Camilla Salerno

Pulse Finder

MC simulation 39Ar and Co-60

MC simulated data: single scatter mix_events

```
payload:
  command: python /dslab/exe/mix_events.py
  args: # anything which differentiates the jobs (e.g. dependence on $DS_JOB_ID) must be specified here
    - -i /darkside/single.csv
    - -o test.fil
    - -m 8 66
    - -g 1
    - -n 3600
    - -p "daq.slice=1*nu.s" # specify slice duration length, for electronics simulation
    - --seeds $DS_JOB_ID $DS_JOB_ID # sets the two numpy and numba seeds to the job number
  outputs: # here we specify what output files need to be copied
    - test.fil # so the output files will be renamed to test_0001.fil etc, based on $DS_JOB_ID
```

In the single.csv file the only background left activated is:

0,39Ar,lar_tpc,/storage/gpfs_ds50/darkside/users/hessel/g4ds10/Linux-g++/background_production/data_model/out_files/ar39.tpcdistribution.10000000.0.fil

39Ar → Activity: 36.4 Bq Efficiency: 0.999999 Activity X Efficiency=36.4

In this example:

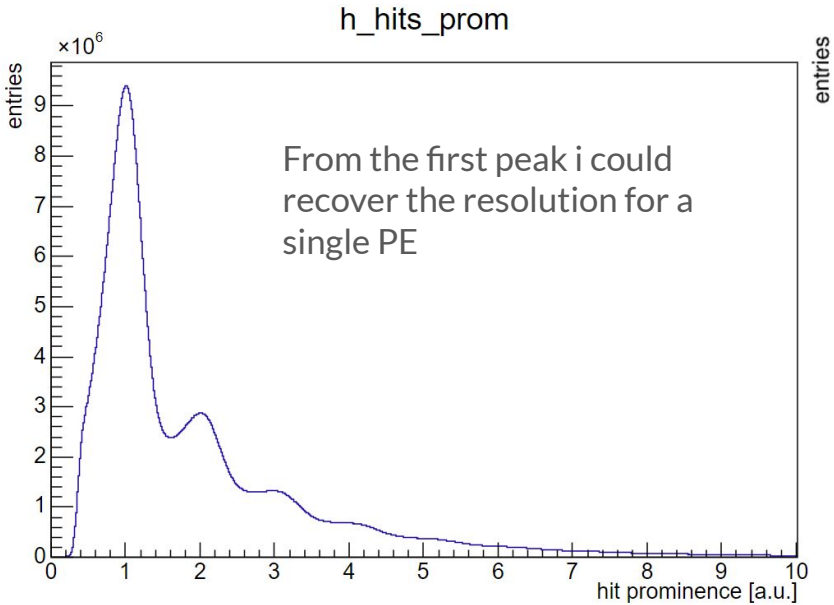
- Sending one job with -n 3600, the output will be a file .fil which contains 3600 simulated gates of 1s
- This file is then splitted in 120 file.fil, each containing 30 gates of 1s

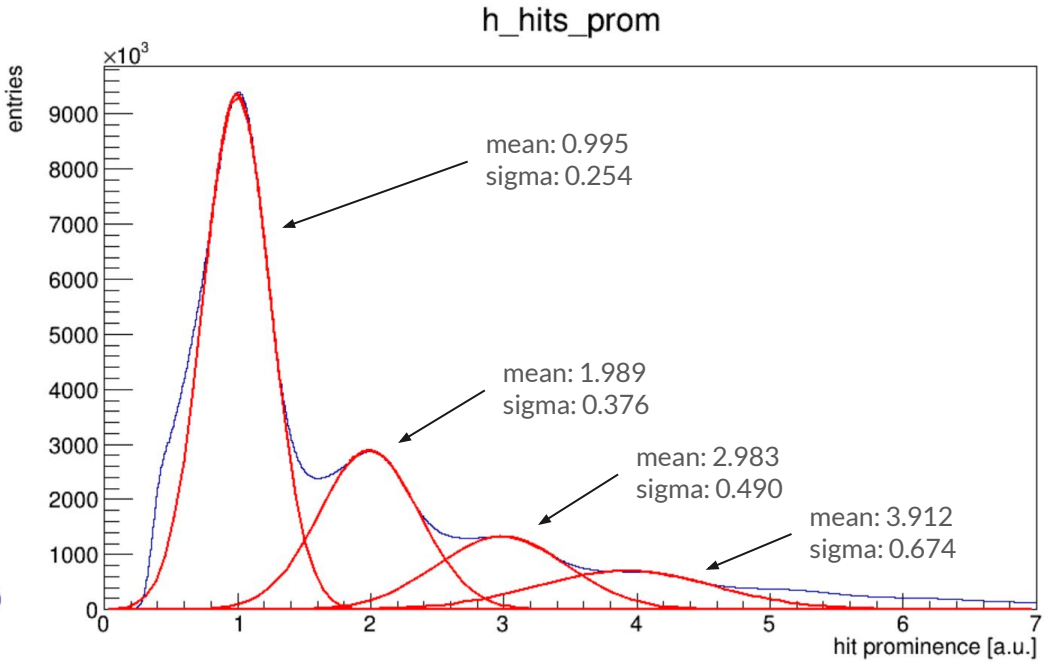
MC simulated data: Full chain

```
- command: python /darkside/bin/g4ds_wrapper.py # we mounted the qwait folder as /darkside
  args: # anything which differentiates the jobs (e.g. dependence on $DS_JOB_ID) must be specified here
    - --input $DS_JOB_INPUT # since this step is executed from another dir, we specify the full path to its input
    - --output $DS_JOB_WORKDIR/s1 # we make sure the output is put on the working directory (local node); g4ds will add .fil/.log
    - --mac /darkside/examples/g4ds_fromdep.mac
    - --jobid $DS_JOB_ID
    - --base-seed 1234
    - --events 30
    - --execute-from /opt/g4ds11/build # which version of g4ds to use
- command: python /dslab/exe/clusters_and_pulses.py
  args: # anything which differentiates the jobs (e.g. dependence on $DS_JOB_ID) must be specified here
    - --s2
    - --veto
    - -i s1.fil
    - -o s2.fil
    - -p "daq.slice=1*nu.s" # specify slice duration length, for electronics simulation
    - --seeds $DS_JOB_ID $DS_JOB_ID # sets the two numpy and numba seeds to the job number
- command: python /dslab/exe/daq_slices.py
  args: # anything which differentiates the jobs (e.g. dependence on $DS_JOB_ID) must be specified here
    - -i s2.fil
    - -o raw.slc
    - -p "daq.slice=1*nu.s" # specify slice duration length, for electronics simulation
    - --seeds $DS_JOB_ID $DS_JOB_ID # sets the two numpy and numba seeds to the job number
    - --truth
```

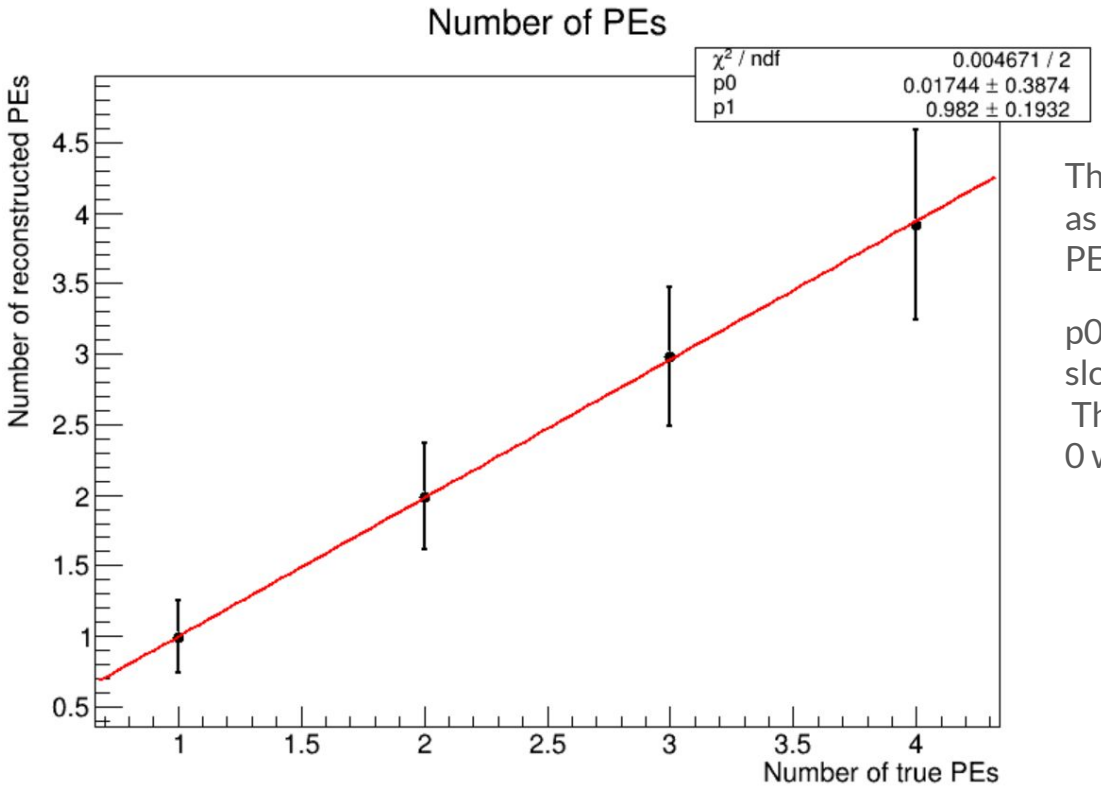
Single PE calibration

Output MC: Hits prominence distribution





PE reconstruction calibration



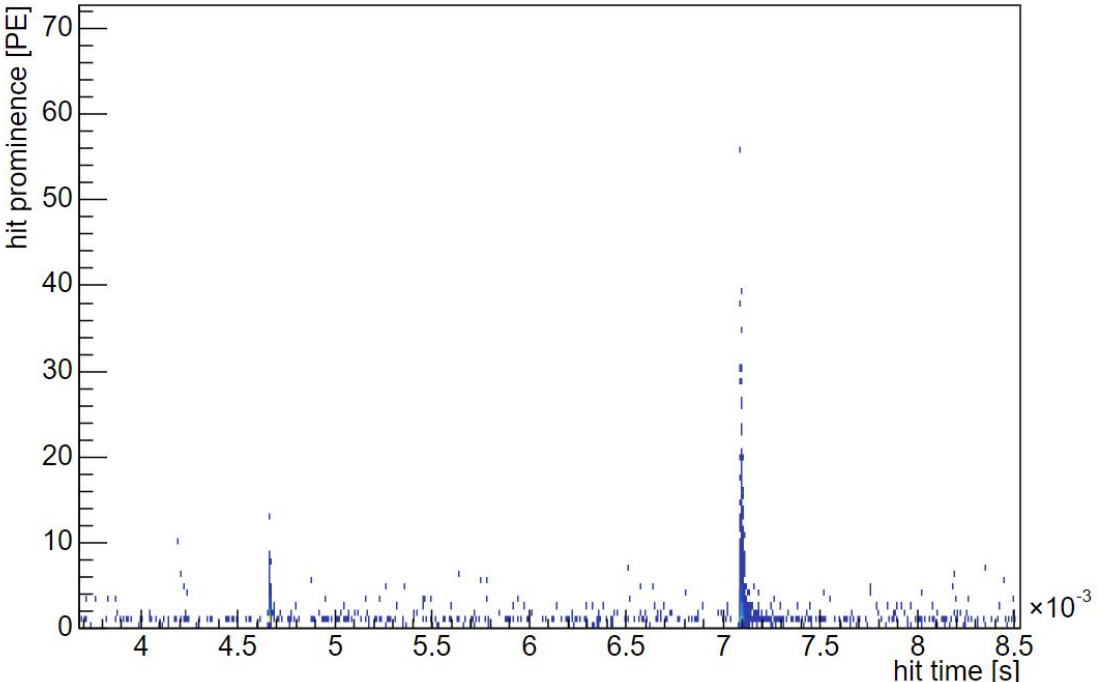
The parameters of the Gaussian Fits as a function of the true number of PEs

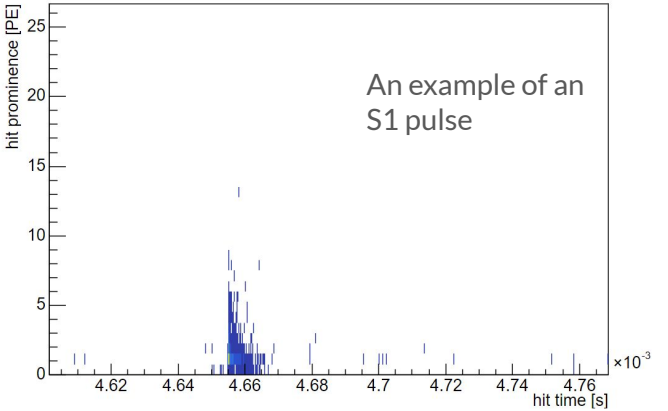
p0 is the y-intercept and p1 is the slope of the linear fit
The y-intercept is compatible with 0 within 1 σ

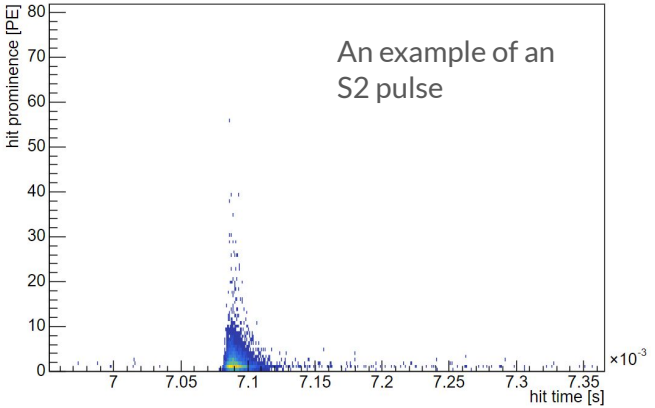
Pulse reconstruction and association

Output MC: Hits distribution

Looking at a fragment of 1 slice these plots are showing an example of a 39Ar event







How the PulseFinder works

➔ **First step: Moving average filter**
- kernel of 55 samples

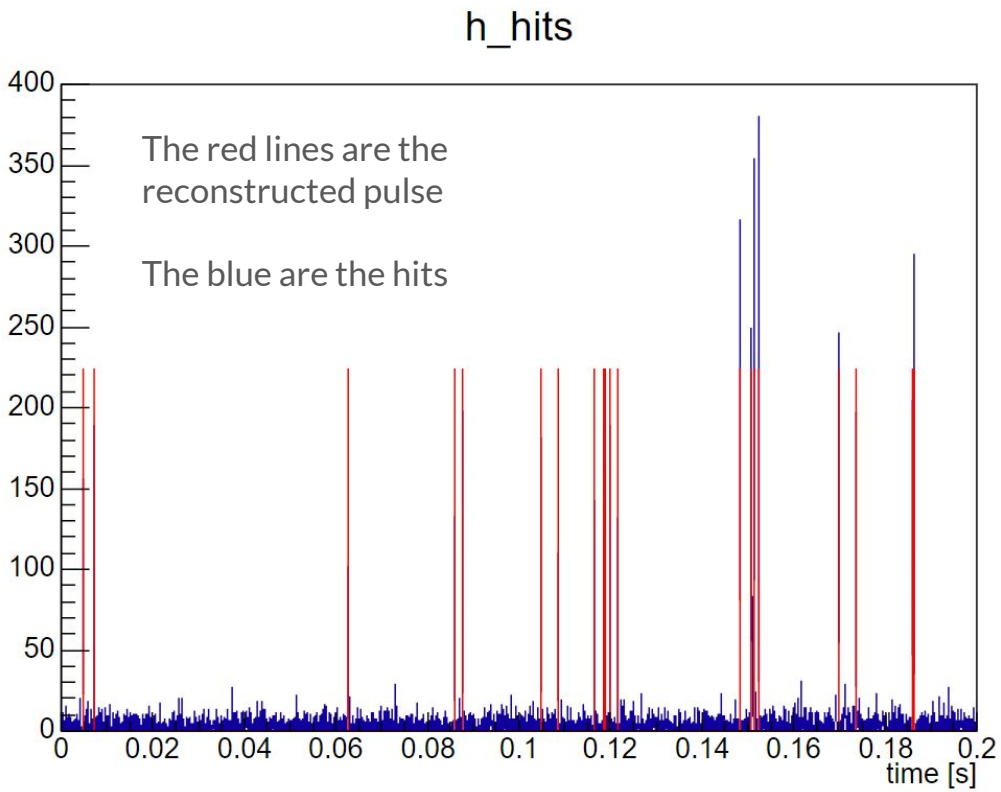
➔ **Second step :clustering**

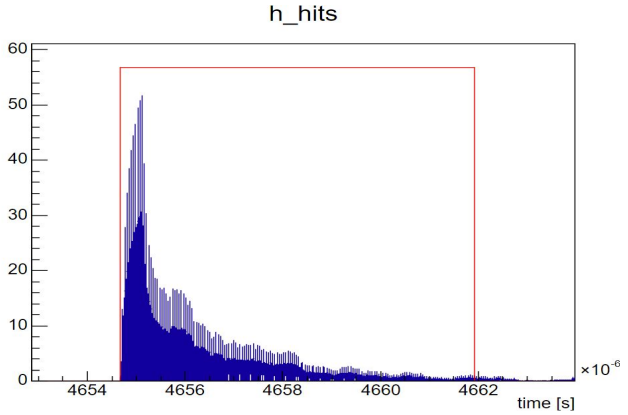
The PulseFinder clusterizes the hits based on some cuts:

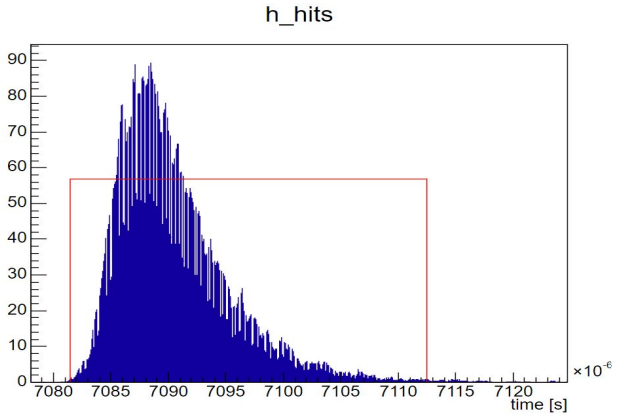
- The algorithm add an hit to the cluster only if it is closer than 30 ns to the last hit added to the cluster
- The algorithm saves the clusters only if the number of hits in the cluster is greater than 3

➔ **Third step : reclustering**
- Pulses found in the first step are clusterized together if they are closer than 15 us
- If one pulse smaller than 0.7 us from the first step is isolated is deleted

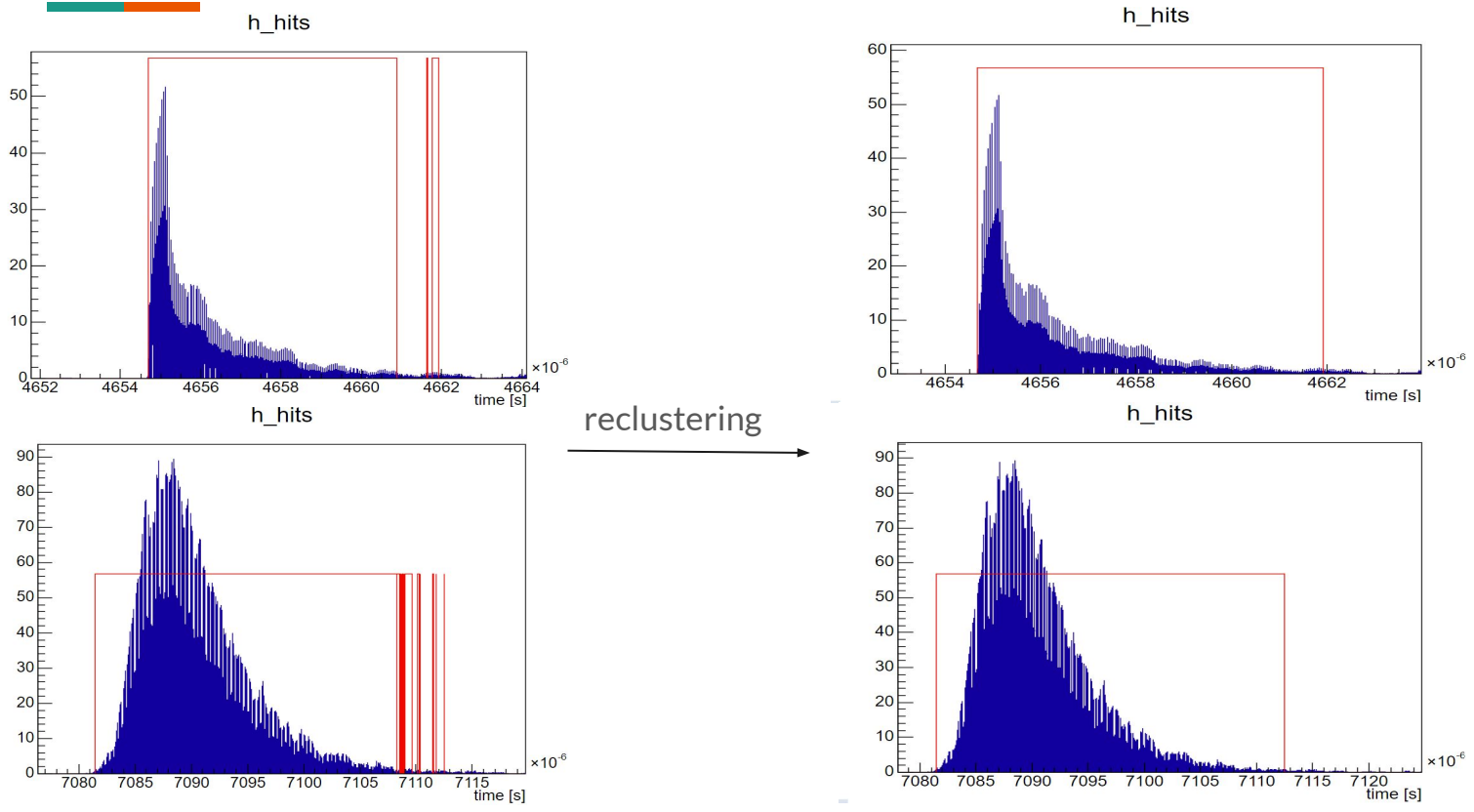
PulseFinder Output







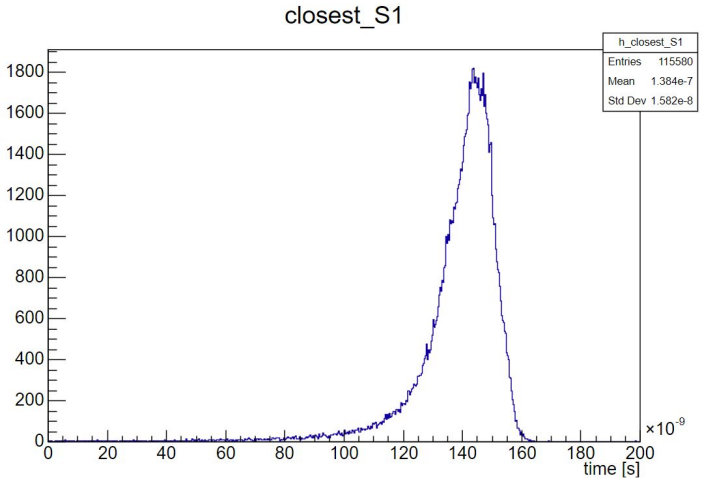
How the PulseFinder works: reclustering

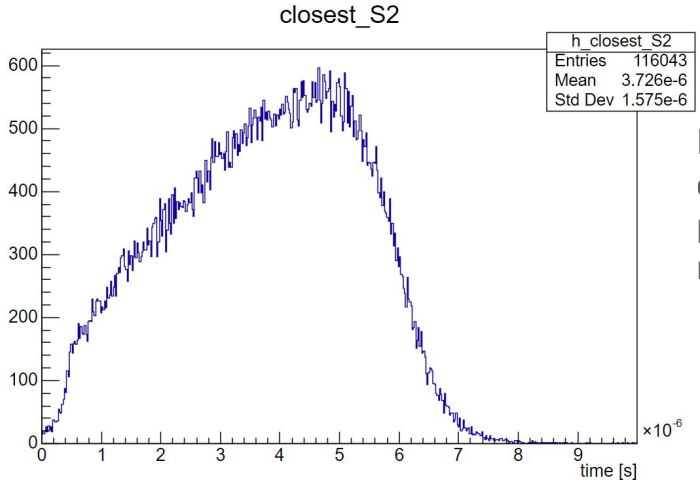


Association with the true pulses

For each true_event the associated S2 time is calculated summing the time of the event and the drift time based on the z position:

- Each true S1 or true S2 is associated with the nearest reconstructed pulse
- The algorithm checks that the association of a reconstructed pulse occurs only once

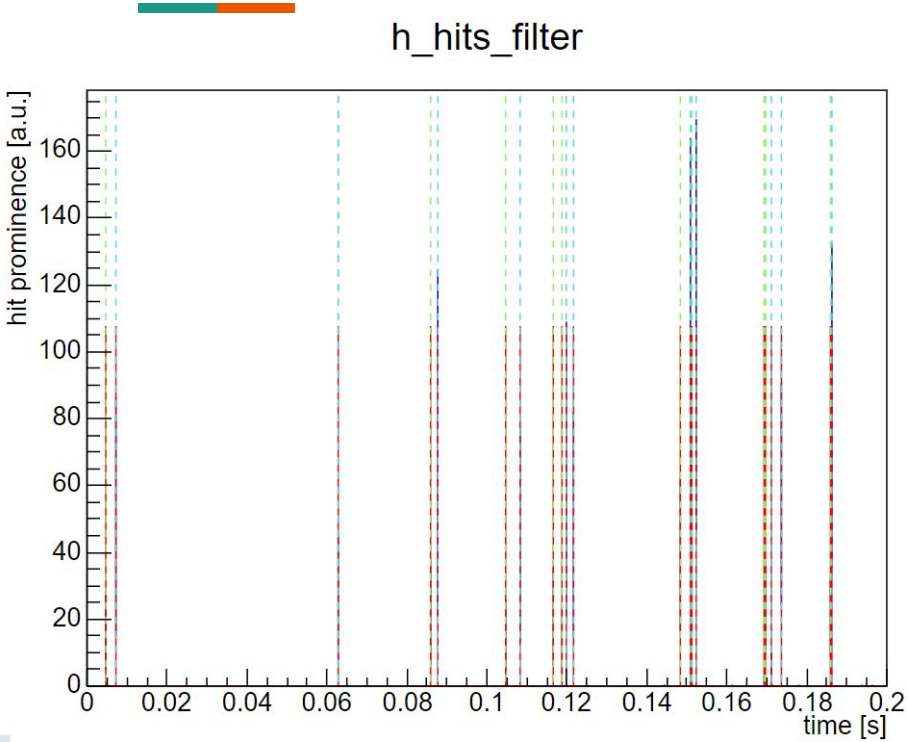


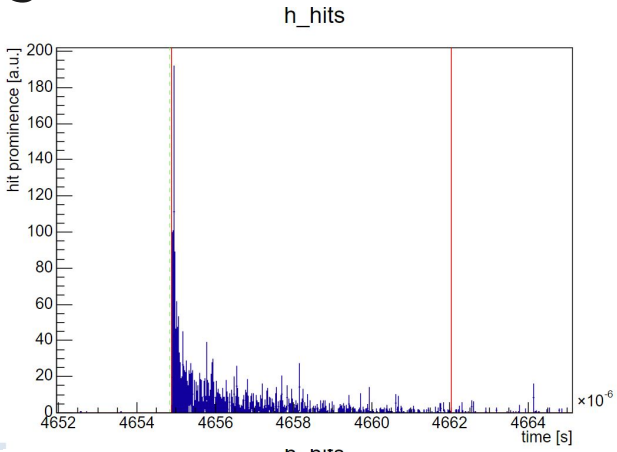


Here are the plots of the time distance between the true pulses and the associated reconstructed ones

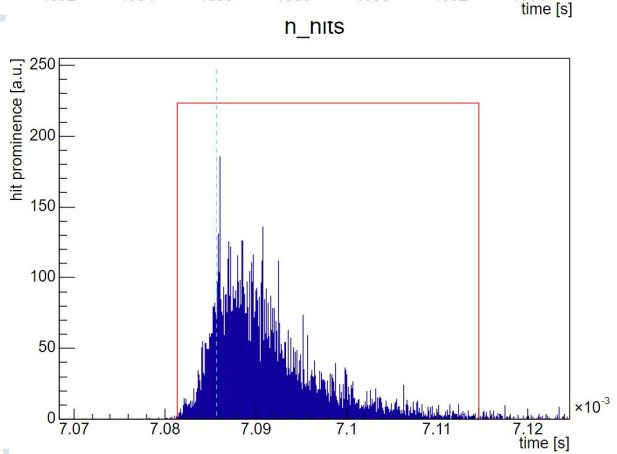
Association with the true pulses

here the waveform is not filtered



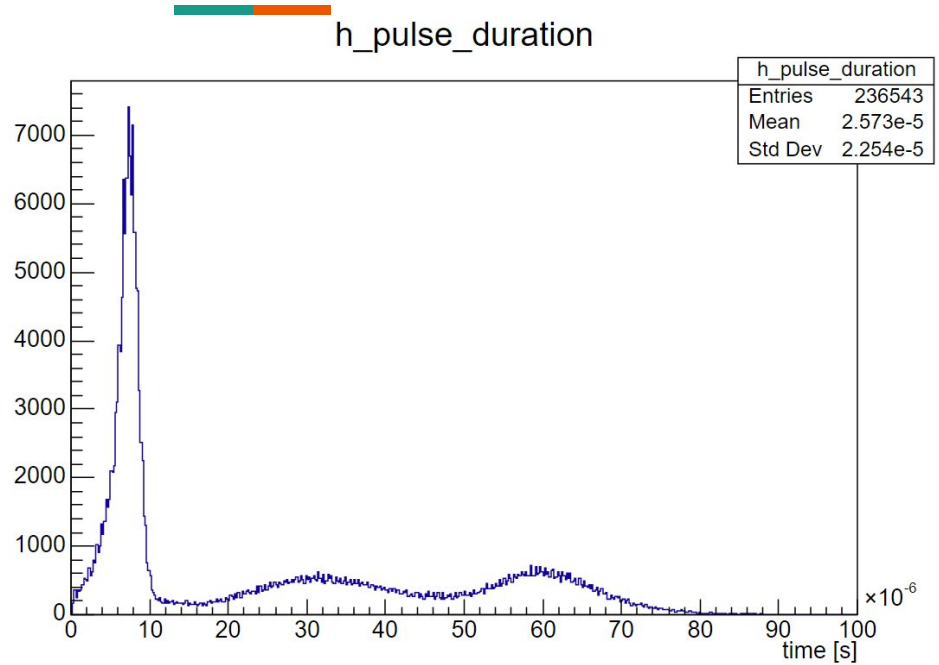


the green dashed line corresponds to the S1 true time

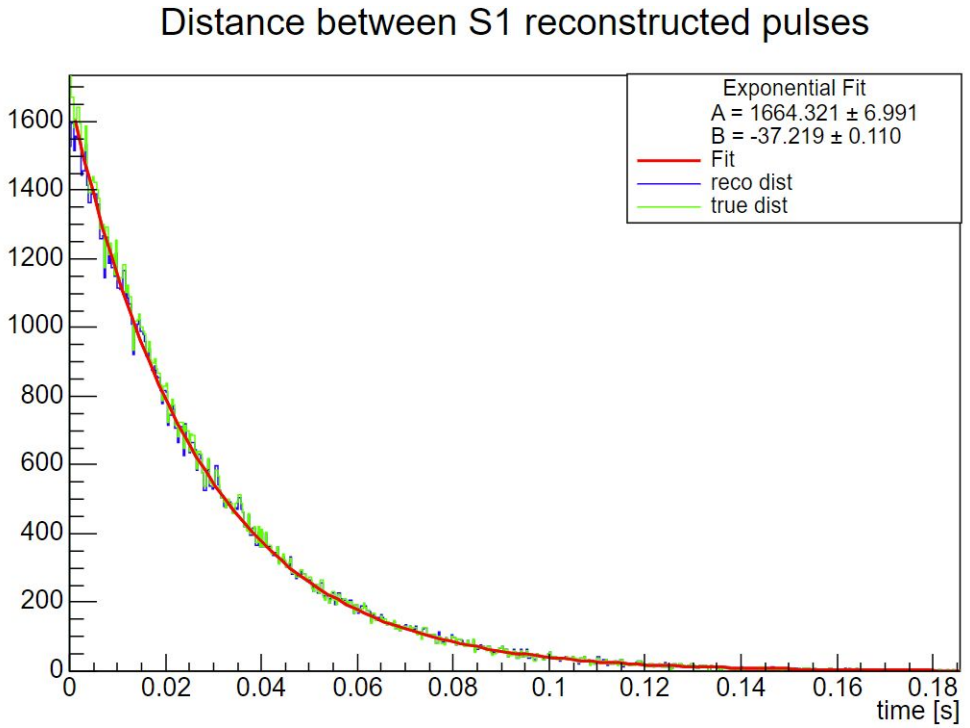


the light blue dashed line corresponds to the S2 true time

Reconstructed Pulses



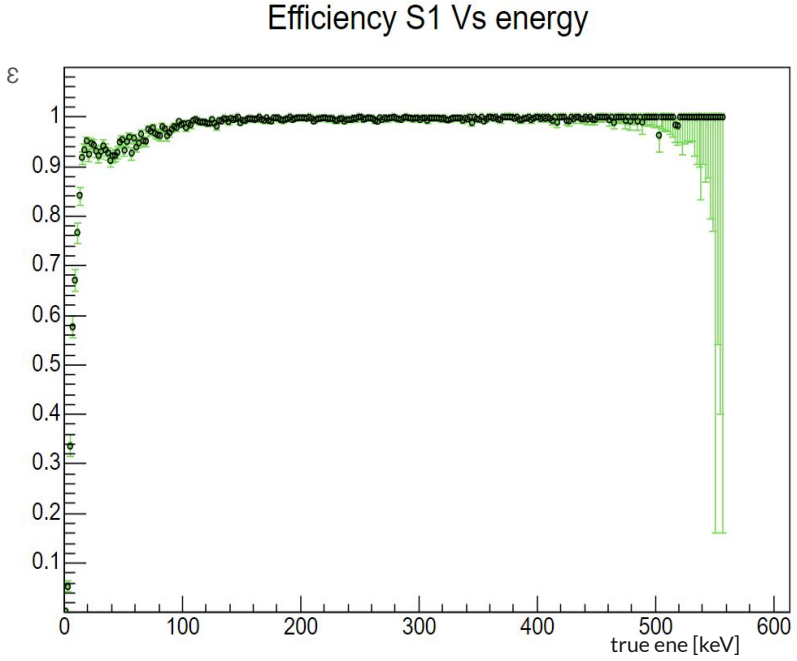
The distribution of the pulse duration is coherent with the typical width of an S1 and an S2

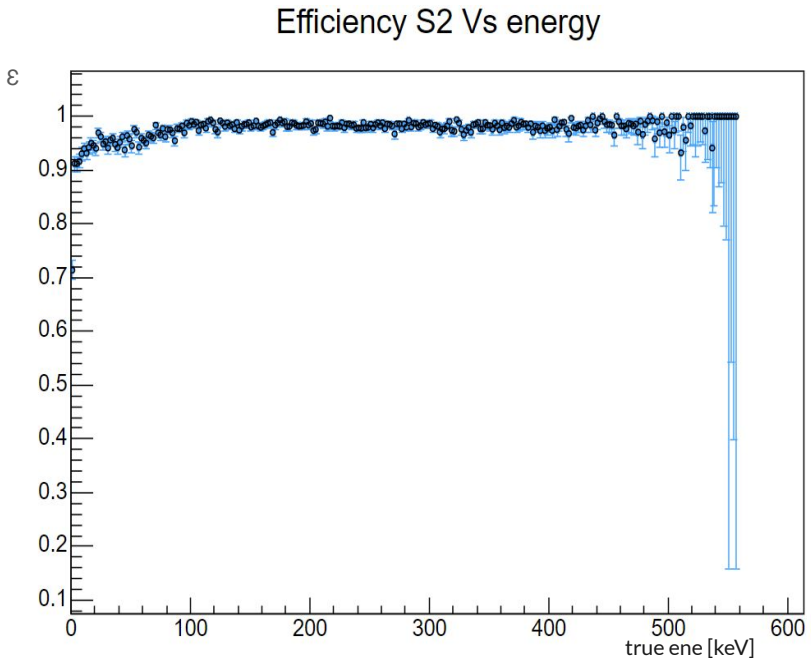


Histogram of the distance between pulses : the fit is of the form A*exp(B*x), where B is of the same order of the rate of true events of 39Ar, even if it is not compatible.

Reconstructed Pulses

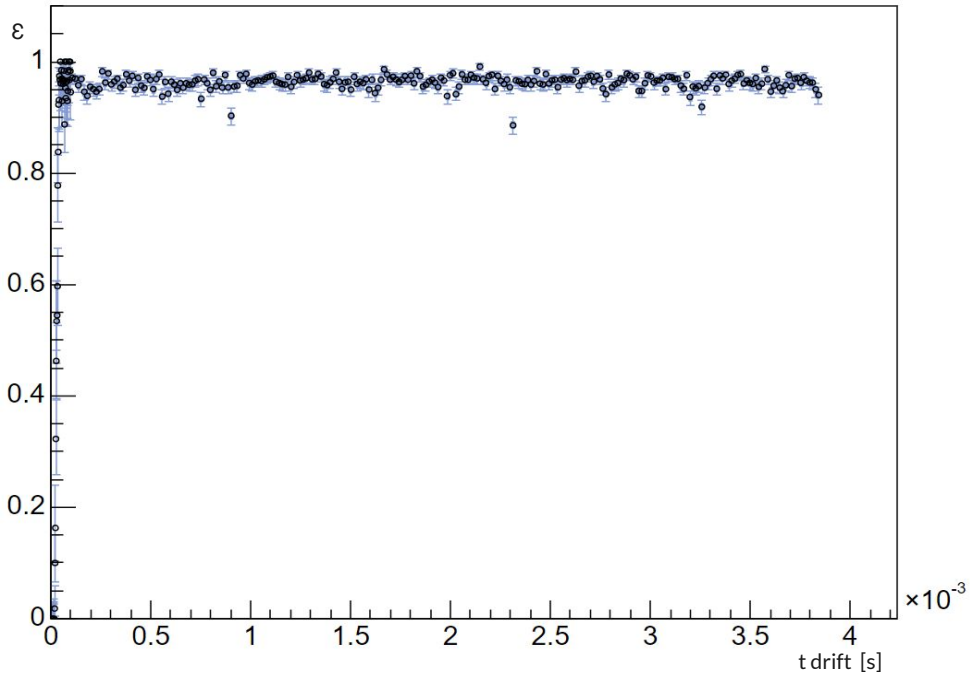
reconstruction efficiency

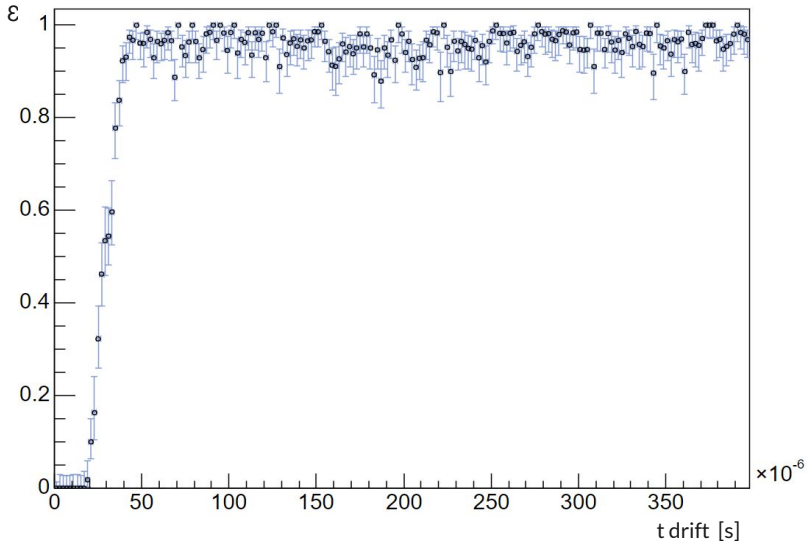




Reconstructed Pulses

Efficiency Vs Drift Time

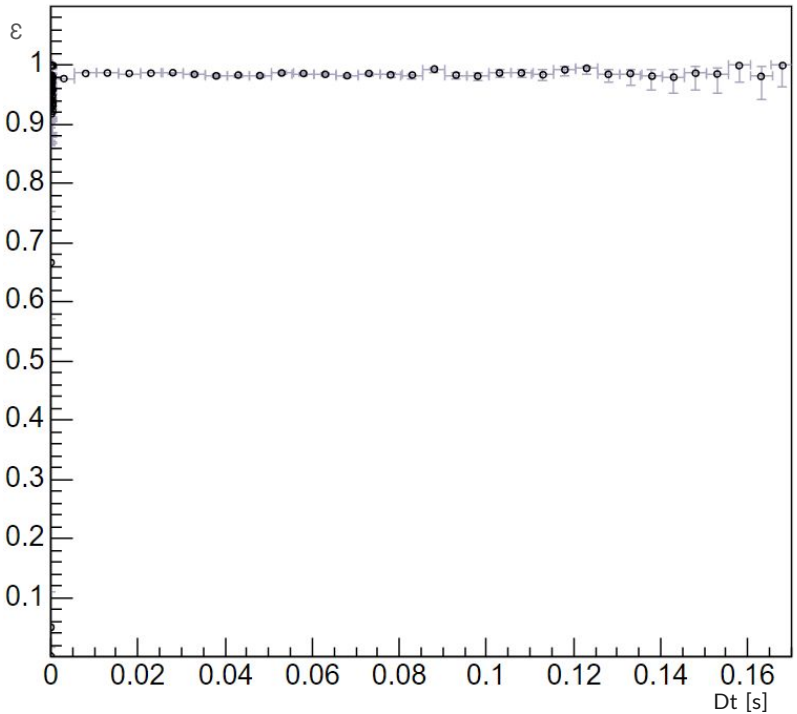


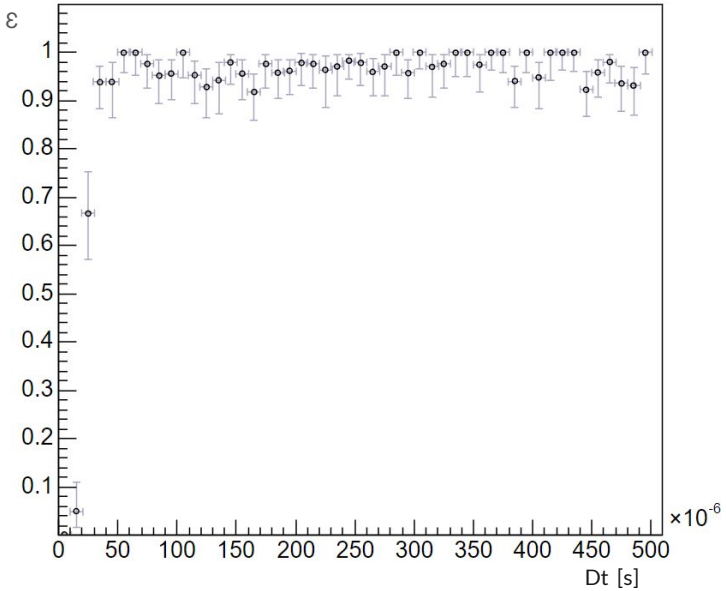


Zooming on the region at lower t_drift, the efficiency starts to decrease at orders of t_drift compatible with the cuts made in the pulse finder, in the reclustering (15 us)

Reconstructed Pulses

Efficiency Vs Dt of true events

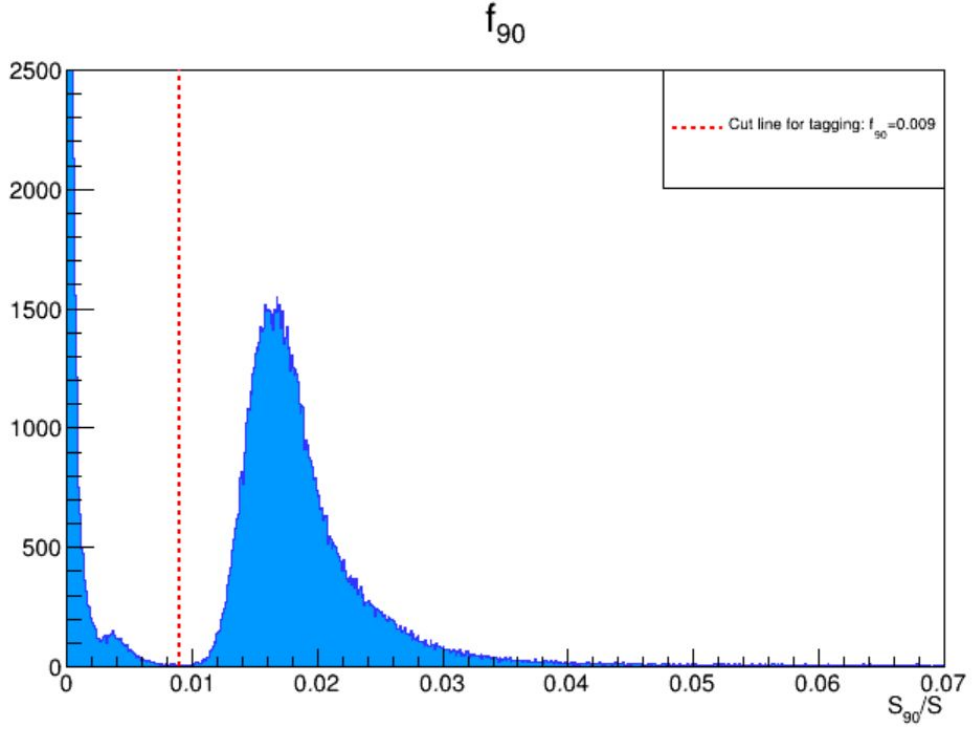


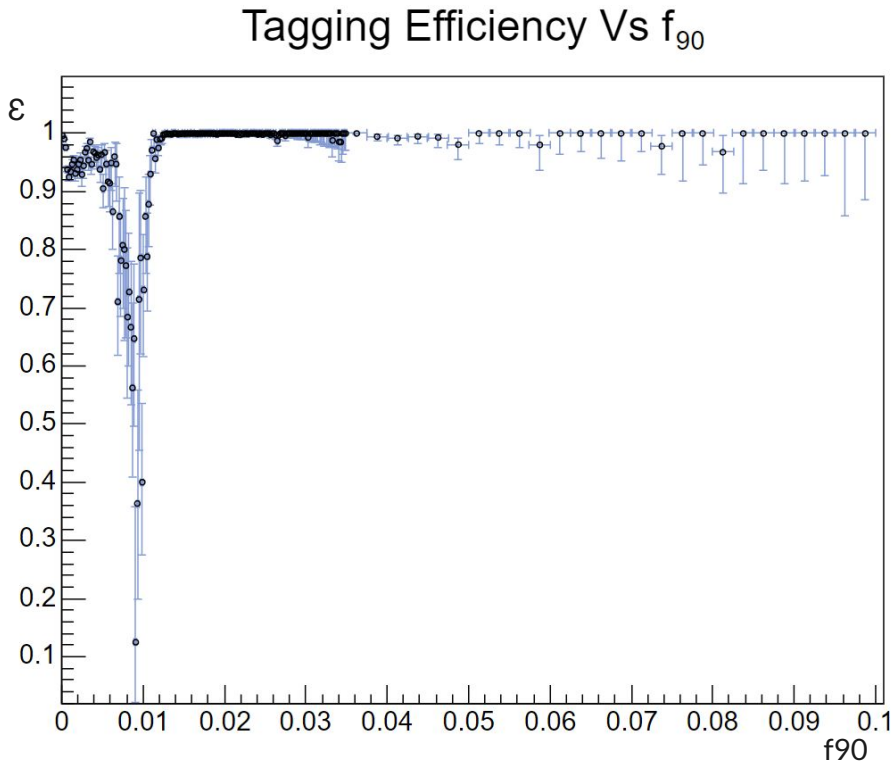


Zooming on the region at lower Dt, the efficiency starts to decrease at orders of Dt compatible with the cuts made in the pulse finder, in the reclustering step (15 us)

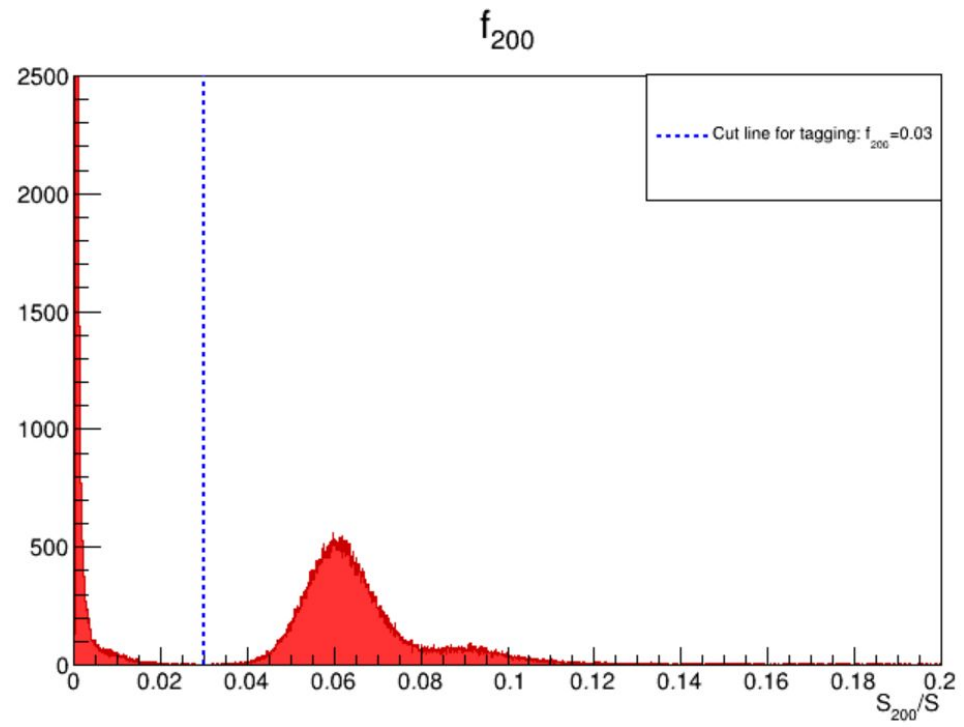
Pulse Classification

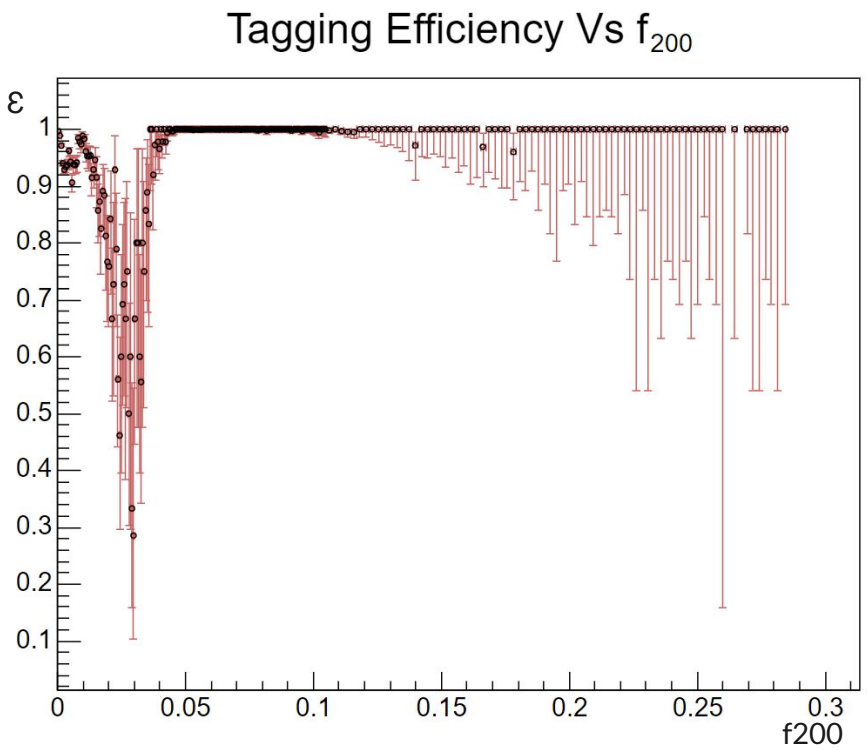
f_prompt: f_90


f_{90}
Cut line for tagging: f_{90}=0.009
2500
2000
1500
1000
500
0
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
S_{90}/S

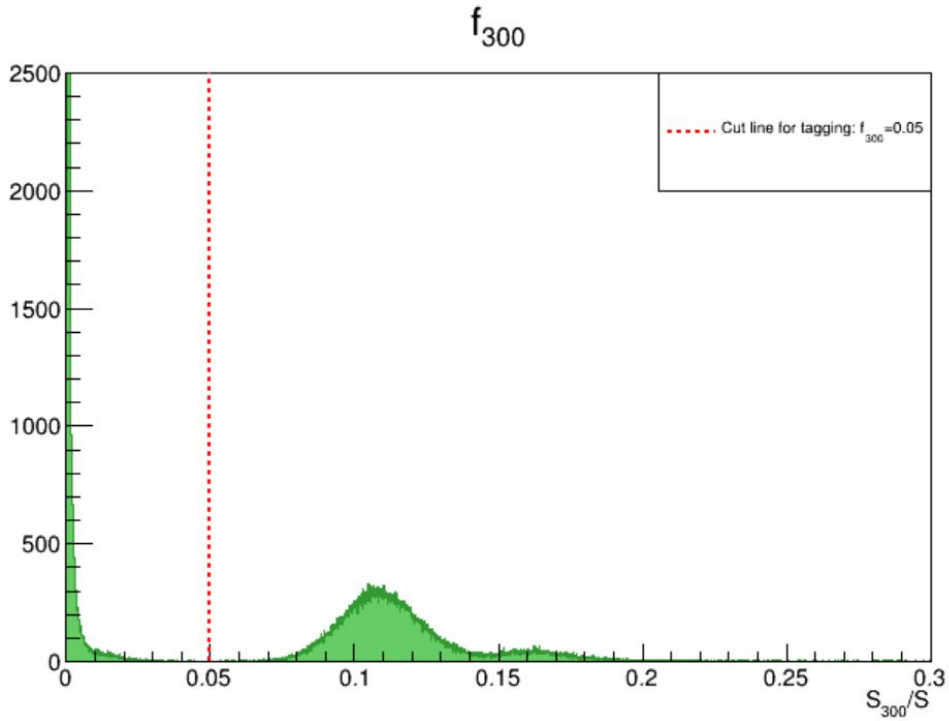

Tagging Efficiency Vs f_{90}
ε
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
f90

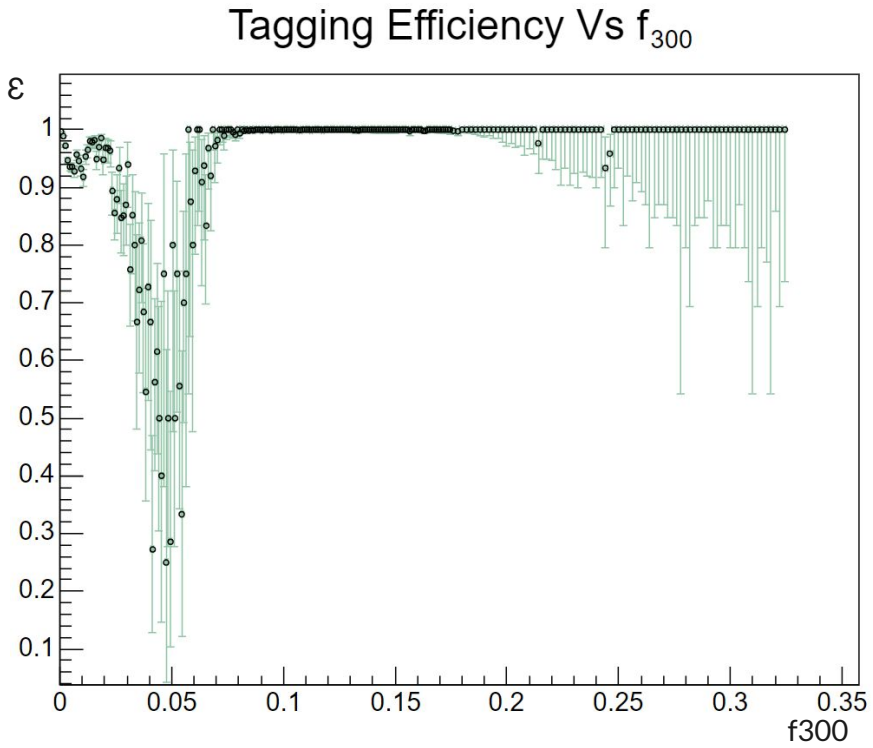
f_prompt: f_200



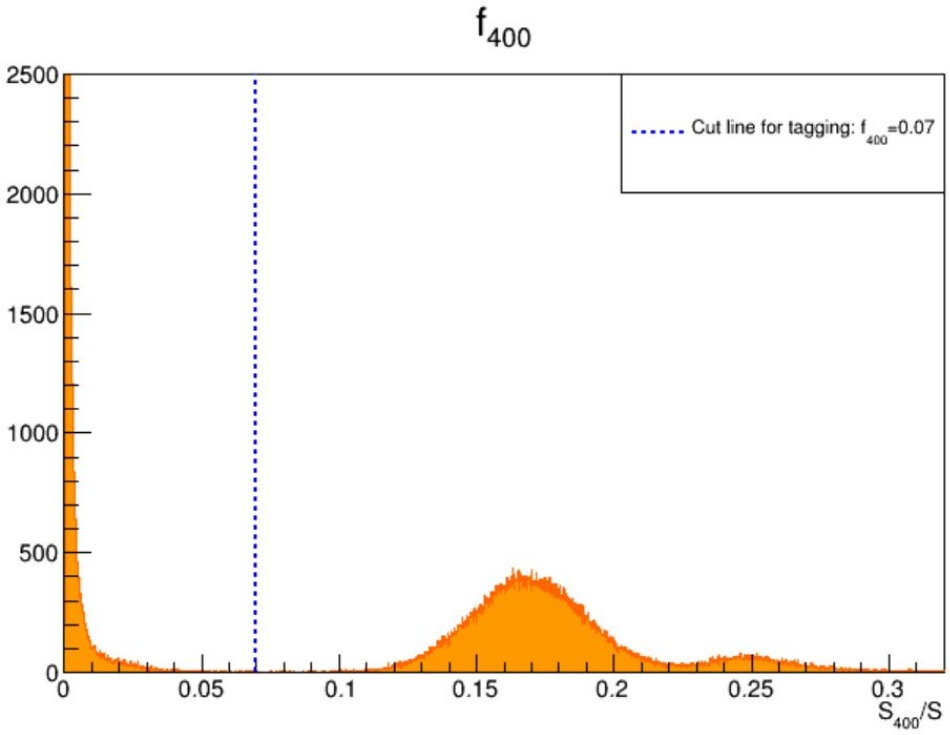


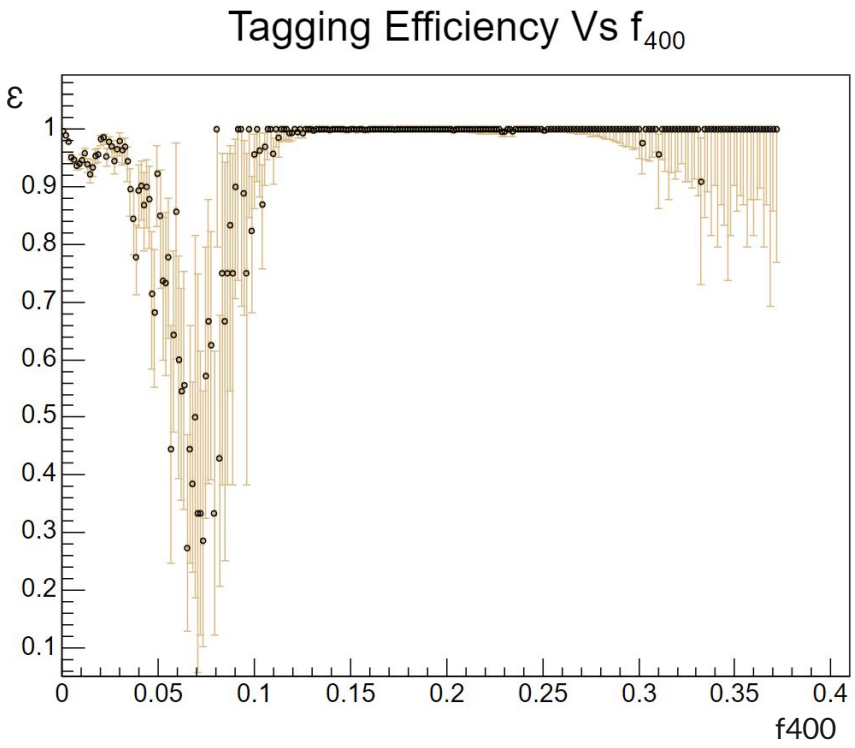
f_prompt: f_300



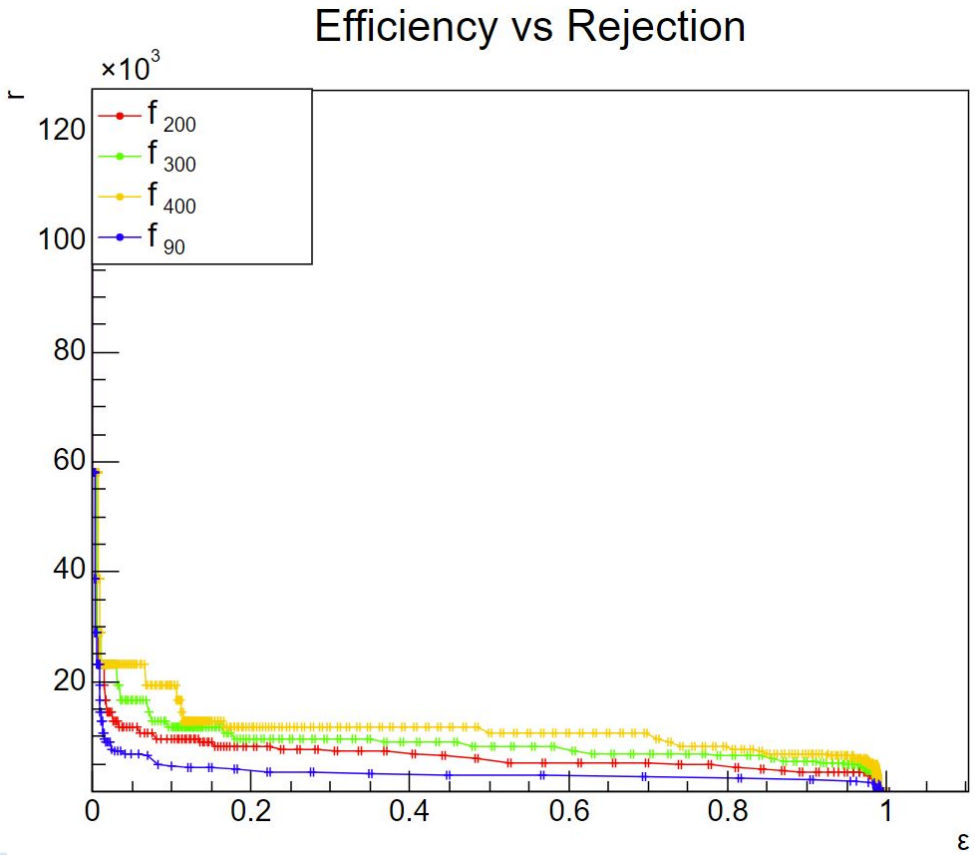


f_prompt: f_400





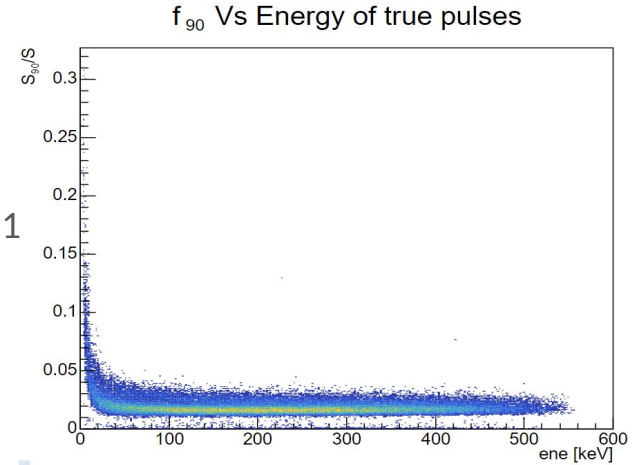
Rejection power and efficiency

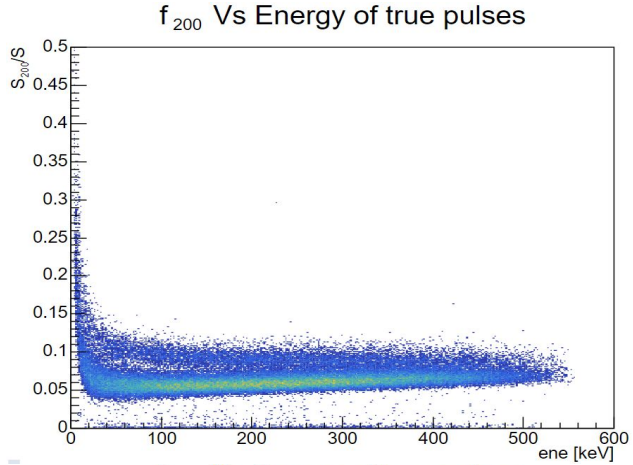


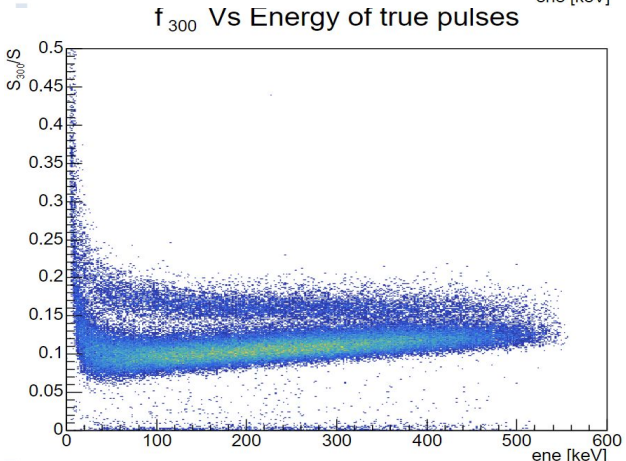
Changing the tagging cut the efficiency was evaluated recognisedS1/totS1 and the rejection power as NOTrecognisedS2/totS2

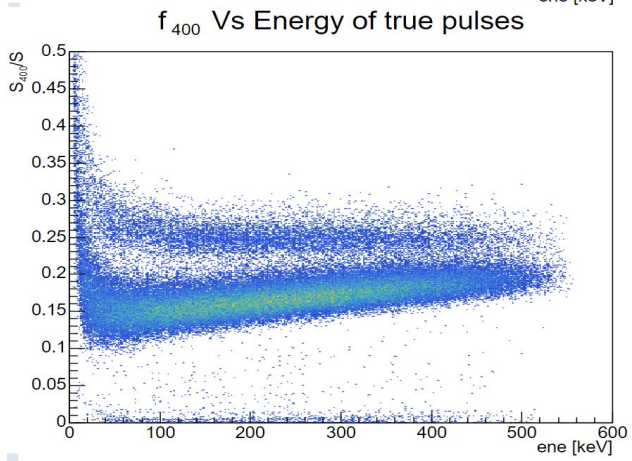
f_prompt Vs Energy

Selecting only S1 reco pulses associated with S1 true pulses

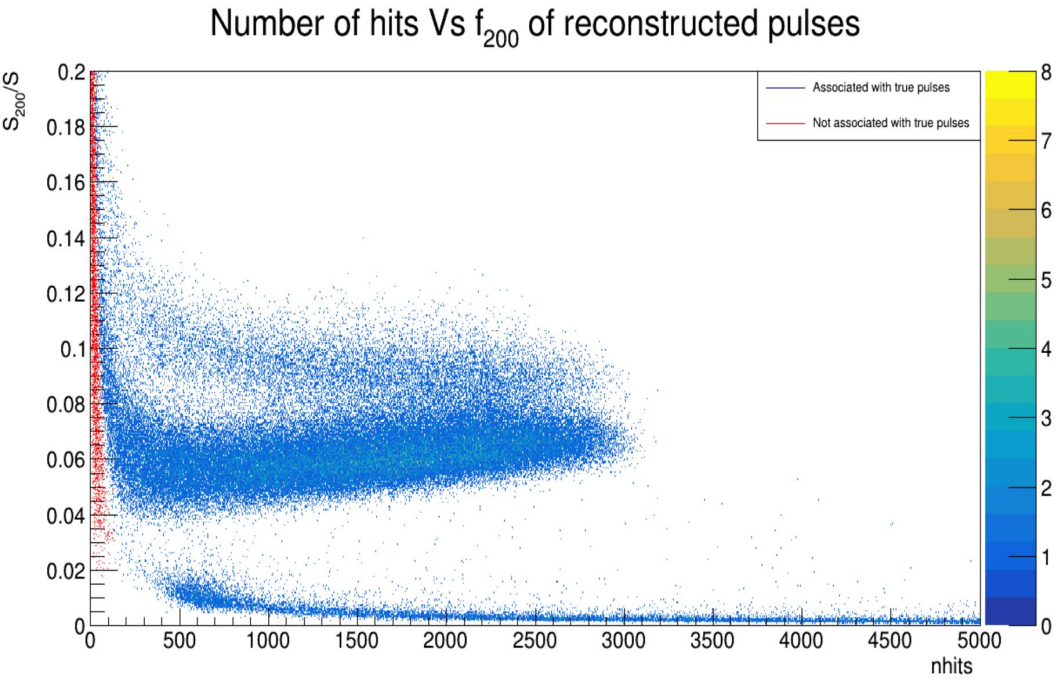






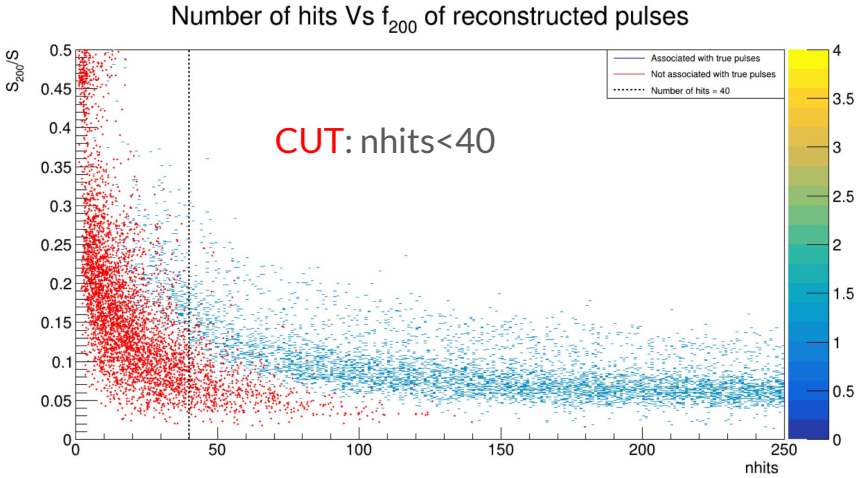


Identification of fake pulses

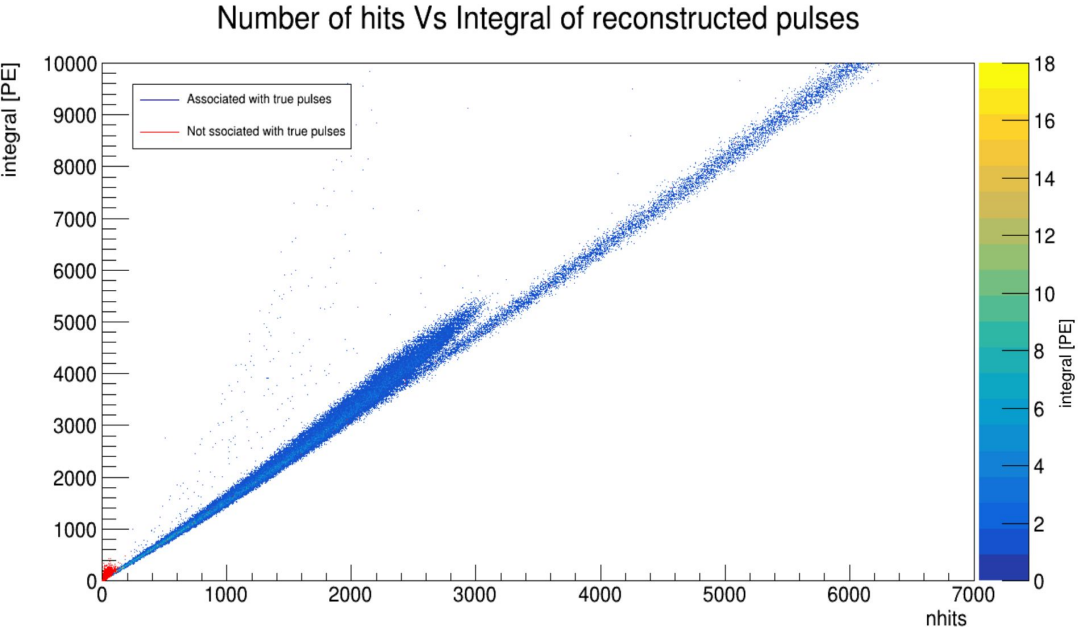


This plot shows a good choice of cut to remove fake pulses, based on the number of hits per pulse

In red the non-associated pulses, considered as fake pulses
In blue all the pulses associated with true pulses



Identification of fake pulses



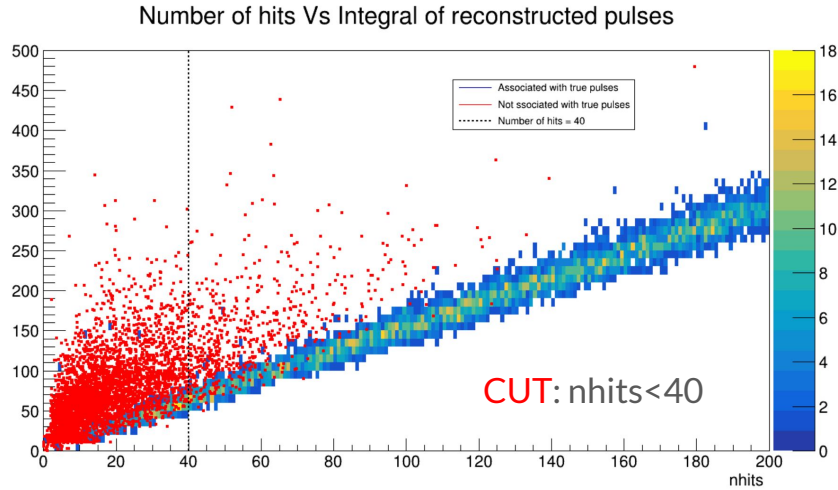
Looking this plot and the one shown in the previous slide, with a cut chosen at nhits=40

- number of cutted real pulses: 456 on a total of 231623 real pulses

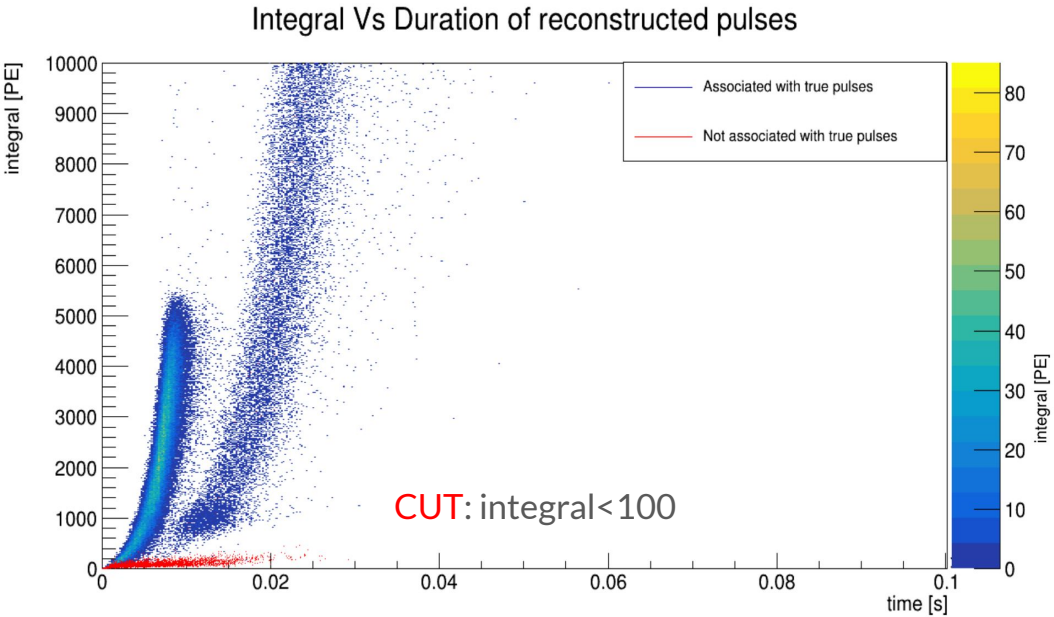
➔ efficiency ε=0.998+/-0.001

- number of cutted fake pulses: 4081 on a total of 4920 fake pulses

➔ rejection power 1/R=5.848+/-0.005



Identification of fake pulses



This plot shows another possible choice of cut based on the pulses' integral :

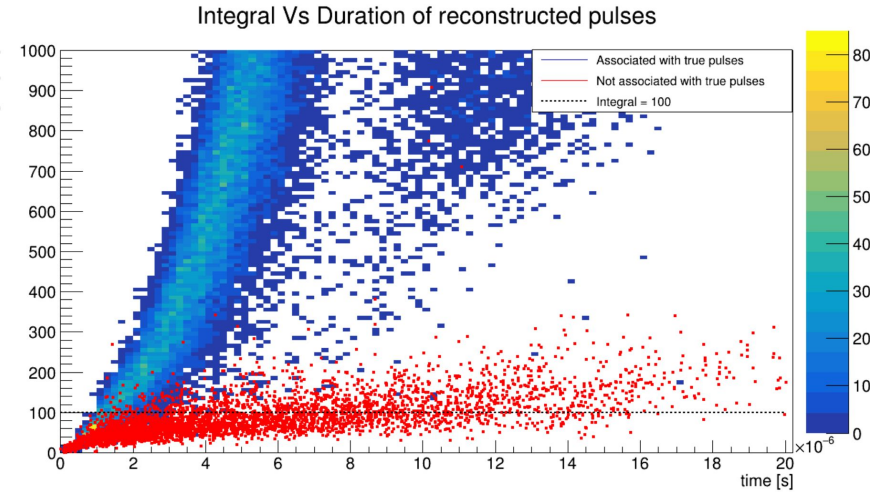
- number of cutted real pulses: 928 on a total of 231623 real pulses

➔ efficiency Ɛ=0.996+/-0.001

- number of cutted fake pulses: 3400 on a total of 4920 fake pulses

➔ rejection power 1/R=3.24+/-0.007

The better choice is the cut on nhits .



Next Steps

Improvements on the PulseFinder algorithm:

- Cut on Dt between reco-true, efficiency study
- Cut in the reclustering for isolated pulses must be lowered for NR (generation of 40Ar sample)
- Study on the pulse duration distribution

Analysis of the Co-60 sample:

- Pulse reconstruction with the Pulse Finder
- Tagging efficiency based on f_prompt