glbreco_v1

A general overview

Yun & Ilaria (for the moment)

Instructions

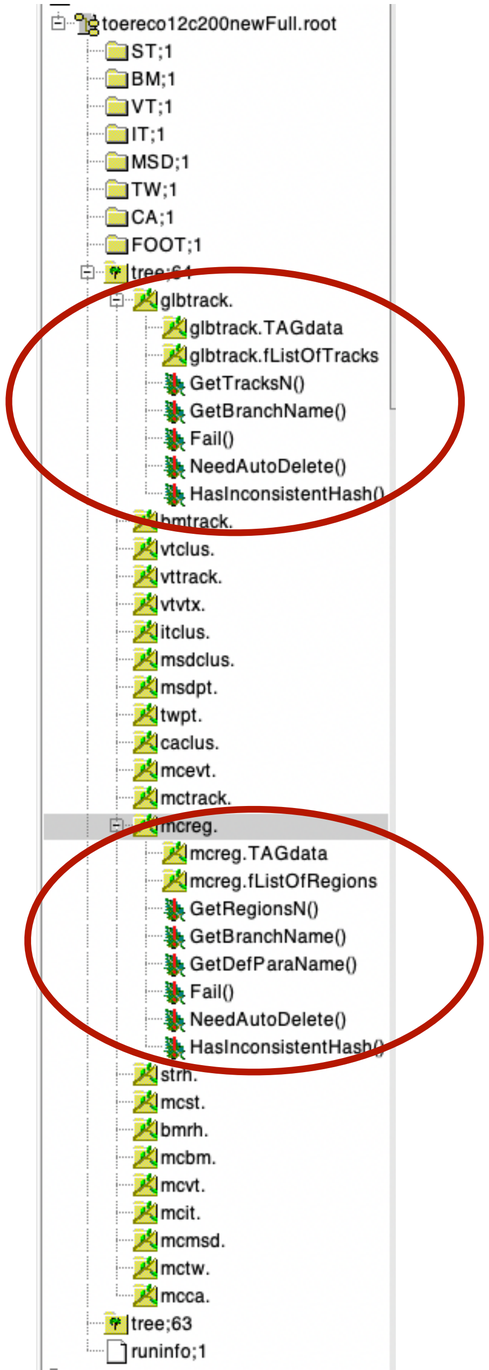
- Download the shoe code, checkout the **branch glbreco_v1** and do the usual build commands (cmake, source, make etc.)

- Get or create a root file with a ttree containing global tracks reconstructed from Genfit or Toe.
  If it is a MC study, the including of crossing regions is highly recommended

- To launch the global analysis from builddir/Reconstruction/fullrec:

  ../../bin/DecodeGlbAna -in toeorgenfitoutputfile.root -out outputfilename -exp 12C_200 -run 1 -mc
  (once we'll have the real data reconstruction, we'll omit the -mc flag)

- Enjoy the output file with cross sections, systematic uncertainty analysis and a first draft of a paper ready to be submitted… almost

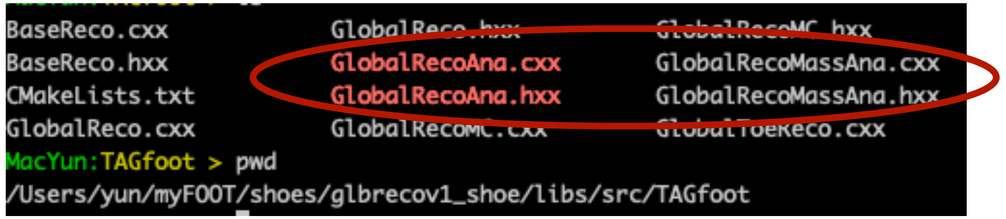


Overview

- The idea of a global analysis branch is to provide a tool with analysis methods and codes developed and verified by all the collaboration
- New analysis techniques can be easily added in the workflow.
- Once the cross section analysis has been done for a specific campaign, it can be easily adapted for all the other campaigns
- For the moment, all the analysis codes are in TAGfoot: GlobalRecoAna.cxx/hxx and GlobalRecoMassAna.cxx/hxx
- The goal is the differential cross section:

$$\frac{d\sigma_i}{d\Omega}(\theta) = \frac{Y_i(\theta)}{N_C \times N_{TG} \times \Delta\Omega \times \epsilon^i_{trk}(\theta)}$$

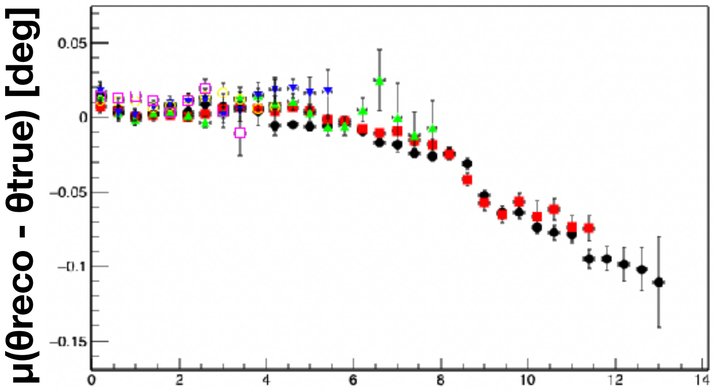
$$\frac{d\sigma_i}{dE_{kin}}(E_{kin}) = \frac{Y_i(E_{kin})}{N_C \times N_{TG} \times \Delta E_{kin} \times \epsilon^i_{trk}(E_{kin})}$$

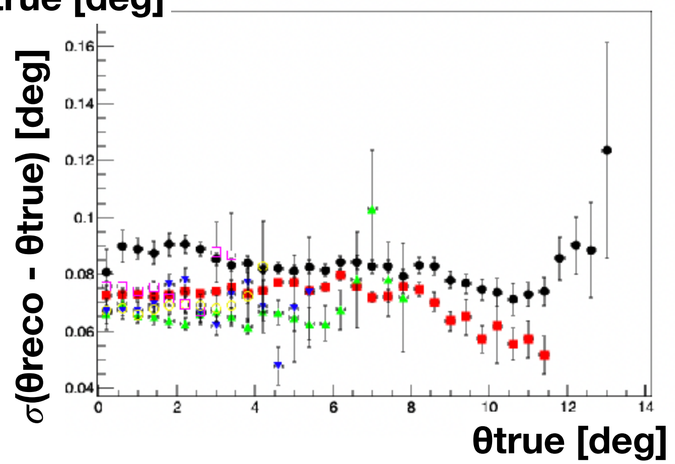


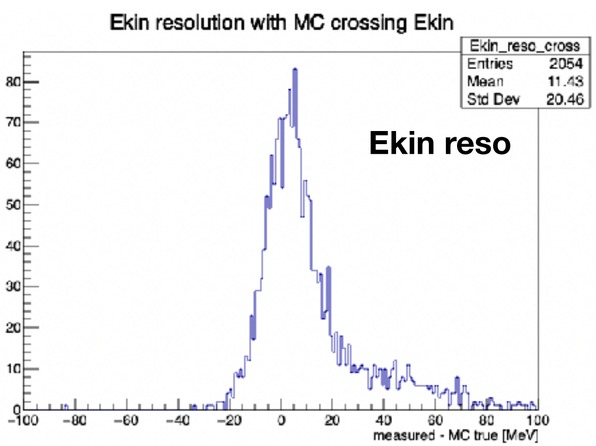
- **For this presentation: no vtx pileup, triggered simulation, TW match**

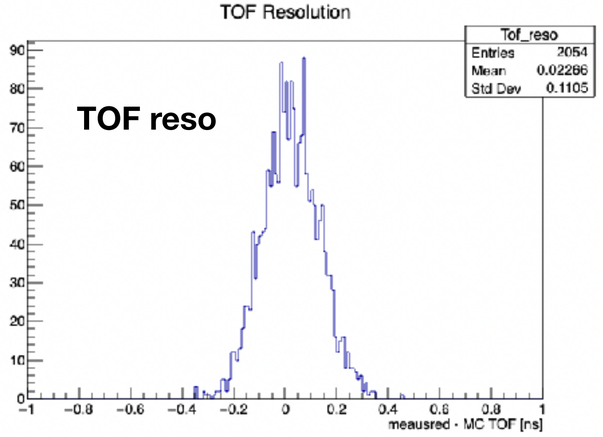
Resolution studies

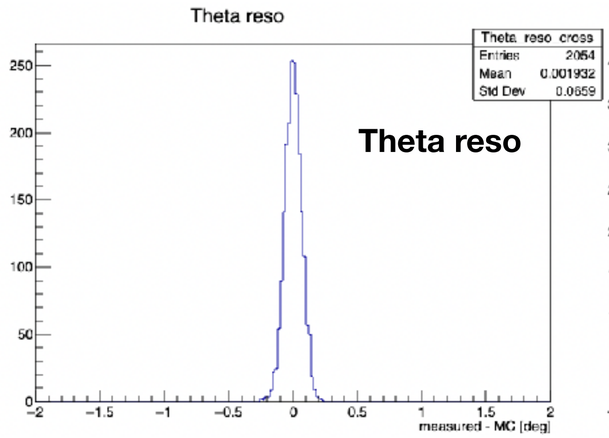
- Theta, Ekin, momentum and TOF resolution plots are calculated
- On the right the plots related to the **Z=3** reconstructed tracks are shown
- **Ekin calculation needs to be checked**

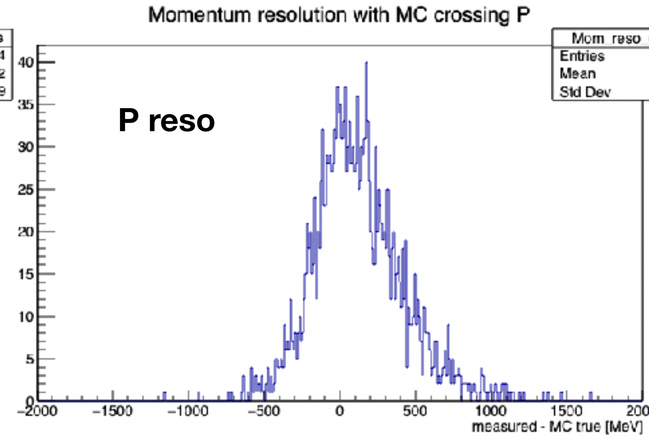






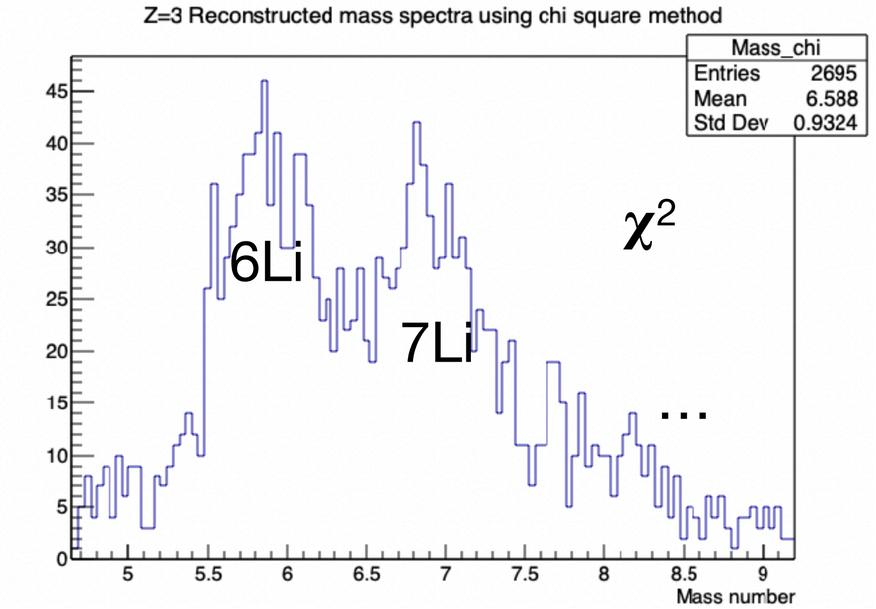


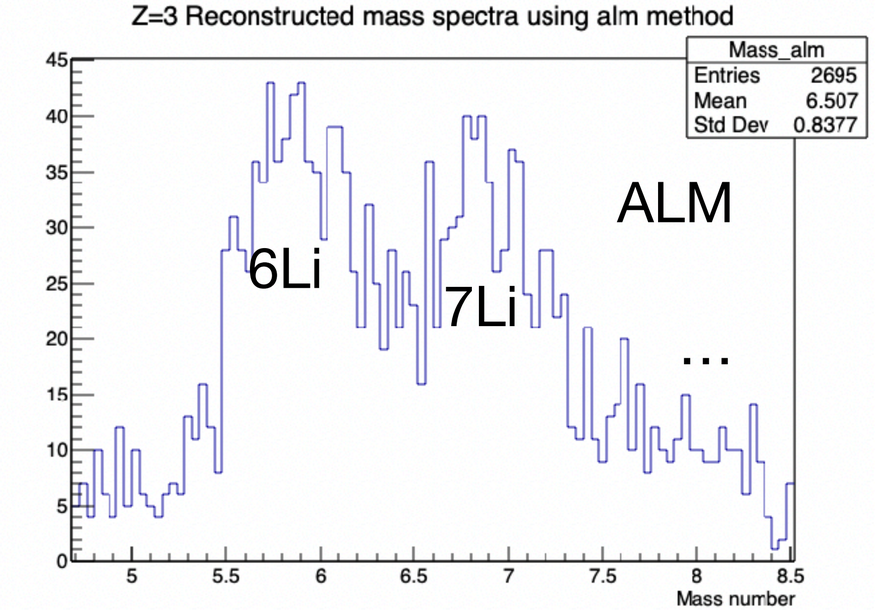




Mass reconstruction

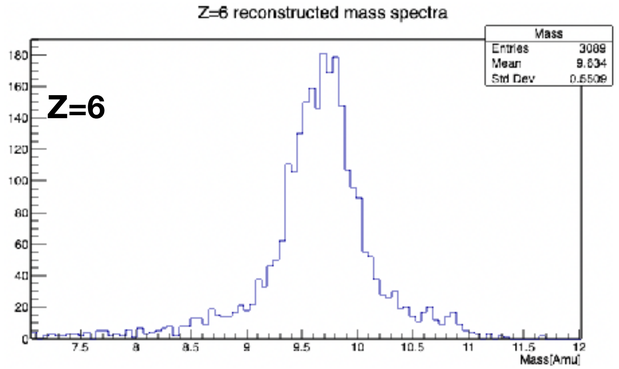
- The mass reconstruction methods are developed in a **dedicated class**: GlobalRecoMassAna.cxx/hxx
- The input parameters are the foot TOF, P, Ekin measurements with the associated uncertainty
- The class provides an estimate of the mass with P-Ekin, TOF-P and TOF-Ekin measurements and It combines them with the Chi2 and Alm methods
- As output one can have 6 mass estimates with the associated error and the estimate of the TOF, P and Ekin from the chi2 and Alm algorithms
- At the moment the Ekin measurement from the Calo has to be checked with dedicated studies (see “calo studies” of this presentation)
- The input parameters (TOF, P, Ekin, Errors) need to be checked
- **The alm and chi2 minimisation parameters need to be studied and optimized**
- A check of the codes for errors, bugs etc. is more than welcome

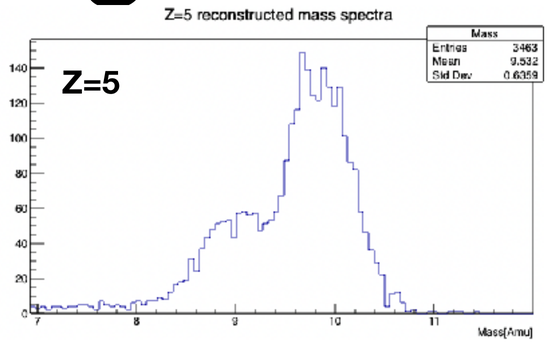


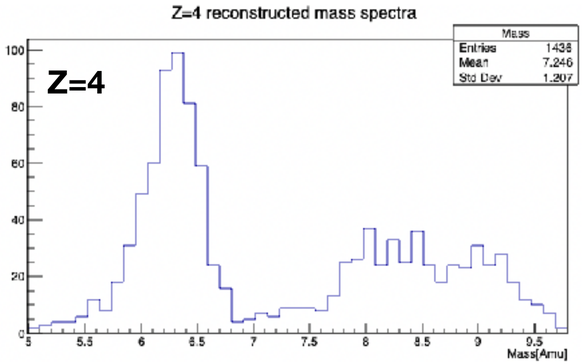


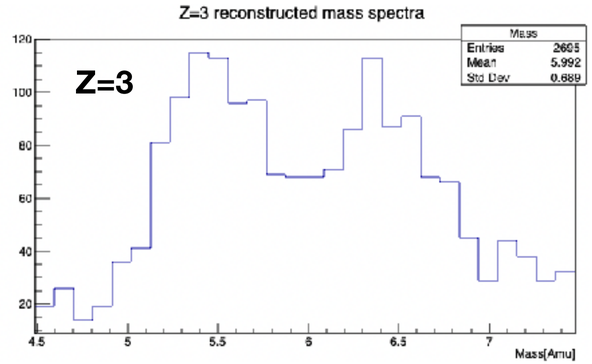
Mass spectra and combinatorial background

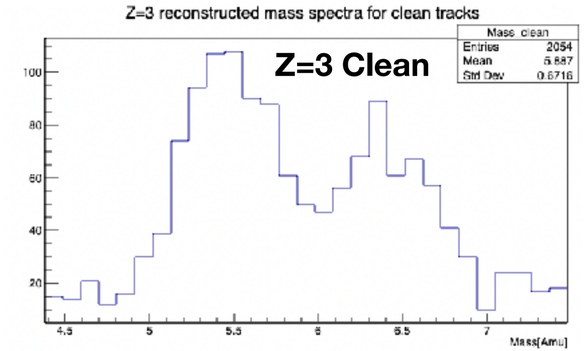
- The mass spectra for each Z are present
- Studies to evaluate the combinatorial background are ongoing
- Track purity seems to be very good
- The Z reconstruction precision is almost 100%, **need to add pileup and check the Z reconstruction actual performances**
- Need for statistics!

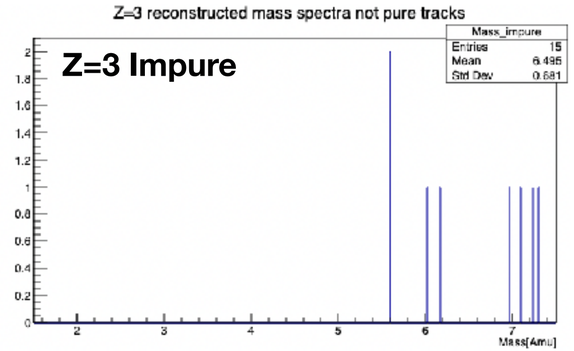






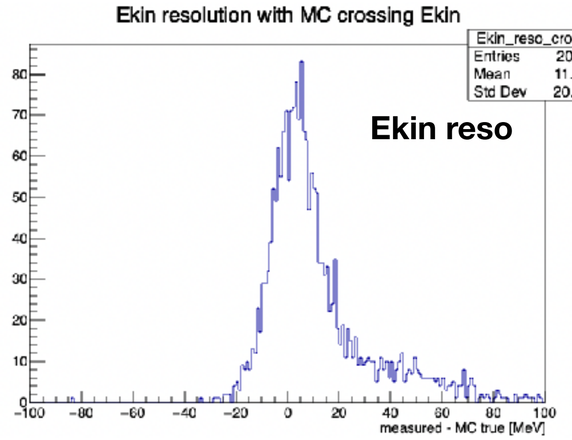


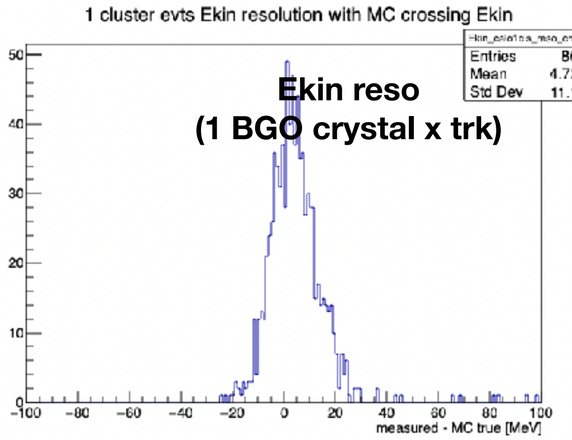


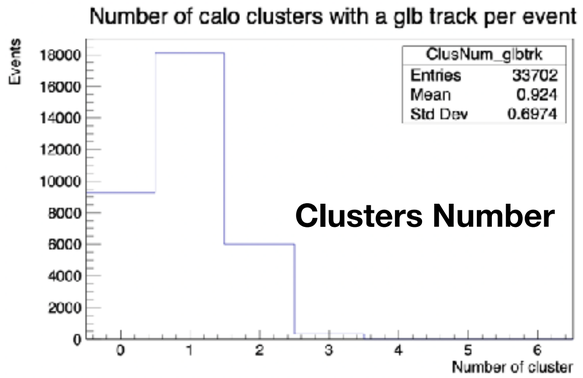


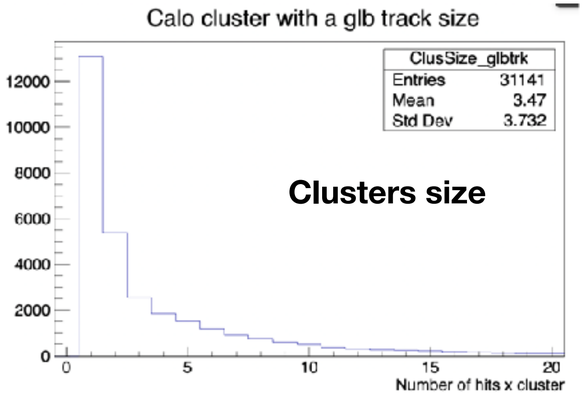
Calo studies

- At the moment the calorimeter Ekin energy measurement is higher than the MC true Ekin
- Considering only the tracks with 1 cluster, this effect is not present anymore
- The calo clustering is copy and paste from the VTX: it starts from a crystal with an over-threshold signal and it adds to the cluster all the adjacent crystals, iteratively
- Needs to verify if this clusterization algorithm is suitable for the calorimeter









To do list

- Ekin unfolding.
  The theta unfolding is not necessary since the matrix is almost diagonal (with this simulation, without vtx pileup etc.)

- Efficiency definition and studies with a full MC simulation

- Full MC simulation with statistics

- Check TOE and Genfit output

- Check in the code for errors, bugs etc and improve the readability of the code