SVT Alignment: status and update

PF, Norman, Alessandra

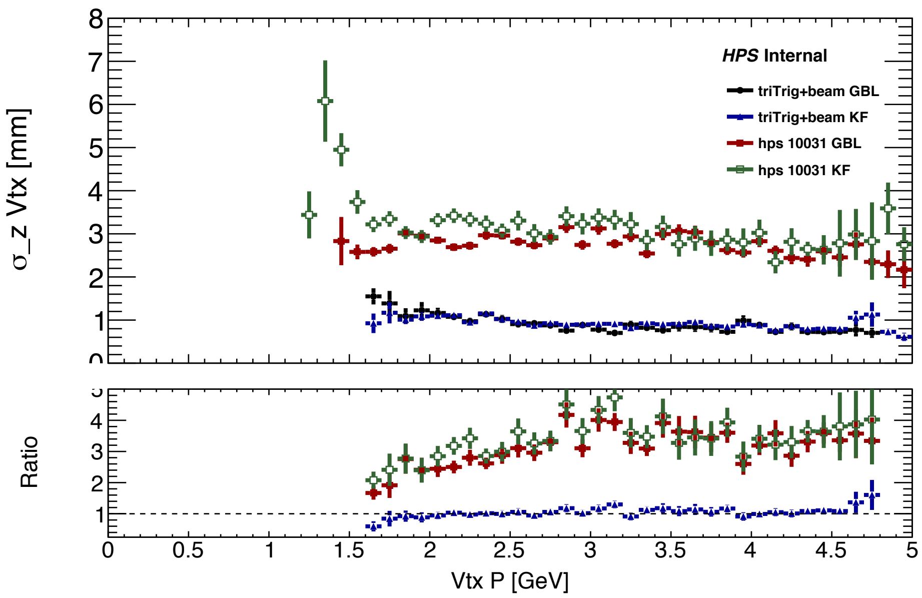
05/14/2020

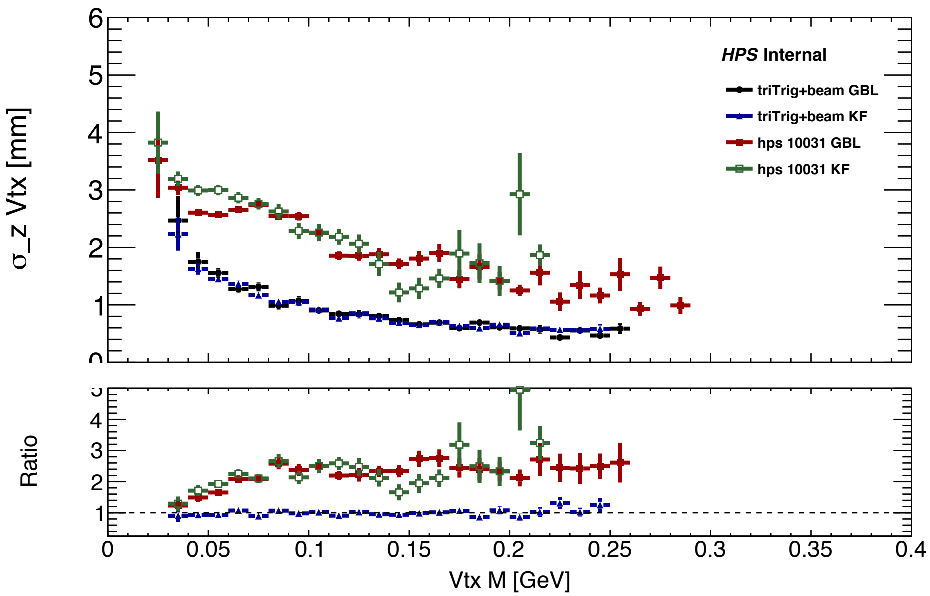
Outline

- **Introduction**
  - Results from the April analysis workshop
  - Detector alignment from last collaboration meeting
  - Available data, skims, MC samples for alignment studies
  - Software status and readiness
- **Global alignment tests**
  - Multiple events vertices fits in FEEs
  - Opening angle analysis
  - Current results and status
- **Local alignment tests**
  - MC based MPII tests
  - Local alignment for 2019 data
- **Current performance aiming for Jeopardy**
  - L1/L2 local alignment
  - Vertex z resolution and bias
- **Conclusions and next steps**

Introduction - Highlights from April's workshop

- First results on 2019 data and MC readiness were shown at the [April's 2019 Analysis Workshop]
- Details on the selection in the backup
- These plots were made on a large fraction 10031 events and triTrig+beam generated by TT back in end of March. **No skims back then**
- MC reproduces the expected resolution plot produced before the upgrade (see slide 25 of this talk)
- Alignment is top priority for 2019 data processing
- **First results have shown a x2-3 worse resolution wrt trident MC + beam**





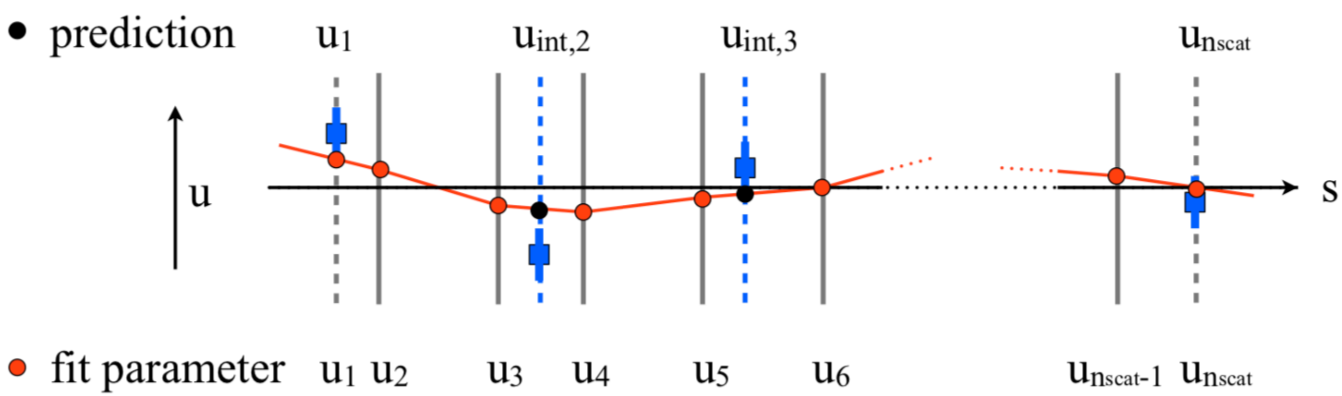
Introduction - Calibration Data and MC samples

- A set of samples have been selected for the SVT calibration:
  - Full Energy Electron (**FEE**) trigger: **10103** and **10104** B-Field ON
  - **FEE** trigger: **10101** B-Field OFF
    - FEE have high momenta tracks to minimise MCS
  - **V0 skims**: **10031** both with Ecal Cluster on Track (V0Skims) or without (V0SkimsLoose)
    - Illuminate both electron (hole) and positron (slot) sensors
- The data sets information is summarised SVT Alignment Skims
- In addition MC samples used for checking perfect geometry are (for the moment):
  - **Tridents** (TriTrig): signal only and signal + beam overlay

- **All through future talks on alignment I'll use L1-L7 nomenclature.**

GBL Tracking - Recap

- **General Broken Lines (GBL) is a track refit algorithm that add the description of multiple scattering to an initial trajectory**
  - Based on propagation in magnetic field
  - Constructed from a sequence of thin scatterers
  - **In the case of silicon detector a scatter also has a measurement (in the form of local residual in the sensitive u direction)**



- The initial trajectory should be 'close enough' to the solution and provide a reasonable estimate of the particle trajectory
- GBL is used in hps-java to refit helical track fits
- It is **iterated** (5 iterations) in our code to ensure convergence of the track parameters corrections

Introduction - SW status and readiness - just for reference

SLAC

- Majority of alignment software is in place since 2016 alignment campaign.
- We use hps-java with custom steering files for producing
  - Output monitoring files - ROOT format - hps-DQ-macros
  - Millepede input files for local alignment for hps-mille
  - SLCIO files for dedicated analysis of the results using hpstr
- Work in the past month has been made on the alignment chain:
  - GBL Code review for global derivatives for local alignment
  - Fix our MPII wrappers for 2019 geometry. MPII can now run on 2019 data/MC
  - Tests on MC misalignments for validation
  - Use of pre-fitted hits for faster processing of iterations
  - Improved monitoring plots/tools and collect all available monitoring drivers useful for alignment purposes
- More informations available 2019 HPS Alignment Notes

HPS Alignment strategy

- HPS geometry is implemented in the software without a direct support for MPII global structures alignment
- Since 2016, the strategy to align the detector was divided in aligning first global structures, i.e. front vs back of the detector, top/bottom angles and relative positions … and then MPII was invoked for aligning the single sensors
- I will go through first tests made on
  - **Global alignment**:
    - Opening angle correction with BField OFF and BField ON
    - Multi Event FEE Vertices
    - Effects on tracking and vertexing
  - **Local alignment**:
    - Validation of MPII on Trident MC for new thin sensors
    - Test of MPII on 2019 V0 skims
    - Effects on tracking and vertexing

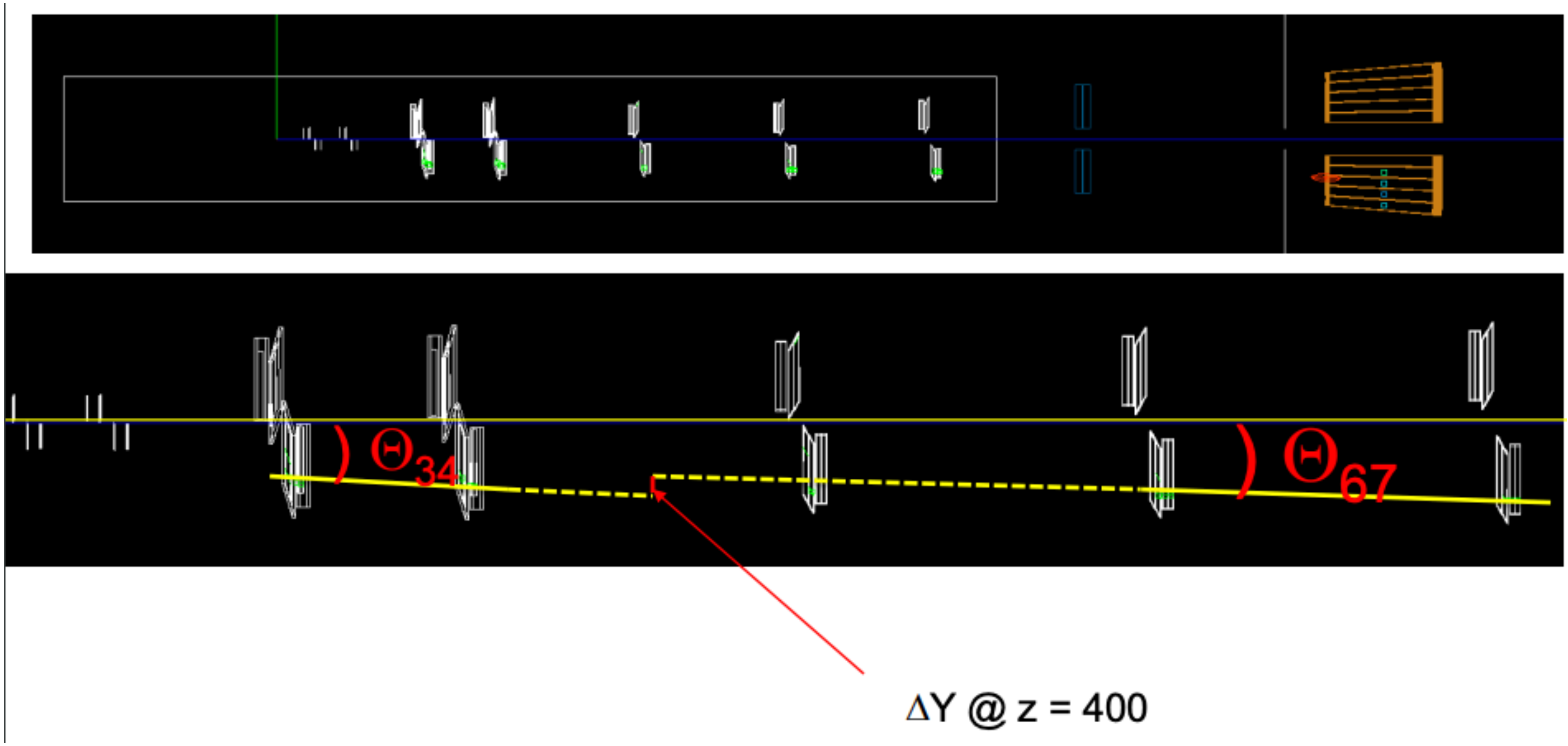
Global alignment test: Bottom Volume opening angle

Global Alignment - Opening Angle studies

- Used **field-off straight tracks** to assess the SVT alignment (run 10101)
- Single cluster events with full energy, 2H02 wire as target (-2267mm upstream), use of 1D strip hits
- Straight tracks do not illuminate L1 & L2, opening angle between front / back from layers 3-4 and 5-6 **(top)** , 3-4 & 6-7 **(bottom)** => impose $\theta^y_{34} + \Delta\theta_y = \theta^y_{67}$
- Offsets are compared at the pivot point at z = 414mm
- Similar exercise using **FEE runs 10103 and 10104 (bottom only as Ly7 Top is not functioning in these runs)**

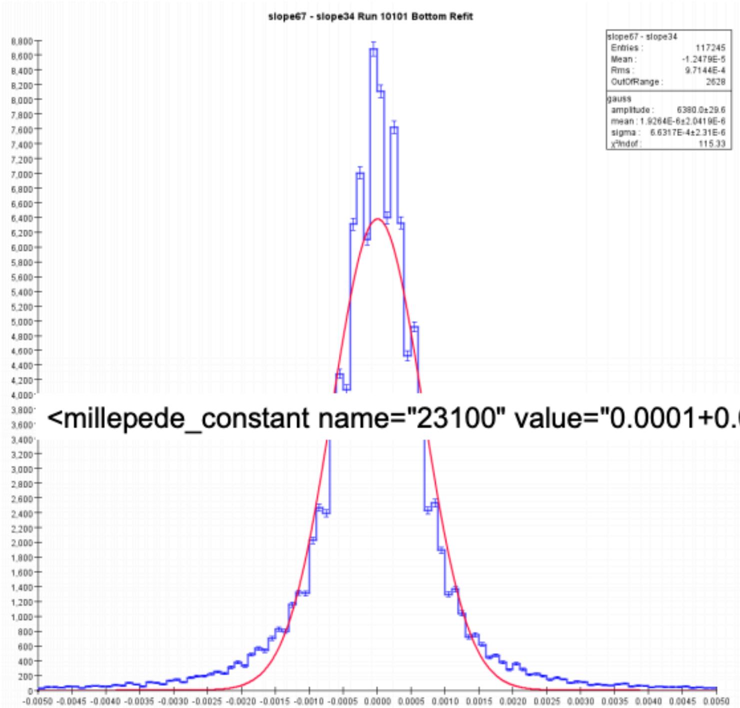
Norman's talk 10 Sept 2019

Norman's talk Apr '20 analysis WS

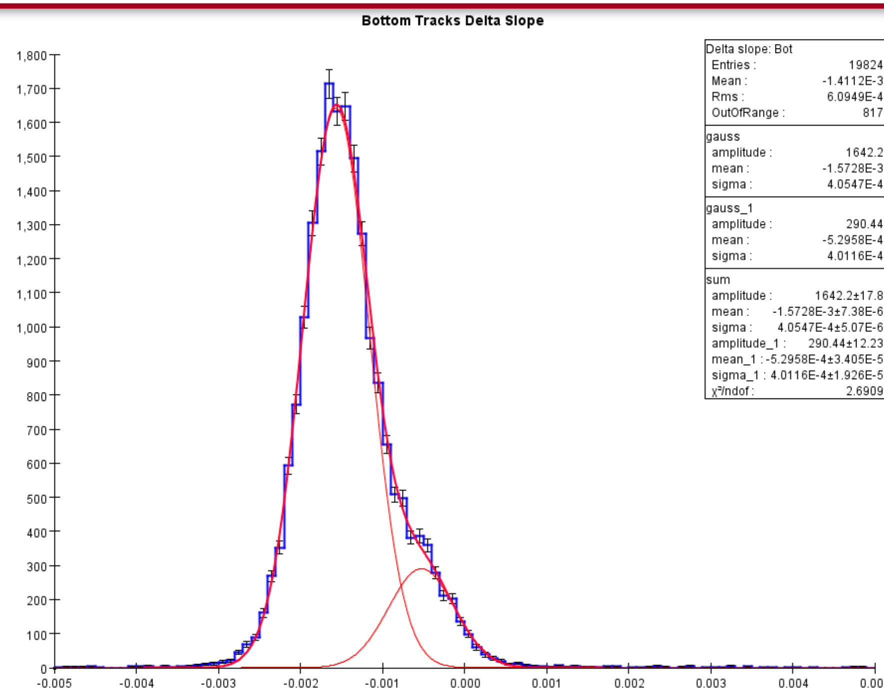


Global Alignment - Opening Angle studies

<millepede_constant name="23100" value="0.0001+0.00157"/>



<millepede_constant name="23100" value="0.0001+0.000911"/>



12

BFIELD OFF

no L1/L2 hits

$\Delta\theta_y(bottom) = 0.9\ mrad$

$\Delta\theta_y(top) \sim 0\ mrad$

BFIELD ON

L1/L2 hits on track

$\Delta\theta_y(bottom) = 1.5\ mrad$

$\Delta\theta_y(top)\ N.A. \rightarrow 0$

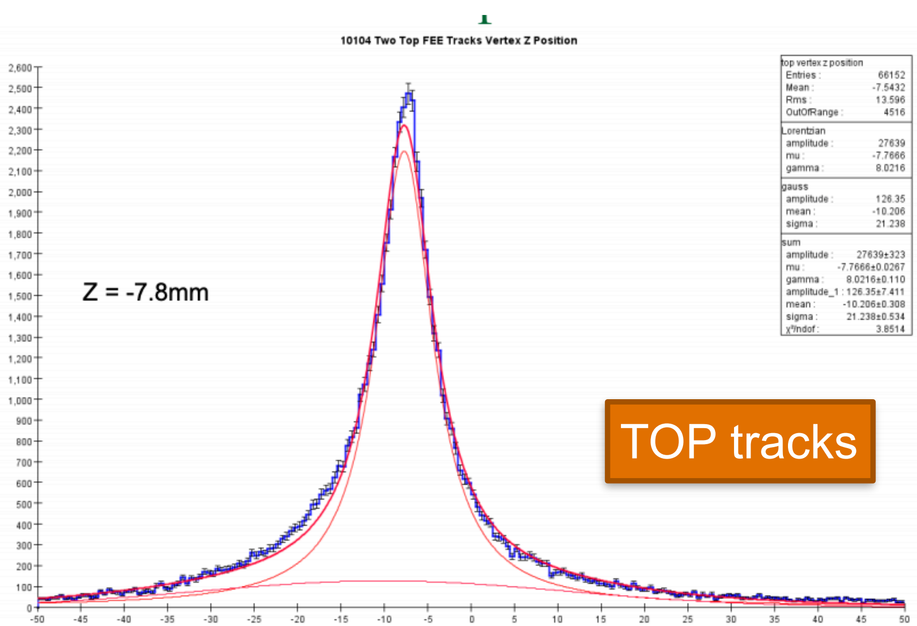
The difference in the two results **might** be due to the tension due to large internal misalignment in the thin layers

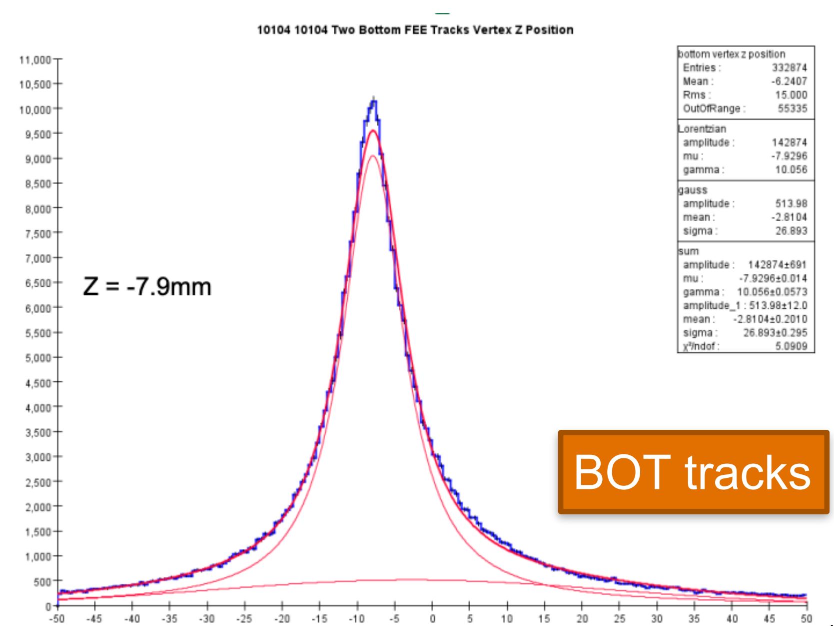
Norman's talk Apr '20 analysis WS

Norman's talk 10 Sept 2019

Global Alignment - Multi events FEE vertexing

- A way to cross-check the effect of opening angle is to **form vertices in top and bottom separately**
- Original study by Norman using **2 FEE tracks** separately for top and bottom from different events and vertex them to get unconstrained vtx position
- The **z position** of the multiple FEE vertex “insensitive” to some global alignment DoF, such as front-back opening angle [see backup]
- The **x-y position** of the vertex can give an indication of $\Delta\theta_y$

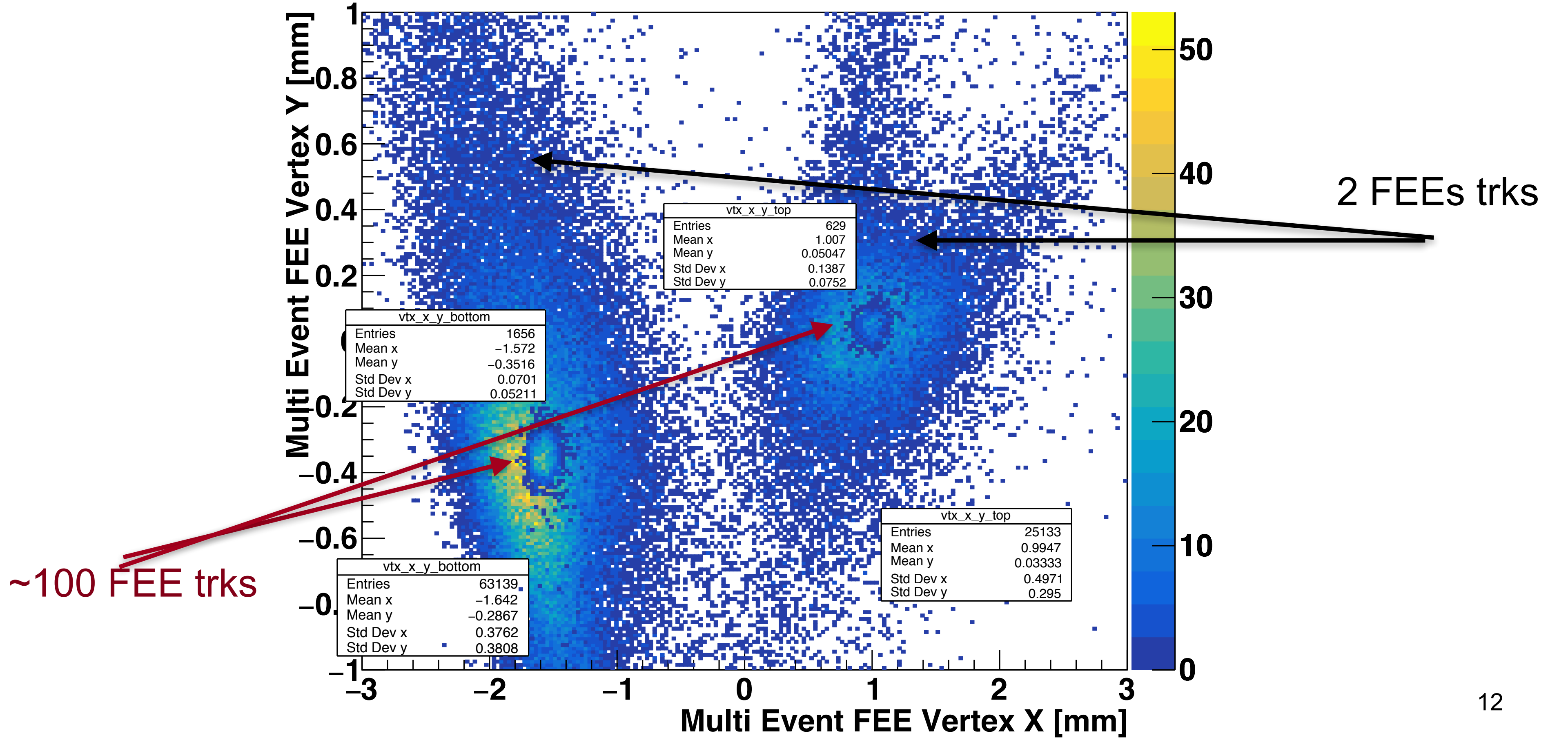




Norman's talk Nov '19 Coll meeting

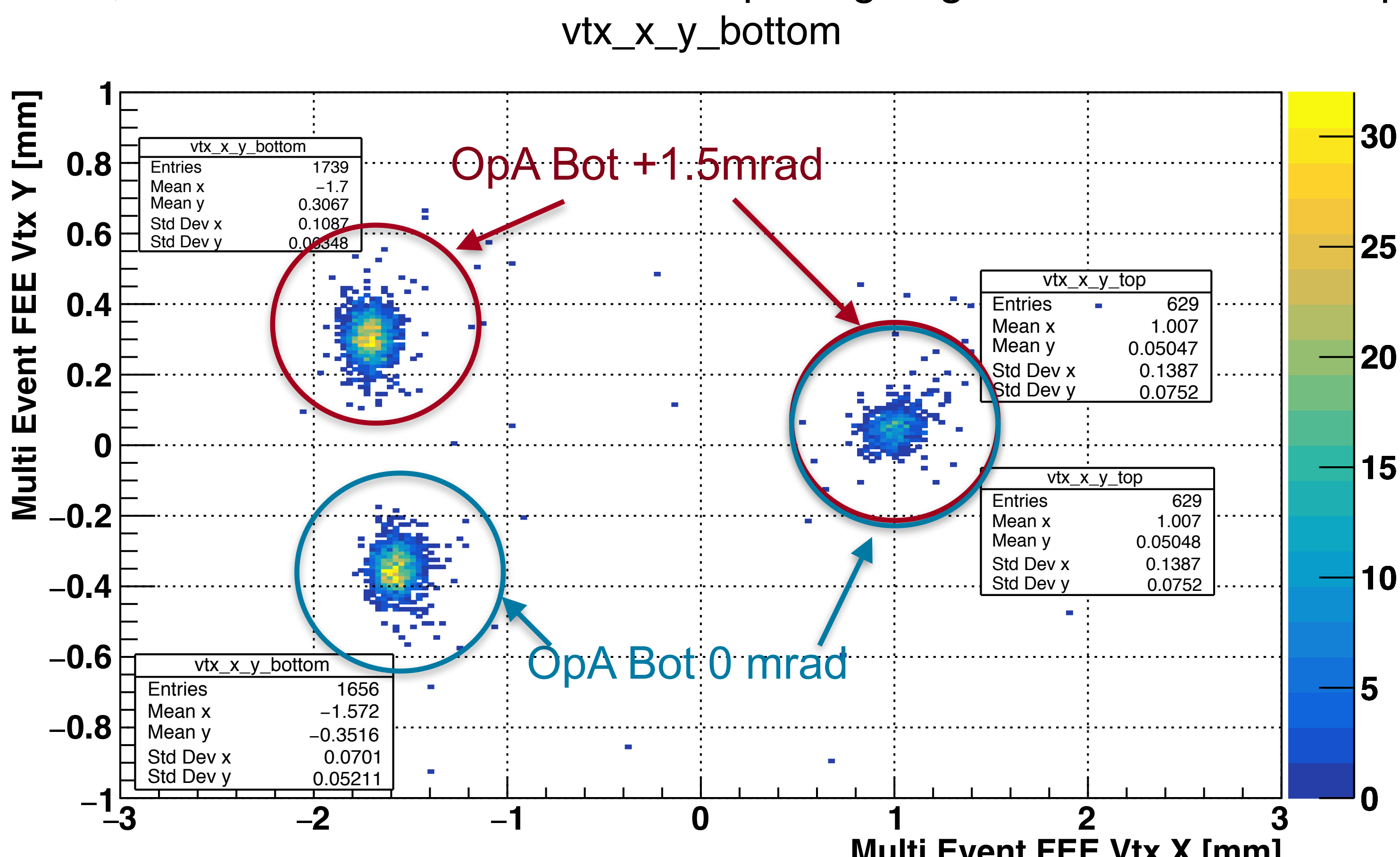
Global Alignment - Multi events FEE vertexing

- First, we updated the Multi Event FEE Vertexer to accept more than 2 tracks per event (see iss687_dev MultiVtxer).
- Clear effect on the x-y position resolution wrt 2-tracks vertices
- Events are collected, vertices are fitted in 100 tracks chunks, or less if not available: i.e. if 150 tracks are found 2 vertices are formed with 100 and 50 tracks, respectively.



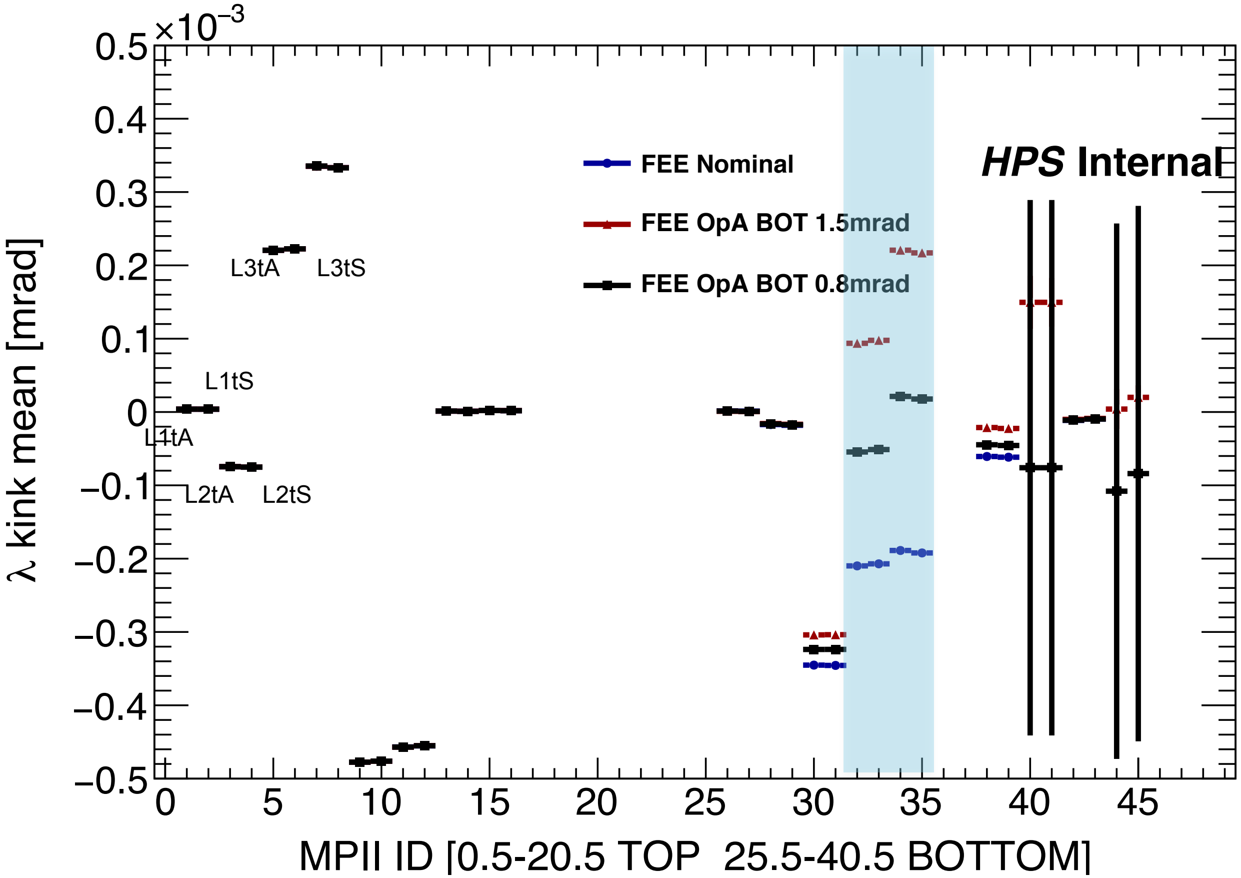
Global Alignment - Multi events FEE vertexing

- $\Delta\theta_y$ move VTX position along Y, movement on X axis (minor), Z is stable.
- We notices that $\Delta\theta_y = 1.5$ mrad seems to "overcorrect" of **~ factor 2** the Y position of the VTX, if we assume that there is no opening angle correction in the top volume.



Global Alignment - Multi events FEE vertexing

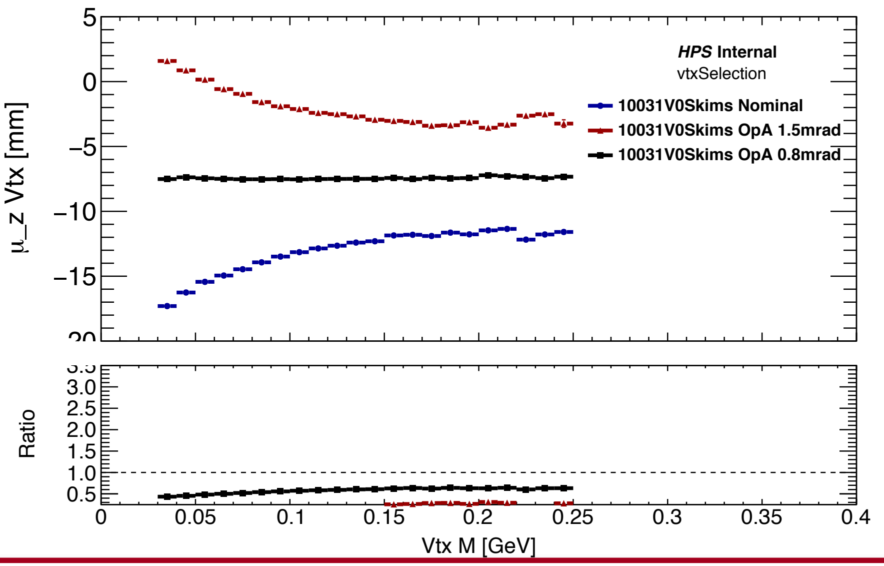
- Additionally, $\Delta\theta_y$ is expected to change the Ly4-Ly5 kinks in GBL refits -> Ideally they should be going to 0 if sensors are aligned [shown later]
- The plot shows the λ kinks as function of the MPII ID of each sensor. We have 20 planes per volume, first 20 are top [some are off and do not show in the plot]
- Shaded area highlights the Ly4-Ly5 kinks. $\Delta\theta_y$ ~ factor 2 larger [kinks flip from negative to positive] from BFieldON

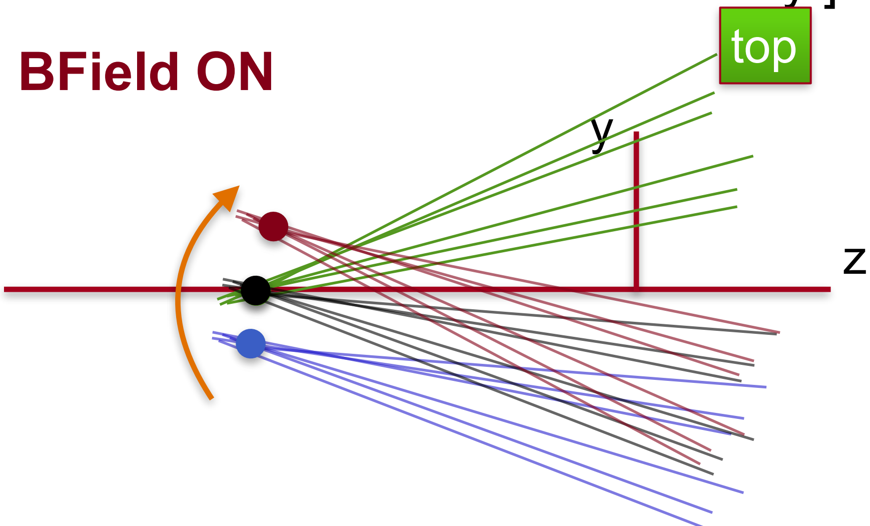


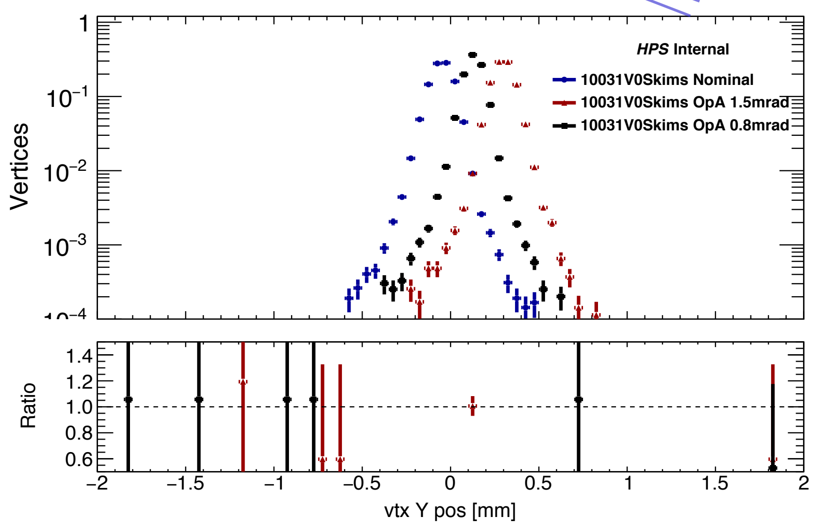
Large error bands for outer sensors are due to low stats:
- Artefact of TProfile I use => will be investigated and fixed soon

Global Alignment - Vertex z vs InvMass

- Opening angle should flatten the Vtx z dependence with InvM.
- Applying $\Delta\theta_y = 0.8$ mrad [At that moment I didn't recall Norman's study!]
- **Seems like it's a factor ~2 too big in BField ON**







Disclaimer: This internally removes the top-bottom tracks opening angle dependence from the z position as we bring y location of the beam spot to be the same between top and bottom
- **Doesn't mean that only changing bottom is correct, but there are some indications this could be a plausible solution**

see Norman's SVT tracking Talk : opening angle for top compatible with 0

Global Alignment - Vertex z and position resolution vs InvM

- This simple global alignment has very low impact on track quality, number of hits-on-tracks [backup]
- Fixing the DeltaY between top/bottom standalone vertices removes the vertex location dependence from opening angle by construction
- The Unconstrained Vtx resolution is extracted by requiring:

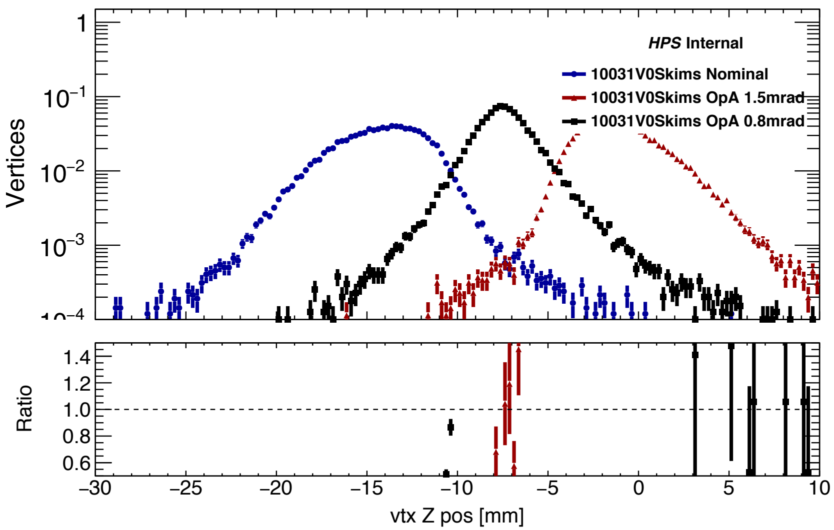
$p(e^-) < 3.4\ GeV$

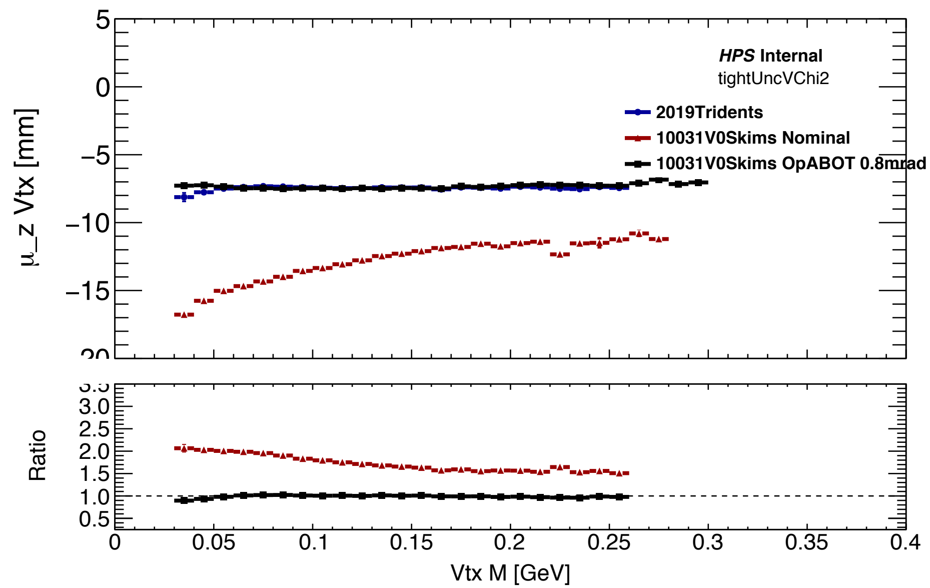
$p(e^-)$ and $p(e^+) > 0.6\ GeV$

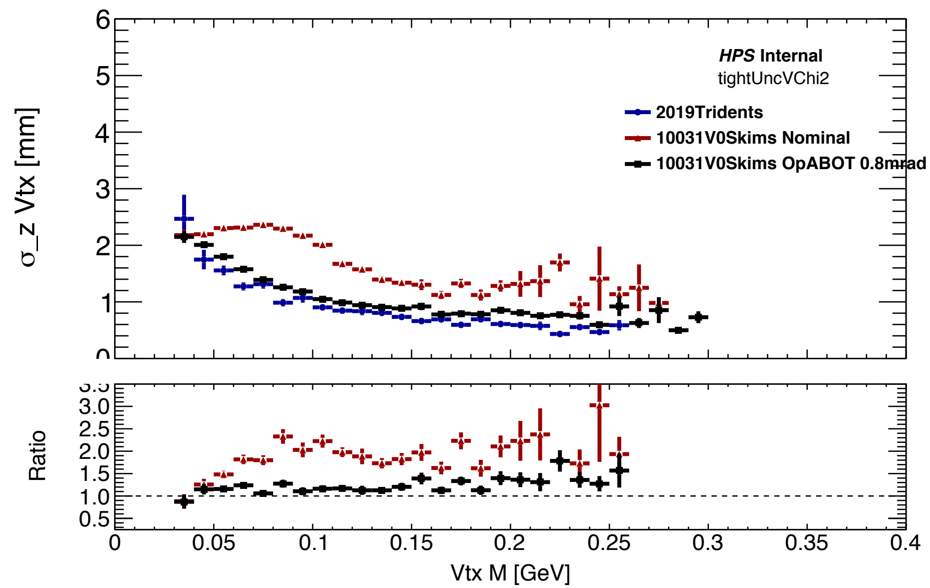
$N_{hits}^{track} > 4$

N_{L1hits} track$(e^+) > 0$

$\chi^2_{unc-vtx} < 10$



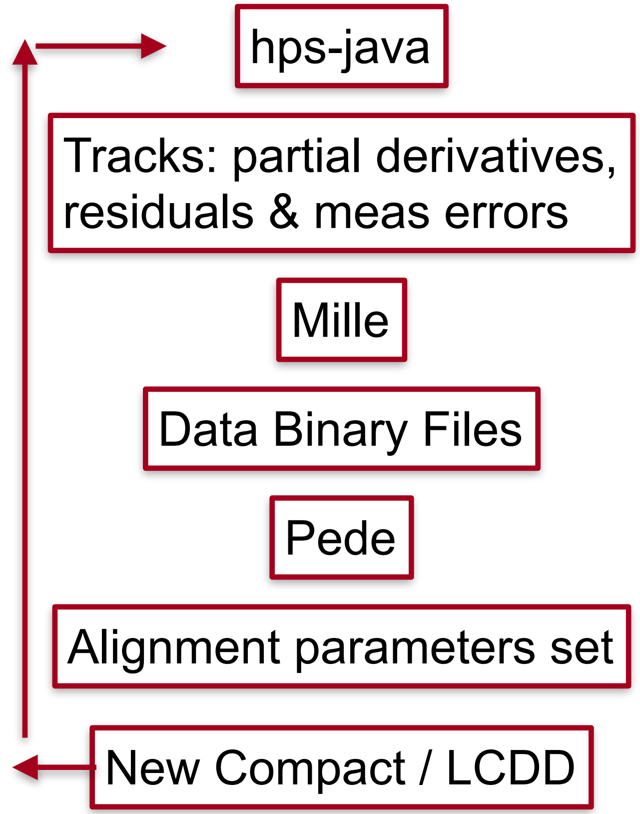




Local alignment tests - MPII code validation and L1/L2 alignment

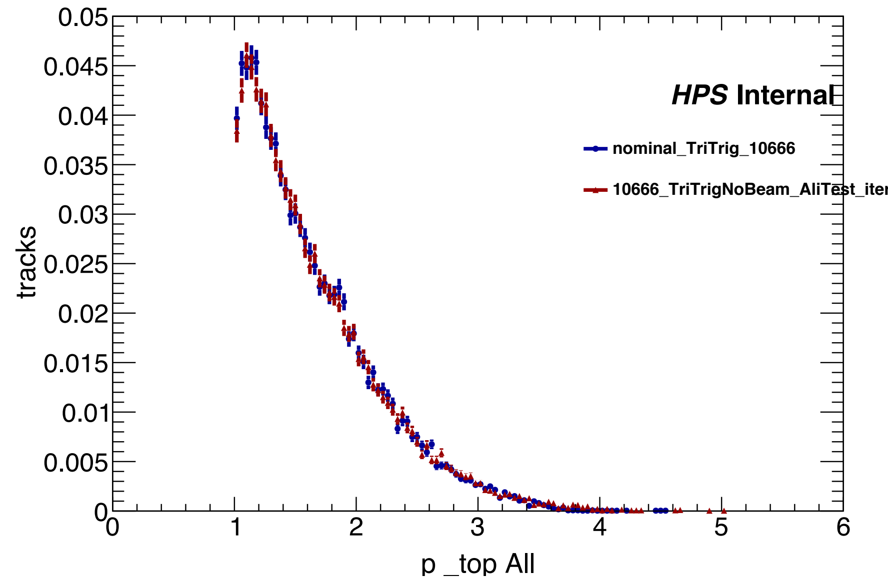
Local Alignment Tests - MPII Validation

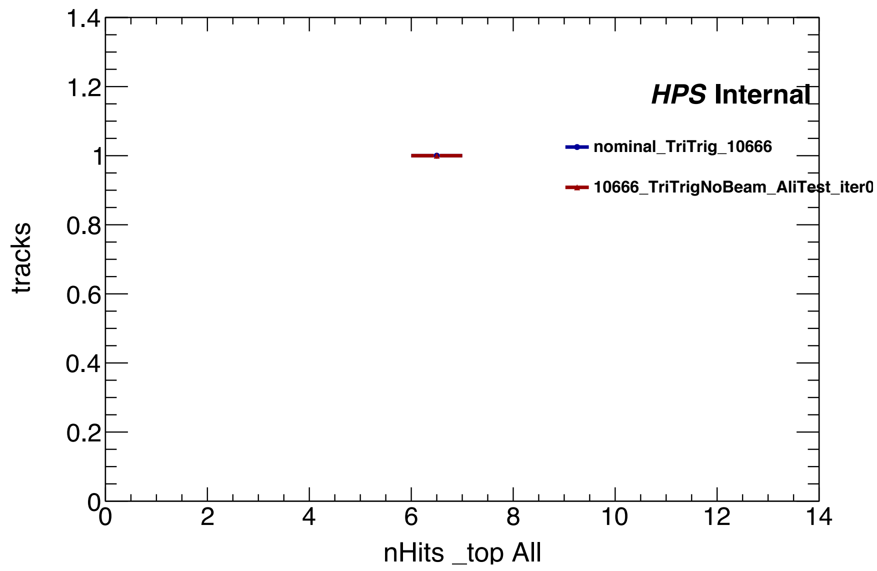
- MPII is the software used for aligning the single sensors in HPS (internal/local alignment)
- First thing to do before jumping on 2019 alignment is to assess the capacity of our alignment packages to:
  - Identify misalignments
  - Properly re-align them
  - Form a new aligned geometry
- These tests were done on MC samples with perfect geometry
- Used TriTrig 2019 with new thin layers L1/L2 geometry modelling and **no beam overlay**



Track selection for alignment tests

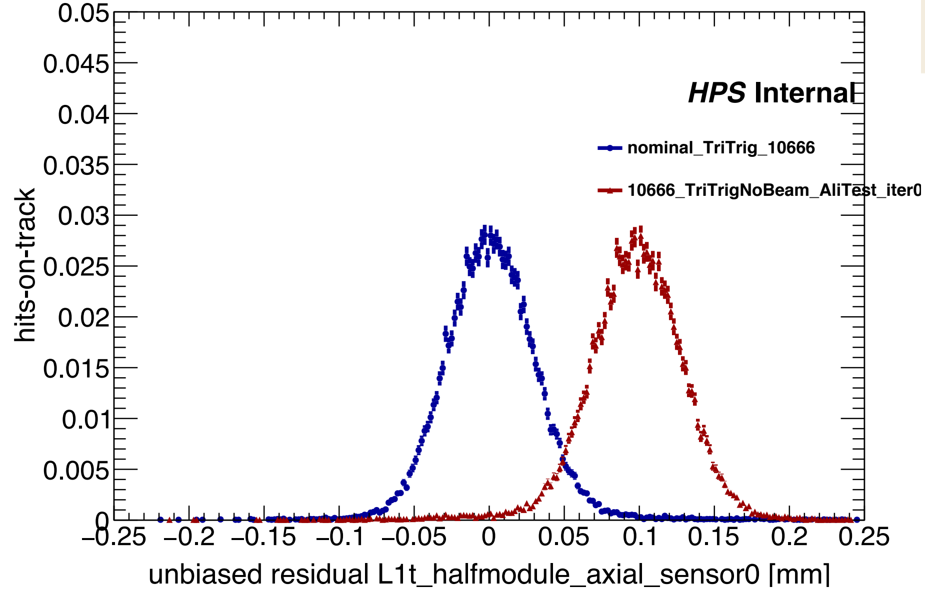
- Alignment track selection:
  - At least 1 vertex in the V0 collection
- Track selection:
  - **p > 1 GeV**: reduce Multiple coulomb scattering contribution
  - **>= 6 3D hits on tracks**: stronger constraints from rest of the detector
  - **X2<50**: better track quality
- **In the MC generated for 2019 there are no tracks with 7 hits due to the conditions that were used to reconstruct the samples.**
- So only 6 hits tracks are selected

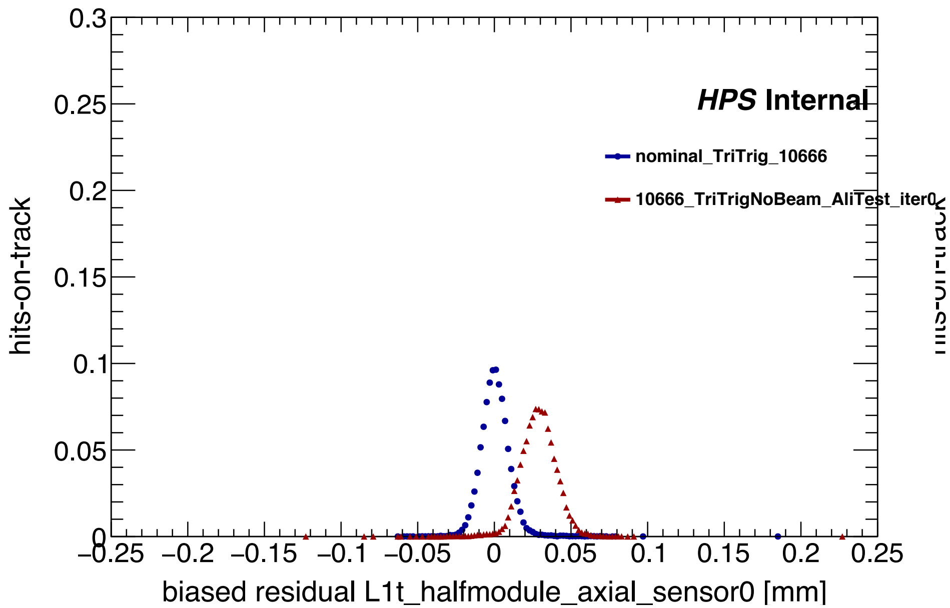


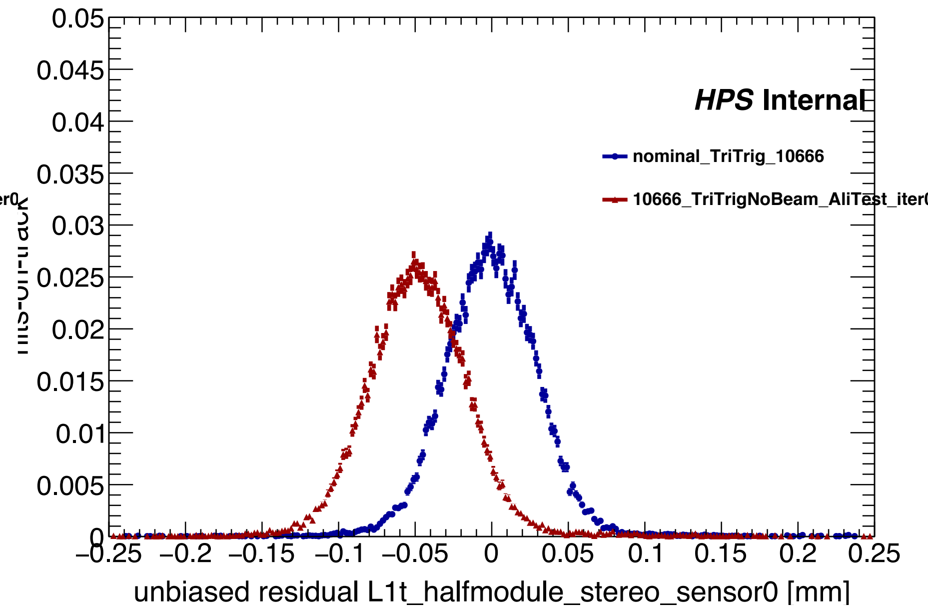


Local Alignment Tests - L1 top axial du translation

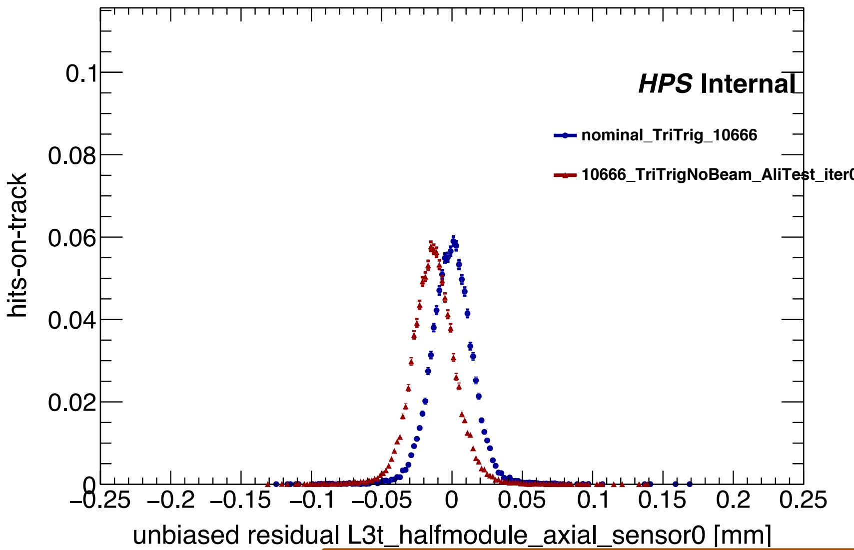
- Moved **L1 top axial sensor** of $du = +100\ \mu$m to validate alignment chain
- The size of misalignment can be identified by GBL unbiased residuals [new]
- The bias in the stereo residual is due to the reflection of the axial side misalignment -> **need to be recognised as correctly placed in the solution**

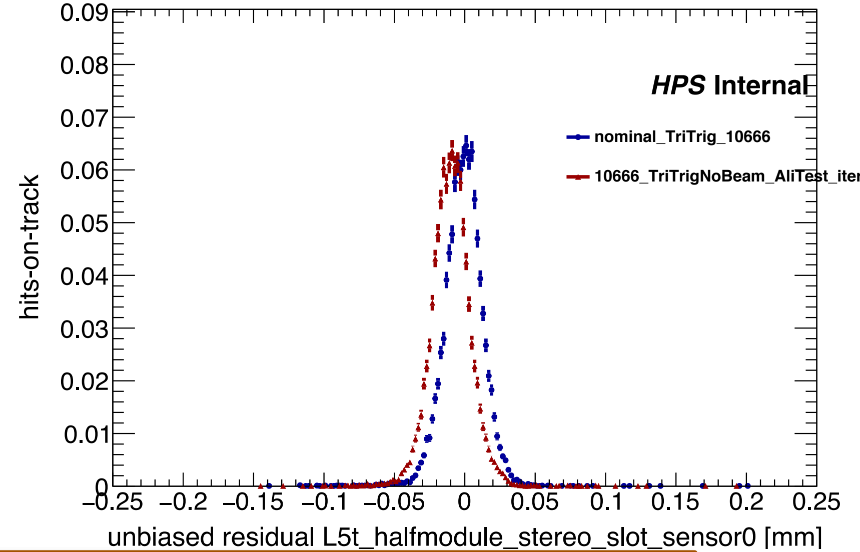






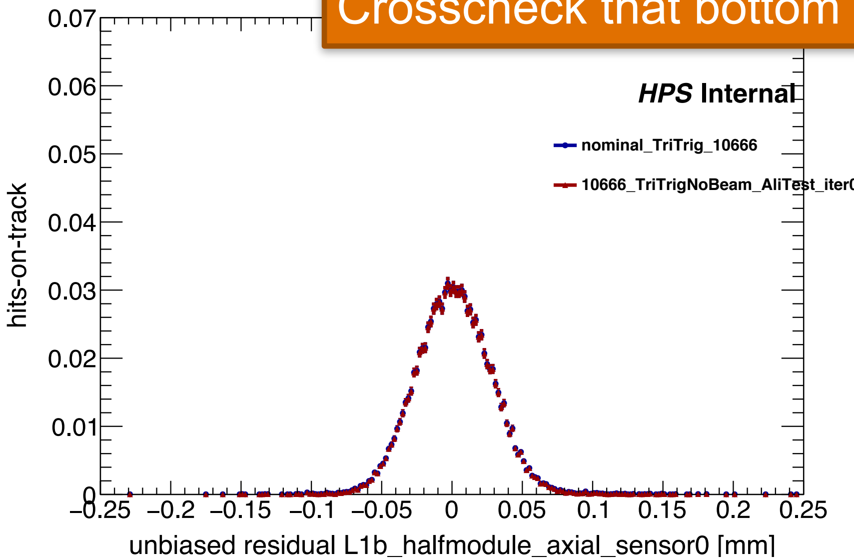
Local Alignment Tests - L1 top axial du translation

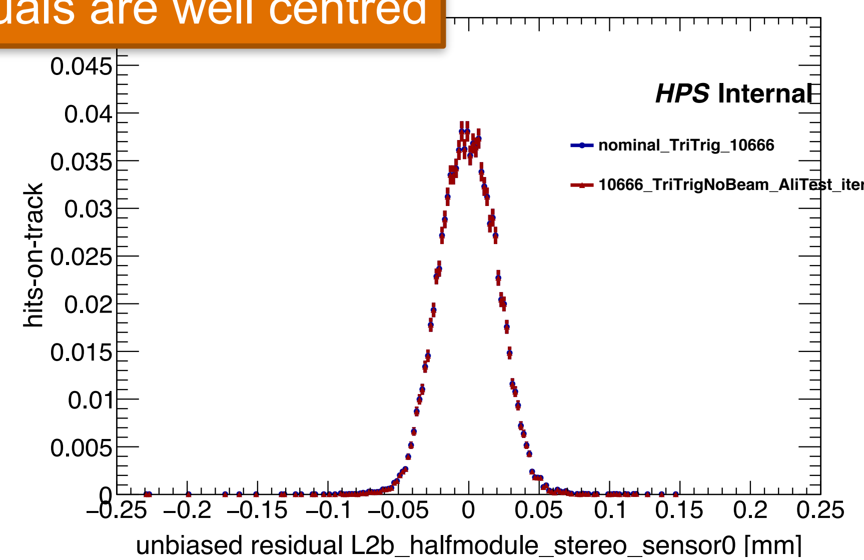




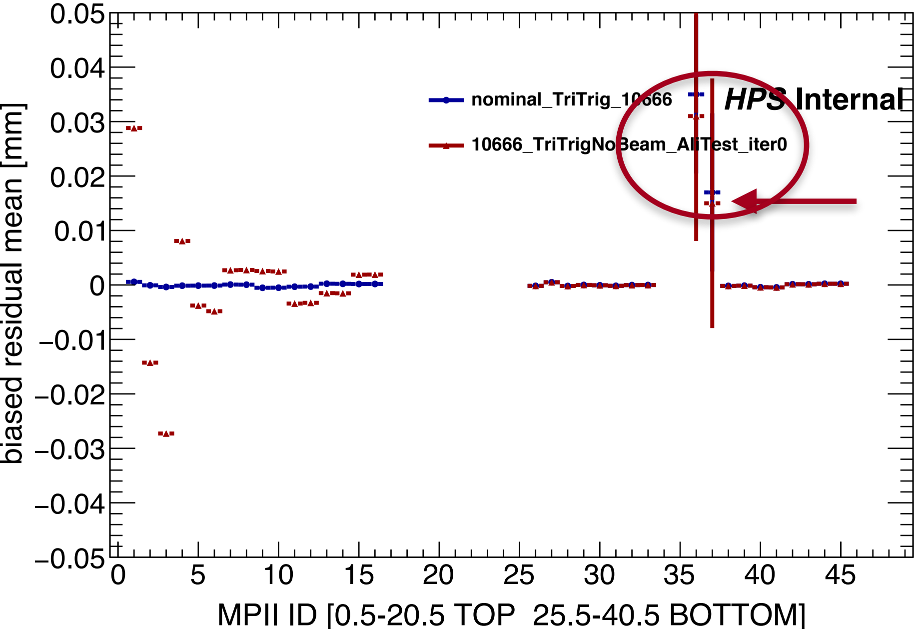
Outer layers are affected too by a single sensor misalignment

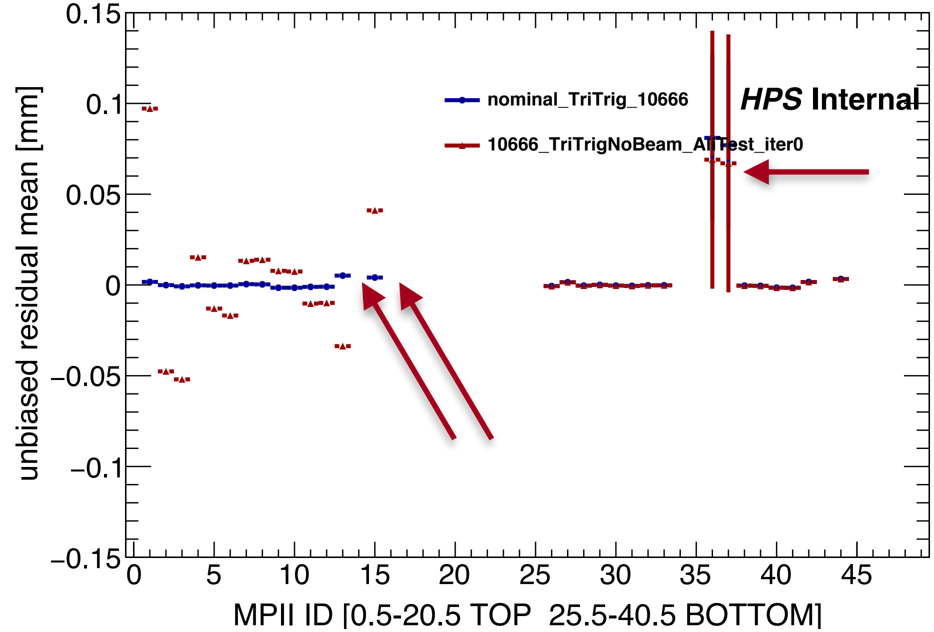
Crosscheck that bottom residuals are well centred





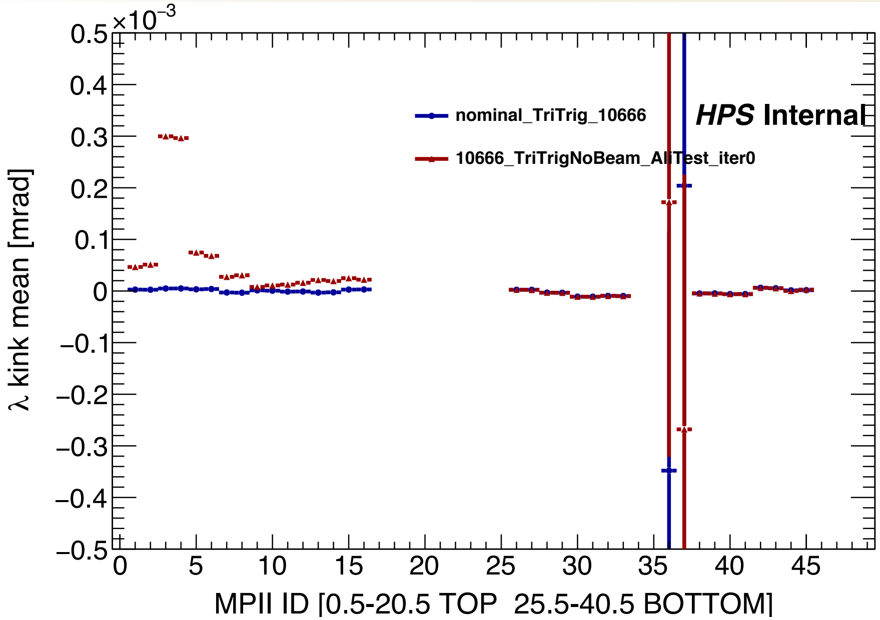
Local Alignment Tests - Summary plots

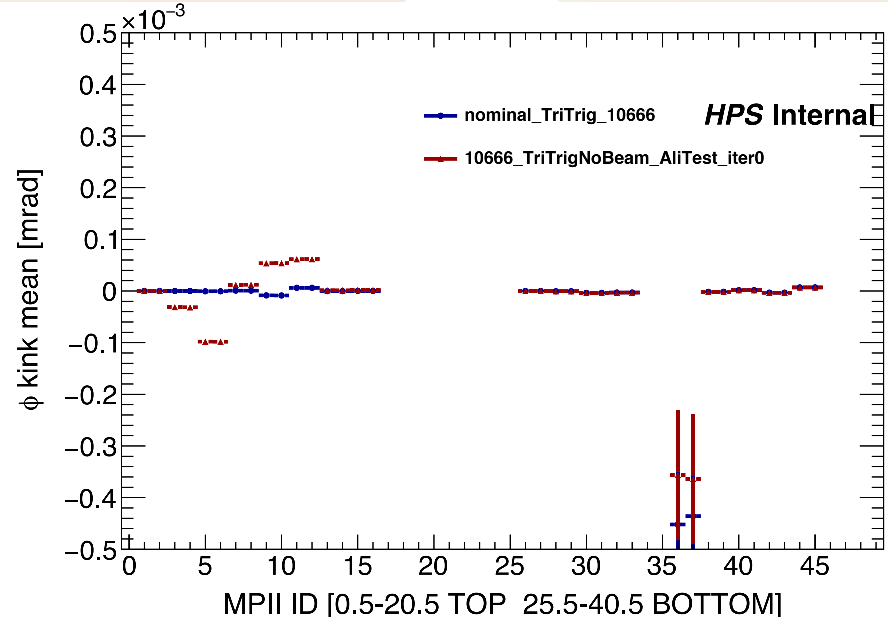




- The plot shows the mean of the biased (left) and unbiased (right) residuals for each MPII ID. Each volume has 20 sensors. First 20 for top, then 20 for bottom
- The misalignments affect all hit-on-track in the top volume. **To be checked in MC**:
  - Why L6t stereo hole and L6t stereo slot (14-16) are dropped in unbiased residuals
  - Reason for largely displaced residuals on L5b axial slot and L5b stereo slot
- Plot indicates how such simple misalignment propagates through the whole volume.

Local Alignment Tests - Summary plots Residuals



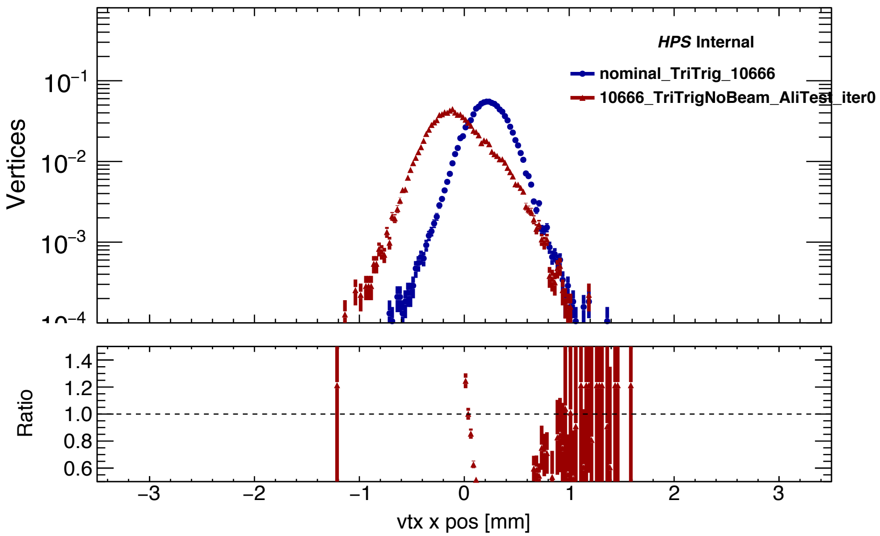


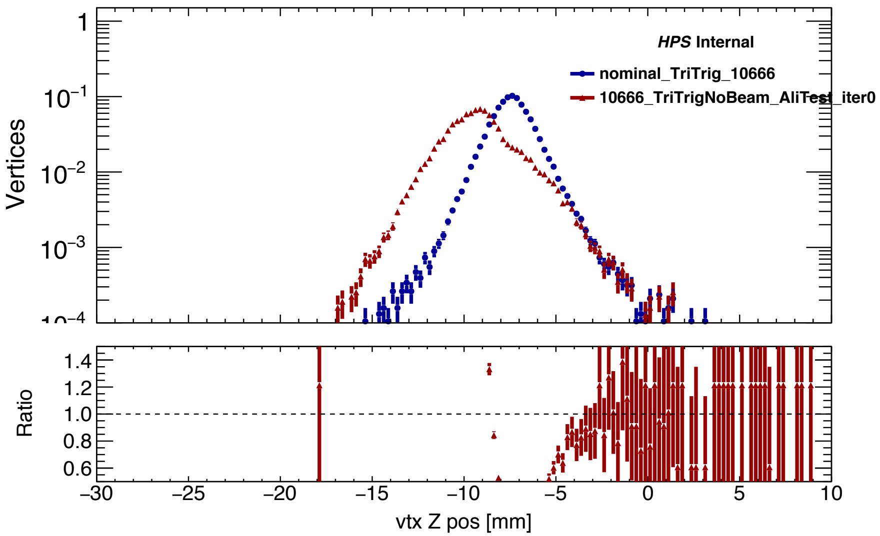
- The plot shows the mean of the λ (left) and ϕ (right) kinks for each MPII ID. Each volume has 20 sensors. First 20 for top, second 20 for bottom
- The misalignments mostly affect all λ kinks in the top volume.

  To be checked in MC:

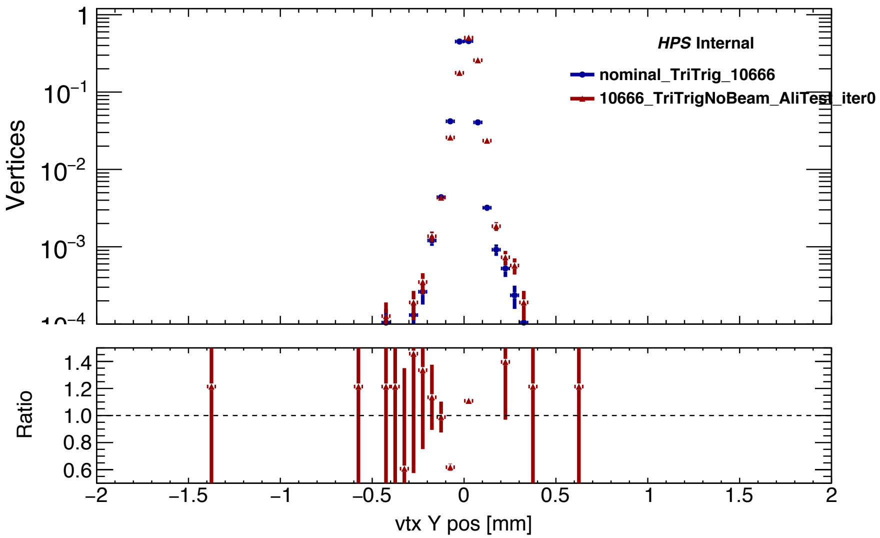
  - Reason for largely displaced kinks on L5b axial slot and L5b stereo slot.
- Plot indicates how such simple misalignment propagates through the whole volume.

Local Alignment Tests - Summary plots Residuals





- Several track quantities are affected
- Finally, such simple (but large) misalignment has large effect on the vtx location and width.



Local Alignment Tests - MPII steering settings and results

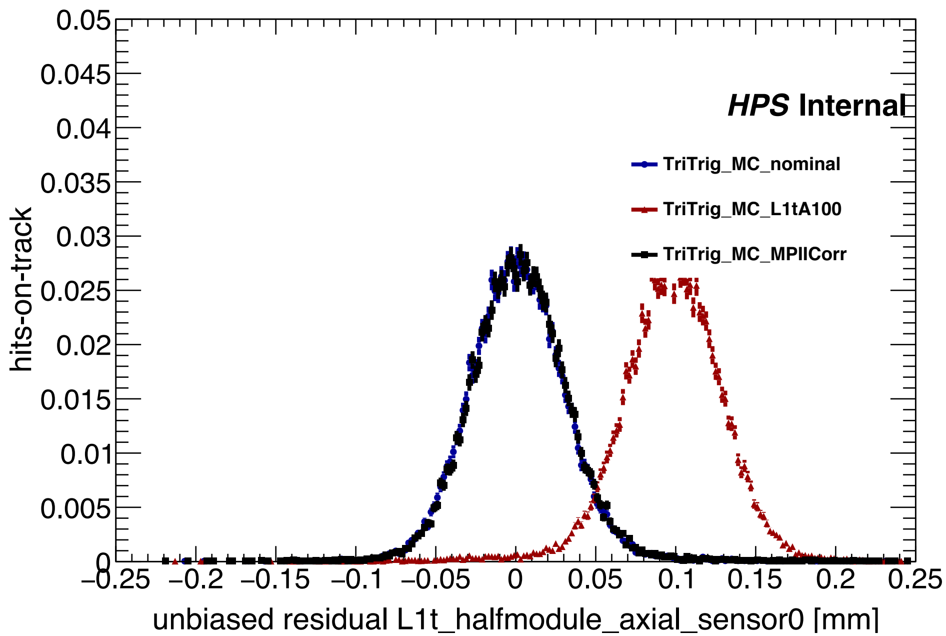
- **Re-alignment strategy:**
  - Fix all outer layers
  - Re-align both L1 top Axial and Stereo
- This is to check if MPII:
  - can recognise displacements on single side when two sides are realigned
  - can recover the same degree of misalignment
- Results are **OK**:
  => MPII finds that L1At is moved of 100um with sub-micron precision
  => MPII finds that L1St is moved of 0.8um with sub-micron precision
- **Simple translations on the new thin sensors can be recovered**

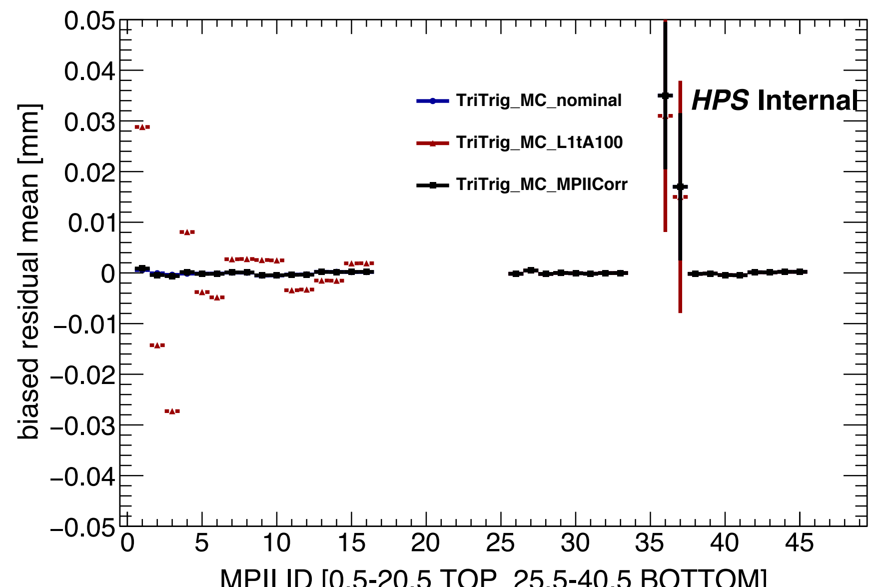
Single L1At movement

```
Solution algorithm:
=======================
                   solution method:    diagonalization
       convergence limit at Delta F=   0.10E+01
        maximum number of iterations=  6
                   using pre-sigmas:   no
                     regularization:   no
   Chi square cut equiv 3 st.dev applied
      ... in first iteration with factor   0.30E+02
     ... in second iteration with factor   0.60E+01
    (reduced by sqrt in next iterations)
Checking feasibility of parameters:
   parameters are feasible  (i.e. satisfy constraints)
```

```
PTLINE: INFO=1  convergence reached
```

```
Parameter    ! first 3 elements per line are significant (if used as input)
     11101   0.99313E-01   0.0000      -0.68652E-03   0.19138E-03
     11102   0.87519E-03   0.0000       0.50875E-01   0.18948E-03
     11103    0.0000      -1.0000
     11104    0.0000      -1.0000
     11105    0.0000      -1.0000
```





Local Alignment Tests - MPII steering settings and results - "multiple correlated misalignment"

SLAC

- **I also tested global movements**
  - L1tL2t A+S = +100um
  - L1bL2b A+S = +500um and was able to re-align correctly [however this assumes the rest is correctly placed]

```
        <millepede_constant name="11101" value="0.100"/>
        <millepede_constant name="11102" value="0.100"/>
        <millepede_constant name="11103" value="0.100"/>
        <millepede_constant name="11104" value="0.100"/>
        <millepede_constant name="11105" value="0.0"/>
        <millepede_constant name="11106" value="0.0"/>
        <millepede_constant name="11107" value="0.0"/>
        <millepede_constant name="11108" value="0.0"/>
        <millepede_constant name="11109" value="0.0"/>
        <millepede_constant name="11110" value="0.0"/>
        <millepede_constant name="11111" value="0.0"/>
-UUU:----F1  compact.xml    30% L199   (nXML Valid) ------
        <millepede_constant name="21101" value="0.50"/>
        <millepede_constant name="21102" value="0.50"/>
        <millepede_constant name="21103" value="0.50"/>
        <millepede_constant name="21104" value="0.50"/>
        <millepede_constant name="21105" value="0.0"/>
        <millepede_constant name="21106" value="0.0"/>
        <millepede_constant name="21107" value="0.0"/>
```

Compact entries for global movement

```
Parameter    ! first 3 elements per line are significant (if used as input)
    11101     0.10191        0.0000       0.10191        0.48554E-03
    11102     0.10095        0.0000       0.10095        0.49947E-03
    11103     0.10052        0.0000       0.10052        0.32385E-03
    11104     0.10053        0.0000       0.10053        0.33074E-03
    11105      0.0000       -1.0000
    11106      0.0000       -1.0000
    11107      0.0000       -1.0000
    11108      0.0000       -1.0000
    11109      0.0000       -1.0000
    11110      0.0000       -1.0000
-UU-:----F1  millepede--L1_L2_MC_GlobMov.res   Top L1     (Fundamental) ----
    12316      0.0000       -1.0000
    12317      0.0000       -1.0000
    12318      0.0000       -1.0000
    12319      0.0000       -1.0000
    12320      0.0000       -1.0000
    21101     0.50224        0.0000       0.50224        0.42774E-03
    21102     0.50398        0.0000       0.50398        0.41657E-03
    21103     0.50071        0.0000       0.50071        0.28678E-03
    21104     0.50135        0.0000       0.50135        0.27921E-03
```

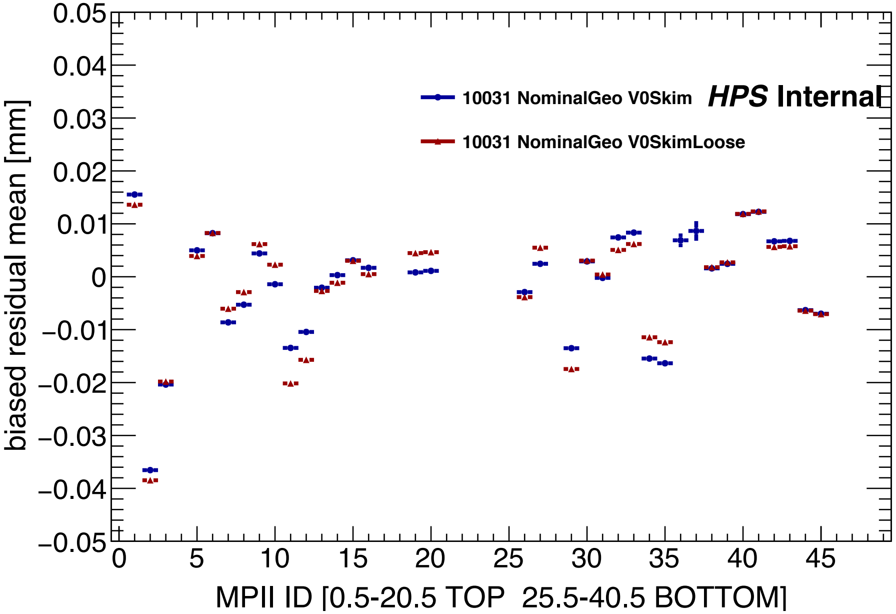
MPII residuals after accumulation and solving

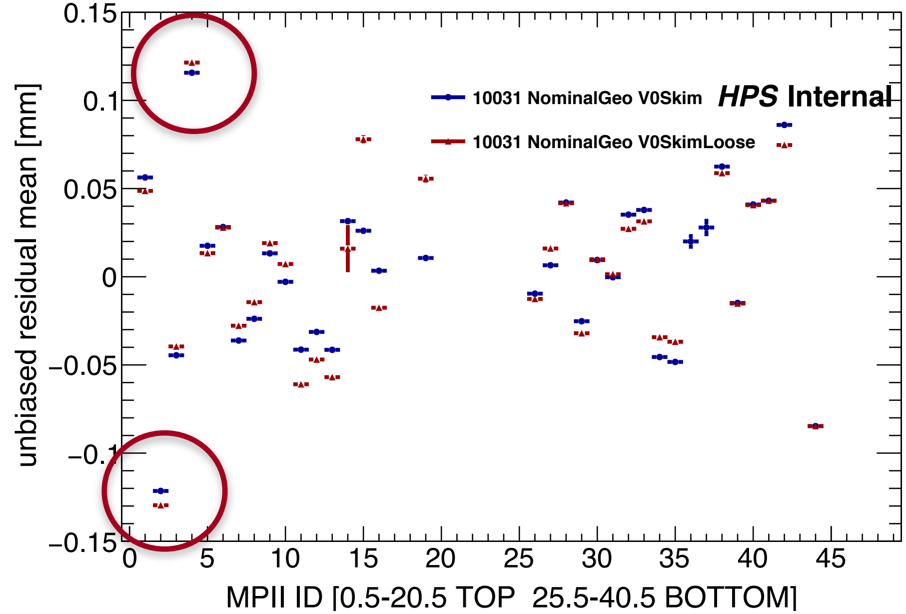
Local Alignment Tests - Bottom line

- Using MC Trident has been shown that MPII is able to recover u-translations of the new thin sensors.
- I also tested global movements
  - L1tL2t A+S = +100um
  - L1bL2b A+S = +500um and was able to re-align correctly **[however this assumes the rest is correctly placed]**
- **Rotations still under investigations [partial derivatives seem to be correct in the code]**
- **- Work ongoing to automatise the procedure to post (human readable) results to www.slac.stanford.edu/~pbutti/alignment/**

Local Alignment Tests - Checks on 2019 Data

- After validation of MPII alignment on MC samples first checks on data were performed
- Checked 10031 both V0Skims (≥1 V0 + cluster on tracks) and V0SkimsLoose (≥1 V0, All tracks) unbiased residuals
- Initial situation (nominal alignment with survey constants in, **no global $\Delta\theta_y$ correction**) shows large residuals for L1/ L2



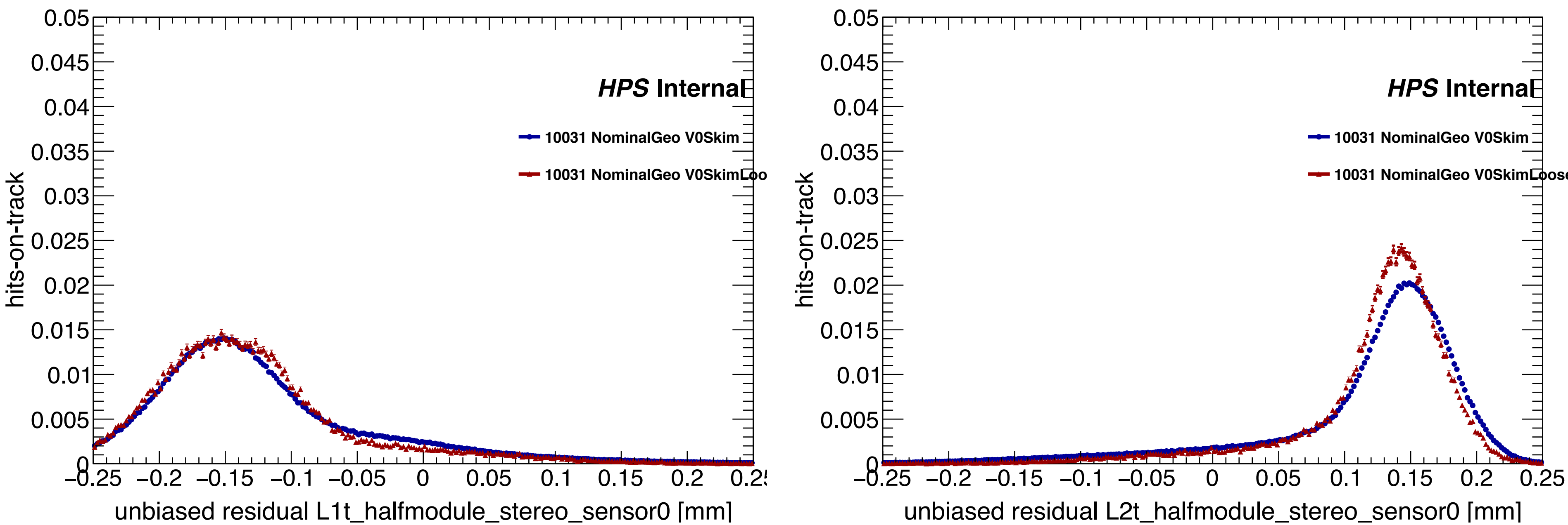


From unbiased residual checks seems like large L1tS / L2tS displacements might be present

Full plots available: v0skim vs v0skimLoose

Local Alignment Tests - Checks on 2019 Data

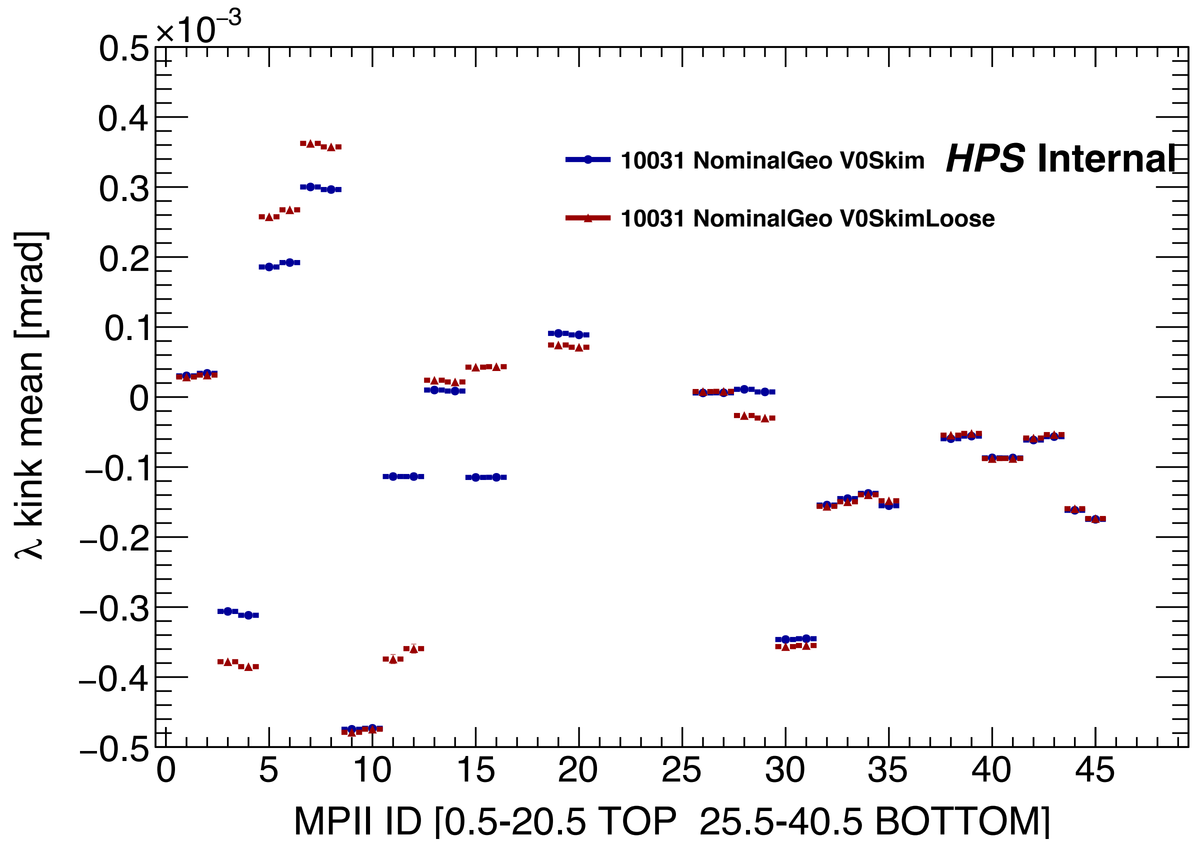
- After validation of MPII alignment on MC samples first checks on data were performed
- Checked 10031 both V0Skims (≥1 V0 + cluster on tracks) and V0SkimsLoose (≥1 V0, All tracks) unbiased residuals
- Initial situation (nominal alignment with survey constants in, **no global $\Delta\theta_y$ correction**) shows large residuals for L1/ L2



Full plots available: v0skim vs v0skimLoose

Local Alignment Tests - Checks on 2019 Data

- Large λ kinks in the innermost layers might point to large internal misalignments of these sensors as GBL might try to increase the angle at each surface to pick the next point => sensitive to u movements



Local Alignment Test - First internal alignment pass

- **For the first tests on 2019 Alignments decided to try to align the innermost sensors**
- **They are the most important for vertex resolution and they were the mounted in the detector during the upgrade [even if other sensors might have moved]**
- **For the first tests I use the V0Skims [faster processing, similar results wrt V0SkimsLoose]**
  - At least 1 vertex in the V0 collection
  - Tracks associated to one ECAL cluster
- Track selection:

  - **p > 1 GeV**: reduce Multiple coulomb scattering contribution

  - **>= 6 3D hits** on tracks: stronger constraints from rest of the detector

  - **X2<50**: better track quality
- **Strategy for the test**:

  - Assumed most of misalignment present on L1/L2 (new thin sensors)

  - Fixed L3->L7 for reference and constraint against global movements (won't try to free them)

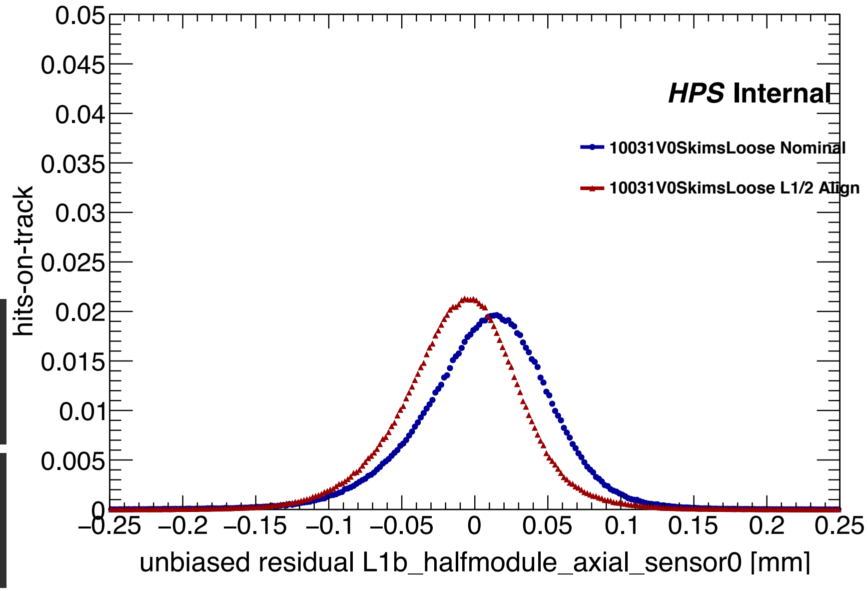
  - Only aligned L1 and L2 top/bottom translations along sensitive direction (tu)

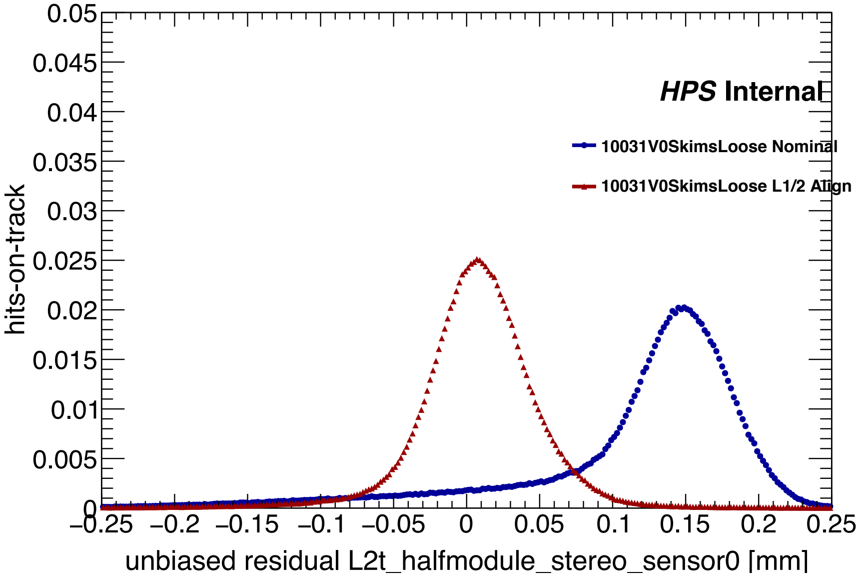
Local Alignment Test - Results

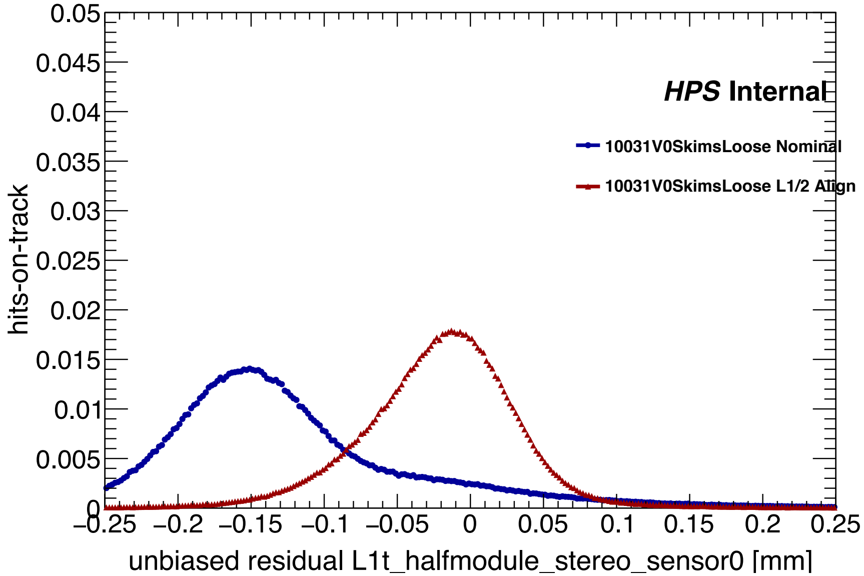
- **3 Alignment iterations [actually 2 for top 1 for bottom]:**
  1) L1t_tu L2t_tu
  2) L1t_tu L2t_tu
  3) L1b_tu L2b_tu
- Corrections show convergence [checked at 4th iteration but not included]

```
<!-- top half-module translations -->
<millepede_constant name="11101" value="0.0 + -0.061797 + 0.011437" />
<millepede_constant name="11102" value="0.0 + -0.037138 + 0.011789" />
<millepede_constant name="11103" value="0.0 + -0.040444 + 0.017099" />
<millepede_constant name="11104" value="0.0 + -0.129850 + -0.029836" />
```

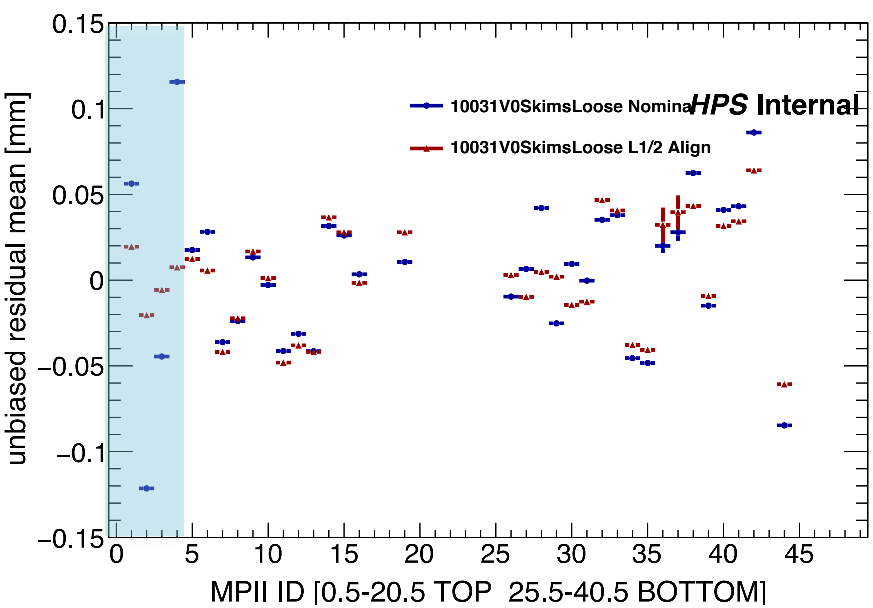
```
<millepede_constant name="21101" value="0.0 + -0.135130" />
<millepede_constant name="21102" value="0.0 + -0.095591" />
<millepede_constant name="21103" value="0.0 + -0.114820" />
<millepede_constant name="21104" value="0.0 + -0.036847" />
```

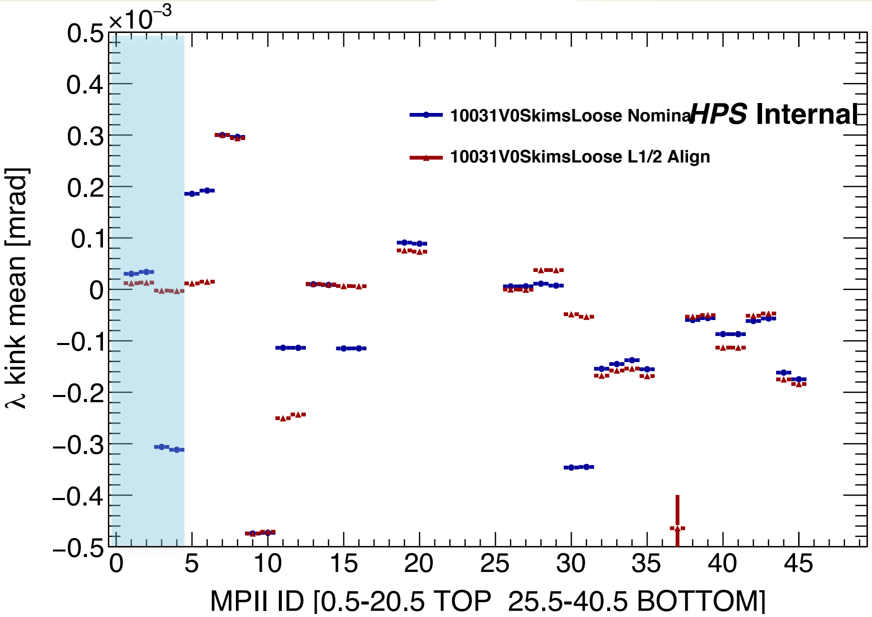






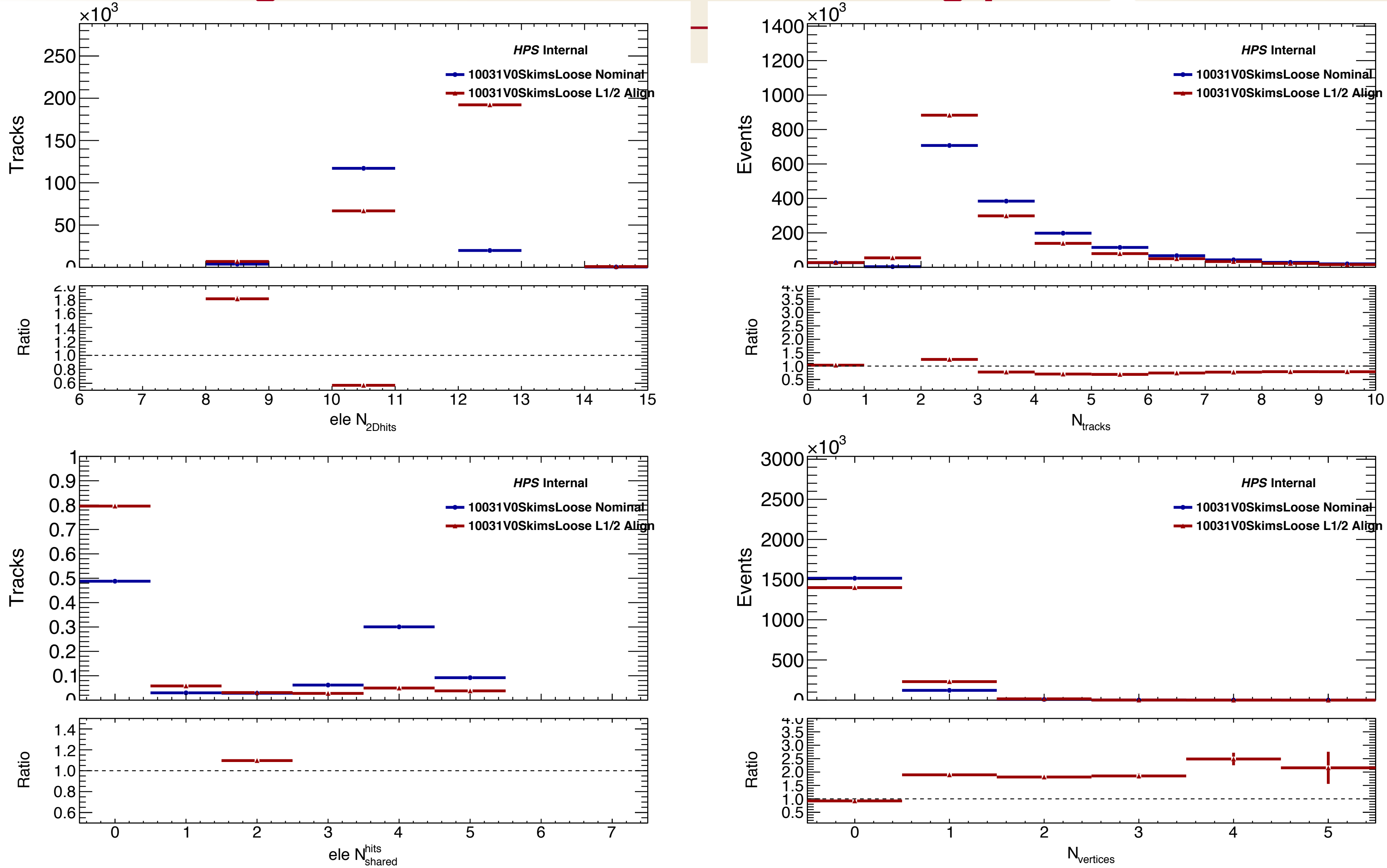
Local Alignment Test - Results residuals/kinks





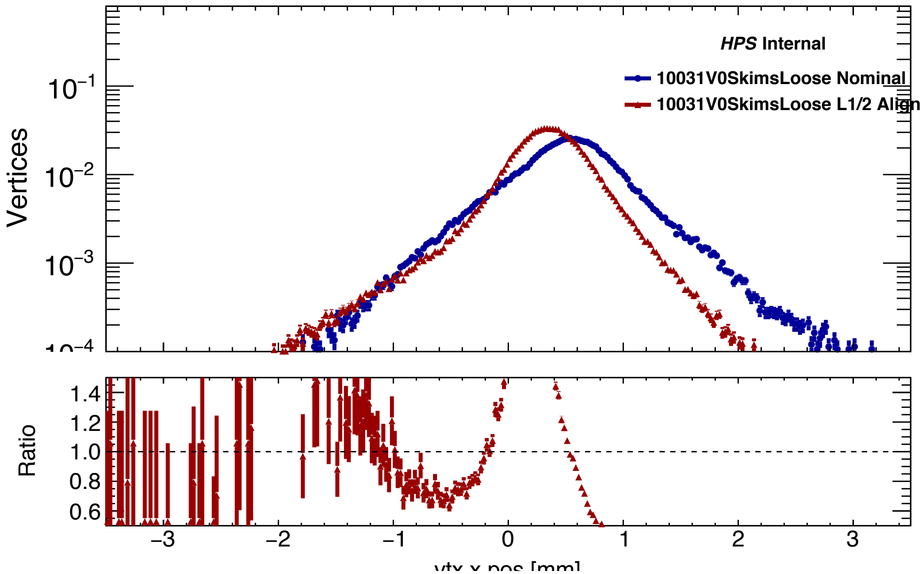
- Shaded area shows the aligned sensors.
- Reduction of residual biases and next-layer lambda kinks.
- Solution leads to small changes in the outer sensors.

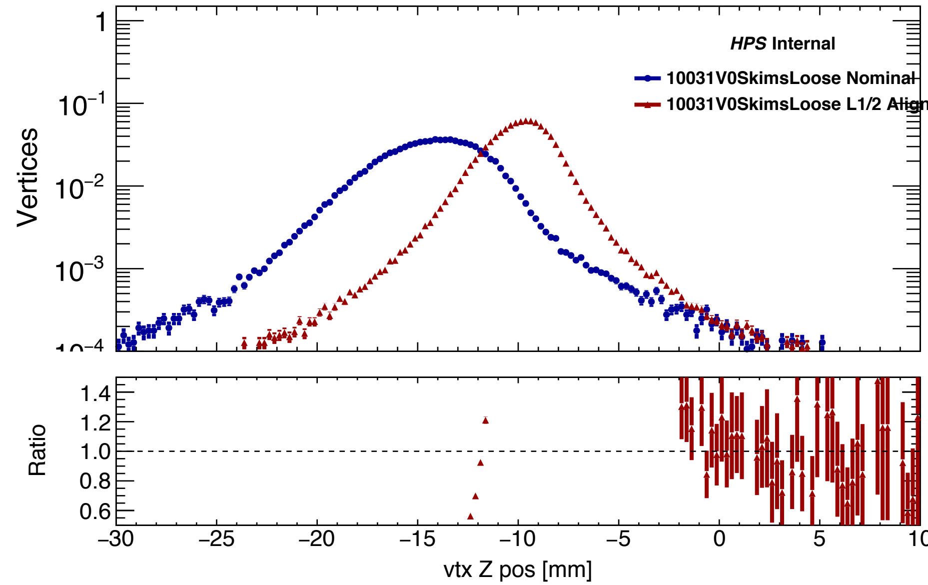
Local Alignment Test - Results tracking performance

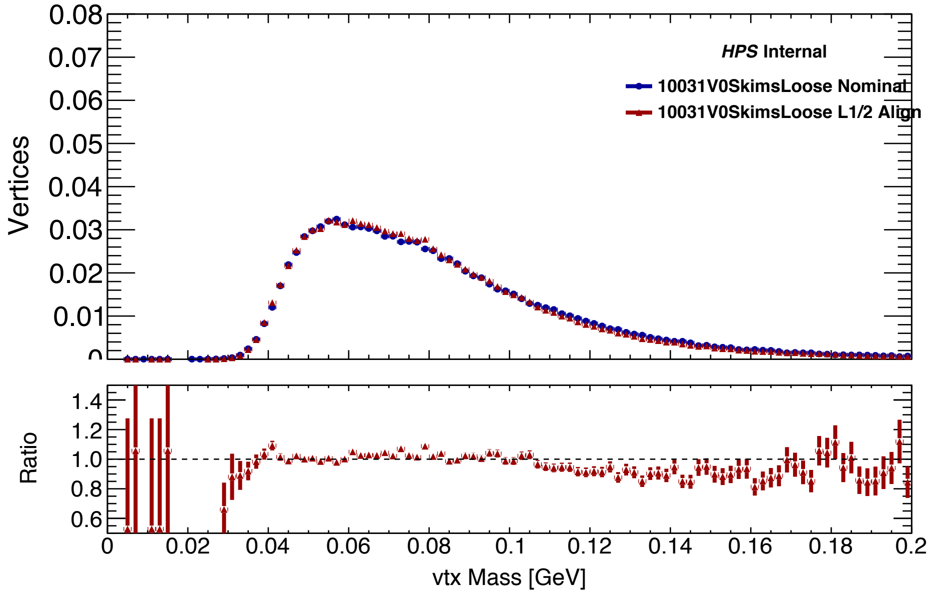


More hits on track, less shared hits, more cleaner tracks, more vertices

Local Alignment Test - Results tracking performance



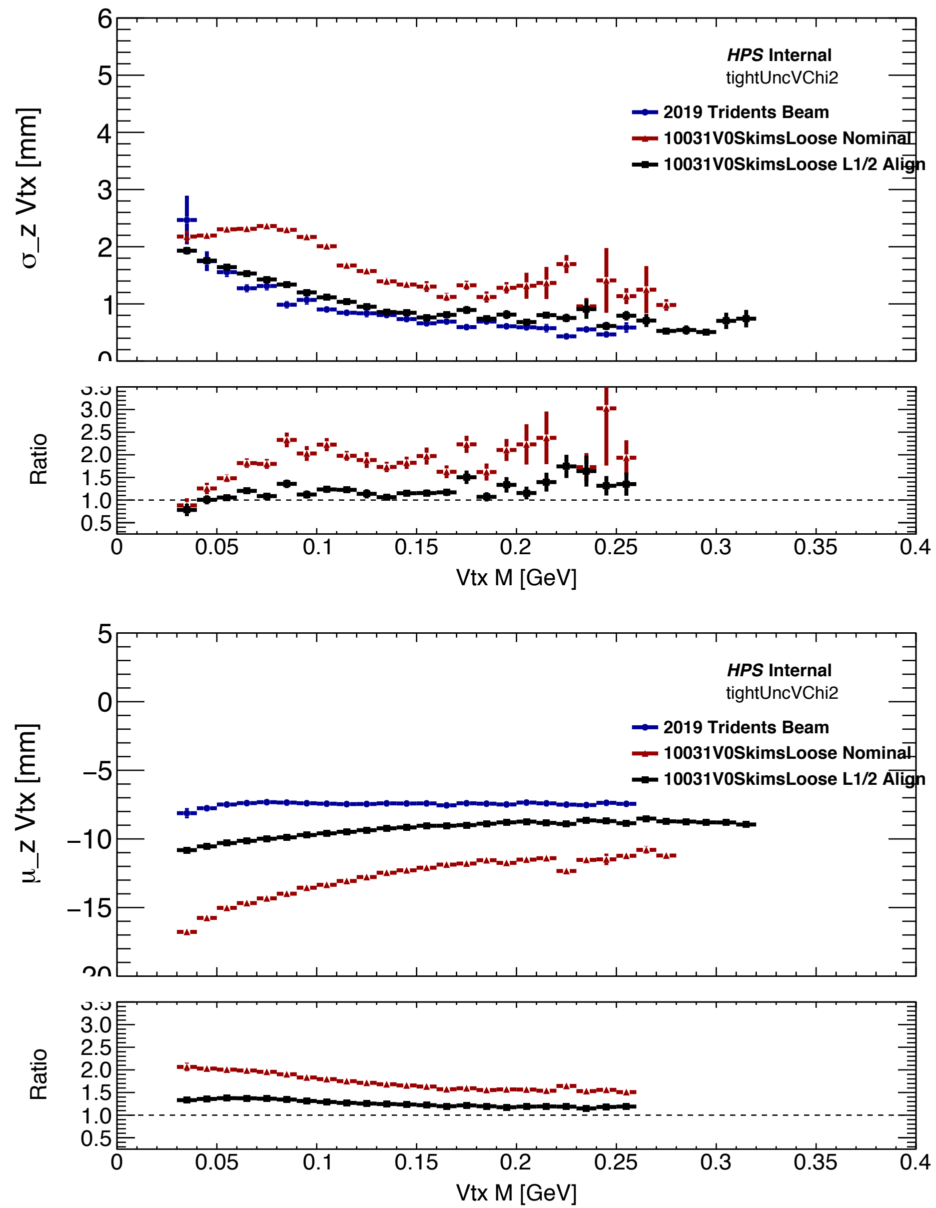




Reduction of core spread and reduction of the tail fraction.
No changes on the shape of the Invariant Mass of unconstrained vertices.
More plots available: L1L2AlignmentPlots

Current performance aiming for Jeopardy

- The Unconstrained Vtx resolution is extracted by requiring:
  - el mom < 3.4 GeV
  - el / pos mom > 0.6 GeV
  - >=4 hits on track
  - pos has L1 hit
  - Unc V Chi2 < 10
- **Still looser than 2016 Tight Signal Region.**
- Flatter trend of the vtx position wrt Inv Mass, moving toward **assumed z position of the vertex [-7.5/-7.8 mm Preliminary!]**, but still far and proceeding cautiously.
- **Lot more work** is needed to get to a proper alignment, but first results sensibly improve resolution



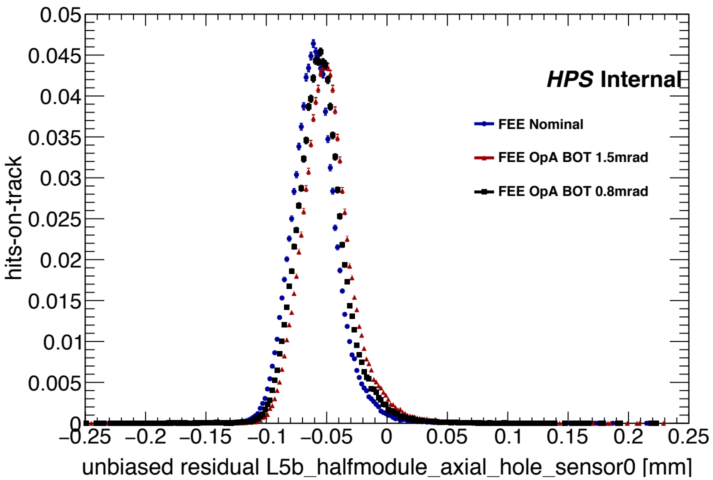
Summary and next steps

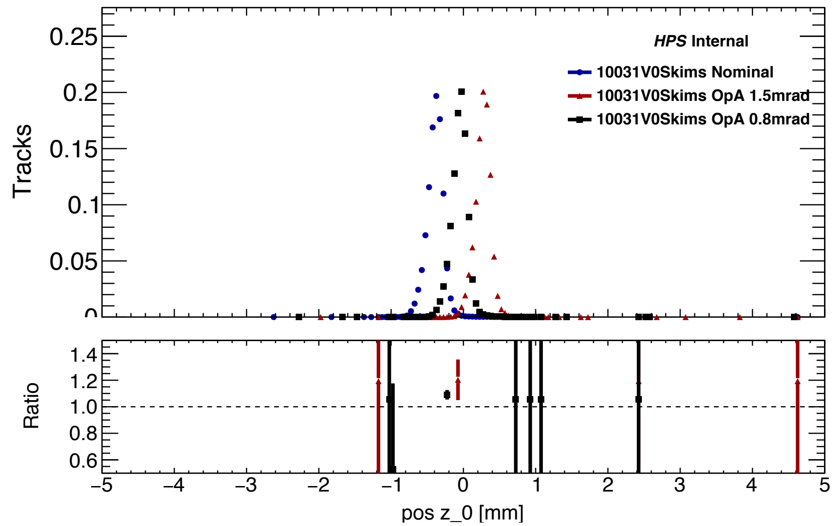
- **Has been shown that different alignment scenarios can bring to improvements in vertex resolution plots for initial assessment of detector performance vs MC simulation.**
  - Harder in 2019 to converge to correct solution without clear grasp on external constraints
  - Although procedures are getting in place and shaping up, lot of work is still needed.
- Checked full chain on two different alignment configurations:
  - **Global alignment test**: correction of the bottom opening angle
    - Removal of VtxZ_u(InvM) by construction
    - Consistent improvement in vertex z position resolution
    - No effects on improved tracking efficiency and track quality
  - **Local alignment:** correction of L1/L2 tu
    - Improvement of VtxZ_u(InvM)
    - Consistent improvement in vertex z position resolution
    - Improved hit-on-track association, track-quality, reduced number of shared hits.
- **Work is ongoing to test combination of both procedures to have global+internal alignment**
- **To fulfil Jeopardy we will be able to provide**
  - Improved track parameters and vertex position resolution in short time scale and comparison to MC simulation with perfect geometry
- **Long time scale**:
  - Review of the geometry survey, constrained alignment and search for external handles for remove alignment weak modes (survey data, resonances, …)

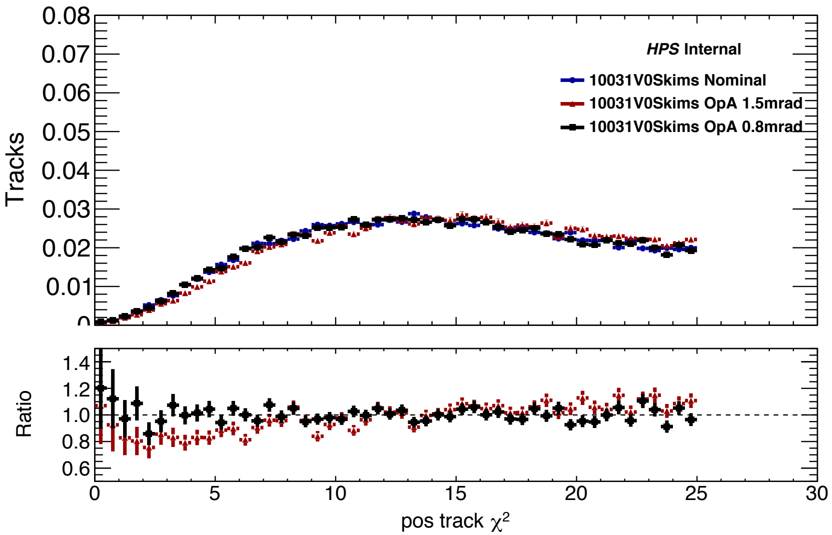
BACKUP

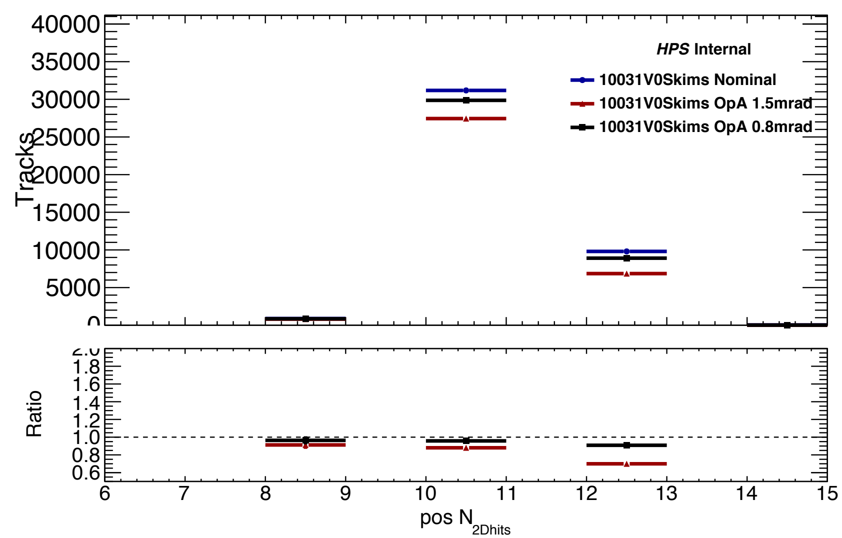
Global Alignment - Vertex z vs InvMass

- Small effect on track quality, improvement of track parameters, residuals…





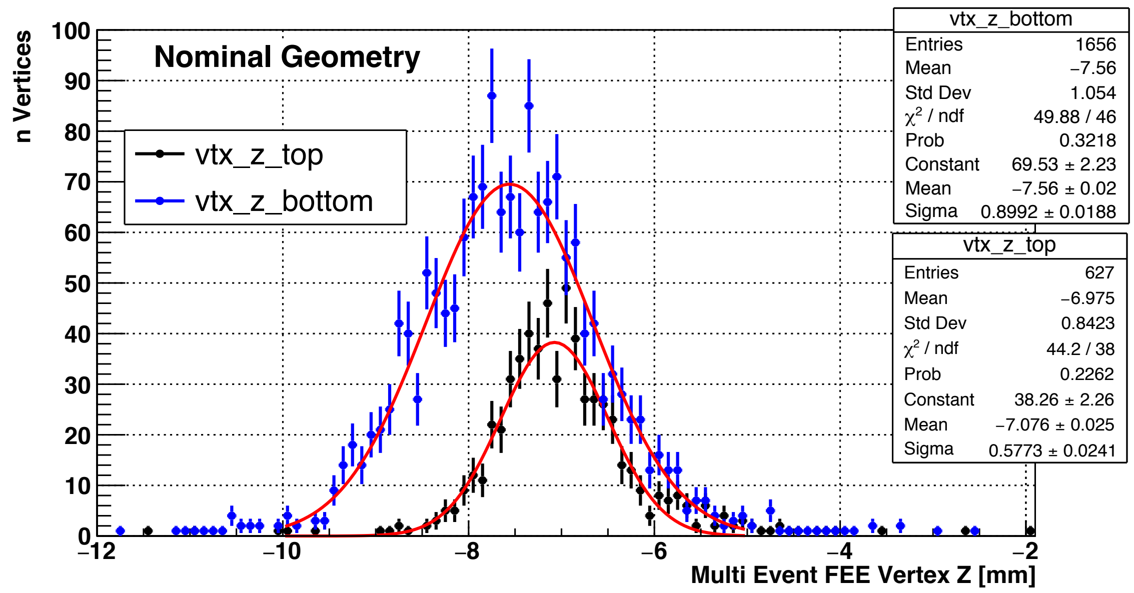


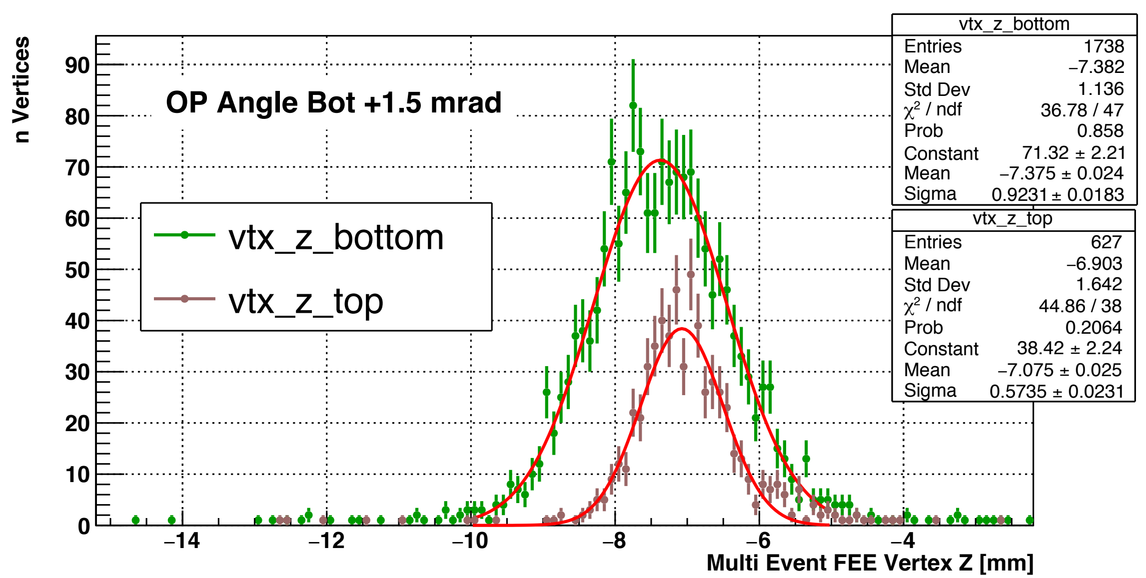


Global Alignment - Multi events FEE vertexing

- The Multi Vtx Fitter has been updated to accept more than 2 tracks per event (see iss687_dev MultiVtxer).
- Clear effect on the x-y position resolution wrt 2-tracks vertices, as the vtx resolution improves with number of tracks.
- Events are collected, vertices are fitted in 100 tracks chunks, or less if not available: i.e. if 150 tracks are found 2 vertices are formed with 100 and 50 tracks, respectively.
- Will use the updated version for alignment studies/monitoring as leads to much clearer visualisation in the following studies
- This method shows a discrepancy between top and bottom z vertex position of ~500um: will have to investigate
- An opening angle of 1.5mrad implies a movement of ~200um on the vertex Z position. $\Delta Z \sim R(1-\cos\theta_y) \sim \frac{1}{2}R\theta_y^2$

  where R is the radius from the pivot to the vertex location, while $\Delta Y \sim R sin\theta_y \sim R\theta_y$

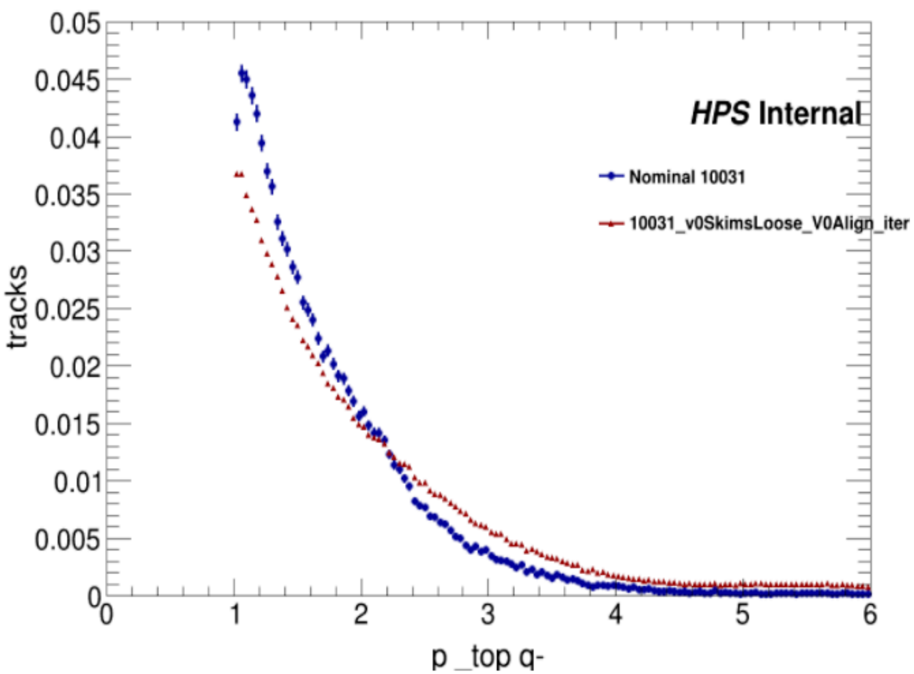


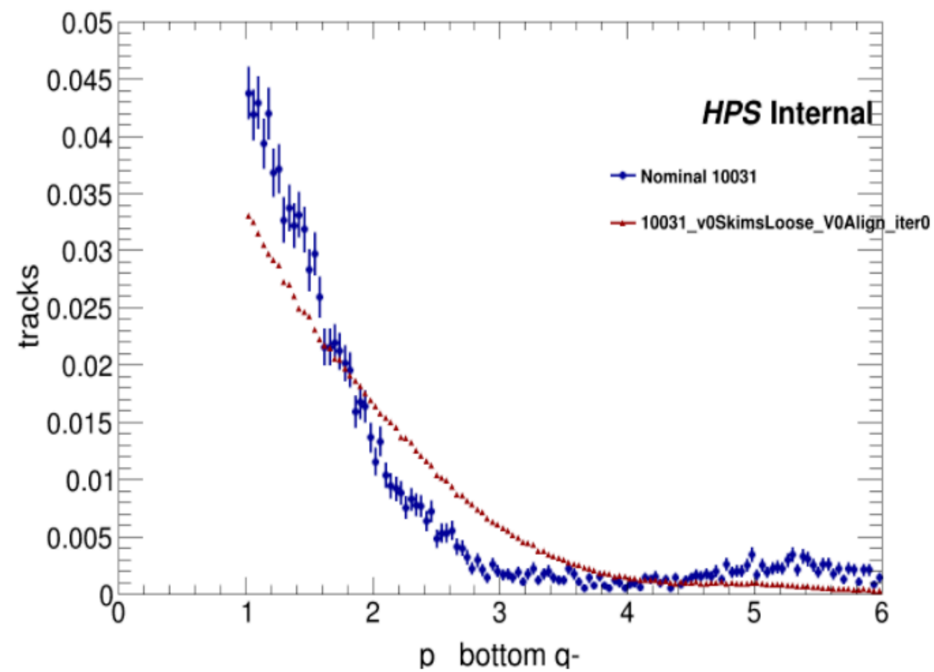


Local Alignment test - Check on data 2019 V0skims 10031

- I've checked the kinematics of the tracks reconstructed in (**V0Skims**) and (**V0SkimsLoose**)
- One issue that has been noticed in the v0skims:
  - **Very low statistics electrons in bottom volume**
- **Probably due to track-to-cluster association algorithm not working properly in 2019 [fundamental!!]**

Observed much lower statistics of electrons in bottom volume in clusters on tracks skims



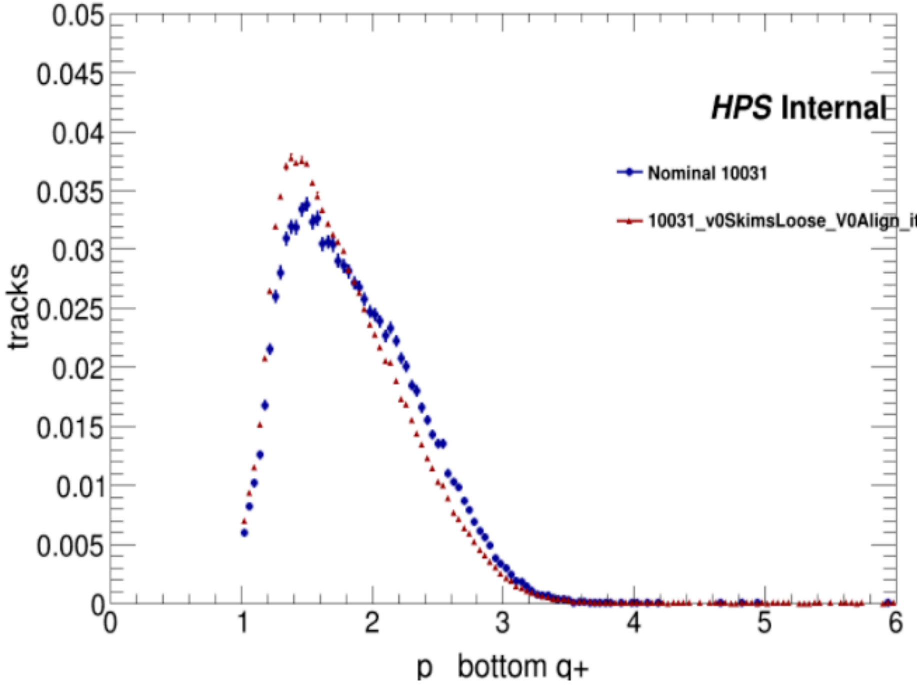


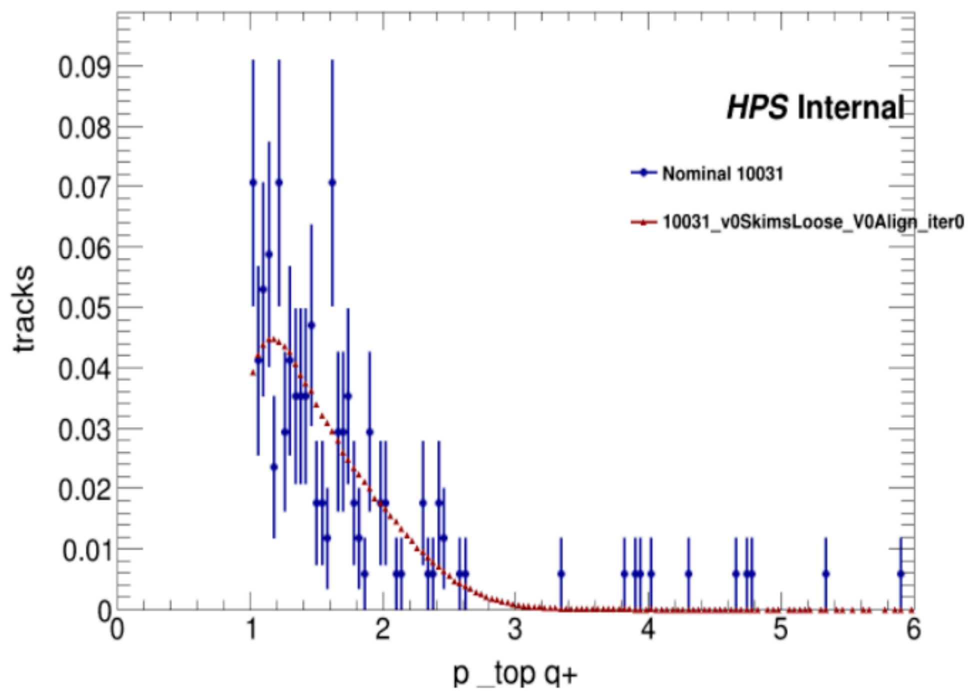
This variable just to show stats. Full plots available: v0skim vs v0skimLoose

Local Alignment test - Check on data 2019 V0skims 10031

- 2019 Data has been skimmed with (**V0Skims**) and without (**V0SkimsLoose**) cluster on track requirement
- One issue that has been noticed in the v0skims:
  - **Very low statistics of positrons in top volume**
- **Probably due to track-to-cluster association algorithm not working properly in 2019 [fundamental!!]**

Observed much lower statistics of positrons in top volume in clusters on tracks skims





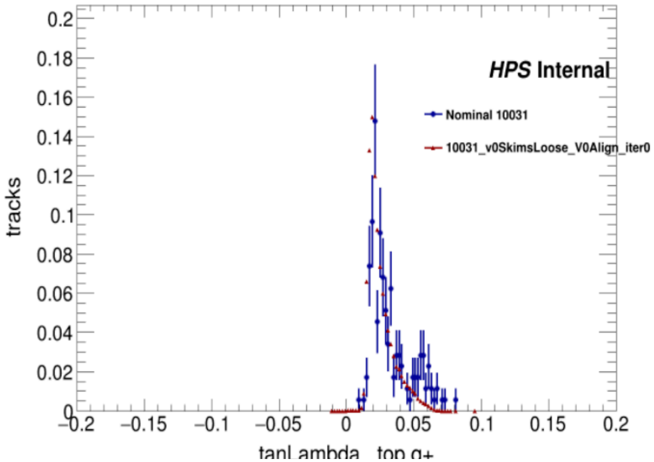
This variable just to show stats. Full plots available: v0skim vs v0skimLoose

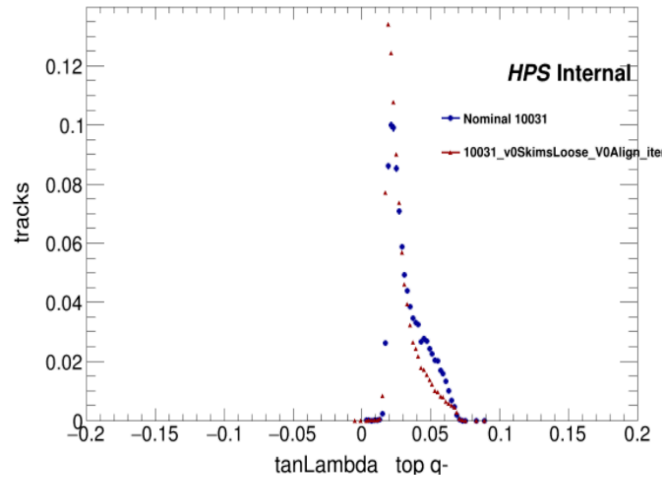
Full plots available: v0skim vs v0skimLoose

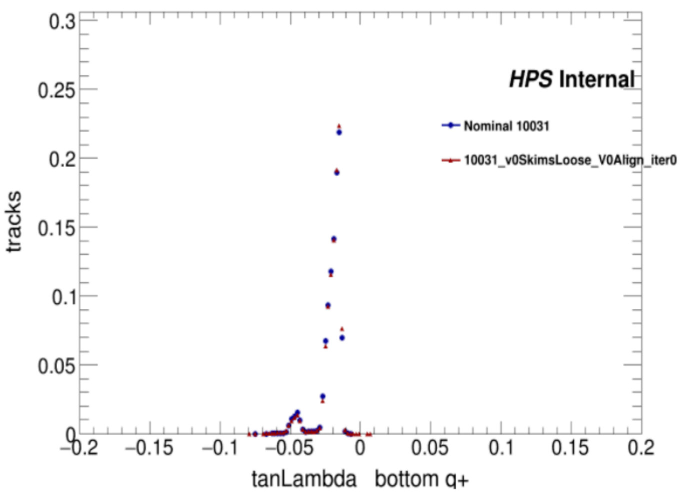
Local Alignment test - Check on data 2019 V0skims 10031

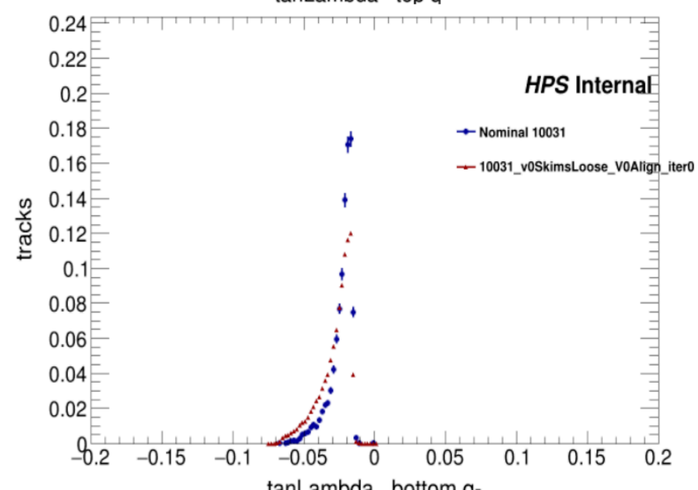
SLAC

- 2019 Data has been skimmed with (**V0Skims**) and without (**V0SkimsLoose**) cluster on track requirement
- One issue that has been noticed in the v0skims:
  - **Very low statistics of positrons in top volume**
  - **Very low statistics electrons in bottom volume**
- **Probably due to track-to-cluster association algorithm not working properly in 2019 [fundamental!!]**









VTX resolution plots selection

- Check over V0 vertices
- The **preselection** is:

  - e P < 3.4 GeV

  - e-/e+ Chi2 < 25

  - e-/e+ P > 0.6 GeV

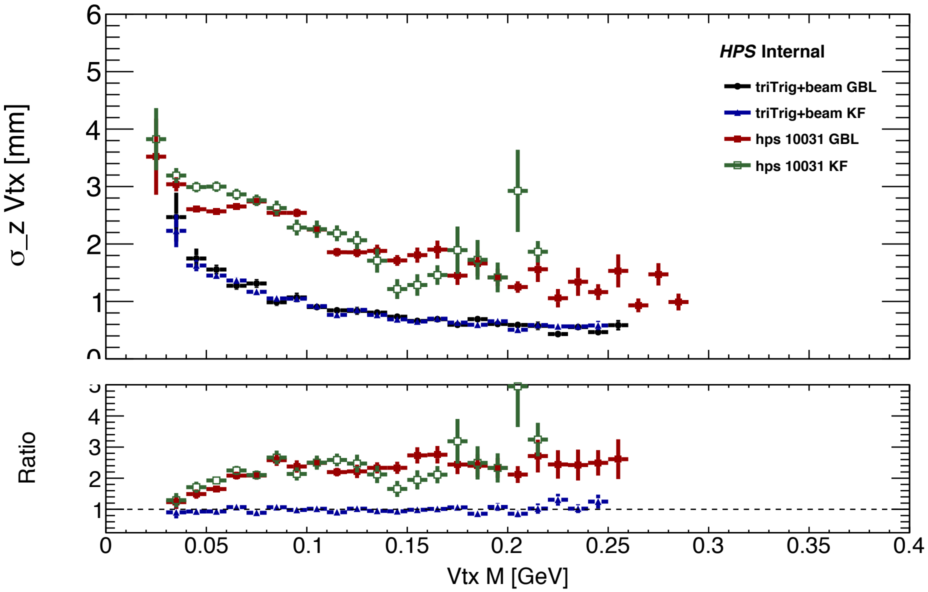
  - 2D hits e-/e+ >= 8

  - e-/e+ NShared < 5 [no effect: MOUSE cuts]

  - Vtx Chi2 < 20
- L1Pos
- **tightUncChi2**:

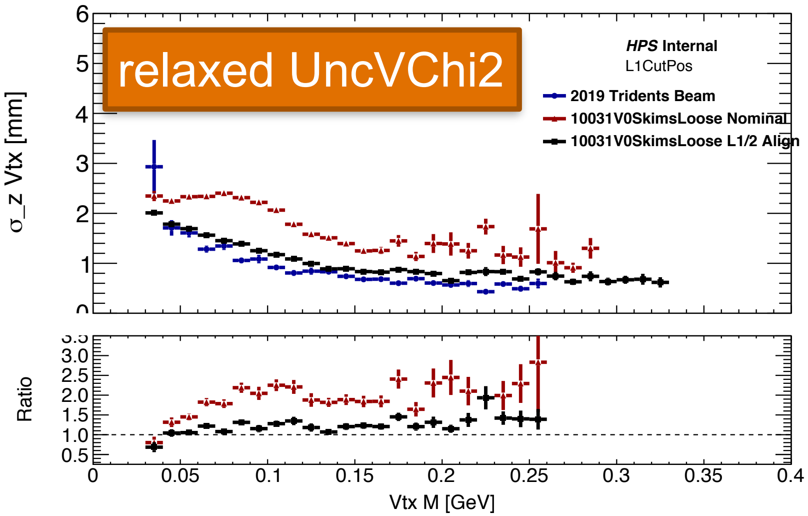
  - UncVChi2 < 10

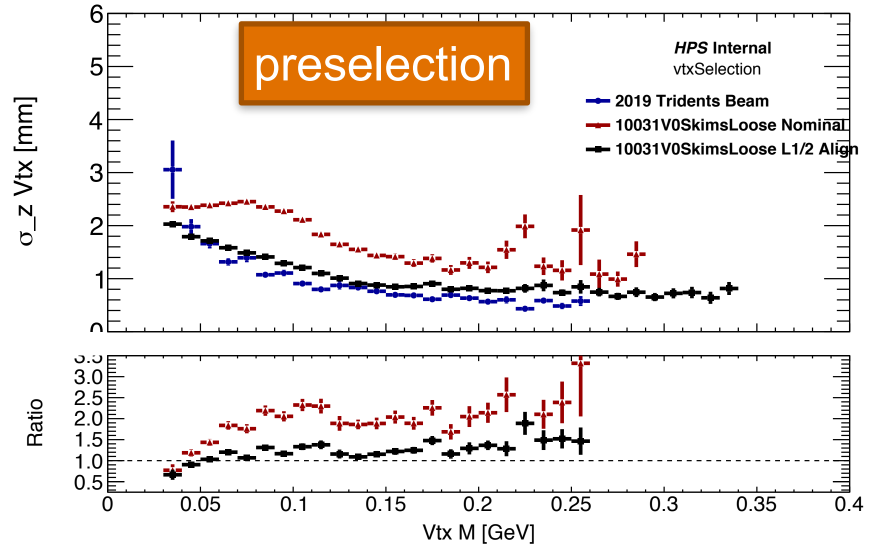
  - L0 Hit on e+ (against WABs)

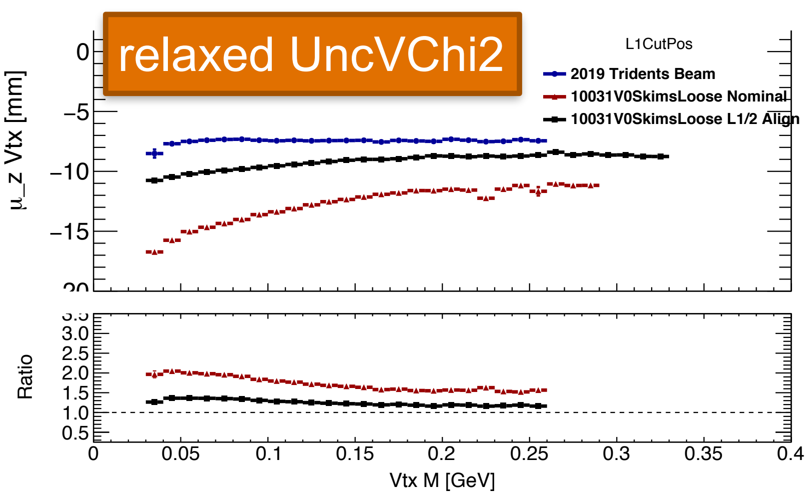


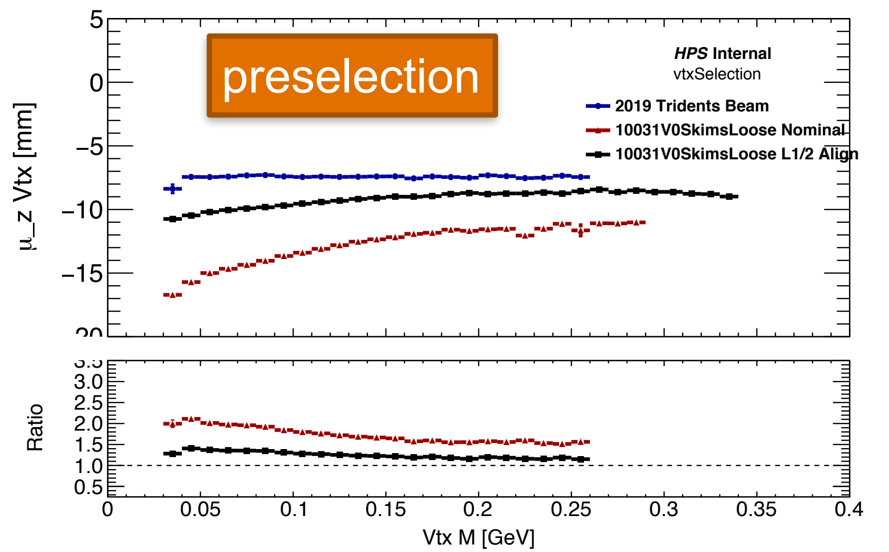
Current performance aiming for Jeopardy

- I also tested looser selections: no major difference observed



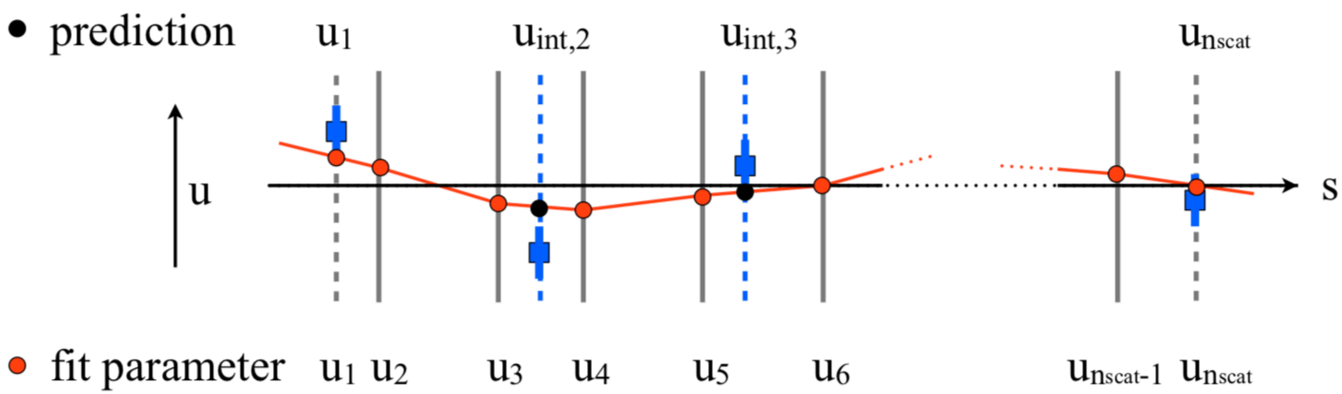






GBL Tracking - Introduction

- General Broken Lines **(GBL) is a track refit algorithm that add the description of multiple scattering to an initial trajectory**
  - Based on propagation in magnetic field & average energy loss
  - Constructed from a sequence of thin scatterers
  - **In the case of silicon detector a scatter also has a measurement (in the form of local residual)**

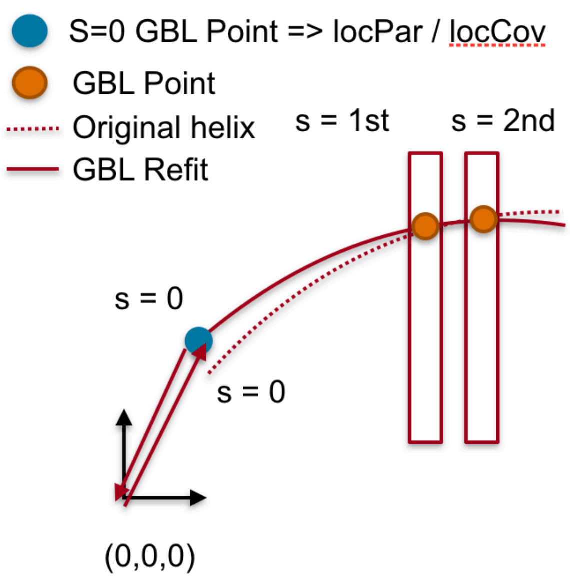


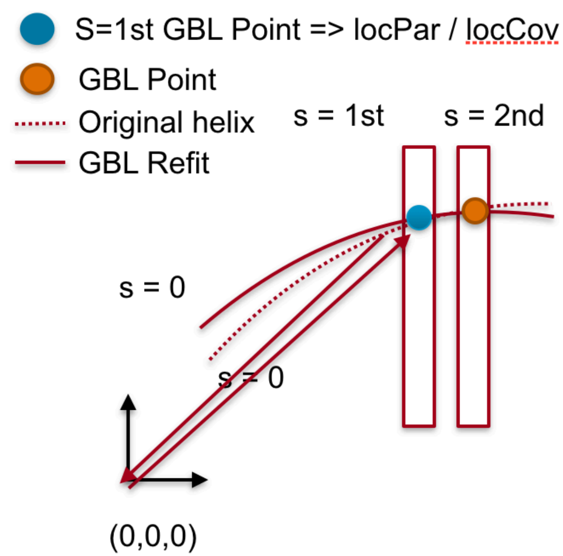
- The initial trajectory should be 'close enough' to the solution and provide a reasonable estimate of the particle trajectory
- GBL is used in hps-java to refit helical track fits
- It is **iterated** (5 iterations) in our code to ensure convergence of the track parameters corrections

GBL Tracking - How corrections are extracted

SLAC

- General Broken Lines provides the track parameters **corrections** and the **full local covariance** matrix at each scatter point
- **An empty scatter point** (no scatter nor measurement) can be used to obtain the corrections to the track parameters at that particular point in space
- This is what is done in our tracking code:
  - **The track parameters with respect to (0,0,0) are obtained from a fictitious GBL point at s=0**
  - The other track states on surface are computed on the hit position on each sensor
- **This implies a the usage of a uniform magnetic field between the first measurement to s=0 point**



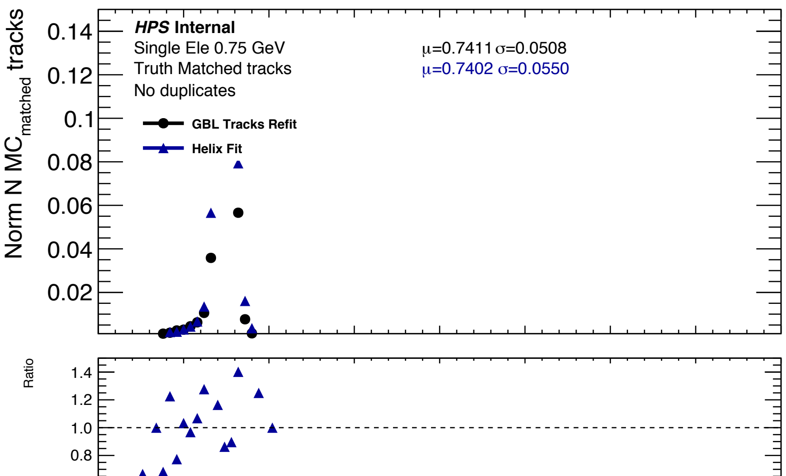


GBL Refit - Track Parameters residuals/pulls

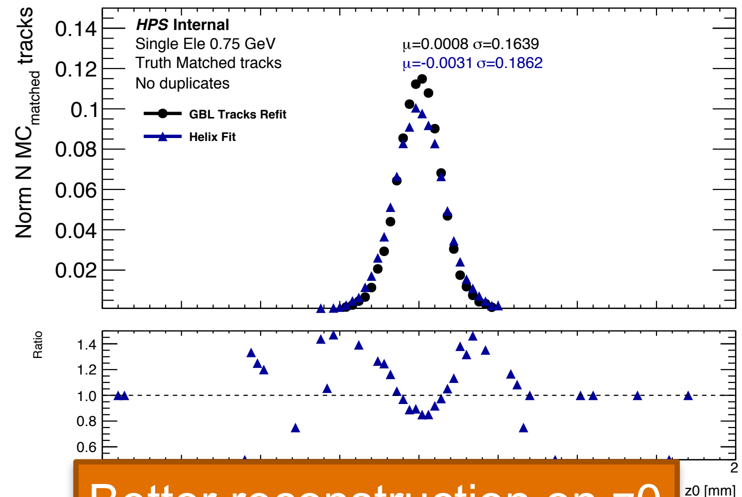
- **Performance of track re-fit is estimated using track parameter residuals and pulls with respect to the matched truth particle**
- Proper estimate of track parameters and their errors is fundamental for vertexing, event reconstruction and eventually analysis.
- Used 2016 Geometry MC (2019 MC readout/reconstruction still work in progress)
- Single electron samples, E=0.75GeV and E=2GeV, perfect detector conditions and alignment. Particles are shot from (0,0,0)
- Last checks presented at a collaboration meeting (I know of) were performed by MattG May2017_Vertexing
- He found **pulls well centered but with errors not properly computed for the linear fit (z, tanLambda)**
- **Last check made in iss154** (Several changes since then in hps-java)

Helix tracks and GBL Refit

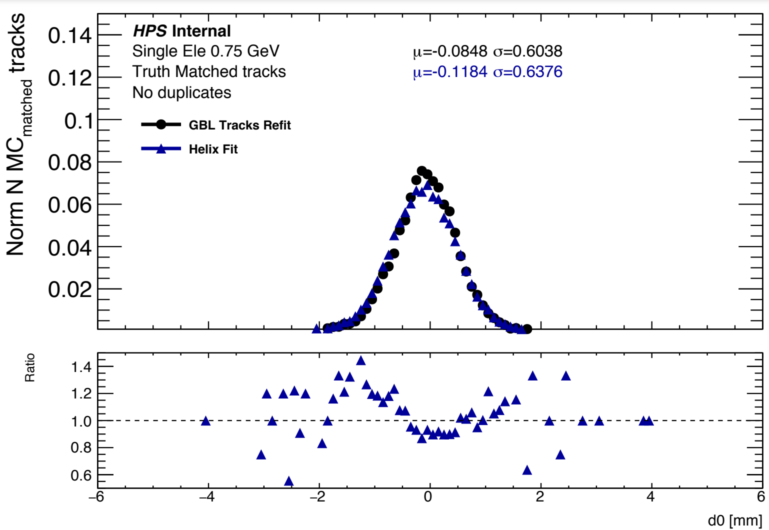
- Helix fits are taken from the GBL Refit relational table. Basic quality cuts are applied

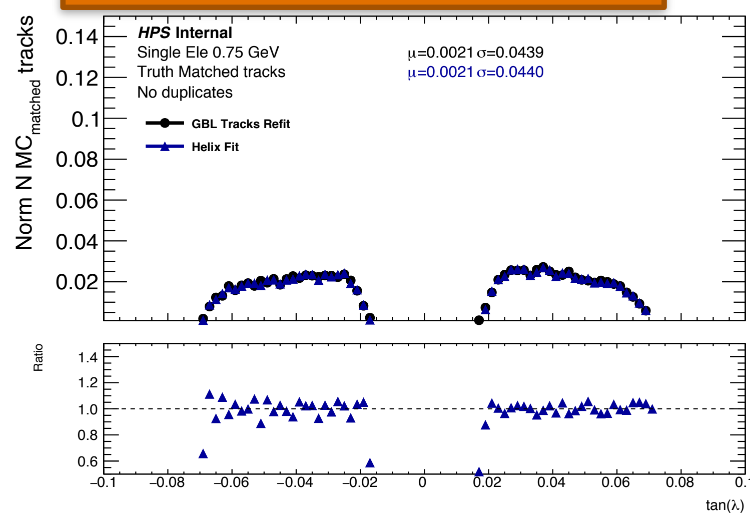


same momentum injected is reconstructed



Better reconstruction on z0

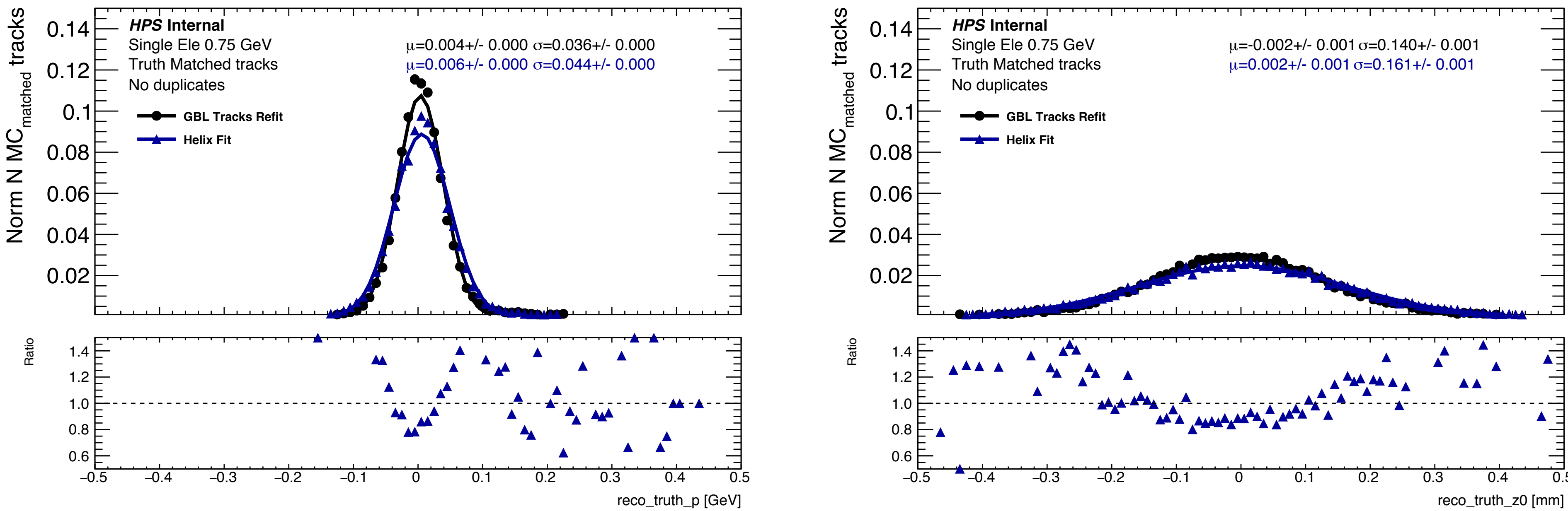




Helix tracks and GBL Refit - Comparison to truth

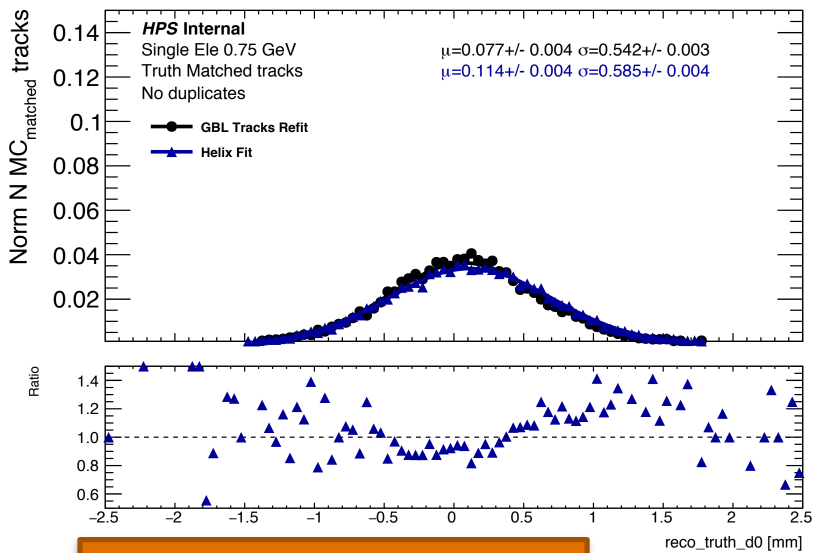
SLAC

- Tracks are requested to be matched to mcParticles in the event
- The Matching Criteria checks which particle from simulation generated the hits-on-track
- **Found about ~10% duplicate rate** in single electron sample (by checking that a different track is matched to more than one MC particle) - quite large and needs to be addressed
- **Found 0% fake rate in this sample**. Suspicious but expect small anyway

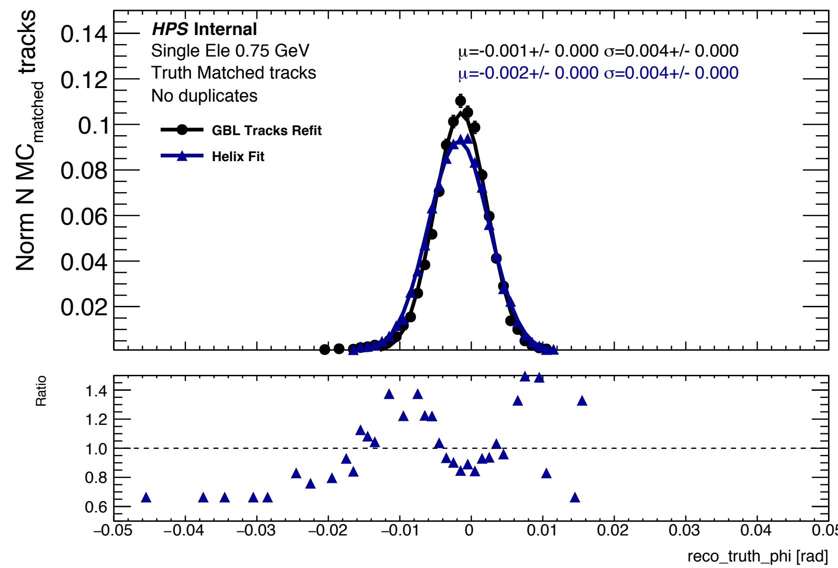


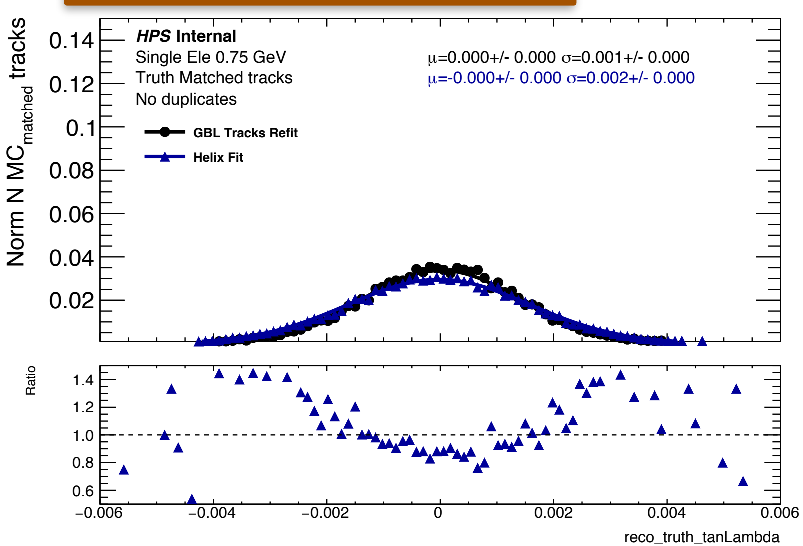
Improvement of p and z0 residuals with respect to truth matched particles
Gaussian shape models ~ok (not momentum, due to energy loss modelling)
Resolution improvements observed from truth: **p: 18%, z0:12 %**

Helix tracks and GBL Refit - Comparison to truth



d0 improvement 7%



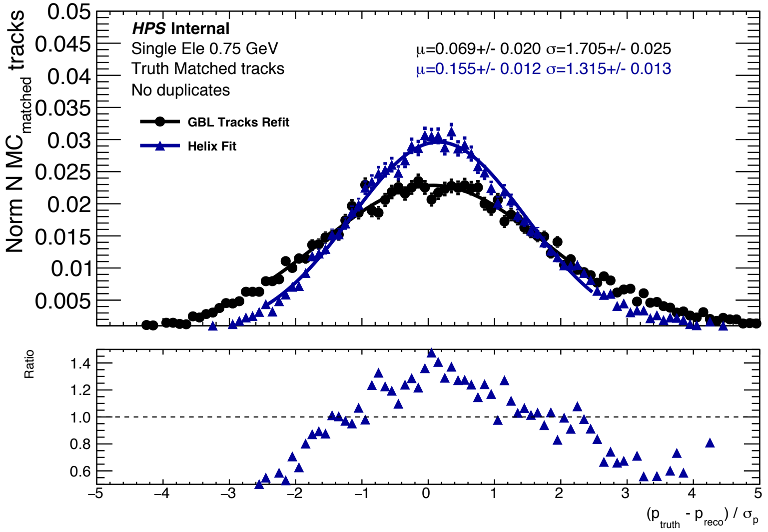


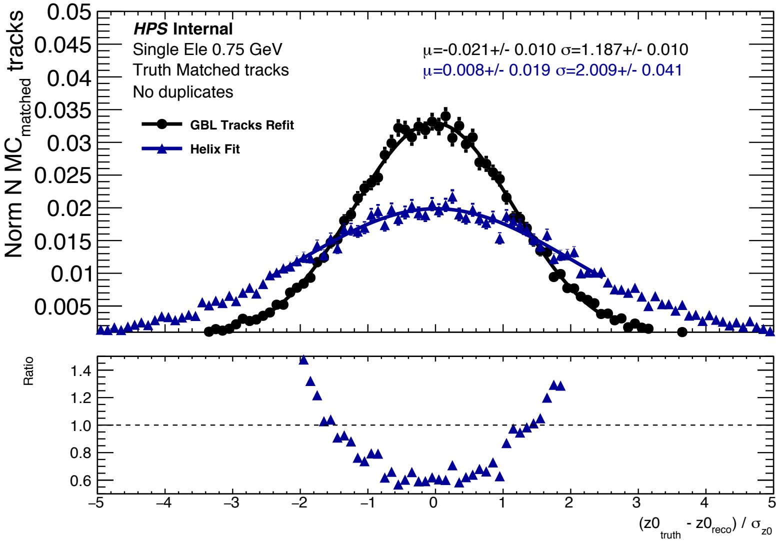
General improvement of all track parameters with respect to truth with respect to Helical Track Fit. Track parameters are wrt ref point (not the best due to b-field non - uniformities

This is in line with what has been observed back in 2017 by MattG

Helix tracks and GBL Refit - Check over the pulls

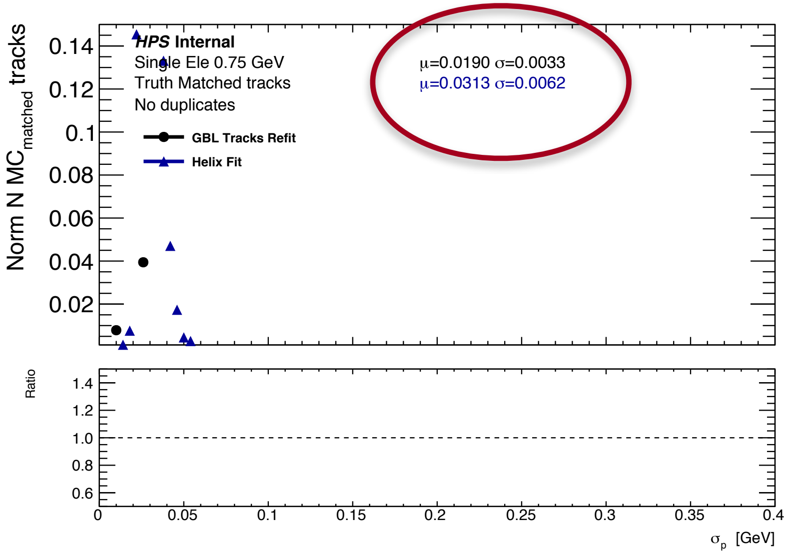
- Pulls are computed dividing the truth residual over the correspondent error from the covariance matrix





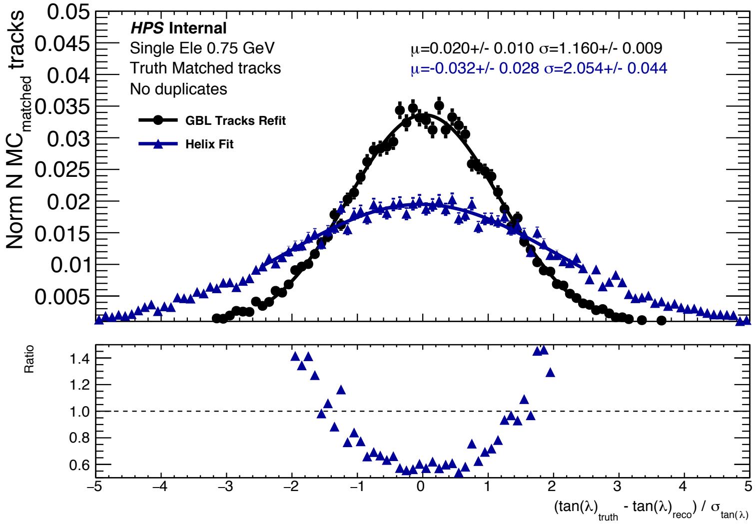
GBL provides a much better guess of the z0 error with respect to seed track However momentum error seems to be largely smaller than expected
We see 18% improve of the residual pull and ~2x smaller estimated error.

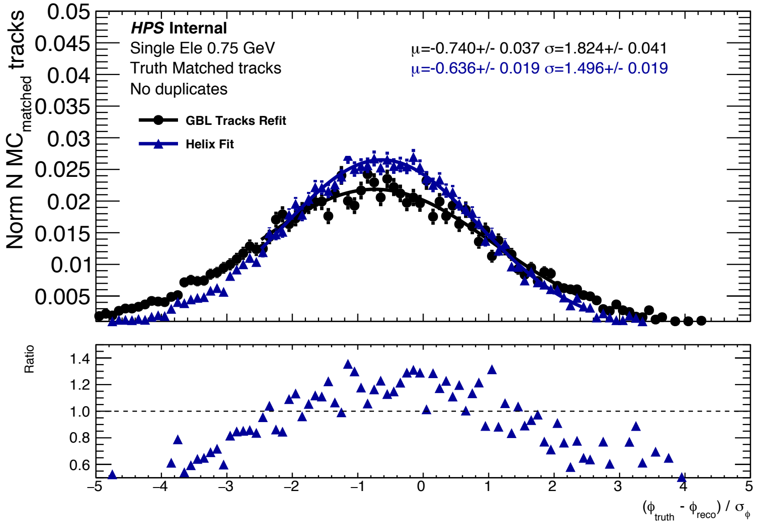
This can be due to:
- Wrong covariance matrix computation
- In-accurate transport of the track params to (0,0,0)



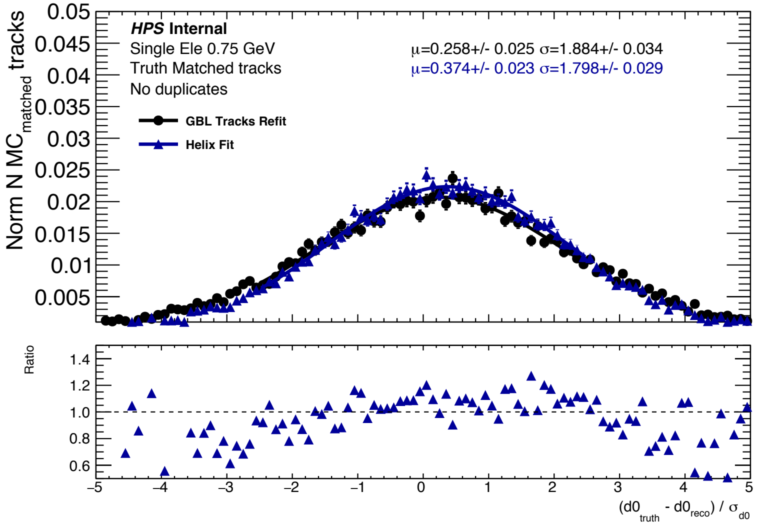
Helix tracks and GBL Refit - Check over the pulls

- Pulls are computed dividing the truth residual over the correspondent error from the covariance matrix

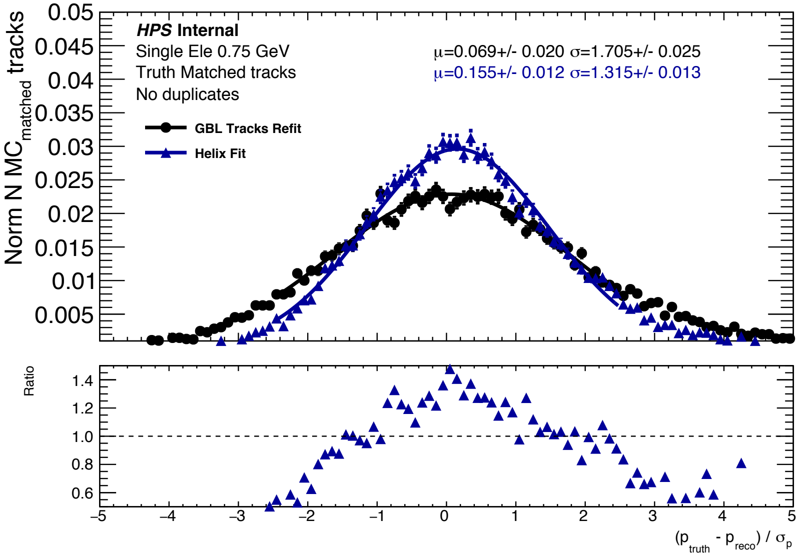


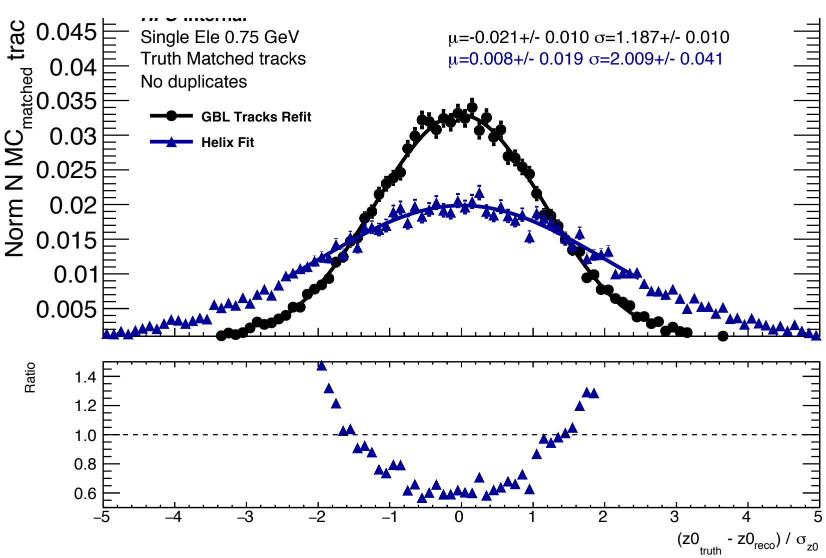


Seems like the effect is present only in the circle fit:
- tanLambda and z0 have pulls with sigma ~1 and bias ~2% => **OK!**
- p, d0, phi0 all have pulls ~ 2 => **phi is largely biased**



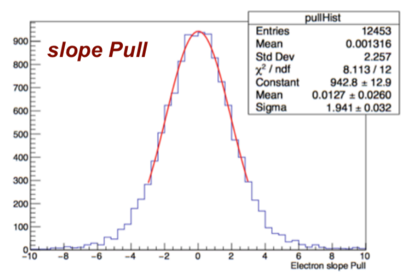
Comparison with A' sample



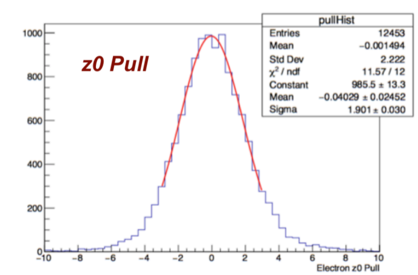


Opposite results with respect what MattG shown in 2017 at the HPS collaboration meeting
- 100mm 40-50MeV A' sample
- Back then (~1.4-1.5 circle fit pull widths)
- x2 pull width for linear fit

Plan to check pulls on measurement instead of reference point (less math)



These are too wide by x2!

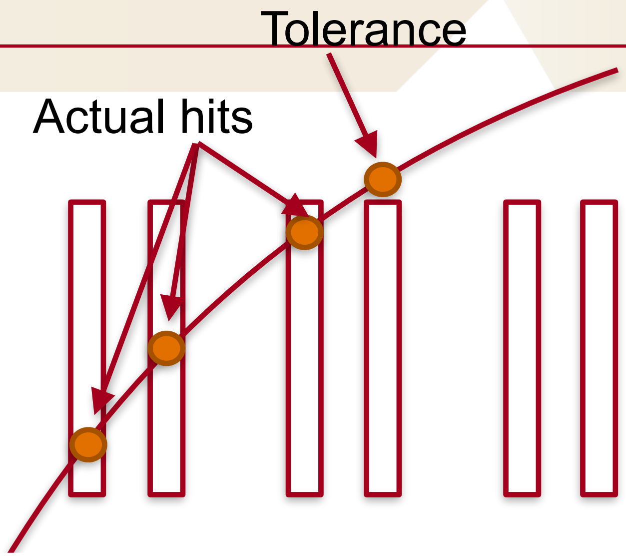


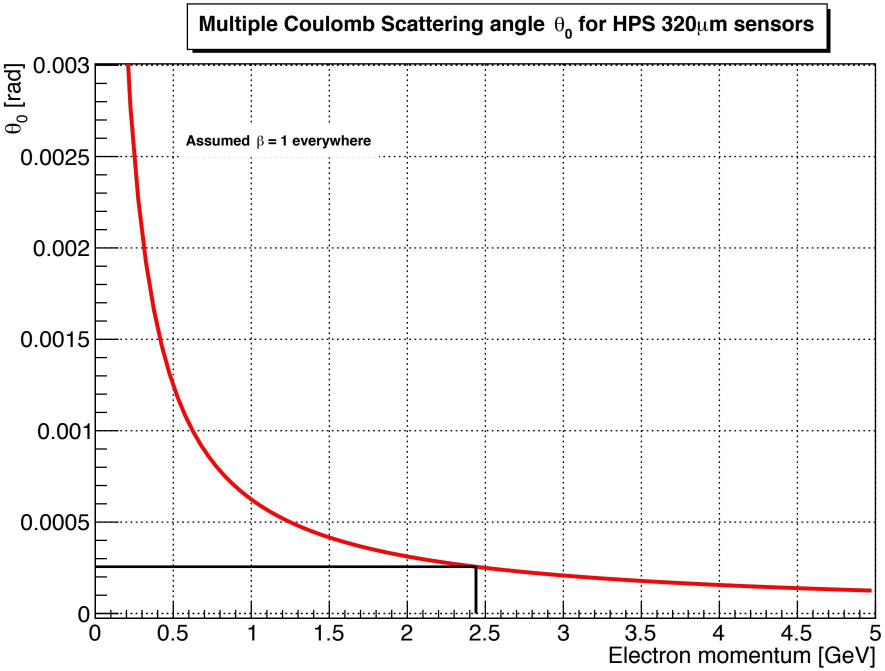
Z-dependence? Momentum-dependence?

MattG

Multiple Scattering treatment

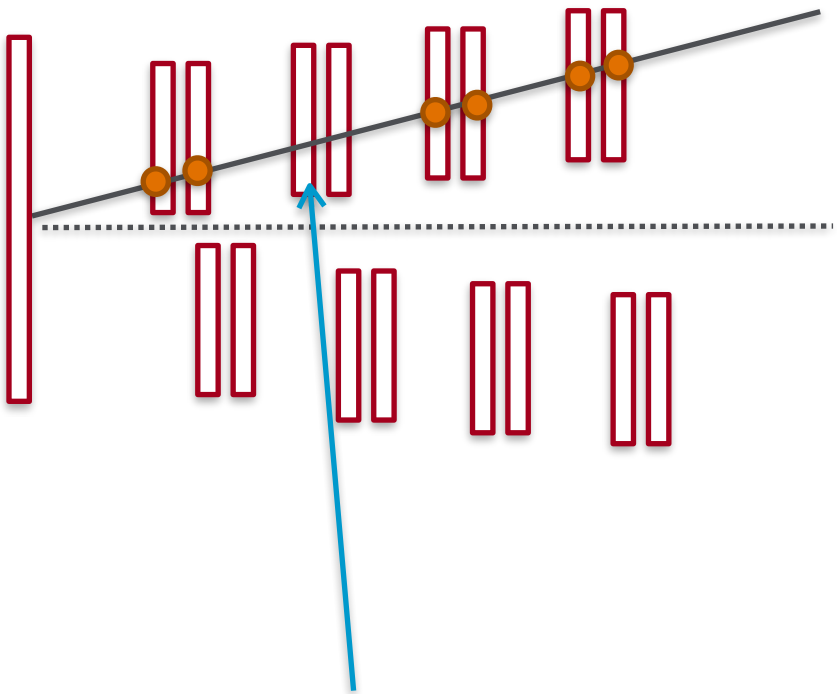
- The Multiple scattering contribution is estimated from the track helical fit:
  (1) Find Scatter Points along Helical Fit
  - Check x (y) > (Δu(v)/2) + 100um
  - Strips are along y
  - 100 um of tolerance (fixed)
- Scattering angle is computed from PDG
- **Found small issue with missing hits and multiple scattering in GBL Refits**
  - **Scatter points were only added for hitsOnTrack**
  - **Holes were neglected**





Multiple Scattering treatment

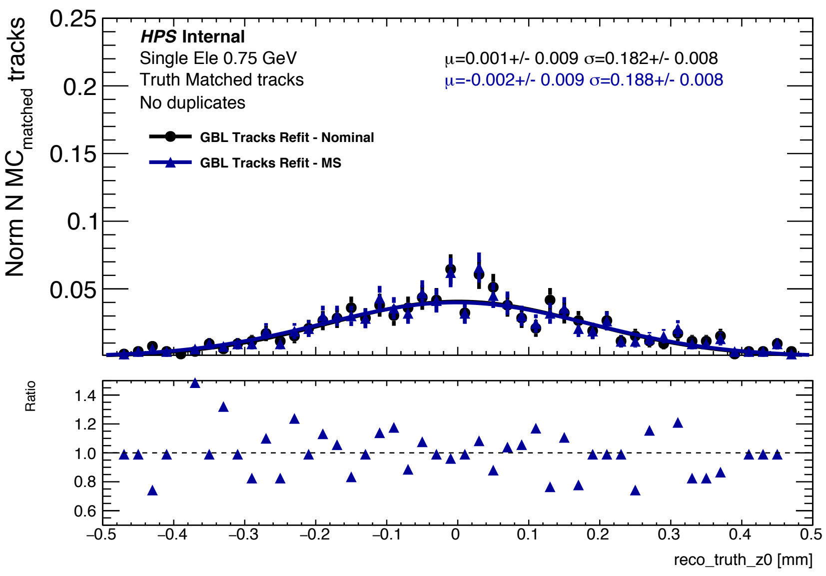
- Treatment of MS not fully understood (by me)
- Second:
  - Multiple scattering only added if hit-on-track is present
- Fixed from iss630
  - Effect on 2016 should be small:
  - Vertex analysis asks for L1 hits in main SR, will affect LXL2 searches
- Different for 2019 as some hybrids are dead in Ly4

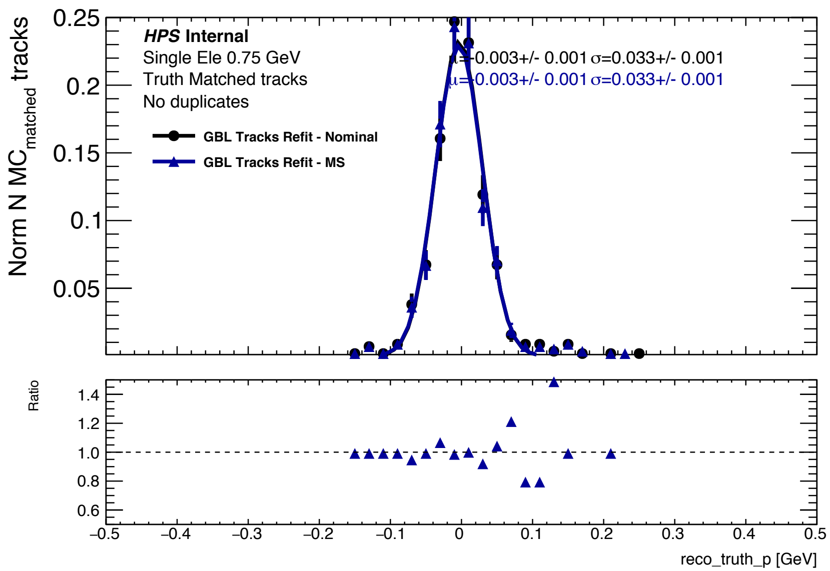


Track misses hits on track here:
MS is not added

Truth residuals and pulls - linear fit

- Single electrons E=0.75 GeV sample
- **Tracks are required to have 5 hits and one hit on L6**
  - Ensures maximum effect for the change done
- Better description of the error for these tracks
- **Black**: “proper” treatment of multiple scattering
- **Blue:** nominal



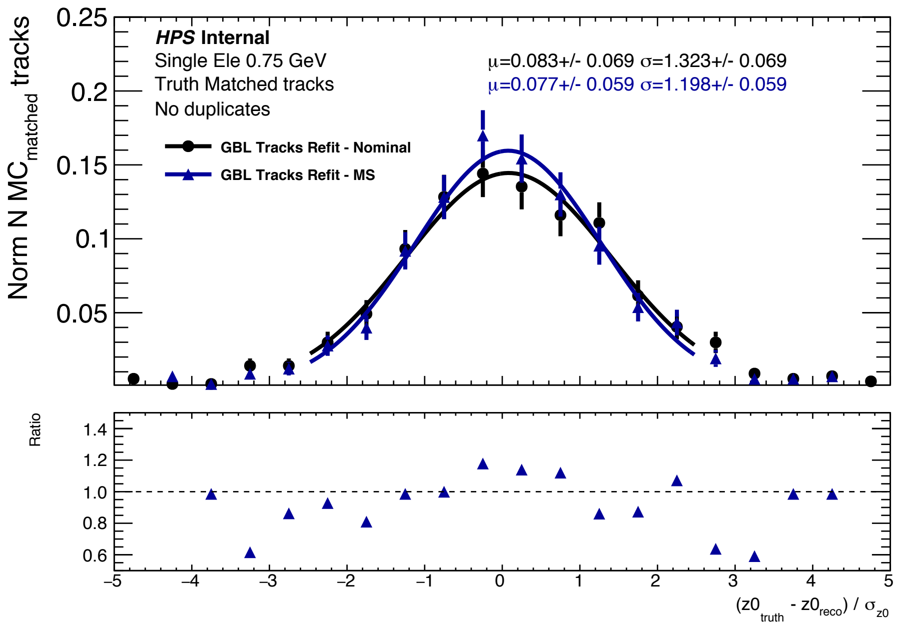


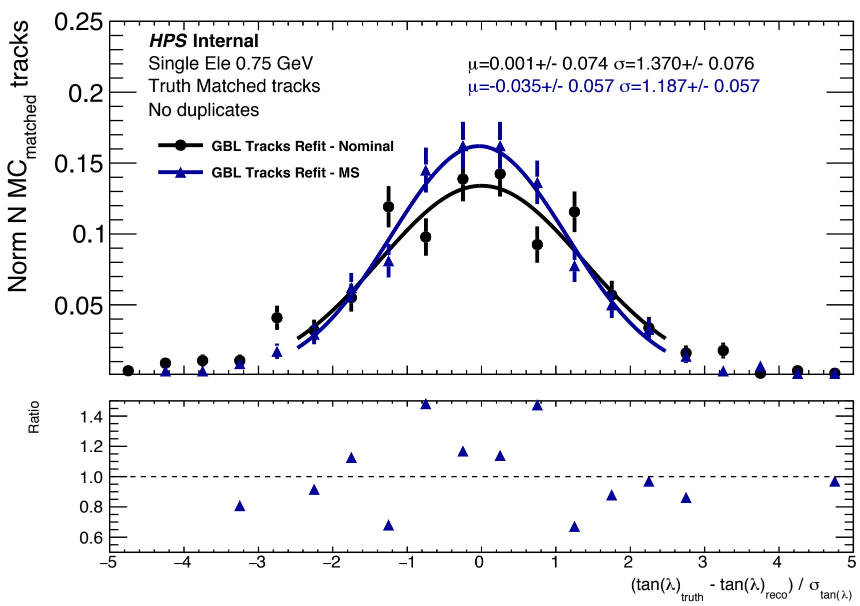
- No effect on truth-residuals with respect to nominal - Expected

Truth residuals and pulls - linear fit

- Single electrons E=0.75 GeV sample
- **Tracks are required to have 5 hits and one hit on L6**
  - Ensures maximum effect for the change done
- No effect on truth-residuals with respect to nominal - OK
- Better description of the error for these tracks
- **Black**: "proper" treatment of multiple scattering
- **Blue:** nominal

z0 / tanLambda => 10% improvement
In error description
Pull similar quality of all track

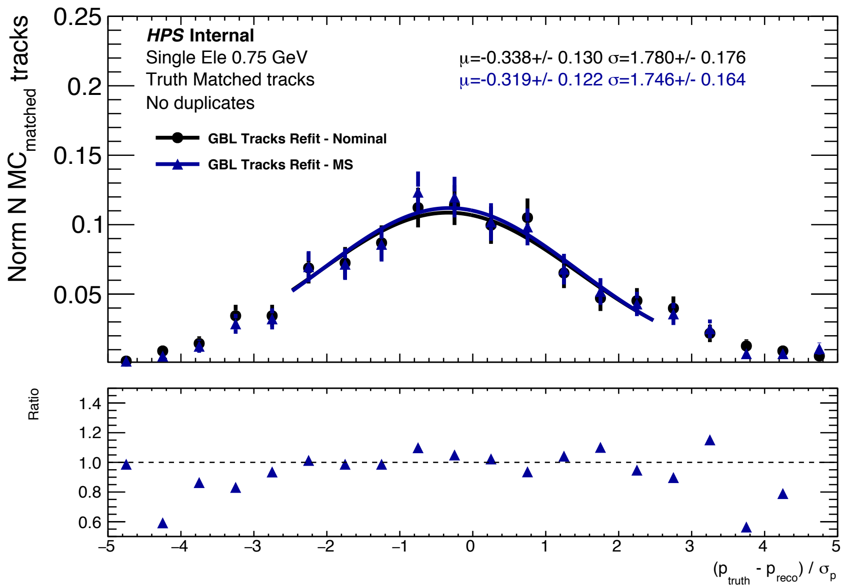


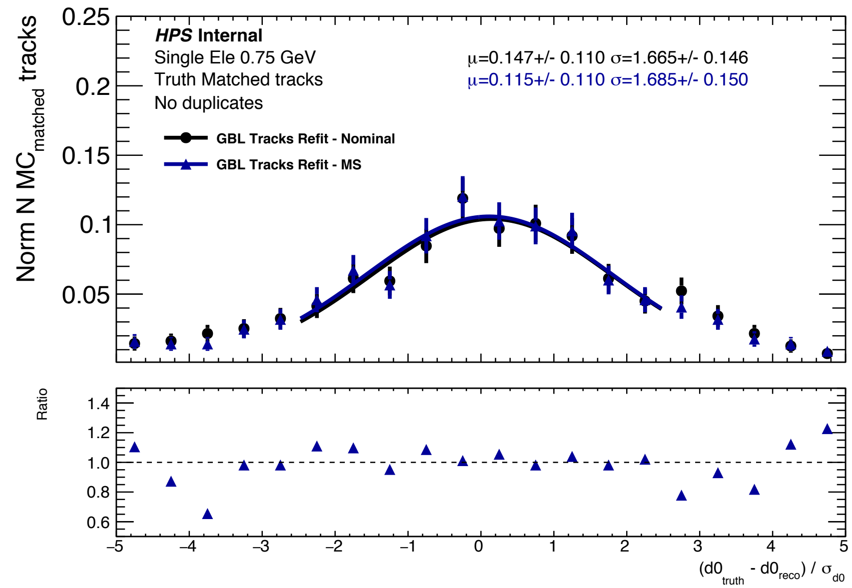


Truth residuals and pulls - circle fit

- Single electrons E=0.75 GeV sample
- **Tracks are required to have 5 hits and one hit on L6**
  - Ensures maximum effect for the change done
- No effect is observed on circle fit
- **Is that expected?**
  - No resolution to guess phi kinks (?)
  - Multiple scattering in phi not properly computed in Java Port of GBL fit (?)
- **Unfortunately another thing to check**

No Effect on Circle Fit ?!
Un-expected

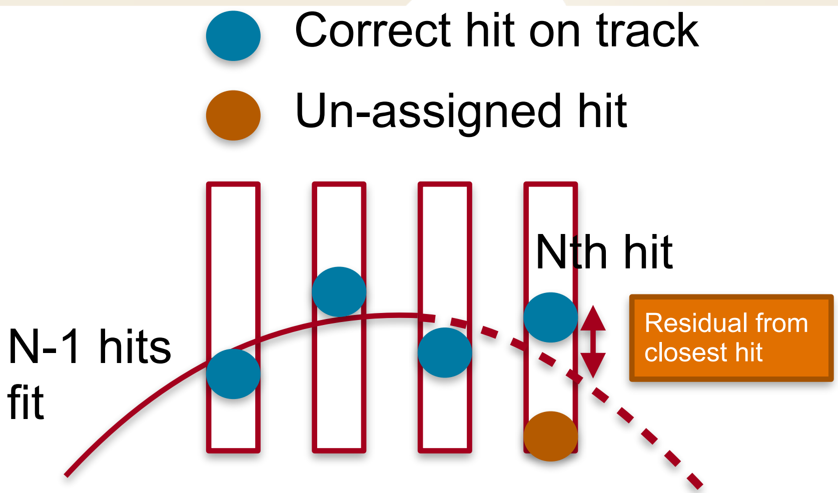


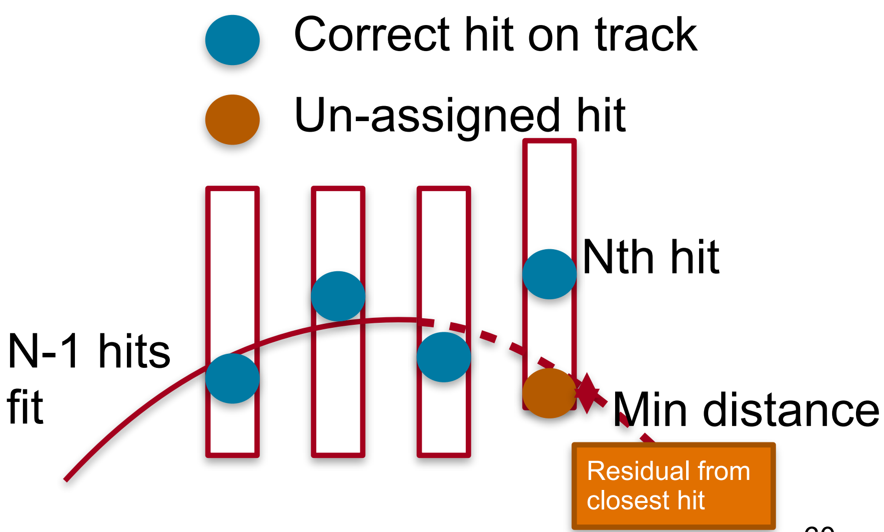


New driver was needed for GBL unbiased residuals

- **Revisited the Unbiased hit-on-track residuals driver - issXX to be opened**
- The reason being that a whole track finding was re-performed removing hits on layers=> residuals were then defined wrt the closest measurement in the removed layer [at least the Unbiased Residual Driver I was pointed to]
  - Doesn't catch properly detector movements in case of other hits on layer
- **Unbiased residuals are now formed refitting the original GBL track**
  - GBLStripClusterData list is persisted
  - GBLPoint under check is removed and substituted with a scatter (to keep MCS effects)
  - GBL Trajectory is refit (*)
  - Hit-on-track is computed
- **This, in principle, should be the right way to compute the GBL residual**

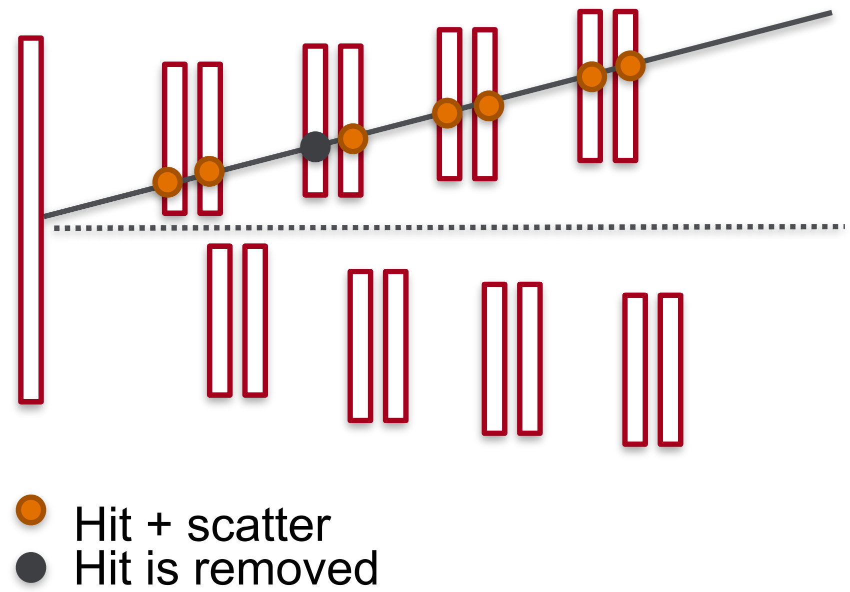
- (*) GBL doesn't converge over a single refit. I haven't iterated the refit yet should be done





Computation of the unbiased residuals

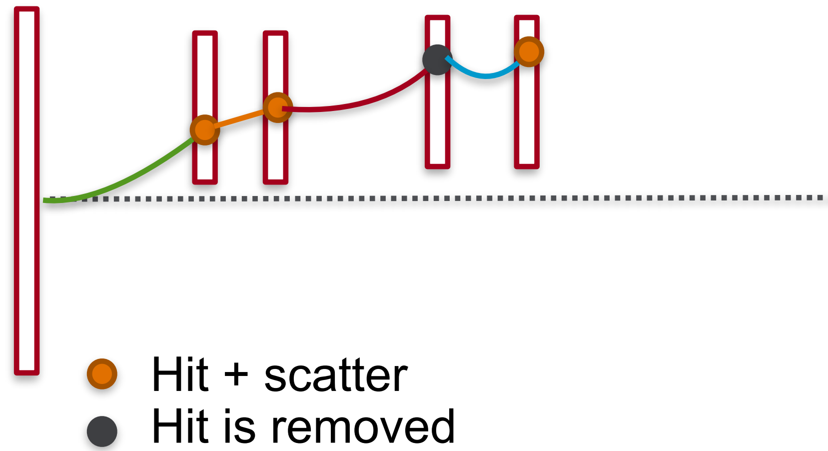
- Added persistency of the GBLStripClusterData associated to a GBL Fit trajectory
- Each GBLStripClusterData object holds:
  - ID for the sensor
  - measurement (+err) in local coord
  - Track fit position (for biased residual)
- Loop on the hits, each hit is removed and substituted with a scatter
- GBL Refit is re-performed

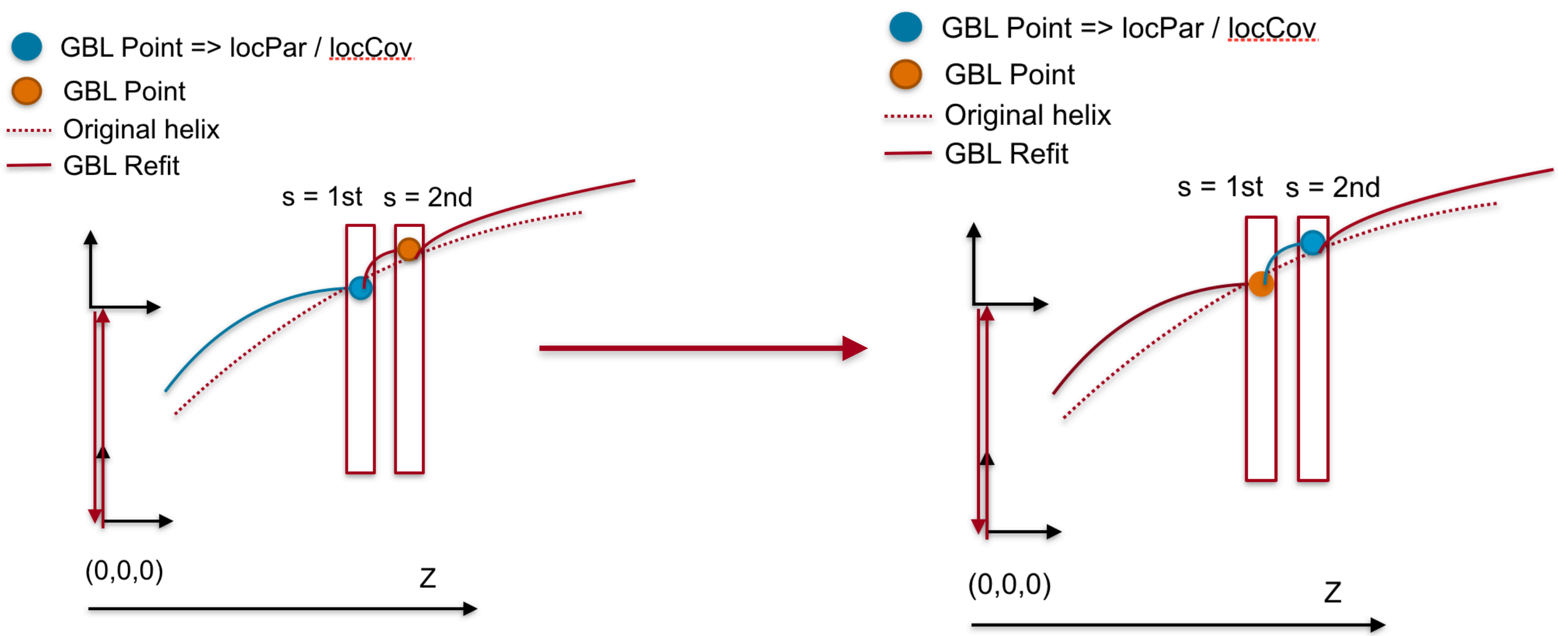


Work in progress being done for fully unbiased residuals (both sides are removed)

Computation of the unbiased residuals

- GBL track has different track states on surface at each sensor [between each measurement a different helix is computed]
- **Extrapolated track position is corrected for each track state on surface**
- The residual is then computed **r = m - e** where
  m= measurement position
  e= extrapolation

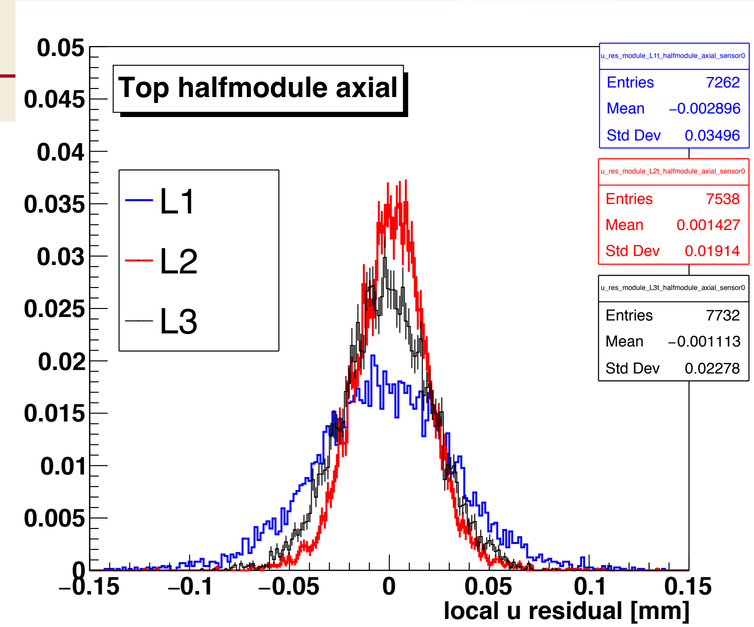


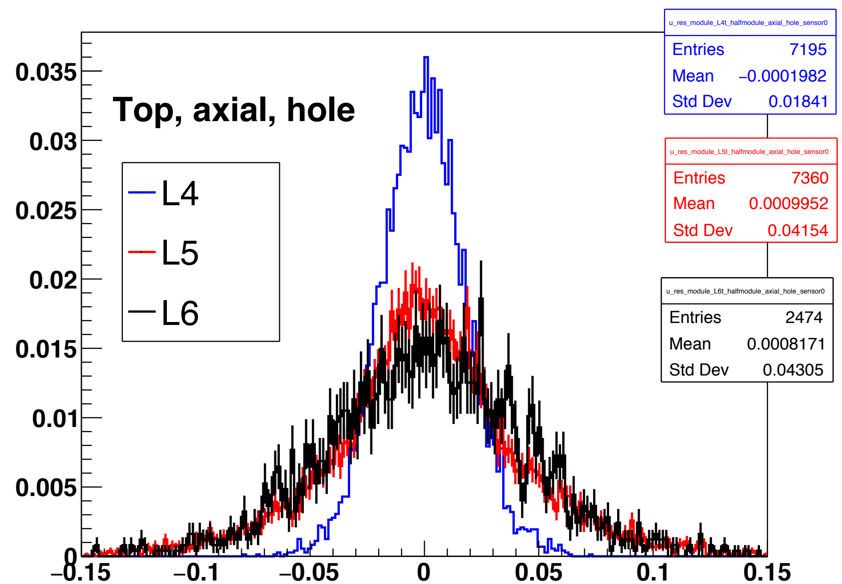


Unbiased Residuals

- Unbiased residuals are centered on zero with a width ~23um [avg] for single electrons at ~2.4GeV
- RMS Ly2 ~ RMS Ly4 (?)
- MS not included for holes-on-tracks
- Single GBL Refit for unbiased track

Re-Observed (originally done by MattS) that Ly4 has best residual with respect to the other layers
- Somewhat un-expected
- Cause should be investigated [perhaps lower priority though?]





Conclusions

- In the process of learning the software for the GBL refitting
- **Found really small issues in**:
  - **Multiple Coulomb Scattering treatment** - corrected in iss634
    - **Effect on z0**: error enlarged, better pull
  - Fix to an element of **CLtoPerigee jacobian**, for the rest is exact - issue to be made
  - Tested proper **application of Jacobian for change for reference frame from s=0 to (0,0,0)**. Minor effects (backup)
    - Minor, as electrons and positron tracks are corrected to Vtx position in analysis.
  - Order of our track parameters is different wrt GBL svn code
    - **If matrix algebra has been copied directly, might cause issues. Algorithm needs a check, in principle.**
- **Strategy to obtain track parameters to ref-point (0,0,0)**
  - Intrinsically uses B-Field uniform => need to be changed for 2019
  - Worth checking on 2016, which is data/MC we understand better
- **Observed no multiple scattering effects on the circular fit. Un-expected.**

Next steps - Track fitting and Tracking performance

- **Recompute** truth_residuals / pulls / errors at first measurement instead at s=0
  - This should the degree of precision of our covariance matrix from GBL port
- Fit in **Ly1-Ly6 and use RK to extrapolate** the track parameters back to reference point or vertex position
  - Should be easy as already implemented for extrapolation to ECAL (code is available)
  - Same as Robert does with KF!
- **Use a step-by-step approach with a full Jacobian between layers (2019)**
  - Robert uses a variable B magnitude + Rotation to align to the direction of b-field. Reference "Jacobians in Homogeneous B-Field" contains the full expression.
  - Worth implementing?

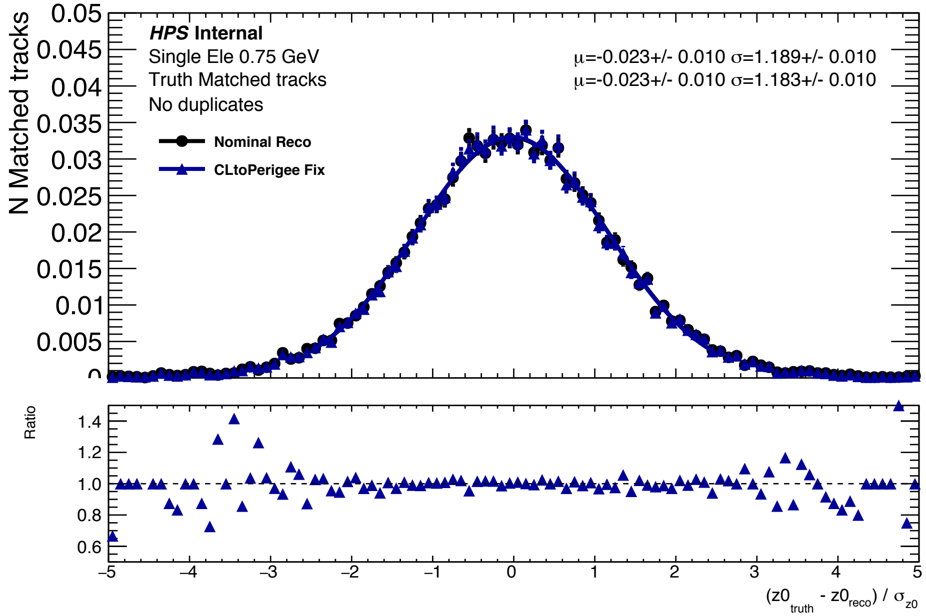
Next steps - Alignment

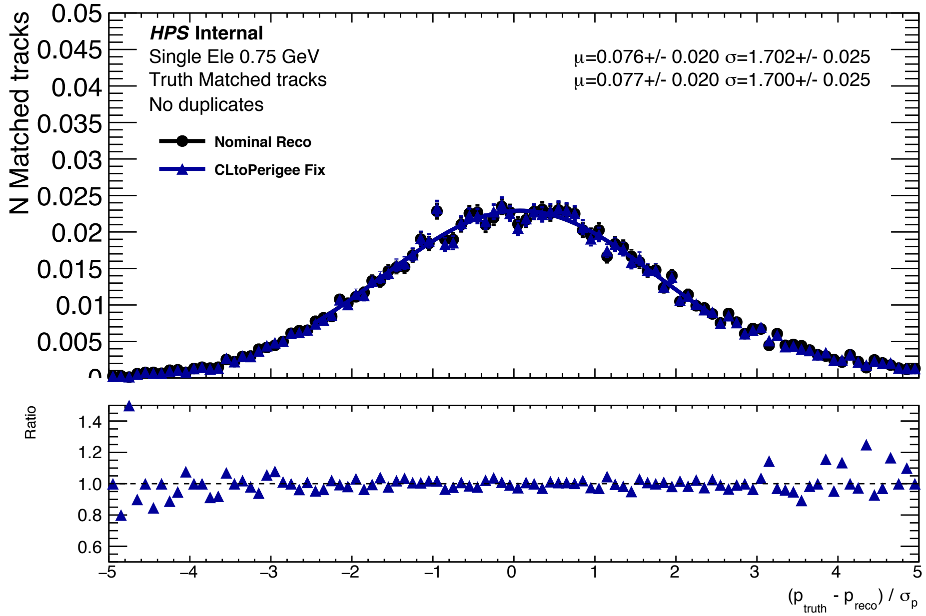
- I've began looking into millepede configuration and code
  - **Ramp-up work still in progress**
  - Sorted out how GBLData is filled, discussed with other collaborations experts
  - **Preferred to have a feeling of what is actually fed into the algorithm before running it, then things started to pile up**
  - Plan to dig into it before Xmas break.
- **High-priority to-do list:**
  - Generate a compact + lcdd with sensors moved by hand and check new code for unbiased residuals
  - Re-align and check results with metrics developed
  - For 2019 need to decide a structure for L0 - L1
    - Fixed Millepede-ID indexing for 2019, iss622, which is a start…

Next steps - Track selection (not only GBL)

- Need to urgently revisit the strategies used for track finding:
  - Strategy efficiency and fake rate should be evaluated and run separately
  - Remove duplicates from analysis level, tracks should (in principle) arrive to analysers **clean** and **non-ambiguous**
- Request to **revisit** and **optimise object identification cuts**
  - Should be possible to address in a short time scale
- Decide a set of generic track quality cuts for analysers
  - Assess selection efficiency and fake rate.
- Aim to a performance support note for 2016 analysis (and 2019)

Curvilinear to Perigee Jacobian Checks





- **The curvilinear to perigee** Jacobian is used when the correction to the track parameters is applied to the original track
- I've checked (to my best knowledge) if the transformation was correct:
  - Found small issue in one element
  - **Minimal effect.**
- Checked pulls after correction:
  - Consistent with the fix
- **After fix, I'd say Jacobian is correct**