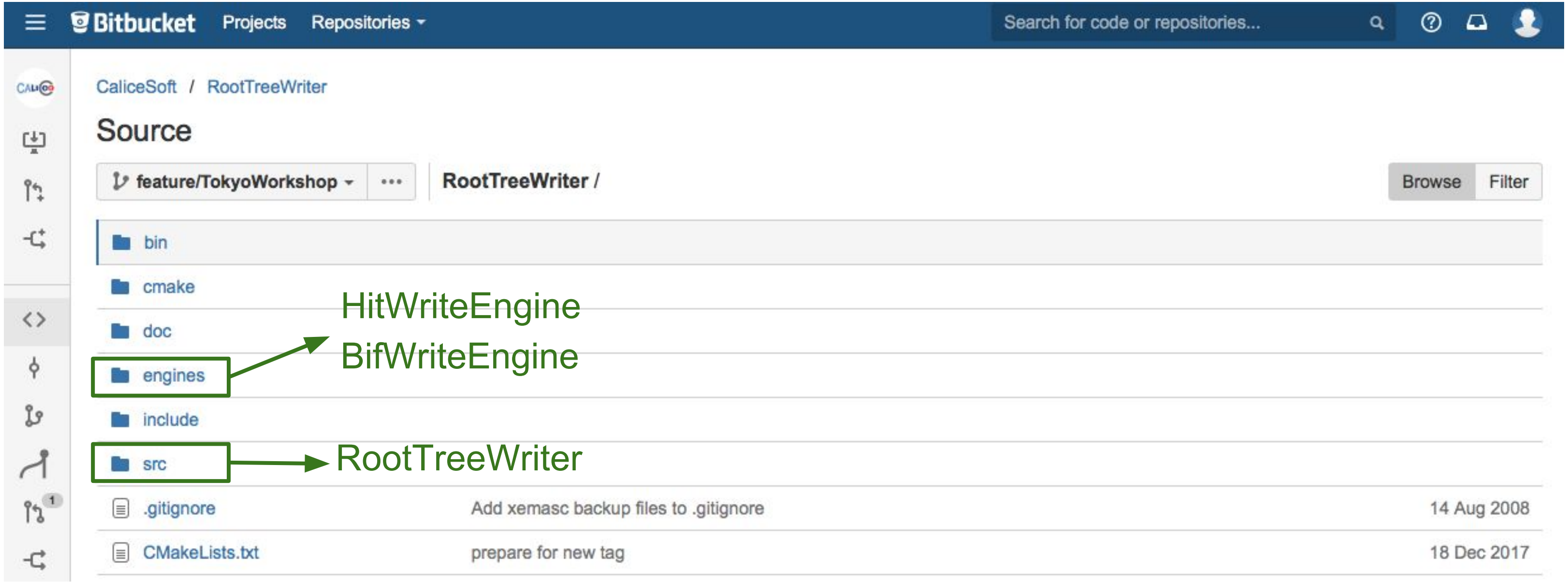
Status of the analysis root tree for the AHCAL testbeam data

Tokyo Analysis Workshop 2018

Saiva Huck

Overview

- Root tree with variables for reconstructed data
- Filled from CalorimeterHit objects (slcio) after reconstruction
- Used for data analysis



CaliceSoft / RootTreeWriter

Source

feature/TokyoWorkshop ··· RootTreeWriter /

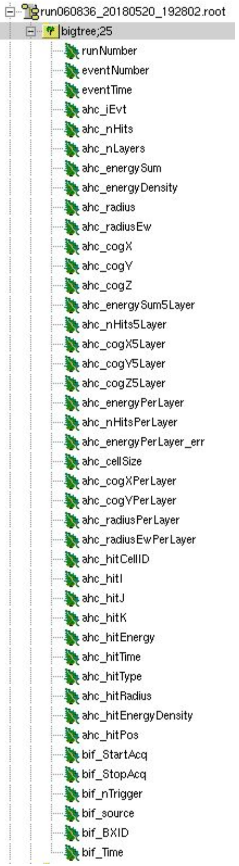
bin		
cmake		
doc		
engines		
include		
src		
.gitignore	Add xemasc backup files to .gitignore	14 Aug 2008
CMakeLists.txt	prepare for new tag	18 Dec 2017

engines → HitWriteEngine, BifWriteEngine

src → RootTreeWriter

Overview

- Root file: ***run***runnumber_date_time.root
- Name: “bigtree”
- 43 variables in total
- Each branch (each variable): info for all events
  - ⇒ Each event contains several objects
  - ⇒ An object can be one value or an array of information for each hit or layer
- Information available for every single hit in every event
  - ⇒ Information in arrays is important, not histograms
  - ⇒ Binning of shown histograms not relevant
- Example run: 60836 (May 2018), 40 GeV pions, no PP

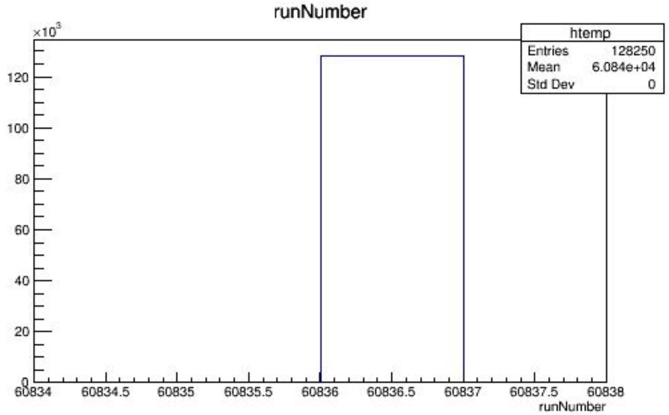


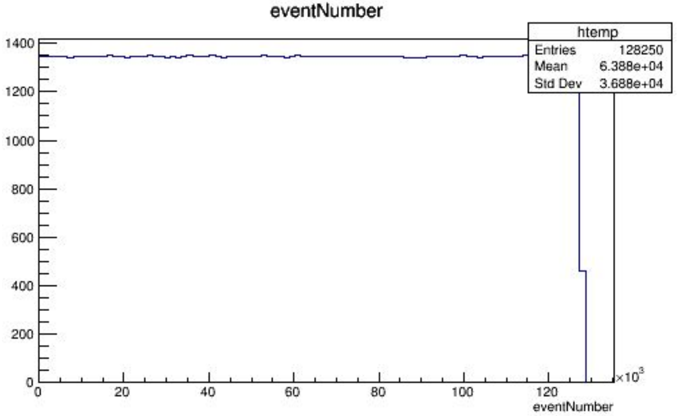
```
run060836_20180520_192802.root
  bigtree;25
    runNumber
    eventNumber
    eventTime
    ahc_iEvt
    ahc_nHits
    ahc_nLayers
    ahc_energySum
    ahc_energyDensity
    ahc_radius
    ahc_radiusEw
    ahc_cogX
    ahc_cogY
    ahc_cogZ
    ahc_energySum5Layer
    ahc_nHits5Layer
    ahc_cogX5Layer
    ahc_cogY5Layer
    ahc_cogZ5Layer
    ahc_energyPerLayer
    ahc_nHitsPerLayer
    ahc_energyPerLayer_err
    ahc_cellSize
    ahc_cogXPerLayer
    ahc_cogYPerLayer
    ahc_radiusPerLayer
    ahc_radiusEwPerLayer
    ahc_hitCellID
    ahc_hitI
    ahc_hitJ
    ahc_hitK
    ahc_hitEnergy
    ahc_hitTime
    ahc_hitType
    ahc_hitRadius
    ahc_hitEnergyDensity
    ahc_hitPos
    bif_StartAcq
    bif_StopAcq
    bif_nTrigger
    bif_source
    bif_BXID
    bif_Time
```

The variables

- runNumber
  - Each event: one value
  - Same for all events in one run

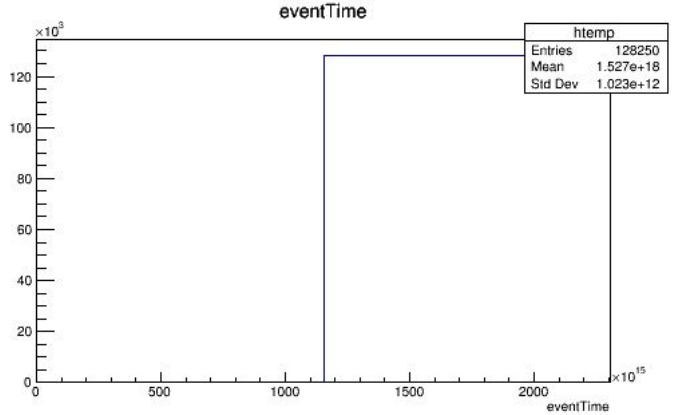
- eventNumber
  - Each event: one value
  - Rather confusing
    - ⇒ Needs to be clarified
    - ⇒ Messes up number of entries
    - ⇒ 38*Entries ≠ Entries of per-layer distributions





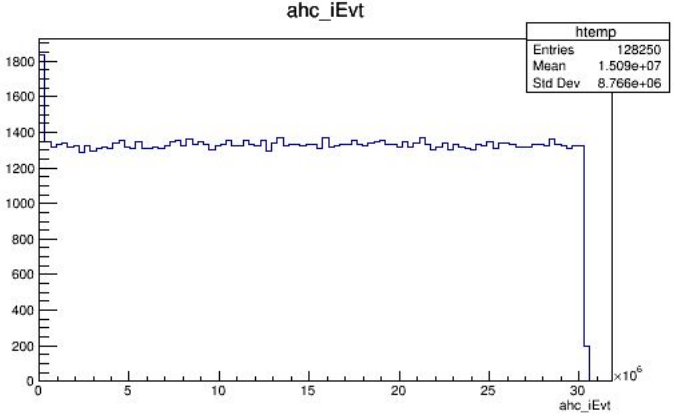
The variables

- eventTime
  - Each event: one value
  - BIF timestamp in unix time



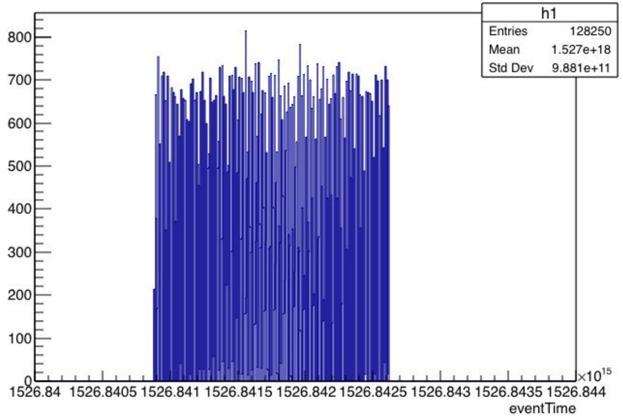
- ahc_iEvt
  - Each event: one value
  - Summed up number of hits up until that event

    ⇒ Is this what we want?



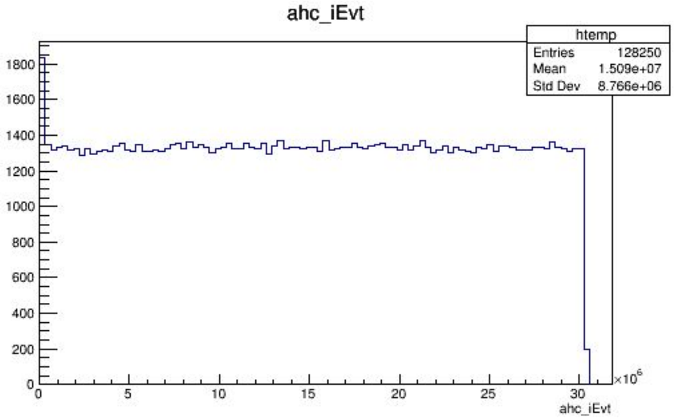
The variables

- eventTime
  - Each event: one value
  - BIF timestamp in unix time



- ahc_iEvt
  - Each event: one value
  - Summed up number of hits up until that event

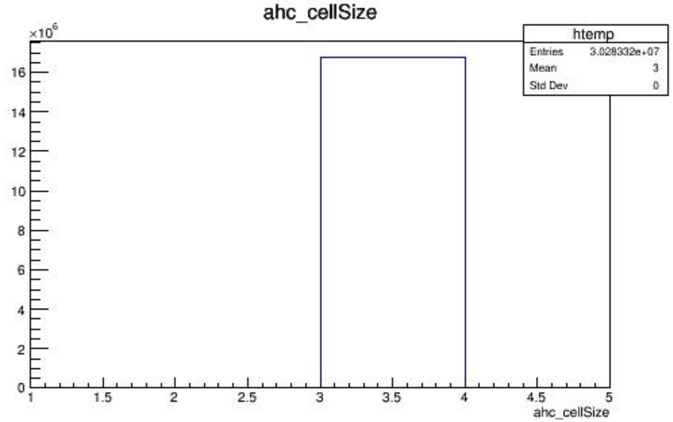
    ⇒ Is this what we want?

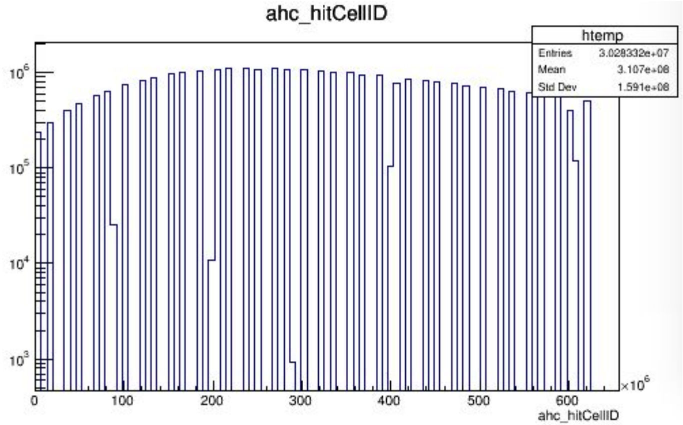


The variables

- ahc_cellSize
  - Each event: array of size of ahc_nHits entry
  - Side length of hit scintillator tile in cm
  - Has now been adapted to current prototype

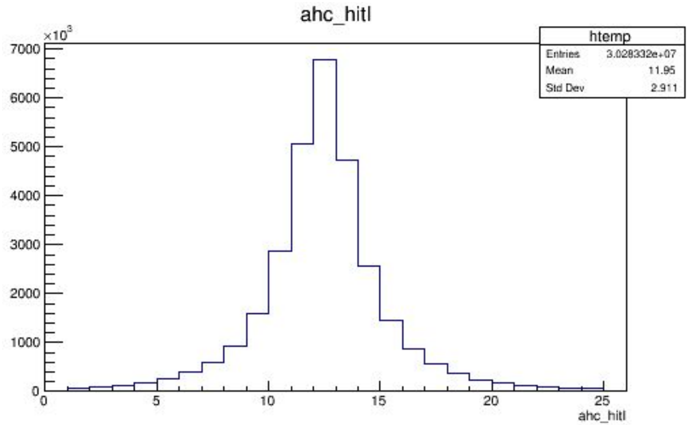
- ahc_hitCellID
  - Each event: array of size of ahc_nHits entry
  - Encoded IJK information of the hit



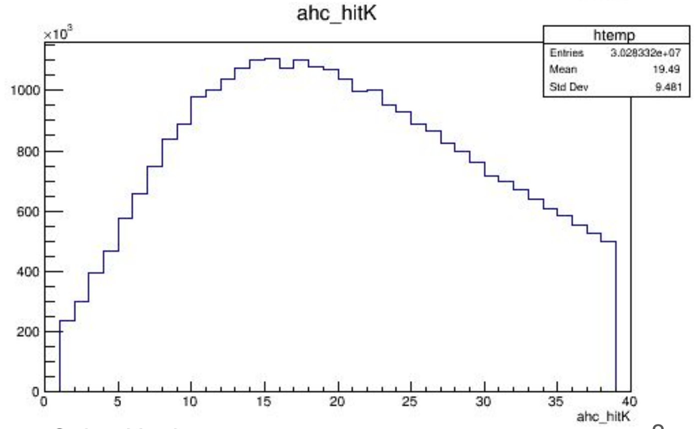


The variables

- ahc_hitI, ahc_hitJ
  - Each event: array of size of ahc_nHits entry
  - Number of hit scintillator tile in x (I) and y (J) direction
  - Tiles numbered from 1 to 24 in each direction
  - Gives the hit position perpendicular to the beam axis



- ahc_hitK
  - Each event: array of size of ahc_nHits entry
  - Number of hit scintillator tile in z direction
  - Corresponds to the layer number
  - Numbered from 1 to 38



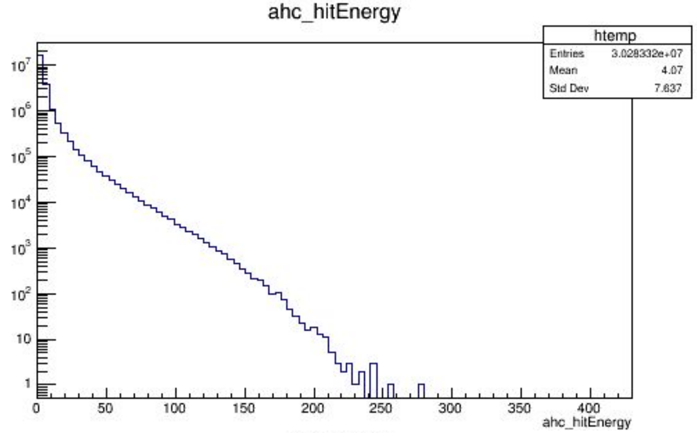
The variables

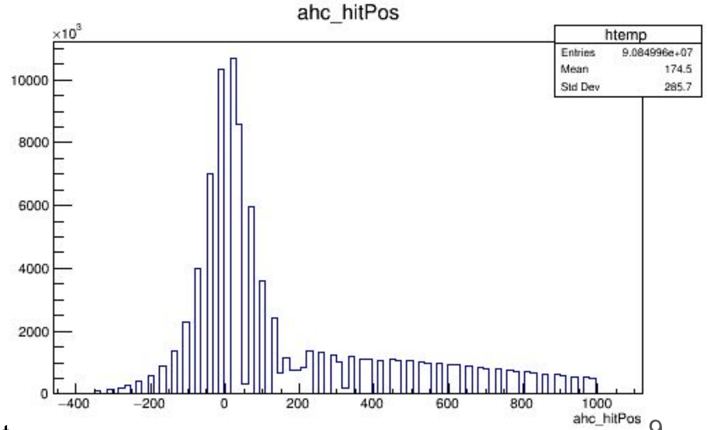
- ahc_hitEnergy

$$E_{\text{calibrated}} = \frac{f_{\text{saturation}}\left[(\text{ADC} - \text{Pedestal}) * \text{IC/Gain}\right]}{\text{IC/Gain} * \text{MIP}}$$

  - Each event: array of size of ahc_nHits entry
  - Energy of single hit in MIP
  - From ADC value, pedestal subtracted
  - For future reference: ahc_hitEnergy = E

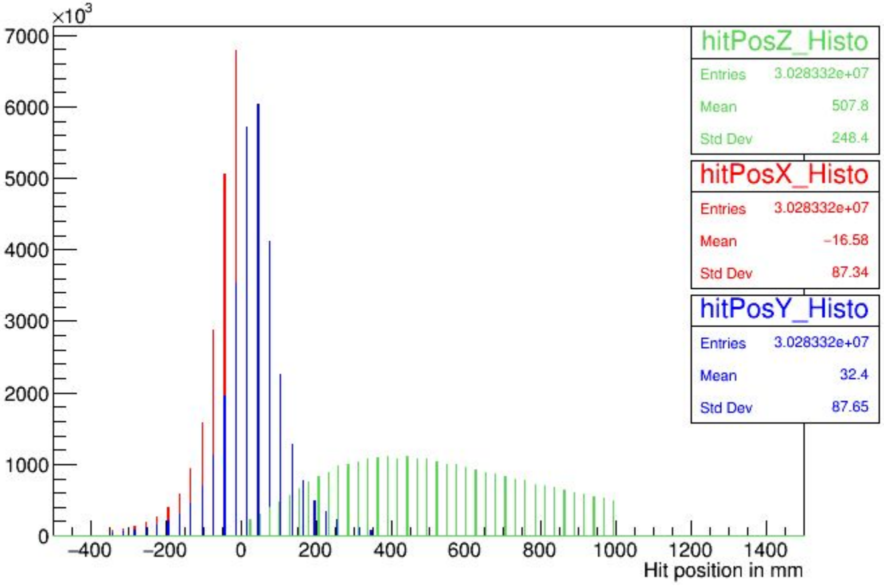
- ahc_hitPos
  - Each event: 3 arrays of size of ahc_nHits entry
  - Hit position in x, y and z in mm (accuracy: one tile)
  - W.r.t. the center of the detector (z: first layer)
  - For future reference:
    - ⇒ ahc_hitPos[0] = x
    - ⇒ ahc_hitPos[1] = y
    - ⇒ ahc_hitPos[2] = z





The variables

- ahc_hitPos
  - Each event: 3 arrays of size of ahc_nHits entry
  - Hit position in x, y and z in mm
  - W.r.t. the center of the detector (z: first layer)
  - For future reference:
    - ⇒ ahc_hitPos[0] = x
    - ⇒ ahc_hitPos[1] = y
    - ⇒ ahc_hitPos[2] = z
  -
    ```
    for(int n = 0; n < bigtree->GetEntries(); n++)
    for(int i = 0; i < ahc_nHits; i++)
    TH1F->Fill(ahc_hitPos[0]); etc.
    ```



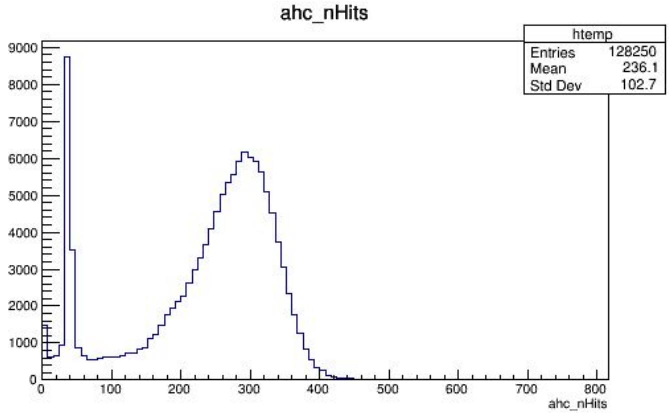
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

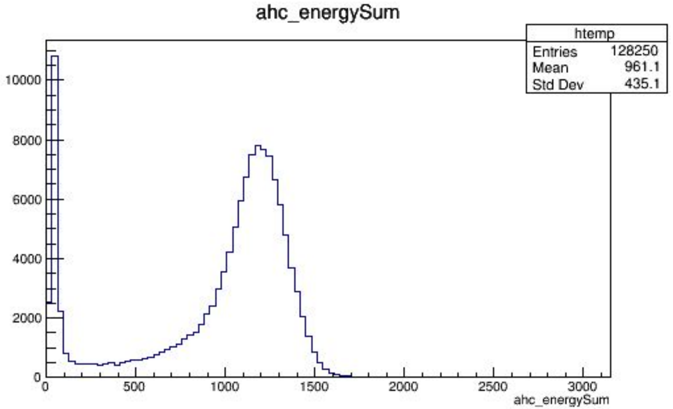
The variables

Calorimeter for ILC

- ahc_nHits
  - Each event: one value
  - Total number of hits above threshold per event
- ahc_energySum
  - Each event: one value
  - Summed up MIP energy of all hits in one event

$$ahc\_energySum = \sum_{i=1}^{ahc\_nHits} E_i$$



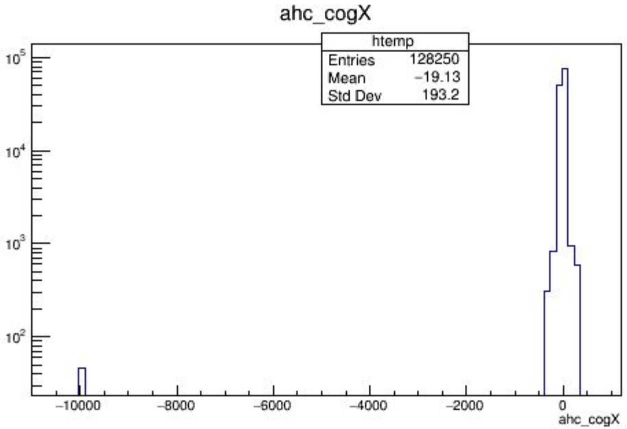


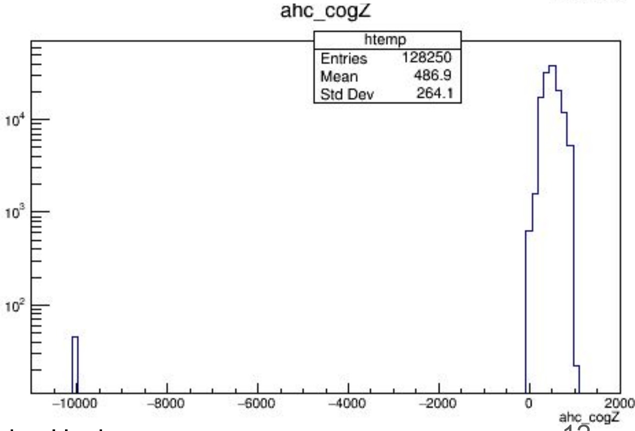
The variables

- ahc_cogX, ahc_cogY
  - Each event: one value
  - Center of gravity of event perpendicular to the beam axis
  - Only if ahc_energySum > 0, otherwise standard value

$$ahc\_cogX = \frac{\sum_{i=1}^{ahc\_nHits} x_i \cdot E_i}{\sum_{i=1}^{ahc\_nHits} E_i} = \frac{\sum_{i=1}^{ahc\_nHits} x_i \cdot E_i}{ahc\_energySum}$$

- ahc_cogZ
  - Each event: one value
  - Center of gravity of event in beam direction
  - Only if ahc_energySum > 0, otherwise standard value





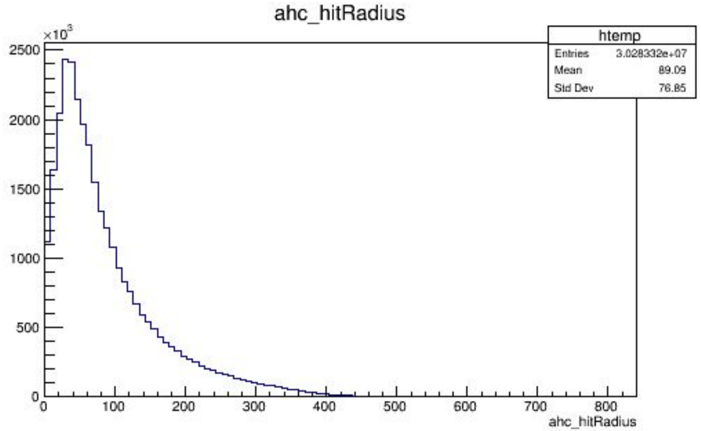
The variables

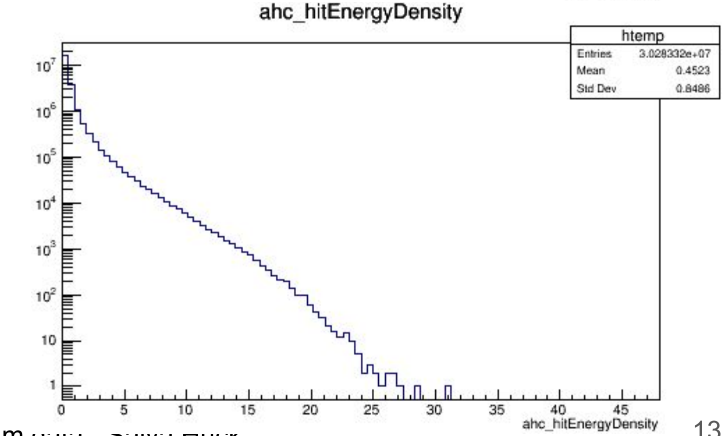
- ahc_hitRadius
  - Each event: array of size of ahc_nHits entry
  - Radial distance from cog of event in x and y in mm

$$ahc\_hitRadius = \sqrt{(ahc\_cogX - x)^2 + (ahc\_cogY - y)^2}$$

- ahc_hitEnergyDensity
  - Each event: array of size of ahc_nHits entry
  - Hit energy divided by tile surface

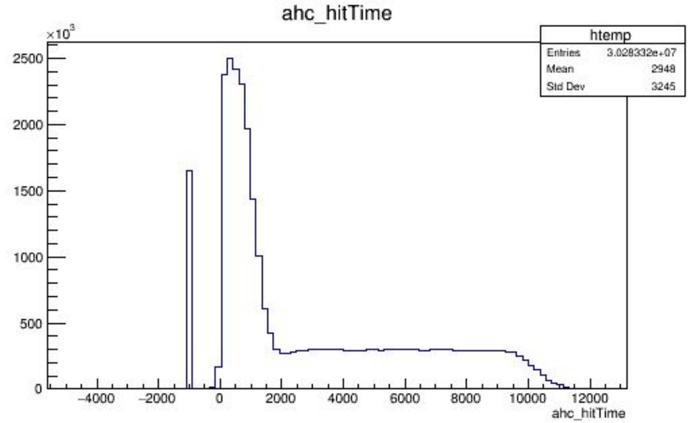
$$ahc\_hitEnergyDensity = \frac{E}{ahc\_cellSize^2}$$



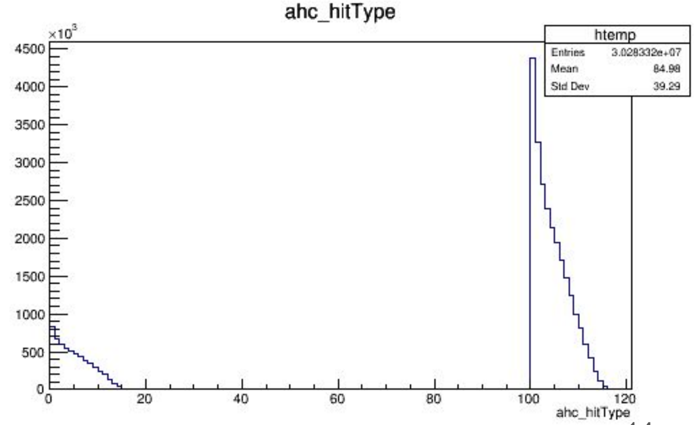


The variables

- ahc_hitTime $\boxed{t[\mathrm{ns}] = \mathrm{TDC} * \mathrm{slope} + \mathrm{Offset} - T_{\mathrm{reference}}}$
  - Each event: array of size of ahc_nHits entry
  - Time of hit in ns
  - From TDC value
  - Calibration constants to follow, not yet correct

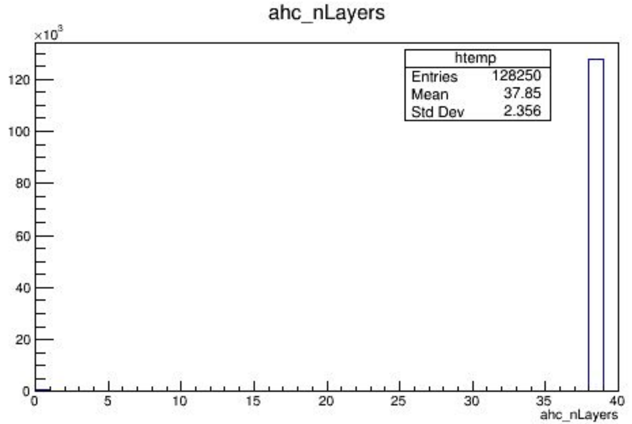


- ahc_hitType
  - Each event: array of size of ahc_nHits entry
  - Definition: Gainbit*100 + memoryCell
  - Information on high (1)/low (0) gain and memory cell

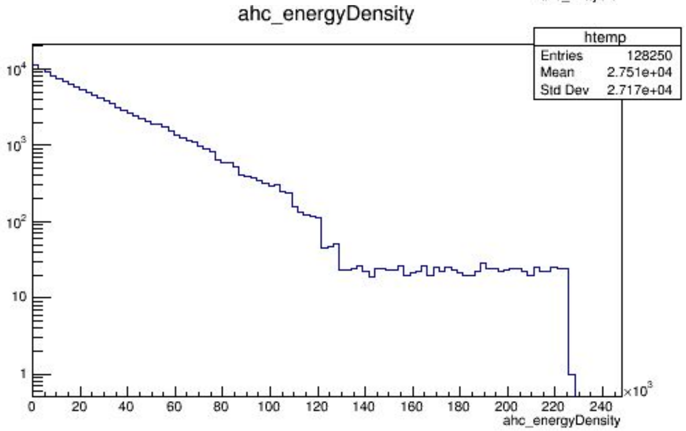


The variables

- ahc_nLayers
  - Each event: one value
  - Should be 38 for all events
  - Has now been adapted to current prototype



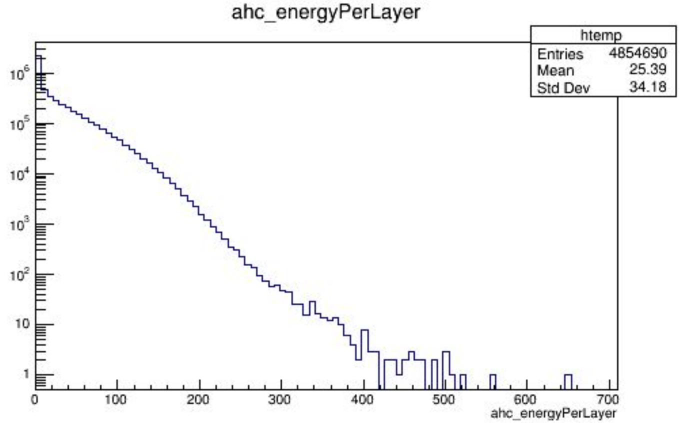
- ahc_energyDensity
  - Each event: one value
  - Energy of all hits in event divided by tile surface
  - Used to not weigh hit distances more in larger tiles



$$ahc\_energyDensity = \sum_{i=1}^{ahc\_nLayers} \left(\sum_{j=1}^{ahc\_nHitsPerLayer_i} \frac{E_j}{{ahc\_cellSize_j}^2} \right)$$

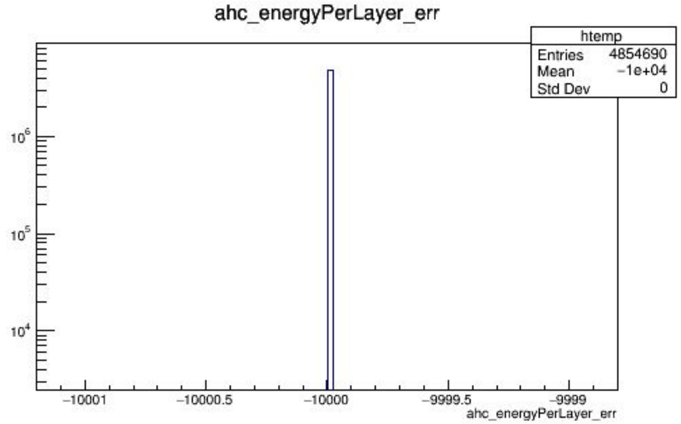
The variables

- ahc_energyPerLayer
  - Each event: array of size of ahc_nLayers entry
  - Energy sum of all hits in one layer per event



- ahc_energyPerLayer_err
  - Each event: array of size of ahc_nLayers entry
  - Information not yet saved in slcio files

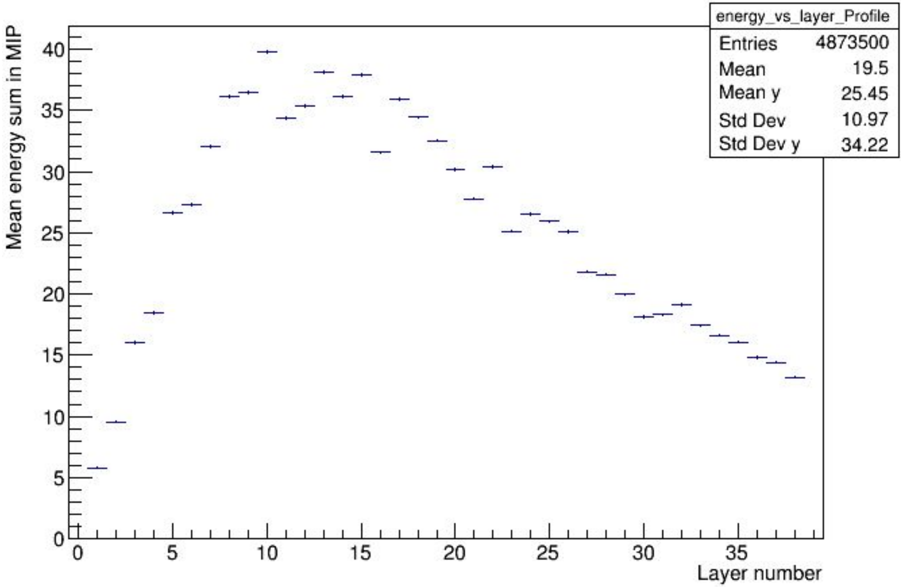
⇒ All entries: standard value



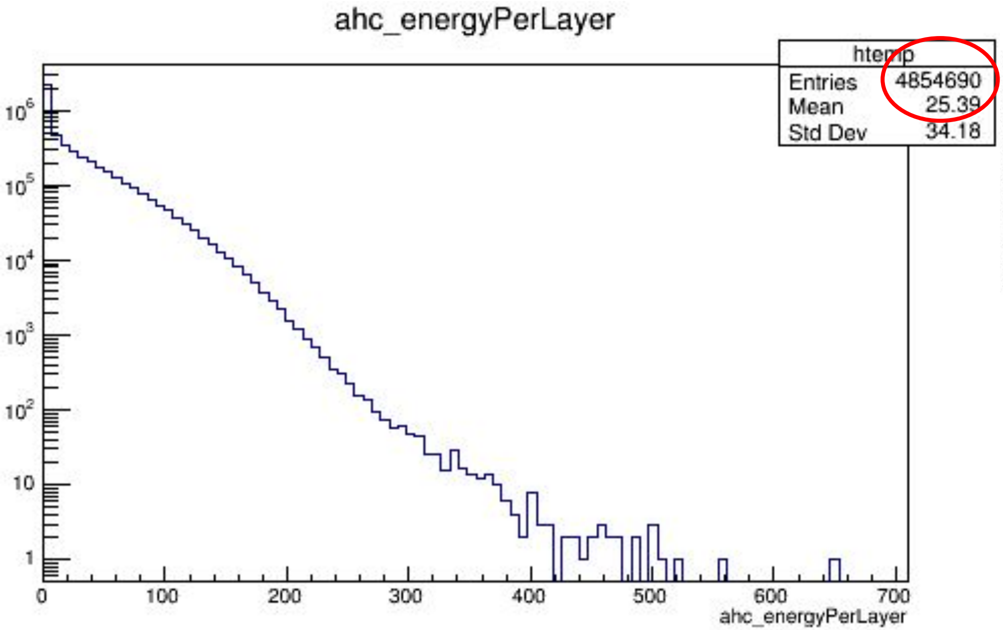
The variables

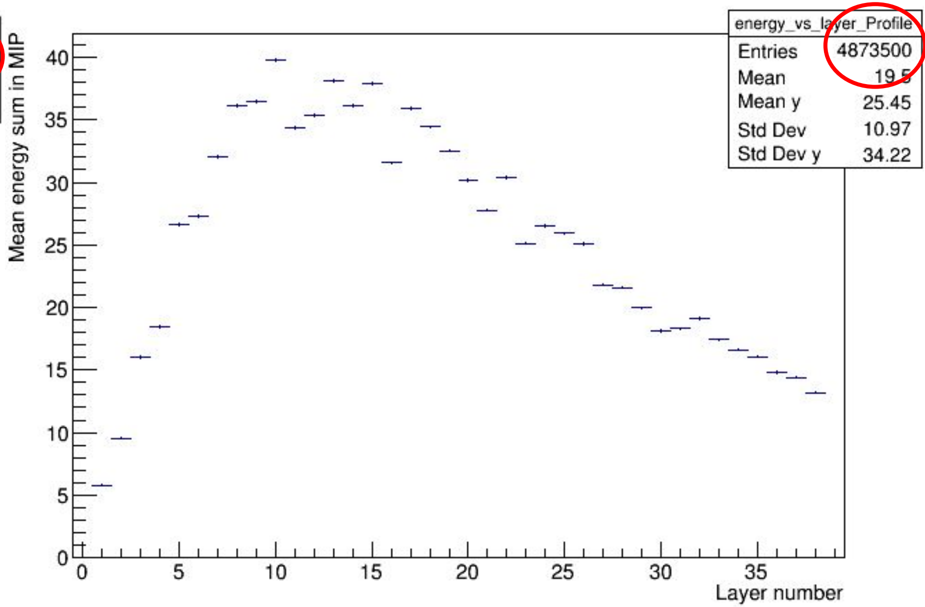
- ahc_energyPerLayer
  - Each event: array of size of ahc_nLayers entry
  - Energy sum of all hits in one layer per event

  -
    ```
    for(int n = 0; n < bigtree->GetEntries(); n++)
    for(int i = 0; i < ahc_nLayers; i++)
    TProfile->Fill(i+1, ahc_energyPerLayer[i]);
    ```



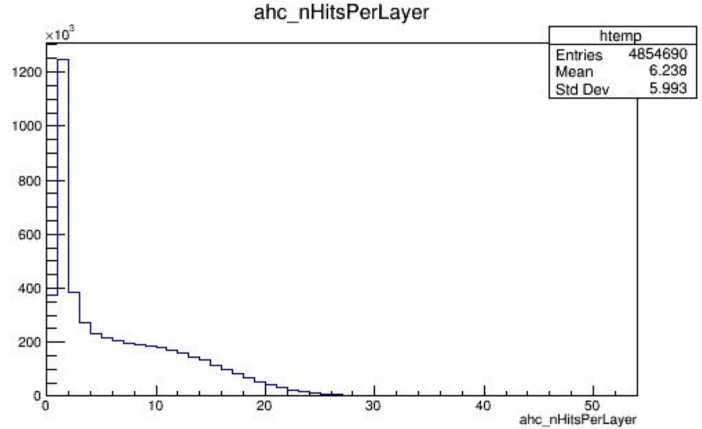
The variables





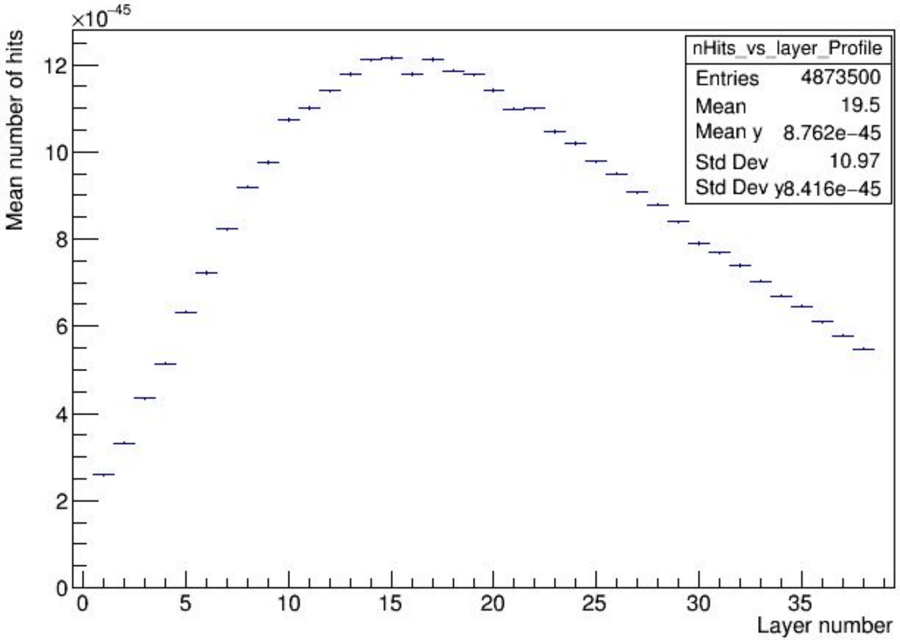
The variables

- ahc_nHitsPerLayer
  - Each event: array of size of ahc_nLayers entry
  - Number of hits in each layer per event



The variables

- ahc_nHitsPerLayer
  - Each event: array of size of ahc_nLayers entry
  - Number of hits in each layer per event
  - for(int n = 0; n < bigtree->GetEntries(); n++)
    for(int i = 0; i < ahc_nLayers; i++)
    TProfile->Fill(i+1, ahc_nHitsPerLayer[i]);

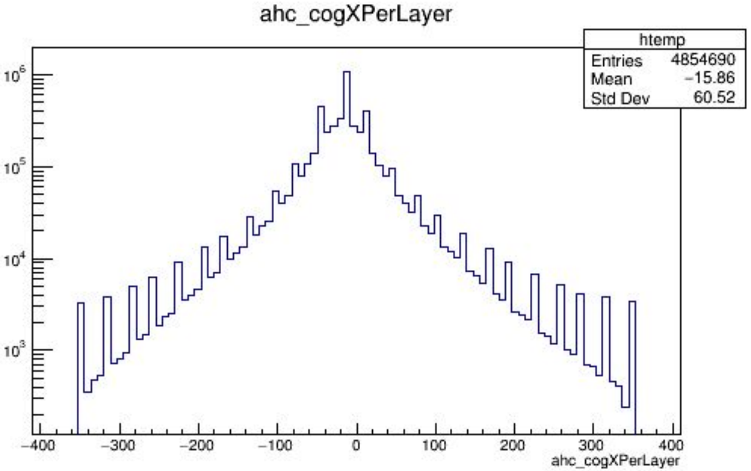


The variables

- ahc_cogXPerLayer, ahc_cogYPerLayer
  - Each event: array of size of ahc_nLayer entry

    ⇒ Only layers with ahc_energyPerLayer > 0
  - Mean center of gravity (x,y) in each layer per event

$$ahc\_cogXPerLayer = \frac{\sum_{i=1}^{ahc\_nHitsPerLayer} x_i \cdot E_i}{ahc\_energyPerLayer}$$

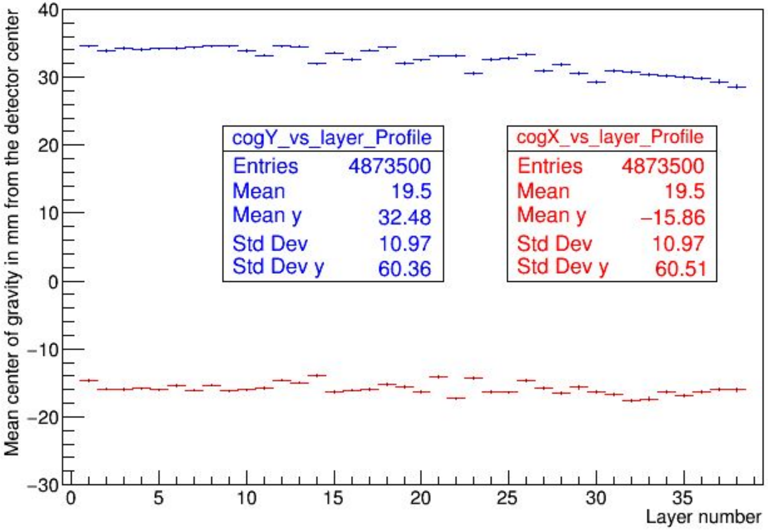


The variables

- ahc_cogXPerLayer, ahc_cogYPerLayer
  - Each event: array of size of ahc_nLayer entry

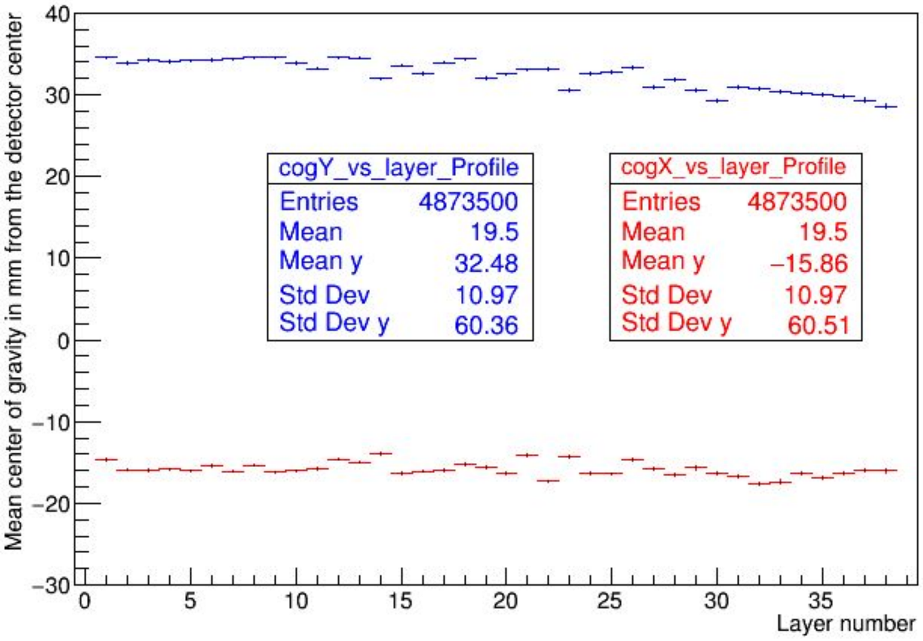
    ⇒ Only layers with ahc_energyPerLayer > 0
  - Mean center of gravity (x,y) in each layer per event

  -
    ```
    for(int n = 0; n < bigtree->GetEntries(); n++)
    for(int i = 1; i < ahc_nLayers; i++)
    TProfile->Fill(i, ahc_cogXPerLayer[i]);
    ```



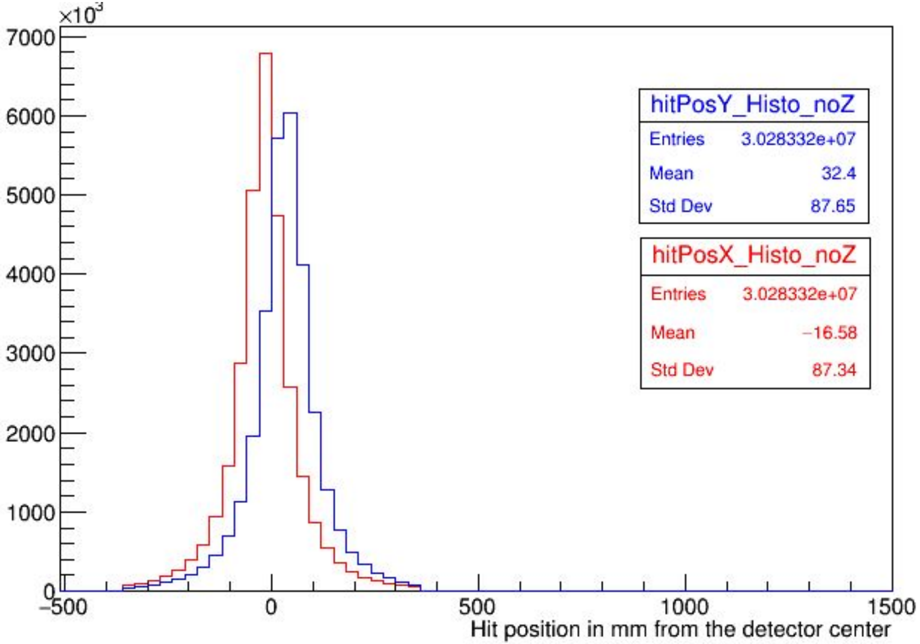
$$ahc\_cogXPerLayer = \frac{\sum_{i=1}^{ahc\_nHitsPerLayer} x_i \cdot E_i}{ahc\_energyPerLayer}$$

The variables



cogY_vs_layer_Profile	
Entries	4873500
Mean	19.5
Mean y	32.48
Std Dev	10.97
Std Dev y	60.36

cogX_vs_layer_Profile	
Entries	4873500
Mean	19.5
Mean y	−15.86
Std Dev	10.97
Std Dev y	60.51



hitPosY_Histo_noZ	
Entries	3.028332e+07
Mean	32.4
Std Dev	87.65

hitPosX_Histo_noZ	
Entries	3.028332e+07
Mean	−16.58
Std Dev	87.34

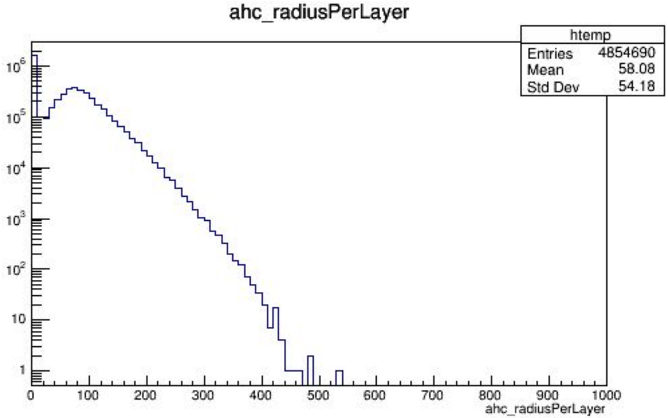
The variables

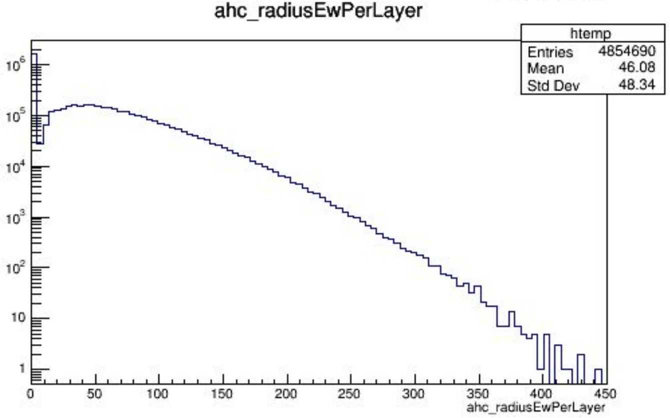
- ahc_radiusPerLayer
  - Each event: array of size of ahc_nLayers entry
  - Mean radius of the hits in each layer in each event

$$ahc\_radiusPerLayer = \frac{\sum_{i=1}^{ahc\_nHitsPerLayer} \sqrt{(ahc\_cogXPerLayer - x_i)^2 + (ahc\_cogYPerLayer - y_i)^2}}{ahc\_nHitsPerLayer}$$

- ahc_radiusEwPerLayer
  - Each event: array of size of ahc_nLayers entry
  - Energy weighted mean radius

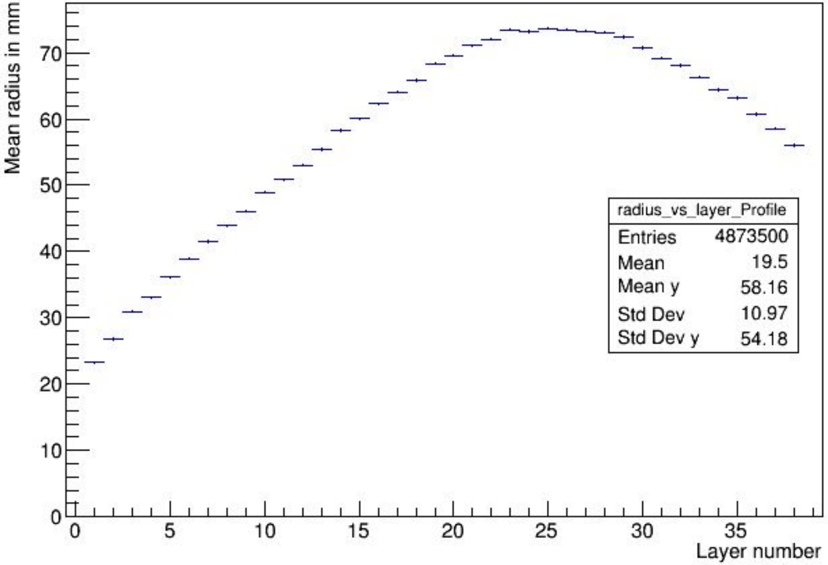
$$ahc\_radiusEwPerLayer = \frac{\sum_{i=1}^{ahc\_nHitsPerLayer} \frac{E_i}{ahc\_cellSize_i{}^2} \cdot \sqrt{(ahc\_cogXPerLayer - x_i)^2 + (ahc\_cogYPerLayer - y_i)^2}}{\sum_{i=1}^{ahc\_nHitsPerLayer} \frac{E_i}{ahc\_cellSize_i{}^2}}$$

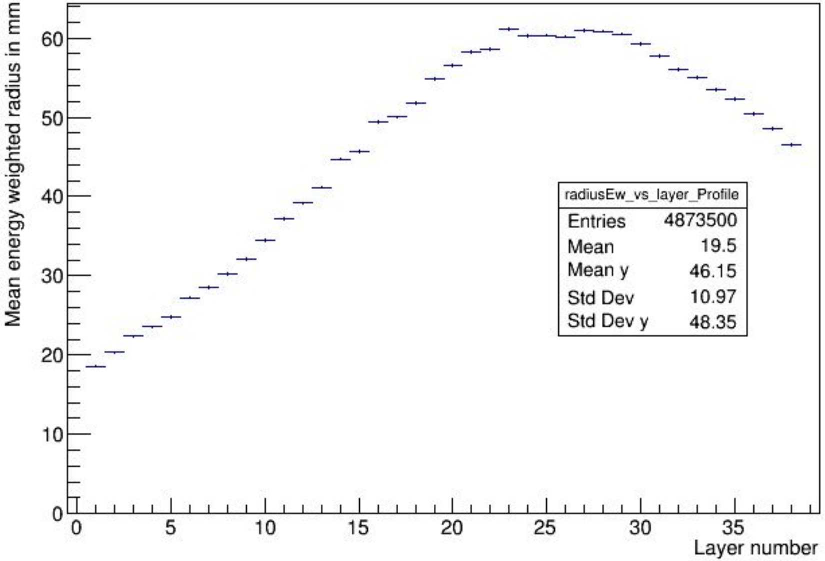




The variables

- ahc_radiusPerLayer, ahc_radiusEwPerLayer
  - for(int n = 0; n < bigtree->GetEntries(); n++)
    for(int i = 1; i < ahc_nLayers; i++)
    TProfile->Fill(i, ahc_radiusPerLayer[i]); etc.





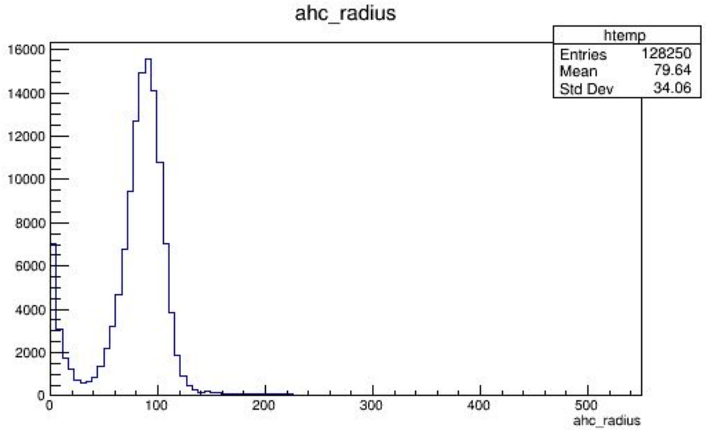
The variables

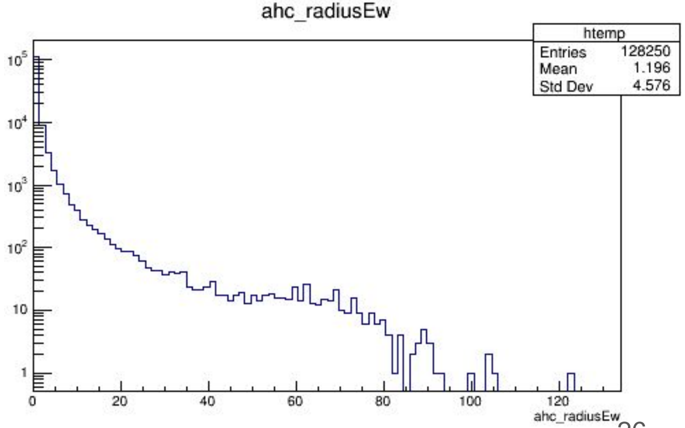
- ahc_radius
  - Each event: one value
  - Mean radial distance of all hits in an event from the cog

$$ahc\_radius = \frac{\sum_{i=1}^{ahc\_nHits} ahc\_hitRadius_i}{ahc\_nHits}$$

- ahc_radiusEw
  - Each event: one value
  - Energy weighted radius

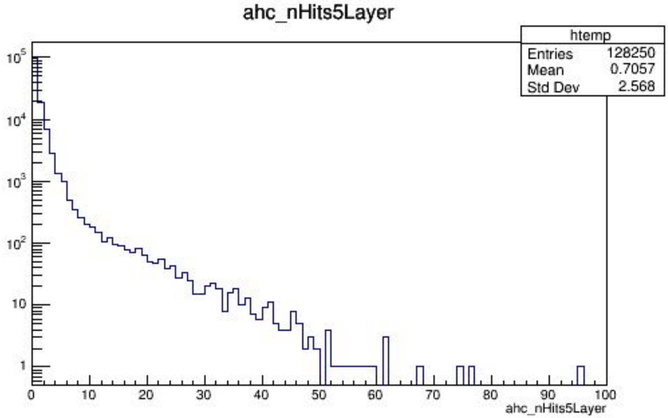
$$ahc\_radiusEw = \frac{\sum_{i=1}^{ahc\_nLayers} \left(\sum_{j=1}^{ahc\_nHitsPerLayer_i} \frac{E_j}{ahc\_cellSize_j{}^2} \cdot \sqrt{(ahc\_cogXPerLayer - x_j)^2 + (ahc\_cogYPerLayer - y_j)^2} \right)}{ahc\_energyDensity}$$



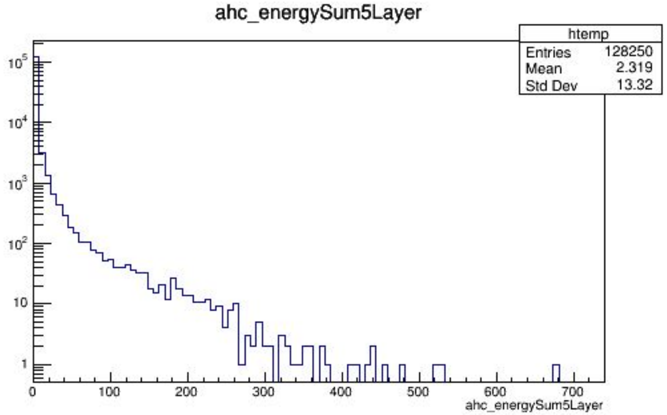


The variables

- ahc_nHits5Layer
  - Each event: one value
  - Number of hits in first 5 layers
    - ⇒ BUT: only counts hits outside a radius of 280 mm
    - ⇒ Measured from center of detector



- ahc_energySum5Layer
  - Each event: one value
  - Energy sum in first 5 layers
    - ⇒ BUT: only counts hits outside a radius of 280 mm
    - ⇒ Measured from center of detector



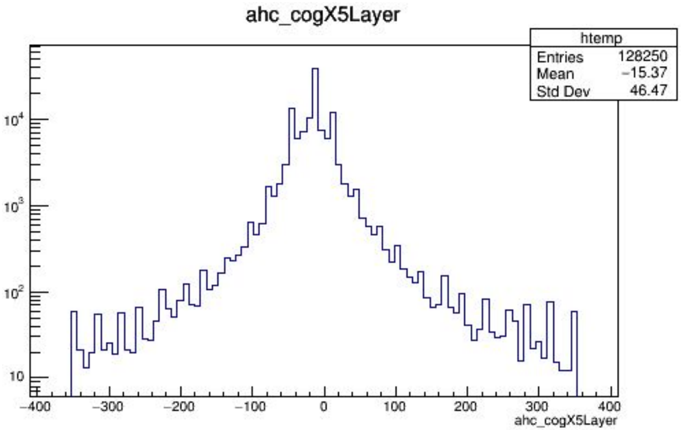
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

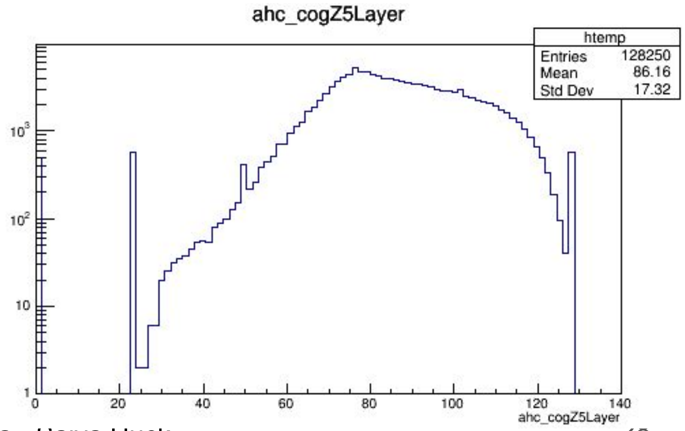
The variables

Calorimeter for ILC

- ahc_cogX5layer, ahc_cogY5Layer
  - Each event: one value
  - Mean center of gravity in first 5 layers in x and y
  - Averaged over ALL hits in first 5 layers

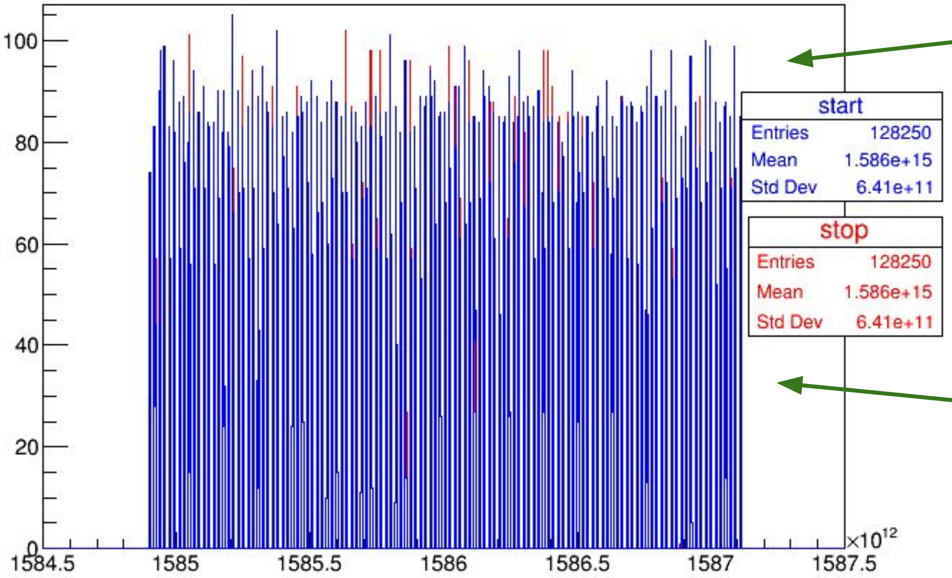
- ahc_cogZ5Layer
  - Each event: one value
  - Mean center of gravity in first 5 layers in z
  - Averaged over ALL hits in first 5 layers

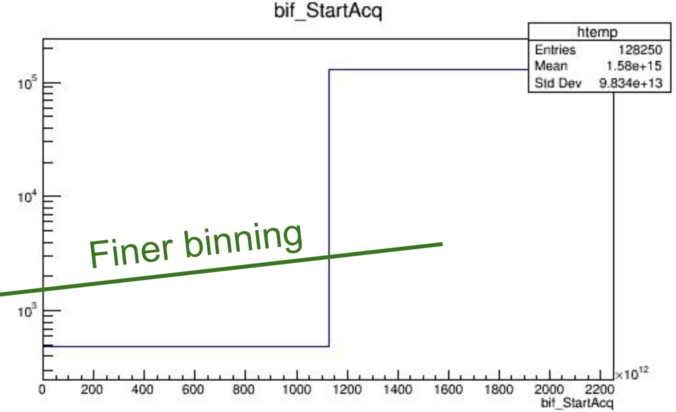




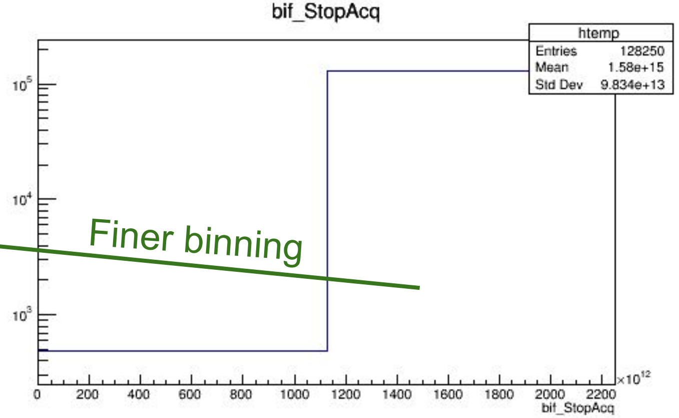
The variables

- bif_StartAcq, bif_StopAcq
  - Each event: one value each
  - Trigger times w.r.t. the time the BIF was switched on





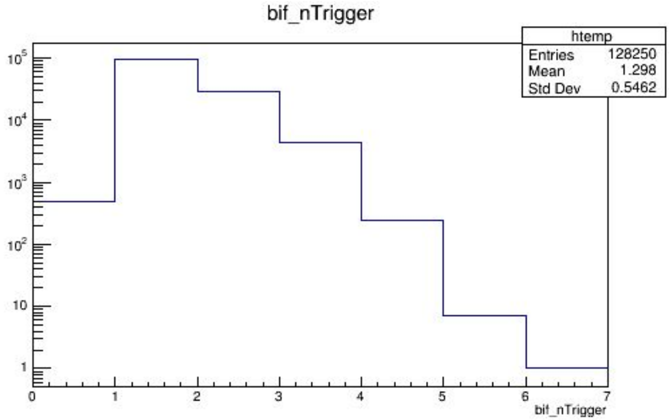
Finer binning



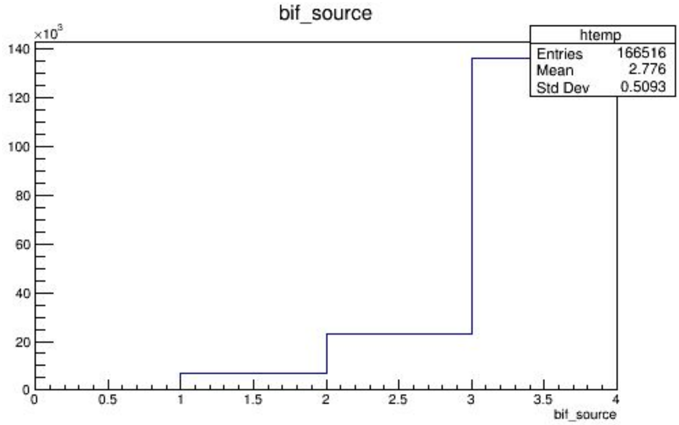
Finer binning

The variables

- bif_nTrigger
    - Each event: one value
    - Number of triggers per event

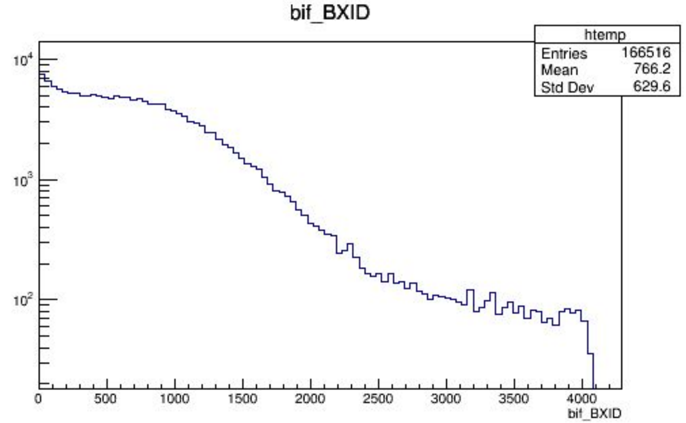


- bif_source
    - Each event: array of size of bif_nTrigger value
    - Up to 4 devices connected to BIF
    - Info which input was used
    - Can be >1 trigger in the same source for the same event

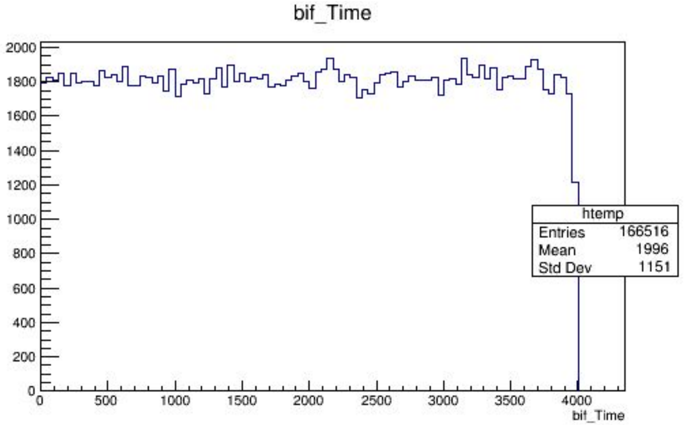


The variables

- bif_BXID
  - Each event: array of size of bif_nTrigger value
  - Bunch crossing ID of each trigger event



- bif_Time
  - Each event: array of size of bif_nTrigger value
  - Time of each trigger event



Ideas for additions?

- Information in physics units (radiation length, ...)
- Errors in slcio objects
- Module/Chip/Channel info in addition to IJK
- RMS/skewness for entire runs?
  - Also layer-wise?
- Keep/adapt 5Layer variables?
- Adapt ahc_iEvt?
- ...