TRD Software status

CBM-TRD Retreat
March 29, 2019

Etienne Bechtel
University of Frankfurt

Overview

Structure of the software in cbmroot
Architecture of the code
Simulation modes

Features in the code
Digitization
Reconstruction

Results
Pulls
Physics performance

Open Issues
Needed features?
Communication and documentation

Architecture of simulation

Generation and transport

Generation is usually UrQMD

Transport is done with Geant 3 or Geant 4

Output is:

- MC Points in the detector
  - MC position
  - MC momentum
  - MC energy deposition

Digitization

Input is one MC point in the detector at a time

Simulation of the Detector response with the given energy deposition

Output is:

- CbmTrdDigis
  - Address
  - Time
  - Deposited charge in one channel
  - Trigger
  - Asic type
  - Error class

Reconstruction

Input is a timeslice at a time

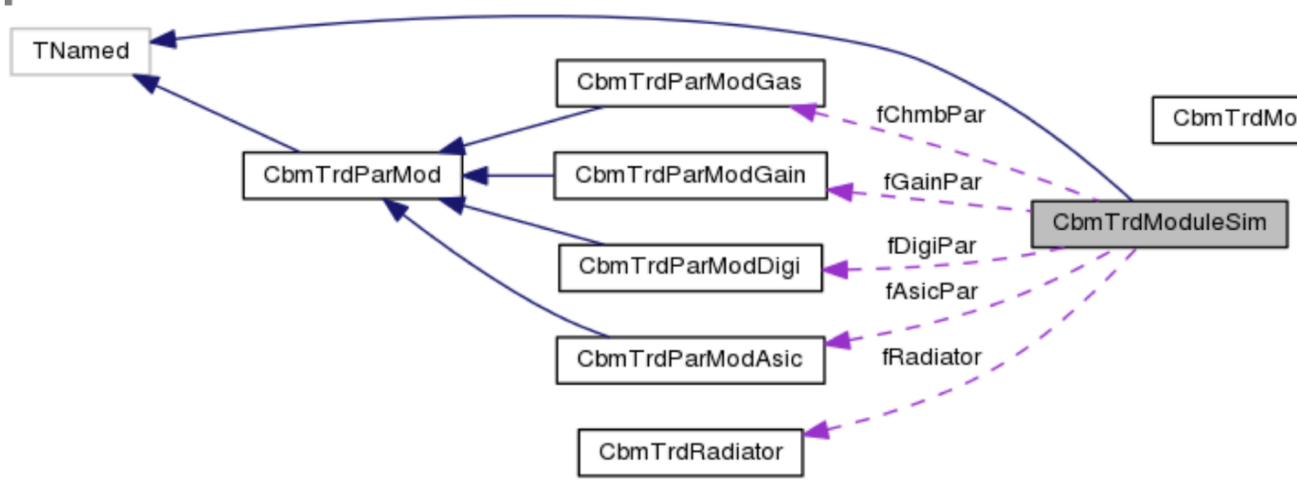
Each timeslice contains an amount of digit depending on the Interaction rate

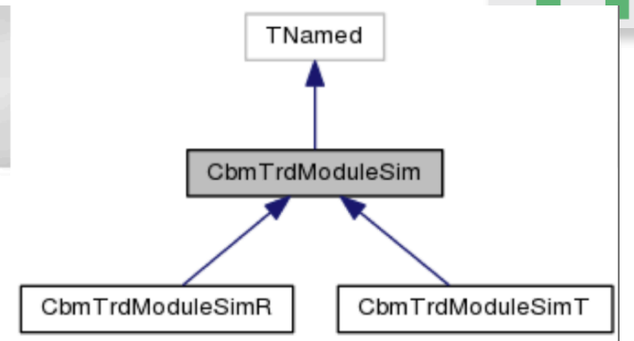
Output is:

- CbmTrdHits
  - Address
  - Time
  - Deposited charge from one particle

Reminder: TRD structural update
Mainly developed by Alexandru

The Object e.g. SimModule





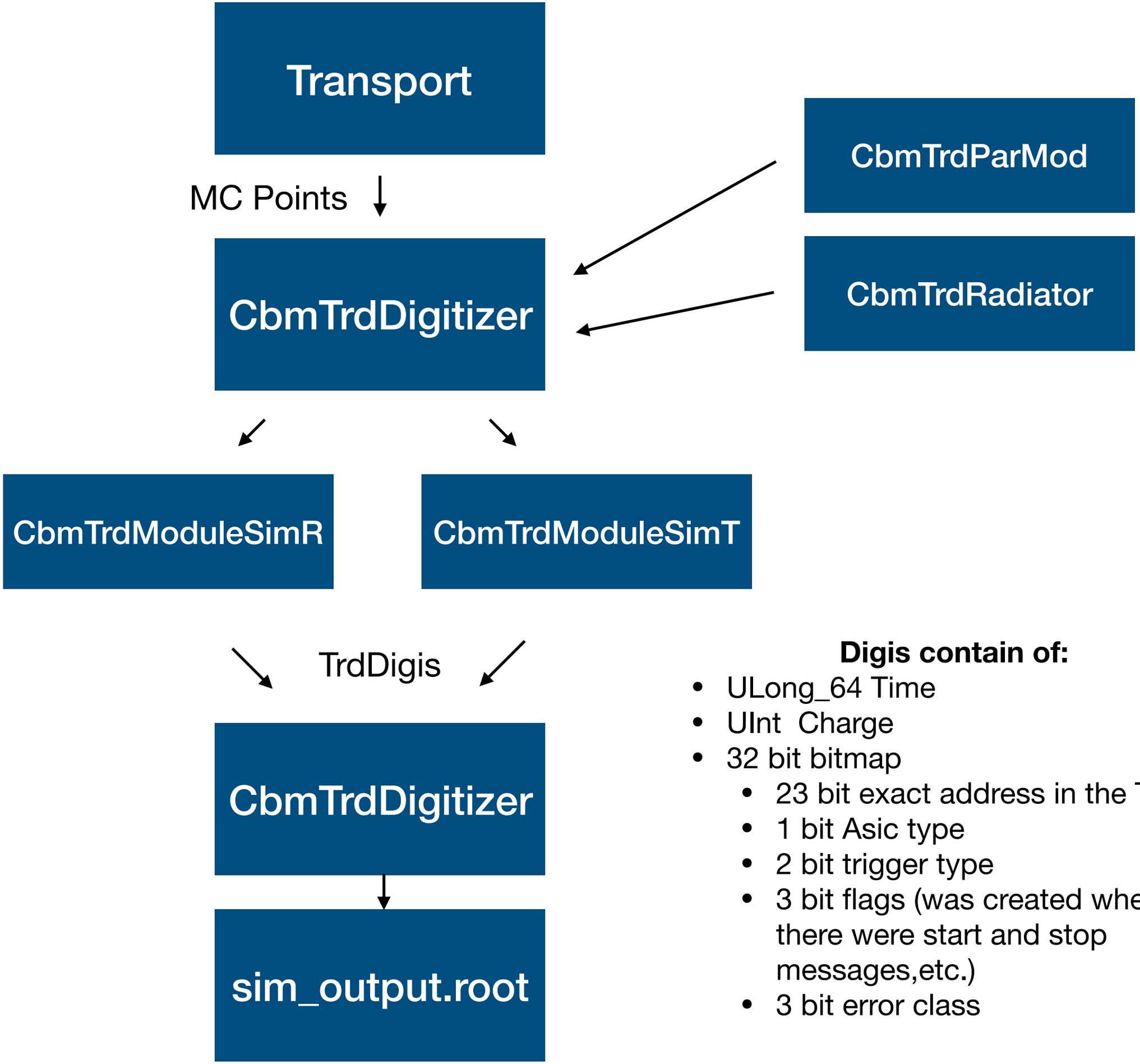
Public Types

enum	**CbmTrdModuleSimDef** { **kTime** = 0, **kNoise**, **kWeightDistance** }

Public Member Functions

	CbmTrdModuleSim () Default constructor.
	CbmTrdModuleSim (Int_t address, Int_t orientation, Double_t x, Double_t y, Double_t z, Double_t dx, Double_t dy, Double_t dz) Constructor with placement.
virtual	**~CbmTrdModuleSim** ()
virtual TClonesArray *	**GetDigiMap** ()
virtual Int_t	**FlushBuffer** (ULong64_t time=0)=0 Flush local digi buffer.
virtual Bool_t	**MakeDigi** (**CbmTrdPoint** *p, ULong64_t time, Bool_t TR=kFALSE)=0 Steering routine for converting MC point to digits.
virtual Bool_t	**MakeRaw** ()=0 Steering routine for converting MC point to raw format.
virtual void	**SetAsicPar** (const **CbmTrdParModAsic** *p)
virtual void	**SetChmbPar** (const **CbmTrdParModGas** *p)
virtual void	**SetDigiPar** (const **CbmTrdParModDigi** *p)
virtual void	**SetGainPar** (const **CbmTrdParModGain** *p)

Slide from Alexandru's talk at the 31th collab. meeting



Digis contain of:

- ULong_64 Time
- UInt Charge
- 32 bit bitmap
  - 23 bit exact address in the TRD
  - 1 bit Asic type
  - 2 bit trigger type
  - 3 bit flags (was created when there were start and stop messages,etc.)
  - 3 bit error class

Digitization (rectangular)

Calculate TR production

Calculate the TR production with the theory of a regular radiator:

$$\frac{dN}{d\omega} = \frac{4\alpha}{\omega(\kappa+1)} \frac{\left(1-e^{(-N\sigma)}\right)}{\left(1-e^{(-\sigma)}\right)} \cdot \sum_n \Theta_n \left(\frac{1}{\varrho_1+\Theta_n} - \frac{1}{\varrho_2+\Theta_n}\right)^2 [1-\cos(\varrho_1+\Theta_n)]$$

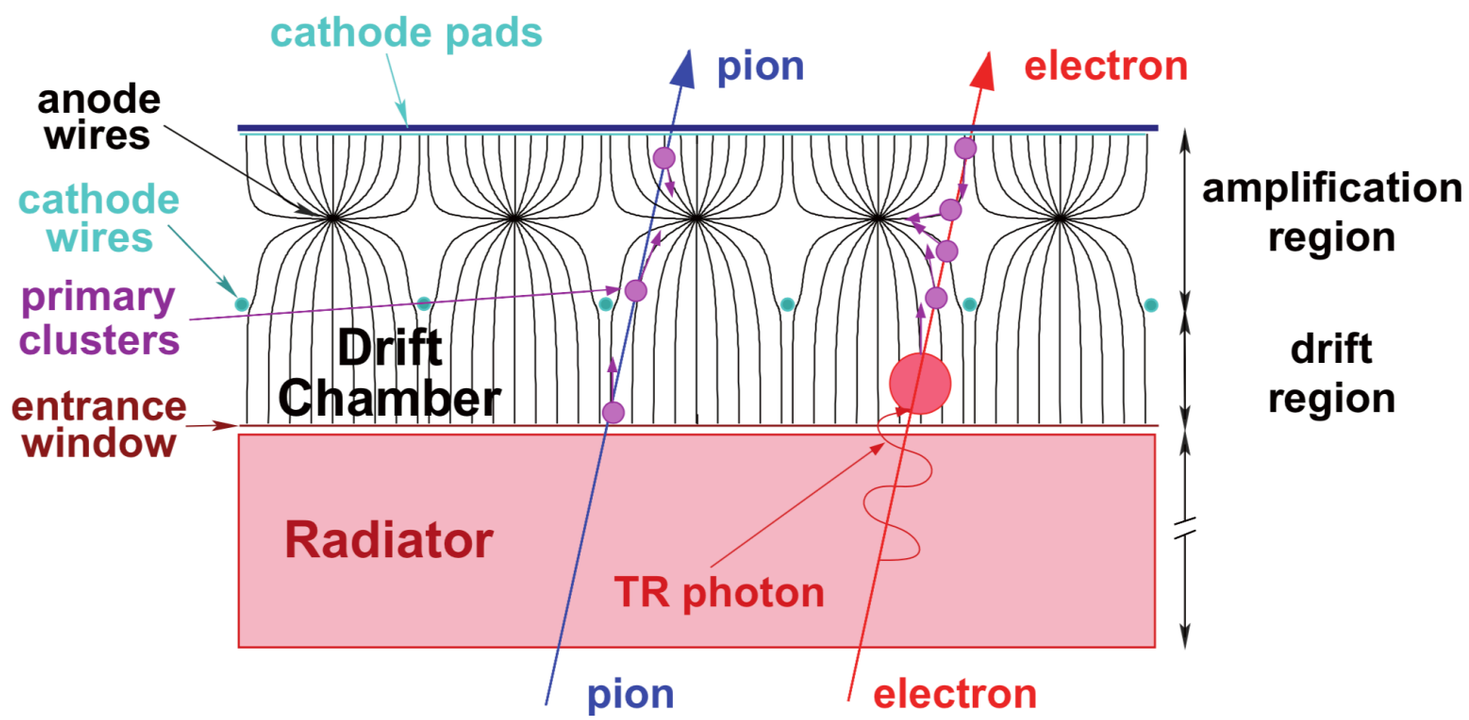
Afterwards we do not have single TR photons but an energy deposition with

$$E = E_{MC} + E_{TR}$$

Digitization

Calculate TR production

Distribute charge in the gas



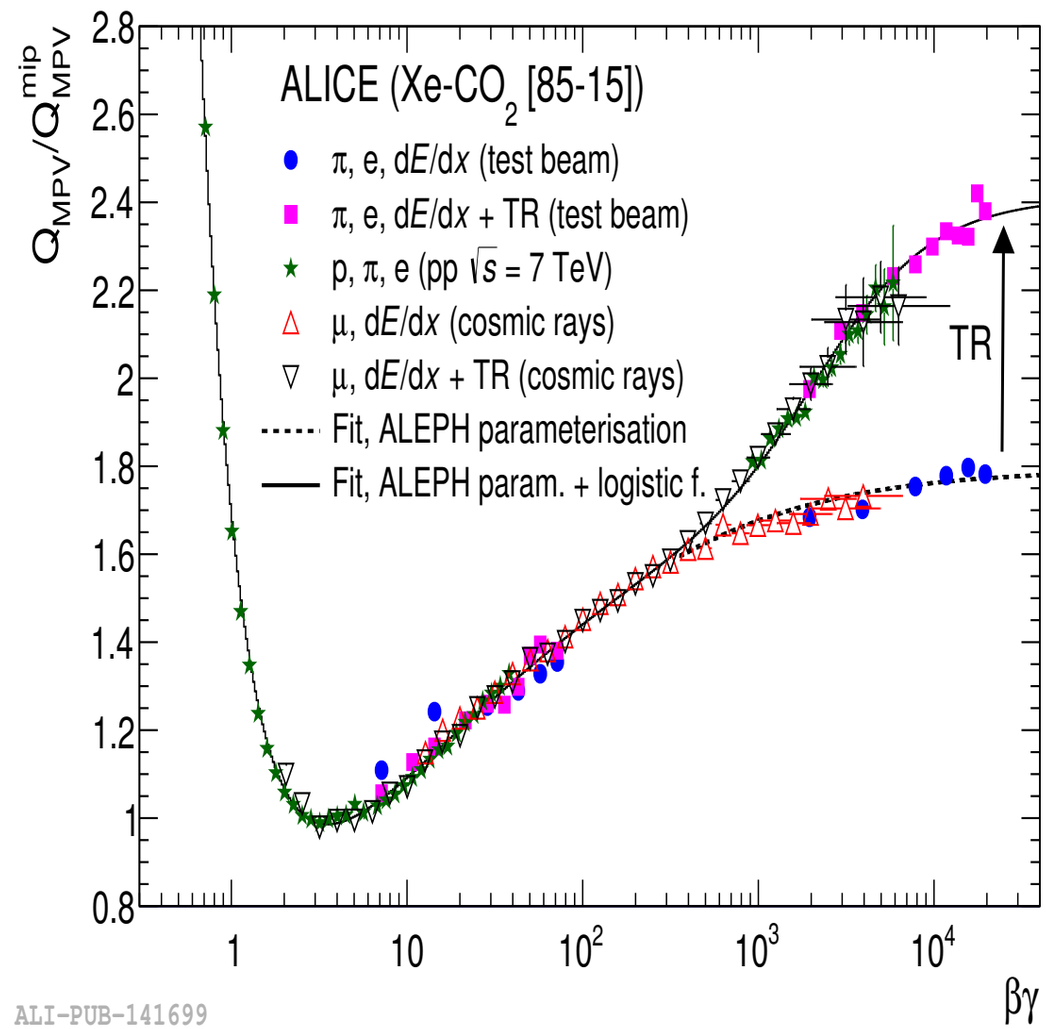
Geant does not give single ionisations

Digitization

Calculate TR production

Distribute charge in the gas

Bethe bloch approach



$$P(s) = \frac{1}{D} exp(-\frac{s}{D})$$

$$D = \frac{1}{< N_{prim} > \cdot f(\beta\gamma)}$$

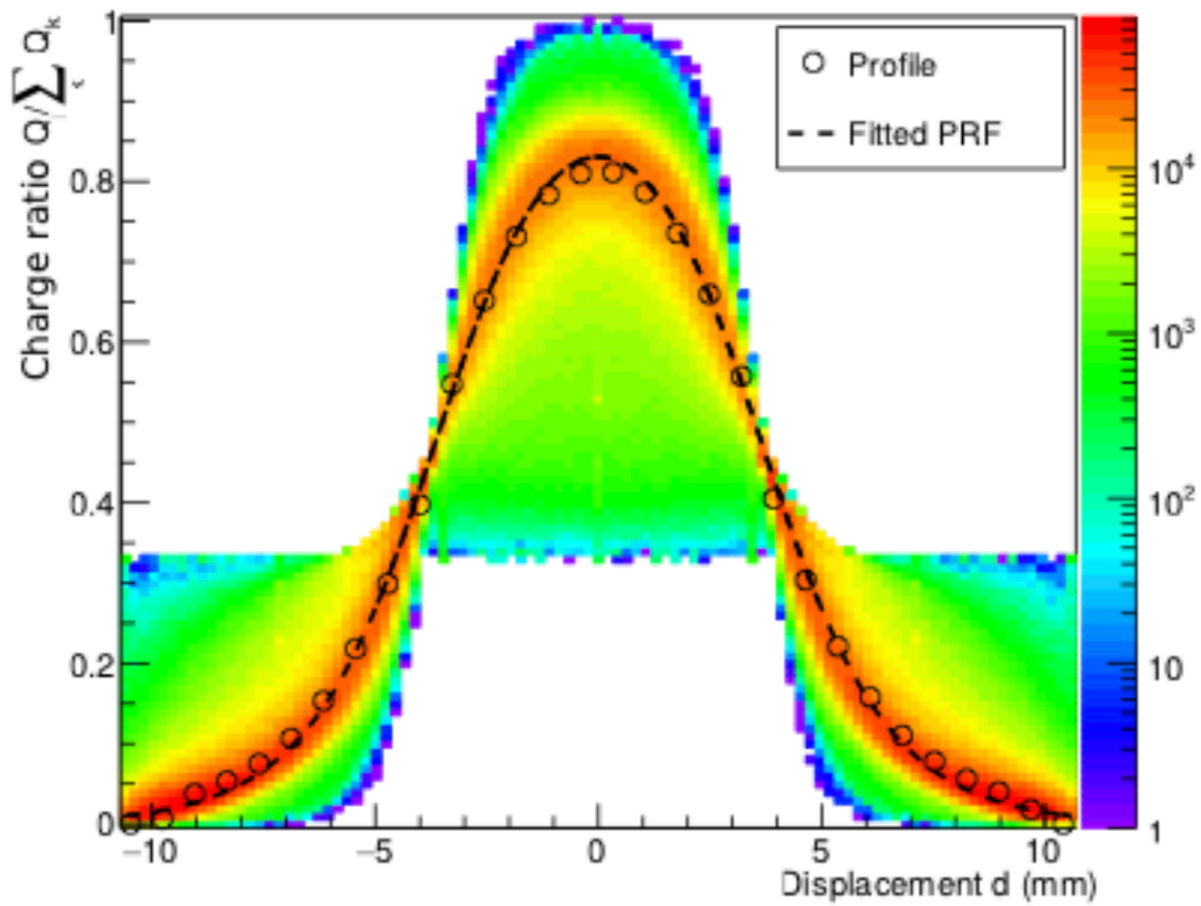
$$< N_{prim} >= 20.5$$ → From Geant 3

Digitization

Calculate TR production

Distribute charge in the gas

Distribute charge over the pad plane



Pad response function of a 3 pad cluster

Digitization

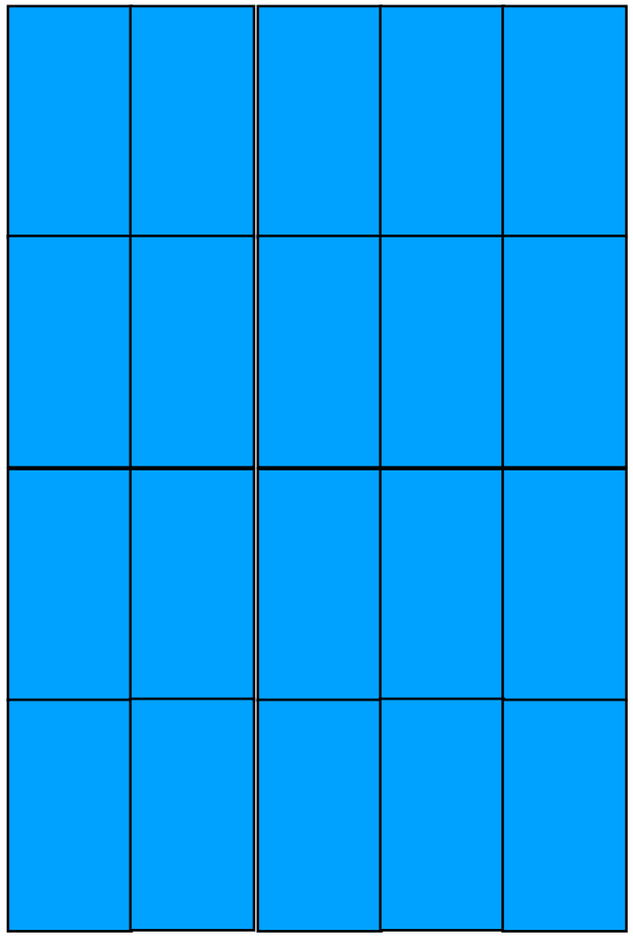
Calculate TR production

Distribute charge in the gas

Distribute charge over the pad plane

Pseudo trigger

Eloss > Threshold



Digitization

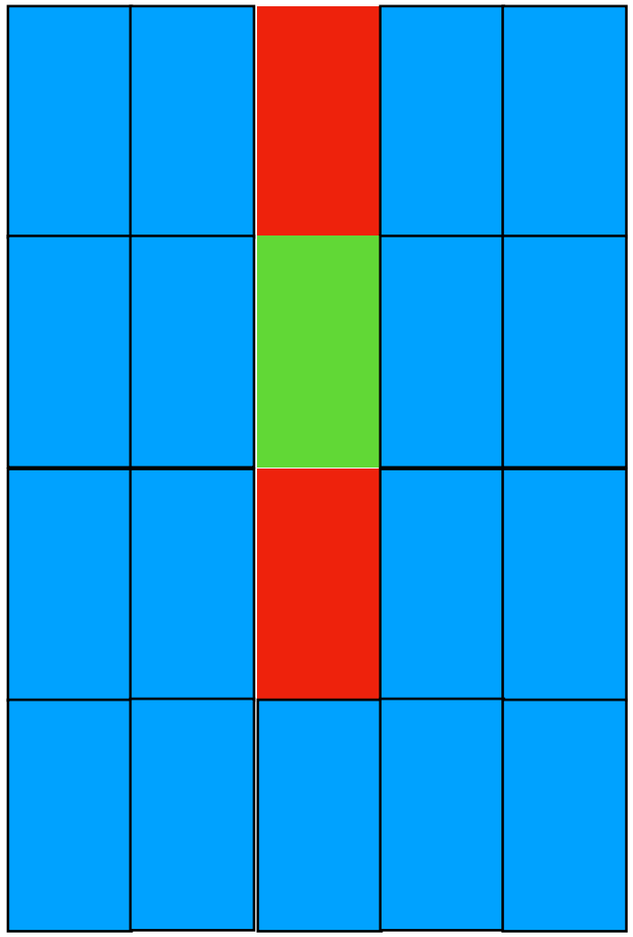
Calculate TR production

Distribute charge in the gas

Distribute charge over the pad plane

Pseudo trigger

Eloss > Threshold



Digitization

Calculate TR production

Distribute charge in the gas

Distribute charge over the pad plane

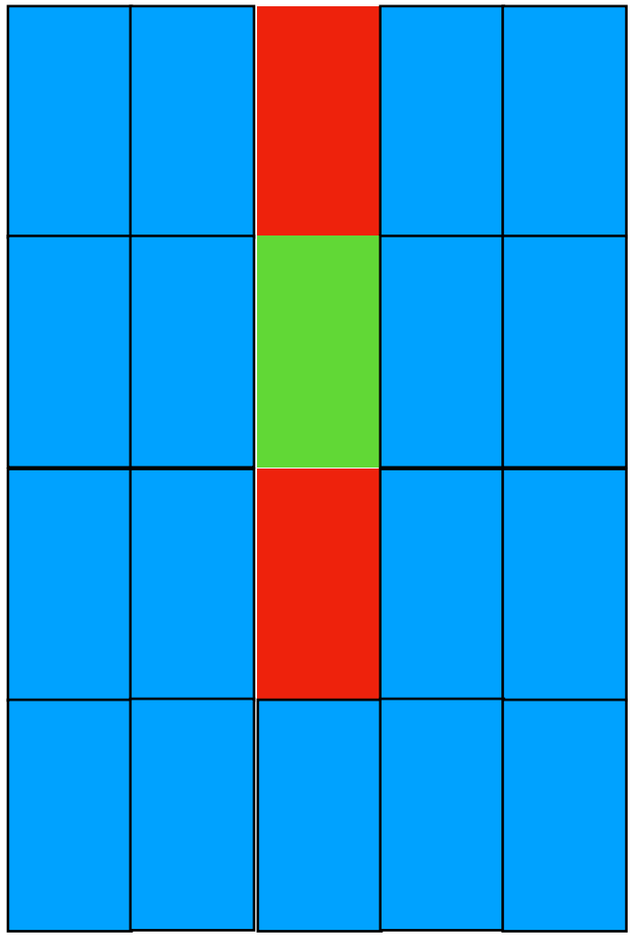
EB

TB

Pulse

Pseudo trigger

Eloss > Threshold



EB	Write info into digis In event pile up is handled via a standard map
TB	Time and charge get buffered Digis are getting released of the buffer when the time difference to the last digit is larger than the signal colle time (hence no interaction) Otherwise charge is just added up
Pulse	Charge is handled on ADC level Triggers are calculated Inter event pile up adds charge respective to its incoming time to existing pulses Multihits can be created

Software requirements

	EB mode	TB mode	Pulse mode
In - event interaction	Yes	Yes	Yes
Inter - event interaction	No	(Yes)	Yes
Realistic Pileup	No	(Yes)	Yes
Realistic Noise	No	(Yes)	Yes
Crosstalk	No	No	(Yes)

Front-end simulation

The Spadic response function is defined as:

$$g(t) = A \cdot \left(\frac{t}{\tau}\right)^2 \cdot \exp\left(-\frac{t}{\tau}\right) \quad (\text{for} \quad t \geq 0)$$

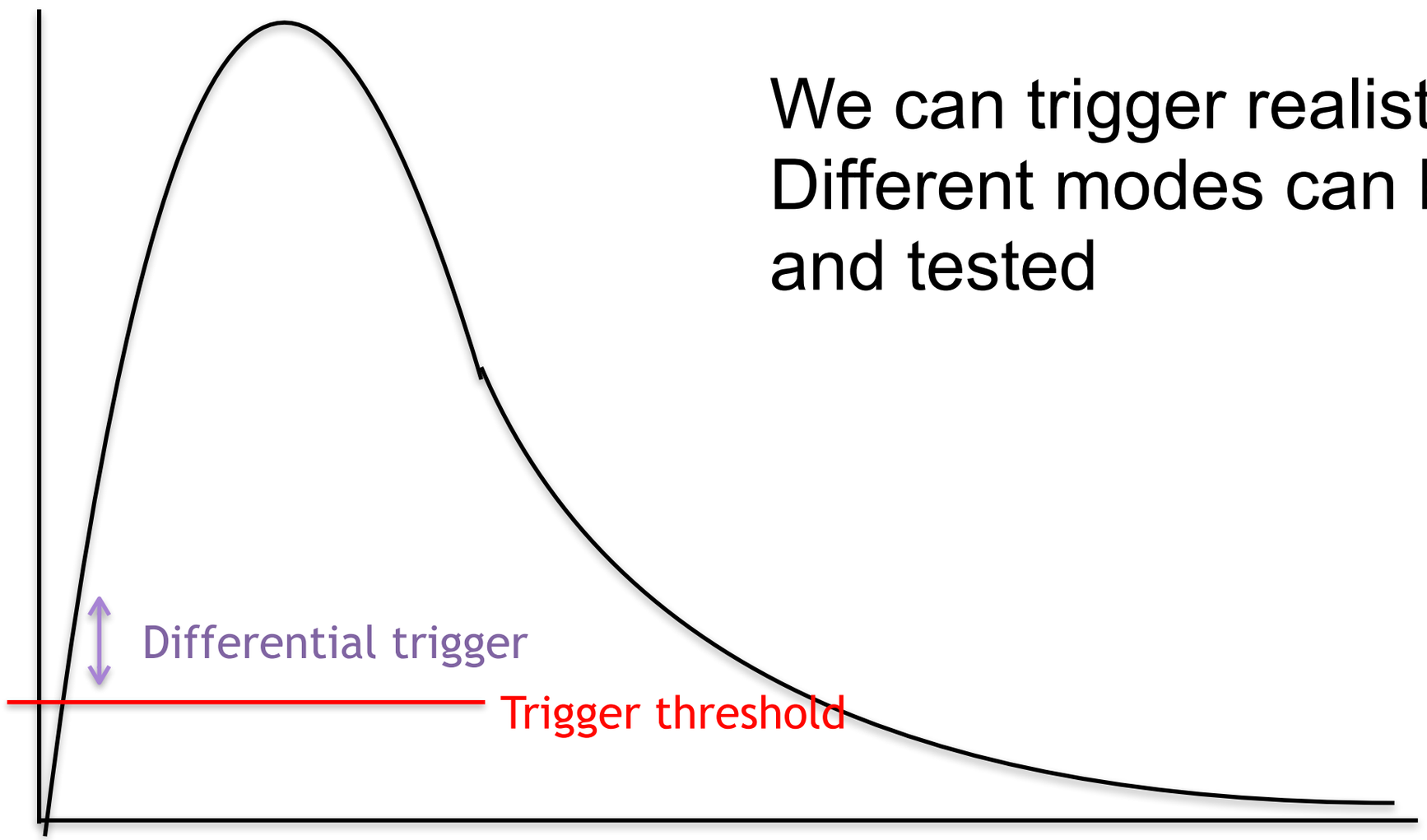
Where A is the calibration based on a MIP, which should be 7% of the ADC range

- 35 ADCs on the central Pad

τ = 120 ns

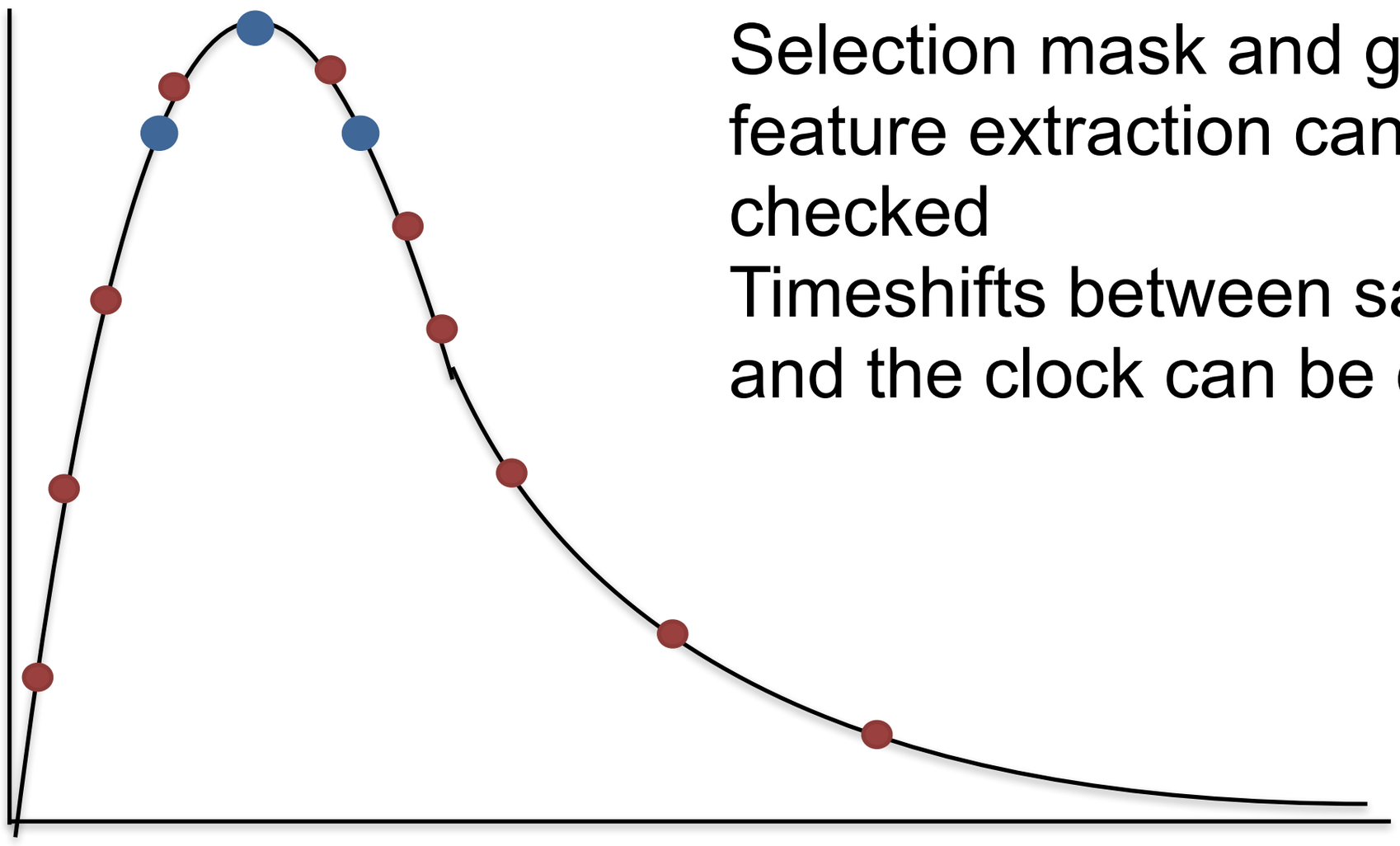
There is a first order shaper with $A \cdot \left(\frac{t}{\tau}\right)$ and a second order shaper with $A \cdot \left(\frac{t}{\tau}\right)^2$

Triggering



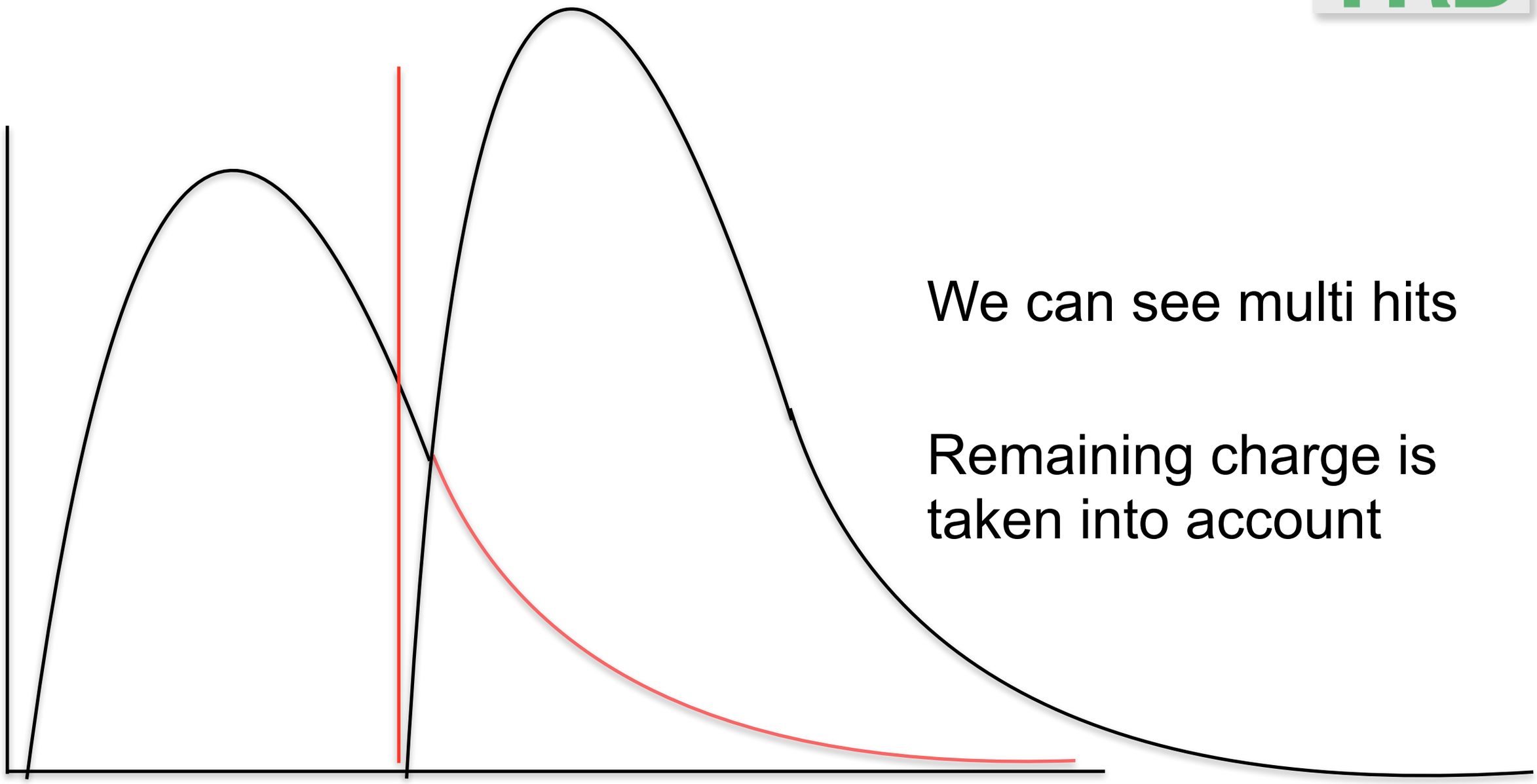
We can trigger realistically
Different modes can be set and tested

Feature extraction



Selection mask and general feature extraction can be checked
Timeshifts between sampling and the clock can be checked

Multihits

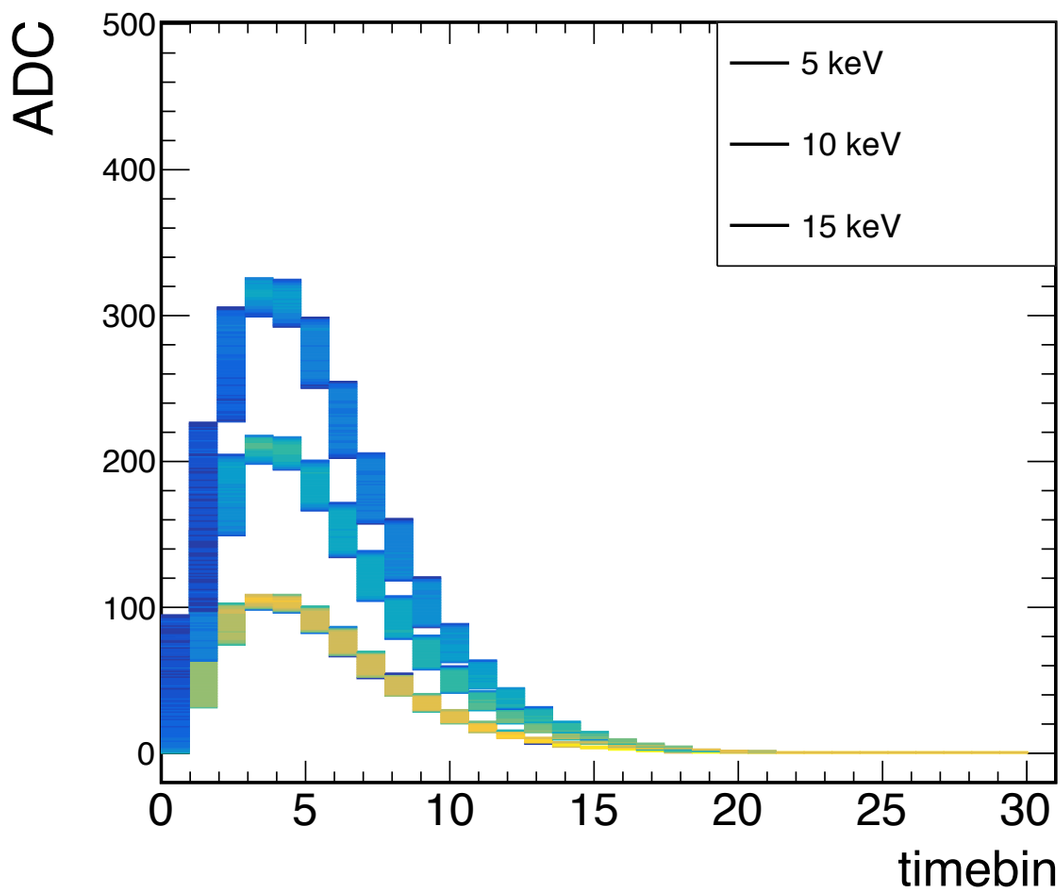


We can see multi hits

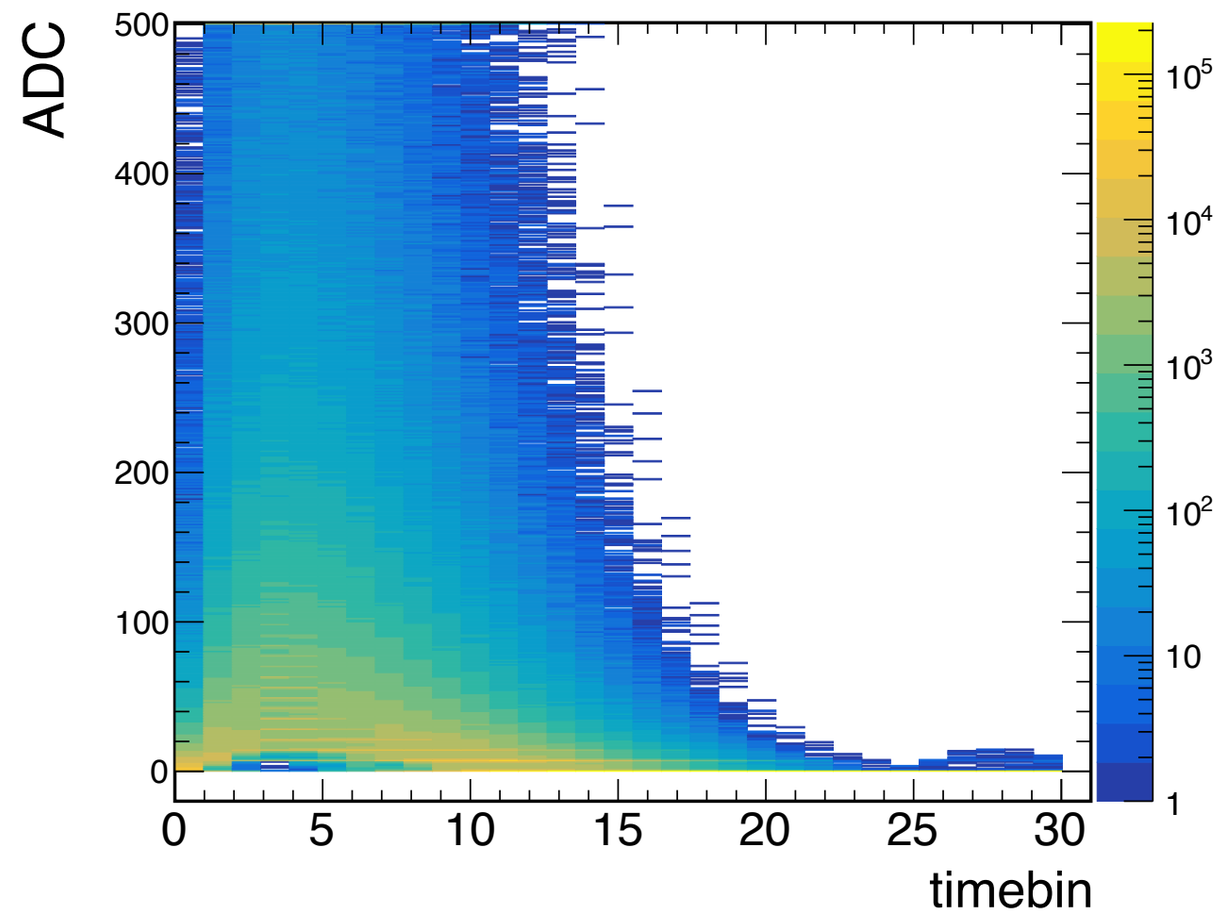
Remaining charge is taken into account

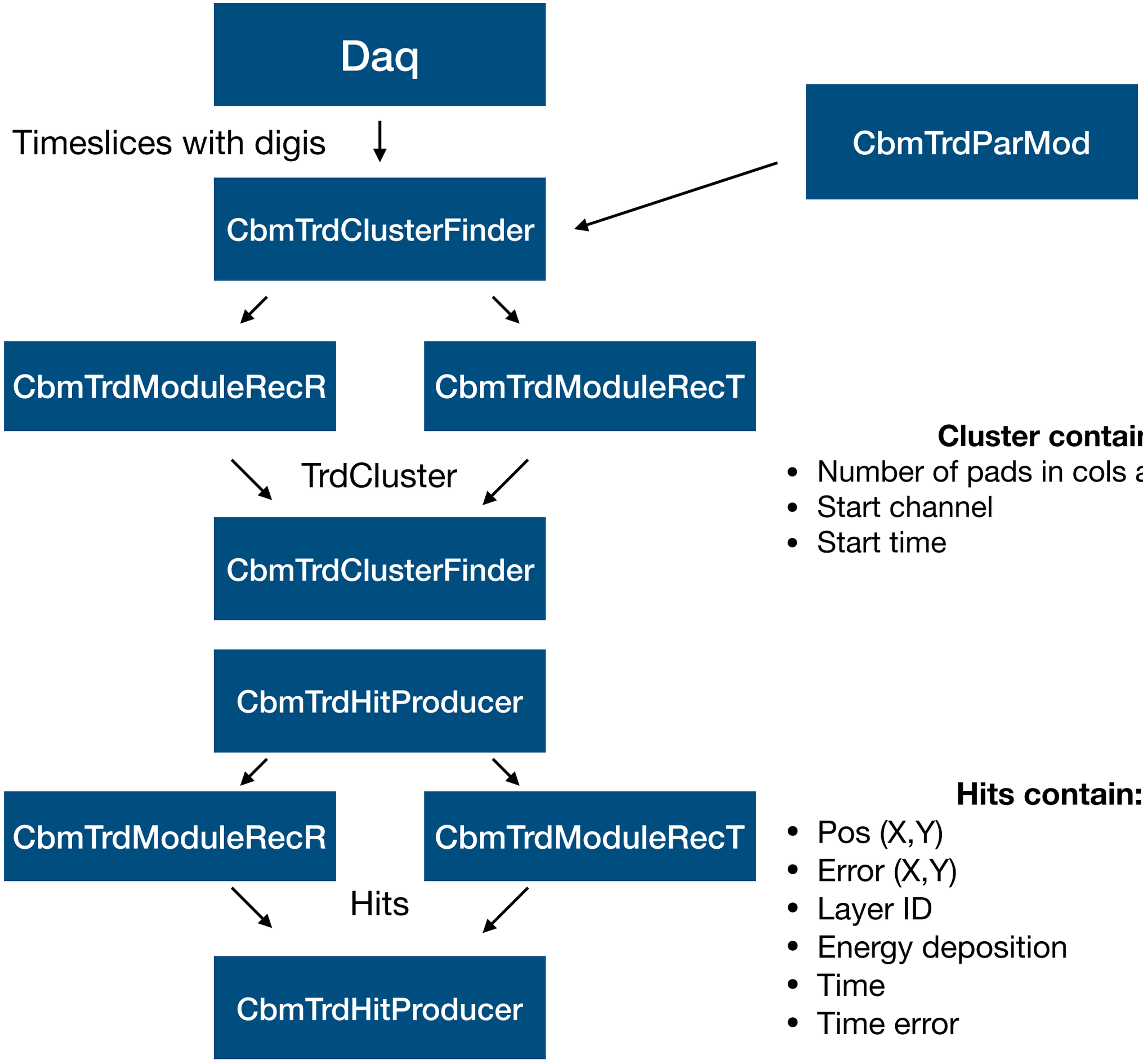
Simulated pulses

Selected energies



Full spectrum





Cluster contain:

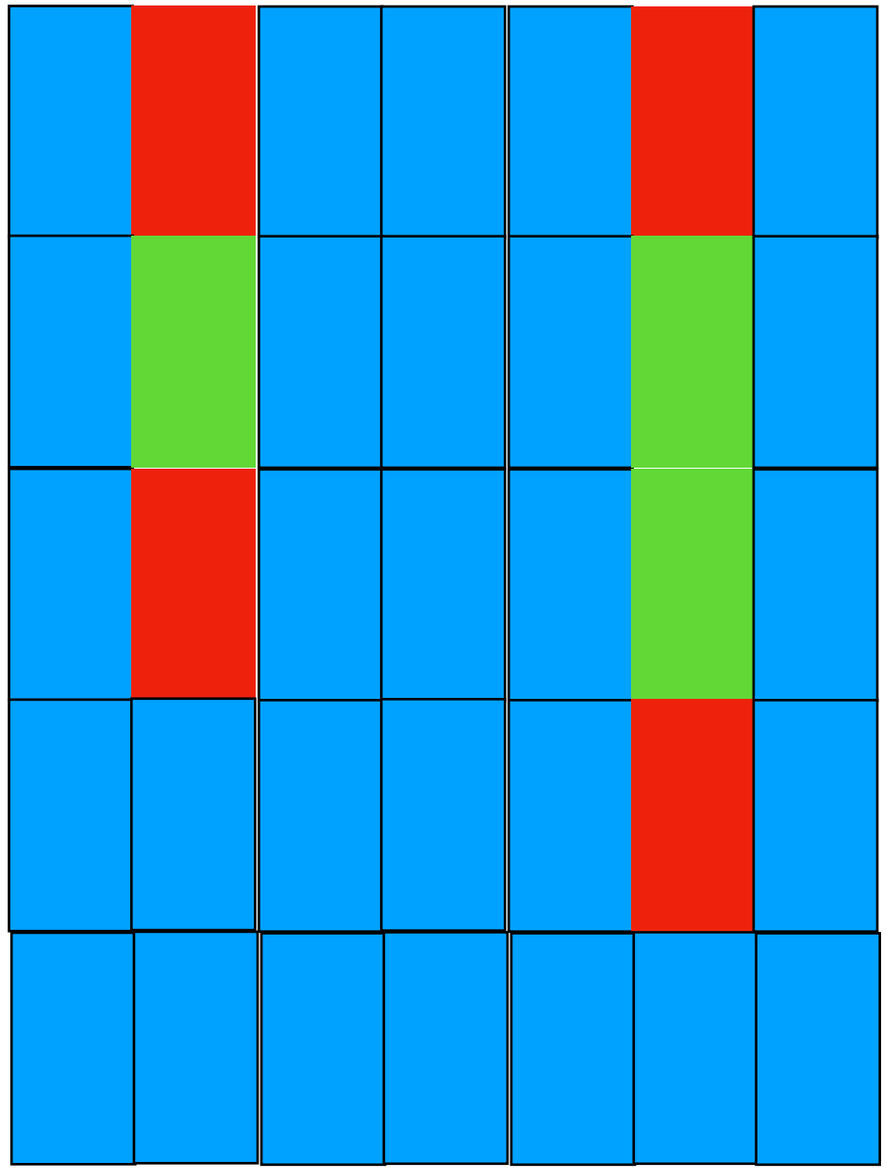
- Number of pads in cols and rows
- Start channel
- Start time

Hits contain:

- Pos (X,Y)
- Error (X,Y)
- Layer ID
- Energy deposition
- Time
- Time error

Reconstruction

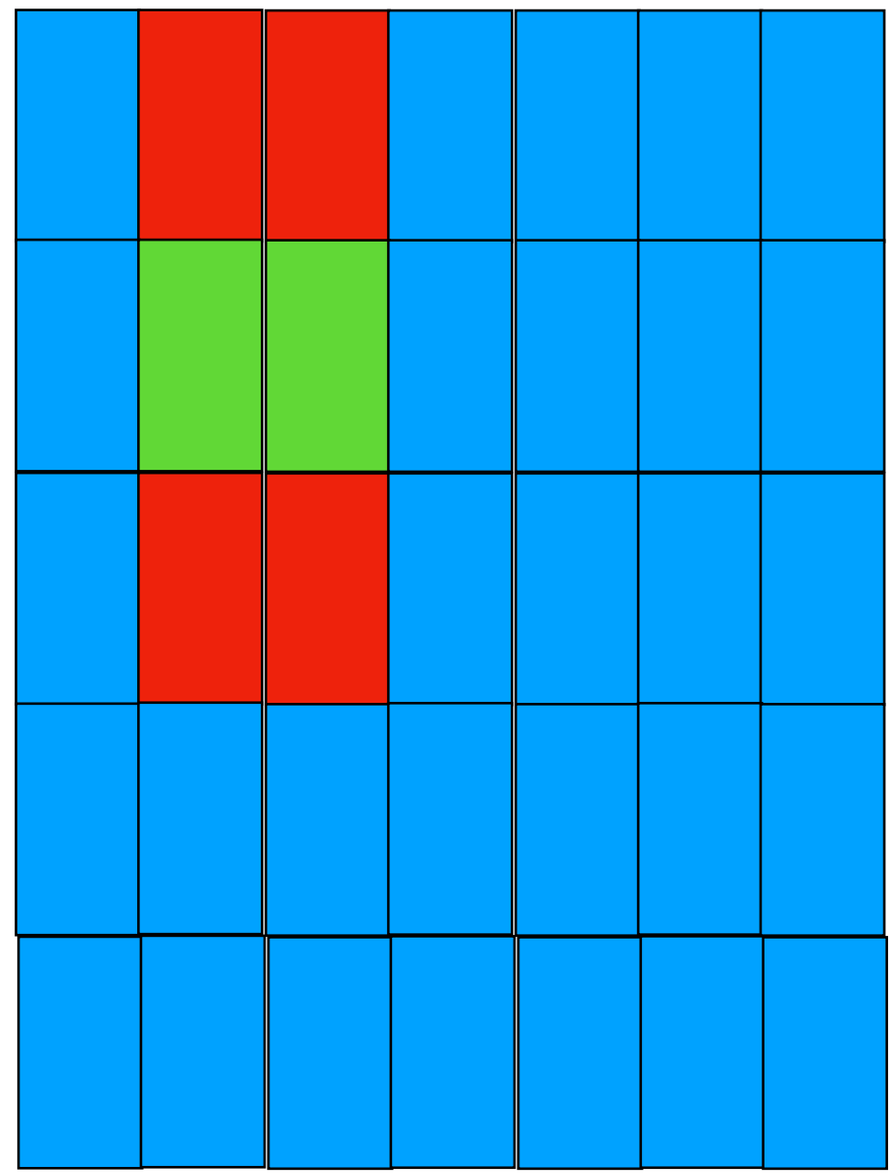
The Algorithm right now just searches self trigger and finds from there the corresponding forced neighbours or the adjacent self trigger



Reconstruction

In case of neighbour rows it compares the centres of gravity for both sub clusters

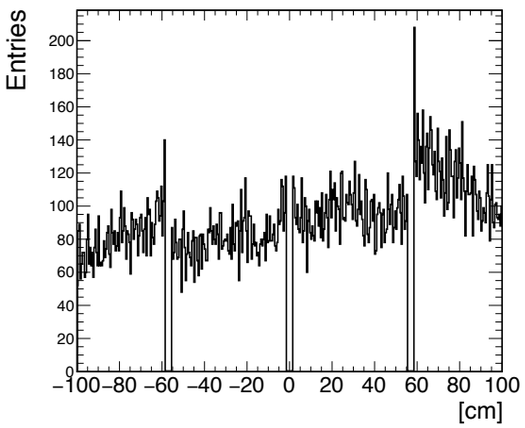
Position reconstruction is then later also done with the center of gravity



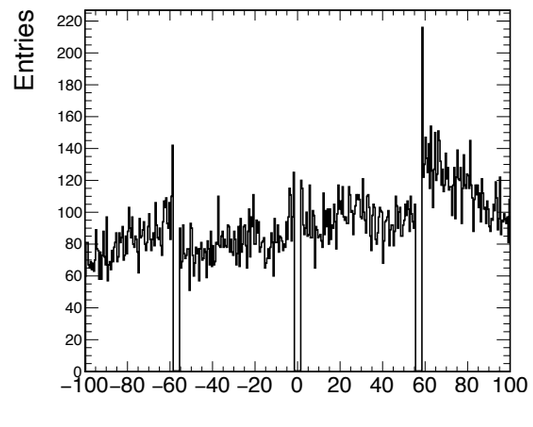
EB - 100 events - SIS100 - first layer

X

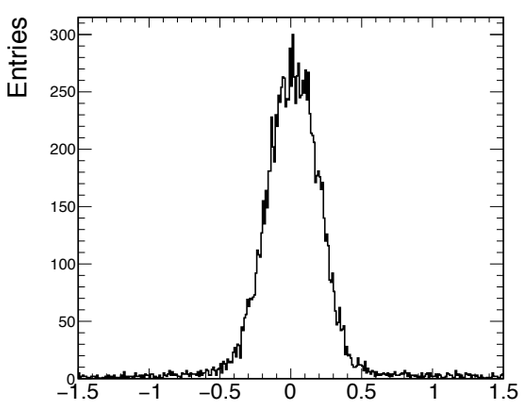
Rec



MC



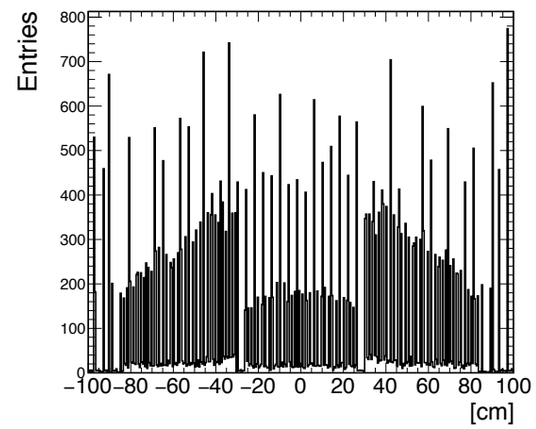
[cm]



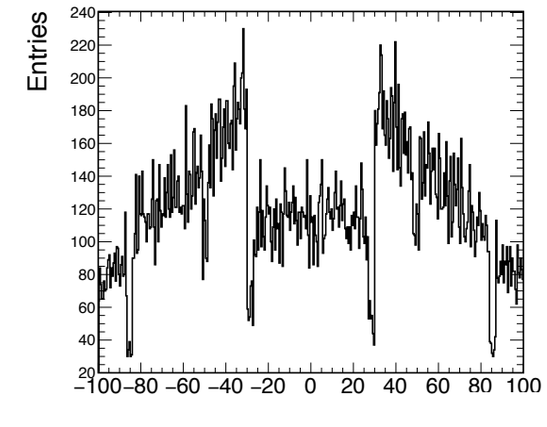
Pull

Y

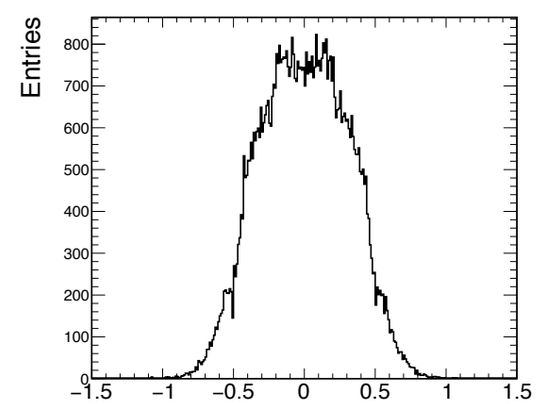
Rec



MC



[cm]

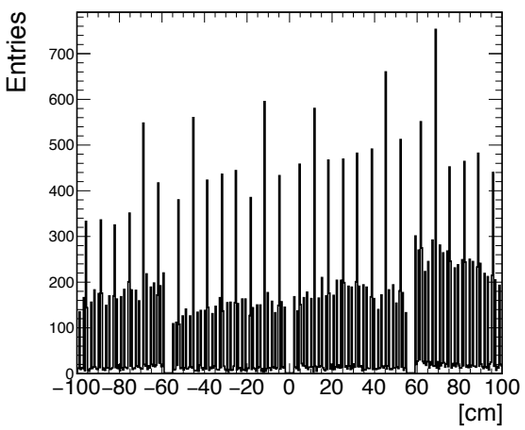


Pull

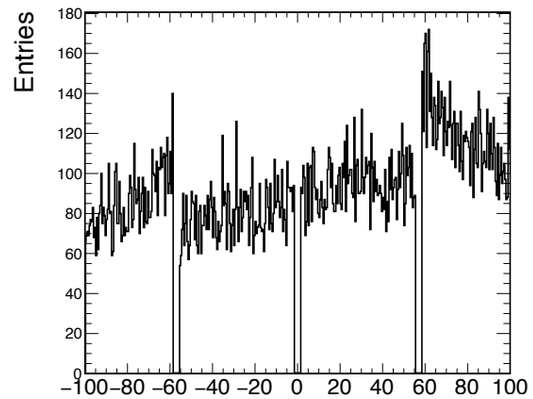
EB - 100 events - SIS100 - second layer

X

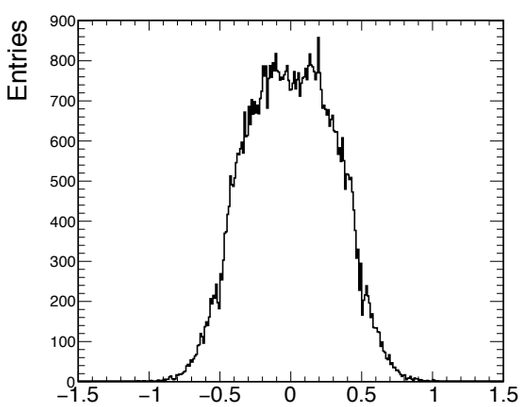
Rec



MC



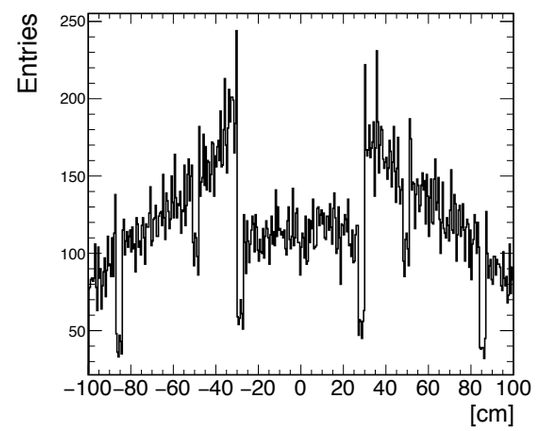
[cm]



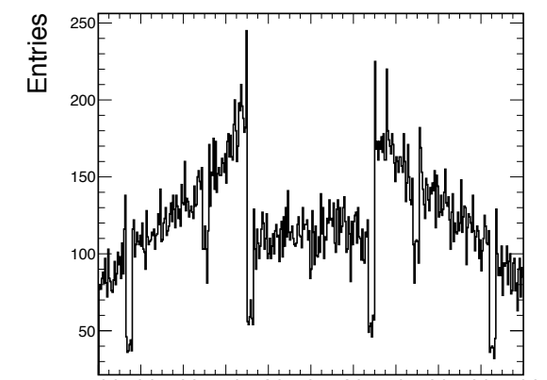
Pull

Y

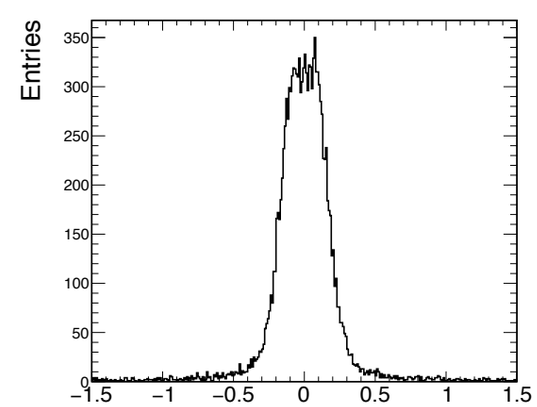
Rec



MC

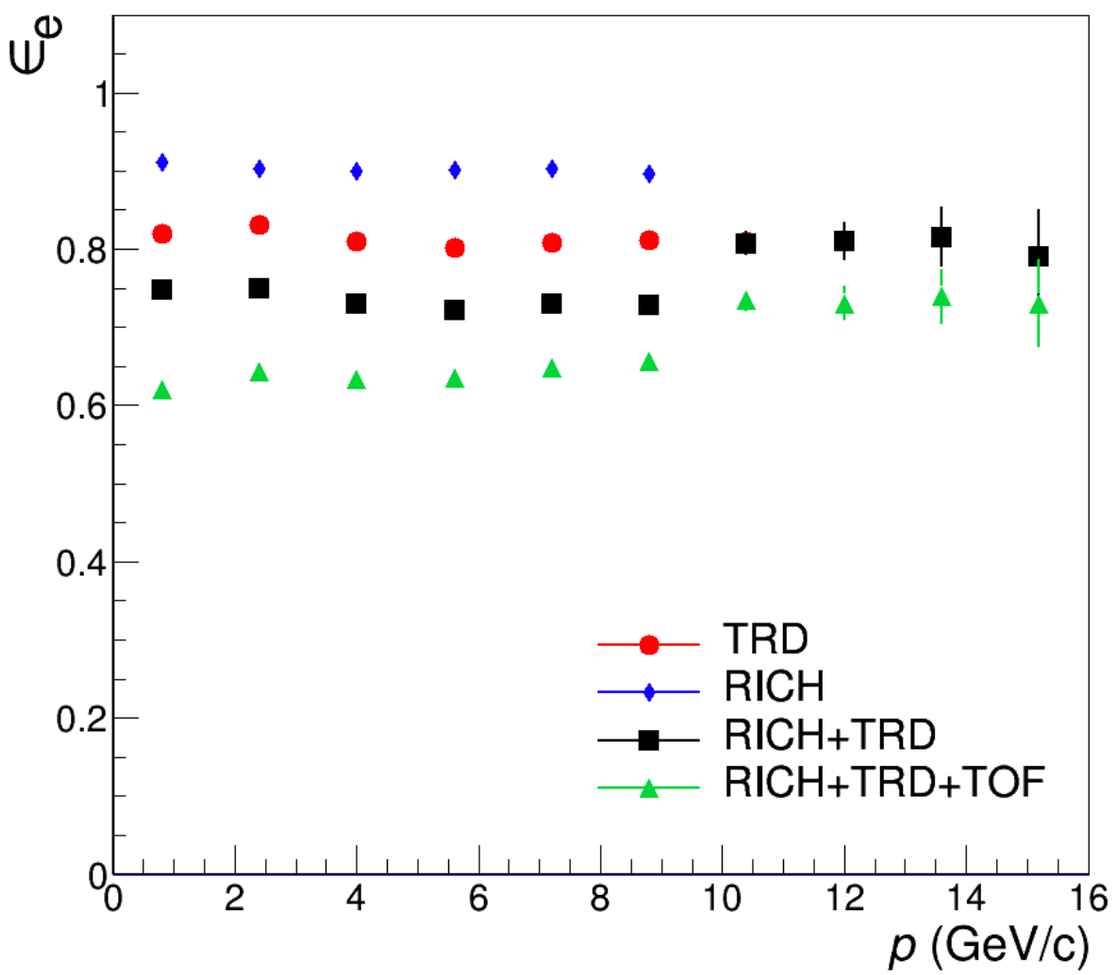


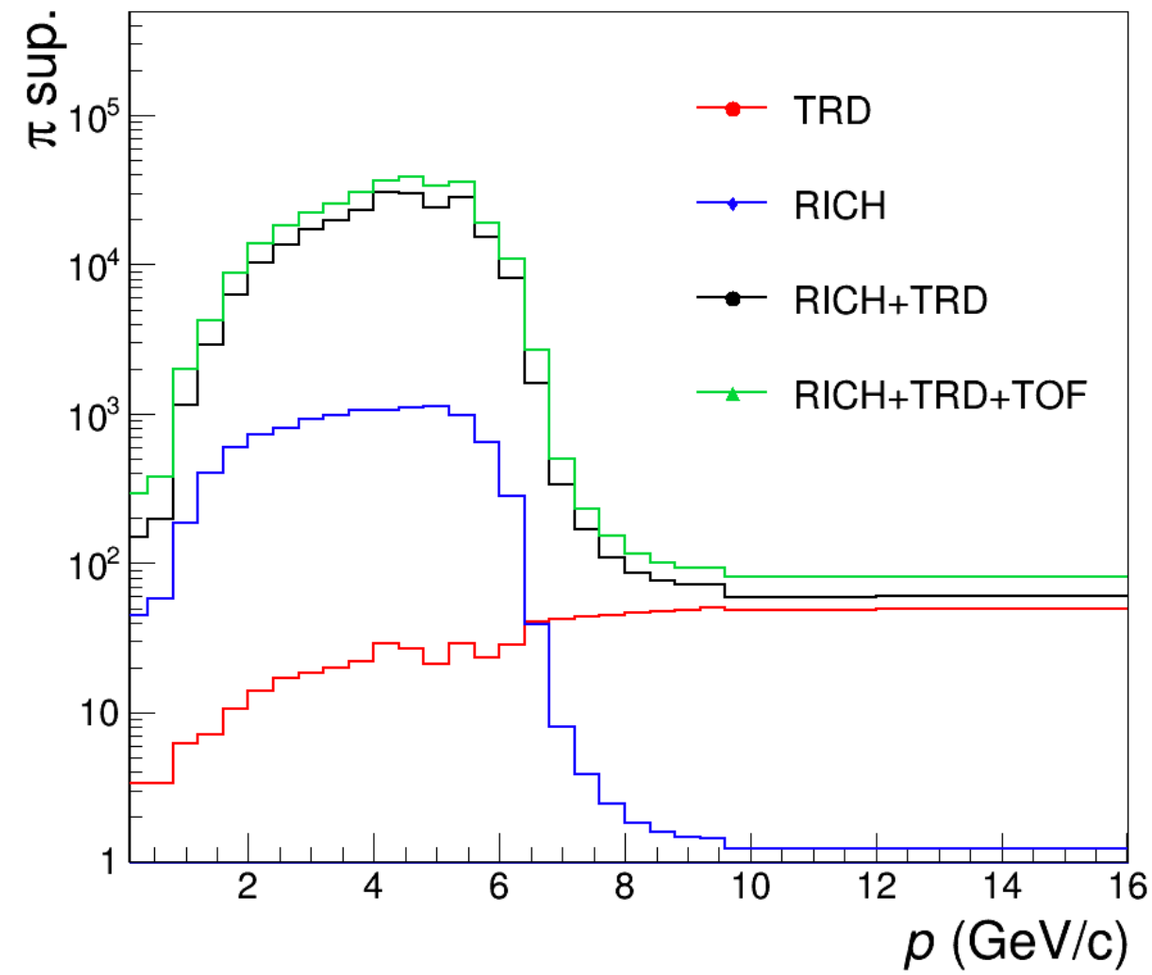
[cm]



Pull

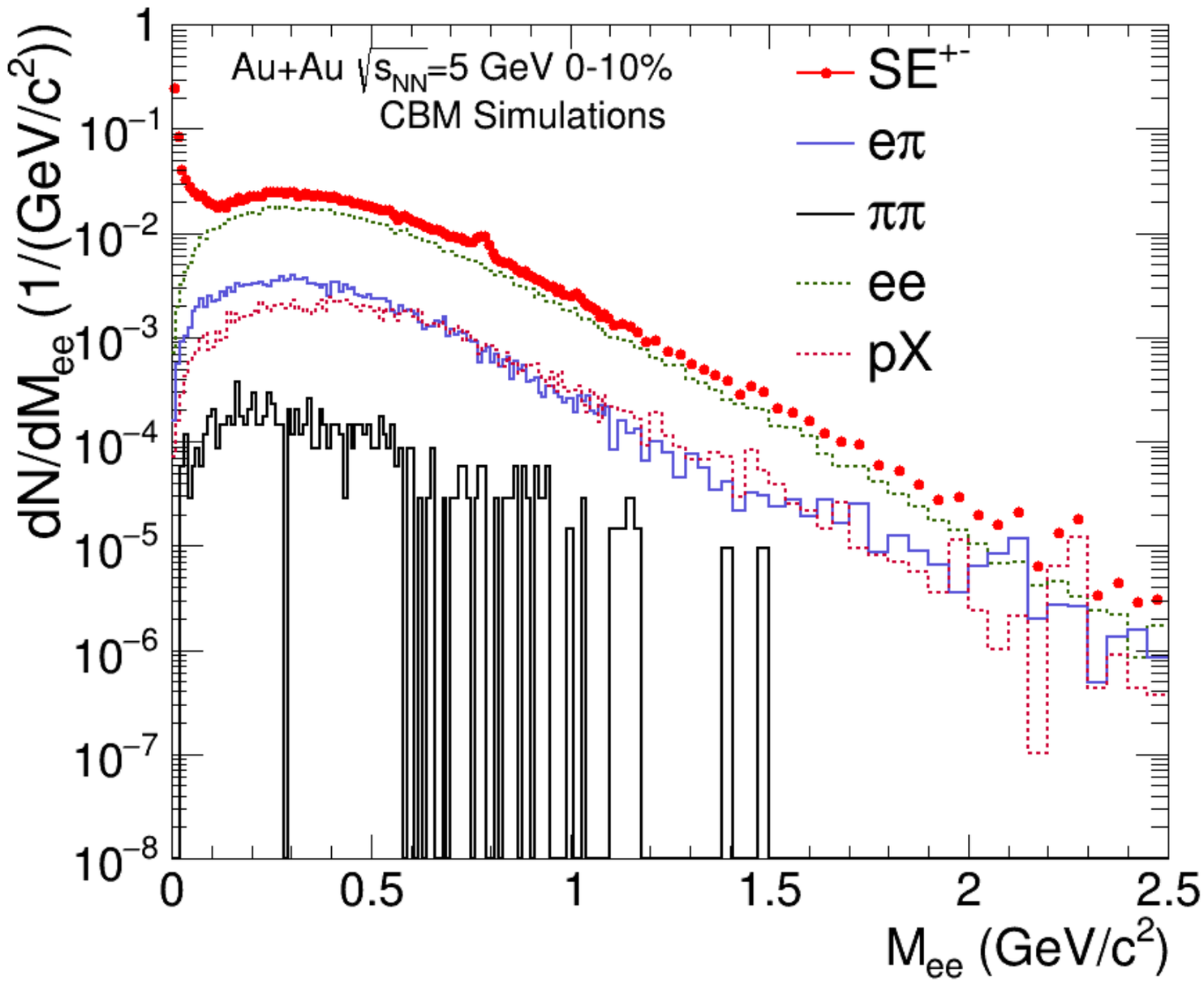
Pion suppression and electron efficiency





Invariant mass spectra of background contributions

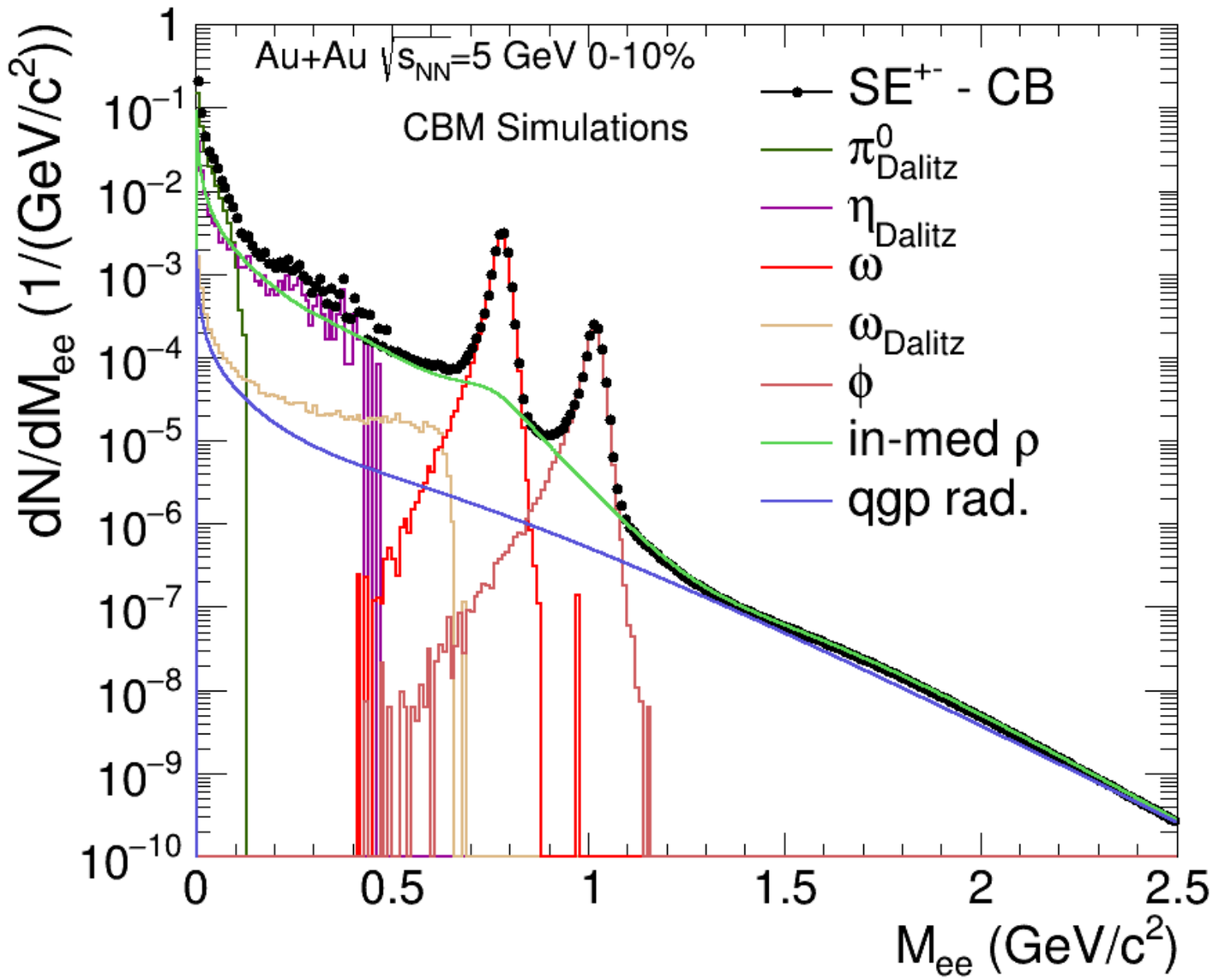
The hadronic background contributions are strongly suppressed



Invariant mass spectra of embedded signals

Clear access to low mass vector mesons and thermal radiation

Thermal radiation is scaled to expected yield at 4 weeks runtime



Open issues and/or discussion

Needed features / Outlook
Simulation of the entrance grid
More verification of the multi hit case in pulse simulation
Comparison of simulation and testbeam measurements
Use reconstruction class in cbmroot for testbeam data

Communication and documentation
Maybe we should think about our group own central software communication (as redmine)

- discuss open issues / remarks or questions from outside
- streamline working processes in similar directions and help students to find a starting point

Documentation….
(i am myself a very very bad example for good documentation but it should be a bit more in the focus)