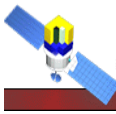
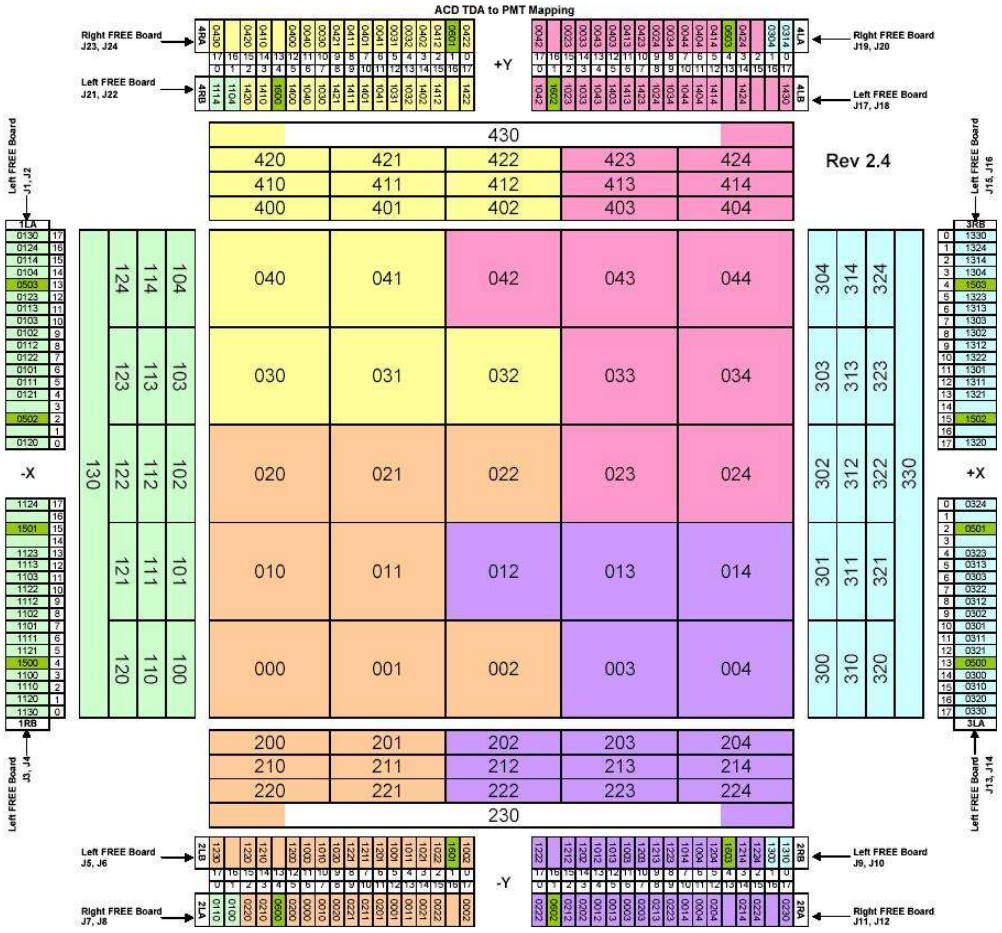
Overview of Acd Reconstruction



- Digitization (AcdDigi in digiRootData)
  - Merge data from different electronics cards into physical space
  - Preserve all data sent off the detector
- Apply MIP Calibration (AcdHit)
  - Remove pedestal (baseline counts) and factor in different electronics gains
  - Express hits in terms of MIPS and MeV
- Extrapolate Tracks to Acd (AcdTkrIntersectionTool)
  - Use GEANT geometry, calculation only as good as input
- Calculate distance from extrapolated track to hit elements and known gaps (AcdRecon / AcdTkrPocaTool)
  - Extract Tile edges and Ribbons locations from GEANT model
  - Calculates miss distance is track doesn't hit element
- Make monitoring plots for SVAC report

Digitization

- Each ACD tile or ribbon is attached to two PMTs.
  - Map from electronic to physical space
    - The tile light collection is designed to equalize between PMTs.
    - This isn't really possible in the ribbons, which are just scintillating fibers attached to PMTs.
    - By design the two PMTs for each element are readout by different FREE cards



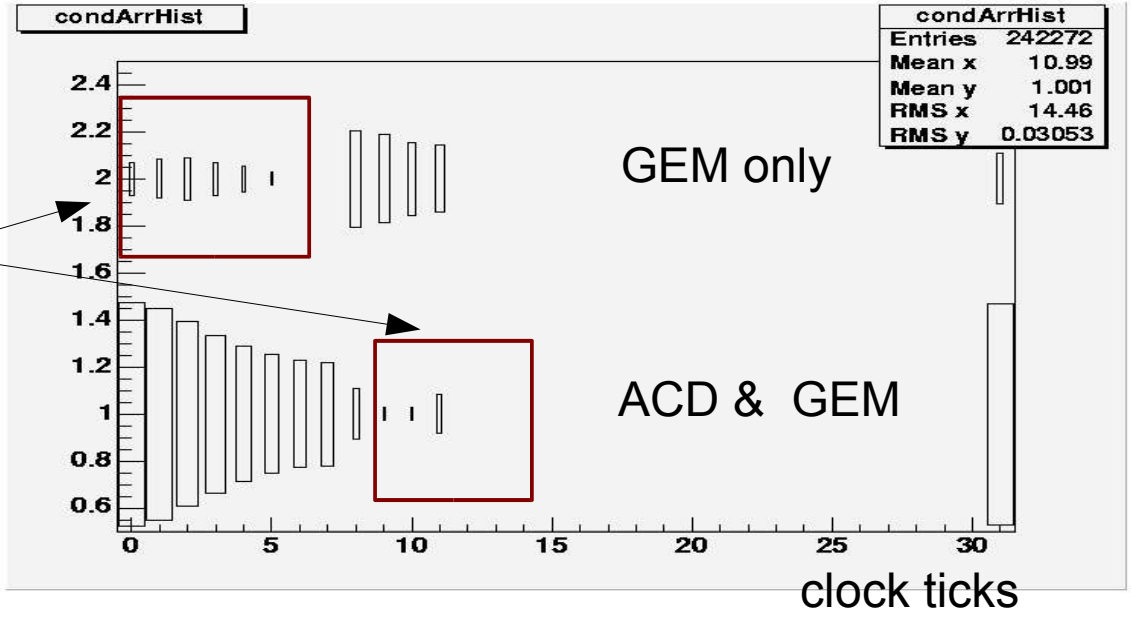
AcdDigi data

- Each AcdDigi has the data for both PMTs
  - PHA (aka pulseHeight or ADC value), (12 bits, 0-4095)
  - Flags
    - range (low or high gain)
    - above Zero suppresion (digital cut) (aka acceptMap bit)
    - above Veto Threshold (analog cut) (aka hitMap bit)
    - above CNO Threshold (analog cut, 1 signal per FREE board)
      - Not being set currently
    - parity errors in data transmission

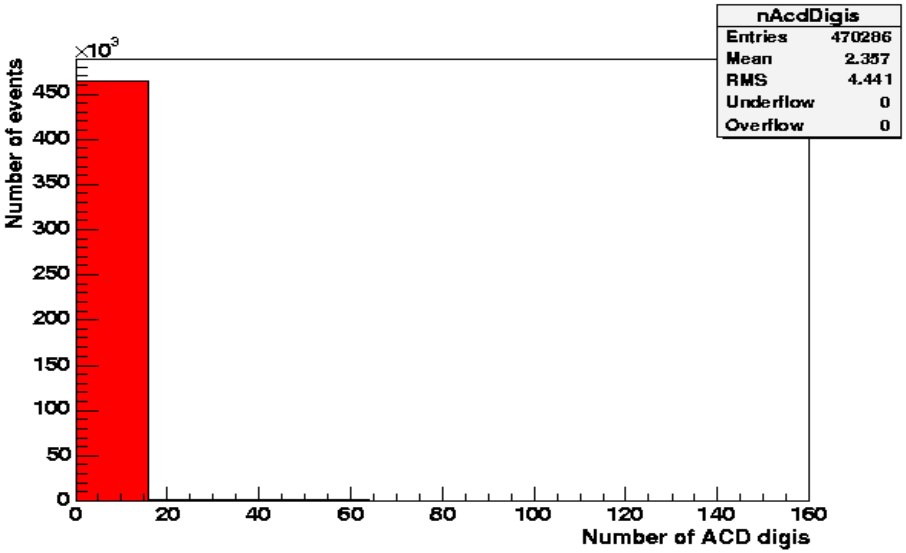
ACD information in GEM

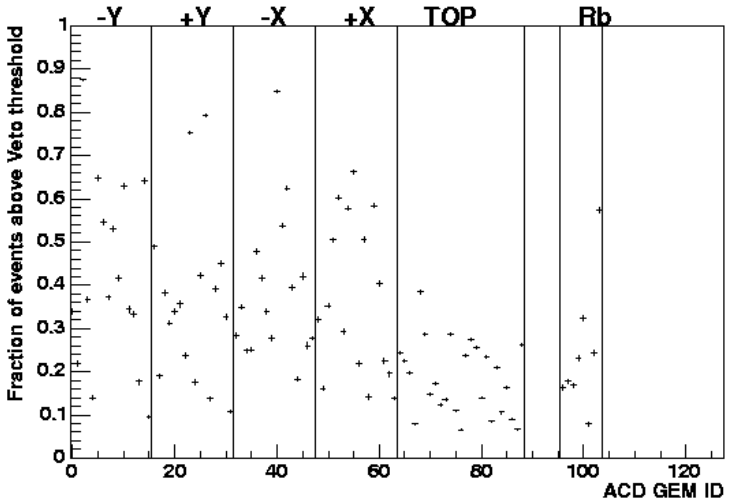
- "or" of Veto signals for channel, but timing is slightly different
- This is in with the GEM data, not the ACD data (so far)
  - If the channel reaches threshold within 6-7 ticks the GEM information should match the ACD

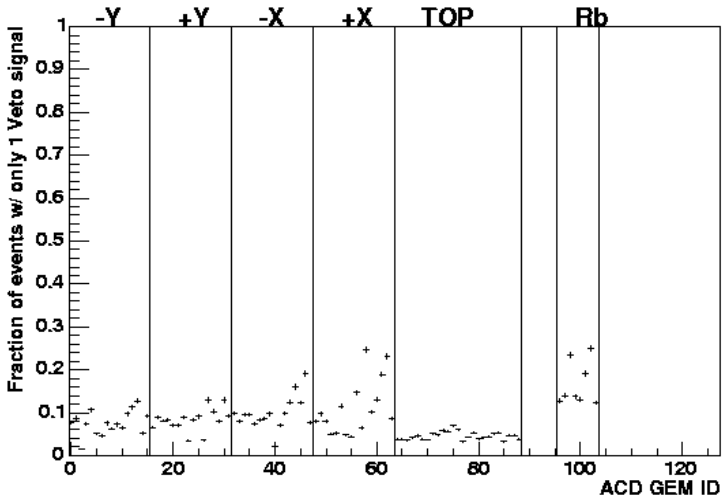
These were traced back to a bug in the digitization process



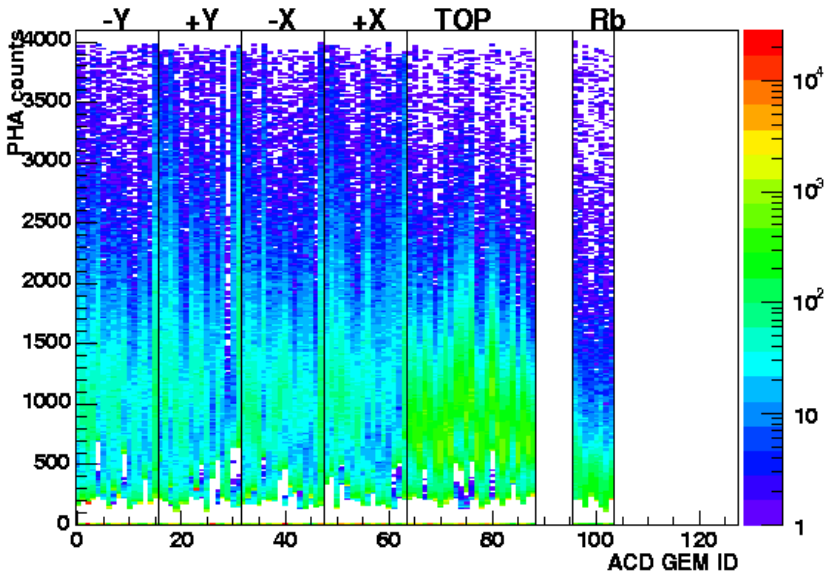
ACD Digi data in SVAC reports (hits and vetos)





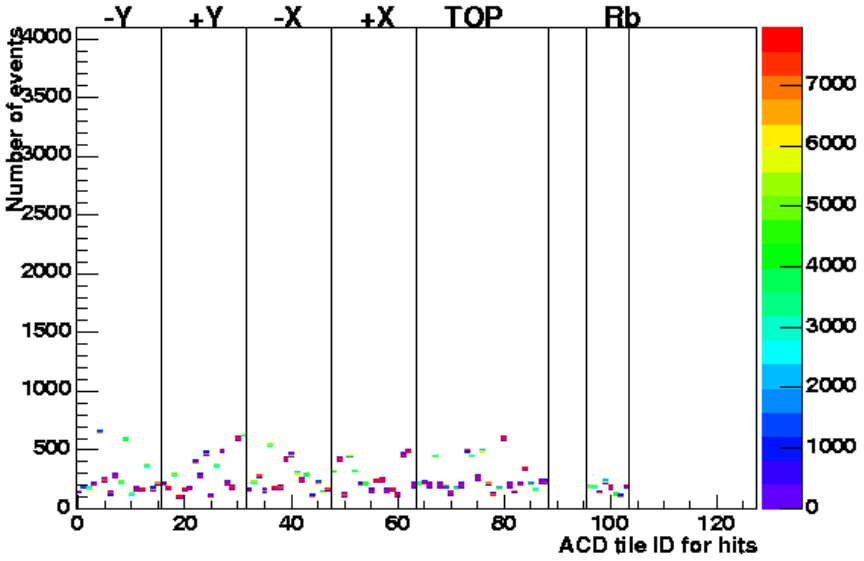


ACD Digi data in SVAC reports (PHA)

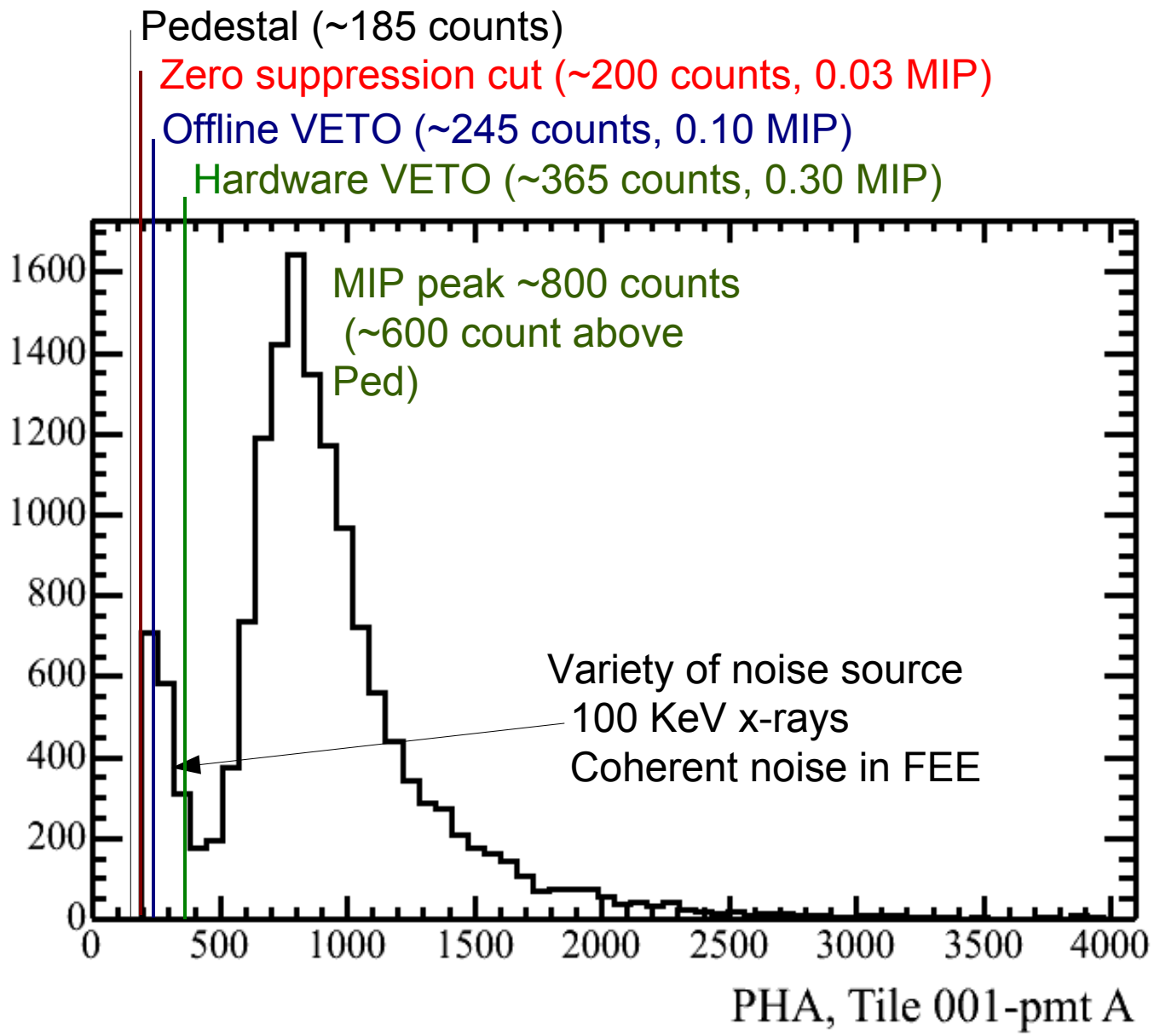


HV on (GOOD)

HV off (BAD)



ACD pulse height basics



Calibrated ACD Hits

- Express pulses in terms of MIPs/ MeV, not PHA counts
  - In Recon ROOT files as AcdRecon::AcdHit
    - ie, tree.Draw("m_acd->getAcdHit(0)->getMips(0)");

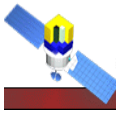
```
Event::AcdHit {

  AcdId m_tileId;          // which tile was hit

  ushort m_pha[2];          // Digi level data for both PMT
  ushort m_flags[2]         // Veto, Accept bits, error flags for both PMT
  float m_mips[2];          // calibrated values for both PMT in MIPs

  // coming soon
  float m_meV[2];           // calibrated values for both PMT in MeV
}
```

The values are also available in the SVAC ntuple as AcdMips

ACD hit flags

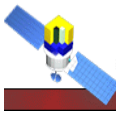
- These flags are defined so far:
  - PMT_ACCEPT // pmt is above zero suppression threshold
  - PMT_VETO // pmt fired veto discriminator
  - PMT_RANGE // pmt was read out in high range
  - PMT_ODD_PARITY_ERROR // pmt has parity error
  - PMT_HEADER_PARITY_ERROR // parity error in header
  - PMT_DEAD // pmt was dead or masked off
  - PMT_HOT // pmt was hot
- Maybe others such as:
  - PMT_IN_ROI // pmt was used in making an ROI coincidence
- Some of these require non-acd data
  - PMT_DEAD, PMT_HOT could require offline tables
    - Leave it alone for now, hopefully we would every need this, but the space is there if we do

ACD Distance variables in AcdRecon/ Merit



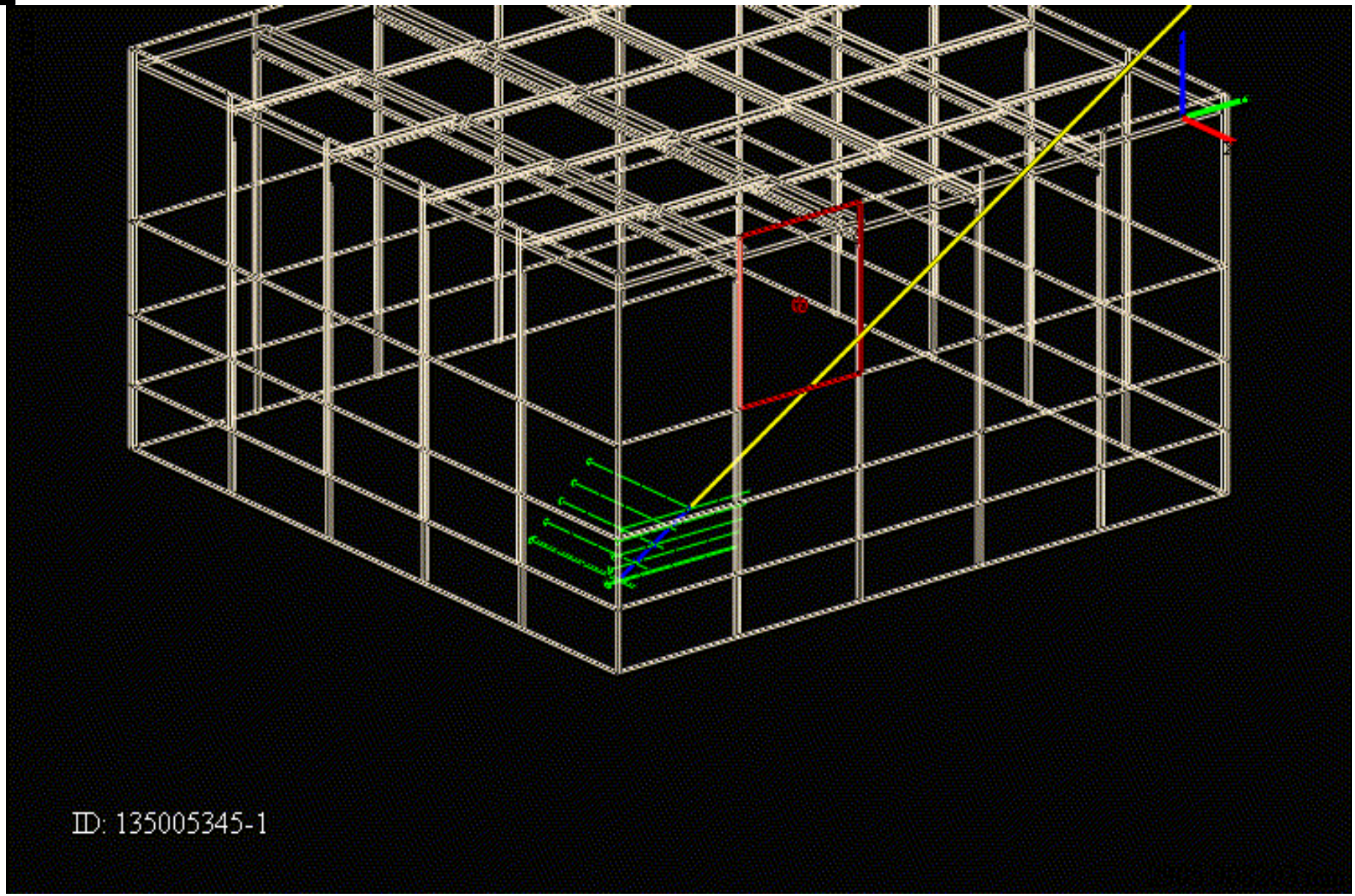
- CAVEAT
  - Lots of deprecated/ Monte Carlo stuff around
    - AcdDoca, AcdActiveDistance (old), AcdEnergy (MC)
- ActiveDistance3D
  - Positive: 2D distance to edge of element if track passes inside
  - Negative: 3D distance from track to edge or corner
    - NOTE: calculation changes as we cross tile edge
- HitRibbonDistance
  - Same as active distance always 2D, w/ simplified ribbon geometry
    - Ribbons made up of 3-segments, top & 2 sides
- CornerDoca
  - 3D distance to the gaps along the corner edges of the ACD
  - Sign takes into account direction of overlaps, as tile extends beyond edge in one direction

Geometry as used by ACD reconstruction

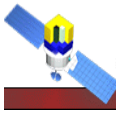


- Tiles
  - Defined as rectangular solids
  - A center & four corners. Perfectly flat, no thickness.
- Ribbons
  - Defined as three line segments
  - Top and two sides.
  - Use nominal with to decide if track hits ribbon
- Gaps
  - Defined as lines running down the sides of the ACD
    - Also, ribbons soon to be included as "gaps" b/c of the slight loss of efficiency in ribbon area
- Current version of code encapsulates the first two of these into simple data structure that can be cached and passed around. AcdTileDim and AcdRibbonDim

•And the same in picture(s)



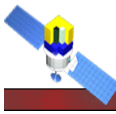
•Extrapolating tracks to the ACD



- Calculate where track leaves ACD volume
- Use G4Propagator to run Kalman fit out to that point
  - Keep track of all active ACD detector elements crossed by track
  - Extract information about intersections

```
Event::AcdTkrIntersection {

  AcdId m_tileId;          // which tile was hit
  int   m_trkId;           // which track did the hitting

  Point  m_global;       // 3D global postion of intersection

  double m_arcLengthToISect;  // distance from last hit to intersection
  double m_pathLengthInTile;  // distance inside active volume

  double m_localPosition[2];      // postion of intersection in local coords
  double m_localCovarience[3];  // projected error on the intersection in plane\

  unsigned int m_hitCode;         // bit map to see if tile was really hit
}
```

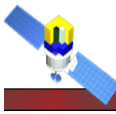
Associations Between Tracks and ACD hits



- We also want to keep track of which track come close to which hit ACD element
- For each track within an arbitrary distance (currently 200mm) of a hit ACD element we store

```
Event::AcdTkrPoca {

  AcdId m_tileId;          // which tile was hit
  int     m_trkId;         // which track did the hitting

  Point   m_location;      // 3D global postion of POCA

  TkrTrackParams m_params;   // track params at the POCA

  int m_region;            // where does the POCA occur
  double m_dist;           // the active distance
  double m_distErr;        // the error on the active distance
}
```

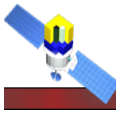
Keeping track of many associations



- Sorting the track-tile coincidences by active distance
  - Largest (ie, most "active") comes first
- Provide functions to access them in that order

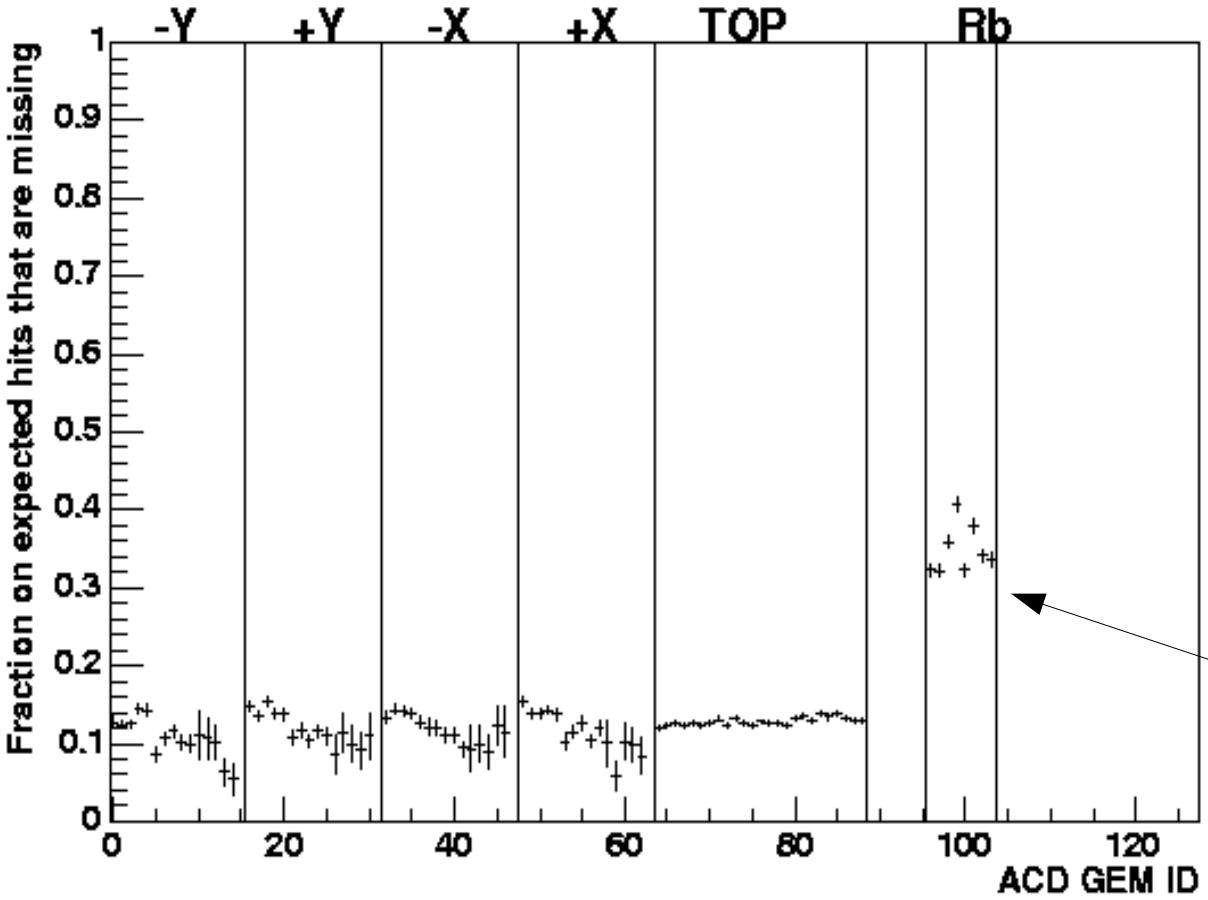
```
Event::AcdPocaMap {

   // these get only the best coincidence
   // they return null pointer if there is none
   Event::AcdTkrPoca* bestPoca(Event::TkrTrack&);
   Event::AcdTkrPoca* bestPoca(AcdId&);

   // these return all the relevent coicidences
   // of course the set could be empty
   set<Event::AcdTkrPoca*> pocas(Event::TkrTrack&);
   set<Event::AcdTkrPoca*> pocas(AcdId&);
}
```

Summary of new Data in AcdRecon/ SVAC



- AcdHit
  - Hit Based calibrated data -> data structure is ready
    - Still need to do calibration code
- AcdTkrIntersection
  - Track based, extrapolation to ACD, independent of if ACD hit
    - In release, variable pulled into SVAC tuple
- AcdTkrPoca & AcdPocaMap
  - Track-Hit correlations
    - Might want to revist exactly what is being stored in AcdTkrPoca to match better with AcdTkrIntersection
- For more information stick around for lunch tutorial

•ACD recon plots in SVAC reports

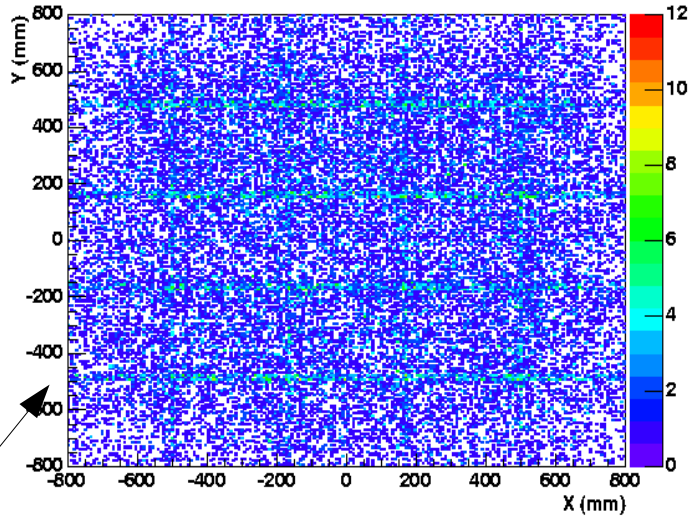


This plots really shows the photon contamination & error in track extrapolation in the surface muon sample, rather than the ACD efficiency

Ribbons show higher ineffiency b/c of track extrapolation errors

$$\frac{\text{GemID}\quad \text{acdTkrIntersection->hitMap()} == 0 \;\&\&\; \text{nTracks} == 1}{\text{GemID}\quad \text{all acdTkrIntersection} \;\&\&\; \text{nTracks} == 1}$$

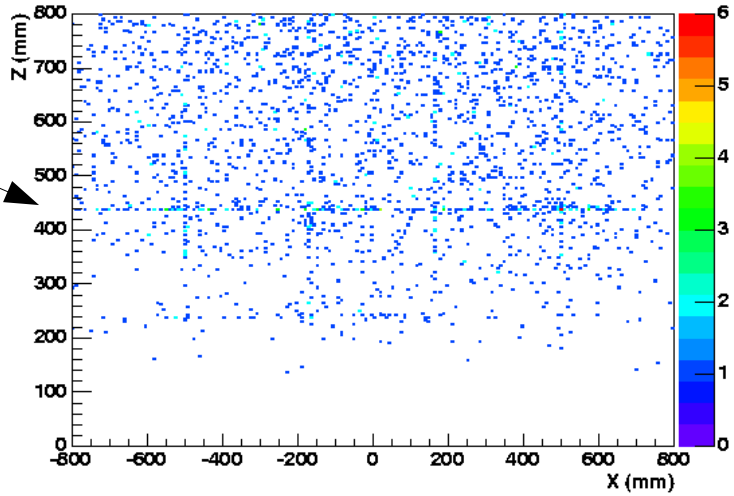
•ACD miss maps in SVAC reports



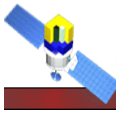
The flat background is from the photon “contamination” of our nice muon sample. Also from mis-reconstructed events.

The structures that we see here are from:

1) the overlaps (2 entries instead of 1)
2) when tracks get extrapolated into the wrong tile, (near the tile edges)



Using Acd recon to look for gaps in the ACD



- The ACD is required to be 0.9997 efficient for charged particle detection
  - PMT efficiency for 0.1 MIP in tile must be very, very close to 100%
  - PMT on ribbons must detect down to 1-2 p.e.
  - Even with this, we need to be able to ID events that snuck in the gaps
- About 30% of cosmic rays are NOT muons
  - Trying to measure 0.3 per mil effect requires very clean muon sample
  - This is also good practice for getting clean MIP sample for other uses

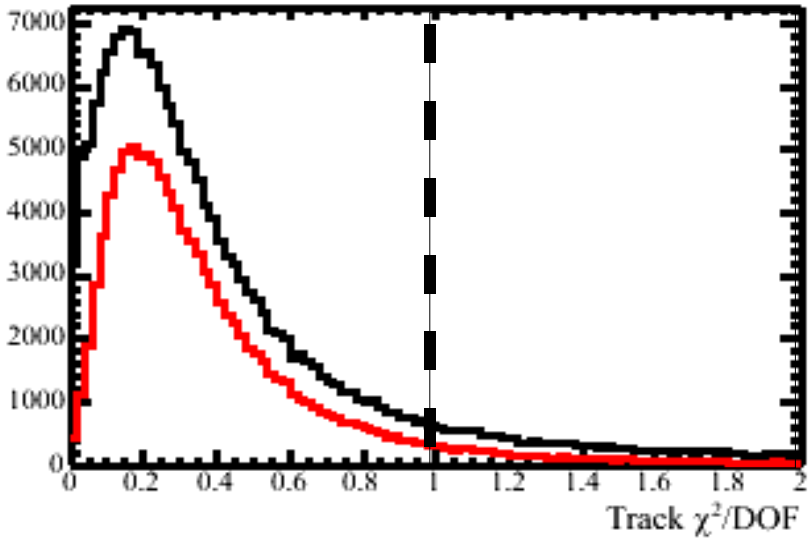
Data Used/ Processing version

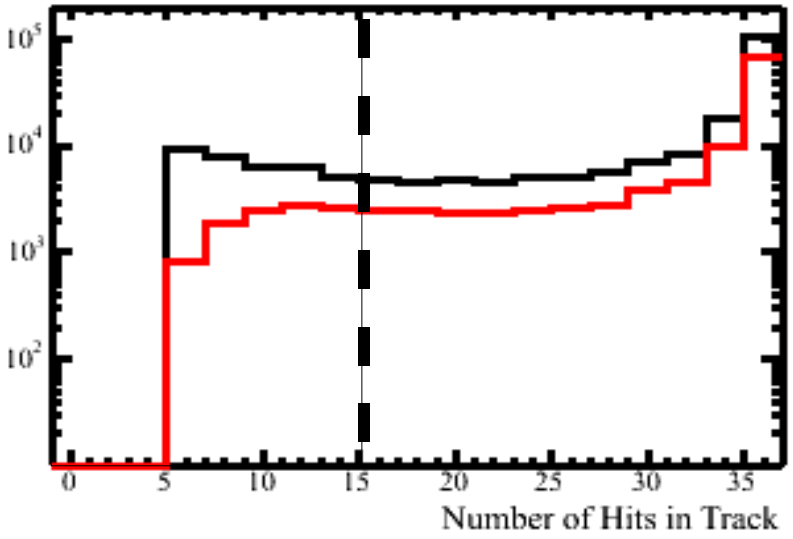
- For muon selection tuning used run 135005345
  - B/2 470K triggers
- For ACD performance plots used all long B/2, B/30 runs
  - About 37.8 M triggers
- For all plots used digitization-v3r4p6 (reprocessing)
- For all plots used recon-v3r4p6 (reprocessing)
  - The reprocessing has calculations of all the POCA between tracks and hit ACD tiles and ribbons stored in the Recon ROOT tree

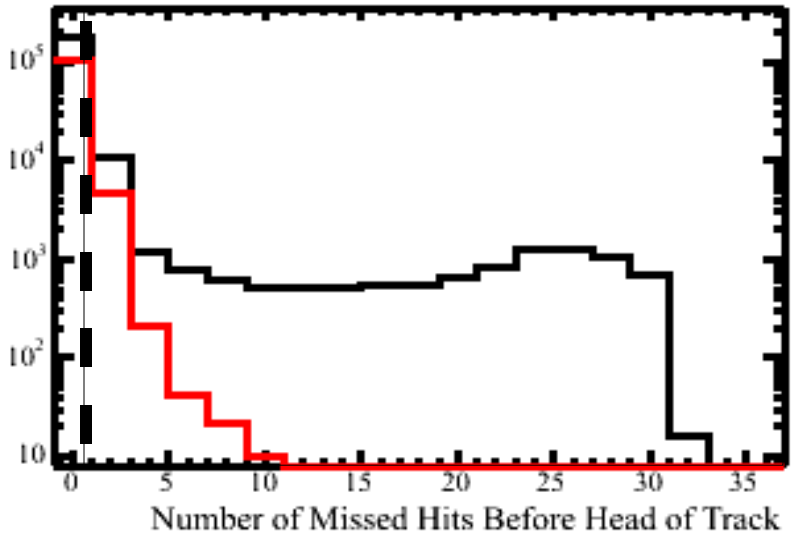
Muon Selection

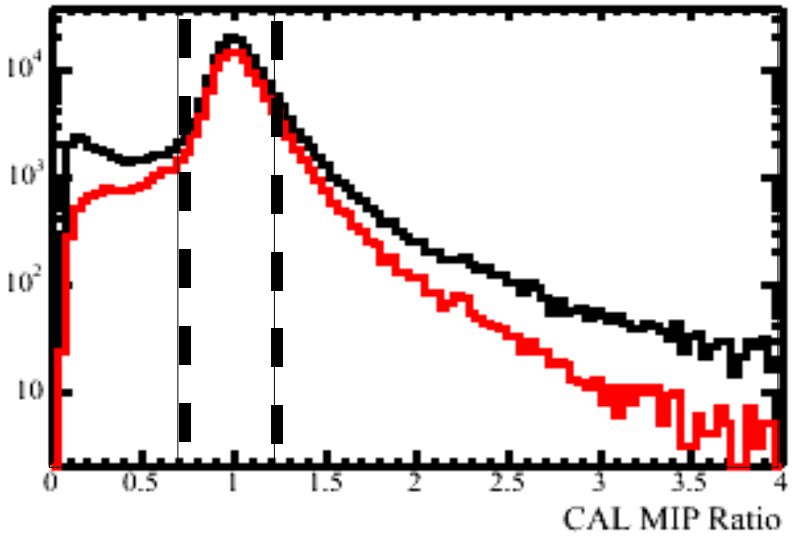
- Exactly one track (the muon)
- Track c2 < 1.
  - when Kalman fit with the default hypothesis 100 MeV e-
- 0.75 < CalMIPRatio < 1.25
  - Ratio of energy in CAL to expected energy for MIP (path-length corrected)
- Number of hit of Track > 15
  - This cuts out the lower part of the sides of the ACD
- "Tkr1SSDVeto" == 0
  - There is a hit in the last plane before the track exists the tracking volume
- These cuts selected 115115 of 470286 triggers in run 135005345
  - 24.48% of triggers

Muon selection, (Red is After all other cuts)

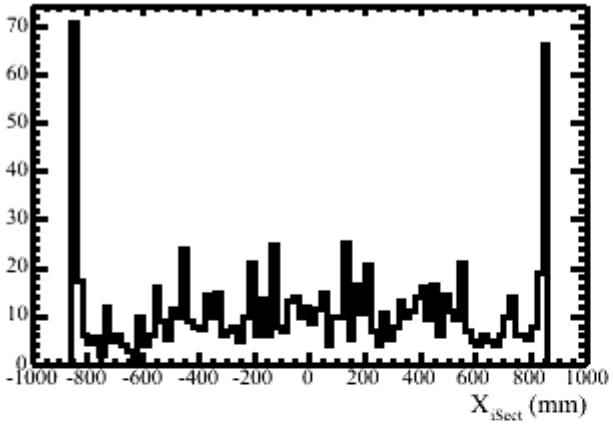


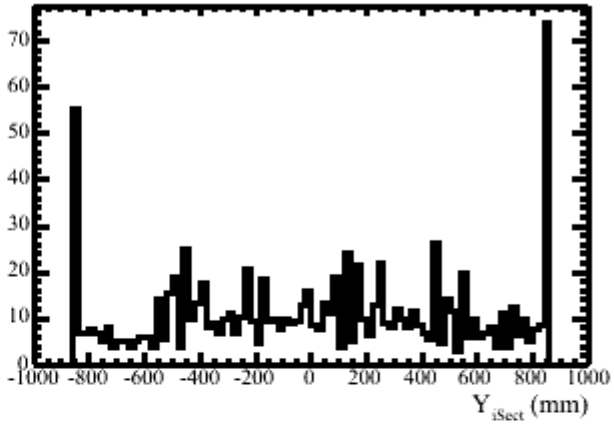


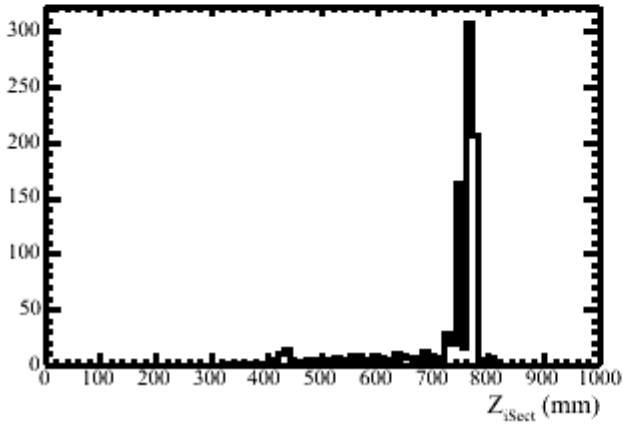




Track extrapolations unmatched to ACD hits







964 Misses out of 37.8M * 0.245 = 9.26 M
Inefficiency of 0.000104

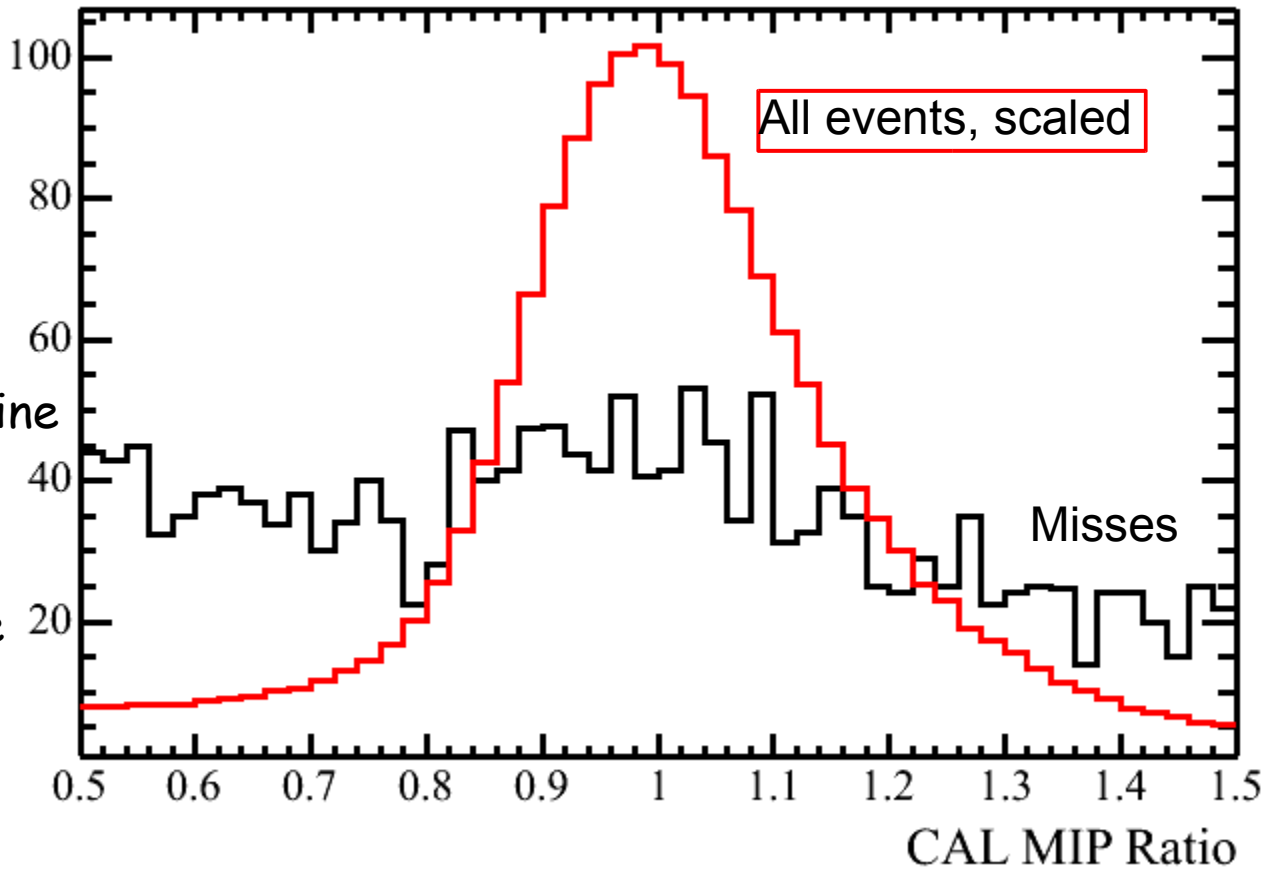
Some hints of ribbons, but not obvious above fairly flat background

Selected muons with no AcdTkrPoca within 150 mm

Are these misses from muons?

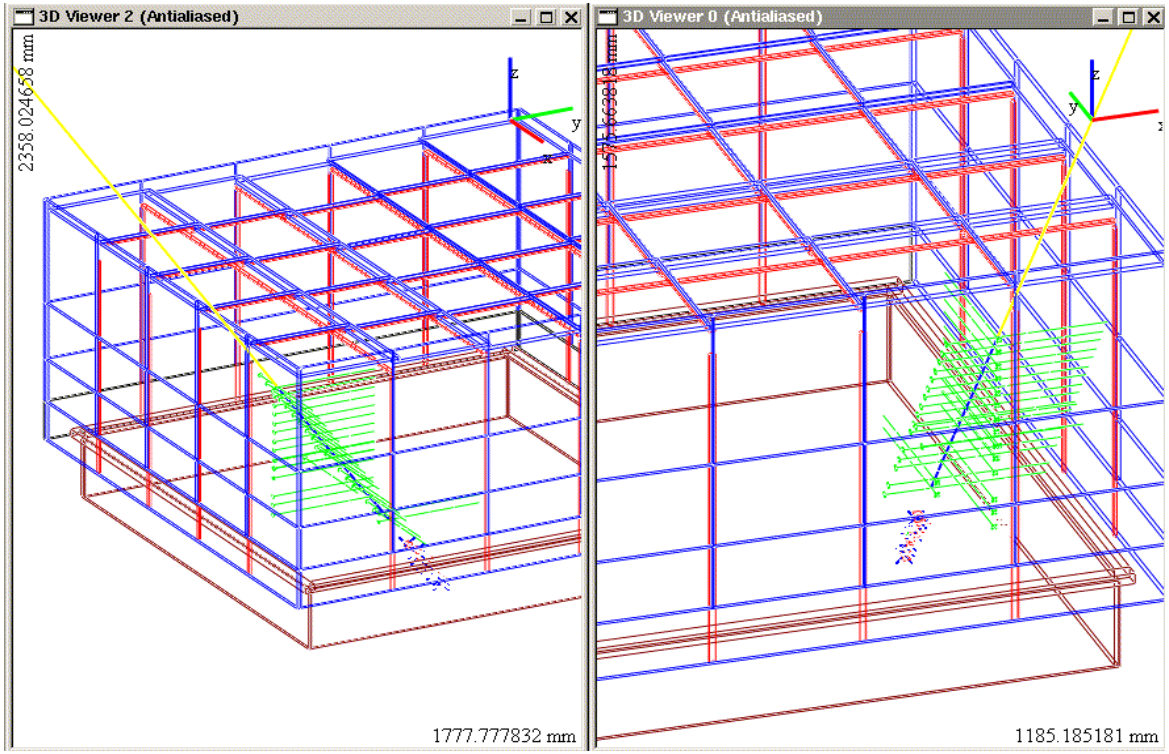
The CalMIPRatio for the events which don't have any nearby hits is quite different from the baseline distribution.

It seems likely that some fraction of these events are not muons

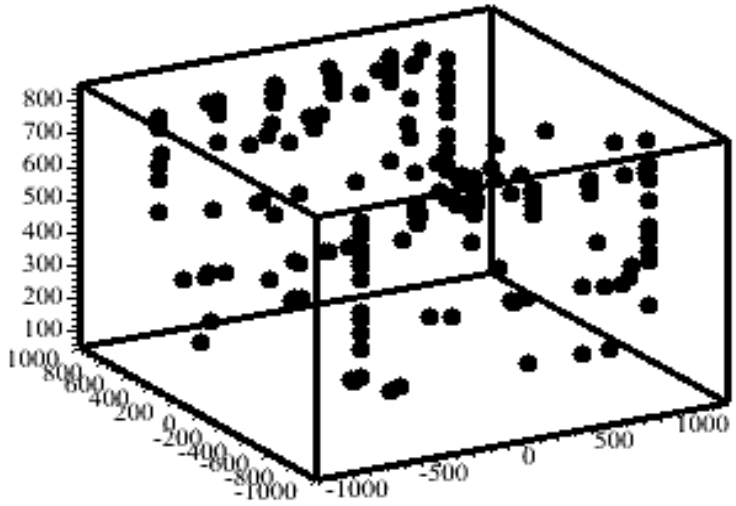


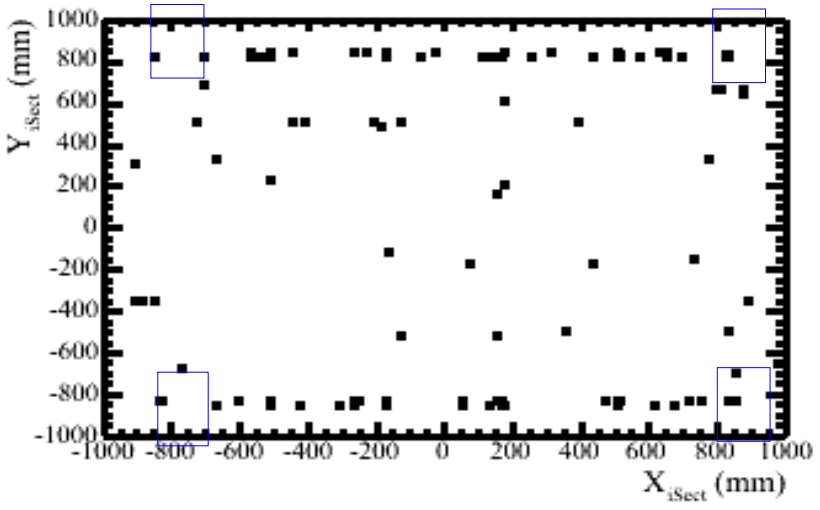
A track hits a gap

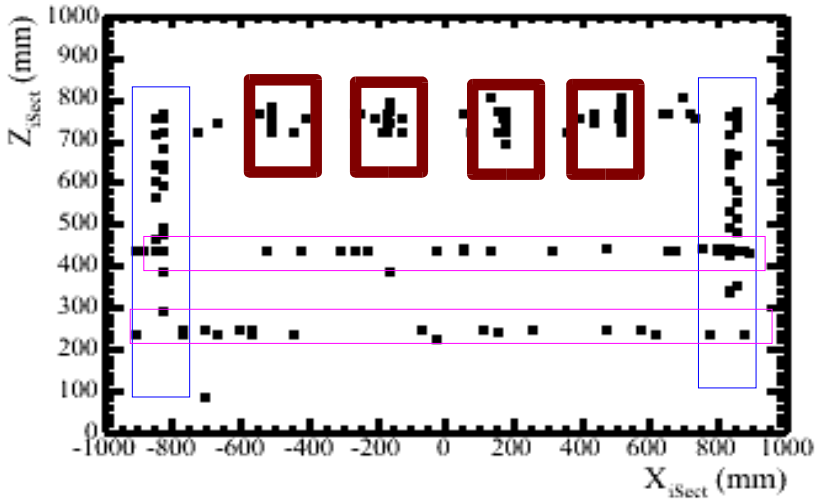
- This happened 29 times in the entire sample we had a track go through a gap in GEANT model & no ACD hit
  - nAcdTkrIntersection == 0 && AcdNumDigis == 0
  - Have scanned 4 of them in FRED, they all hit the gaps at the top of the Y sides

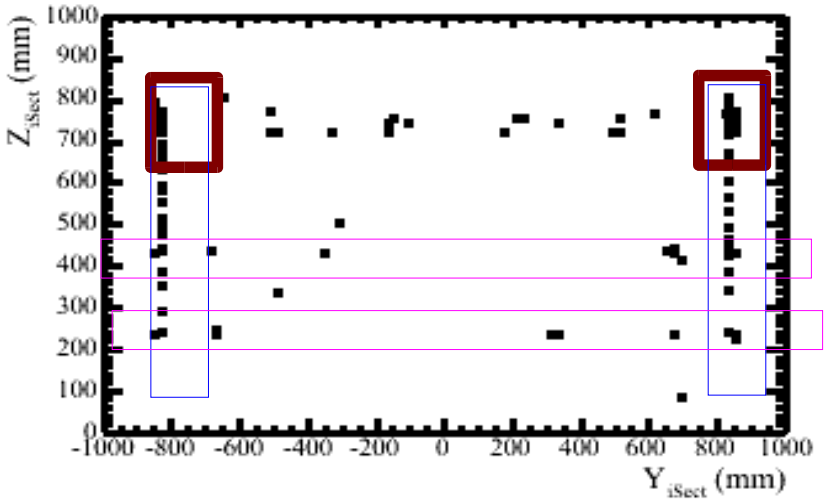


Gaps in the GEANT model



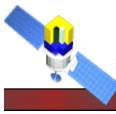






POCA for nAcdTkrIntersection == 0 but nAcdTkrPoca != 0

Status of ACD Recon



- Getting there. Most of the big stuff is under control
  - Merit variables for background rejection
  - Extrapolating tracks to ACD
  - Finding hits near tracks
- A some big things still missing
  - The geometry model is not totally accurate
    - This is something of a sore point
  - Calibrating to MIP/ MeV
    - This is probably the highest priority item for looking at upcoming ground data.
  - Handling the ribbons/ gaps
    - What is the ribbon efficiency?
    - How do we best use the ribbons?

Wish List

- AcdCalibSvc
  - To read in calibration xml files and go from PHA -> MIPs/ MeV
- Additions to AcdGeomSvc
  - Basically getting the list and position of the gaps we care about
- A plan for the ribbons?
- Some other (long term) project ideas have come up
  - Using the ACD to catch stuff going sideways in tracker
  - Using the ACD to cross check the tracker performance
    - Look at the distributions near tile edges to get tracker resolution.