Preliminary beam test results

F. Grancagnolo

Dec. 1st, 2021

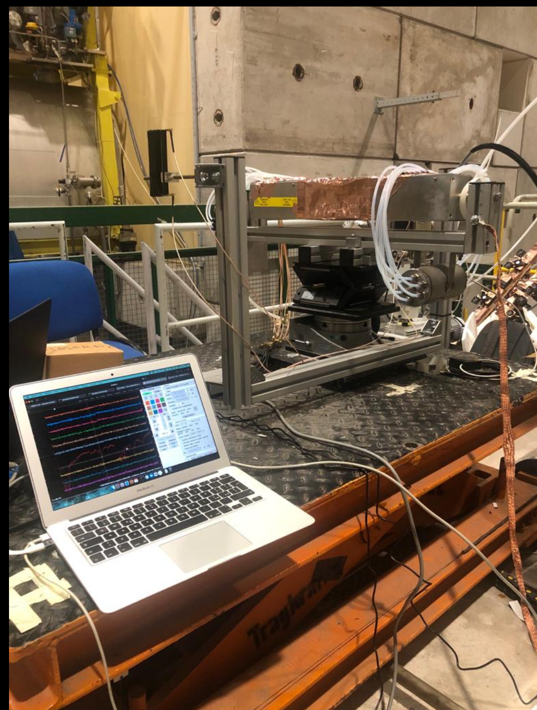
The test

Purpose

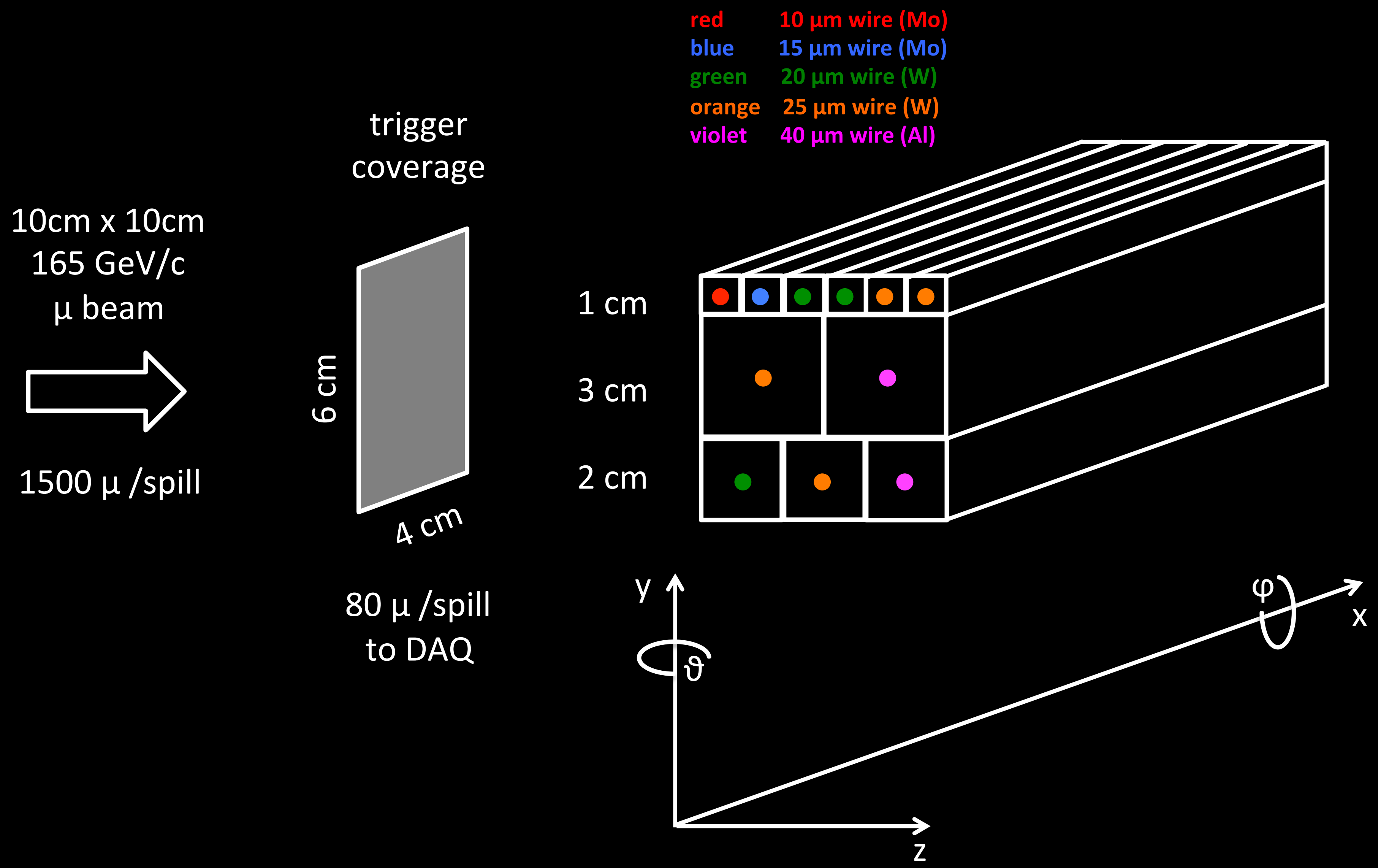
- **Demonstrate the ability to count clusters at a fixed βγ (e.g. muons at a fixed momentum – 165 GeV/c) by changing:**
  - the cell size (1 – 3 cm)
  - the track angle (0° to 60°)
  - the gas mixture (90/10: 12 cl/cm, 80/20: 20 cl/cm, for m.i.p.)
- **Establish the limiting parameters for an efficient cluster counting:**
  - cluster density as a function of impact parameter
  - space charge (by changing gas gain, sense wire diameter, track angle)
  - gas gain stability
- **Train different cluster counting algorithms**

SHIFTERS

Claudio CAPUTO	UC Louvain
Federica CUNA	INFN Lecce
Nicola DE FILIPPIS	INFN Bari
Francesco GRANCAGNOLO	INFN Lecce
Matteo GRECO	INFN Lecce
Sergey GRIBANOV	BINP Novosibirsk
Kurtis JOHNSON	U of Florida
Sasha POPOV	BINP Novosibirsk
Angela TALIERCIO	UC Louvain
Shuiting XIN	IHEP Beijing

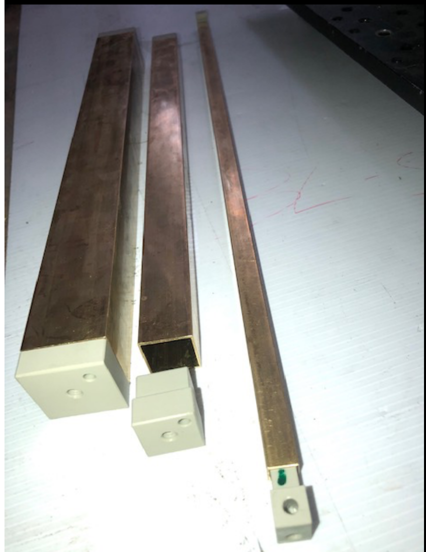


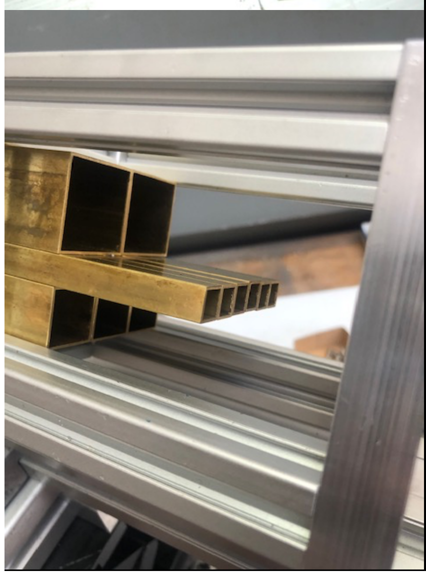
Drift tubes schematics



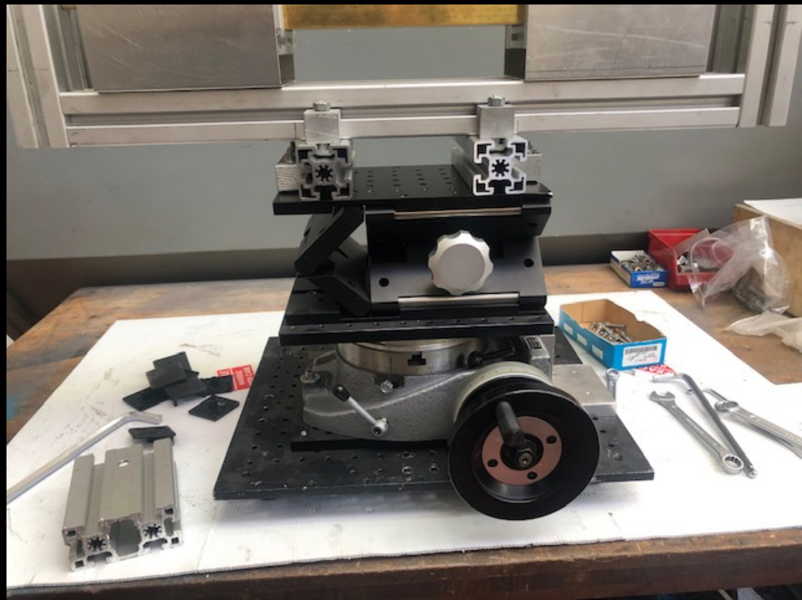
The Hardware

The drift tubes



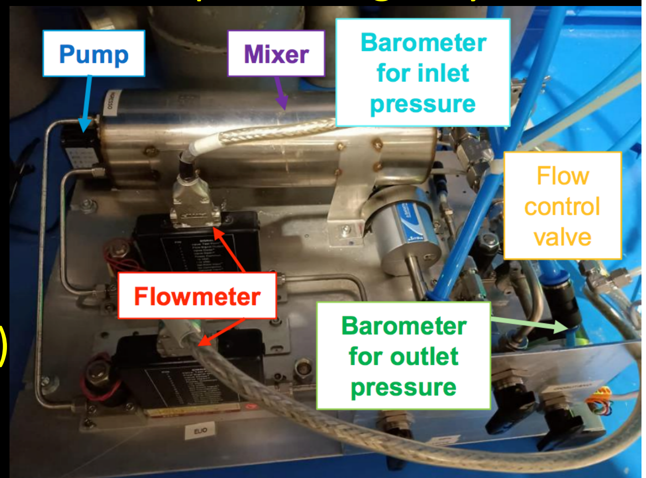


The rotating table

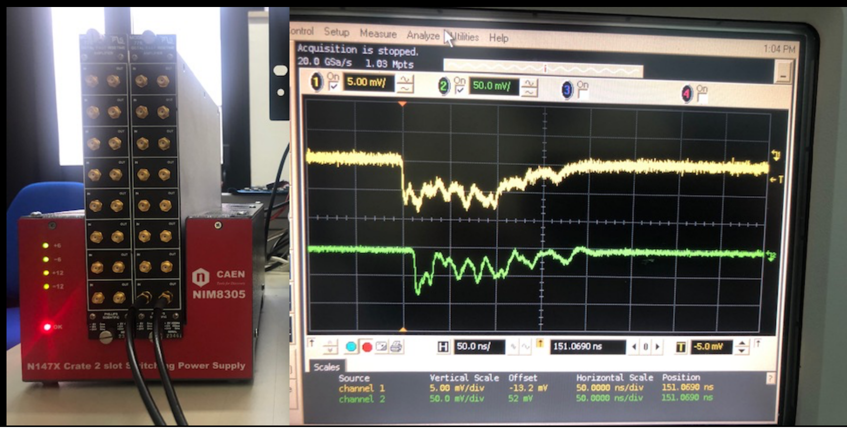


The HV system

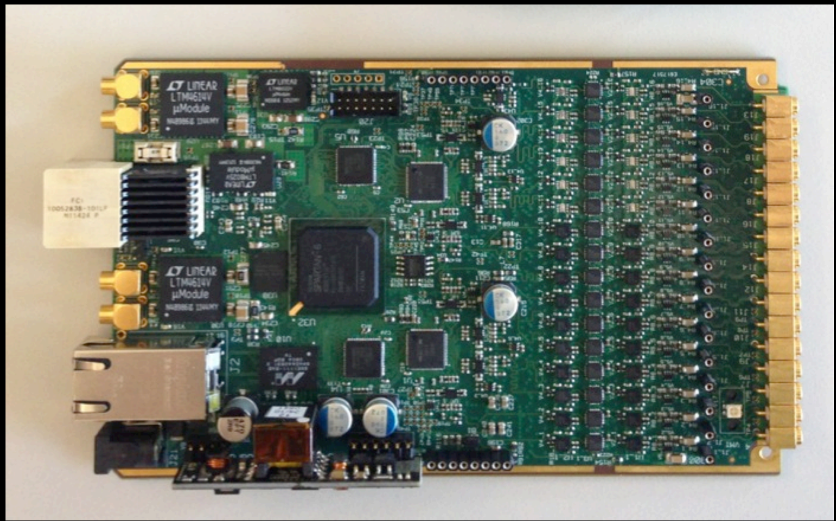
The portable gas system



The gain 10 - 1.7 GHz amplifier (Phillips 775)



DRS4 DAQ board and Trigger

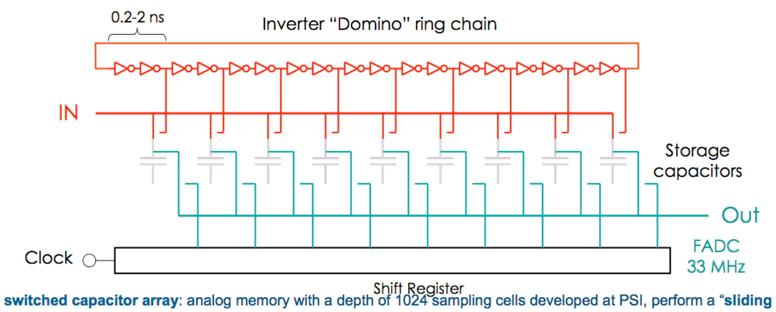


16 channels data acquisition board designed and used by the MEG2 experiment at PSI ($\mu \rightarrow e + \gamma$)
(credit to S. Ritt, Paul Sherrer Institute, Zurich, Switzerland)

12cm x 6cm upstream and downstream scintillator tiles (designed and used as timing counter of the MEG2 experiment at PSI) used in coincidence and readout by SiPM's

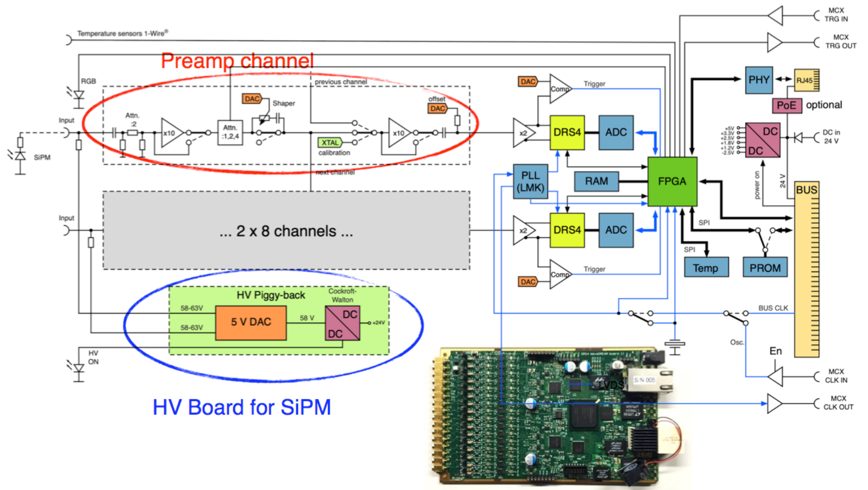
The DAQ board

DRS4 chip

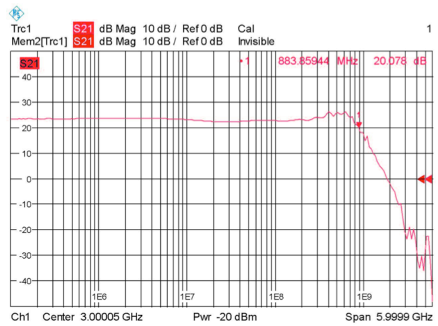


* Analog switched capacitor array: analog memory with a depth of 1024 sampling cells developed at PSI, perform a "sliding window" sampling.
* 500MSPS ↔ 5GSPS sampling speed with 11.5 bit signal-noise ratio
  * 8 analog channels + 1 clock-dedicated channel for sub 50ps time alignment
* Pile-up rejection O(~10 ns)
* Time measurement O(10 ps)
* Charge measurement O(0.1%)

WaveDREAM Board (WDB)

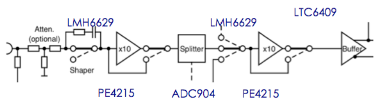


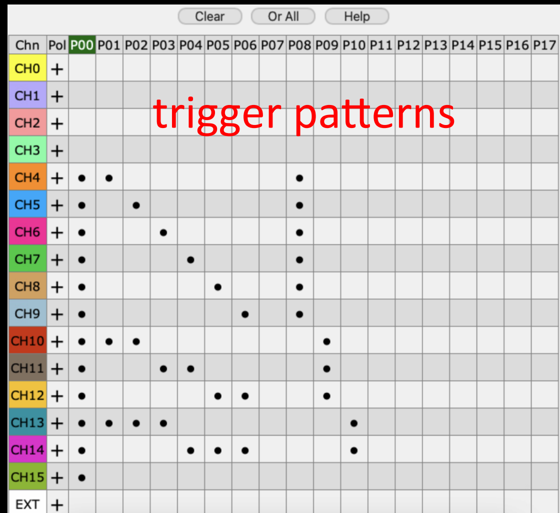
Preamp channel

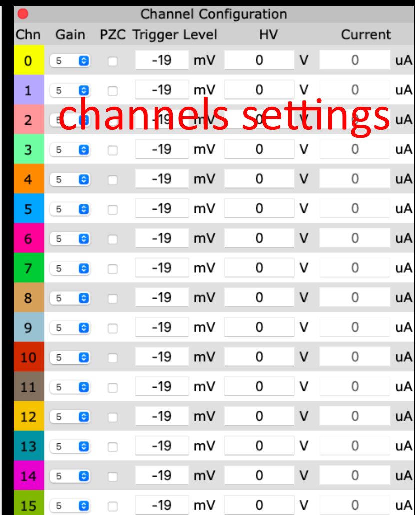


Gain	BW_{3db} (MHz)	Noise (mV)
1	940	0.37
10	880	0.40
100	300	1.2
100	500	1.7
100	800	3.3

Different compensations







Data format

WDS: Binary format

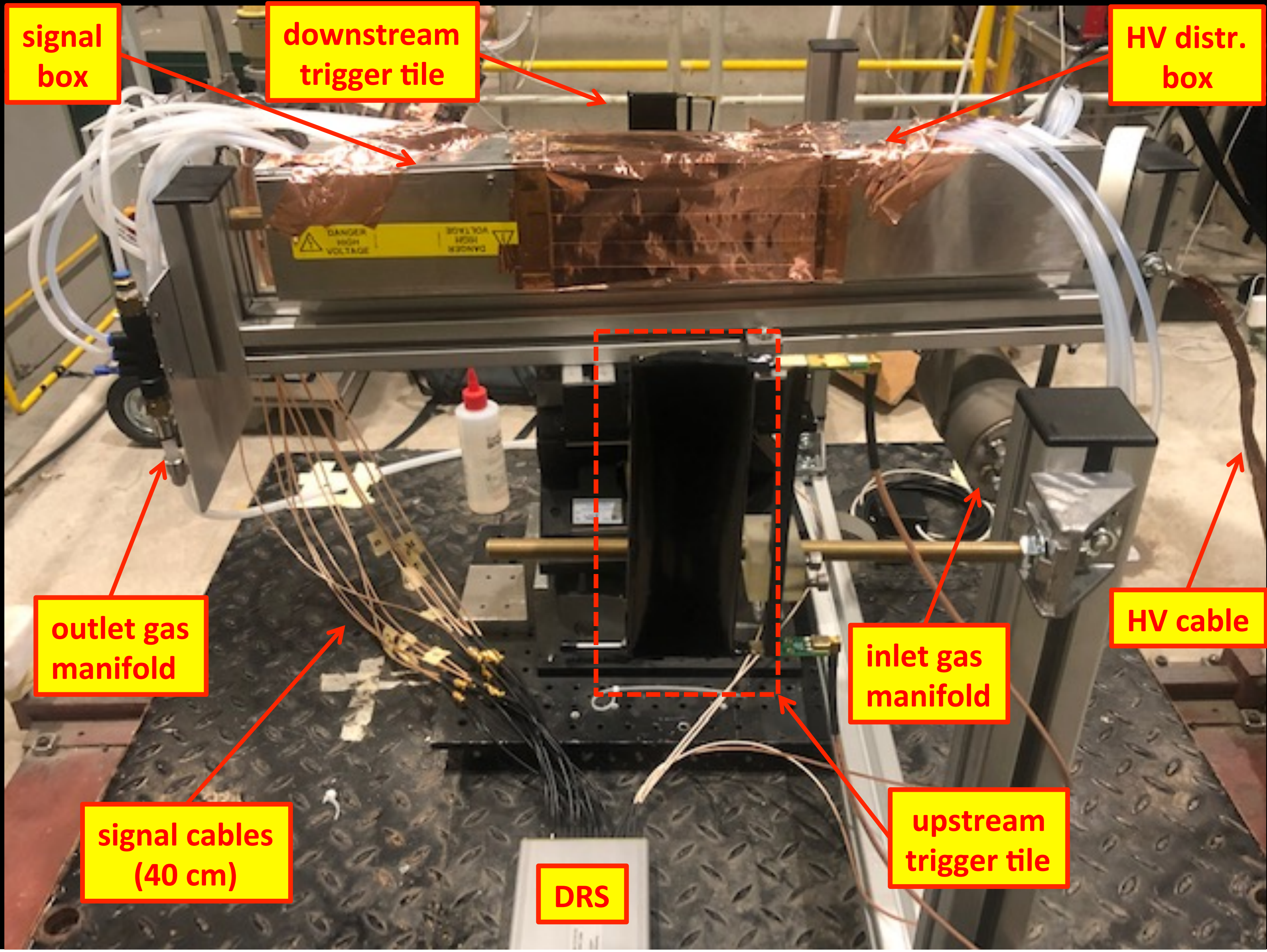
- Header relating to the board consisting of the words:
  - DRS8
  - TIME
  - B#XXX (XXX represents the card number and changes according to the WDB, in this case 033)
  - Calibration information

- Header EVENT
  - Serial. Number
  - Time information
  - Channel Information

<u>The macro for converting binary files to root files is ready.</u>

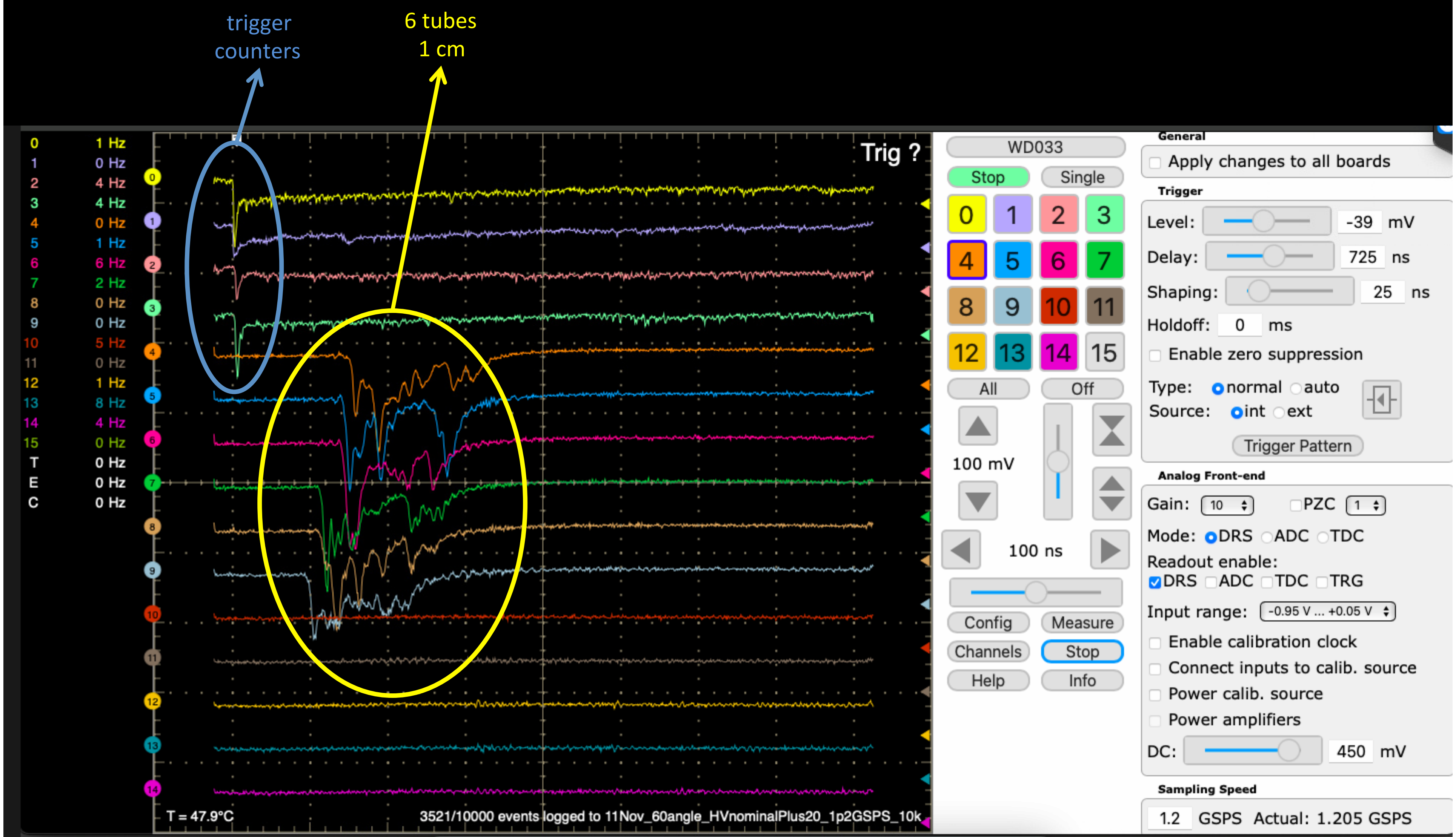
Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	'R'	'S'	'8'	File header, Byte 3 = version
1	'T'	'I'	'M'	'E'	Time Header
2	'B'	'#'	Board number		Board serial number
3	'C'	'0'	'0'	'0'	Channel 0 header
4	Time Bin Width #0				Effective time bin width in ns for channel 0 encoded in 4-Byte floating point format
5	Time Bin Width #1				
...	...				
1027	Time Bin Width #1023				
1028	'C'	'0'	'0'	'1'	Channel 1 header
1029	Time Bin Width #0				Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
1030	Time Bin Width #1				
...	...				
2052	Time Bin Width #1023				
2053	'E'	'H'	'D'	'R'	Event Header
2054	Event Serial Number				Serial number starting with 1
2055	Year		Month		Event date/time 16-bit values
2056	Day		Hour		
2057	Minute		Second		
2058	Millisecond		Range		Range center (RC) in mV
2059	'B'	'#'	Board number		Board serial number
2060	'C'	'0'	'0'	'0'	Channel 0 header
2061	Scaler #1				Scaler for channel 0 in Hz
2062	'T'	'#'	Trigger cell		Channel 0 first readout cell
2063	Voltage Bin #0		Voltage Bin #1		Channel 0 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2064	Voltage Bin #2		Voltage Bin #3		
...	...		...		
2574	Voltage Bin #1022		Voltage Bin #1023		
2575	'C'	'0'	'0'	'1'	Channel 1 header
2576	Scaler #2				Scaler for channel 1 in Hz
2077	'T'	'#'	Trigger cell		Channel 1 first readout cell
2578	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2579	Voltage Bin #2		Voltage Bin #3		
...	...		...		
3089	Voltage Bin #1022		Voltage Bin #1023		
3090	'E'	'H'	'D'	'R'	Next Event Header
...					

The setup

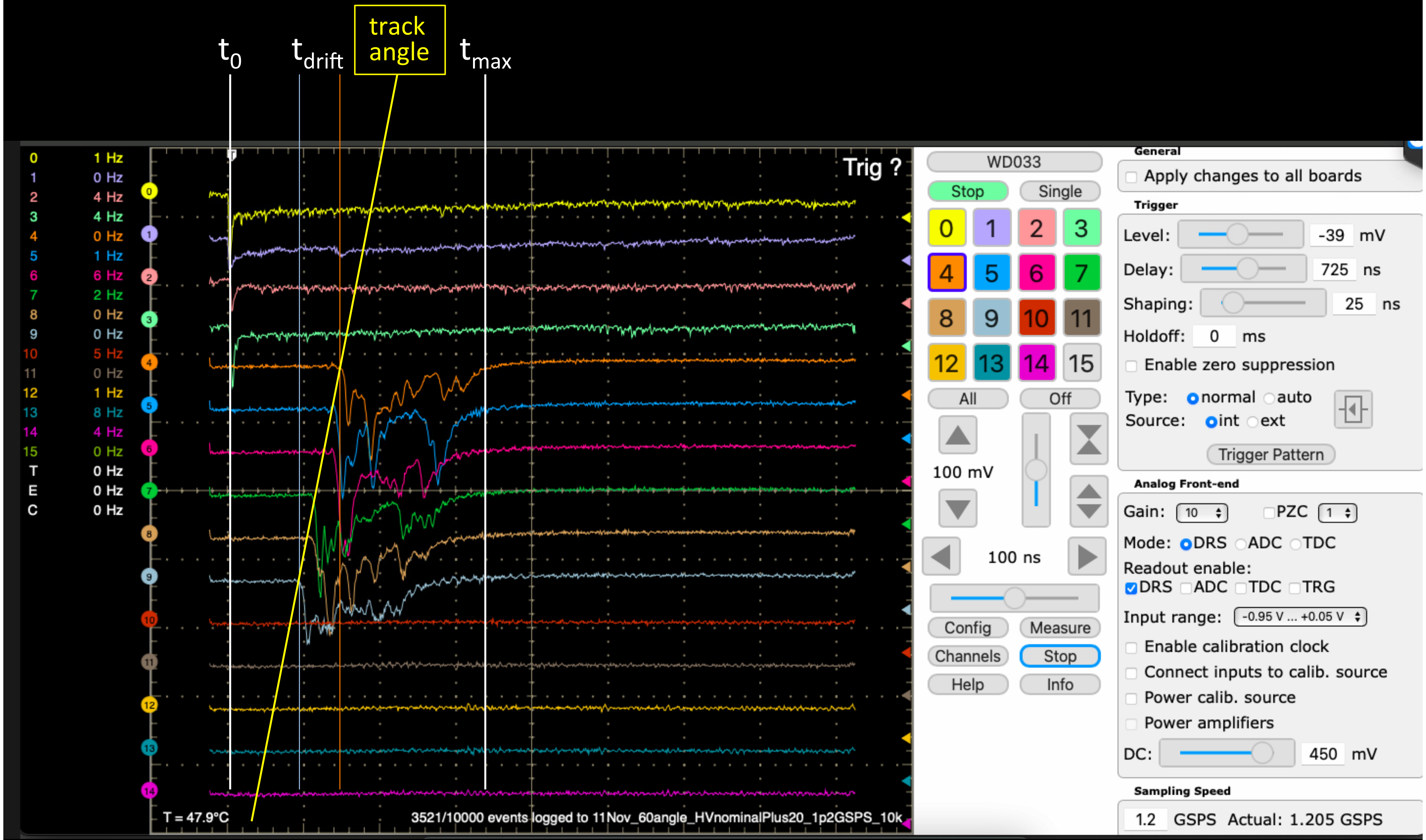


signal box
downstream trigger tile
HV distr. box
outlet gas manifold
signal cables (40 cm)
DRS
inlet gas manifold
upstream trigger tile
HV cable

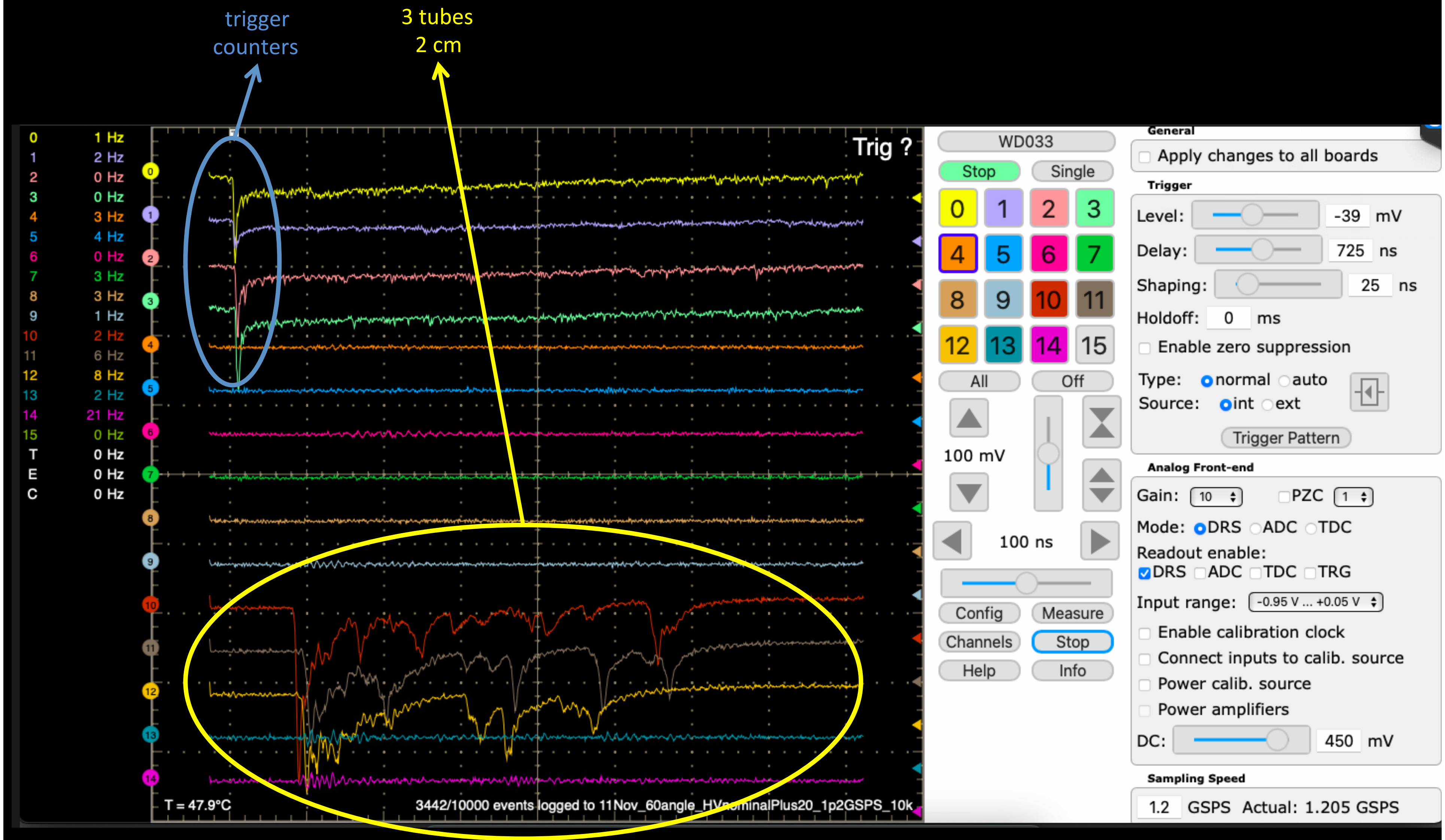
Event display



Event display



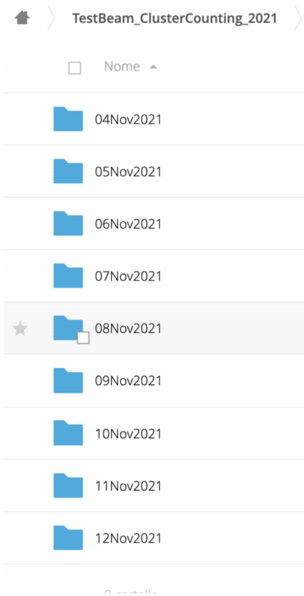
Event display



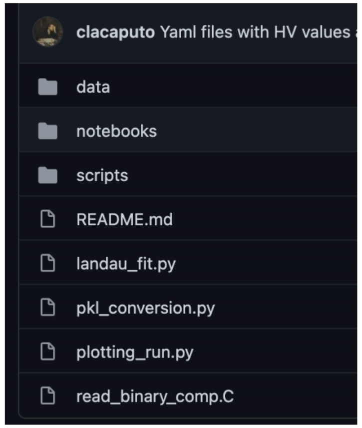
Data storage

- Raw data stored on CERNBox: https://cernbox.cern.ch/index.php/s/lzI6PygC4tx1DCE
  - Public accessible; you can download it
  - Only binary files provided
- Github with online analysis code and preliminary offline code: https://github.com/clacaputo/drifttubes_analysis
  - Code for converting RAW data to ROOT and PKL files is provided in the repository
  - Basic Runs database, based on YAML files, is provided
- Log Book (To be Updated): https://codimd.web.cern.ch/9UXozxEwRK6vsJ4ilia9BA

- https://cernbox.cern.ch/index.php/s/lzI6PygC4tx1DCE
- One folder for each day of data taking, each folder should contain:
  - txt file with informations about the run (To be moved in the log book)
  - RAW files (binary),
  - ROOT and pkl files can be easily created with the code provided in my repo (more in next slides)



- Github: https://github.com/clacaputo/drifttubes_analysis
  - YAML config files for the different runs
  - Decoding code (Thanks to Gianluigi!)
  - Conversion code to ROOT, pkl files, parallellized on HTCondor
  - script for submitting on HPC facilities, easily customisable
  - code for online Landau fit, plus plot productions
  - Peak finding (BETA version)



- YAML config files for the different runs, example at this link

Analysis plan

- Determine **gas gain** for the different (9) configurations and for different gas mixtures
- Study the **space charge** limitations for an efficient cluster collection
- Train different **cluster counting algorithms** by comparing to a well known method (PeakFit)
- Check **Poisson nature** of cluster counting (tube size, track angle, gas mixture)

Some considerations on gas gain

Capacitance per unit length (cylinder approximation):

$$C_L = 2\pi\varepsilon_0/\ln(R/r_w) \qquad C_{tube} = C_L \times L$$

Inductance per unit length (cylinder approximation):

$$L_L = 2\times10^{-7}\ \ln(R/r_w)\ \mu H/m$$

our drift tubes: $L = 30, 40$ cm $\quad R = 0.4, 0.9, 1.4$ cm

$r_w = 5.0, 7.5, 10.0, 12.5, 20.0\ \mu m$

Characteristic Impedance: $Z = \sqrt{L/C}$

At **gas gain = 10^5**, one single electron deposits a charge

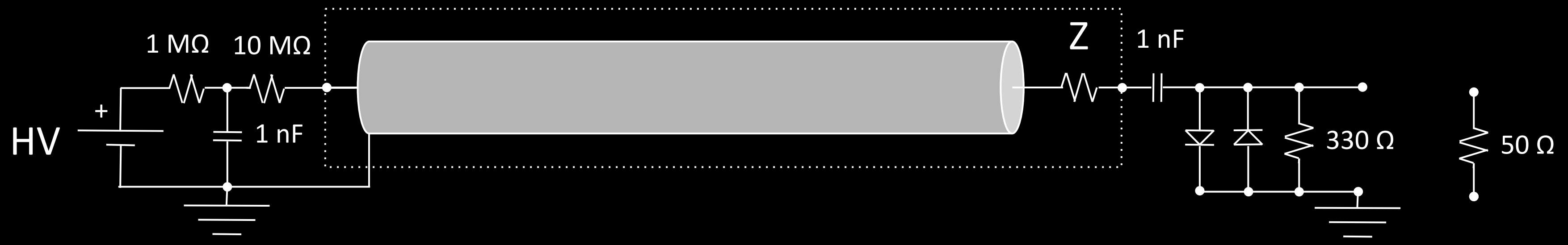
$Q_0 = 1.6\times10^{-19}\times10^5$ Coul = **16 fCoul**

and the pulse height generated by a single electron would be:

$$\Delta V = Q_0 / C_{tube}$$

Charge distribution

equivalent circuit



Impedance mismatch: $\Delta V_{reflect} = (Z-330)/(330+Z) \times \Delta V$ $\Delta V_{transmit} = 660/(330+Z) \times \Delta V$

Current divider: $\Delta V' = 330/(330+50) \times \Delta V_{transmit} = 0.87 \times \Delta V_{transmit}$

combined result: $\Delta V' = 330/(330+50) \times 660/(330+Z) \times Q_0/C_{tube}$

A few numbers

configuration number	R	$2\times r_w$	L	C_L	L_L	C_{tube}	Z	ΔV (10^5 gain)	ΔV' (10^5 gain)
1	0.4 cm	10 µm	0.4 m	8.37 pF/m	1.34 µH/m	3.35pF	400 Ω	**4.78 mV**	**3.75 mV**
3	0.4 cm	15 µm	0.4 m	8.91 pF/m	1.26 µH/m	3.56 pF	376 Ω	**4.49 mV**	**3.65 mV**
5 - 7	0.4 cm	20 µm	0.4 m	9.33 pF/m	1.20 µH/m	3.73 pF	359 Ω	**4.29 mV**	**3.57 mV**
9 - 11	0.4 cm	25 µm	0.4 m	9.69 pF/m	1.15 µH/m	3.88 pF	344 Ω	**4.12 mV**	**3.50 mV**
13	0.9 cm	20 µm	0.3 m	8.22 pF/m	1.36 µH/m	2.47 pF	407 Ω	**6.48 mV**	**5.04 mV**
15	0.9 cm	25 µm	0.3 m	8.50 pF/m	1.32 µH/m	2.55 pF	394 Ω	**6.27 mV**	**4.96 mV**
17	0.9 cm	40 µm	0.3 m	9.15 pF/m	1.22 µH/m	2.75 pF	365 Ω	**5.82 mV**	**4.80 mV**
19	1.4 cm	25 µm	0.3 m	7.96 pF/m	0.94 µH/m	2.39 pF	344 Ω	**6.69 mV**	**5.02 mV**
21	1.4 cm	40 µm	0.3 m	8.54 pF/m	0.85 µH/m	2.56 pF	315 Ω	**6.25 mV**	**5.30 mV**

ΔV'(10^5 gain) = single electron pulse height at readout, for an HV producing a reduced electric field: E/P, corresponding to a gas gain of 10^5

Determining HV (10^5 gain)

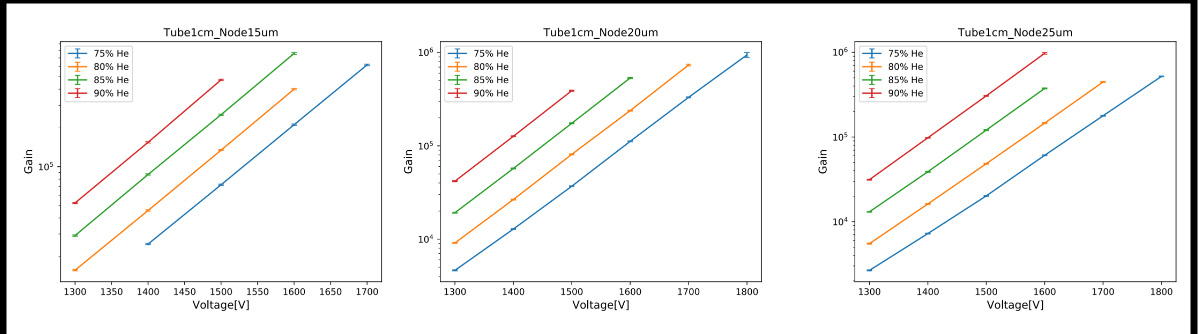
Garfield run on different configurations by Shuiting for P = 760 torr, T = 20°C

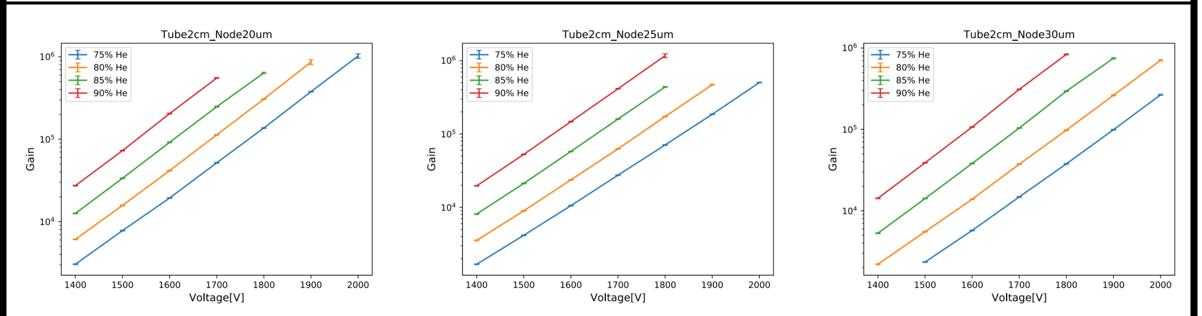
|

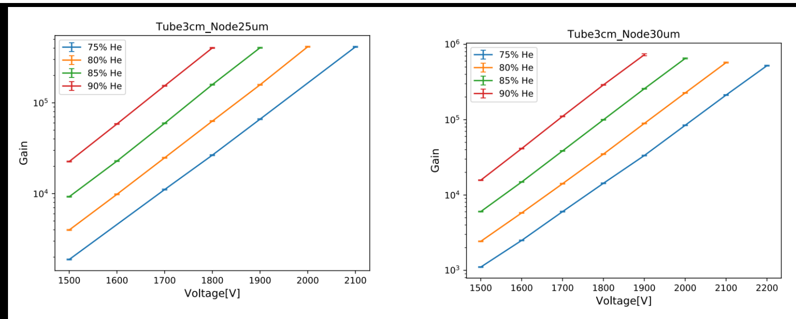
renormalized for P = 725 torr and extrapolated for 10 and 40 μm wires

|

and readjusted by observing average maximum pulse height







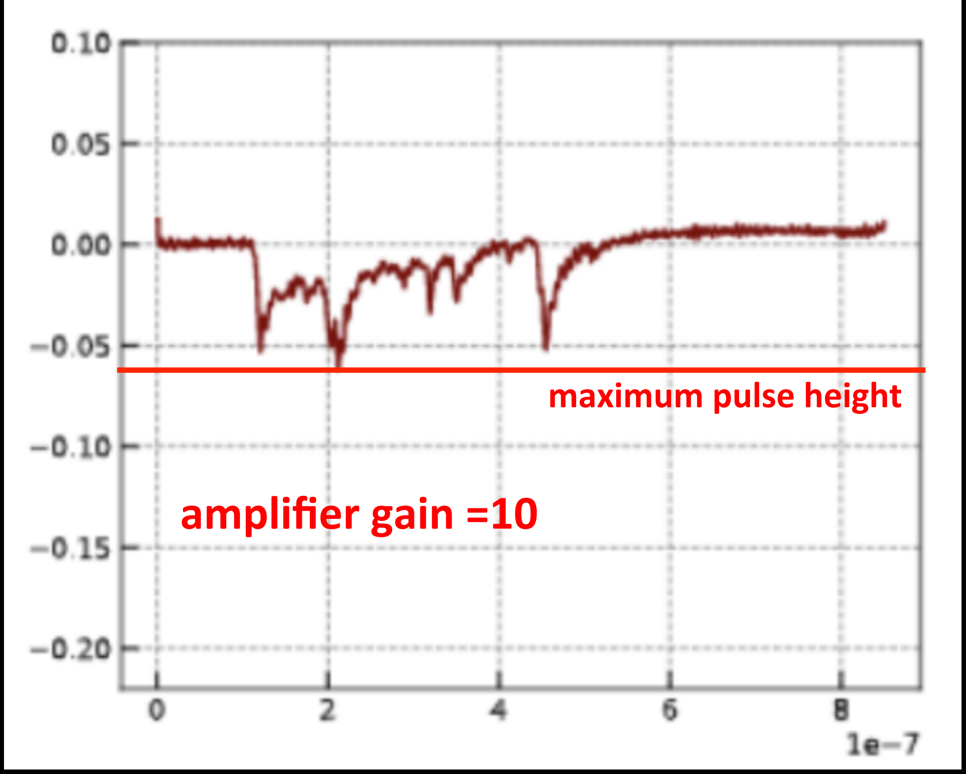
HV table (He/iC_4H_{10} = 90/10, P = 725 torr)

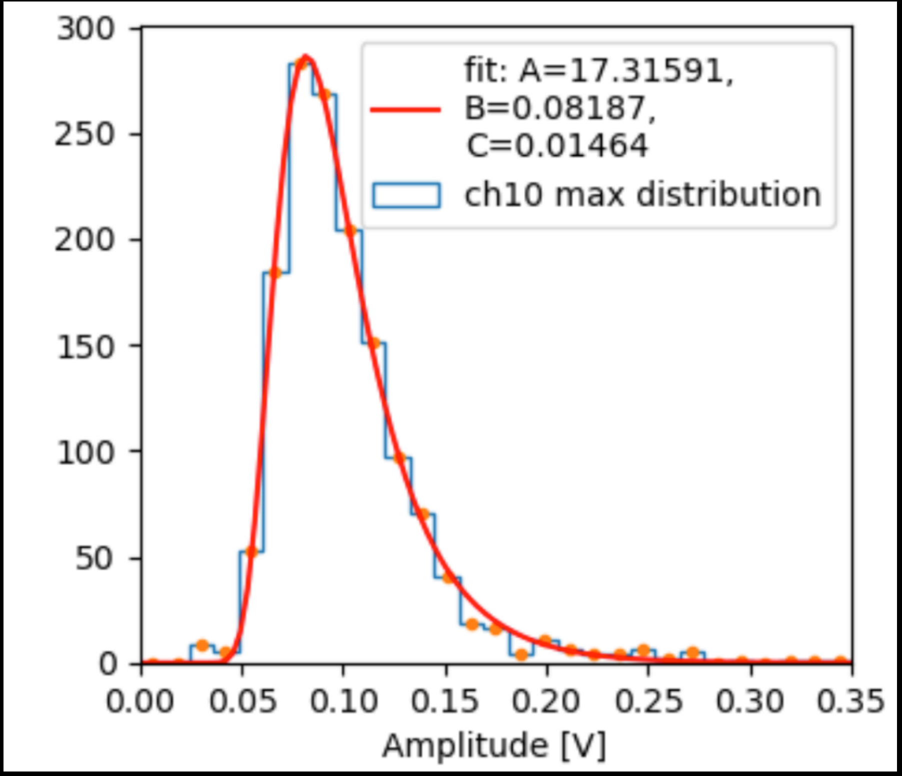
1st attempt (from Garfield)

configuration number	HV	relative gas gain (ΔV)
1	1230 V	2.6
3	1255 V	1.4
5 - 7	1305 V	1.3
9 - 11	1330 V	0. ...
13	1470 V	0.6
15	1545 V	0.9
17	1670 V	0. ...
19	1620 V	0. ...
21	1765 V	0. ...

Average maximum pulse height

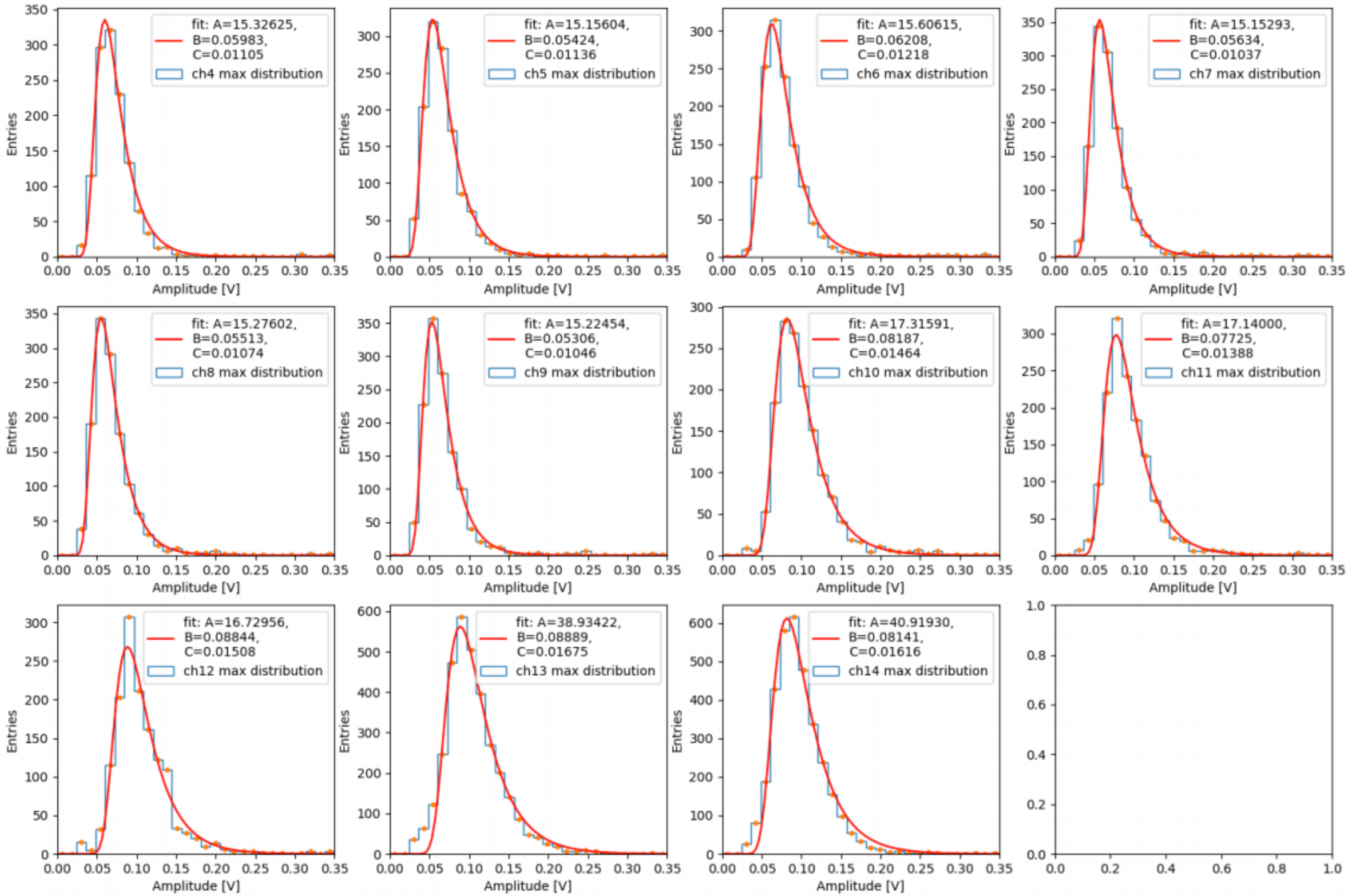
Assume that the average maximum pulse height ≅ single electron pulse height and fit to a Landau distribution by taking into account the amplifier gain (x 10)





Average maximum pulse height

- Max amplitude/ per channel distribution, fitted with a landau - **11Nov_45angle_HVnominal_1p2GSPS_10k_LANDAU**
  - https://github.com/clacaputo/drifttubes_analysis/blob/main/landau_fit.py
- More of these plots: https://cernbox.cern.ch/index.php/s/yjoJLkgUbPC1ELG



HV table (He/iC_4H_{10} = 90/10, P = 725 torr)

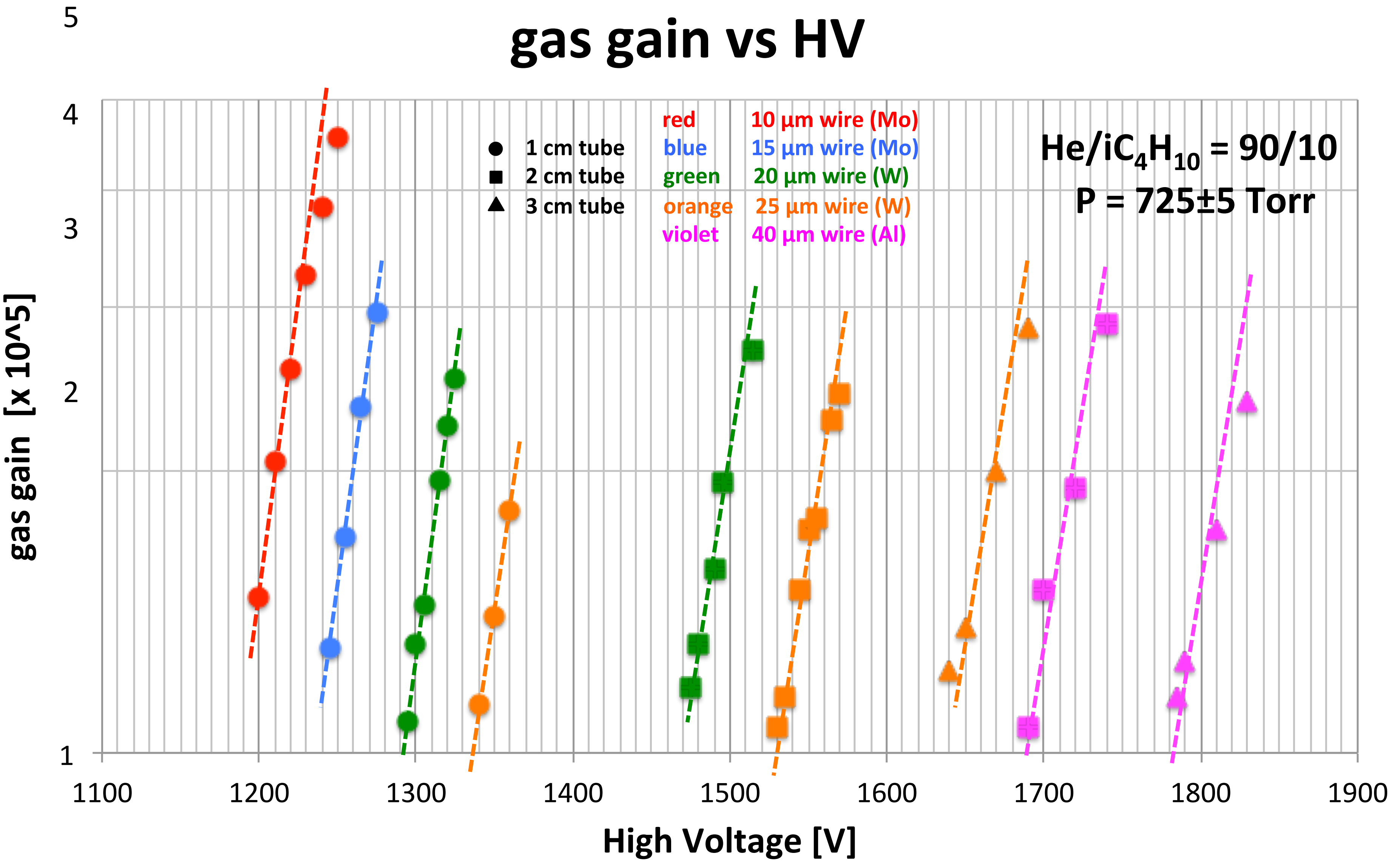
1[st] attempt (from Garfield)

configuration number	HV	relative gas gain (ΔV)
1	1230 V	2.6
3	1255 V	1.4
5 - 7	1305 V	1.3
9 - 11	1330 V	0. ...
13	1470 V	0.6
15	1545 V	0.9
17	1670 V	0. ...
19	1620 V	0. ...
21	1765 V	0. ...

2[nd] attempt (corrections to Garfield)

HV	relative gas gain (ΔV)	actual gas gain (ΔV')
1230 V – 30 V	1.2	1.47
1255 V – 10 V	1.1	1.29
1305 V – 5 V	1.1	1.31
1330 V + 10 V	1.1	1.27
1470 V + 25 V	1.1	1.57
1545 V + 5 V	1.0	1.49
1670 V + 50 V	1.2	1.92
1620 V + 50 V	1.1	2.00
1765 V + 45 V	1.0	1.73

Gas gain



gas gain vs HV
gas gain [x 10^5]
High Voltage [V]
1 cm tube
2 cm tube
3 cm tube
red 10 µm wire (Mo)
blue 15 µm wire (Mo)
green 20 µm wire (W)
orange 25 µm wire (W)
violet 40 µm wire (Al)
He/iC4H10 = 90/10
P = 725±5 Torr
1
2
3
4
5
1100
1200
1300
1400
1500
1600
1700
1800
1900

Cluster counting preliminary algorithms

- Novosibirsk algorithm
- Louvain algorithm
- Beijing algorithm
- Lecce algorithm

Please, see for details:

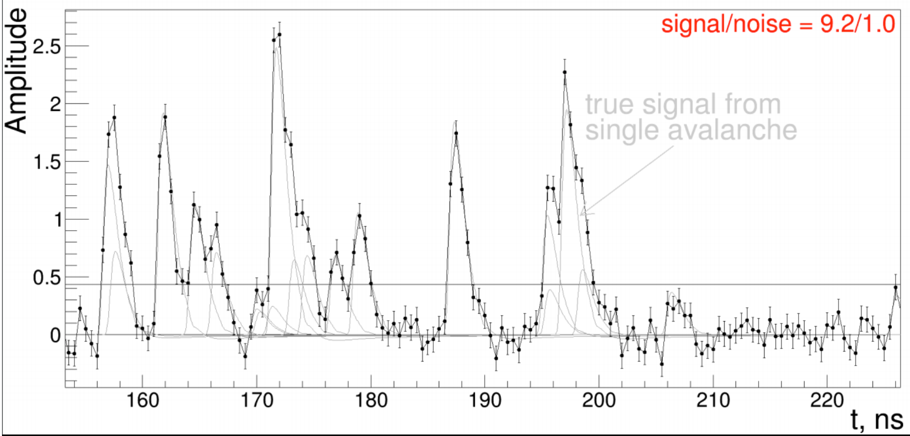
https://indico.ihep.ac.cn/event/15640/

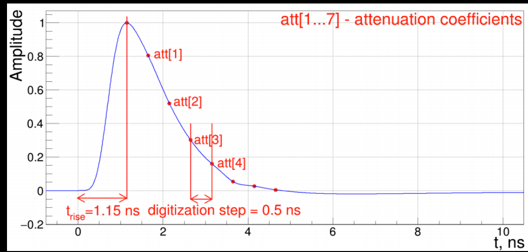
Novosibirsk algorithm (courtesy of Slava Ivanov)

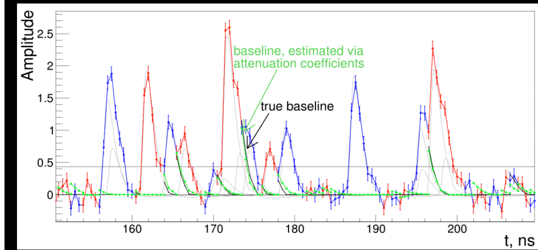
- Based on dynamic estimation of the baseline level

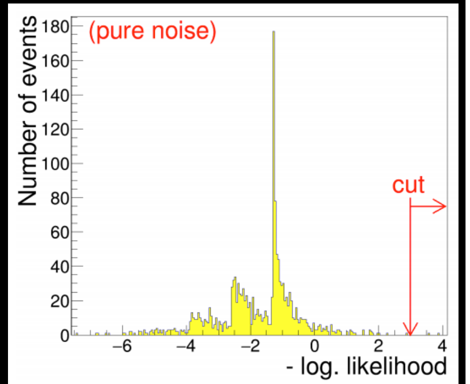
Typical waveform (muons)

- Digitization time step is 0.5 ns (freq. = 2 GHz), signal/noise = 9.2/1.0 (according to V.M. Aulchenko)





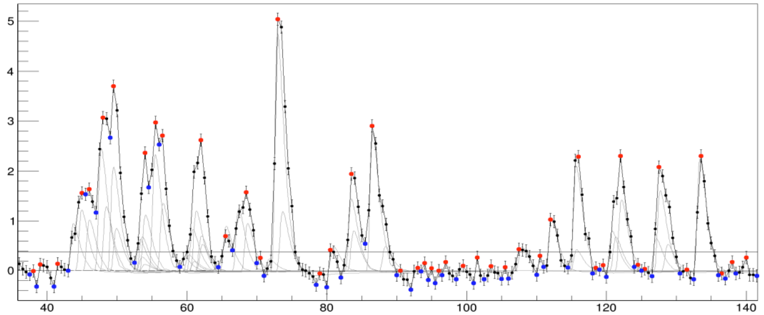




$$q = \sum_{i=1}^{N} \frac{(a_i - b_i)^2}{\sigma_{noise}^2} + N \cdot \ln(2\pi\sigma_{noise}^2)$$

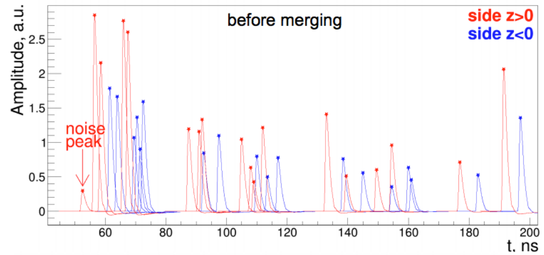
The idea of my peak finding approach

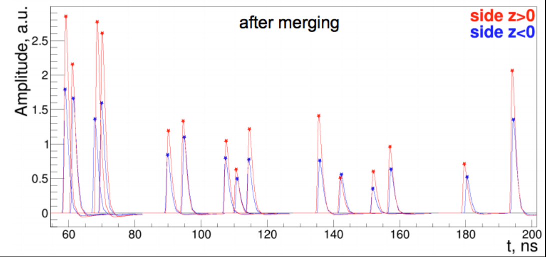
- The waveform contains local minimums and local maximums
- **Each** waveform segment "loc. min. – loc. max. – loc. min" is considered as *peak candidate*
- Peak candidate is identified as *real peak* if it satisfies a *quality criterion*. Currently one peak candidate can give only one real peak
- To calculate the peak quality correctly, one should account for the **baseline** shift, caused by the previous peaks
- Thus, for each peak candidate we should estimate the **baseline** it resides on ("running baseline")



Merging the waveforms from the wire ends

- After the peak finding we merge the waveforms from the wire ends
- This is done by finding the time shift between waveforms via the maximization of their cross-correlation function
- After the alignment of signals in time we find the pairs of peaks with maximum overlap (= dot product) and "merge" them
- The remaining unpaired peaks are considered to be wrong or noise peaks





Novosibirsk algorithm (courtesy of Slava Ivanov)

Peaks clusterization algorithm

- The decisive point for the cluster counting and timing is the *possibility* of peaks clusterization
- We consider each merged peak as possible cluster and assign the gaussian to it. The time μ_i, $i = 1, \ldots, N_{clusters}$, of each gaussian is equal to the corresponding peak time t_j, $j = 1, \ldots, N_{peaks}$ and sigma equals to the time spread due to diffusion, $\sigma_i = \sigma_{diff}(\mu_i)$). Thus, our model to describe the set of peak times is a gaussian mixture model
- Using the Expectation Maximization (EM) algorithm we iteratively recalculate (until convergence) the positions μ_i of gaussians and their weights w_i using the probabilities $p_{i,j}$ that the peak $\#j$ was produced by the cluster (gaussian) $\#i$:

$$p_{i,j} = \frac{w_i}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(t_j - \mu_i)^2}{2\sigma_i^2}\right)$$

→ total "responsibility" of cluster $\#i$:

$$m_i = \sum_{j=1}^{N_{peaks}} p_{i,j}$$

→ new weights:

$$w_i = \frac{m_i}{\sum_{i=1}^{N_{clusters}} m_i}$$

→ new cluster times:

$$\mu_i = \frac{1}{m_i} \sum_{j=1}^{N_{peaks}} p_{i,j} t_j$$

→ next iteration

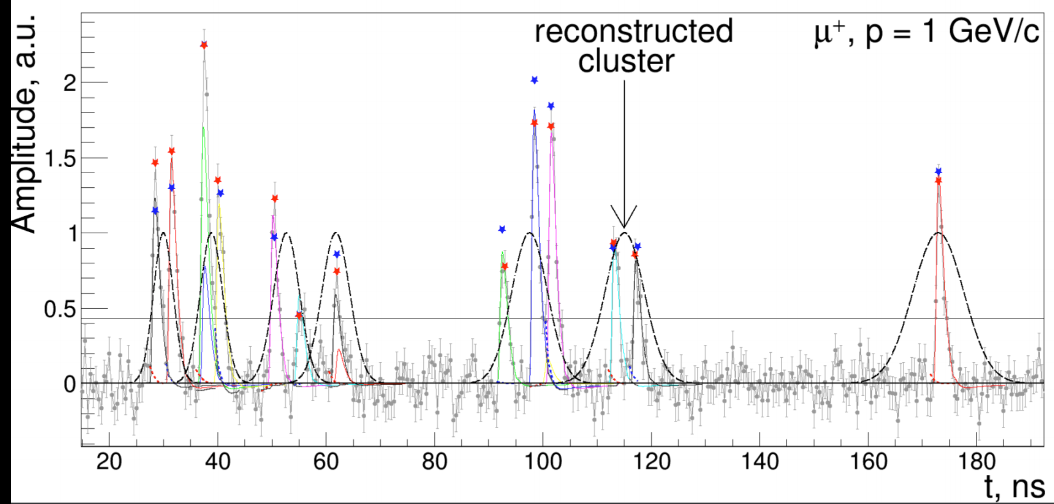
- After the convergence was reached, we usually find that some gaussians are "stucked together" (~ "clusterization happened"), some other gaussians have almost zero weights. These effects are the signs of too large model complexity
- To simplify the model, we should choose the cluster to be removed. To make a best choice, we try to remove each one of them and delegate its "responsibility" m_i to its *nearest neighbor*, and calculate the resulting change of the log-likelihood of data description L. We remove the cluster giving the smallest likelihood loss and start clusterization from the very beginning (without removed gauss)
- To find the optimal number of gaussians we use the Akaike information criterion (AIC), which finds the balance between the likelihood of data description and the model complexity

$$L = \sum_{j=1}^{N_{peaks}} \ln\left(\sum_{i=1}^{N_{clusters}} p_{i,j}\right) \qquad AIC = 2N_{clusters} - 2L$$

- Current version of algorithm shows poor results for the cluster counting (~50% efficiency for m.i.ps). Improvements are necessary
- Ideas: use the described algorithm for *cluster timing*, and try to estimate the most probable number of clusters within each gaussian using the peak amplitudes (= separate cluster *timing* and cluster *counting* tasks); use the information from all the cells on the track (e.g. the distribution of time separations between peaks);
- **I have a standalone code (only C++ and ROOT) to generate the set of waveforms (~"track"), to find the peaks and to clusterize them. I can share it if anybody would like to try his own ideas for the clusterization algorithm. The joint efforts are necessary, any ideas are welcome!**

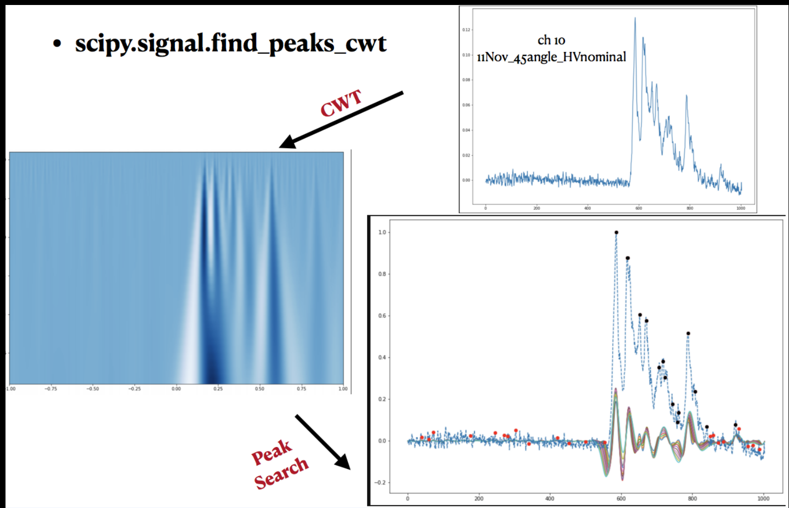
Peaks clusterization algorithm

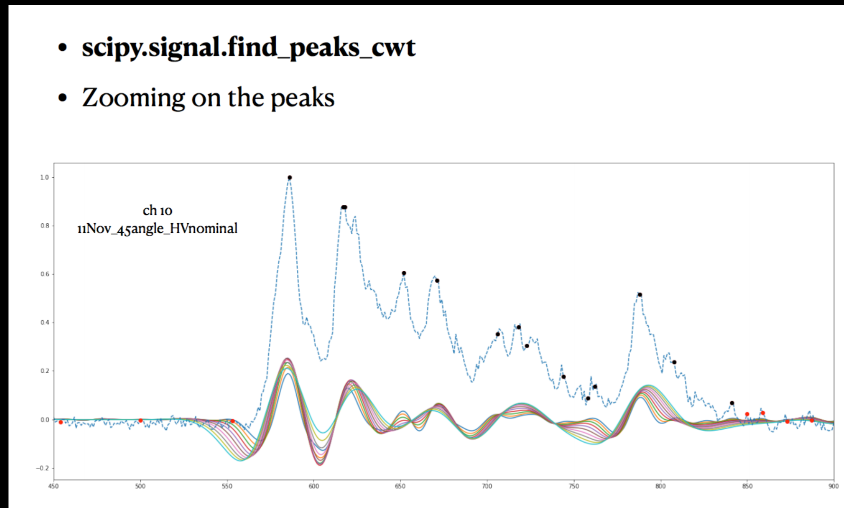
- Peaks clusterization for muons:



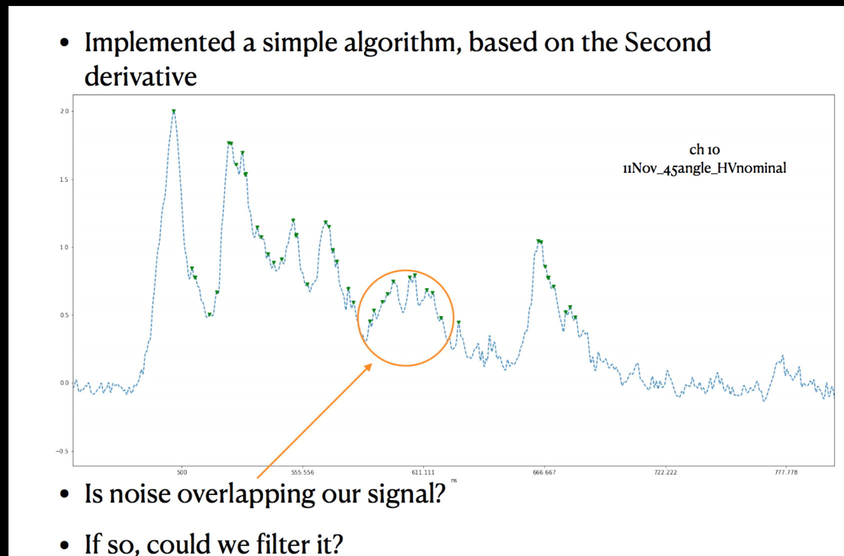
Louvain algorithm (courtesy of Claudio Caputo)

- Based on continuous wavelet transform on data, using the wavelet function.
- Very preliminary





Attempt at using
the second derivative
by implementing
the Satisky-Gorlay
filter

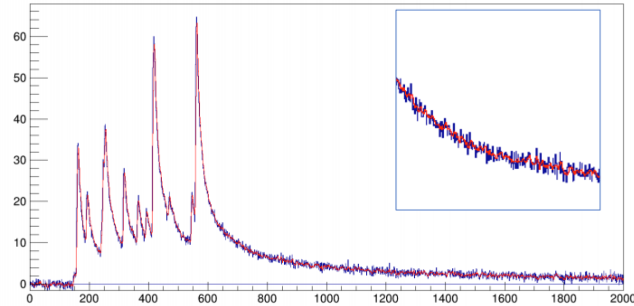


Beijing algorithm (courtesy of Guang Zhao)

- Based on moving average smoothing and on first and second derivatives

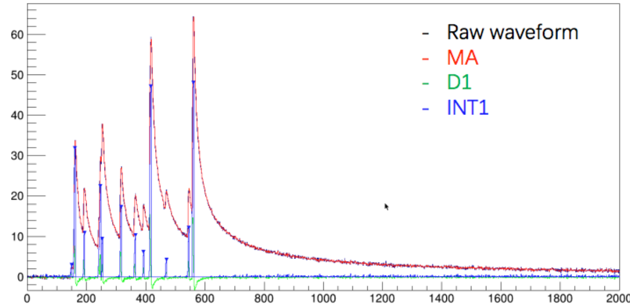
Peak finding algorithm

- **Noise reduction**
  - Filter out high frequency noises in the waveforms in order to improve the S/N ratio
  - Moving average: $\mathbf{MA[i]} = \frac{1}{M} \times \sum_{k=0}^{K<M} \boldsymbol{S[i-k]}$



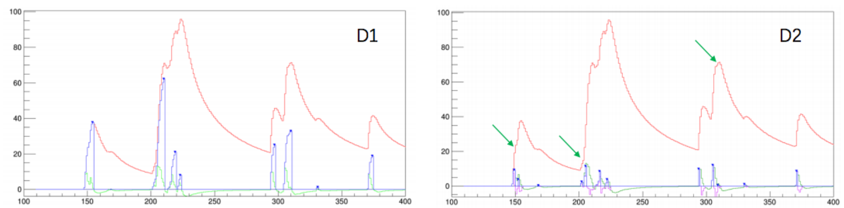
Cluster counting algorithm (II)

- **First derivative and integration**
  - First derivative (**D1**): D1[i] = MA[i] – MA[i – 1]
  - Integration on the positive D1 (**INT1**): recover the rising edge and removing falling edge
  - Hit detection: Passing a threshold

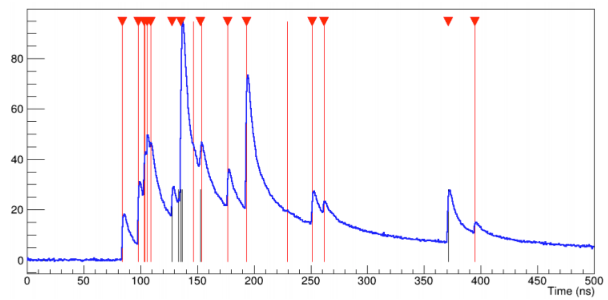


Cluster counting algorithm (III)

- **Second derivative and integration:** **recover pile-up peaks on the rising edge**
  - Second derivative (**D2**): D2[i] = INT1[i] – INT1[i – 1]
  - Integration on the positive D2 (**INT2**)
  - Hit detection: Passing a threshold



Waveform with "MC truth" times



- Red line: primaries
- Black line: secondaries
- Triangle: detected

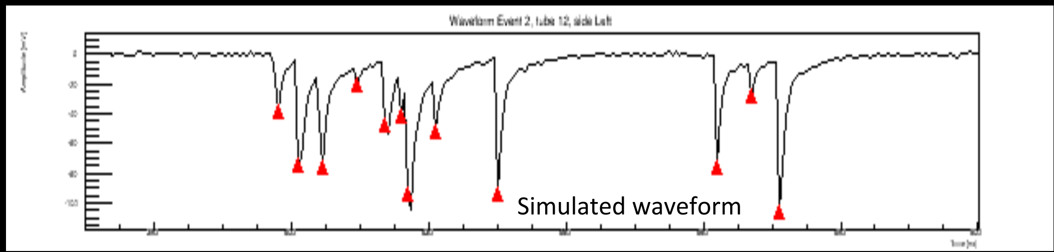
Lecce algorithm (courtesy of Federica Cuna)

ClusterFinder approach

- Based on moving average smoothing and on first and second derivatives

Δb being the number of bins (signal rise time) over which the average value of f is calculated

A peak (assumed to be an ionization electron) is found when *Δf*, *f'* and *f''* are above
a threshold level, defined according to the r.m.s. noise of the signal function *f*,
and when the time difference with a contiguous peak is larger than the time bin resolution.



Cylindrical cell geometry

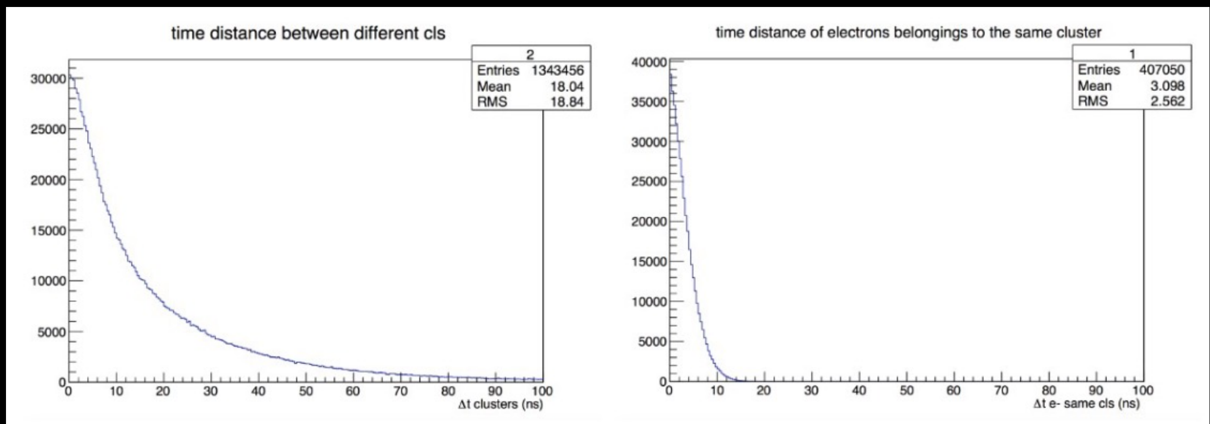
$He\text{-}iC_4H_{10}$

Gas gain 4x10^5

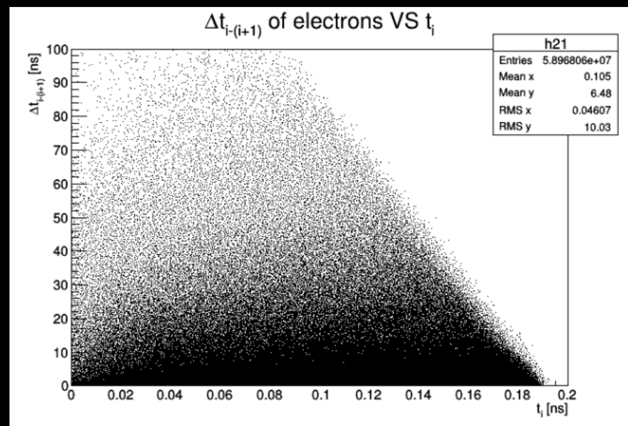
FE gain 10

How to convert found peaks in clusters?

Look to the time difference between electrons belonging to different clusters and those to the same cluster



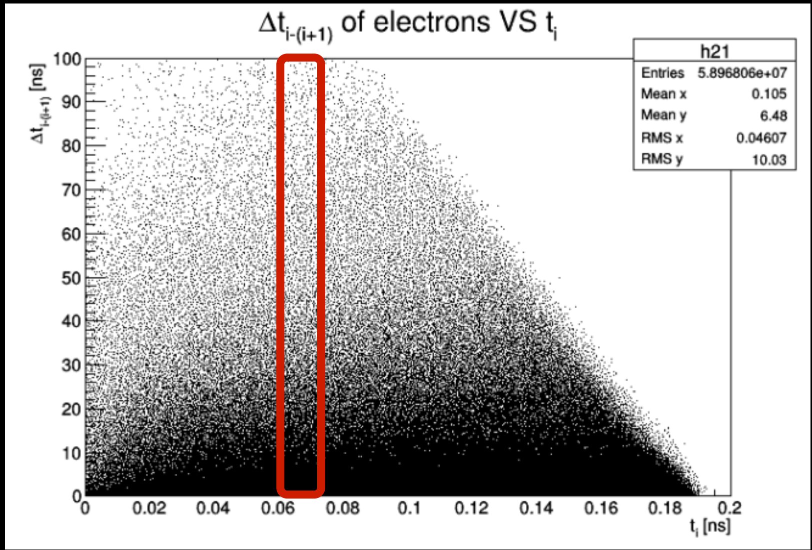
Event by event plot the difference
t_i - t_{i+1} Vs t_i

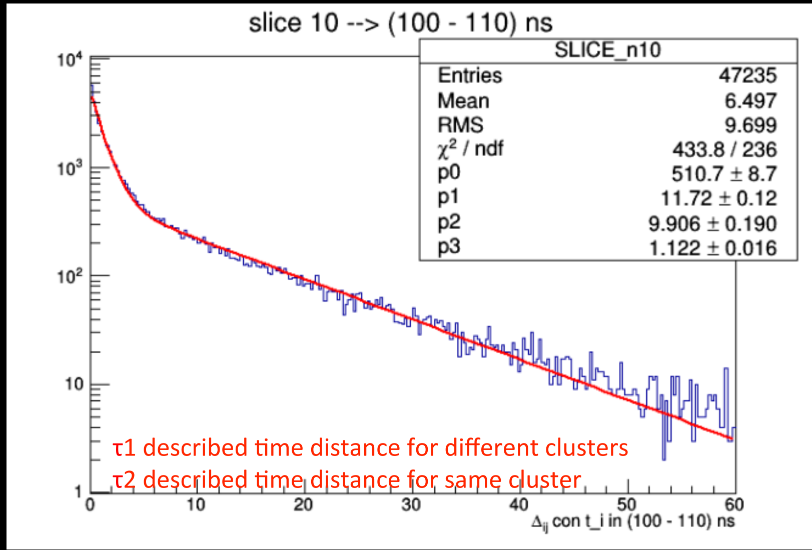


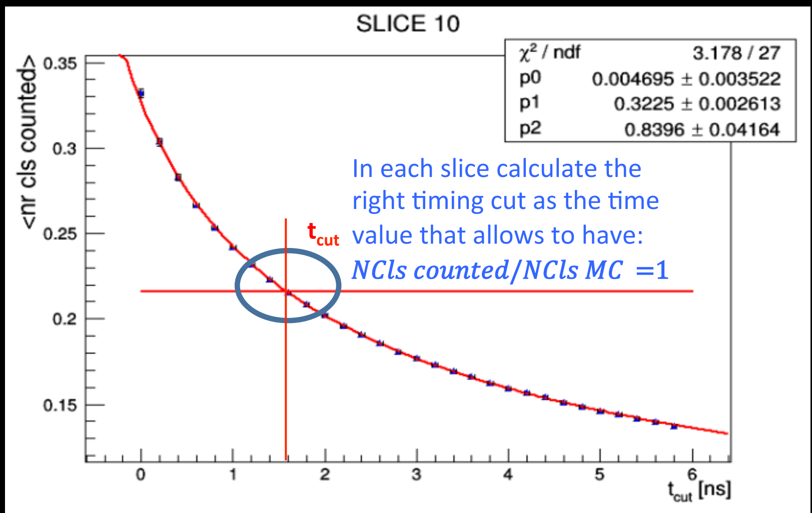
Lecce algorithm (courtesy of Federica Cuna)

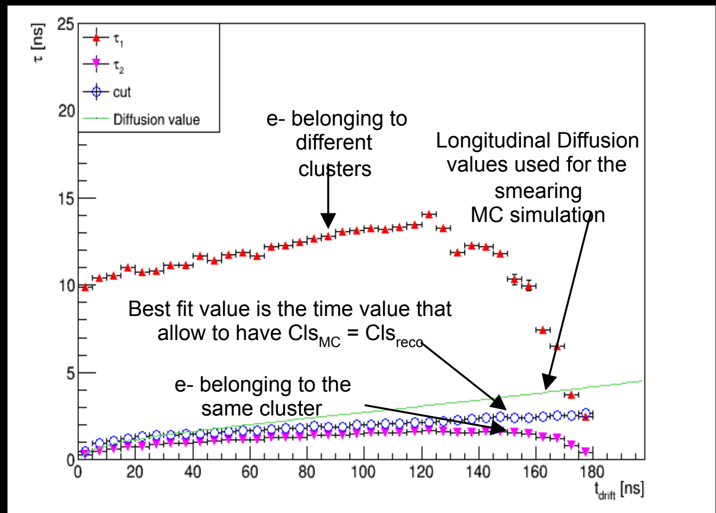
Cluster association approach

Cut into slices along x – axis and fit with a double exponential function





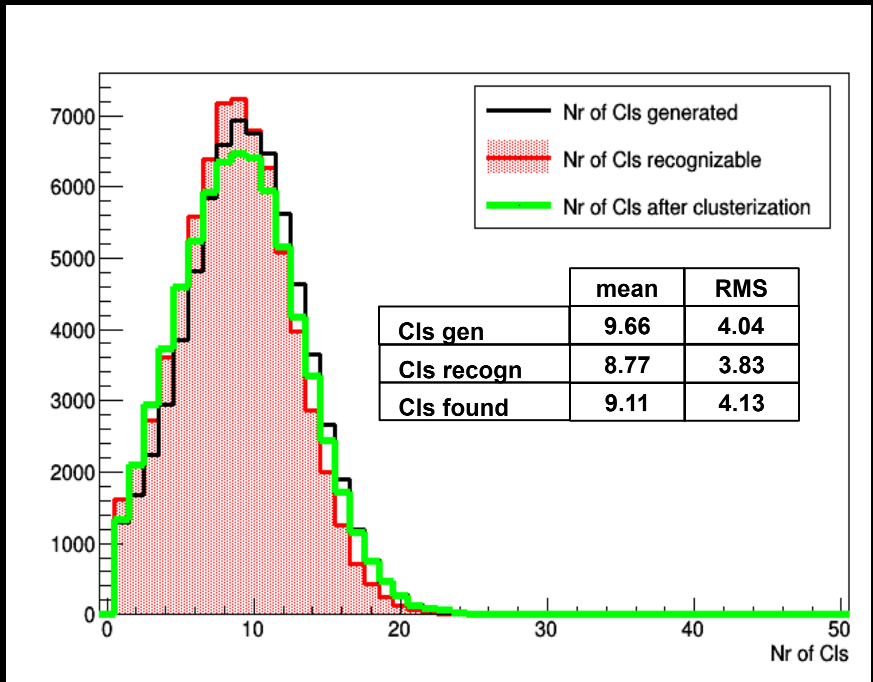




The association of electrons in clusters is based on the time difference between consecutive electrons. Electrons belonging to same cluster are separated by time differences which are compatible with single electron diffusion.

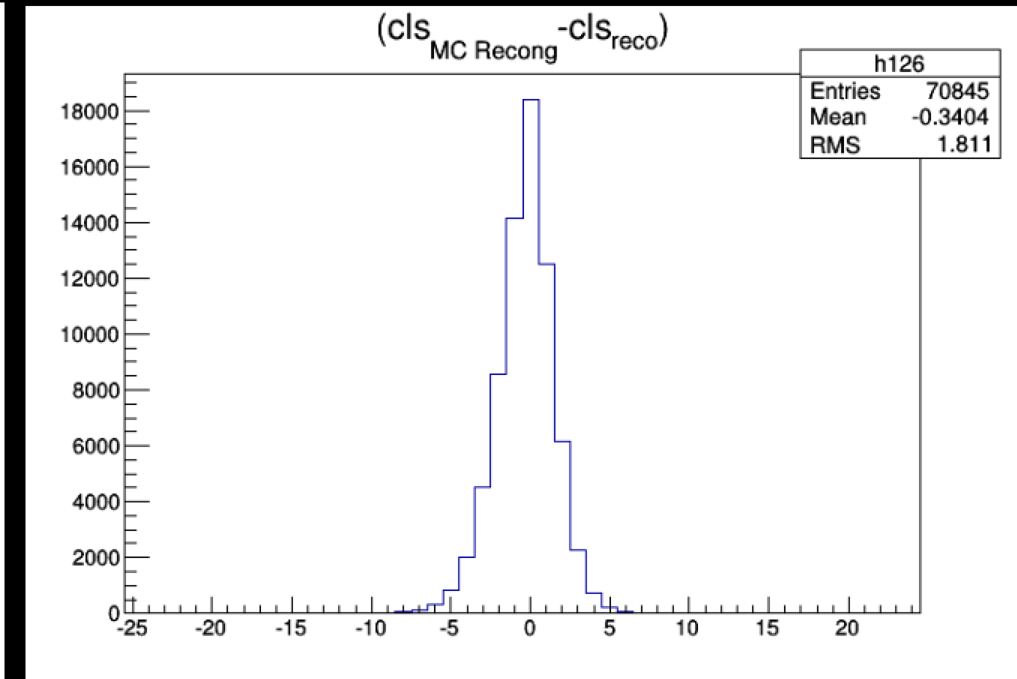
Lecce algorithm (courtesy of Federica Cuna)

Performance



$\varepsilon = Cls\ found/Cls\ recogn = 103.9\%$

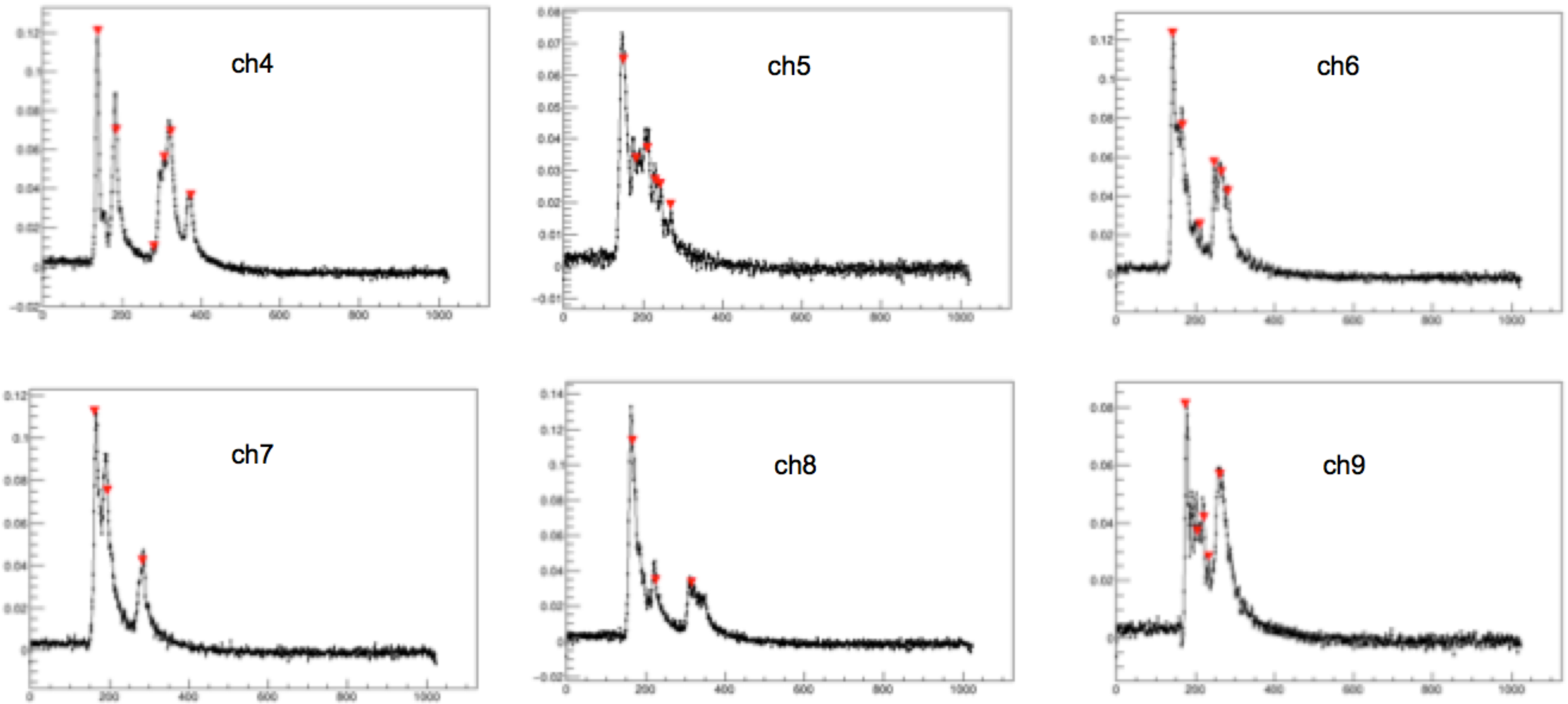
$\varepsilon = Cls\ found/Cls\ gen = 94.3\%$



Time difference between MC generated cluster and reconstructed cluster

Lecce algorithm (courtesy of Federica Cuna)

Algorithm applied to the beam test data



45°
Nominal HV +20

1cm drift tubes

All the parameters need to be tuned

NO attempt made yet to tune the search parameters