Off-line Electron Seeding Validation - Update

Caleb Fangmeier
Ilya Kravchenko, Greg Snow

University of Nebraska - Lincoln

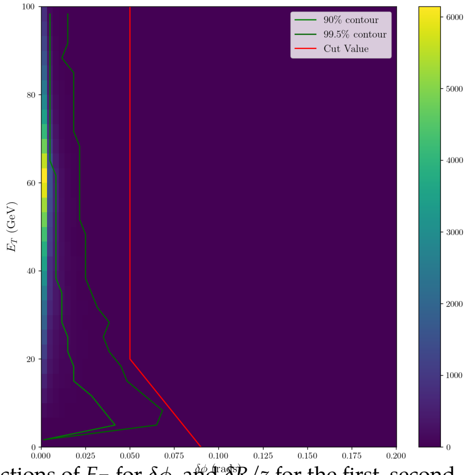
EGamma Reco/Comm/HLT Meeting — February 16, 2018

Introduction

- Our goal is to study **seeding** for the **off-line** GSF tracking with the **new pixel detector**.
- Specifically, we want to optimize the new pixel-matching scheme from HLT for use in off-line reconstruction.
- This Talk:
  - Show the effect of linearly scaling matching windows up and down
  - Show first set of **optimized** windows
  - Next steps
- Full set of results is available here
  https://eg.fangmeier.tech/seeding_studies_2018_02_15_12/output/

$\delta\phi$ Residuals

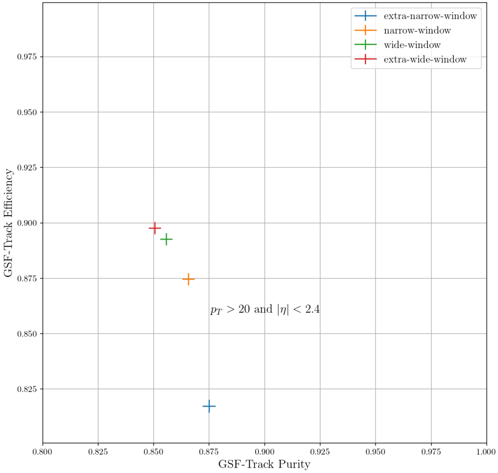
- Distribution of $\delta\phi$ residuals for first matched hits in truth-matched seeds where the hit was in BPIX-L1
- Truth-matching requires sufficient (75%) matched hits with a sim-track as well as less than 10% energy discrepancy between super-cluster and sim-track.
- Differential in E_T of the matched super-cluster
- Red line shows the default (aka HLT) window.
- Contour lines are biased by the matching cut necessarily being applied before deriving the contours.



Cut windows are specified as functions of E_T for $\delta\phi$, and $\delta R/z$ for the first, second, and third matched hits.

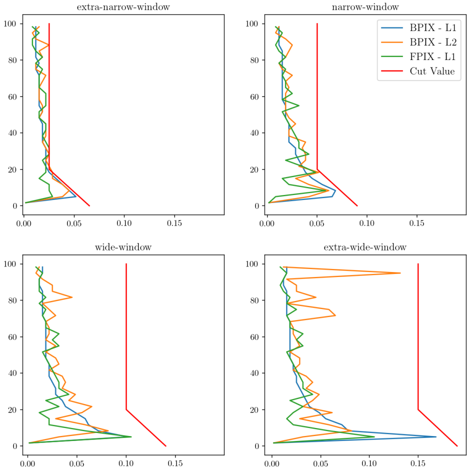
Linear Scaling of Windows

- Modified windows with uniform scaling
  - x0.5(`extra-narrow`)
  - x1.0(`narrow`)
  - x2.0(`wide`)
  - x3.0(`extra-wide`)
- Uniform scaling draws out a clear curve in efficiency v. purity.
- But can we do better? Find windows with points above the curve?



Finding more optimal windows

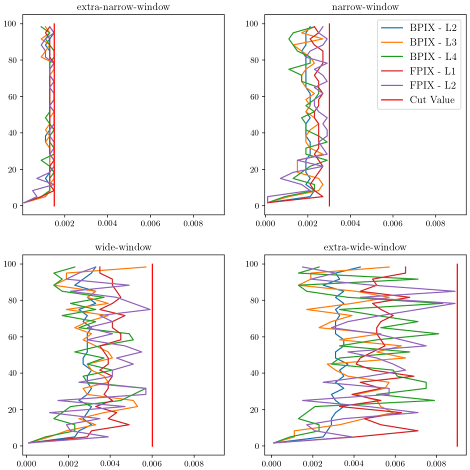
- Figure: first-hit $\delta\phi$ 99% contours for all relevant[1] pixel regions.
- Procedure: Select a cut that tends to reasonably follow the 99% contours in the `extra-wide` windows.
- Repeat this for each of the six windows.
- In this case, the `narrow` window seemed appropriate so this particular window was unchanged.



[1] meaning the subdetectors that have a substantial portion of first hits

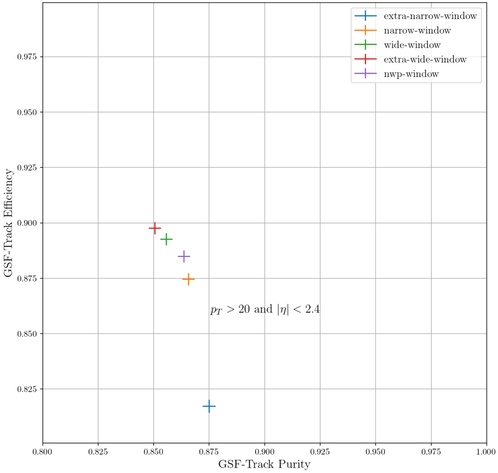
Finding more optimal windows - 2

- Figure: second-hit $\delta\phi$ 99% contours for all relevant pixel regions.
- Quite low statistics in some regions + looking at tails of distribution results in high variability
- Despite this, estimate an appropriate cut to be 0.005



Proposed New Working Point Performance

- New working point sets slightly above the linear-scaling curve
- Hints that better performance is achievable, but it's not obvious how to achieve
- Many ways to vary parameters...



Outlook

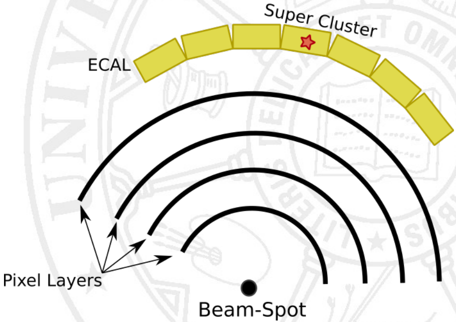
- Next steps
  - Testing with an complementary dataset (currently looking at $Z \to ee$ only)
  - Possibly breaking down windows sizes in η (code supports this, but is currently unused).
- Other Thoughts
  - What is an appropriate working point, and what performance can be deemed adequate?
  - Are there different figures-of-merit that must be balanced (cpu performance, specific background rejections.)?

BACKUP

- **Sim-Track -** A track from a simulated electron originating from the luminous region of CMS (beam-spot +- 5σ)
- **ECAL-Driven Seed -** A seed created via a matching procedure between Super-Clusters and General Tracking Seeds (Either from `ElectronSeedProducer` or `ElectronNHitSeedProducer`)
- **GSF Track -** A track from GSF-Tracking resulting from an **ECAL-Driven Seed**
- **Seeding Efficiency -** The fraction of **Sim-Tracks** that have a matching **ECAL-Driven Seed** (based on simhit-rechit linkage)
- **GSF Tracking Efficiency -** The fraction of **Sim-Tracks** that have a matching **GSF Track** (again, based on simhit-rechit linkage)
- **ECAL-Driven Seed Purity -** The fraction of **ECAL-Driven Seeds** that have a matching **Sim-Track**
- **GSF Tracking Purity -** The fraction of **GSF Tracks** that have a matching **Sim-Track**

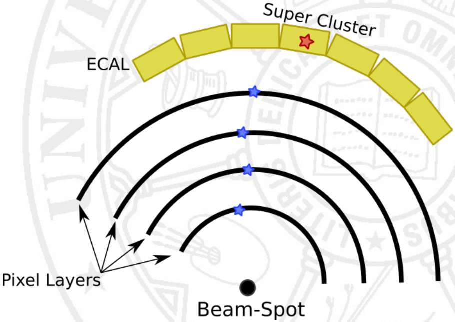
Triplet Electron Seeding - Setup

- Begin with ECAL super cluster and beam spot



Triplet Electron Seeding - Introduce Seed

- Now, examine, one-by-one seeds from general tracking*
- Note that we do not look at all hits in an event, but rather rely on general tracking to identify seeds.

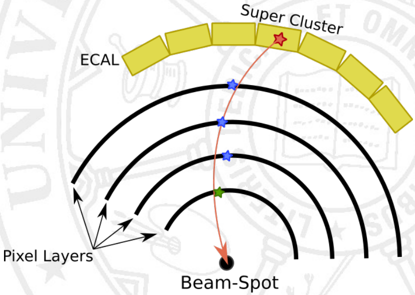


---

*initialStepSeeds, highPtTripletStepSeeds, mixedTripletStepSeeds, pixelLessStepSeeds, tripletElectronSeeds, pixelPairElectronSeeds, stripPairElectronSeeds

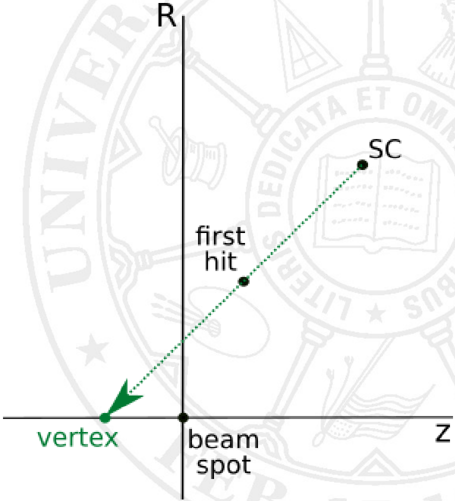
Triplet Electron Seeding - Match First Hit

- Using the beam spot, the SC position, and SC energy, propagate a path through the pixels.
- Next, require the first hit to be within a $\delta\phi$ and δz window. ($\delta\phi$ and δR for FPIX)
- δz window for first hit is huge as SC and beam spot positions give very little information about z.



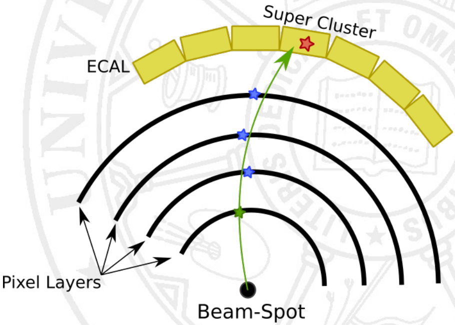
Triplet Electron Seeding - Extrapolate Vertex

- Once we have a matched hit, use it with the SC position, to find the vertex z.
- Vertex x and y are still the beam spot's.
- Just a simple linear extrapolation.



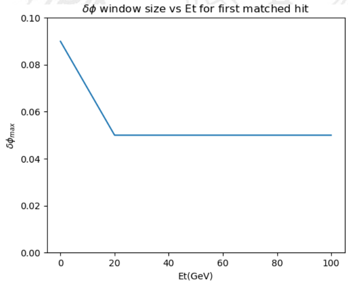
Triplet Electron Seeding - Match Other Hits

- Now forget the SC position, and propagate a new track based on the vertex and first hit positions, and the SC energy.
- Progress one-by-one through the remaining hits in the seed and require each one fit within a specified window around the track.
- Quit when all hits are matched, or a hit falls outside the window. No skipping is allowed.
- However, *layer* skipping is not ruled out if the original seed is missing a hit in a layer



Triplet Electron Seeding - Window Sizes

- The $\delta\phi$ and $\delta R/z$ windows for each hit are defined using three parameters.
  - `highEt`
  - `highEtThreshold`
  - `lowEtGradient`
- From these, the window size is calculated as $\texttt{highEt} + \min(0, \texttt{Et} - \texttt{highEtThreshold}) * \texttt{lowEtGradient}$.
- For the first hit, these parameters for the $\delta\phi$ window are,
  - $\texttt{highEt} = 0.05$
  - $\texttt{highEtThreshold} = 20$
  - $\texttt{lowEtGradient} = -0.002$

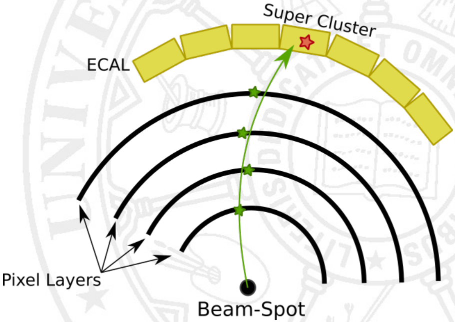


These parameters can be specified for each successive hit, and in bins of η, so optimization is a challenge!

Triplet Electron Seeding - Handle Missing Hits

- Finally, calculate the expected number of hits based on the number of working pixel modules the track passes through.
- Require exact[1] number of matched hits depending on the expected number of hits.
  - If $N_{\text{exp}} = 4$, require $N_{\text{match}} = 3$
  - If $N_{\text{exp}} < 4$, require $N_{\text{match}} = 2$
- If the seed passes all requirements, all information, including
  - Super cluster
  - Original Seed
  - Residuals (For both charge hypotheses)

  are wrapped up and sent downstream to GSF tracking



[1] Exact, rather than minimum to deal with duplicate seeds in input collection. Could switch to minimum with offline cross-cleaned seeds.