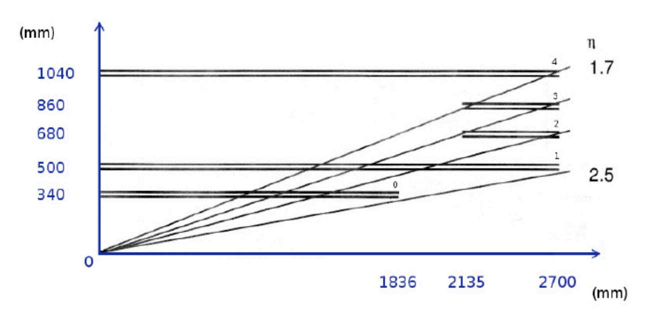
UF | UNIVERSITY of FLORIDA

CSC+Tracker Trigger Algorithm

D. Acosta, A. Ballado, M. Fisher, I. K. Furić, L. Kaplan, A. Madorskiy, B. Scurlock, B. Williams

CSCTT Algorithm

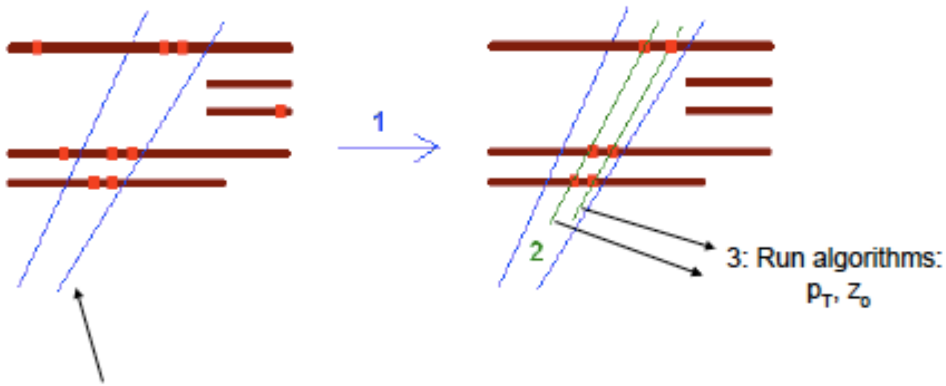
- SLHC conditions expected to yield very high single muon rate at Level 1
  - Combining Tracker data with CSC Track-Finder (CSCTF) data can help reduce fake rate due to noise, detector effects, and CSCTF mis-measurement
- CSC+Tracker Trigger (CSCTT) Algorithm:
  - Define regions of interest to help pre-sparsify tracker readout
  - Assume clustered stub information is read out from tracker
  - Define narrow roads in ϕ, z to further filter tracker readout
  - Tracker stubs have excellent positional resolution utilize internal correlation
  - Attempt fit using tracker-only information (best measurement at low momenta)
- Current CSCTT model developed in context of the Long barrel geometry developed by Tracker upgrade simulation group
  - 100 micron x 1 mm pixels
  - 10 Layers ("stacks"), sensors ~ O(mm) → Stubs
  - Grouped into 5 "double stacks", stacks ~ O(cm)
  - Our studies use FastSim, simHits Stubs
  - See Laura Fields talk in Tracking/Trigger session for more detail



- CSCTT code is being committed to CVS this week
- Internal Note under development. First draft will be available soon.

CSCTT Algorithm

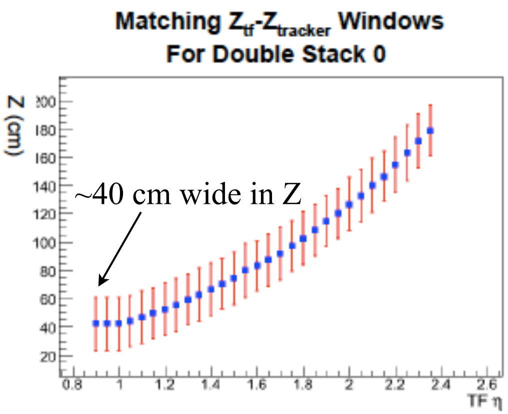
Illustration



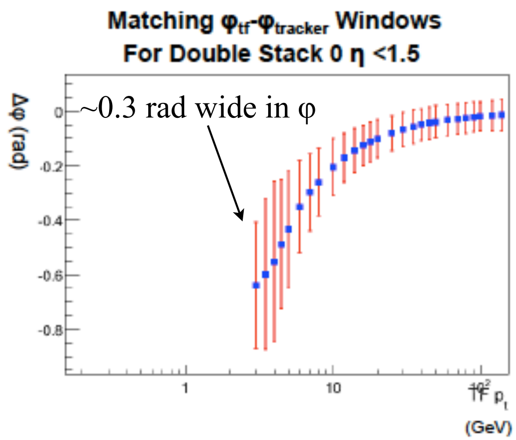
- Step 1: Use matching windows to cut stubs based on $\text{Trackfinder}_{z,\varphi}$ - $\text{Tracker}_{z,\varphi}$
- Step 2: Only keep stubs that are correlated in $\Delta\varphi$ & $\Delta\cot\theta$ (ie $\varphi_{dstack2}-\varphi_{dstack0}$)
- Step 3: Apply r–z algorithm → $\cot(\theta)$ & z_o and r–φ algorithm → p_T

CSC+Trigger Matching Windows

Examples of **For Double Stack 0 :**



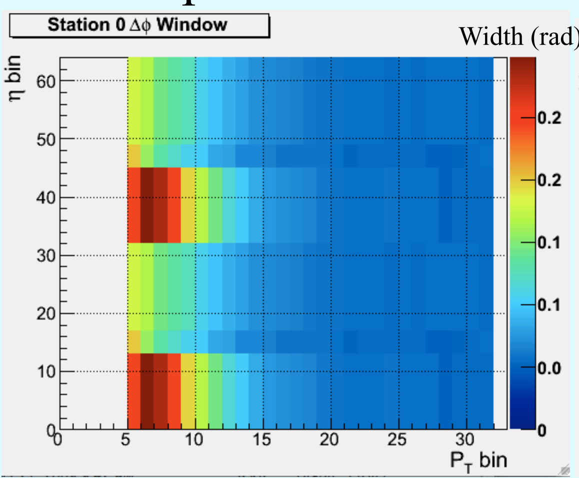
Widths ~ 6 cm

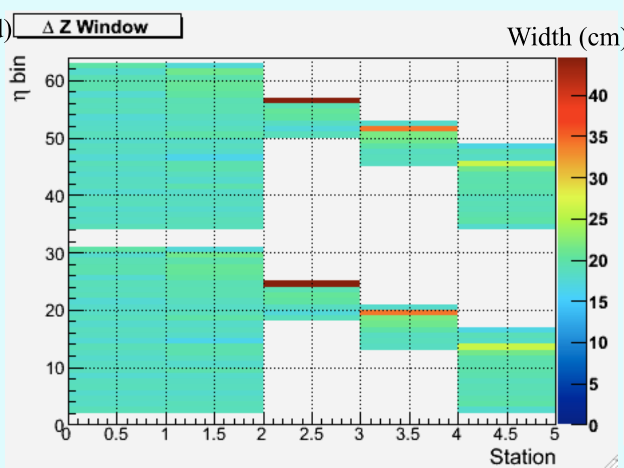


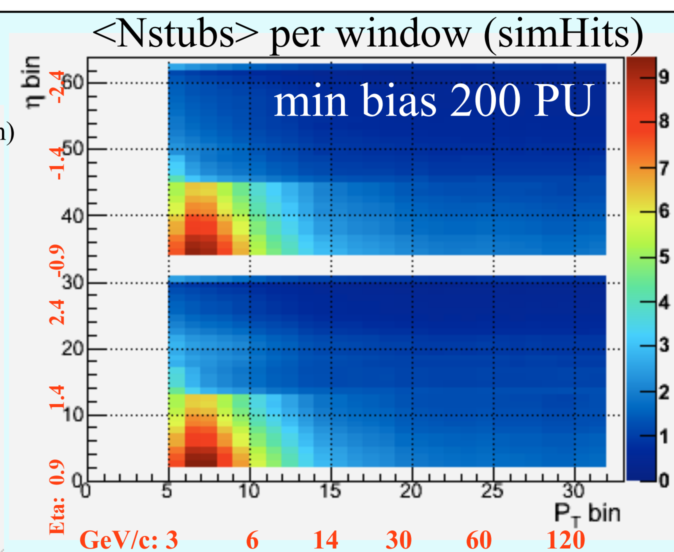
Widths =O(~0.1) - O(~0.01) rad
η dependence low p_T due to inhom. B-field
Can be tightened if necessary

Matching windows are defined for all possible CSCTF-P_T (5 bits) and CSCTF-η (5 bits per endcap) values. Average match-window-occupancy plots shown below are a function of these CSCTF bins and were made with min bias events (200 PU).

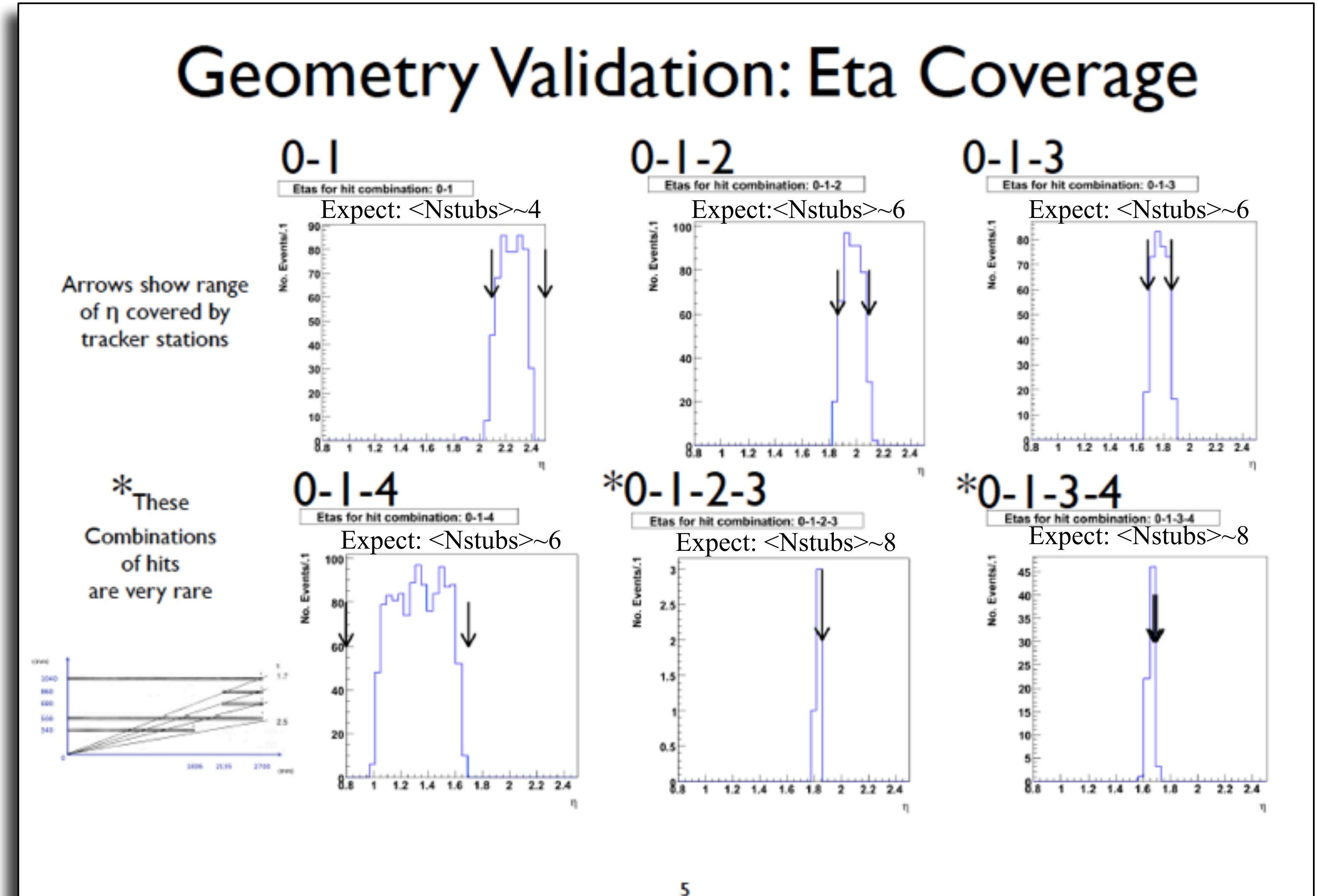
Example Match window sizes versus CSCTF bins



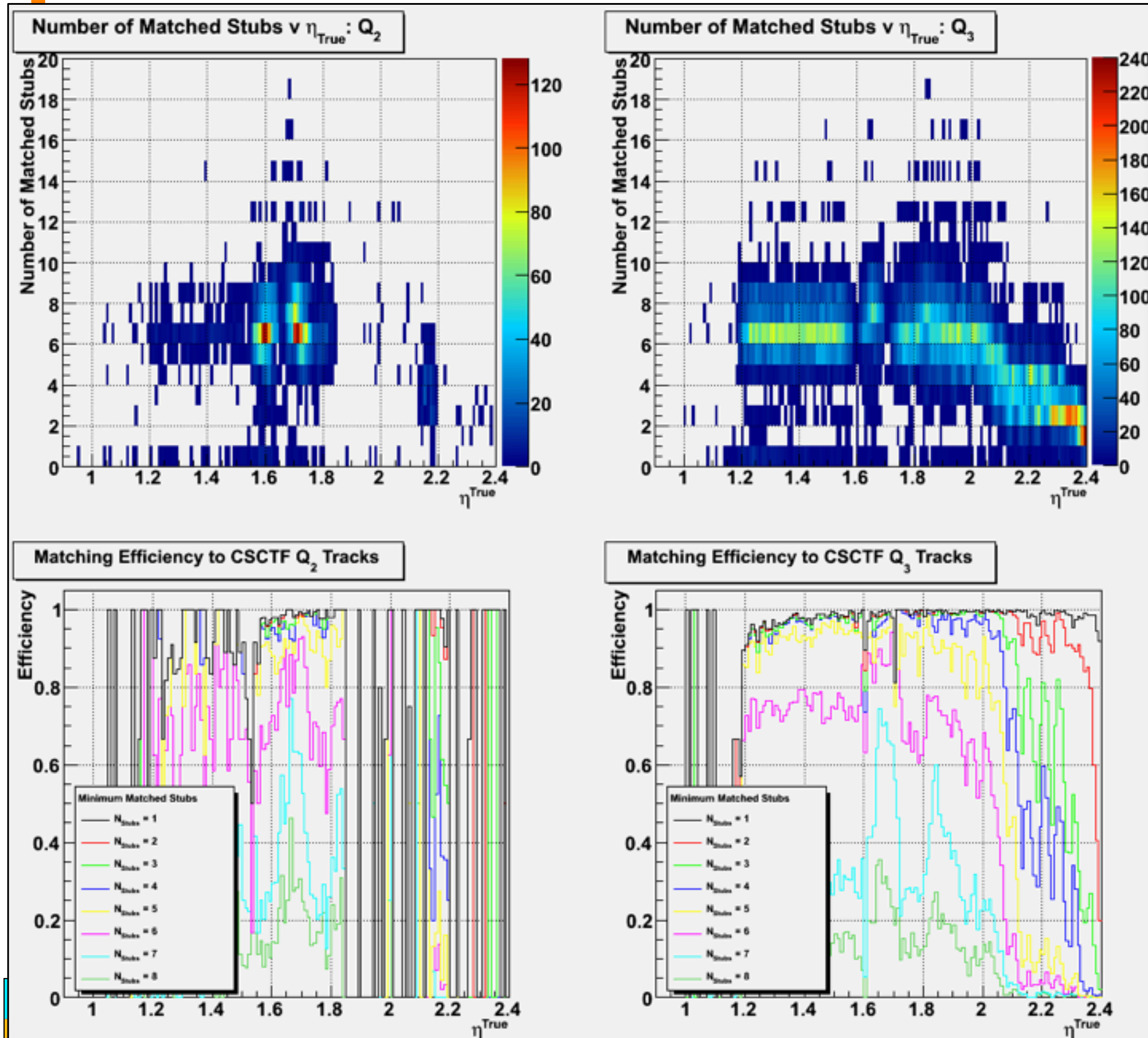


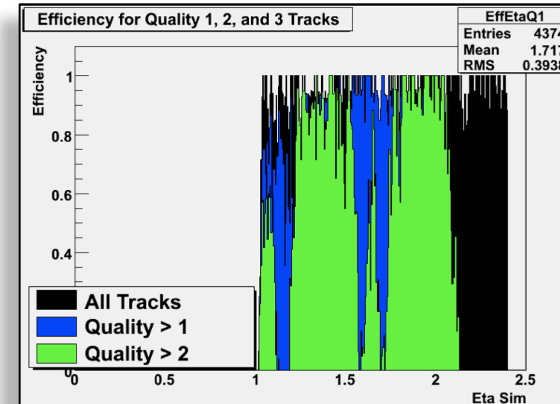


Expected Eta Coverage (Long barrel)



Matching Windows Efficiency (room for fine-tuning)



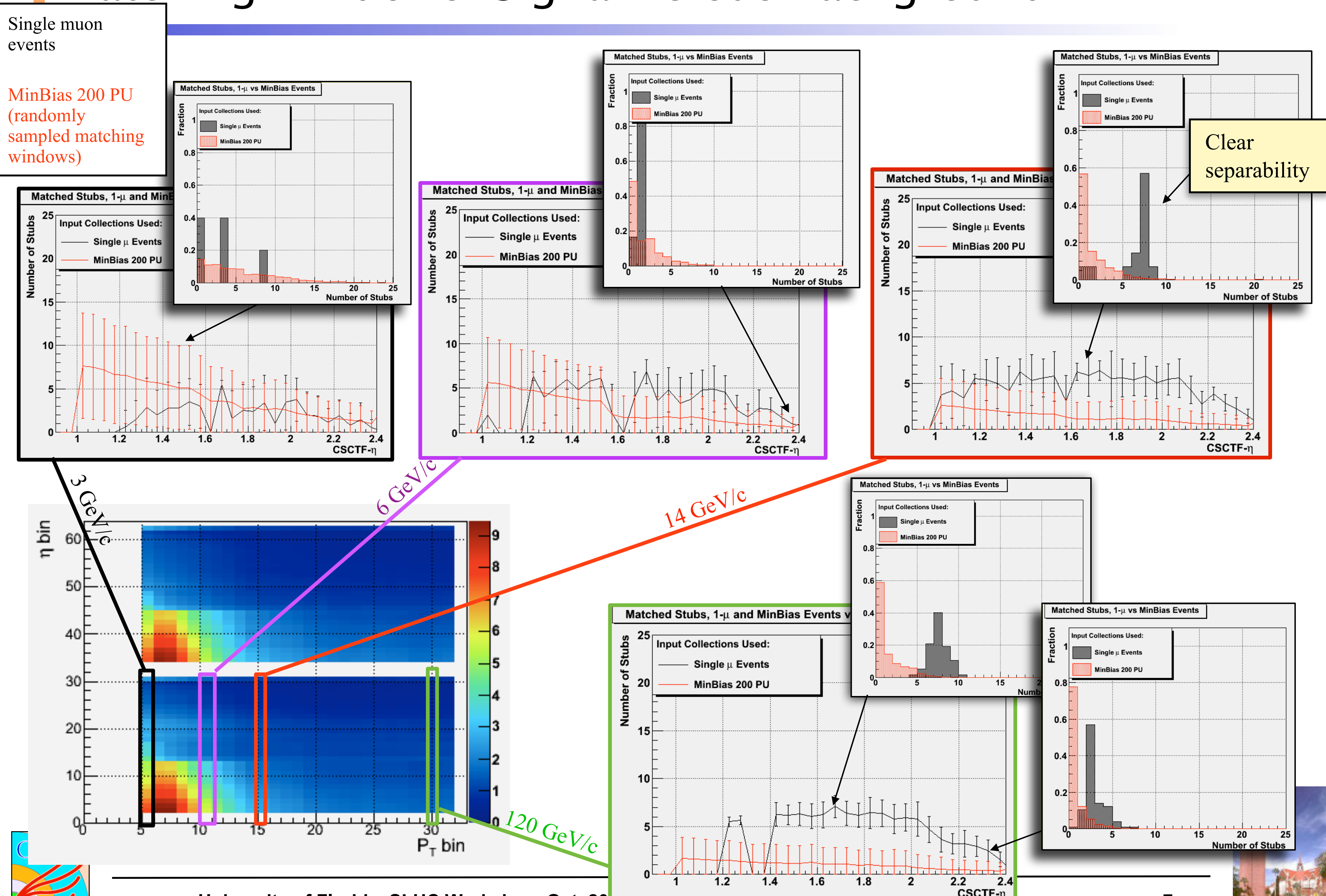


Here we see origin of inefficiency caused by Nstubs cut.

(1) |Eta|>2.1 Nstubs is seen to drop to ~2 (expect 4)

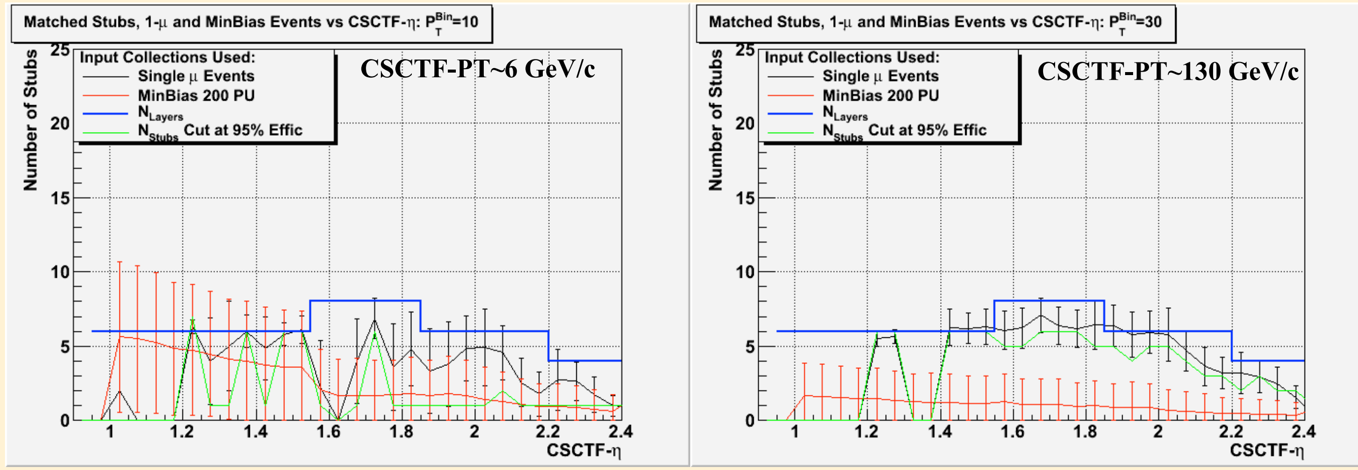
(2) Eta dependent switch in CSCTF-track quality assignment due to gap between inner and outer rings of ME2 and ME3 (matching windows are tuned for Q3 tracks). Q3:Q2 ~ 8:1 for 1 mu events (cf Q3:Q2 ~ 2:1 for MinBias)

Matching Windows: Signal versus Background



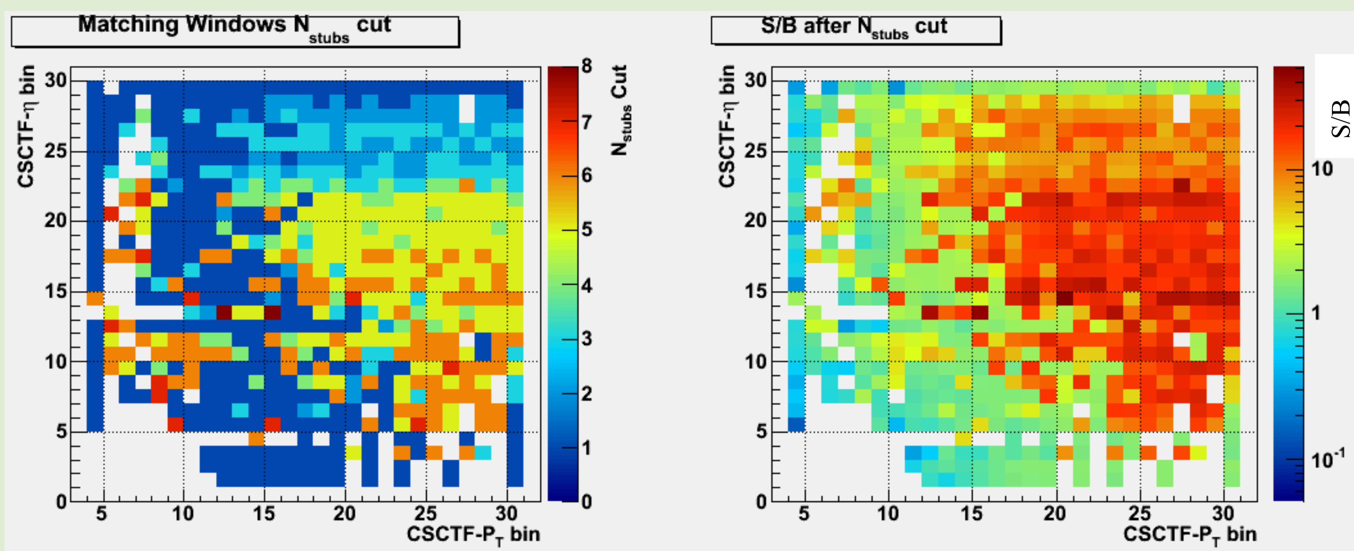
Matching Windows: Separating Signal from Background

Signal and Background vs CSCTF-η

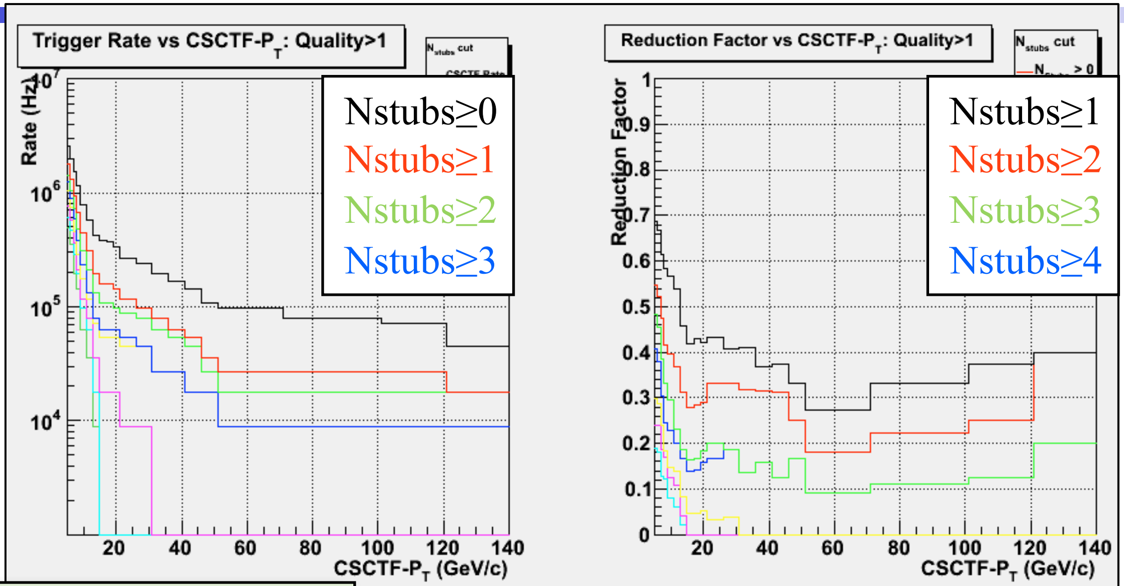


Once matching windows are re-tuned, expect that counting can provide a powerful handle for rate reduction from noise and CSCTF mis-measurement.

Example exercise: tune matching window bin-by-bin N_{stubs} threshold to accept 95% of signal stubs. Cuts and S/B versus bin seen on right ➙

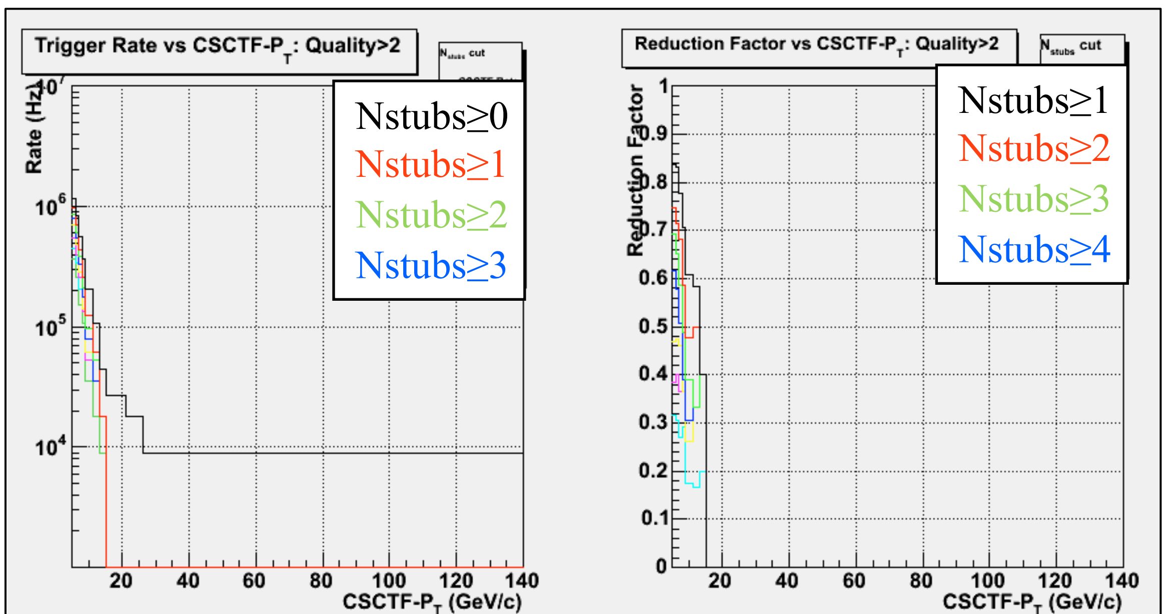


Rate Reduction from stubs in Matching Windows



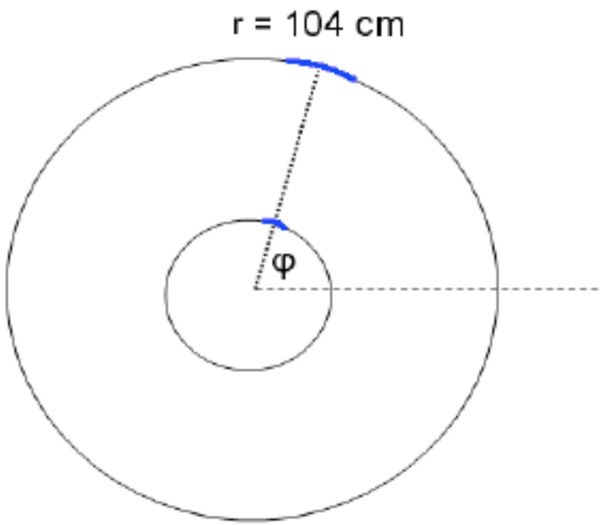
Rate and relative reduction contours with Quality≥2 CSCTF Tracks (versus Nstubs matching window cuts)

Rate with Quality=3 CSCTF Tracks



Rate reduction power is mostly related to CSCTF Quality. Improperly seeded windows miss underlying stubs. This can be a powerful weapon against CSCTF mis-measurement!

P_T Estimate 1: Using Δφ



Circle Fit Approximation:

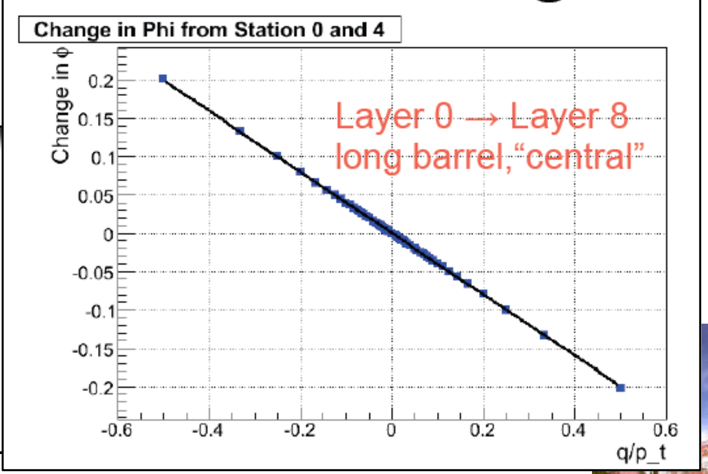
$\phi = \phi_0 + \arcsin(\zeta R / p_T)$

linear approximation:

$\Delta\phi \sim 1/p_T$

$\Delta\phi \sim \Delta R$

- sensors report local coordinate → global ϕ
- measure ϕ in 100 μm units of arc length at 104 cm
- $\Delta\phi_{09} = \Delta\phi_{ij} \cdot \Delta R_{09} / \Delta R_{ij}$
- $\Delta\varphi_{09} \rightarrow 1/p_T \rightarrow p_T$



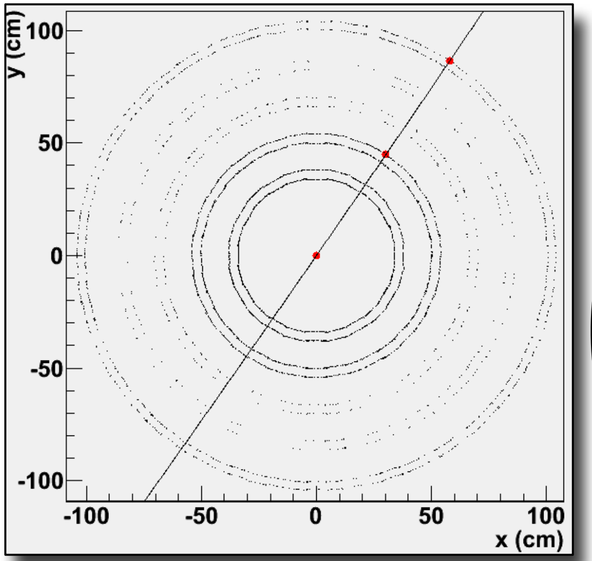
Approach demonstrated to achieve 2% P_T resolution

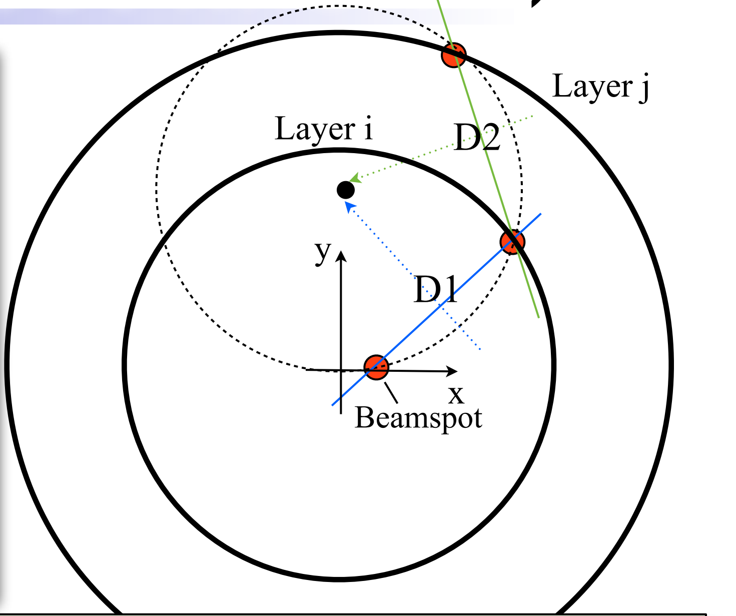
P_T with Beam spot drift

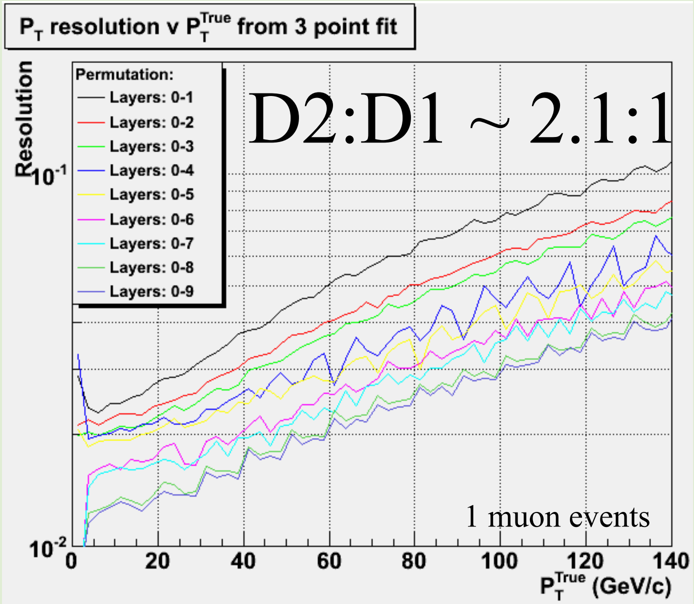
- Current algorithm takes filtered stub candidates and assigns P_T by finding effective $\Delta\varphi_{09}$ between tracker Layers
  - Uses linear fit between $1/P_T$ and $\Delta\varphi_{09}$, with (0,0,0) beamspot
  - This algorithm can be re-tuned to accommodate off-center (not investigated yet)
- Can we use the CSCTT model framework to accommodate beam spot drift?
  - Take filtered stub candidates and use a 3-point circle fit to find P_T
    - Algorithm 1: Assume a known beam spot and use stubs available from two tracker Layers
    - Algorithm 2: Assume unknown beam spot and use stubs available from three tracker Layers
      - Can then use DCA to provide beam spot location
    - Both algorithms fit two lines: L1 = $Point_i$ to $Point_{i+1}$ and L2 = $Point_{i+1}$ to $Point_{i+2}$ (points increasing in radius). Solve for the intersection of the two orthogonal lines which bisect L1 and L2
    - Working with engineer to understand how we can implement algorithm in HW

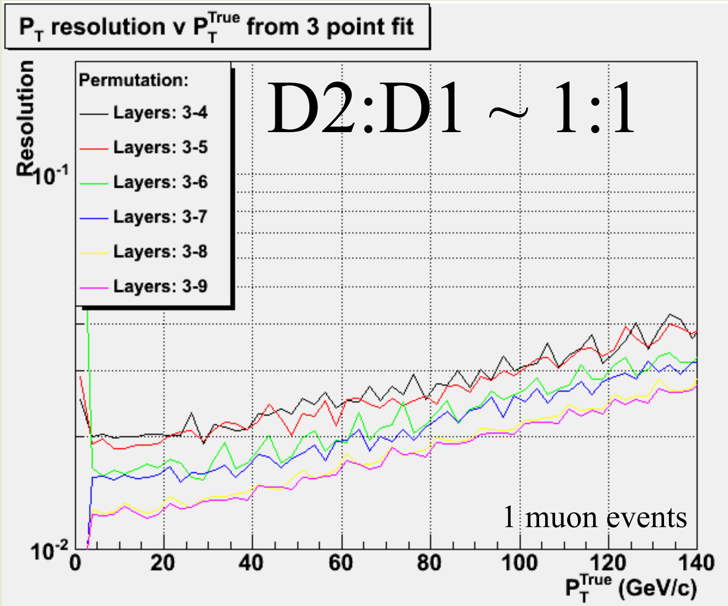
Circle-Fit P_T (beam spot known)

Once candidate stubs are identified, try all pair-wise combinations for 3-point fit (assume known beam spot) and look at resolutions (below).



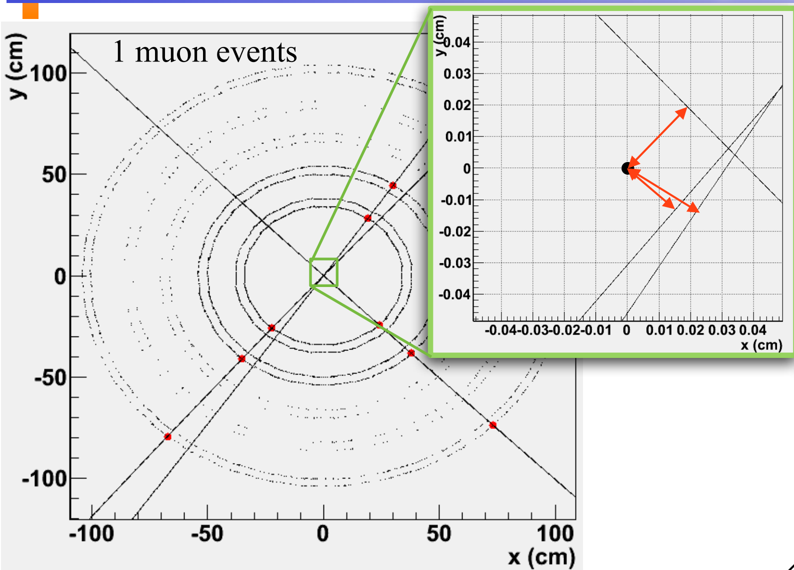


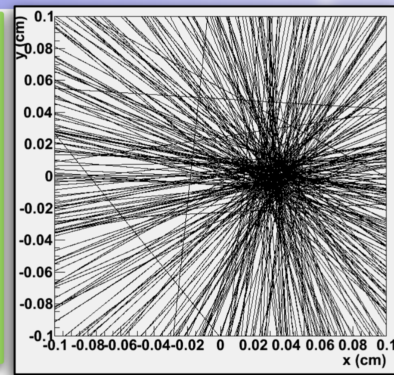


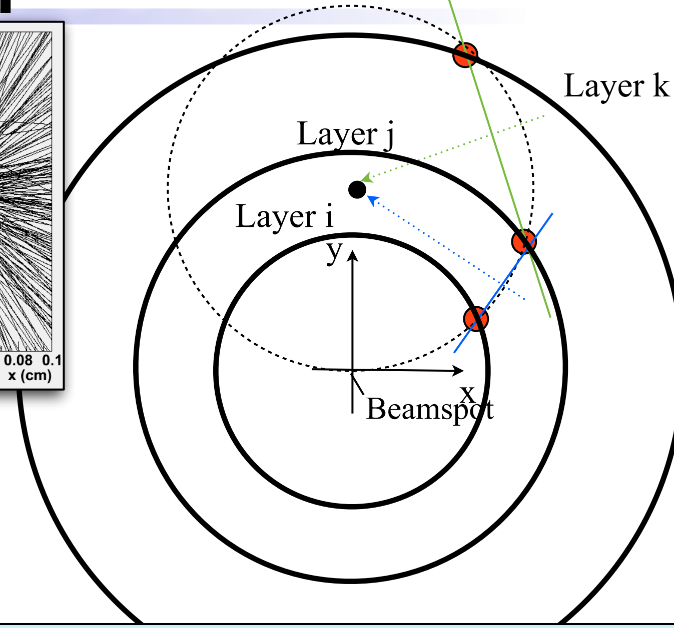


Result not too surprising, best resolution found when chords are nearly equal in length. 3 →9 gives best result with resolution ~ 1 - 3%

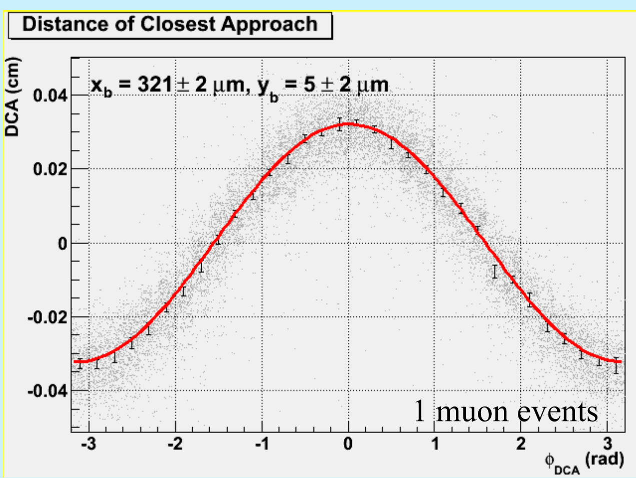
Circle-Fit P_T (beam spot unknown)

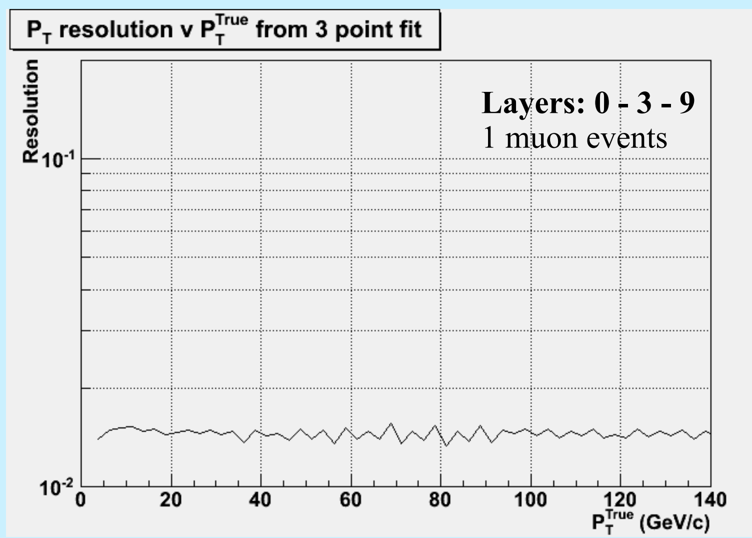






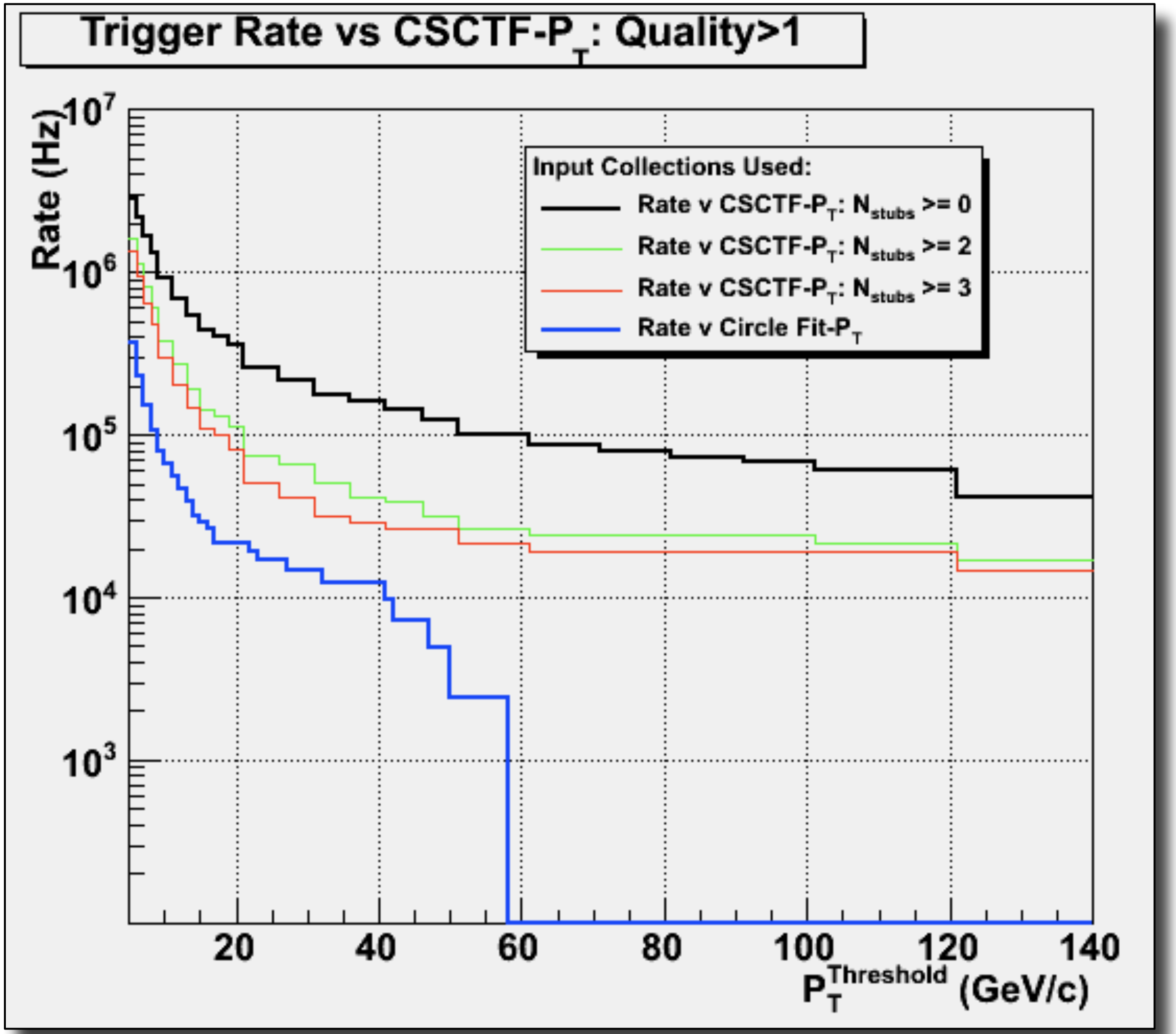
Estimate of beamspot using 3-point fit:
xbgen=0.0325 cm, ybgen=0.00506 cm





Flat P_T resolution when 3-Layer fit is used. We see resolution ~ 1.5%

Rate with Circle Fit



Take either 2 OR 3 (preference to 3 Layer) Layer Circle fit, reevaluate rate

cot(θ) & Z_0

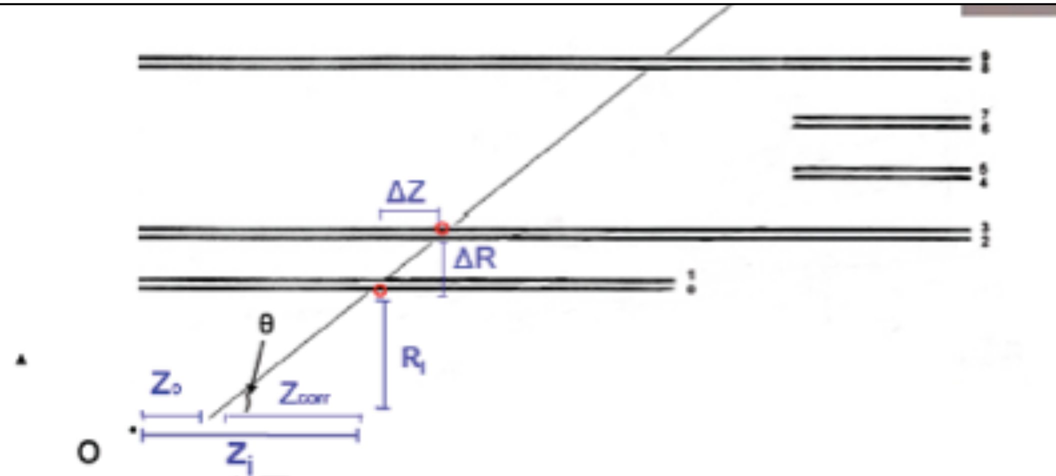
(1) Get cotθ from

$\Delta z_i = z_{stub1} - z_{stub2}$ and

$\Delta R_i = R_{stub1} - R_{stub2}$

(2) Get Z_{corr} from R_i (known) and cotθ

(3) Get Z_0 from Z_{stub1} and Z_{corr}



$$Z_{corr} = R_i \times \frac{\Delta Z}{\Delta R} \qquad Z_o = Z_i - Z_{corr} \quad \& \quad \cot(\theta) = \frac{\Delta Z}{\Delta R}$$

- Pairs of stubs used, first stub always from double stack #0-3
- Similar triangles
- cot(θ) and Z_{corr} calculated then stored in a lookup table.

CSTT model has been demonstrated to achieve Z_0 resolution 640 μm and cot(θ) resolution 0.002

Conclusions

- UF group has developed a CSC+Tracker Trigger (CSCTT) model for Level 1
- Seeding regions of interest within Tracker volume by using CSC Tracker-Finder allows one to make precise estimates of Pt via $\Delta\varphi$ or circle fit, Z_0, cotθ at L1. Beam spot possible as well.
- Matching windows cut on Nstubs can be a powerful weapon in rejecting fake muons or mis-measured CSC Track-Finder muons
- Various features of CSCTT algorithm are better understood
  - Currently studying various improvements to matching windows and filtering, as well as CSCTF-quality (lots of handles for fine-tuning)
- CSCTT code will likely enter CVS this week
- Internal Note in progress (hope to have first draft out soon)