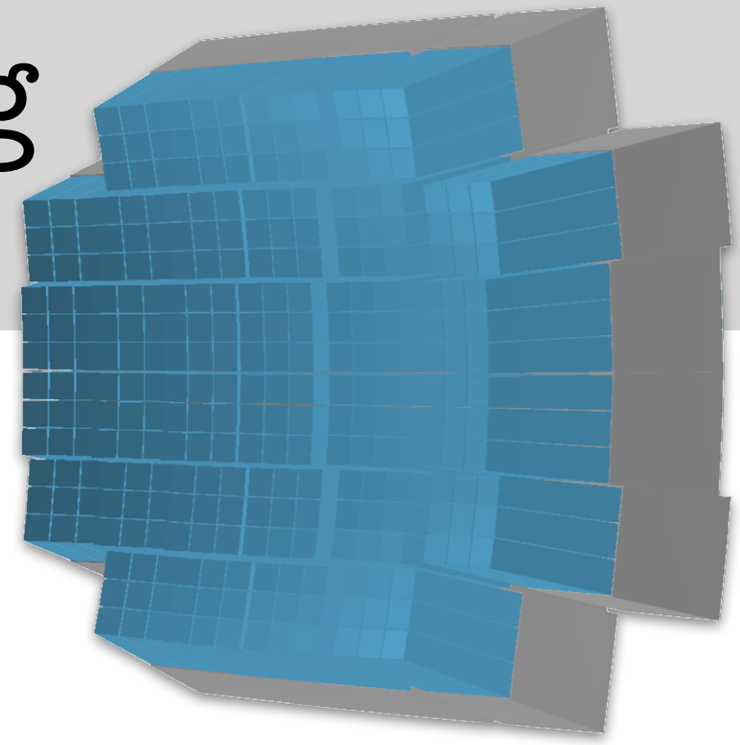
25/03/2020

Status of Calorimeter

Software Meeting

Lorenzo Scavarda



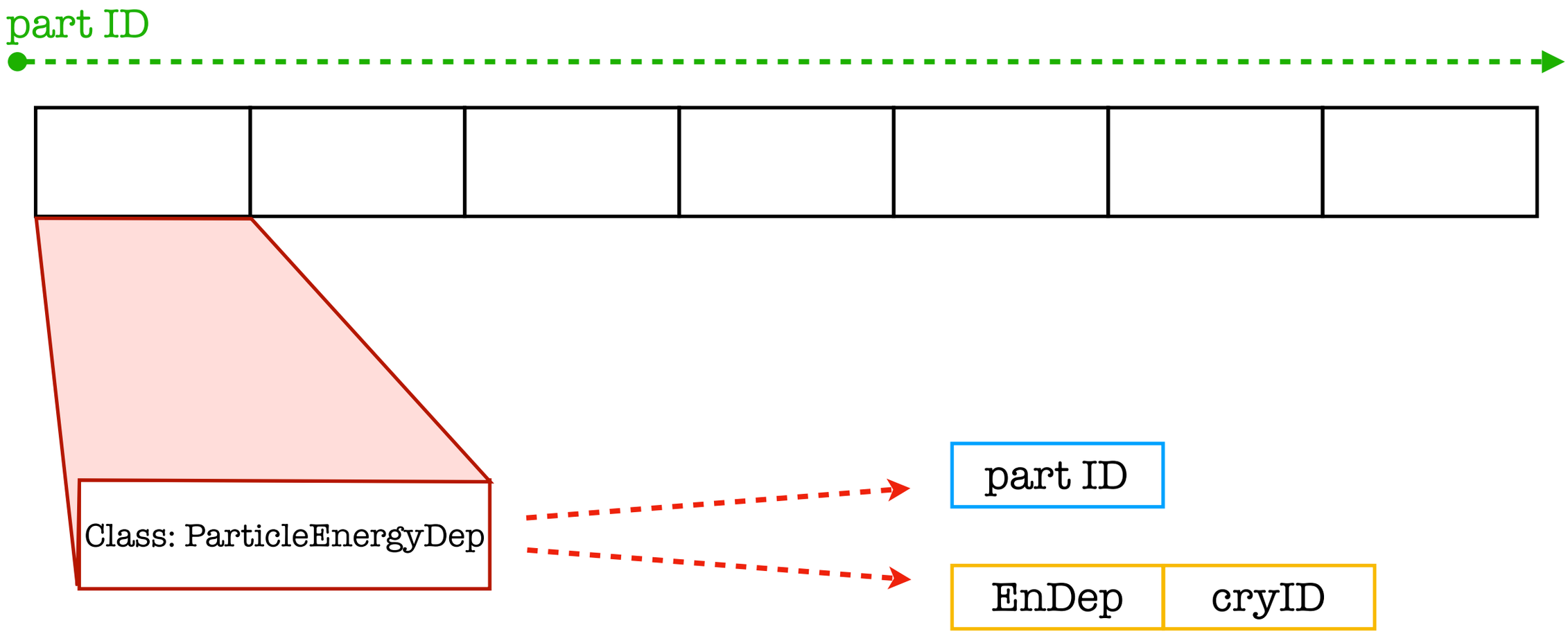
TACAactNtuMC (1)

Useful quantities:

- Total energy per crystal
- Total energy per particle
- Number of crystal hit from each particle

TACAactNtuMC (2)

TObjArr(m_eventStruct->TRn):



A TObjarr is filled with the **partID** and a pair of **EnDep** and **cry hit** of that particle

```
ParticleEnergyDep(int partId, int i) :
   partId(partId), iRelease(i) {}
int partId;                                  // particle ID
int iRelease;                                // index in EvnStr ---> i (the number of the energy deposition)
std::vector<EnergyDep> energyDeps;           // vector of pairs of: (cryID - endep)
```

TACAactNtuMC (3)

Example:

For each event:

for (int i=0; i<m_eventStruct->CALn; i++) loop on all energy releases in Calo

partID: 0	partID: 1	✗	part ID: 3	✗	✗	partID: 6

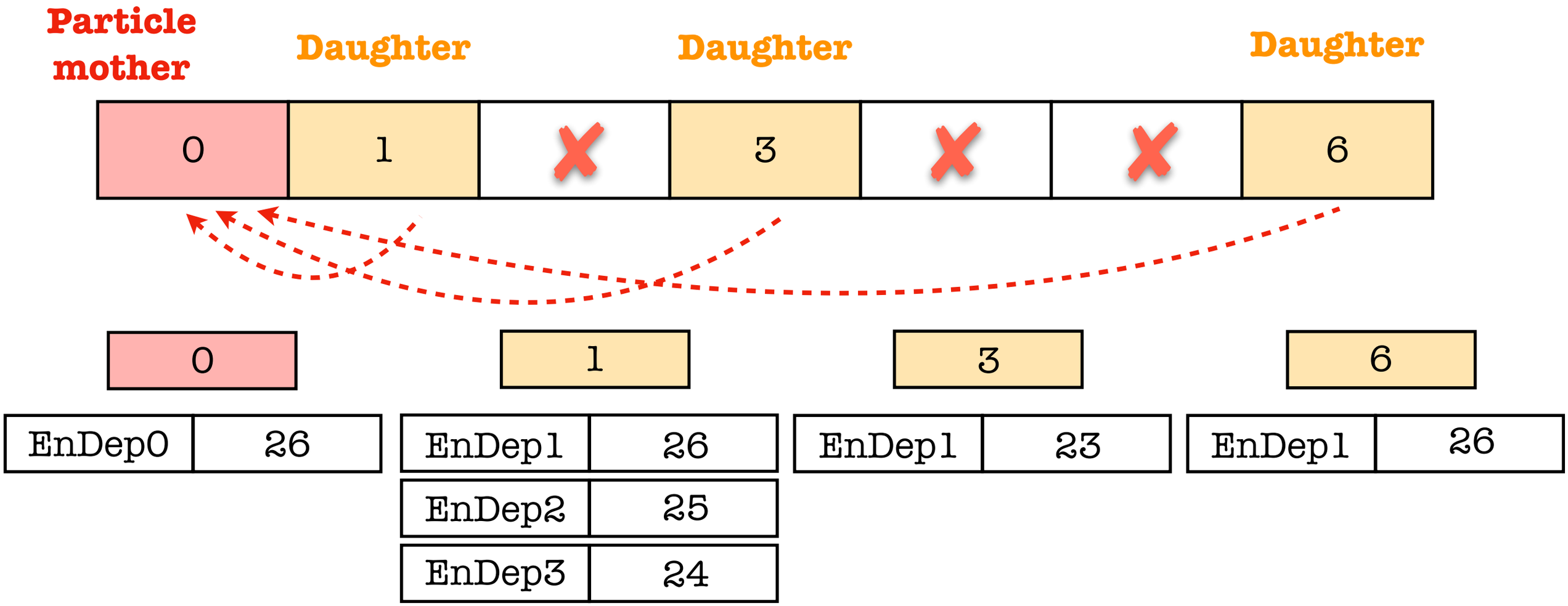
0		1		3		6	
EnDep	cryID	EnDep	cryID	EnDep	cryID	EnDep	cryID

Each element is labeled with the partID number

TACAactNtuMC (4)

Assuming that the TObjarr is sorted by creation and the particle mothers are before the daughters:

```
for ( int i=npart-1;  i>0;  - -i )
```



Remove daughter slots and compress (remove the empty slots) from the TObjarr

TACAactNtuMC (5)

In the TObjarr will remain only the particle mothers.
The mother slots contain the information about all their daughters.

Example:

0	8	14	16	17	24	28

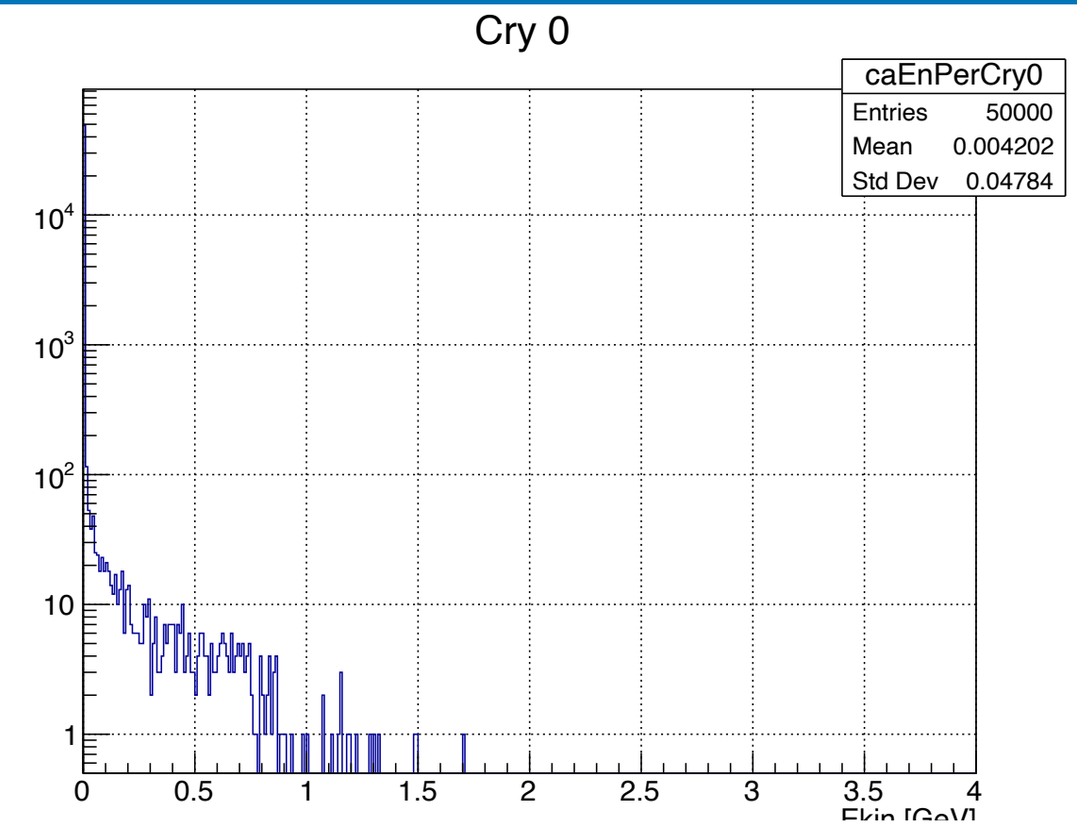
EnDep0_1	26
EnDep1_1	26
EnDep1_2	24
EnDep1_3	25
EnDep3_1	23
EnDep6_1	26

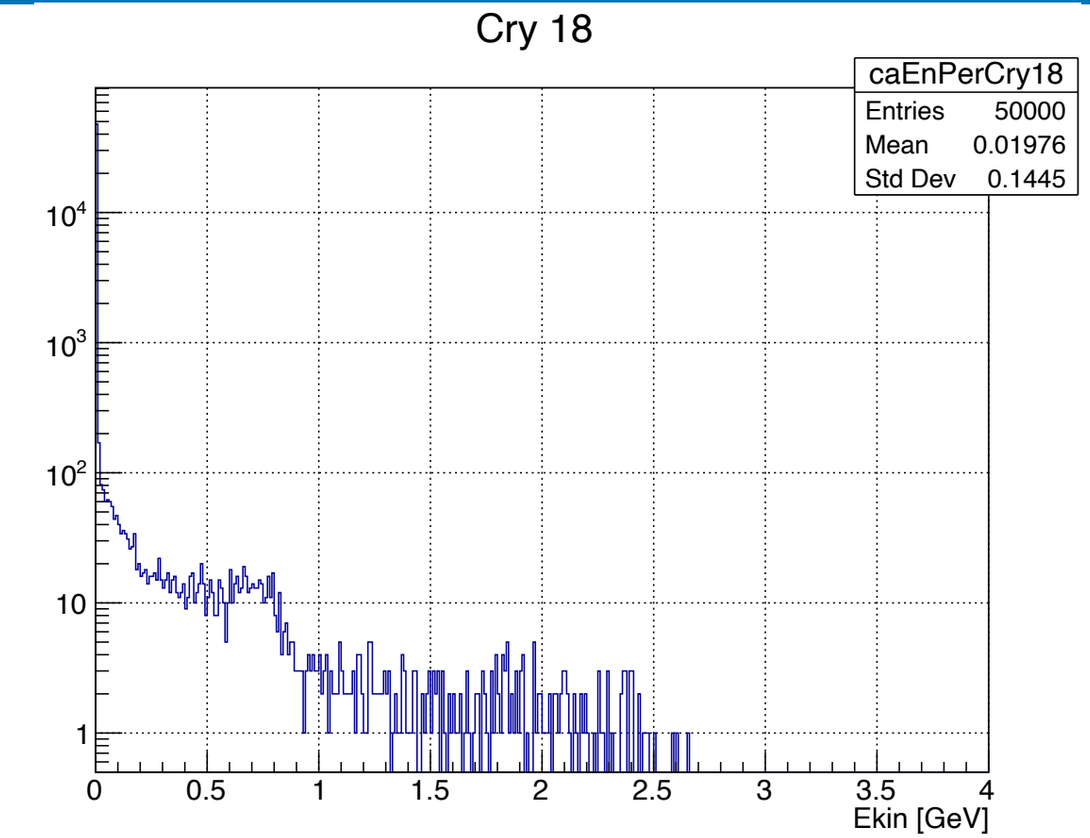
ParticleEnergyDep class

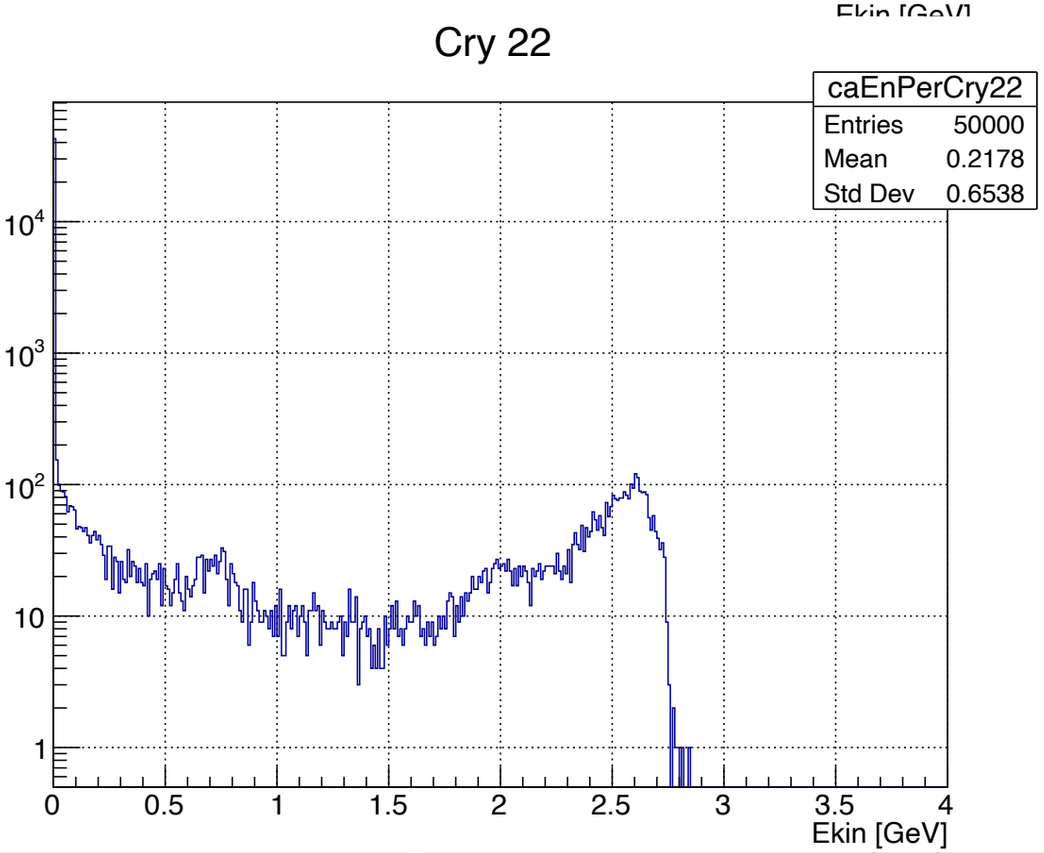
```
//Sum the energy blocks of one particle (trackID)
float getTotalEnergyDep() {
   float totalEnergyDep(0);
   for (std::vector<EnergyDep>::iterator it = this->enerc
      totalEnergyDep += (*it).energyDep;
   }
   return totalEnergyDep;
}

std::set<int> getUniqueCryIds() {
   std::set<int> uniqueCryIds;
   for (std::vector<EnergyDep>::const_iterator it = this
      uniqueCryIds.insert((*it).crystalId);
   }
   return uniqueCryIds;
}

float getNCrystals() {
   return this->getUniqueCryIds().size();
}
```

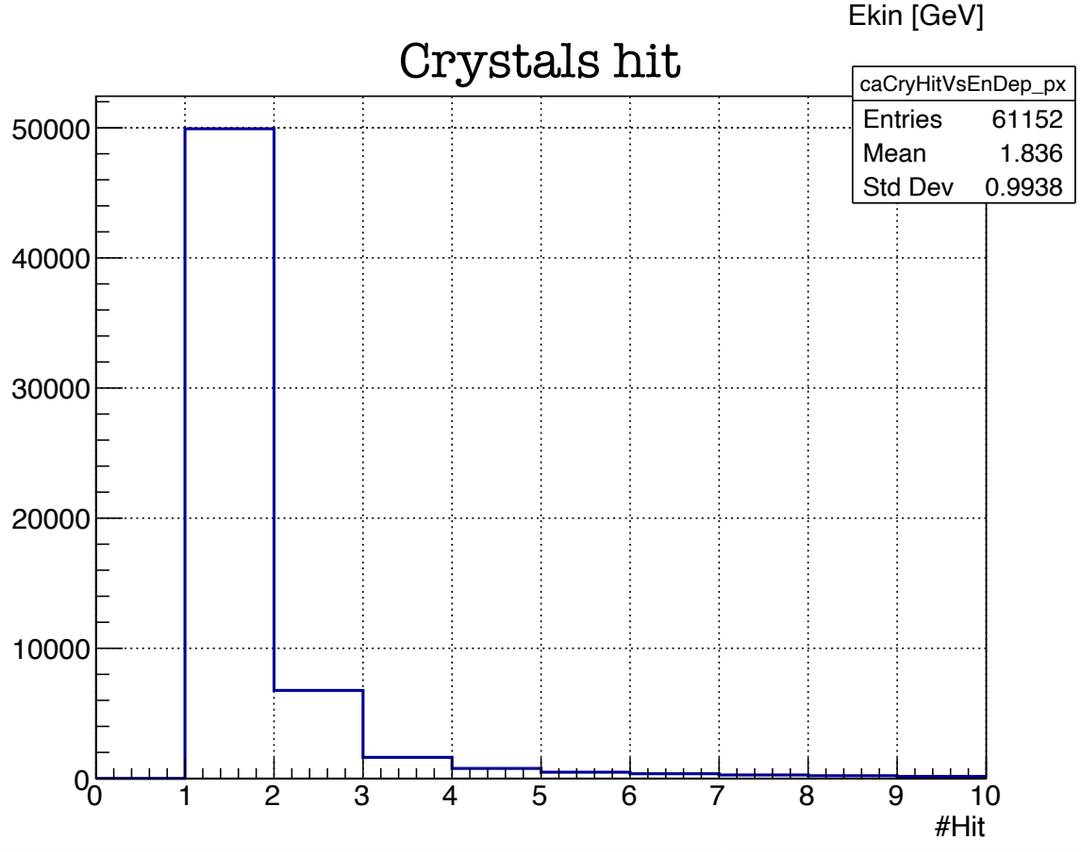
getTotalEnergyDep: EnDep0_1 + EnDep1_1 + EnDep1_2 + …
getUniqueCryIds: 26, 25, 24, 23
getNCrystals: 4

TACAactNtuMC Plots

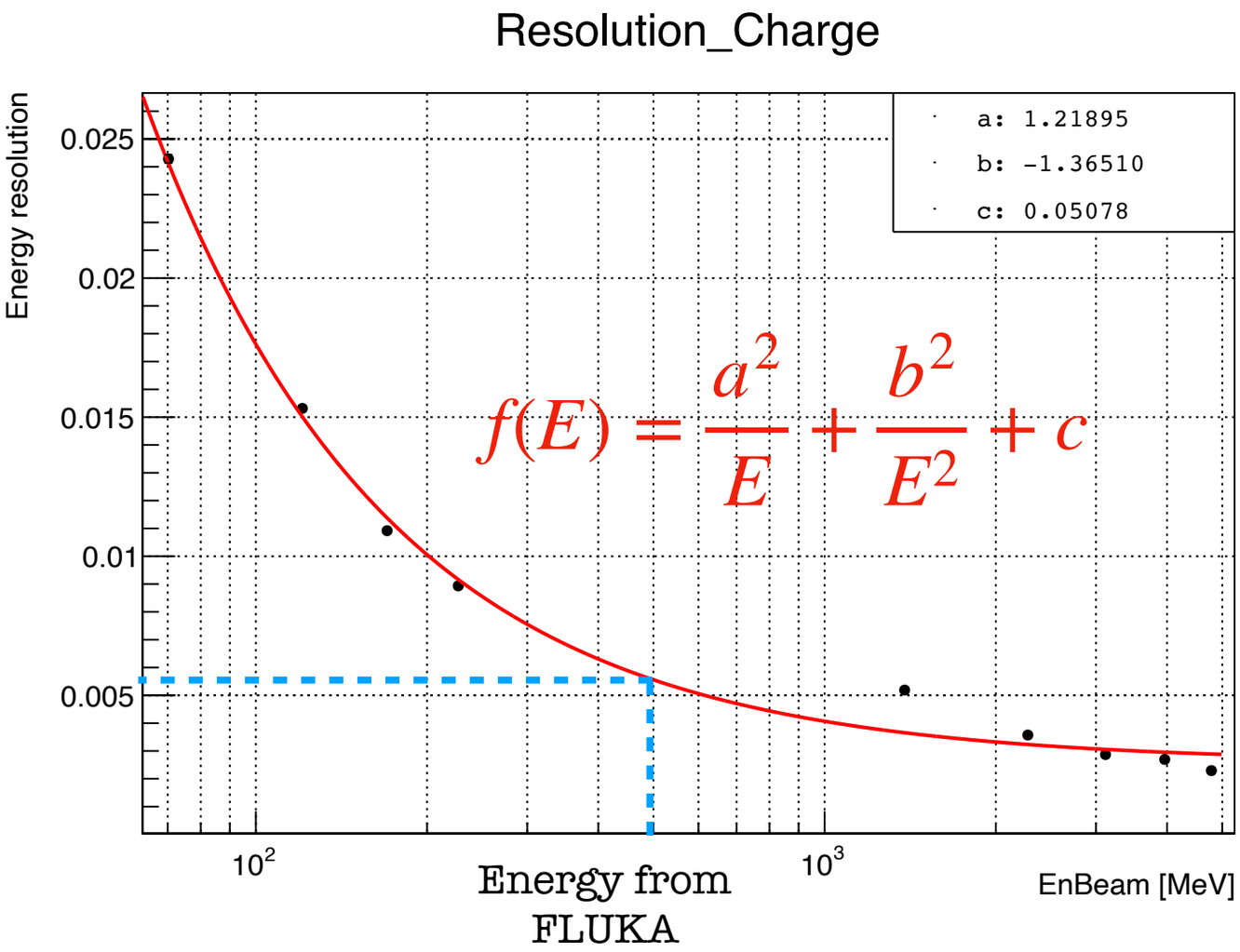








TACAdigitizer

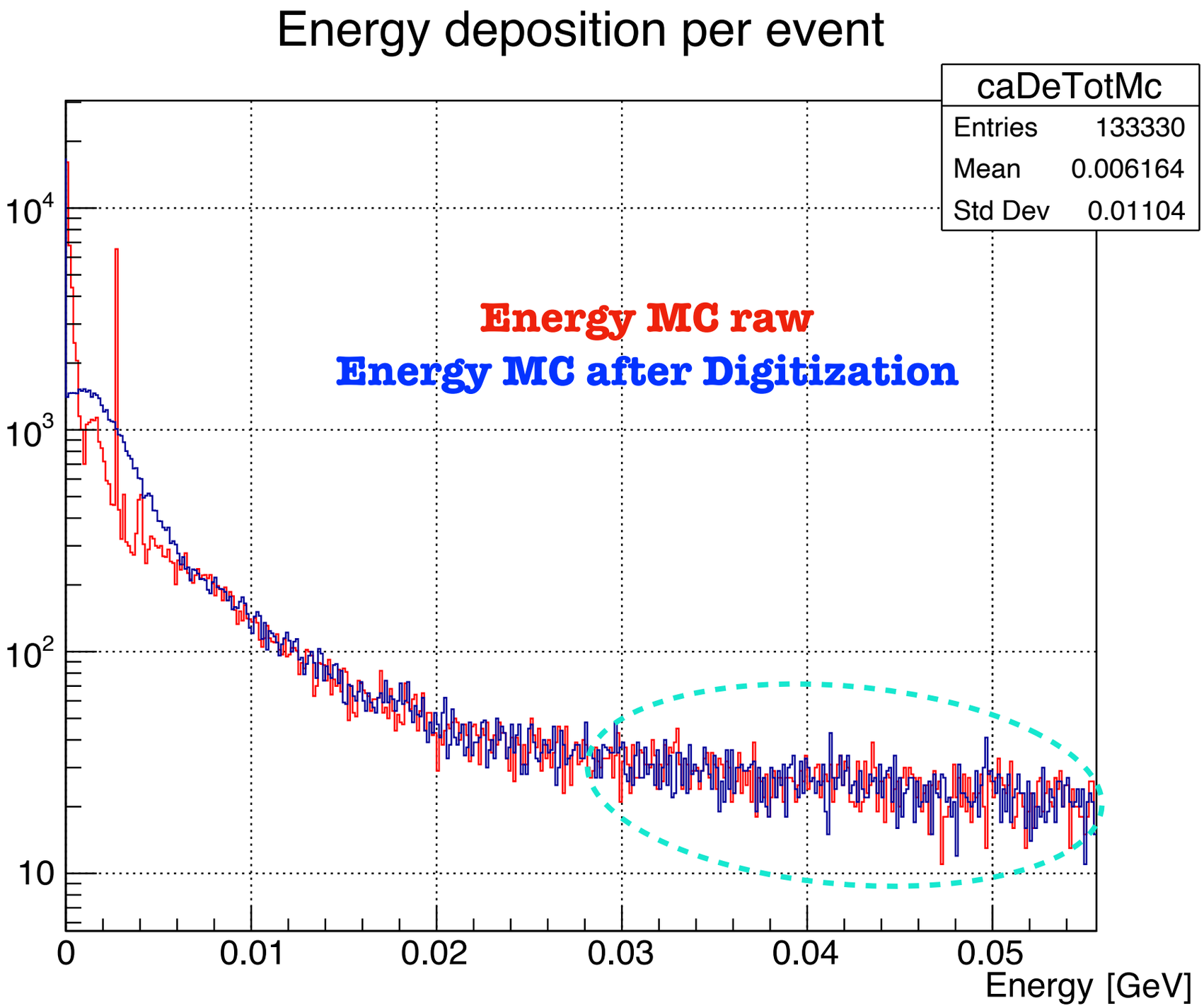


From 16_02_2020 CNAO test beam

```
//_____
Double_t TACAdigitizer::ResEnergy(Double_t* x, Double_t* par)
{
   Float_t energy = x[0];
   Float_t res = (par[0]*par[0]/energy + par[1]*par[1]/(energy*energy) + par[2]*par[2]);
   // Float_t res = (par[0]*par[0]/energy + par[1]*par[1]);

   // res /= 100;

   return res;
}
```

```
//apply smearing to the energy
edep *= 1000; //Convert MC energy from GeV to MeV

Float_t resEnergy = GetResEnergy(edep);
edep += gRandom->Gaus(0, resEnergy);

edep /= 1000;  //Convert energy from MeV to GeV

fCurrentHit = (TACAntuHit*)fpNtuRaw->NewHit(id, photonsN, edep, time);
```

TACAdigitizer Plot



To Do

TACAdigitzer class:

- Do a simulation with a simple setup as we used at CNAO, to compare MC simulation after digitization with some data
- Go on and improve the digitization