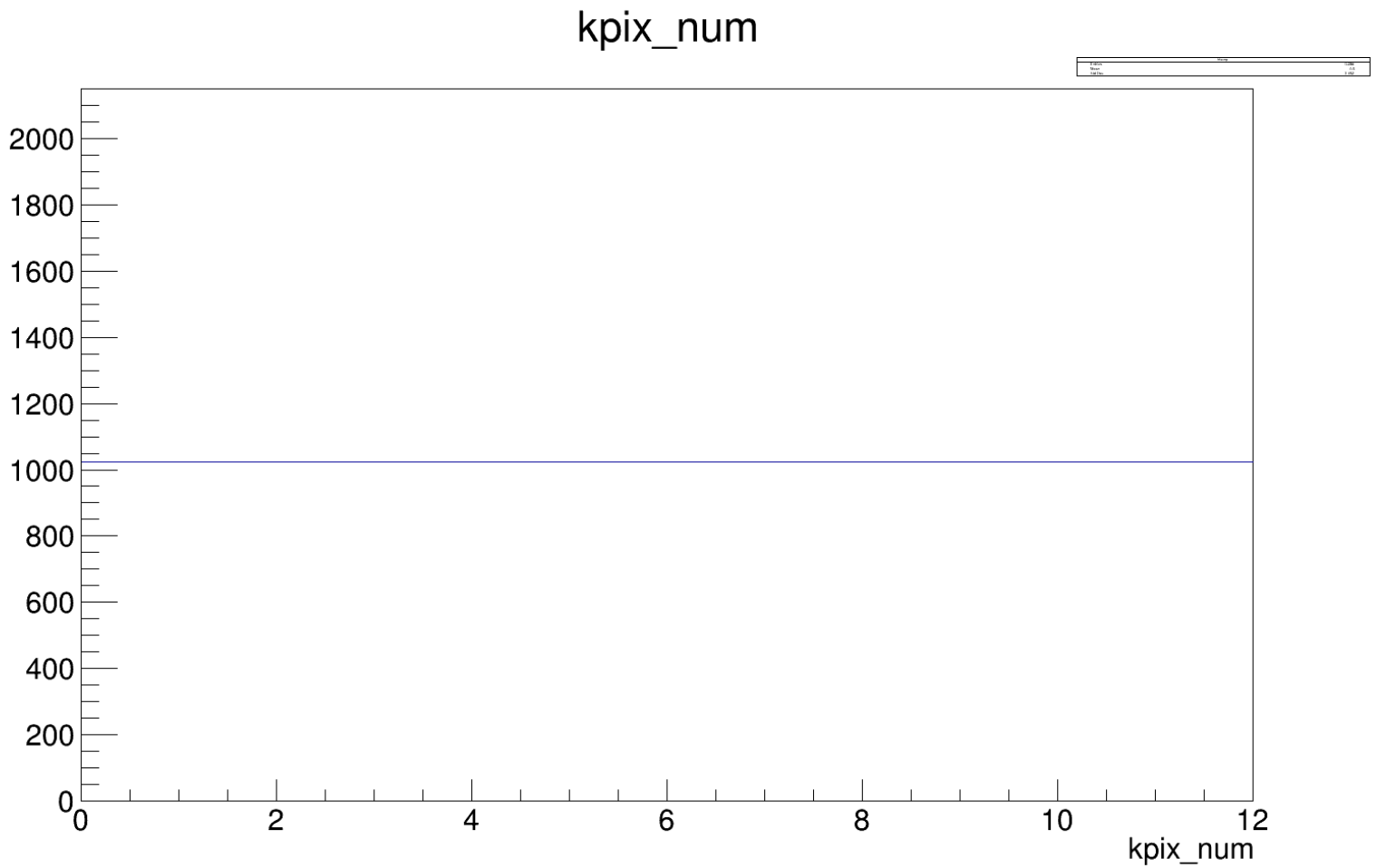
New DAQ version

- In a repeat with larger step size (5 instead of 1) it then disabled a different KpiX?
- Repeating both measurements does not show that issue again. No clue where it came from or where it went.



New DAQ version

- Normal running operation with TLU auto trigger as input seems to be fine with no missing KpiX at least in 2/2 runs. Gonna take a long run tomorrow to check if that works.



Defining Efficiency

- Current method of defining efficiency based on ratio of 6 hit to 5 hit is biased.
- Golden method would be to use Mimosa tracks to verify Lycoris hit numbers.
- But Mimosa frame readout time is ~230 us while Lycoris frame readout time is between 2.5 and 20 us (last testbeam was set to 20 us)

- Idea is to gradually reduce Signal over Noise (S/N) from the clustering further and further until every mimosa track finds a match even when this is noise. And slowly increase it as this should give us an S curve from which we can then choose the best compromise between Efficiency and Purity.

- Issue was that no matter how loose the cut could be chosen I was never able to match 1 track 1 hit.

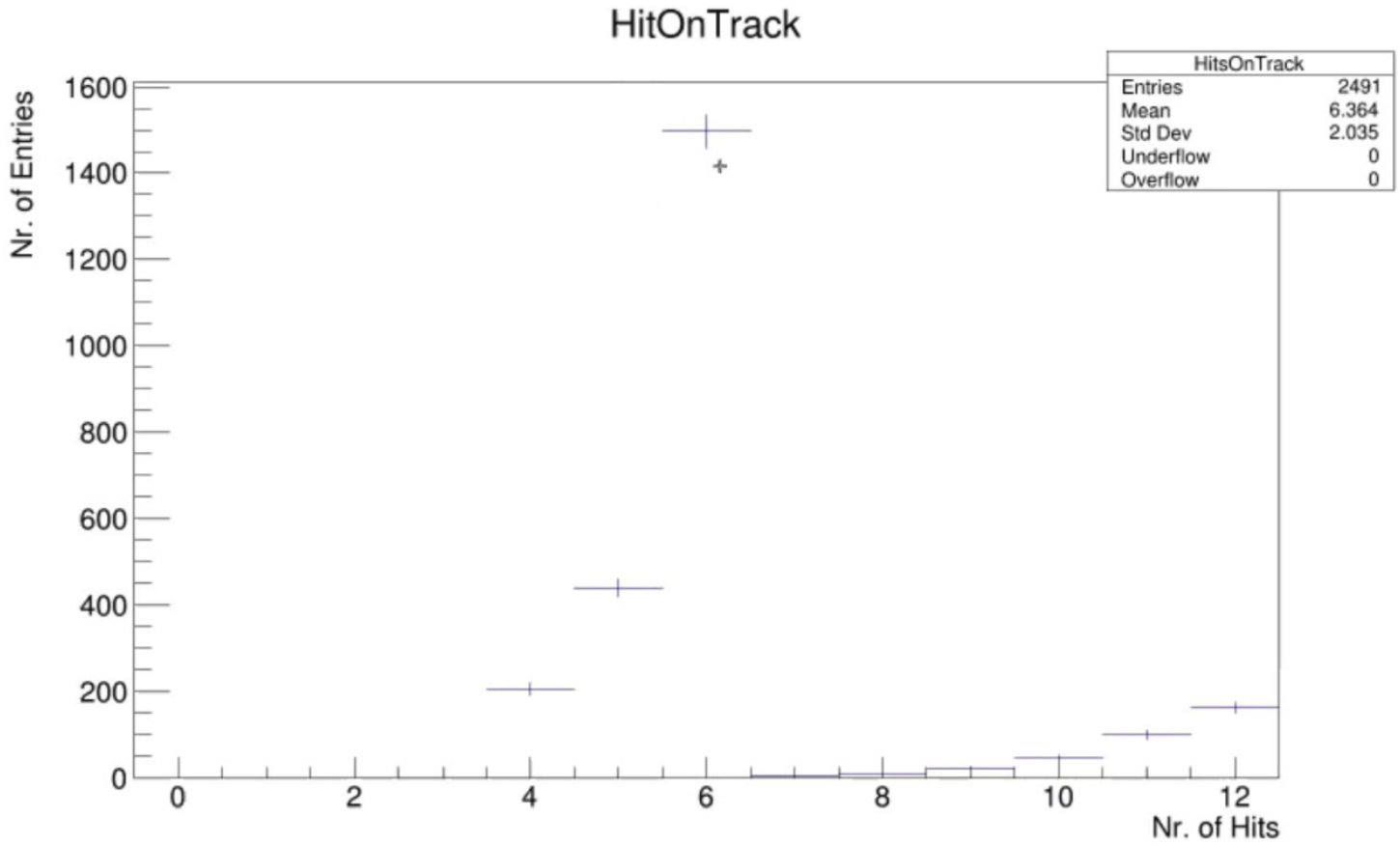
Defining Efficiency

- Plot shows this deficiency Out of 2400 tracks only a small fraction has more than 6 hits (6 mimosa + N lycoris)
- This was done with an extremely low threshold of cluster input SoN (minimum SoN a strip needs to have to be clustered) of 0.5 and no requirement on the finalized cluster.

Strip SoN for input into cluster = 0.5

No cluster SoN requirement

No uniqueness requirement of matching hits



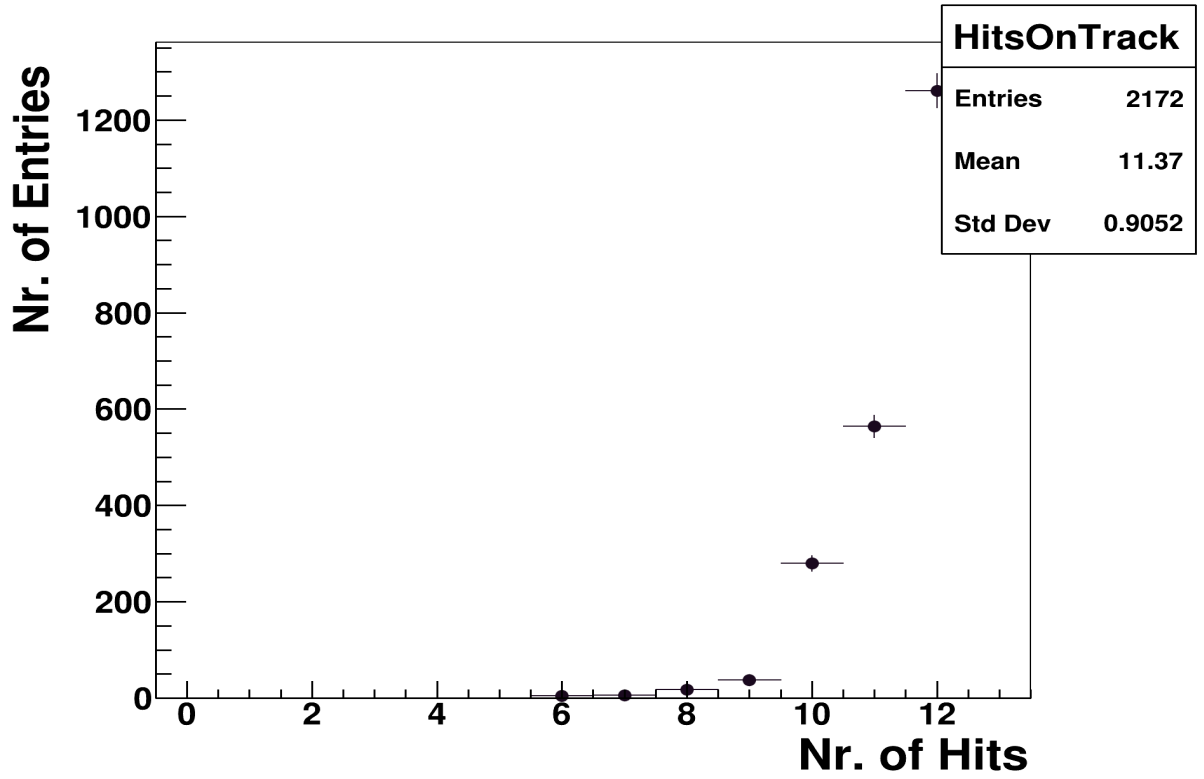
Defining Efficiency

- This problem was solved today as can be seen in the plot below.
- Mimosa is not power pulsed and not limited in the number of events.
  - While KpiX records 1 trigger per spill (because buckets 1-3 are unuseable)
  - Mimosa can record triggers as long as triggers arrive.
- This picture is more flattering than it actually is because all cuts were lowered extremely wide.

Strip SoN for input into cluster = 0.5

No cluster SoN requirement

No uniqueness requirement of matching hits



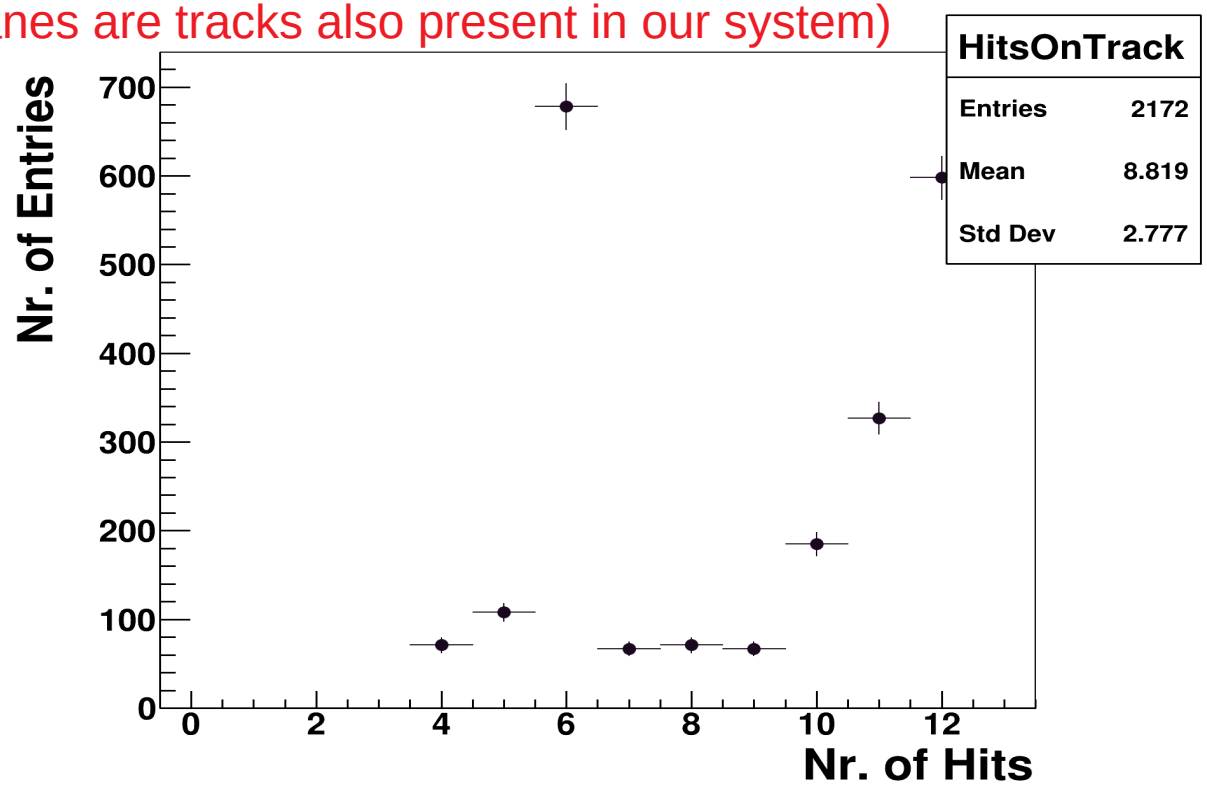
Defining Efficiency

- Below is what it looks like for my normal clustering input currently still without uniqueness and this picture is closer to the truth.
- The problem is that Mimosa readout time is 230us as opposed to KpiX 20 us. Time resolution of Mimosa is quite bad and except for event number no further timing information is available
  - Just because a track is within the mimosa event data does not mean it is also in the KpiX event data.
  - As such we would not expect to have 100% matches and it shows that the previous plot has a lot of noise matches
  - Need to find a way to determine which tracks are present in both data streams (current plan is to say only hits with at least 3 matched lycoris planes are tracks also present in our system)

Strip SoN for input into cluster > 2.0

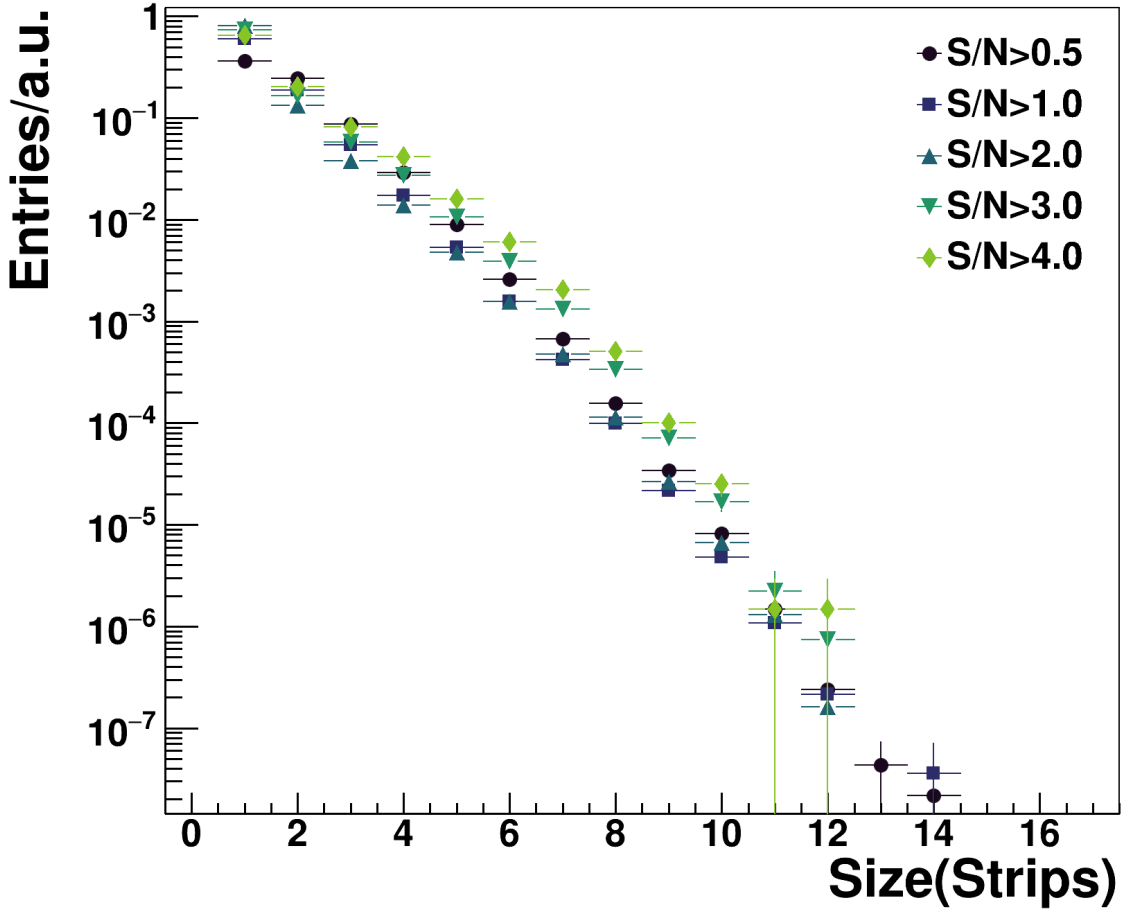
Cluster SoN requirement > 4.0

No uniqueness requirement of matching hits



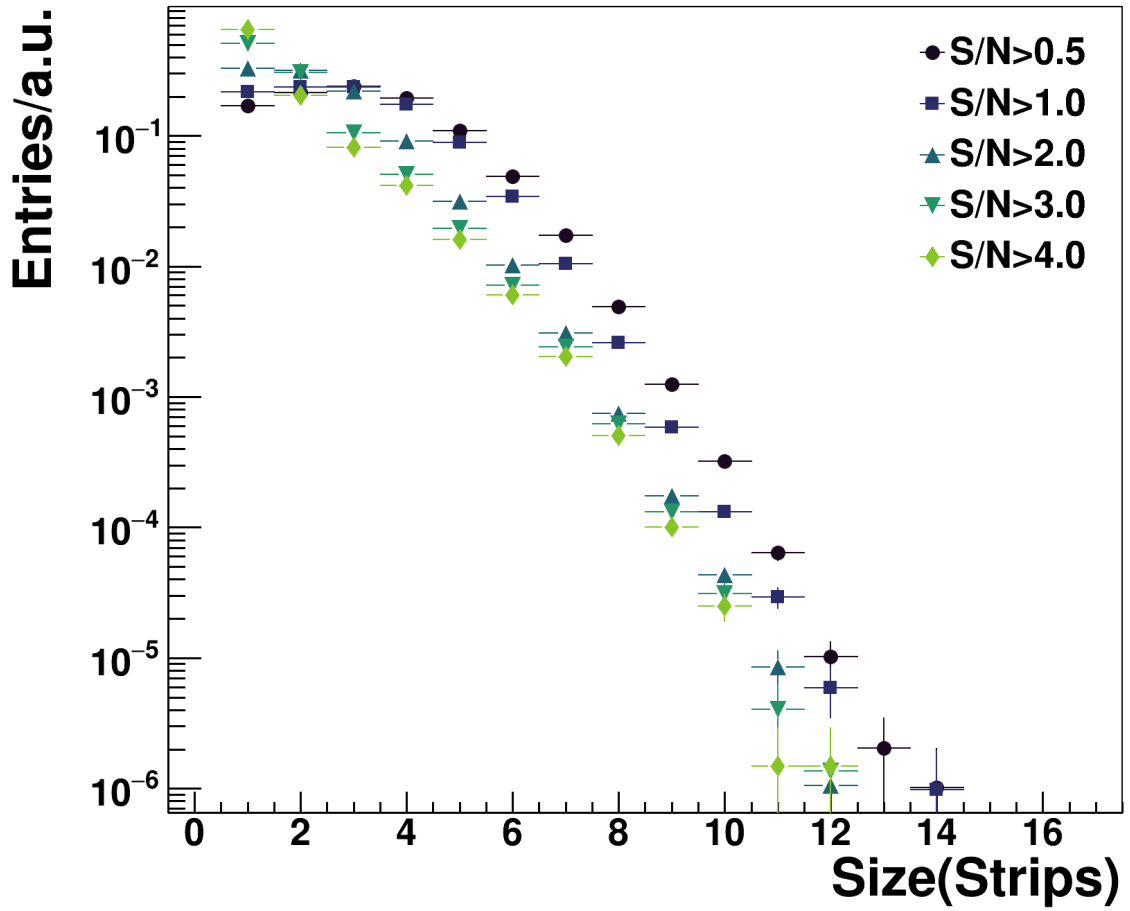
Investigating the Strip SoN Cut

- Also decided to investigate some more my cut criteria to see if they are sensible.
- One thing I checked was how the cluster size changes with the input SoN cut.
- To the right we can see the result for all reconstructed clusters.
- This makes it seem like S/N>4 results in more high size clusters.
- But this is basically just an illusion resulting from the norming to the total number of entries
- S/N>0.5 = 4E7 entries
- S/N>4.0 = 6E5 entries



Investigating the Strip SoN Cut

- When looking only at clusters whose combined SoN is greater than 4 the picture becomes a lot clearer.
- Almost all of them have the same amount of entries (7E5 to 1E6)
- Now it becomes clear that with increased strip SoN cut the cluster size shifts to lower average sizes
- This plot also nicely shows that our S/N>2 cut is fairly well chosen as we expect a roughly equal amount of 1 and 2 strip clusters which is only the case for S/N>2. (though one could of course fine tune it further)

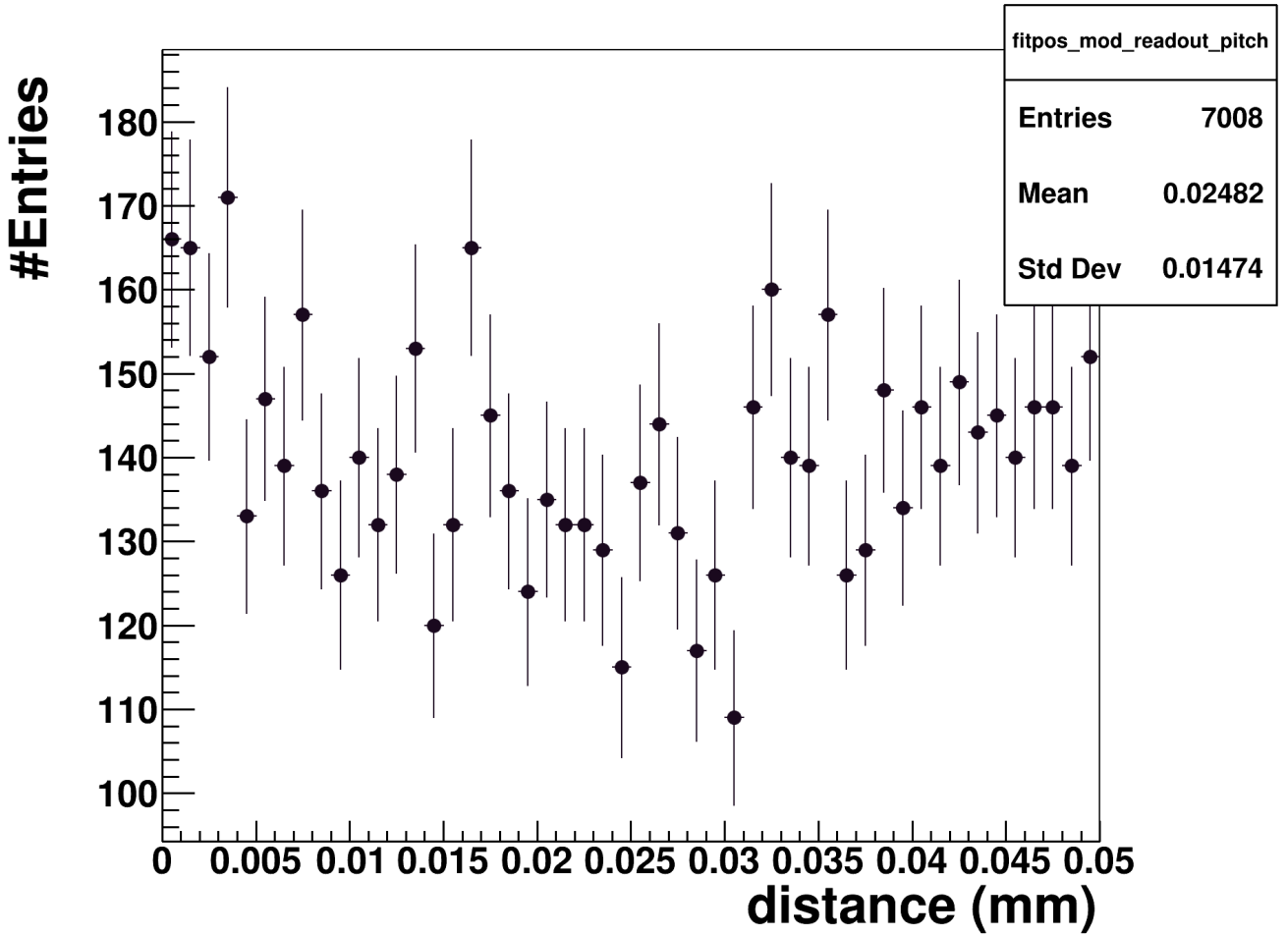


Hit distribution on the sensor

- The question came up on the relation between the cluster position relative to the hit position on the sensor.
- Similar to my floating and strip charge distributions the hit position was determined using the Mimosa telescope
- It does look like there are fluctuations resulting in more hits at the edge but considering the size of the errorbars this is not really significant.
- When applying the same cut as Dieter I find

  (d<0.125 || d >0.375) → Readout
  (0.125<d<0.375) → Floating
  - 3637 readout hits = 51.8% of all hits
  - 3371 floating hits = 48.2% of all hits

- I intend to check how this changes with different clustering input cuts



Hit distribution on the sensor

- Just to make sure the issue is not in the file I repeated the measurement on a subsample of the file Dieter analyzes
- At least in the sub sample I get basically the same result
- 2137 readout hits = 51.6% of all hits
- 2006 floating hits = 48.4% of all hits