

Helium-based gas mixtures : status of the analysis review

E.Bossini

Analysis summary

Steps:

- review the results obtained with the PISA-01 and REND-01 telescopes, fluxed with He+R1234ze.
- Investigate (& solve) eventual critical aspects.
- Proceed to publication

DATA: acquired by the two stations in the period September 2021 – June 2022. To be compared with std (R134a+SF6) gas mixture.

ANALYSIS OUTPUT:

- | | |
|--|---|
| <ul style="list-style-type: none">• Efficiency• Cluster multiplicity• Streamer fraction | } Test chamber with He mixture, triggering chambers with std mixture
Dedicated reco. |
| <ul style="list-style-type: none">• Angular distribution, speed, TOF of the reconstructed tracks• Stability in time (rate, %reco,...) | } All chambers with He mixtures data
from Standard reconstruction. (~ready) |
| <ul style="list-style-type: none">• Time resolution without TW correction? | |

Analysis steps



☑ Review of efficiency code(s) :

- CNAF official reco code (from binary to “debug tree”)
- Dedicated efficiency code by S.Boi.

2 main critical aspects :

- Efficiency, streamer fraction and cluster multiplicity are not obtained in a consistent way (different codes and/or different cuts). -> Difference in efficiency between CNAF eff. code and dedicated code (tuned for streamer % computation) ~10%
- Streamer are not $\ll 1\%$, but of the order of 10% (50/50 mixture)

☑ Debug/improve actual code. Main changes:

- New clusterization algorithm
- New calibration procedure (simultaneous time/space calibration)
- New selection cuts

At present the code can extrapolate streamer and efficiency simultaneously. Difference in efficiency between CNAF eff. code and dedicated code below 2%

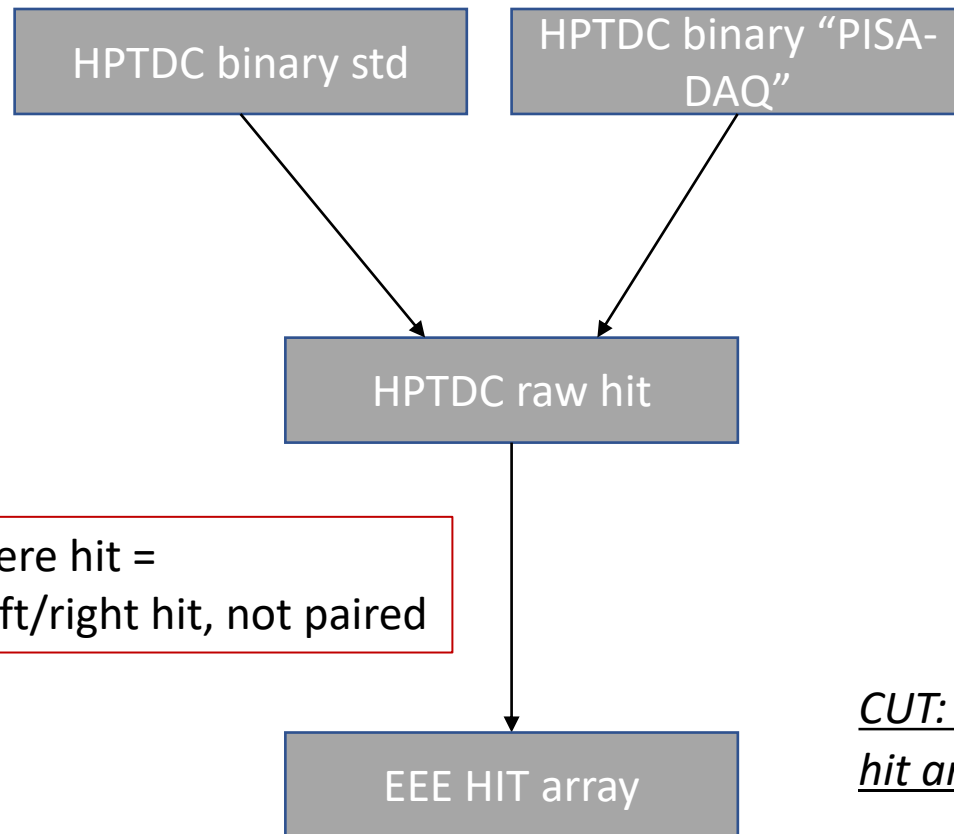
☑ Further optimization/automatization of the code, target discrepancy below 1% (Autom./optim. to be refined)

☑ Validation on a larger set of runs (at present I'm using a PISA run with 50/50 mixture @ eff. plateau, worst condition in terms of reconstruction).

☑ Recompute efficiency for the selected efficiency scans (2 telescopes, ~4 mixtures)

☑ Re-reco of PISA data after fix of DST producer -> New plots of parameter distributions (beta, Theta, ToF,...) -> C.Ripoli

Several codes used for the efficiency analysis, I decided to base the analysis review on the code developed by S.Boi. It takes as input the DST files generated by the CNAF reconstruction code, using a low level TTree.

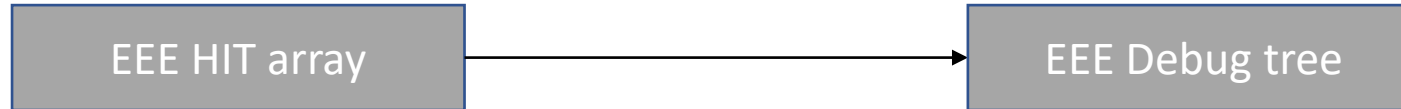


1. multi-hit on the same channel possible
2. Trailing edges without prior leading edge are discarded
3. leading edges without a trailing edge are registered with TOT=0

CUT: hit is discarded is the time of arrival is outside the limits taken by the configuration file «if (timeHit >= fCalib->GetTbLowLimitRight() && timeHit < fCalib->GetTbHighLimitRight())»

Here hit =
left/right hit, not paired

CUT: For each channel a maximum of 6 hit are passed to the hit array

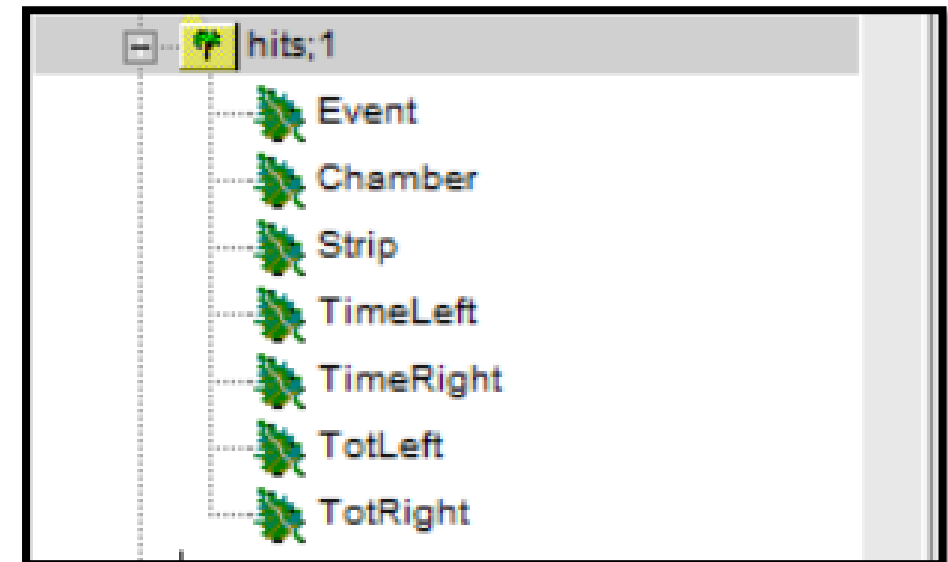


Note: If a strip has no hit on one side, the time on that channel is set to 0 and the TOT=-1

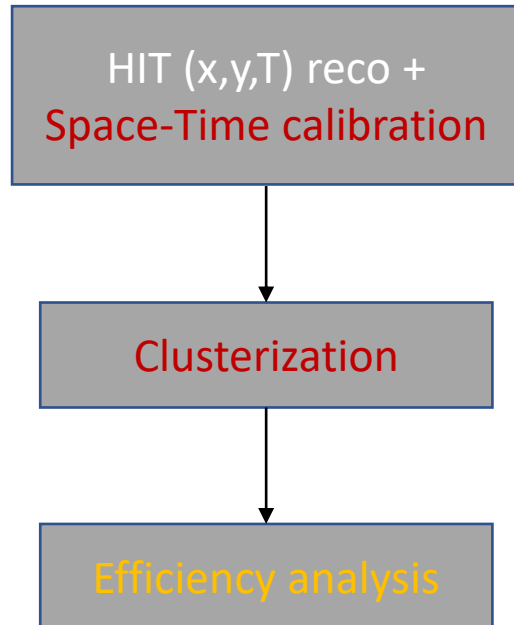
CUT: ONLY the first hit per channel is transmitted, other are discarded

- ❑ Data from “PISA” DAQ are reconstructed with hardcoded values:
 - of the geometry (in particular distance between chambers, wrongly set) -> **RE-RECO DONE**
 - of the architecture (NINO version, correctly set)

DST file content



Workflow based on the workflow of S.Boi.



Keeping the same code infrastructure (well done and with an event display!), the following sections were changed:

- New calibration procedure (simultaneous time/space calibration)
- New clusterization algorithm
- Upgraded selection cuts
- Minor fixes (not discussed here) and more control histograms

Details are given [here](#)

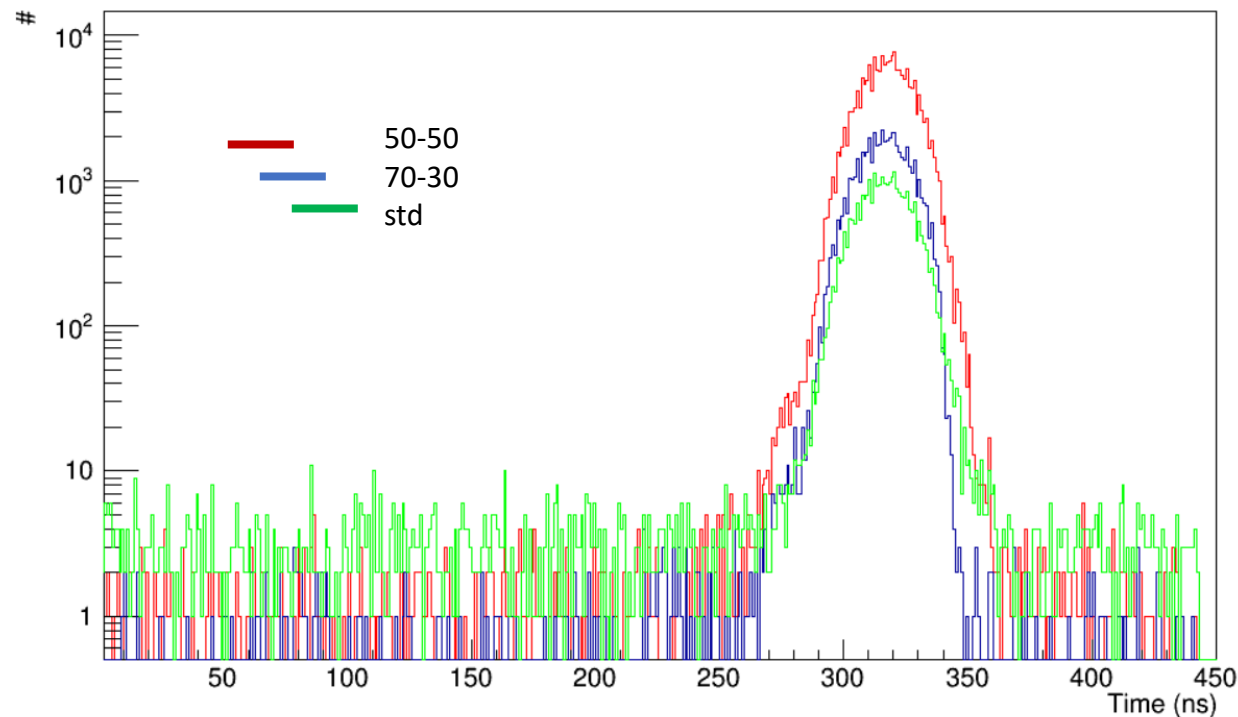
Analysis workflow – T Limit selection

For each mixture, a “pilot” run is selected, with a voltage of the chamber under test in the plateau region. Plateau is known from the previous analysis on the dataset.

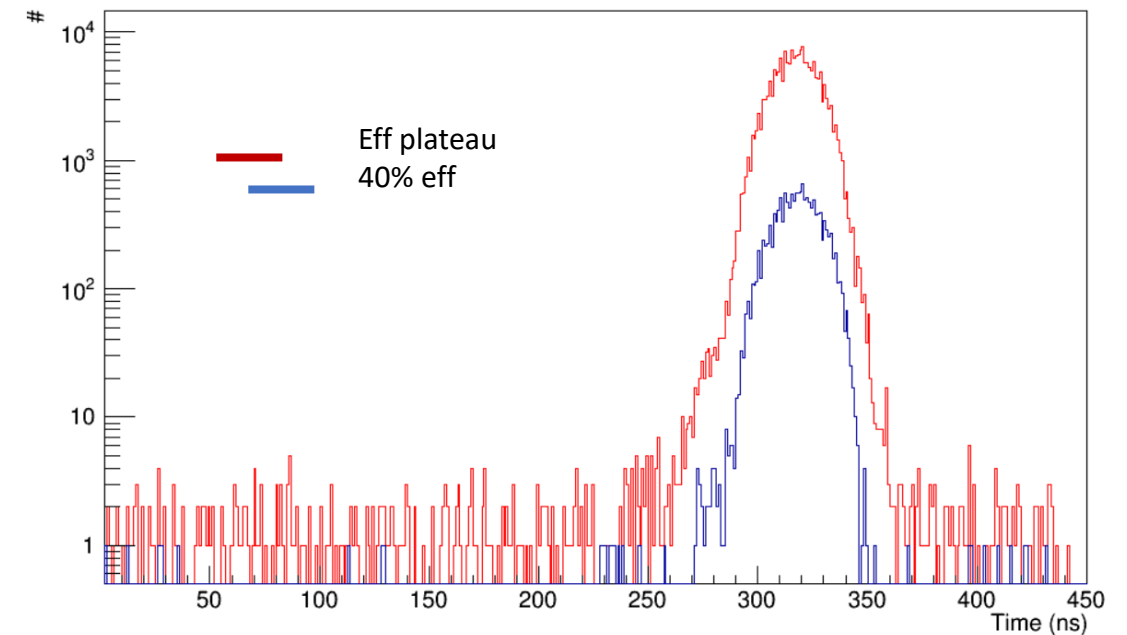
The pilot run is reconstructed without T-cuts at CNAF and the hits reconstructed with the analys workflow (1st stage). The raw T distributions are then used to optimize the T-cut.

All runs with the same mixture are then reconstructed with the same cut.

PISA-01 Bottom camber



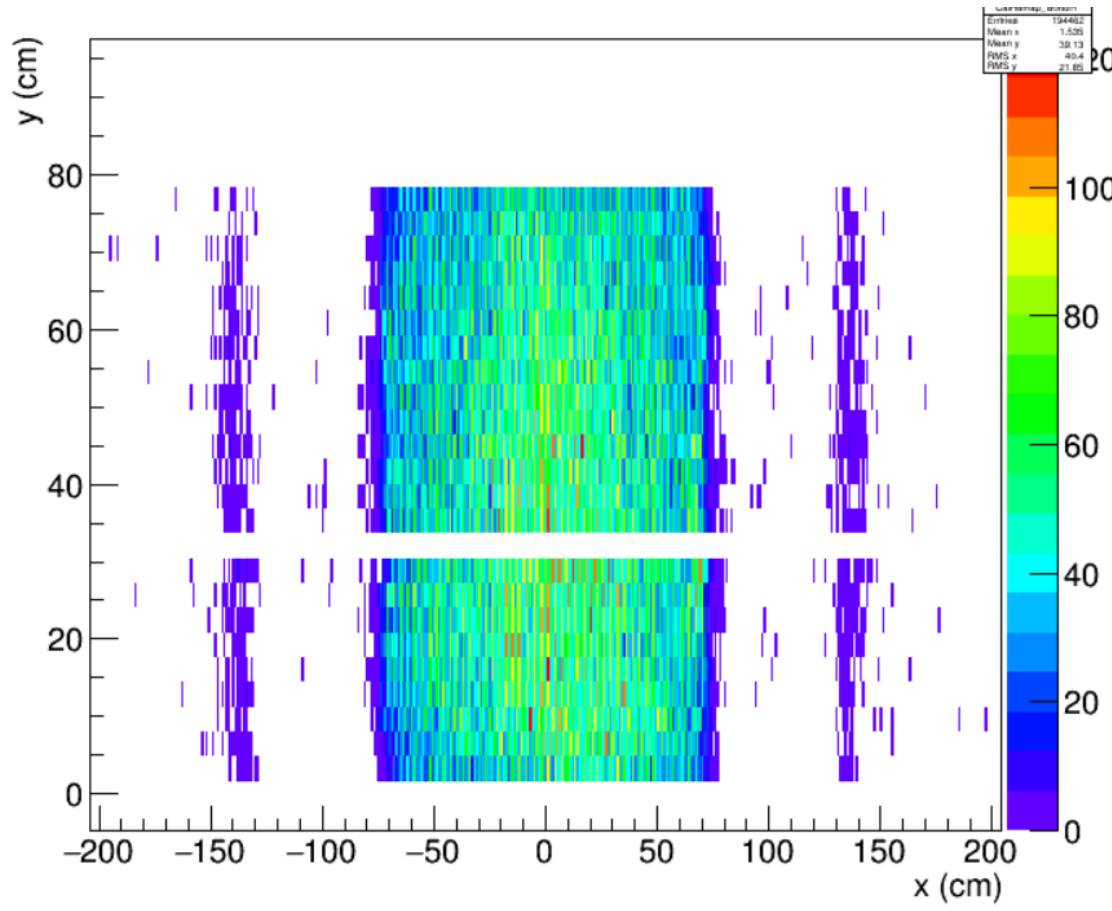
PISA-01 Bottom camber



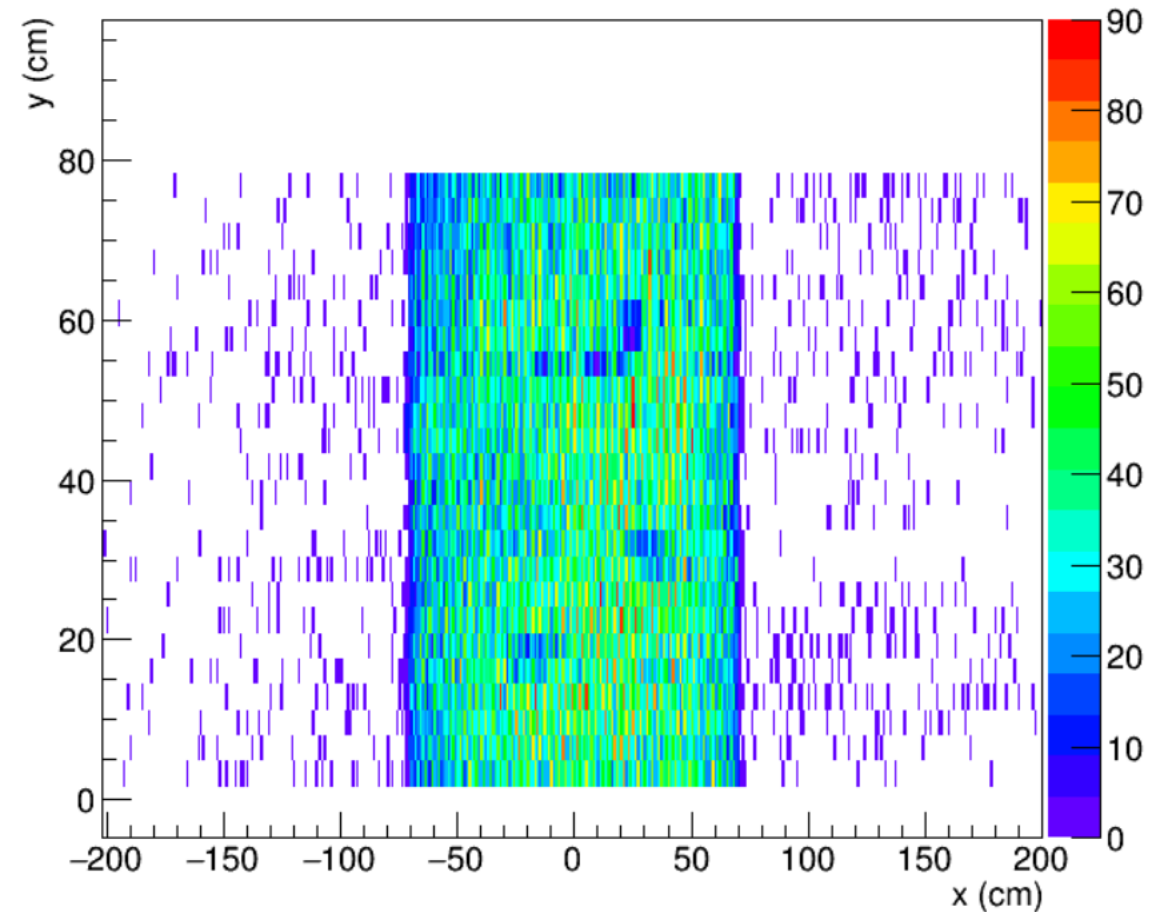
T-cuts are quite stable w.r.t. different mixtures and HVs

Hits outside chamber edges

PISA-01 Bottom camber, 50-50



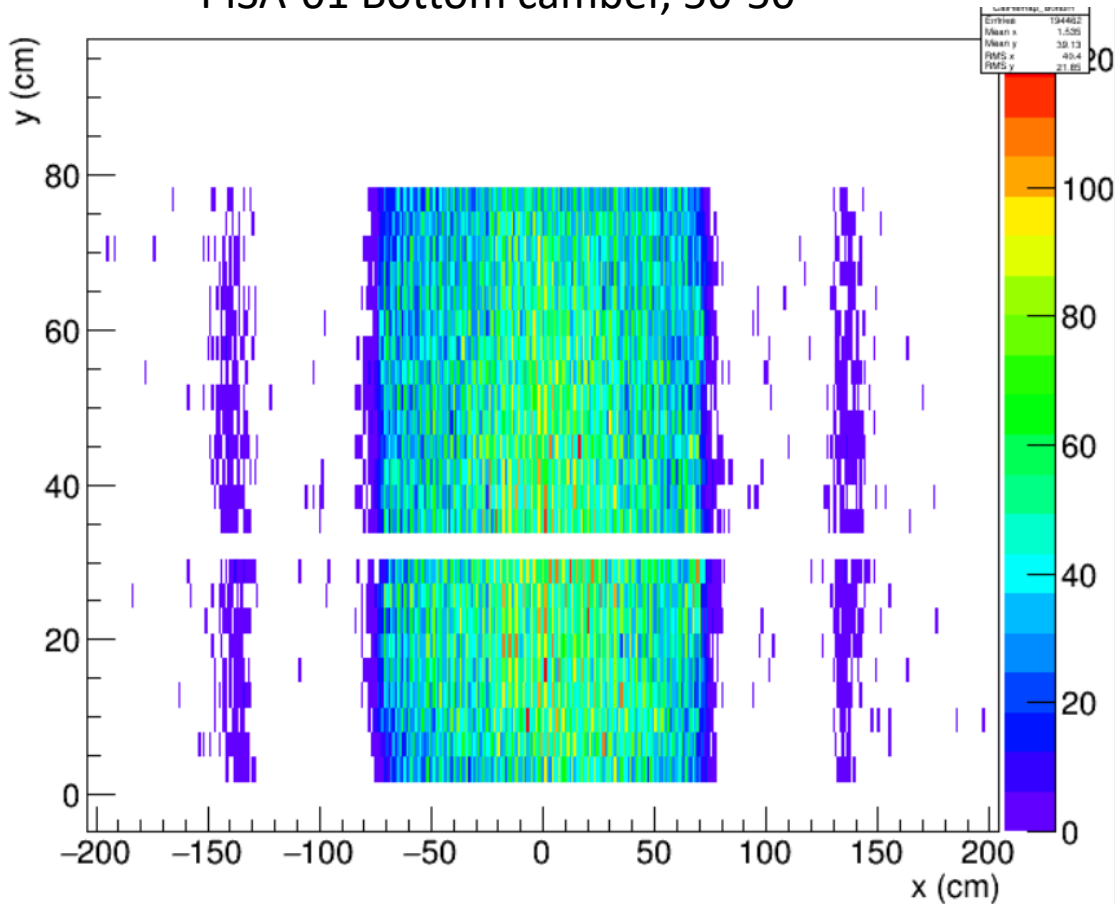
PISA-01 Middle camber, std



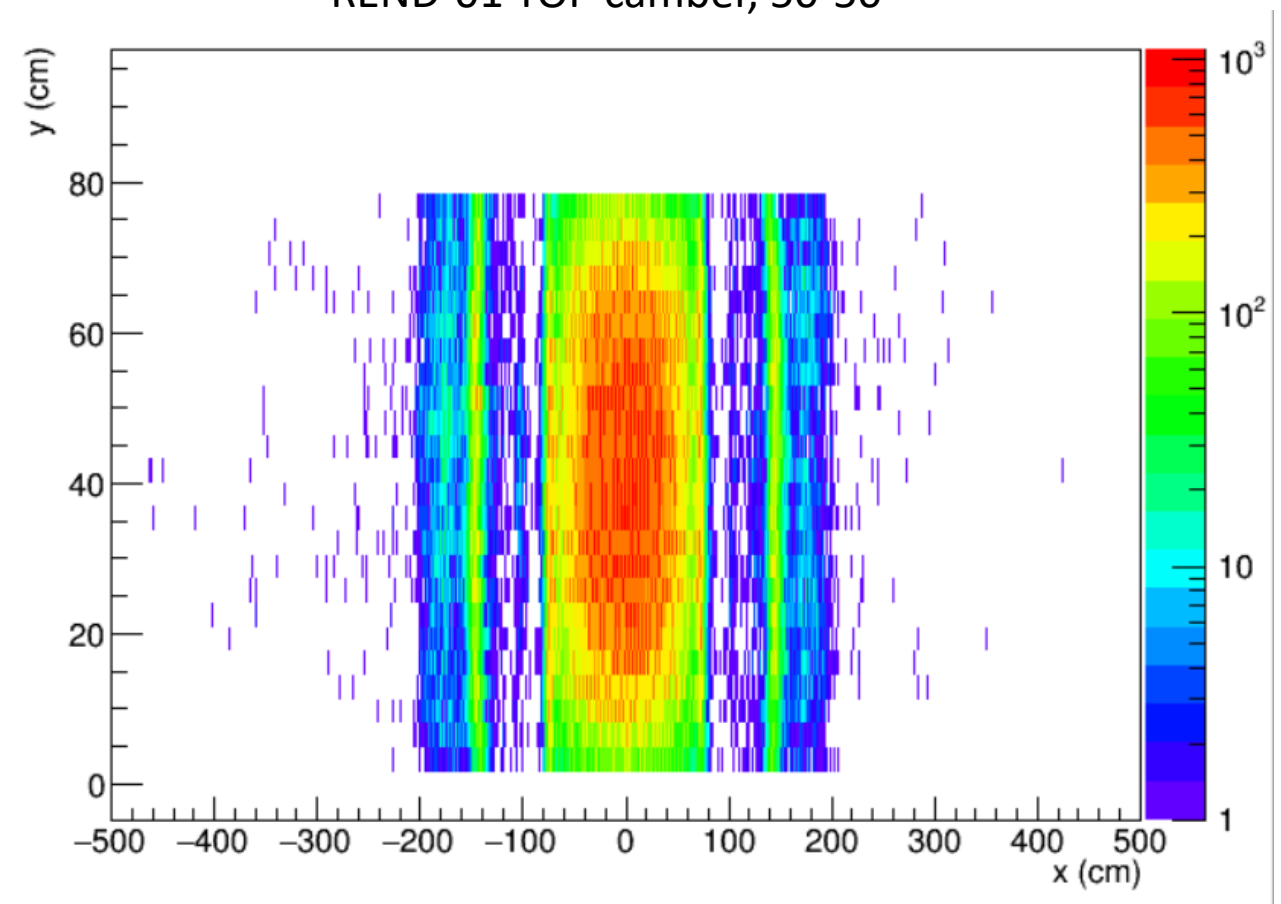
In test chamber, hits reconstructed outside active region of the chamber are accumulated in a well-defined area. Reason still not clear. The peaks are at ± 140 cm Reflections on patch FE cards??

Hits outside chamber edges

PISA-01 Bottom camber, 50-50



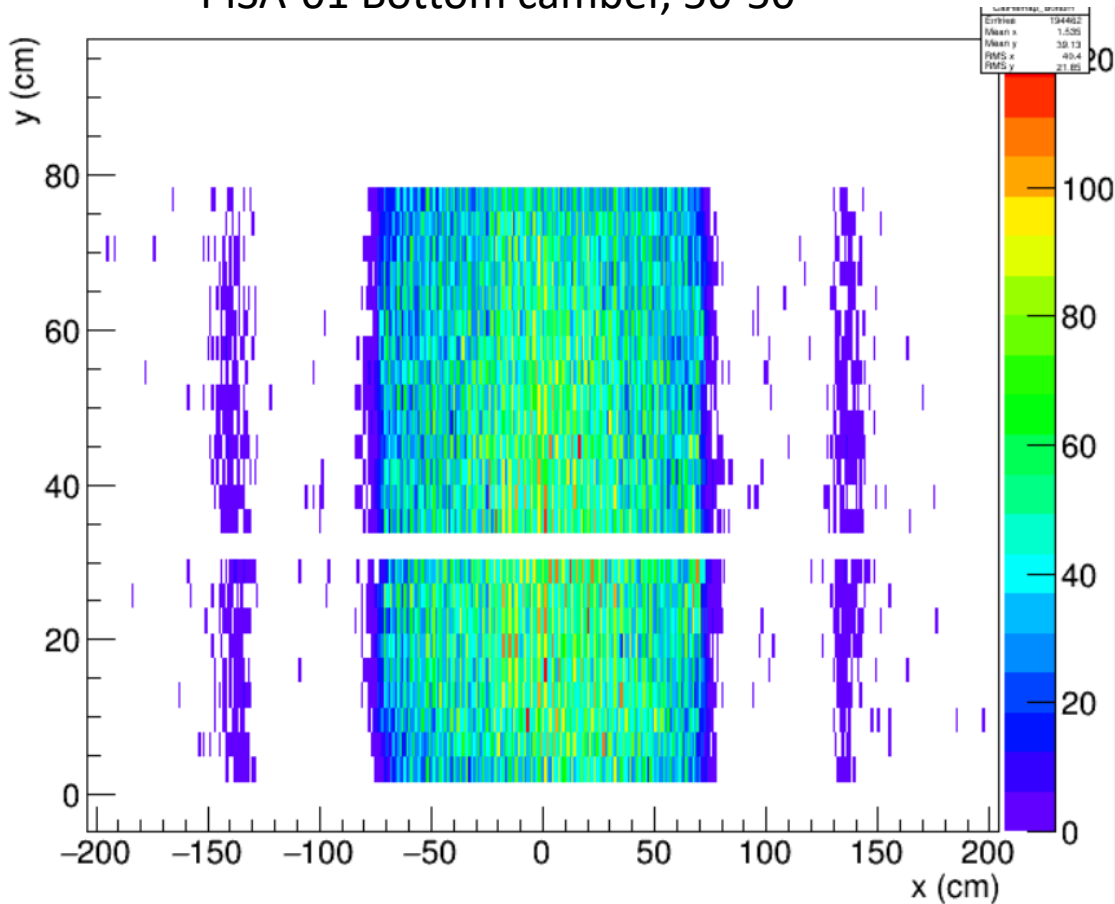
REND-01 TOP camber, 50-50



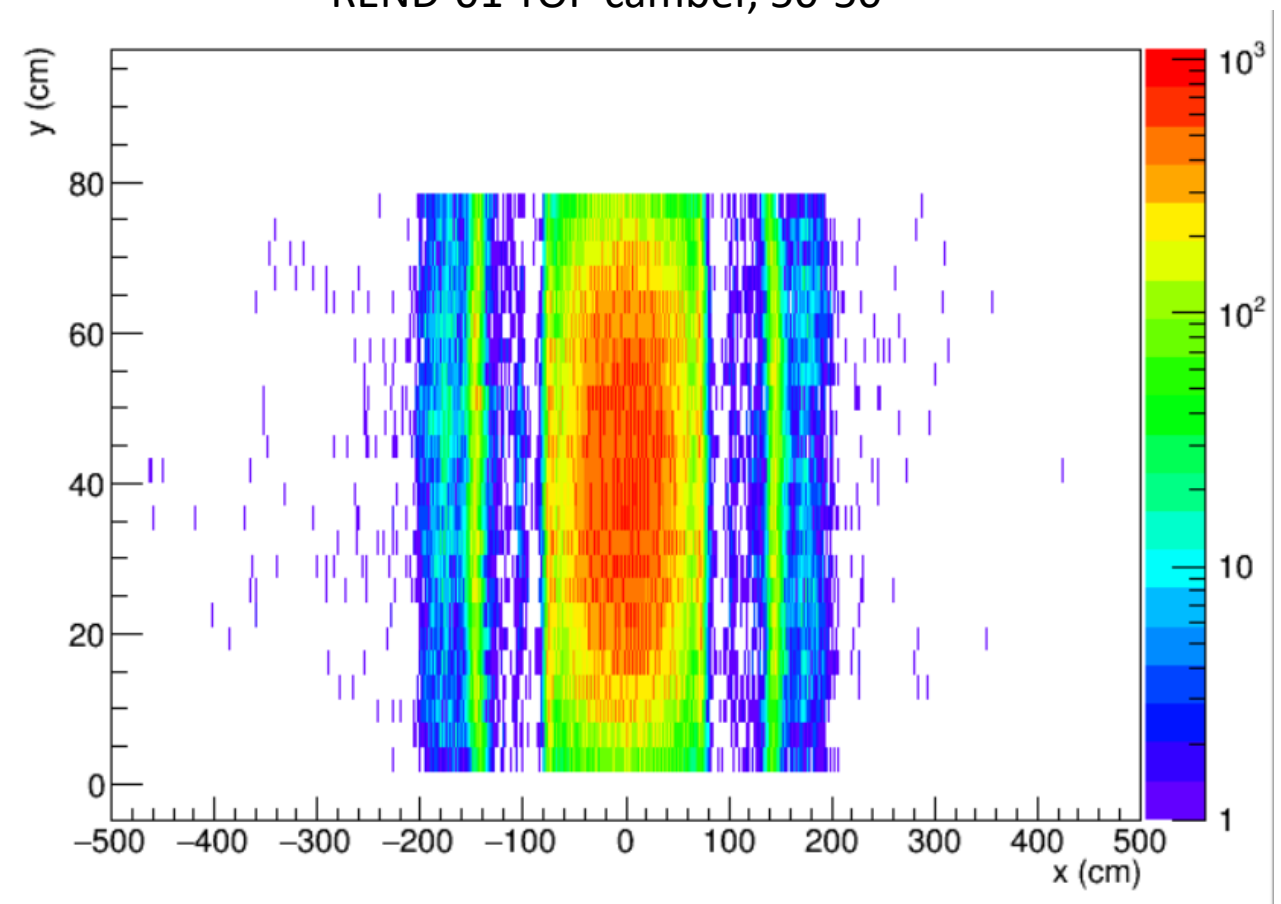
In test chamber, hits reconstructed outside active region of the chamber are accumulated in a well-defined area. Reason still not clear. The peaks are at +/-140cm Reflections on patch FE cards??

Hits outside chamber edges

PISA-01 Bottom camber, 50-50

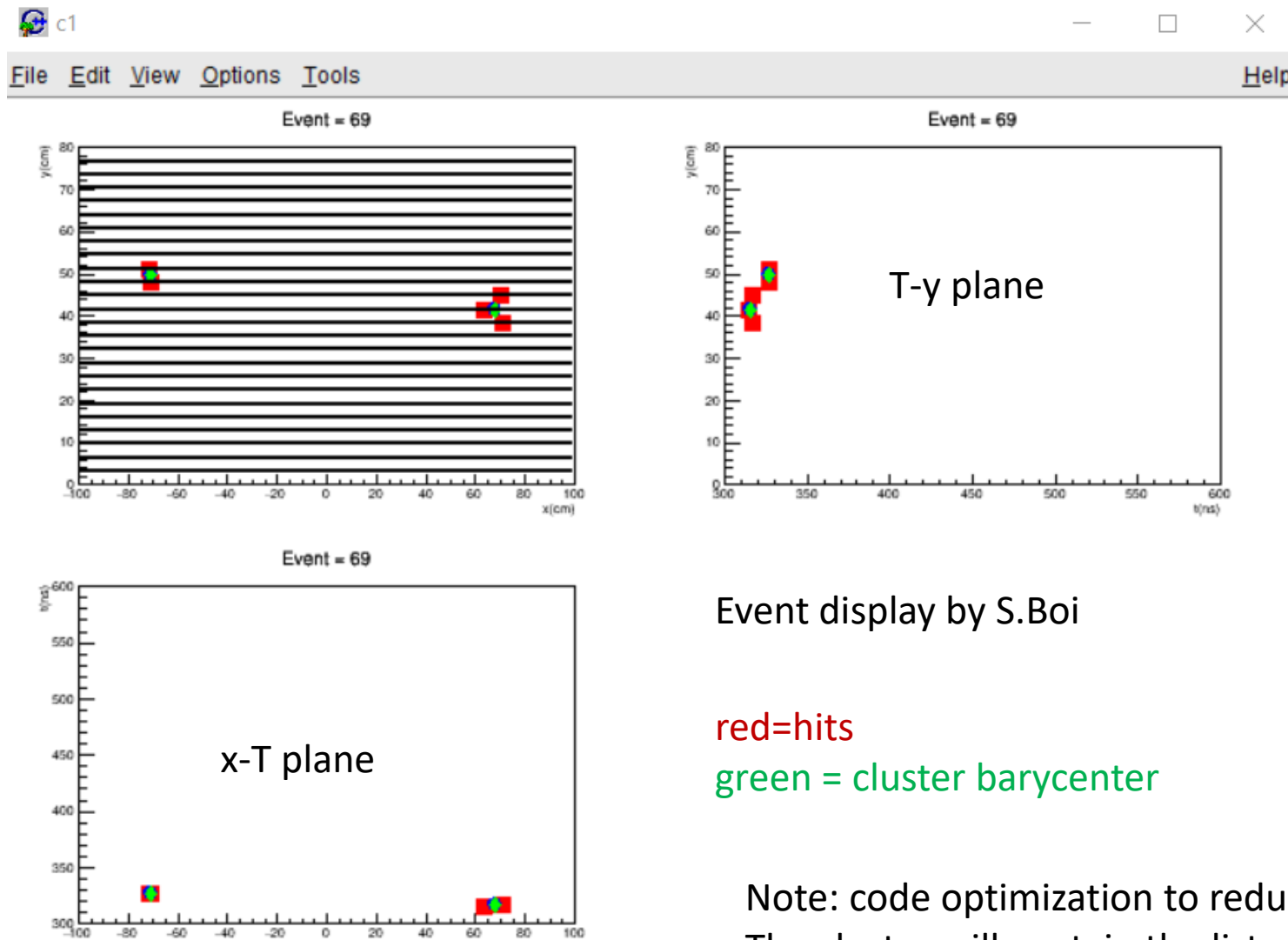


REND-01 TOP camber, 50-50



In test chamber, hits reconstructed outside active region of the chamber are accumulated in a well-defined area. Reason still not clear. The peaks are at +/-140cm Reflections on patch FE cards??

Clusters



Event display by S.Boi

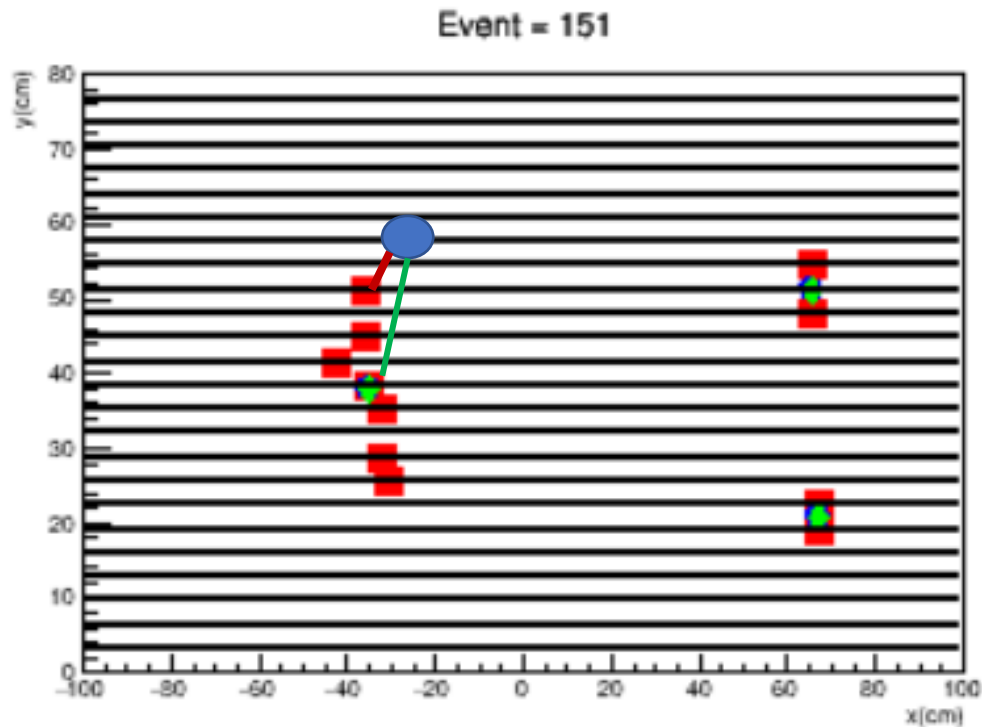
red=hits
green = cluster barycenter

Note: code optimization to reduce clusterization step to few seconds (100K events)
The cluster will contain the list of hits -> useful in the last stage of the analysis

The algorithm in short:

1. First hit is promoted to cluster and removed from hit array.
2. Scan over the hit array to find the first hit with XY distance below 10cm. Metrics: minimum distance between the hit and all the hits already part of the cluster
3. If some hit is added to cluster, remove it from the hit array and go back to point 2.
4. When no more hits can be added to the present cluster, compute cluster parameter (baricenter, T, average ToT)
5. if the hit array is not empty, create a new cluster with first hit and go back to point 2.

Event selection as in the past but new metrics to check the “distance” between a cluster and the projected hit



Old metrics: Distance to barycenter

New metrics: distance to closest hit of the cluster

Cluster multiplicity and streamer are computed from the multiplicity of best-matching cluster.

Selection cut on the triggered events (some tuning still needed):

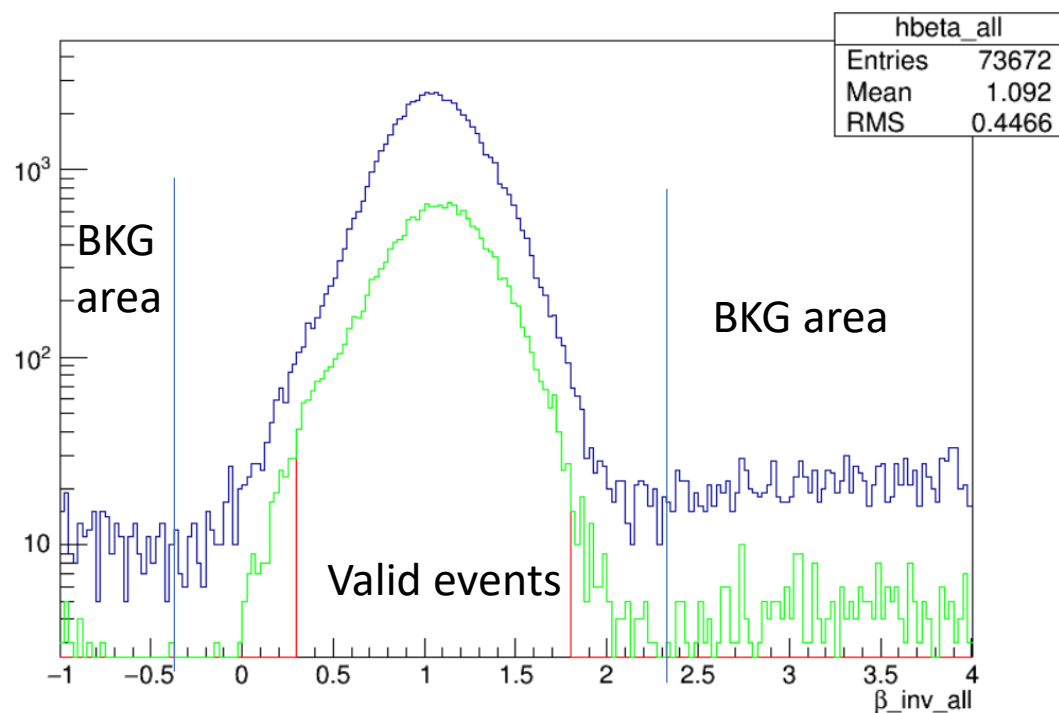
- Extrapolated hit within test chamber acceptance ($5 < y < 75$ cm, $-60 < x < 60$ cm)
- $z_{dir} > 0.9$ (**DISABLED**)
- particle inverse beta within correct window (see next slide)

Criteria for efficiency:

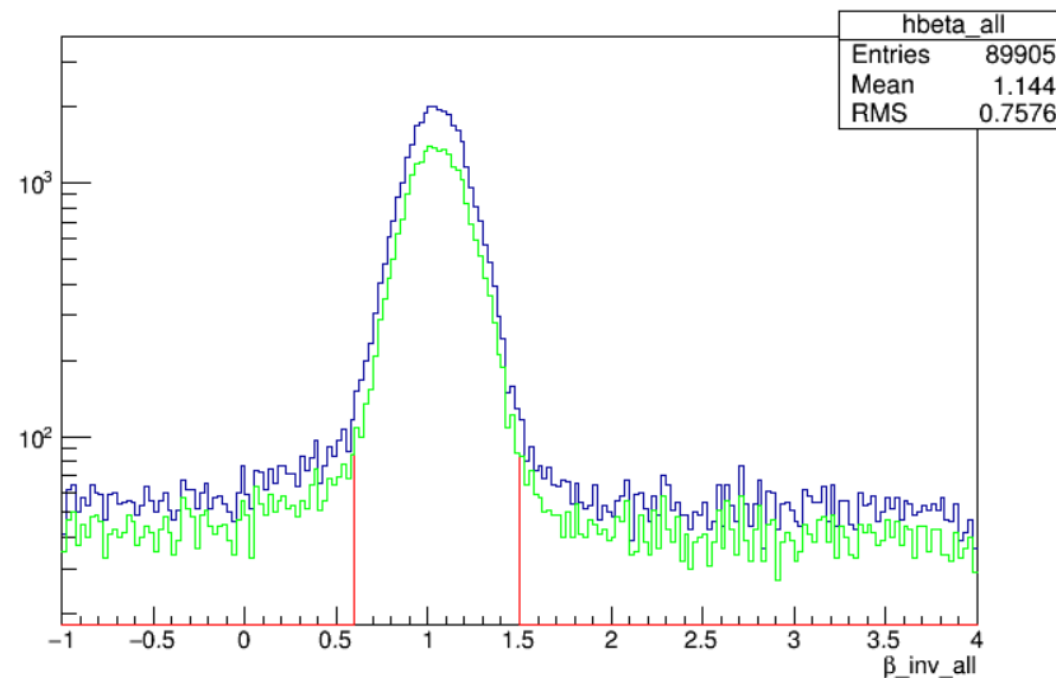
- distance between the extrapolated hit and the closest cluster below 15 cm
- Time difference between the extrapolated hit and the closest cluster below 10 ns

Efficiency

PISA-01



REND-01



- All events with only one hit in triggering chambers
- After fiducial cut
- INV beta cut

- Wider distribution for PISA-01: top and middle chamber known to be not very good
- Non negligible background for REND-01

Corrections

3 typologies of corrections considered:

- selected event was generated by noise in triggering chambers, considered inefficient -> false negative
- selected event was generated by noise in triggering chambers, matched in test chamber-> triple noise coincidence (considered negligible)
- selected event is a real particle, matched in test chamber by a noise hit -> false positive

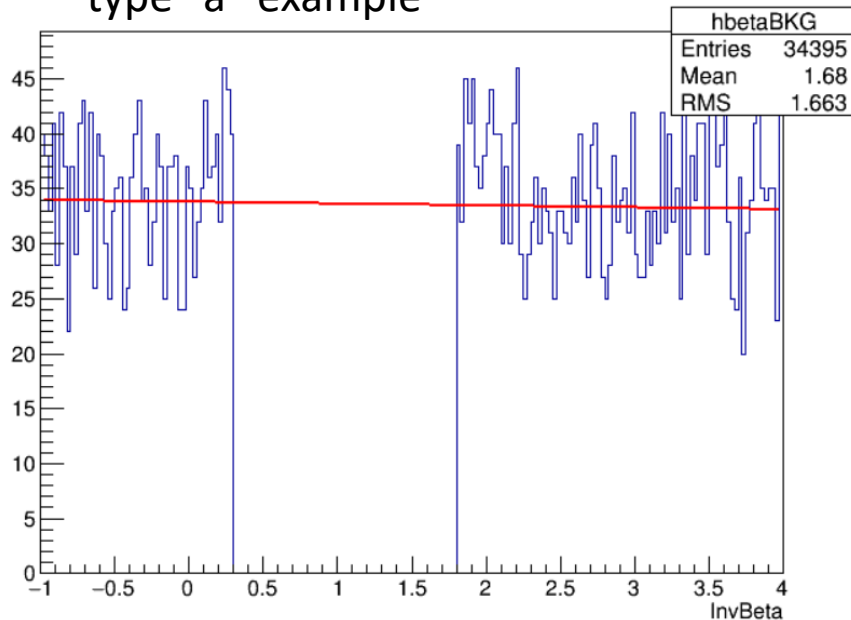
Correction can be performed by estimating the number of event for each category and modifying the numerator/denominator for the efficiency

a -> reduction of the denominator

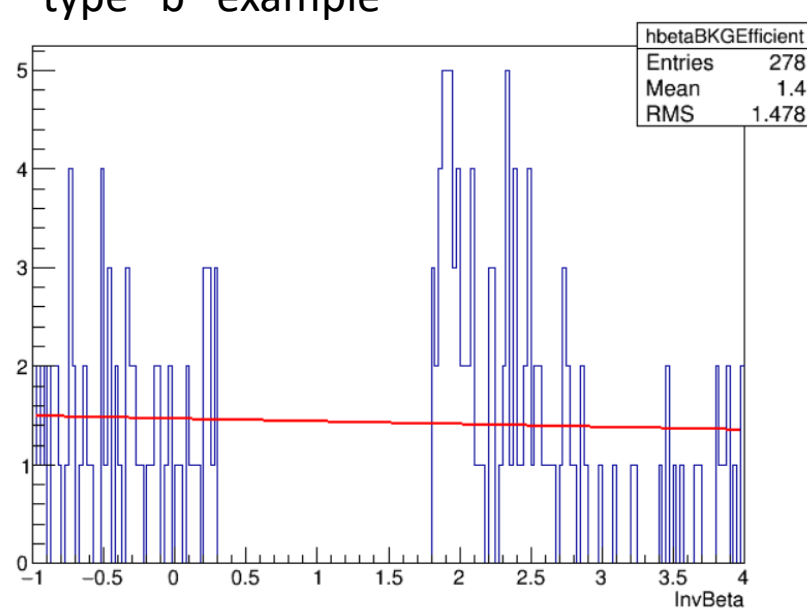
b -> reduction of both numerator and denominator

c ->reduction of numerator

type "a" example



type "b" example



The corrections are computed by fitting the invbeta distribution in the BKG region and integrating the fitting function in the signal region.

- For REND-01 a positive relative correction of ~6% is found
- Type b correction are usually negligible

Datasets

PISA:

50/50 : 22-23 Dic 2021
60/40: 7-8 Dic 2021
70/30: 6-7 Nov 2021
std: 13 Ott 2021

RENDE:

50/50 : 14 Oct 2021
60/40: 15 Oct 2021
70/30: 16 Oct 2021
std: 32 Dic 2021

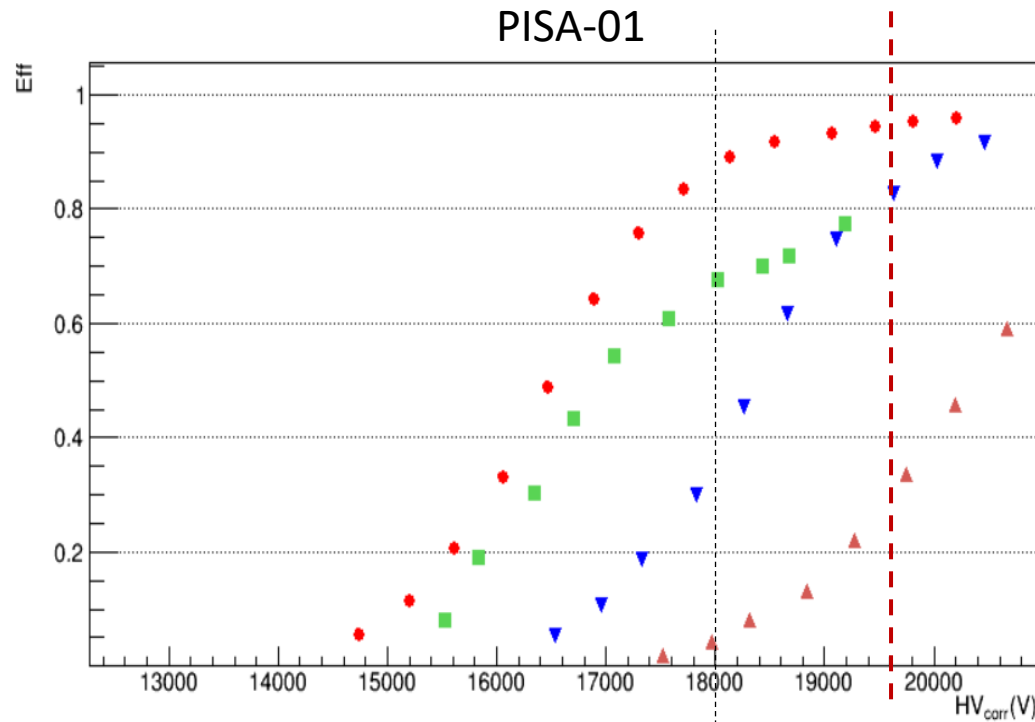


Same dataset used for plots shown at previous conferences

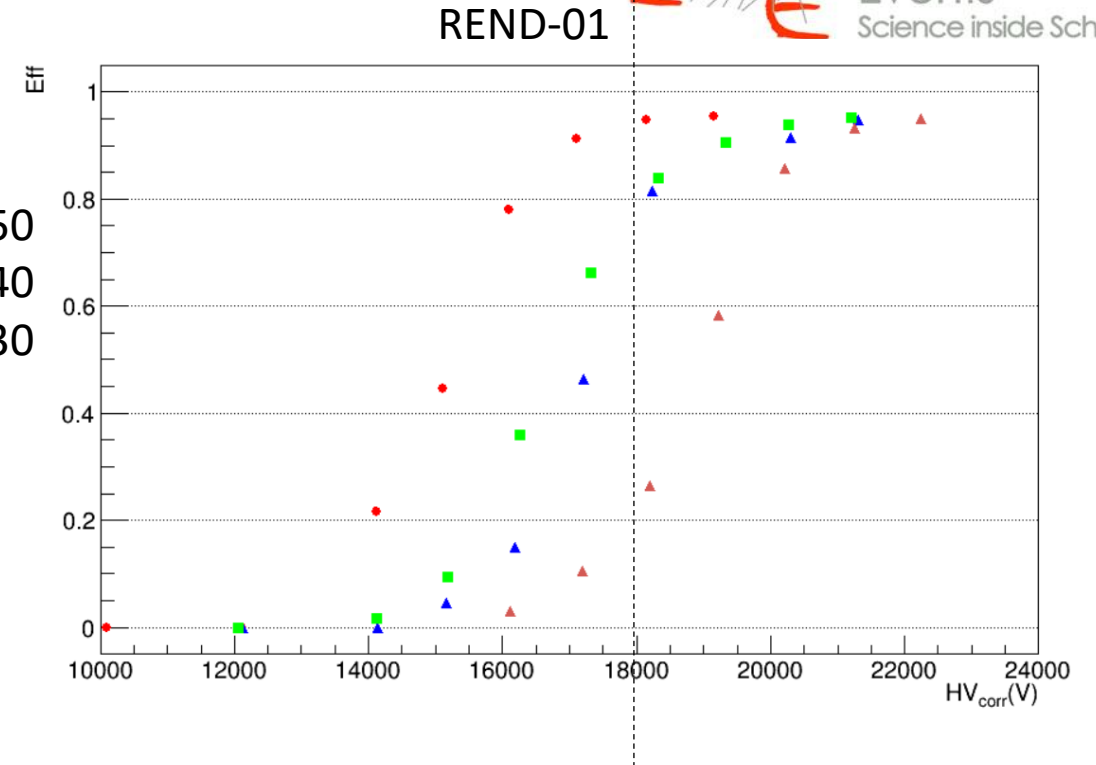
- Std, 50-50, 60-40 and 70-30 mixtures studied on each telescope
- Triggering chambers with fixed HV and std mixture
- Bottom chamber (PISA) and middle chamber (REND-01) used as test chamber
- Chamber gaps : 300um
- Thr ~600mV

Data at different Thrs are available for both stations, but not used in the present analysis

Efficiency

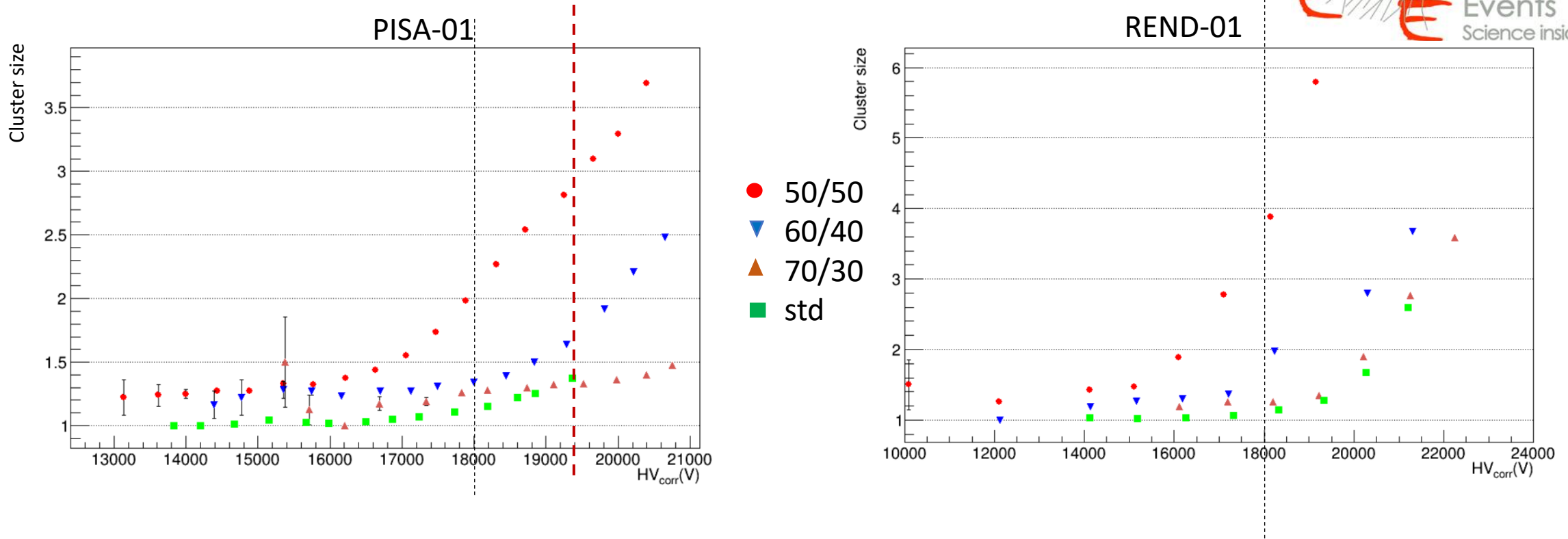


● 50/50
▼ 60/40
▲ 70/30
■ std



- Weird behavior of PISA-01 “std” mixture
- Curves looks shifted by $\sim 1.5\text{kV}$ \rightarrow potentially due to HV readout offset, chamber differences
- Std” curve lies between 50/50 and 60/40 for both stations

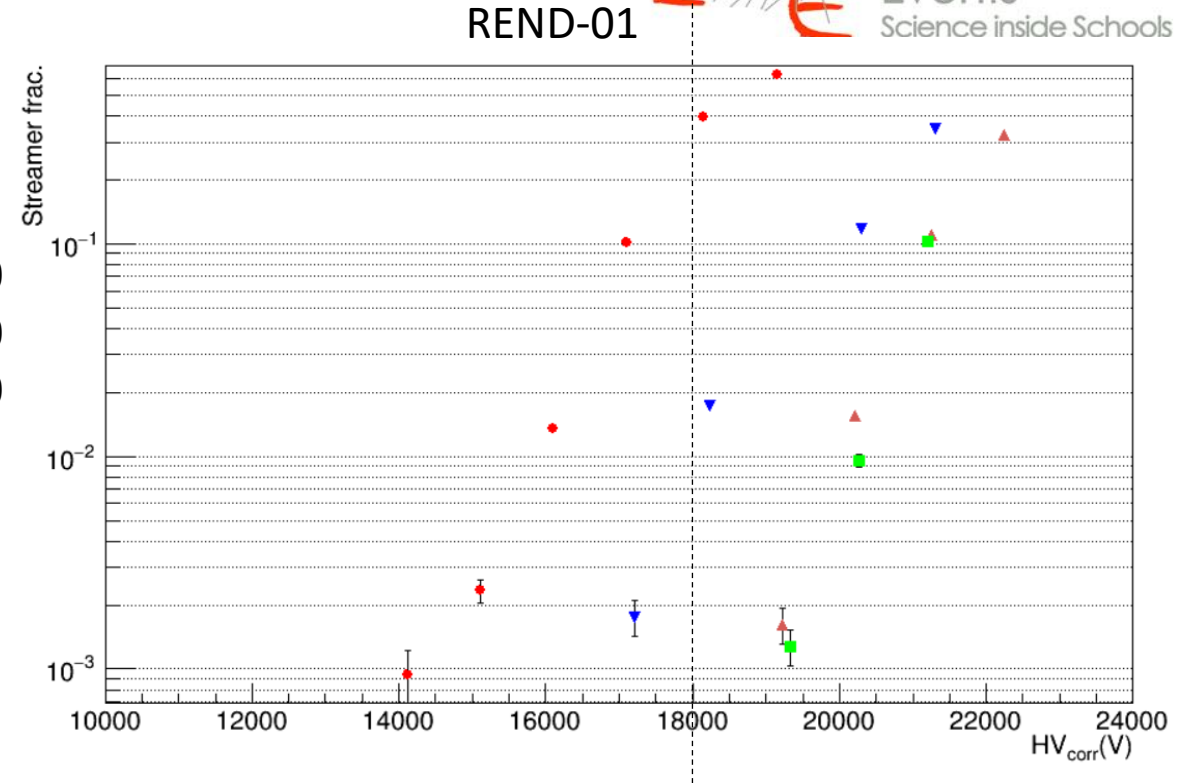
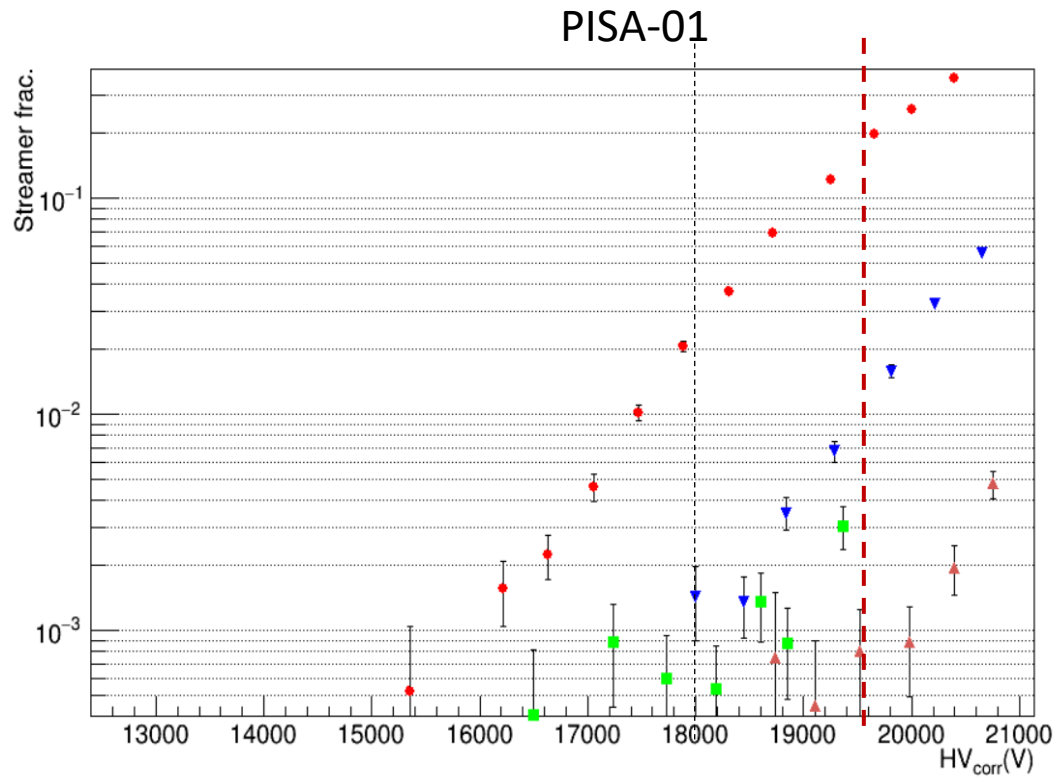
Cluster size



Def: Cluster size is the number of hits forming the cluster closer to the extrapolated point in the test chamber.
The uncertainty is the RMS of the cluster size distribution divided the square root of the number of entries in the histogram

Same behavior between the two station, apart for the HV “offset”

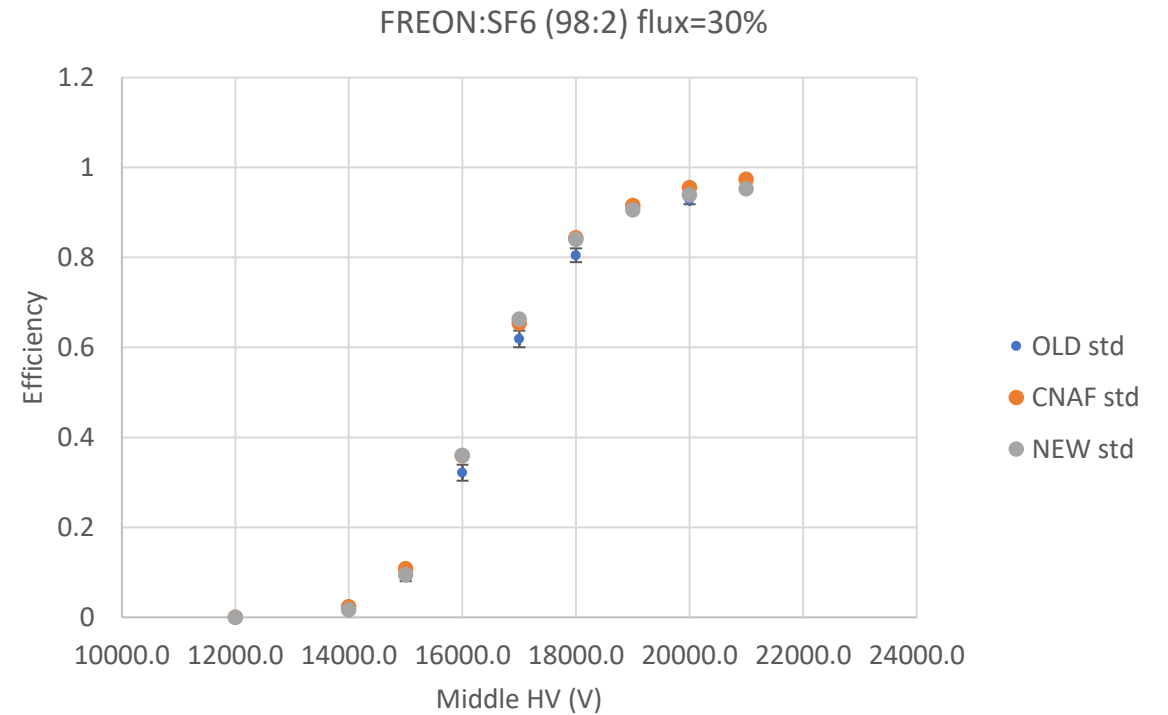
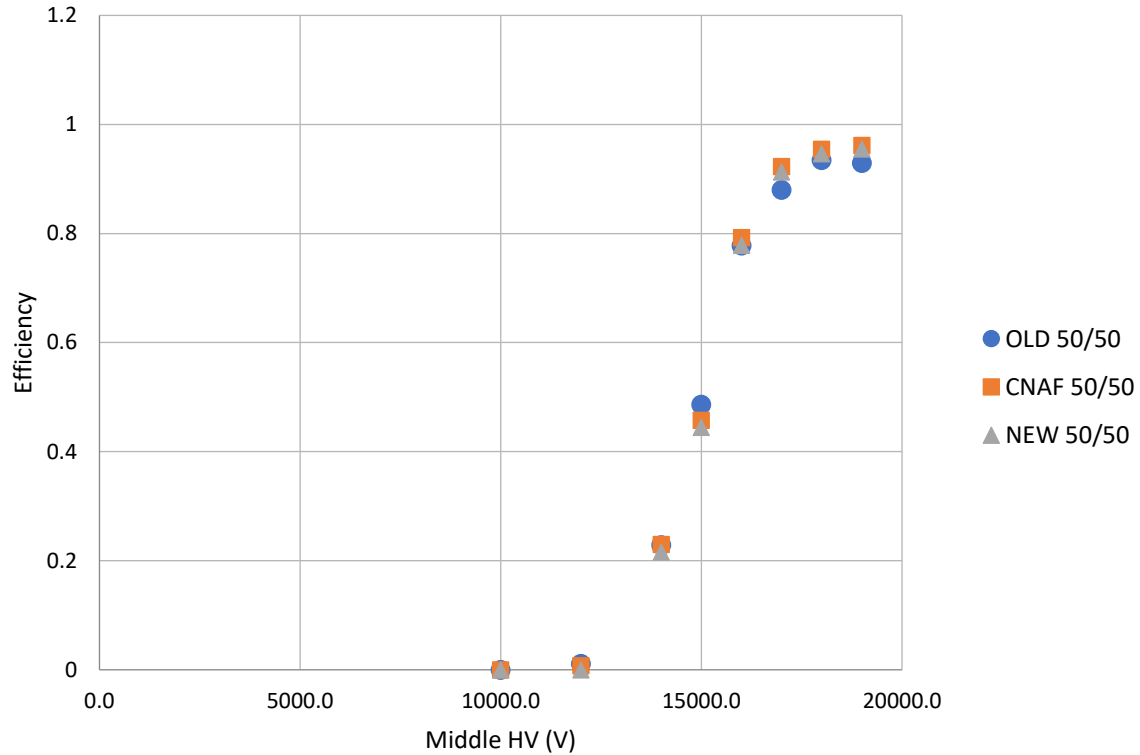
Streamer fraction



Def: Streamer fraction is the fraction of efficient event were the cluster size in the test chamber was > 3
Same behavior between the two station, apart for the HV “offset”

- Streamer fraction reach high values for both 60/40 and 50/50 mixtures (i.e., it is already $\sim 1\%$ @ 80% efficiency for 60/40 mixture)

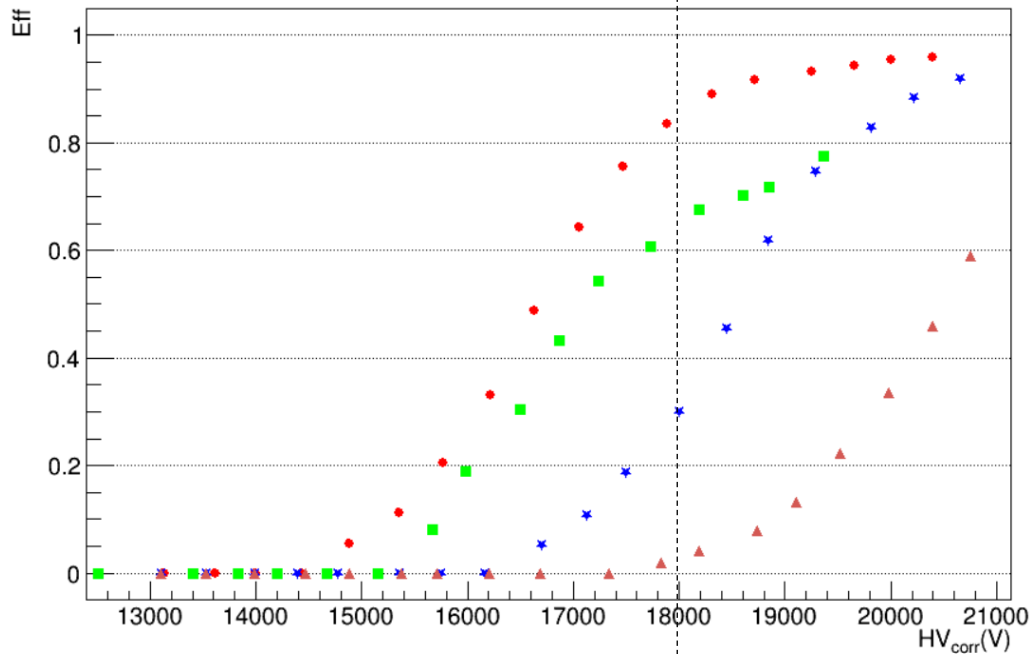
REND-01: CNAF comparison



Comparison of analysis results of REND-01 data are consistent between CNAF, present analysis and old studies

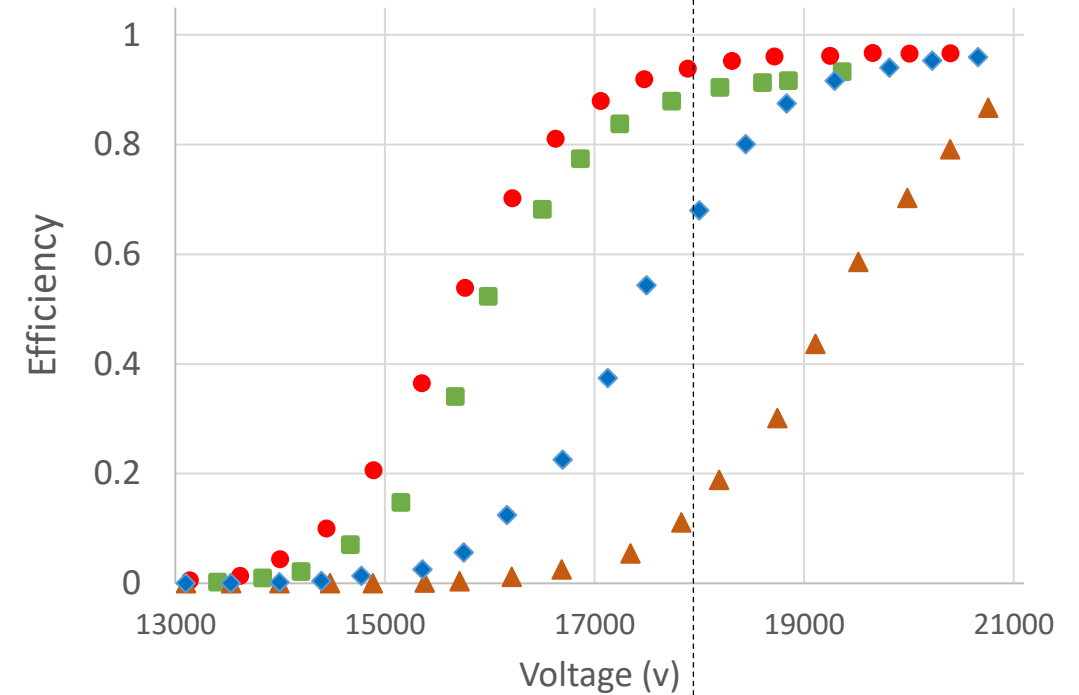
PISA-01: CNAF comparison

Latest results



- 50/50
- ▼ 60/40
- ▲ 70/30
- std

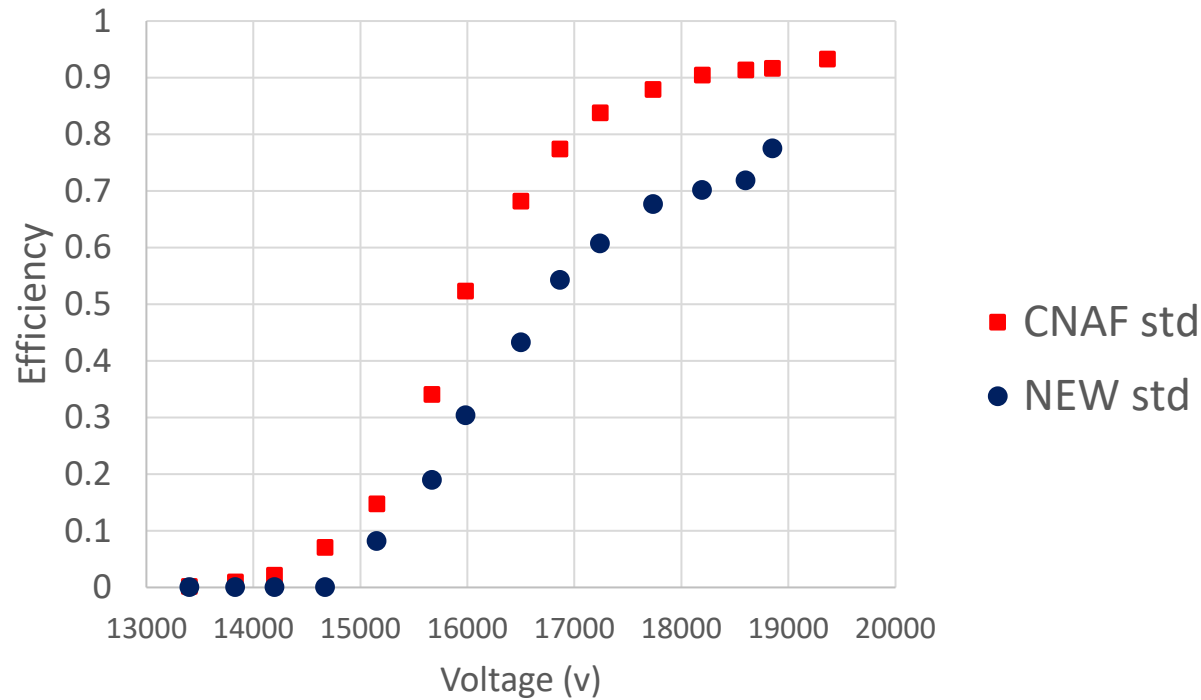
CNAF



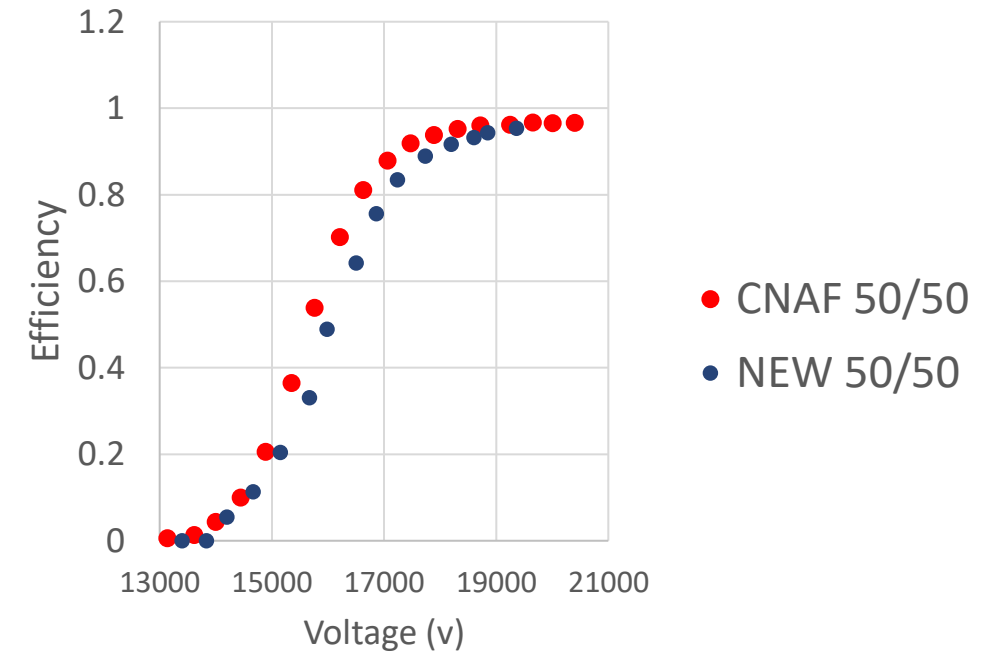
Comparison of analysis results of PISA-01 data shows inconsistent results between CNAF and present analysis

PISA-01: CNAF comparison

Efficiency Vs normalized HV



Efficiency Vs normalized HV



Comparison of analysis results of PISA-01 data shows inconsistent results between CNAF and present analysis

This aspect requires further investigation, which is ongoing

Conclusion



- A Review of efficiency code(s) have been performed
- Critical aspects of the workflow have been identified, corrected and code is optimized for fast processing
- A complete dataset has been analyzed for both stations with different mixtures/compositions

- Few aspects need further investigation

backup

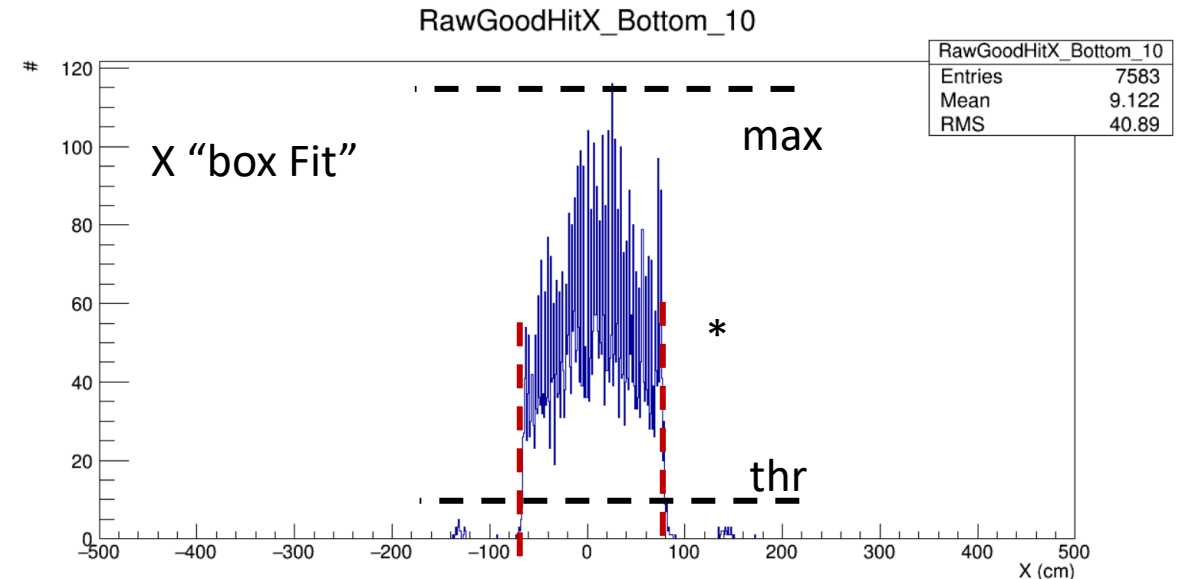
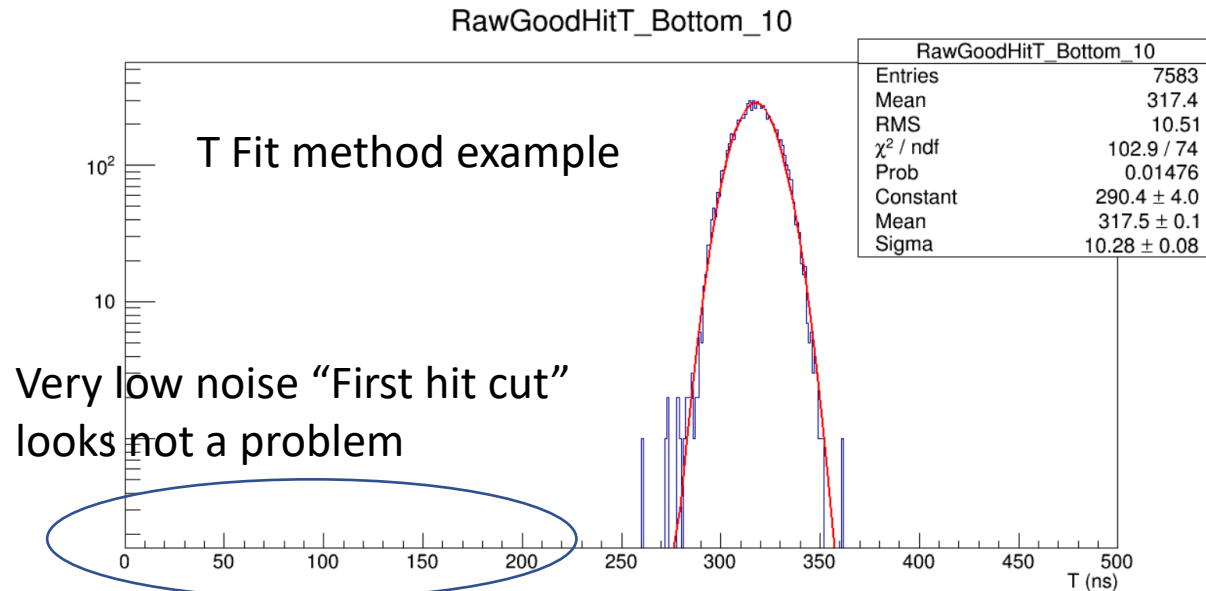
Calibration

The calibration performs a simultaneous calibration of space and time.
It also calibrates the average middle time to be centered w.r.t. the outer chambers.

1. For each strip compute
 - average x coordinate
 - average hit T ($T_l + T_r / 2$) distribution
2. For each chamber:
 - Average hit T distribution

T distribution have large shape variation between telescopes and clk distribution system. The average T can be computed with two parametrized modes:

- Gaussian Fit
- “Box fit” (as for the x coordinate)



Simple average affected by large calib. error for non uniform hit distributions

3. Space/ correction are independently applied

```
if(!external_calib)
{
    std::cout<<"Computing corrections..."<< std::endl;
    double delta_time,strip_offset, Avg_T_adjust[3],x_offset;

    Avg_T_adjust[0]=.0;
    Avg_T_adjust[1]=ch_time[1] - ((ch_time[2]+ch_time[0])/2);
    Avg_T_adjust[2] = .0;

    std::cout << "Mean hit time adjust for chamber 0: " << Avg_T_adjust[0] << std::endl;
    std::cout << "Mean hit time adjust for chamber 1: " << Avg_T_adjust[1] << std::endl;
    std::cout << "Mean hit time adjust for chamber 2: " << Avg_T_adjust[2] << std::endl;

    for (int chamb_idx = 0; chamb_idx < 3; chamb_idx++) //chambers loop
    {
        //adjust strip offset
        for (int str_idx=0;str_idx<24;str_idx++) // secondo loop sulle strip
        {
            //offset tra la media della strip e la media della camera
            if (MeanT_map[make_pair(chamb_idx,str_idx)]>0)
                delta_time=((ch_time[chamb_idx] - Avg_T_adjust[chamb_idx]) -MeanT_map[make_pair(chamb_idx,str_idx)]); //ns
            else delta_time=0.0;
            Calib_map[make_pair(chamb_idx,str_idx)].first=delta_time; //hit time = (Tleft+Tright)/2
            Calib_map[make_pair(chamb_idx,str_idx)].second=delta_time;

            //std::cout<<"strip "<<strIdx<<" time correction: "<< delta_time<<std::endl;
            //correzione spaziale

            x_offset = MeanX_map[make_pair(chamb_idx,str_idx)];
            std::cout<<"strip "<<str_idx<<" x cm offset: "<< x_offset<<std::endl;
            strip_offset=x_offset*2.0/EEEHit::SpeedOfPropagation; //mean ns strip offset
            //std::cout<<"strip "<<strIdx<<" y ps offset: "<< strip ps offset<<std::endl;

            Calib_map[make_pair(chamb_idx,str_idx)].first+=strip_offset/2.0; //negative side
            Calib_map[make_pair(chamb_idx,str_idx)].second-=strip_offset/2.0; //positive side
            std::cout << "Chamber " << Chamber_name[chamb_idx] << ", strip " << str_idx << " left:" << Calib_map[make_pair(chamb_idx,str_idx)].first << s
            std::cout << "Chamber " << Chamber_name[chamb_idx] << ", strip " << str_idx << " right:" << Calib_map[make_pair(chamb_idx,str_idx)].second <<
        }
    }
}
```

Average chamber correction

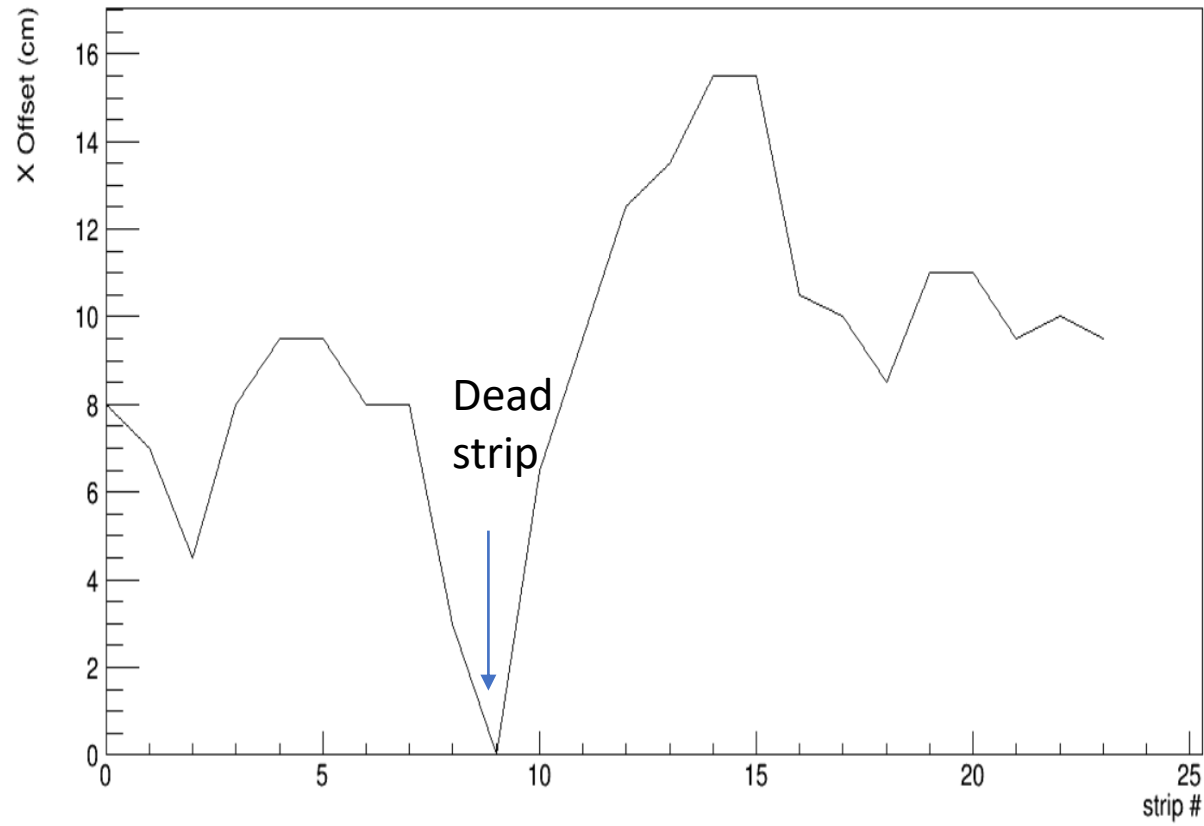
Time calibration

Space calibration

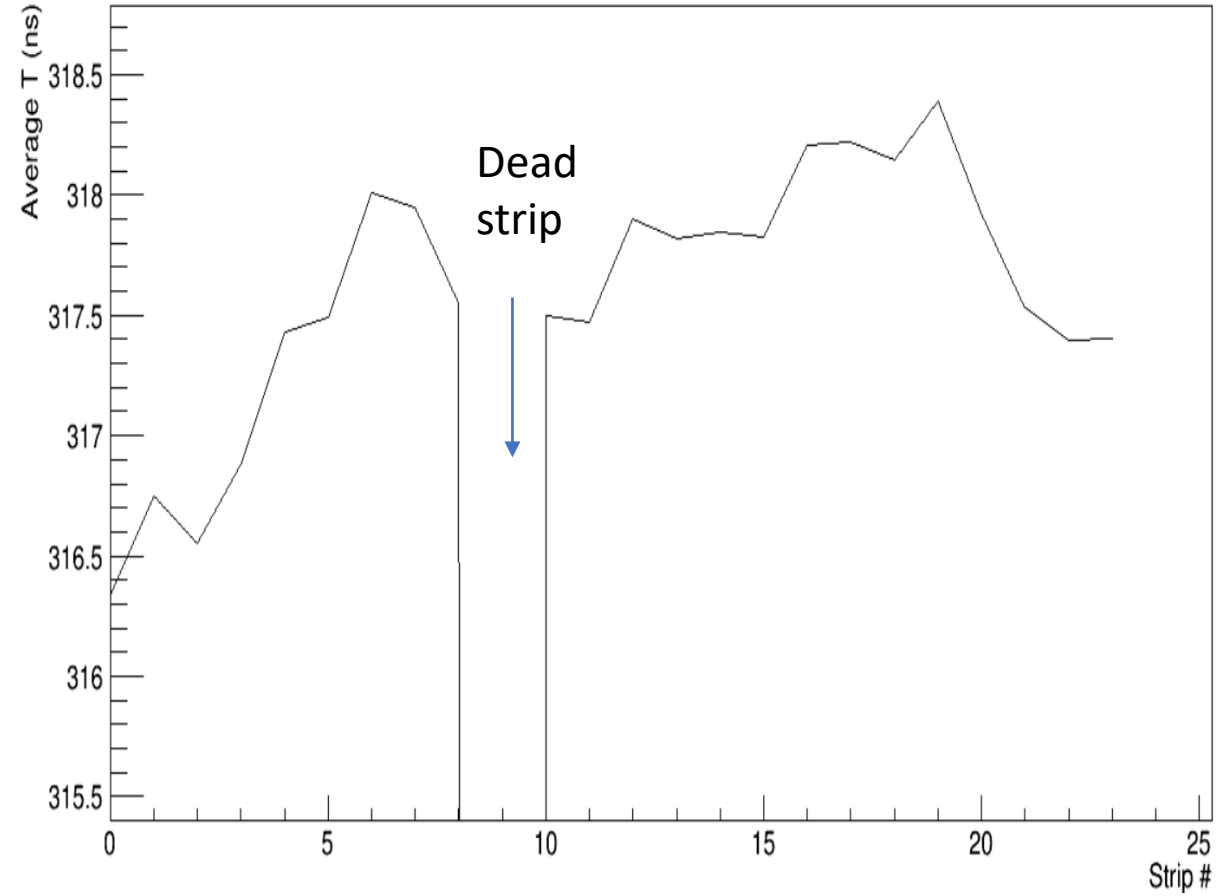
Calibration can be saved/retrieved

Raw data offset

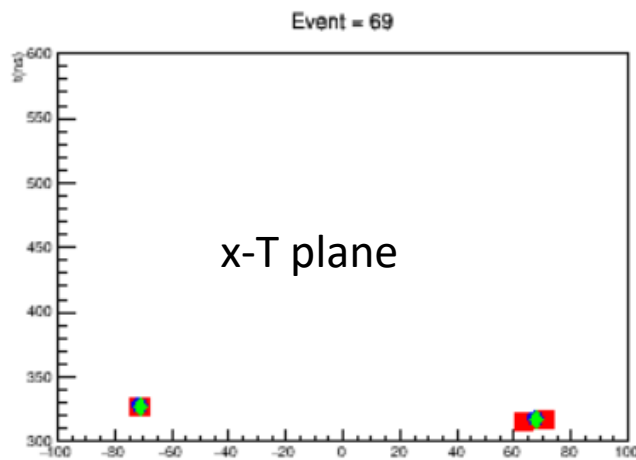
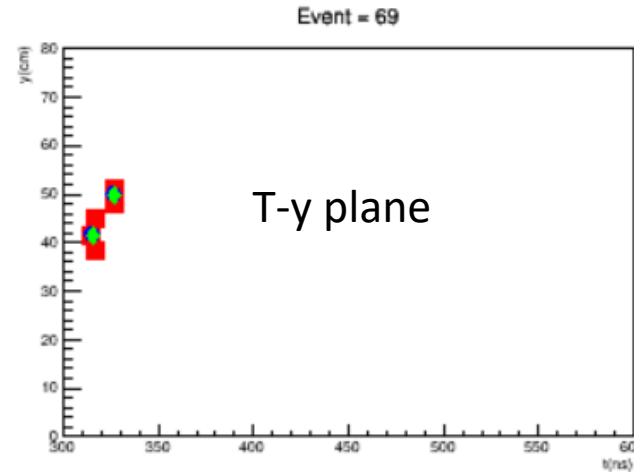
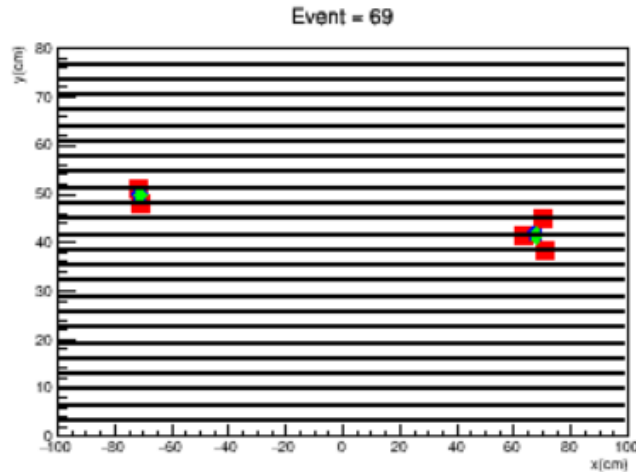
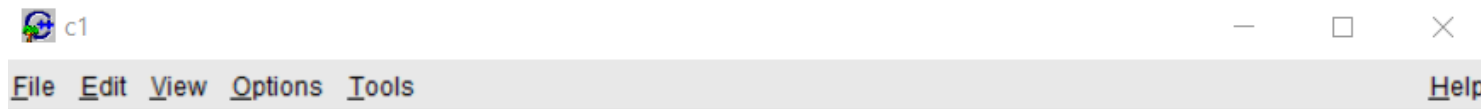
Mean X Trend Chamber_0



Mean Time Trend Chamber_0



Clusters



Event display by S.Boi

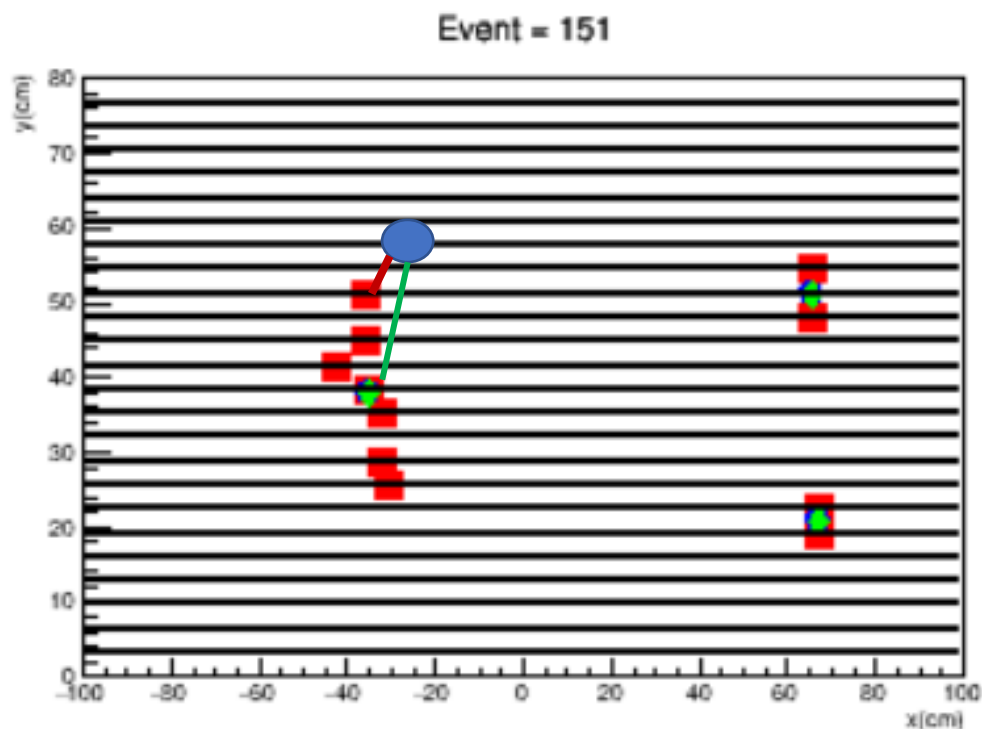
red=hits
green = cluster barycenter

The algorithm in short:

1. First hit is promoted to cluster and removed from hit array.
2. Scan over the hit array to find the first hit with XY distance below 10cm. Metrics: minimum distance between the hit and all the hits already part of the cluster
3. If some hit is added to cluster, remove it from the hit array and go back to point 2.
4. When no more hits can be added to the present cluster, compute cluster parameter (baricenter, T, average ToT)
5. if the hit array is not empty, create a new cluster with first hit and go back to point 2.

Note: code optimization to reduce clusterization step to few seconds (100K events)
The cluster will contain the list of hits -> useful in the last stage of the analysis

Event selection as in the past but new metrics to check the “distance” between a cluster and the projected hit



Old metrics: Distance to barycenter

New metrics: distance to closest hit of the cluster

Cluster multiplicity and streamer are computed from the multiplicity of best-matching cluster.

Selection cut on the triggered events (some tuning still needed):

- Extrapolated hit within test chamber acceptance ($5 < y < 75$ cm, $-60 < x < 60$ cm)
- $z_{dir} > 0.9$
- particle inverse beta within 0.2 from the average inverse beta peak

Criteria for efficiency:

- distance between the extrapolated hit and the closest cluster below 15 cm
- Time difference between the extrapolated hit and the closest cluster below 4 ns

Option to remove residual background based on the inverse beta distribution available in the original code: check ongoing

Test file results

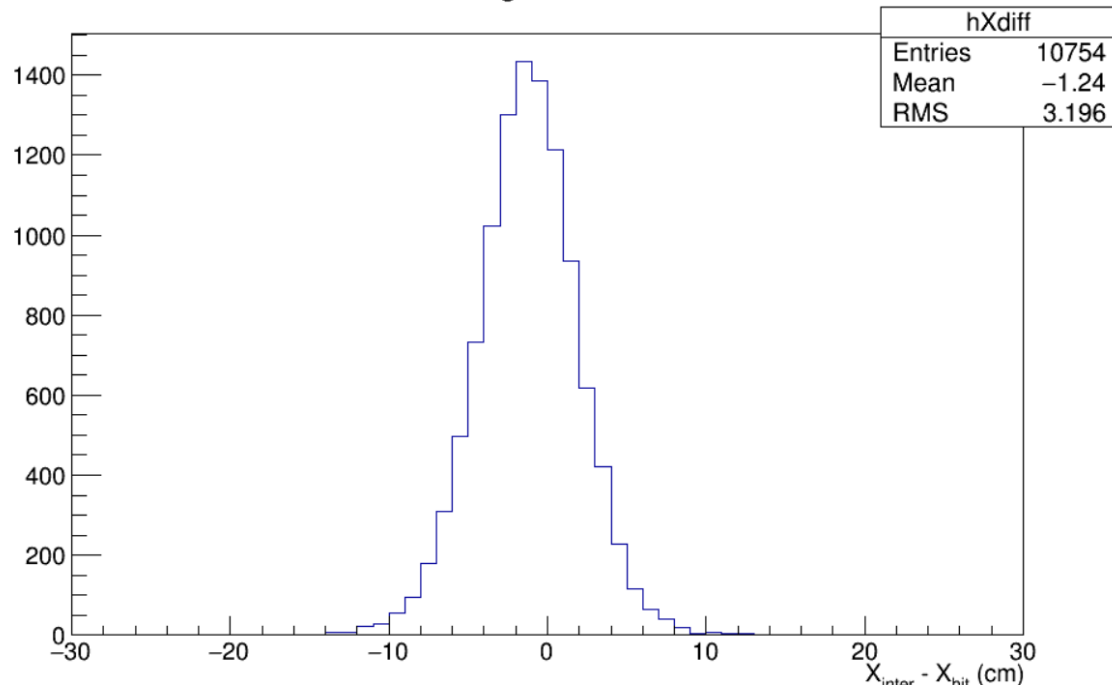
Results with test file:

Global efficiency = 0.94 (previously 86% with same selection/cuts)

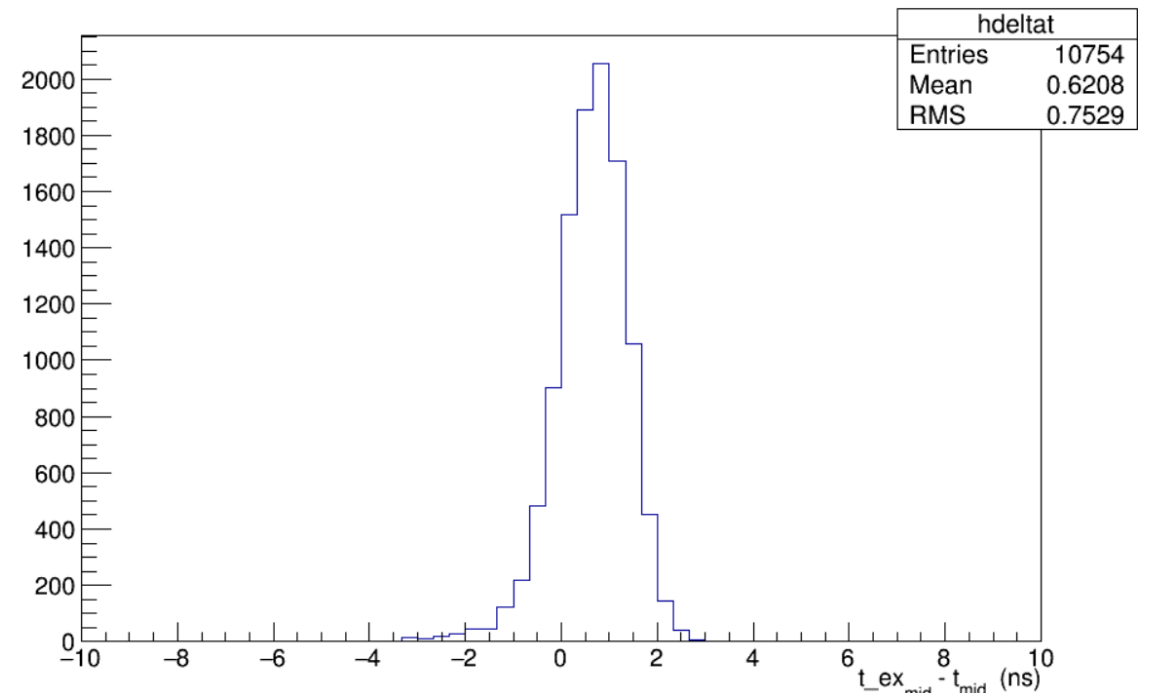
Streamer fraction = 0.095

Efficiency close to the one computed with CNAF efficiency code (0.96%)

ΔX longitudinal res.



ΔT time resolution



Residuals computed w.r.t. cluster barycenter

Offset on X residuals due to x calibration method (binning width)

Offset on T residuals under investigation (probably due to non-weighted average of the hits)

Test file results

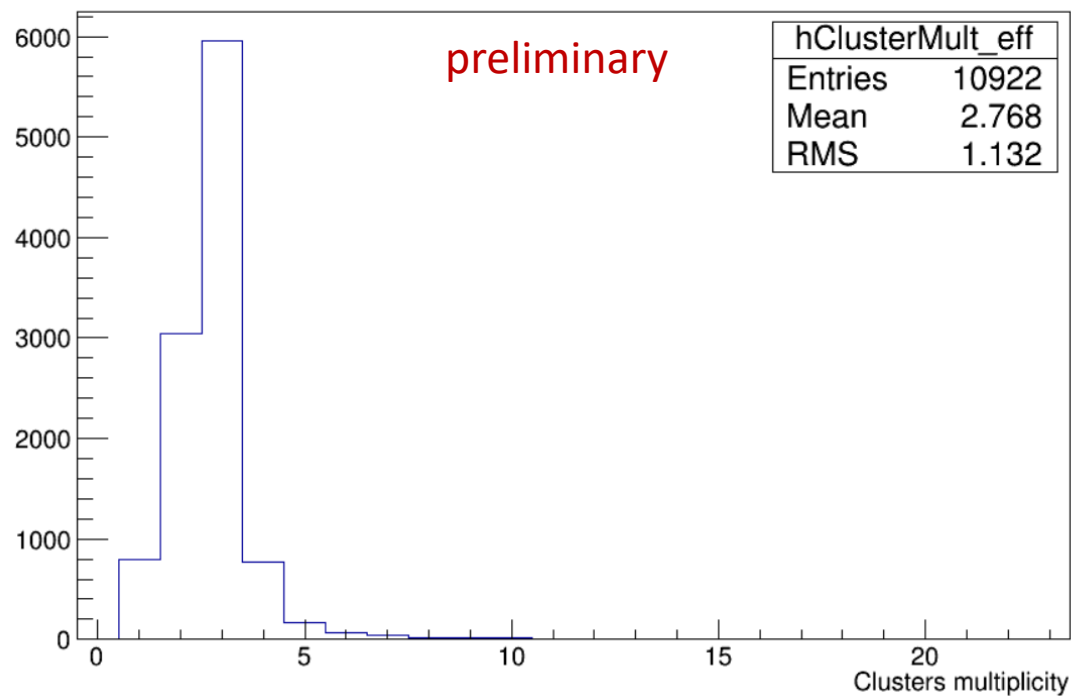
Results with test file:

Global efficiency = 94% (previously 86% with same selection/cuts)

Streamer fraction = 0.095

Efficiency close to the one computed with CNAF efficiency code (96%)

Cluster multiplicity test chamber ($|\beta - 1.05| < 0.20$, $z_{dir} > 0.9$)



Extrapolate MID ch.

