ECAL Studies in ND-GAr using AnaTrees: Electrons and Muons

Vivek Jain, SUNY Albany

Sep 13, 2021

Many thanks to Eldwan Brianne for implementing the geometry, Leo for generating the sample

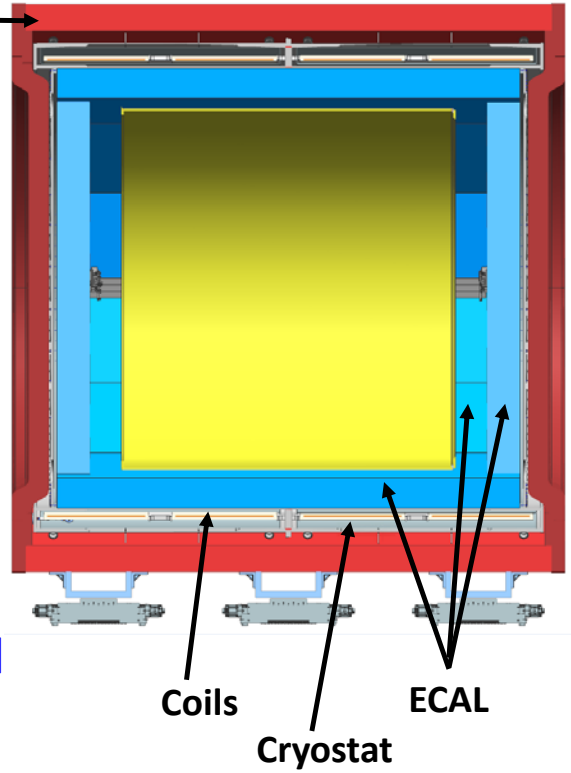
Introduction

- If we want to look at resolution of the CALO for electrons, we need to compare the energy of the reconstructed electron when it enters the CALO with the energy of the cluster(s) that have been “back-tracked” to the truth electron particle
  - Will use the TrajMCPE variable, i.e., energy at the last trajectory point before the CALO

- First few plots are looking at BackTracker information with only fiducial volume cuts

Latest Geometry

- New ECAL geometry –
  - 42 layers of Pb-Scintillator sandwich (~ 10.5 X_0)
    - Scintillator: 8 layers, each 0.5 cm thick, and 34 layers, each 1 cm thick
    - Pb – 8 layers, each 0.7 mm thick, 34 layers, each 1.4 mm thick
    - I believe endcaps are 6 + 36 layers
  - Barrel has 12 fold symmetry
  - Newly optimized SPY magnet and cryostat as the pressure vessel
    - No extra material between the ECAL and TPC



TPCRadius = 273 cm, TPCLength = 259

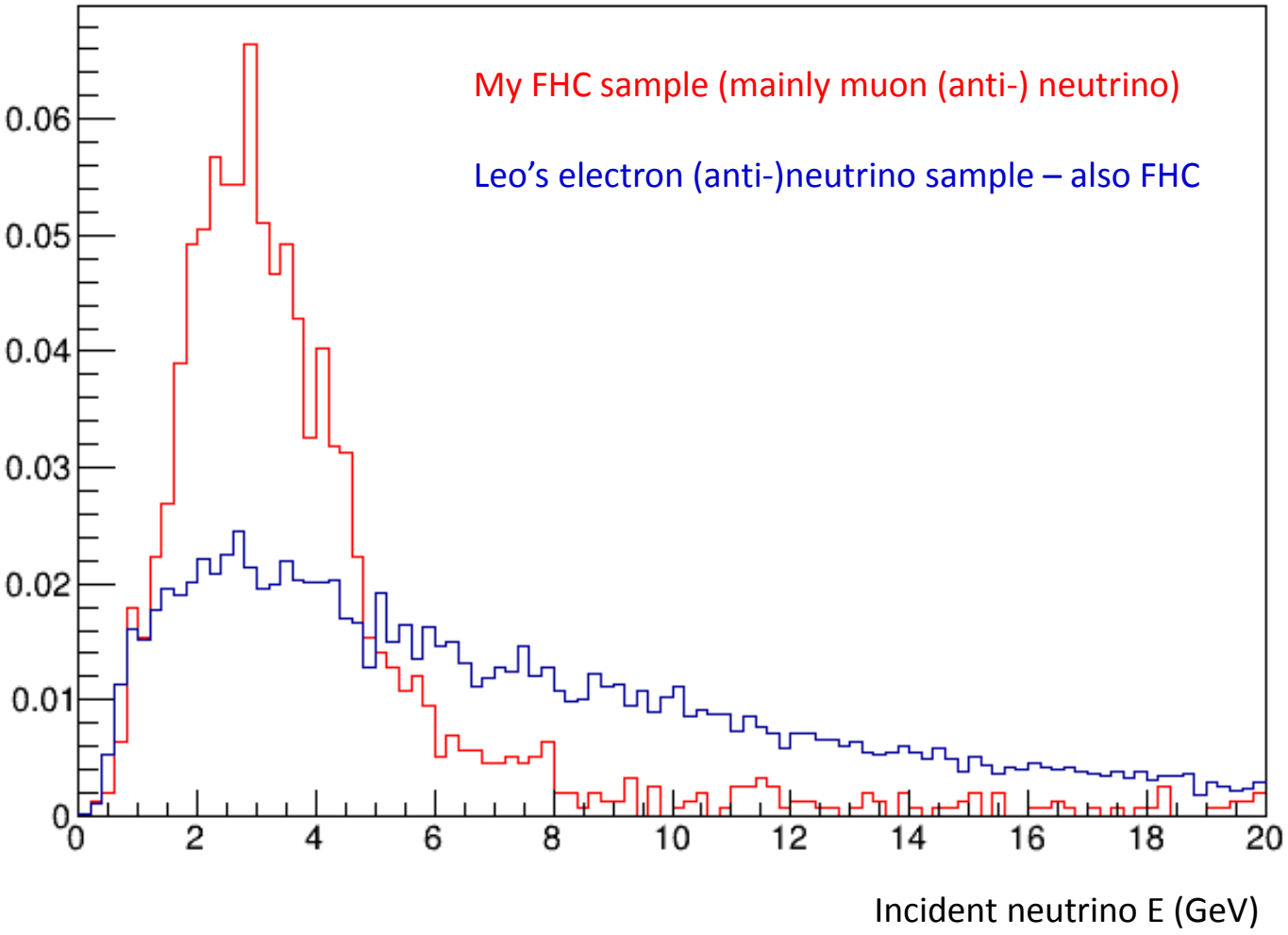
TPCFidRadius = 222.5, cm TPCFidLength = 215

ECALInnerRadius = 278, ECALOuterRadius = 334 cm

ECALStartX = 328 , ECALEndX = 375 cm

Samples

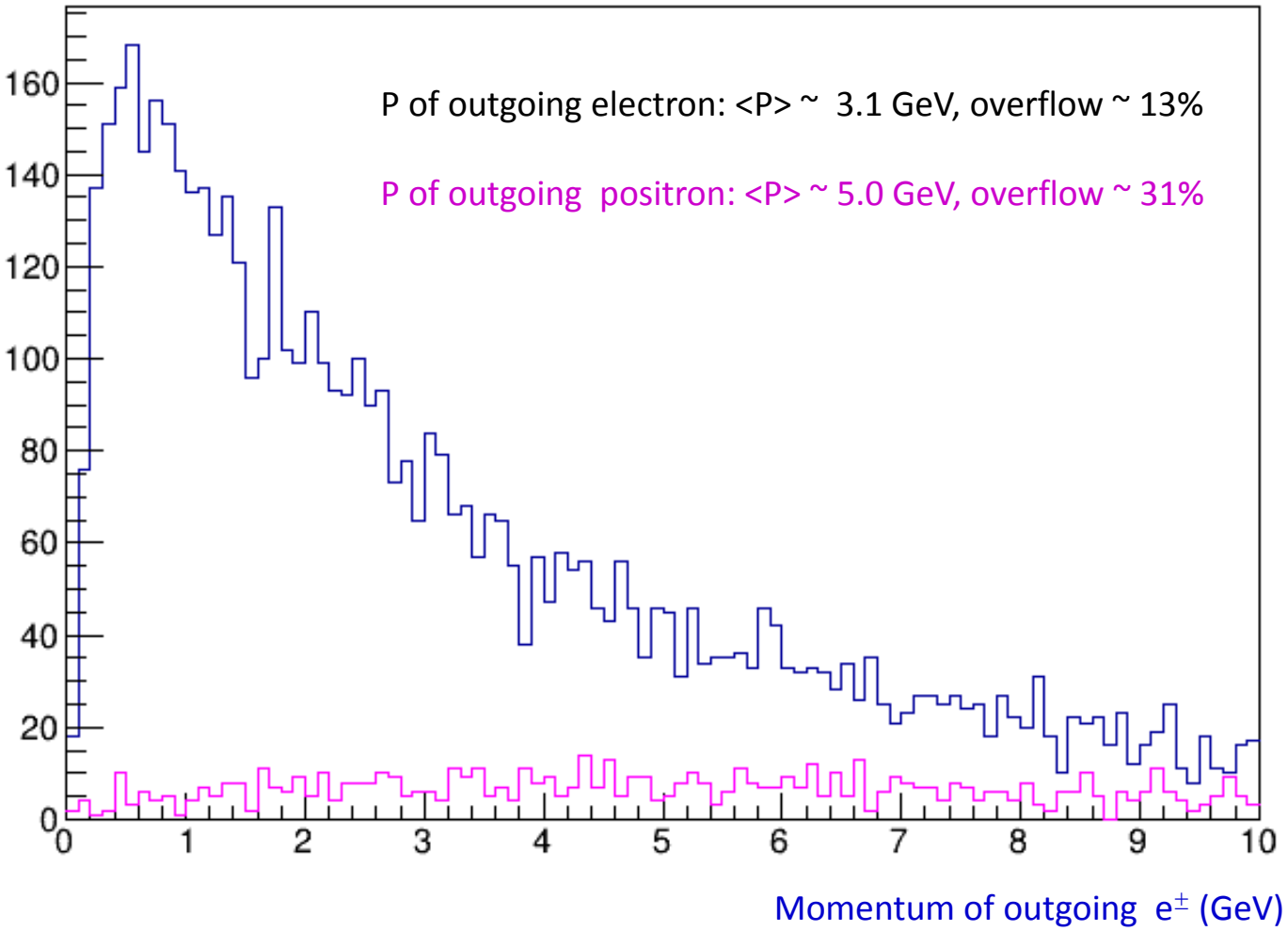
- Leo generated a sample of (FHC) electron neutrino events using GArSoft, so GENIE v3.0.6 (and not using edep-sim)
  - For comparison with muons, I used my locally generated FHC sample (GENIE v3.0.6 and no edep-sim)
    - This is mainly muon (anti-) neutrinos (see backup for details)
- Look at cluster information, and how well clusters match the truth particle (latter uses BackTracker information)
- **All plots in slides are for neutrino interactions in TPC fiducial volume**



Muon neutrinos come from charged Pion and Kaon decays,

whereas electron neutrinos primarily come from charged Kaon decays

Maybe that explains the difference



Leo's electron sample

CC nu_e: 6670 events
CC anti-nu_e: 944 events

NC: 2741 events

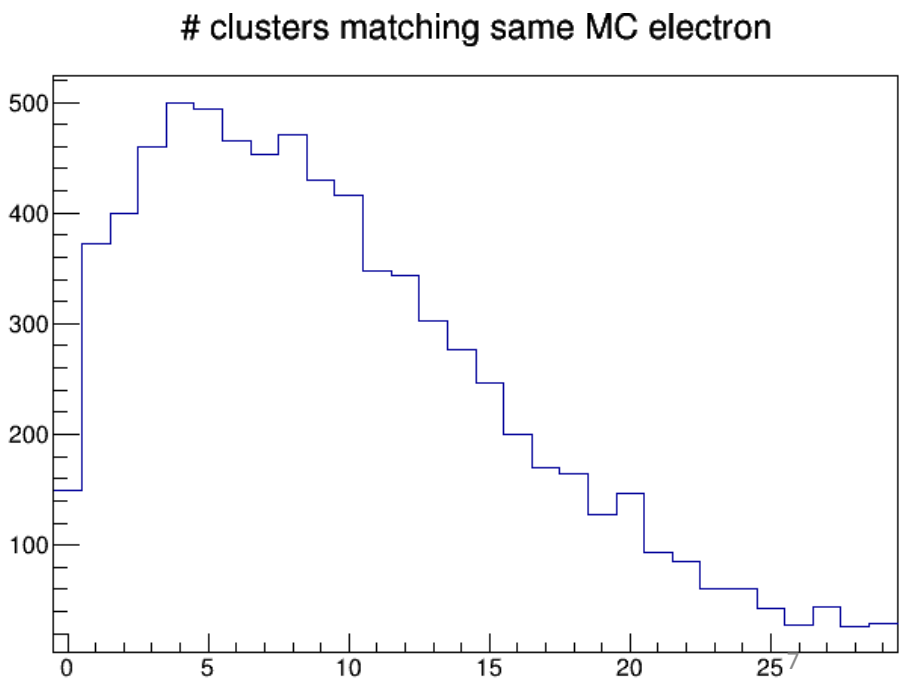
15 di-electron events in sample

Upcoming plots will be for primary electrons and positrons

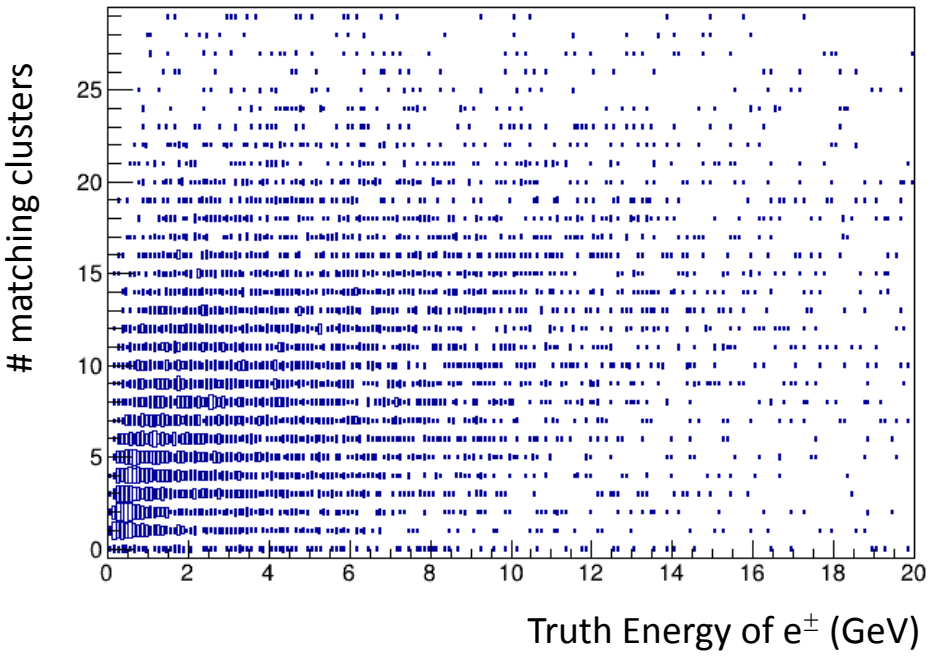
(for comparison, will also use primary $\mu^{\pm}$)

CALO Cluster and BackTracker information – electron sample

- BackTracker algorithm connects reconstructed objects back to truth particles
  - Look at CALO clusters – ClusterMCIndex variable gives the index of the MCParticle that the cluster is posited to be coming from
  - Require that the MCParticle corresponding to **primary electron/positron stops in or goes through the CALO**
    - If ending rad > 278 cm or |x_end| > 328, assume track stops in/goes through CALO

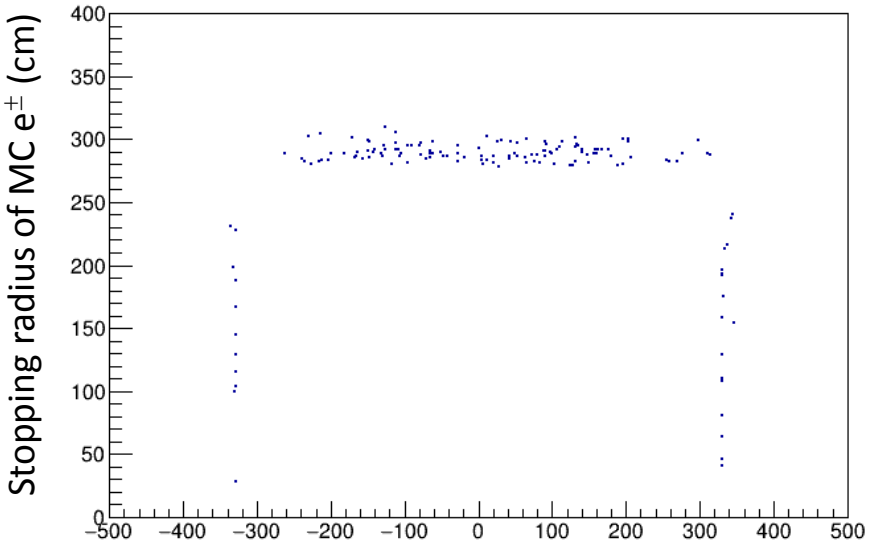


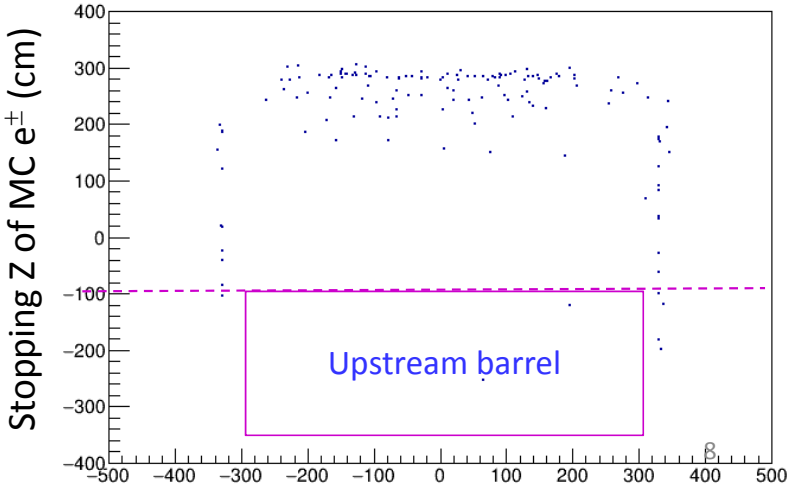
- 2% MC $e^{\pm}$ have 0 matching clusters??
- 5% MC $e^{\pm}$ have 1 match
- 93% of MC $e^{\pm}$ match more than 1 clusters?
  - Why so many matches? Is this due to the strip splitting algorithm used to find hits?



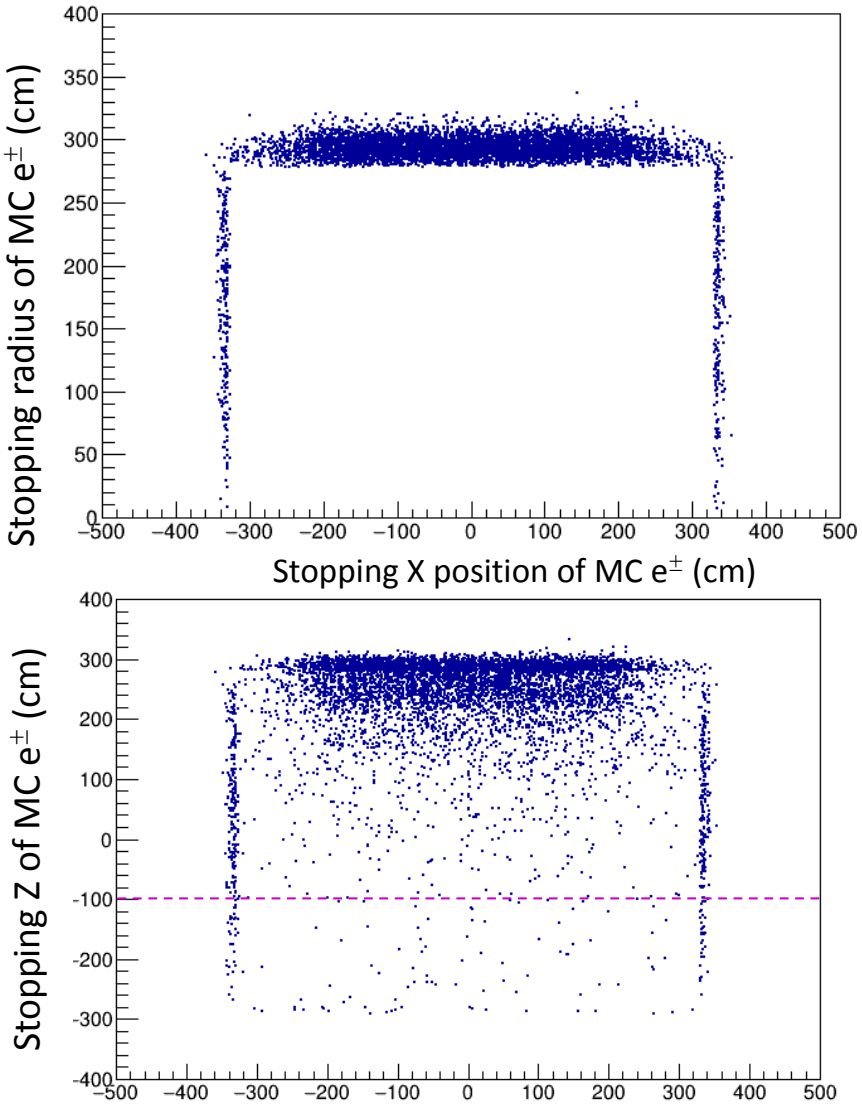
Electron sample

0 cluster matches – stopping position





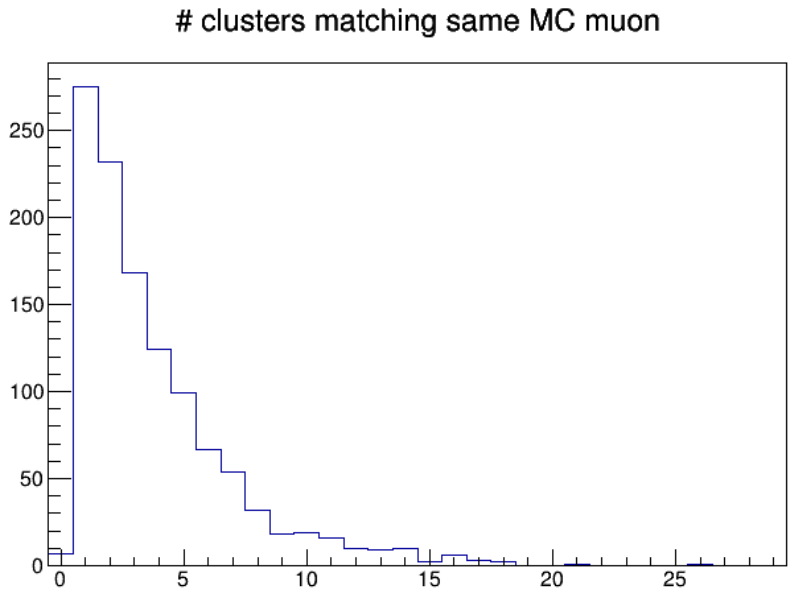
≥ 1 matching clusters - stopping position



Electron sample

	# with 0 cluster match	# with ≥ 1 cluster match
Total # MC e±	149	7350
# stopping in Endcap	28 (19%)	493 (7%)
# stop in upstream barrel	2 (1%)	86 (1%)
# stop in downstream barrel	119 (80%)	6771 (92%)

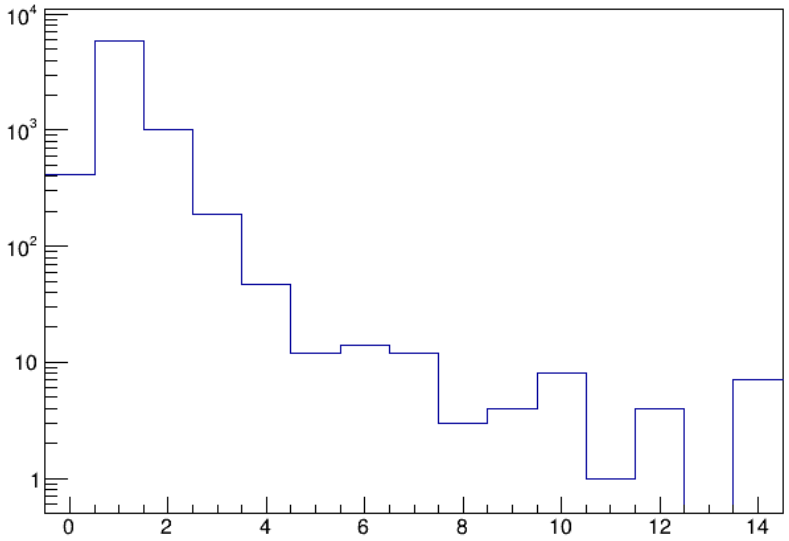
What do muons in CALO look like?



$\leq 1\%$ MC $\mu^{\pm}$ have 0 matching clusters
24% MC $\mu^{\pm}$ have 1 match
75% of MC $\mu^{\pm}$ match more than 1 clusters

(compare with electrons (slide 7) – fewer multiple matches and many more 1 cluster matches)

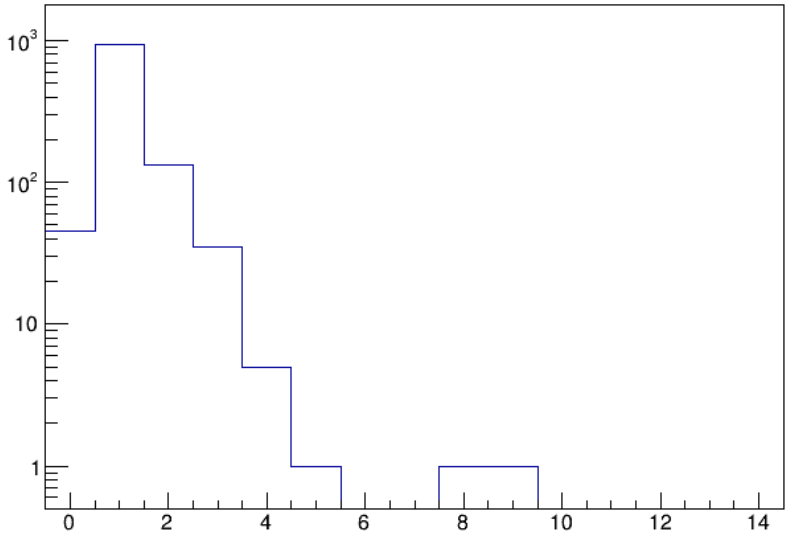
How does the BackTracker perform for TPC tracks?



TPC tracks matching truth primary electron

0-match: ~ 5%
1-match: ~ 76%
≥ 1 match: ~ 19%

This looks more reasonable – still a long tail.



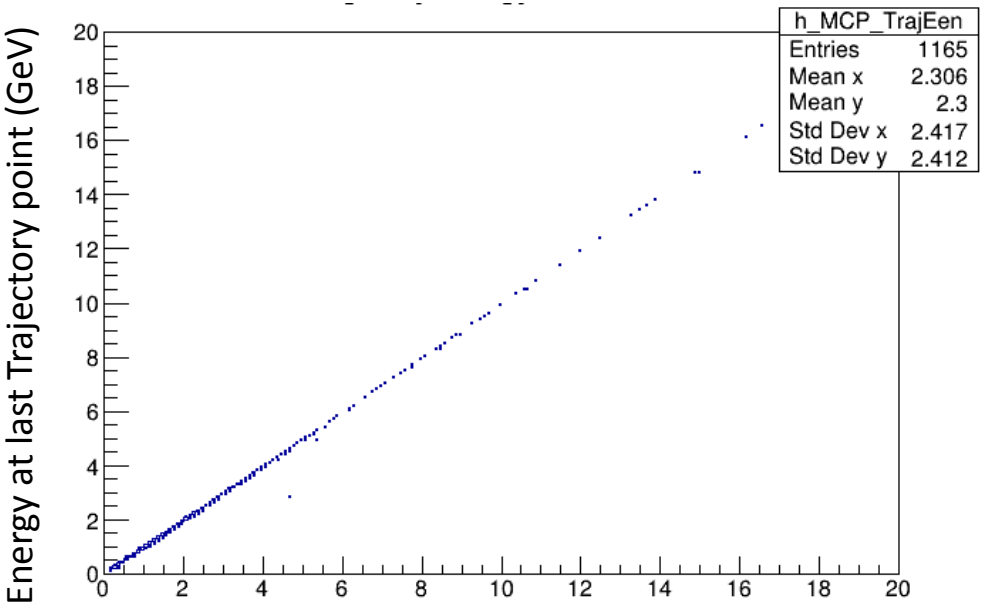
TPC tracks matching truth primary muon

0-match: ~ 4%
1-match: ~ 81%
≥ 1 match: ~ 15%

This looks more reasonable – small tail

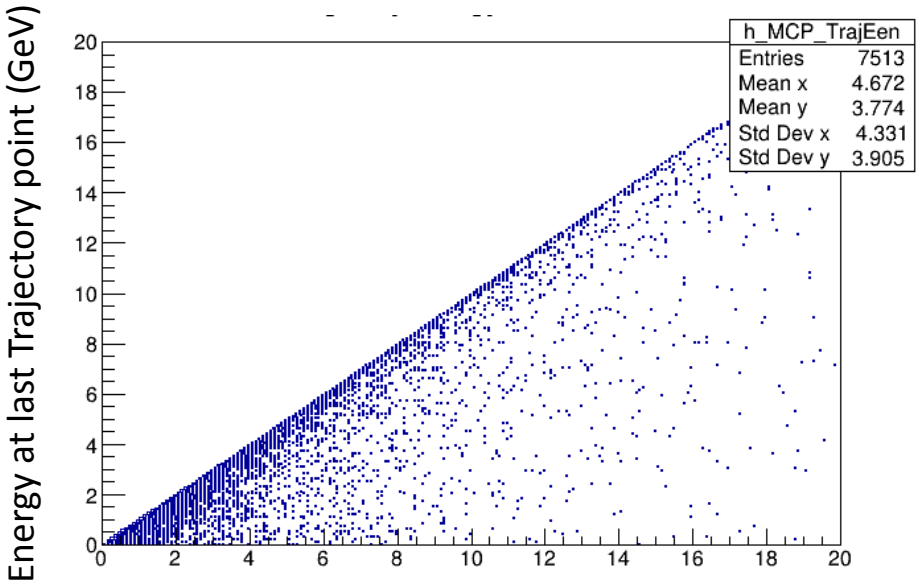
Energy resolutions

- If we want to study electron resolutions in CALO, we will need to compare the Σ(cluster E) with an appropriate value for truth energy
  - Anatrees have TrajMCPP[X, Y, Z] and TrajMCPE variables, the latter being energy, which is what we want.
  - Look at the energy at the last trajectory point before the CALO
  - Require that primary MC truth electron (or muon) stops in/goes through CALO, and that we find trajectory points corresponding to this MC Particle



Muon truth P (Gev)
which is ~ E

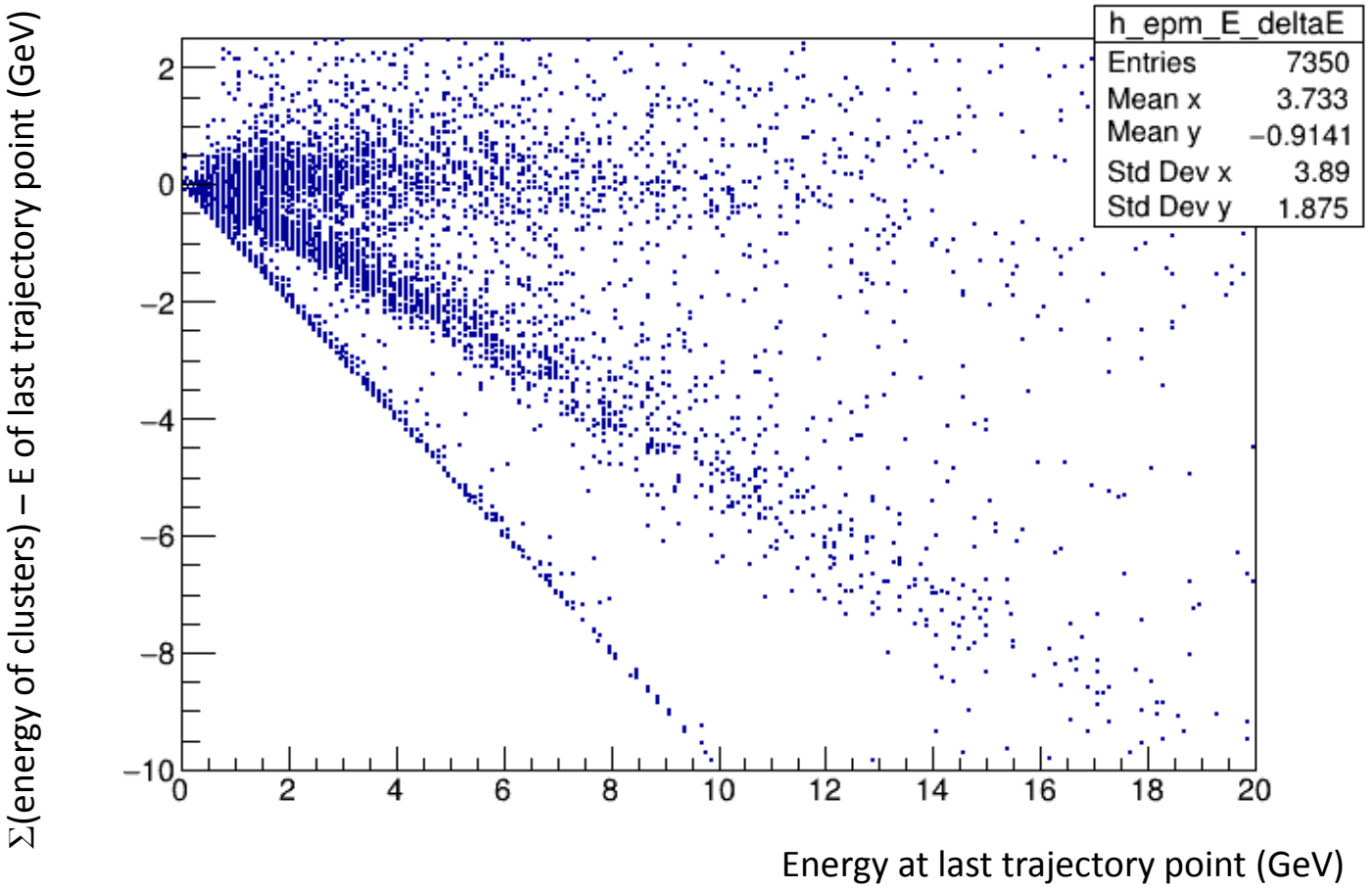
Muons don't lose any energy as they traverse the TPC



Electron truth P (Gev)
which is = Energy

but electrons do lose energy

Here I require that there be at least 1 CALO cluster “back-tracked” to the truth electron



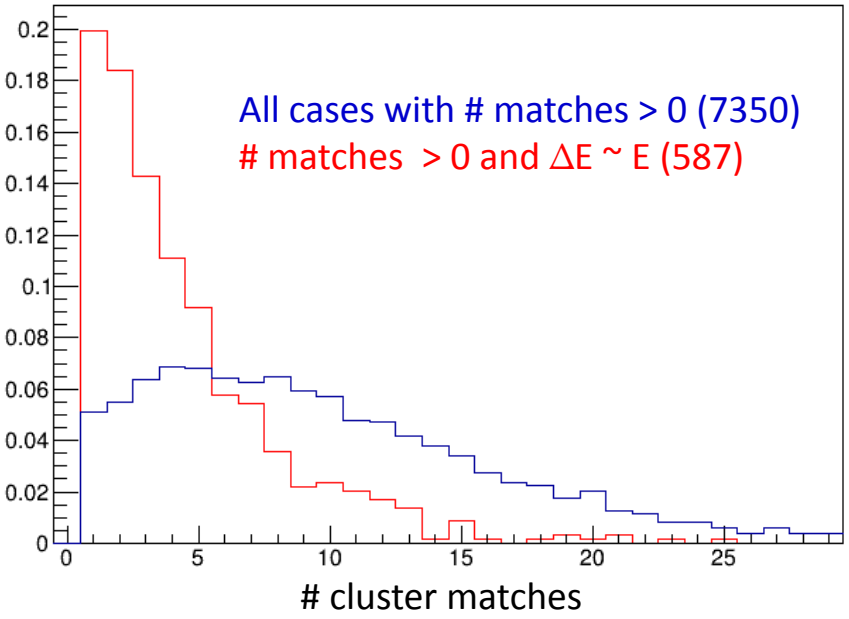
Only for electrons

The lowest band is almost diagonal, i.e., deltaE ~ E

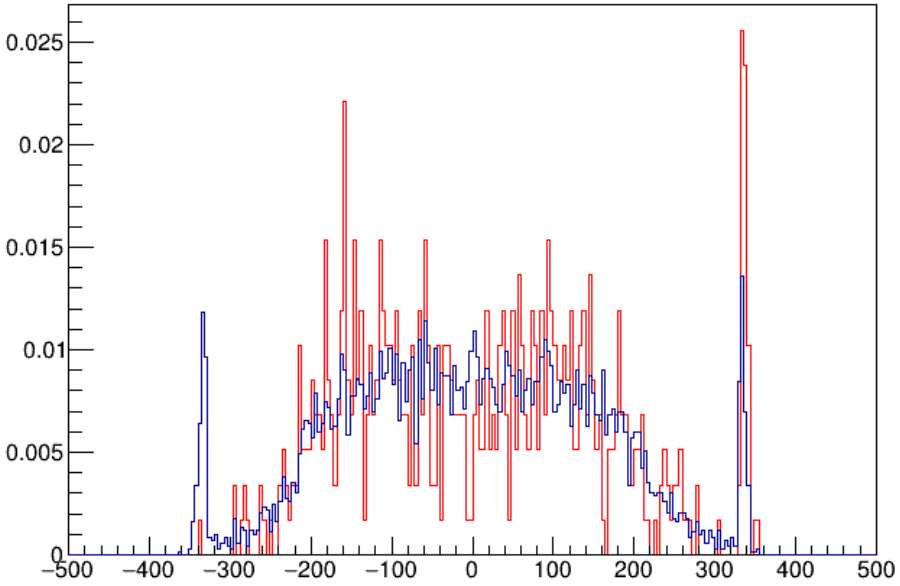
What’s going on?

Corresponding muon plot just has the lowest diagonal band, but that is understandable since muons lose very little energy in CALO.

Looking at cases where ΔE ~ E (electrons) – when E > 0.4 GeV and ΔE < 0.1 GeV

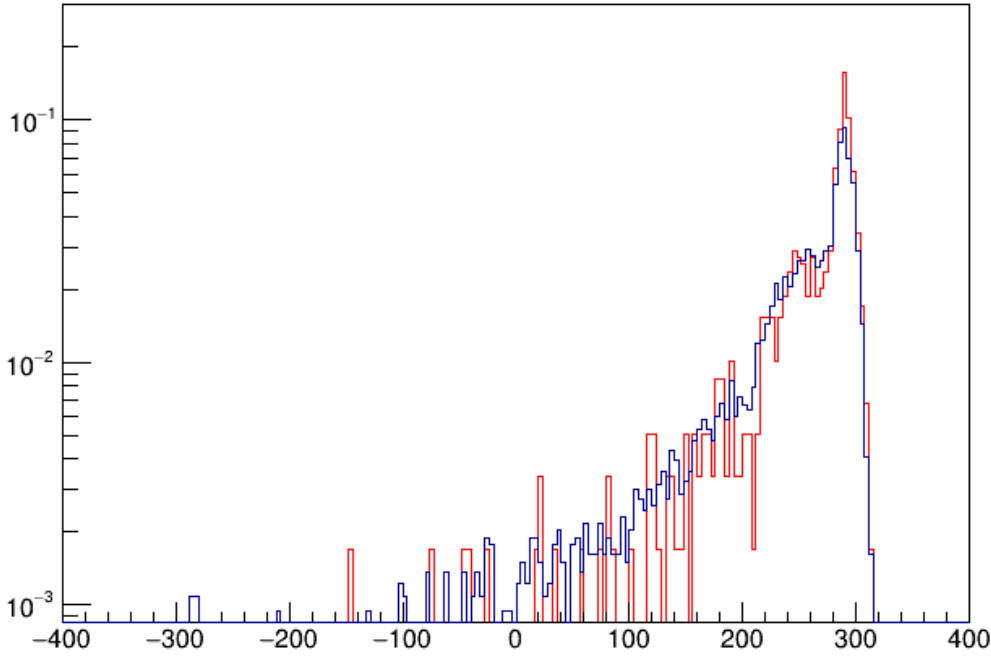


If I make exclusive categories, the averages are 4.0 vs 10.0



Stopping X position of truth electron (cm)

Minor differences in endcaps, and a little bit in the downstream barrel, but nothing too obvious (blue includes the red points)



Stopping Z position of truth electron (cm)

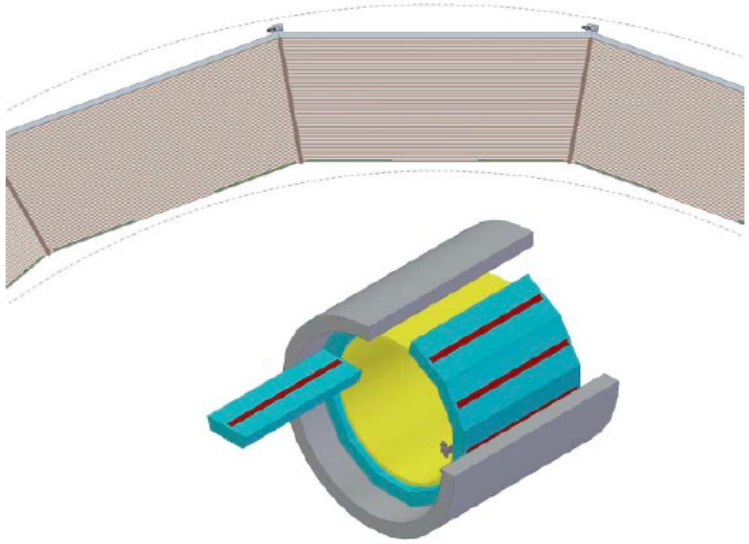
Summary

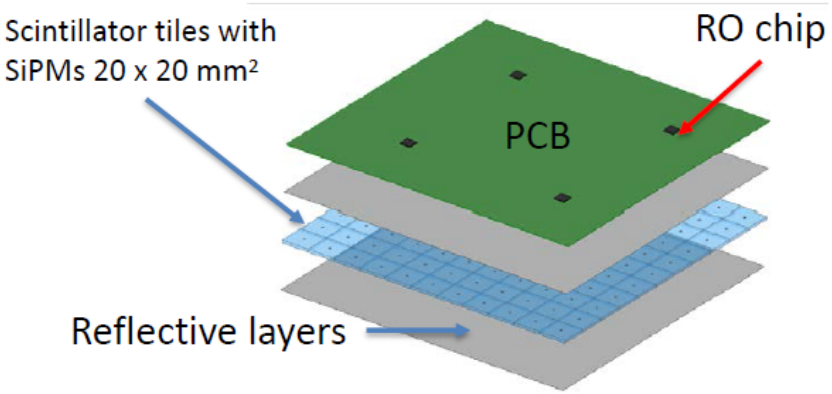
- Why are so many clusters being matched to a truth particle
- Why three bands in the delta E plot (on previous slide)?

Extra –

ECAL: Engineering status

- Based on the ILD design
- Preliminary designs
  - Check ECAL space between TPC/Magnet and put in realistic tolerances.
  - First ideas on how to fix the ECAL
    - Self-supporting
    - Individual rails
  - Module/Layer design: Lead too soft/toxic - most likely in a super-module" made in carbon fiber





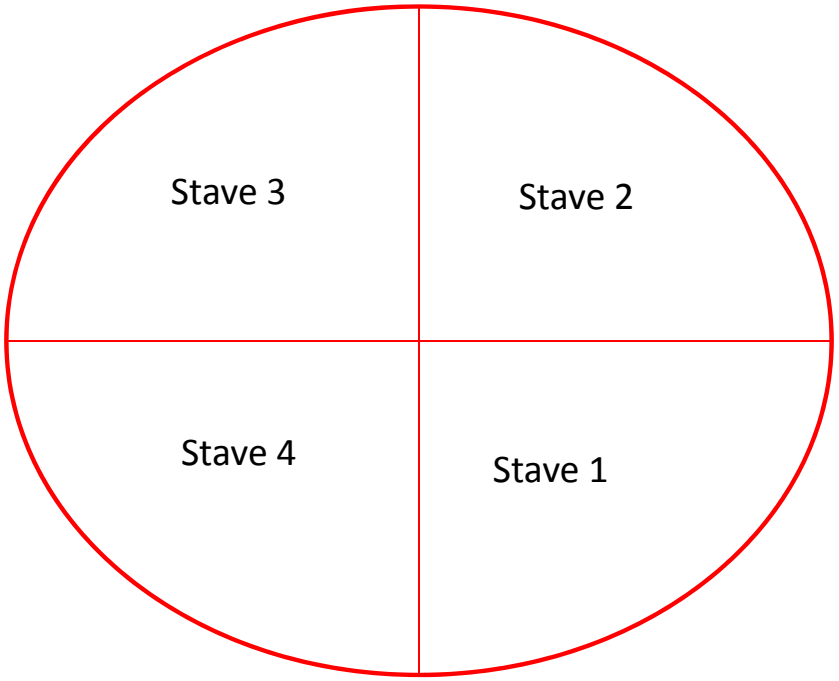
From a talk by either Alan or Eldwan

Nomenclature

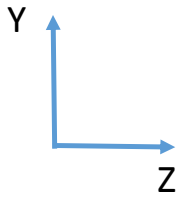
- ECAL is made up of detector id, staves, modules, slices and layers – these are used to encode the CellID (which is being used during digitization)
  - det_id = 1 (Barrel ECAL), = 2 (Endcap ECAL) [Yoke barrel appears to be 4]
  - Barrel ECAL: has 12 staves (as you go around in phi)
    - Module = 1 for Tile, = 2 for Strips
    - Slice = 1 for absorber, = 2 for Scintillator
    - Layers go from 1 to 42 (first 8 are tiles, remainder are strips)
  - Endcap ECAL: has 4 staves, appear to match the four quadrants
    - Module: = 0 for negative X, = 3 for positive X. So no distinction between strips/tiles?
    - Slice = 1 for absorber, = 2 for Scintillator
    - Layers go from 1 to 42 (first 6 are tiles, remainder are strips)
- Nodenames (in the code) look like
  - BarrelECal_stave10_module02_layer_21_slice2_vol_0
  - EndcapECal_stave02_module03_layer_13_slice2_vol_0

Staves in the Endcap – results based on hand scanning output in log file

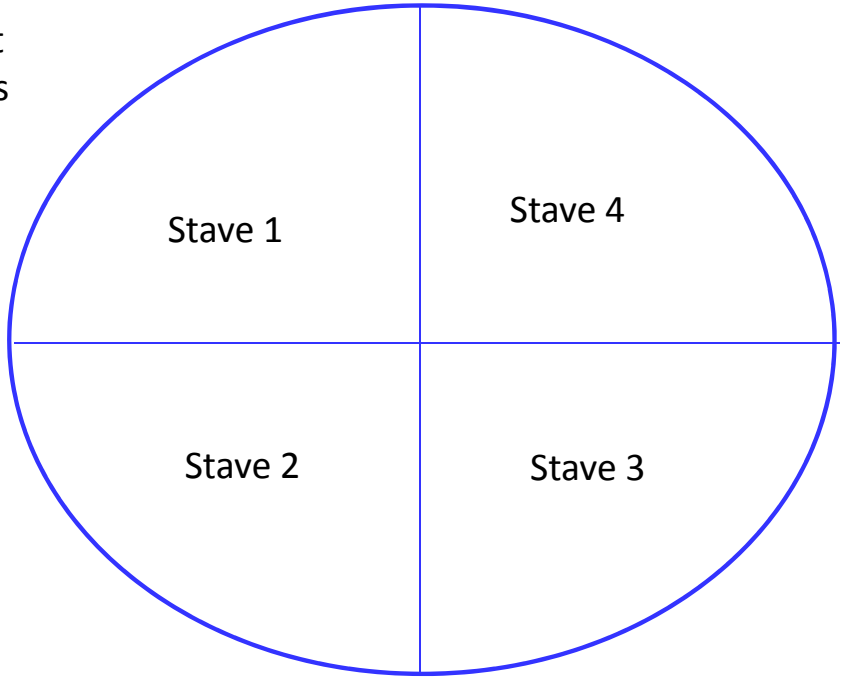
Module 0, i.e X < 0



Boundaries are not along the Y & Z axes
There is small amount of slop across the axes



Module 3, i.e X > 0



Structure of AnaTrees

- Although you have arrays like CCNC, Interaction Type, etc., whose length is the number of interactions within a spill, the reconstructed information in CALO, TPC is aggregated over all of them.
  - Current files have 1 interaction/spill, so it's a moot point

- In these studies, I use the variables SimHitCellID and RecoHitCellID to compare Sim and Reco information
  - I still need to trace CellEncoding in the code, but hits with the same cellID need not be at the same location in space (later)

/pnfs/dune/persistent/users/ebrianne/ProductionSamples/ND-GAr/ nd_hall_mpd_only_ECal12sides_42l_SPY_v3_wMuID/

- Contain Reco files, Anatree files, CAFMaker files and ParamSim files

My sample*

*My files are in /dune/data/users/vj/test4/work/

- Built GArSoft locally using develop branch (on July 6)

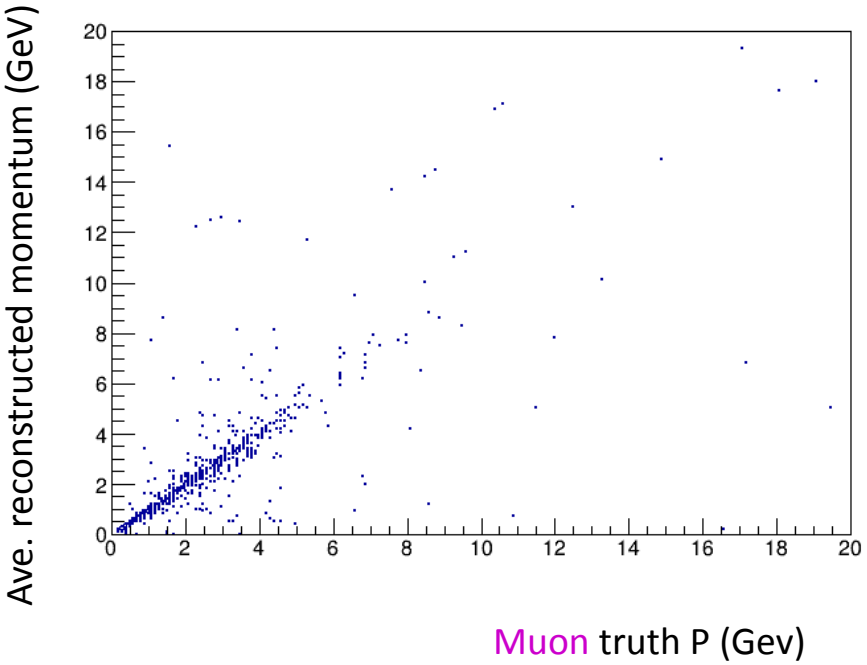
- art –n 1000 –c prodgenie.fcl (I don't know what .fcl/flux files Eldwan used)
  - *"…New detector geometry loaded from: /nashome/v/vj/datadir/test4/localProducts_garsoft_develop_e20_prof/garsoft/v02_10_03/gdml/nd_hall_mpd_only/nd_hall_mpd_only_ECal12sides_42l_SPY_v3_wMuID.gdml …"*
  - *In GENIEGen.fcl, I see the following:*
    - *# Optimized flux, Nov 2017:*
    - *FluxType: "dk2nu"*
    - *FluxFiles: ["g4lbne_v3r5p4_QGSP_BERT_OptimizedEngineeredNov2017_neutrino_?????.dk2nu.root"]*
    - *FluxCopyMethod: "IFDH"*
    - *FluxSearchPaths: "/pnfs/dune/persistent/users/ljf26/fluxfiles/g4lbne/v3r5p4/QGSP_BERT/OptimizedEngineeredNov2017/**neutrino**/flux/dk2nu"*

- art –c readoutsimjob.fcl –s genie_gen.root (debug statements in Digi code)
- art –c recojob.fcl –s readoutsim.root
- art –c anajob.fcl –s reco.root
  - Changed the default anatree.fcl to store BackTracker info for CALO, as well as storing CALO DigiHits

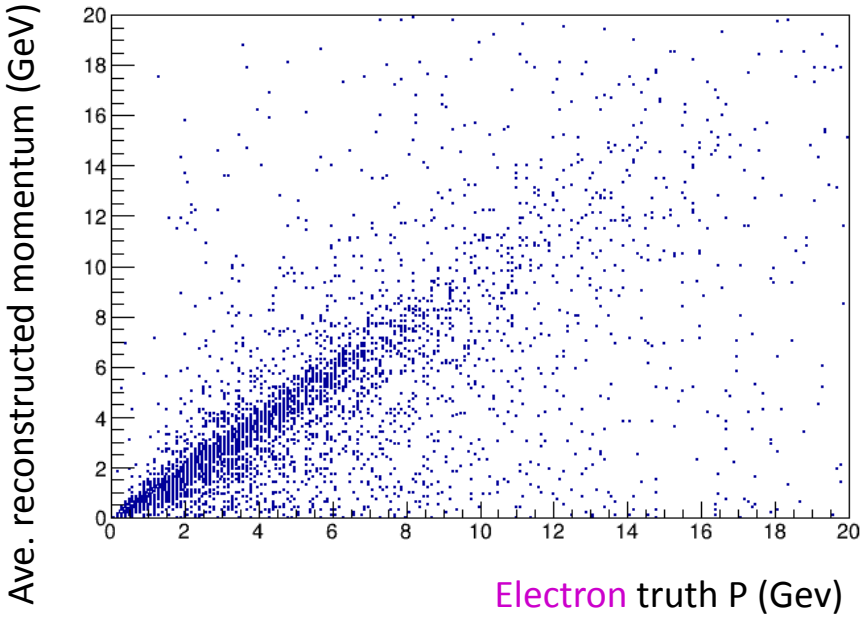
- **I made a total of 3000 events – because the flux files are in the /neutrino/ sub-dir, I am guessing they are FHC**

Energy resolutions – this is old stuff

- If we want to study electron resolutions in CALO, we will need to compare the Σ(cluster E) with an appropriate value for truth energy
  - Current anatrees do not contain TrajMCPP[X,Y,Z] variables, so will have to come up with an estimate.
  - Require that there be only one TPC track matching a MCParticle
  - Each reconstructed track has two momenta associated with it – starting and end. Since start/end are arbitrarily defined, take the average

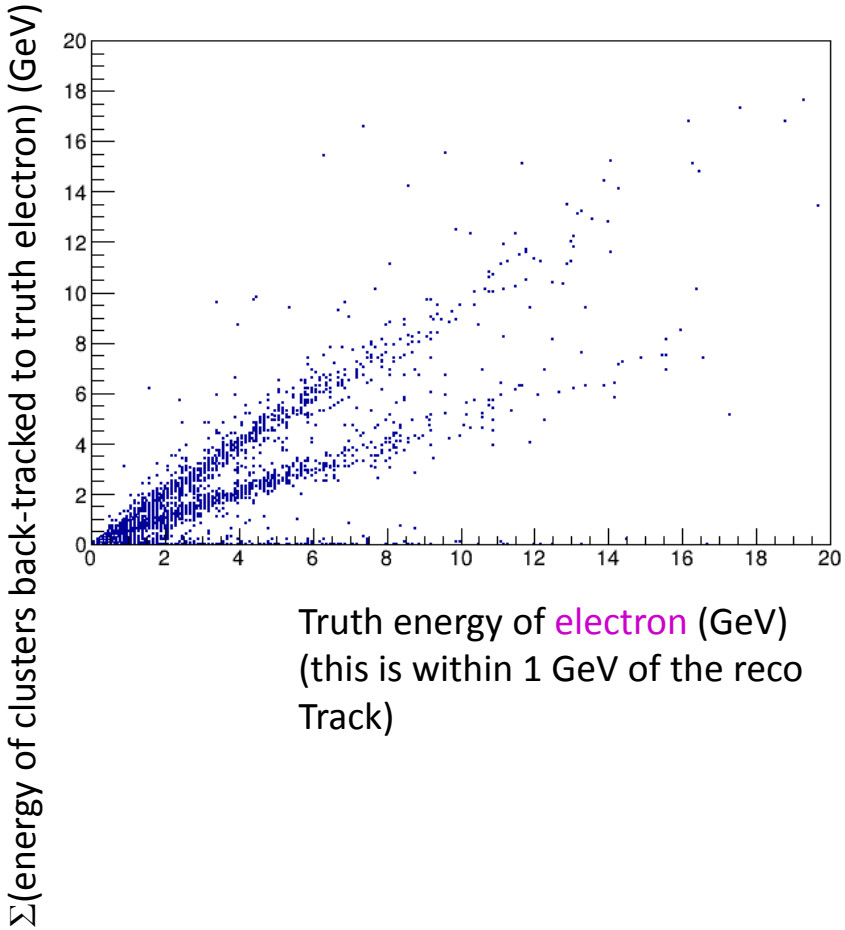


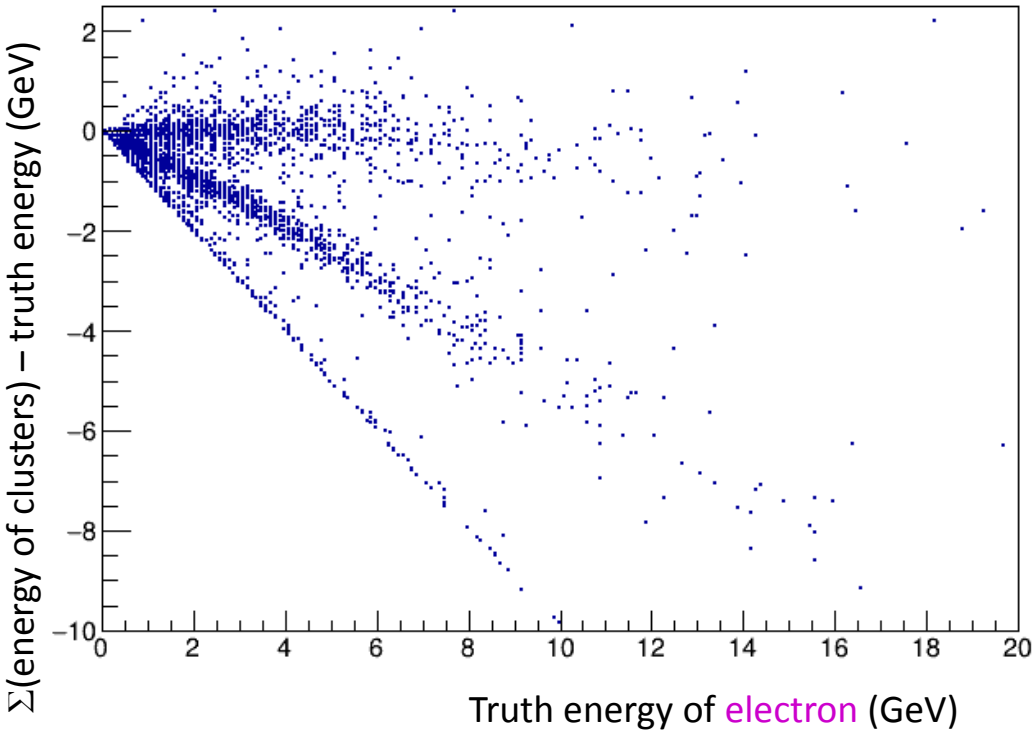
entries = 939
with ave. reco P
within 1 GeV of truth P = 833 (89%)



entries = 5817
with ave. reco P
within 1 GeV of truth P = 3674 (63%)

Require ave. reco P to be within 1 GeV of truth momentum


Σ(energy of clusters back-tracked to truth electron) (GeV)
Truth energy of electron (GeV)
(this is within 1 GeV of the reco Track)


Σ(energy of clusters) – truth energy (GeV)
Truth energy of electron (GeV)
(this is within 1 GeV of the reco Track)