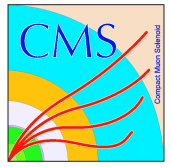


"Slow" Refitting of TeV Muon Showers

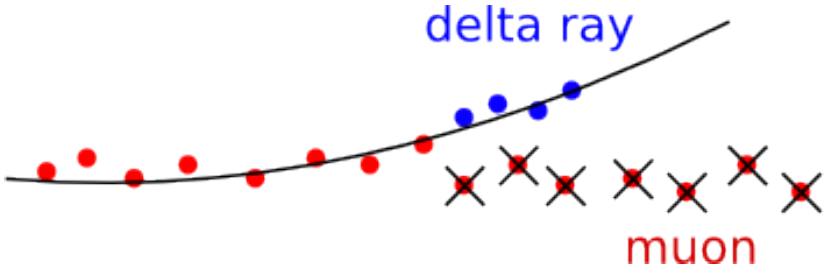
Jim Pivarski

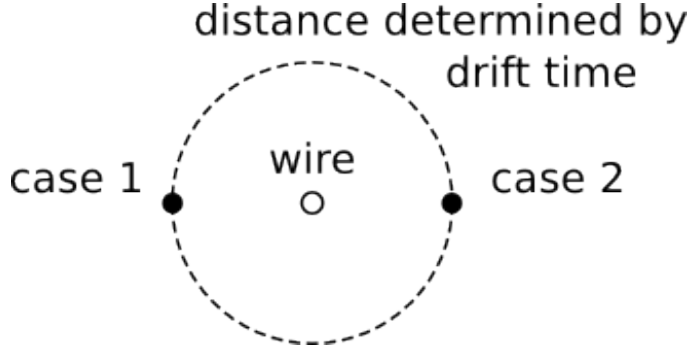
Texas A&M University

16 October, 2008

The Problem

- TeV muons shower, making it difficult to identify which hits belong to the muon and which belong to the delta rays
- We try dropping bad hits from tracks to improve the resolution: FirstStation, Picky, and Cocktail algorithms
- But we have no mechanism to add hits— no ability to recover from an earlier mistake
- More rigorous treatment would return to the whole RecHit collection for TeV candidates and spend more time to find the best set
- For example, segment-finding might pick the wrong side of a DT hit in a very crowded environment; can be better resolved when we know more about the full track





Slow refitting algorithm

- Consider a new track-refitting algorithm which also revisits hit selection from scratch. Algorithm must be:

  1. accurate
  2. ~~fast~~ No!

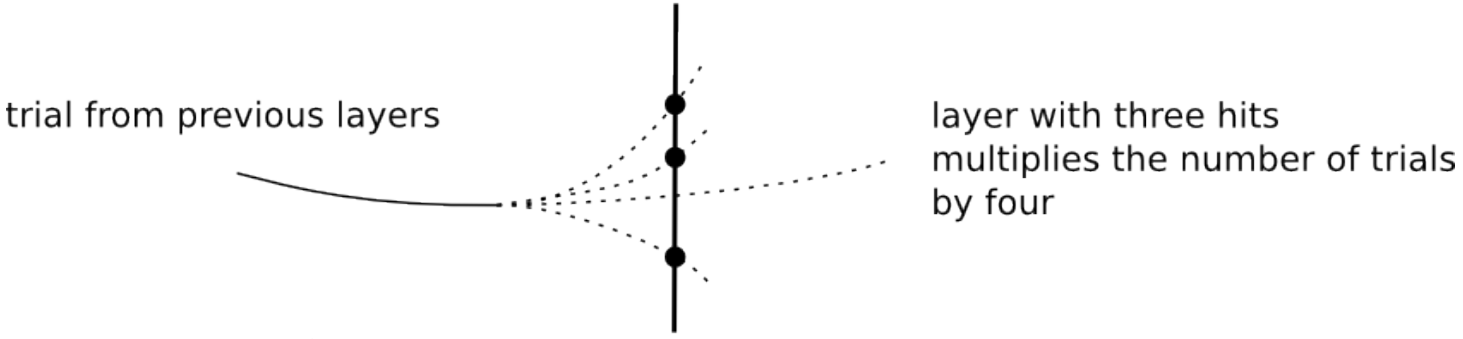
- Only need to refit rare muons in massive dimuon-pair events
- We can afford to spend more CPU time at the highest energies in the offline analysis
- With infinite time, we could try every combination of hits

$$\left(\#\text{hits per layer} \right)^{\#\text{layers}} \approx \text{billions of combinations}$$

- We can almost certainly find the optimum in a more directed search with about 1 minute/muon or less

Algorithm details

1. Starting with tracker track (assume no showering in tracker), make a tree of possible extensions into the muon system
   - At each layer, create a new branch for every possible decision: one for each hit, and one for no hit (>1 hit/layer not allowed)

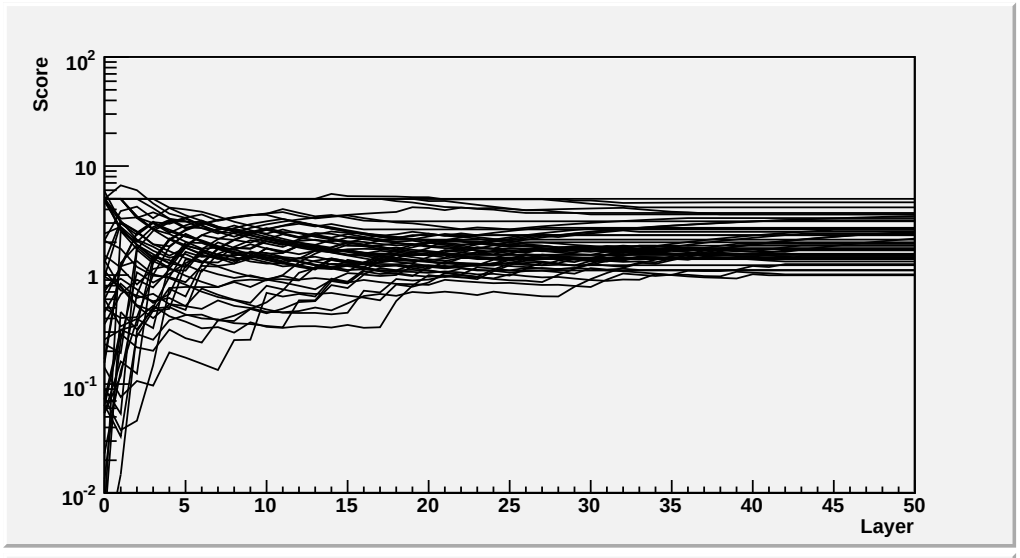


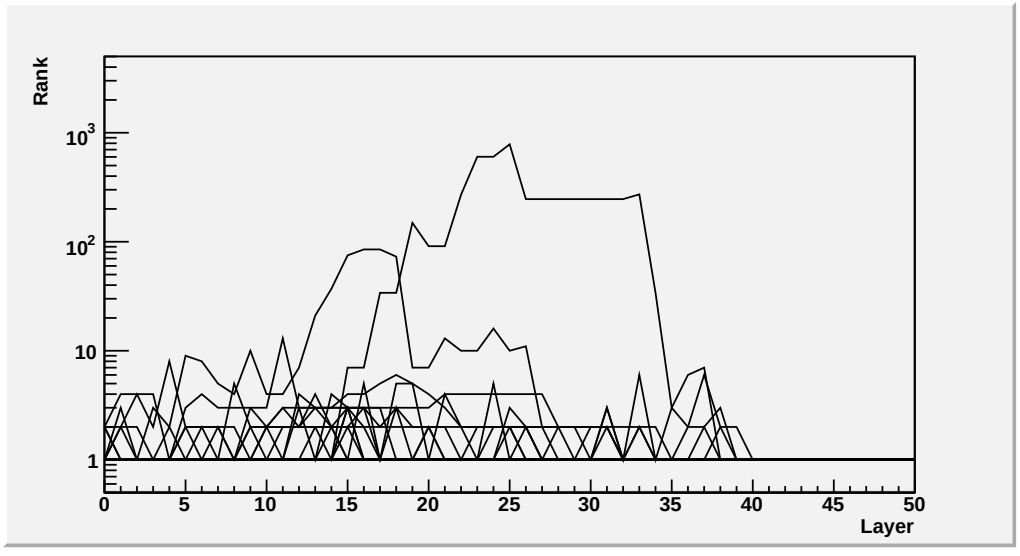
   - $\text{score} = \dfrac{\sum \chi^2_i + N_{\text{missing hits}} \times \text{missing hit penalty}}{N_{\text{layers}}}$
2. Sort trials by score, prune branches with too-high scores at each layer
3. Completely refit the best 100 or so (to get forward-backward smoothed χ^2s), publish the winning trial as the refitted track

Analogy with game theory: from tree of all possible moves, consider only those that lead to winning end-states

Speed optimization

Unlike a game, our opponent isn't trying to trick us
Trials that start to go bad don't improve (because they're following delta rays)





- Among 50 trials that eventually won, score (reduced χ^2) never got above 10

  $\longrightarrow$ apply a cut at 100

- They were always in the top 1000

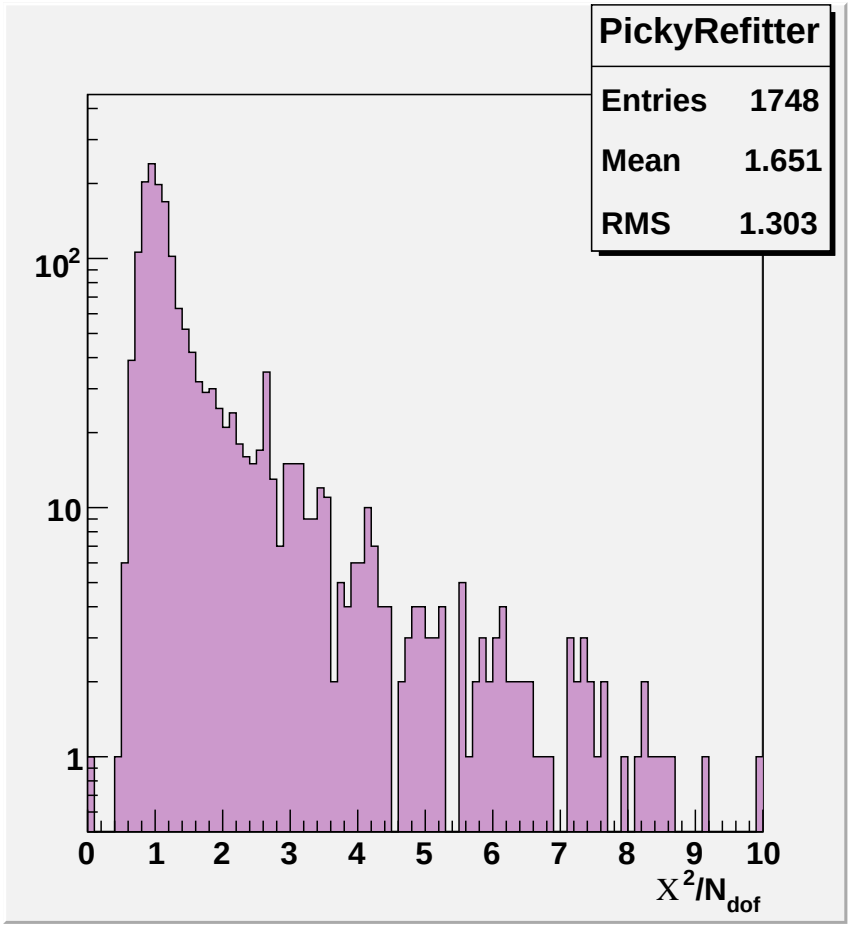
  $\longrightarrow$ cut at 10,000

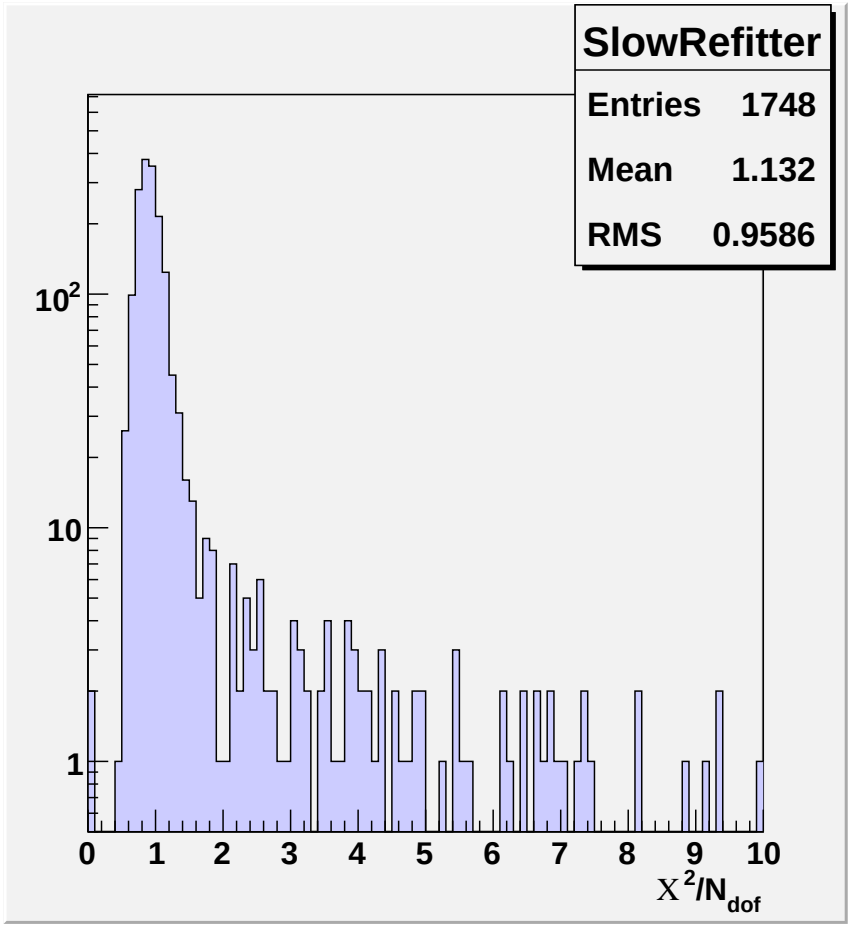
- Cut on rank stabilizes running time

with these cuts: 48 sec/muon

Performance tests: χ^2/N_{dof}

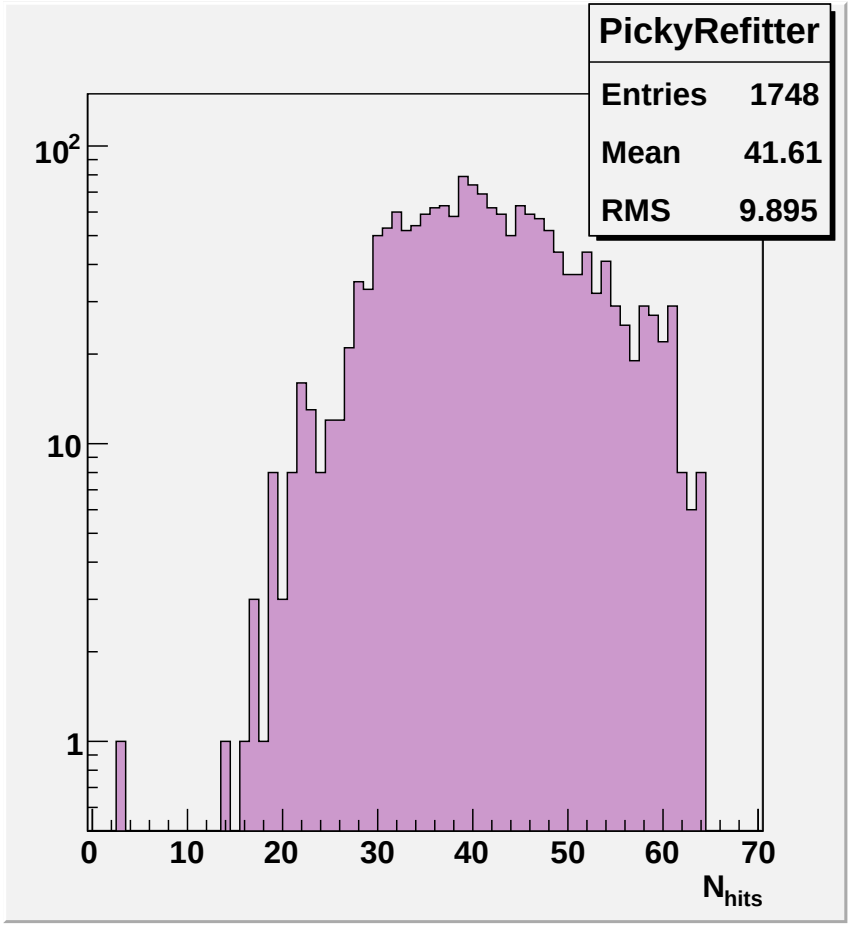
- 1000-event sample of 2 TeV $Z' \to \mu\mu$ in 2_1_9 (private sample)
- Use Picky as a baseline for comparison
- This is the real track χ^2, not score; peak is closer to 1.0, smaller tail

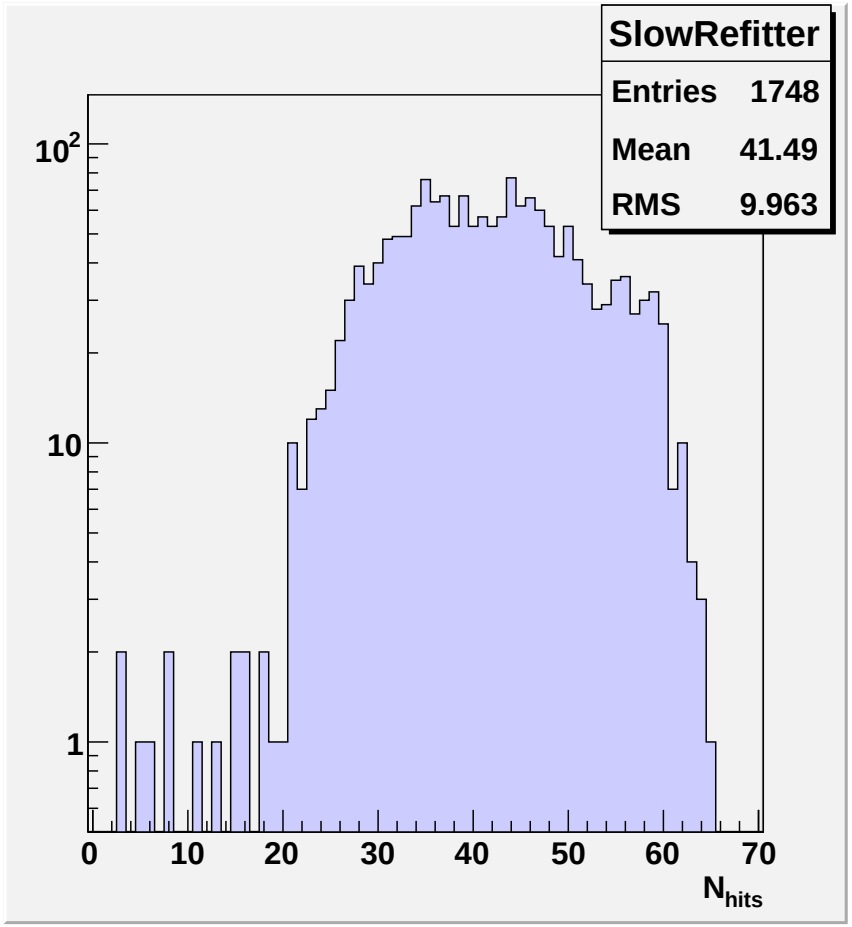




Performance tests: #hits

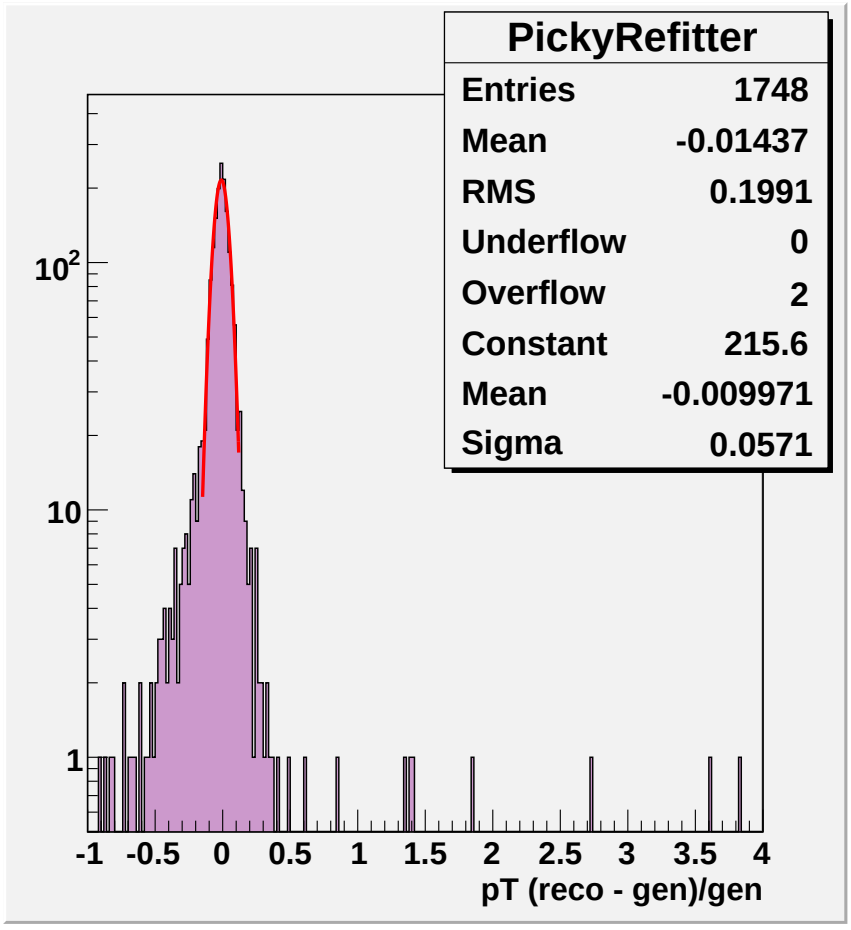
- It is not reducing χ^2 by throwing away more hits
- Missing hit penalty = 5, unoptimized
  Could be made rigorous by converting hit efficiencies into $(\#\sigma)^2$

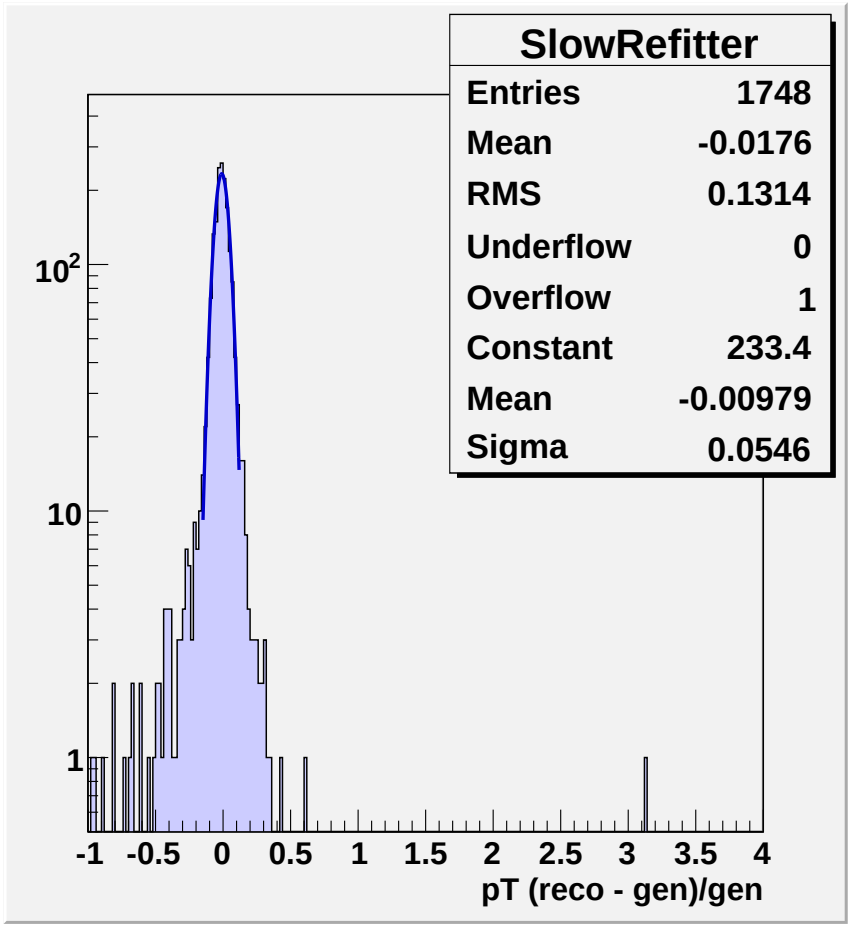




Performance tests: p_T

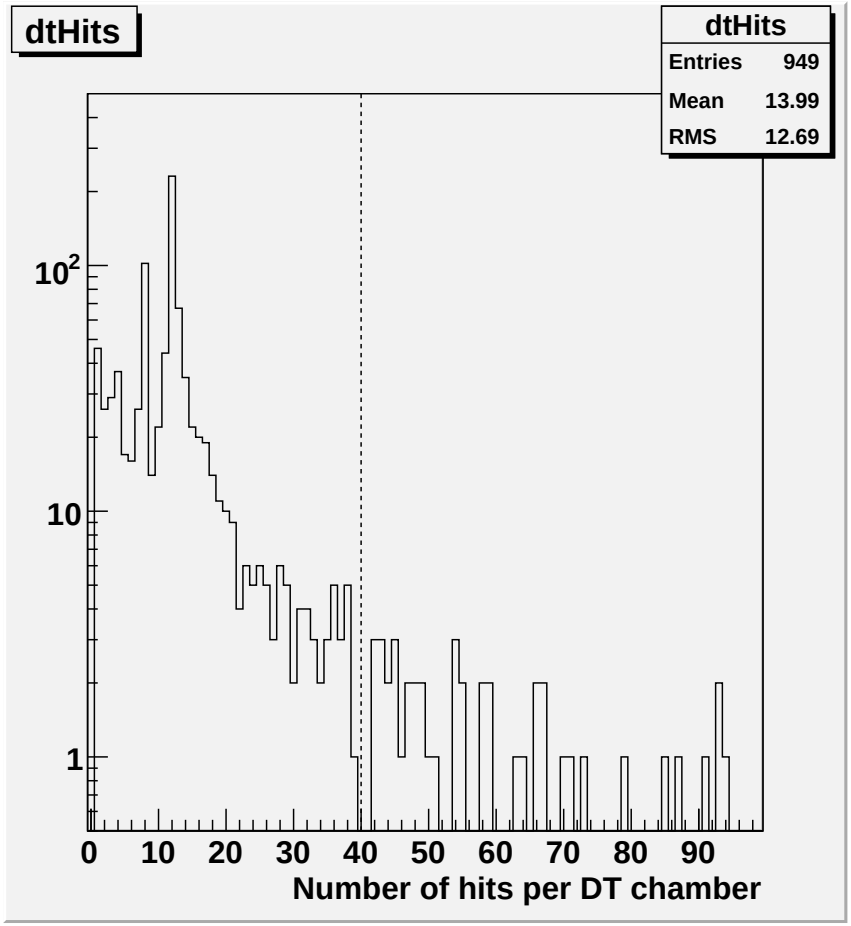
- No impact on core of p_T accuracy: 5.5% in both cases
- But it cleans up tails (could be relevant for Drell-Yan backgrounds)
- Picky RMS: 20% Slow RMS: 13%

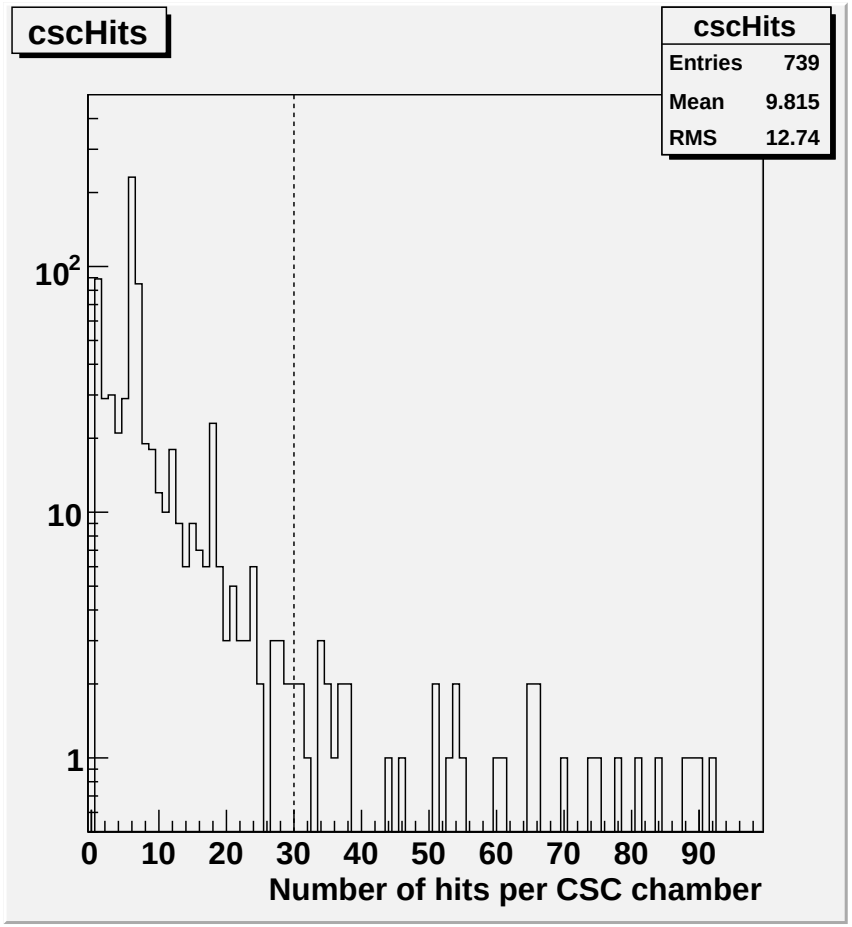




Another (possibly) useful cut

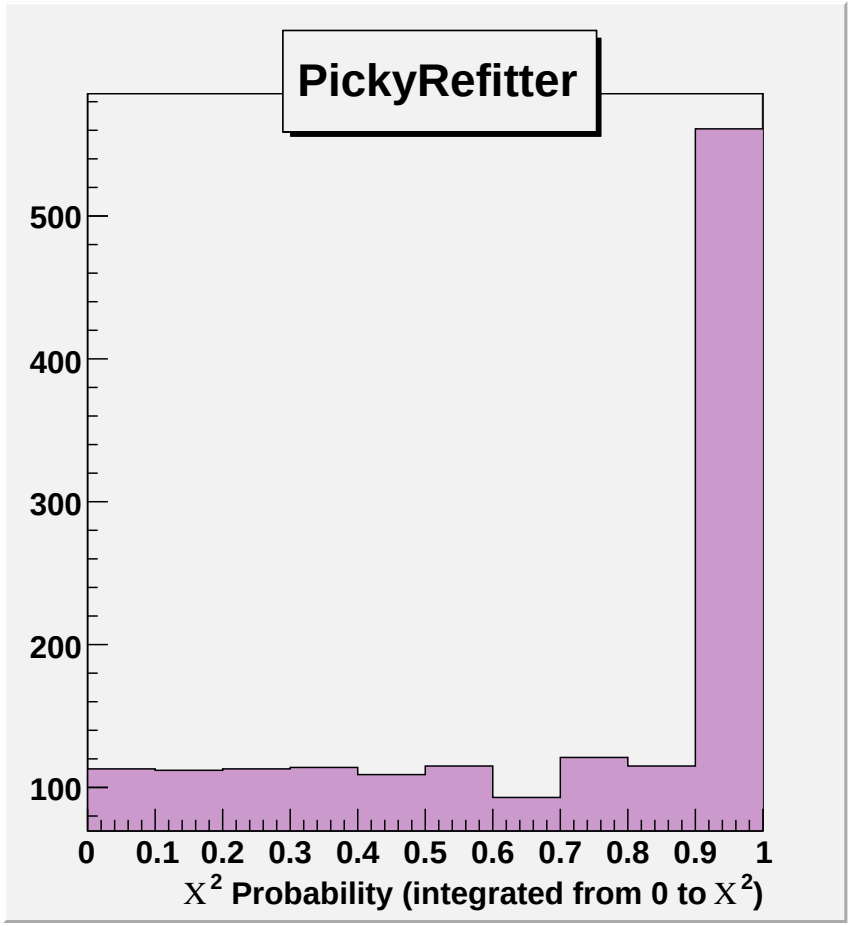
- Pre-count number of hits on each chamber to exclude chambers with lots of showering
  - more likely to pick up delta-ray hit if they are too dense
- Presented cut values were used in this study (but not chosen carefully!)

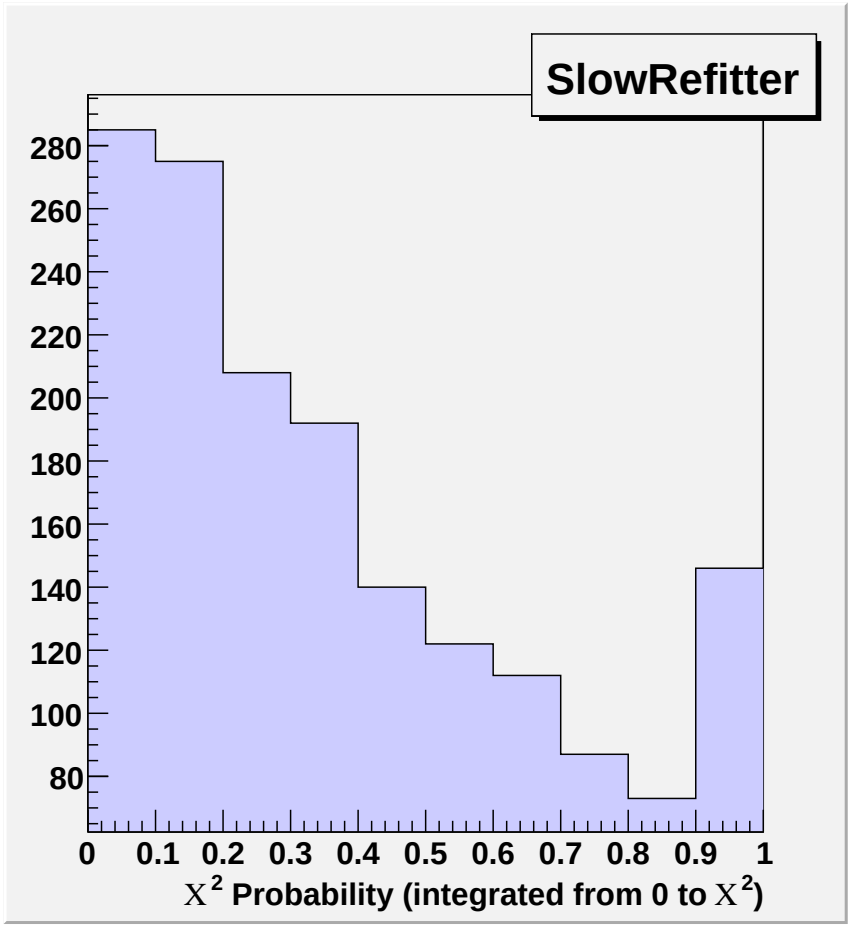




Evident need for more tuning

- Computed probability of track χ^2 from page 6 (should be flat)
- Picky clearly has a tail of too-high χ^2 which Slow refitter mostly fixes
- But Slow refitter also biases the central distribution toward too-low χ^2
- Problem with (arbitrarily-chosen) value of missing hit penalty?





Conclusions

- We can more carefully fit our best muons if we're willing to wait
- Cleans up tails of distributions, but parameters need to be tuned

Open questions

- Is the χ^2 distribution issue inherent to the method, or a function of the parameters?
- Are these tracks stable under multiple refits (fixing the set of hits)?

  If so, perhaps it can provide insight into why GlobalMuons and TeVRefittedMuons are not.