MIDAS data analysis of the 2021'PSI campaign

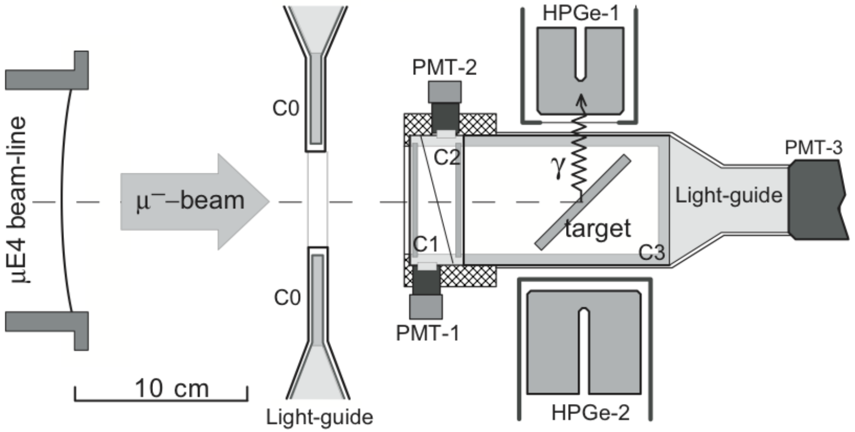
(technical part).

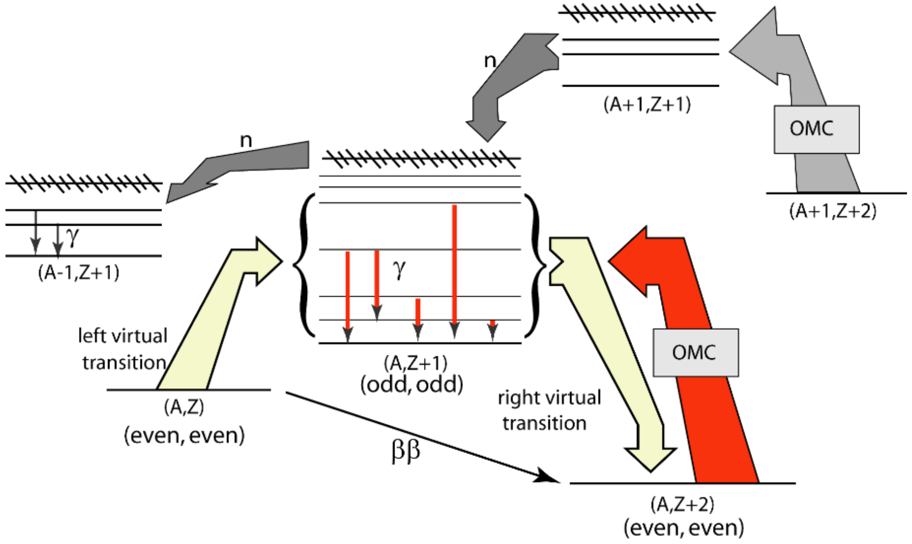
Igor Zhitnikov

OMC4DBD General collaboration meeting 2022

25.02.2022

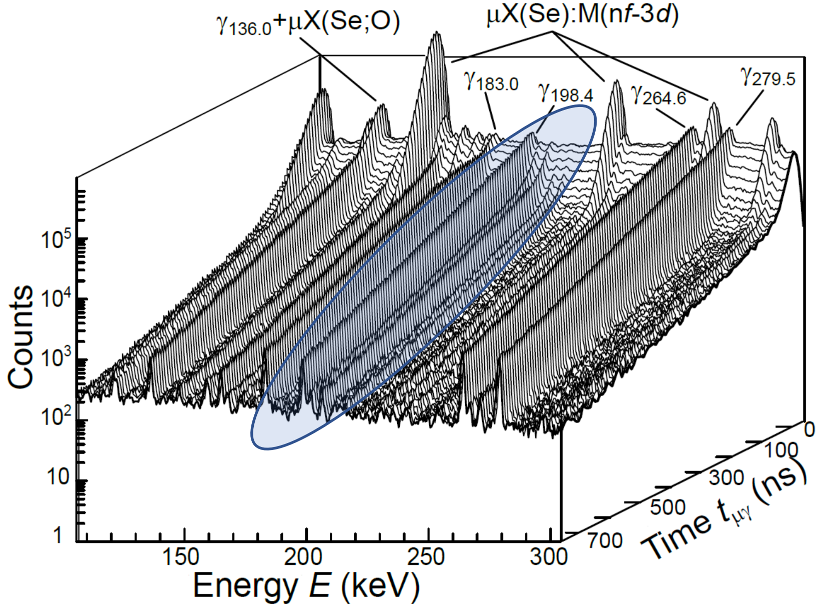
Experimental method of OMC

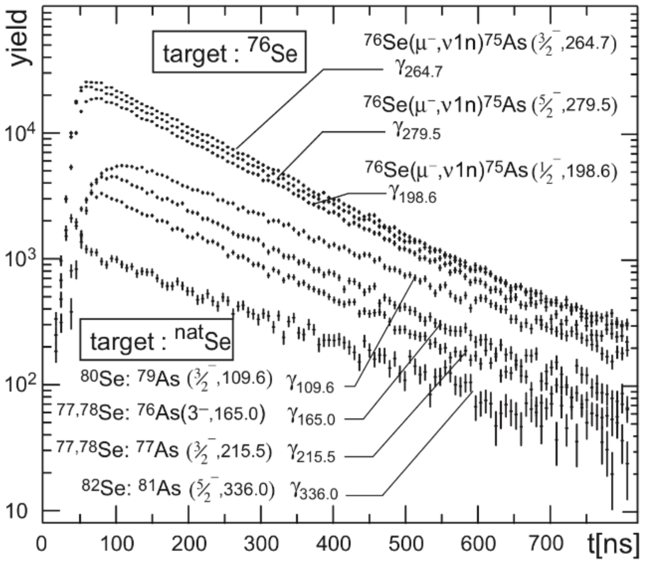




PSI: μE4 beam-line

Number of μ-stop = (8 – 25) x 10³ with 20 – 30 MeV/c

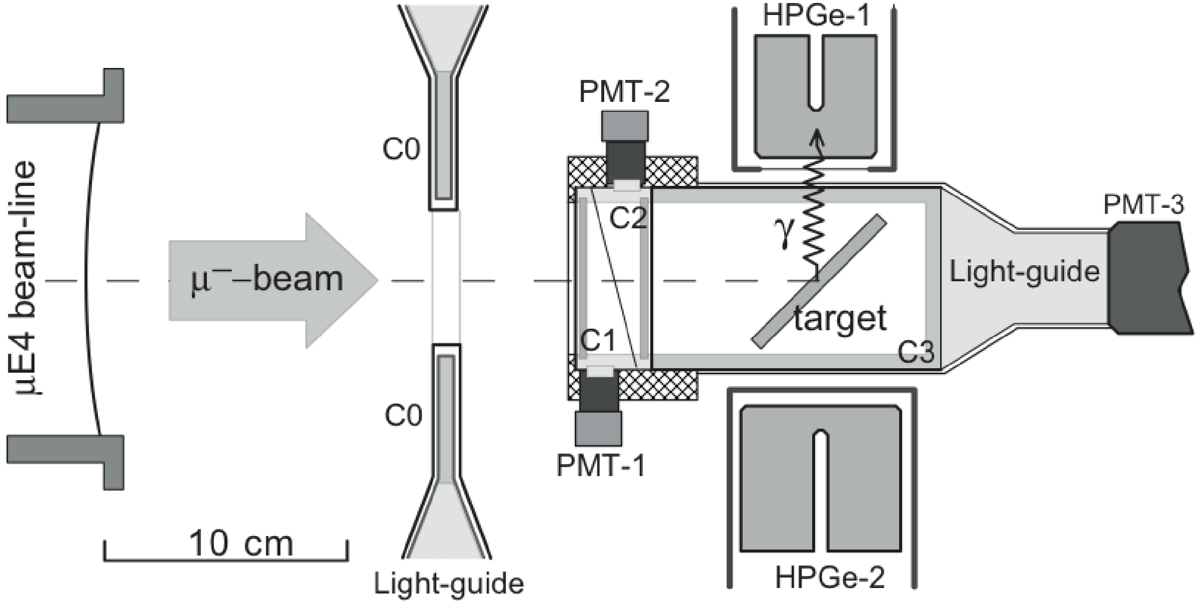




D. Zinatulina, V. Egorov et al. // Phys. Rev. C 99(2019)024327

Experimental method
(data recording)

- Event timestamp and energy for counters C0, C1, C2, C3
  (very high rates, pulses durations ~ 100 ns)
- Event timestamp and energy (with good resolution) for Ge-detectors (Ge1,…, Ge8)
  (high rates, pulses durations ~ 50 mks)

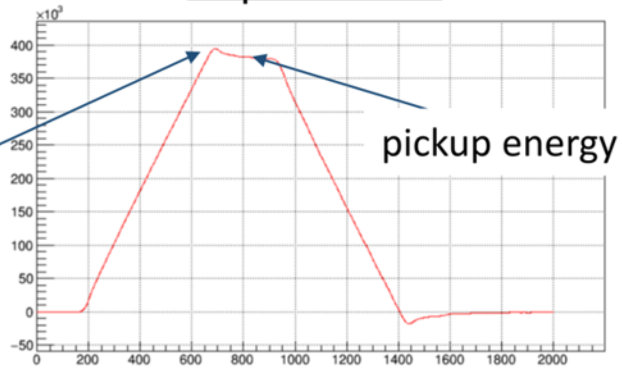


Hardware

Struck SIS3316 digitizer
16-channels, 14-bit resolution,
sampling rate of 250 Mhz
trapeziodal filter firmware

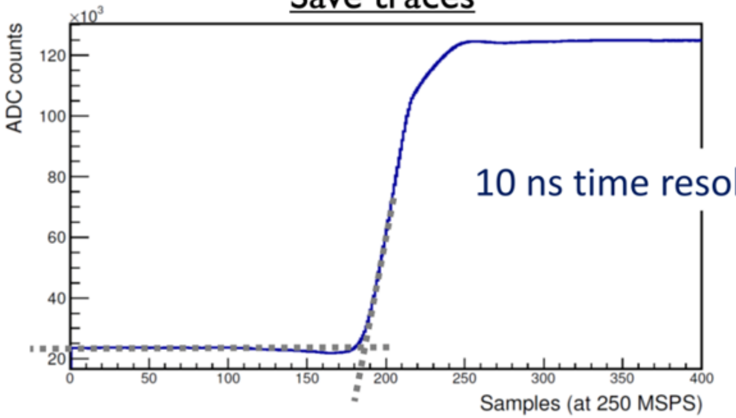
preamp overshoot

Trapezoid filter



pickup energy

Save traces



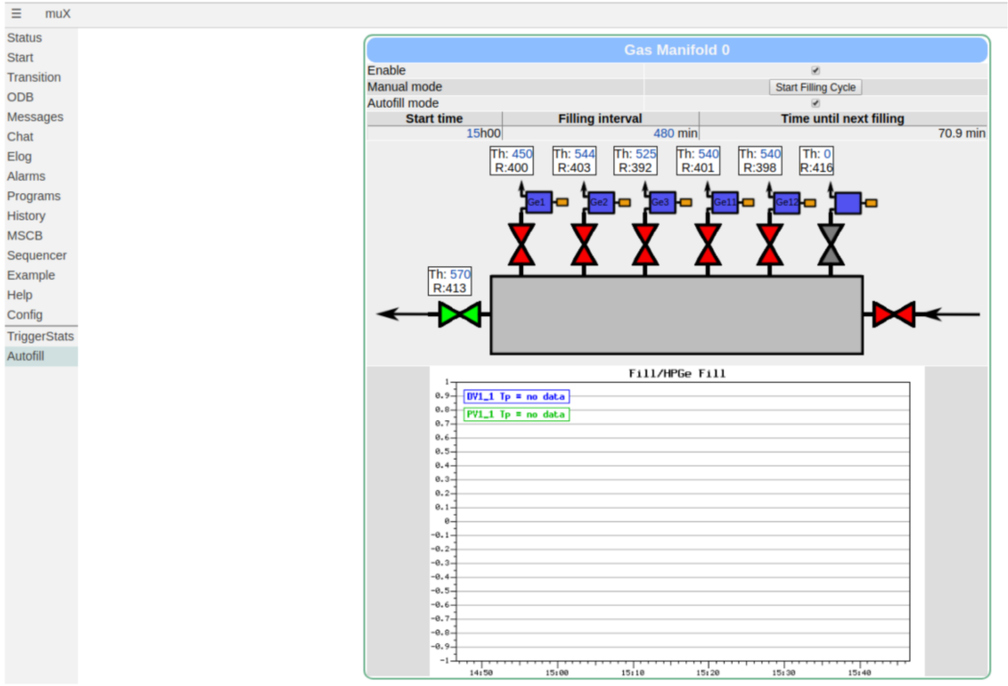
10 ns time resolution

(first order time from
trigger filter)

Data recording. MIDAS.

(Maximum Integration Data Acquisition System)

- Developed and supported by PSI (https://www.psi.ch/en/lin-no-computing/midas)
- Used at experiments MEG, Mu3e, g-2, **muX**, T2K…
- web-interface (remote monitoring and control)



SIS3316 Module 0

Channel Nr	Channel Name	Rate (Hz)	OffSet	Trigger
1	MuonC0	47284.5	16000	50
2	MuonC1	47151.5	16000	10
3	MuonC2	28482.9	16000	50
4	MuonC3	32411.5	16000	50
5	Ge1	4027.4	49152	37
6	Ge2	2807.4	16000	25
7	Ge3	3880.7	49152	20
8	Ge4	3490.7	49152	20
9	Ge5	4588.5	16000	48
10	Ge6	2788.9	16000	20
11	Ge7	3314.8	49152	40
12	Ge8	3954.1	16000	50
13	ch13	0.0	32768	10
14	ch14	31.3	32768	10
15	proton	1954.1	32768	1000
16	Clock	1001.0	32768	1000

SIS3316 Module 1

Channel Nr	Channel Name	Rate (Hz)	OffSet	Trigger

SIS3316 Module 2

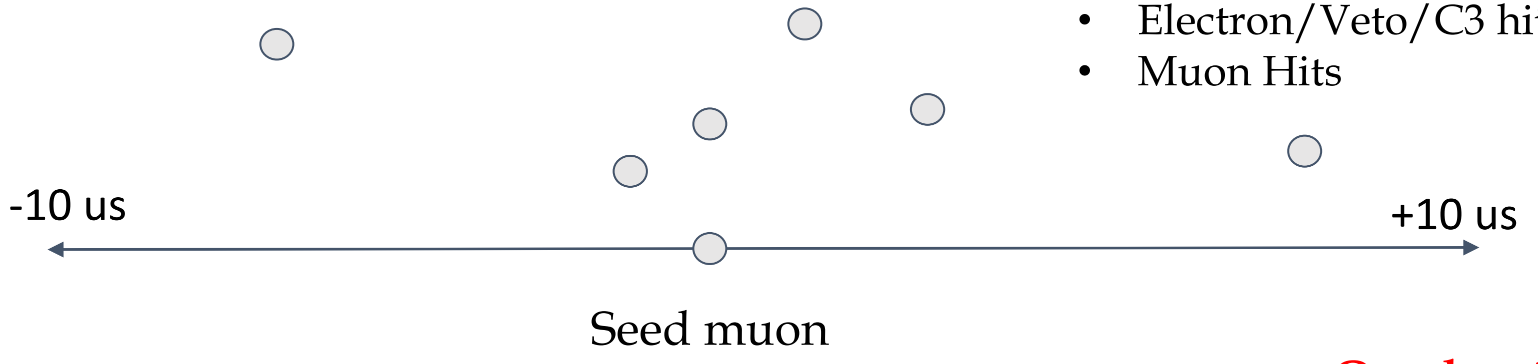
Channel Nr	Channel Name	Rate (Hz)	OffSet	Trigger

Run 45113 Running

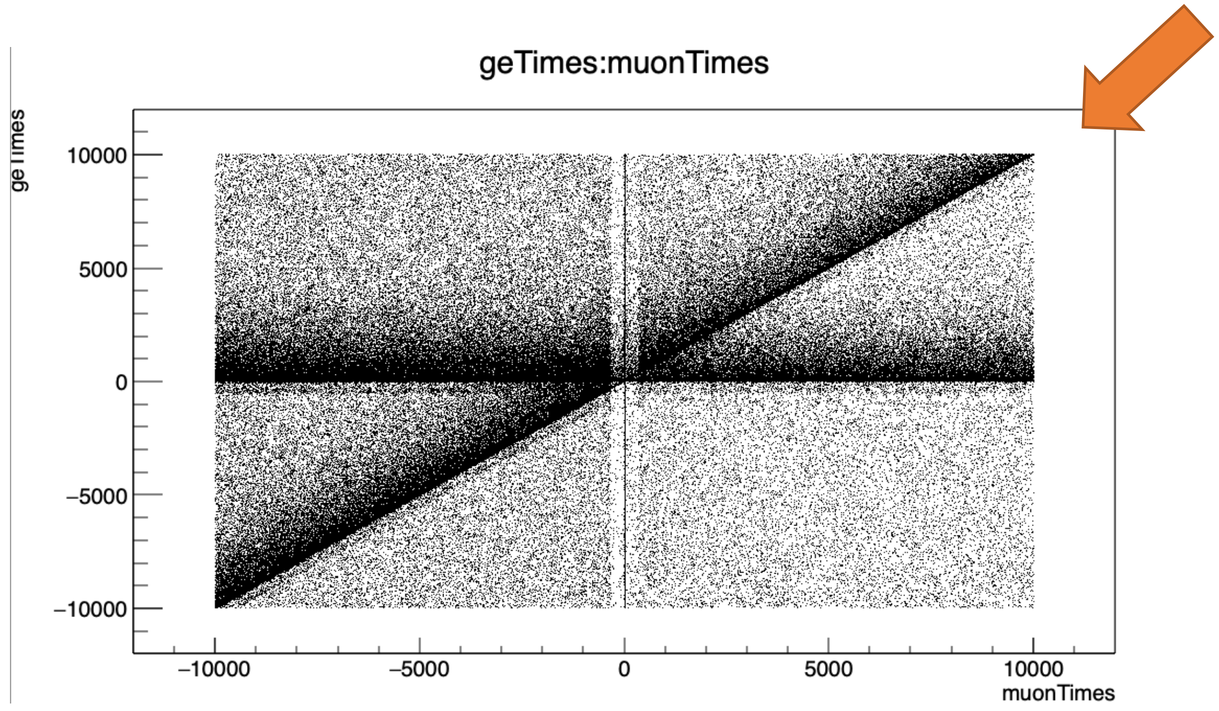
Analyser software from muX collaboration

(needs **ROOT v5**)

- Ge Hits
- Electron/Veto/C3 hits
- Muon Hits



Overlaps!



MuonEvent
(default TTrees from analyzer)

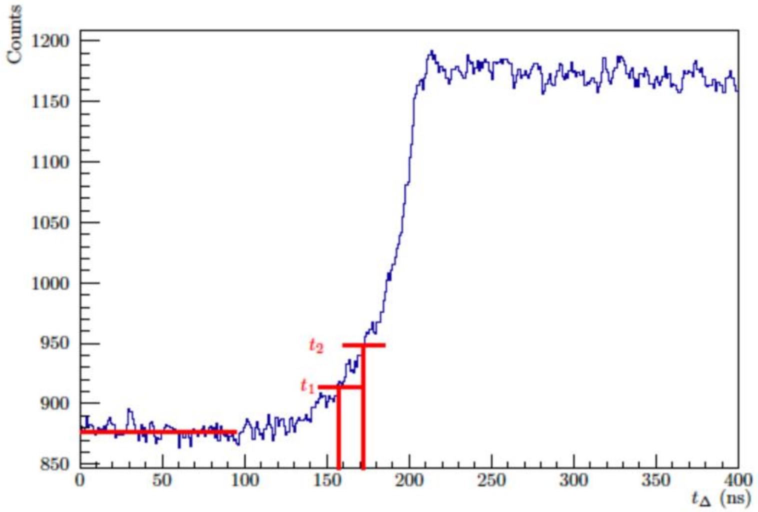
```
struct Hit_t {
  float energy
  float energyADC;
  float time; // time relative to the muon
  bool pile_up; // Pile up flag from sis module
  unsigned short int channel;
  unsigned short int module;
  std::vector<unsigned int> raw_samples;
  float trigger_time;
};


struct MuonEvent_t {
  ULong64_t muonTime;   // muon time in clock ticks
  bool pp;  // Pile-Up Protected muon event
  bool goodEvent;  // flexible good event flag
  std::vector<Hit_t> geHits; // at [-10,10] us time window
  std::vector<Hit_t> muonHits;
  std::vector<Hit_t> electronHits;
  std::vector<Hit_t> neutronHits;

  std::vector<ClusterHit_t> geClusters;
};
```

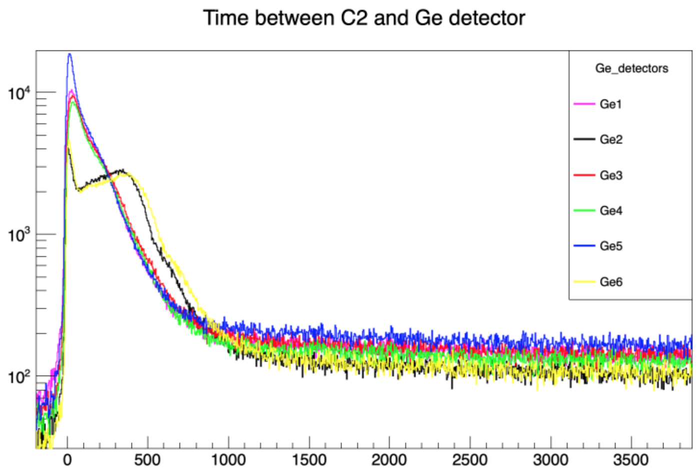
Time correction for Ge-detectors timestamp (ELET)

ELET – Extrapolated Leading Edge Threshold

```
//Determine time of the lower threshold
for (i = fitRangeStop; i <  sis3316trigger->raw_samples.size(); i++ )
{
   if(sis3316trigger->raw_samples[i] >= baseline+threshold)
   {
       tLowerThreshold = i;
       break;
   }
}

//Determine time of the upper threshold

for (i = fitRangeStop; i <  sis3316trigger->raw_samples.size(); i++ )
{
   if(sis3316trigger->raw_samples[i] >= (baseline+threshold*factor) )
   {
       tUpperThreshold = i;
       break;
   }
}
//The ELET algorithm

// cout << 2.*tLowerThreshold - tUpperThreshold - offset << " " << offset << endl;
return (sis3316trigger->time + 2.*tLowerThreshold - tUpperThreshold - offset);
```

$$\Delta t = t_1 - t_0$$
$$\Delta t = t_1 - t_0$$
$$t_0 = 2t_1 - t_2$$





Dubna version of TTree (simplified)

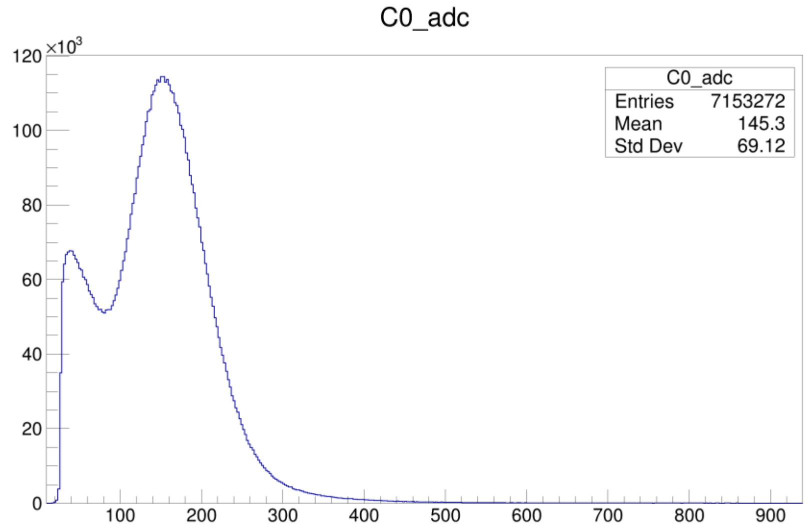
- Original internal structure of event

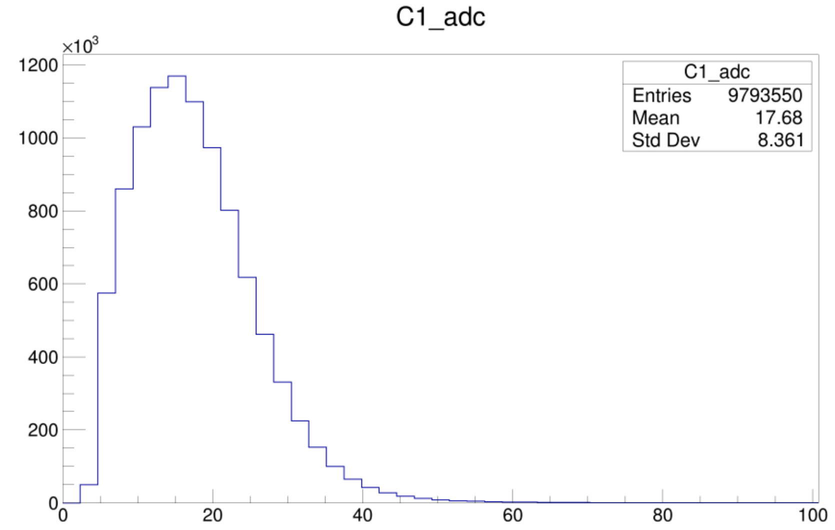
```
struct Hit_t {
  float energy
  float energyADC;
  float time; // time relative to the muon
  bool pile_up; // Pile up flag from sis module
  unsigned short int channel; // channel of
                              // struct module
  unsigned short int module;
  //trace
  std::vector<unsigned int> raw_samples;
  float trigger_time;
};
```

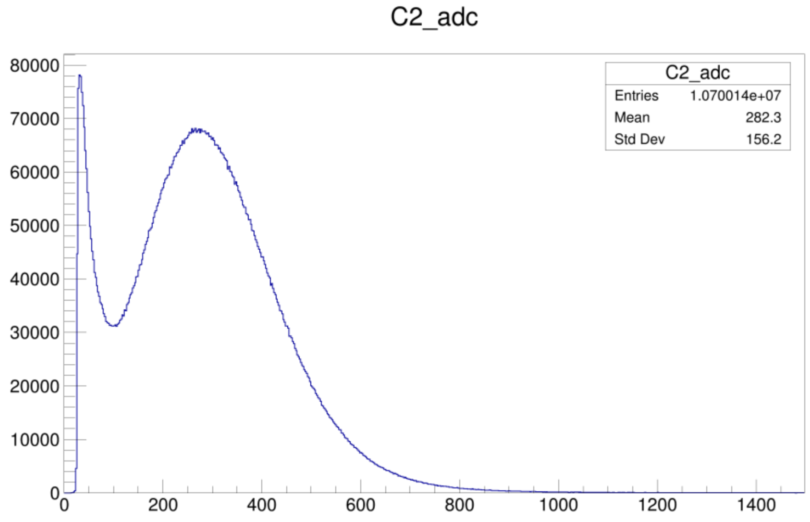
- Dubna structs for TTree's

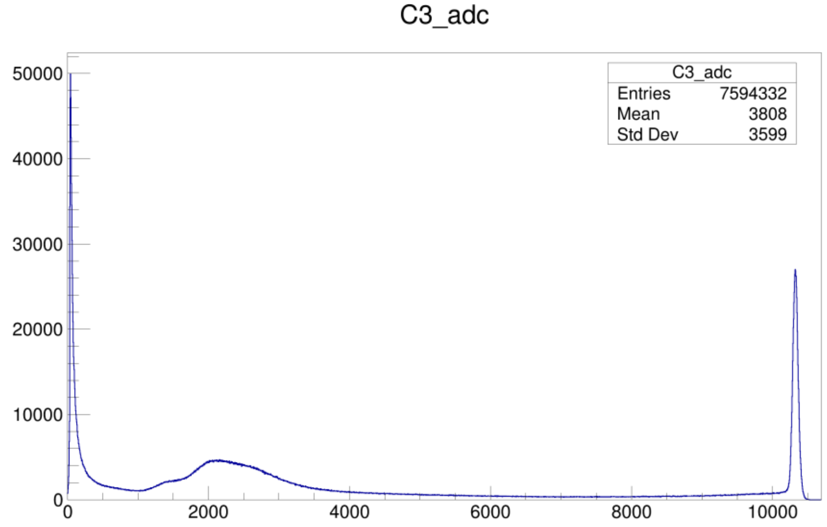
```
struct base_event
{
    Double_t energy;    // for Ge events only
    Double_t energyADC;
    Double_t time;
    uint16_t module;
    uint16_t channel;
    uint8_t status_flag;
};

TTree * mu_tree = (TTree*)f->Get("MuonTree");
TTree * ge_tree = (TTree*)f->Get("GeTree");
```

C# detectors

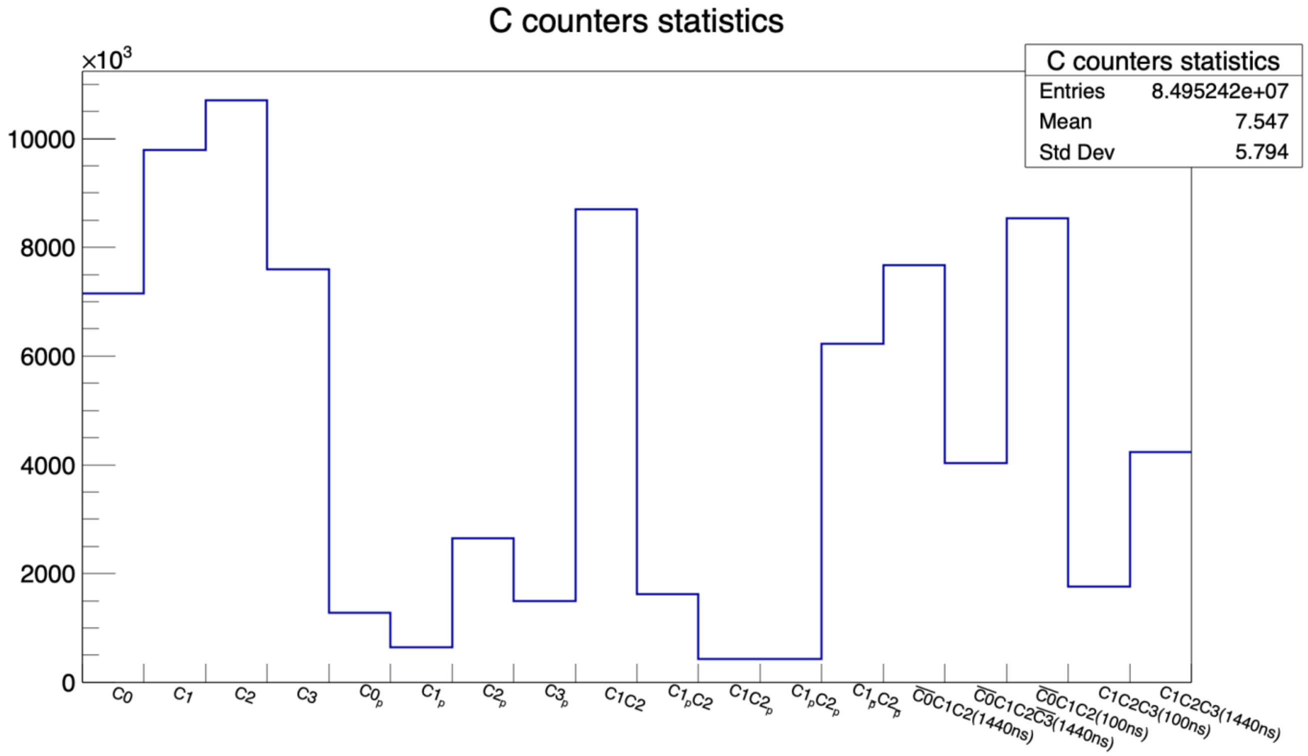

C0_adc
C0_adc
Entries 7153272
Mean 145.3
Std Dev 69.12
×10³
120
100
80
60
40
20
0
100
200
300
400
500
600
700
800
900

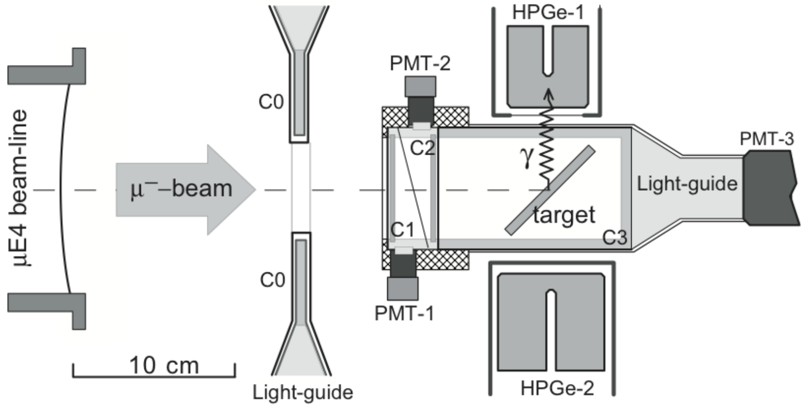

C1_adc
C1_adc
Entries 9793550
Mean 17.68
Std Dev 8.361
×10³
1200
1000
800
600
400
200
0
0
20
40
60
80
100


C2_adc
C2_adc
Entries 1.070014e+07
Mean 282.3
Std Dev 156.2
80000
70000
60000
50000
40000
30000
20000
10000
0
0
200
400
600
800
1000
1200
1400

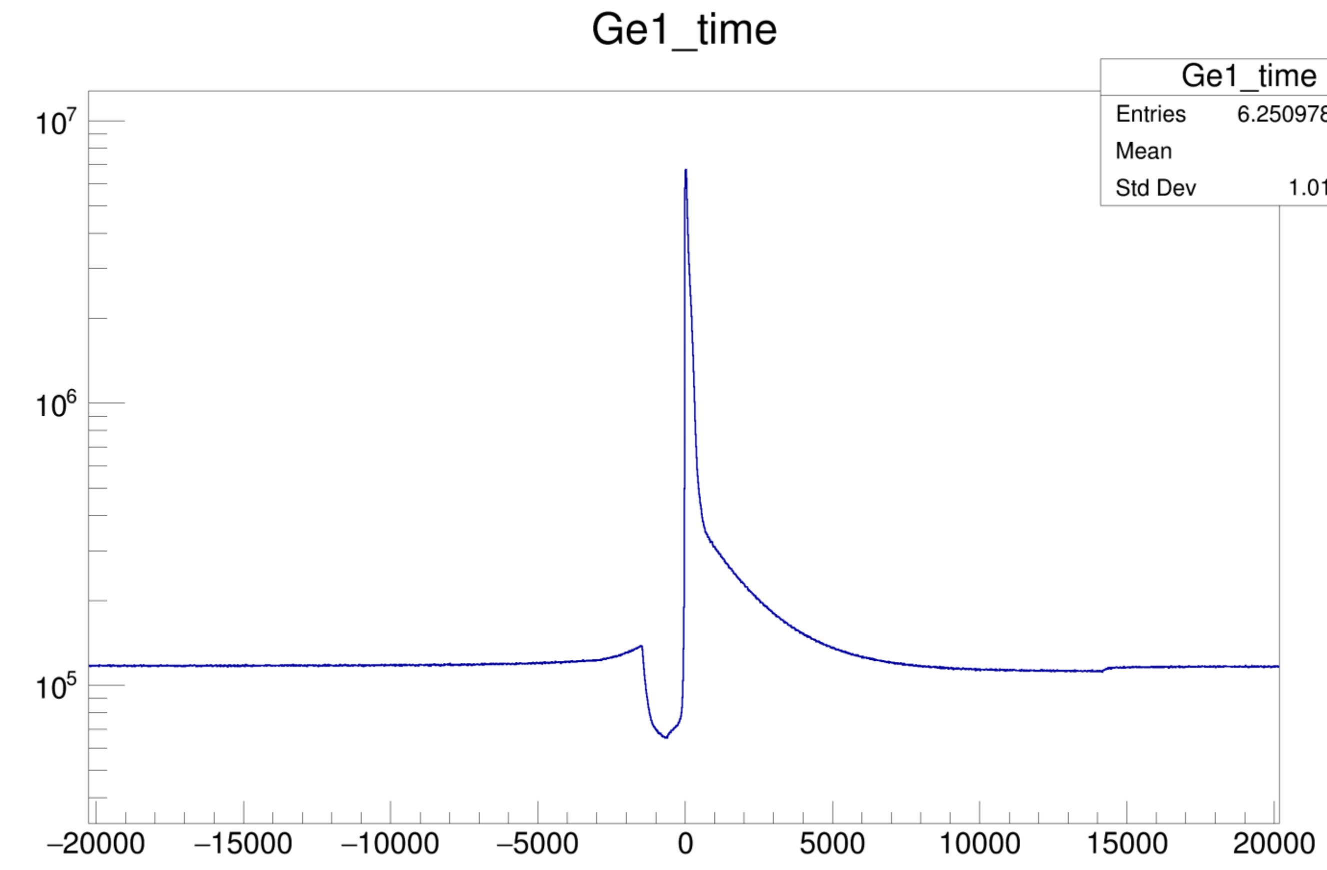

C3_adc
C3_adc
Entries 7594332
Mean 3808
Std Dev 3599
50000
40000
30000
20000
10000
0
0
2000
4000
6000
8000
10000

C counters statistics
(1440ns time window)

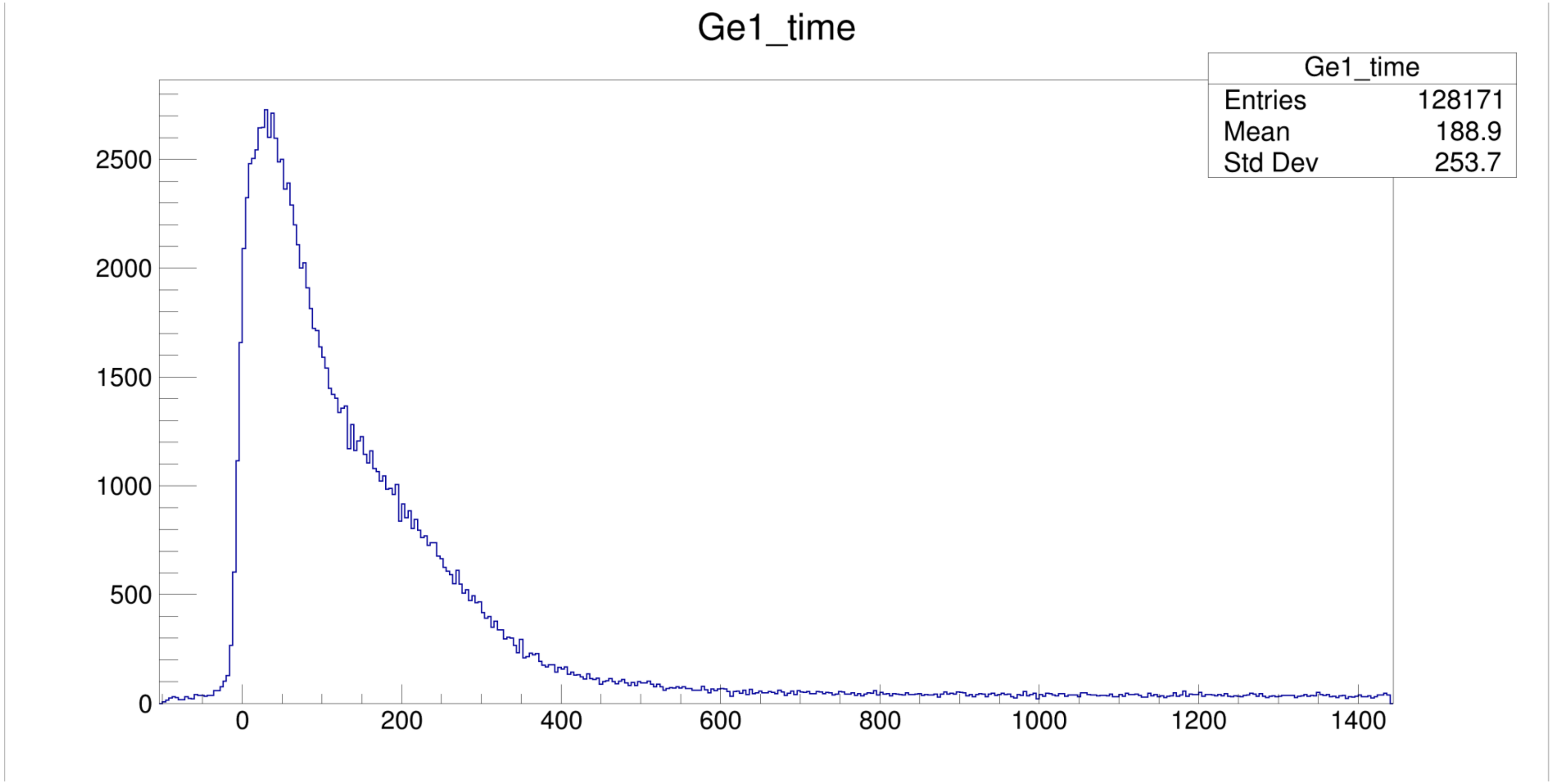




time_Ge1 – time_C1



Ge1_time – Trigger_time



Constructing spectra for further analysis

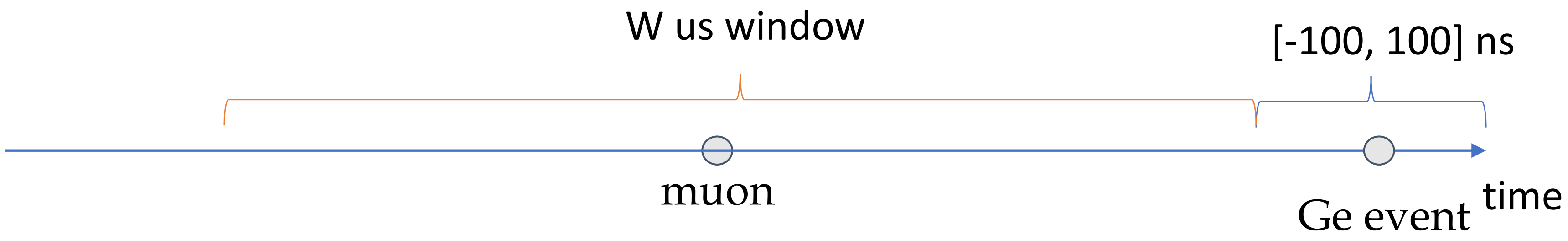
- **all** – all event from Ge detector
- **correlated** - event from Ge detector, if we detected **trigger** event during **W** us before
- **uncorrelated** - event from Ge detector, if we detected **0 C#** events during **W** us before
- **rejected** - event from Ge detector, if we detected **trigger + additional C# events** during **W** us before
- **prompt** - event from Ge detector, if we detected **trigger** event during 100 ns before and after (because time distribution)
- **delayed** - event from Ge detector, if we detected **trigger** event from 100 ns to **W** us before

prompt + delayed = correlated

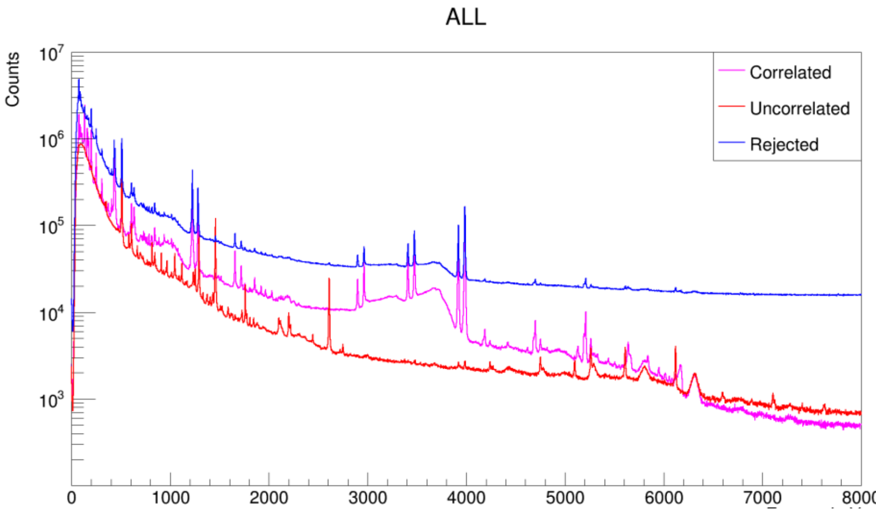
correlated + uncorrelated + rejected = all

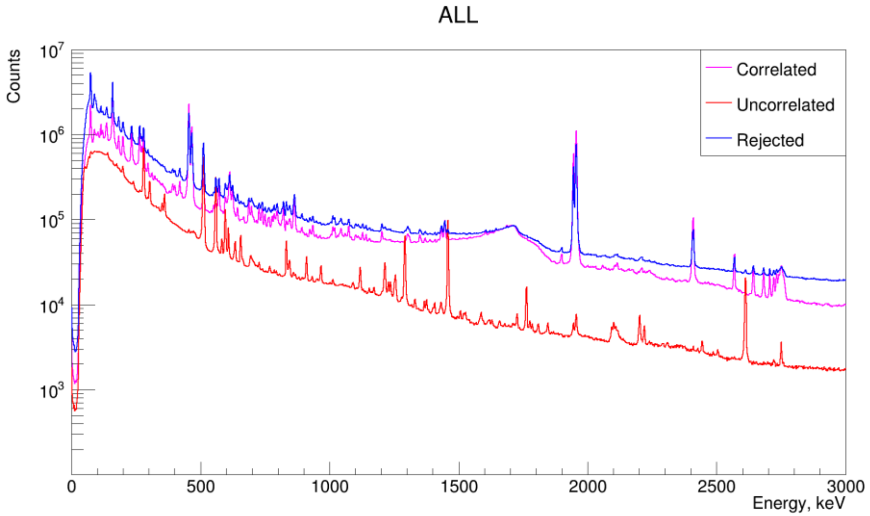
W = 1440 ns (1000ns better?),

Trigger = $\overline{C0}$ & $C1$ & $C2$ & $\overline{C3}$

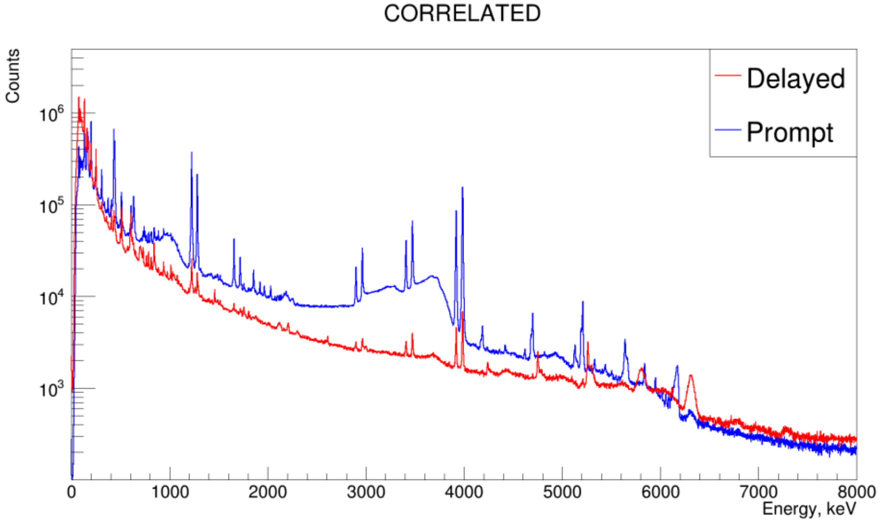


Constructing spectra for further analysis
(part 2, data from Ge1 detector)

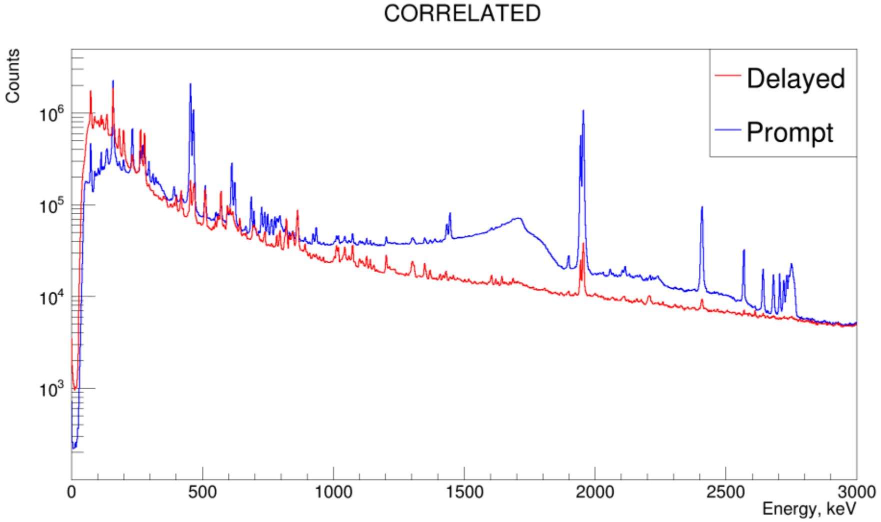




<- ^{136}Ba



^{76}Se ->



Data storage (JINR)
(2019+2021)

Linux server (ubuntu 16.04):

```
Filesystem   Size  Used  Avail Use%  Mounted on
/dev/sdi  7,3T 2,5T 4,4T 36% /monument/data-1    read/write
/dev/sdf1 7,2T 6,5T 312G 96% /monument/run-2019-1read-only (MIDAS data)
/dev/sdh1 7,3T 6,4T 564G 92% /monument/run-2019-2read-only (MIDAS data)
/dev/sde  7,2T 5,5T 1,4T 81% /monument/run-2019-3read-only (MIDAS + 2021 offline data)
/dev/sdd  9,1T 8,2T 420G 96% /monument/run-2021-1read-only (MIDAS data)
/dev/sdj  9,1T 8,6T 0 100%   /monument/run-2021-2read-only (MIDAS data)
/dev/sdg  9,1T 6,0T 2,7T 70% /monument/run-2021-3read-only (MIDAS data)
```

+

BACKUP 3 x 10 TB HDD (run 2021) at Igor desk,
3 x 8 TB HDD (run 2019) at Danya desk

+

In progress: plan to save all ROOT TTrees of run 2021 + some scripts at TUM server

Data translation scheme:

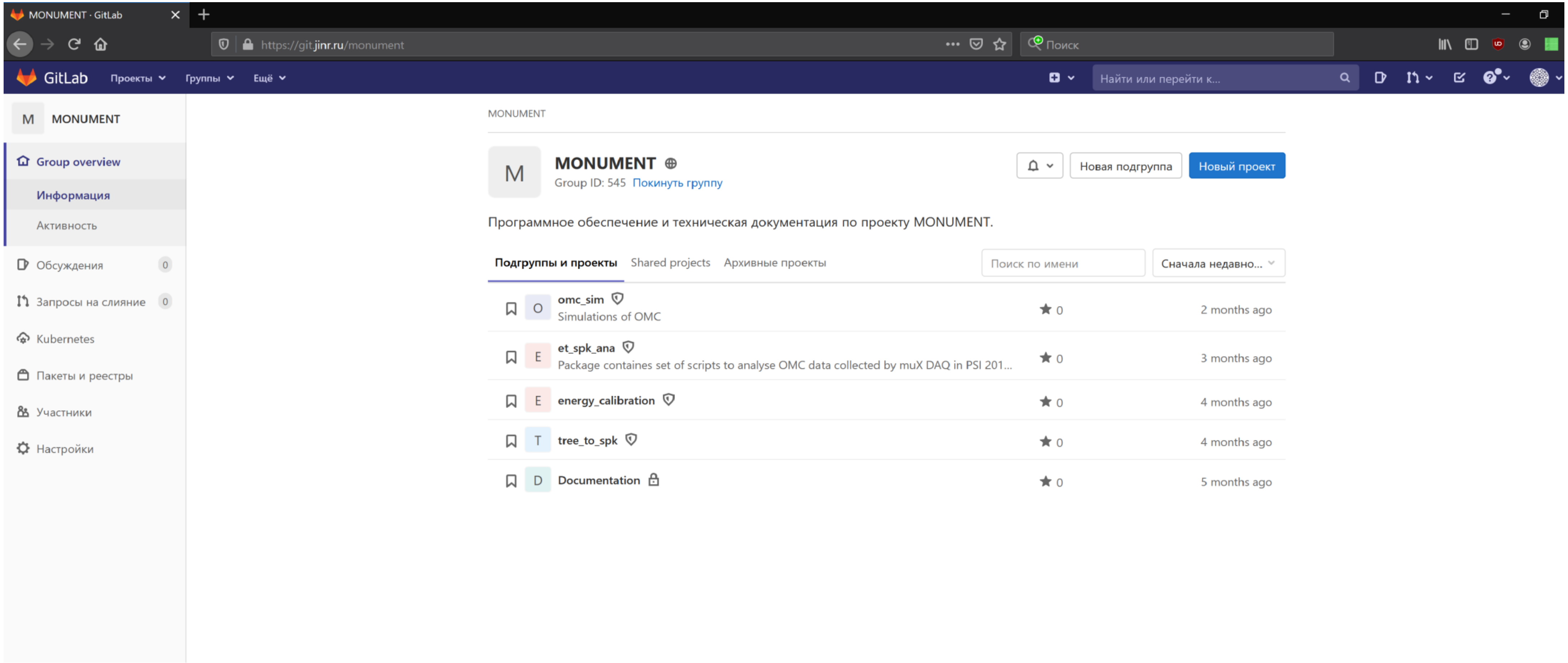
- Original data are stored at MIDAS files:

  run%05d.mid.gz
- *mid.gz -> **analyzer** (+ *.odb midas configuration) -> *.root (TTree)
- *.root (TTree) -> **tree2spk** -> *.root (spectra)

  or

  *.root (TTree) -> **user's script/program** -> *.root (spectra)

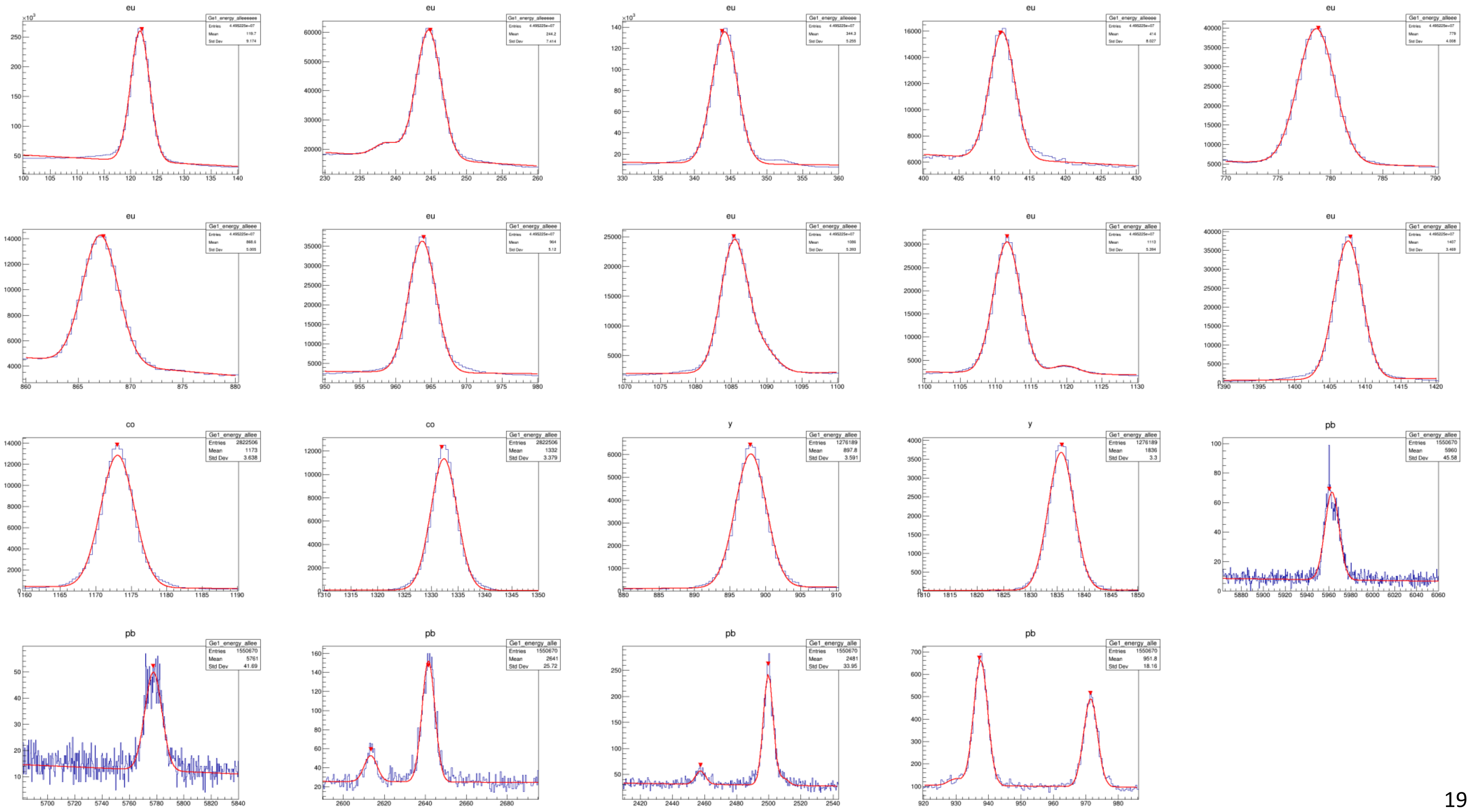
Software storage – JINR git server
(slightly abandoned)



List of scripts (2019, 2021 runs)

- **tree_to_spk/tree2spk** (updated for 2021 run, uploaded to TUM server)
- **fit_lines** (updated for 2021 run, uploaded to TUM server)
- **efficiency_calculation** (updated for 2021 run, uploaded to TUM server)
- **check_calibration** (stability, recalibration of data, not updated yet)

Fit lines: Eu, Co, Y, Pb-nat (Ge1)



Method to obtain *relative* efficiency parameters

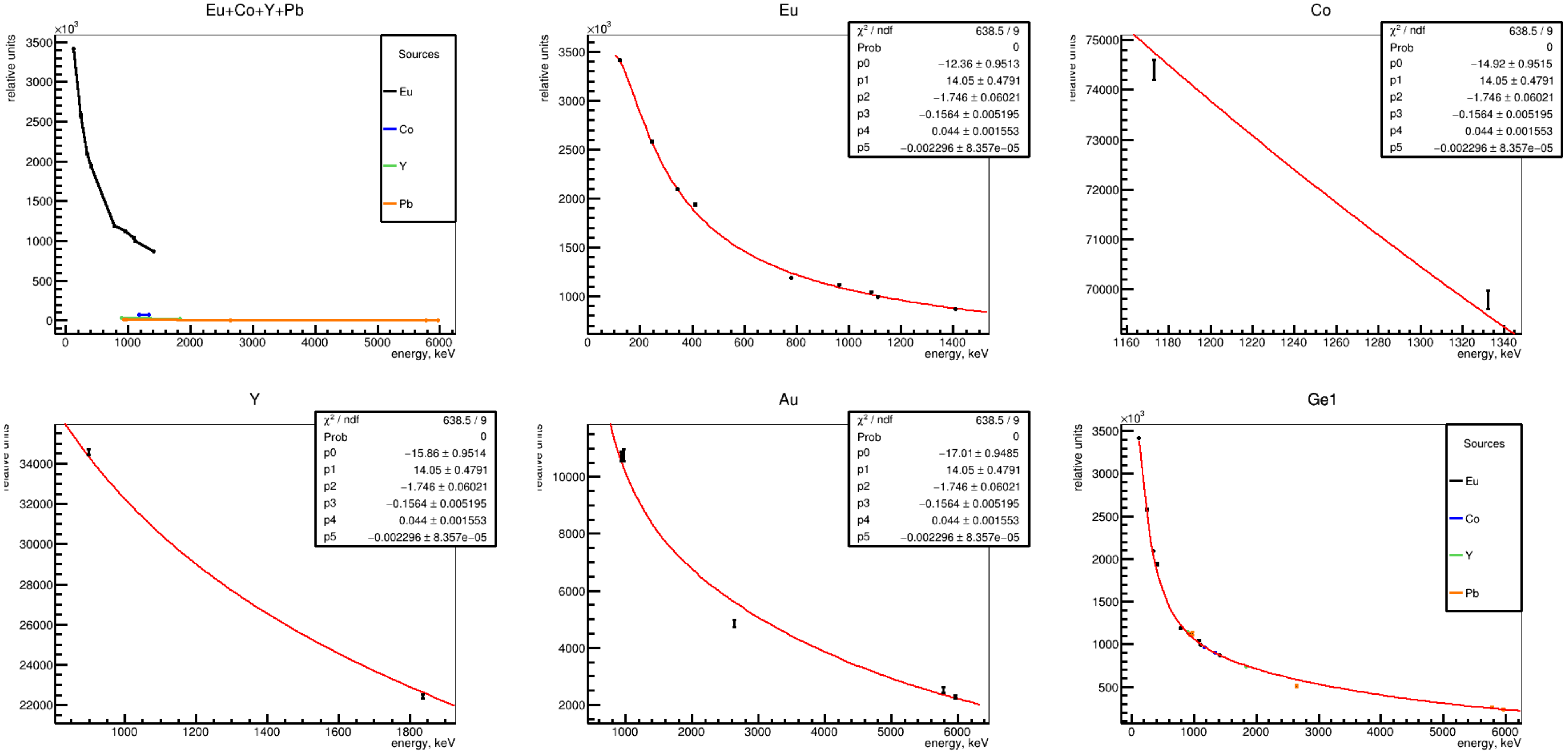
- Take measurements with radioactive source with at least 2 gamma-lines (Eu, Co, Y, Pb)
- Avoid sources activity information (we don’t need this)
- Fit gamma-lines and compare intensities from tabulated data
- Fit simultaneously different sources data (by algorythm from https://root.cern/doc/master/combinedFit_8C.html)
- For efficientcy curve we take function:

$$Eff = e^{p_0+p_1 log(x)+p_2 log^2(x)+p_3 log^3(x)+p_4 log^4(x)+p_5 log^5(x)}$$

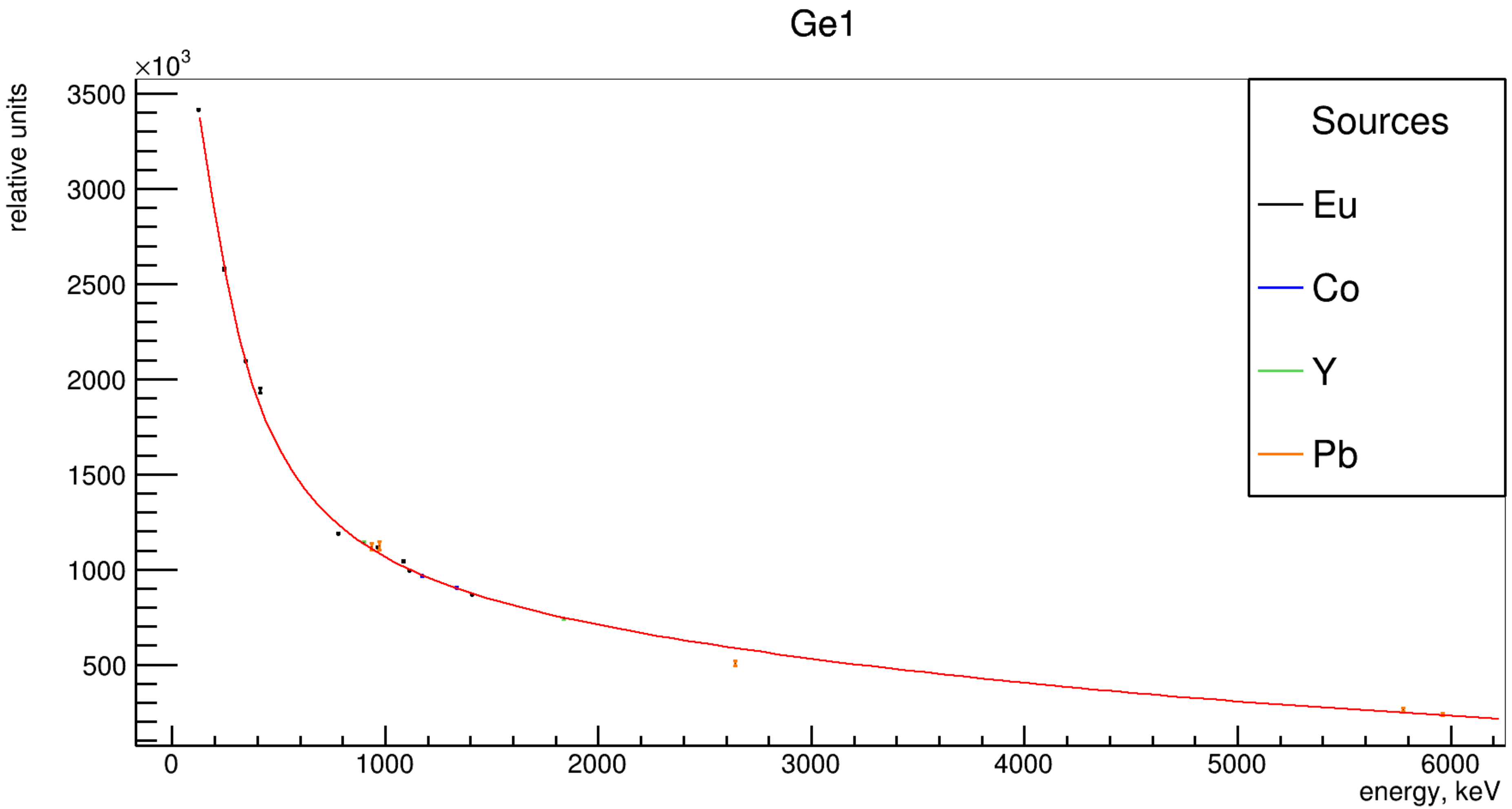
- Find normirating coefficients to Eu points: $K_{normOTHER} = exp(p_{0EU} - p_{0OTHER})$
- Plot all point at “Eu scale”

Efficiency fit

(Ge1 detector)



Efficiency curve



Plans

- Improve/tune all scripts for 2021 run’s data:
  - Puleups rejection/cuts (to get better gamma-lines shape)
  - Efficiency calculations
  - Gamma line fit procedure (add steps + left tails to fit function)
  - Fix time alignment for BEGe detectors (#2, #6) by processing waveforms
- Organize all scripts (git) and prepare a nice documentation
- Continue copying of MIDAS data/update scripts at TUM server