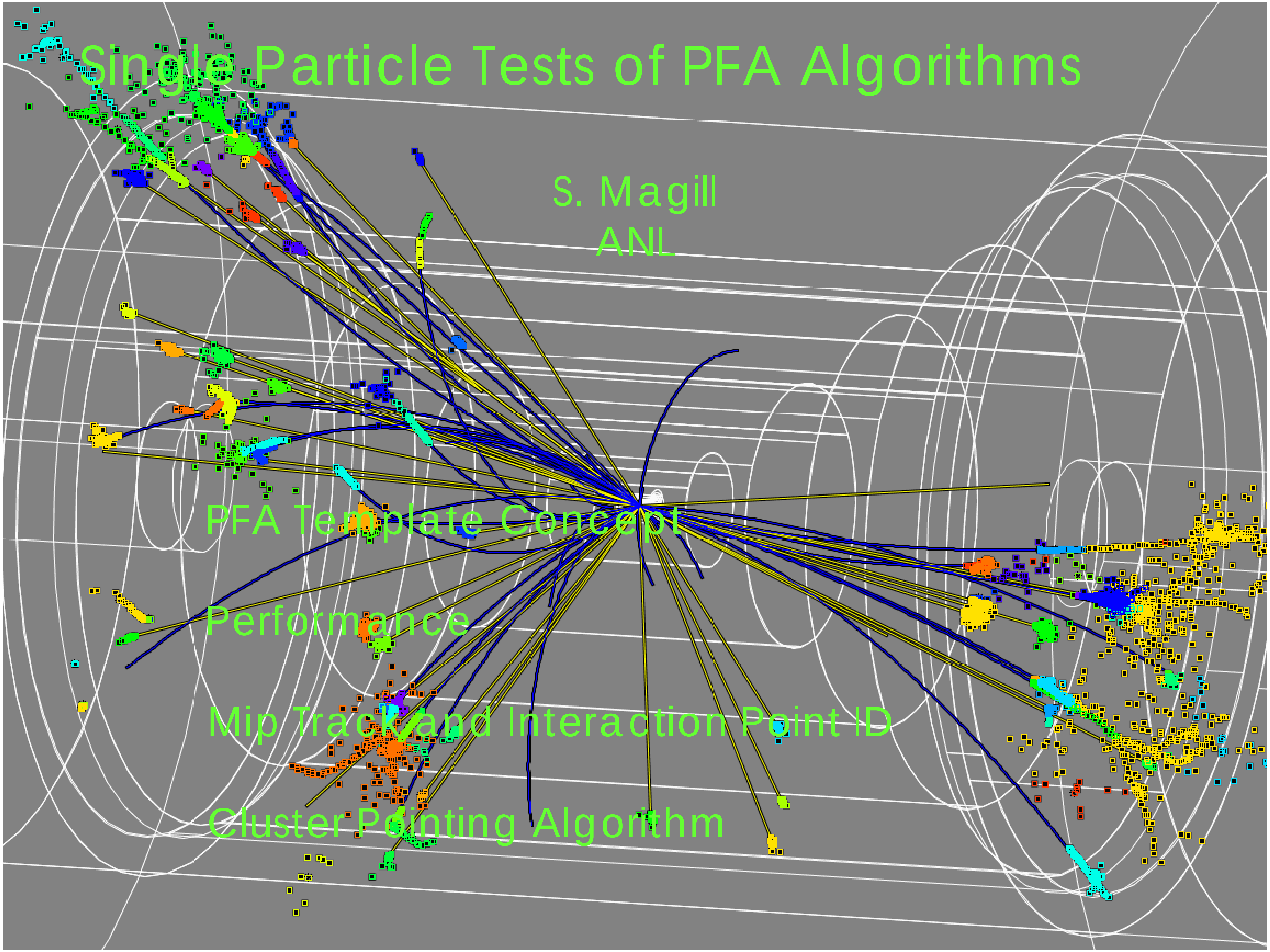


Single Particle Tests of PFA Algorithms

S. Magill
ANL

PFA Template Concept

Performance

Mip Track and Interaction Point ID

Cluster Pointing Algorithm

PFA Template Concept

Modular PFA composed of multiple individual particle ID algorithms

Common IO throughout PFA for cluster, ID algorithms

- at each step, complete set of subdetector hitmaps modified by the previous algorithm (allEM, allHAD, allHITS)
- allows interchangeability of algorithm order, cluster and ID algorithms
- for example, different optimized clustering can be used at each step
- ease of algorithm import

Relies as much as possible on single particle tuning of individual algorithms (as opposed to process tuning)

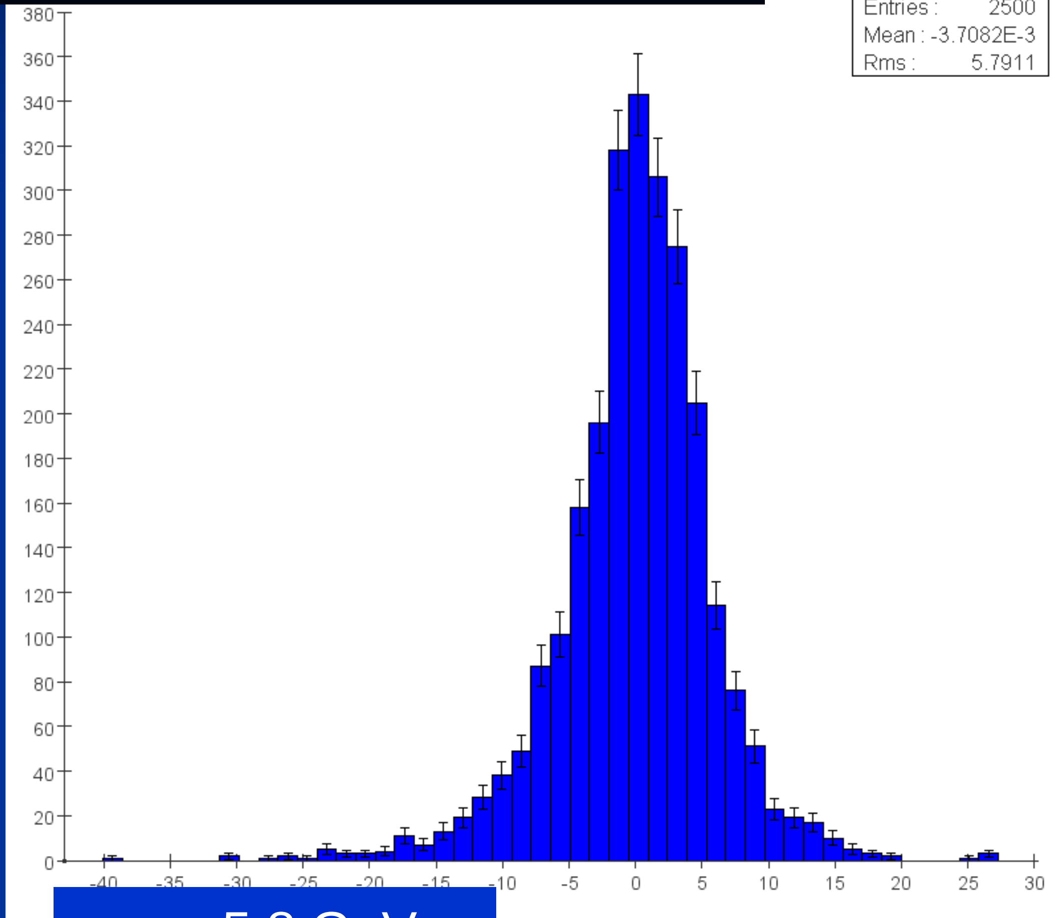
- can test/tune individual algorithms in test beam(s)

Current PFA Template – PFAMain.java

DigiSim – hit digitization, timing, threshold cuts
Perfect PFA – standard Perfect RPs, cheated tracks
Cheated/Reconstructed Tracks
Track Extrapolation Maps – spacepoints along extrapolated tracks ..., layer 0 ECAL/HCAL, ECAL shower max
Track-Mip Association – mip segment, interaction point of charged particles
Cluster Pointing Algorithm – 3 cluster classes; points at charged particle interaction spacepoint, points at IP, non-pointing
Photon Finder I – subset of IP-pointing clusters based on tra distance
Photon Finder II (R. Cassell) – multi-variable evaluation of DT clusters
Track-Cal Cluster Matching – iterative matching of clusters to track using distance, E/p
Photon Finder III – low energy photon clusters
Track Proximity Cleaner – photon candidates trimmed near tracks
EM/HAD Cluster Merger – merges EM and HAD clusters in cone
Neutral Hadron Finder – leftover clusters
Reconstructed Particles -> Energy Sum, Jet Finding

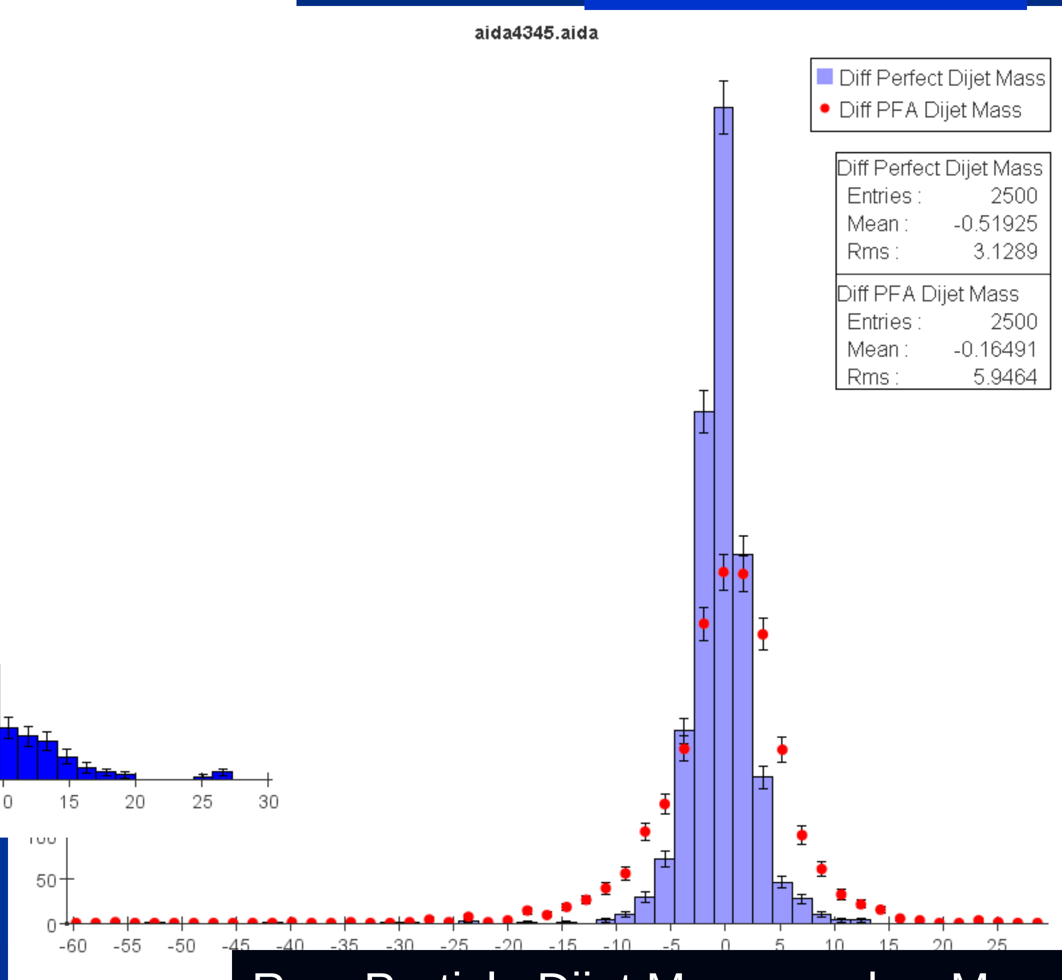
PFA Template Performance – qqbar100

RecoParticle ESum – qqbar ESum



rms = 5.8 GeV
rms90 = 3.7 GeV
alpha90 = 0.37

rms = 5.9 GeV
rms90 = 3.8 GeV
alpha90 = 0.38



RecoParticle Dijet Mass – qqbar Mass

Associating Cal Showers with Tracks
Track/Mip and Track/Shower Algorithms for PFA Template

Tracks

- cheated, from Perfect PFA (ReconFSTracks)
- extrapolated using helical swimmer with MC p, MC origin, charge, Bz
- ready for real track extrapolation with measured p, origin, charge, Bz

Track Extrapolation Map Utility

- maps spacepoint to track extrapolated to E0, EM Shower Max, H0

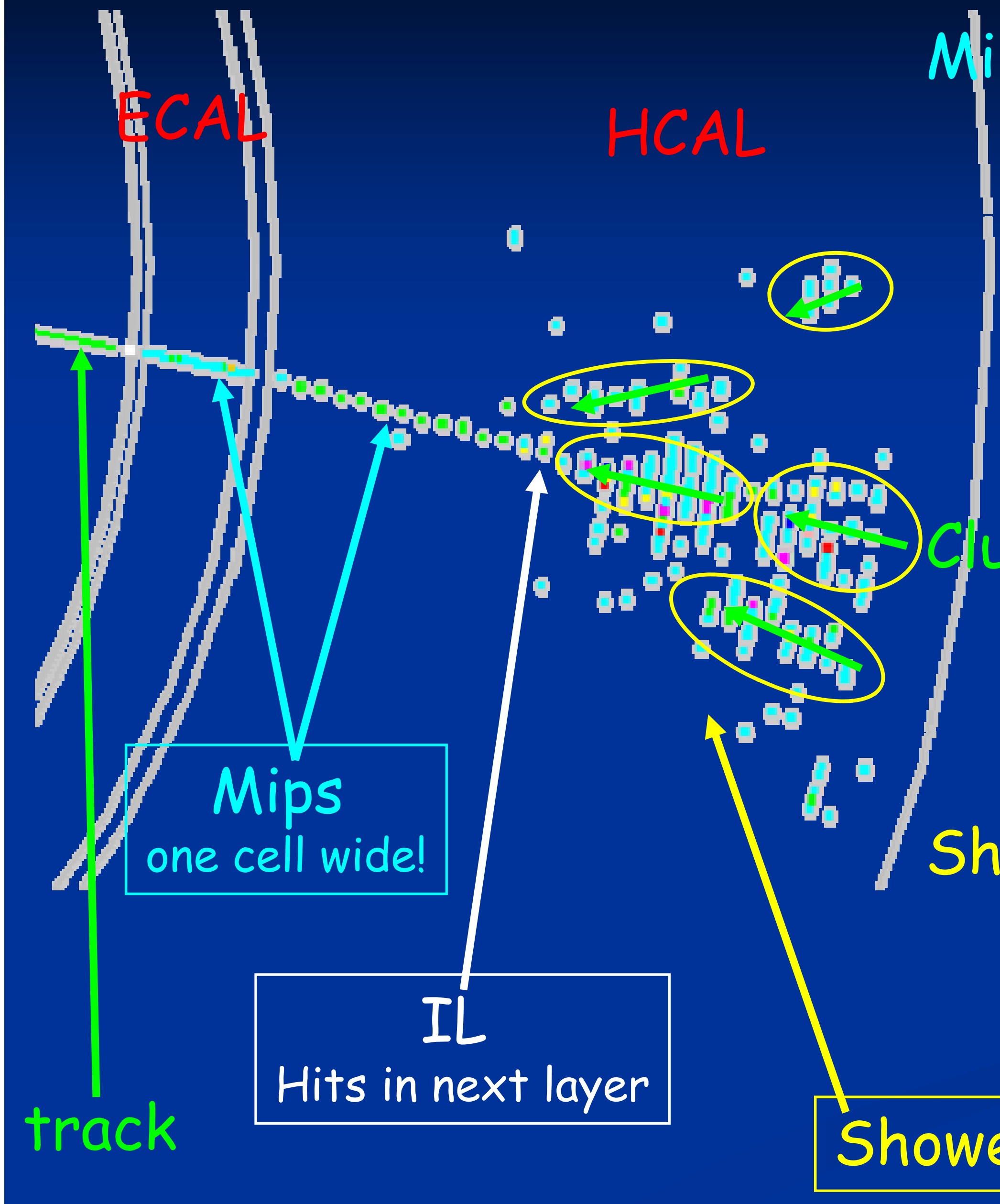
Track Mip Cluster and Interaction Layer Finder

Cluster Pointing Algorithm

Track Shower Cluster Finder

- associates clusters to tracks starting from IL
- first, finds core clusters by searching in same region as mip finder
- uses cluster proximity ($\Delta\theta$,$\Delta\phi$) and E/p measure based on CAL resolution for p
- iterates expanding cone until E/p window is met or max cone size is reached
- outputs are track shower clusters (includes mips, core, and shower)

Shower reconstruction by track extrapolation



Mip reconstruction :

Extrapolate track through CAL layer-by-layer
Search for "Interaction Layer"
-> Clean region for photons (ECAL)
-> mip clusters matched to tracks

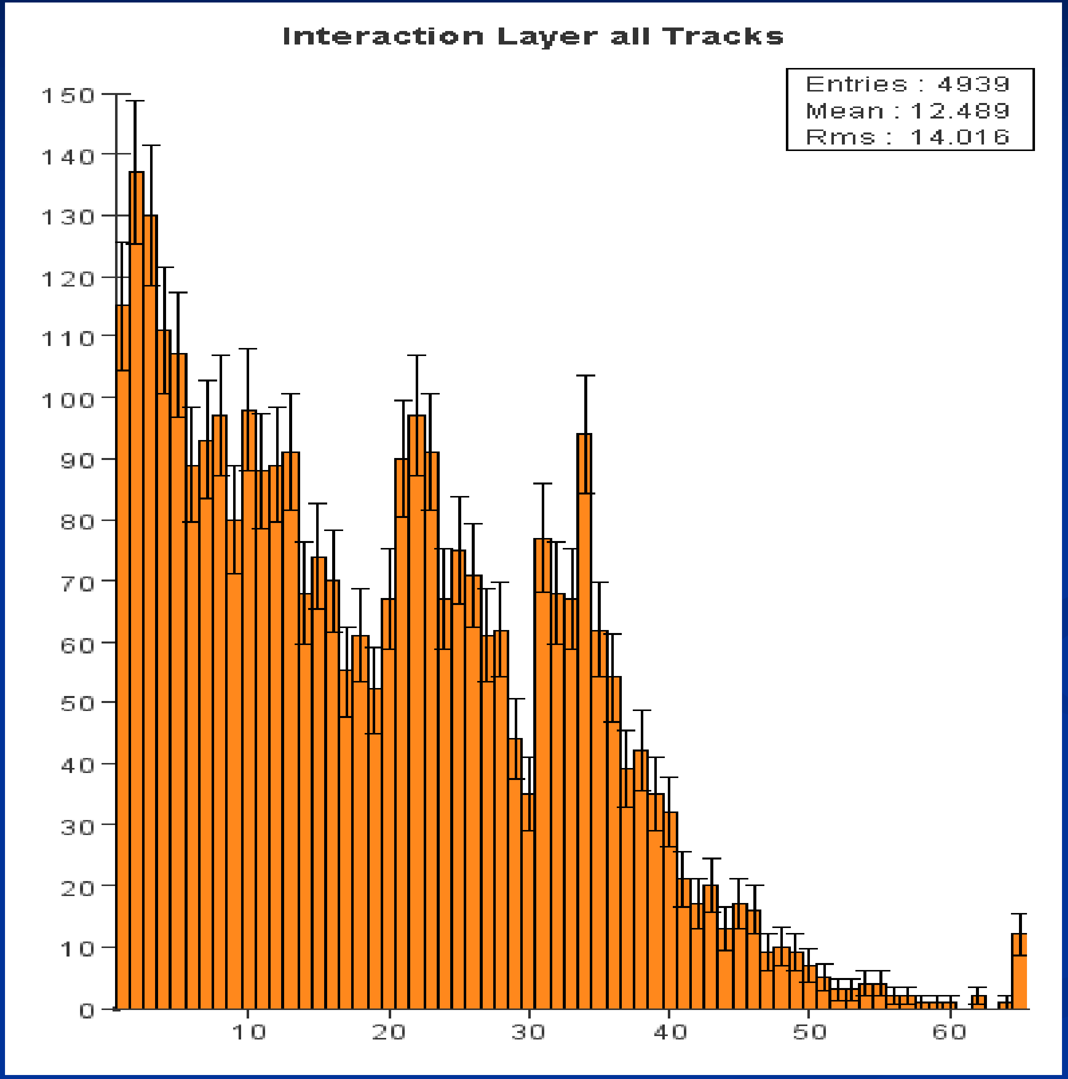
Cluster Pointing :

Determine pointing vector
-> Match ILSP pointing clusters to track

Shower reconstruction :

Optimize matching, iterating in E,HCAL separately (E/p test)

Track-Mip Algorithm – Interaction Layer

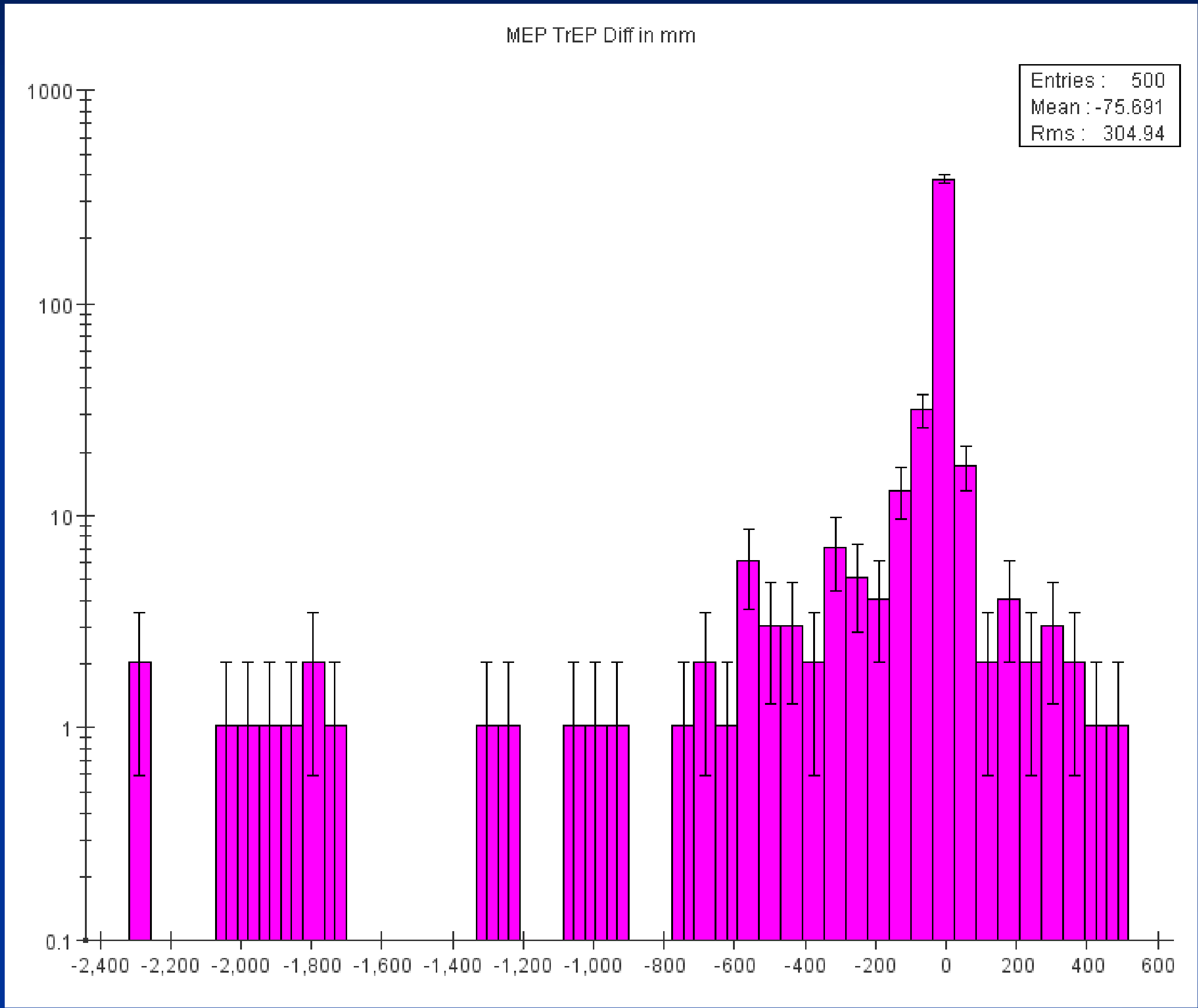


Interaction layer for all tracks

-> exponential behavior for each section of CAL – 20/10 layer ECAL sections and 34 layer HCAL

Also some non-interacting pions

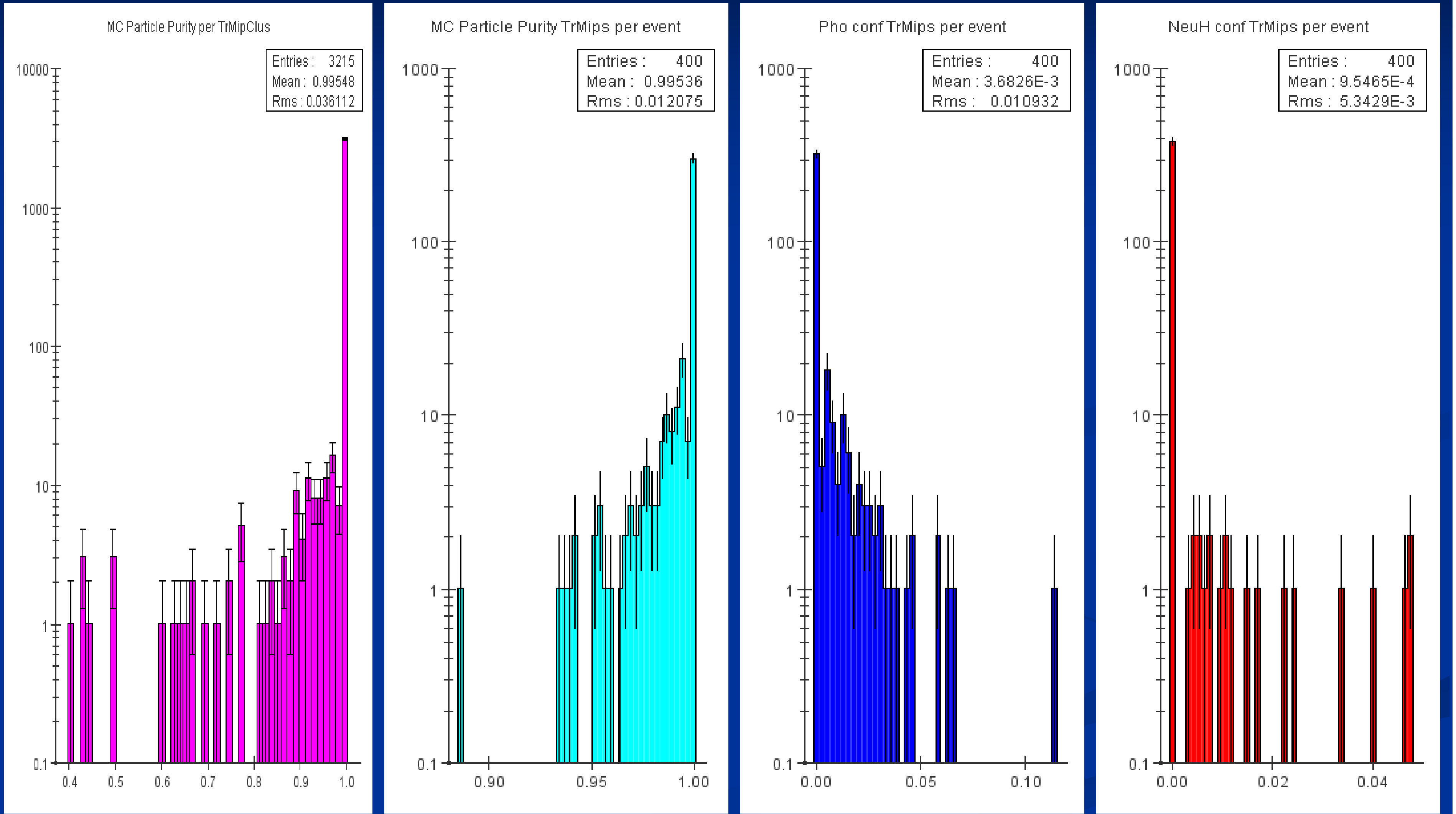
Comparison of Mip Endpoint to MC Track Endpoint



rms = 30 cm

rms90 = 4.5 cm

MIP Finder Performance in qqbar100 events



Track-Mip Algorithm Summary

The Track-Mip Algorithm associates hits to a track with almost no loss in purity

In simulated physics events, values of the purity of the found mip clusters are typically >99%

In addition, the interaction point of charged hadron showers is also obtained – 90% occuring within 5 cm of the MC track endpoint

As a standalone program, this algorithm can be evaluated and tuned with test beam data

Plans are to produce a C++ version of the algorithm that can be used in the MarlinReco framework

Cluster Pointing Algorithm

Cluster hits with DT clusterer – 4 hit minimum for principal axes determination

- plan to test other cluster algorithms

Compare cluster pointing direction to IL spacepoint direction and IP direction :

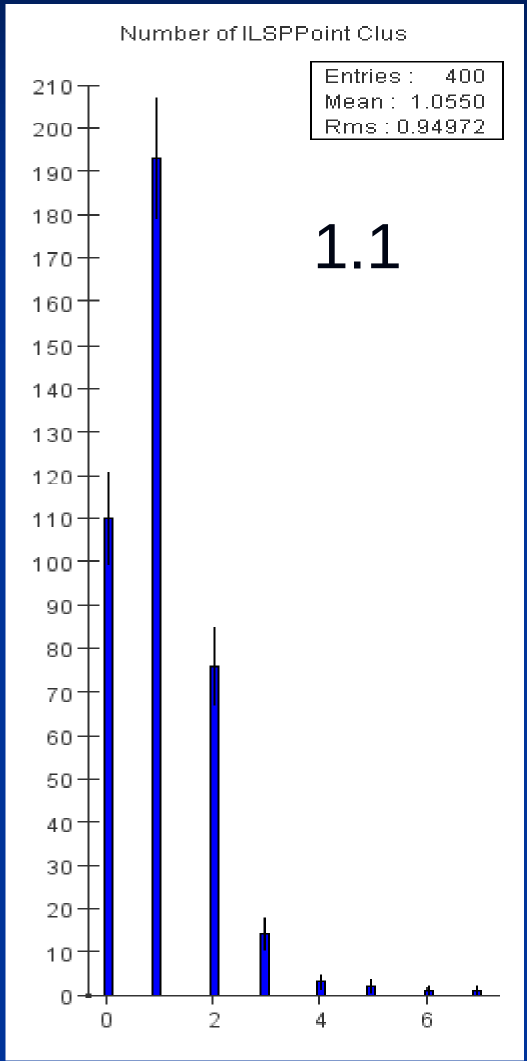
If IL spacepoint comparison < IP comparison -> ILSP Cluster

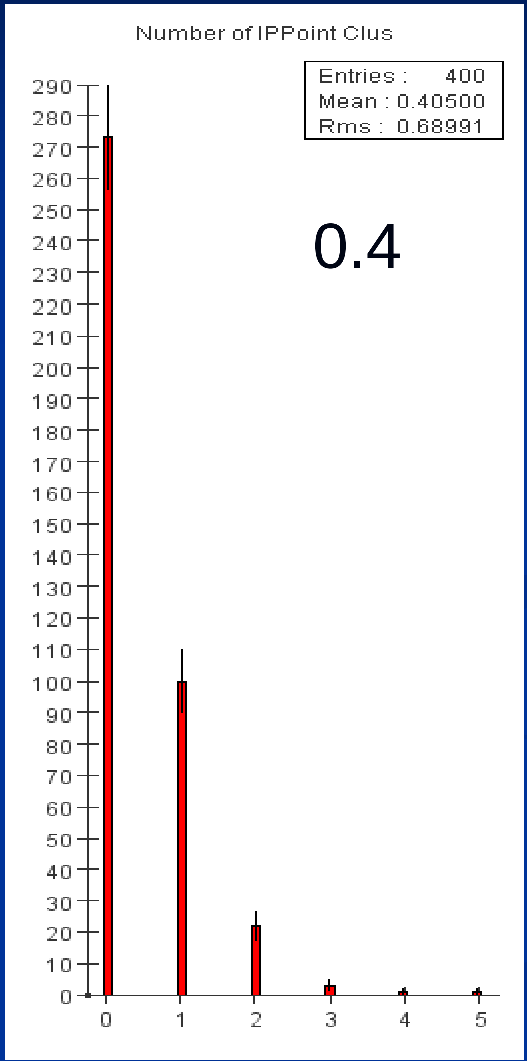
Else if IP direction comparison small enough -> IP Cluster

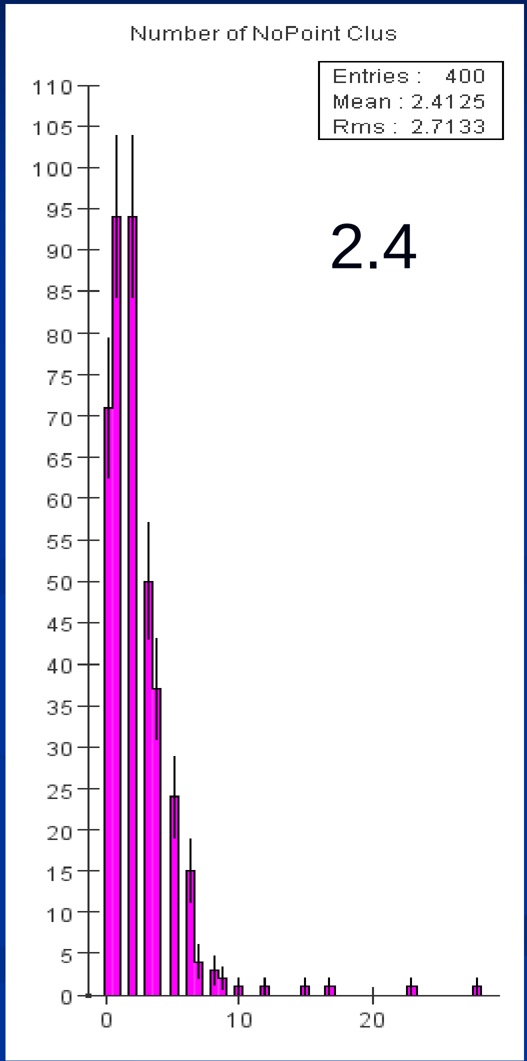
Else NP (non-pointing) Cluster

Do cluster fragments of charged hadrons point to the interaction point?

Single pion cluster pointing results

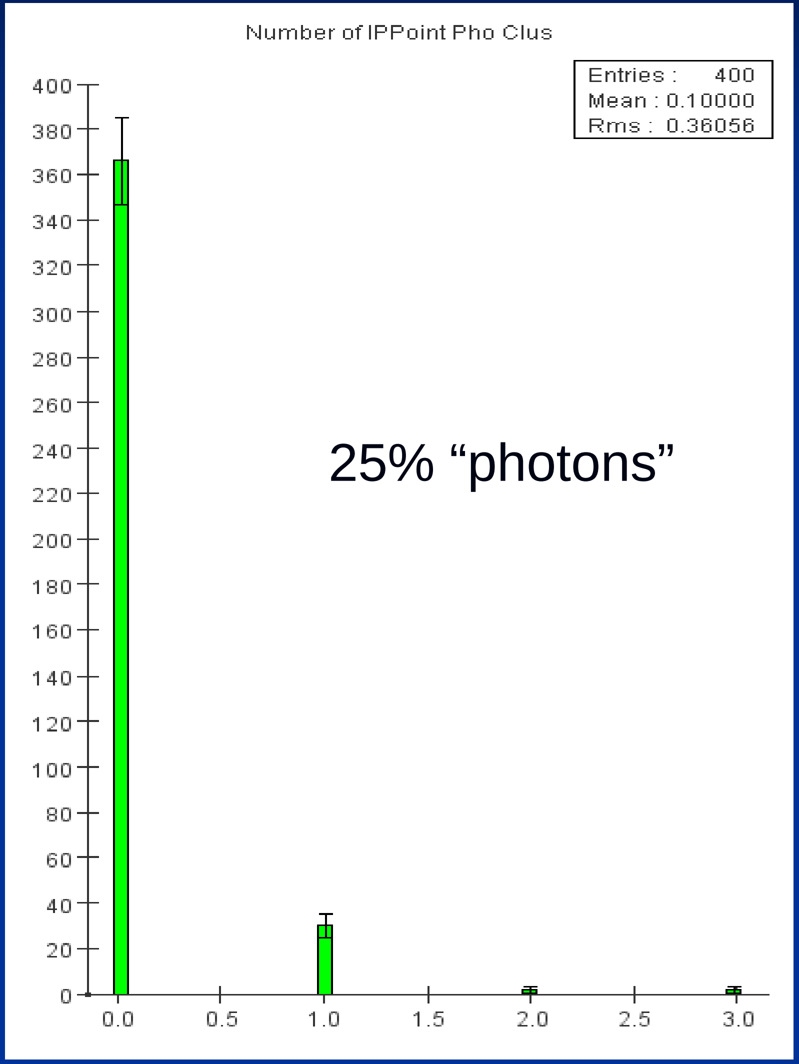


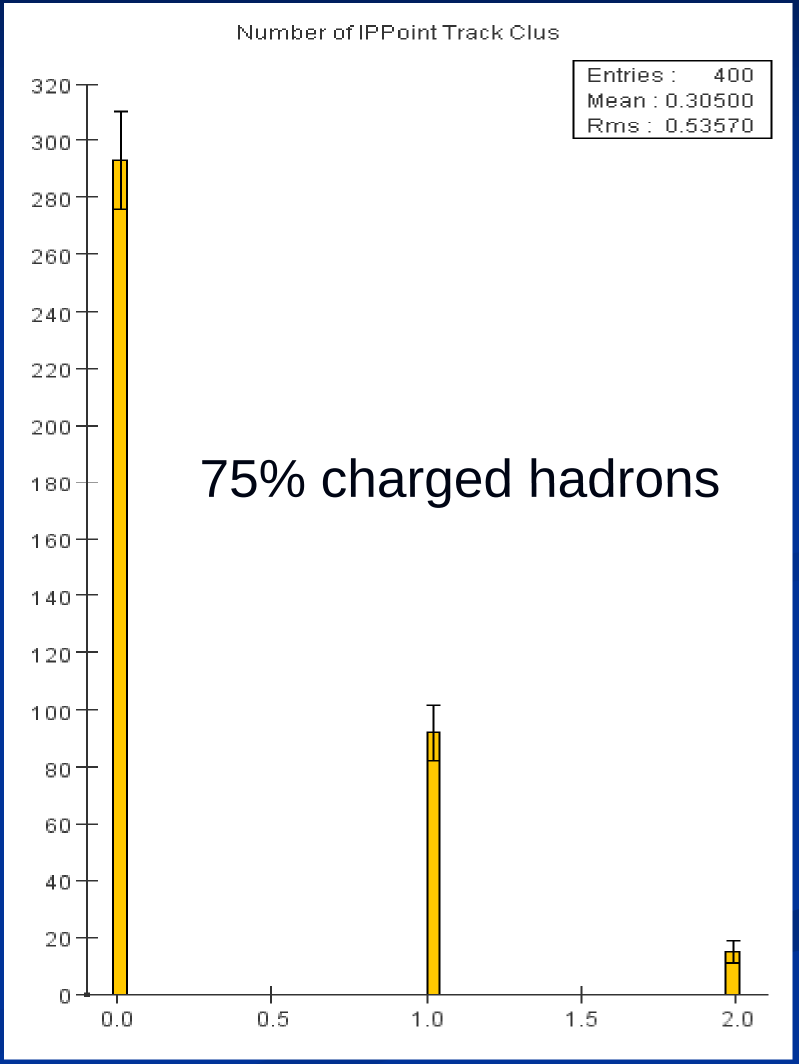




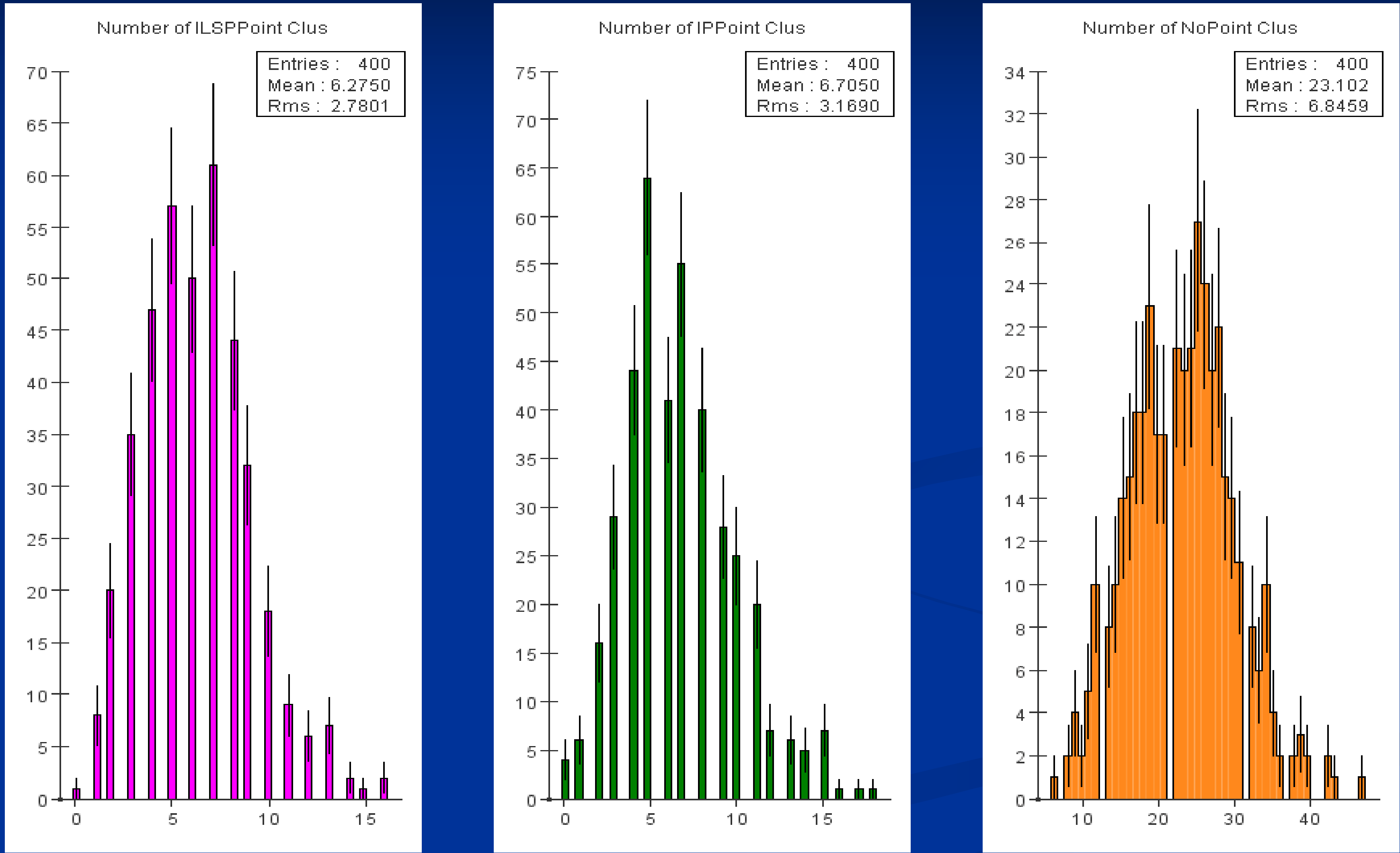
DT Clustering with 4 hit minimum, after mip finder, 1-10 GeV pions, 4-176 degrees

IP Cluster subdivision

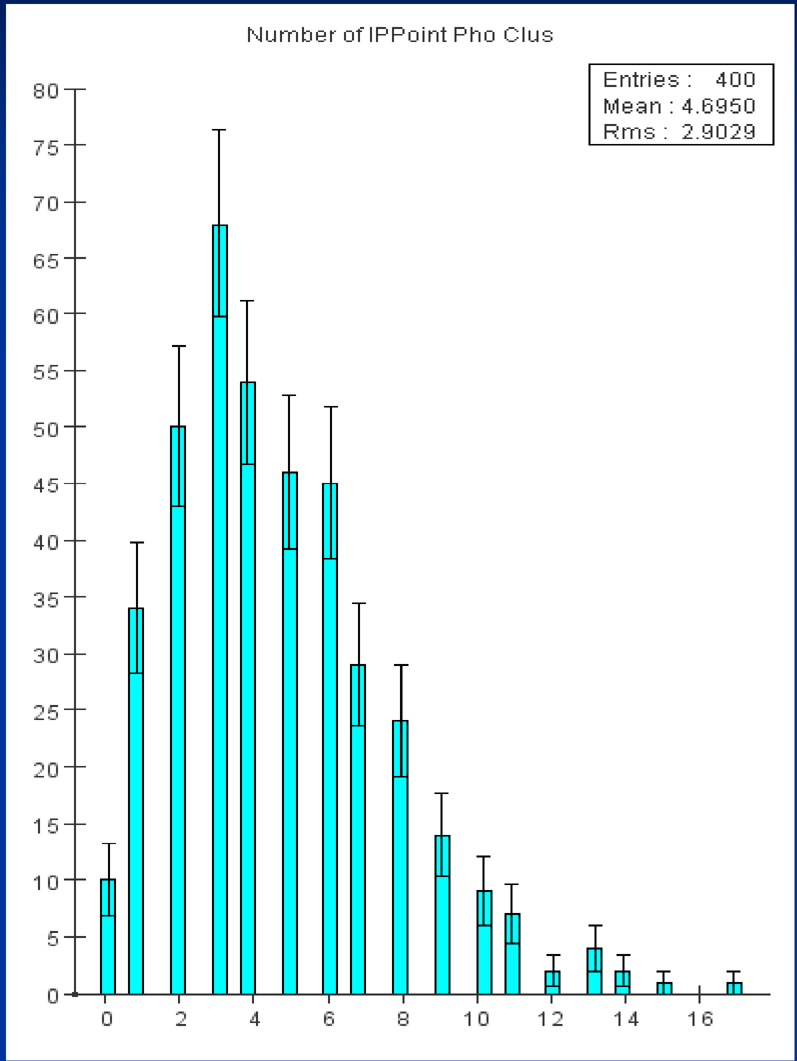


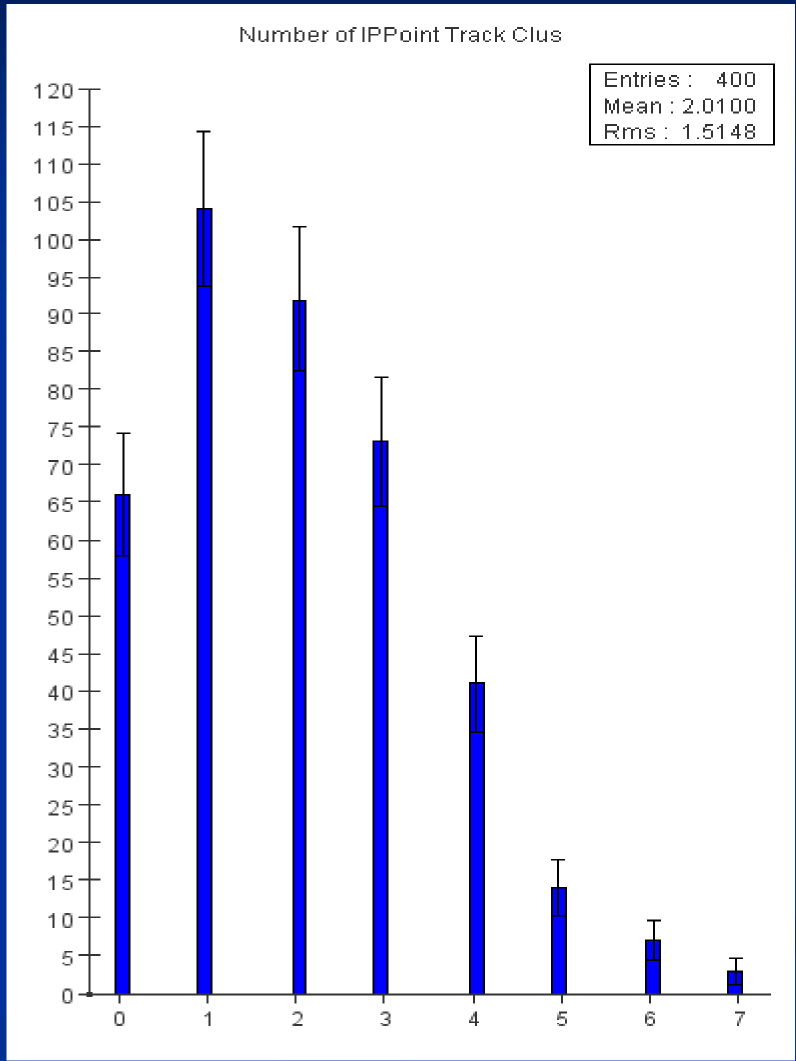


Cluster pointing results in qqbar100 events

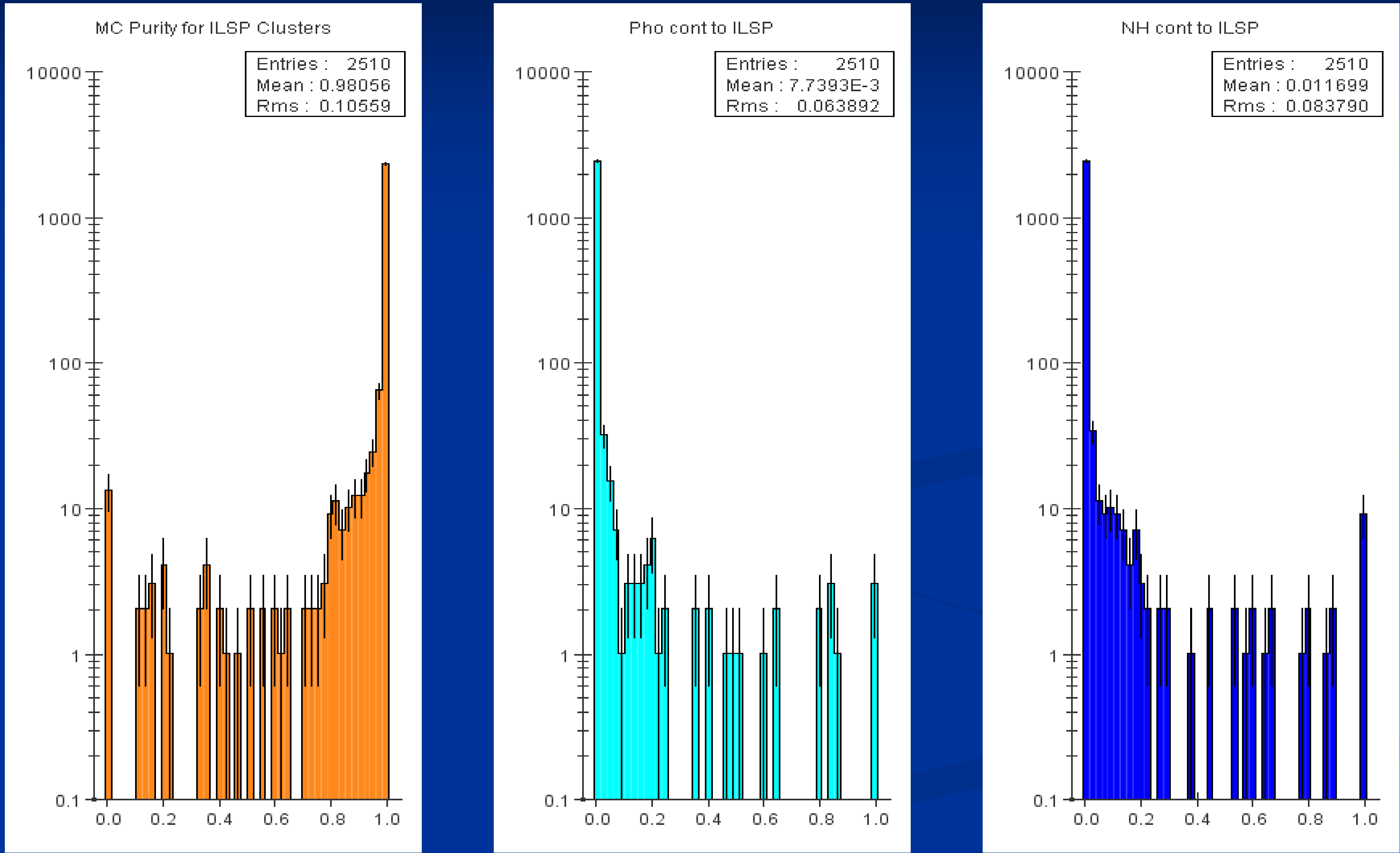


IP Cluster subdivision

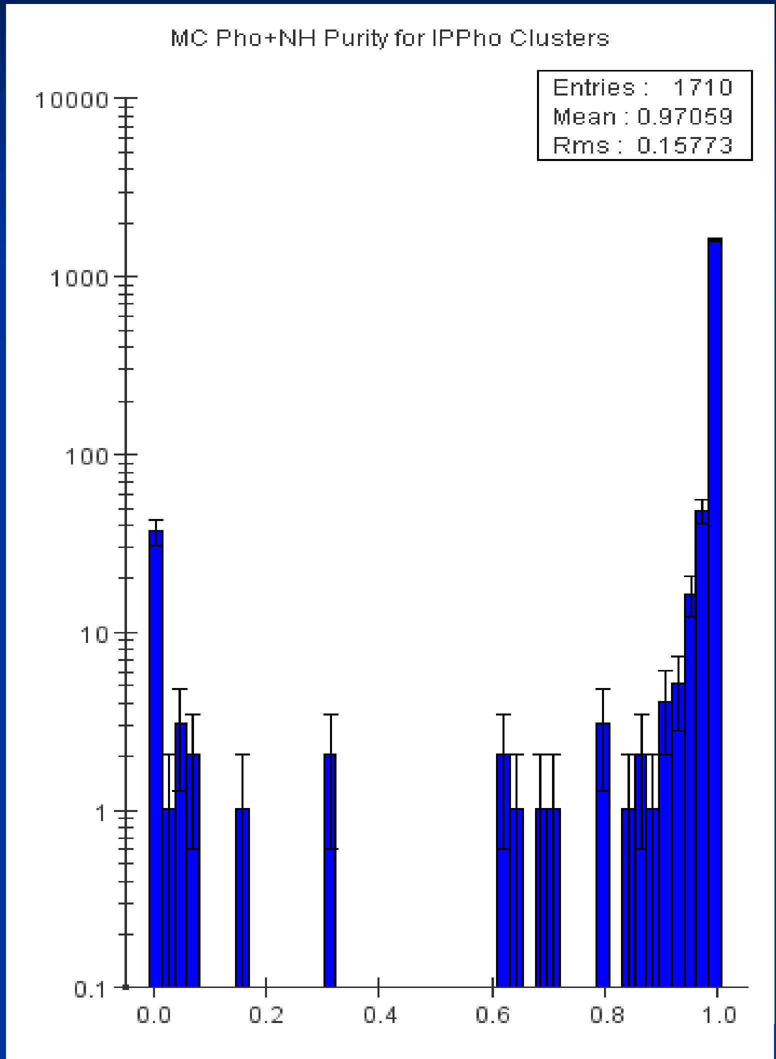


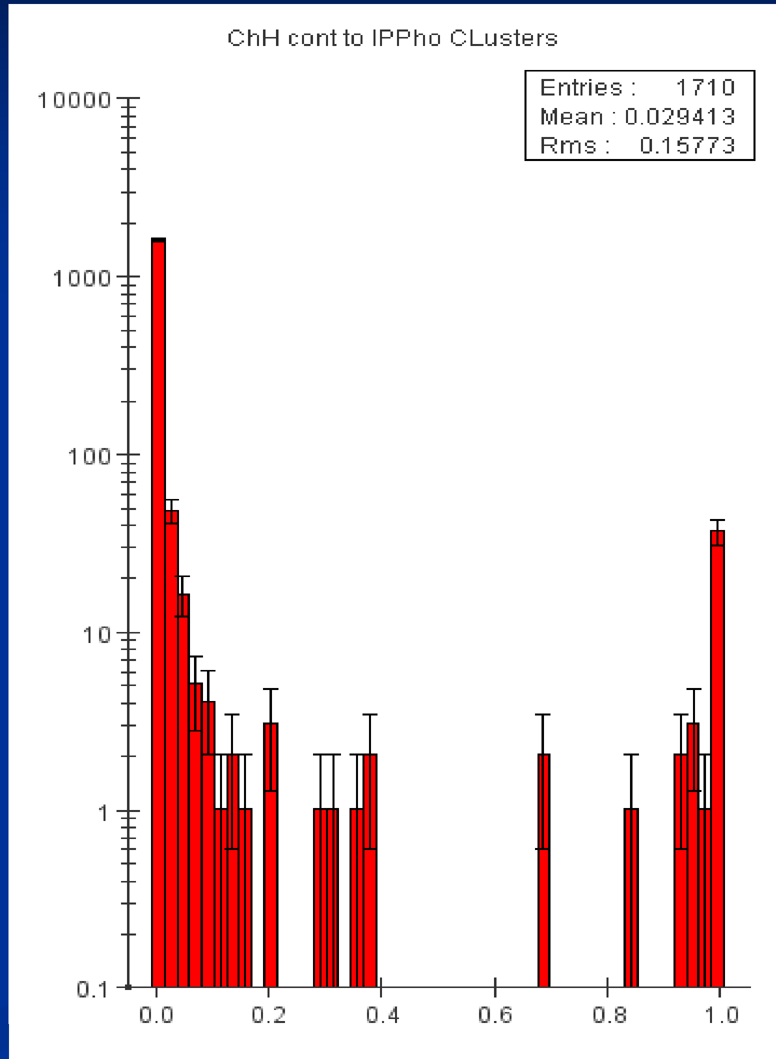


Purity of ILSP Clusters (assume charged hadron)

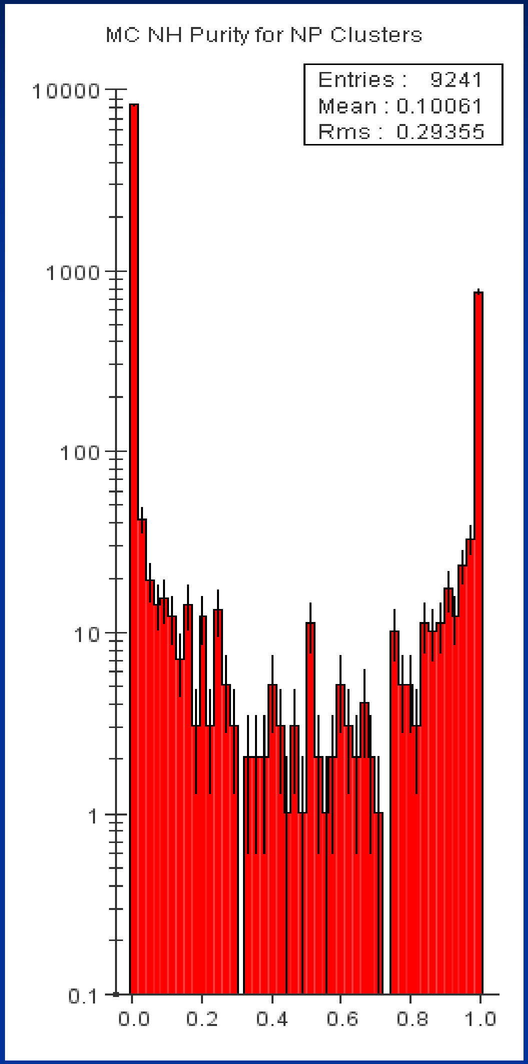


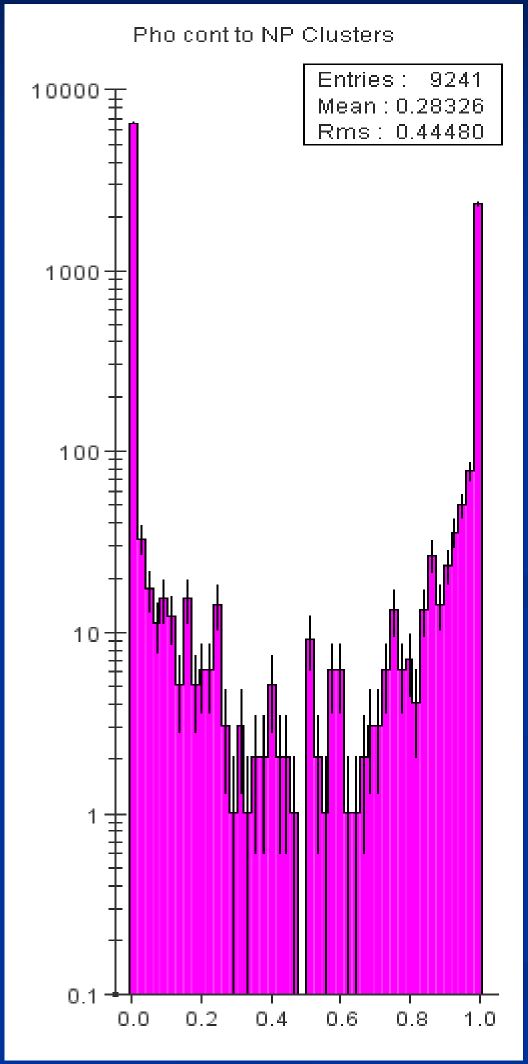
IPPho Clusters (EM only)

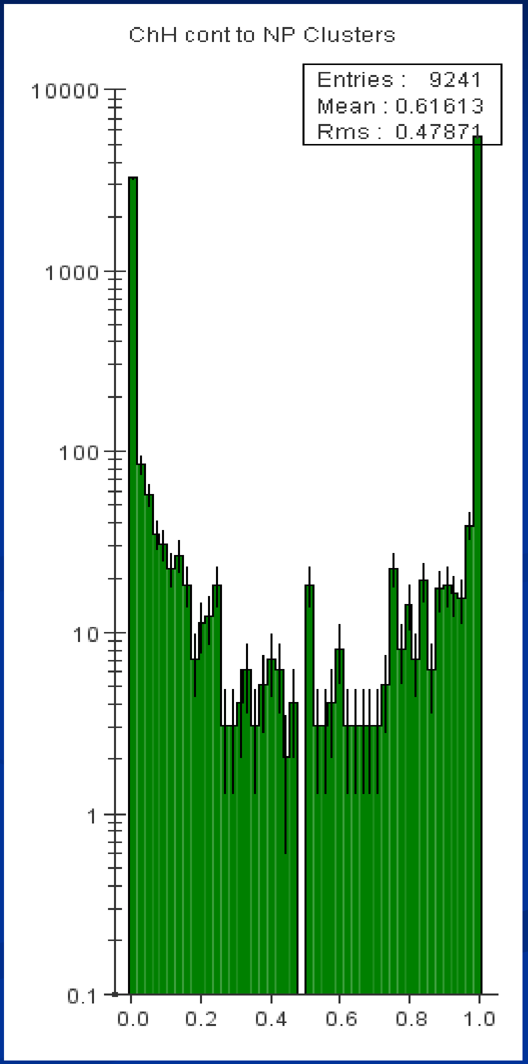




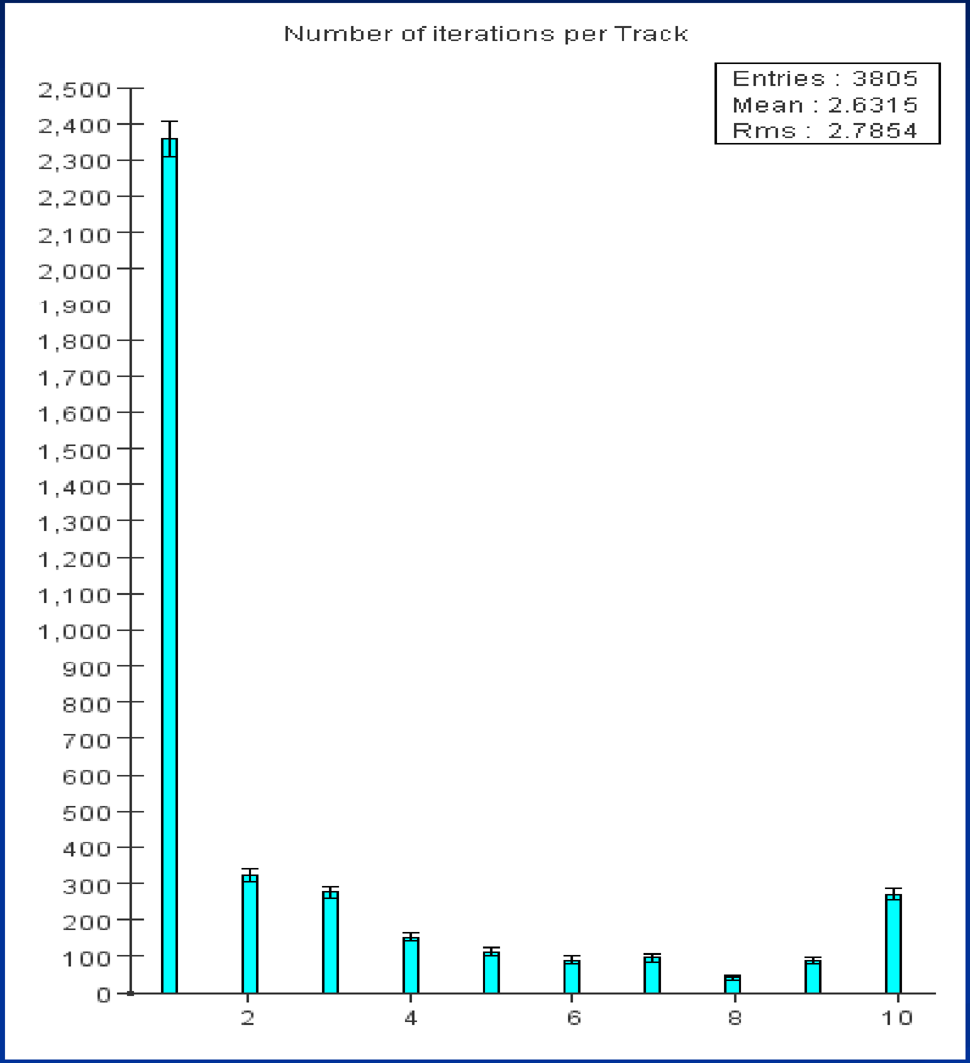
Non-Pointing Clusters





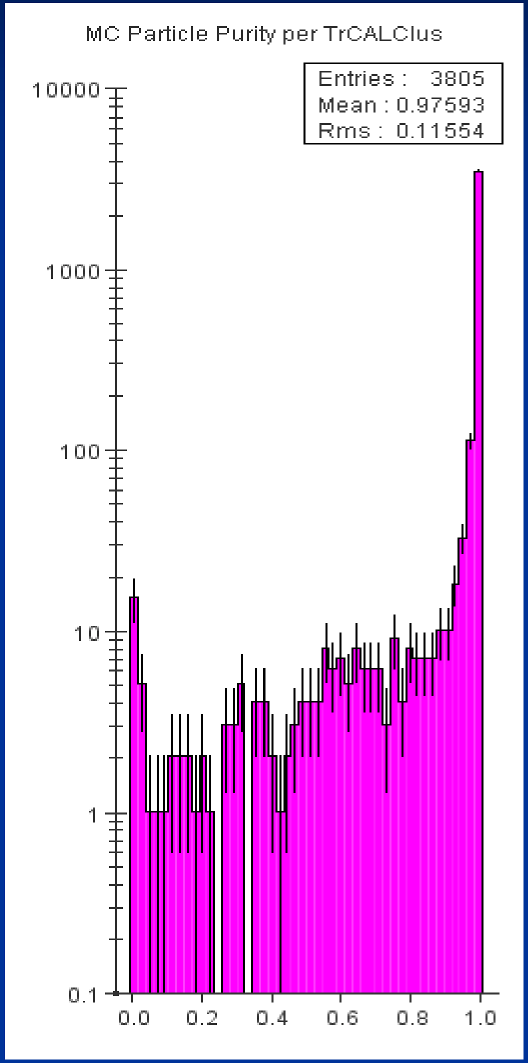


Track-CAL Performance in qqbar100 events

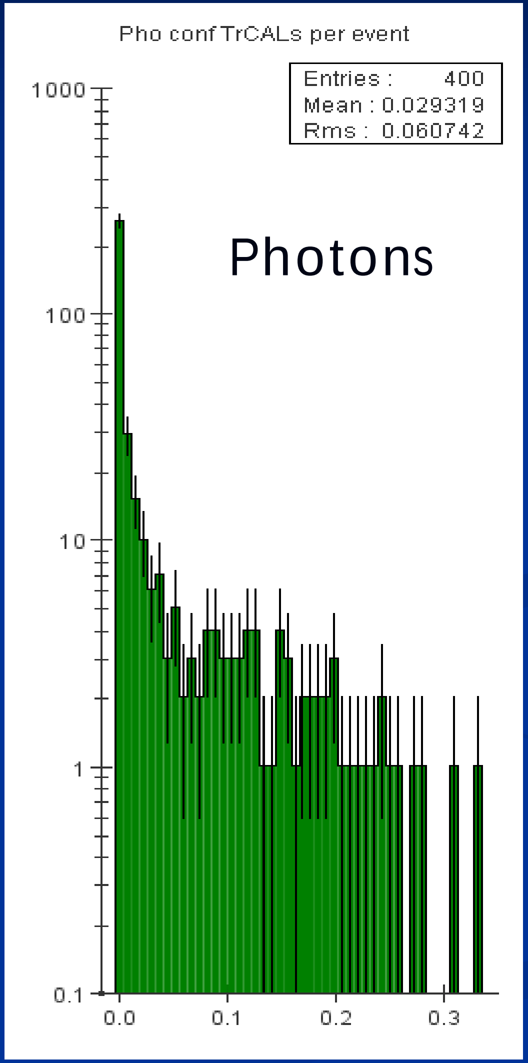


Average size of matched charged hadron cluster – 0.030 ($\Delta\theta$,$\Delta\phi$)

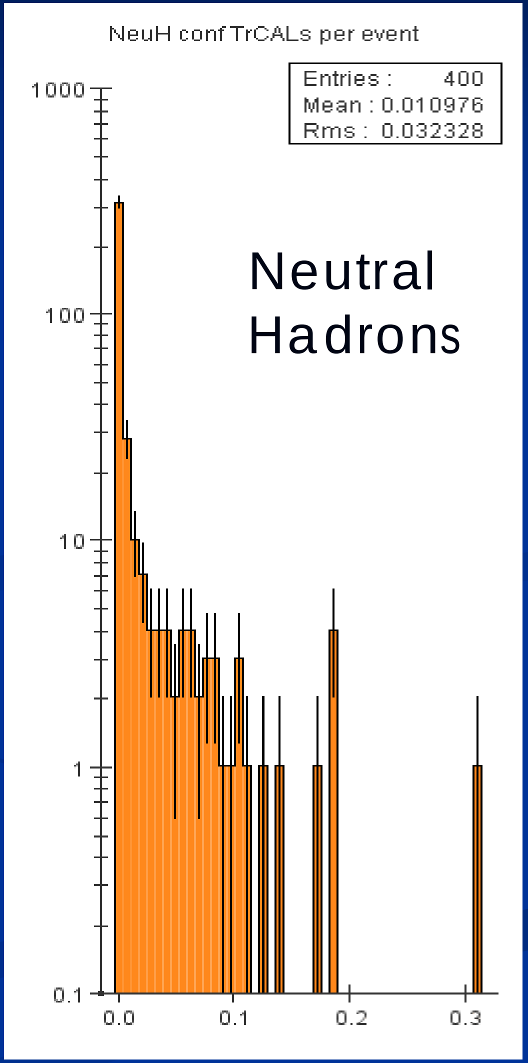
Track-CAL Performance in qqbar100 events



97.6% purity



2.9% contr.



1.1% contr.

Cluster Pointing Algorithm Summary

A cluster pointing algorithm has been developed which, at present, forms 3 classes of clusters, with one being further subdivided into 2 pieces

As tested so far with the DT clusterer, high charged particle purities are obtained for clusters pointing at the interaction layer spacepoint (>98%) – other clusterers will be tested

Also, high photon purities are obtained for clusters which point at the IP and which are not too close to a track – (>95%)

This algorithm, used in conjunction with the Track-Mip Algorithm, can be evaluated and tuned with test beam data

Plans are to produce a C++ version of this algorithm which can run in the MarlinReco framework.

Summary

The PFA Template approach lends itself to the development of modular cluster and particle ID algorithms which and be tested as standalone programs in test beam. The cluster pointing algorithm is probably sensitive to the choice of hadron shower models in the simulation, so it is important to test it with real data. Plans are to produce C++ versions of the algorithms discussed in this talk, and also for any other algorithms use in the Template for which test beam data is now or will become available.