Corryvreckan integration modules for MPGDs

Elena Sidoretti

University of Roma Tor Vergata and INFN Roma2 - elena.sidoretti @roma2.infn.it

3rd DRD1Collaboration Meeting – December 2024

Overview

Corryvreckan's[1] **modular approach** simplifies the process of adapting it to various types of detectors and enhances the efficiency of data analysis by providing a streamlined, **user-friendly** interface.

- Reconstruct and analyse data from **pixel R/O**
- Modular structure
- Highly flexible and configurable

… also **strips**, combining X and Y

Corryvreckan

The Maelstrom for Your Test Beam Data

[1]D. Dannheim *et al* 2021 *JINST* **16** P03008

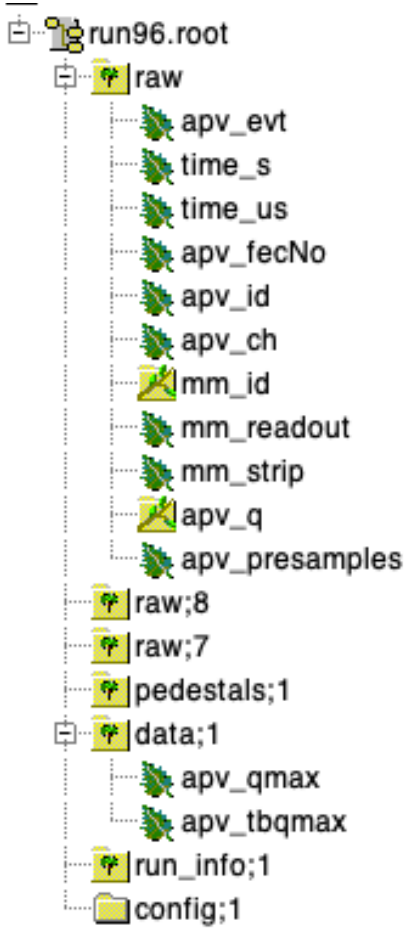
Recap from last year RD51 collaboration meeting

- Implementing an interface for the **Scalable Readout System** (SRS[2])+**APV25**[3] read by **mmDAQ1**[4]→ a very popular readout chain for acquiring and processing MPGD signals.
- **New Analysis modules:**
  - APVReader – useful for debugging
  - APV1Dto2D – reads two 1D chambers transforming them in 2D virtual detectors
  - APVReader2D – reads 2D chambers
- **New Geometry features:**
  - apv_1 and apv_2 – (geometry files) the APV number corresponding to the detector readout which is read by the APVReader module

My work until now:
Last year talk: https://indico.cern.ch/event/1327482/contributions/5708135/
16th Pisa meeting poster: https://agenda.infn.it/event/37033/contributions/227406/
16th Pisa meeting proceeding: https://doi.org/10.1016/j.nima.2024.169799

[2]S. Martoiu et al. JINST, 8:C03015, 2013.
[3]M. French et al. NIM A, 466(2):359–365, 2001
[4]M. e. a. Bianco. PoS, TIPP2014:202, 2015

Output from mmDAQ1



mmdaq1 (APV25 only , 1 FEC)

event number	apv_evt
daq time stamp	time_s
daq time stamp	time_us
srs fec number	apv_fecNo
chip id in fec	apv_id
channel id in chip	apv_ch
chamber name	mm_id
readout number*	mm_readout
strip number	mm_strip
ADC data	apv_q → vector containg the waveform of the charge signal (sampled every 25 ns)

How is the analysis performed

Configuration

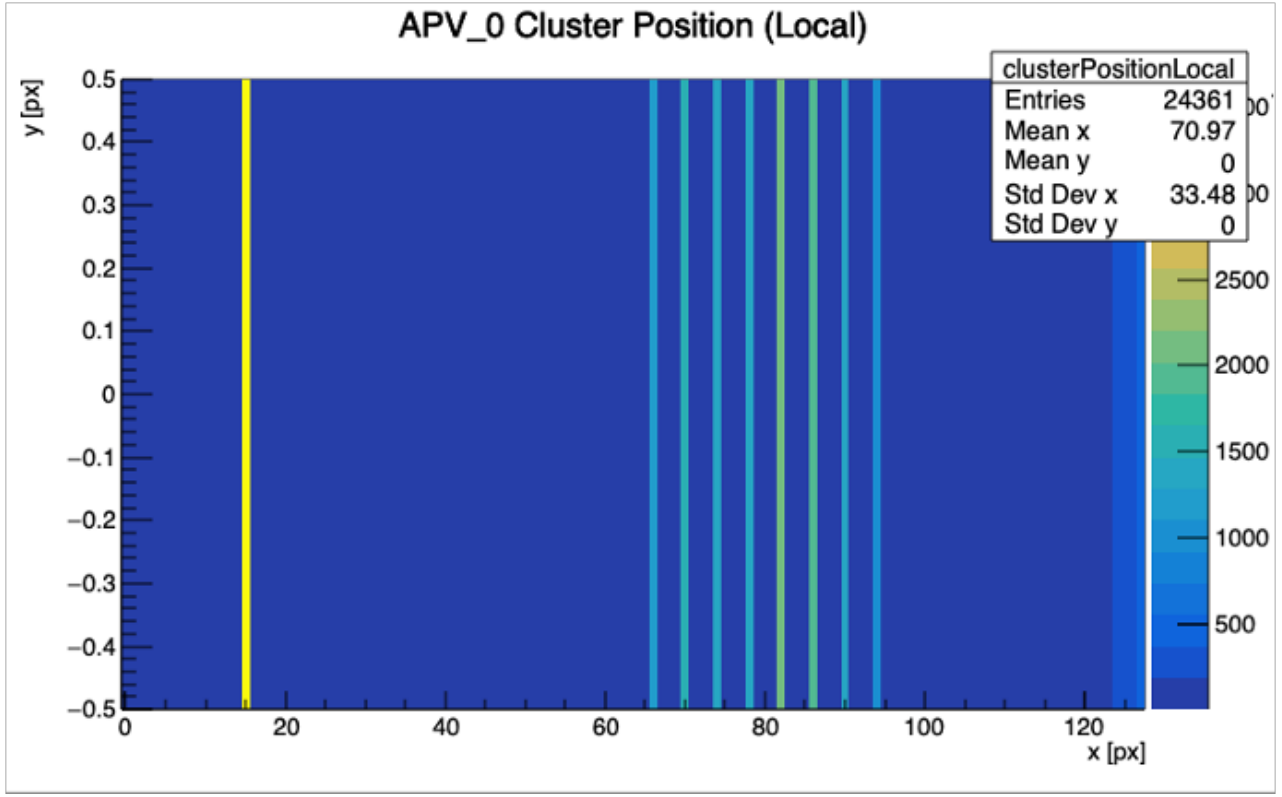
- Global module
  - Calls for detector **Geometry** file and sets the output directory
- Event Loader module
  - **APVReader** modules collect the data from the input file, and generate the pixels according to the readout layout
- Clustering module
  - Generate **clusters**
- Tracking module
  - Performs **tracking**
- Detector Under Test association module
- Analysis DUT
- Analysis Efficiency

Geometry

- Detectors info
  - Dimension
  - Orientation and position
  - Number of pixels
  - Pixel pitch
  - Type
  - APV number

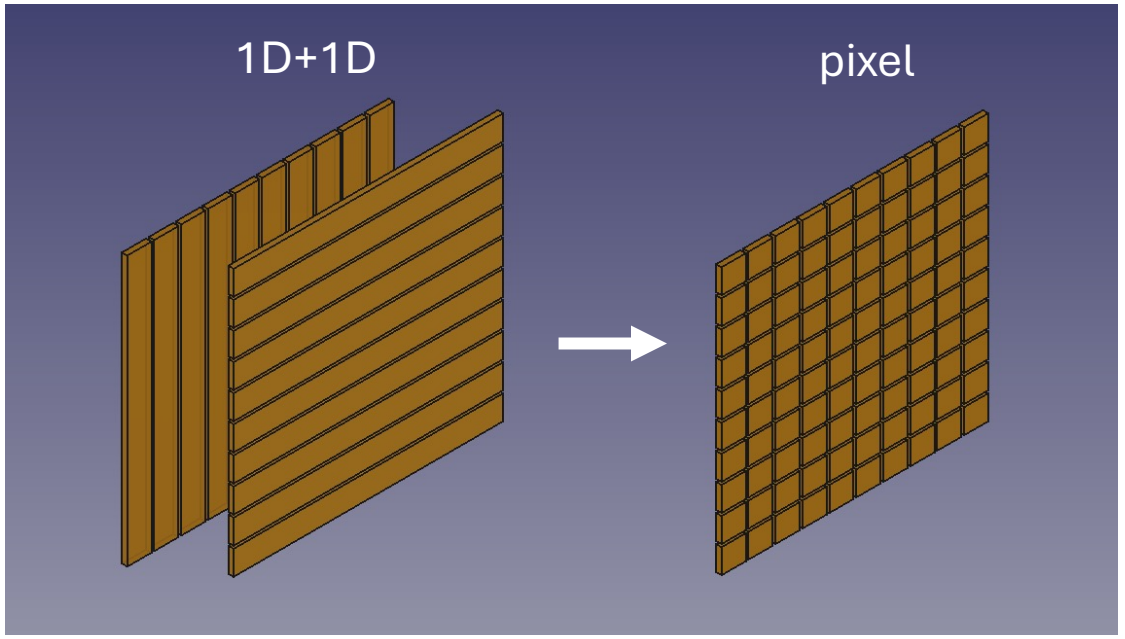
APVReader

- Reads the raw data from the detector
- The pixels are identified with row and column indexes, each strip is treated as a pixel with the length of the detector's active area
- The APV channel of the ROOT file given by mmDAQ (going from 0 to 127 in this case) corresponds to the 'columns', and the 'row' is set constant
- Useful for detector debugging and commissioning



APV1Dto2D

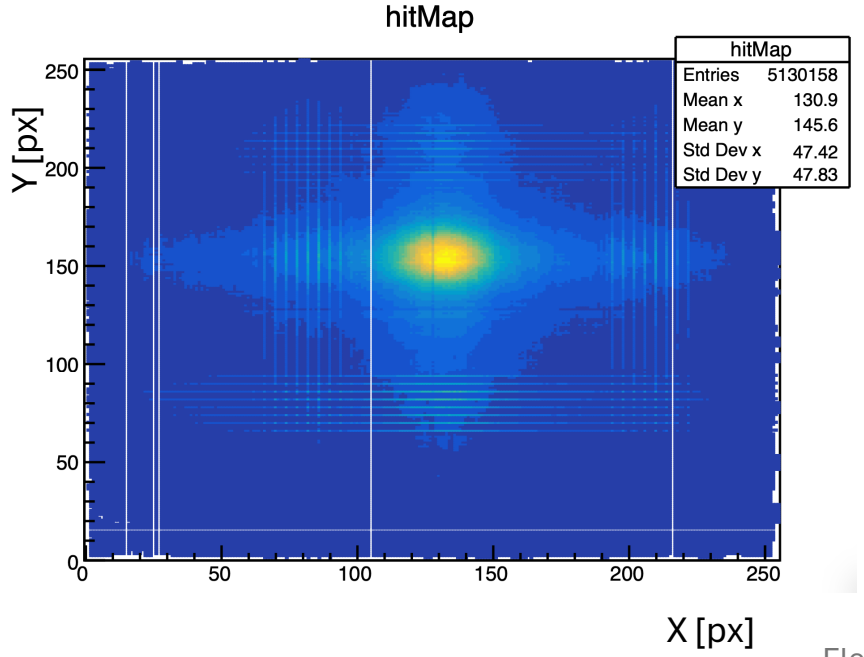
- **1D+1D = VIRTUAL 2D DETECTOR**
  The module APV1Dto2D creates a **Virtual detector** combining X and Y views and placing it between the two actual detectors
- The column of the detector is defined by the strips along the X-view direction
- The row of the detector is defined by the strips along the Y-view direction
- The charge of the pixel is the average of q_x and q_y
- The pixel cluster size is given by the product of the number of the contiguous hits along X and Y
- *(!!) this is not a native pixel R/O, ghost hits may occur*

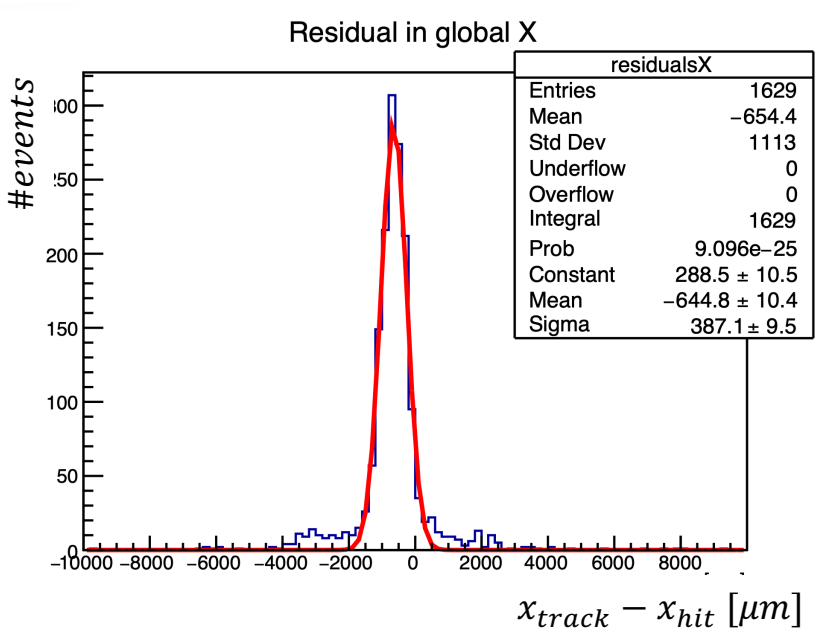


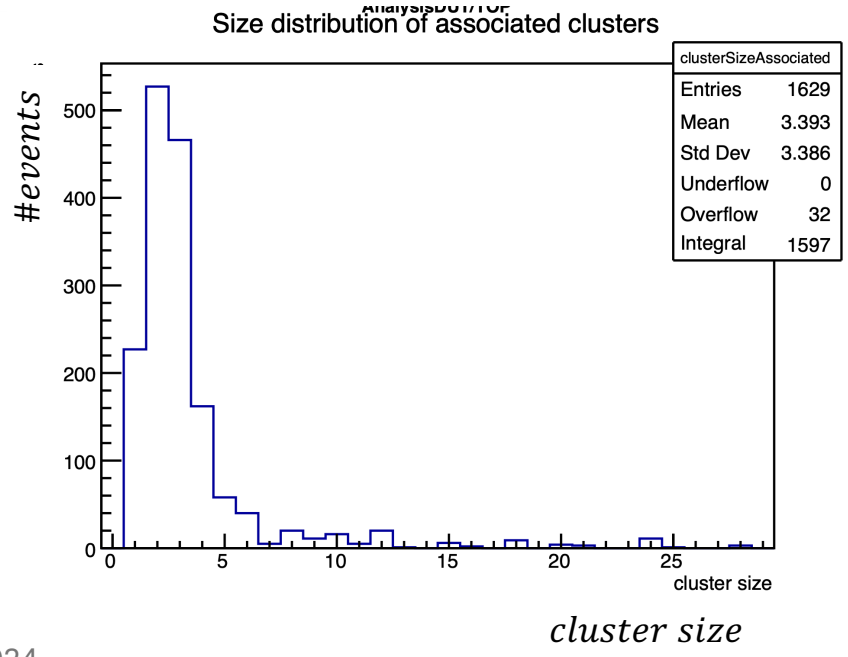
Analysis output

- Corryvreckan performs clustering, tracking, Detector Under Test's (DUT) cluster association to tracks, alignment, and it also provides detector efficiency and spatial resolution
- A cluster on a DUT plane is associated to a track when their distance is smaller than a fixed chosen value
- Resolution is given by the RMS of the associated track residuals
- 2D efficiency is calculated as:

$$\frac{\#\ tracks\ with\ associated\ cluster\ on\ the\ DUT}{\#\ tracks\ passing\ trought\ the\ DUT}$$

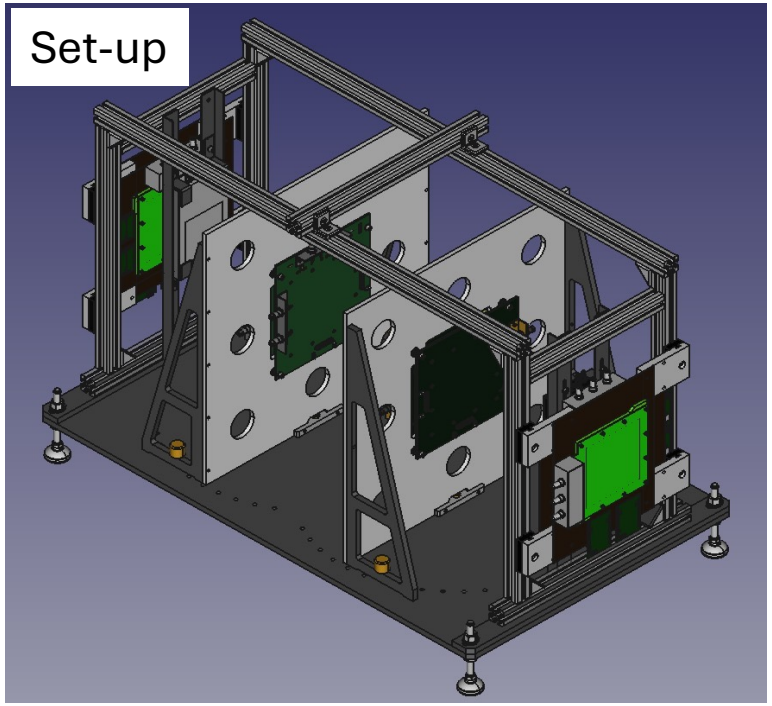






New Features

- SRS mmDAQ3 with APV25
- Multiple FECs
- Tracking is performed using GEM-uRWELL with 2D Compass like readout
- Detectors Under Test:
  - 2 GEM-uRWELL with 2D Compass like readout
  - 2 uRGroove with 2D groove readout
- Each pair of same type DUT is anchored together very closely, positioned in “enemy mode”, meaning with the cathode toward each other, and are supposed to have similar performance

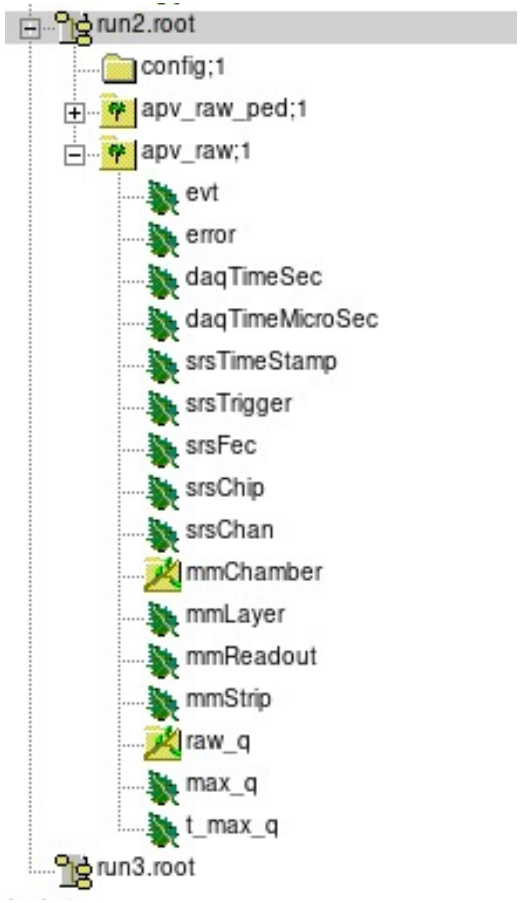


Check the results on my WG2 talk
https://indico.cern.ch/event/1442324/contributions/6257413/

What's new

- **Integration of mmDAQ3 data**
- **New detector geometry flags:**
  - FEC number – used for multiple FECs in mmDAQ3
  - 2D Readout type – Compass like, top read out and PAD
  - "Enemy" mode – to perform analysis of residuals in enemy
  - Gas gap – definition of the thickness of the gas gap
- **New pixel:**
  - information about charge along X and Y
  - time stamp along X and Y (and respective errors)
  - Time stamp of the pixel as: $\frac{t_x+t_y}{2} + event\_time\_stamp$
- **mmDAQ timing introduced:**
  - The events duration is set to be 25 ns (one time bin in mmDAQ) times n (to be set in the configuration)
- **Configuration flags:**
  - Run number is now a global parameter
  - Charge cut in APVReading modules can be set from configuration file, differentiated for X and Y
- **New Analysis modules!!**

Integration of mmDAQ3 data



mmdaq3 trees (apv_raw, multiple FECs)

event number	evt
error n/a yet	error
daqTime in seconds	daqTimeSec
daqTime in micro-seconds	daqTimeMicroSec
SRS time stamp (counter of clock cycles)	srsTimeStamp
trigger number	srsTrigger
FEC id number	srs Fec
Chip number within FEC	srsChip
Channel number within chip	srsChan
Chamber name	mmChamber
Chamber Layer number	mmLayer
Chamber readout strips	mmReadout
Strip number within readout	mmStrips
ADC data	raw_q

New detector geometry flags and new modules

Configuration

- Global module
  - Calls for detector geometry file and sets the output directory
- Event Loader module NEW IMPROVEMENTS!
  - APVReader modules collect the data from the input file, and generate the pixels according to the readout layout
- Clustering module
  - Generate clusters NEW IMPROVEMENTS!
- Tracking module
  - Performs tracking NEW IMPROVEMENTS!
- Detector Under Test association module
- Analysis DUT NEW IMPROVEMENTS!
- Analysis Efficiency

Geometry

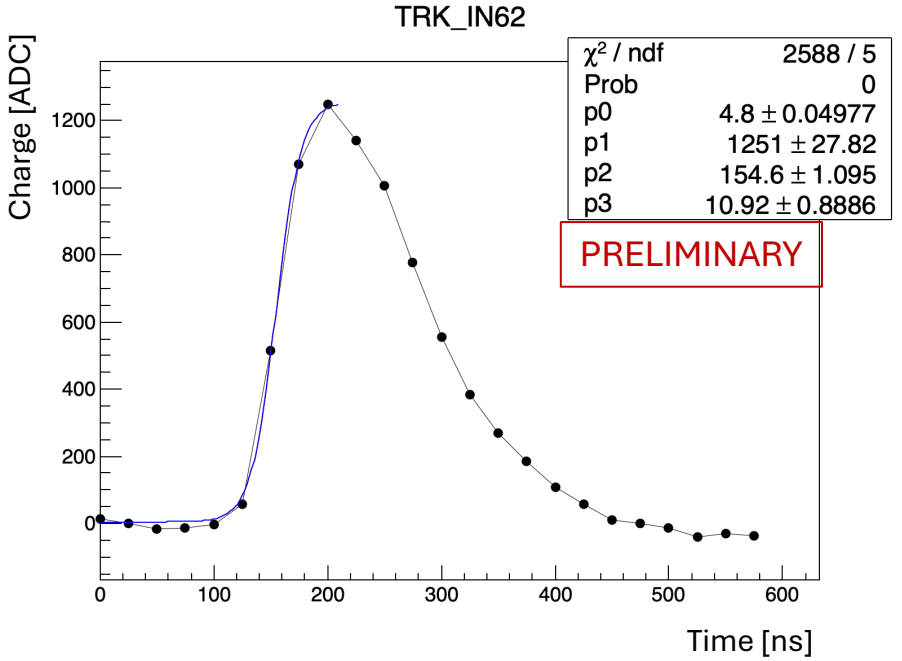
- Detectors info
  - Dimension
  - Orientation and position
  - Number of pixels
  - Pixel pitch
  - Type
  - APV number
  - FEC number NEW!
  - Readout type NEW!
  - Gas gap NEW!
  - Enemy role NEW!

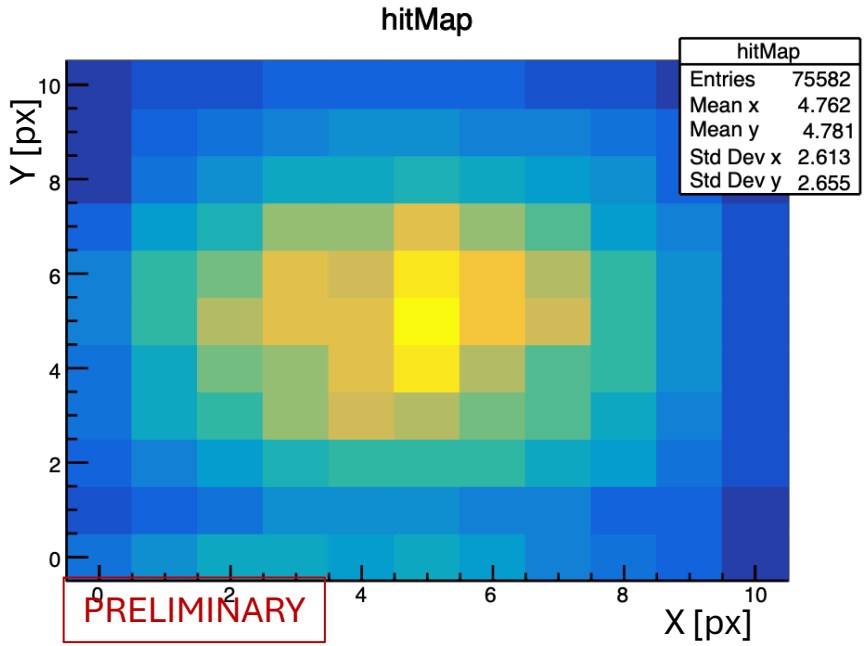
New Pixel

- **The original pixel:**
  - *timestamp* - timestamp in nanoseconds
  - *raw* - charge-equivalent pixel raw value. If not available set to 1
  - *charge* - pixel charge in electrons. If not available, set to raw for correct charge-weighted clustering
- **The new pixel CONSTRUCTOR:**
  - *raw* – charge of the strip along X (column) in ADC
  - *charge* - pixel charge in ADC
  - *time_col* – time of the hit in the strip along X
  - *time_row* – time of the hit in the strip along Y
  - *timestamp* = $\frac{t_x+t_y}{2} + event\_time\_stamp$
- In the analysis modules you will now be able to get information about the strips separately from their pixel reconstruction, without losing information
- You will be able to set the timing along X and Y also after performing other modules (es. uTPC analysis WIP!)

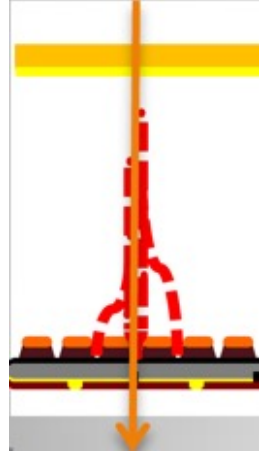
New modules: APVReader2D

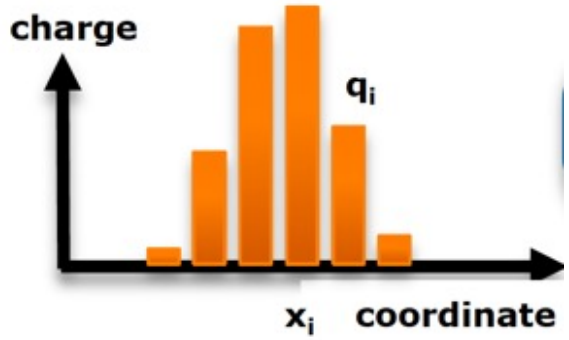
- For different readout type chambers, the pixel charge is calculated differently
  - For TOP readout it is given as the charge on the X hit
  - For Compass readout it is the sum of the charge on X and Y and the events with only one view fired are discarded
- The TGraph *charge vs time* of each strip signal is fitted with a Fermi-Dirac function to extrapolate the time of the hit as the inflection point of the rising edge
- Extra: APVReaderPAD - new module to store as Corryvreckan's pixels the information from a MPGD with PAD Readout





New modules: ClusteringStrips

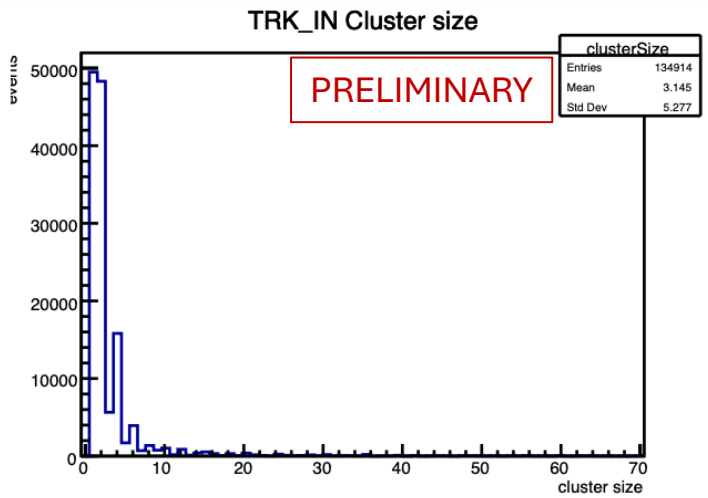


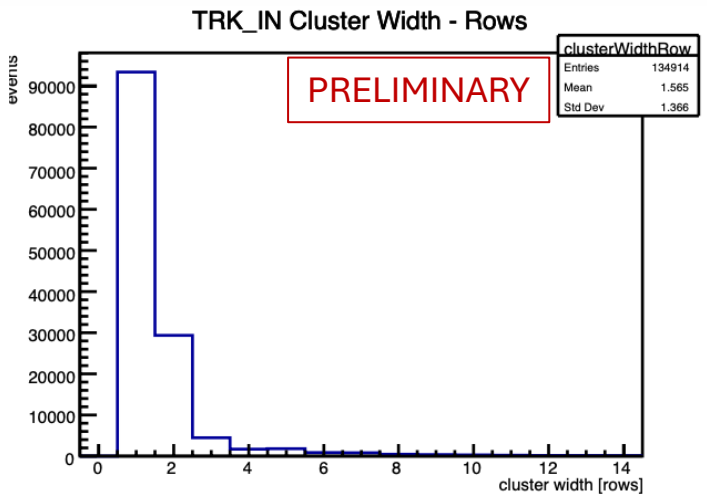


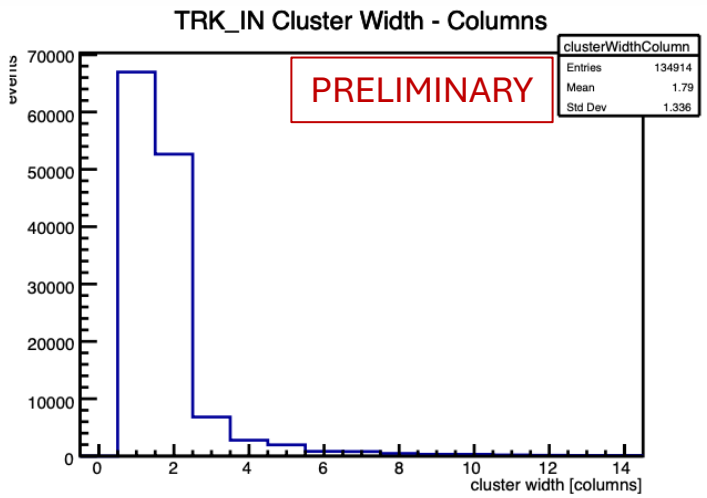
- To store a pixel in the same cluster it is required:
  - maximum one contiguous strip not fired to select the next
  - a time window selection between strips (can be set from configuration file)

if (col_diff >= (neighbor_radius_col_ +1)|| row_diff >= (neighbor_radius_row_+1) || time_col_diff >time_cut_ || time_row_diff > time_cut_) {
return false;

- The clusters vector will be stored to be charge ordered, with maximum charge pixel first
- The position of the cluster is calculated with the charge centroid method, using the charge along X and along Y separately for each coordinate

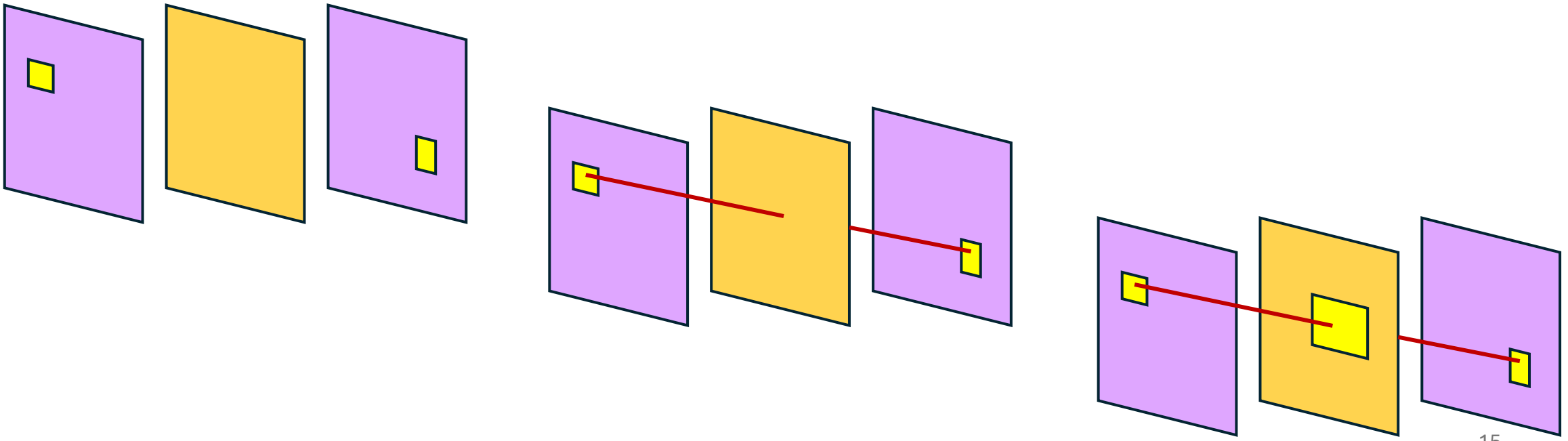






New modules: Tracking4D

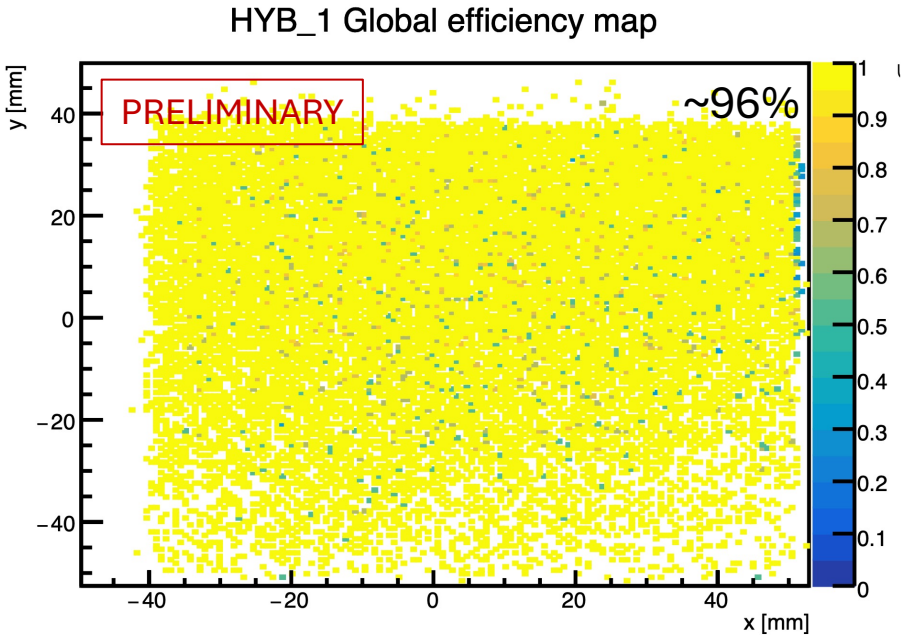
- The tracking is performed with «GoldenClusters», meaning with maximum charge clusters
- The trackers are the only detectors required to be fired in order to start building a track

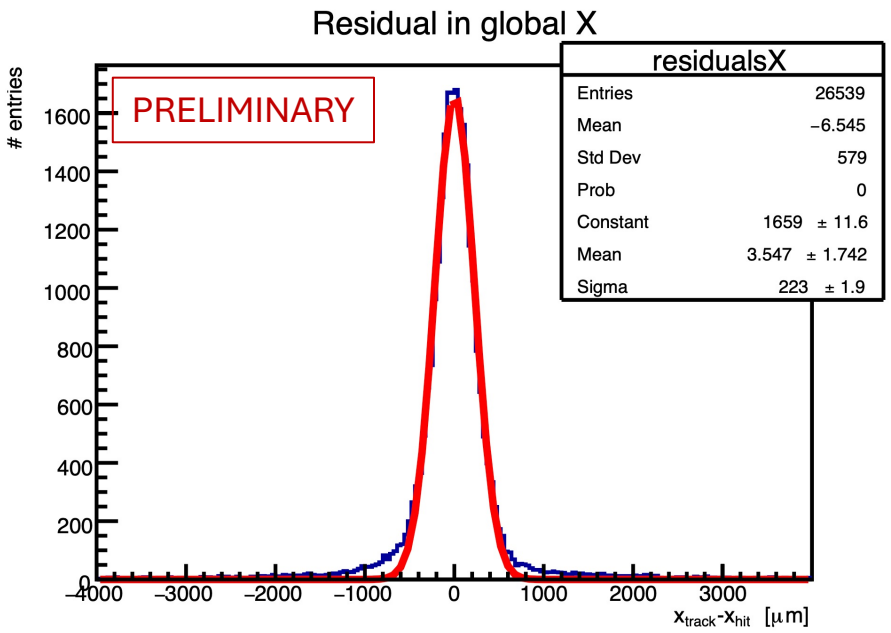


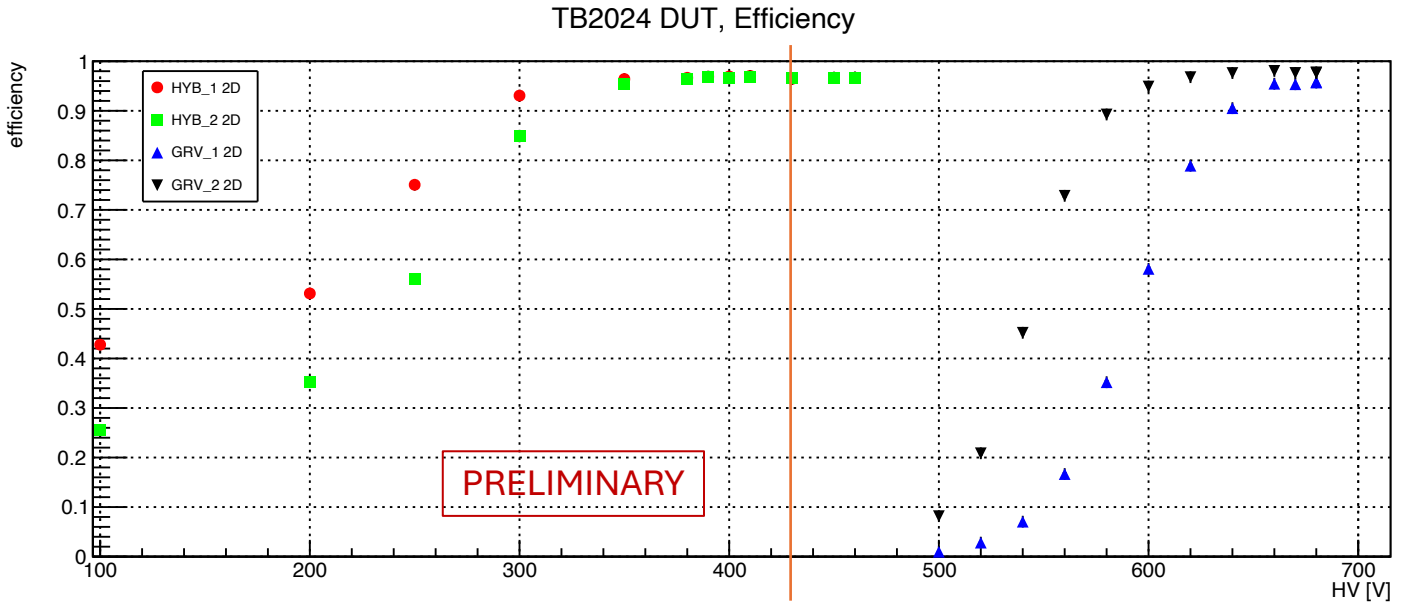
New Features: first results

Example of efficiency calculated with Corryvreckan.

$$\text{Efficiency} = \frac{\#\ tracks\ with\ associated\ cluster\ on\ the\ DUT}{\#\ tracks\ passing\ trought\ the\ DUT}$$



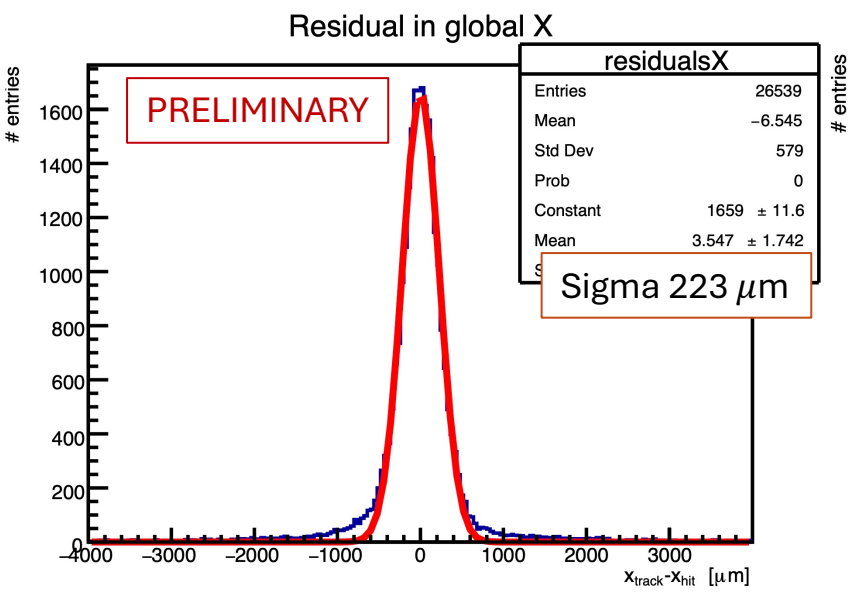


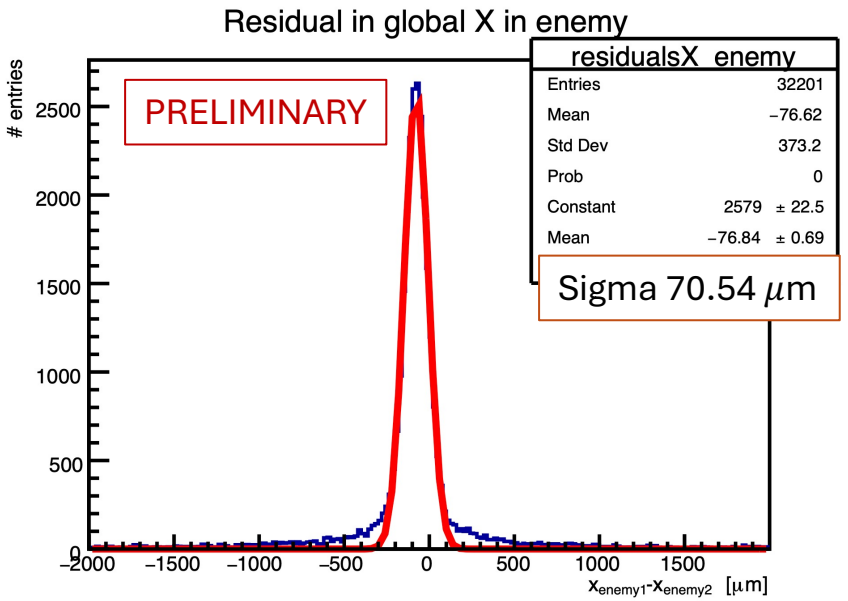


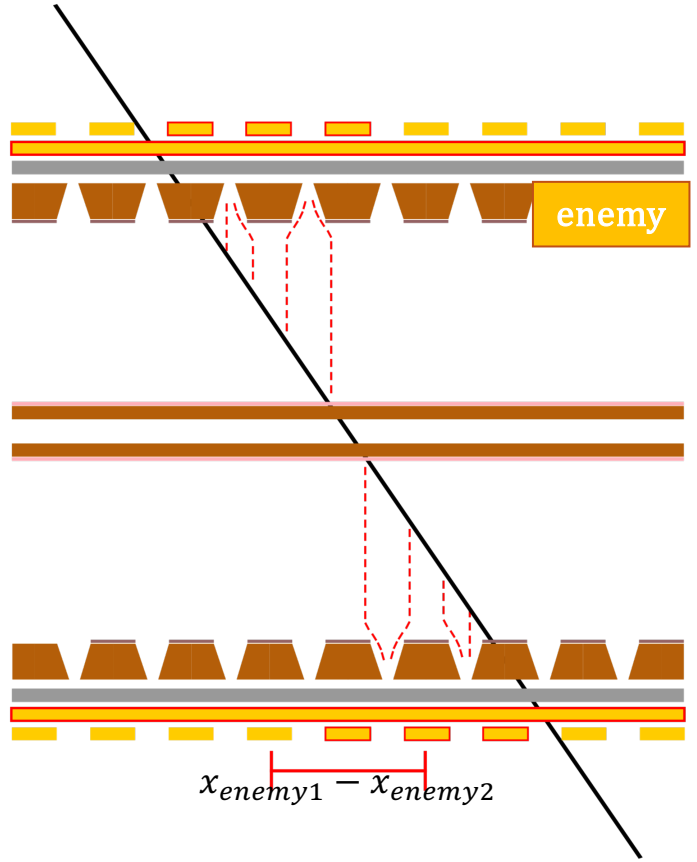
New Analysis modules: AnalysisDUT_enemy

Implementation to study inclined tracks

- If the detector has the flag «enemy» true, the module will calculate the Residuals also with the enemy method
- The enemy method calculates the distance between the clusters' centre on each readout

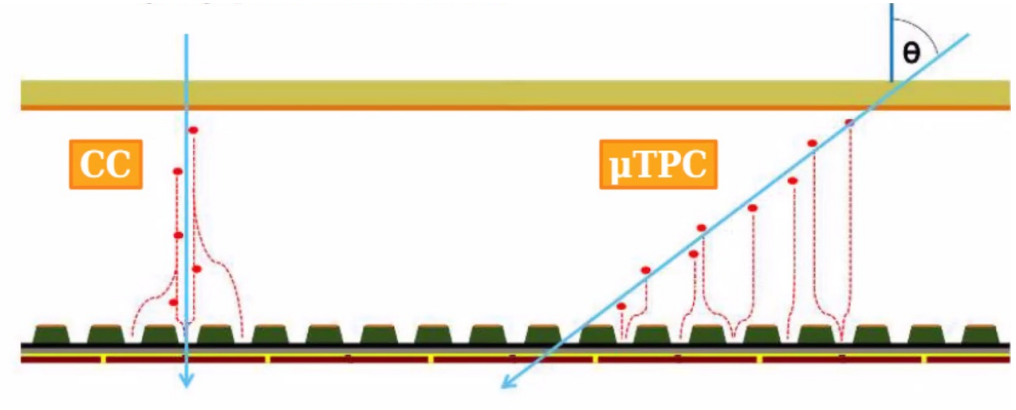


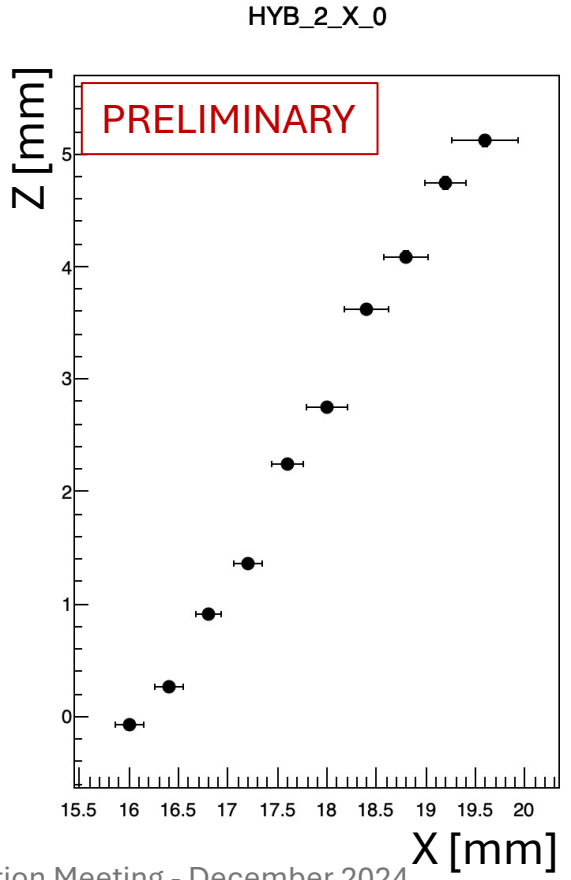


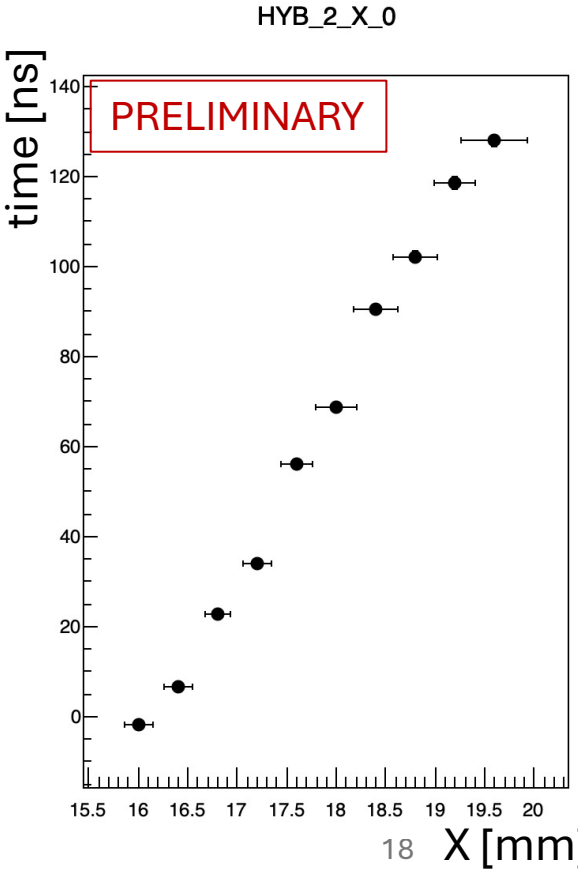


New analysis modules: WIP μTPC

- the clusters position is modified correcting it with the μTPC
- In APVReader will be required the flag “fit” to be true
  - Each Charge_vs_Time histogram is fitted with a Fermi Dirac function to find the time of receiving signal
- Time is used with the drift velocity to perform μTPC and select the position along z in the gas gap in which the particle passed







Next steps

In order to make the code as generic as possible:

- Fit to the signal in APVReader will be set from configuration file allowing several different functions to be used
- Use of the detector geometry flags to generalize different steps of the analysis
- Introduction of new tracking module to be used also for 1D detectors

- Suggestions for further improvements needed by the community?

Thank you!

Thanks to

Marinagela Bondì, Annalisa D’Angelo, Riccardo Farinelli, Matteo Giovannetti, UniRoma 2 & LNF-INFN w.g.