Data Quality Tools

N. Feege, E. Garutti, S. Morozov, A. Vargas

Motivation

Large amount of data on tape & coming which we want to analyse!!

======> Check the quality of our devices e.g HCAL

- Check global variables (e.g total Energy)
- Identify local Problems

Define a Run Status: Good, Medium, Bad

- all Information have to be stored in a DB

Global Variables

- Trigger rates
- Beam Quality
  * Beam size -> store beam position & spread for MC
  * double particles -> how many particles can we use ?
  * Beam composition: e,μ,π content
    -e.g 30%pions would define a Medium pion run
  * Beam energy determination -> compare with elog entry
    and tag differences to be investigated
- Drift chamber Quality
  * First: check if it is properly converted
  * correlation of x,y Ecal vs DC track ==> values can go to DB
  * Efficiency

Hcal Problems

HCAL related:

- Identification of Dead Channels	info to DB
- Half or complete missing Module	Hcal Bad
- Pedestal stability	info since fluctuations can be corrected
- Temperature behaviour	info to DB
- Led stability	info to DB

Similar criteria can be defined for other detector components

Available Information

	2007	2008
Trigger	ok	ok
Ecal	ok	?
Hcal	ok	ok
Tcmt	ok	ok
Drift Chamber	ok	-
Number of Runs	~200	~80

====> Begin with 2007 data

RootTreeWriter

Tool: RootTreeWriter

Modular Root-Ntuple where Engines can be switched on/off.

Engines:

Trigger, Ecal ,Hcal, Tcmt & Drift Chamber

- starting point: LCIO files
- Hcal: variables at several level are stored
  e.g global/layer-wise/cell-level

Data Quality tools

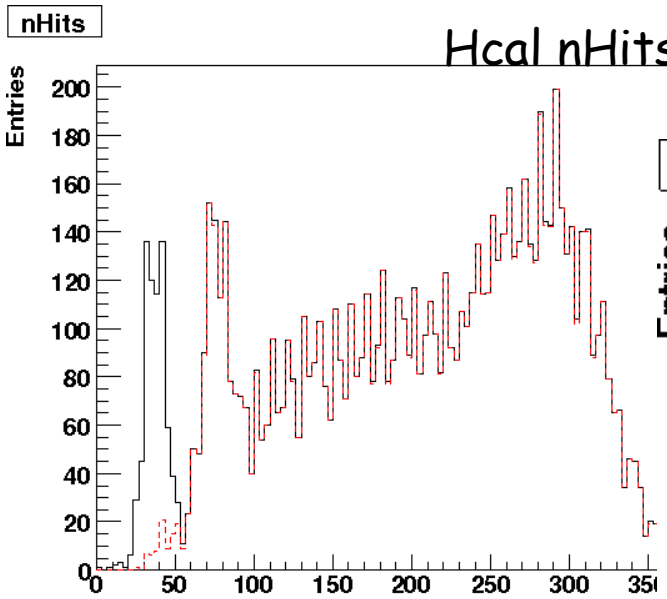
Use the RootTreeWriter as the main tool to perform the data quality analysis.

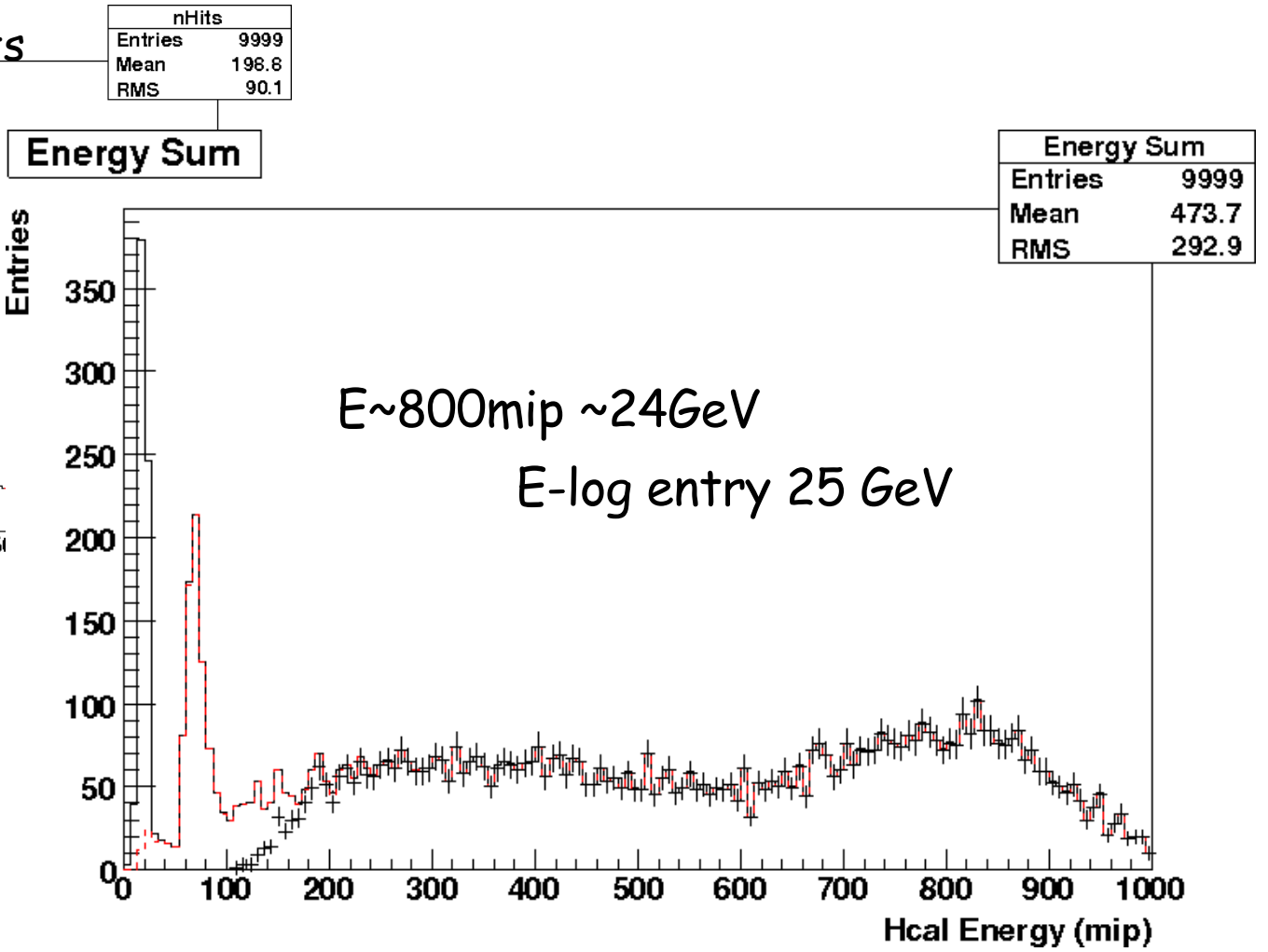
But, 1 Run has ~ 250,000 evts ---> root-Ntuple ~ 2 G

Reduce the size:

- Saving the Ntuple just each 100th event ===> ~20 M
- Save all events but exclude cell-level information ===> ~200M

First root-Ntuple



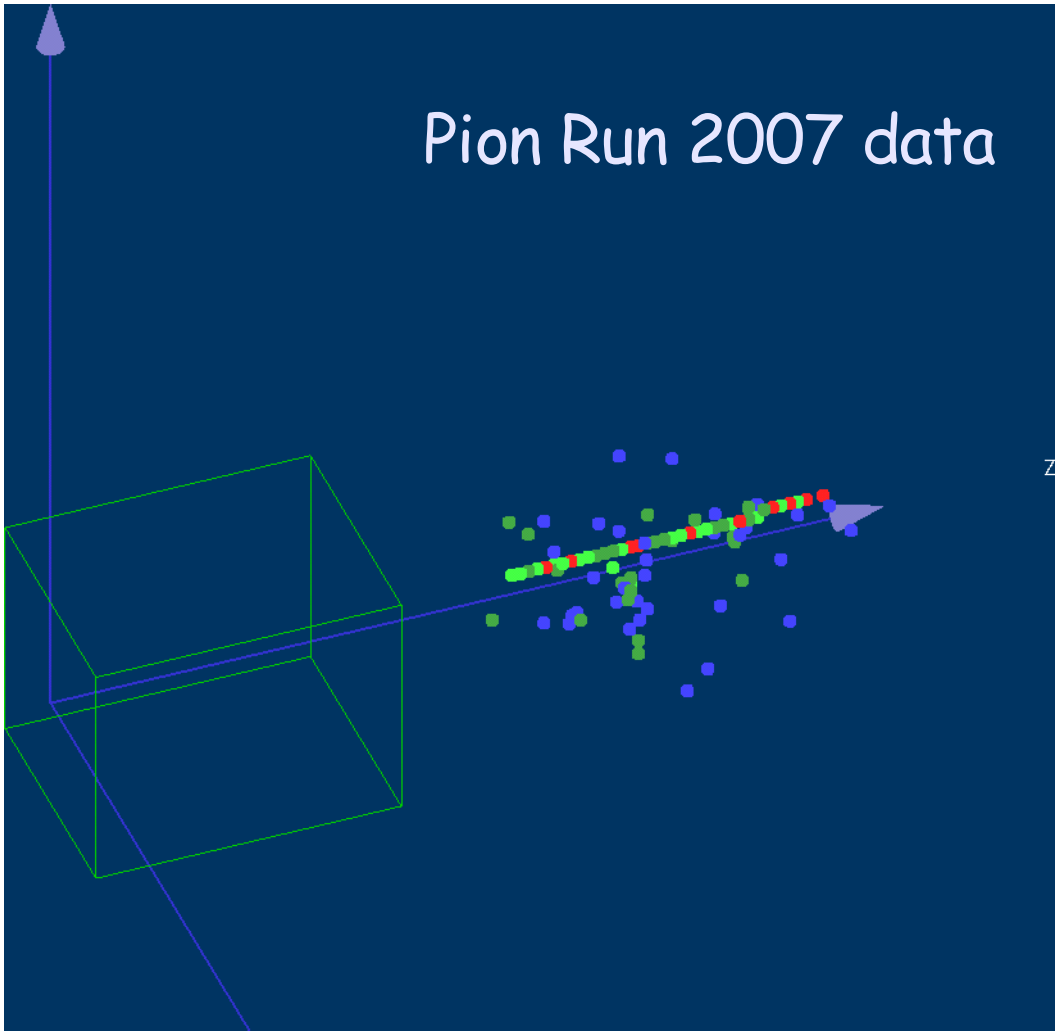


First root-Ntuple

Using global quantities
-layer-wise:

Muon Finder:

Hcal Esum< 100 mip &&
at least 25 layers
having 0< nHits <4



Outlook

- Data quality checks for 2007 data have started!
  - Ntuples to be produced in the local cluster

- Goal is to have a complete DQ-chain which provides a list of information to be stored in a DB

  - from LCIO: run#, run type, Number of events, E, x, y, theta
  - from Elog: E, theta
  - from DQ: number of expected phys events, ... HCAL,ECAL,TCMT,DC,beam good/medium/bad ...

Back-up

```
int   nHits;
int   nLayers;          // number of layers < MAXLAYERS !
float energySum;
float cogX;            // center of gravity weighted by energy on whole calo
float cogY;
float cogZ;
float radius;          // shower radius (in x,y plane) calculated w.r.t. The cog
float radiusEw;        // shower radius (in x,y plane) calculated w.r.t.
                                the cog Energy weighted
```

Back-up

```
int   cellID[MAXCELLS];
float hitEnergy[MAXCELLS];
float hitRadius[MAXCELLS];            // hit distance to the cog
                                      (ideally one wants to replace the cog with the DC track)
float hitEnergyDensity[MAXCELLS];     // hit energy weighted by tile_size^2*hitRadius
                                      (dimention: E/mm^3)
float hitPos[MAXCELLS][3];

int   lNHits[MAXLAYERS];              // layer by layer number of hits
float lEnergy[MAXLAYERS];             // layer by layer Energy
float lEnergy_err[MAXLAYERS];         // layer by layer Energy error
                                        (not yet filled in CalorimeterHit class)

float lCogX[MAXLAYERS];         // center of gravity weighted by energy in one layer
float lRadius[MAXLAYERS];       // shower radius (in x,y plane) in one layer
                                   calculated w.r.t. the cog
float lRadiusEw[MAXLAYERS];     // Energy weighted shower radius (in x,y plane)
                                   in one layer calculated w.r.t. the cog
```