Updates to disambiguation
Tracking efficiencies for a cosmic sample
Monte Carlo Challenge 3

Karl Warburton with guidance from Tingjun Yang

17/06/2015

Reminder from last week

- I re-did Muhammads work on tracking efficiencies for a primary anti-muon sample (10K events).
  - Observed that disambiguation was main place where improvement could be made. Cheated disambiguation with pandora close to 100% above 40 cm.
  - Martin form Warwick showed an analogous analysis he and Jon (Sheffield) had done. Got similar results to me.
- Generated a sample of 1K 10 drift window CRY events, saw that efficiency was very low, due to poor disambiguation.
  - Disambiguation was only selecting largest cluster in each TPC, multiple clusters leads to all but largest being discounted -> Bad efficiency.

Updates to disambiguation

How does disambiguation work?

- Disambiguation tries to match collection, U and V hits into triplets with a common time and a small intersection distance over the wires.
- Then clusters these hits into objects which it will eventually use to create tracks.
- Works out which clusters to keep.
  - Formerly takes only largest cluster in each TPC.
  - Now checks that clusters do not overlap in time in a TPC.
- Removes any outlier hits from this cluster, and tries to add any missed hits to the cluster (‘clean and fill’).

How does it work out whether clusters overlap in time?

- After clustering all hits, works out start/end time of each cluster.
  - Does this by looking at the peak times of all hits, and if before/after the stored start/end time of the cluster it belongs to setting it to the peak time of this hit.
- Loops through all clusters (cluster 1)
  - For each cluster look at all clusters (cluster 2)
    - If cluster 2 has more hits then cluster 1, check if cluster 1 is entirely contained within cluster 2.
      - If it is then this is a ‘bad’ cluster so do not use to seed a track.
- Only have time separated clusters, so ‘clean and fill’ as per previous slide.

Definition of a ‘bad’ cluster

- A cluster which is entirely contained within another, larger cluster.
- Due to ambiguity of wire wrapping hits on a given U/V channel can be in any of 3 locations.
- Can reconstruct triplets on any of these three sections, but would expect almost all to be in correct section.
- Want to remove these fake hits which make ‘bad’ clusters, so check for any clusters which are entirely contained within larger clusters.
  - Can also extend to checking channel numbers of these smaller clusters to prevent losing legitimate but time contained clusters.

Monte Carlo Challenge 3

Monte Carlo Challenge 3

- Apologies that it has been a long time since MCC2.

- Will try to make sure there isn't such a large gap between the current and next MCC.

- But lots of improvements, see Tingjun's talk last week.

Stages (Recap)

- Each sample goes through five stages;
  - Gen - CRY or TextFileGen
  - G4 - Geant4 Simulation (including TPC's, counters and photon detectors)
  - Detsim - TPC readout simulation
  - Reco - Full reconstruction
  - Mergeana - Merge art output files (only these are uploaded to SAM), run anatree on those files.
- The fcl files used are saved in lbnecode/fcl/lbne35t

File sizes

	Gen	G4	Detsim	Reco	Mergeana
Anti-Mu	11 MB	120 GB	120 GB	124 GB	126 GB
CRY - 10 drift window	5.3 MB	19 GB	20 GB	26 GB	31 GB

- CRY sample much smaller now (was 127 GB last time) due to ROI compression - make a larger set?

Location of files

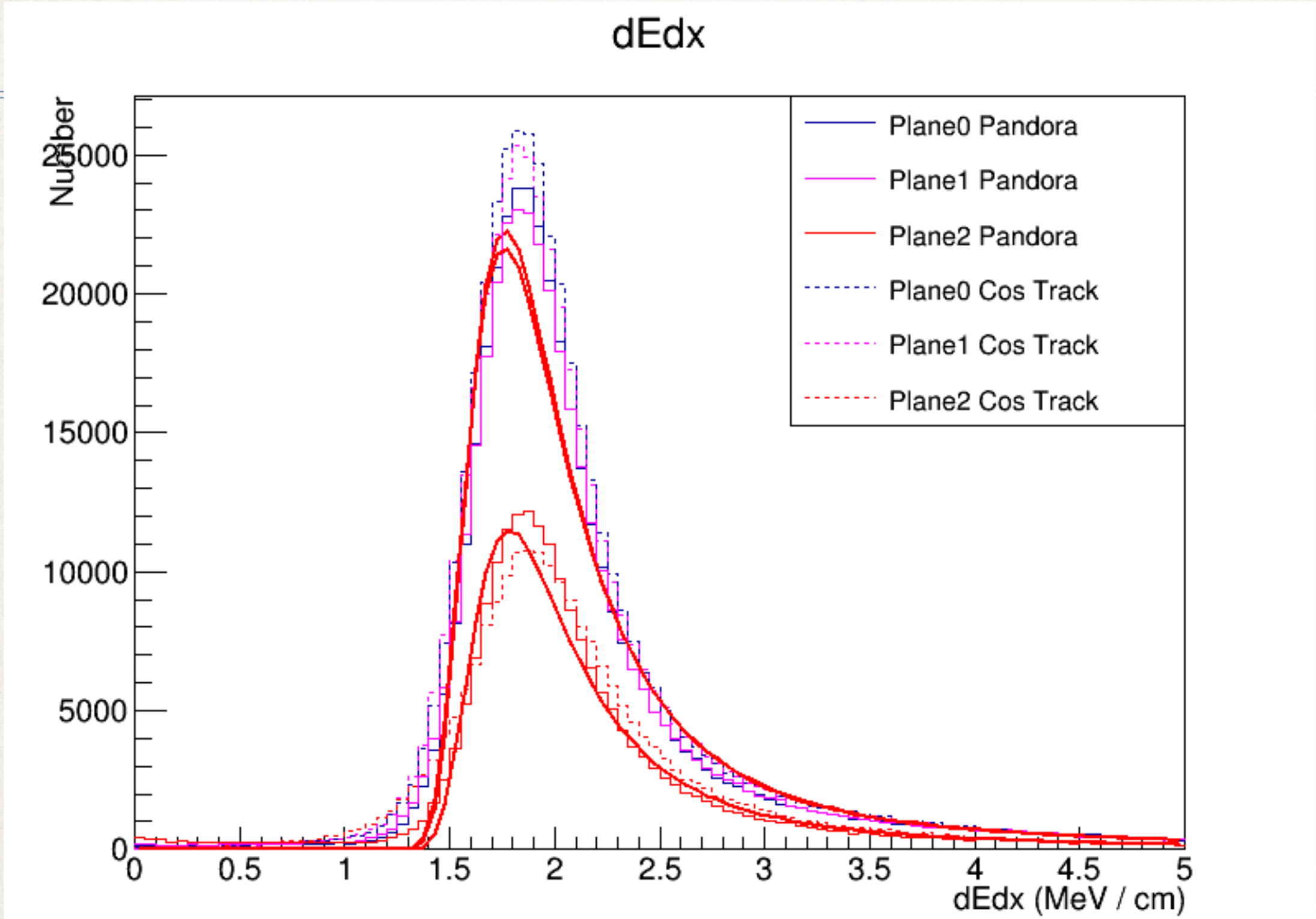
- Follow the same structure as for that of MCC1 and MCC2.
- /pnfs/lbne/scratch/lbnepro/v04_12_00/(STAGE)/(PROJECT_VERSION)/(JOB_ID)/(FILE)
- Currently two project versions;
  - 10,000 Anti-Muon events
  - 1,000 CRY 10 drift window events
- Mergeana stage is uploaded to enstore.
- Wiki is updated to reflect addition of the new sample.
- Plan to add some filtered samples. Any requests?
  - Proton / Pion / Horizontal Muon

Monitoring of MCC 3

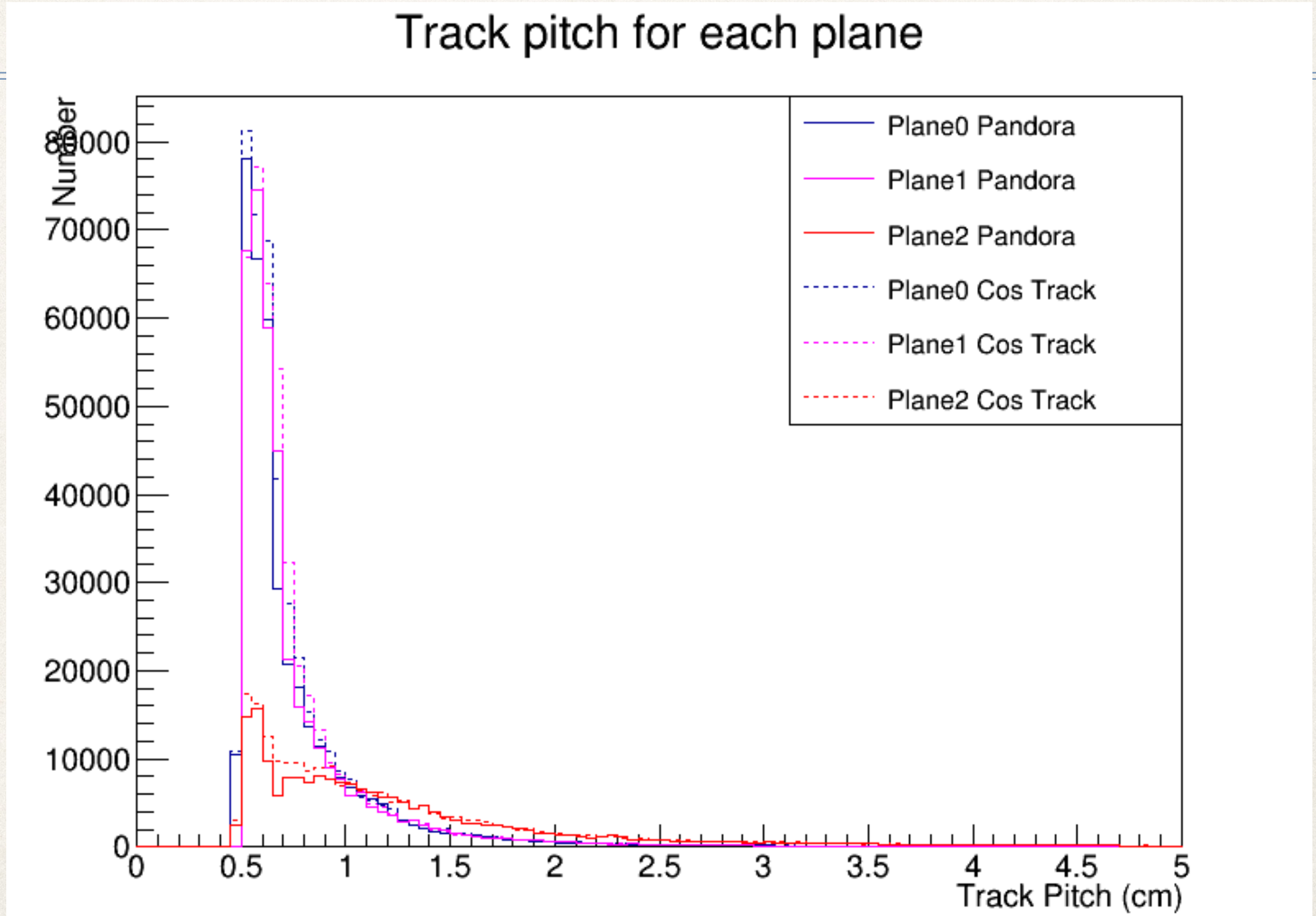
What monitoring?

- After generate a new dataset want to check is consistent with older generated files, and certain quantities are correct.
  - Reasonable tracking efficiencies
  - Correct calorimetry for muons
  - Will want to do others too I imagine. Any ideas?
- As I have done this I have re-done these analyses for the new MCC.

dEdx for Pandora and Cosmic Track



Track pitch for Pandora and CosTrk



Definition of Tracking Efficiency

- Cleanly separated numerator and denominator so both can be expressly defined in code.
- Numerator defined as;
  - MCTruth information for matched tracks.
  - Reconstructed track length of between 75% and 125% of MC track length, which is non-zero.
  - Only one track to be filled per MCTruth GEANT4 ID.
- Denominator defined as;
  - MCTruth particle information.
    Only Anti-muons with non-zero track length in the detector.
- Can be shown for any combination of protons, muons, electrons, pions and kaons.

Definition of matched track

- Loop through each track
  - Loop through each MCParticle
    - If GEANT4 trackId of track which caused track is equal to MCParticle then are matched.
- I get GEANT4 trackId from backtracker, using the MCTruthT0 calculation.

Tracking Efficiencies Monitoring

- For each sample (Anti-Mu and CRY) will show;
  - Length Efficiency
  - Theta vs Phi Efficiency
- For Cosmic Tracker and Pandora;
  - Reconstructed vs Truth Length
  - Reconstructed vs Truth Theta
  - Reconstructed vs Truth Phi

Layout of subsequent slides

CRY, reconstructed using Cosmic Tracker

CRY, reconstructed using Pandora

Anti-Muon, reconstructed using Cosmic Tracker

Anti-Muon, reconstructed using Pandora

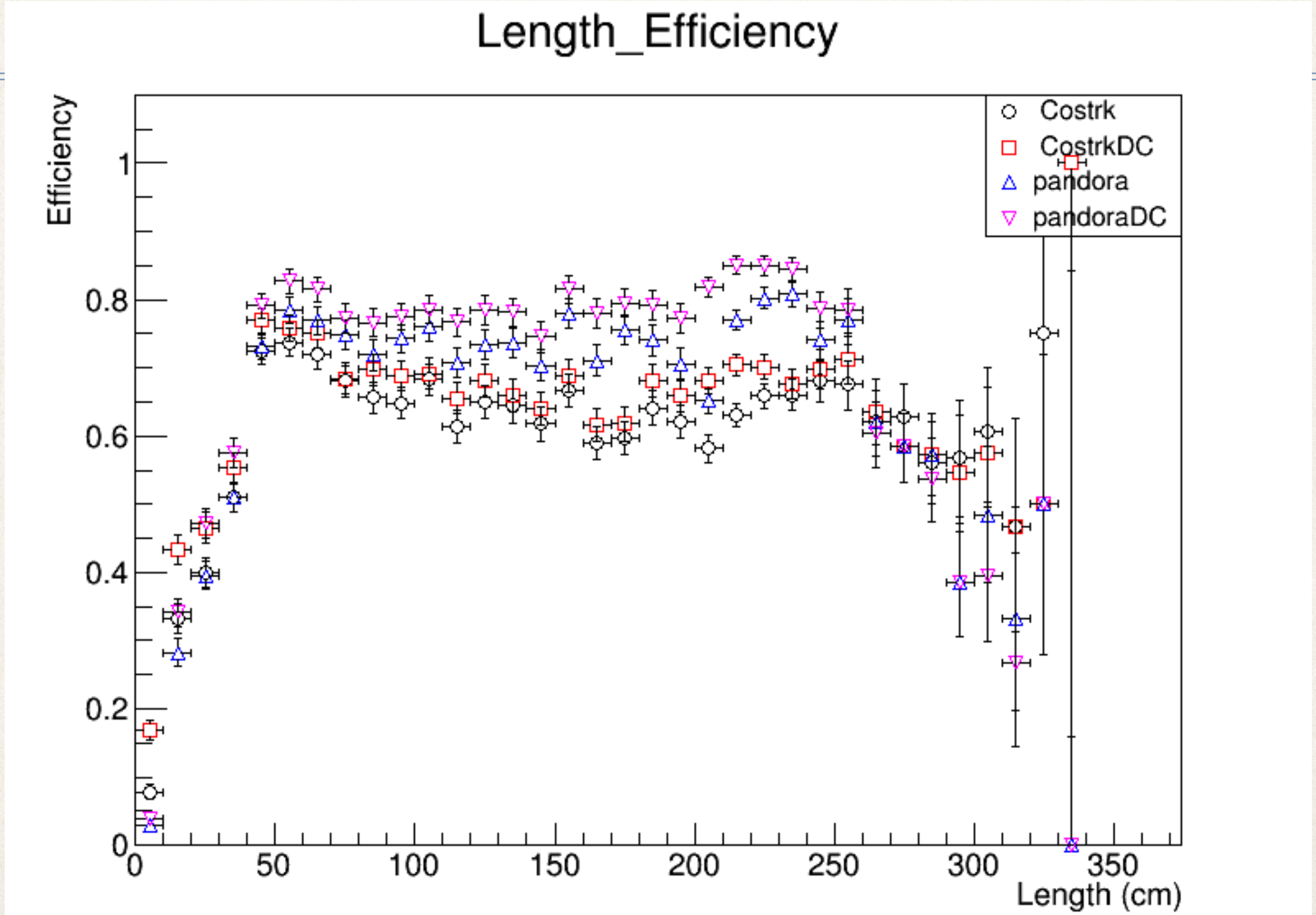
Tracking Efficiencies II

- Again I invite people to look at;

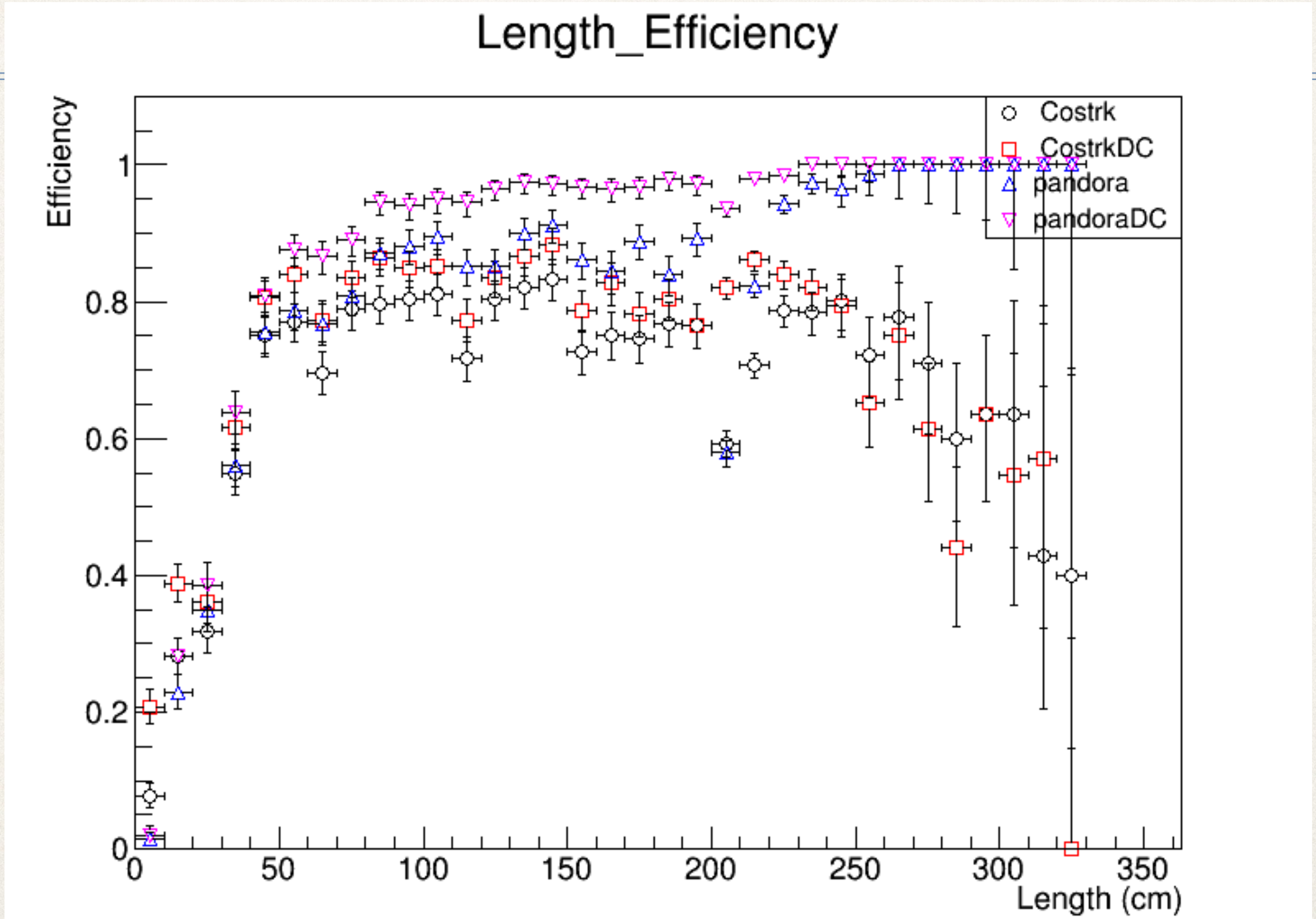
/lbne/app/users/php13tkw/LarDevelop/workspace/TrackingEfficiencies/MCC3_(Sample)

- Important note in comparing Pandora and Cosmic Track, Reconstructed vs Truth angular plots;
  - Cosmic Tracker makes many more tracks (~15k) than particles present (~6k) for Anti-Mu and ~36k tracks for ~13k particles for CRY.
  - Pandora makes far fewer ~6.5k and ~13.5k.
  - So that Cosmic Tracker comparison plot doesn't have lots of points from these bad tracks, I have only plotted fully matched tracks in these plots.

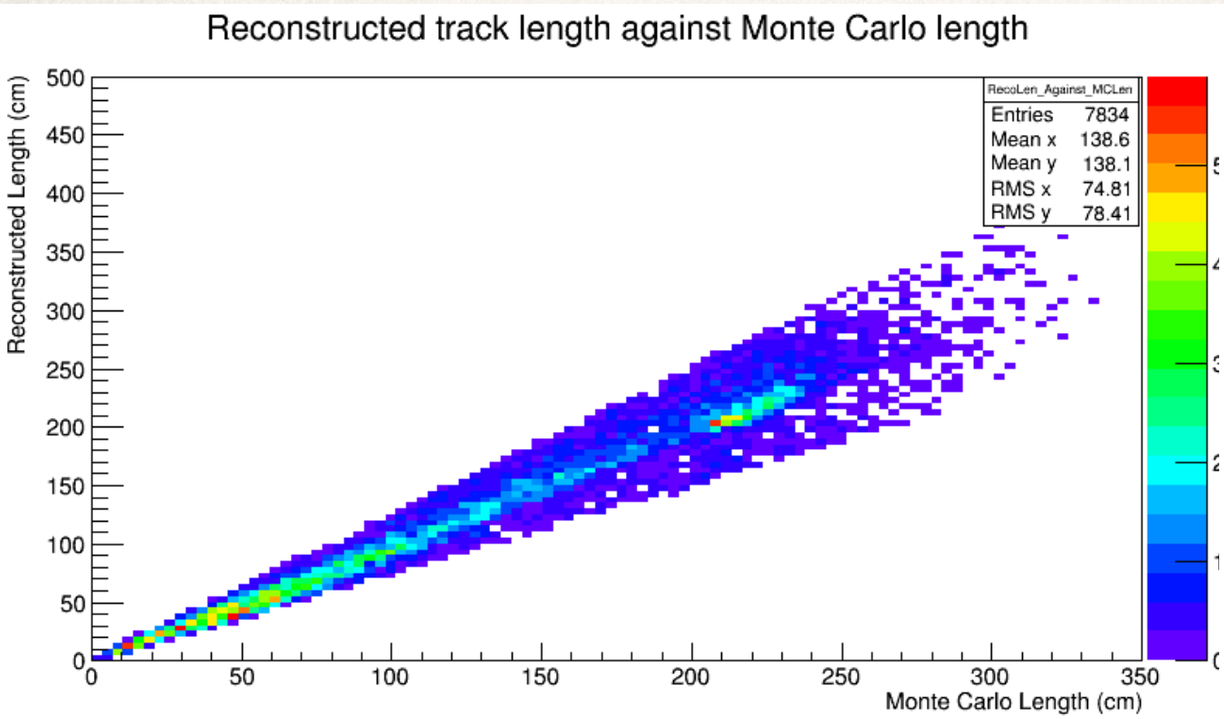
CRY, Length Efficiency

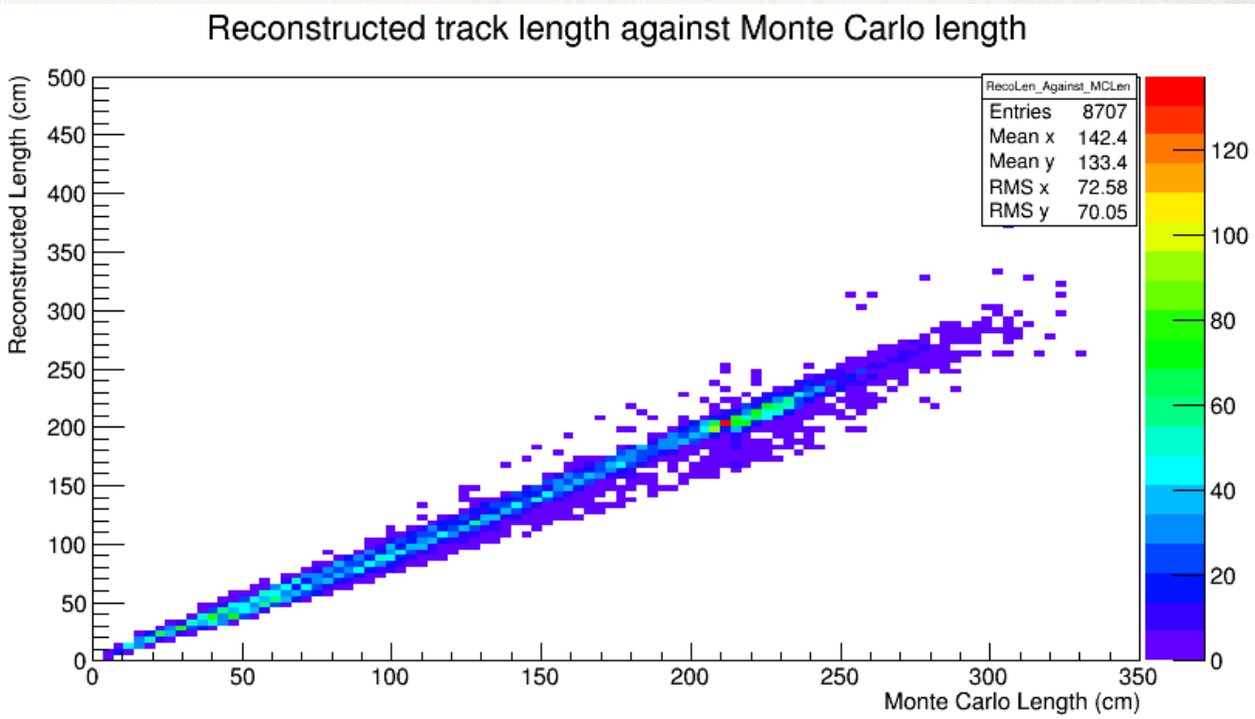


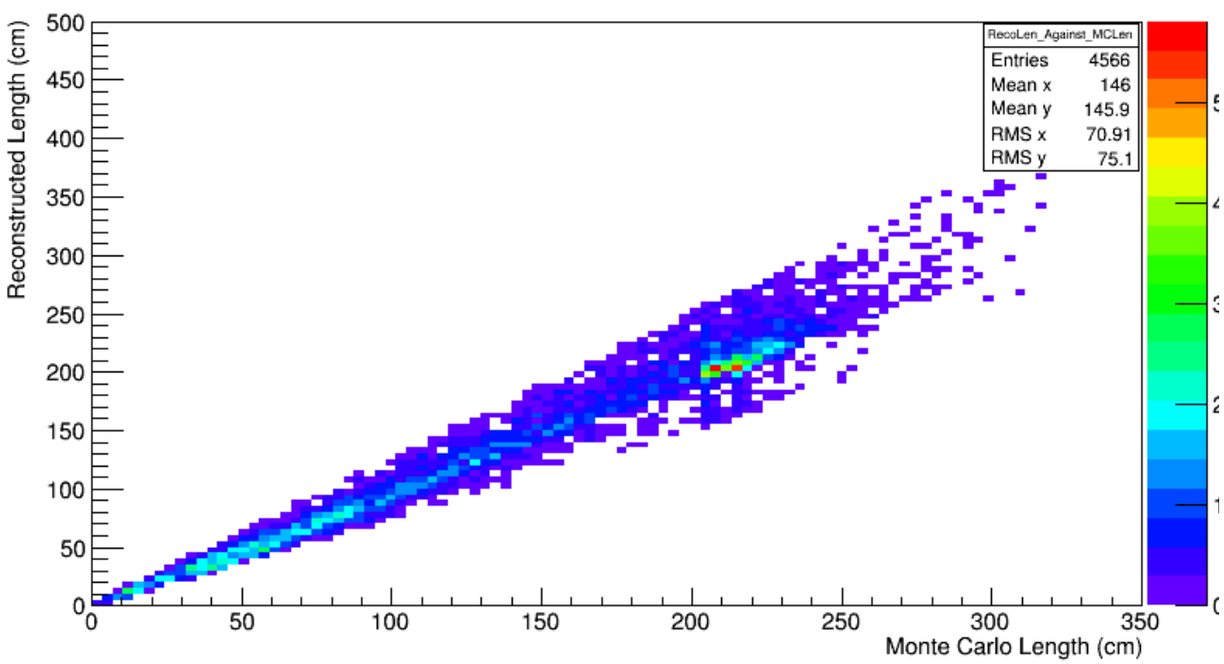
Anti-Muon, Length Efficiency

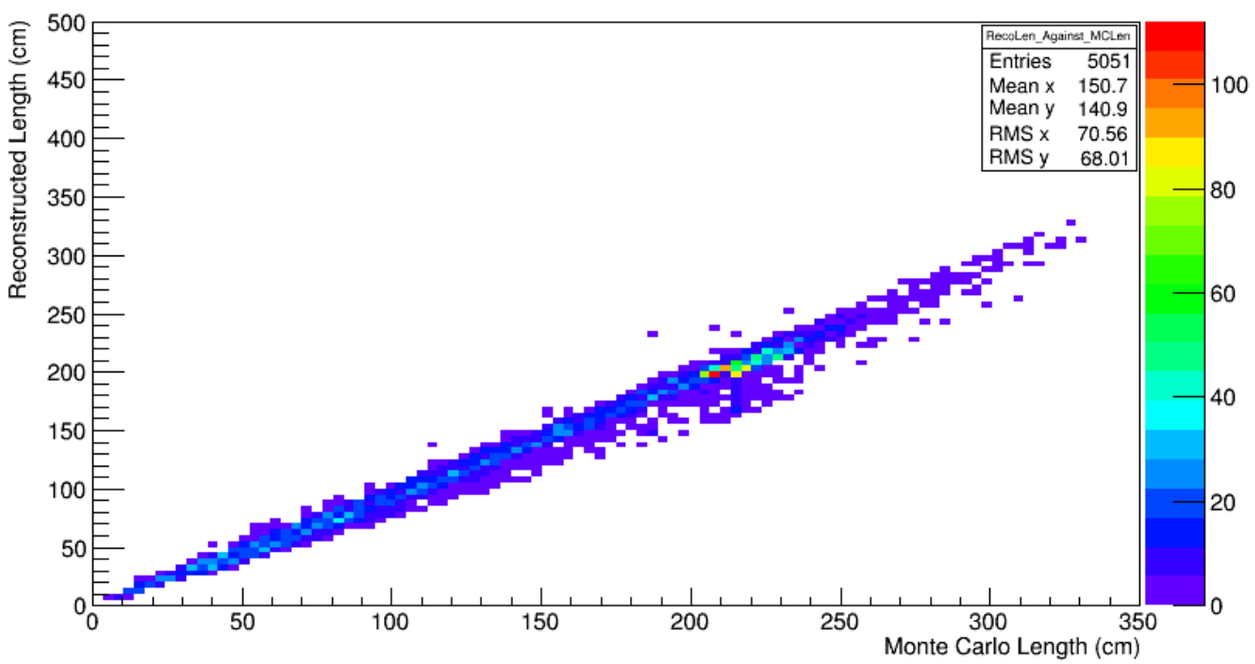


Reco vs True Length - Matched

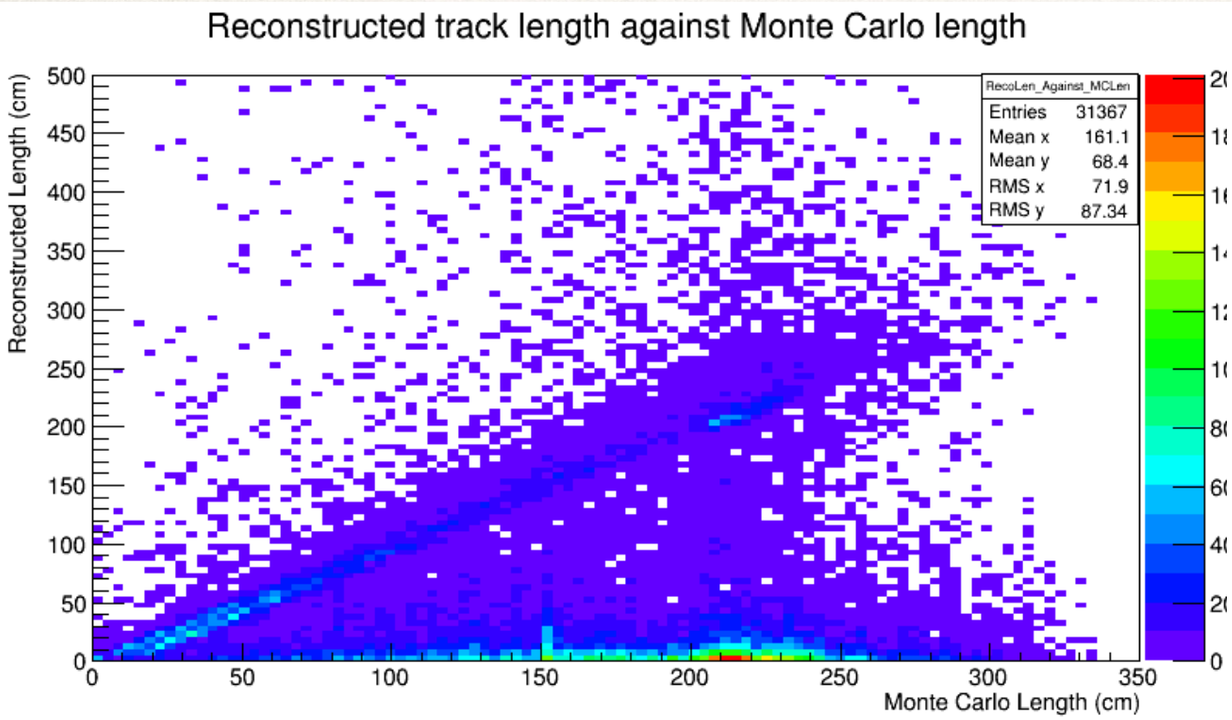


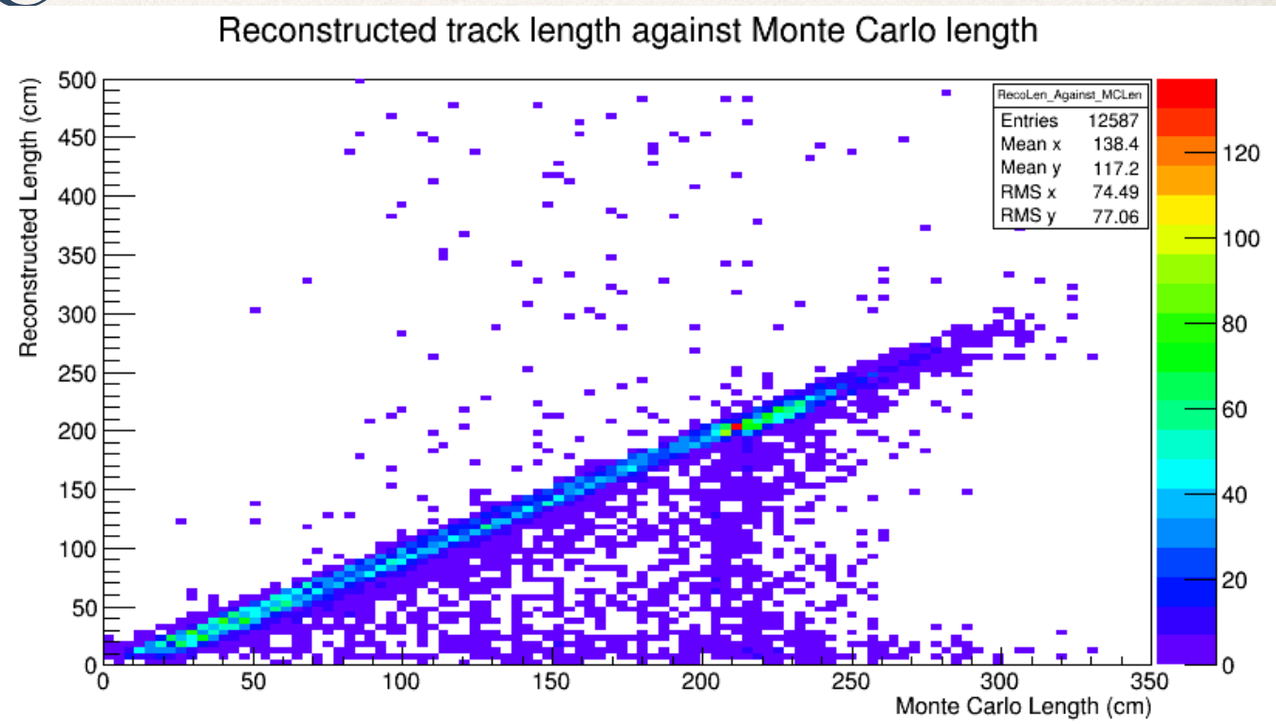


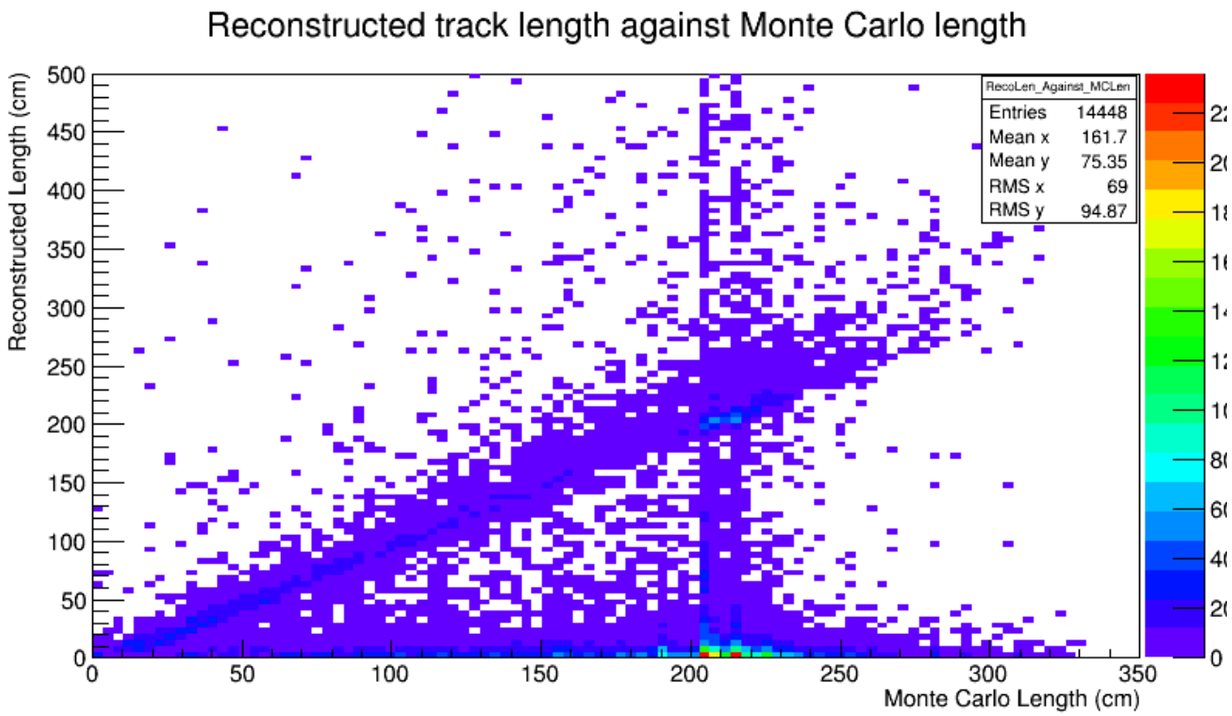


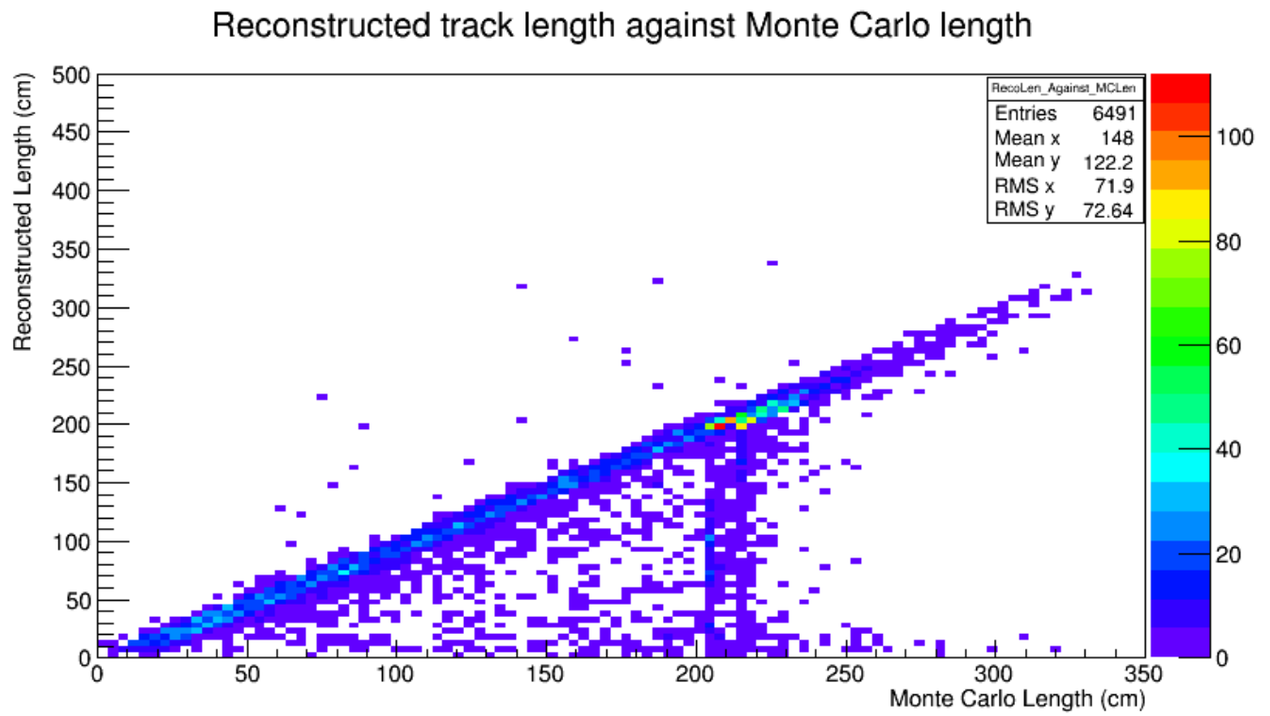


Reco vs True Length - All

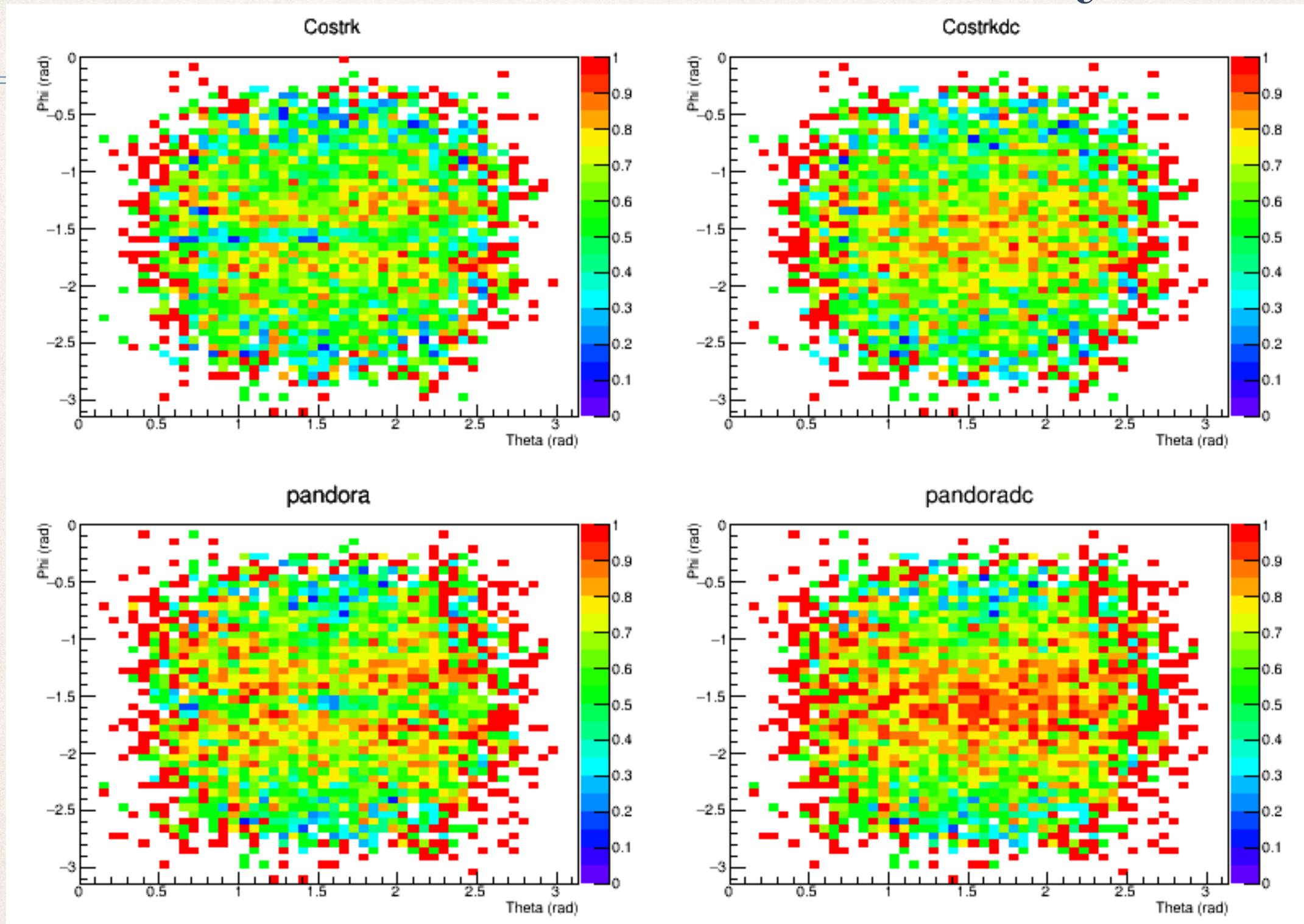




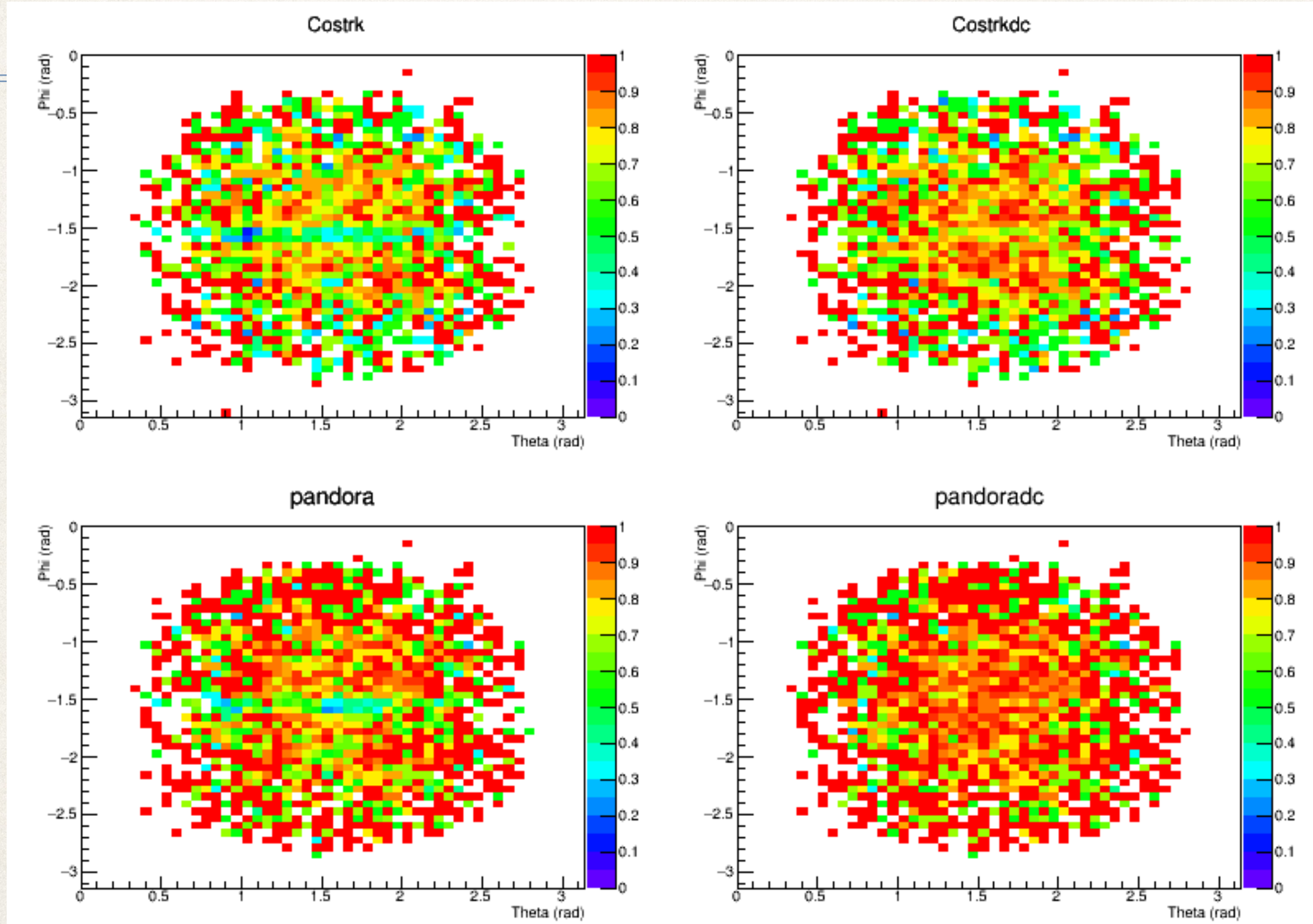




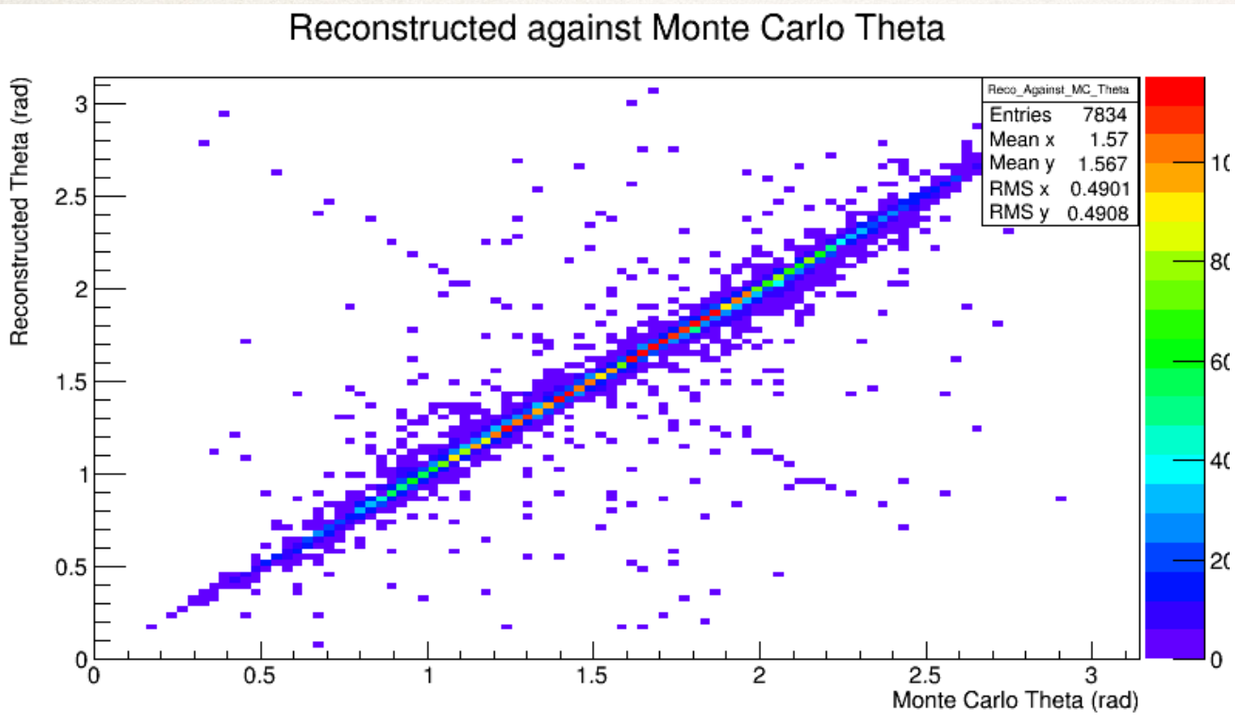
CRY, Theta vs Phi Efficiency

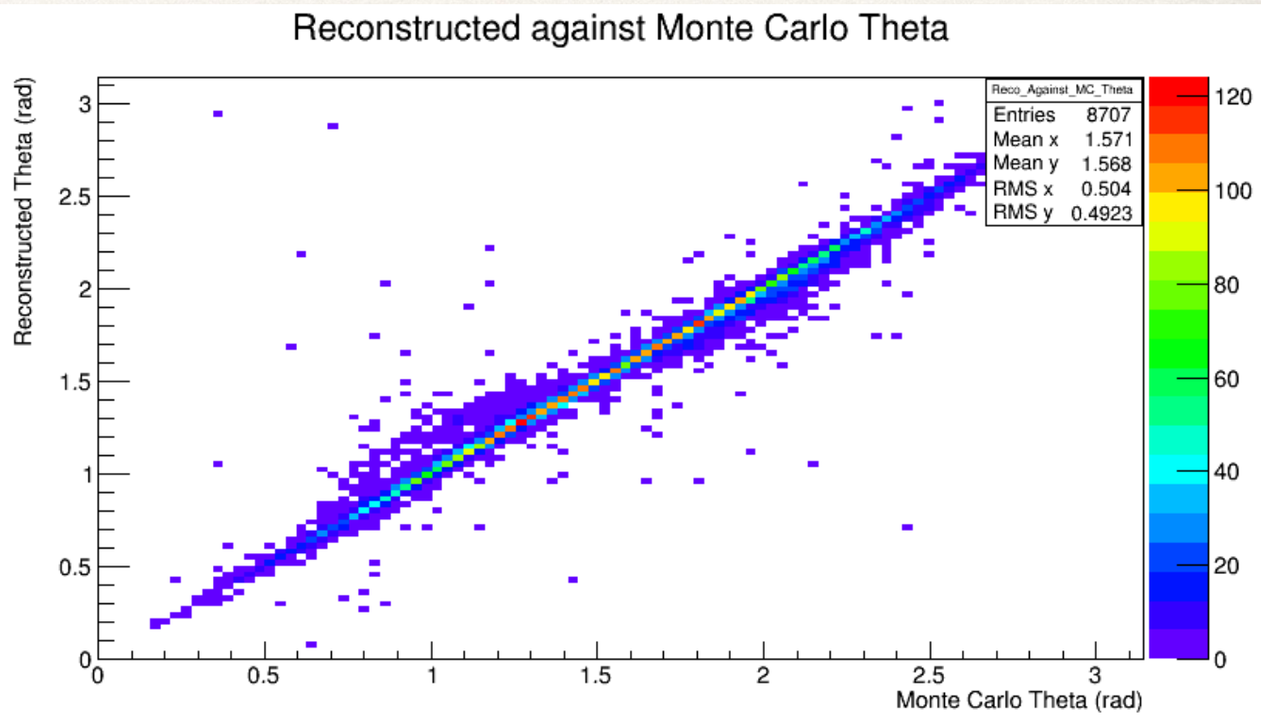


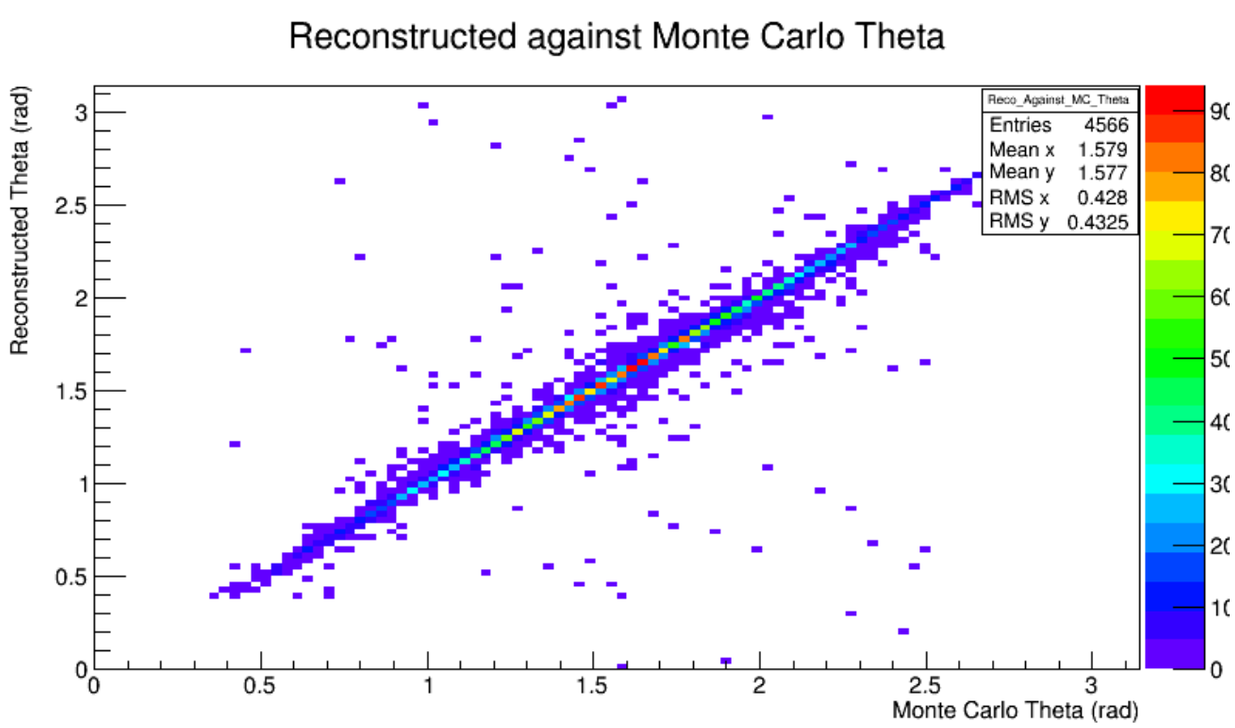
Anti-Muon, Theta vs Phi Efficiency

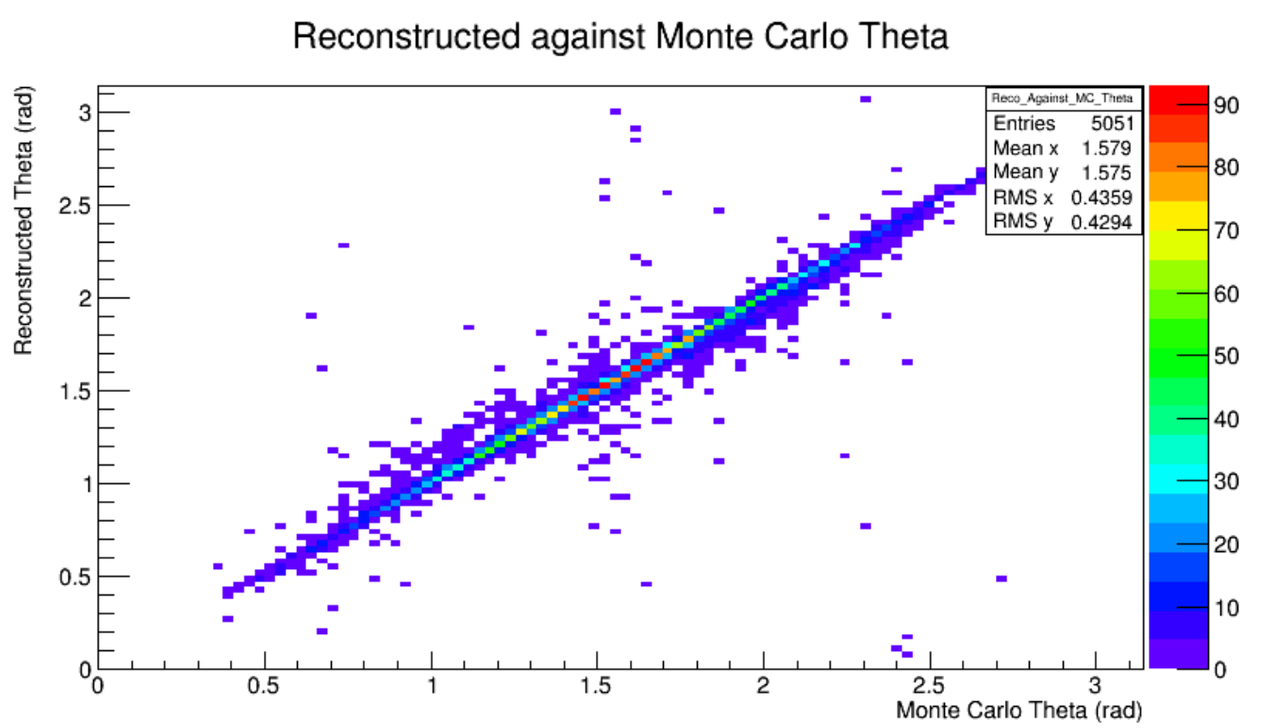


Reco vs True Theta - Matched

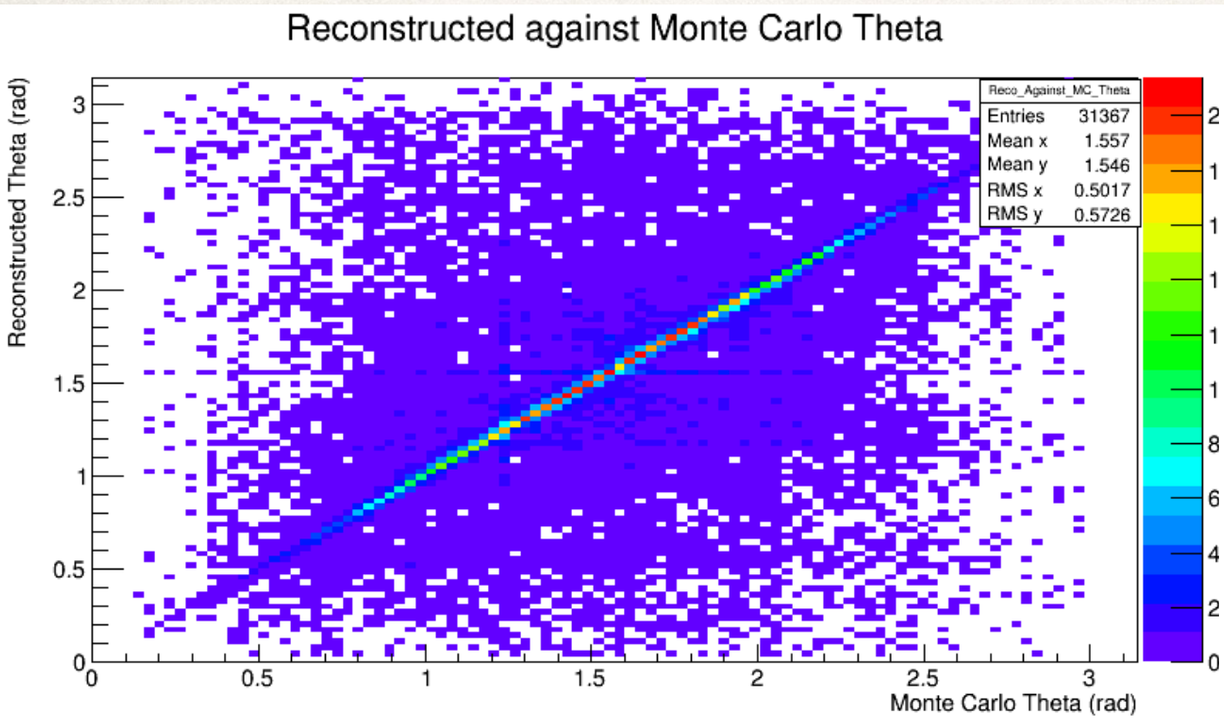


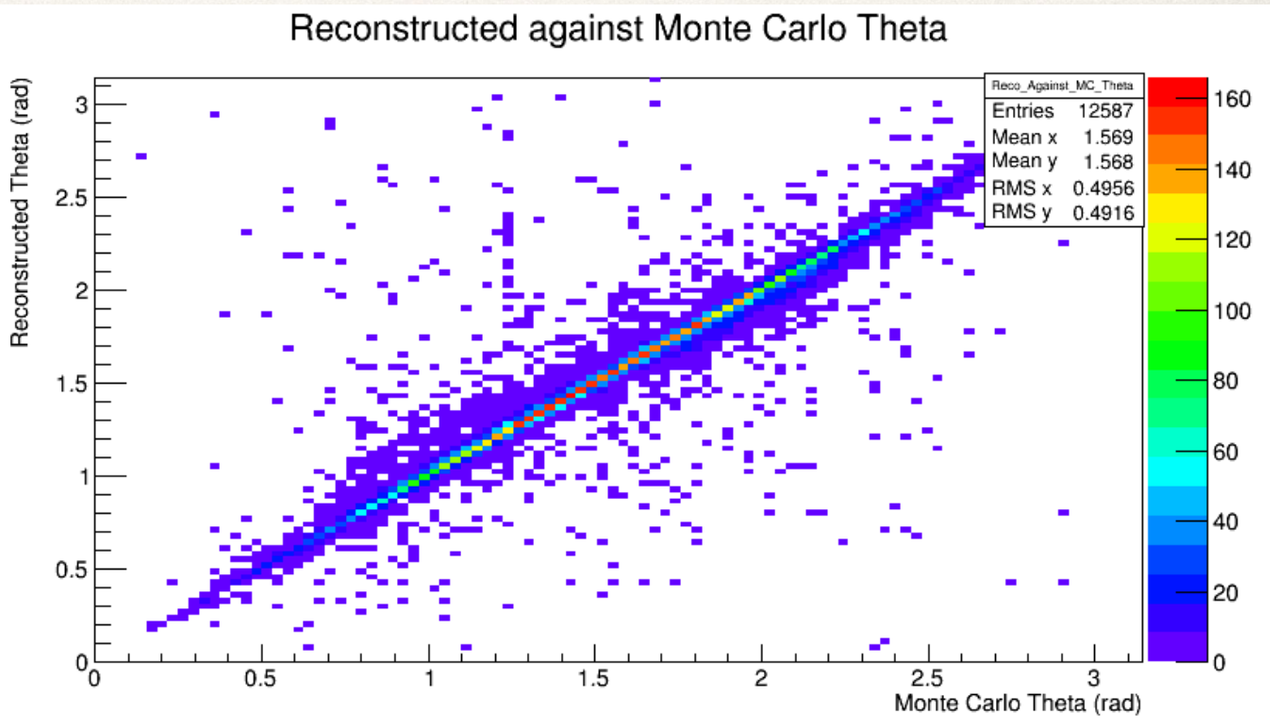


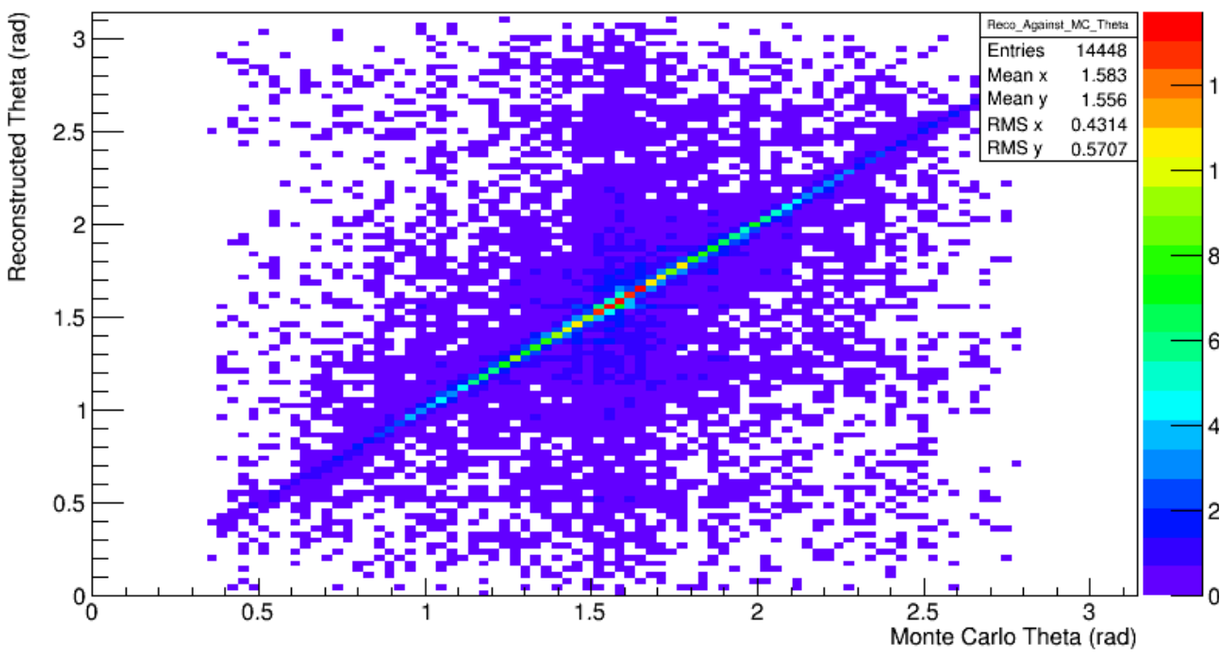


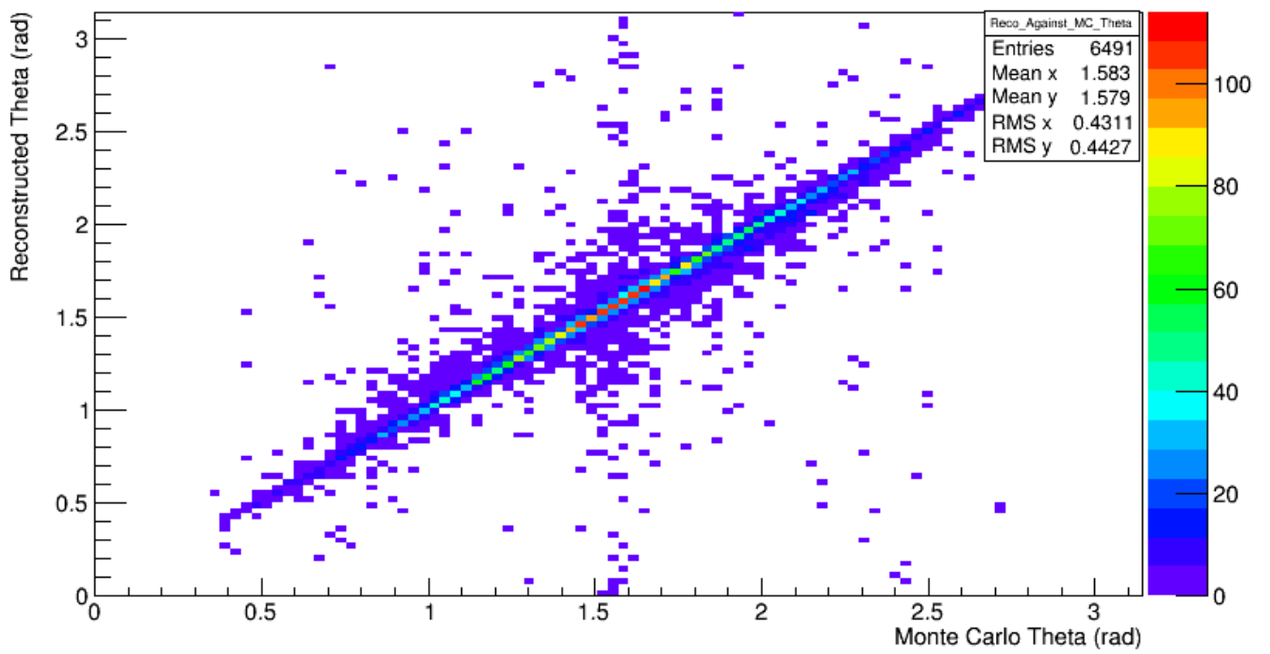


Reco vs True Theta - All

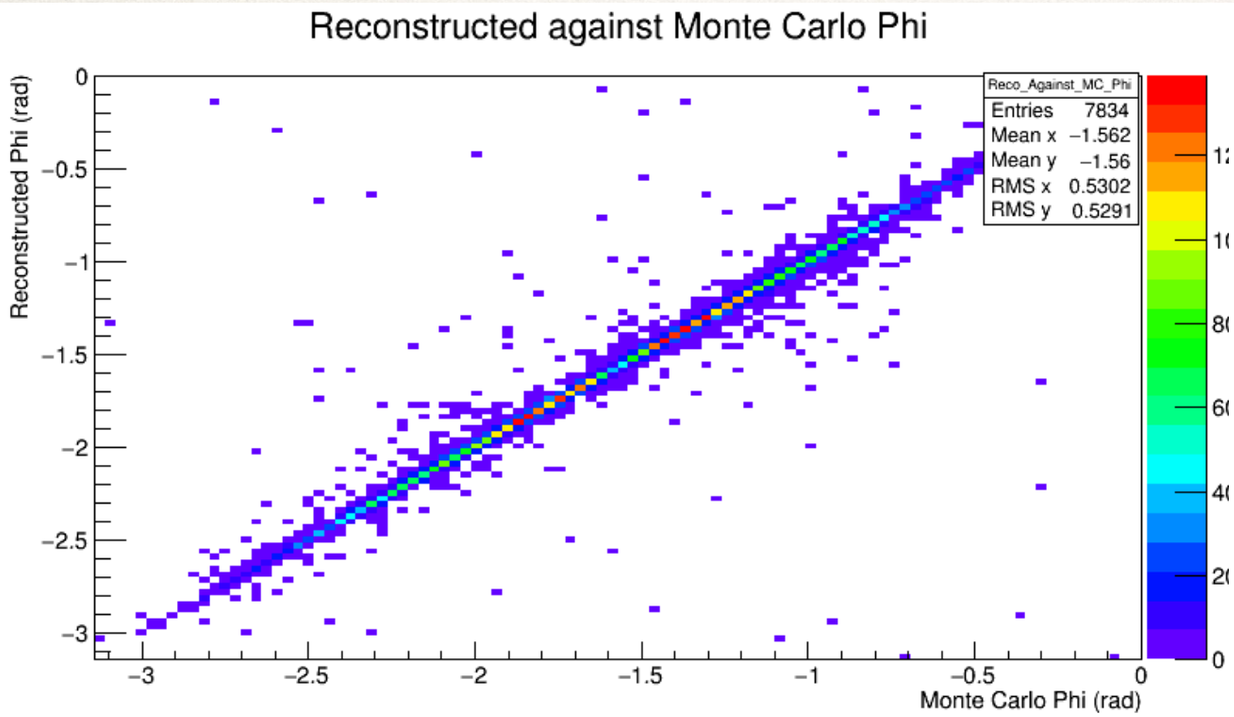


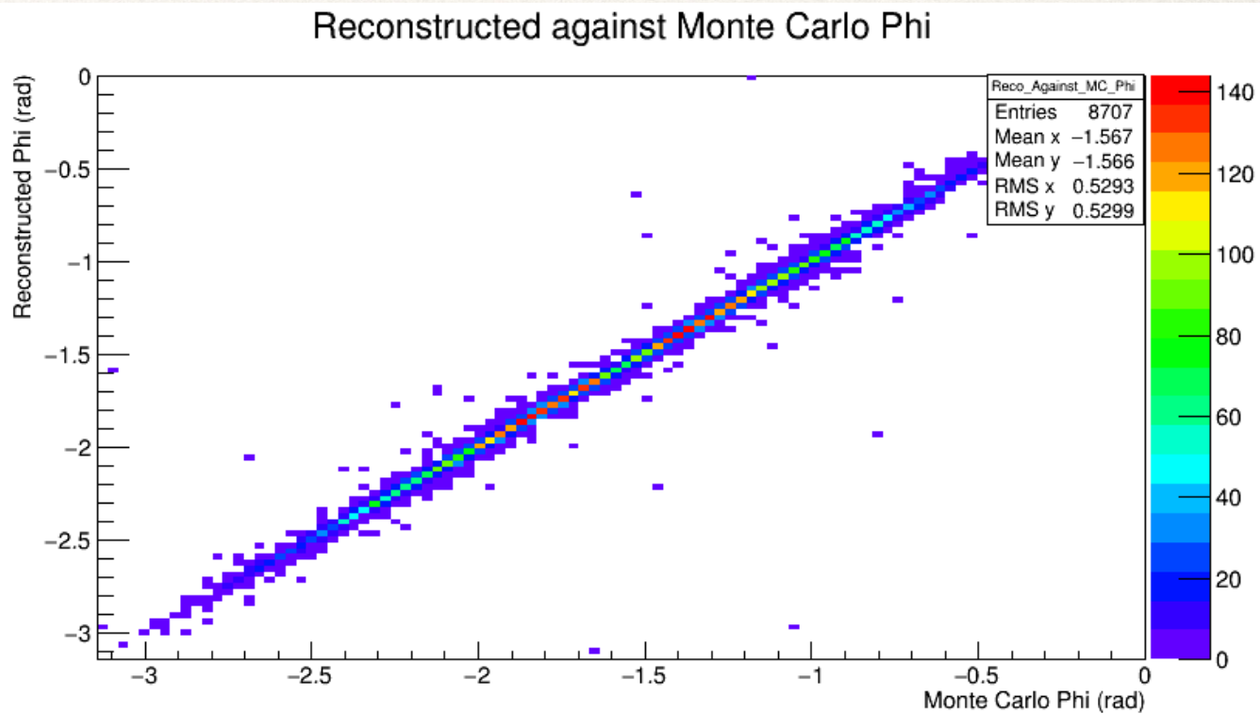


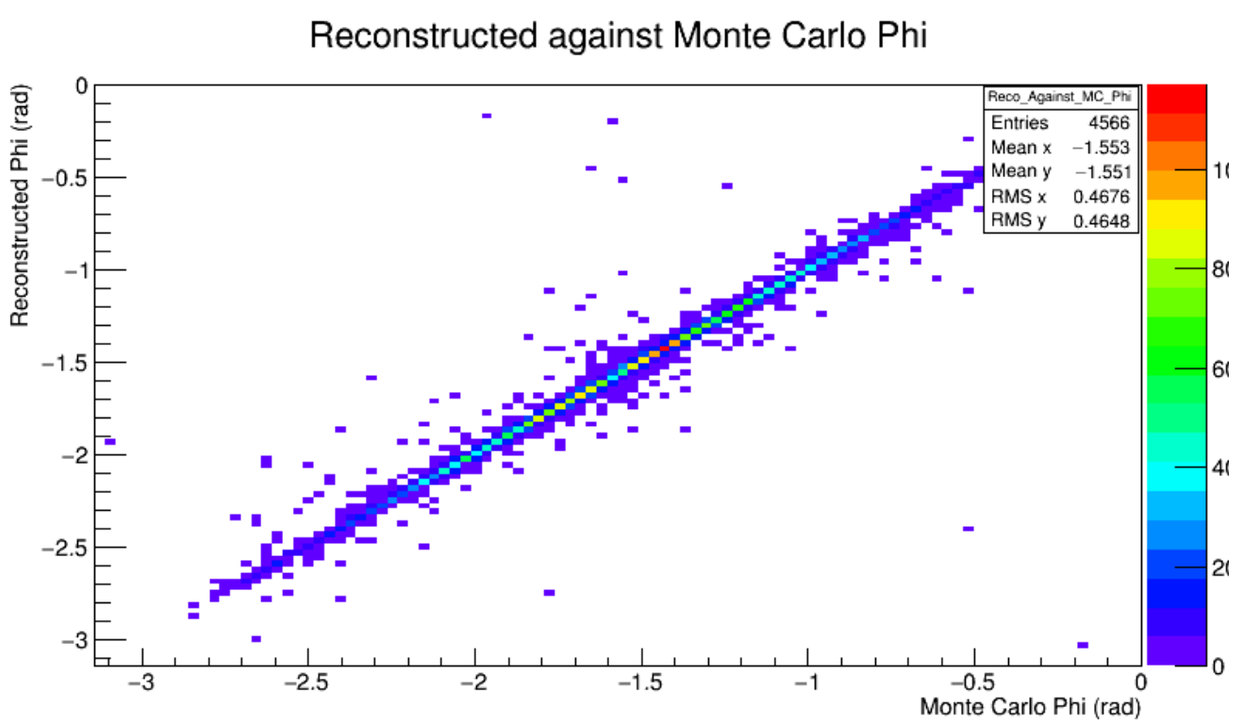


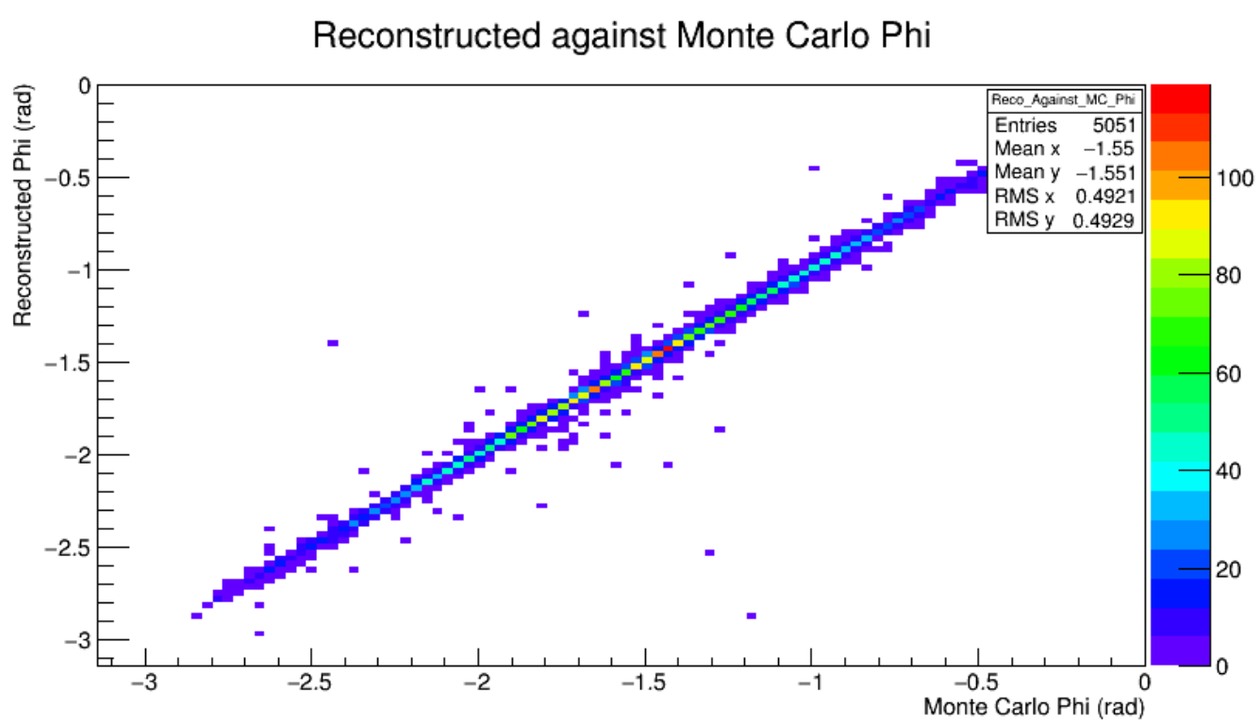


Reco vs True Phi - Matched

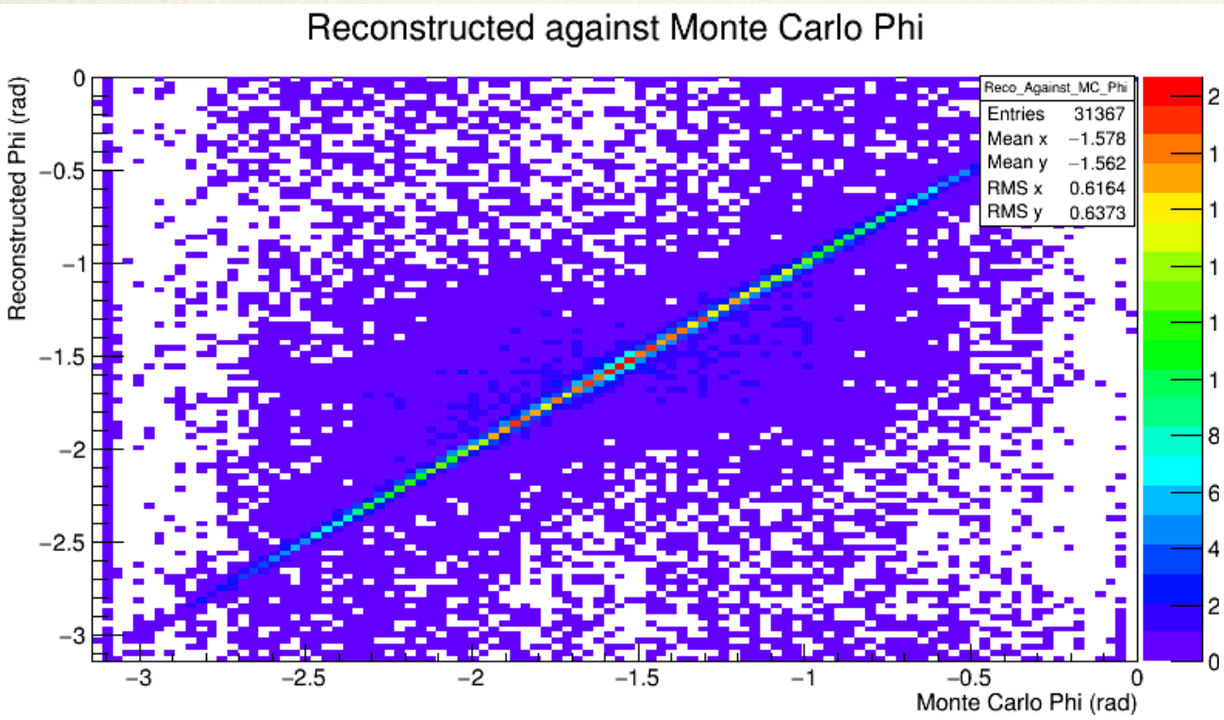


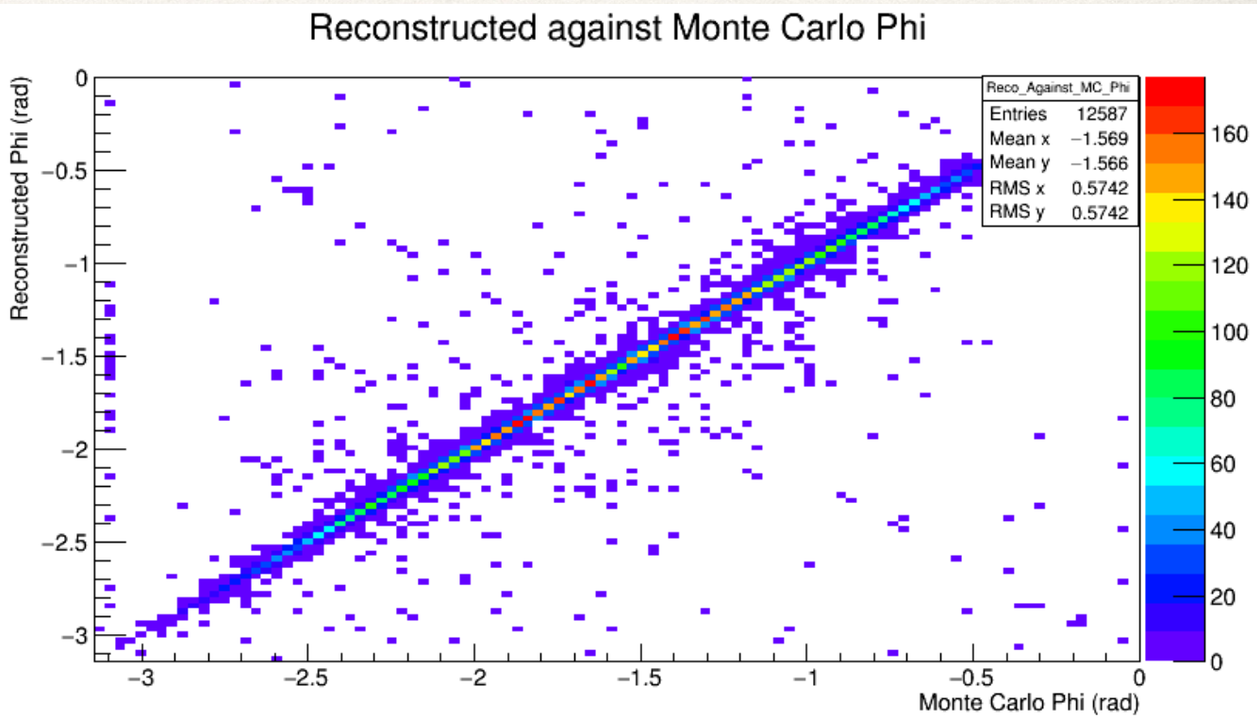


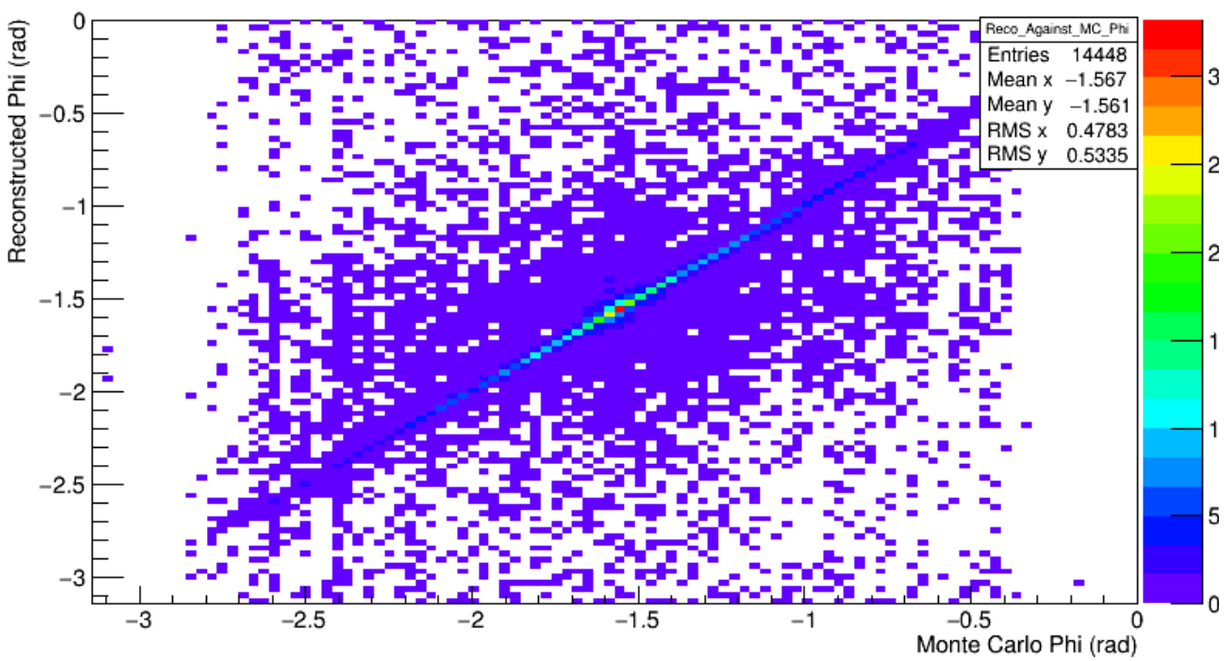


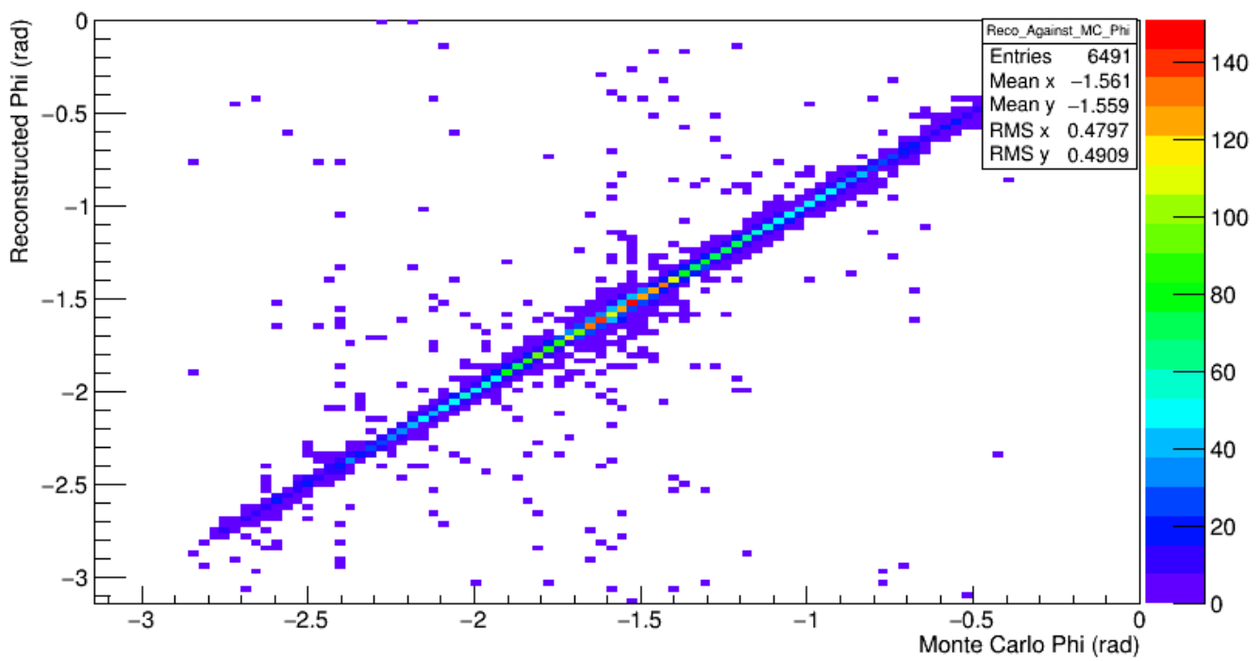


Reco vs True Phi - All









Conclusions

- MCC 3.0 is complete and files are ready to be used.
  - Cosmic Tracker is much improved, as is disambiguation.
  - Calorimetry is tuned correctly.
  - Pandora and CosTrk both have high efficiencies. Pandora appears to be better at longer tracks, CosTrk better at shorter ones.
  - CosTrk makes many tracks from delta rays, and mis-identifies track angles much more than pandora does.
  - Tracks which Pandora makes are much less spread out over phase space (eg track length, theta, phi) than those that CosTrk makes.