The GRETINA tracking code

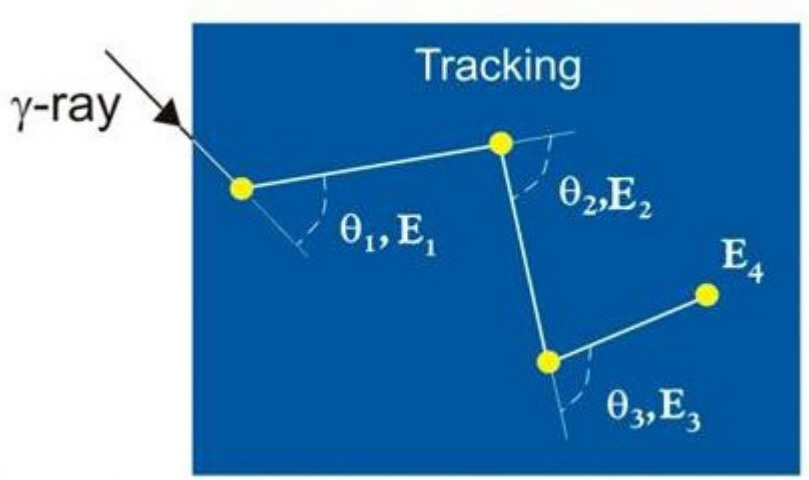
- Intro, 'review'
- How to use

  Few new developments:
- efirst function
- elast function

  Other tests

Torben Lauritsen, ANL
Amel Korichi, CSNSM

Tracking 101: determining the interaction sequence and how 'good' a gamma ray is



Cluster, then find interaction sequence

Evaluate scattering angle <–> energy consistency with the Compton scattering formula:

$$E'_\gamma = \frac{0.511}{1 + \frac{E_\gamma}{0.511}(1 - cos(\theta))}$$

FOM < ~0.6-0.8 considered GOOD

FOM > ~0.8 considered BAD (Compton events)

The GT

$$FOM = \frac{\sqrt{\sum_i (\theta_i^{theo} - \theta_i^{obs})^2}}{n_i - 1}; n_i > 1$$

(in rad)

Not the only one possible

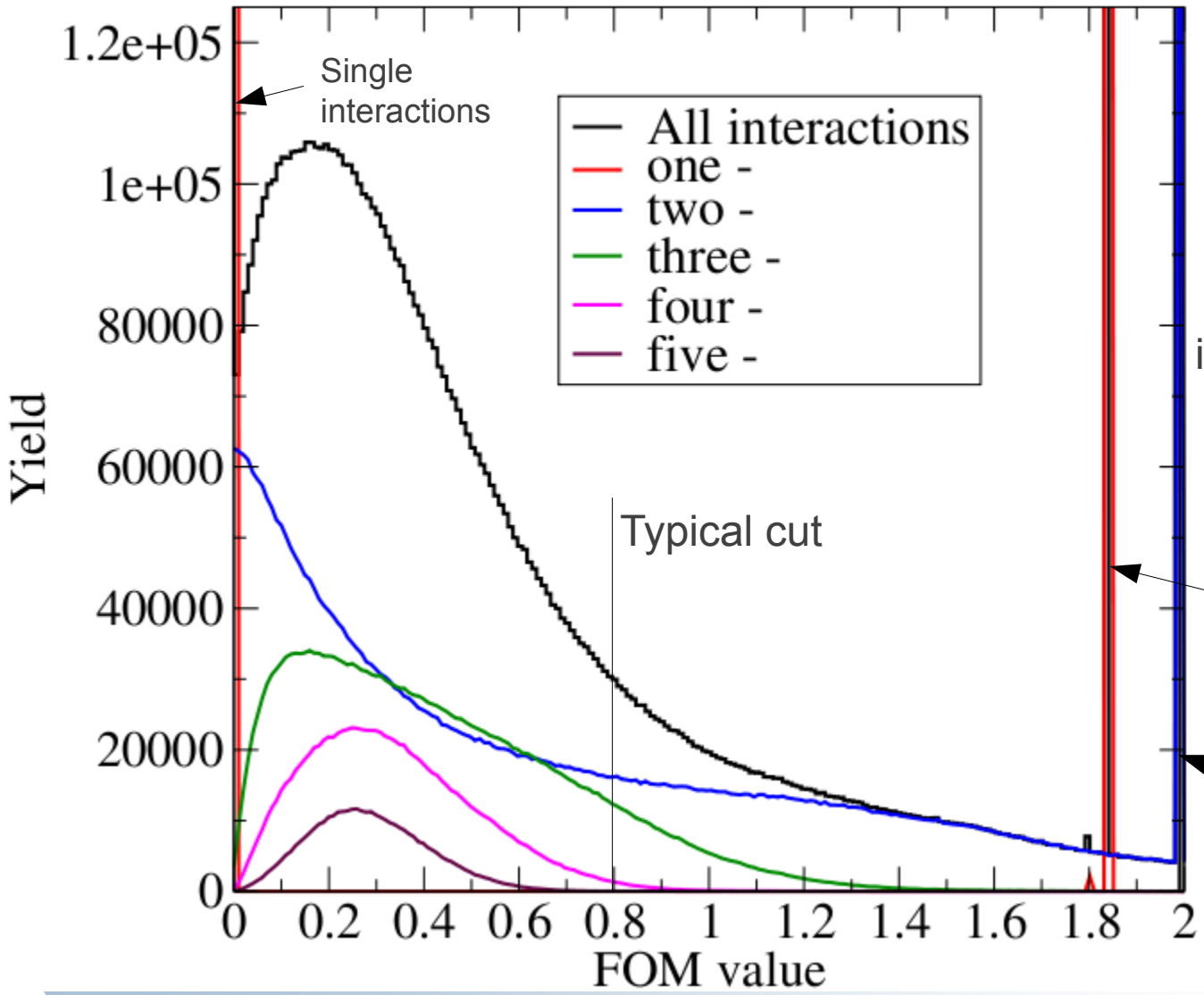
Note: Single interactions cannot be tracked

1: We find the interaction sequence

2: We evaluate how 'good' the gamma rays is

(BTW: We re-scale to CC energy before tracking)

FOM: ***a measure of how well the interaction angles and interaction energies follow the Compton scattering formula for the interaction points in a gamma ray.*** ***Typical spectrum of FOM values:***



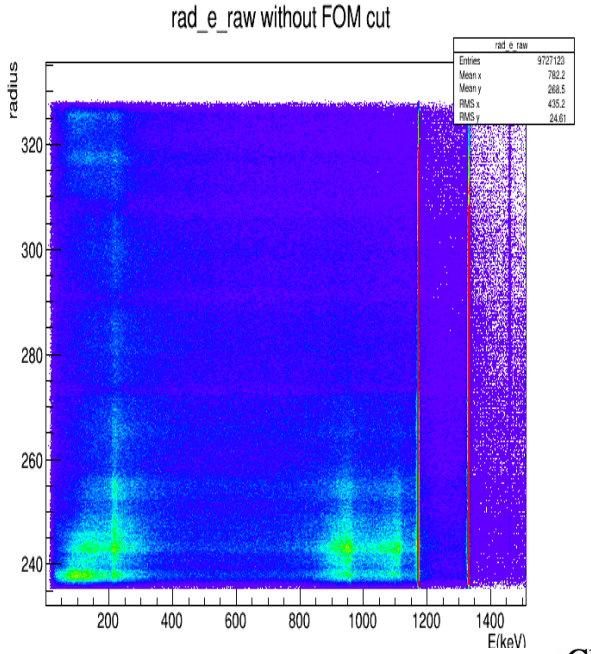
Typical FOM cut: 0.8 (up to the user to choose)

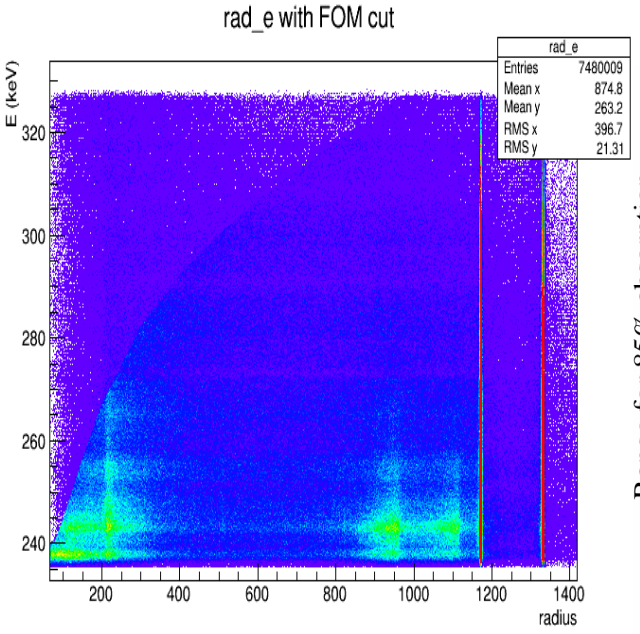
Automatically, preferentially cut two interaction gamma rays

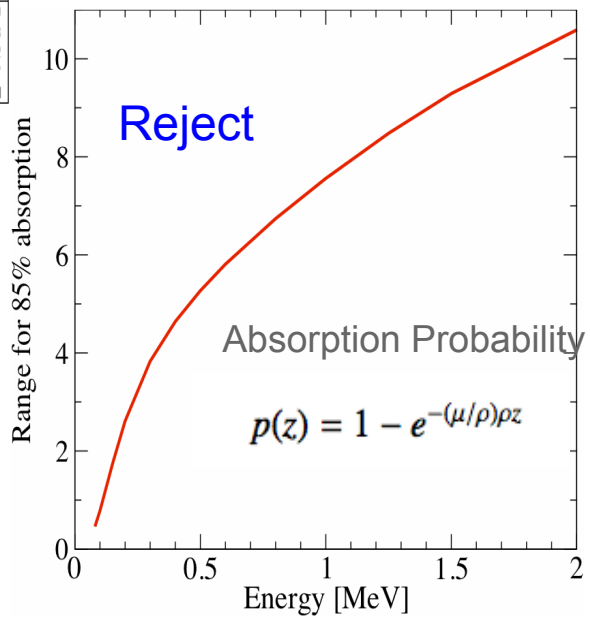
Over range single interactions

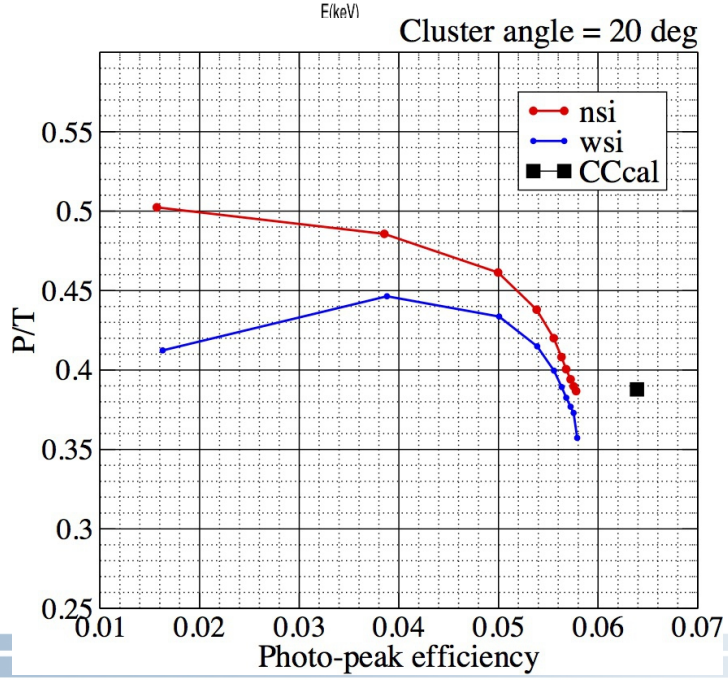
overflow

For single hits: We can improve the tracking









The same concept applied to OFT :

Improvement for AGATA@GANIL

Selected Chat file options:

```
./trackMain \
  track_GT.chat  \
  GTDATA/mode2.dat  \
  GTDATA/mode1.gtd >
GTDATA/trackMain.log
```

```
dtwin          30
target_x 0
target_y 0
target_z 0
CCcal CCenergy.cal
useCCEnergy
clusterangle   1  20
clusterangle  30  20
enabled "0-180"
trackingstrategy 1 0
trackingstrategy 2 0
trackingstrategy 3 0
trackingstrategy 4 0
trackingstrategy 5 0
trackingstrategy 6 5 ggtttt
trackingstrategy 7 5 gggtttt
trackingstrategy 8 5 gggttttt
```

(10 nsec units)

GTAG1

```
recluster1   0.01  0.1  3 10  0.90
nprint 20
singlehitmaxdepth 23 1.9 18.5 1.0
0.000 0.59
.
.
8.000 10.17
10.00 10.01
16.3  20.0
```

There are many more options!
Here we just show the basic ones.

We add mode1 data to the mode 2 data!!!!

BTW: We can handle AGATA data too!

Some functions in ANL tracking code

- Single interaction range [GTAG1]
- Splitclusters: try to split clusters that have a bad FOM into two gamma rays that have good FOMs. [SUMMED LINES!?]
- Combine clusters: try to combine that have bad FOMs into one gamma rays that has a good FOM
- Recluster: split gamma rays with bad FOM decreasing the clustering angle. [TBD: *can go the other way too*]
- Matchmaker: combine two single interaction gamma rays into one gamma ray with a good FOM [tricky!]
- PairProd: TBD

GTAG1

We can execute these functions iteratively until we have made the best out of the data we were given

The problem: ***sometimes we make the wrong call because the experimental data is not perfect (i.e., we accidentally destroy what were actually good gamma rays)***

What is new?

J. van der Marel and B. Cederwall. **Backtracking as a way to reconstruct Compton scattered gamma rays.** *Nuclear Instruments and Methods in Physics Research Section A*, 437:538, 1999.

> See if adding a penalty if the last interaction is not in the range from ~10 to ~300keV helps, [elast function]

I. Piqueras, F. A. Beck, E. Pachoud, and G. Duchene. **A probabilistic γ-ray tracking method for germanium detectors.** *Nucl.Instrum.Methods Phys.Res.*, A516:122, 2004.

> Add a penalty if the first interaction point does not have the largest energy deposit, if the energy is larger than ~500keV, [efirst function]

Can be gauged by the Doppler correction

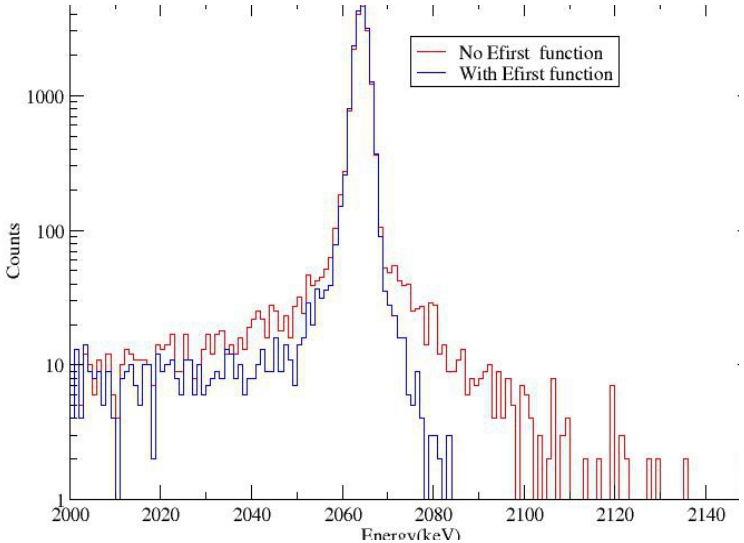
Look at 2064 keV line of in-beam experimental data. v/c~8.2%, so first interaction can be gauged by the Doppler correction

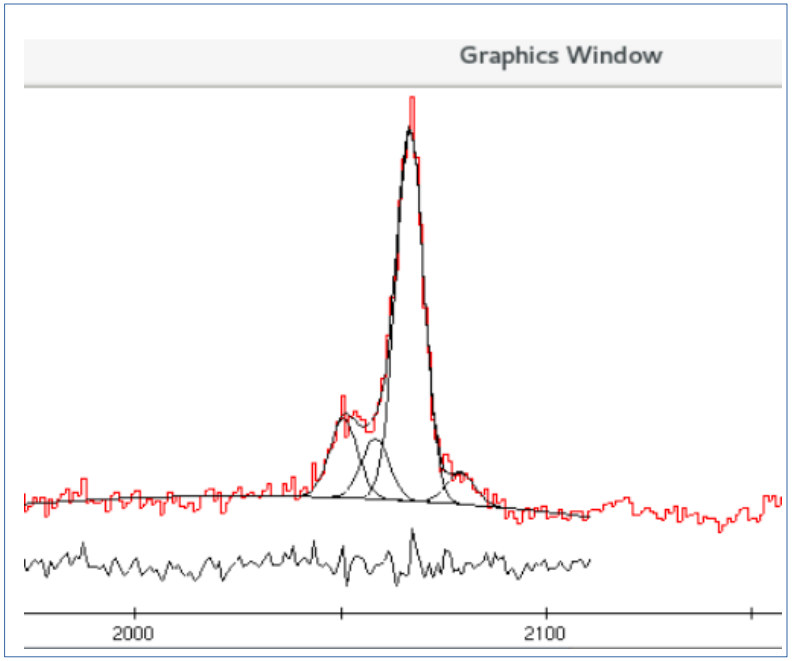
Efirst function :

Tested using simulated and experimental data
(Mater project of V. Lellaseux)

$^{12}C(^{84}Kr[394MeV], 4n)^{92}Mo$

in-beam experimental data. v/c~8.2%,
so first interaction can be gauged by the Doppler correction





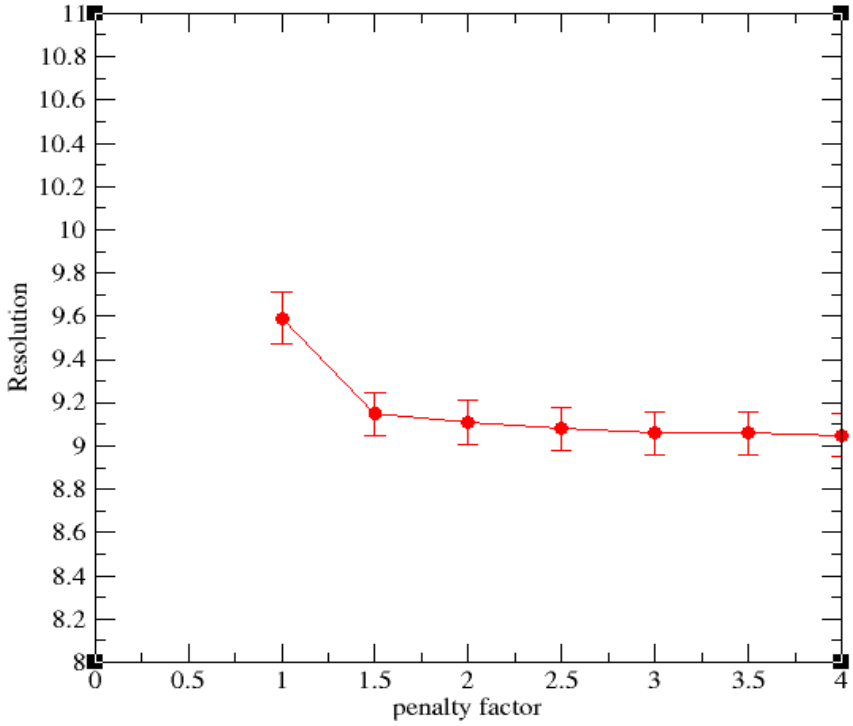
Fit results

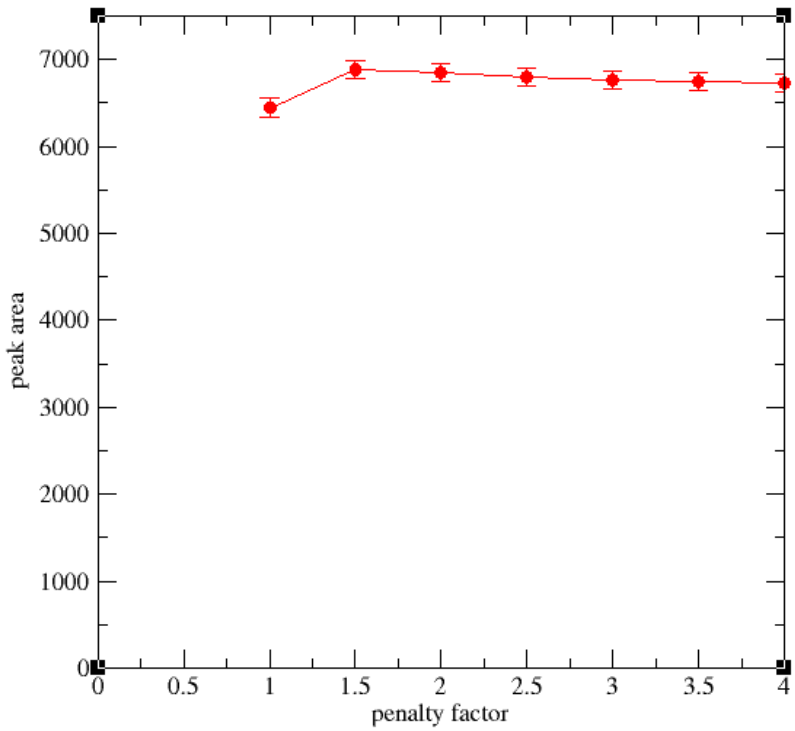
Energy	Width	Area	X2	
2064 keV	7.98(13)	8116(164)	1.437	without Efirst
2064 keV	7.70(12)	8604(170)	1.154	with Efirst

The peak width is reduced by 5(2)%
The counts in the peak goes up by 7(3)%

Efirst function

```
#       + min gamma ray energy (MeV)
#       |       + penalty factor
efirst 0.500 3.5
```





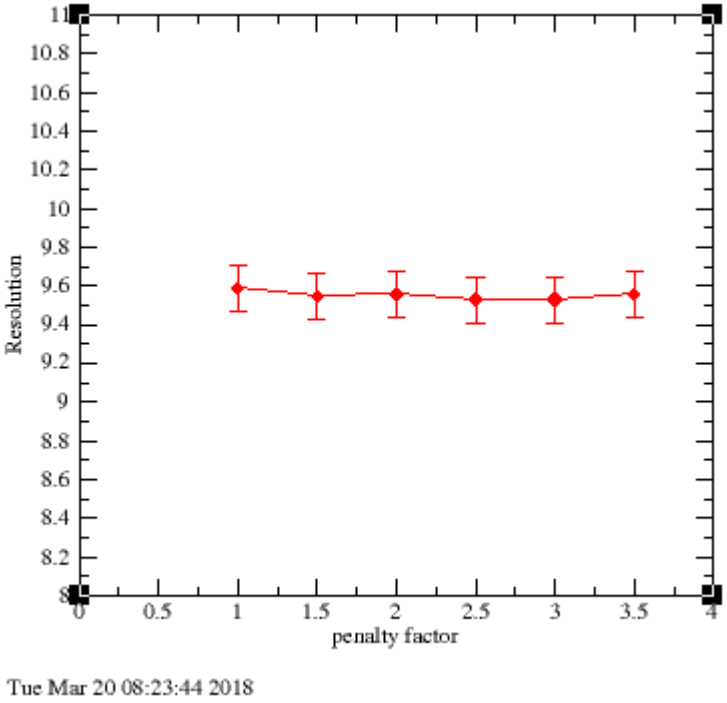
Function seems to help us find the first interaction point better :)

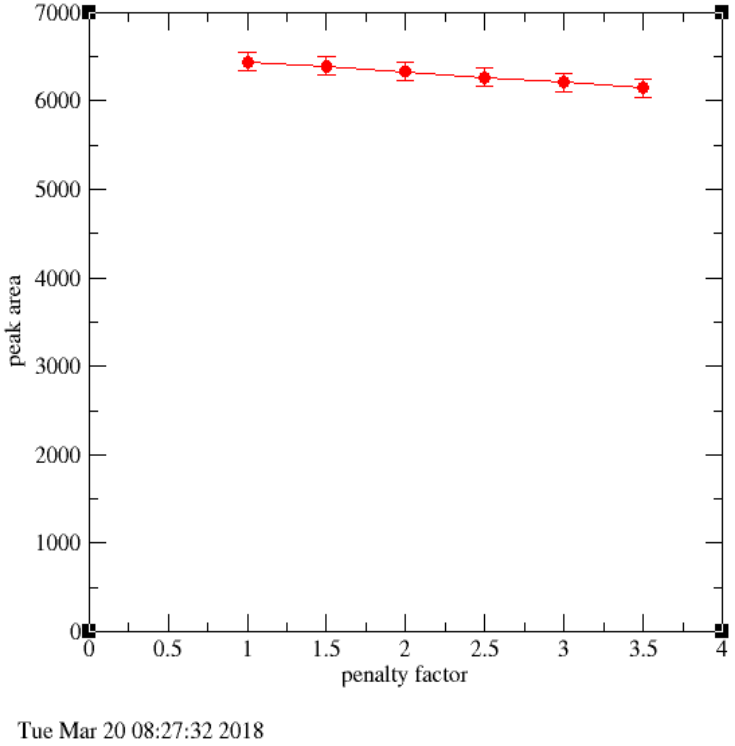
~5% better energy resolution, ~5% more peak area for the same FOM cut

elast function:

```
#       +-- lowest accepted energy (MeV)
#       |       +-- highest accepted energy (MeV)
#       |       |       +- penalty factor
elast 0.050 0.300 1.5
```

As a function of the penalty factor, resolution and peak area:

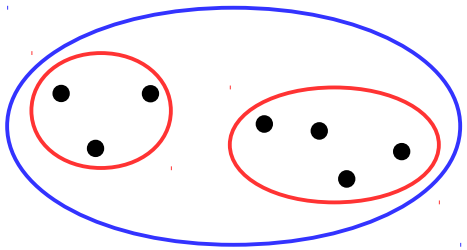




… does not seem to help us

Combine cluster function :

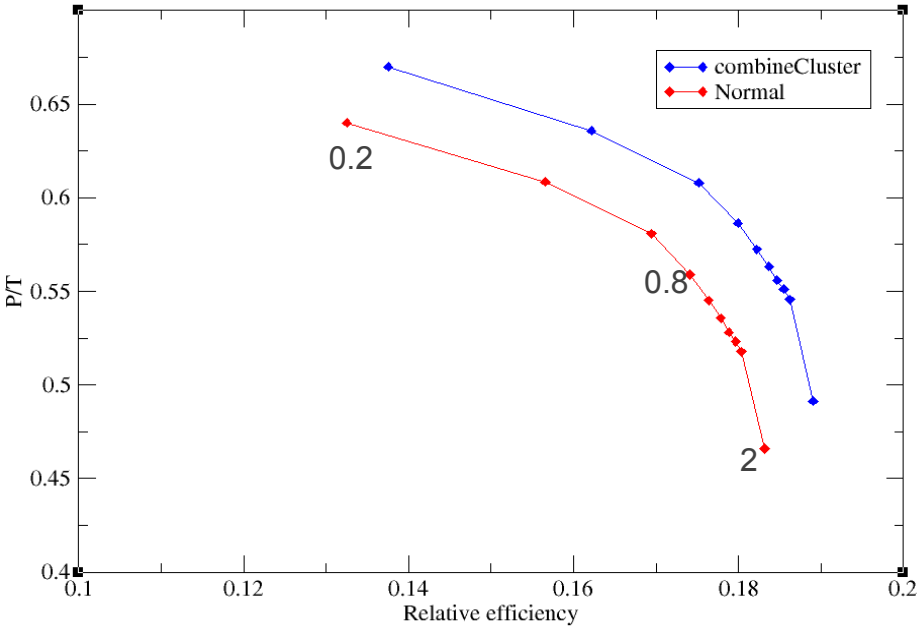
If a cluster has a bad FOM, we merge it with another cluster of the same event and check if FOM is better.



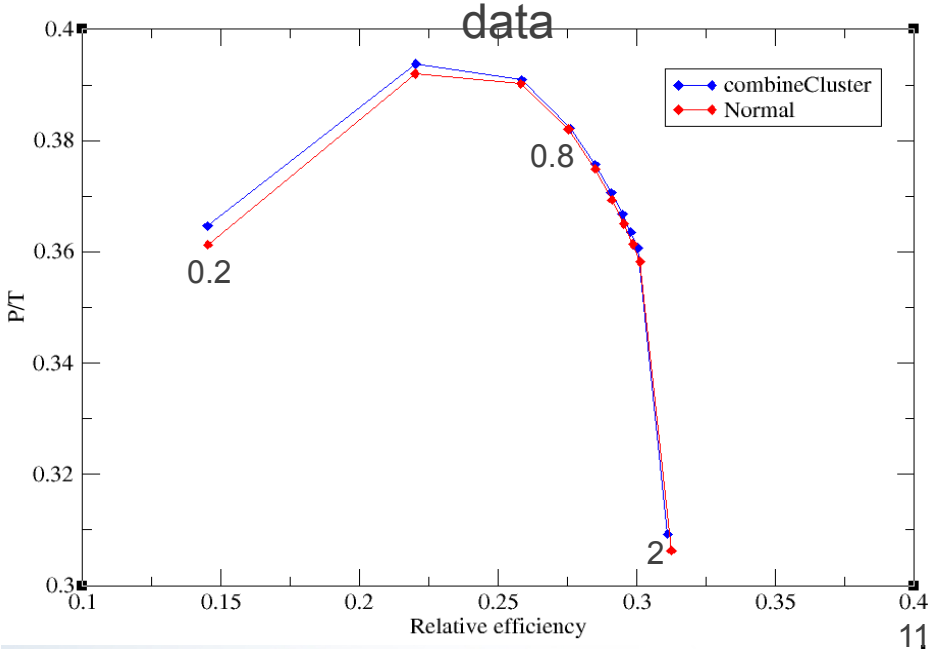
```
#                +- threshold FOM for combining
#                |   +- ndet maximum for combining
#                |   |  +- max distance for inclusion attempt (cm)
#                |   |  |
combineclusters 0.8   8  270
```

^{60}Co data AGATA @GANIL

Simulated data

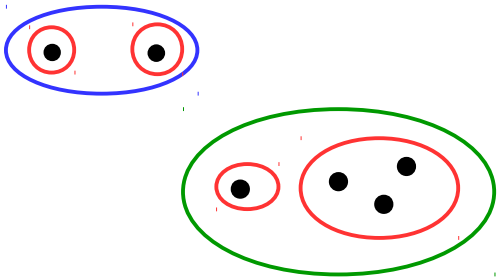


Experimental data



Other functions :

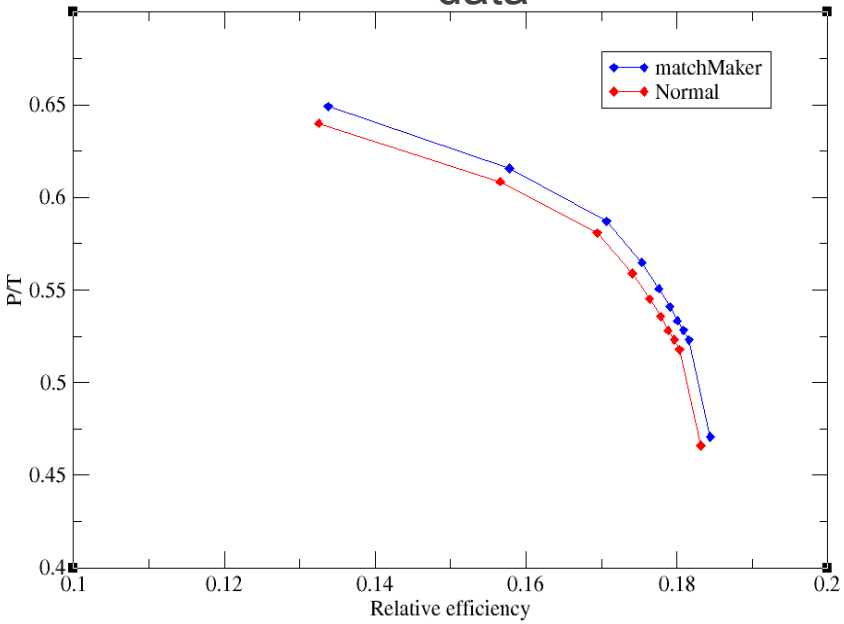
MatchMaker : Merge two single interactions
MatchMake++ : Merge a single interaction with another cluster that had a bad FOM



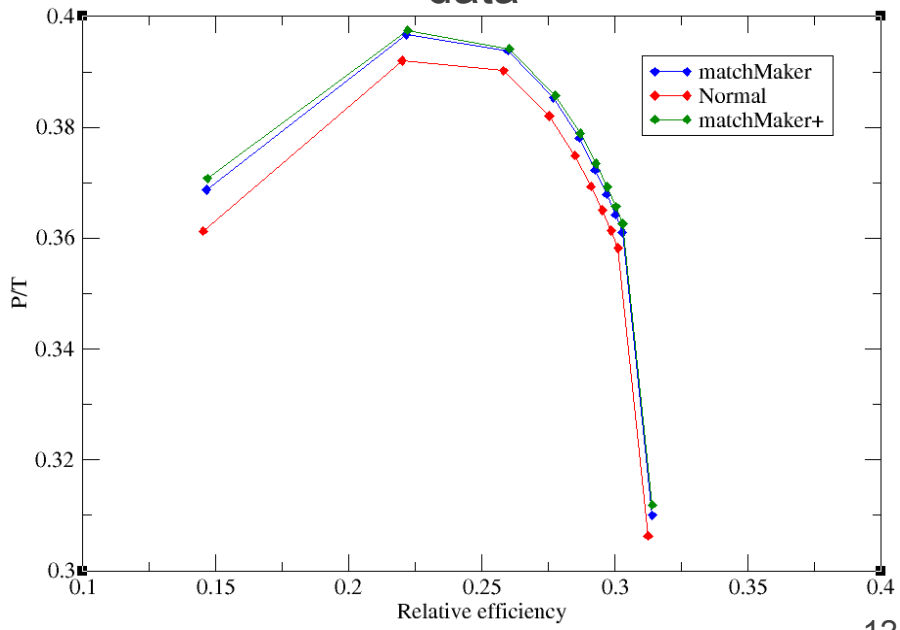
```
#           +- 'kickout' FOM value
#           |    +- max distance for inclusion attempt (cm)
#           |    |
matchmaker 0.5  150
```

^{60}Co data AGATA @GANIL

Simulated data



Experimental data



Conclusions and future

- Not many changes to the GT tracking code since last
- But tried the new efirst and elast functions
- elast does not seem to work. Maybe the same problem as with backtracking which also does not work so well? Is the assumption wrong or is it bad data?
- The efirst function appears to work. It effectively **trivializes** the tracking at high energy; *but so be it*. The function and the functional form of the penalty needs to be optimized and we need to make sure it does not hurt us in other energy regions
- We still need to add tracking of pair production to the code using Machine learning?
- Try other FOM measures?