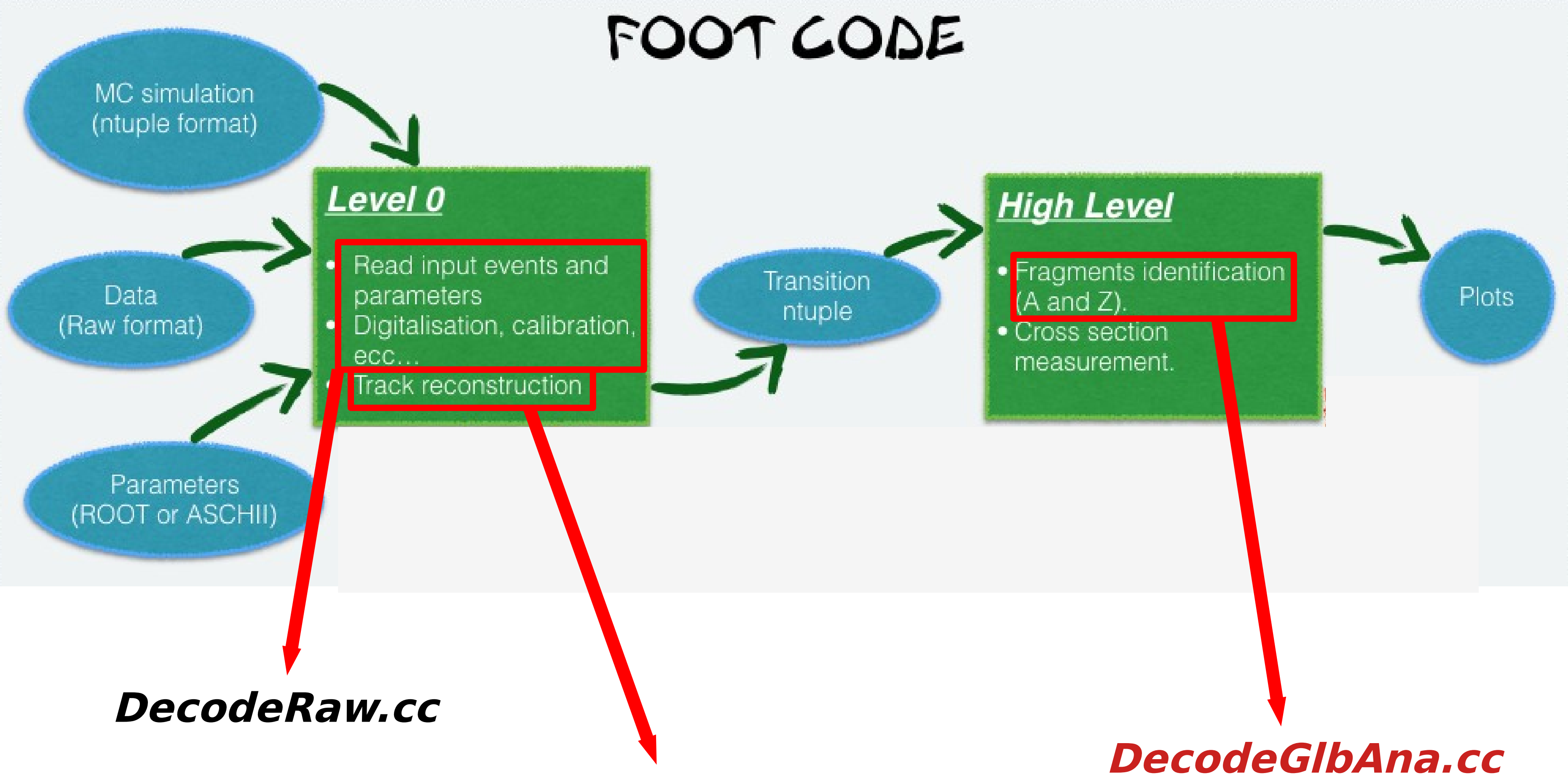
Procedure for Cross Section measurements analysis

SHOE organization

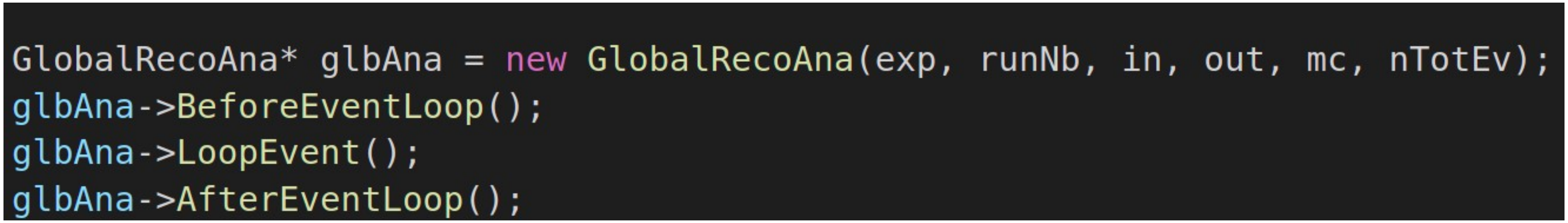

FOOT CODE
MC simulation (ntuple format)
Data (Raw format)
Parameters (ROOT or ASCHII)
Level 0
• Read input events and parameters
• Digitalisation, calibration, ecc…
• Track reconstruction
Transition ntuple
High Level
• Fragments identification (A and Z).
• Cross section measurement.
Plots

DecodeRaw.cc

DecodeGlb.cc

DecodeGlbAna.cc

DecodeGlbAna



```
GlobalRecoAna* glbAna = new GlobalRecoAna(exp, runNb, in, out, mc, nTotEv);
glbAna->BeforeEventLoop();
glbAna->LoopEvent();
glbAna->AfterEventLoop();
```

- It initializes a pointer to the class *GlobalRecoAna*
- For every event, loop methods are called.

glbAna->BeforeEventLoop()

- Reading of the input parameters
- Creation of the output file (trees and plots)
- Setting of variables about beam and target

glbAna->AfterEventLoop();

- Stamp luminosity (needed for cross-section measurements)
- Writing on files plots of variables computed in glbAna→LoopEvent();
- Closing files

glbAna->LoopEvent()

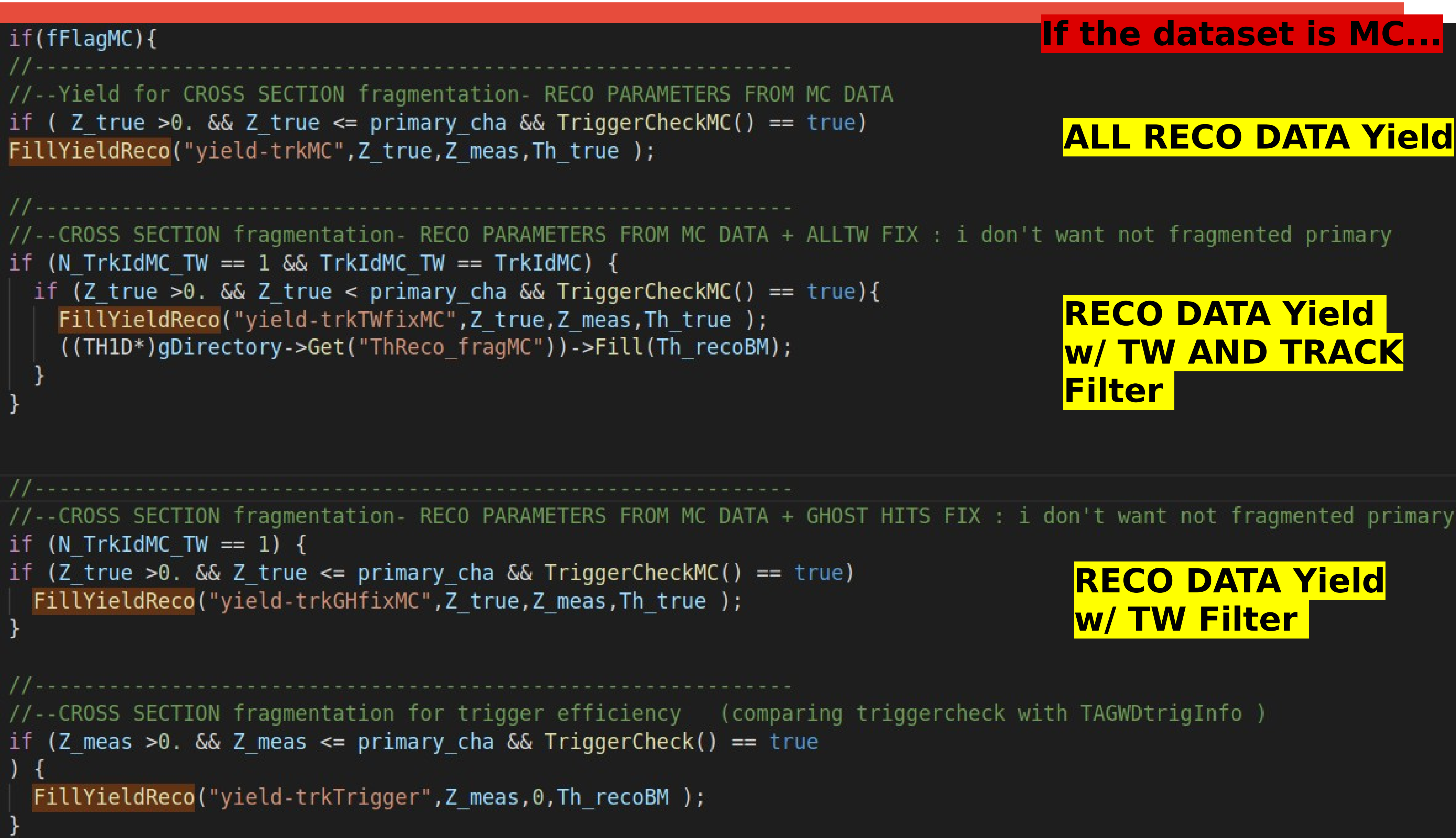
- **Loop on every event**
  - ClustersPositionStudy(currEvent) ~ *Study of variables wrt to cuts for glb track reconstruction*
  - study of bm tracks: to be moved
  - AlignmentStudy(currEvent, nt, 0) ~ studies about *alignment*
  - FragTriggerStudies() ~ study efficiency of hardware trigger (for real data)
  - TrackVsMCStudy() ~ *TW ghost hits studies (for MC data)*

  - **Loop on every global track**
    - GlbTrackPurityStudy()
    - MC studies
      - Crossing region studies
    - RECO values
      - FillMassPlots() ~ *Mass reconstruction*
      - study on ek binning
      - Unfolding

- **Loop on every global track (continue)**
  - **FillYieldReco**("yield-trkMC",Z_true,Z_meas,Th_true); ~ ***RECO/MC TRACKS counts for cross section measurements***
    - ***TriggerCheckMC() / TriggerCheck()*** **~** simulazione del trigger hardware

- **Loop on every MC particle**
  - **FillYieldMC**("yield-true_cut",charge_tr,theta_tr,Ek_tr_tot); ~ ***MC counts for cross section measurements***
  - TWAlgoStudy(); ~study of TW charge reconstruction

With these functions I obtain the yields in terms of the variables I am interested in,
From which i compute all the efficiencies and then cross section



If the dataset is MC...

```
if(fFlagMC){
//-------------------------------------------------------------
//--Yield for CROSS SECTION fragmentation- RECO PARAMETERS FROM MC DATA
if ( Z_true >0. && Z_true <= primary_cha && TriggerCheckMC() == true)
FillYieldReco("yield-trkMC",Z_true,Z_meas,Th_true );
```

ALL RECO DATA Yield

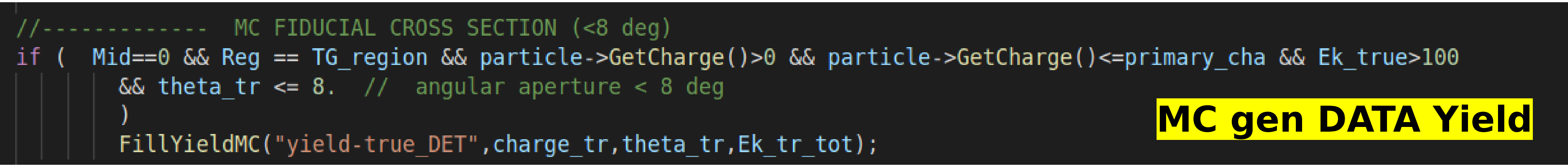
```
//-------------------------------------------------------------
//--CROSS SECTION fragmentation- RECO PARAMETERS FROM MC DATA + ALLTW FIX : i don't want not fragmented primary
if (N_TrkIdMC_TW == 1 && TrkIdMC_TW == TrkIdMC) {
  if (Z_true >0. && Z_true < primary_cha && TriggerCheckMC() == true){
    FillYieldReco("yield-trkTWfixMC",Z_true,Z_meas,Th_true );
    ((TH1D*)gDirectory->Get("ThReco_fragMC"))->Fill(Th_recoBM);
  }
}
```

RECO DATA Yield w/ TW AND TRACK Filter

```
//-------------------------------------------------------------
//--CROSS SECTION fragmentation- RECO PARAMETERS FROM MC DATA + GHOST HITS FIX : i don't want not fragmented primary
if (N_TrkIdMC_TW == 1) {
if (Z_true >0. && Z_true <= primary_cha && TriggerCheckMC() == true)
  FillYieldReco("yield-trkGHfixMC",Z_true,Z_meas,Th_true );
}
```

RECO DATA Yield w/ TW Filter

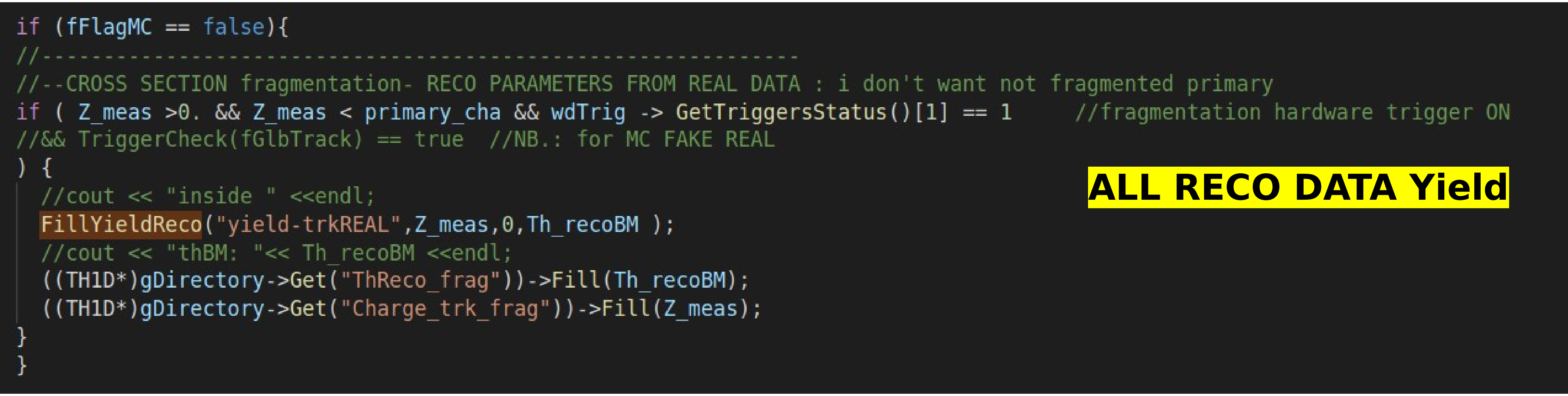
```
//-------------------------------------------------------------
//--CROSS SECTION fragmentation for trigger efficiency   (comparing triggercheck with TAGWDtrigInfo )
if (Z_meas >0. && Z_meas <= primary_cha && TriggerCheck() == true
) {
  FillYieldReco("yield-trkTrigger",Z_meas,0,Th_recoBM );
}
```



```
//-------------  MC FIDUCIAL CROSS SECTION (<8 deg)
if (  Mid==0 && Reg == TG_region && particle->GetCharge()>0 && particle->GetCharge()<=primary_cha && Ek_true>100
        && theta_tr <= 8.  //  angular aperture < 8 deg
        )
        FillYieldMC("yield-true_DET",charge_tr,theta_tr,Ek_tr_tot);
```

MC gen DATA Yield

If the dataset is REAL...



```
if (fFlagMC == false){
//-------------------------------------------------------------
//--CROSS SECTION fragmentation- RECO PARAMETERS FROM REAL DATA : i don't want not fragmented primary
if ( Z_meas >0. && Z_meas < primary_cha && wdTrig -> GetTriggersStatus()[1] == 1     //fragmentation hardware trigger ON
//&& TriggerCheck(fGlbTrack) == true  //NB.: for MC FAKE REAL
) {
  //cout << "inside " <<endl;
  FillYieldReco("yield-trkREAL",Z_meas,0,Th_recoBM );
  //cout << "thBM: "<< Th_recoBM <<endl;
  ((TH1D*)gDirectory->Get("ThReco_frag"))->Fill(Th_recoBM);
  ((TH1D*)gDirectory->Get("Charge_trk_frag"))->Fill(Z_meas);
}
}
```

ALL RECO DATA Yield

ALL RECO DATA Yield

RECO DATA Yield w/ TW AND TRACK Filter

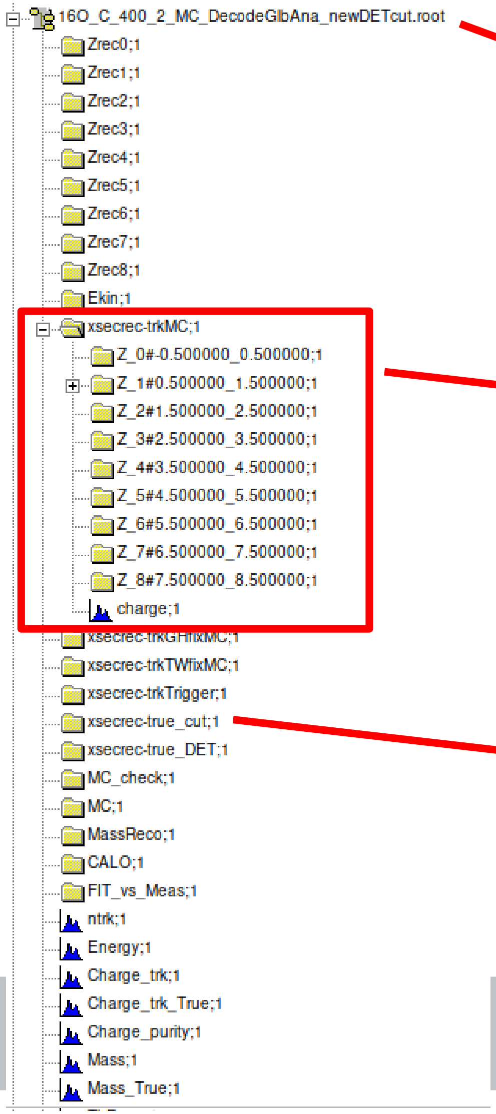
MC gen DATA Yield

Mainly the yields in these 3 folders are used in my analysis.

DecodeGlbAna output

- ROOT file after DecodeGlbAna.cxx execution
  Example from ROOT TBrowser:

Results of FillYield() are in these folders:



```
160_C_400_2_MC_DecodeGlbAna_newDETcut.root
  Zrec0;1
  Zrec1;1
  Zrec2;1
  Zrec3;1
  Zrec4;1
  Zrec5;1
  Zrec6;1
  Zrec7;1
  Zrec8;1
  Ekin;1
  xsecrec-trkMC;1
    Z_0#-0.500000_0.500000;1
    Z_1#0.500000_1.500000;1
    Z_2#1.500000_2.500000;1
    Z_3#2.500000_3.500000;1
    Z_4#3.500000_4.500000;1
    Z_5#4.500000_5.500000;1
    Z_6#5.500000_6.500000;1
    Z_7#6.500000_7.500000;1
    Z_8#7.500000_8.500000;1
    charge;1
  xsecrec-trkGHfixMC;1
  xsecrec-trkTWfixMC;1
  xsecrec-trkTrigger;1
  xsecrec-true_cut;1
  xsecrec-true_DET;1
  MC_check;1
  MC;1
  MassReco;1
  CALO;1
  FIT_vs_Meas;1
  ntrk;1
  Energy;1
  Charge_trk;1
  Charge_trk_True;1
  Charge_purity;1
  Mass;1
  Mass_True;1
```

Name of the file

Events are organized in subfolders of Z
<variable name>_<n° of bins>#<lower limit>_<upper limit>

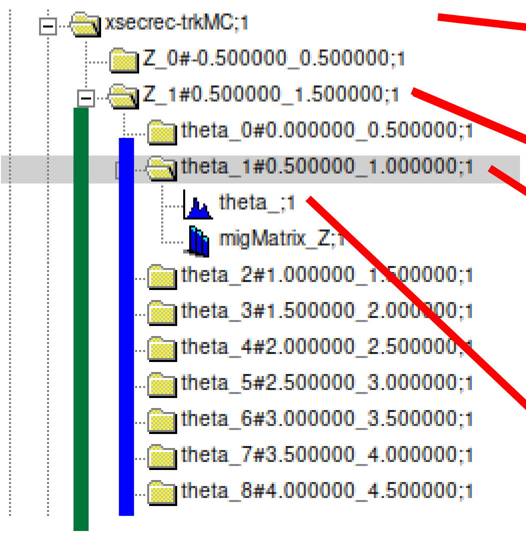
Folders in which there are plots of events from which
Obtaining cross section measurements

First approximation logic

ROOT dataset
From SHOE

- ROOT file after DecodeGlbAna.cxx execution
  Example from ROOT TBrowser:

Every Yield folder is made in this way:



```
xsecrec-trkMC;1
  Z_0#-0.500000_0.500000;1
  Z_1#0.500000_1.500000;1
    theta_0#0.000000_0.500000;1
    theta_1#0.500000_1.000000;1
      theta_;1
      migMatrix_Z;1
    theta_2#1.000000_1.500000;1
    theta_3#1.500000_2.000000;1
    theta_4#2.000000_2.500000;1
    theta_5#2.500000_3.000000;1
    theta_6#3.000000_3.500000;1
    theta_7#3.500000_4.000000;1
    theta_8#4.000000_4.500000;1
```

Folders in which there are plots of events from which
Obtaining cross section measurements

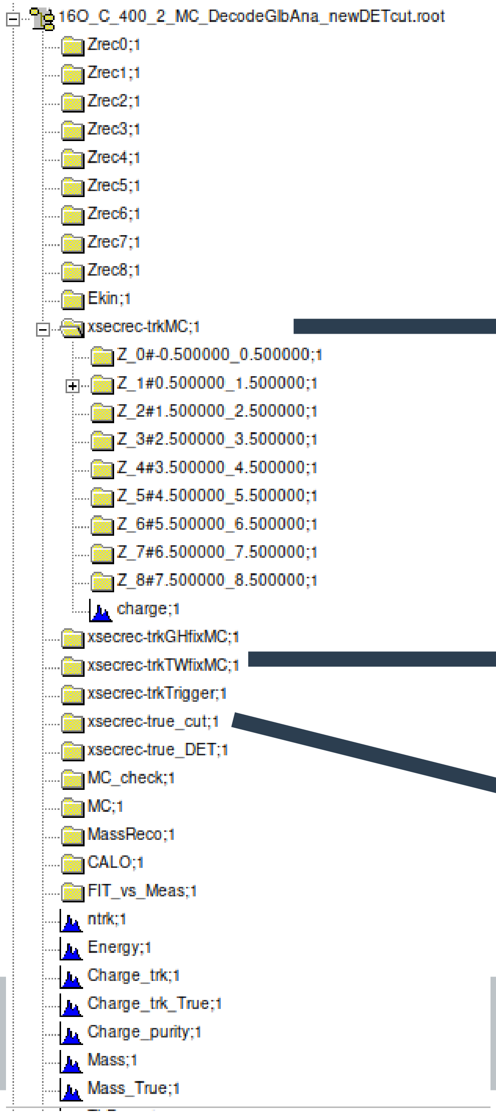
Plots are organized in subfolders of Z

For every Z, plots are organized in subsubfolders of theta

Events distribution in terms of theta

SHOE dataset results

ROOT dataset
From SHOE



```
16O_C_400_2_MC_DecodeGlbAna_newDETcut.root
  Zrec0;1
  Zrec1;1
  Zrec2;1
  Zrec3;1
  Zrec4;1
  Zrec5;1
  Zrec6;1
  Zrec7;1
  Zrec8;1
  Ekin;1
  xsecrec-trkMC;1
    Z_0#-0.500000_0.500000;1
    Z_1#0.500000_1.500000;1
    Z_2#1.500000_2.500000;1
    Z_3#2.500000_3.500000;1
    Z_4#3.500000_4.500000;1
    Z_5#4.500000_5.500000;1
    Z_6#5.500000_6.500000;1
    Z_7#6.500000_7.500000;1
    Z_8#7.500000_8.500000;1
    charge;1
  xsecrec-trkGHfixMC;1
  xsecrec-trkTWfixMC;1
  xsecrec-trkTrigger;1
  xsecrec-true_cut;1
  xsecrec-true_DET;1
  MC_check;1
  MC;1
  MassReco;1
  CALO;1
  FIT_vs_Meas;1
  ntrk;1
  Energy;1
  Charge_trk;1
  Charge_trk_True;1
  Charge_purity;1
  Mass;1
  Mass_True;1
```

- **These are the folders of yield I am using in my analysis:**

ALL RECO DATA Yield

RECO DATA Yield w/ TW AND TRACK Filter

MC gen DATA Yield

Analysis procedure

To compute elemental cross section and angular differential cross section:

$$\boxed{\sigma(Z) = \frac{Y(Z) - B(Z)}{N_{beam}\, N_{target}\, \epsilon(Z)}}$$

$$\boxed{\frac{d\sigma}{d\theta}(Z,\theta) = \frac{Y(Z,\theta) - B(Z,\theta)}{N_{beam}\, N_{target}\, \Omega_\theta\, \epsilon(Z,\theta)}}$$

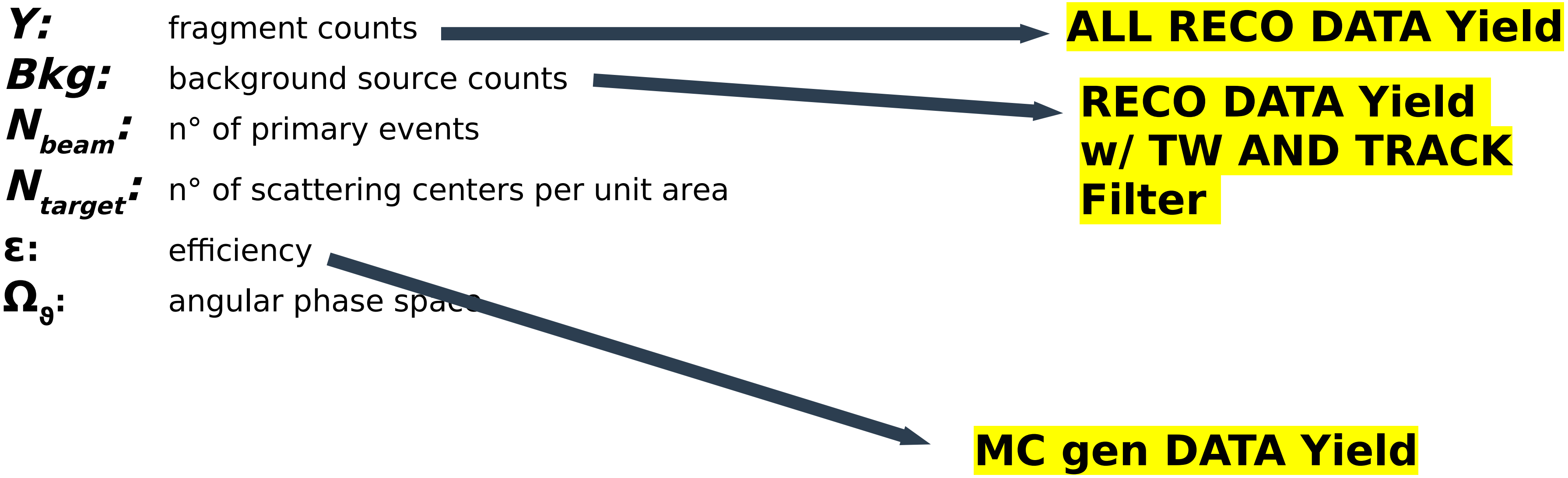
Y:	fragment counts
Bkg:	background source counts
N_{beam}:	n° of primary events
N_{target}:	n° of scattering centers per unit area
ε:	efficiency
Ω_ϑ:	angular phase space

Analysis procedure

To compute elemental cross section and angular differential cross section:

$$\sigma(Z) = \frac{Y(Z) - B(Z)}{N_{beam}\, N_{target}\, \epsilon(Z)}$$

$$\frac{d\sigma}{d\theta}(Z,\theta) = \frac{Y(Z,\theta) - B(Z,\theta)}{N_{beam}\, N_{target}\, \Omega_\theta\, \epsilon(Z,\theta)}$$



Y: fragment counts → **ALL RECO DATA Yield**

Bkg: background source counts → **RECO DATA Yield w/ TW AND TRACK Filter**

N_{beam}: n° of primary events

N_{target}: n° of scattering centers per unit area

ε: efficiency → **MC gen DATA Yield**

Ω_ϑ: angular phase space

Analysis procedure

$$\sigma(Z) = \frac{Y(Z) - B(Z)}{N_{beam}\, N_{target}\, \epsilon(Z)}$$

1) Starting from a **MC dataset** of 10^6 events generated by FLUKA to simulate detectors and beams of GSI 2021 campaign.

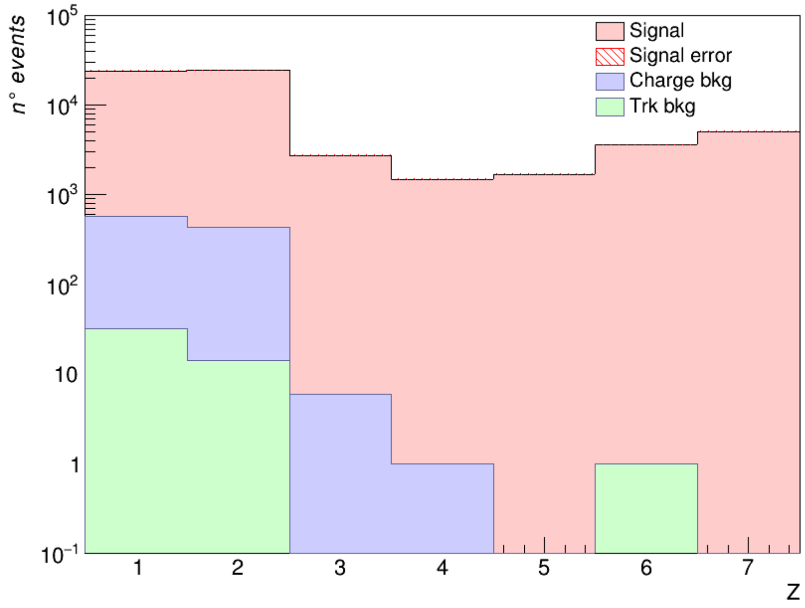
2) **Yield of Z** obtained from reconstructed tracks

- Exploiting **charge** reconstruction algorithm
- Exploiting **tracking** reconstruction algorithm
- Simulating a **"trigger"** in order to consider only fragments

3) **Background** obtained from MC cuts on:

- **Charge** algorithm mis-reconstruction
- **Tracking** algorithm mis-reconstruction

Z yield and Bkg sources



ALL RECO DATA Yield

Ratio between

ALL RECO DATA Yield

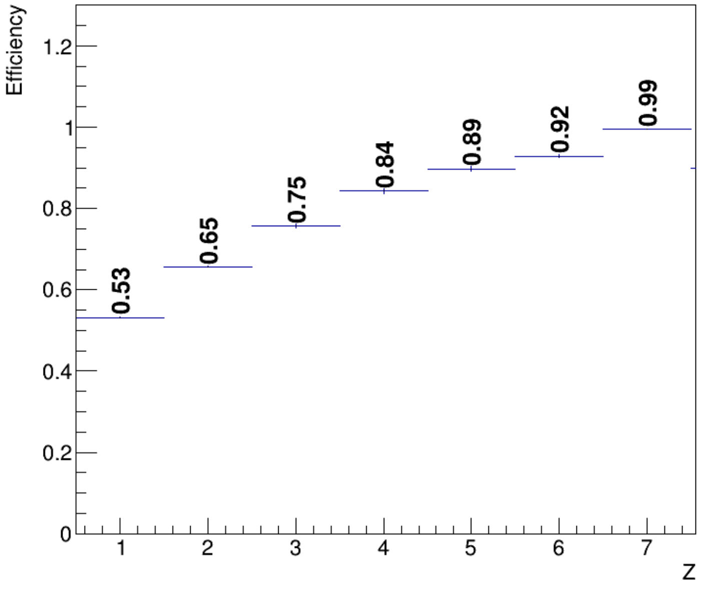
RECO DATA Yield w/ TW AND TRACK Filter

Analysis procedure

$$\sigma(Z) = \frac{Y(Z) - B(Z)}{N_{beam}\, N_{target}\, \epsilon(Z)}$$

4) Track efficiency obtained as:

$$\boxed{\epsilon(Z) = \frac{N_{track}(Z)}{N_{true}(Z)}}$$



where

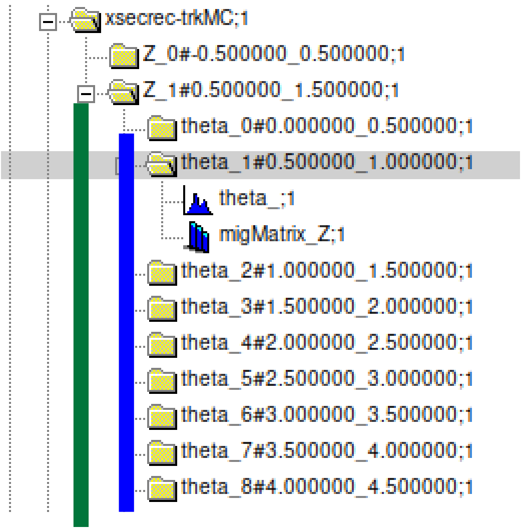
- N_{track} is obtained by tracking algorithm
- N_{true} are generated particles from the simulation with angular acceptance **θ≤8°**

Ratio between **ALL RECO DATA Yield** **MC gen DATA Yield**

First approximation logic

ROOT dataset
From SHOE

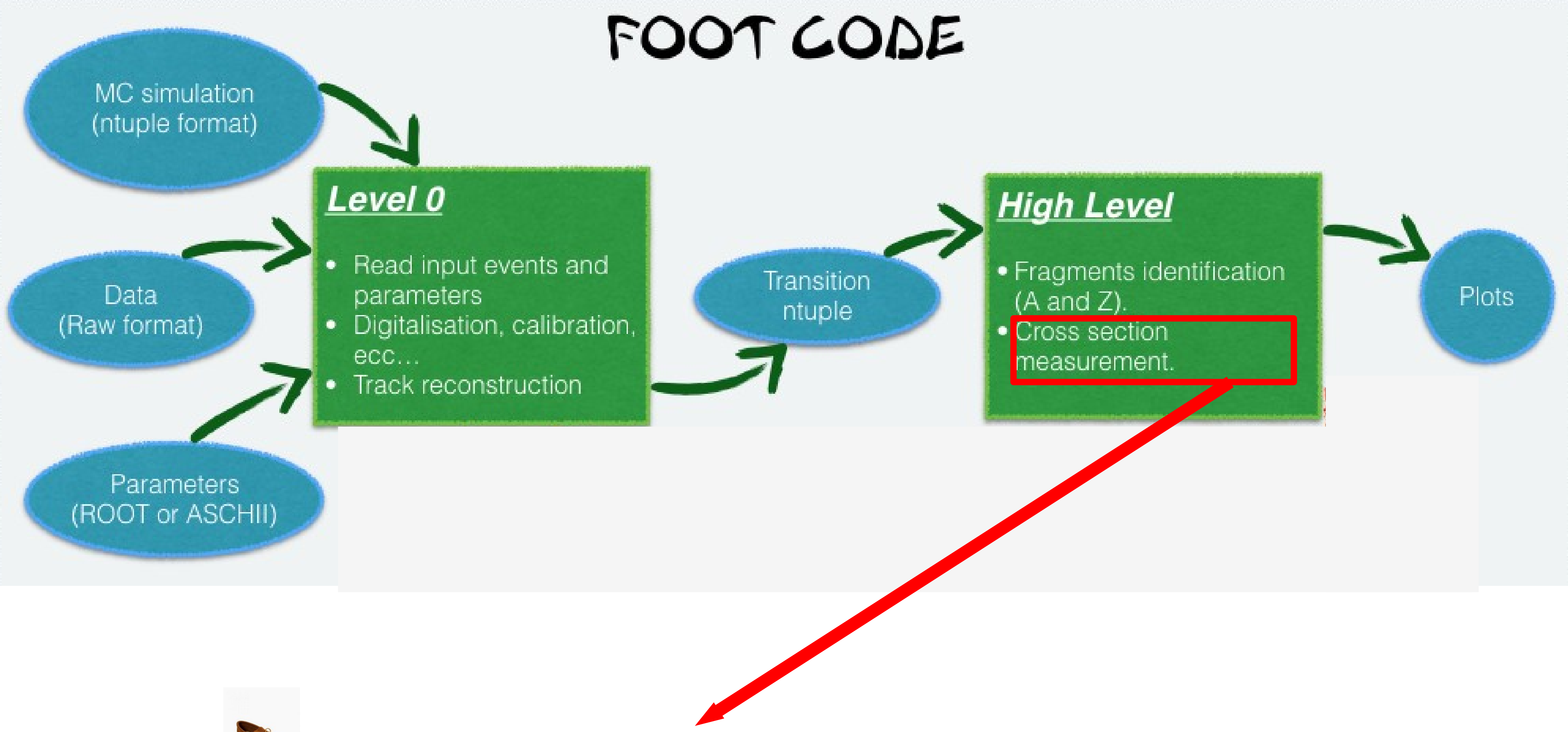
- ROOT file after DecodeGlbAna.cxx execution
  Example from ROOT TBrowser:



```
xsecrec-trkMC;1
    Z_0#-0.500000_0.500000;1
    Z_1#0.500000_1.500000;1
        theta_0#0.000000_0.500000;1
        theta_1#0.500000_1.000000;1
            theta_;1
            migMatrix_Z;1
        theta_2#1.000000_1.500000;1
        theta_3#1.500000_2.000000;1
        theta_4#2.000000_2.500000;1
        theta_5#2.500000_3.000000;1
        theta_6#3.000000_3.500000;1
        theta_7#3.500000_4.000000;1
        theta_8#4.000000_4.500000;1
```

I need to sum events in histograms
According to the cross section
I want to compute
→ MOCCASIN

SHOE organization



Level 0

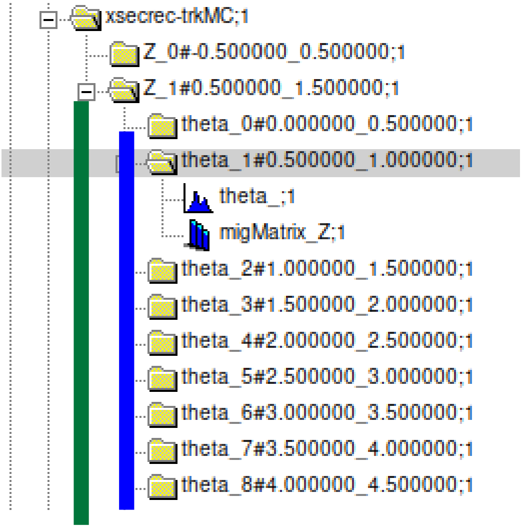
- Read input events and parameters
- Digitalisation, calibration, ecc…
- Track reconstruction

High Level

- Fragments identification (A and Z).
- Cross section measurement.

MOCCASIN
Measurements **O**f **C**ross-sections **C**omputed **A**nalyzing **S**HOE **I**nput **N**tuples

MOCCASIN – first step



```
xsecrec-trkMC;1
  Z_0#-0.500000_0.500000;1
  Z_1#0.500000_1.500000;1
    theta_0#0.000000_0.500000;1
    theta_1#0.500000_1.000000;1
      theta_;1
      migMatrix_Z;1
    theta_2#1.000000_1.500000;1
    theta_3#1.500000_2.000000;1
    theta_4#2.000000_2.500000;1
    theta_5#2.500000_3.000000;1
    theta_6#3.000000_3.500000;1
    theta_7#3.500000_4.000000;1
    theta_8#4.000000_4.500000;1
```

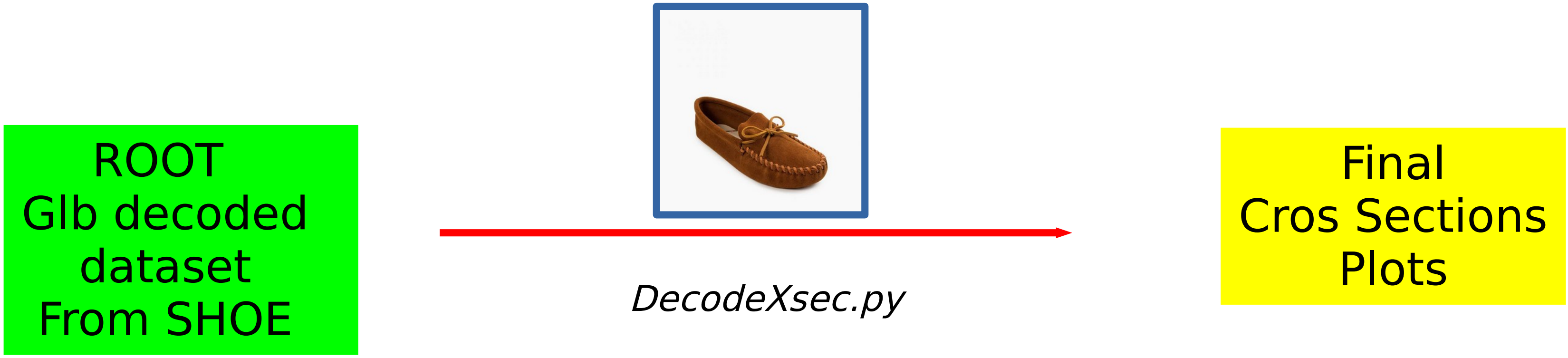
- The first goal is to recollect the specific plots of events according to requested variables.

- F.e. obtaining a distribution in Z
  of events already distributed in theta for every Z

- Specific theta_ plots are then summed and converted according to the requested variables for final cross section measurement

MOCCASIN logic

- Python code
- https://baltig.infn.it/gubaldi/moccasin



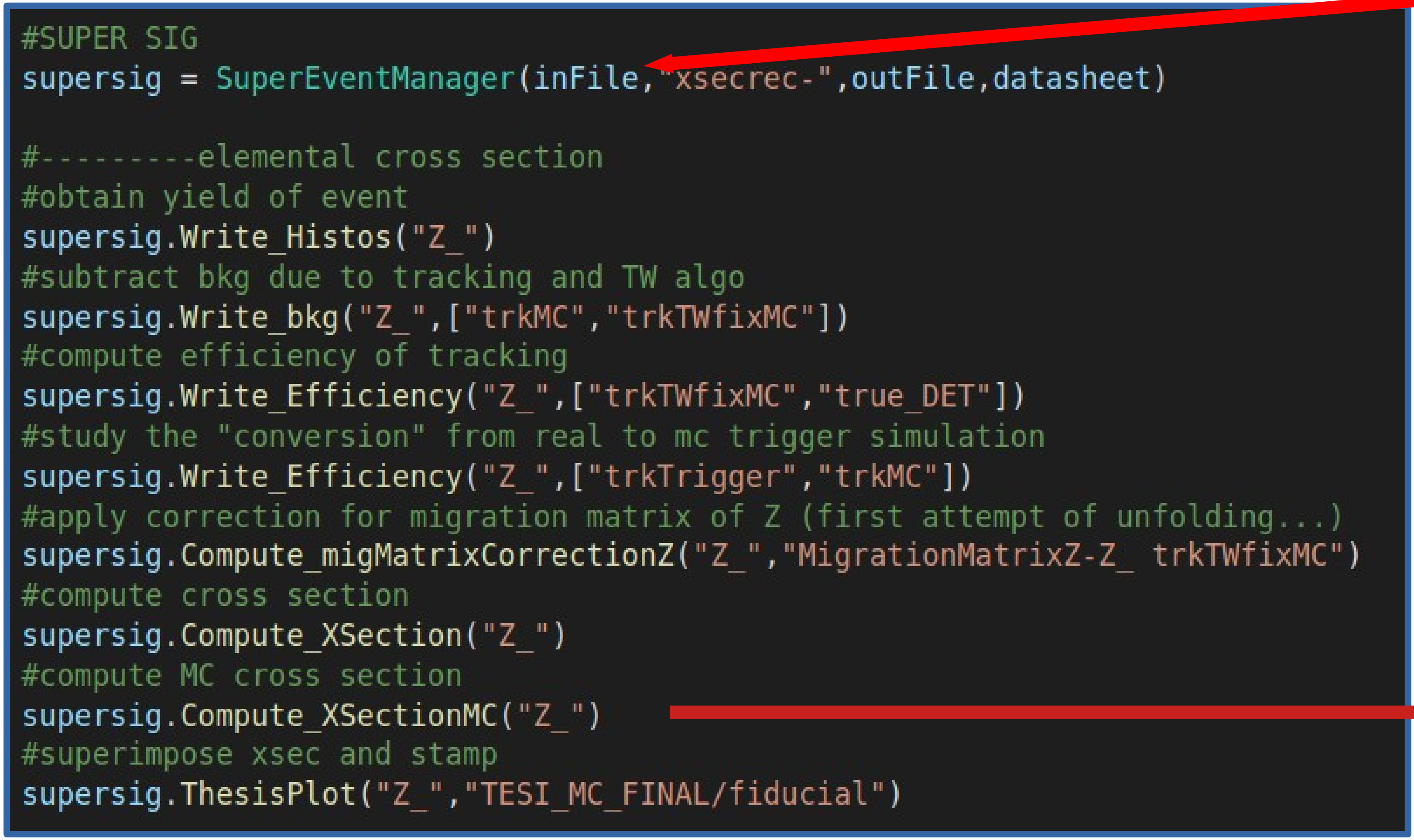
ROOT
Glb decoded
dataset
From SHOE

DecodeXsec.py

Final
Cros Sections
Plots

MOCCASIN logic

DecodeXsec.py



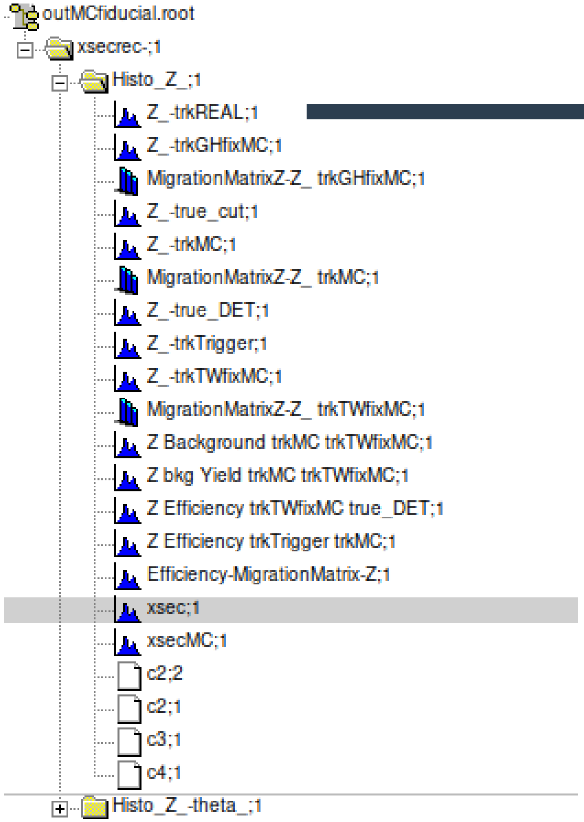
```
#SUPER SIG
supersig = SuperEventManager(inFile,"xsecrec-",outFile,datasheet)

#---------elemental cross section
#obtain yield of event
supersig.Write_Histos("Z_")
#subtract bkg due to tracking and TW algo
supersig.Write_bkg("Z_",["trkMC","trkTWfixMC"])
#compute efficiency of tracking
supersig.Write_Efficiency("Z_",["trkTWfixMC","true_DET"])
#study the "conversion" from real to mc trigger simulation
supersig.Write_Efficiency("Z_",["trkTrigger","trkMC"])
#apply correction for migration matrix of Z (first attempt of unfolding...)
supersig.Compute_migMatrixCorrectionZ("Z_","MigrationMatrixZ-Z_ trkTWfixMC")
#compute cross section
supersig.Compute_XSection("Z_")
#compute MC cross section
supersig.Compute_XSectionMC("Z_")
#superimpose xsec and stamp
supersig.ThesisPlot("Z_","TESI_MC_FINAL/fiducial")
```

ROOT dataset
From SHOE

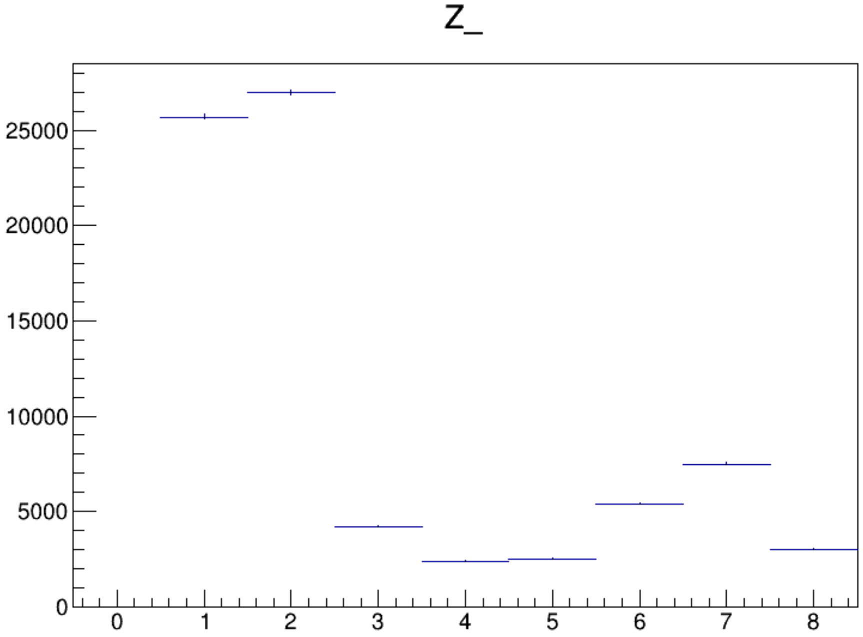
Final
Cros Sections
Plots

MOCCASIN logic

MOCCASIN output



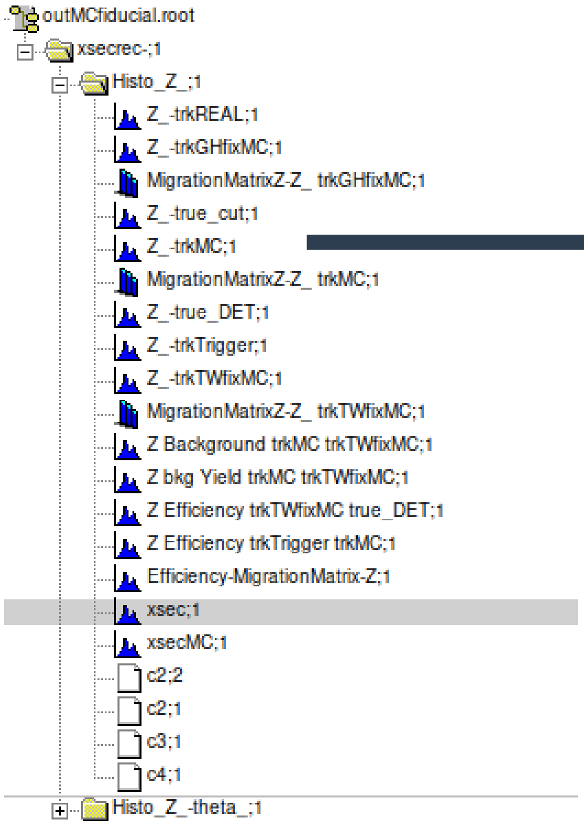
```
outMCfiducial.root
  xsecrec-;1
    Histo_Z_;1
      Z_-trkREAL;1
      Z_-trkGHfixMC;1
      MigrationMatrixZ-Z_ trkGHfixMC;1
      Z_-true_cut;1
      Z_-trkMC;1
      MigrationMatrixZ-Z_ trkMC;1
      Z_-true_DET;1
      Z_-trkTrigger;1
      Z_-trkTWfixMC;1
      MigrationMatrixZ-Z_ trkTWfixMC;1
      Z Background trkMC trkTWfixMC;1
      Z bkg Yield trkMC trkTWfixMC;1
      Z Efficiency trkTWfixMC true_DET;1
      Z Efficiency trkTrigger trkMC;1
      Efficiency-MigrationMatrix-Z;1
      xsec;1
      xsecMC;1
      c2;2
      c2;1
      c3;1
      c4;1
    Histo_Z_-theta_;1
```



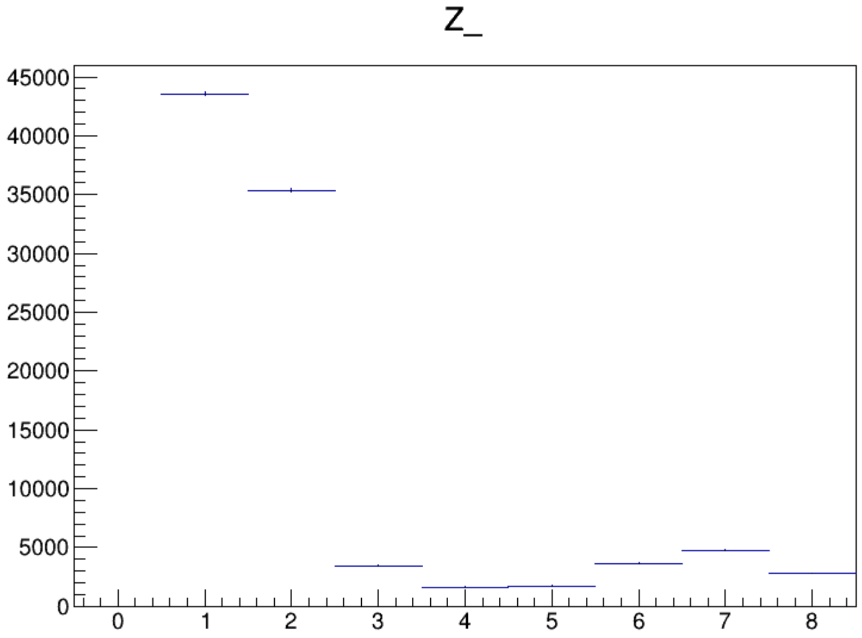
from **ALL RECO DATA Yield**

MOCCASIN logic

MOCCASIN output



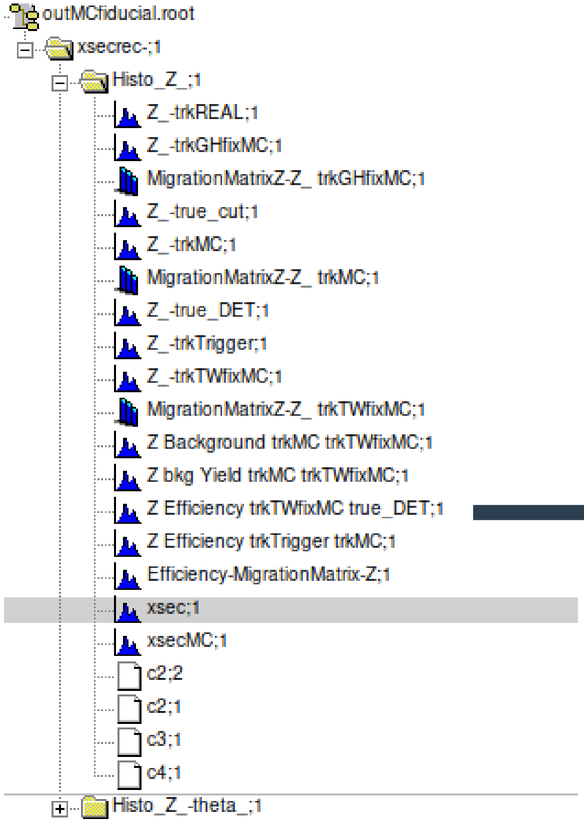
```
outMCfiducial.root
  xsecrec-;1
    Histo_Z_;1
      Z_-trkREAL;1
      Z_-trkGHfixMC;1
      MigrationMatrixZ-Z_ trkGHfixMC;1
      Z_-true_cut;1
      Z_-trkMC;1
      MigrationMatrixZ-Z_ trkMC;1
      Z_-true_DET;1
      Z_-trkTrigger;1
      Z_-trkTWfixMC;1
      MigrationMatrixZ-Z_ trkTWfixMC;1
      Z Background trkMC trkTWfixMC;1
      Z bkg Yield trkMC trkTWfixMC;1
      Z Efficiency trkTWfixMC true_DET;1
      Z Efficiency trkTrigger trkMC;1
      Efficiency-MigrationMatrix-Z;1
      xsec;1
      xsecMC;1
      c2;2
      c2;1
      c3;1
      c4;1
    Histo_Z_-theta_;1
```



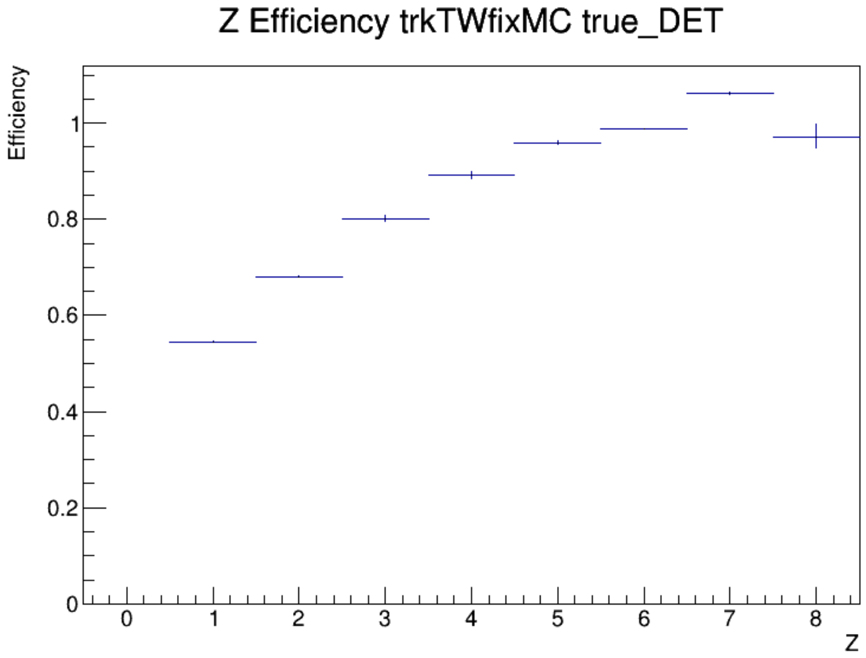
from **MC gen DATA Yield**

MOCCASIN logic

MOCCASIN output



```
outMCfiducial.root
  xsecrec-;1
    Histo_Z_;1
      Z_-trkREAL;1
      Z_-trkGHfixMC;1
      MigrationMatrixZ-Z_ trkGHfixMC;1
      Z_-true_cut;1
      Z_-trkMC;1
      MigrationMatrixZ-Z_ trkMC;1
      Z_-true_DET;1
      Z_-trkTrigger;1
      Z_-trkTWfixMC;1
      MigrationMatrixZ-Z_ trkTWfixMC;1
      Z Background trkMC trkTWfixMC;1
      Z bkg Yield trkMC trkTWfixMC;1
      Z Efficiency trkTWfixMC true_DET;1
      Z Efficiency trkTrigger trkMC;1
      Efficiency-MigrationMatrix-Z;1
      xsec;1
      xsecMC;1
      c2;2
      c2;1
      c3;1
      c4;1
    Histo_Z_-theta_;1
```



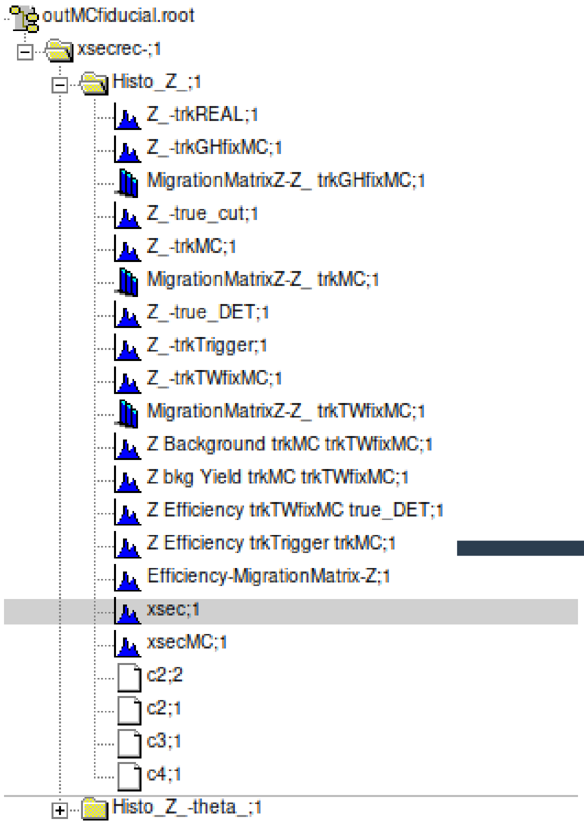
from
Ratio between
For bkg

ALL RECO DATA Yield

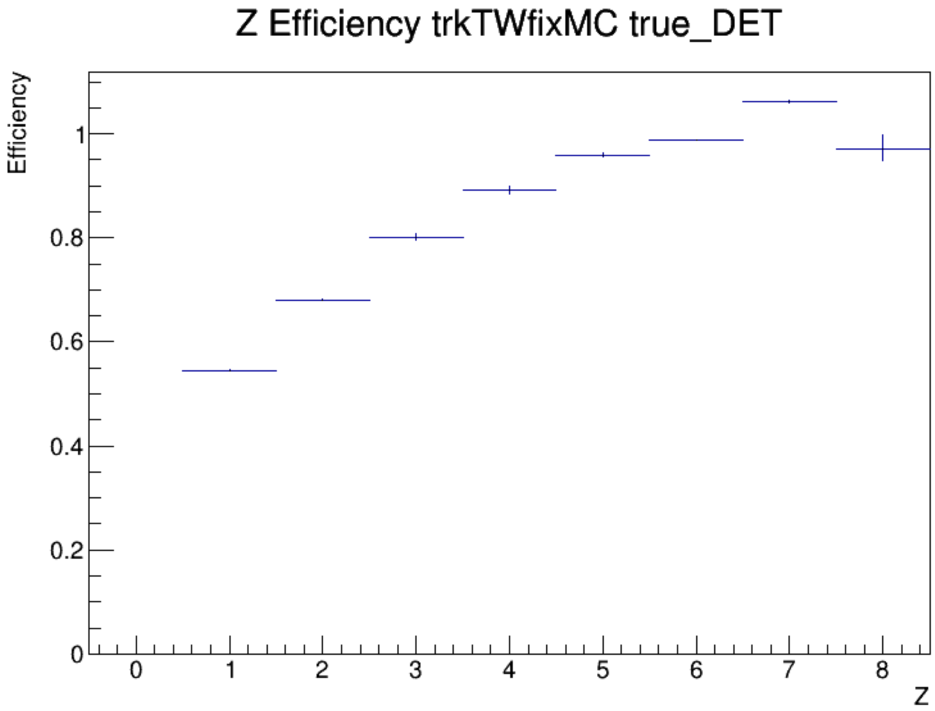
RECO DATA Yield
w/ TW AND TRACK
Filter

MOCCASIN logic

MOCCASIN output



```
outMCfiducial.root
  xsecrec-;1
    Histo_Z_;1
      Z_-trkREAL;1
      Z_-trkGHfixMC;1
      MigrationMatrixZ-Z_ trkGHfixMC;1
      Z_-true_cut;1
      Z_-trkMC;1
      MigrationMatrixZ-Z_ trkMC;1
      Z_-true_DET;1
      Z_-trkTrigger;1
      Z_-trkTWfixMC;1
      MigrationMatrixZ-Z_ trkTWfixMC;1
      Z Background trkMC trkTWfixMC;1
      Z bkg Yield trkMC trkTWfixMC;1
      Z Efficiency trkTWfixMC true_DET;1
      Z Efficiency trkTrigger trkMC;1
      Efficiency-MigrationMatrix-Z;1
      xsec;1
      xsecMC;1
      c2;2
      c2;1
      c3;1
      c4;1
    Histo_Z_-theta_;1
```



from

Ratio between
For efficiency

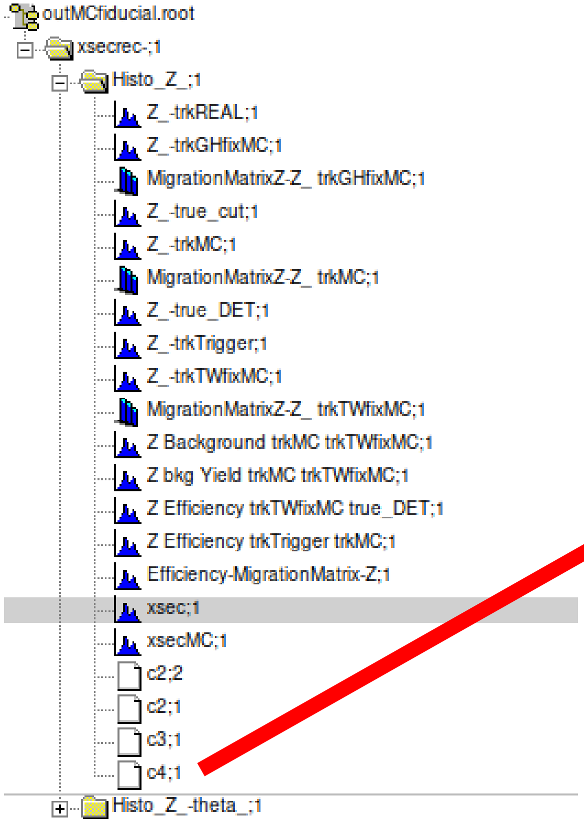
ALL RECO DATA Yield

MC gen DATA Yield

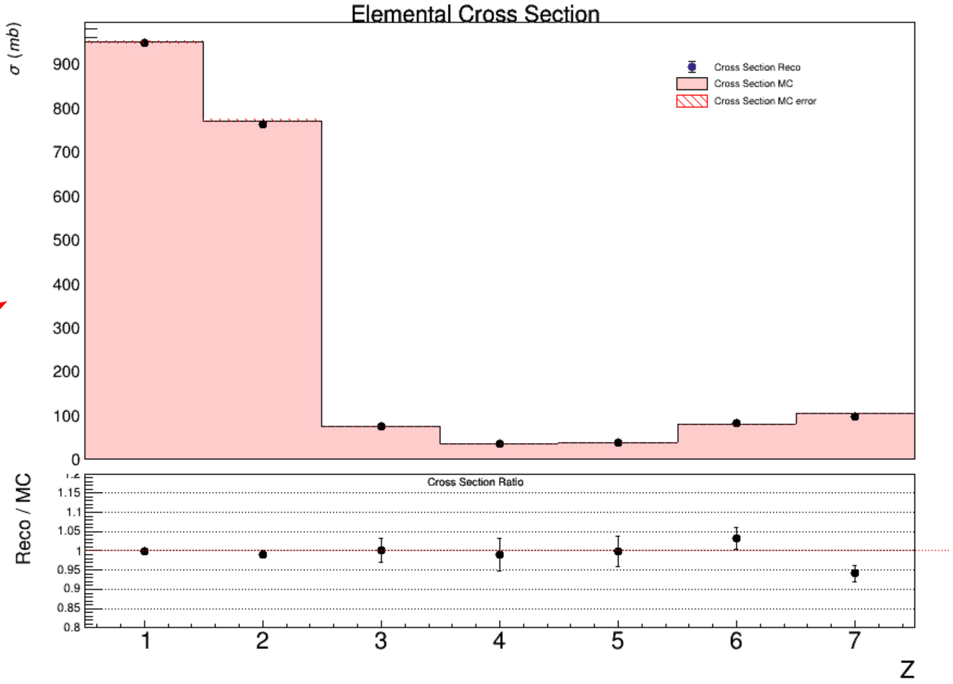
MOCCASIN logic

MOCCASIN output

Putting togheter all plots...



```
outMCfiducial.root
  xsecrec-;1
    Histo_Z_;1
      Z_-trkREAL;1
      Z_-trkGHfixMC;1
      MigrationMatrixZ-Z_ trkGHfixMC;1
      Z_-true_cut;1
      Z_-trkMC;1
      MigrationMatrixZ-Z_ trkMC;1
      Z_-true_DET;1
      Z_-trkTrigger;1
      Z_-trkTWfixMC;1
      MigrationMatrixZ-Z_ trkTWfixMC;1
      Z Background trkMC trkTWfixMC;1
      Z bkg Yield trkMC trkTWfixMC;1
      Z Efficiency trkTWfixMC true_DET;1
      Z Efficiency trkTrigger trkMC;1
      Efficiency-MigrationMatrix-Z;1
      xsec;1
      xsecMC;1
      c2;2
      c2;1
      c3;1
      c4;1
    Histo_Z_-theta_;1
```

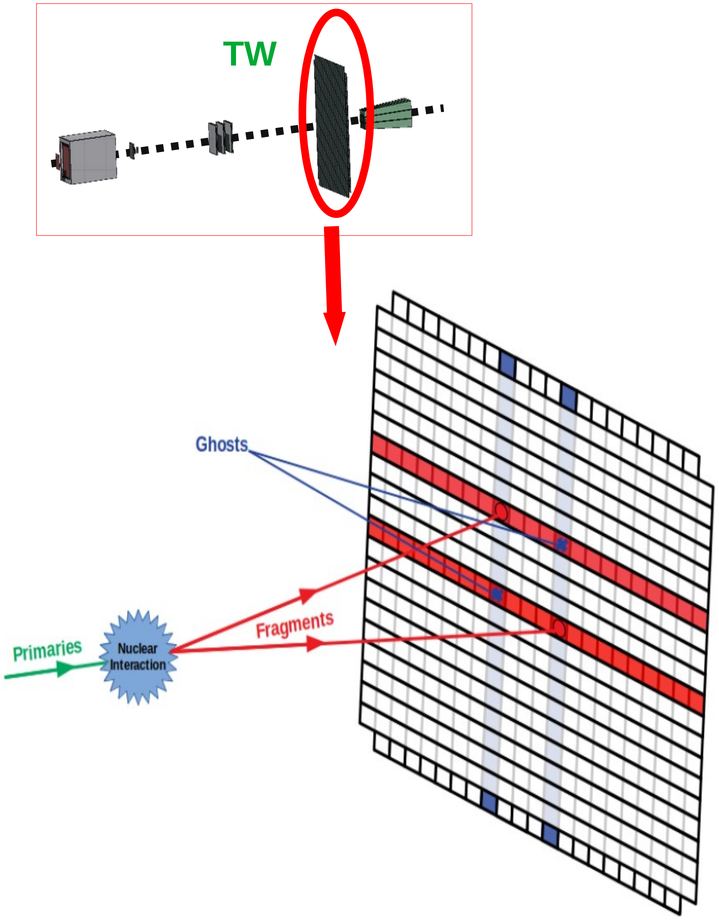


MOCCASIN logic

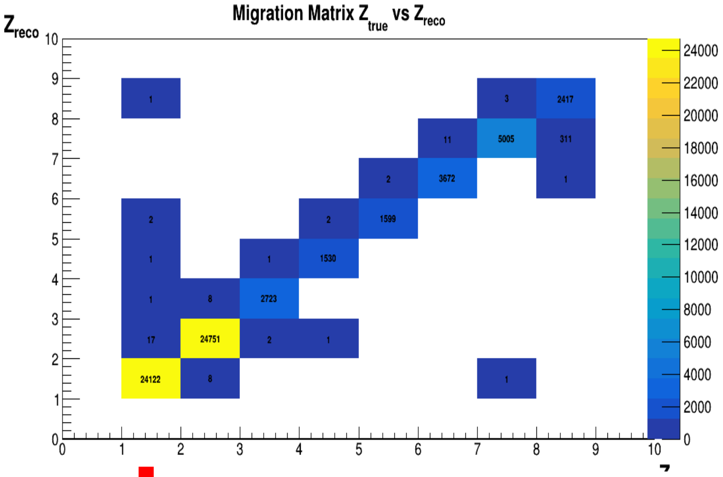
- **It is possible to add dinamically new efficiencies**
- **It is possible to add other subprocesses (es unfolding)**
-

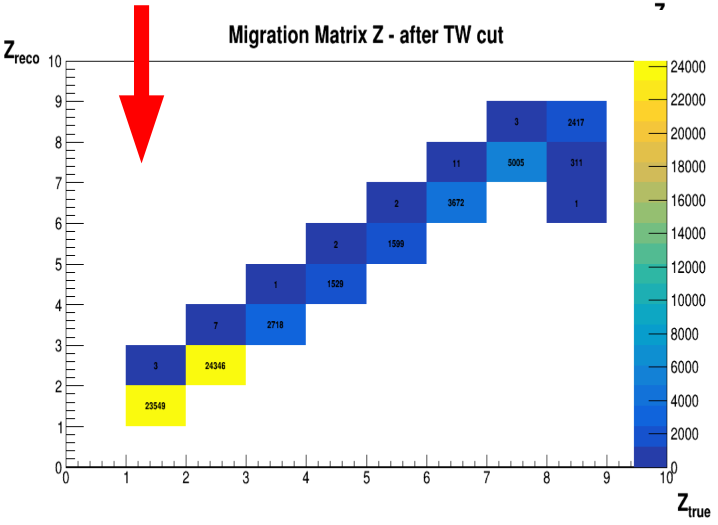
Example: Track reconstruction

It is possible that every bar layer of the TW is hit by more than a fragment at the same time:
multiple hits / ghost hits mis-reconstruction



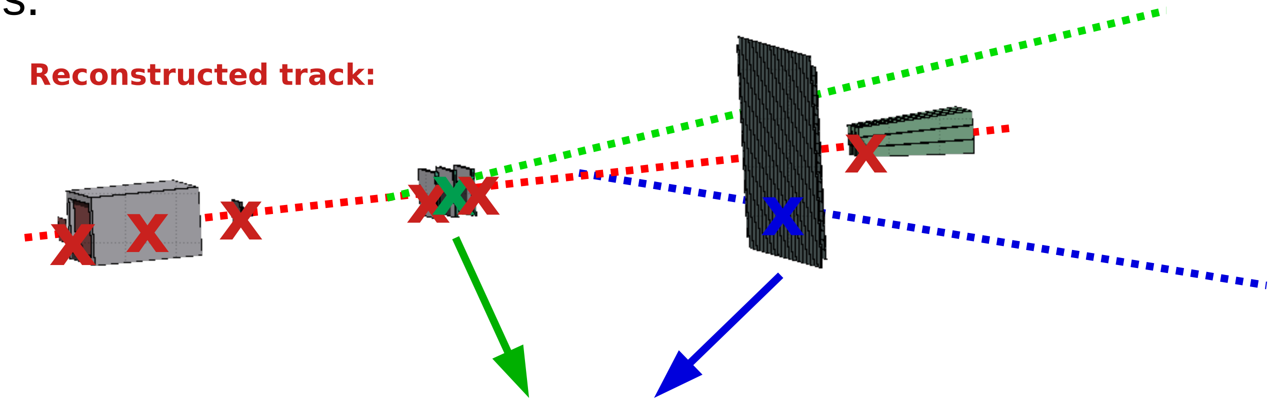
Applying TW cut:
Matrix more diagonal





Reconstruction, Track Algo

However, due to the presence of a lot of secondary fragmentation, some points can belong to other particles.



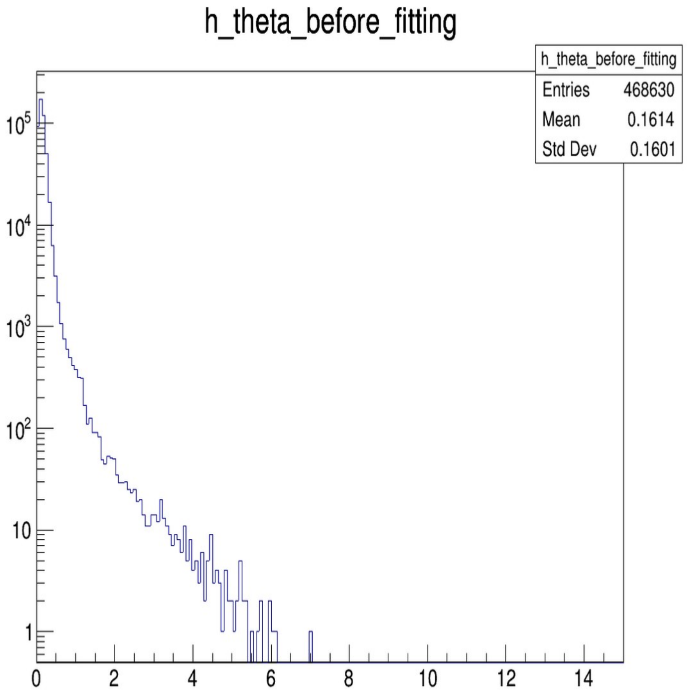
Wrong points collected in the track

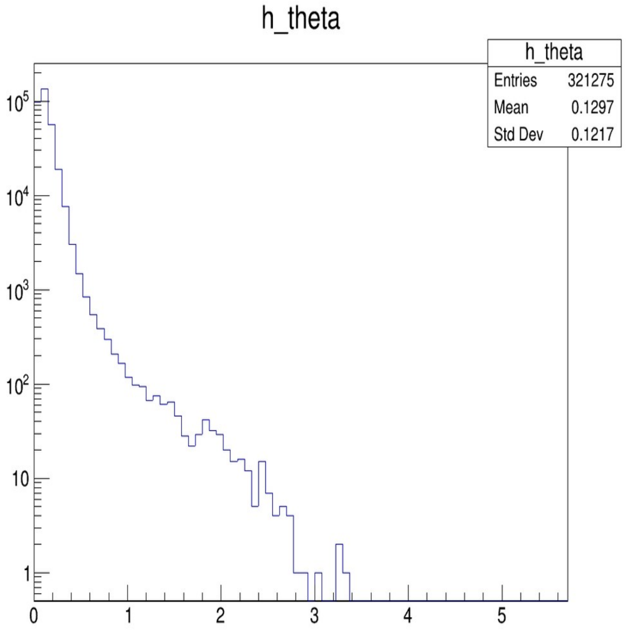
The McId of the track is given by the most present particle in the collection

However, if the TWPoint is of another particle → **its McId is different**

→ filter out all the tracks in which **$\mathrm{McId}_{\mathrm{track}} \neq \mathrm{McId}_{\mathrm{TWPoint}}$**

Track candidates



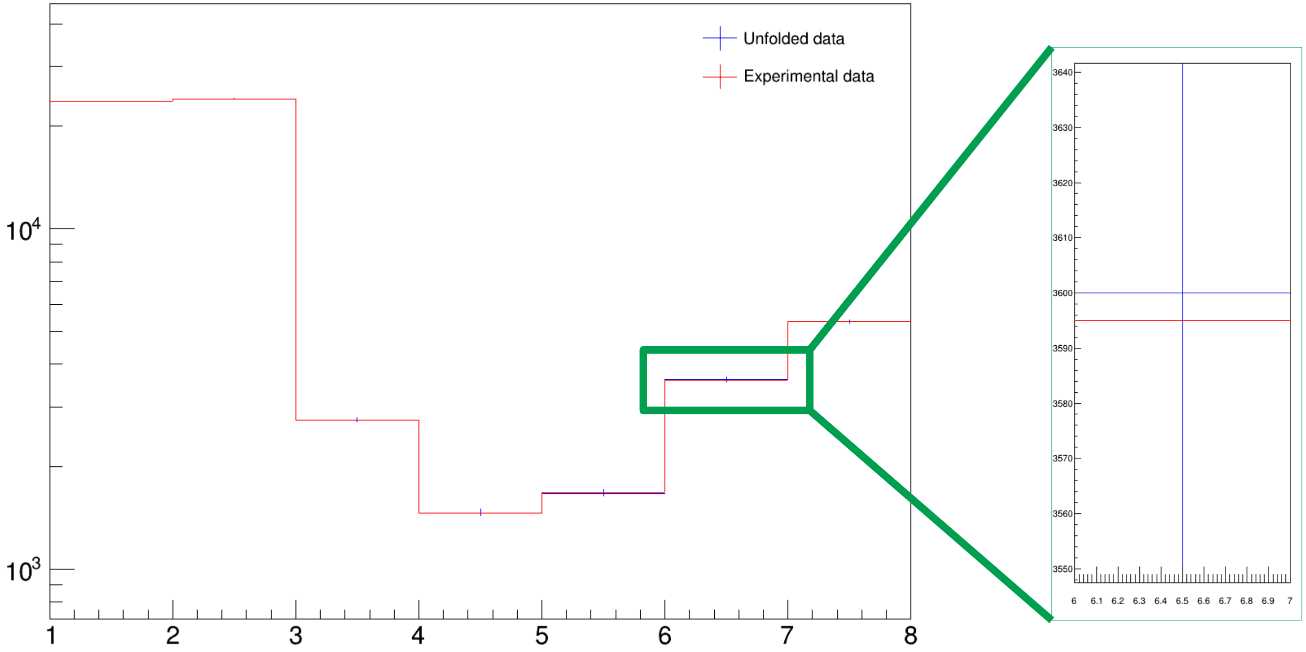


Possible vtx tracks → Selected tracks for glb reco

(No study of match between vtx track and tw point)

Implementation of Unfolding

$$\sigma(Z) = \frac{\left(Y(Z) - B(Z)\right)^{U}}{N_{beam}\, N_{target}\, \epsilon(Z)}$$

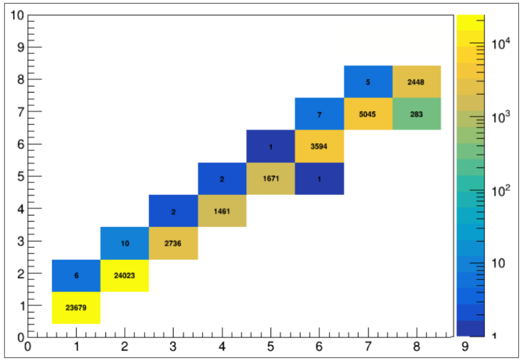


Thanks to Sofia C.

$$y_i = M_{ij}\, x_j$$

$$\longrightarrow \quad x_j = M_{ij}^{-1}\, y_i$$

- Little variation because the migration matrix is very diagonal


10
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9
23679
6
24023
10
2736
2
1461
2
1671
1
1
3594
7
5045
283
5
2448
10^4
10^3
10^2
10
1