

Introduction to ancillary analysis

Daniele Brugnara

How is the workshop structure: Wednesday

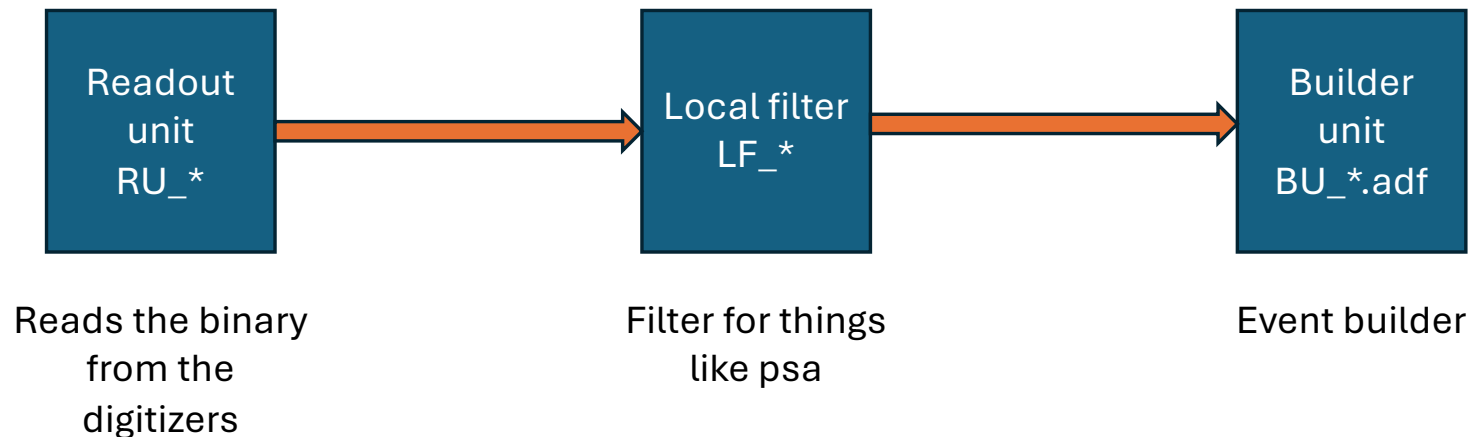
09:00	12 - Global data replay How to produce the final data from the AGATA side <i>RDC-202 - 202, CCIN2P3</i>		<i>Jeremie Dudouet</i> 09:00 - 10:00
10:00	coffee break <i>RDC-202 - 202, CCIN2P3</i> 10:00 - 10:30		
11:00	13 - Selector General info on the selector: what it is and what can you do with it <i>RDC-202 - 202, CCIN2P3</i>		<i>Daniele Brugnara</i> 10:30 - 12:00
12:00	Lunch <i>RDC-202 - 202, CCIN2P3</i> 12:00 - 13:30		
13:00			
14:00	9 - PRISMA: general introduction Detailed description of PRISMA and the data analysis <i>RDC-202 - 202, CCIN2P3</i>	<i>Filippo Angelini</i> 13:30 - 15:00	18 - Other ancillaries General information on the ancillaries, data format and DAQ <i>RDC-202 - 202, CCIN2P3</i>
15:00	coffee break <i>RDC-202 - 202, CCIN2P3</i> 15:00 - 15:30		
16:00	10 - PRISMA: steps of the analysis with examples Step-by-step guide on the PRISMA analysis <i>RDC-202 - 202, CCIN2P3</i>	<i>Filippo Angelini</i> 15:30 - 17:00	11 - Other ancillaries Information on handling the RAW data of the ancillaries and the optimization procedure <i>RDC-202 - 202, CCIN2P3</i>
17:00			

How is the workshop structured: Thursday

09:00	14 - Hands on session ancillaries Read the raw data and build the ancillary events <i>RDC-202 - 202, CCIN2P3</i> Daniele Brugnara et al. 09:00 - 10:00
10:00	coffee break <i>RDC-202 - 202, CCIN2P3</i> 10:00 - 10:30
11:00	22 - Hands on session ancillaries Produce the root data with a global replay Run the selector Prisma analysis I <i>RDC-202 - 202, CCIN2P3</i> Daniele Brugnara et al. 10:30 - 12:00
12:00	Lunch <i>RDC-202 - 202, CCIN2P3</i> 12:00 - 13:30
13:00	15 - Hands on session ancillaries Try the user selector Prisma analysis II <i>RDC-202 - 202, CCIN2P3</i> Daniele Brugnara et al. 13:30 - 15:00
15:00	coffee break <i>RDC-202 - 202, CCIN2P3</i> 15:00 - 15:30
16:00	23 - Hands on session ancillaries Optimization and bkg subtraction <i>RDC-202 - 202, CCIN2P3</i> Daniele Brugnara et al. 15:30 - 17:00
17:00	

Handling the RAW data

- The DAQ that runs online (xDAQ) consists of some actors that handle different tasks.
- Each actor can write data to disk



Handling the RAW data

- It can happen that one of the actors of the daq is overloaded during experiment and stops the data flow
- This can happen for the actor that builds events if there is a sudden increase in rate in one of the channels
- This is usually due to a mistake on the experimental side
- **Nevertheless the data CAN be recovered by reading the output of the readout unit**
- Another reason could be to change the parameters for the PSA of the ancillaries

Possible issues and how to solve them

Coincidences stop at some point	Online building problem		Run ReadCaenRaw and femul
Loss of statistics	Online building problem		
Multiple peaks	Ancillaries or cores not aligned	Align with <code>genconf.py</code> or <code>ReadCaenRaw.set</code>	
Exponential shape	The global time offset is wrong	Find the coincidence peak as explained in <code>Scripts/TimeOffsetPeak</code>	
No peak	There is no global offset		

ReadCaenRaw: how to run and how to configure

- Configuration file-based program
- One needs to set:
 - **Time window**
 - **Digitizer board informazion**
 - **Timestamp offset**
 - **Minimum fold**
- Can also output root files for the selector

```
daniele@MacBookPro AncMerging % ./ReadCaenRaw
Usage: ./ReadCaenRaw [OPTIONS]

Options:
  --help, -h                Show this help message and exit
  --root <OutputRootFileName> Specify the output root file name
  --global-anc-tsoffset <value> Set the global ancillary timestamp offset (double)
  --nrevts <value>          Set the number of events to process (integer)
  --prisma <file1> [file2 ...] Specify one or more input Prisma files
  --input <file1> [file2 ...] Specify one or more ancillary input files
  --adf <outAdfFileName>   Specify the output ADF file name

Example:
  ./ReadCaenRaw --labrslave --root output.root --global-anc-tsoffset 1.5 --nrevts 100 \
  --dante 5 --prisma prisma1.dat prisma2.dat --input anc1.dat anc2.dat \
  --adf output.adf
```

ReadCaenRaw.set_labrprisma 825 B

```
1 #window: width of the time window used for the event building (in ns)
2 window 500
3
4 #boardDef:
5 # arg 1: boardId
6 # arg 2 :boardVersion(V1725/V1730==1 - VX2740 ==2)
7 # arg 3 :number of channels
8 # arg 4: FWVersion (PHA or PSD)
9 # arg 5: ns per timestamp
10 # arg 6: ns per sample
11 # arg 7: data key: 0xFA0201A2 = SPIDER ; LABR = 0xFA0201A5 ; DANTE = 0x
12 board 1 1 16 PSD 2 2 labr
13
14 #minFold:
15 # arg 1: detector
16 # arg 2: minFold for this detector
17 # arg 3: keep only events if coincidence with other board
18 minfold 1 0
19
20 # board   channel   timeOffset
21 tsoffset 1 0 130
22 tsoffset 1 1 130
23 tsoffset 1 2 130
24 tsoffset 1 3 130
25 tsoffset 1 4 130
26 tsoffset 1 5 130
27 tsoffset 1 6 130
28 tsoffset 1 7 130
```


The selector project

- **High level analysis based on configuration files**
- The analysis procedure is common to all experiments and there is little benefit of repeating the same steps over and over
- The code was created for the near-line analysis and has since evolved with more refinements with full analysis capabilities
- The user is expected to take a look at the code and check the output data
- The nearline analysis is a great STARTING point and can be recalled with a simple *git clone*



The starting point

- Femul produces a root file containing MANY leaves
- The selector allows us to have a common code debugged over several experiments
- This also means that sometimes some changes/improvements are made and could break backward compatibility (for instance for the configuration file).
- The README.md file in the root directory, as well as other ones serve as a first documentation
- We try to log on the CHANGELOG.md significant changes in different versions
- Current supported detectors:
 - AGATA
 - PRISMA
 - SPIDER
 - EUCLIDES
 - DANTE
 - LABR₃
 - SAURON (S1)
 - OSCAR
 - EXOTIC's MCPS

Structure of the selector.conf

KEYWORD | value(s) | unit of measure | comment

Detectors considered in the analysis

Configuration of the folders, the file patterns, and the TTree names

Configuration of the reaction, multiple ions of interest can be added

Target thickness and rotations, used for energy loss calculations. The presence of a degrader before or after the target is also possible.


```
#Configuration file for the selector
#Format: | KEYWORD | value(s) | Unit of measure | Comment |
#Comments are ignored unit of measure # means none
#-----
#
DETECTORS_PRESENT
EUCLIDES          NO      #      Euclides is present YES/NO
PRISMA            NO      #      Prisma is present YES/NO
DANTE             NO      #      Dante is present YES/NO
LABR              NO      #      Labr is present YES/NO
SPIDER           NO      #      Spider is present YES/NO
AGATA             NO      #      Agata is present YES/NO
#-----
#
REPLAY_CONF
ENABLED_HISTOS    enabled_histos.conf #      File name with list of enabled histograms
TREE_NAME        TreeMaster #      Input tree name
SUM_FILE_PATTERN sum #      Hadded file pattern
OUT_FILE_PATTERN run_ #      Output file pattern
IN_FILE_PATTERN  Tree_ #      Input file pattern
REPLAY_DIR_PATTERN run_ #      Replay directory pattern
IN_SUB_PATH      /Out/Analysis #      Input sub path
CONF_PATH        ./Conf #      Replay conf folder path
OUT_PATH         ./Out #      Output path
IN_PATH          ./Data #      Input path
#-----
#
REACTION_CONF
REACTION_POSITION 0      0.5 #      Position of the reaction in the taget 0->front 0.5->middle, 1->back
ENERGY            0      MeV #      Beam energy
TARGET            1 1 #      Target ion A Z
BEAM              1 1 #      Beam ion A Z
ION               1 1 #      Fragment of interest for binary reaction calculation: A Z (those detected)
#-----
#
TARGET_CONF
DEG_DISTANCE      0      um #      Degrader distance in um
DEG_THICKNESS     0      mg/cm2 #      Degrader thickness in mg
ROTATIONZ         0      deg #      Target rotation on the Z axis in degrees
ROTATIONX         0      deg #      Target rotation on the X axis in degrees
TILT              0      deg #      Target tilt in degrees; Negative values for clockwise rotations
THICKNESS         0      mg/cm2 #      Target thickness-density in mg/cm2 or units alike
DEG_PRESENT       NO      #      Degrader present YES/NO
DEG_MATERIAL      none #      Degrader material
MATERIAL          none #      Target material
DEG_POS          AFTER #      Degrader position BEFORE/AFTER
#-----
```


The UserSelector

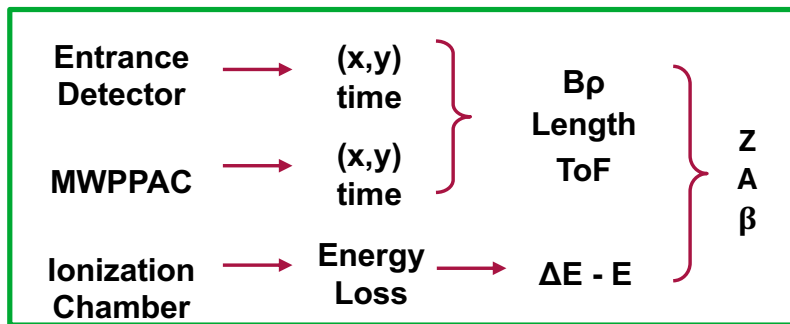
- If a part of the analysis is of general interest, it should be added to the regular part of the code under src/Selector
- However, in many cases some things are experiment-specific and can be handled by the UserSelector
- The experiment specific part is set at compile time

h UserSelector.h 790 B

```
1 #pragma once
2 #include "AgataSelector.h"
3
4 class UserSelector : public AgataSelector {
5     public:
6         UserSelector(const std::string& options) : AgataSelector(options){};
7         virtual Bool_t Process(Long64_t entry) override;
8         virtual void SlaveBegin(TTree* tree) override;
9         virtual void SlaveTerminate() override;
10
11     private:
12         // USER variable and histogram definition section
13
14         unsigned long long oldTS{0};
15         unsigned long long initTS{0};
16         int runNr{-1};
17         int counter{0};
18         double totCR{0};
19         double aliveT{0};
20
21         struct UserHistograms {
22             std::vector<TObject*> ptrs;
23             TDirectory *dir{nullptr};
24             TH1D* h_alive_time = nullptr;
25             TH1D* h_avgCR = nullptr;
26         } userHistograms;
27     };
28
```

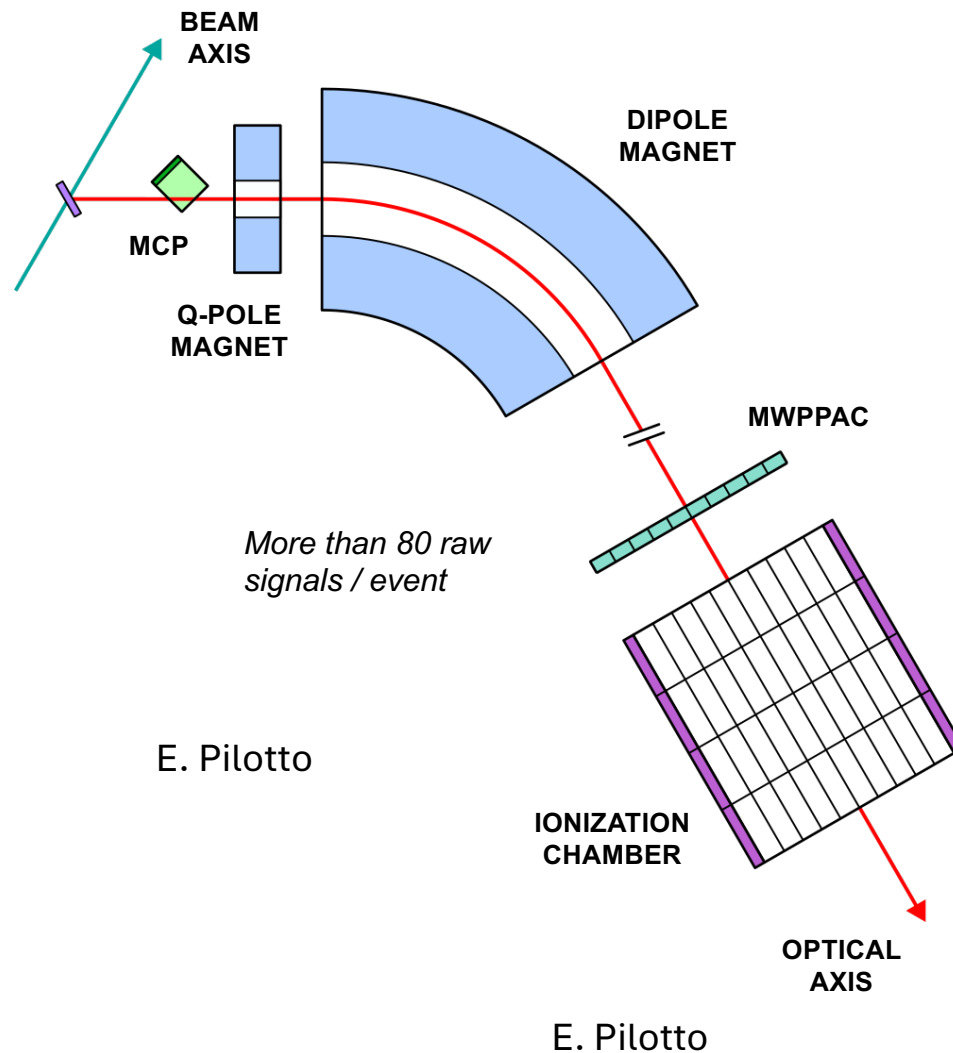


Prisma



Typical performance

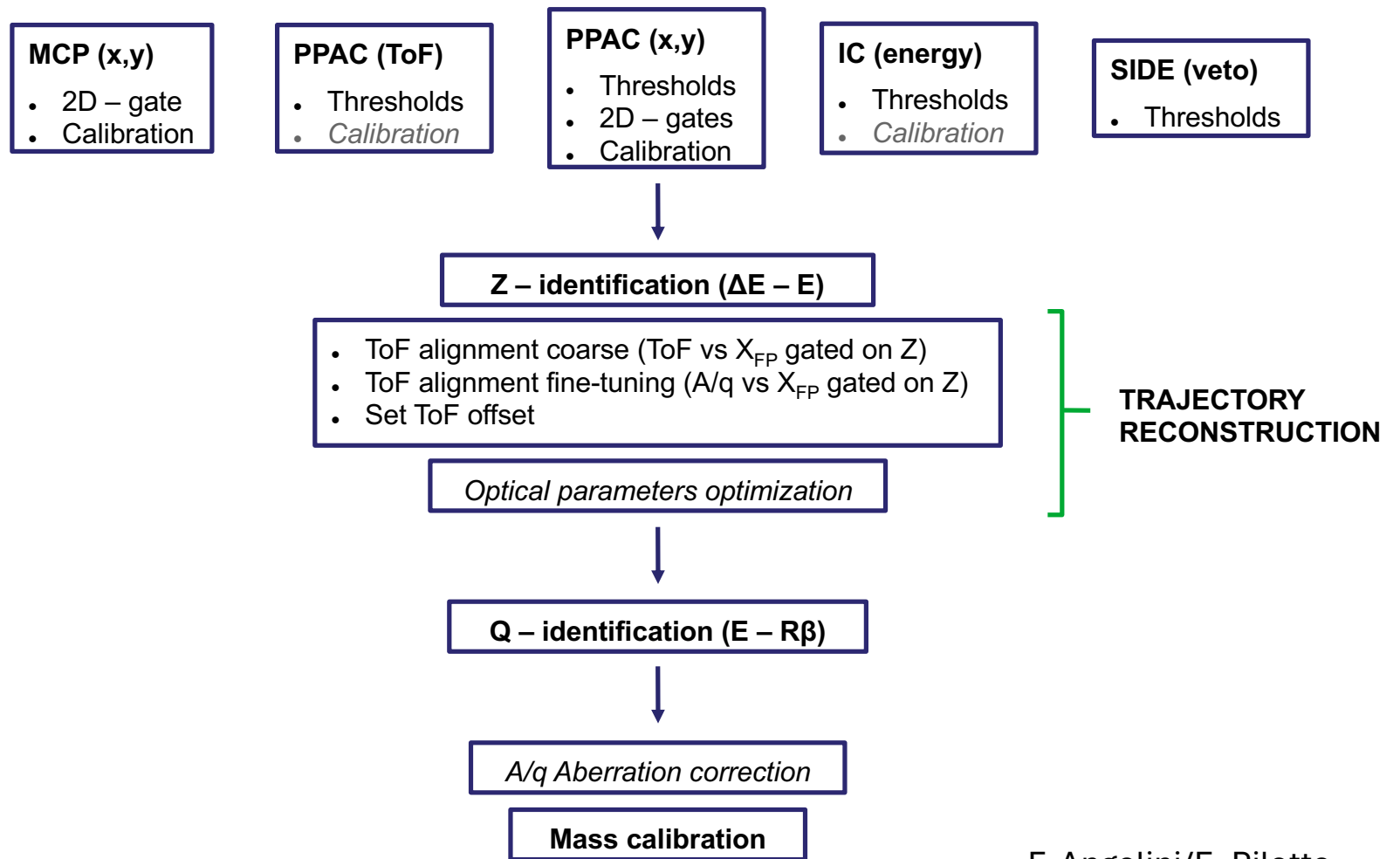
- Z resolution: ~ 1/60
- A resolution: ~ 1/300



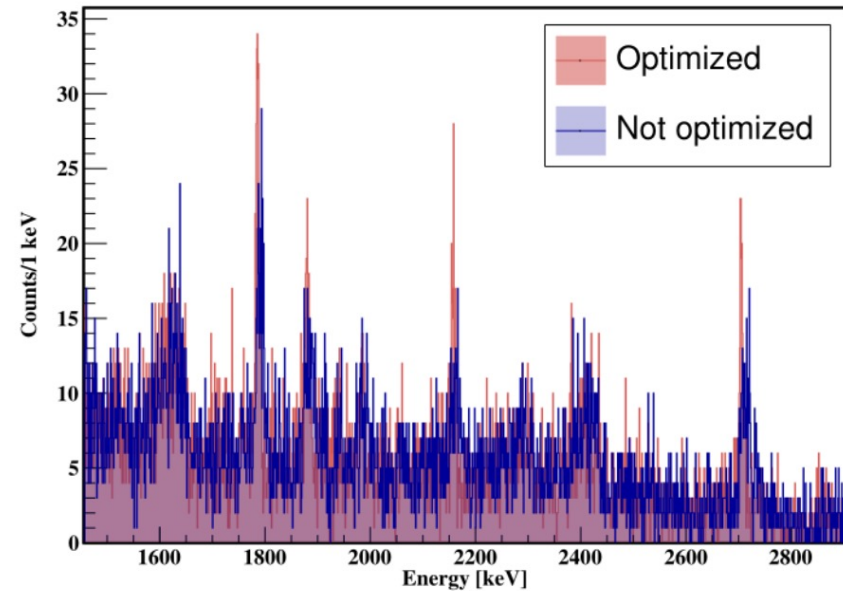
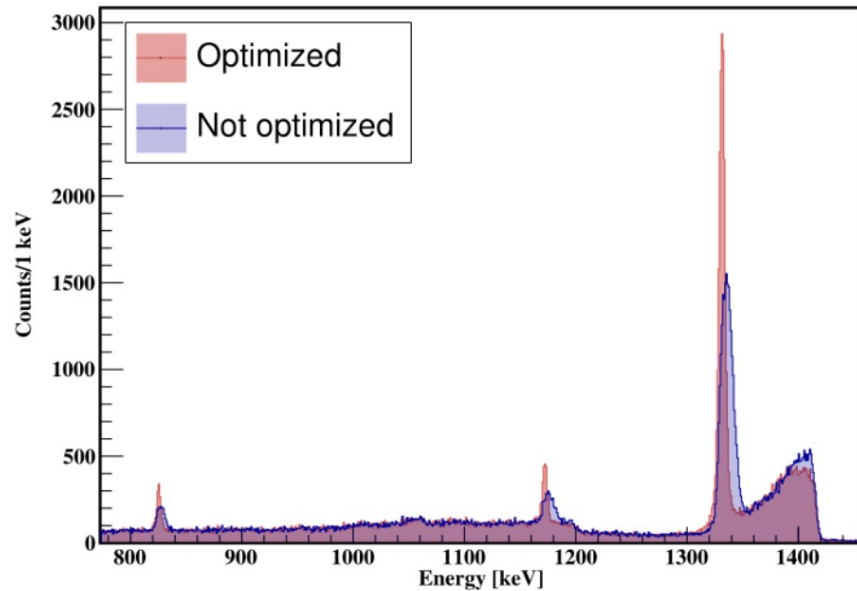
The analysis of PRISMA

- The analysis can be handled by the Prismafilters+prisma library or by the selector
- The outcome is EQUIVALENT
- The analysis with the selector allows for more automated optimization procedures and adds some more corrections/improvements
- **If the PRISMA analysis is concluded and one is satisfied with it, there is no need to repeat it with the selector**
- The selector can take the already analyzed prisma data of the prisma library and just perform the remaining steps of the analysis

Prisma analysis schematics



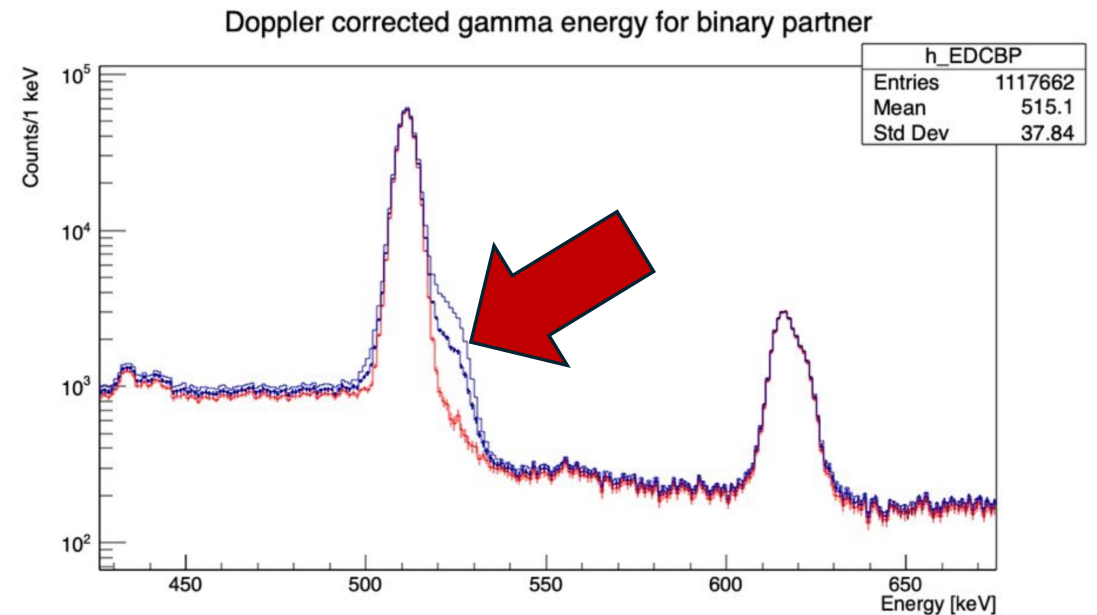
The optimization procedure



- Remarkable improvements are possible with the automatic optimization but are experiment dependent.
- There are two ways to perform it:
 - Multiparameter minimization
 - Scans of a single parameter

Background subtraction and other features

- The selector has a background subtraction option that can really improve the results in some cases
- Other things to check/features are the energy loss evaluations and kinematic reconstructions



General remarks

- You are welcome (and encouraged) to use your one data for the workshop, this will help you to debug issues and ask us questions
- You will be exposed to detectors you might not be interested in at the moment, but it's good to know they exist for proposals/ideas
- Ask questions during the talks, let's keep it informal