Tracking: Status and perspectives

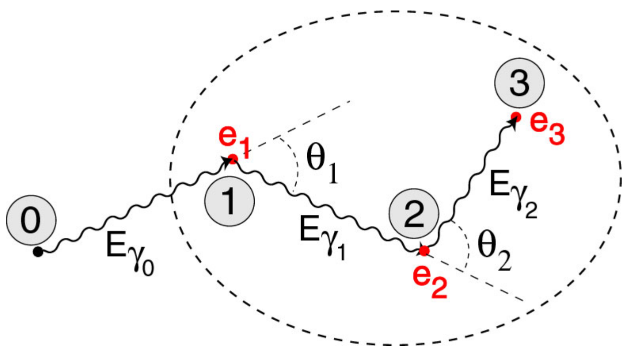
A. Lopez-Martens

OFT recent changes

- Maximum angle for clusterization can be scaled down

$$\alpha_{max} = \cos^{-1}\left(1 - \frac{2}{\left((n_{int}+2)/3\right)^{0.9}}\right)$$

- 1st vertex evaluation parameter ***a=1*** in AGATA package



$$L = \prod_{n=1}^{N-1} P_n \exp^{-a\left(\frac{E_{\gamma n} - E_{\gamma n,pos}}{\sigma_E}\right)^2}$$

- Single interaction procedure

minprobsing limit on product of range and cross section for photoelectric effect -> hardcoded energy-dependent probability to find the interaction point at a given distance inside AGATA

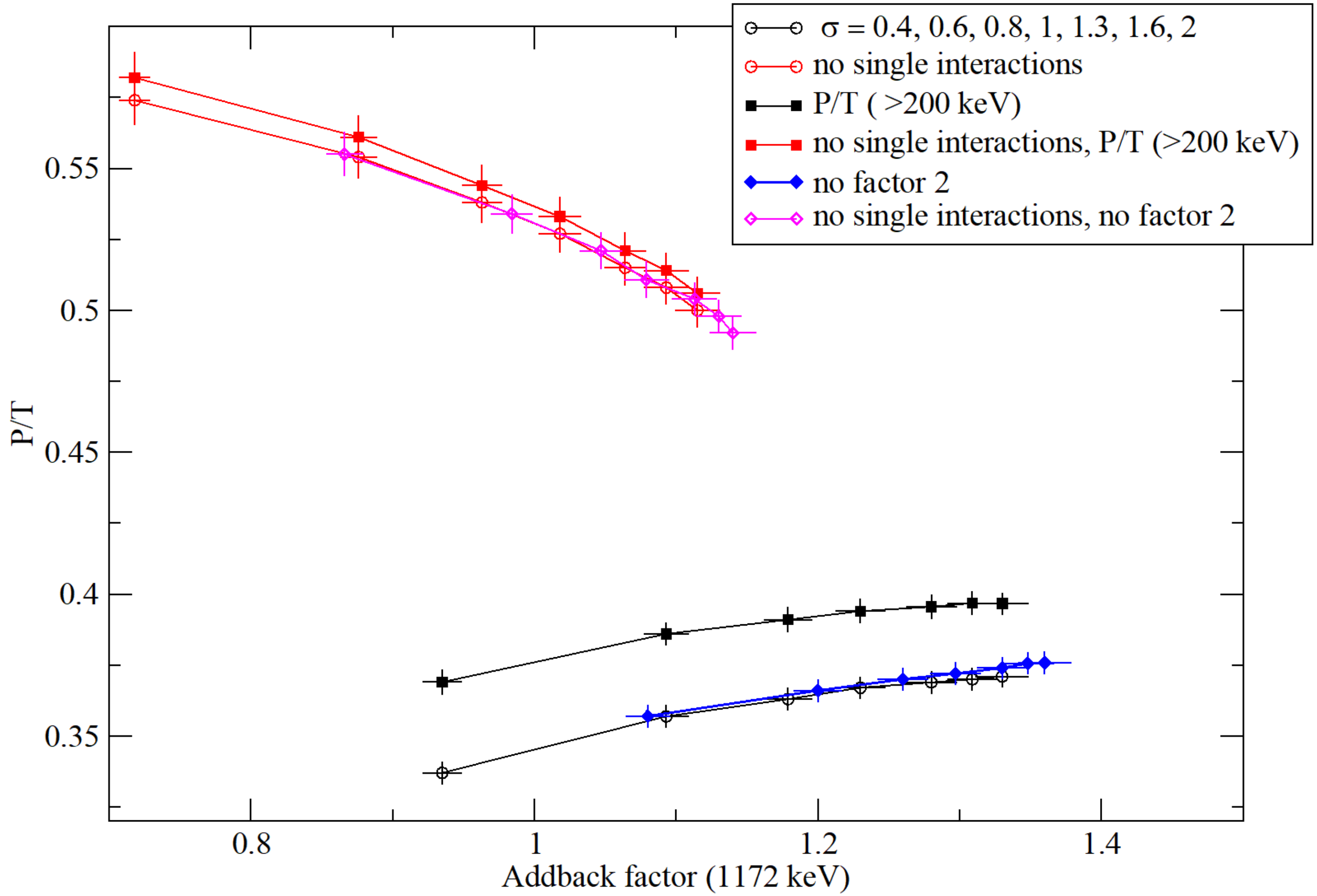
OFT recent changes

- Some bug fixes and improvements (thanks to Jeremy !):

Some events with wrong crystal ID in the Ge sphere correction to compute effective distance in Ge (in online/replay software, not in the AGATA package)

Possibility to have different packets for tracking based on the time of the interaction points

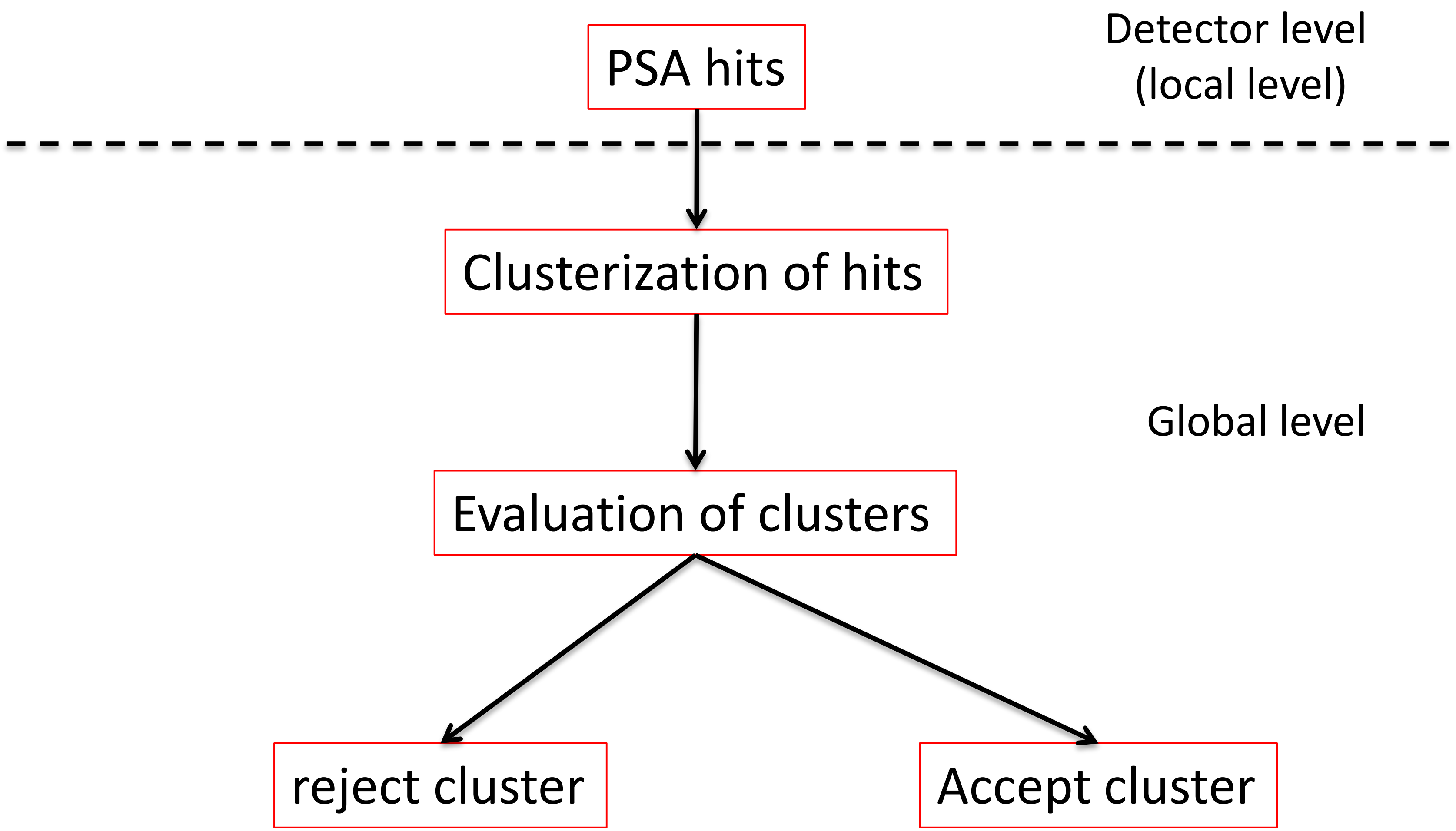
Results for 1172 keV from ^{60}Co (2016)



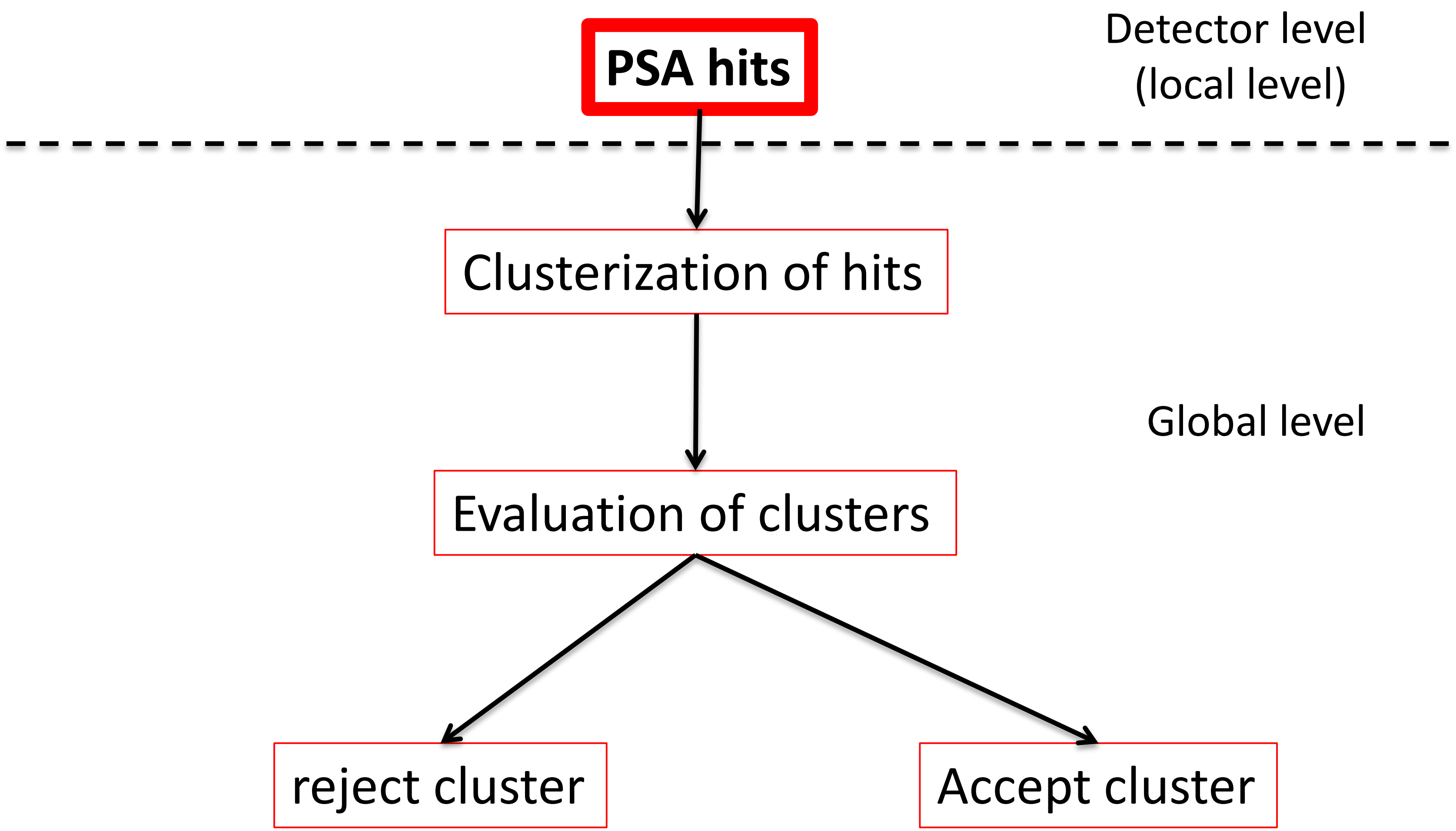
P/T (σ_θ =0.8) goes from 34 to 36% with new single interaction procedure

Addback factor (σ_θ =0.8) goes from 1.18(2) to 1.26(2) removing the factor 2

What can be improved ?



What can be improved ?

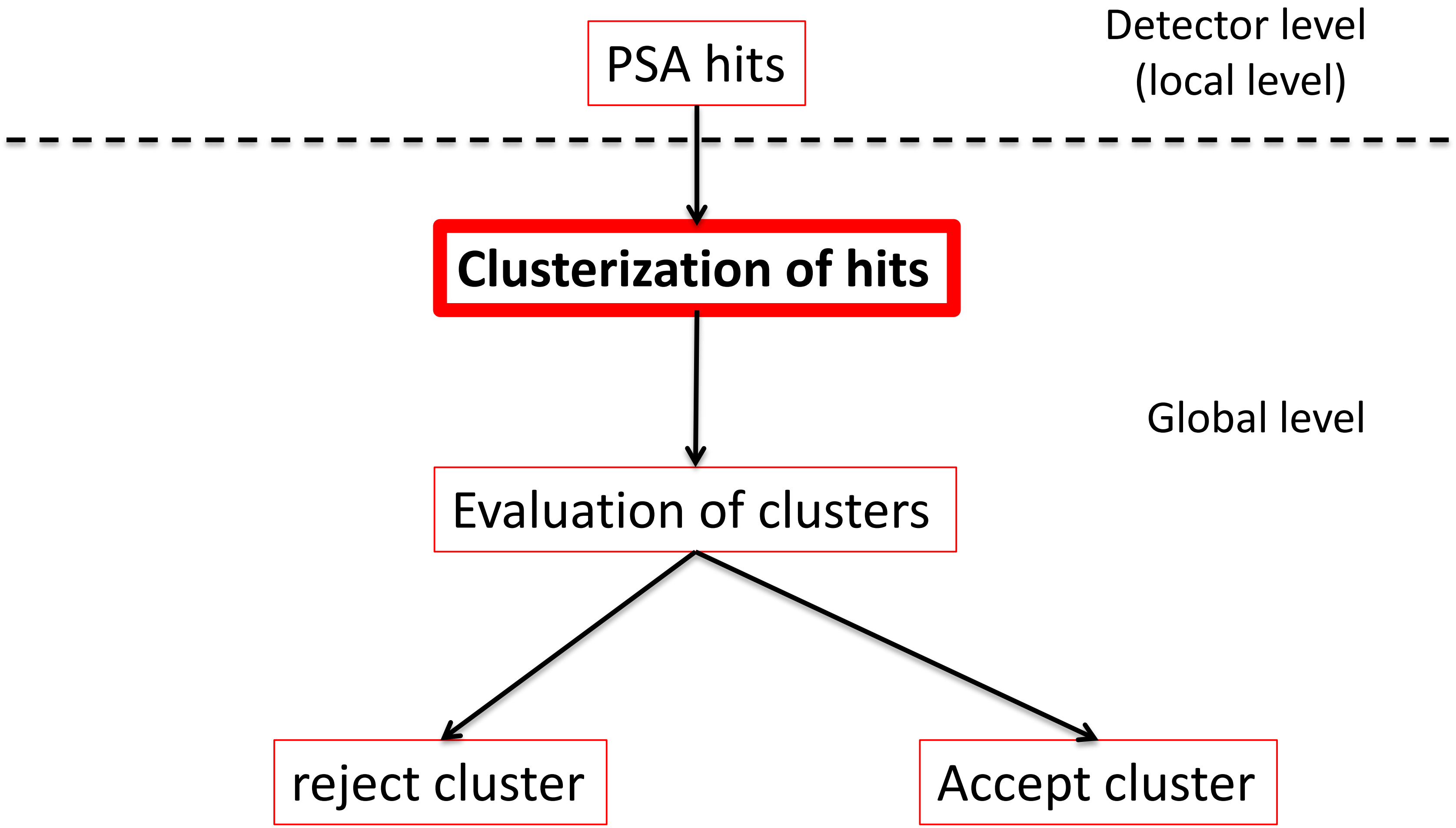


PSA hits

- Improve the fidelity of the signal data base
- Provide uncertainties of the hit positions/energies
- Provide multiple-hit/segment solution as well as single-hit/segment

- Feedback loop from tracking to aid/guide PSA ?

What can be improved ?



Clusterization

- Geometric method (mgt, OFT, Gretina): **link clustering**

  Free parameter: maximum opening angle α_{max}

  (may vary with the number of interactions)

Clusterization

- Geometric method (mgt, OFT, Gretina): **link clustering**

  Free parameter: maximum opening angle α_{max}

  (may vary with the number of interactions)

- Minimization of an objective function via Fuzzy loggic : ***Fuzzy C-Means*** and ***Hyperconic clustering*** **(**or ***Dancing Cones*****)**

$$J(f) = \sum_{j \in N, k \in K} f_{k,j}^{m} d_{k,j}^{2}$$

j - labels the N interaction points

k - labels the K clusters

d - geometrical distance between point j and cluster k

$f_{k,j}$ degree of membership of point j to cluster k

m - fuzzyfier

Free parameter: D_{max} (to get an estimate of the number of clusters to be found)

Clusterization

- Minimization of the Free Energy of the system: ***Deterministic annealing fil*****ter**

$$F = D - TS$$

$$F = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n} p(c_j / v_i).d(v_i, c_j) + \frac{T}{N}\sum_{i=1}^{N}\sum_{j=1}^{n} p(c_j / v_i).\ln\left[p(c_j / v_i)\right]$$

$p(c_j/\nu_i)$ – probability that point *i* belongs to cluster *j*

$d(\nu_i,c_j)$ Euclidean distance of point *i* to cluster *j*

Free parameters: rate of change of temperature ***T*** and $\boldsymbol{T_{stop}}$

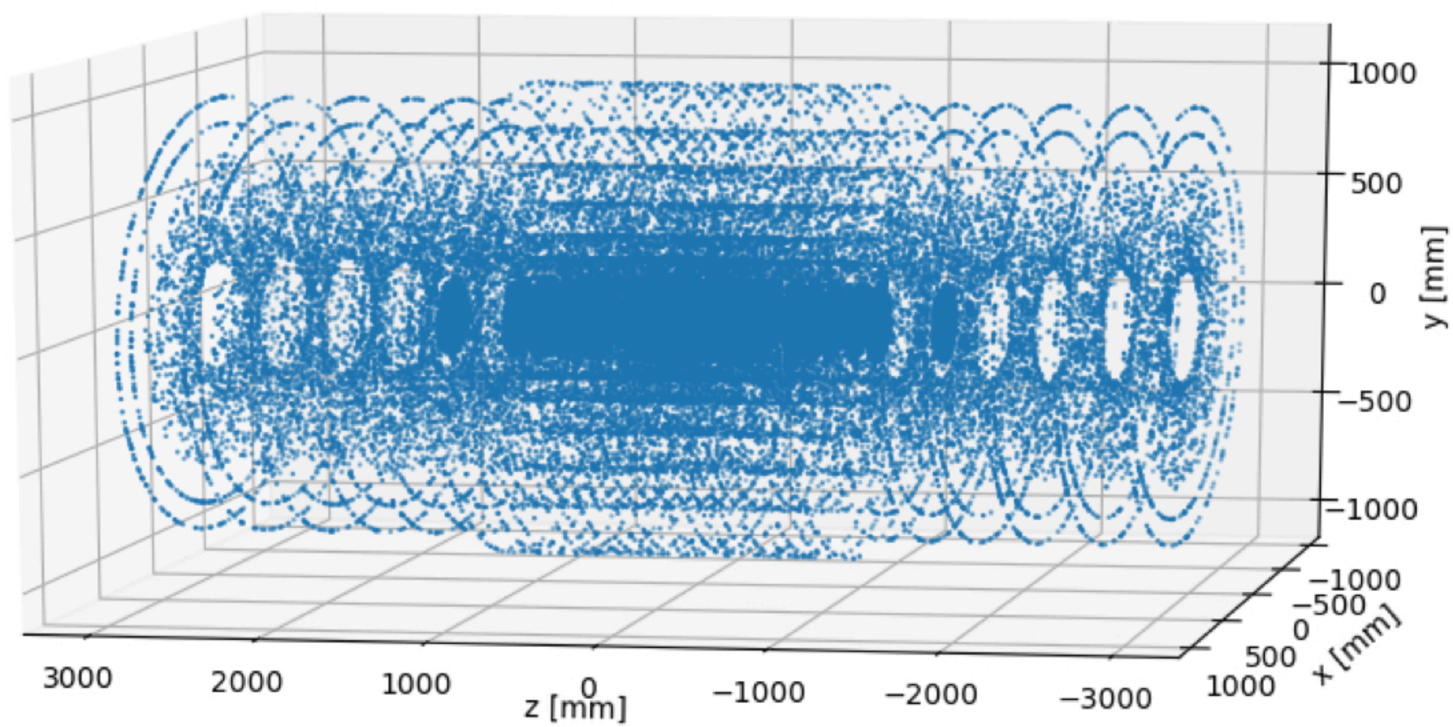
30 photons of 1 MeV into AGATA 4π (only reaction chamber present & no extra dead layers)

Algorithm	Efficiency (%)	Peak-to-Total (%)
MGT[9]	28	49
OFT[8]	24	55
Fuzzy C-Means	27	46
Deterministic Annealing[11]	26	48

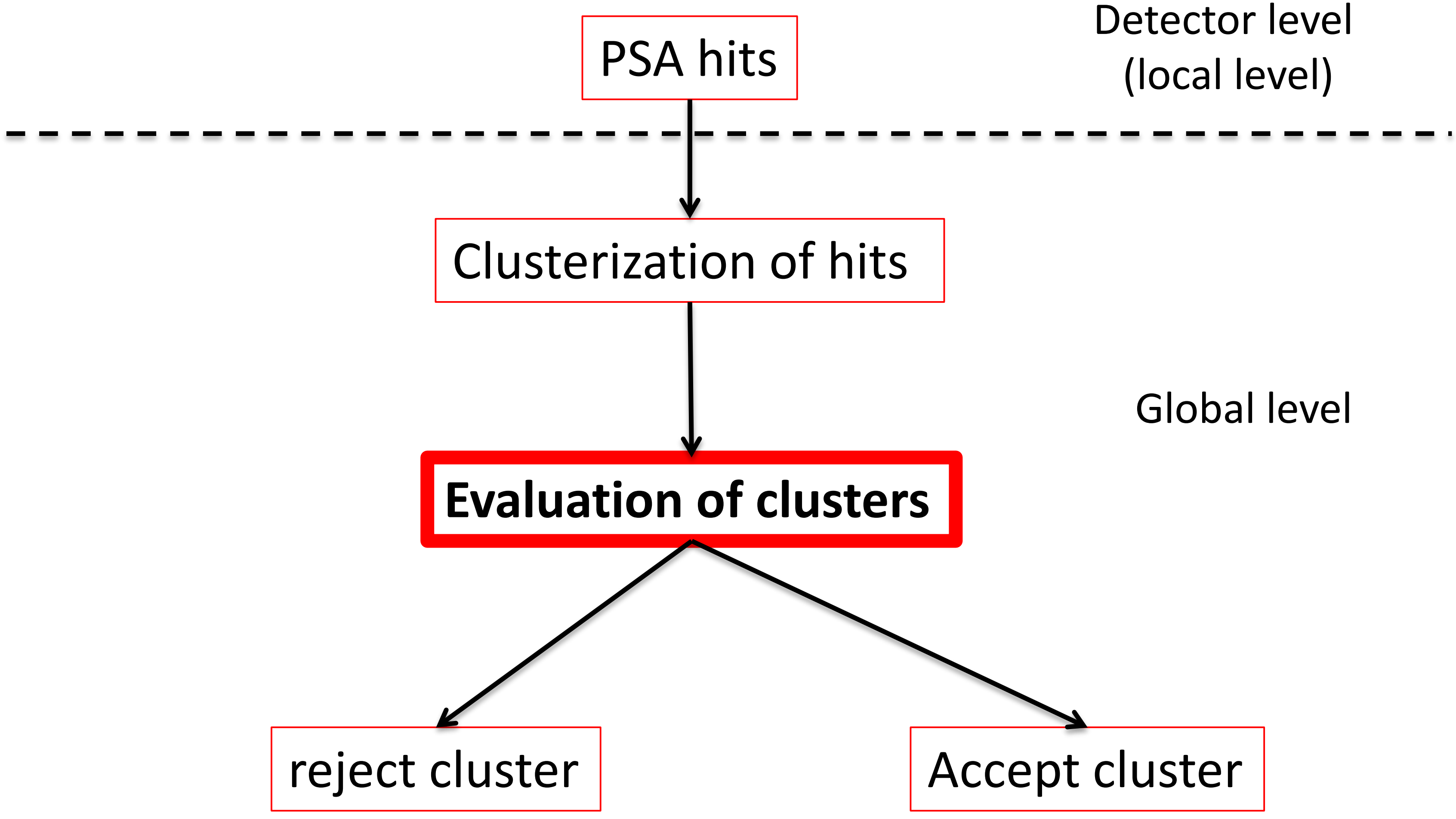
G. Suliman & D. Bucurescu (2010)

Clusterization

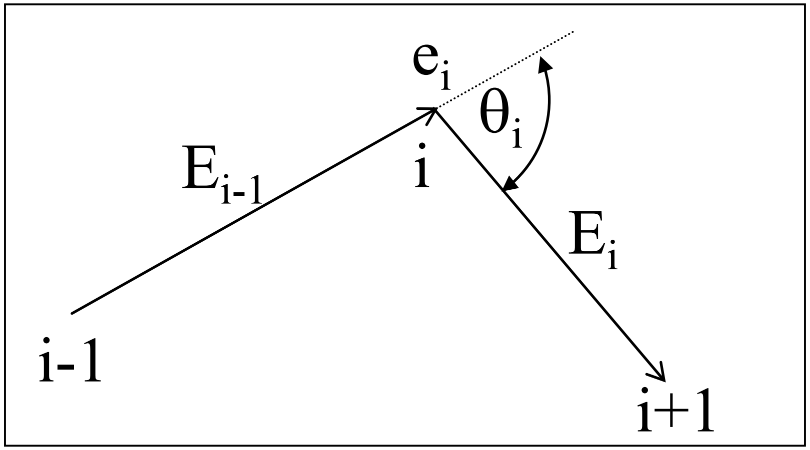
- Machine learning algorithms for cluster recognition on the basis of total cluster energy, number of interactions, spatial extent, ….?



What can be improved ?



Test of Compton vertex



$$E_i = \frac{E_{i-1}}{1 + \frac{E_{i-1}}{mc^2}\left(1 - \cos\theta_i\right)}$$

1) $V_i^E = E_i - E_i^P$

2) $V_i^e = e_i - e_i^P$

3) $V_i^{\cos\theta} = \cos\theta_i^E - \cos\theta_i$

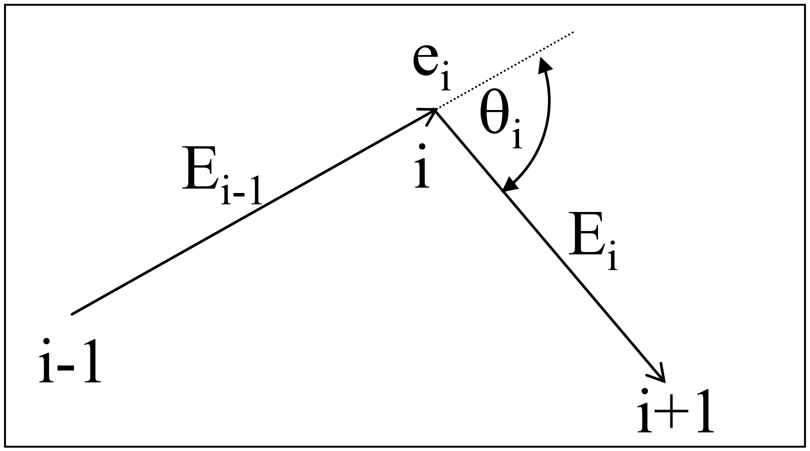
4) $V_i^\theta = \theta_i^E - \theta_i$

$$\chi^2 = \sum_{i=1}^{N_V} \left(\frac{V_i^x}{\sigma\left(V_i^x\right)} \right)^2$$

assuming normality and independence :

$$\sigma^2(V_i^x) = \sum \left(\frac{\partial V_i^x}{\partial w} \Delta w \right)^2$$

Test of Compton vertex



1) $V_i^E = E_i - E_i^P$

2) $V_i^e = e_i - e_i^P$

3) $V_i^{\cos\theta} = \cos\theta_i^E - \cos\theta_i$

4) $V_i^\theta = \theta_i^E - \theta_i$

mgt:

$$\chi^2 = \sum_{i=1}^{N_V} w_i \left(\frac{V_i^E}{E_{i-1}} \right)^2$$

Gretina:

$$FOM = \frac{1}{N_V} \sqrt{\sum_{i=1}^{N_V} \left(V_i^\theta \right)^2}$$

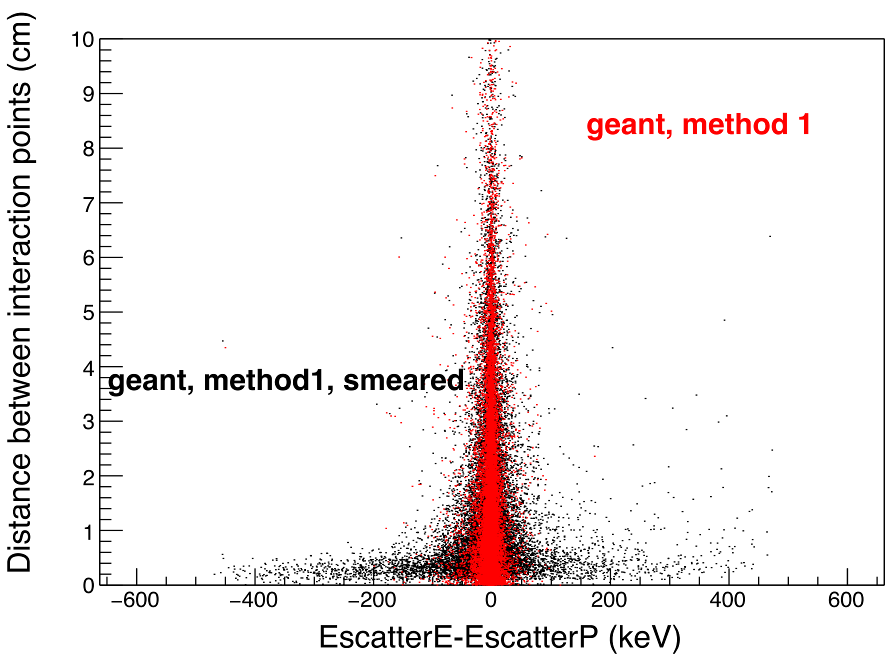
OFT:

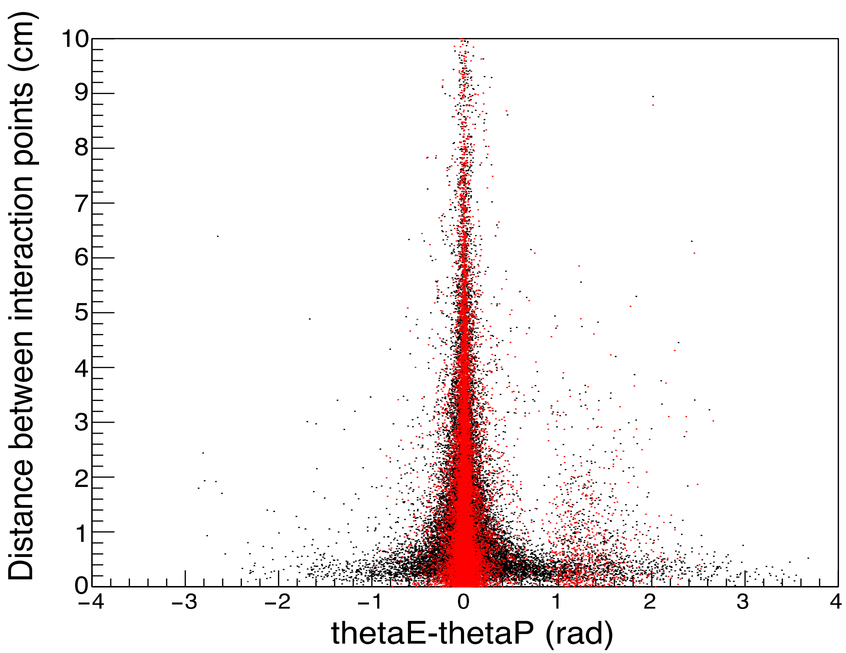
$$L = \prod_{i=1}^{N_V} P_i \exp^{-a\left(\frac{V_i^E}{\sigma_E}\right)^2}$$

What does the data look like ?

662 keV photon interacting in AGATA with 2 interactions (primary gamma interaction points) unphysical cosines are set to correspond to thetaE= 7 rad.

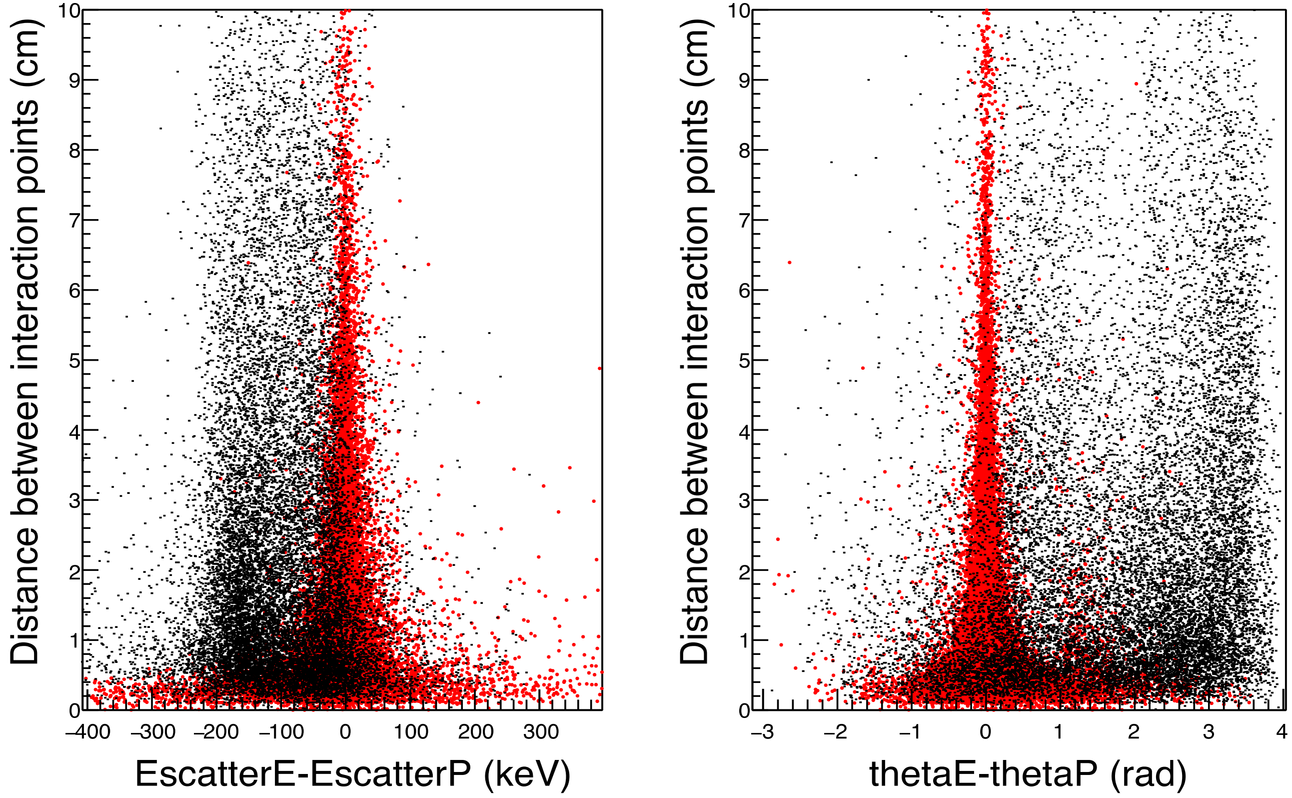
Total absorption events



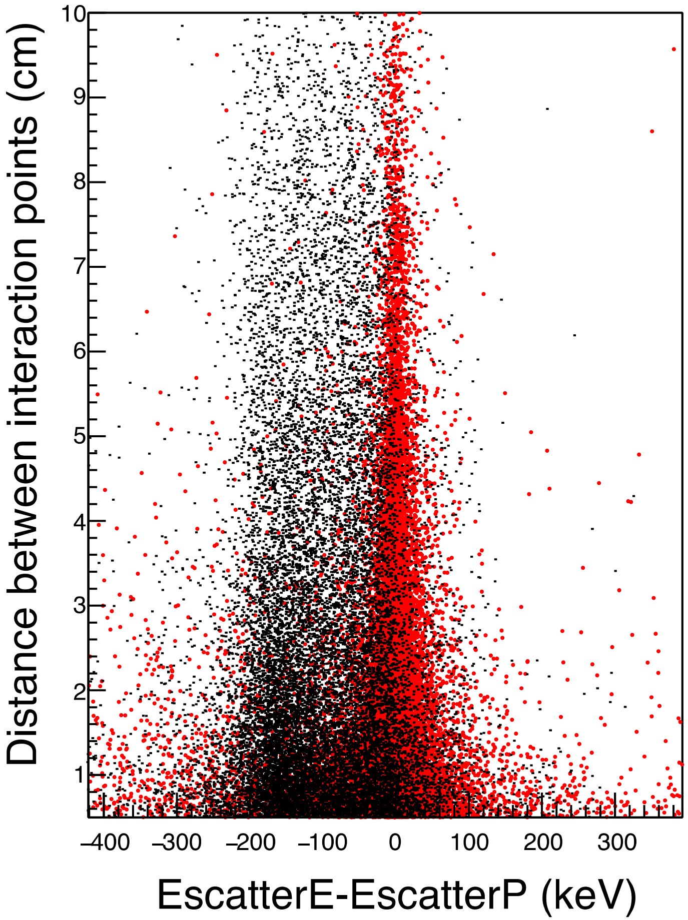


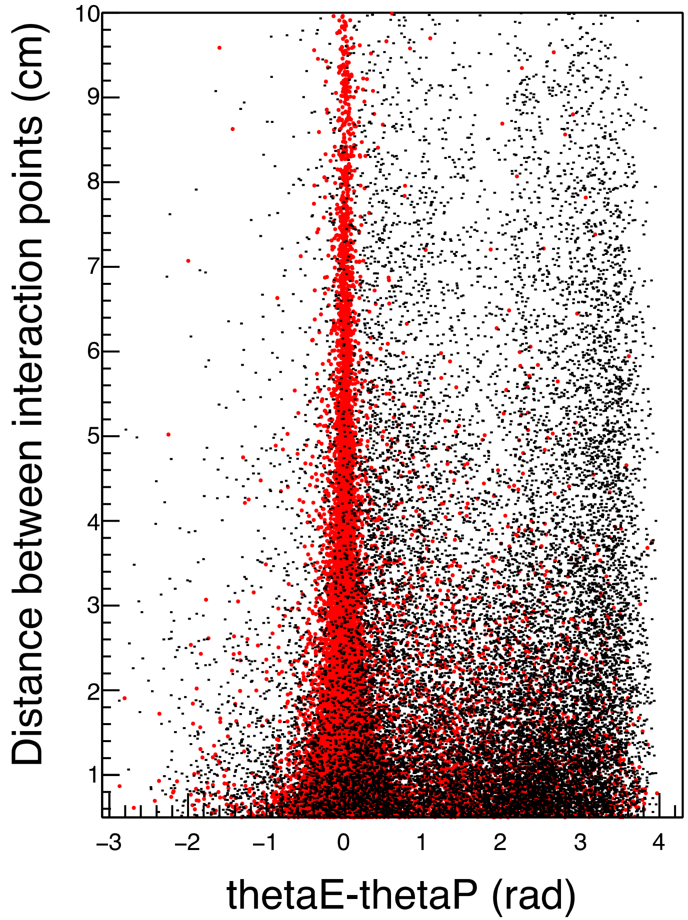
What does the data look like ?

Total absorption vs partial absorption (smeared), **when the 1st interaction is known**



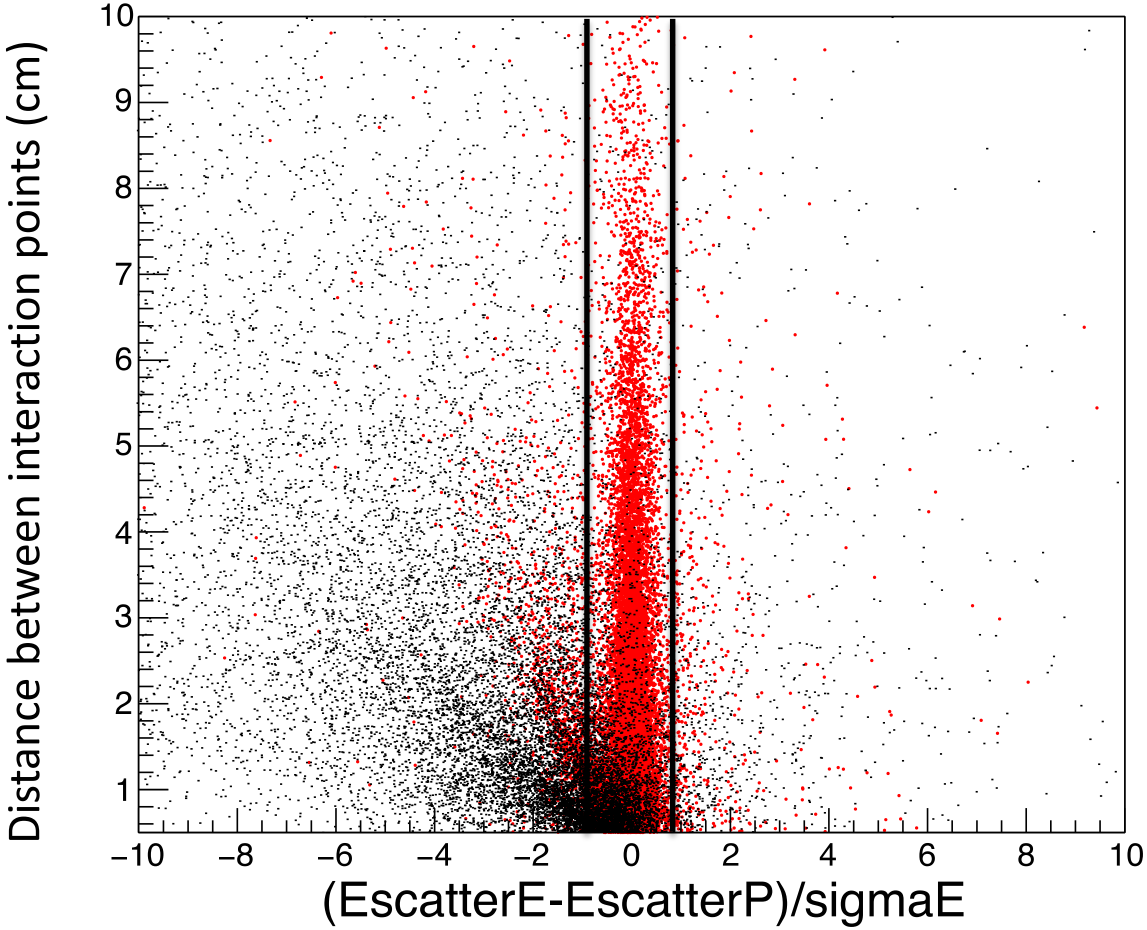
Adding packed 3-interaction events





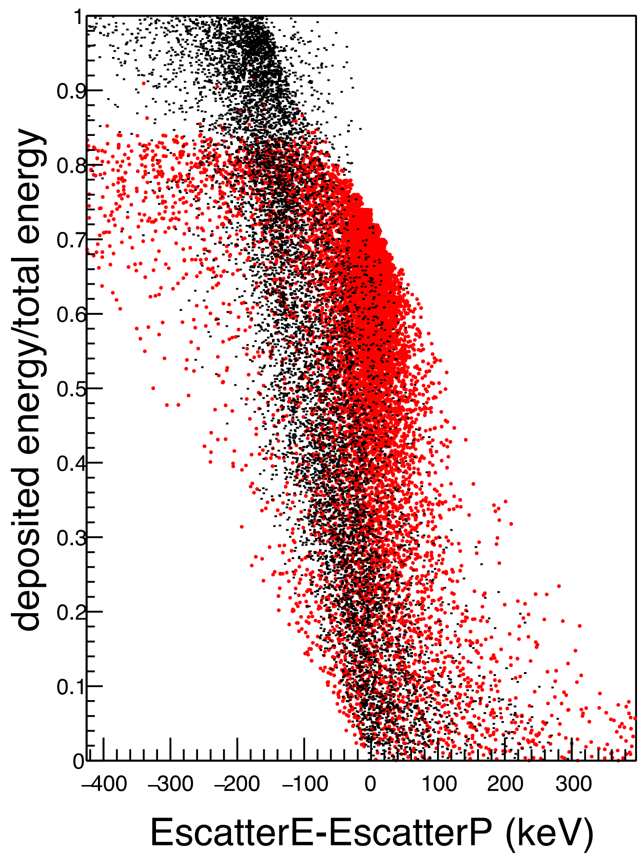
Correcting for position uncertainties

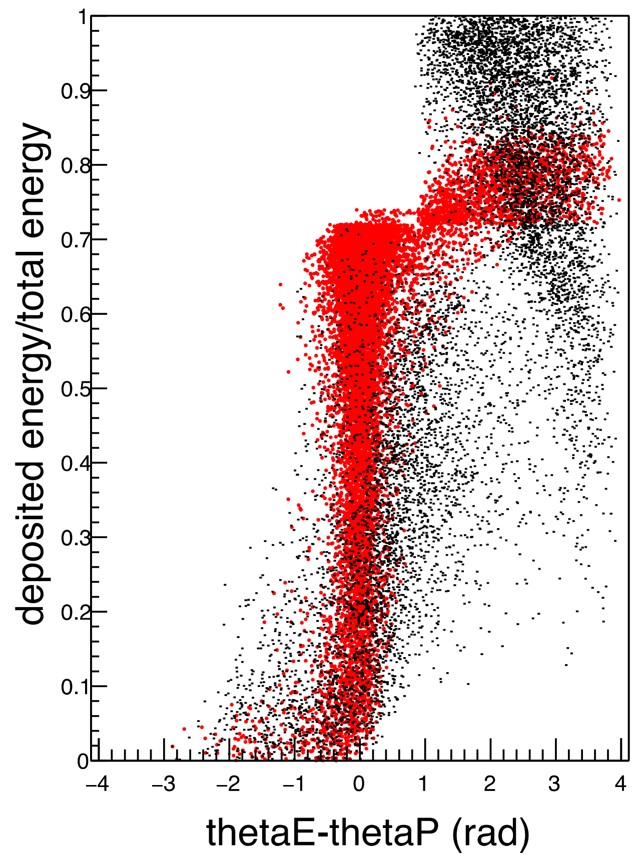
σ_θ=0.8 cm



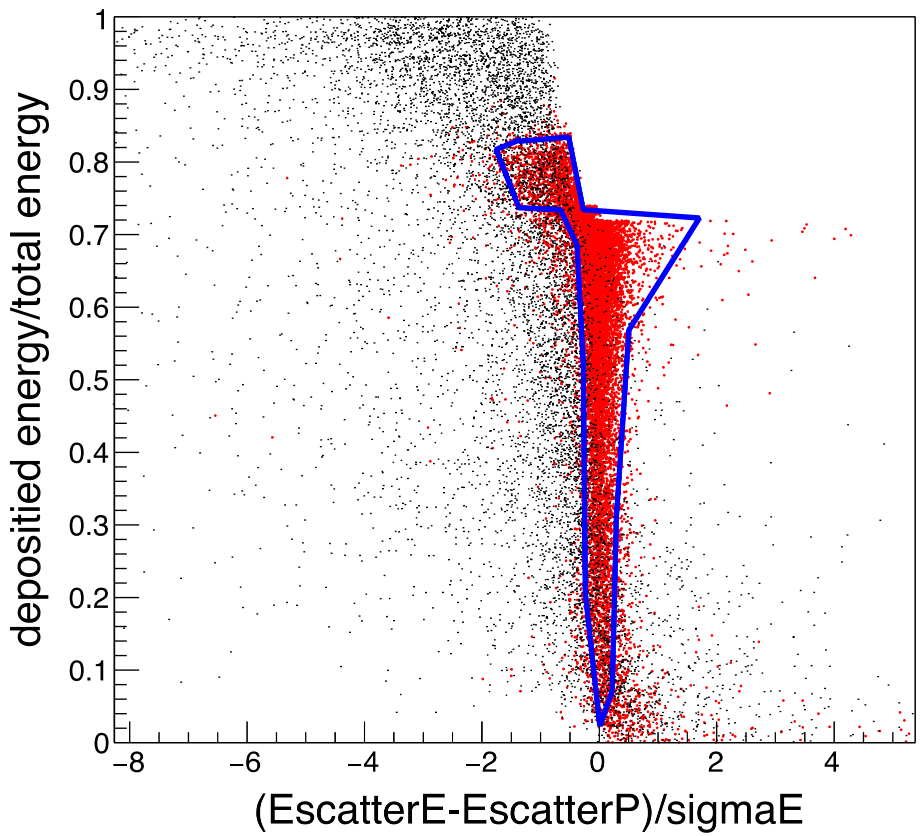
Using other observables

For distances <2 cm, where background & peak coexist





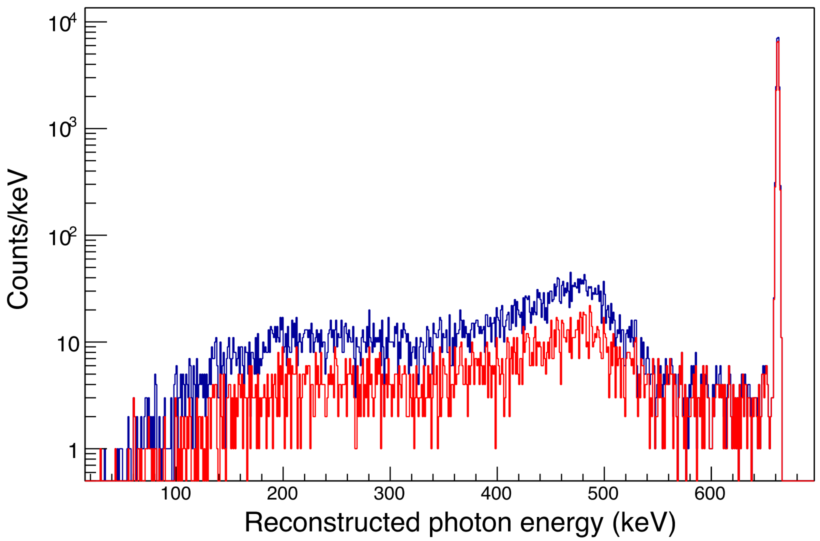
With uncertainty correction



Selection on (EscaterE-EscatterP)/sigmaE

vs

Same selection for r>2 and specific selection of events when r<2



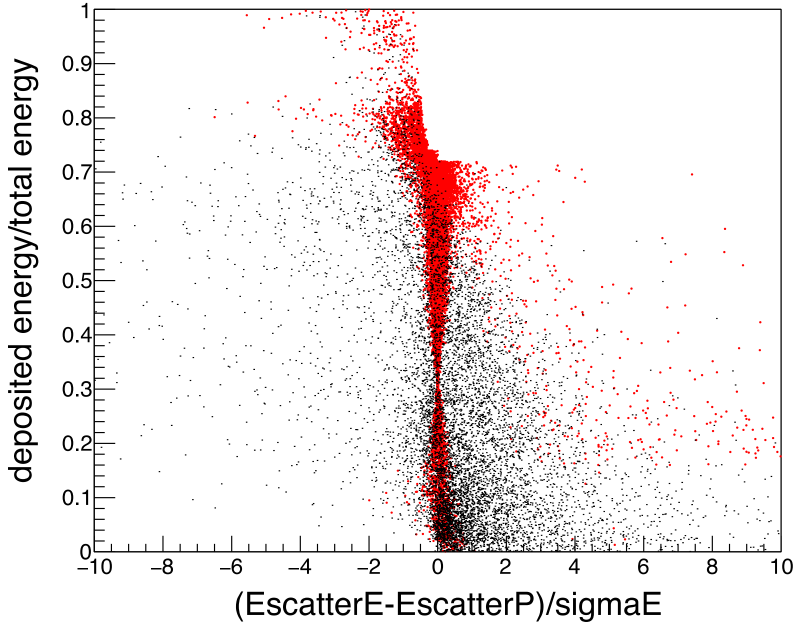
92% of the peak, with ½ the background

The cut can be improved !

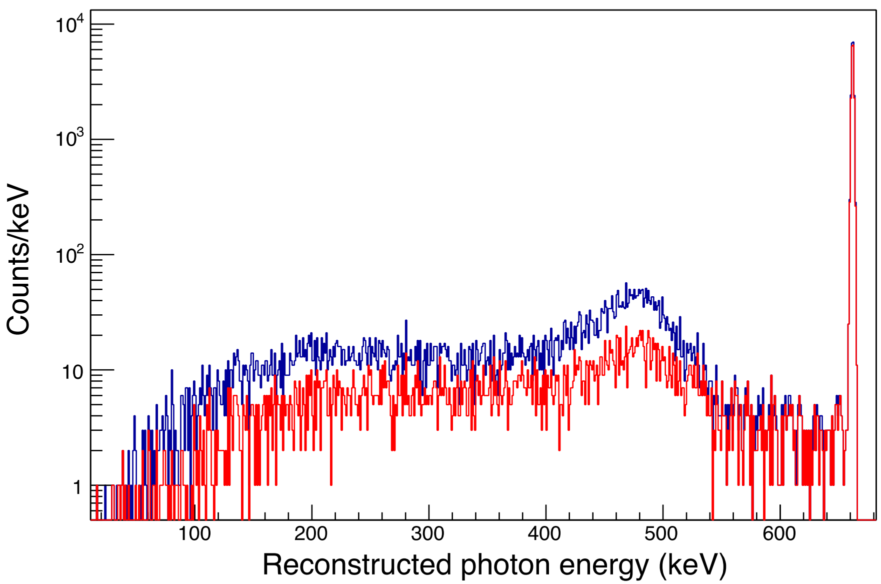
What happens when the 1st interaction point is not known….?

Determine the first interaction point as the one that minimizes the difference between EscatterE and EscatterP

- ➔ 15% total absorption events end up with the wrong 1st interaction
- ➔ 65% of partial absorption events end up with the wrong 1st interaction



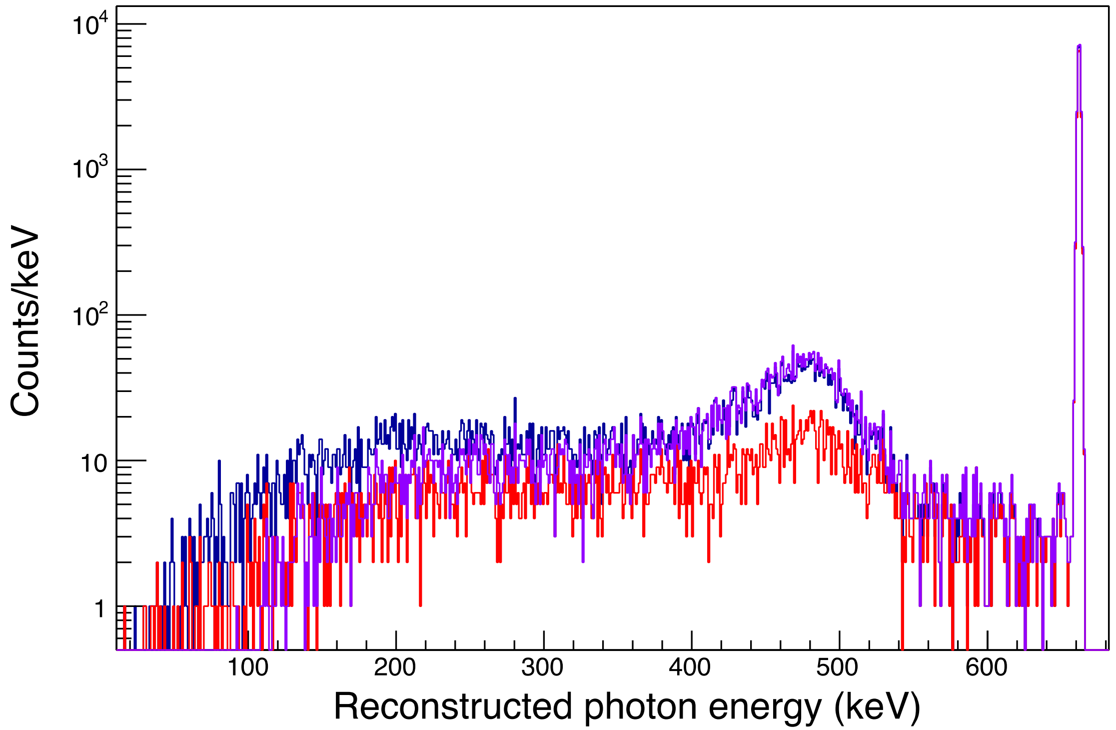
Using same selections as before



P/T is worse in both cases, but the previous differences remain

How does this compare to OFT ?

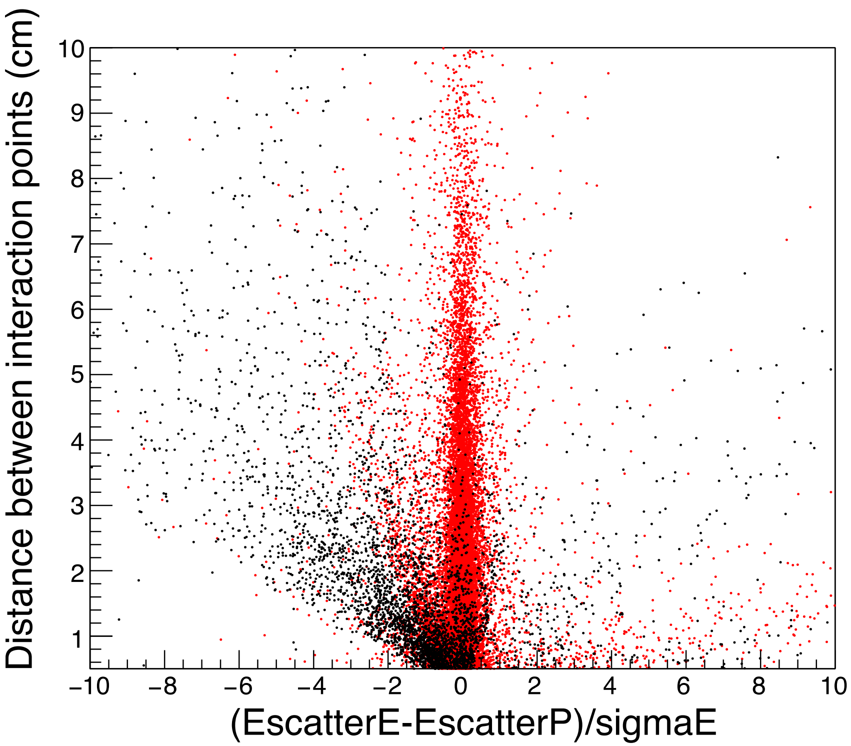
2-interaction-point events (from original 2s and 3s)

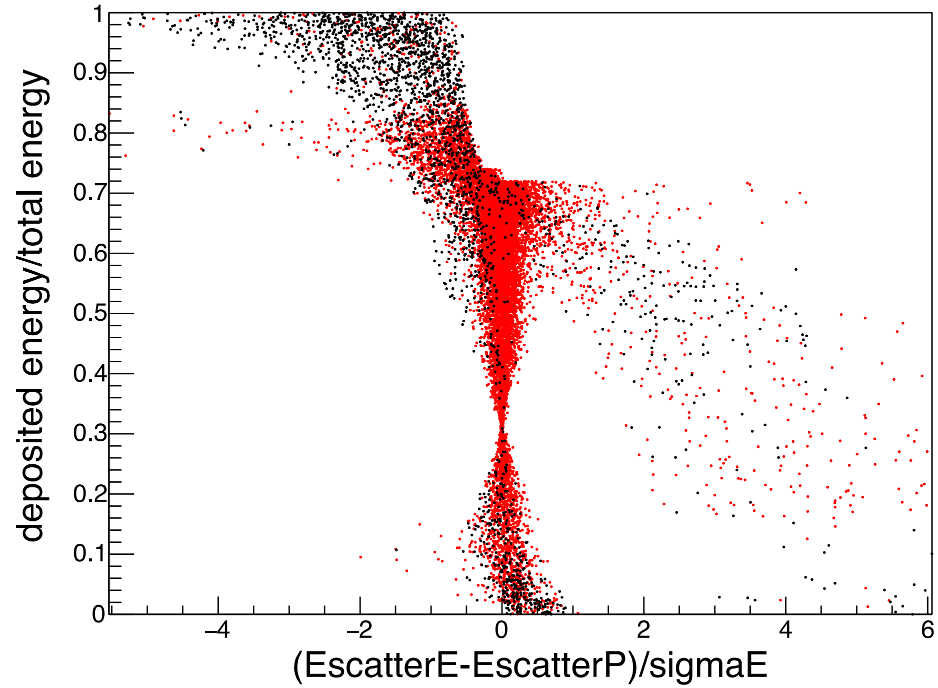


Spectrum	Eff.	P/T
Core	1	32%
Red	1.1	83%
Blue	1.2	70%
Tracked (σ_θ=0.8)	1.2	74%

How different are good and bad 662 keV events ?

662 kev total absorption vs 662 partial absorption (from 900 keV scattered events)





What to conclude ?

- Certainly room for « pattern » recognition in tracking
- How do patterns change with the number of interactions in the clusters and the total incident energy ?
- How do patterns change with real data ?
- What other observables can/should be used ?
- How to calculate a probability or likelihood in order to compare clusters, which contain some common interaction points ?
- Should clusterization & cluster validation be decorrelated ?
- …

NB: algorithm timing is not an issue– as final « good » tracking can be done offline