

Tracking Meeting

PF, Cameron, MattS

09/16/2019



U.S. DEPARTMENT OF
ENERGY

Stanford
University

SLAC NATIONAL
ACCELERATOR
LABORATORY

- Updates on Hipster software
- Clusters on track charge distributions for HV bias runs
- Some plots on Refit tracks procedure
- Comments and next steps

Analysis Software updates

- Hipster is a software developed by Omar to produce ROOT ntuples containing objects such as tracks, particles, clusters, etc from LCIO files
- Analysis skeleton in place:
 - Each task can be performed by a different processor
 - General configuration from python file
- **Updates:**
 - Fixed Cmake compilation for MacOS
 - Added some histogramming classes to produce basic histograms and new processors
 - Added functionality to run the same scheme starting from ROOT files to produce ROOT histograms
 - Started developing processors/tools for offline analysis
 - Branch available in <https://github.com/pbutti/hpstr -b ntupleProcessing>

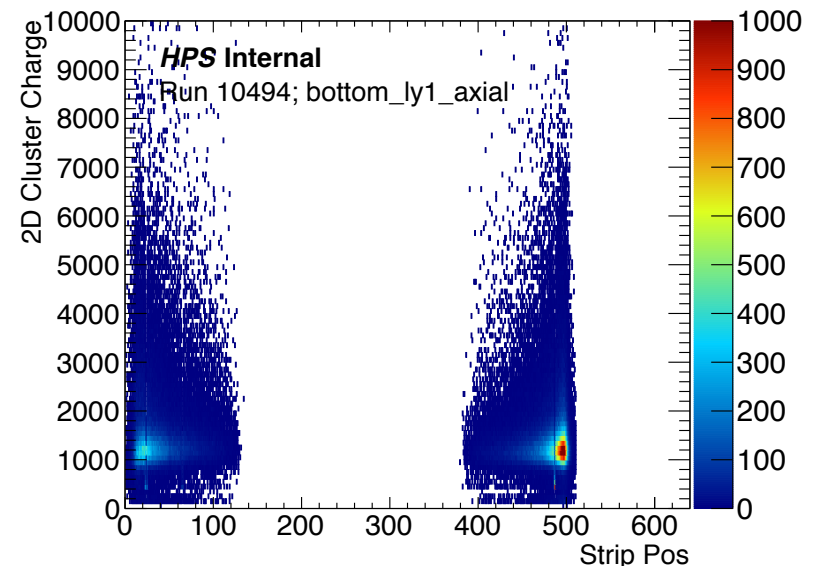
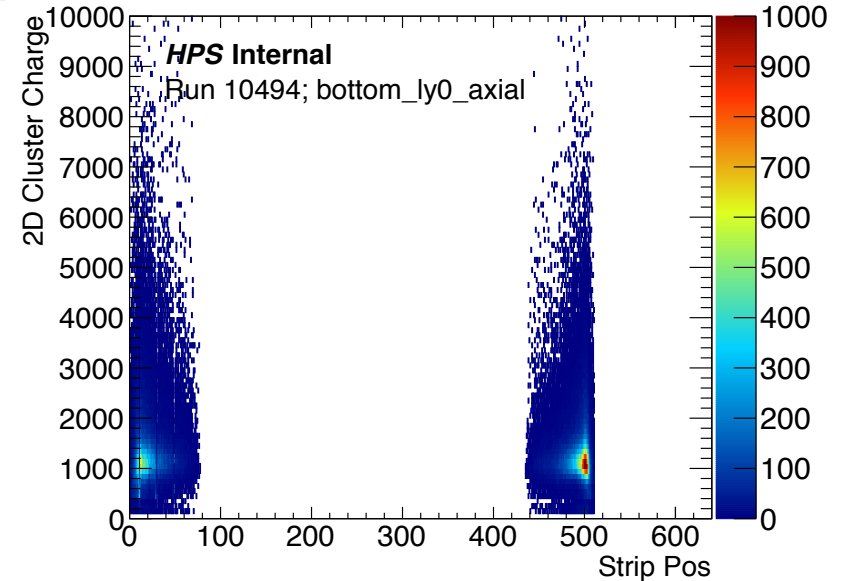
Cluster on track charge checks on recent HV scans

- Request to study charge distribution along y-direction on SVT modules in 2019 data
- Tim checked the distributions of charge vs Y in 10030
 - He applies charge >600e-
 - Cluster time within +/- 40 ns
 - Fit bins >500 entries to gaussian + p1 and plot gaussian mean vs Y
- I used
 - 10491/10492/10494 HV bias scans. Not fully sure of 10494 run conditions but should reflect production runs in that period
 - Plan to use more recent bias voltage scans but I have issues ssh to ifarm

Run/ HV	L0-1 (V)	L2-3 (V)	L4-6 (V)
10491	70	270	180
10492	40	100	180
10494	?	?	?

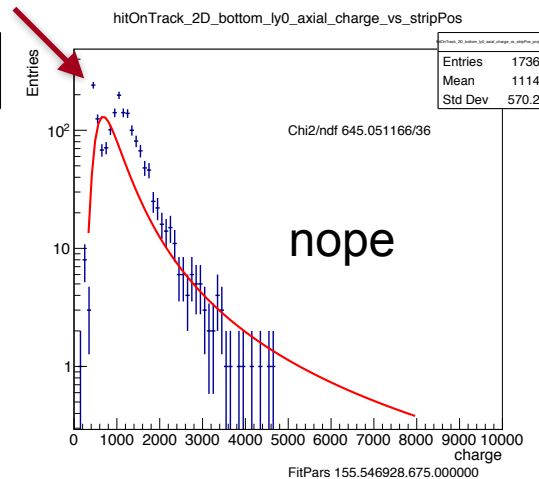
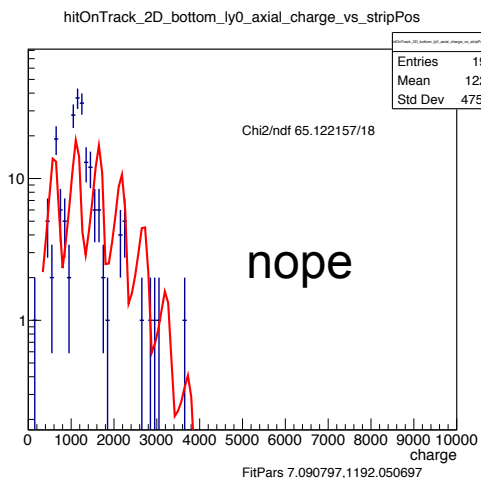
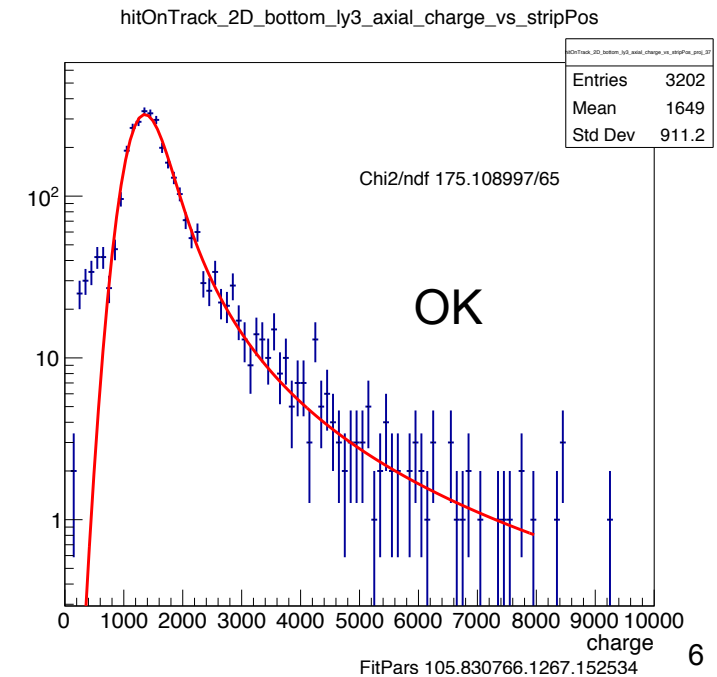
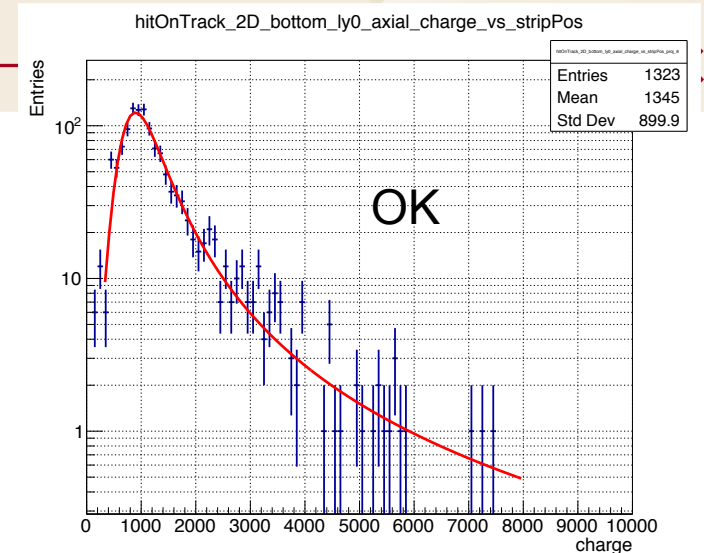
Cluster on track distribution

- **All tracks** passed reconstruction **are used** for these plots, no quality cuts applied.
- From tracks get the 3D clusters then obtain raw information and fits
 - **No quality cuts** are applied on clusters.
 - **No correction for track incident angle** applied.
- 2D cluster charge and position respectively obtained from:
 - **Sum of charge of 1D strips in the 2D cluster**
 - **Charge weighted average of the strip positions**
- This allows us to get the cluster position in terms of strip number.

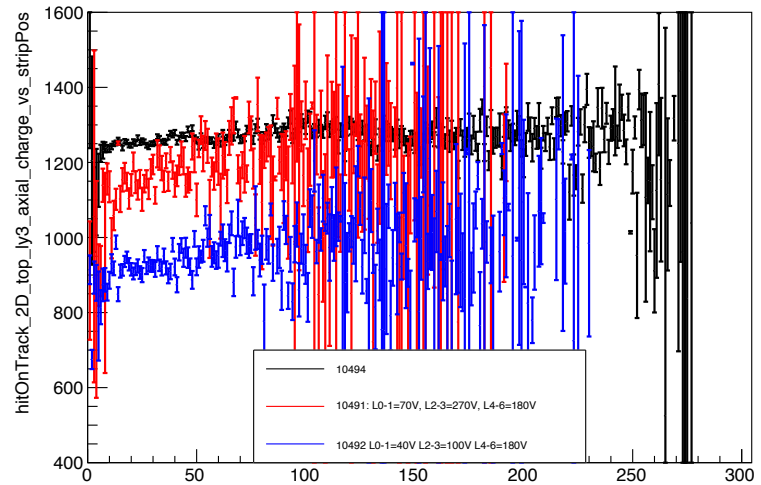
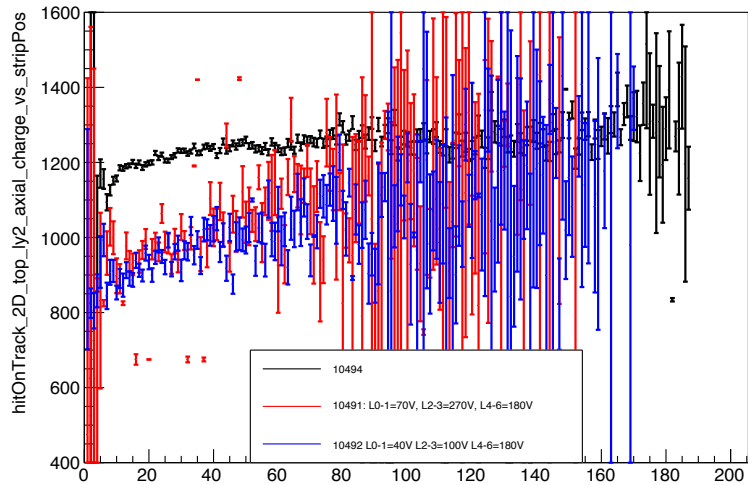
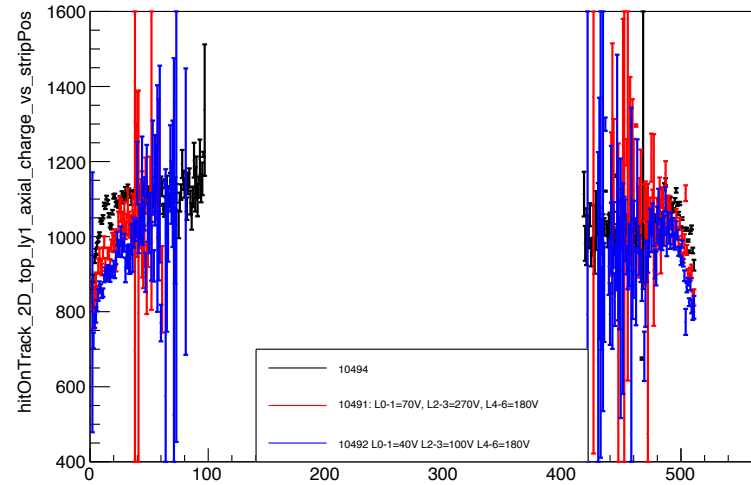
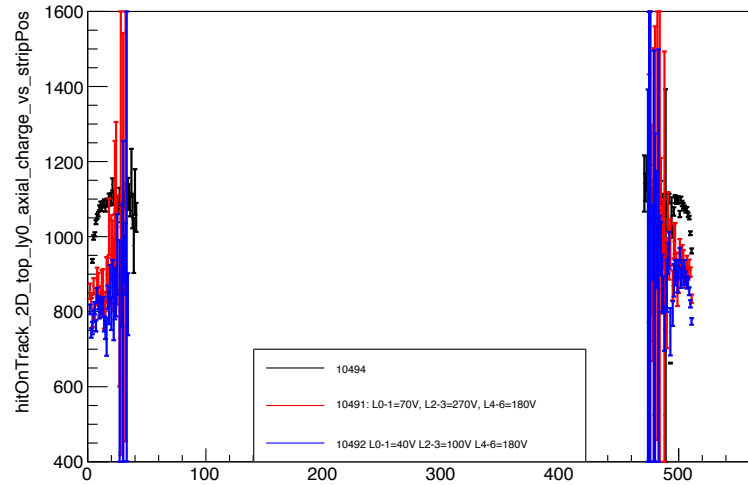


Langau fit

- Charge distribution fit is made with a landau convoluted with a gaussian
 - Reasonable modeling of the charge distribution
- >25 events per projection to perform the fit (need to be revisited..?)
- Largest part of the fits are OK
 - Very few failed fits due to strange channels behavior (such as very high low-charge occupancy)

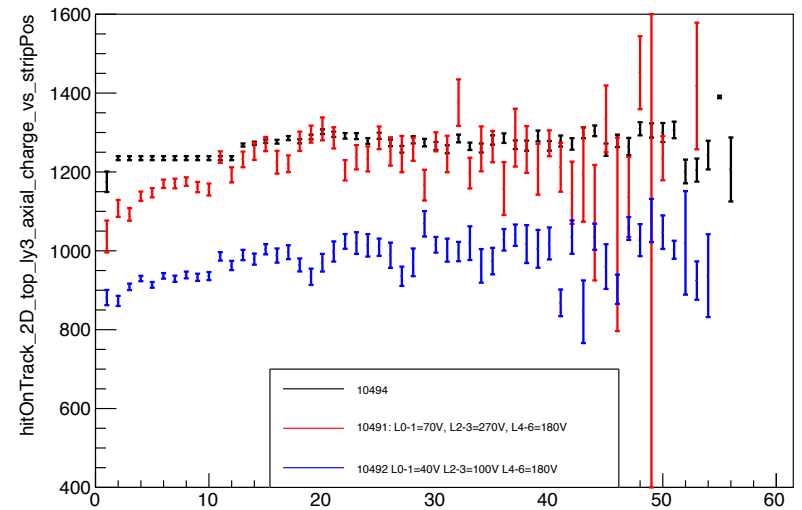
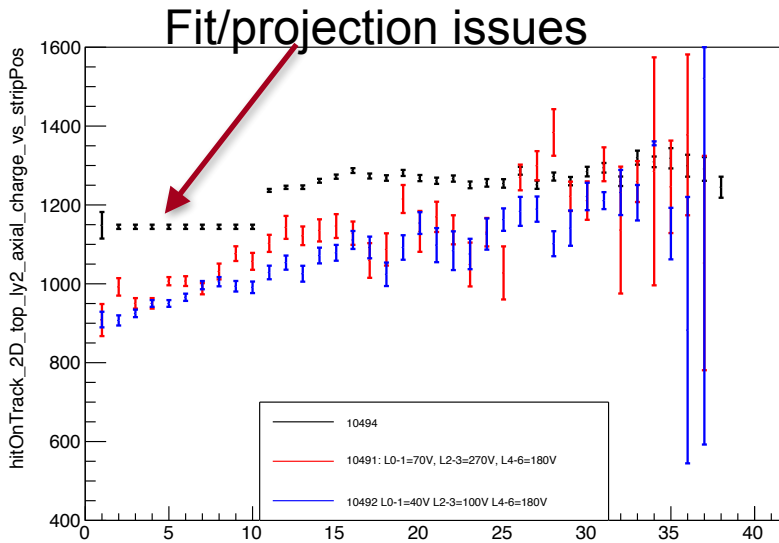
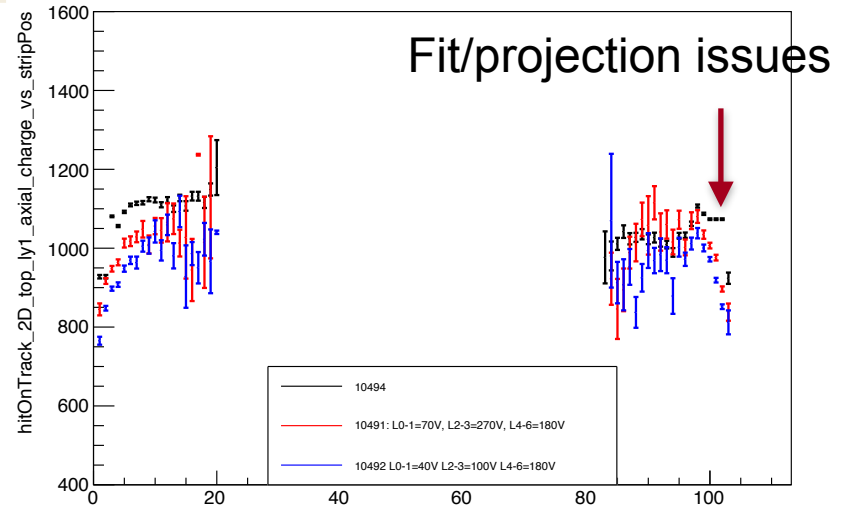
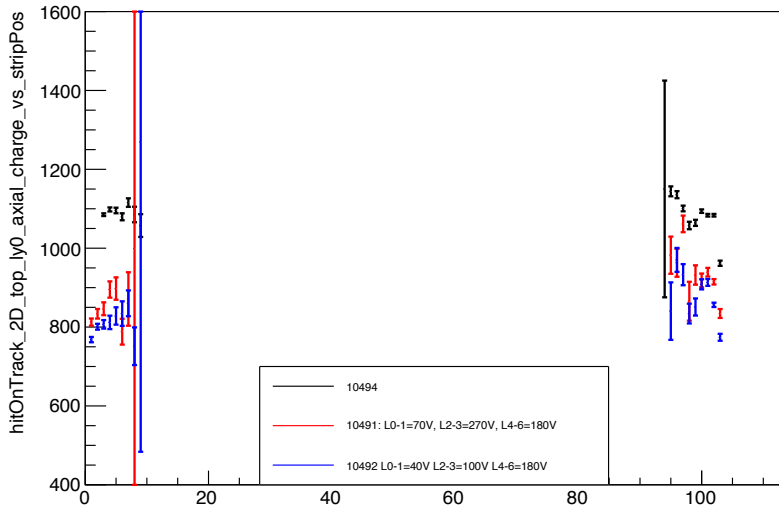


Cluster charge vs stripNumber



Cluster charge vs stripNumber

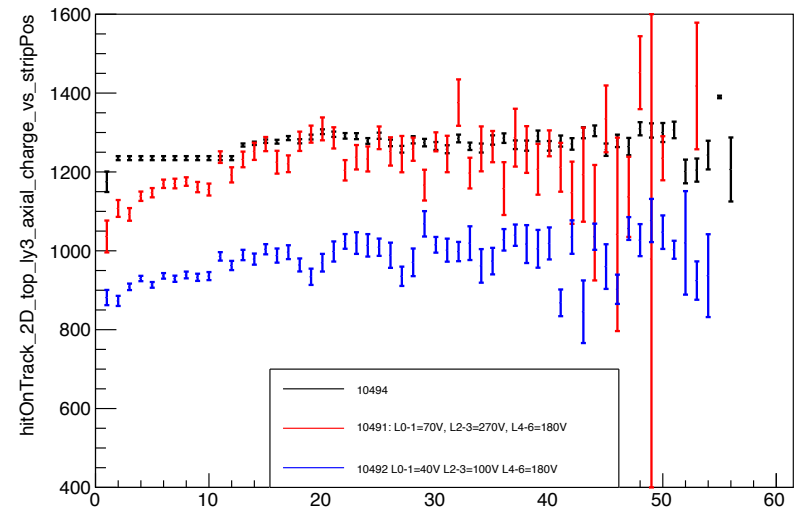
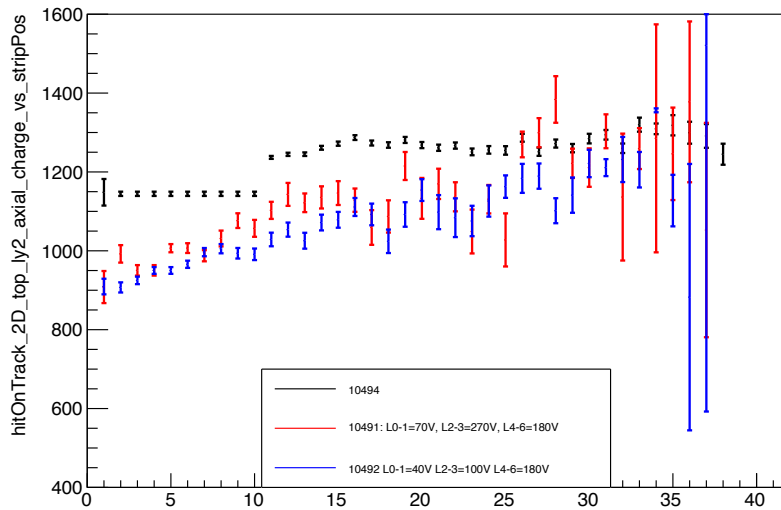
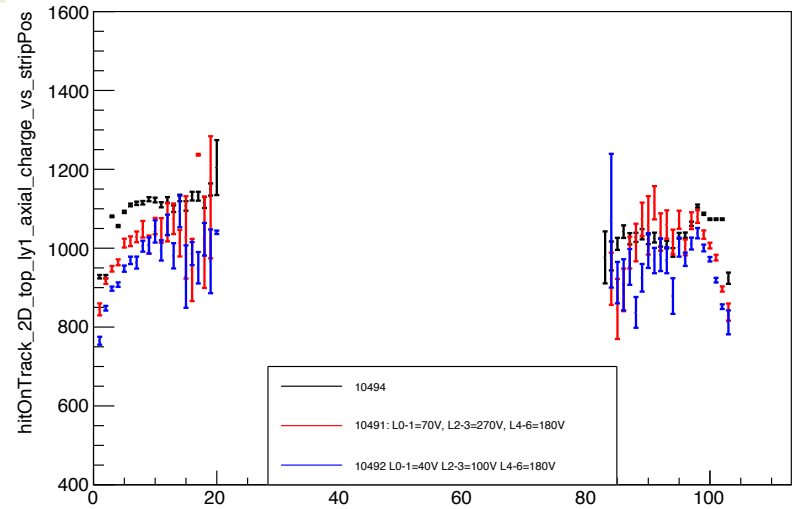
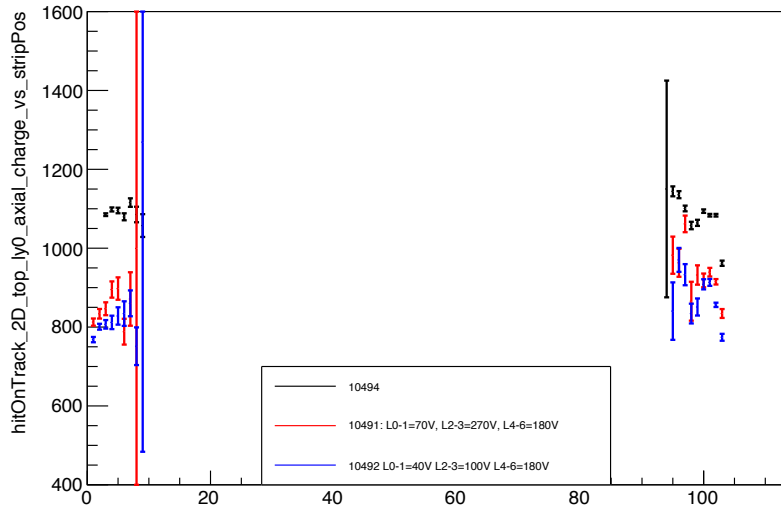
SLAC



Strips grouped in groups of 5.

Cluster charge vs stripNumber

SLAC



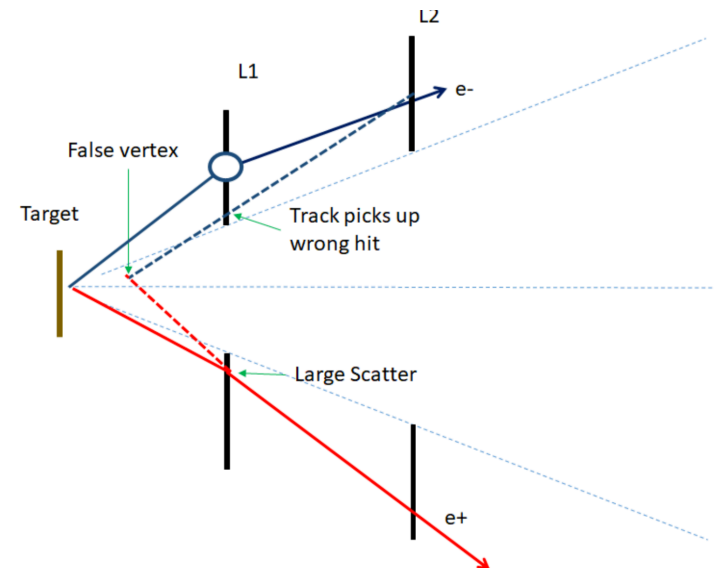
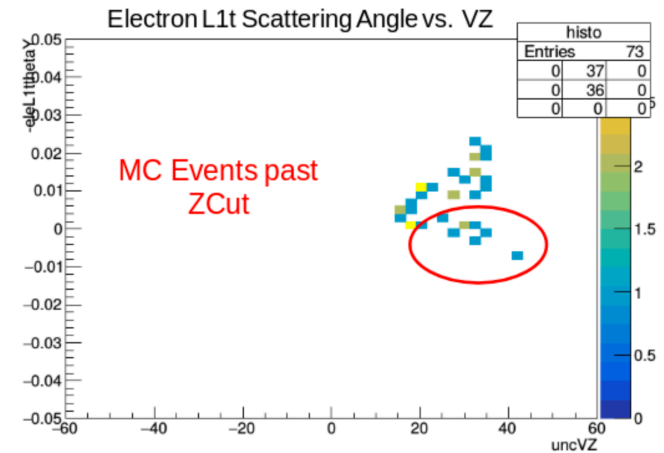
Strips grouped in groups of 5.

Comments and plans

- Planning to check latest HV bias scans:
 - 10711/2/3/4
 - Had issues ssh to the ifarm to retrieve the data past days
- Obtain plot of charge collection fraction vs HV
- Run-by-run evolution of charge collection fraction per module
- Correct cluster charge by incidence angle (wrt 3D cluster)

Refit Tracks

- RECAP: Motivation:
 - Part of the events with $z > Z_{cut}$ comes from event with scattering angle after L1 pointing toward the beam
 - These events seem to come from tracks picking up the wrong hit il L1
- RECAP : What's in the sample
 - For every track find the layer with the first hit
 - Grab all 3D hits in the layer that aren't associated with the track
 - Refit the GBL using the other 3D hits in the event and form a new collection of tracks. Each refit is linked to the original GBL track
 - Refit the vertices using the GBL tracks

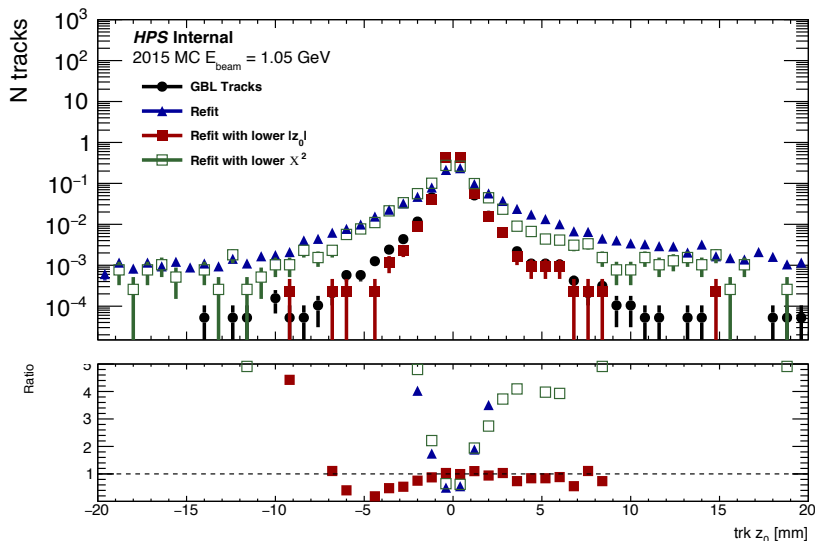
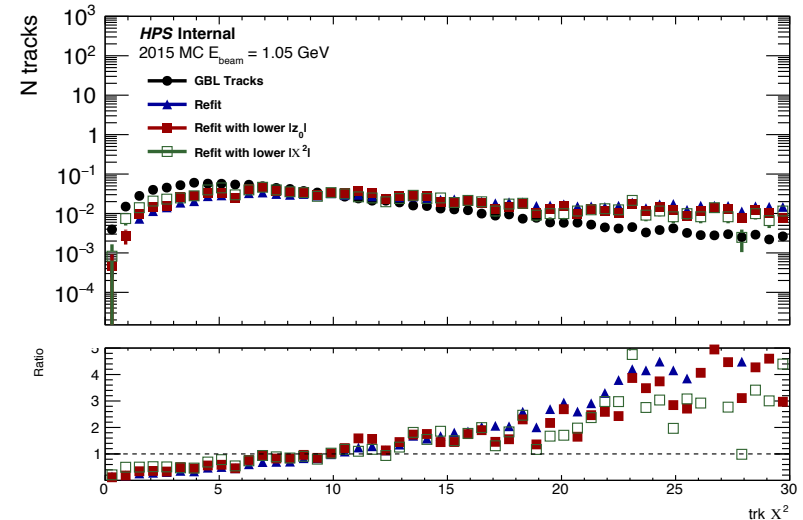


Refit Tracks checks

- Requested to check the refit track sample produced by MattS
- 2015 MC geometry with beam $E=1.05$ GeV
- Use of hipster analysis code as hard to use MattS' flat Ntuple format
 - Written new LCIO processor to get the refitted tracks linked to the GBL tracks.
 - Made small code for checking basic track properties.
 - No selection on GBL tracks for the moment
 - No check on vertices yet

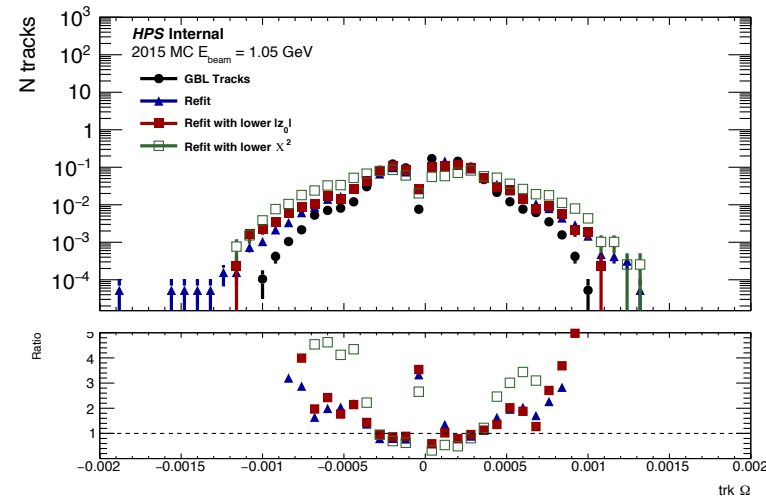
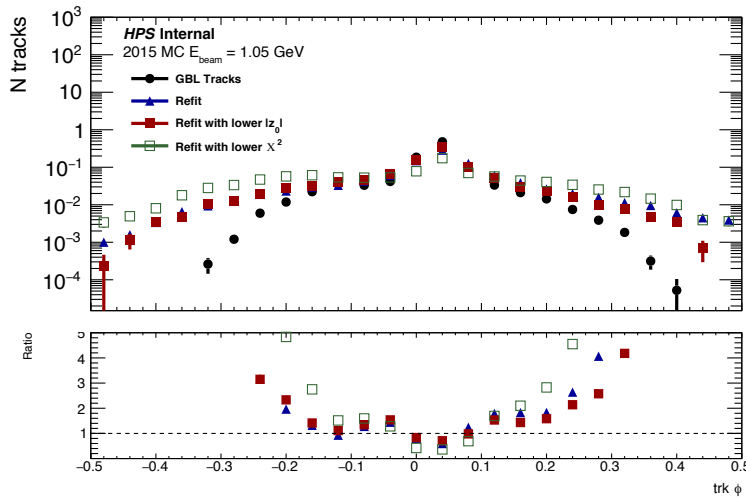
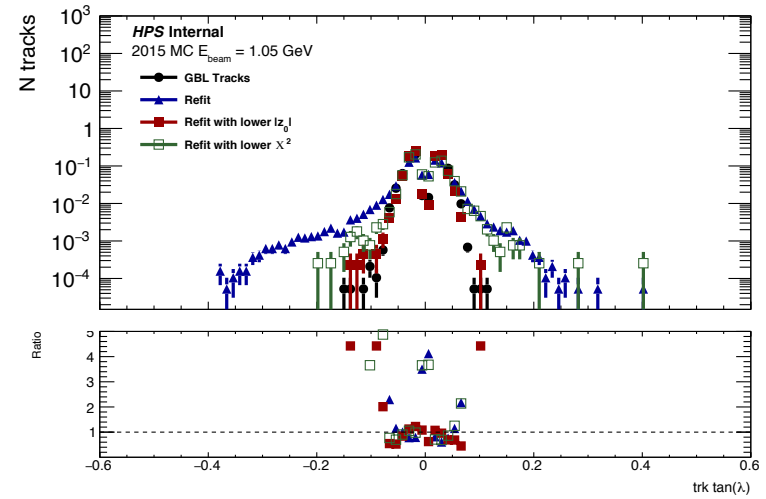
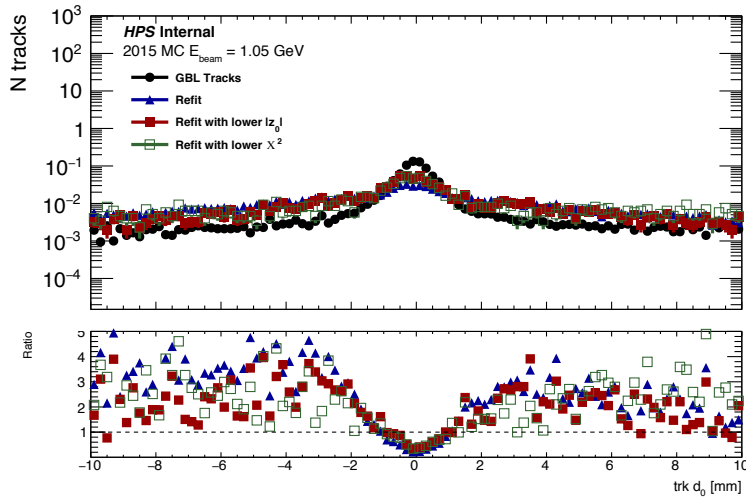
Track properties comparison

- For each GBL multiple refit tracks can be created (according to how many hits are present in L1)
- Only the refit-track with the best X2 is kept
 - No cut on refit track X2 done yet
- 4 Tracks categories:
 - GBL Tracks
 - Refit tracks
 - Refit tracks with $|z_0| < \text{GBL Track } |z_0|$
 - Refit tracks with $X_2 < \text{GBL Track } X_2$

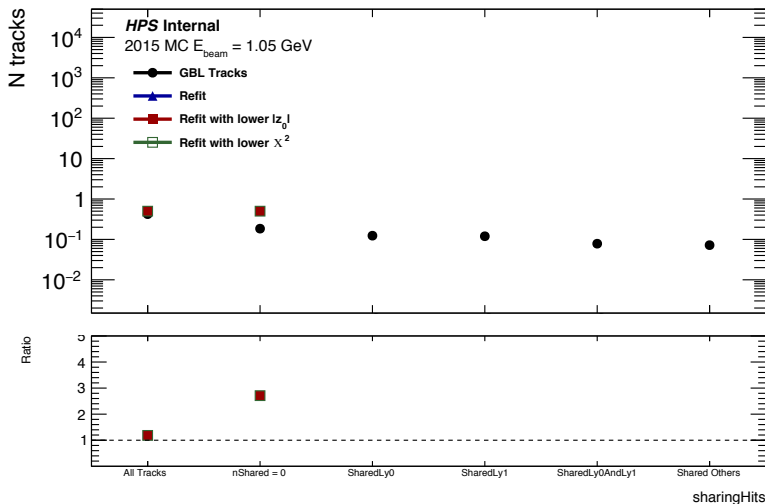
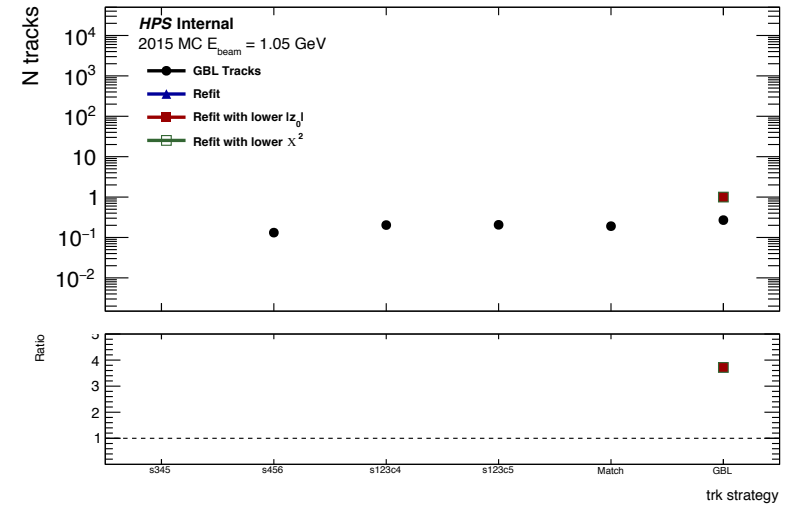
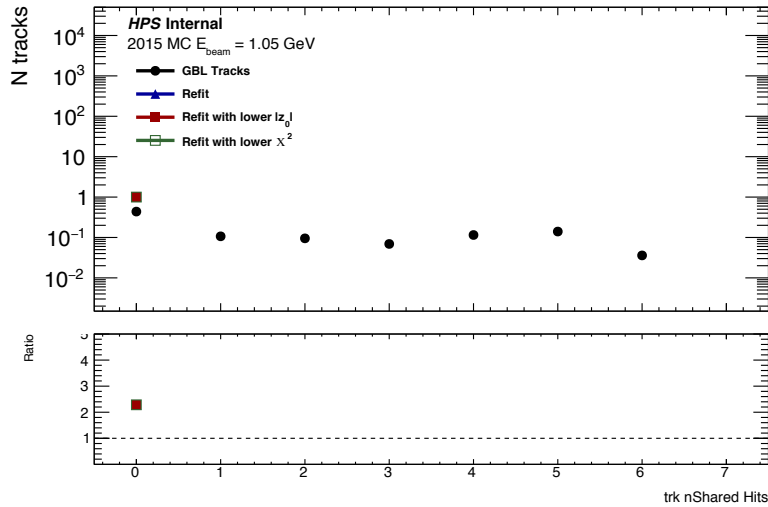


Only GBL tracks that have at least 1 refit are selected
Only the “best” X_2 refit is used
 X_2 distribution gets larger tails, z_0
Distribution generally widened wrt (0.0.0)

Track properties comparison



Check on track strategy and shared hits



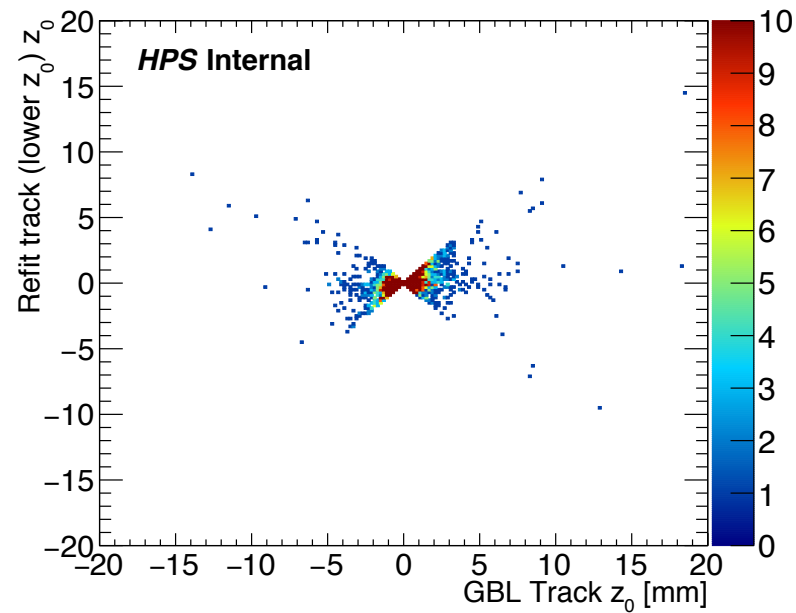
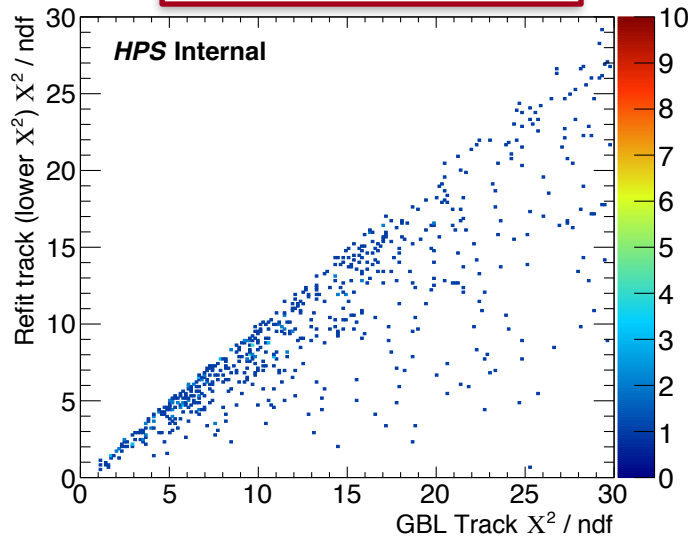
- GBL Tracks can have large number of shared hits (up to 6, same track different strategy?)
- Innermost and second innermost layer tend to have a non-negligible fraction of shared hits. Often both layers have shared hit.
- L1-Refit on 123 seeded tracks shouldn't be done as track road is biased from L1 hit
 - Found that s345 tracks are absent in collection: bit-by-bit logic seems OK to set the type
 - No s345 driver or bug in track type setting?

- Performed first check on Refit procedure.
- Only basic checks and plotting fw on track properties, hit content. Basic comparison done.
- Planning to:
 - Improve number of plots produced
 - Check vertex quality, position and other properties
 - More refined event / track selection?
- Some concern when applying this refit procedure:
 - What metric to use to retain refit track?
 - Only tracks that aren't seeded using L1 hits could, in principle, be refitted.
 - Avoid object reconstruction biases.

Backup

2D hosts

Refit $X^2 < \text{GBL } X^2$



Refit $|z_0| < \text{GBL } |z_0|$

