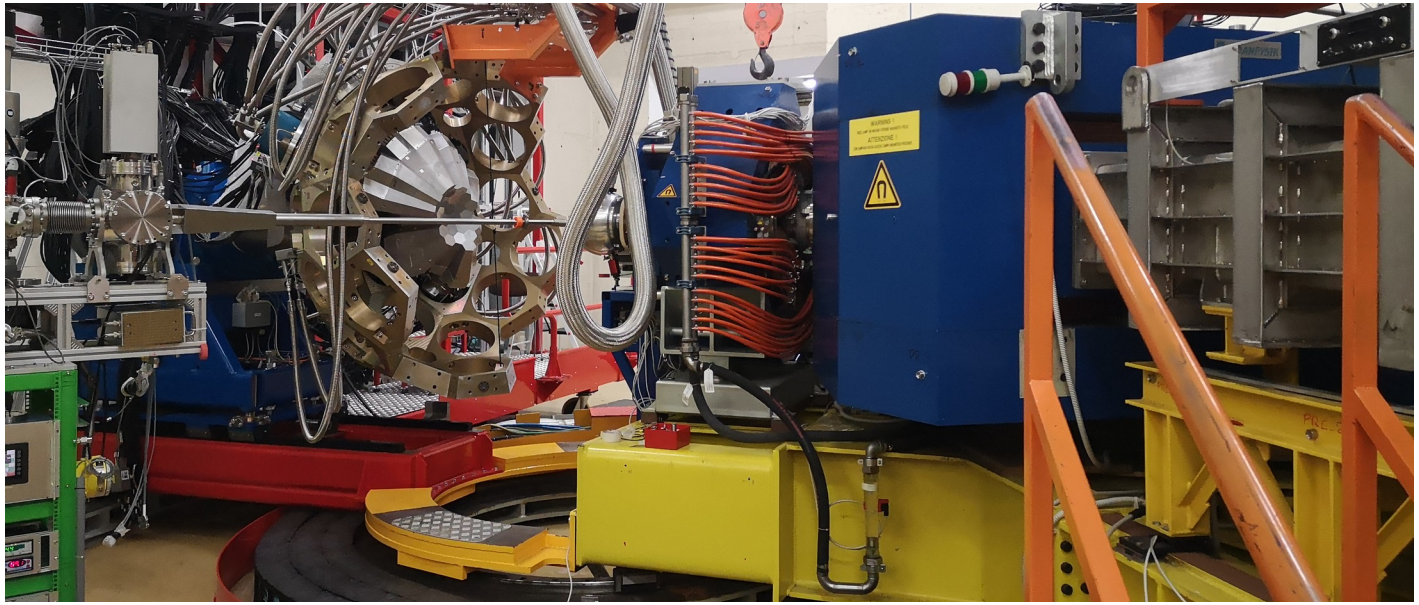
AGATA analysis workshop

September 2023

The PRISMA magnetic spectrometer: analysis and data-processing



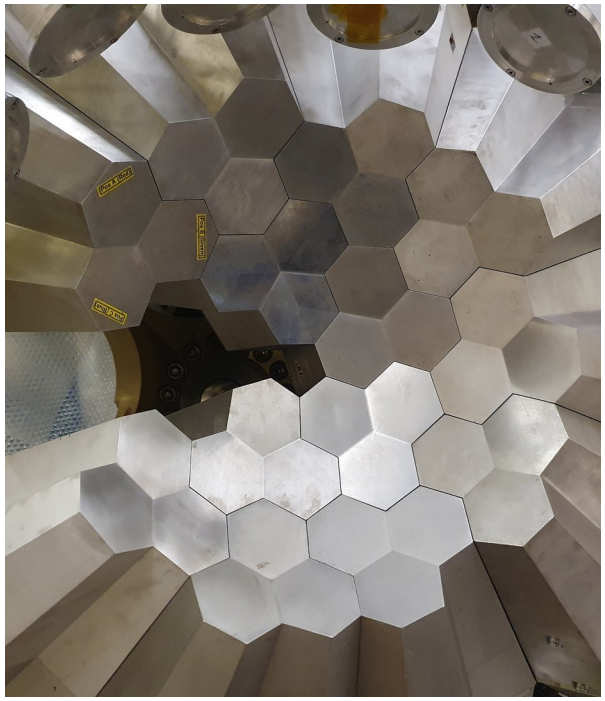
Speaker: Elia Pilotto

Overview

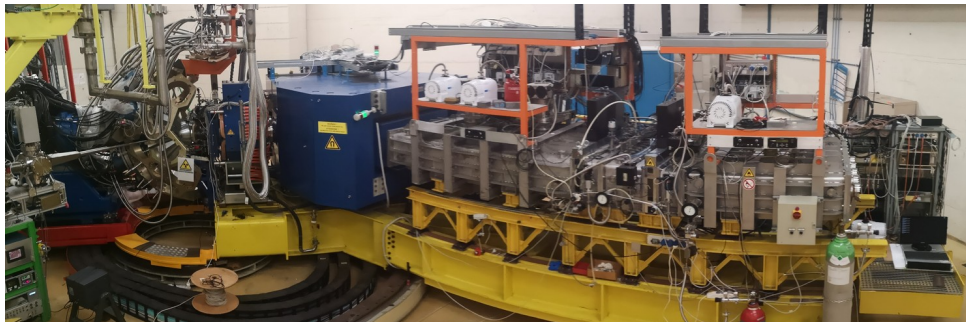
INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

The main objective of this presentation is to provide a full and in-depth explanation on how to perform the analysis of an experiment with the Prisma magnetic spectrometer.

- **The analysis software:**
  - structure, installation, how to run, output format, configuration
- **Steps of the analysis:**
  - Calibration of MCP, PPAC and IC
  - Z identification
  - Trajectory reconstruction
  - Mass identication
- **Tools to help in the process:**
  - PrismaOnlinePackage
  - agataselector



Picture taken by R. M. Pérez-Vidal



The PRISMA magnetic spectrometer

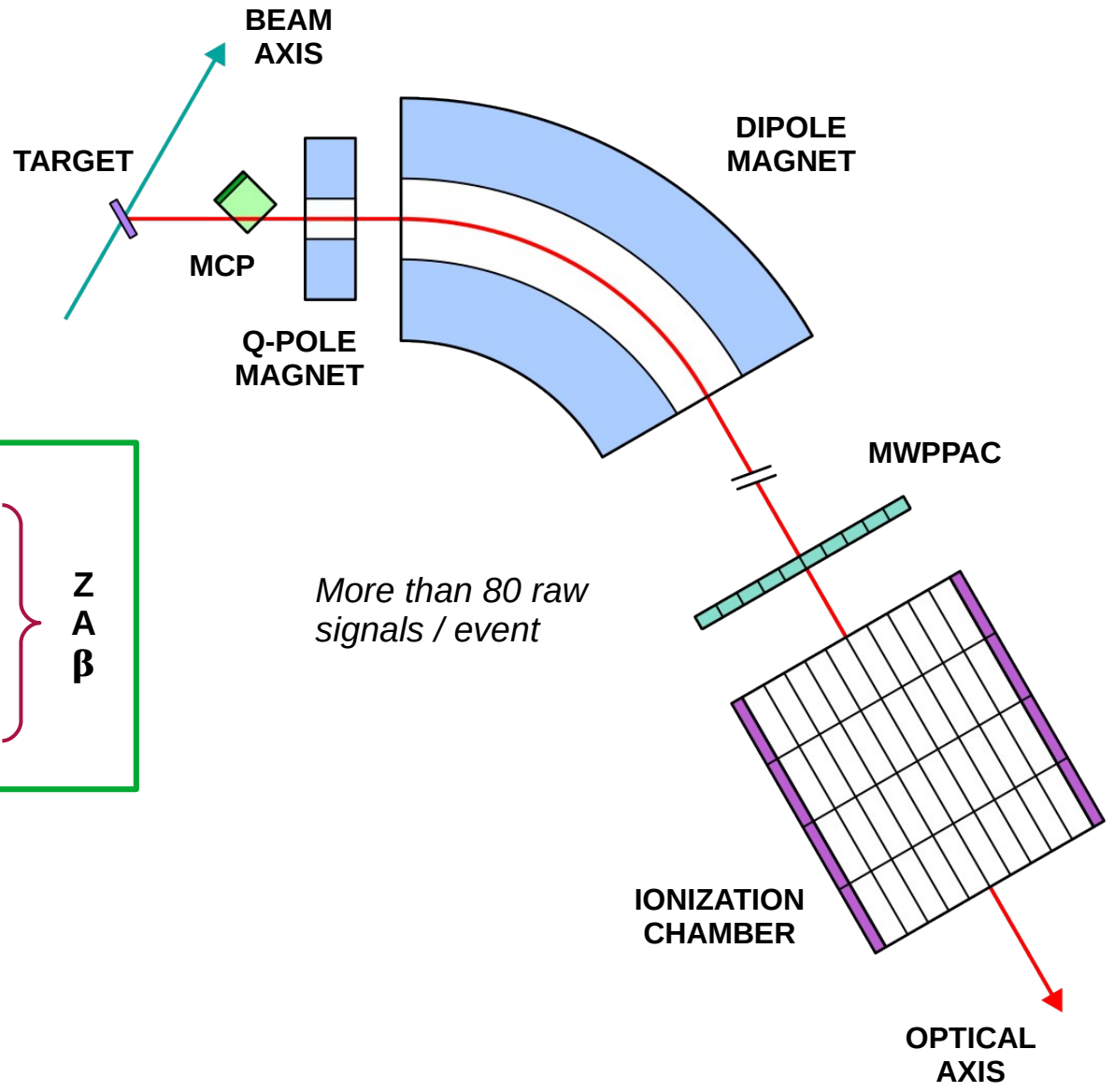
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Main objective:

Identification of incoming nuclei in terms of **nuclear charge**, **mass** and **velocity vector**.

Working range:

3-10 MeV/u, 20 < A < 150



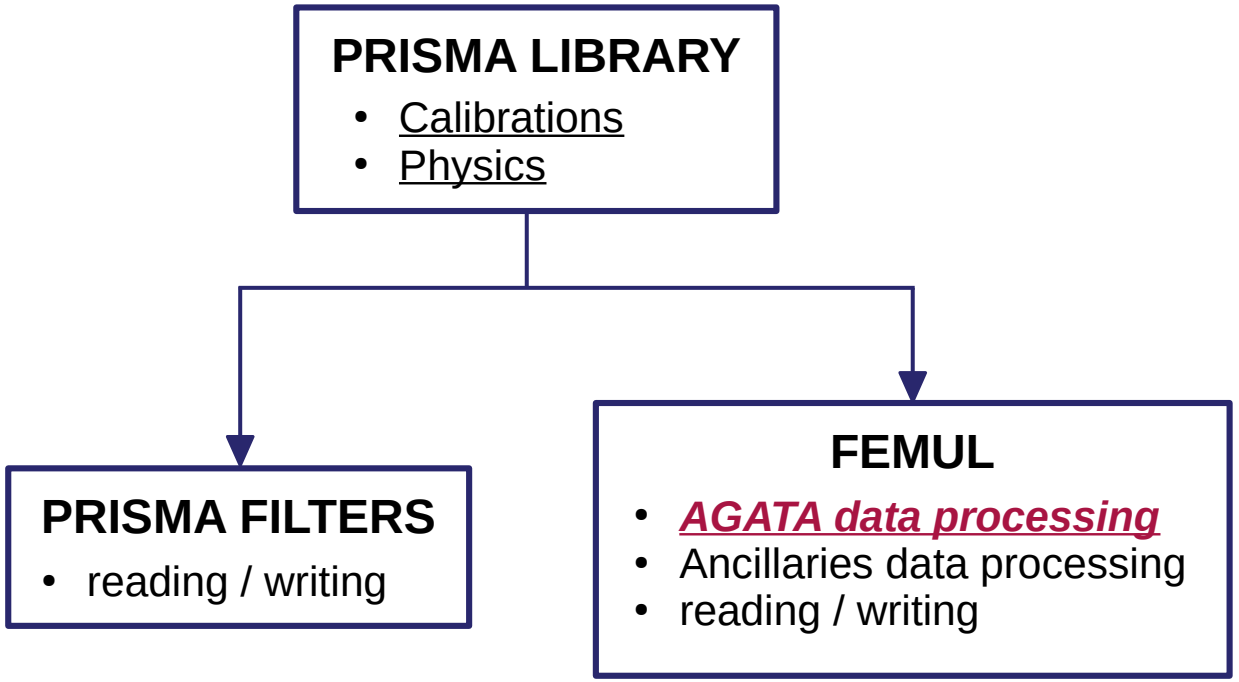
Entrance Detector	(x,y) time	Bρ Length ToF	Z A β
MWPPAC	(x,y) time		
Ionization Chamber	Energy Loss	ΔE - E	

More than 80 raw signals / event

Typical performance

- Z resolution: ~ 1/60
- A resolution: ~ 1/300

The analysis software



Acquisition time

~10 min *~10 hr* *~10 hr*

PrismaFilters installation

```
YOUR_PATH=$PWD

git clone https://baltig.infn.it/prisma/prisma_library.git
cd prisma_library
mkdir build lib
cd build
cmake .. -DCMAKE_INSTALL_PREFIX=$YOUR_PATH/prisma_library/install
cmake --build . --target install

export PRISMA_DIR=$YOUR_PATH/prisma_library/install
export LD_LIBRARY_PATH=$PRISMA_DIR/lib:$LD_LIBRARY_PATH

cd $YOUR_PATH
git clone https://baltig.infn.it/prisma/prismafilters.git
cd prismafilters
mkdir build
cd build
cmake .. -DROOT_OUTPUT=ON
make -j4

cd $YOUR_PATH
```

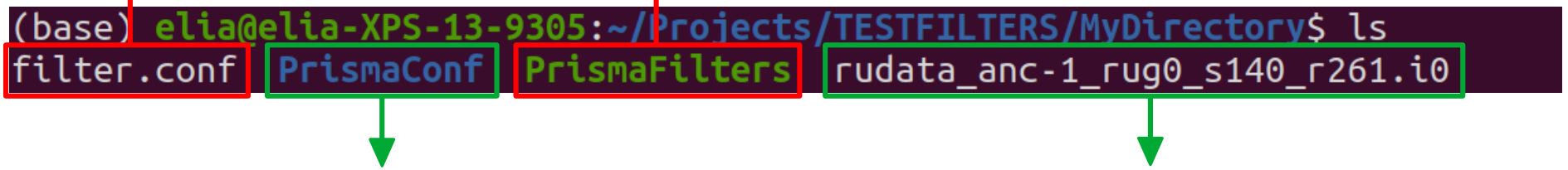
Check also the manual (pdf format):

$YOUR_PATH/prismafilters/PRISMA_manual/ManualPRISMARoot.pdf

Basic directory setup

Configuration file

PrismaFilters executable (or link to)



Configuration folder

*Take from **PrismaOnlinePackage***

Raw data

- Readout Unit (RU) format
- Builder Unit (BU) format

PrismaFilters femul

Run with:

```
./PrismaFilters -if rudata_anc-1_rug0_s140_r261.i0 -cfg filter.conf
```

Also get help with:

```
./PrismaFilters -h
```

Event structure

After running:

```
(base) elia@elia-XPS-13-9305:~/Projects/TESTFILTERS/MyDirectory$ ls
filter.conf  PrismaConf  PrismaFilters  rudata_anc-1_rug0_s140_r261.i0  Tree_0000.root
```

Output file

Contains a single ROOT Tree called PrismaTree

```
******************************************************************************
*Tree    :PrismaTree: PrismaTree                                             *
*Entries :   108680 : Total =        61384900 bytes  File  Size =   20614936 *
*        :          : Tree compression factor =   2.98                       *
******************************************************************************
*Br    0 :TSPrisma  : TSPrisma/l                                             *
*Entries :   108680 : Total  Size=     872593 bytes  File Size  =     503090 *
*Baskets :       28 : Basket Size=      32000 bytes  Compression=   1.73     *
*.............................................................................*
*Br    1 :Raw       : MCP_raw[3]/s:PPAC_Xleft_raw[10]/s:                     *
*         | PPAC_Xright_raw[10]/s:PPAC_Cathode_raw[10]/s:PPAC_Y_raw[2]/s:    *
*         | TOF_raw[10]/s:IC_A_raw[10]/s:IC_B_raw[10]/s:IC_C_raw[10]/s:      *
*         | IC_D_raw[10]/s:IC_A_Drift_raw[10]/s:IC_B_Drift_raw[10]/s:        *
*         | IC_C_Drift_raw[10]/s:Side_A_raw[2]/s:Side_B_raw[2]/s:            *
*         | Side_C_raw[2]/s:Side_D_raw[2]/s:Monitors_raw[2]:TAC_LT_VTS/s     *
*Entries :   108680 : Total  Size=   27472373 bytes  File Size  =   12467244 *
*Baskets :      863 : Basket Size=      32000 bytes  Compression=   2.20     *
*.............................................................................*
*Br    2 :Analyzed  : MONITOR_1/F:MONITOR_2/F:MCP_X/F:MCP_Y/F:MCP_Q/F:       *
*         | MCP_Theta/F:MCP_Phi/F:X_FP/F:Y_FP/F:TOF/F:IC_Pads[40]/F:IC_E/F:  *
*         | IC_DE_A/F:IC_DE_AB/F:IC_RANGE/F:IC_Drift_A/F:IC_Drift_B/F:       *
*         | IC_Drift_C/F:Theta/F:Phi/F:Beta/F:Length/F:Radius/F:RBeta/F:     *
*         | A_over_q_uncal/F:A_over_q/F:Mass/F:Qvalue/F:Theta_BP/F:Phi_BP/F: *
*         | Beta_BP/F:TAC_LT_LTS/F:IC_col_a/b:IC_col_b/b:IC_col_c/b:         *
*         | IC_col_d/b:IC_a_numpads/b:IC_b_numpads/b:IC_c_numpads/b:         *
*         | IC_d_numpads/b:Z_Nr/b:Q_Nr/b:A_Nr/b:mcp_ok/b:tof_ok/b:traj_ok/b: *
*         | side_ok/b:ic_ok/b:z_ok/b:q_ok/b:a_ok/b                           *
*Entries :   108680 : Total  Size=   33039277 bytes  File Size  =    7623124 *
*Baskets :     1036 : Basket Size=      32000 bytes  Compression=   4.33     *
*.............................................................................*
```

Prisma Timestamp

Raw Branch

Contains raw variables

Analyzed Branch

Contains calibrated / processed variables

Configuration file and configuration folder

filter.conf

```
Domains          0 0
SaveDataDir      .
SpecPrefix       PRISMA
vmeADC           5 32
vmeADC           6 32
vmeADC           7 32
vmeADC           8 32
TSOffset         120
PRISMAManager    /MyPath/PrismaConf/manager.conf
PRISMALUTFile    /MyPath/PrismaConf/lutPRISMA.txt
RootOutputFile   /MyPath/Tree_0000.root
RawDataBranch
AnaDataBranch
```

Don't change

Definitely change

May change

PrismaConf/

```
(base) elia@elia-XPS-13-9305:~/Projects/TESTFILTERS/MyDirectory/PrismaConf$ ls
ban                cal          lutPRISMA.txt  mass.conf  ppac.conf  side.conf    threshold
binarypartner.conf ionch.conf   manager.conf   mcp.conf   README.md  solver.conf  zed.conf
```

- Folders:
  - `ban/`
  - `cal/`
  - `threshold/`
- Lookup table: `lutPRISMA.txt`
- Configuration manager: `manager.conf`

Configuration files:

- `mcp.conf`
- `ppac.conf`
- `ionch.conf`
- `side.conf`
- `solver.conf`
- `zed.conf`
- `mass.conf`
- `binarypartner.conf`

For femul is same but without manager.conf

Configuration file and configuration folder

manager.conf

```
files_path          = /MyDirectory/PrismaConf/
mcp_conf            = mcp.conf
ppac_conf           = ppac.conf
ionch_conf          = ionch.conf
side_conf           = side.conf
solver_conf         = solver.conf
zed_conf            = zed.conf
mass_conf           = mass.conf
binarypartner_conf = binarypartner.conf

manager_ndet    =  4

ndet_mcp        =  1
ndet_ppac       = 10
ndet_ionch      = 10
ndet_side       =  2

ind_mcp         =  0
ind_ppac        =  1
ind_ionch       =  2
ind_side        =  3
```

Insert correct path

lutPRISMA.txt

```
...
6       0       120     IC_0_DE_A
6       1       121     IC_0_DE_B
6       2       122     IC_0_DE_C
...
```

Shouldn't have to be changed

Steps of the analysis

MCP (x,y)
- 2D – gate
- Calibration

PPAC (ToF)
- Thresholds
- *Calibration*

PPAC (x,y)
- Thresholds
- 2D – gates
- Calibration

IC (energy)
- Thresholds
- *Calibration*

SIDE (veto)
- Thresholds

↓

Z – identification (ΔE – E)

- ToF alignment coarse (ToF vs X_{FP} gated on Z)
- ToF alignment fine-tuning (A/q vs X_{FP} gated on Z)
- Set ToF offset

Optical parameters optimization

TRAJECTORY RECONSTRUCTION

↓

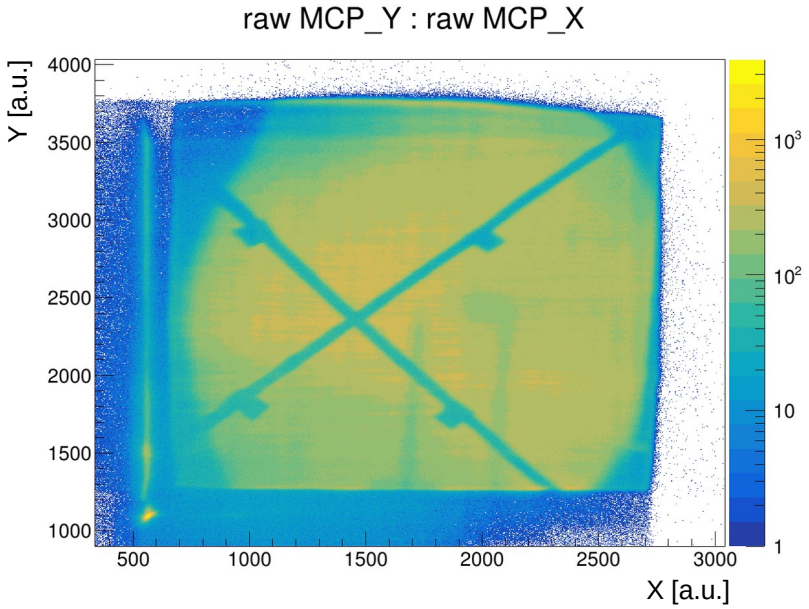
Q – identification (E – Rβ)

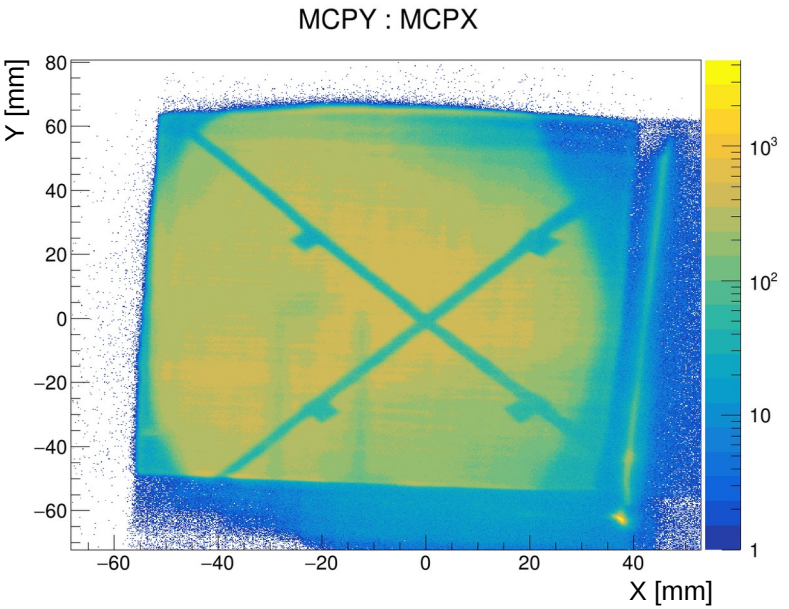
↓

A/q Aberration correction

Mass calibration

MCP calibration





Raw variables

- MCP_raw[3] → 0 = x
  1 = y
  2 = “q” (not used)

Analyzed variables

- MCP_X
- MCP_Y
- MCP_Q
- MCP_Theta
- MCP_Phi

  Z axis is beam,
  Y axis is up, origin is target
- Theta
- Phi

  Z axis is from target to Prisma,
  Y axis is up, origin is target
- mcp_ok

MCP calibration

mcp.conf

```
ind_xm = 0
ind_ym = 1
ind_um = 2

# if you want to ignore the check of the banana just set ignore_banana to 1.
All the events in the MCP_X : MCP_Y will be considered as valid.
mcp_banana = ban/mcp_banana.ban
ignore_banana = 0
ban_res_x = 10096
ban_res_y = 10096

xm_file = cal/x_mcp.cal
ym_file = cal/y_mcp.cal
um_file = cal/u_mcp.cal

xm_gain = 1.0
xm_offs = 0.0

ym_gain = 1.0
ym_offs = 0.0

mcp_mix_x_0 = 1.
mcp_mix_x_1 = -0.078
mcp_mix_y_0 = 0.056
mcp_mix_y_1 = 1.

angle_prisma = 55.
rotation_mcp = 0.2
mcp_target_d = 250.
mcp_angle = 135.
```

2D – gate

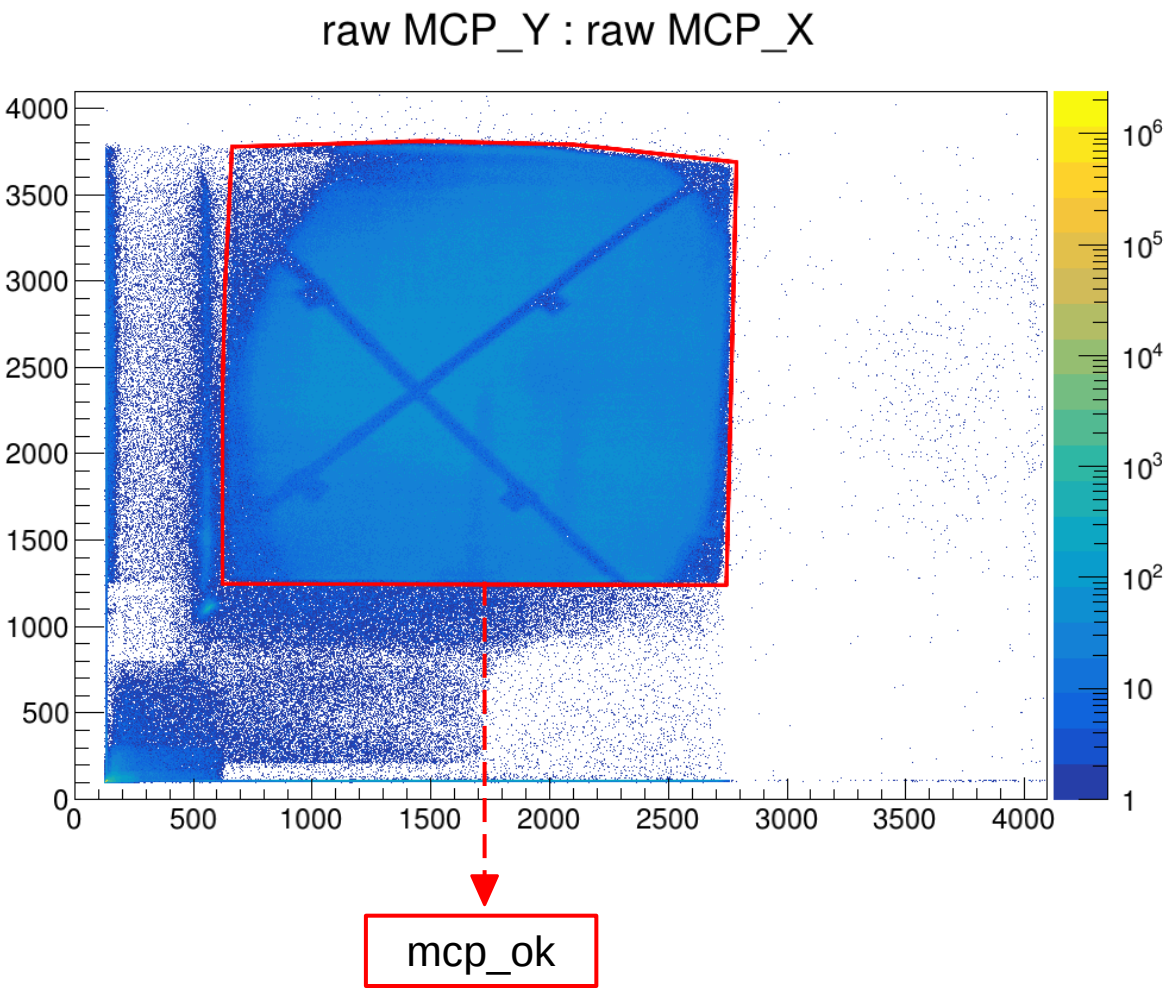
Calibration files

Calibration parameters

Prisma angle

Calibration parameter

MCP calibration



ban/mcp_banana.ban

```
# X_raw       Y_raw
#
659.491       3821.24
625.245       3149.19
615.46        2564.52
605.675       1919.35
625.245       1213.71
855.186       1200.27
2738.75       1220.43
2753.42       2208.33
2753.42       3169.35
2763.21       3693.55
1823.87       3760.75
913.894       3787.63
664.384       3747.31
659.491       3821.24
```

MCP calibration

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x_0 & x_1 \\ y_0 & y_1 \end{pmatrix} \begin{pmatrix} x_{raw} \\ y_{raw} \end{pmatrix}$$

$$\downarrow$$

$$\begin{cases} x'' = a + bx' + c\,(x')^2 \\ y'' = d + ey' \end{cases}$$

$$\downarrow$$

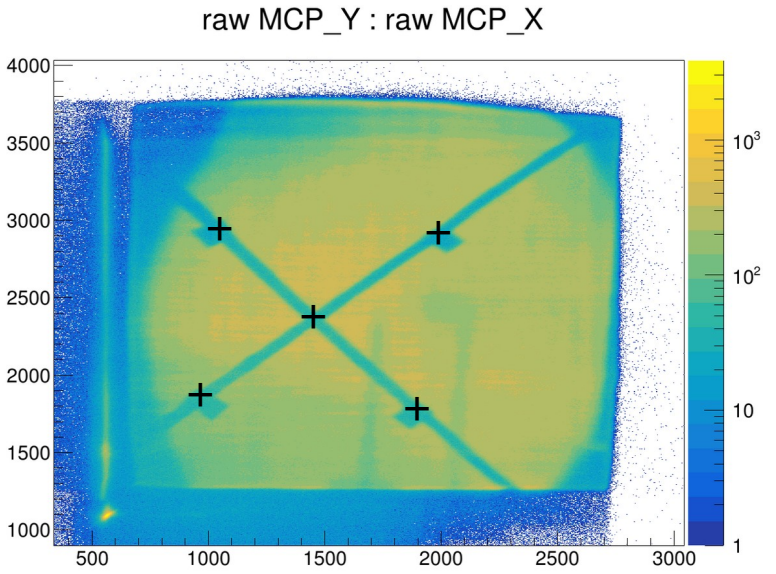
$$\begin{pmatrix} x_f \\ y_f \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x'' \\ y'' \end{pmatrix}$$

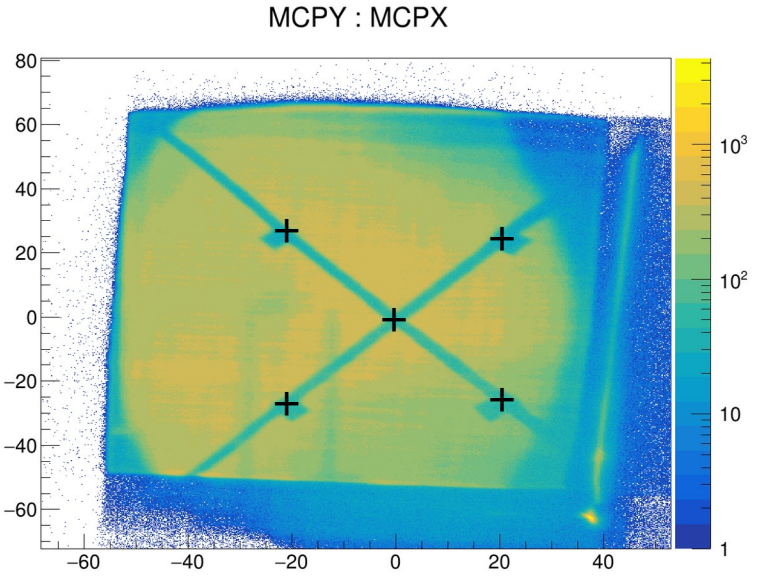
cal/x_mcp.cal

```
99 0 3   64.6318  -0.0579827    4.57388e-06
```

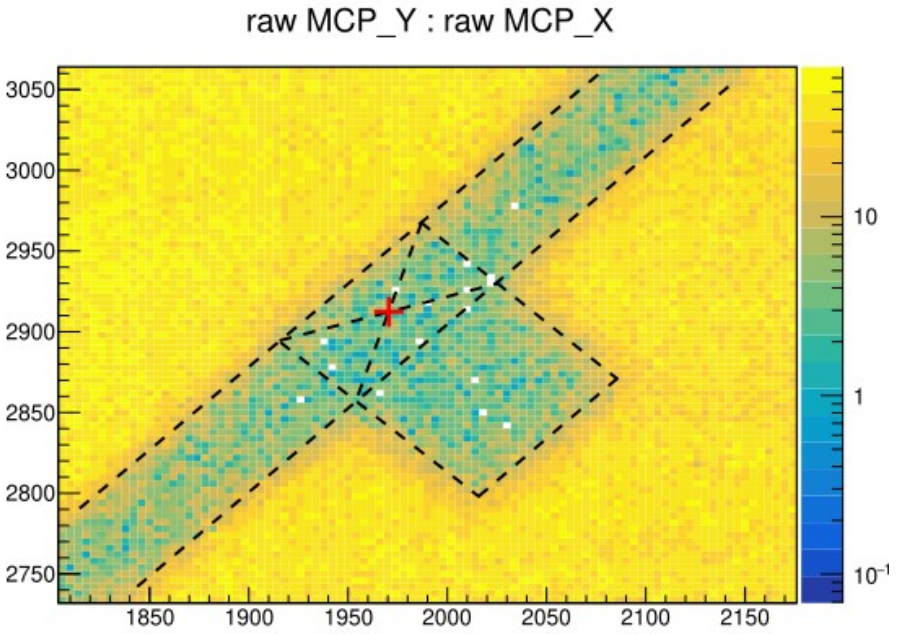
cal/y_mcp.cal

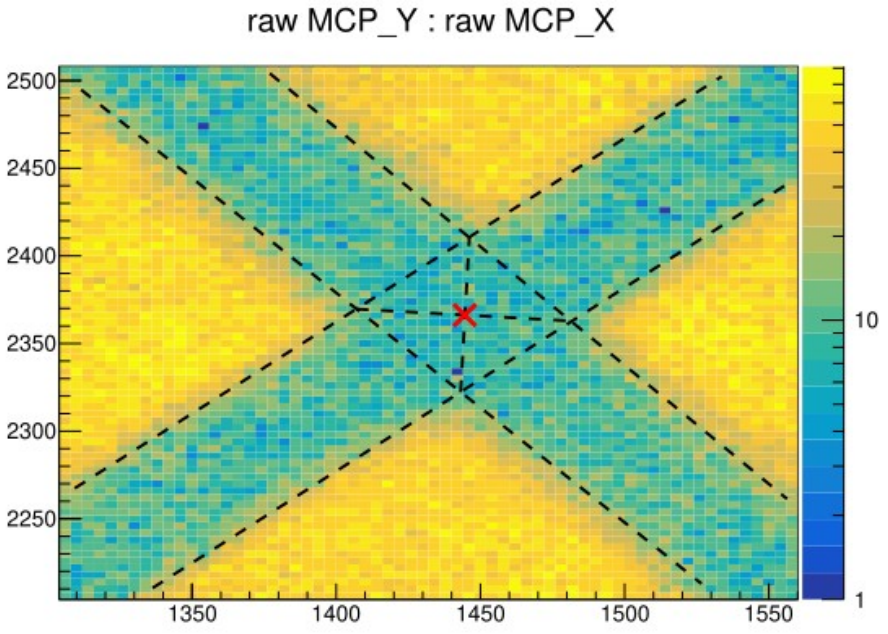
```
99 0 2   -116.334      0.047883
```





MCP calibration





Calibrated			**Raw**		
position	x [mm]	y [mm]	*position*	x [a.u.]	y [a.u.]
center	0	0	center	1500	2500
top left	-21.5	26.5	top	2000	3000
top right	21.5	26.5	top left	1000	3000
bottom left	-21.5	-26.5	bottom right	2000	2000
bottom right	21.5	-26.5	bottom left	1000	2000

Reference position and location in **real** coordinates

Approximate position and location in **raw** coordinates

PPAC calibration

ppac.conf

```
ind_yu  = 0
ind_yd  = 1
ind_xl  = 2
ind_xr  = 3
ind_xc  = 4
ind_tof = 5

##used for calibration and to put Cath-L+R gates
ppac_banana = ban/Cath-L+R.ban
##used to analyze experimental data
ban_res_x = 10000
ban_res_y = 10000

xl_file   = cal/cath-left.cal
xr_file   = cal/right-cath.cal
xfp_file  = cal/xfp-mm.cal
yfp_file  = cal/yfp-mm.cal
tof_file  = cal/tof-total.cal
tof_ofile = cal/alignment-ns.cal

xl_threshold_file  = threshold/x_left.thres
xr_threshold_file  = threshold/x_right.thres
xc_threshold_file  = threshold/x_cathode.thres
tof_threshold_file = threshold/tof.thres

xfp_gate_lowerTh = cal/xfp-gate-lower.cal
xfp_gate_upperTh = cal/xfp-gate-upper.cal

tof_offs = 395.5
```

2D – gates

Calibration files

Threshold files

ToF offset

PPAC calibration – Time of Flight (ToF)

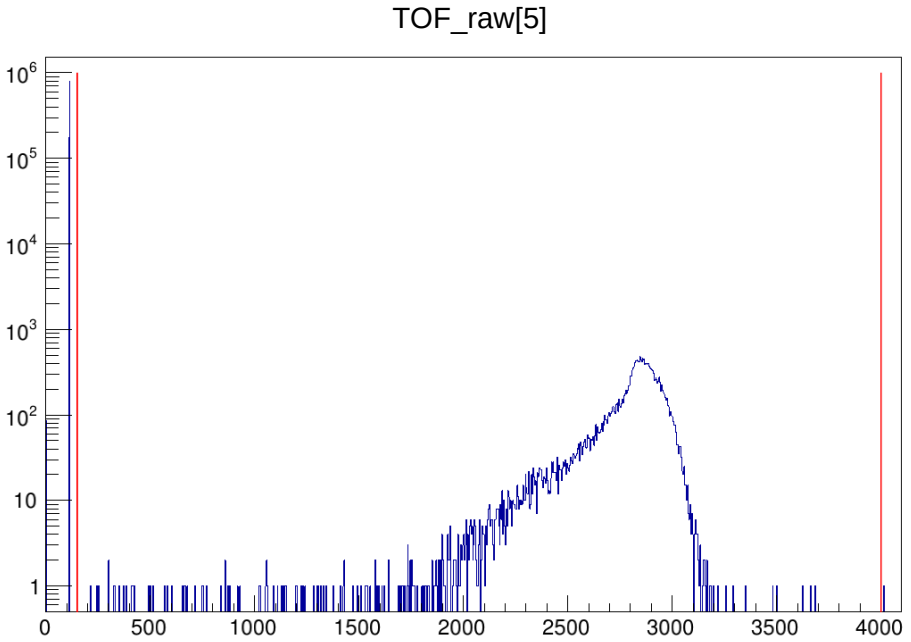
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Raw variables

- ToF_raw[10]

Analyzed variables

- TOF → The one of the section for which position was recorded



threshold/tof.thres

```
99 0 2   150.     4000.
99 1 2   150.     4000.
99 2 2   150.     4000.
99 3 2   150.     4000.
99 4 2   150.     4000.
99 5 2   150.     4000.
99 6 2   150.     4000.
99 7 2   150.     4000.
99 8 2   150.     4000.
99 9 2   150.     4000.
```

cal/tof-total.cal

```
0 0 2    1.95377      -0.0496
0 1 2    7.20707      -0.05028
0 2 2    1.45085      -0.04937
0 3 2    7.57760      -0.04915
0 4 2    4.41215      -0.04952
0 5 2    7.39904      -0.04951
0 6 2    3.94844      -0.04948
0 7 2    1.24438      -0.04928
0 8 2    0.70445      -0.04949
0 9 2    0.29052      -0.04908
```

Shouldn't be changed

PPAC calibration – (x,y) position

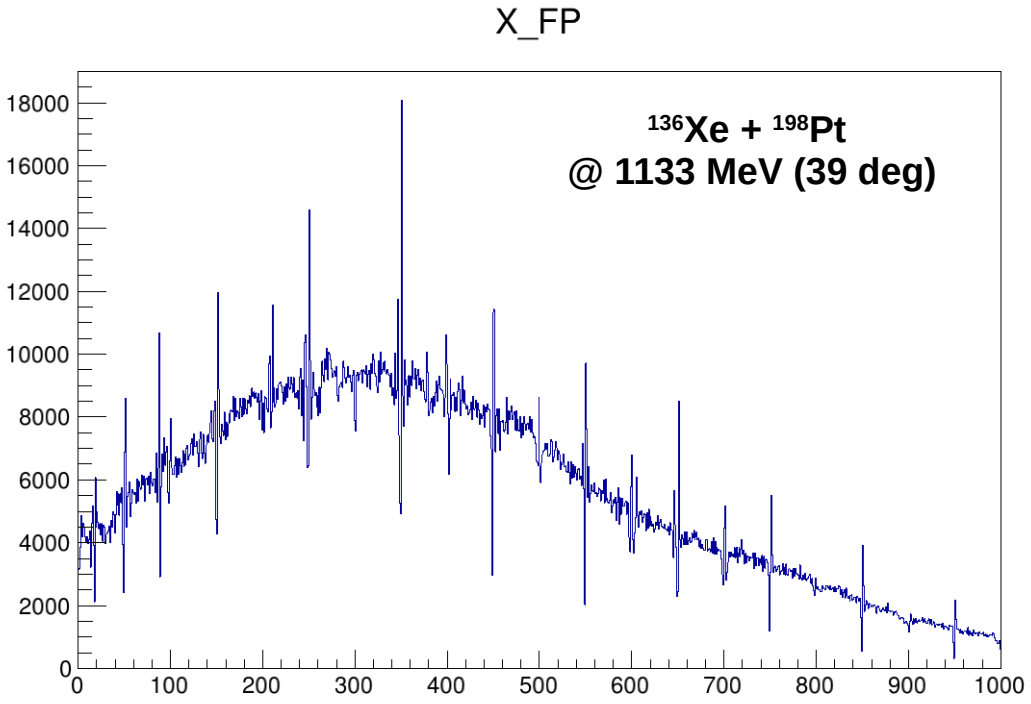
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

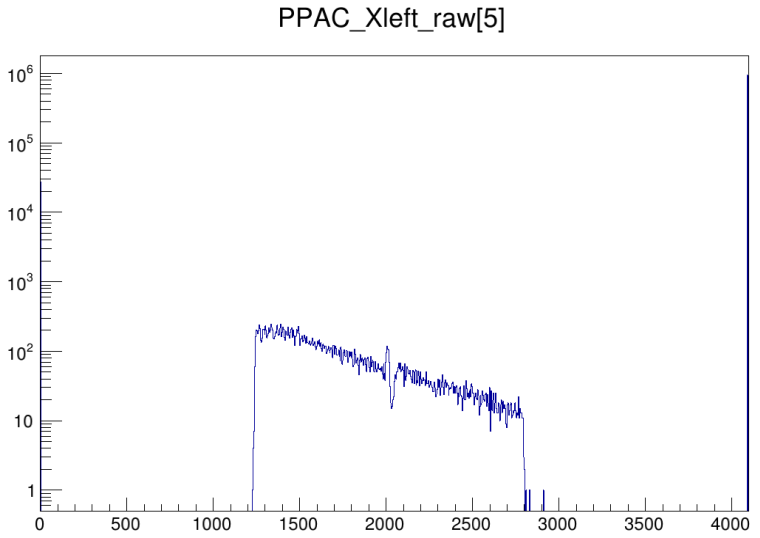
Raw variables

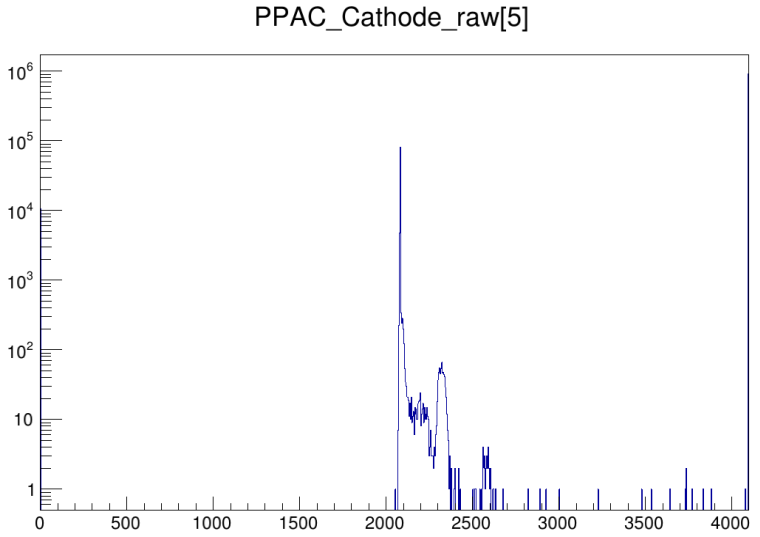
- PPAC_Xleft_raw[10]
- PPAC_Xright_raw[10]
- **PPAC_Cathode_raw[10]**
- PPAC_Y_raw[2]

Analyzed variables

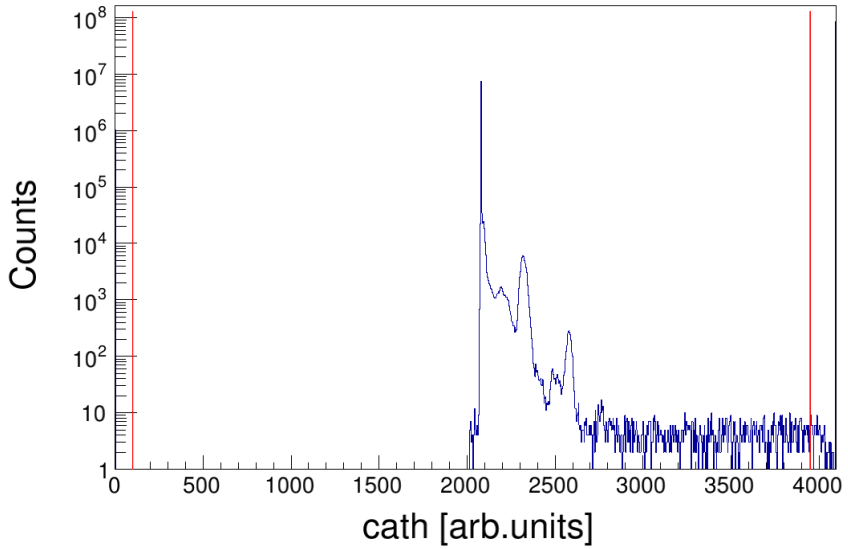
- X_FP
- Y_FP





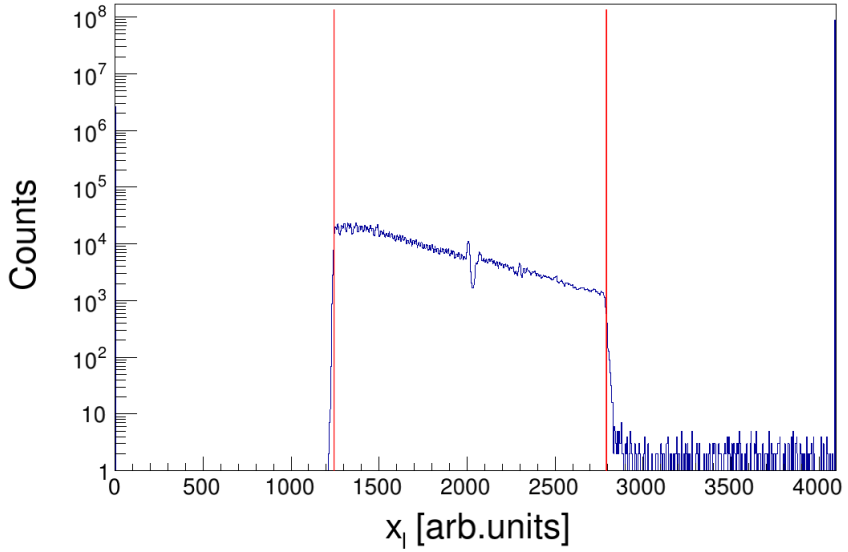


PPAC calibration – (x,y) position



threshold/cath.thres

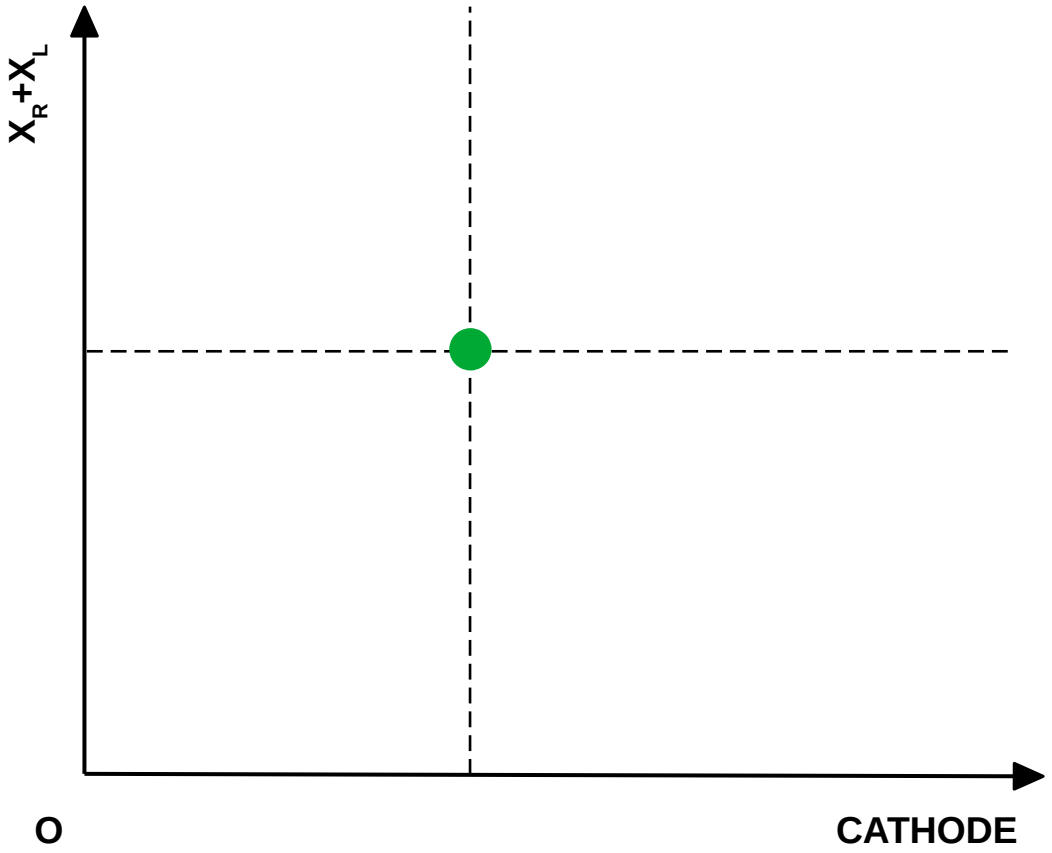
```
99 0 2   150.     4000.
99 1 2   150.     4000.
99 2 2   150.     4000.
99 3 2   150.     4000.
99 4 2   150.     4000.
99 5 2   150.     4000.
99 6 2   150.     4000.
99 7 2   150.     4000.
99 8 2   150.     4000.
99 9 2   150.     4000.
```



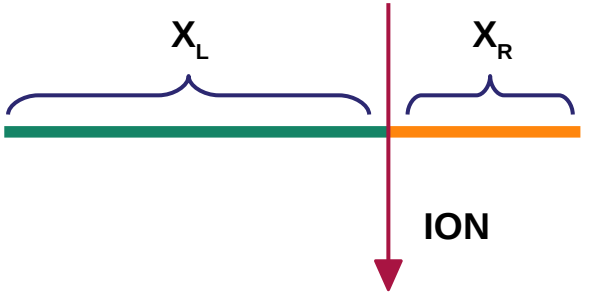
threshold/x_left.thres

```
99 0 2   1380     2916
99 1 2   1073     2663
99 2 2   1153     2654
99 3 2   1131     2649
99 4 2   1298     2782
99 5 2   1244     2792
99 6 2   1293     2818
99 7 2   1340     2841
99 8 2   1634     3161
99 9 2   1511     3080
```

PPAC calibration – (x,y) position



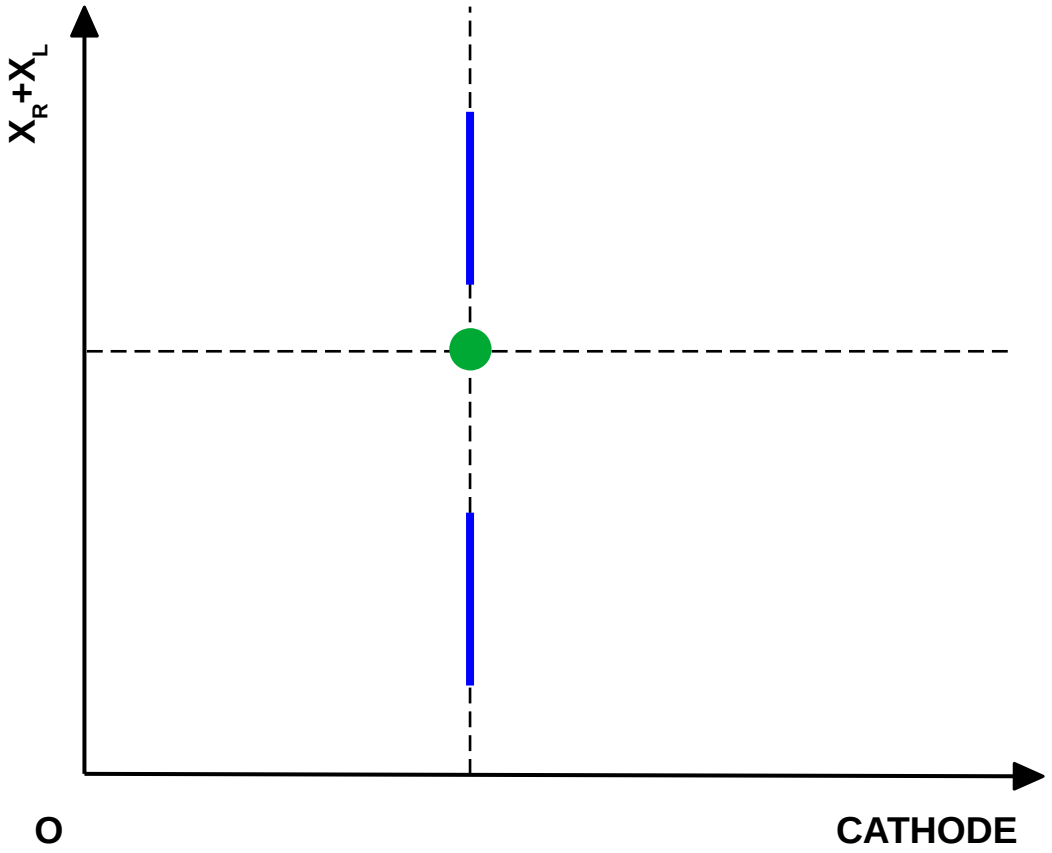
In ideal conditions, the variables represented on both axis should be constant

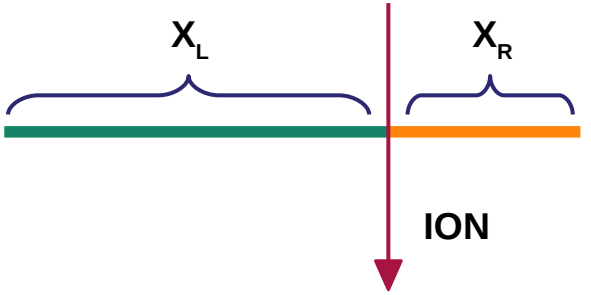


The CATHODE variable is given by the time difference between the cathode signal (START) and the delayed OR of all cathodes (STOP).

The $X_{RIGHT\ /\ LEFT}$ variable is given by the time difference between the cathode signal (START) and the RIGHT / LEFT signal (STOP).

PPAC calibration – (x,y) position





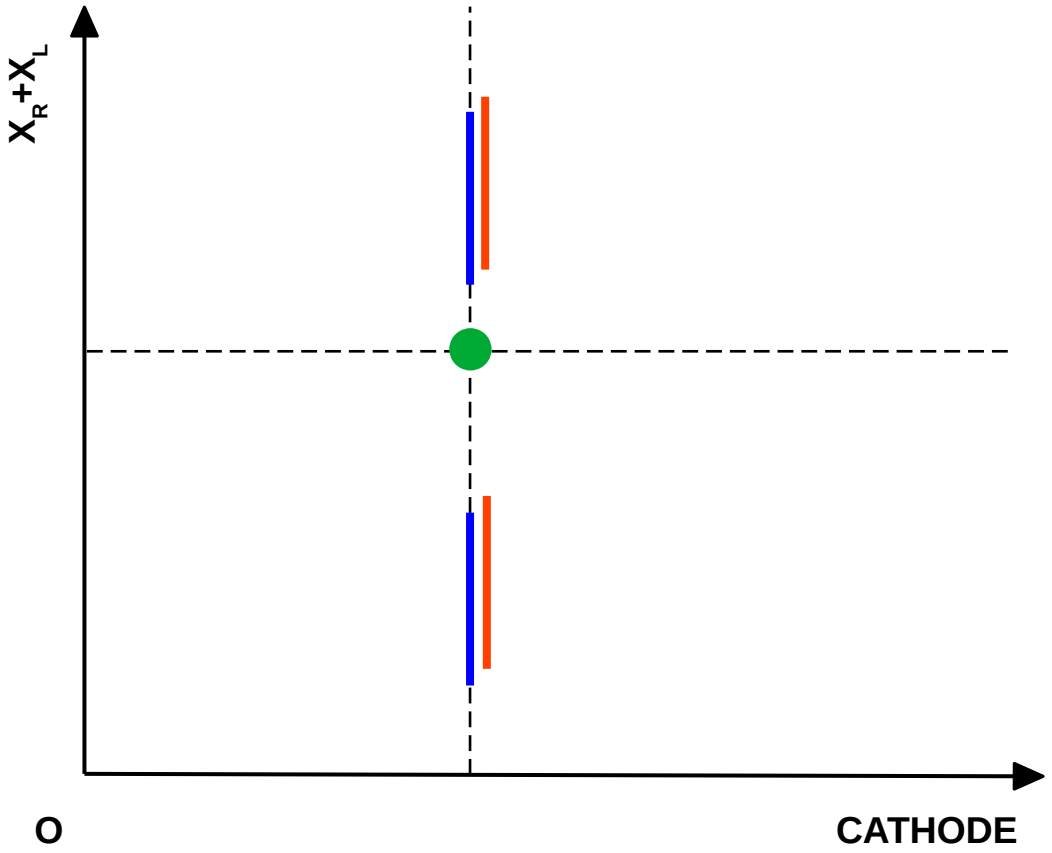
The CATHODE variable is given by the time difference between the cathode signal (START) and the delayed OR of all cathodes (STOP).

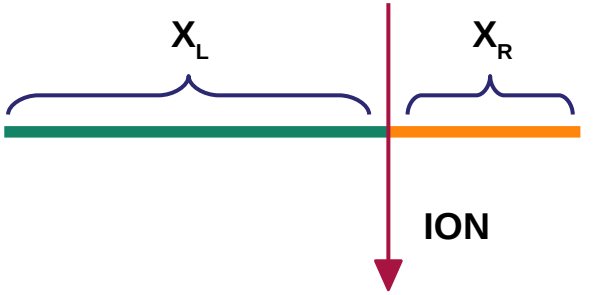
The $X_{RIGHT\ /\ LEFT}$ variable is given by the time difference between the cathode signal (START) and the RIGHT / LEFT signal (STOP).

Due to efficiency, sometimes one of the two signals (RIGHT or LEFT) is missing. Therefore:

$$X_R+X_L = X_{R/L} \quad \text{or} \quad X_R+X_L = X_{R/L} + 4096$$

PPAC calibration – (x,y) position





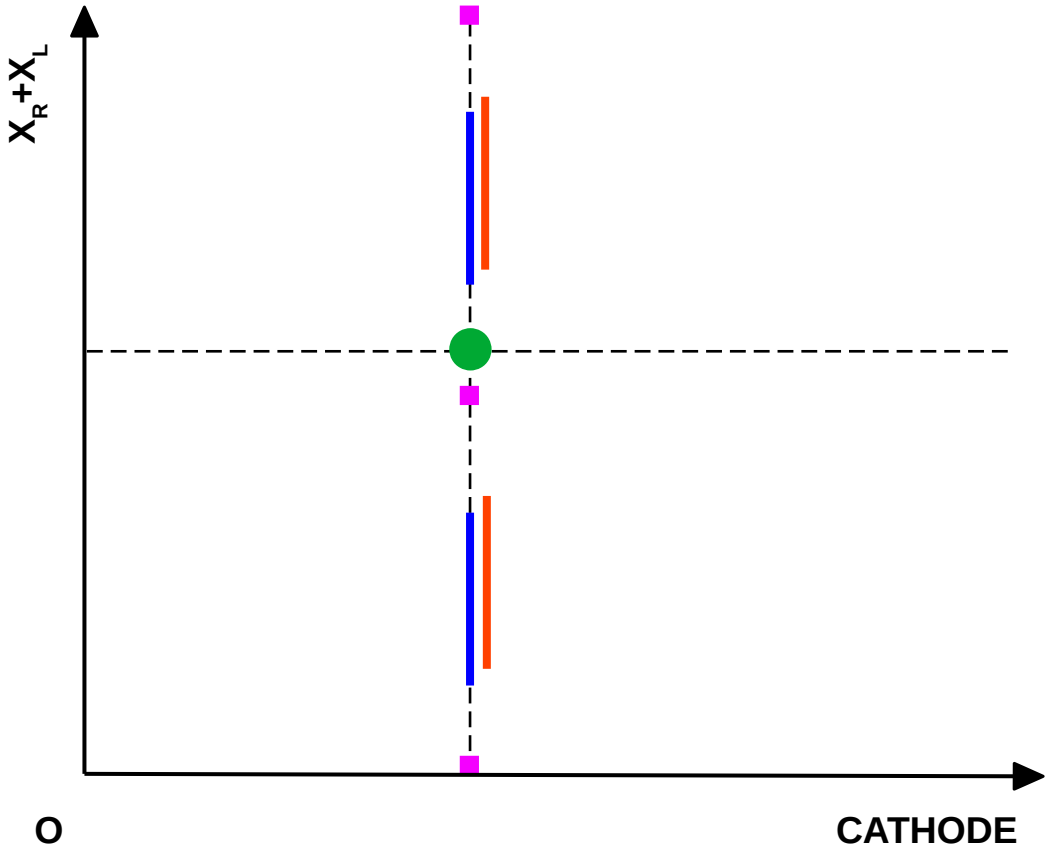
The CATHODE variable is given by the time difference between the cathode signal (START) and the delayed OR of all cathodes (STOP).

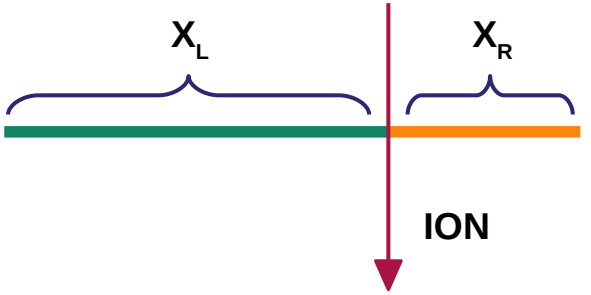
The $X_{RIGHT\ /\ LEFT}$ variable is given by the time difference between the cathode signal (START) and the RIGHT / LEFT signal (STOP).

Due to efficiency, sometimes one of the two signals (RIGHT or LEFT) is missing. Therefore:

$$X_R+X_L = X_{R/L} \quad \text{or} \quad X_R+X_L = X_{R/L} + 4096$$

PPAC calibration – (x,y) position





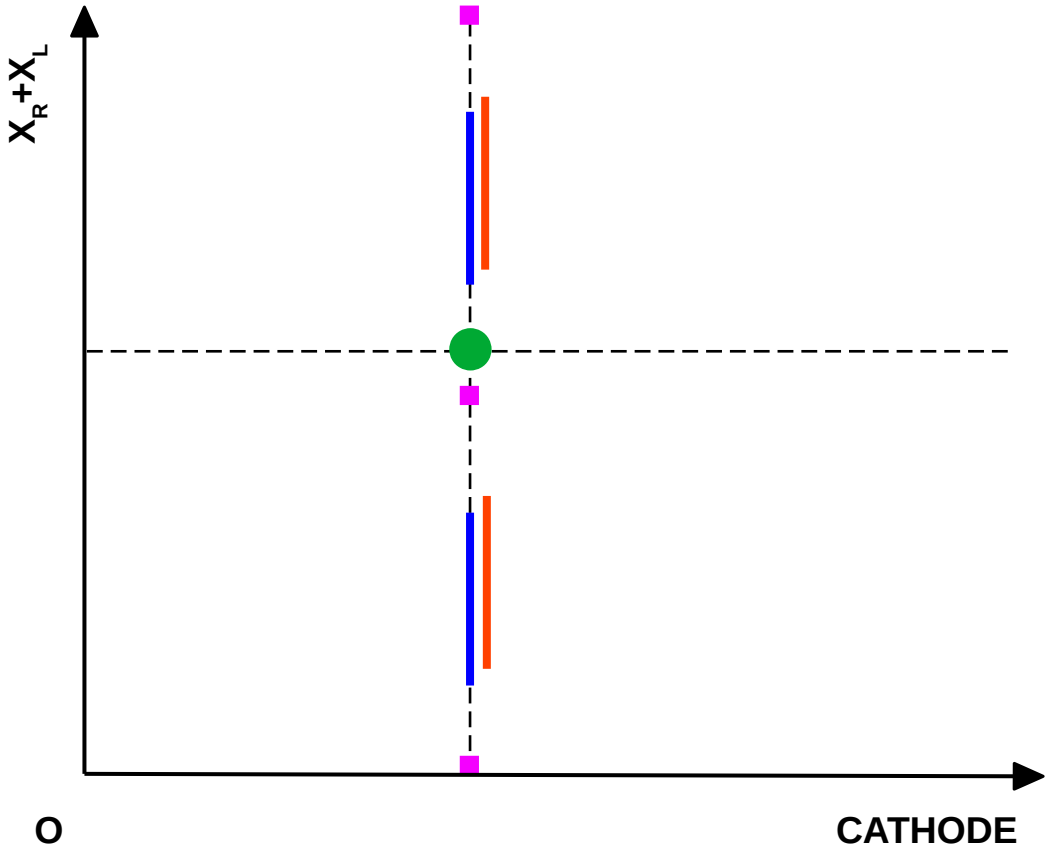
The CATHODE variable is given by the time difference between the cathode signal (START) and the delayed OR of all cathodes (STOP).

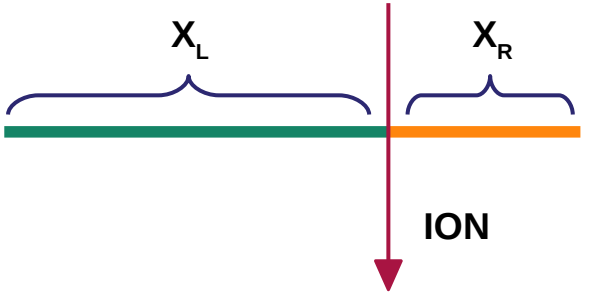
The $X_{RIGHT\ /\ LEFT}$ variable is given by the time difference between the cathode signal (START) and the RIGHT / LEFT signal (STOP).

Sometimes both RIGHT and LEFT are missing.

X_R+X_L = 0, 4096, 8192

PPAC calibration – (x,y) position





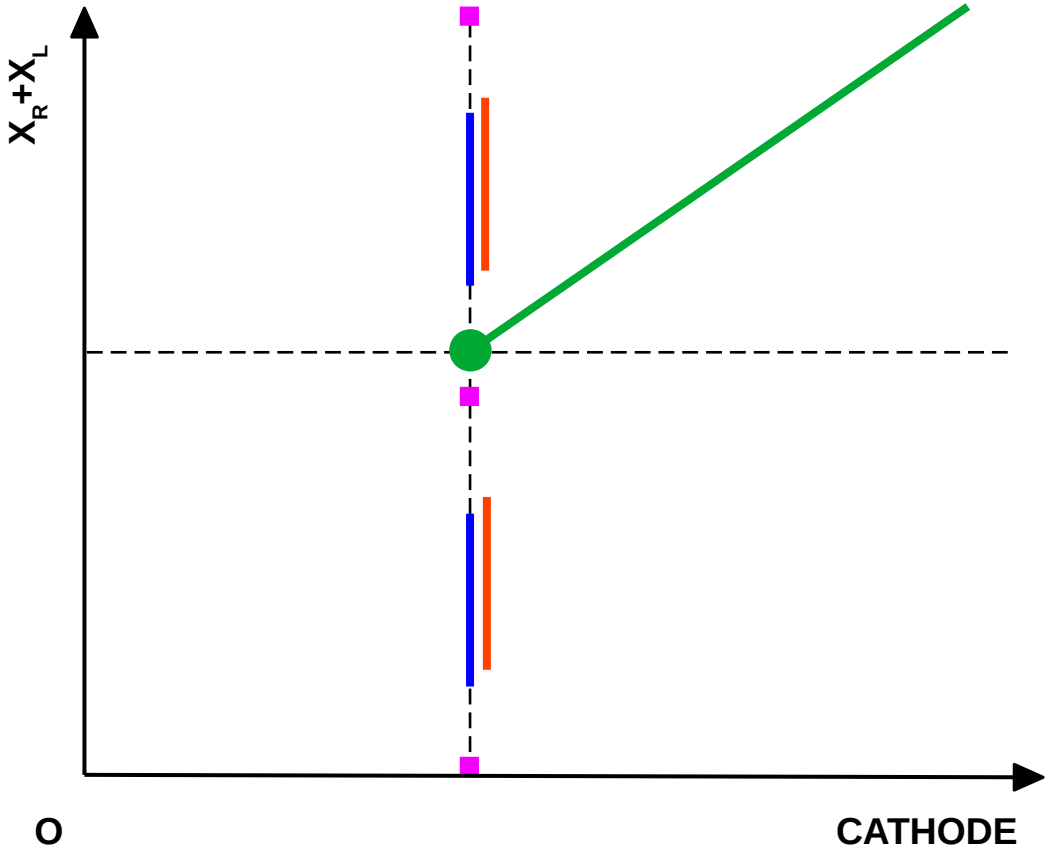
$$cath^{(real)} = cath + drift$$

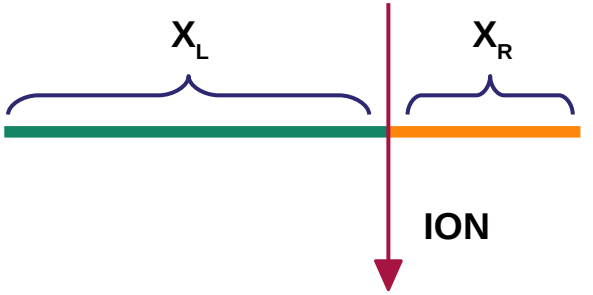
$$X_{left}^{(real)} = X_{left} + drift$$

$$X_{right}^{(real)} = X_{right} + drift$$

In reality, a random event-by-event drift can be observed on all three signals

PPAC calibration – (x,y) position





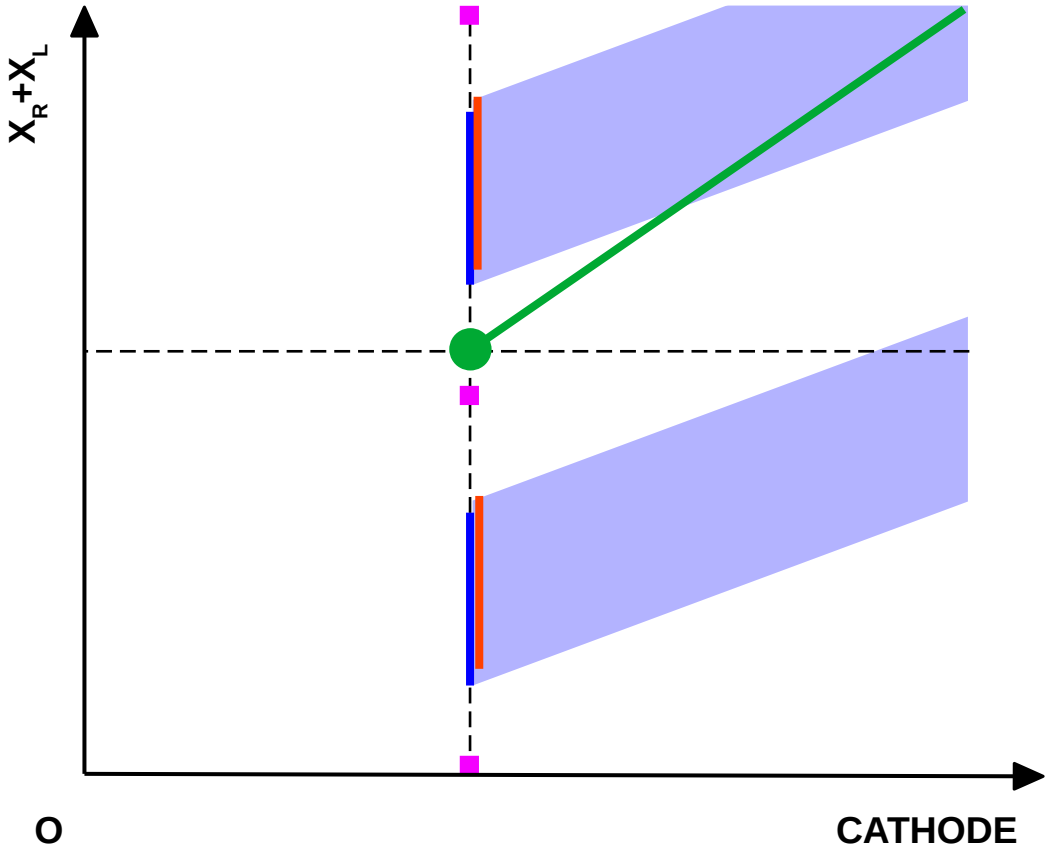
$$cath^{(real)} = cath + drift$$

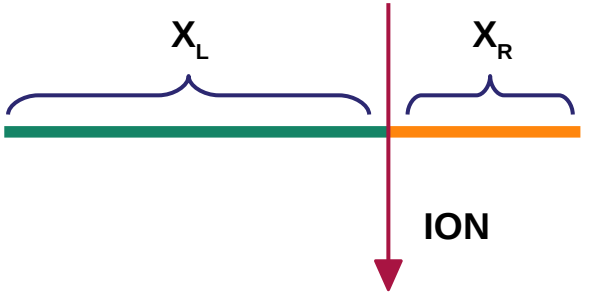
$$X_{left}^{(real)} = X_{left} + drift$$

$$X_{right}^{(real)} = X_{right} + drift$$

In reality, a random event-by-event drift can be observed on all three signals

PPAC calibration – (x,y) position





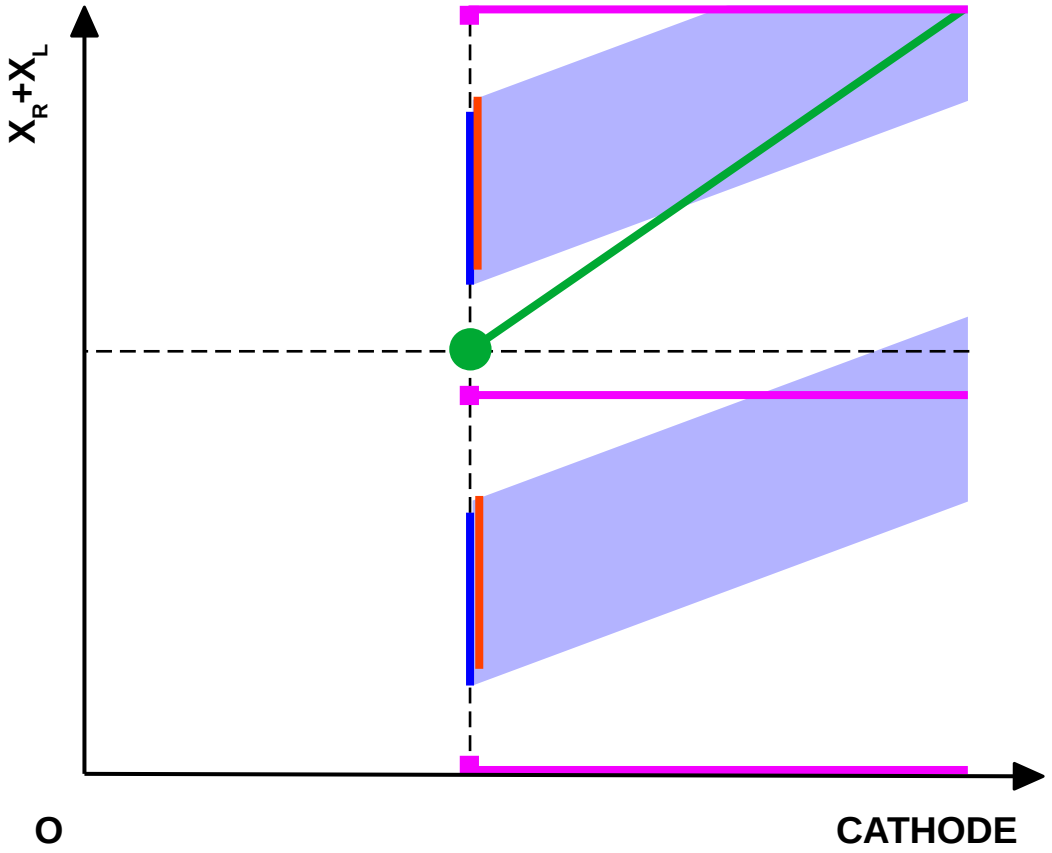
$$cath^{(real)} = cath + drift$$

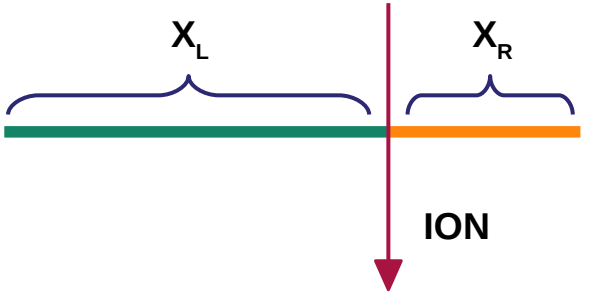
$$X_{left}^{(real)} = X_{left} + drift$$

$$X_{right}^{(real)} = X_{right} + drift$$

In reality, a random event-by-event drift can be observed on all three signals

PPAC calibration – (x,y) position





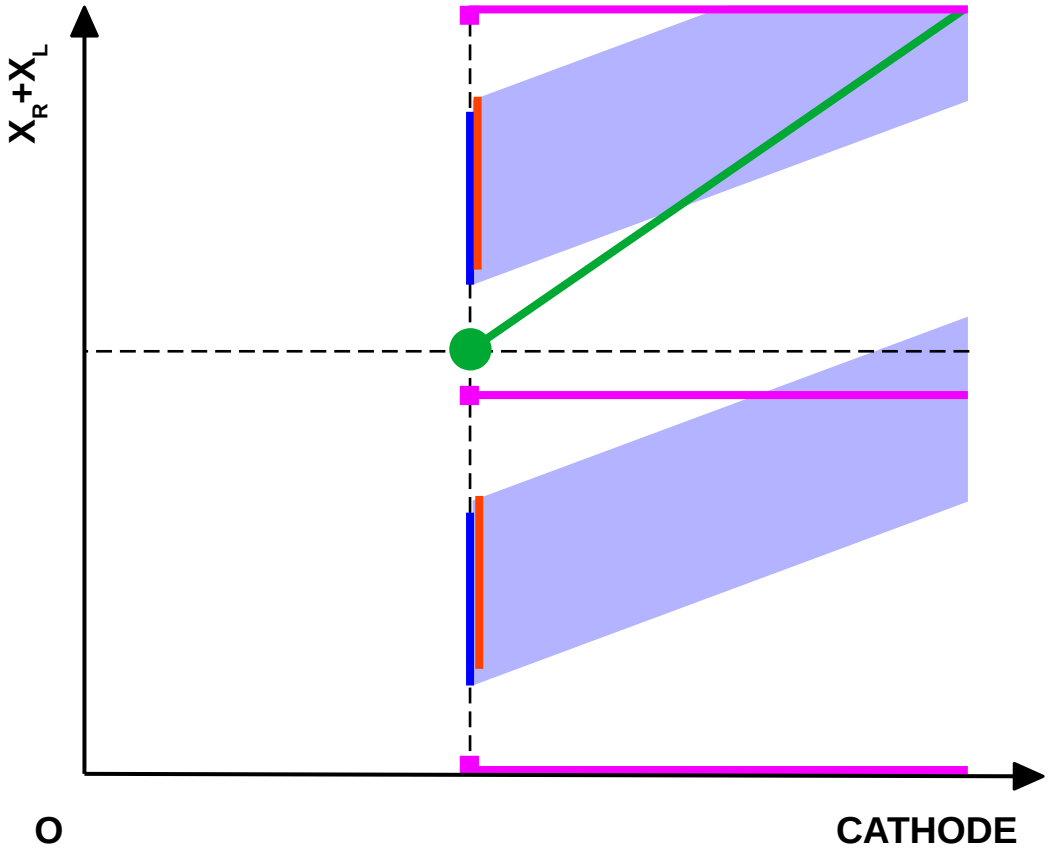
$$cath^{(real)} = cath + drift$$

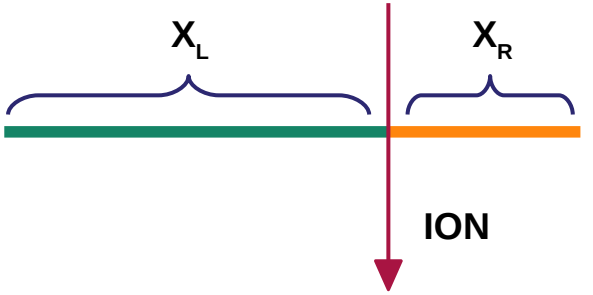
$$X_{left}^{(real)} = X_{left} + drift$$

$$X_{right}^{(real)} = X_{right} + drift$$

In reality, a random event-by-event drift can be observed on all three signals

PPAC calibration – (x,y) position





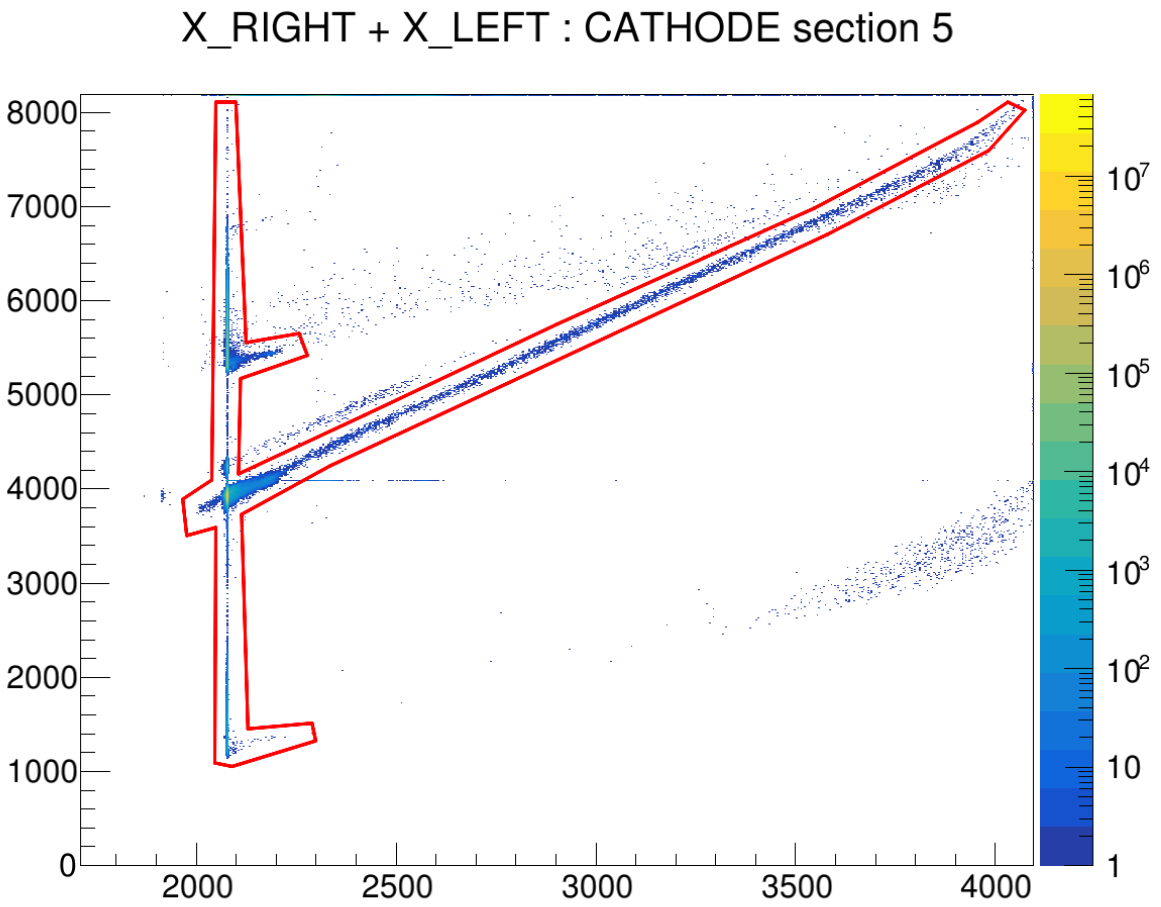
$$cath^{(real)} = cath + drift$$

$$X_{left}^{(real)} = X_{left} + drift$$

$$X_{right}^{(real)} = X_{right} + drift$$

Everything else should be rejected

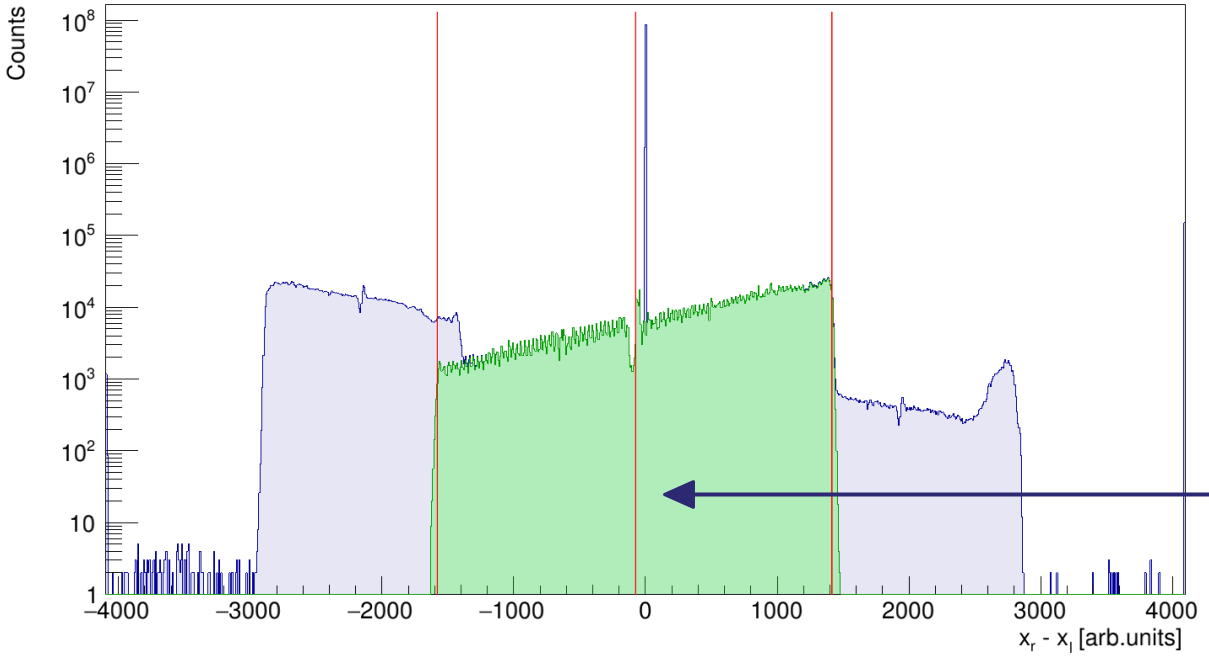
PPAC calibration – (x,y) position



ban/Cath-L+R.ban

```
ADC 0
2164.36        5614.43
2752.65        7183.06
3179.31        6856.78
2749.41        4083.45
2720.32        1159.53
2096.48        1059.14
2164.36        5614.43
ADC 1
1915.47        657.569
2170.83        682.667
2132.04        3619.14
2335.68        4271.69
2161.13        4848.94
1902.55        4497.57
1915.47        657.569
ADC 2
...
```

PPAC calibration – (x,y) position



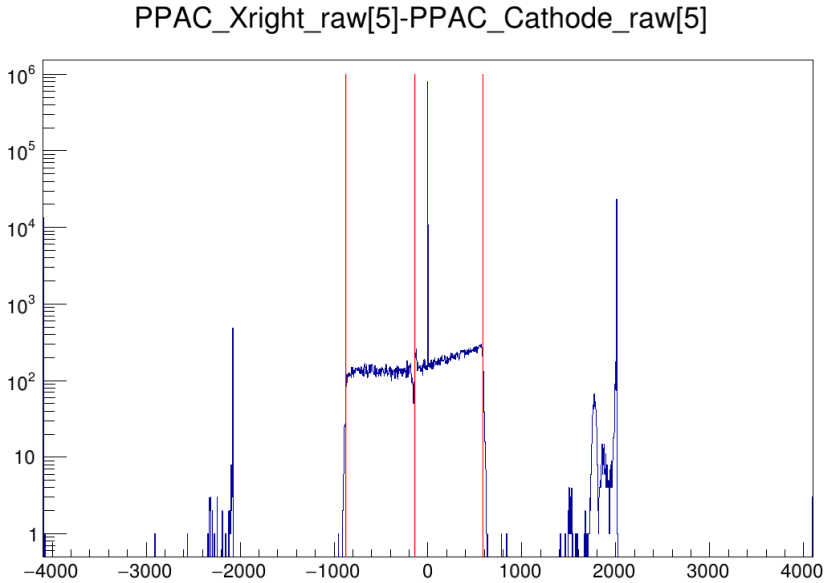
The two central wires are short-circuited for reference

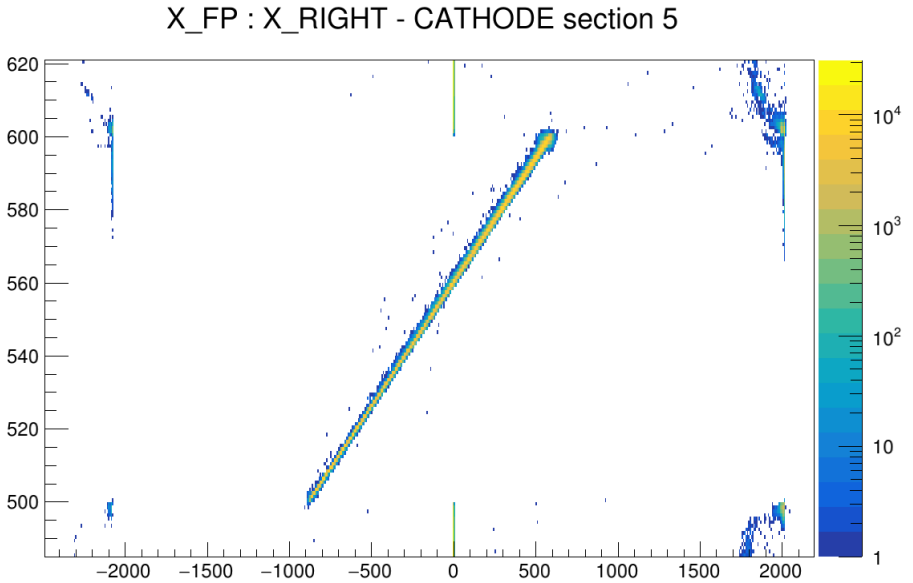
cal/xfp-mm.cal

```
0    0    2      22.8390        0.033068
0    1    2      140.473        0.031969
0    2    2      246.771        0.033510
0    3    2      345.847        0.033046
0    4    2      449.119        0.033455
0    5    2      552.932        0.033260
0    6    2      649.307        0.032957
0    7    2      753.357        0.033023
0    8    2      854.932        0.032099
0    9    2      954.912        0.032319
```

Linear calibration is obtained through interpolation

PPAC calibration – (x,y) position





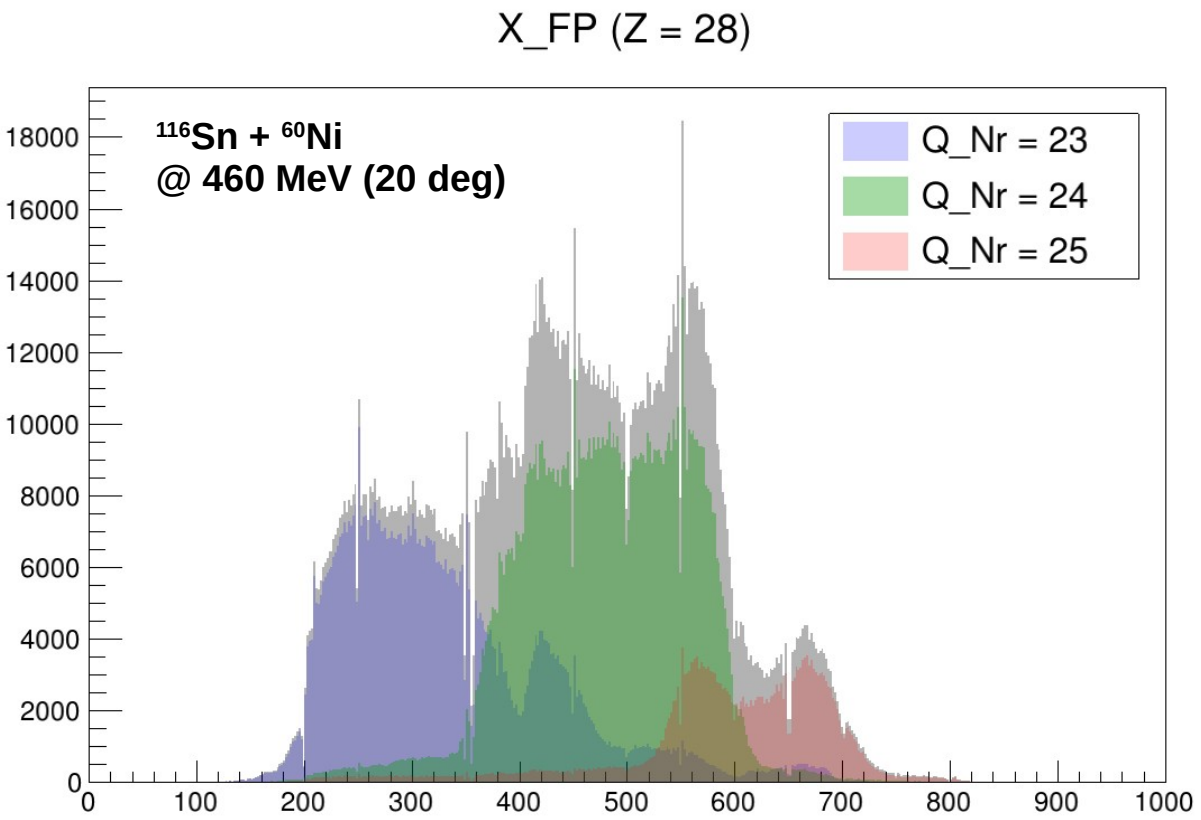
cal/cath-left.cal

```
2 0 2    34.2953      0.057012
2 1 2    140.138      0.060773
2 2 2    235.959      0.062755
2 3 2    334.528      0.063331
2 4 2    448.666      0.066711
2 5 2    546.093      0.065287
2 6 2    643.451      0.064010
2 7 2    744.054      0.063801
2 8 2    849.711      0.065035
2 9 2    946.486      0.065151
```

cal/right-cath.cal

```
2 0 2    7.85676      0.062334
2 1 2    143.224      0.063618
2 2 2    245.719      0.064189
2 3 2    357.075      0.064529
2 4 2    449.238      0.063880
2 5 2    560.522      0.068181
2 6 2    657.069      0.066841
2 7 2    762.603      0.065166
2 8 2    861.401      0.065061
2 9 2    966.342      0.066576
```

PPAC calibration – (x,y) position



OBS: To assign the correct ToF, X_FP has to be determined

↓

tof_ok = ToF & X_FP

OBS: For lighter systems, different charge states end up in different regions of the focal plane

Ionization Chamber (IC)

ionch.conf

```
ind_DE_A    = 0
ind_DE_B    = 1
ind_DE_C    = 2
ind_DE_D    = 3
ind_DRIFT_A = 4
ind_DRIFT_B = 5
ind_DRIFT_C = 6

ic_A_Calibration_file = cal/IC_gain_300_30_6_A_2022_chargeInject.cal
ic_B_Calibration_file = cal/IC_gain_300_30_6_B_2022_chargeInject.cal
ic_C_Calibration_file = cal/IC_gain_300_30_6_C_2022_chargeInject.cal
ic_D_Calibration_file = cal/IC_gain_300_30_6_D_2022_chargeInject.cal

drift_A_Calibration_file = cal/calDefault_x_Drift_A.cal
drift_B_Calibration_file = cal/calDefault_x_Drift_B.cal
drift_C_Calibration_file = cal/calDefault_x_Drift_C.cal

ic_A_Threshold_file = threshold/IC_A.thres
ic_B_Threshold_file = threshold/IC_B.thres
ic_C_Threshold_file = threshold/IC_C.thres
ic_D_Threshold_file = threshold/IC_D.thres

drift_A_Threshold_file = threshold/DRIFT_A.thres
drift_B_Threshold_file = threshold/DRIFT_B.thres
drift_C_Threshold_file = threshold/DRIFT_C.thres
...
```

Calibration files

Threshold files

Ionization Chamber (IC)

ionch.conf

```
...
threshold_before_cal=1
veto_threshold = 4000.
max_veto_id = 10

ic_A_gain = 1.0
ic_A_offs = 0.0

ic_B_gain = 1.0
ic_B_offs = 0.0

ic_C_gain = 1.0
ic_C_offs = 0.0

ic_D_gain = 1.0
ic_D_offs = 0.0
```

Shouldn't be changed

Ionization Chamber (IC)

Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Raw variables

- IC_A_raw[10]
- IC_B_raw[10]
- IC_C_raw[10]
- IC_D_raw[10]

→ Raw energy losses

- IC_A_Drift_raw[10]
- IC_B_Drift_raw[10]
- IC_C_Drift_raw[10]

→ Raw drift times with respect to PPAC OR of all cathodes

Analyzed variables

- IC_Pads[40] → Calibrated energy losses
- IC_E → Total energy lost (sum of all IC_Pads)
- IC_DE_A
- IC_DE_AB

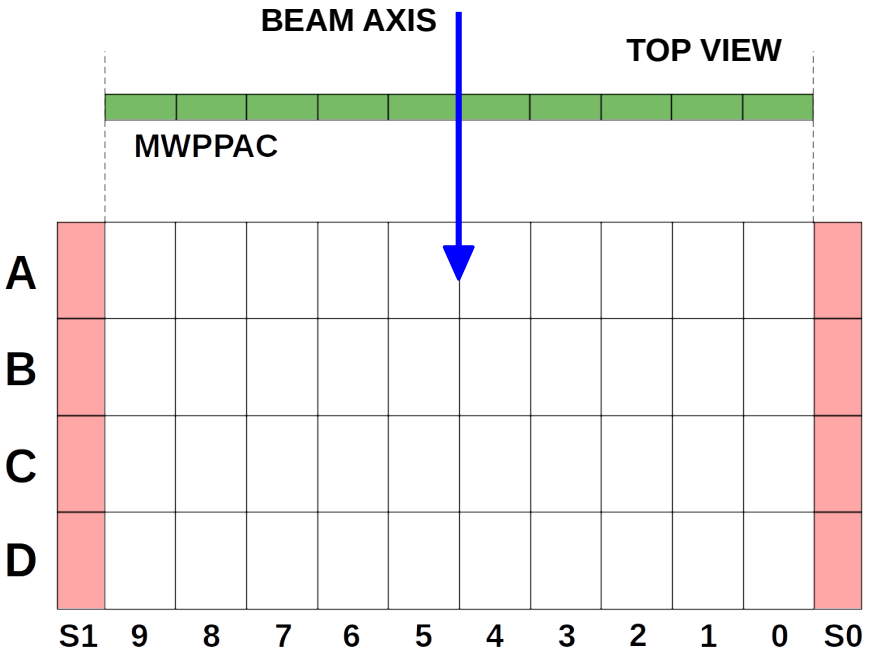
→ Total energy lost in rows A / AB

- IC_RANGE → Range calculated as weighted average of energy losses
- IC_Drift_A
- IC_Drift_B
- IC_Drift_C

→ Calibrated drift times

- IC_col_a
- IC_col_b
- IC_col_c
- IC_col_d

→ Column number with highest energy deposition calculated row by row

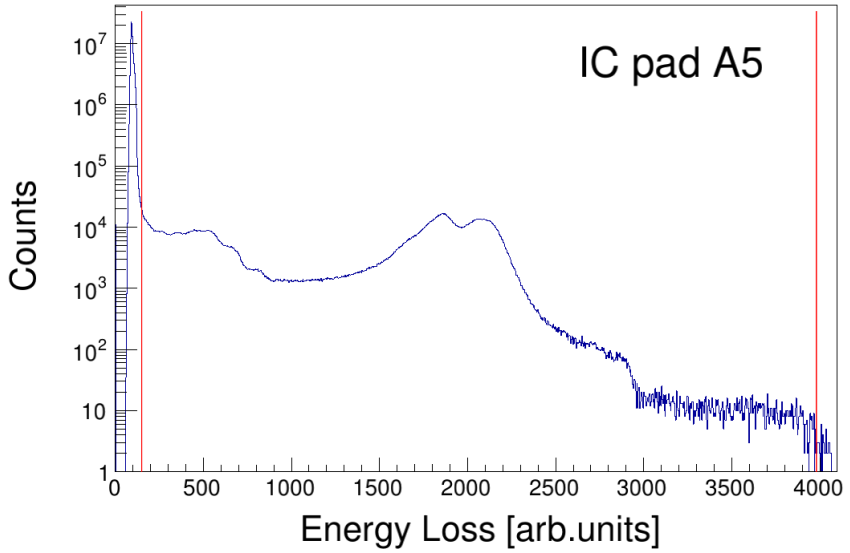


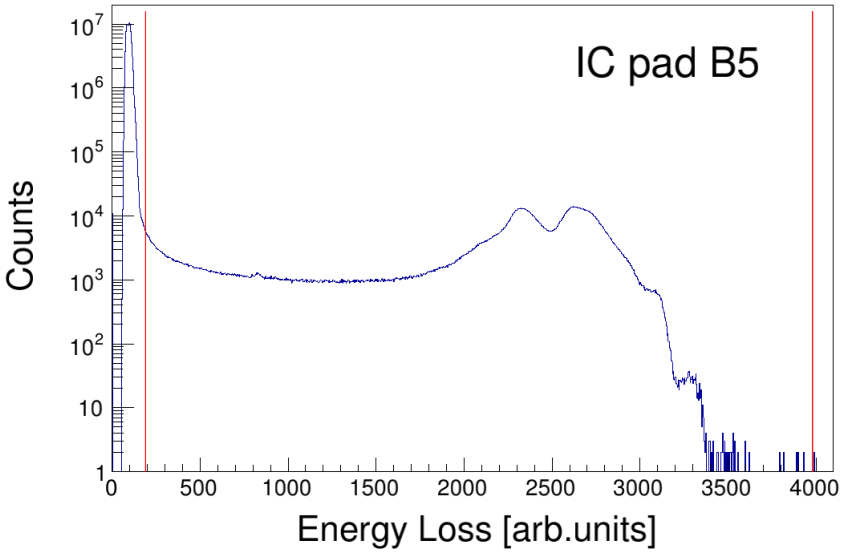
- IC_a_numpads
- IC_b_numpads
- IC_c_numpads
- IC_d_numpads

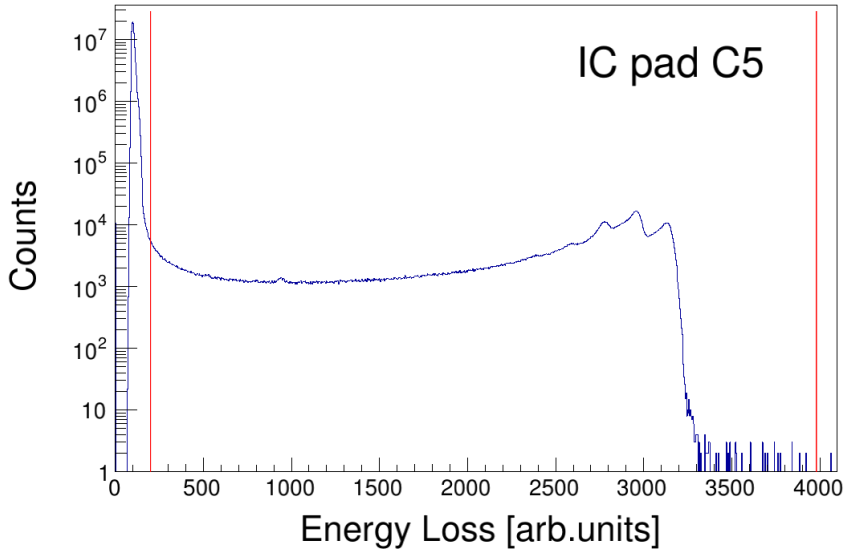
→ Number of pads above threshold in each row

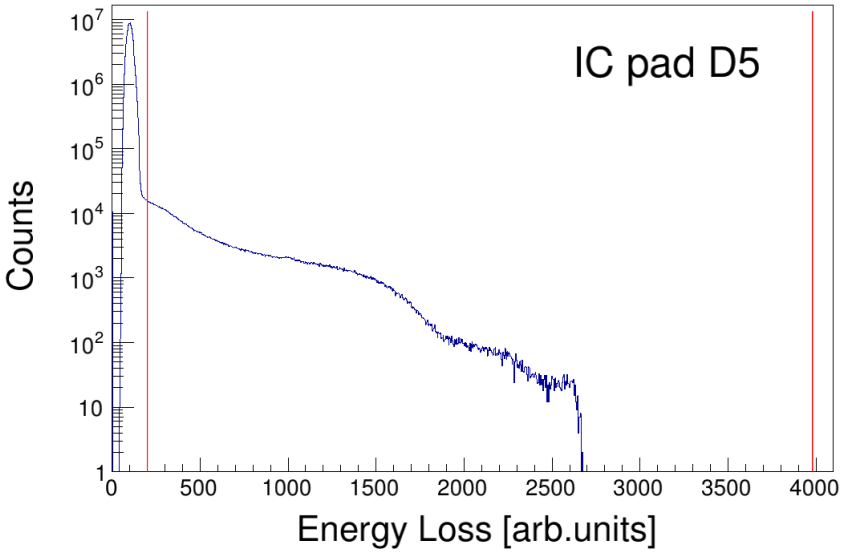
- ic_ok → At least one pad has fired

Ionization Chamber (IC)









Side detectors

side.conf

```
ind_s1 = 0
ind_s2 = 1
ind_s3 = 2
ind_s4 = 3

side_A_Threshold_file = threshold/side_A.thres
side_B_Threshold_file = threshold/side_B.thres
side_C_Threshold_file = threshold/side_C.thres
side_D_Threshold_file = threshold/side_D.thres
```

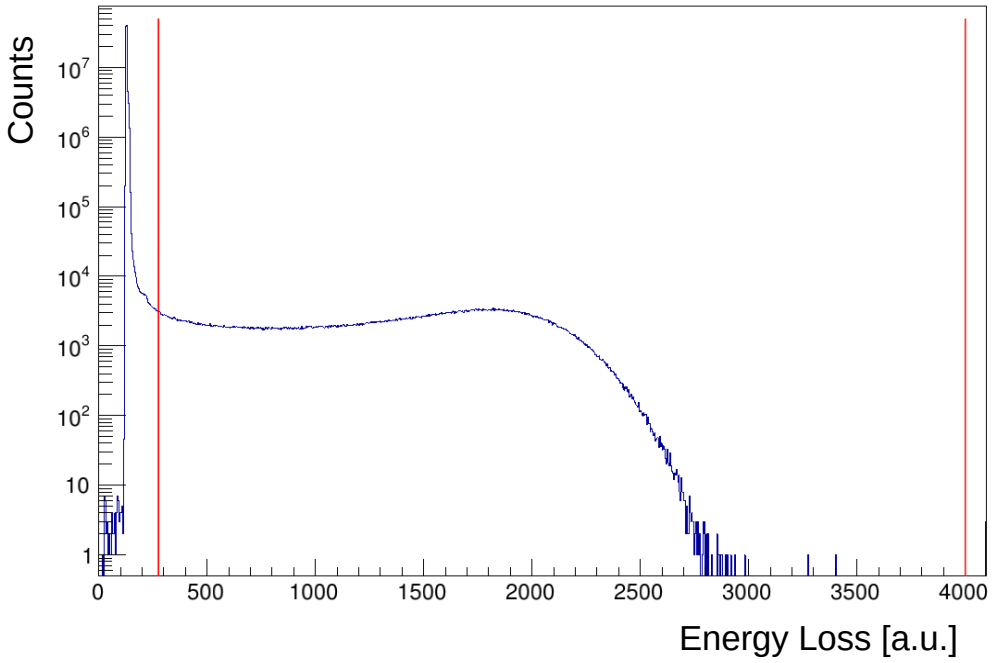
Threshold files

Raw variables

- Side_A_raw[2]
- Side_B_raw[2]
- Side_C_raw[2]
- Side_D_raw[2]

Analyzed variables

- side_ok



Steps of the analysis - update

MCP (x,y)
- 2D – gate
- Calibration

PPAC (ToF)
- Thresholds
- *Calibration*

PPAC (x,y)
- Thresholds
- 2D – gates
- Calibration

IC (energy)
- Thresholds
- *Calibration*

SIDE (veto)
- Thresholds

Z – identification (ΔE – E)

- ToF alignment coarse (ToF vs X_{FP} gated on Z)
- ToF alignment fine-tuning (A/q vs X_{FP} gated on Z)
- Set ToF offset

Optical parameters optimization

Q – identification (E – Rβ)

A/q Aberration correction

Mass calibration

Z - identification (ΔE – E)

zed.conf

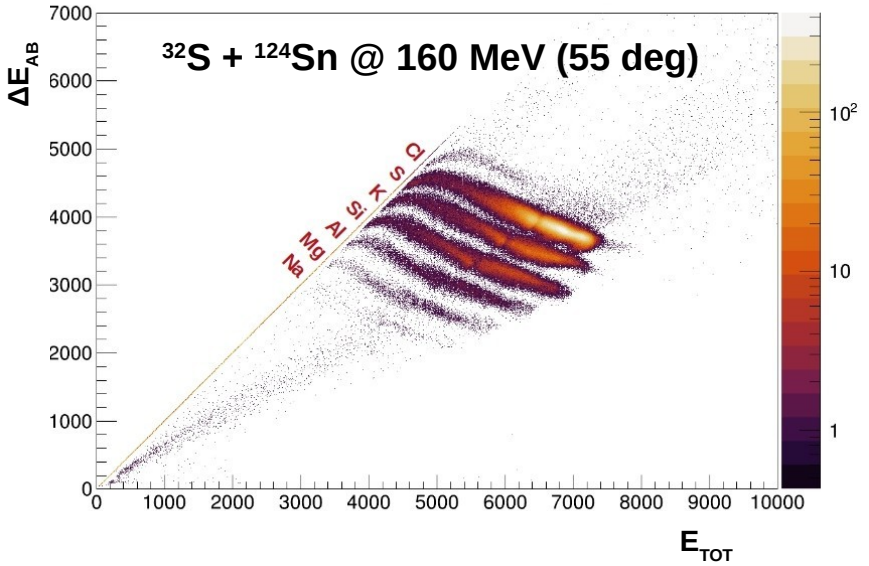
```
z_min = 10
z_max = 17

#Enable certain cuts for determine the Z of the particle
use_dea  = false
use_deab = true
use_path = false

#bananas for the DEA_E Cut
ban_file_base_dea = ban/zed_ban_dea_

#bananas for the DEAB_E Cut
ban_file_base_deab = ban/zed_ban_deab_

#bananas for the path cut
ban_file_base_path = ban/zed_ban_range_

ban_res_x = 16000
ban_res_y = 16000

gain_energy = 1.0
gain_range  = 1.0
```

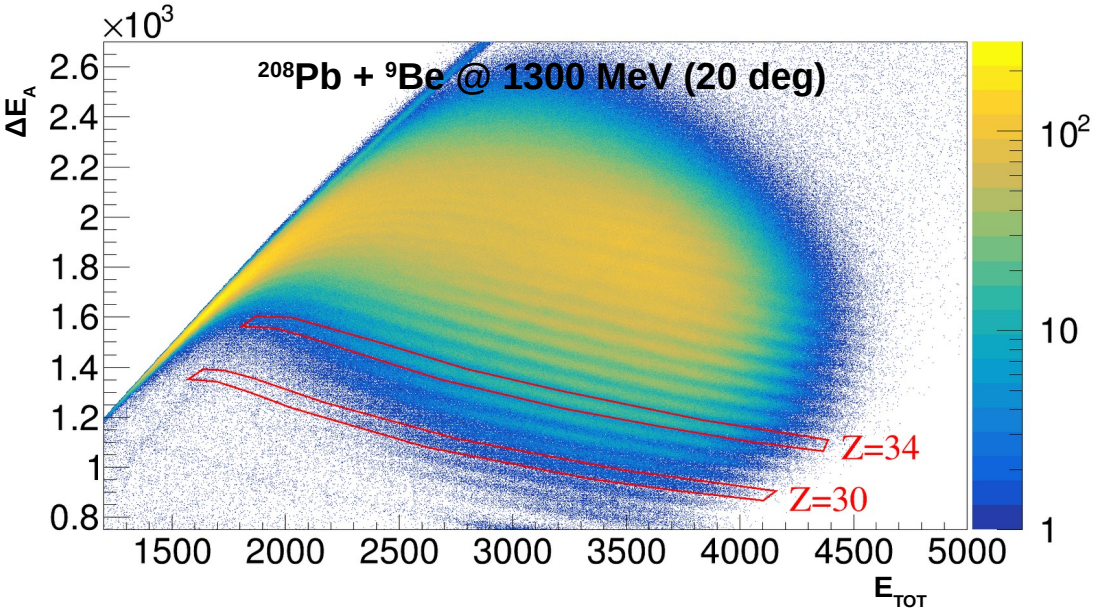
Example:
`ban/zed_ban_deab_16.ban`

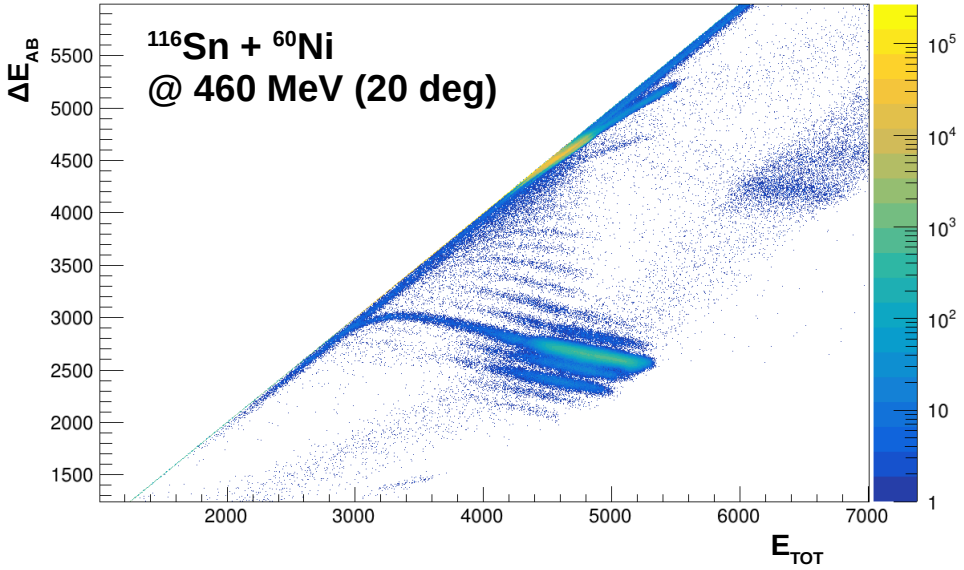
Analyzed variables

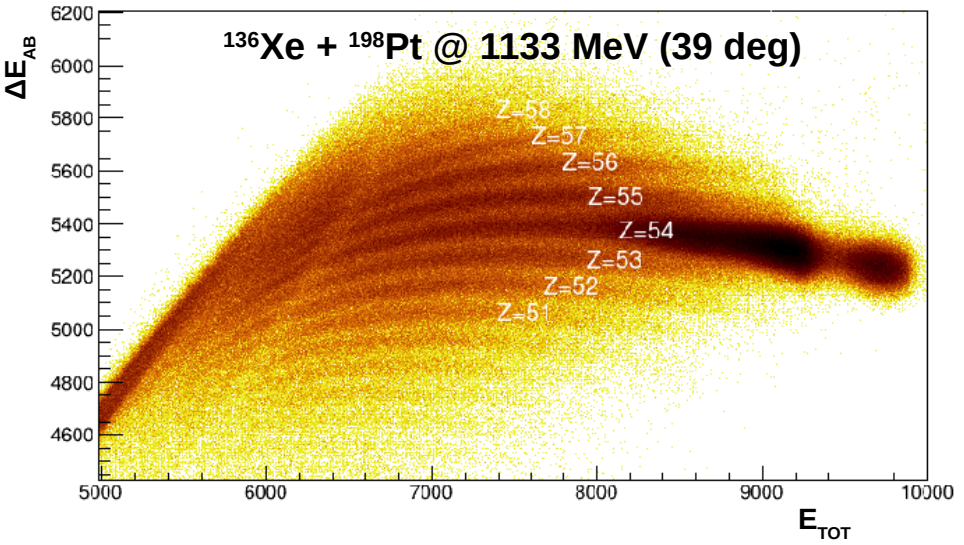
- Z_Nr
- z_ok

Z - identification (ΔE – E)









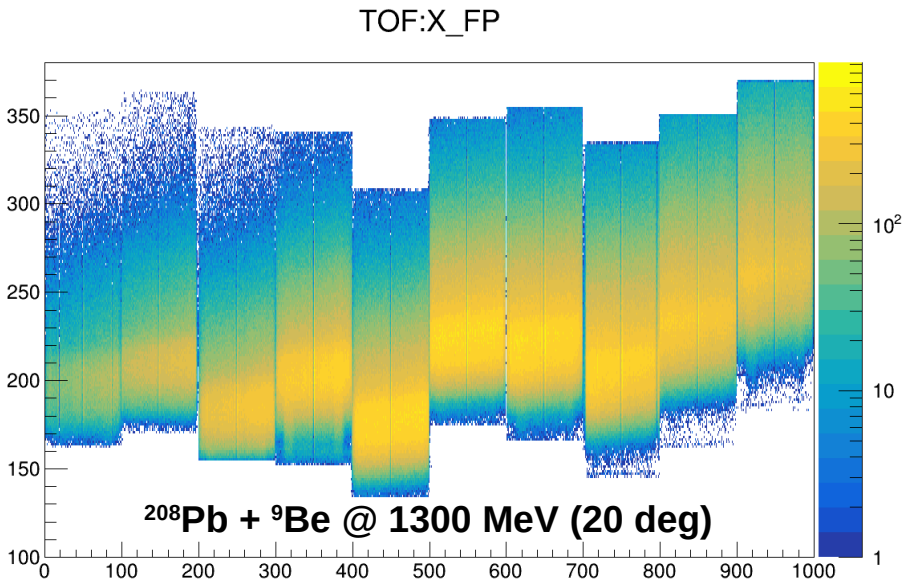
ToF alignment and ToF offset

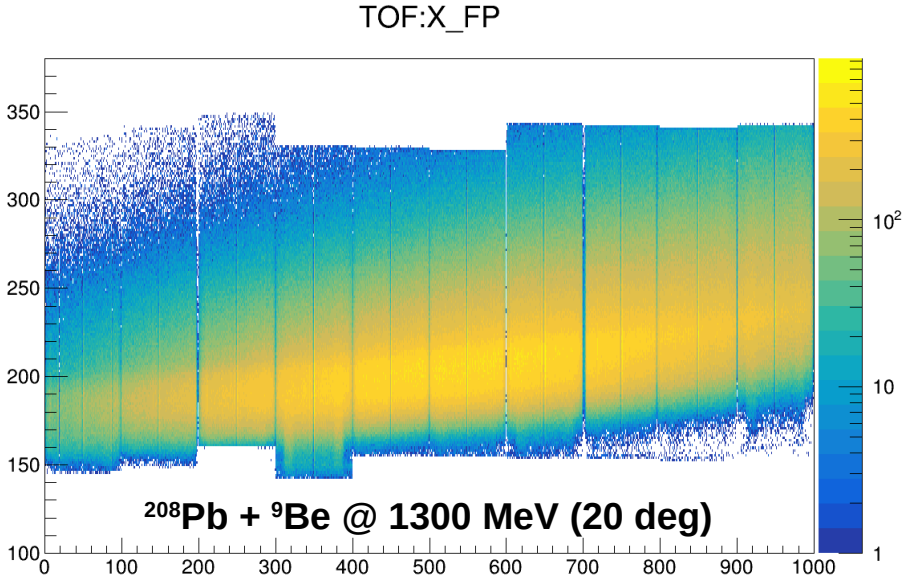
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

cal/alignment-ns.cal

```
1    0    2   -53.9    1
1    1    2   -54.2    1
1    2    2   -42.1    1
1    3    2   -40.2    1
1    4    2   -39.0    1
1    5    2   -41.8    1
1    6    2   -32.1    1
1    7    2   -29.6    1
1    8    2   -29.6    1
1    9    2   -29.6    1
```

Section-by-section offsets

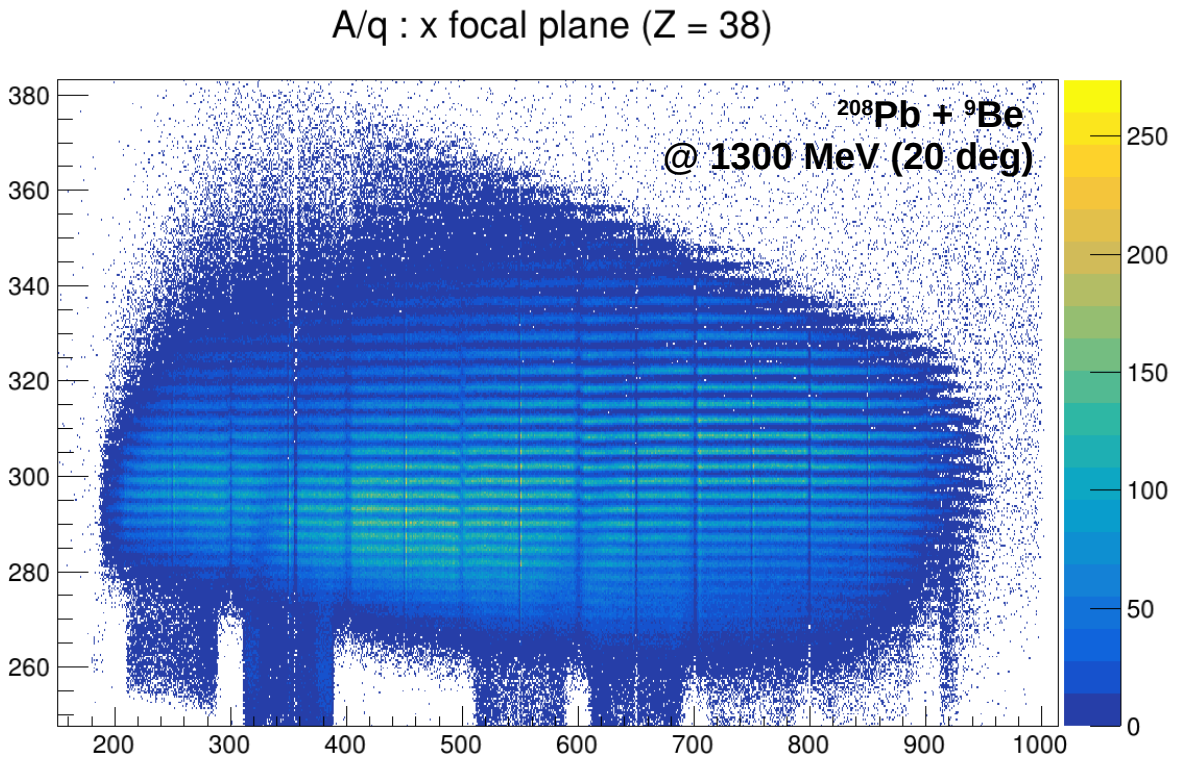




ToF alignment and ToF offset

$$\frac{A}{q} = \frac{B\rho}{L} \cdot ToF \cdot const$$

For fine tuning of ToF alignment use A/q vs X_FP matrix (gated on Z)



ToF offset

- You can use the variable `tof_offs` in the file `ppac.conf` to move all sections together
- For a first estimate of ToF, consider a flight path inside the Prisma spectrometer ~ 6 m long
- Fine tuning has to be done using the Doppler correction from known gamma-ray transitions

Optimization of optical parameters

solver.conf

```
target_mcp_distance = 250.
angle_mcp = 135.

quad_radius = 157.
geom_radius = 150.
#quad_length=420 is nominal
quad_length = 400.
dipole_entrance_angle = 20.
#dipole_exit_angle=125 is nominal
dipole_exit_angle = 125.
#target_quad_distance=500 is nominal
target_quad_distance = 500.

target_dipole_distance = 1600.
dipole_height = 200.

dip_fp_dist = 3285.0

fp_half_length = 500.

guess_radius = 1200.;
tolerance = 1.

prisma_dipole = 0.542190
prisma_quadrupole = 0.492521
#field_ratio = 0.953096

...
```

Many parameters can be optimized

Magnetic field intensities in Tesla

Optimization of optical parameters

Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

solver.conf

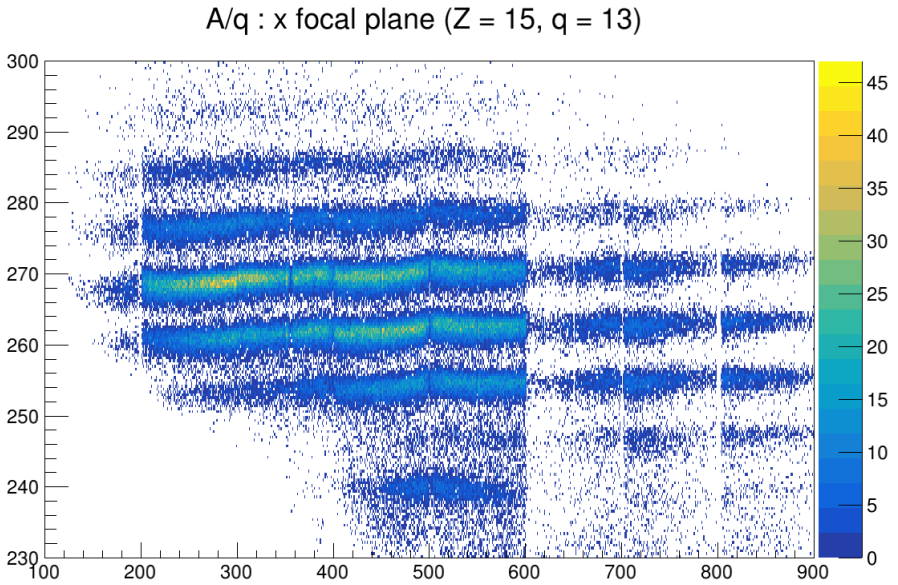
```
...

ic_fp_distance = 720.0
ic_depth = 250.0
ic_width = 100.0

ic_threshold = 10
ic_veto_threshold = 10
ic_up_thresh = 4090
ic_sections = 4

#default value
angular_tolerance = 1.3

#Possibility to correct the path and not the A/Q value itself
xmc_file = cal/xmcTrajempty.cal
ymc_file = cal/ymcTrajempty.cal
xfp_file = cal/xfpTrajempty.cal
```

Should not be changed

Analyzed variables

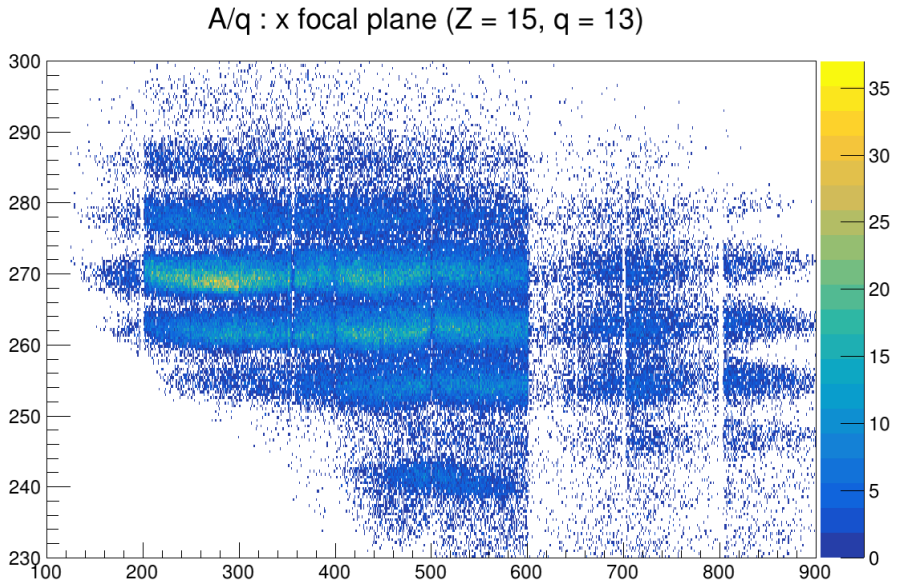
- Beta
- Length
- Radius
- RBeta
- A_over_q_uncal
- traj_ok

Optimization of optical parameters



$^{32}S + ^{208}Pb$ @ 230 MeV (51 deg)

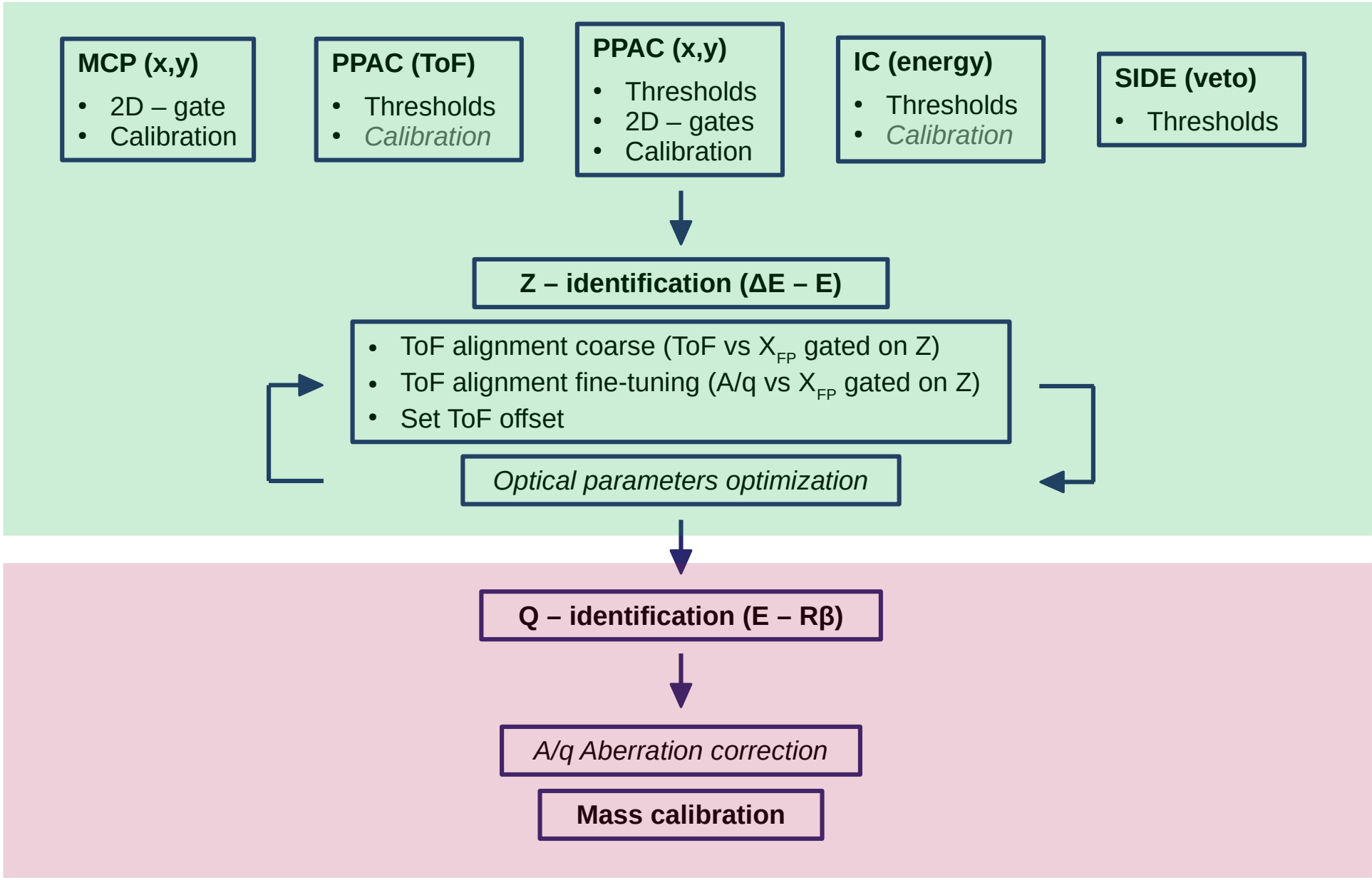
GOOD



$^{32}S + ^{208}Pb$ @ 230 MeV (51 deg)

BAD

Steps of the analysis - update



Charge and mass identification

mass.conf

```
z_min = 48
z_max = 58

q_min = 39
q_max = 44

a_min = 120
a_max = 150

ban_file_base = ban/charge_ban_
mas_file_base = ban/mas_ban_
mass_number_from_banana = 0

ban_res_x = 10000
ban_res_y = 10000

cal_file = cal/a_over_q.cal

xmc_file = cal/xmcempty.cal
xmc_spli = cal/xmcempty.spli
xmc_uspl = 1

ymc_file = cal/ymcempty.cal
ymc_spli = cal/ymcempty.spli
ymc_uspl = 1

...
```

Example:
`ban/charge_ban_16_14.ban`
`ban/mas_ban_16_032.ban`

A/q calibration file

A/q aberration corrections against MCP coordinates

Charge and mass identification

mass.conf

```
...

yfp_file = cal/yfpempty.cal
yfp_spli = cal/yfpempty.spli
yfp_uspl = 0

xfp_file = cal/xfpempty.cal
xfp_spli = cal/xfpempty.spli
xfp_uspl = 1

chg_file_base = cal/charge_cal_

a_over_q_gain = 1.0
a_over_q_offs = 0.0

xmc_gain = 1.0
xmc_offs = 0.0
ymc_gain = 1.0
ymc_offs = 0.0

xfp_gain = 1.0
xfp_offs = 0.0

r_fact = 1.0
e_fact = 1.0

B_dipole = 0.829976
```

A/q aberration corrections against PPAC coordinates

Charge state “re-calibration” file

Magnetic field intensity

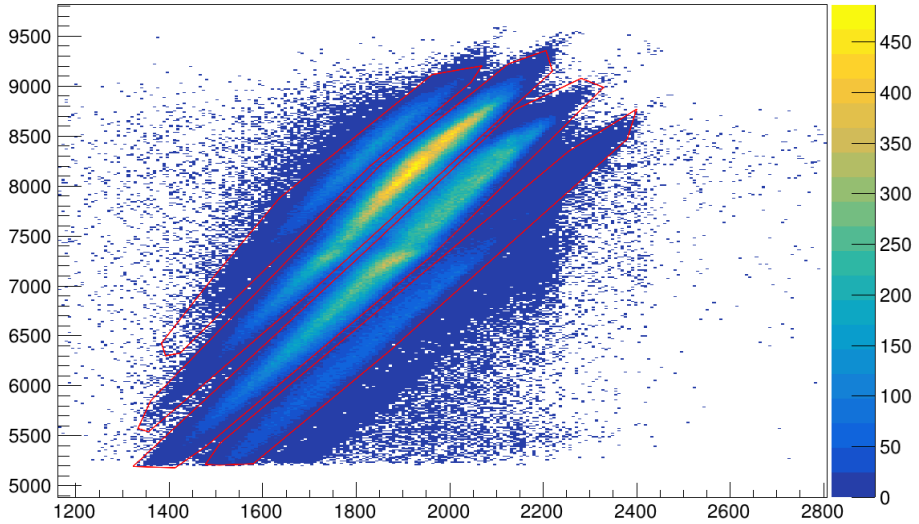
Analyzed variables

- A_over_q
- Mass
- Q_Nr
- A_Nr
- q_ok
- a_ok

Q – identification (E – Rβ)

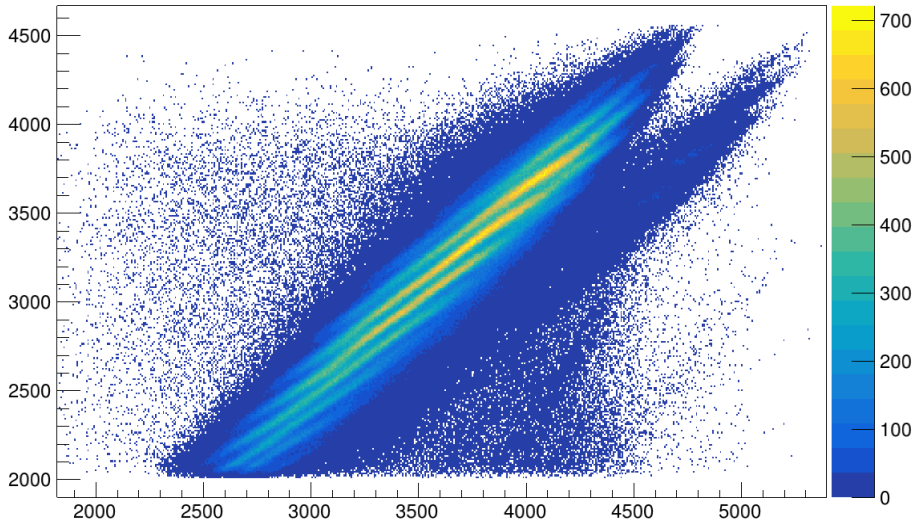
$$B\rho = \frac{p}{q} \rightarrow B\rho \cdot v = \frac{mv^2}{q} \rightarrow E = q \cdot \rho\beta \cdot \frac{Bc}{2}$$

IC E : R*Beta (Z = 15)



^{36}S + ^{208}Pb @ 250 MeV (51 deg)

IC E : R*Beta (Z = 37)



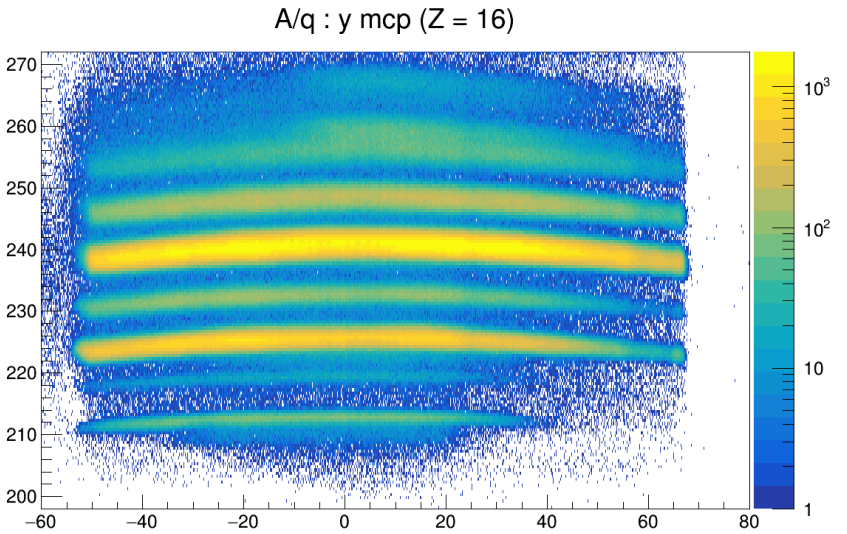
^{208}Pb + ^{9}Be @ 1300 MeV (20 deg)

To be done for each combination of Z and Q

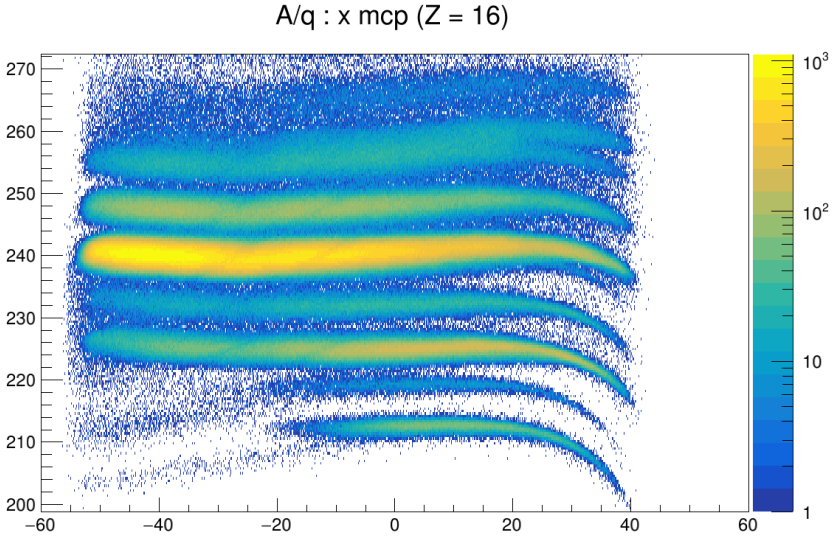
Aberration corrections

What are aberrations?

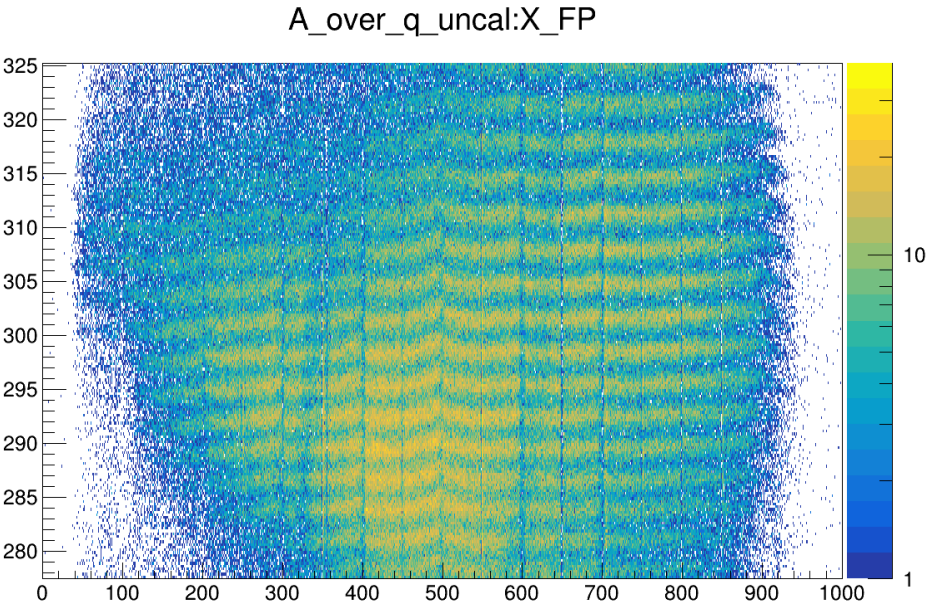
- They are <u>systematic effects</u> that concern the trajectory reconstruction and would decrease the mass resolution of the spectrometer
- They happen because the magnetic field geometry in the algorithm is approximated by considering ideal dipoles and quadrupoles, but <u>can be corrected</u>



^{32}S + ^{124}Sn @ 160 MeV (55 deg)

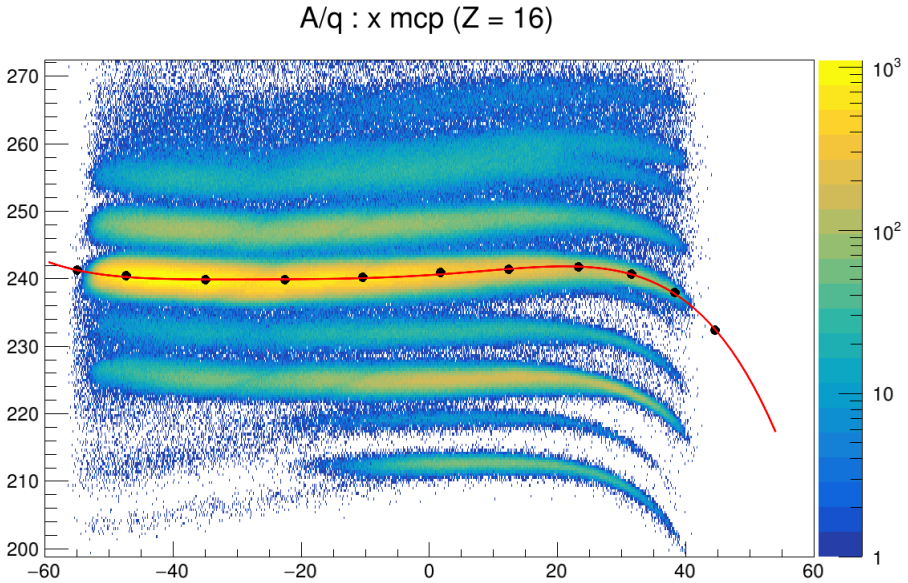


^{32}S + ^{124}Sn @ 160 MeV (55 deg)

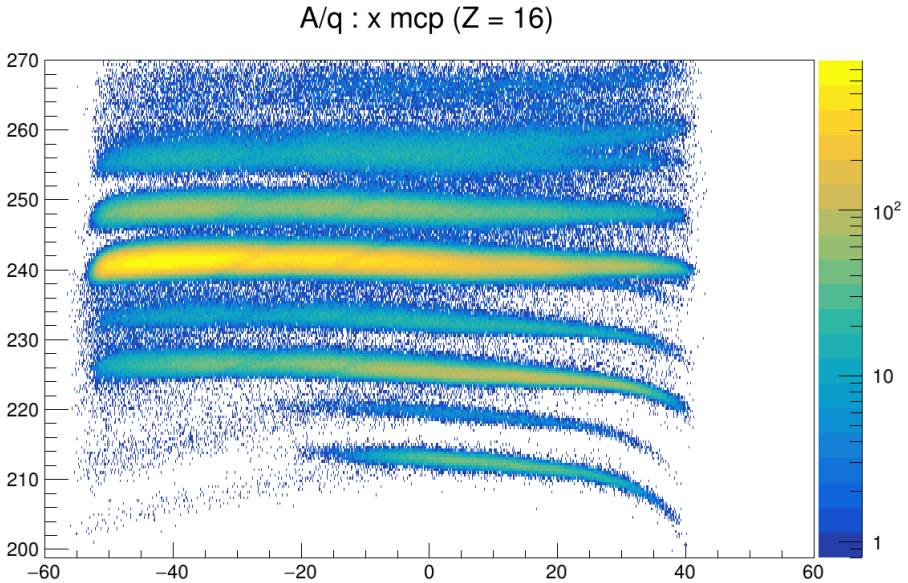


^{208}Pb + ^{9}Be @ 1300 MeV (20 deg)

Aberration corrections



^{32}S + ^{124}Sn @ 160 MeV (55 deg)



^{32}S + ^{124}Sn @ 160 MeV (55 deg)

- The function is subtracted from A/q
- Can be **polynomial** or **cubic spline**

`cal/xmc.cal`

```
...
99   14   6   0.   0.07992    0.0002788    -5.80e-05    -8.90e-07     -5.92e-09
99   15   6   0.   0.07992    0.0002788    -5.80e-05    -8.90e-07     -5.92e-09
99   16   6   0.   0.07992    0.0002788    -5.80e-05    -8.90e-07     -5.92e-09
```

For each Z

Aberration corrections

cal/xmc.spli

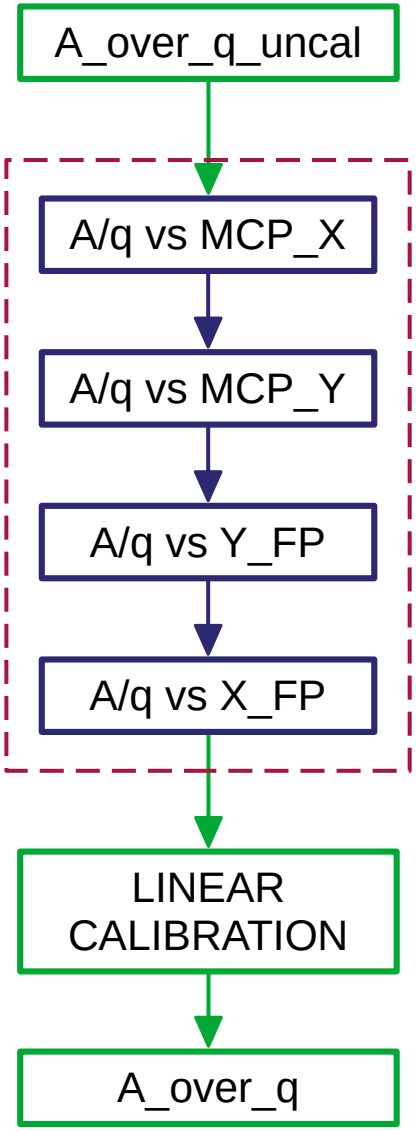
```
ADC 16
-80.0      0.0
-40.0      0.0
  0.0      0.0
 40.0      0.0
 80.0      0.0
ADC 15
...
```

→ Z_Nr

cal/xfp.cal

```
16   0    2    0.   0.
16   1    2    0.   0.
16   2    2    0.   0.
16   3    2    0.   0.
16   4    2    0.   0.
16   5    2    0.   0.
16   6    2    0.   0.
16   7    2    0.   0.
16   8    2    0.   0.
16   9    2    0.   0.
#
15   0    2    0.   0.
15   1    2    0.   0.
15   2    2    0.   0.
15   3    2    0.   0.
...
```

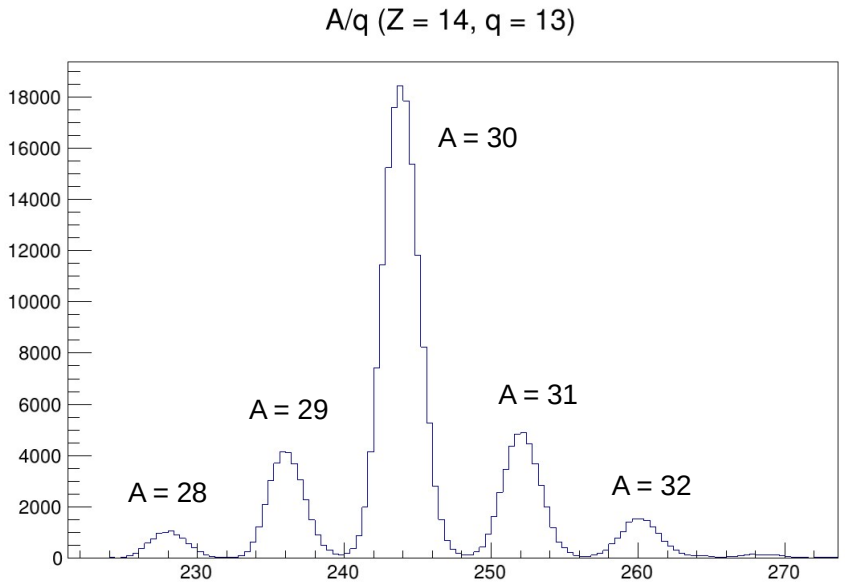
For X_FP the polynomial correction is done section by section



ORDER IS VERY IMPORTANT

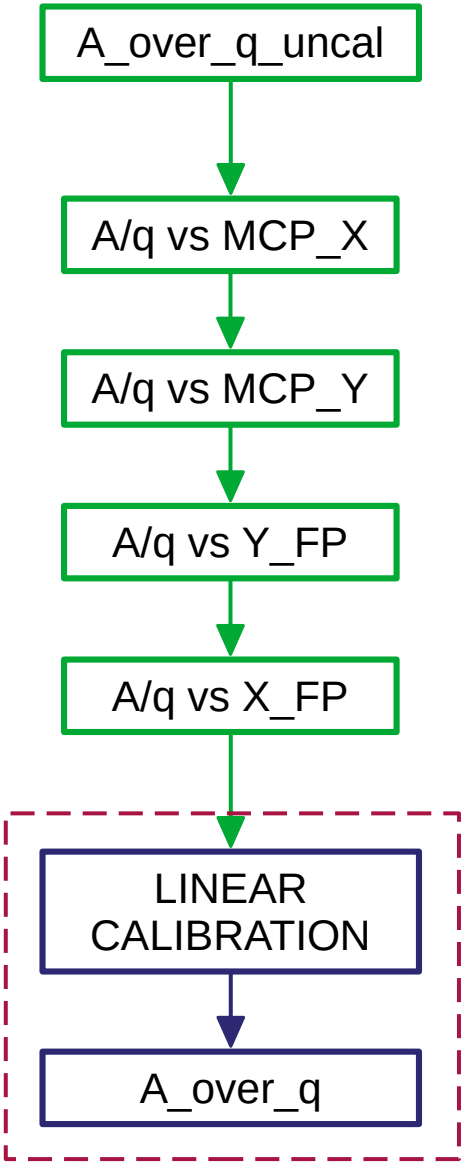
Mass calibration

Linear calibration is performed to match the centroids of the peaks in the A/q spectrum (gated in Z and Q) to the expected A/q values.



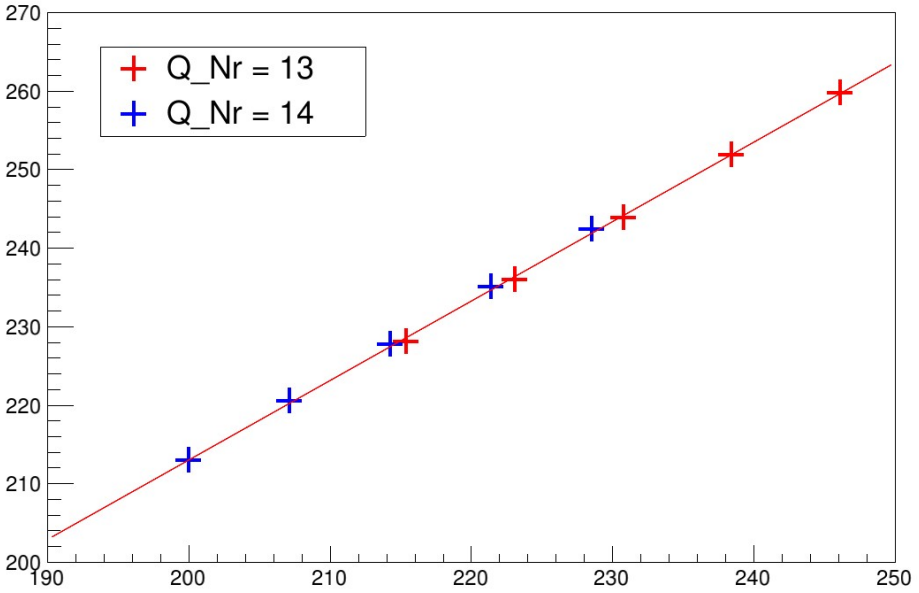
Example

Expected	Observed
28 / 13 x 100 = 215.385	227.9
29 / 13 x 100 = 223.077	236.1
30 / 13 x 100 = 230.769	243.8
31 / 13 x 100 = 238.462	252.3
32 / 13 x 100 = 246.154	260.1



ORDER IS VERY IMPORTANT

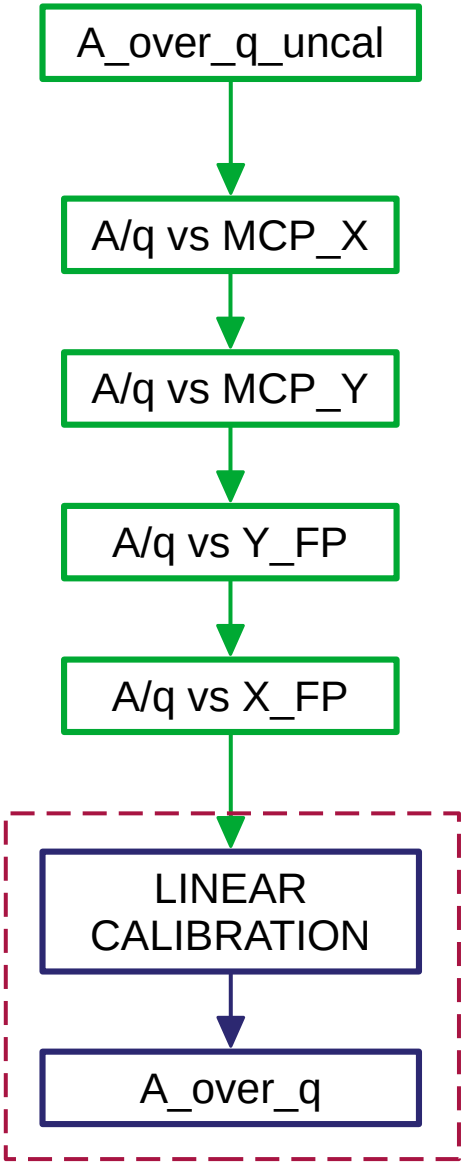
Mass calibration



cal/a_over_q.cal

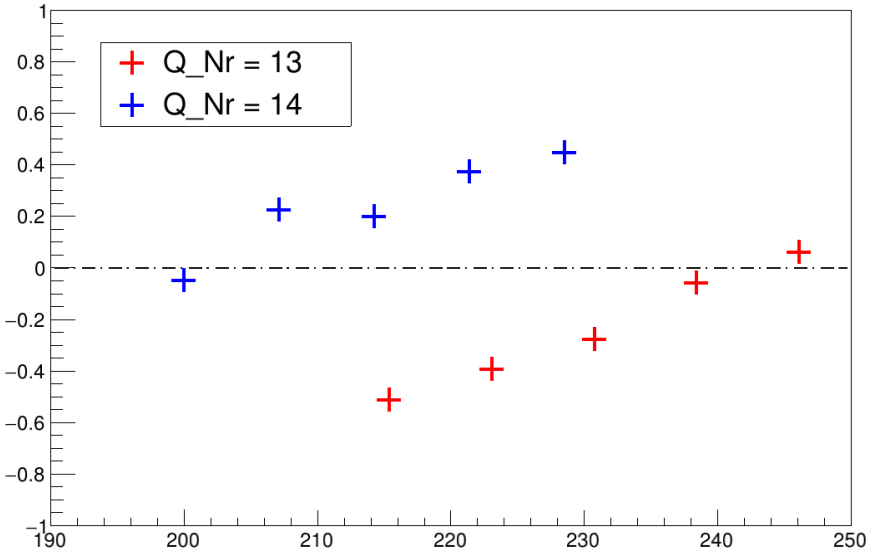
```
1   16    2    -7.96874335    1.120640853
1   15    2    -6.3795417     1.112781635
1   14    2    -6.427071018   1.112698609
```

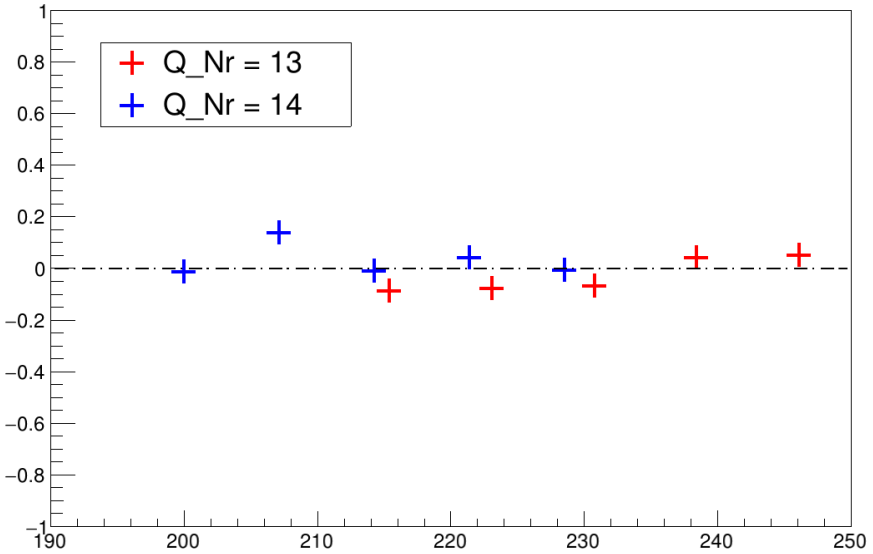
For each Z



ORDER IS VERY IMPORTANT

Mass calibration





$$mass = \left(\frac{A}{q}\right)_{cal} \cdot q_{eff}$$

cal/charge_cal_16.cal

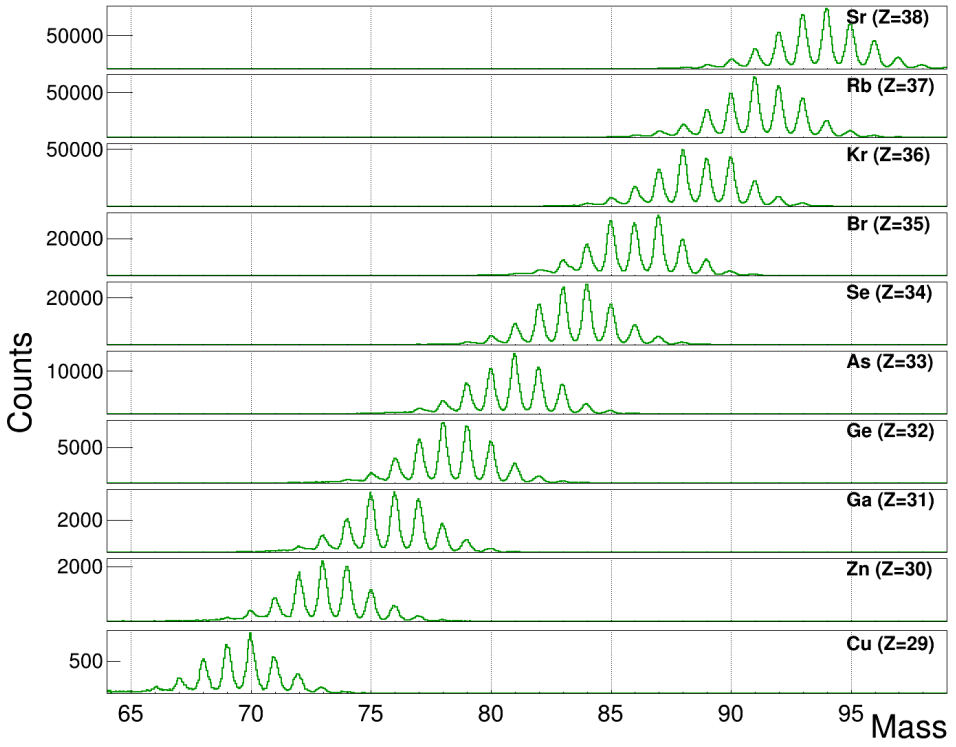
```
99    12    2    11.95475   0.
99    13    2    12.92442   0.
99    14    2    13.95171   0.
99    15    2    14.93272   0.
99    16    2    15.95846   0.
```

For each Z and Q

Used for fine tuning

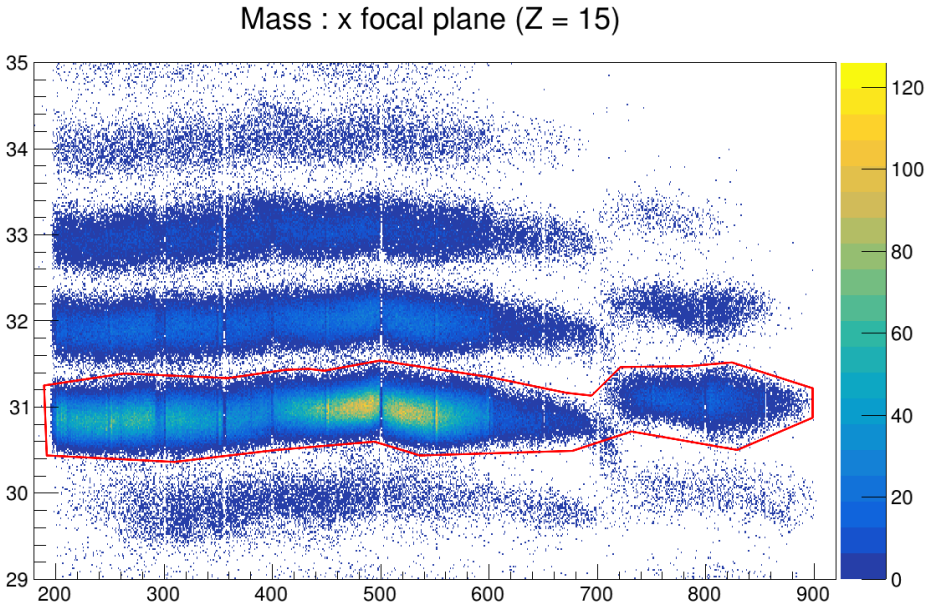
Mass calibration

Final mass spectrum



^{208}Pb + ^{9}Be @ 1300 MeV (20 deg)

A_Nr is calculated either by rounding to the nearest integer or by applying a 2D-gate in the Z-gated Mass vs X_FP matrix



^{32}S + ^{124}Sn @ 160 MeV (55 deg)

Binary partner

binarypartner.conf

```
#Setup for beam and Target
#it takes the calculated Z and A of the prismalibrary to calculate the binary
partner and the reaction
z_beam = 16
a_beam = 32

z_target =  50
a_target =  124

#Thickness of Target in mg/cm^2
target_thickness = 0.500

# 0   - At the beginning of the target
# 0.5 - At the middle
# 1   - At the exit
reaction_place = 0.5

#Beam Energy in MeV (NOT PER NUCLEON!)
beam_energy = 160

excitation_projectile = 0.0
excitation_target = 0.0

#Angle between target and beam in Degree
theta_target = 35.0

#Angle of Prisma in degrees
angle_prisma = 55.0

#Atomic mass file http://www.nndc.bnl.gov/amdc/masstables/ 2020 format
mass_table_file = cal/massnumbers2020.cal
```

Suggestion:
Just use the kinematic reconstruction built in the **agataselector** instead of this

Analyzed variables

- Qvalue
- Theta_BP
- Phi_BP
- Beta_BP

Steps of the analysis - update

MCP (x,y)
- 2D – gate
- Calibration

PPAC (ToF)
- Thresholds
- *Calibration*

PPAC (x,y)
- Thresholds
- 2D – gates
- Calibration

IC (energy)
- Thresholds
- *Calibration*

SIDE (veto)
- Thresholds

↓

Z – identification (ΔE – E)

- ToF alignment coarse (ToF vs X_{FP} gated on Z)
- ToF alignment fine-tuning (A/q vs X_{FP} gated on Z)
- Set ToF offset

Optical parameters optimization

↓

Q – identification (E – Rβ)

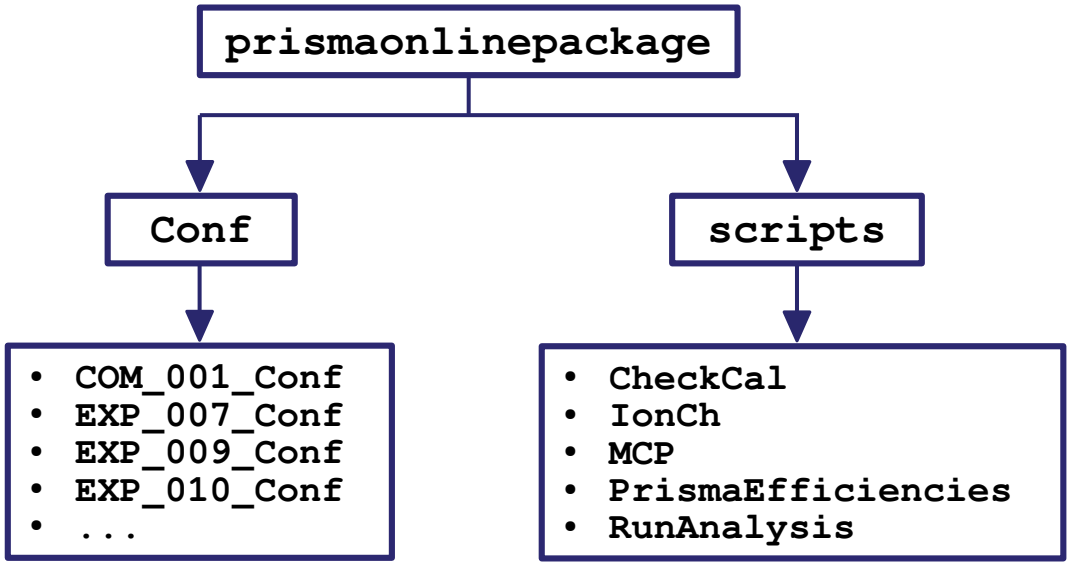
↓

A/q Aberration correction

Mass calibration

Analysis tools – the PrismaOnlinePackage

```
git clone https://baltig.infn.it/prisma/prismaonlinepackage.git
```



Main folder

Subfolders

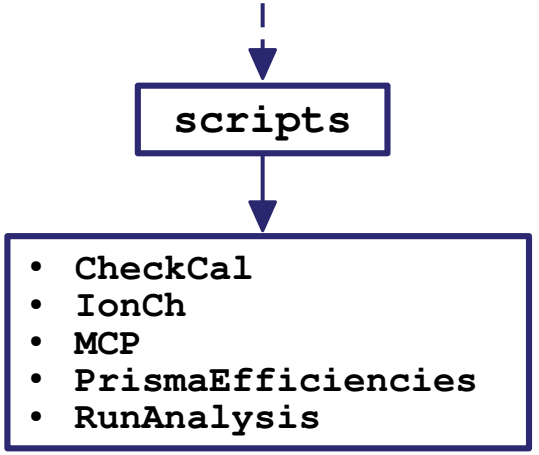
Online analysis configuration folders

Can be used as an example and as starting point

Scripts used for online analysis

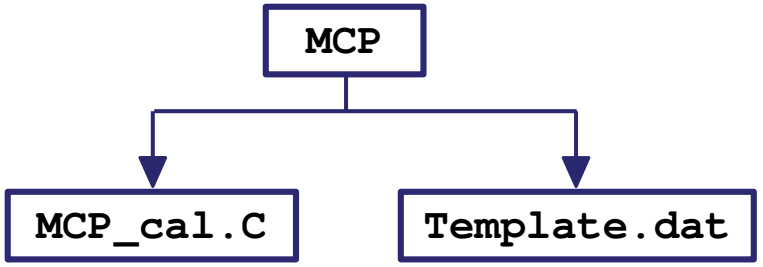
See next slides ...

Analysis tools – the PrismaOnlinePackage



- **`CheckCal`** → Visually check most calibrations and thresholds
- ***`IonCh`*** → *Increase the quality of the IC calibration (only use if necessary)*
- **`MCP`** → Perform MCP calibration
- ***`PrismaEfficiencies`*** → *Evaluate efficiency of some elements of the Prisma spectrometer*
- **`RunAnalysis`** → Run PrismaFilters with basic multi-thread *+ run agataselector*

MCP



Template.dat

pos	x_mm	y_mm	x_chan	y_chan
center	0	0	1500	2500
tl_cal	-21.5	26.5	2000	3000
tr_cal	21.5	26.5	1000	3000
bl_cal	-21.5	-26.5	2000	2000
br_cal	21.5	-26.5	1000	2000

Run with:

```
$ root -l
$ .L MCP_cal.C
$ MCP_cal("filename")
```

The routine implements a Minuit2 minimizer from ROOT to minimize the quadratic sum of the distance between reference and calibrated points.

MCP

MCP_cal.C

```
...
void calibration::process(){
  ...
  min->SetVariable(0,"x0",1.00, 1e-6);
  min->SetVariable(1,"x1",0.01, 1e-6);
  min->SetVariable(2,"y0",0.01, 1e-6);
  min->SetVariable(3,"y1",1.00, 1e-6);
  min->SetVariable(4,"calxa",50, 1e-6);
  min->SetVariable(5,"calxb",1, 1e-6);
  min->SetVariable(6,"calxc",0.01, 1e-6);
  min->SetVariable(7,"calya",50, 1e-6);
  min->SetVariable(8,"calyb",1, 1e-3);
  min->SetVariable(9,"angle",0.01, 1e-6);//in degrees
  min->SetVariableLimits(9,-10, 10);//in degrees

  //min->FixVariable(0);
  //min->FixVariable(1);
  //min->FixVariable(2);
  //min->FixVariable(3);
  //min->FixVariable(4);
  //min->FixVariable(5);
  //min->FixVariable(6);
  //min->FixVariable(7);
  //min->FixVariable(8);
  //min->FixVariable(9);
  min->Minimize();

  for(int i=0; i<10; i++) par.push_back(min->X()[i]);
}
```

Here to toggle which parameters to use in the minimization

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x_0 & x_1 \\ y_0 & y_1 \end{pmatrix} \begin{pmatrix} x_{raw} \\ y_{raw} \end{pmatrix}$$

$$\downarrow$$

$$\begin{cases} x'' = a + bx' + c\,(x')^2 \\ y'' = d + ey' \end{cases}$$

$$\downarrow$$

$$\begin{pmatrix} x_f \\ y_f \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x'' \\ y'' \end{pmatrix}$$

MCP

Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

Output:

```
...
Nfcn  = 1479
x0    = 0.712126  +/-  0.000116715
x1    = -0.0403526    +/-  0.000241851
y0    = -0.0889732    +/-  0.0576092
y1    = -2.38539  +/-  0.00796766
calxa     = 58.3466   +/-  7.1939
calxb     = -0.0603181    +/-  0.0165054
calxc     = -3.72841e-10  +/-  8.51033e-06
calya     = -135.322  +/-  3.14078
calyb     = -0.0221951    +/-  0.000418261
angle     = 2.63223   +/-  1.08814    (limited)

Calibrated reference points:
5.22987e-05  7.87512e-06
-21.5        26.5
21.5         26.5
-21.5        -26.5
21.5         -26.5

Calibration parameters:
sum dist sq: 3.54675e-09
|x0 x1|  |0.712126    -0.0403526|
|y0 y1|  |-0.0889732  -2.38539  |
cal x: 99 0 3    58.3466  -0.0603181   -3.72841e-10
cal y: 99 0 2    -135.322 -0.0221951
angle: 2.63223
```

CheckCal

Load with:

```
$ root -l
$ .L CheckCal.C
```

Suggestion: check directly the section labeled "**`For the USER`**" in the code where the fuctions are defined if you need to use it.

`TCutToBan`

```
//--------------------------------------------------------------------------------|
//For the USER
void TCutToBan(string cutFileName, string banFileName);
//example: TCutToBan("cutName.root","banName.ban")
...
```

IMPORTANT: cut title and cut filename have to be the same.

OBS: Can be automatized with a bash script

```
root -l << EOF
    .L CheckCal.C
    TCutToBan("z16.root","PrismaConf/ban/zed_ban_deab_16.ban")
    TCutToBan("z15.root","PrismaConf/ban/zed_ban_deab_15.ban")
    TCutToBan("z14.root","PrismaConf/ban/zed_ban_deab_14.ban")
    TCutToBan("z13.root","PrismaConf/ban/zed_ban_deab_13.ban")
    TCutToBan("z12.root","PrismaConf/ban/zed_ban_deab_12.ban")
    TCutToBan("z11.root","PrismaConf/ban/zed_ban_deab_11.ban")
EOF
```

*Example is located in **CheckCal** folder*

CheckCal

Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

TCutToSpli

```
...
void TCutToSpli(string cutFileName, string spliFileName, double val);
//example: TcutToSpli("cutName.root","spliName.spli",250)
...
```

```
...
void TCutToSpli( std::vector<string> cutFileName,
                 string spliFileName,
                 std::vector<double> val);

//example:
TCutToSpli(  {"cutName1.root","cutName2.root","cutName3.root"},
             "spliName.spli",
             {250,260,270})
...
```

Second version is used to sum multiple spline files

Good for iterative corrections

CheckCal

Draw functions

```
void DrawPPAC_XrCal (string inFileName,
                     string calFileName,
                     bool selFlag = true,
                     int nrEvts = -1);

//example: DrawPPAC_XrCal("treeName.root","calName.cal",false,1e6)
//example: DrawPPAC_XrCal("outName.root","calName.cal")
```

```
void DrawCharge (string inFileName,
                 string banFilePattern,
                 int Z, int qmin, int qmax,
                 bool selFlag = true,
                 int nrEvts = -1);

//example: DrawCharge("treeName.root","banPatt_",16,13,15,false,1e6)
//example: DrawCharge("outName.root","banPatt_",16,13,15)
```

Repeating-ish structure

CheckCal

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

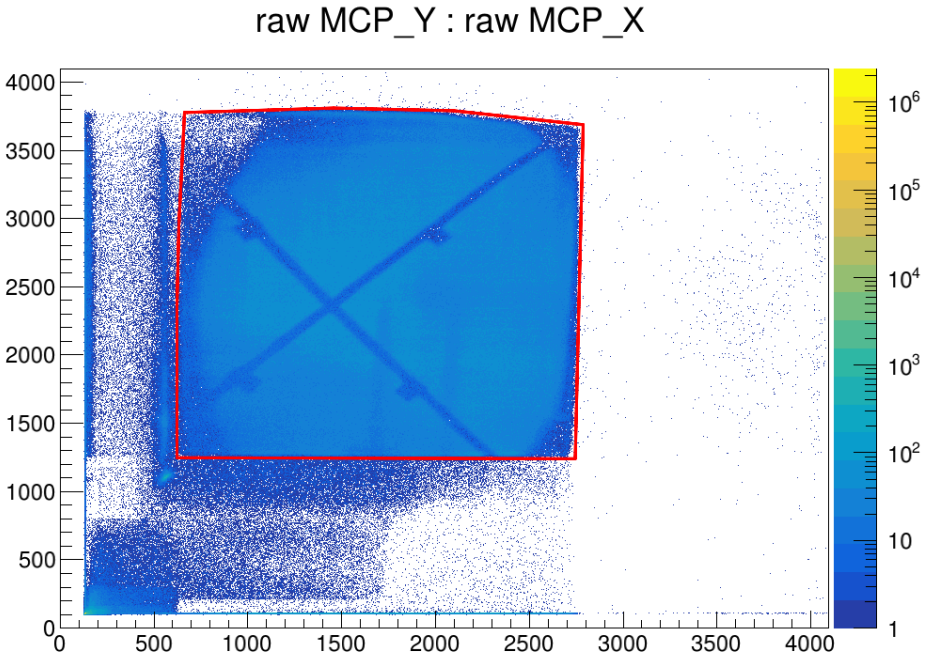
DrawAll

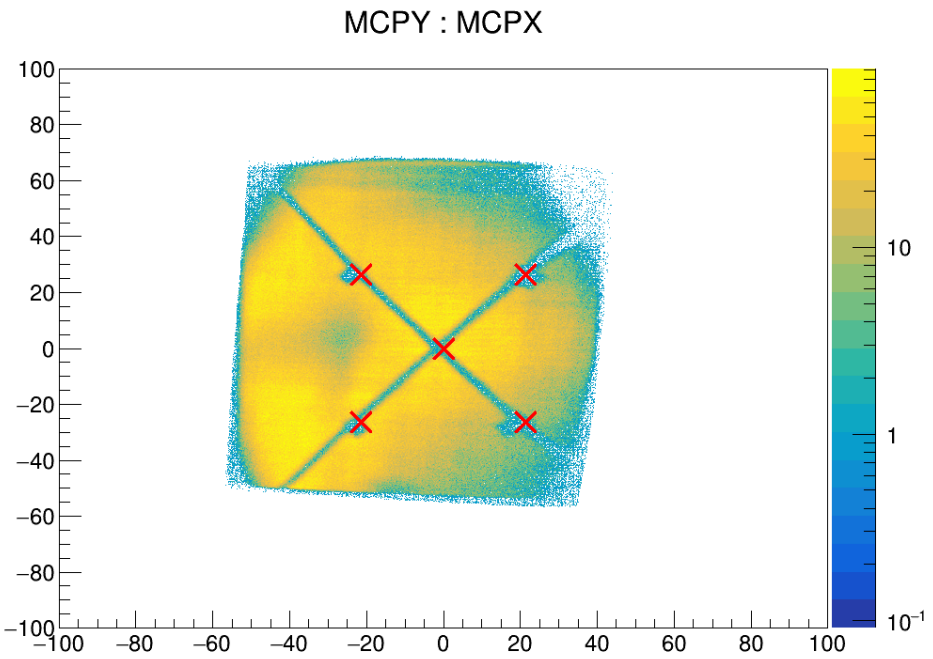
```
const string defTreeName =    "Runs/run_0000/Tree_0000.root";
const string defOutName =     "Out/sum-0_1.root";

void DrawAll(string inTreeName = defTreeName, string inOutName = defOutName);
```

```
void DrawAll(string inTreeName = defTreeName, string inOutName = defOutName){
     //DrawMCPraw          (inOutName,"PrismaConf/ban/mcp_banana.ban");
     //DrawMCPana          (inOutName);
     //DrawPPAC_Right      (inTreeName,"PrismaConf/threshold/x_right.thres",1e6);
     //DrawPPAC_Left       (inTreeName,"PrismaConf/threshold/x_left.thres",1e6);
     //DrawPPAC_Cath       (inTreeName,"PrismaConf/threshold/x_cathode.thres",1e6);
     //DrawPPAC_Cuts       (inOutName,"PrismaConf/ban/Cath-L+R.ban");
     //DrawPPAC_XfpCal     (inTreeName,1e6);
     //DrawPPAC_XrCal      (inOutName,"PrismaConf/cal/right-cath.cal");
     //DrawPPAC_XlCal      (inOutName,"PrismaConf/cal/cath-left.cal");
     //DrawToFs            (inTreeName,"PrismaConf/threshold/tof.thres",1e6);
     //DrawSidePads        (inOutName,"PrismaConf/threshold/side_");
     //DrawIonChPads       (inOutName,"PrismaConf/threshold/IC_");
     //DrawZedAB           (inOutName,"PrismaConf/ban/zed_ban_deab_",12,16);
     //DrawCharge          (inOutName,"PrismaConf/ban/charge_51deg_ban_",16,12,16);
     //DrawCharge          (inOutName,"PrismaConf/ban/charge_51deg_ban_",15,12,15);
     //DrawCharge          (inOutName,"PrismaConf/ban/charge_55deg_ban_",15,12,15);
     //DrawCharge          (inOutName,"PrismaConf/ban/charge_55deg_ban_",14,11,14);
     //DrawCharge          (inOutName,"PrismaConf/ban/charge_55deg_ban_",13,11,13);
     //DrawCharge          (inOutName,"PrismaConf/ban/charge_55deg_ban_",12,10,12);
}
```

CheckCal

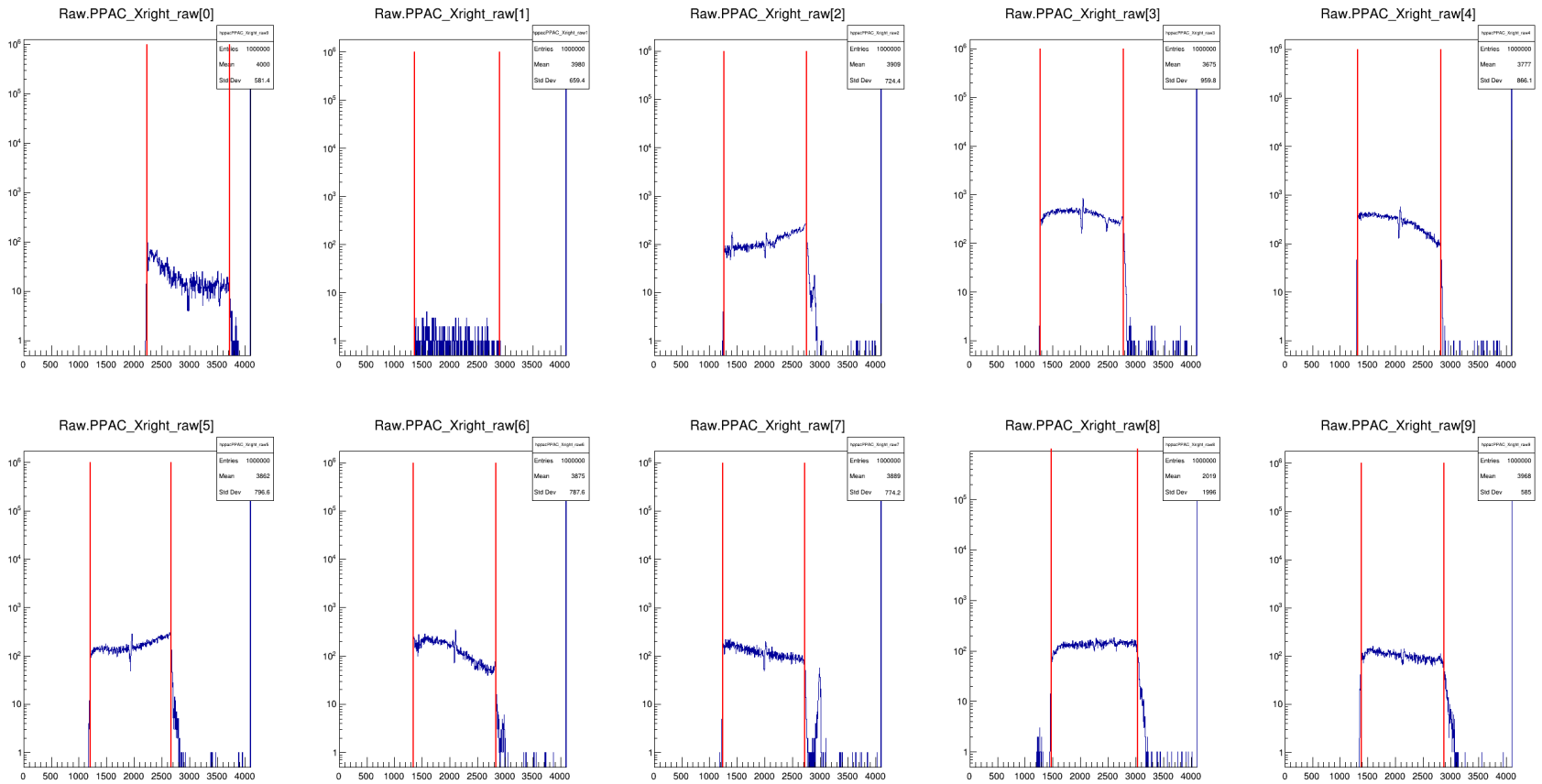
```
void DrawMCPraw(string inFileName, string banFileName, bool selFlag = true, int nrEvts = -1);
//example: DrawMCPraw("treeName.root","banName.ban",false,1e6)
//example: DrawMCPraw("outName.root","banName.ban")

void DrawMCPana(string inFileName, bool selFlag = true, int nrEvts = -1);
//example: DrawMCPana("treeName.root",false,1e6)
//example: DrawMCPana("outName.root")
```





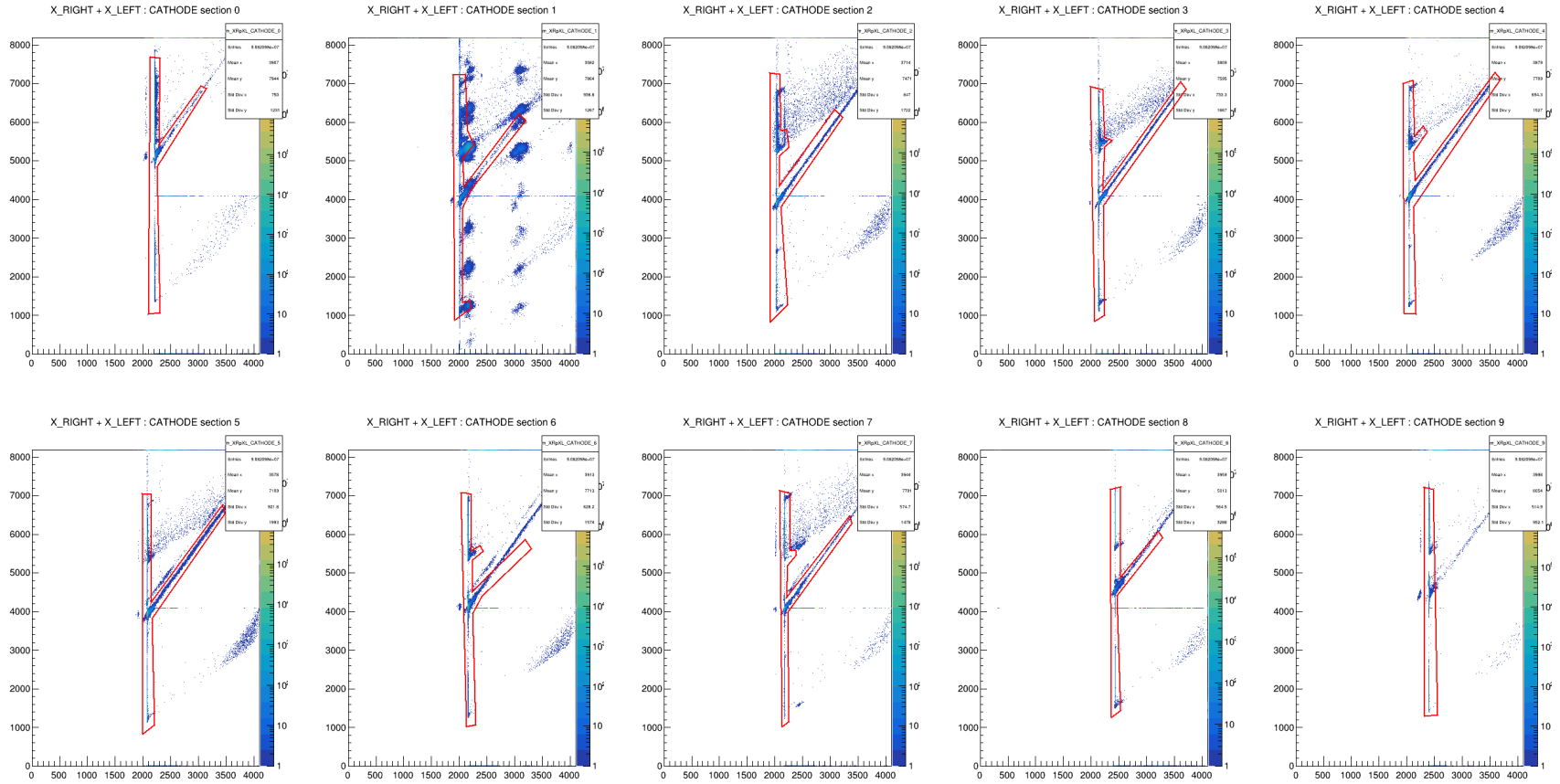
CheckCal

```
void DrawPPAC_Right(string inFileName, string thrFileName, int nrEvts = -1);
void DrawPPAC_Left (string inFileName, string thrFileName, int nrEvts = -1);
void DrawPPAC_Cath (string inFileName, string thrFileName, int nrEvts = -1);
//example: DrawPPAC_Right("treeName.root","thrName.thres",false,1e6)
```



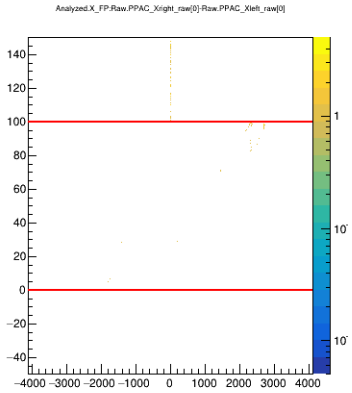
CheckCal

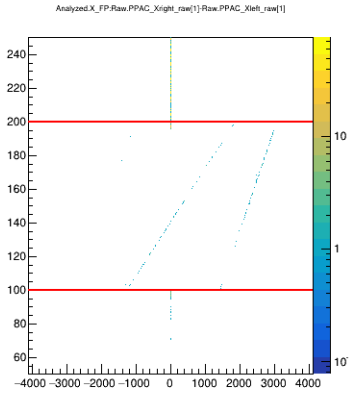
```
void DrawPPAC_Cuts (string inFileName, string banFileName, bool selFlag = true, int nrEvts = -1);
//example: DrawPPAC_Cuts("treeName.root","banName.ban",false,1e6)
//example: DrawPPAC_Cuts("outName.root","banName.ban")
```

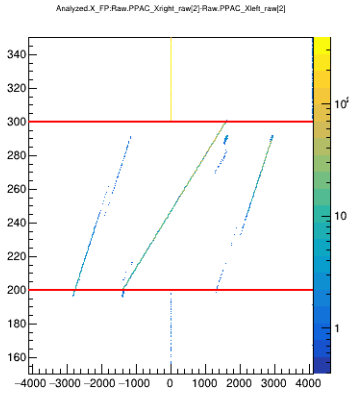


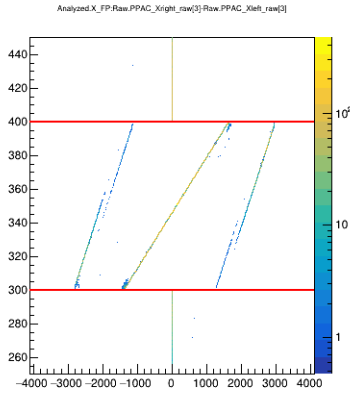
CheckCal

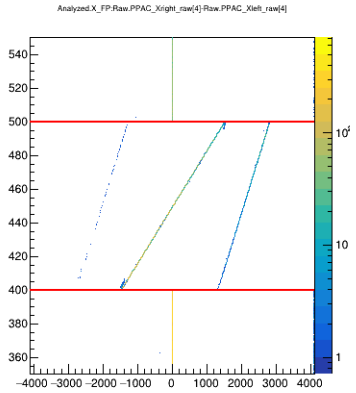
```
void DrawPPAC_XfpCal (string inFileName, int nrEvts = -1);
//example: DrawPPAC_Cuts("treeName.root",1e6)
```

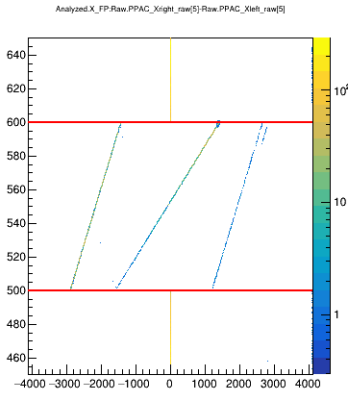


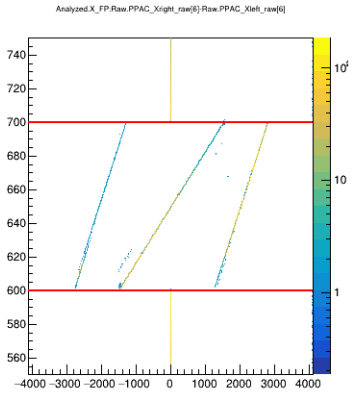


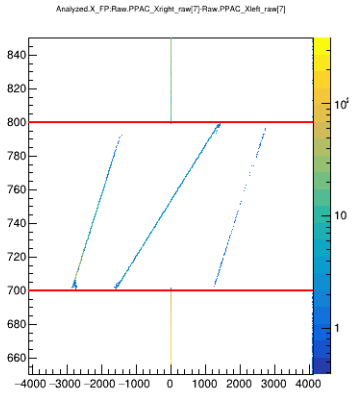


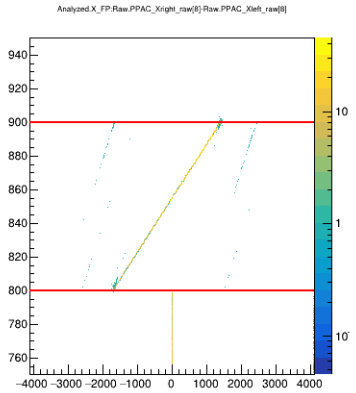


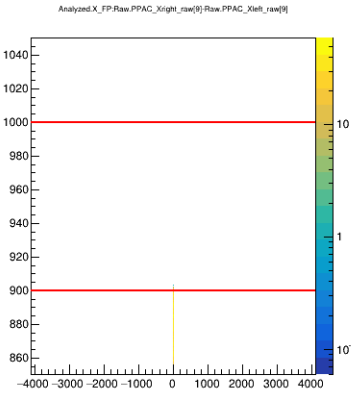






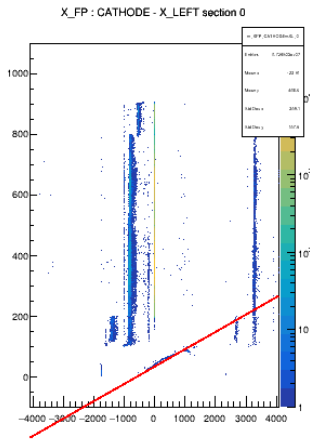


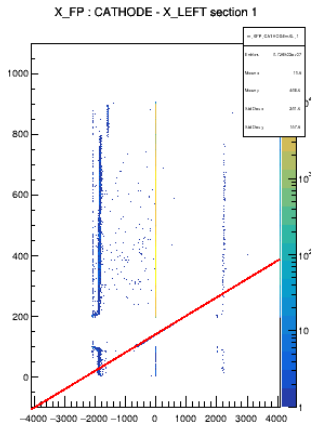


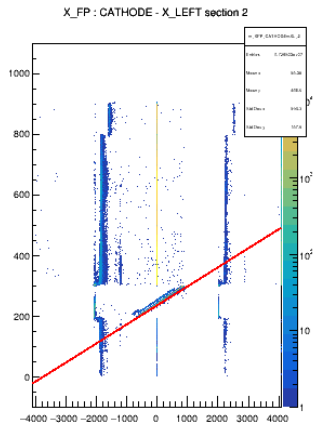


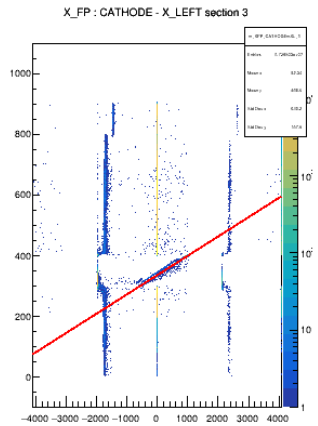
CheckCal

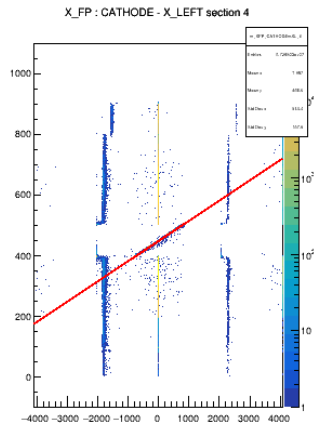
```
void DrawPPAC_XrCal (    string inFileName, string calFileName,
                         bool selFlag = true, int nrEvts = -1);
void DrawPPAC_XlCal (    string inFileName, string calFileName,
                         bool selFlag = true, int nrEvts = -1);
//example: DrawPPAC_XrCal("treeName.root","calName.cal",false,1e6)
//example: DrawPPAC_XrCal("outName.root","calName.cal")
```

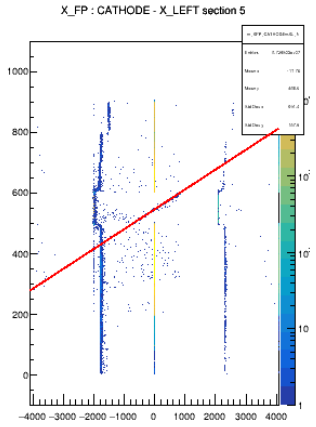


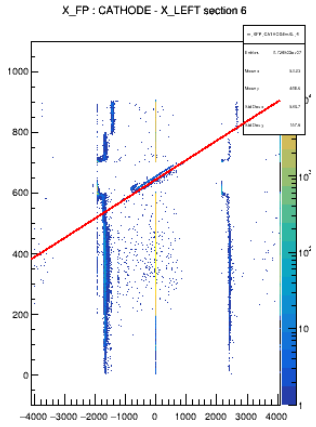


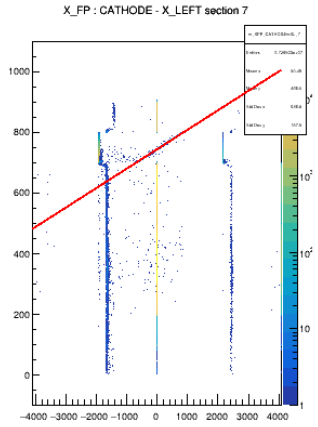


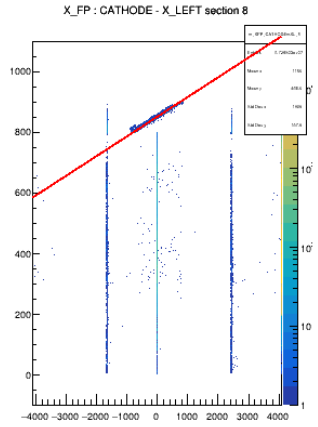


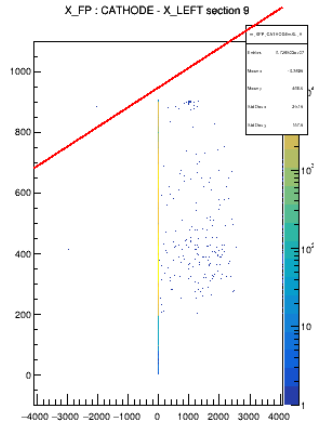






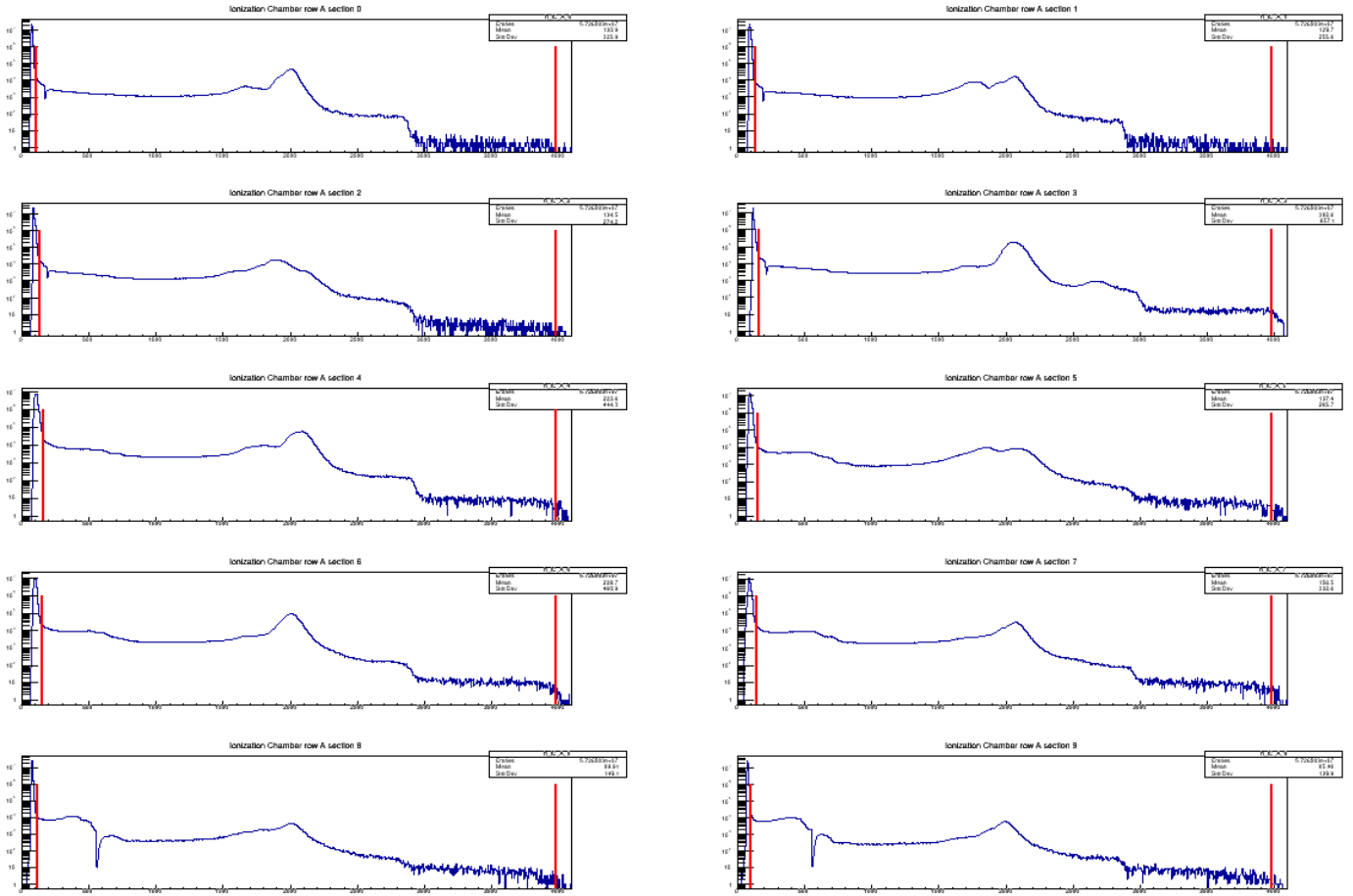






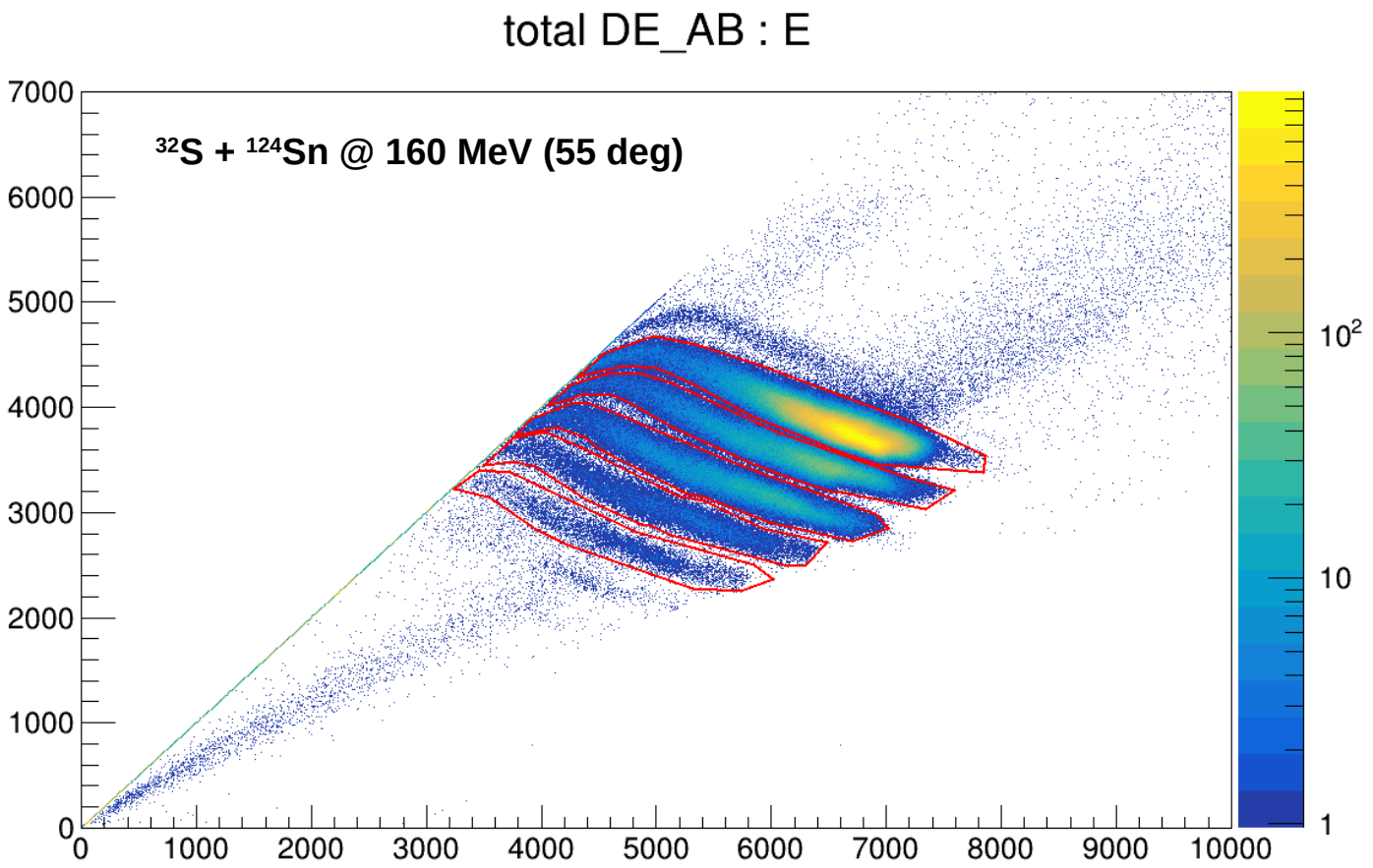
CheckCal

```
void DrawToFs (string inFileName, string thrFileName, int nrEvts = -1);
//example: DrawToFs("treeName.root","thrName.cal",1e6)

void DrawSidePads (string inFileName, string thrFilePattern, bool selFlag = true, int nrEvts = -1);
void DrawIonChPads (string inFileName, string thrFilePattern, bool selFlag = true, int nrEvts = -1);
//example: DrawSidePads("treeName.root","thrPatt_",false,1e6)
//example: DrawSidePads("outName.root","thrPatt_")
```



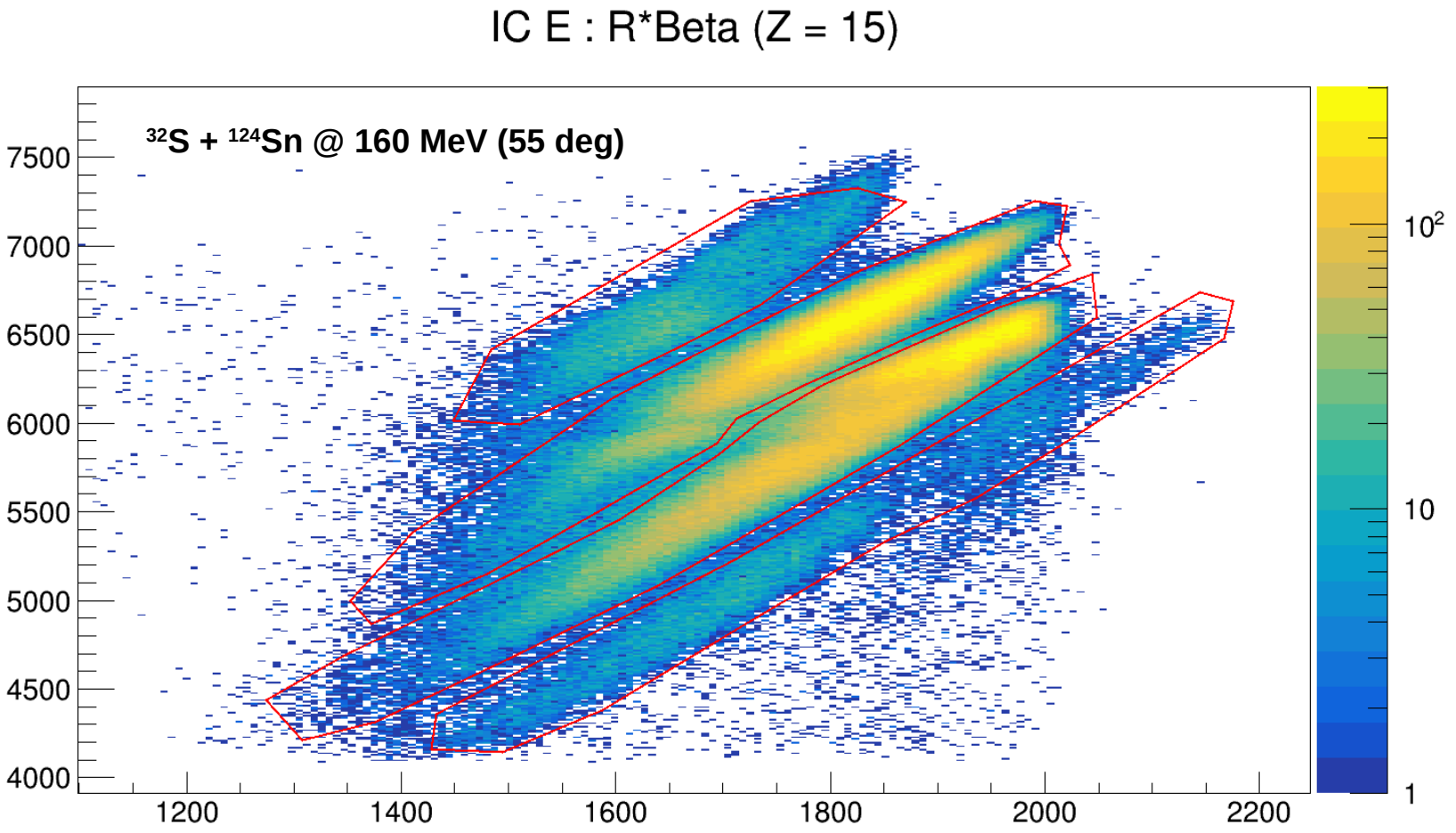
CheckCal

```
void DrawZedA  (    string inFileName, string banFilePattern,
                    int Zmin, int Zmax, bool selFlag = true, int nrEvts = -1);
void DrawZedAB (    string inFileName, string banFilePattern,
                    int Zmin, int Zmax, bool selFlag = true, int nrEvts = -1);
//example: DrawZedA("treeName.root","banPatt_",12,16,false,1e6)
//example: DrawZedA("outName.root","banPatt_",12,16)
```



CheckCal

```
void DrawCharge (   string inFileName, string banFilePattern,
                    int Z, int qmin, int qmax, bool selFlag = true, int nrEvts = -1);
//example: DrawCharge("treeName.root","banPatt_",16,13,15,false,1e6)
//example: DrawCharge("outName.root","banPatt_",16,13,15)
```



RunAnalysis

Run PrismaFilters with basic multi-thread *+ run agataselector*

Setup

PrismaFilters
executable (or link to)

Contains outputs, configuration files and inList files

Runs folder

Configuration folder

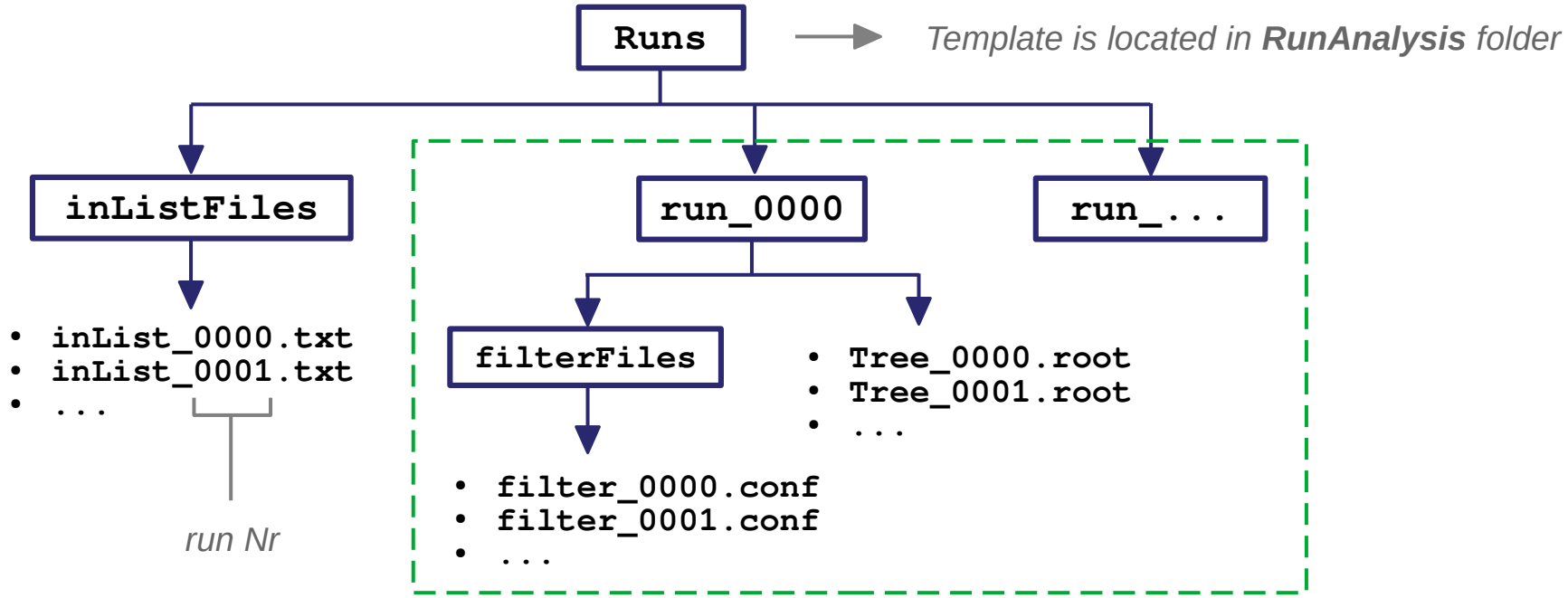
Take from ***PrismaOnlinePackage***

RunAnalysis
executable (or link to)

Compile using the script ***Compile.sh*** *inside the folder* ***RunAnalysis***

RunAnalysis

Runs folder



Produced by the RunAnalysis executable

inList_0000.txt

```
inPath/ruData_0000.xxx
inPath/ruData_0001.xxx
inPath/ruData_0002.xxx
inPath/ruData_0003.xxx
...
```

→ Basic multi-threading is obtained by processing different files on different threads

RunAnalysis

INFN LNL
Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

You can get help with:

```
./RunAnalysis -h          or       ./RunAnalysis --help
```

Help screen:

```
./RunAnalysis [options, ...] [#run, ...]
    --mode             [#]  Select what you want to run, default is 3:
                            1 - PrismaFilters
                            2 - Selector
                            3 - Both
    --nrthr            [#]  Specify maximum number of threads, default is 4
    --verb             [#]  Specify selector verbose level, default is 0
    --adf                   Set output in adf format
    --prismaconf       [#]  Conf directory path, default is    "./PrismaConf"
    --run_dir_pattern[#]  Run directory pattern, default is "run_"
```

Example:

```
./RunAnalysis 0 1 2 3 --mode 1 --nrthr 6
```

RunAnalysis

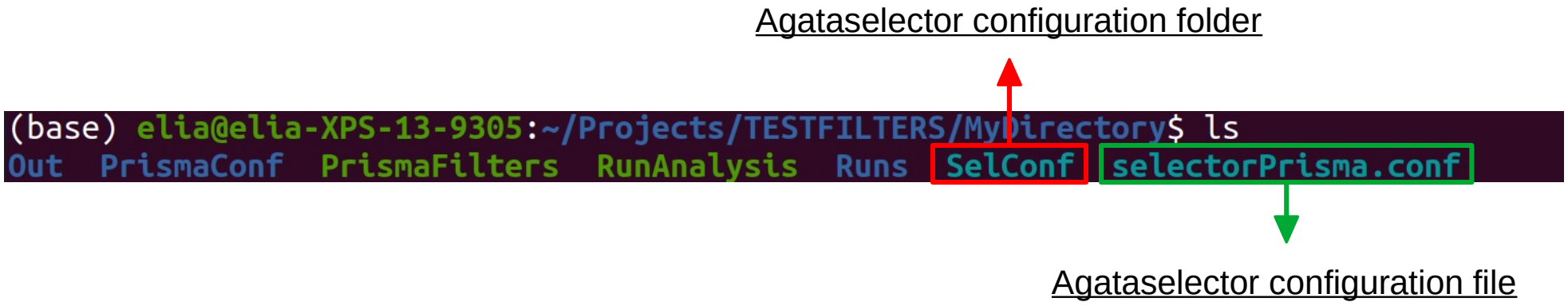
Regarding the agataselector, the command

```
./RunAnalysis 0 1 2 3 --mode 1 --nrthr 6
```

Is completely equivalent to

```
RunSelector --conf selectorPrisma.conf -no_user_sel 0 1 2 3 --nrthr 6 --verb 0
```

Expected folder structure



RunAnalysis

selectorPrisma.conf

```
#-------------------------------------------------------
REPLAY_CONF
ENABLED_HISTOS            enabled_histos.conf   #   File name with list...
SUM_FILE_PATTERN          sum                   #       Hadded file pattern
REPLAY_DIR_PATTERN        run_                  #       Replay directory pattern
TREE_NAME                 PrismaTree            #       Input tree name
IN_FILE_PATTERN           Tree_                 #       Input file pattern
CONF_PATH                 ./SelConf             #       Replay conf folder path
OUT_PATH                  ./Out                 #       Output path
OUT_FILE_PATTERN          run_                  #       Output file pattern
IN_SUB_PATH               /.                    #       Input sub path
IN_PATH                   ./Runs                #       Input path
#-------------------------------------------------------
```

The "`--adf`" option

The option "`--adf`" will produce output files in adf format:

`prismaBU.adf_00` → raw branch
`prismaBU.adf_01` → analyzed branch

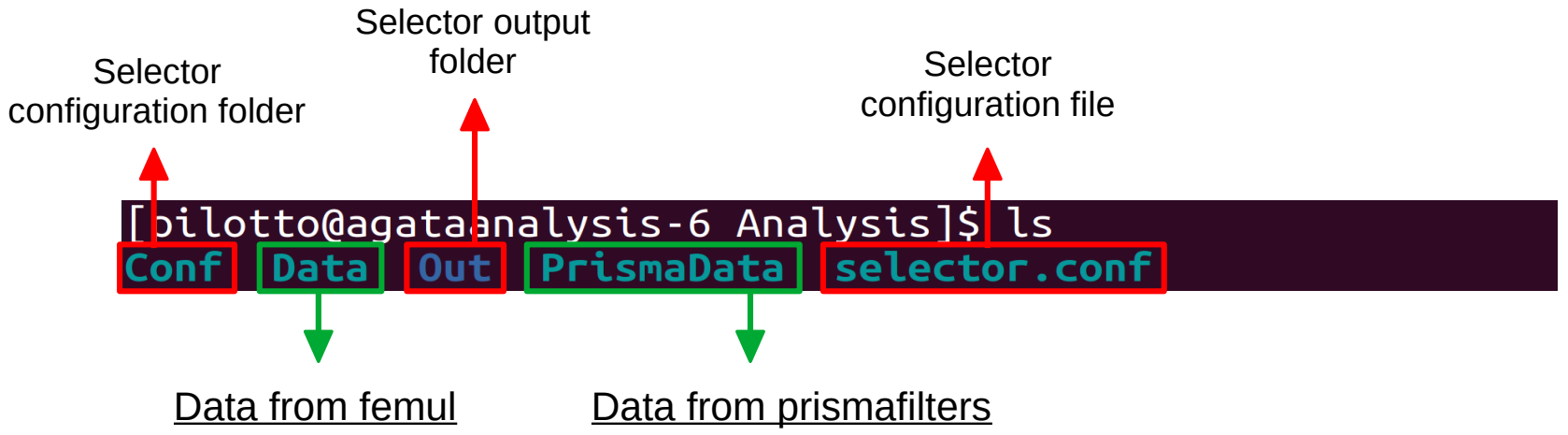
These will be located as normal in the "`Runs/`" folder.

Analysis tools – data merging and the agataselector

The **agataselector** can be used to produce almost all the plots needed to perform the analysis of an experiment involving the AGATA – PRISMA setup.

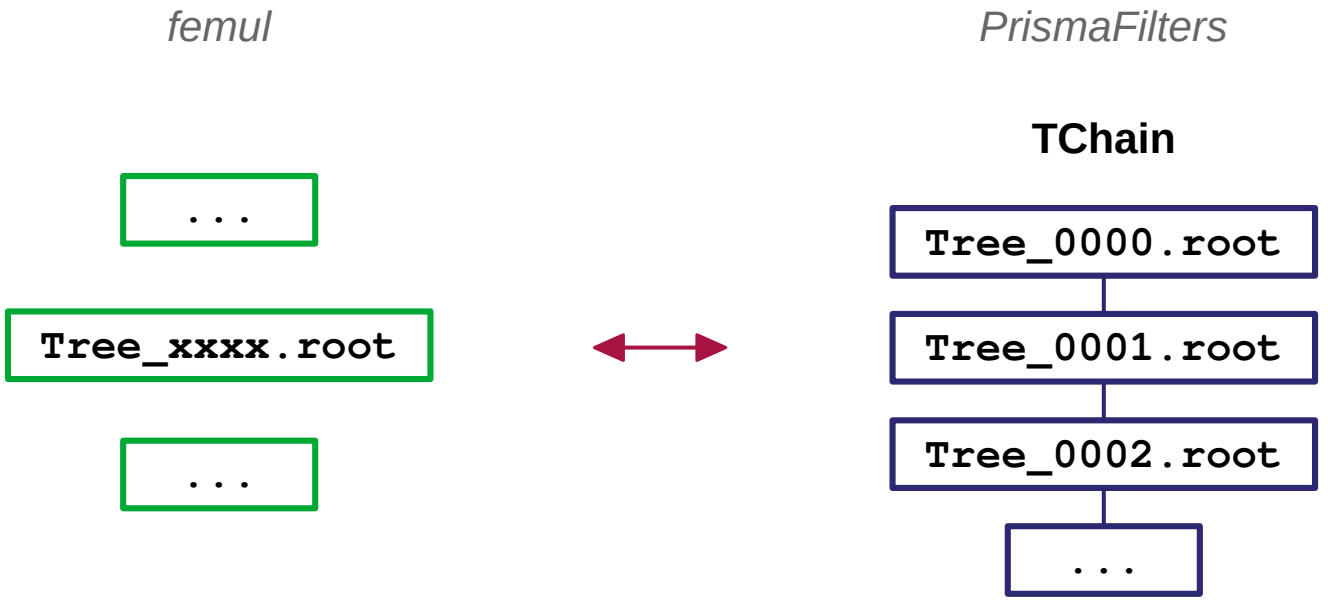
Data merging

- The **`--update_prisma`** option of the *agataselector* will use the files produced with the *PrismaFilters* code to update those produced using the *femul* code.
- In general, running the PrismaFilters code followed by the "update_prisma" procedure tends to be faster than running the femul code for the same dataset.



*"Runs/" folder if you are using **RunAnalysis***

Analysis tools – data merging and the agataselector



The merging algorithm

1. In femul data, from beginning, skip events until non-zero Prisma timestamp
2. Binary search of this timestamp in PrismaFilters data
3. Match! Update analyzed variables in femul data using PrismaFilters data
4. Scan both Trees, skipping zeroes and advancing only on the side with the lower timestamp

+ Basic multi-threading by processing different files on different threads

Thank you for your attention

Aknowledgements

G. Andreetta, F. Angelini, R. N. Del Alamo, B. Gongora, J. Pellumaj

Speaker: Elia Pilotto