Track and Vertex Refitting Tools

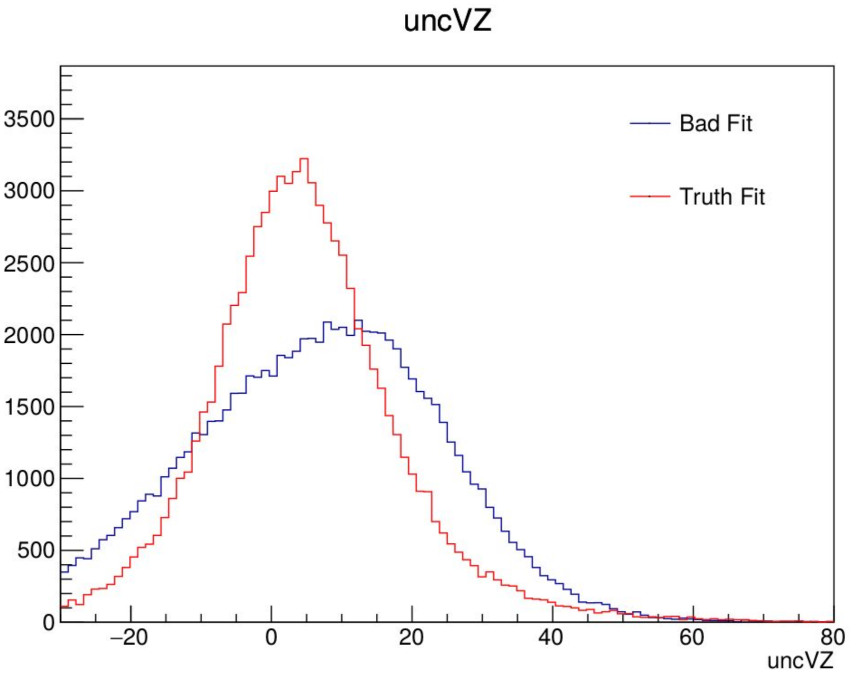
Matt Solt
9/16/2019

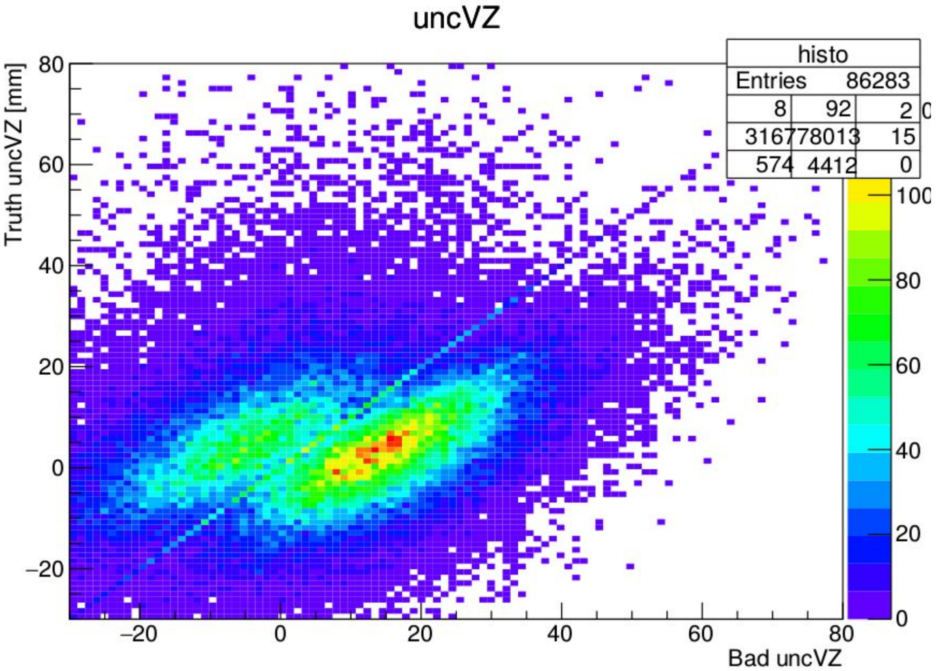
Track/Vertex Refitting Introduction

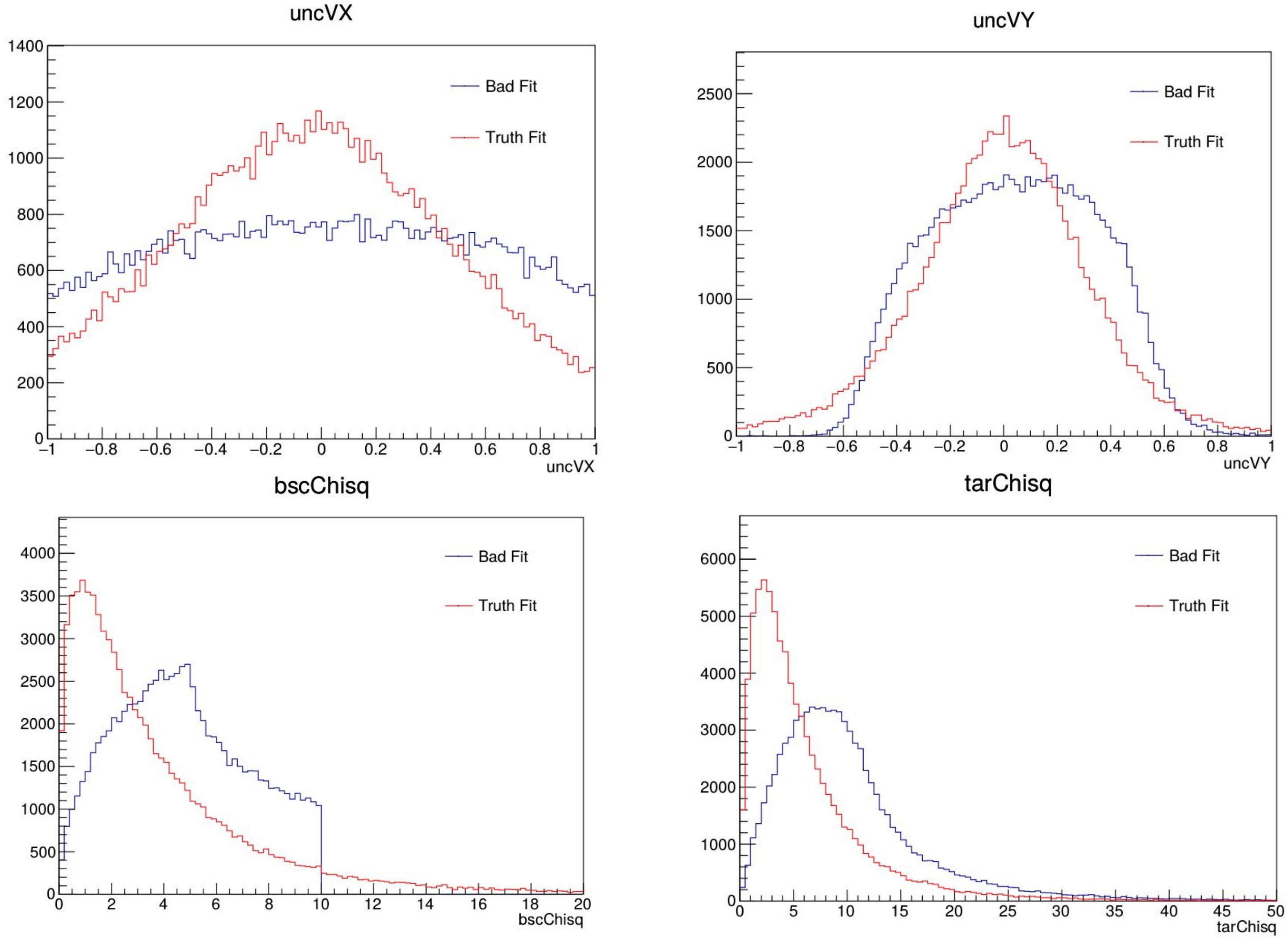
- I have developed a (mostly) complete package for refitting tracks/vertices
- Identify bad tracks and refitting both track/vertex with truth hits
    - Previously showed brief preview. Update: I ran over the entire 2015 MC
    - I have full truth info (e.g. scattering, MCParticle Truth) including truth for the particle responsible for the bad hit
- Refitted tracks/vertices for all tracks using all hits in the first layer
- The goal is to find a reasonable criteria for selecting the “correct” track and V0 particle, or reject the event entirely
- These tools are also useful for tracking, and I recruited some help for the future of this project (PF)

Mistracked V0s vs V0s With Only Truth Hits

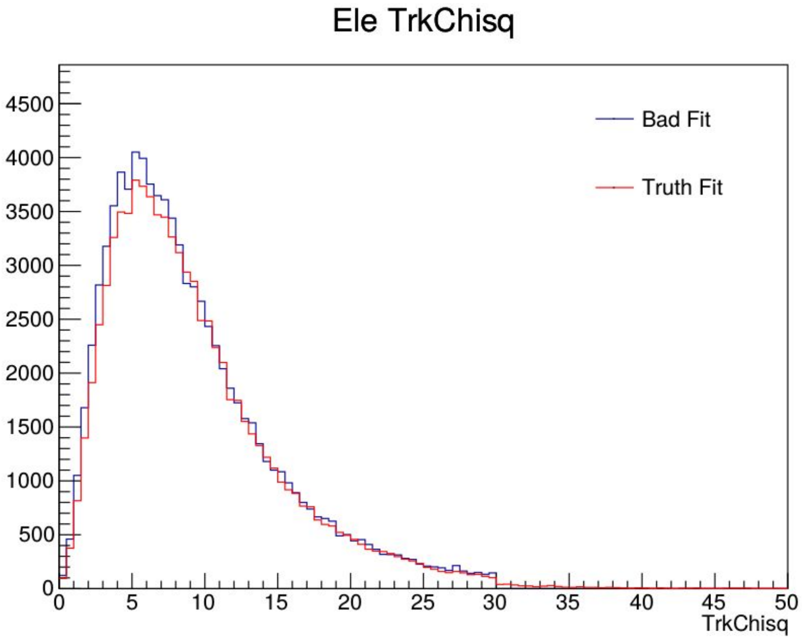
- “Bad” means original track/V0 with mistracks. “Truth” means refit with truth hits

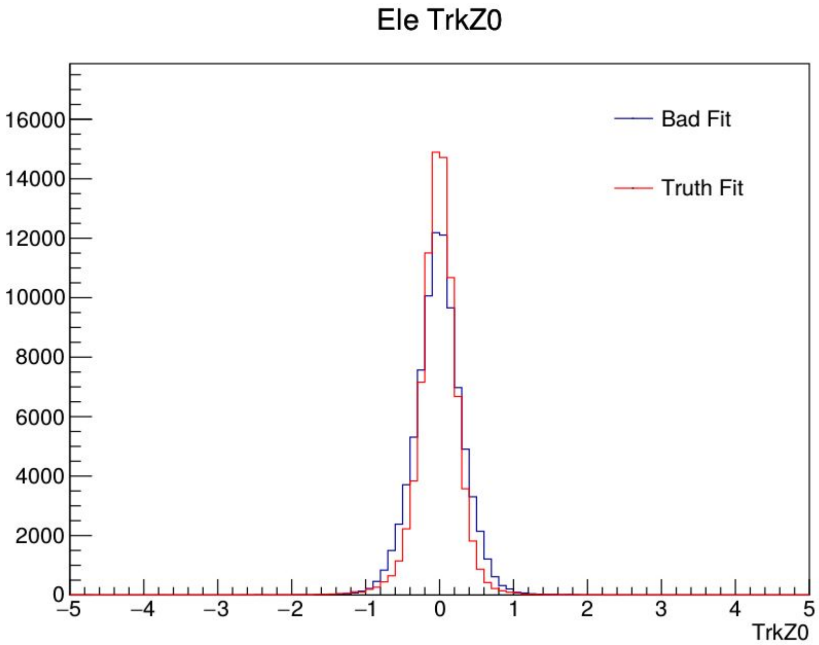





uncVX
1400
1200
1000
800
600
400
200
0
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
uncVX
Bad Fit
Truth Fit
uncVY
2500
2000
1500
1000
500
0
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
uncVY
Bad Fit
Truth Fit
bscChisq
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20
bscChisq
Bad Fit
Truth Fit
tarChisq
6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
40
45
50
tarChisq
Bad Fit
Truth Fit

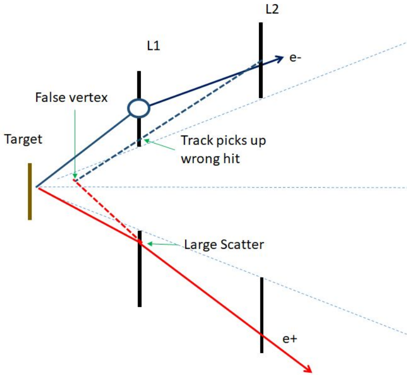
Tracking Distributions



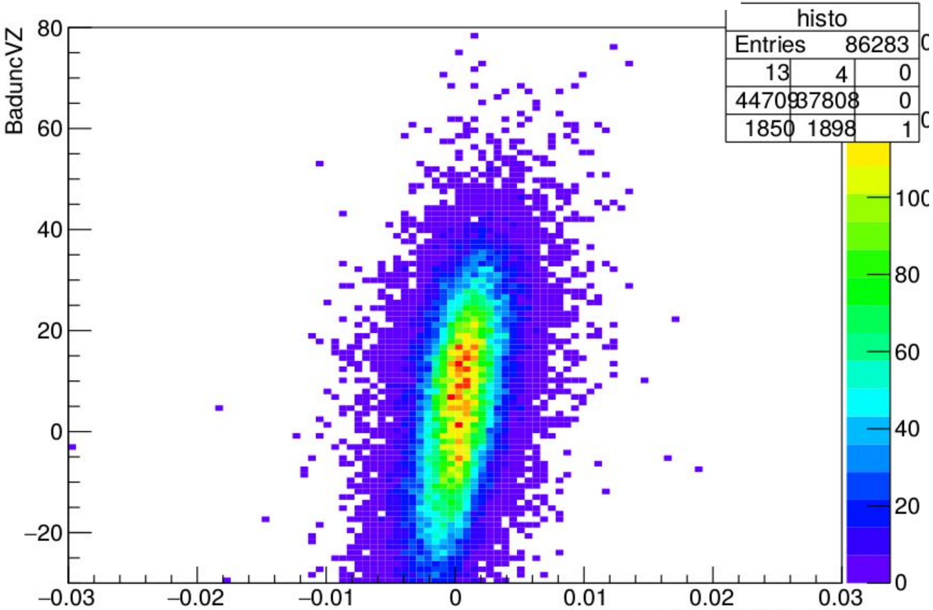


I also have kink distributions, but the refit distributions look funny so I think I have a bug

Bad/Truth Vz vs. L1 Scattering

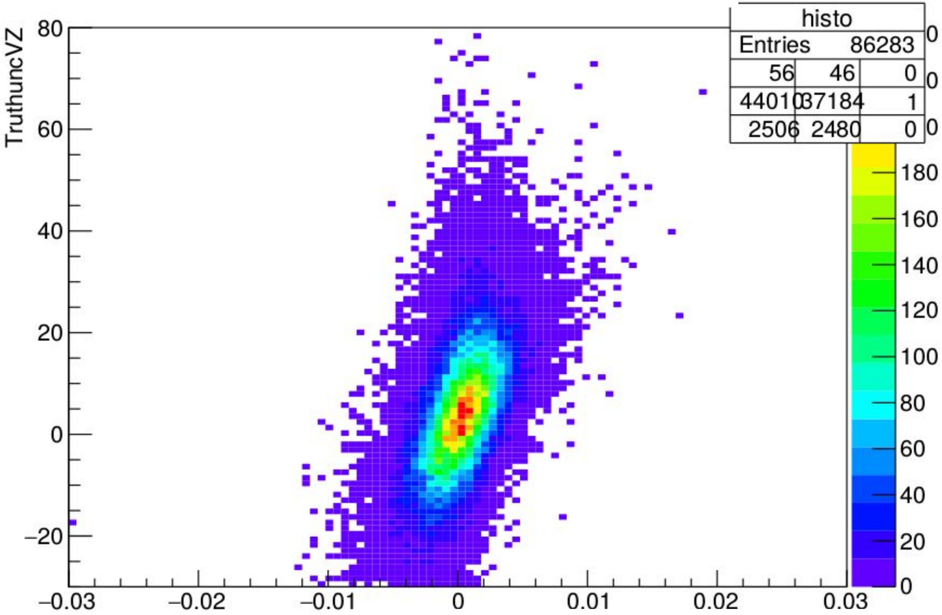


BadunclVZ vs L1 Scatter



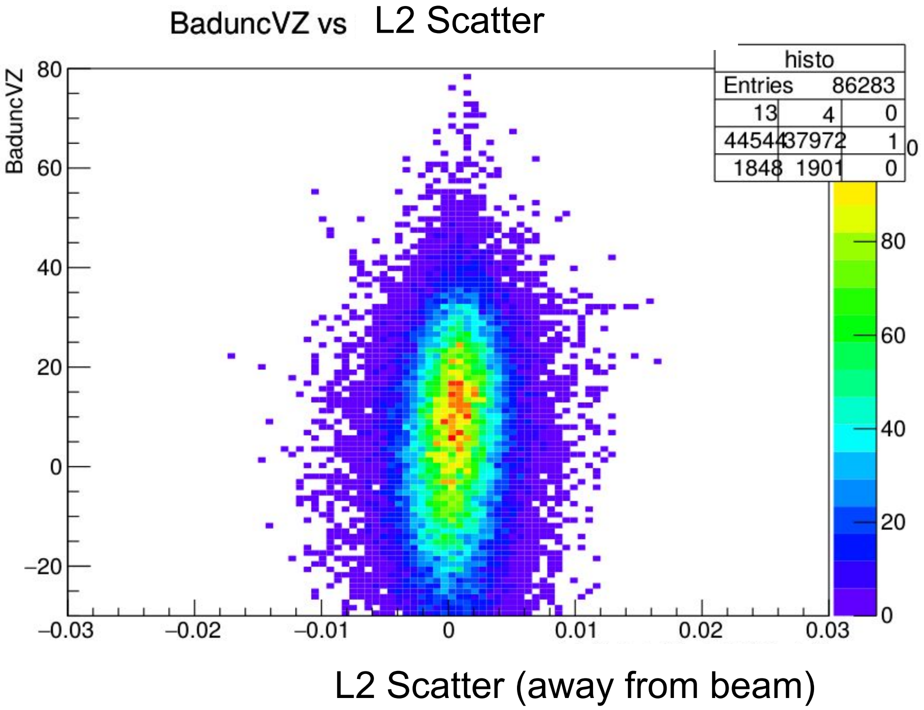
L1 Scatter (away from beam)

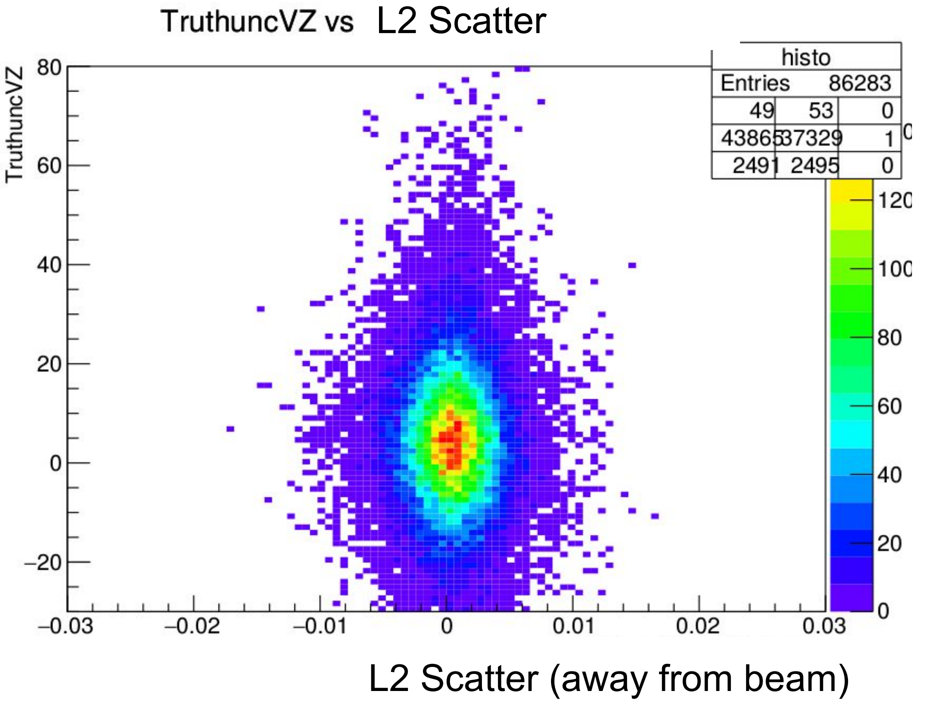
TruthuncVZ vs L1 Scatter



L1 Scatter (away from beam)

Bad/Truth Vz vs. L2 Scattering

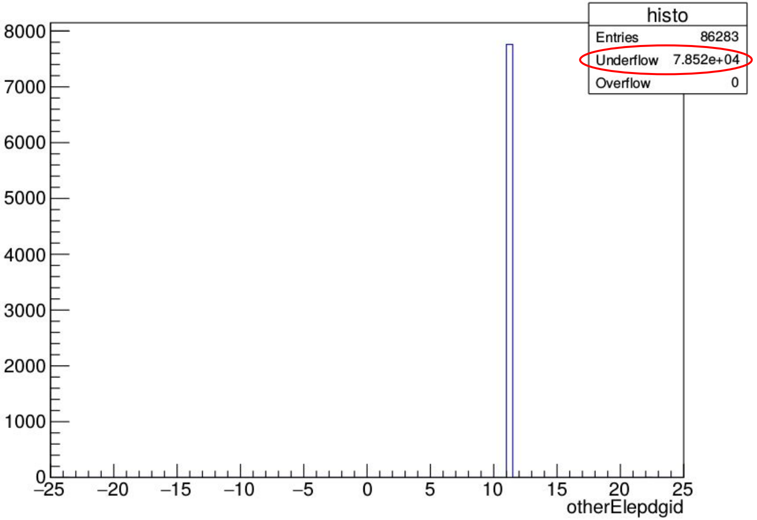


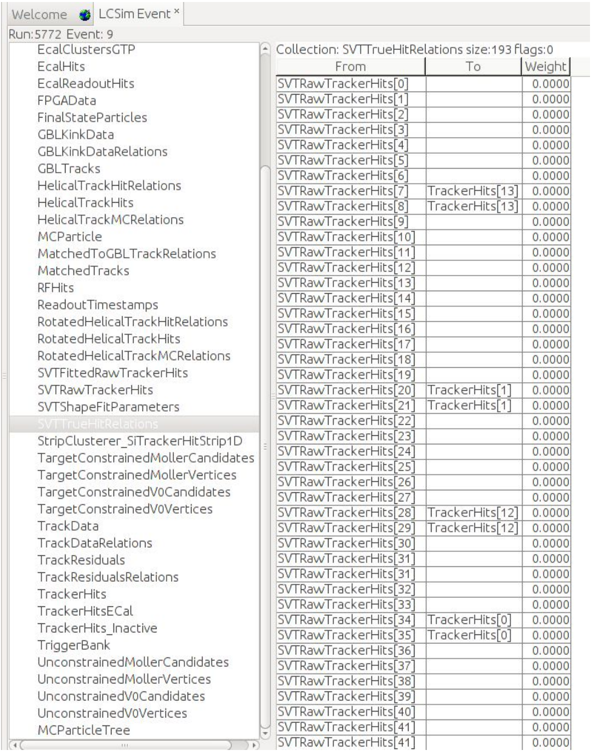


Looking at the Particle Responsible for the Bad Hit

- RawTrackerHits are not always associated with SimTrackerHits. Why? Ideas?
- This appears to be fixed with 2016 MC.

“Bad” Particle PDGID





Run:5772 Event: 9

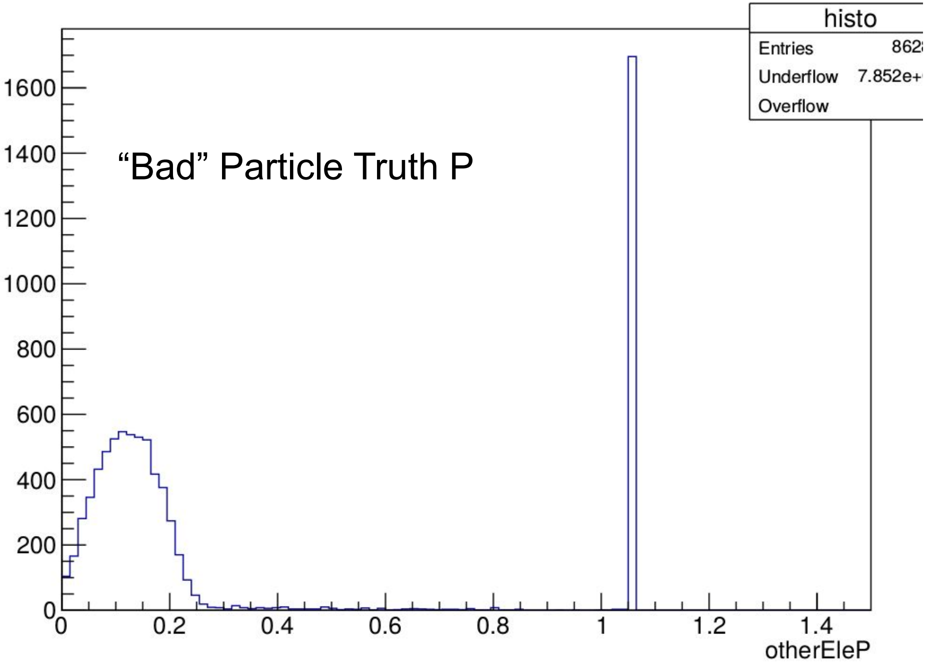
EcalClustersGTP
EcalHits
EcalReadoutHits
FPGAData
FinalStateParticles
GBLKinkData
GBLKinkDataRelations
GBLTracks
HelicalTrackHitRelations
HelicalTrackHits
HelicalTrackMCRelations
MCParticle
MatchedToGBLTrackRelations
MatchedTracks
RFHits
ReadoutTimestamps
RotatedHelicalTrackHitRelations
RotatedHelicalTrackHits
RotatedHelicalTrackMCRelations
SVTFittedRawTrackerHits
SVTRawTrackerHits
SVTShapeFitParameters
SVTTrueHitRelations
StripClusterer_SiTrackerHitStrip1D
TargetConstrainedMollerCandidates
TargetConstrainedMollerVertices
TargetConstrainedV0Candidates
TargetConstrainedV0Vertices
TrackData
TrackDataRelations
TrackResiduals
TrackResidualsRelations
TrackerHits
TrackerHitsECal
TrackerHits_Inactive
TriggerBank
UnconstrainedMollerCandidates
UnconstrainedMollerVertices
UnconstrainedV0Candidates
UnconstrainedV0Vertices
MCParticleTree

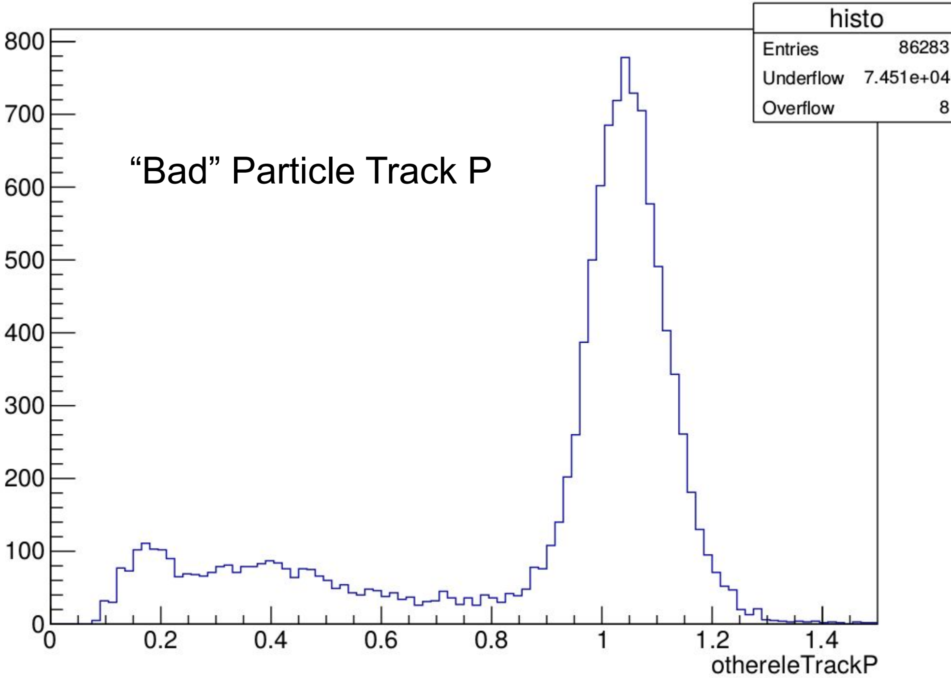
Collection: SVTTrueHitRelations size:193 flags:0

From	To	Weight
SVTRawTrackerHits[0]		0.0000
SVTRawTrackerHits[1]		0.0000
SVTRawTrackerHits[2]		0.0000
SVTRawTrackerHits[3]		0.0000
SVTRawTrackerHits[4]		0.0000
SVTRawTrackerHits[5]		0.0000
SVTRawTrackerHits[6]		0.0000
SVTRawTrackerHits[7]	TrackerHits[13]	0.0000
SVTRawTrackerHits[8]	TrackerHits[13]	0.0000
SVTRawTrackerHits[9]		0.0000
SVTRawTrackerHits[10]		0.0000
SVTRawTrackerHits[11]		0.0000
SVTRawTrackerHits[12]		0.0000
SVTRawTrackerHits[13]		0.0000
SVTRawTrackerHits[14]		0.0000
SVTRawTrackerHits[15]		0.0000
SVTRawTrackerHits[16]		0.0000
SVTRawTrackerHits[17]		0.0000
SVTRawTrackerHits[18]		0.0000
SVTRawTrackerHits[19]		0.0000
SVTRawTrackerHits[20]	TrackerHits[1]	0.0000
SVTRawTrackerHits[21]	TrackerHits[1]	0.0000
SVTRawTrackerHits[22]		0.0000
SVTRawTrackerHits[23]		0.0000
SVTRawTrackerHits[24]		0.0000
SVTRawTrackerHits[25]		0.0000
SVTRawTrackerHits[26]		0.0000
SVTRawTrackerHits[27]		0.0000
SVTRawTrackerHits[28]	TrackerHits[12]	0.0000
SVTRawTrackerHits[29]	TrackerHits[12]	0.0000
SVTRawTrackerHits[30]		0.0000
SVTRawTrackerHits[31]		0.0000
SVTRawTrackerHits[31]		0.0000
SVTRawTrackerHits[32]		0.0000
SVTRawTrackerHits[33]		0.0000
SVTRawTrackerHits[34]	TrackerHits[0]	0.0000
SVTRawTrackerHits[35]	TrackerHits[0]	0.0000
SVTRawTrackerHits[36]		0.0000
SVTRawTrackerHits[37]		0.0000
SVTRawTrackerHits[38]		0.0000
SVTRawTrackerHits[39]		0.0000
SVTRawTrackerHits[40]		0.0000
SVTRawTrackerHits[41]		0.0000
SVTRawTrackerHits[41]		0.0000

Looking at the Particle Responsible for the Bad Hit

- Nice FEE peak! (and possibly recoil electron peak)





Future and Questions to Answer (Using Truth)

- What tracking/vertexing variables are useful?
- How can we determine what features (and methods) are useful for selecting the “correct” track/V0? Will this be computationally feasible?
- What is reasonable to accomplish for 2016 analysis? What do we want to use for 2019 data?
- How will this impact displaced A’ (as a function of z)?
- Info about a bad track may be contained in the corresponding track (assuming they both came from the target).
    - We must also be careful not to assume too much!

Track/Vertexing Refitting With First Hits

- I have also written drivers that refit tracks/vertices without truth
  - For every track, find the layer with the first hit (either L1 or L2).
  - Grab all 3D hits in the layer that are not associated with the track.
  - Refit using GBL all the hits on this track (except the first hit) with each of these other 3D hits in the event. Map these new GBL tracks to the original GBL track.
  - Refit the vertices with these new GBL tracks.
- The goal here is to come up with a criteria select the “correct” vertex or eliminate the event (PF will help me).
- This is overkill, but it’s a good start to eventually doing something smart
- PF has some very preliminary results coming soon. He also had the idea of looking at raw hit and cluster information
  - He also (rightfully) says that we are biasing tracks with seed 123. These studies separated by different seeds would also be interesting

Conclusion

- We now have the ability to refit tracks and vertices to fully analyze
    - We can filter out bad tracks and refit tracks/V0s with truth hits
    - We have the ability to refit tracks/vertices using arbitrary 3D hits in the first layer
    - The goal is to use all available info to make decisions of “correct” V0s/tracks, or eliminate the event
    - I am most interested in reducing high Z events due to mistracks for vertexing analysis, but this could be useful for tracking purposes also
- This obviously needs a close look with A’ MC (as a function of z)
- Need to think about dealing with inefficiencies in data
- I (admittedly) hacked this together. As we explore more what is actually useful from these studies, we can make a more robust package.