VXD Alignment

Peter Kvasnicka, Tadeas Bilka
Peter.Kvasnicka@mff.cuni.cz
Tadeas.Bilka@gmail.com

Charles University in Prague

16th International Workshop on DEPFET detectors and applications,
25-28 May 2014, Kloster Seeon, Germany

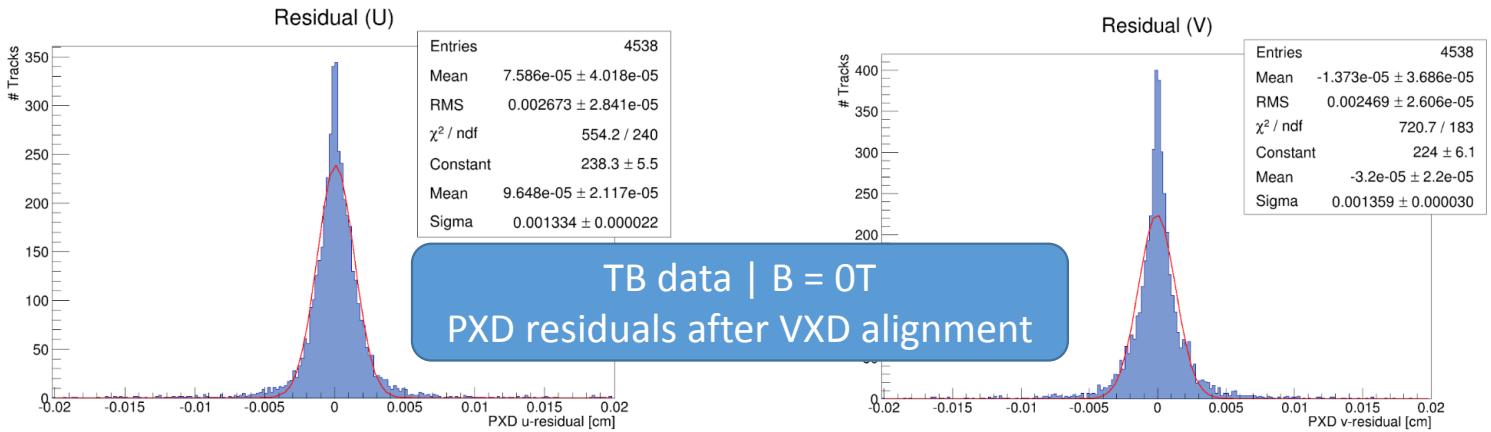
Outline

Current status | VXD alignment in basf2

- General Broken Lines (GBL)
  - Track model / fast refit with proper description of multiple scattering
  - Adds multiple scattering effects to initial trajectory as additional fit parameters
    https://www.wiki.terascale.de/index.php/GeneralBrokenLines
  - **GBLfit** module in basf2 → production of alignment data files for Millepede
    - Basic track selection (point Chi2, track p-value, minimum degrees of freedom)
    - genfit::TrackCand → genfit::Track → gbl::GblTrajectory → Mille binary
- Millepede II
  - Implementation of the Millepede algorithm for global alignment
    https://www.wiki.terascale.de/index.php/Millepede_II
  - Linear least squares fit for *very* large number of parameters
  - **MillepedeIIalignment** module in basf2 → can compute (basic) constraints, run Pede and analyse results → xml with alignment
    - Python ***alignment_tools*** (in testbeam package)… allow to sum alignment in two xmls or sum alignment in txt (from Pede) with xml

Current status | Progress in basf2 since last F2F meeting in Prague

- ❑ GBL fit working with both TrueHits/Clusters
- ❑ Material treatment using thin/thick scatterers
- ❑ Alignment procedure for testbeam and Belle II VXD (fully working in MC)
- ❑ Telescopes supported (except real track finding)
- ❑ Basf2 modules for GBL fitting + Millepede alignment
- ❑ Testbeam geometry reflects the experiment
  - Local coordinates, sensor positions …
  - Note that PXD is shifted by almost 5mm from nominal position!



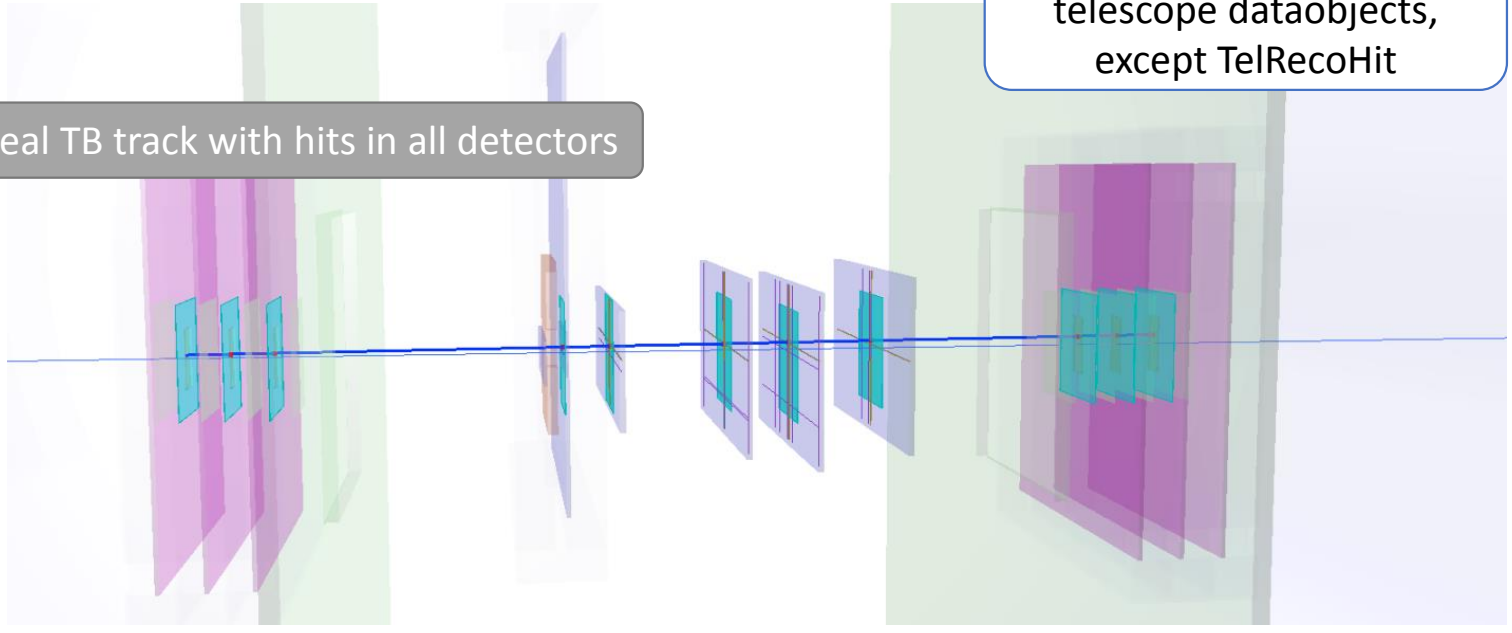
Current status | MC tracking for EUDET telescopes

- Cloned modules with support for telescopes added
  - EUDET telescope detector: Const::TEST
  - EUDET dataobjects: TelSimHit, TelTrueHit, TelCluster, TelRecoHit

✓ **TrackFinderMCVXDTB** … MC telescope + VXD track finding
✓ **GenFitterVXDTB** … fitting for telescopes + VXD
✓ No changes necessary for alignment

✗ Display does not support telescope dataobjects, except TelRecoHit

Real TB track with hits in all detectors



Current status | Full Belle II VXD alignment

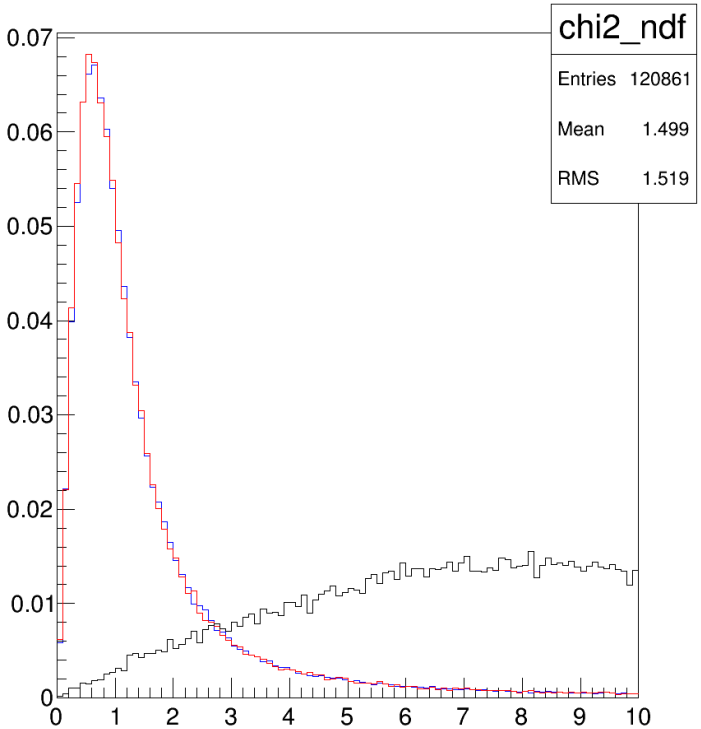
- Basic example available in release
- See alignment/examples
  - Step 1 a/b … generates ip/cosmics sample
  - Step 2 a/b … uses misaligned geometry to fit sample and produce data for alignment
  - Step 3 … alignment with Millepede on combined sample
  - Step 4 a/b … fit samples again with alignment parameters from step 3
- Only misalignment in geometry at reconstruction level
  - 100µm in U/V, 1 mrad in gamma
  - Available as xml in svn
  - Official version does not fully support slanted SVDs (solved locally)
- On following slides: results of single alignment procedure iteration
  - > 1000 alignment parameters fitted (around 200 000 tracks)
  - Takes < 1min
- Known issue: Need to remove TracksToMCParticles relations from input in steps 2 and 4

Current status | Full Belle II VXD alignment

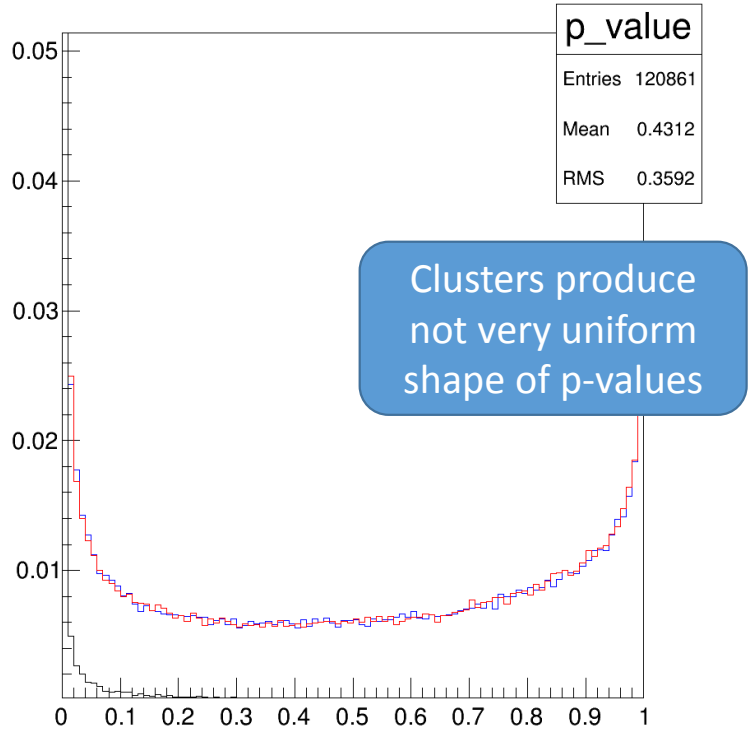
- > 100k muons from IP + > 100k cosmic muons (field off)
- Generated average misalignment 100um in u, v; 1mrad in gamma
- Plots for B=0 (cosmic muons)

Ideal geometry | misaligned | after alignment (using constraints)

Track Chi2/NDF



Track P-value

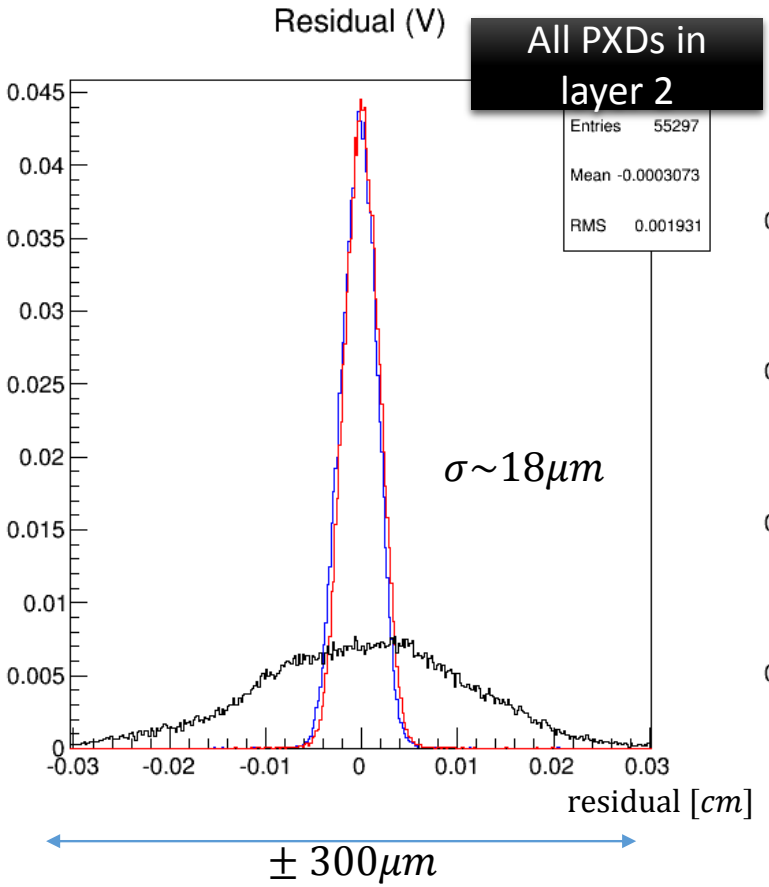


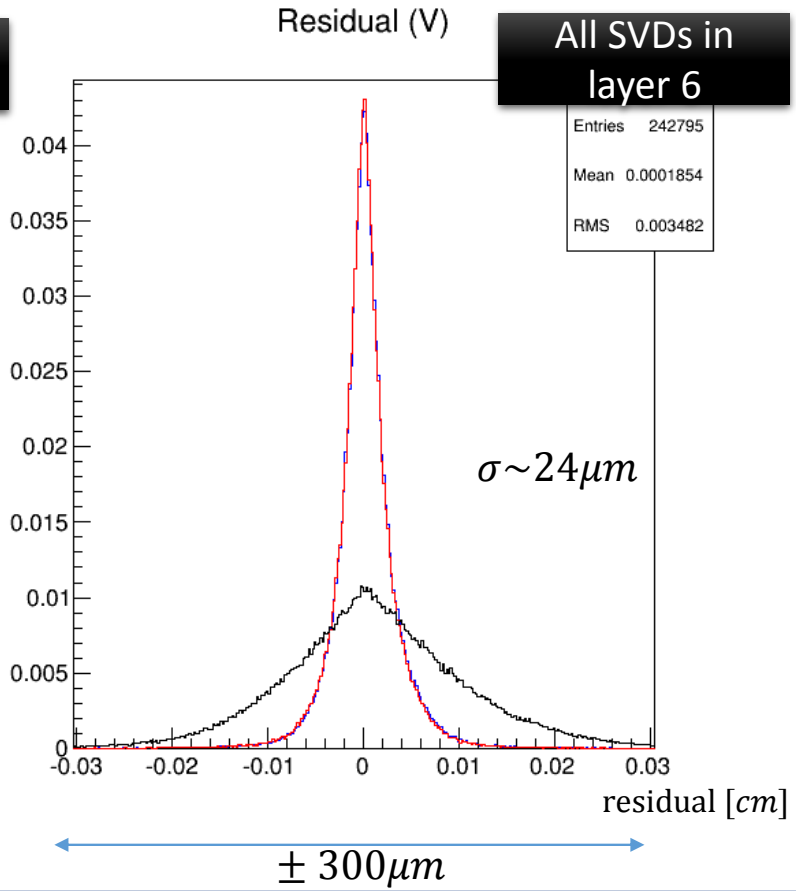
Clusters produce not very uniform shape of p-values

Current status | Full Belle II VXD alignment

- Residuals in ***Z*** in layer 2 (PXD) and layer 6 (SVD)
- Plots for *B = 0T* (cosmic muons)

Ideal geometry | misaligned | after alignment

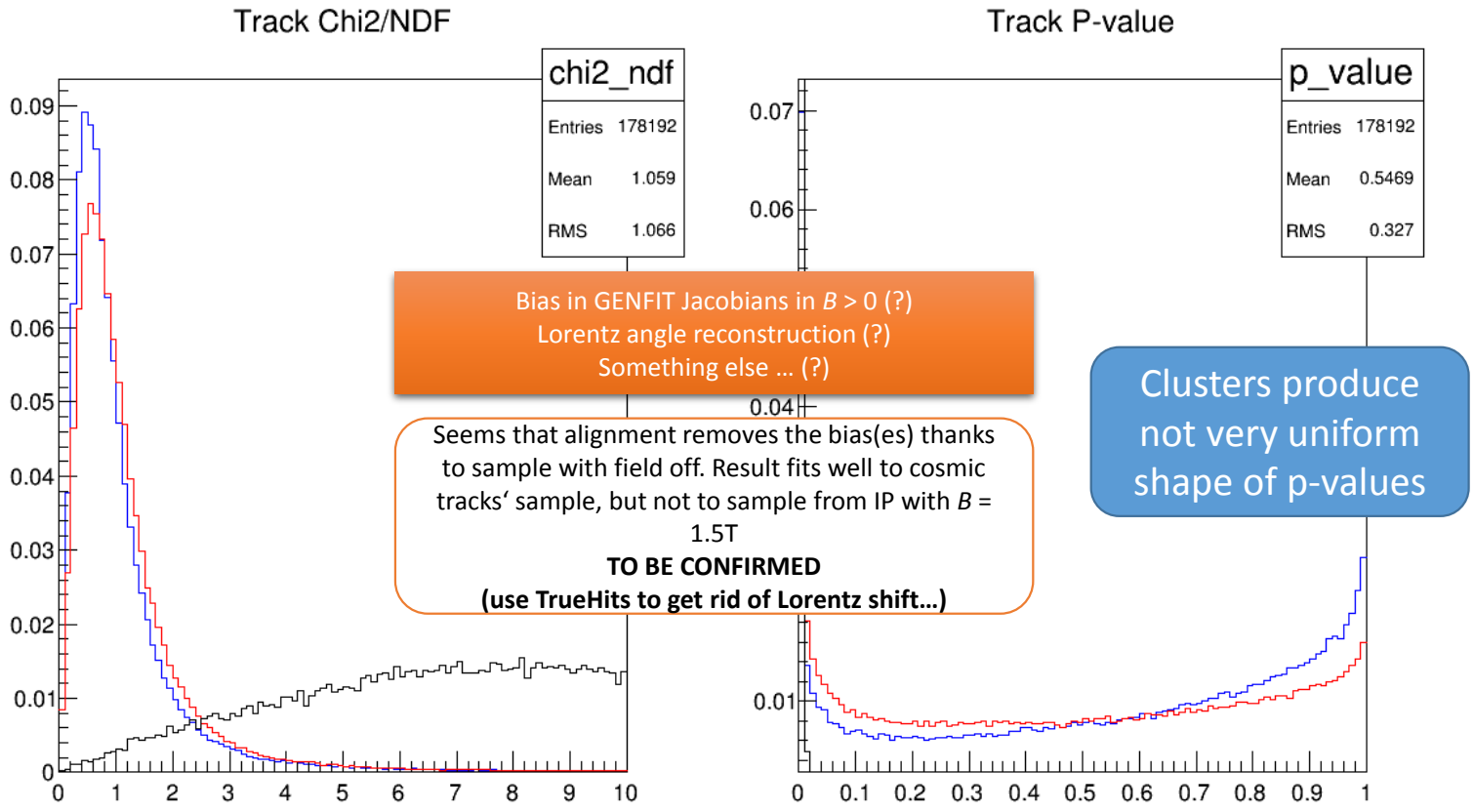




Current status | Full Belle II VXD alignment : Issues with magnetic field

- Chi2/NDF and P-values. Plots for B=1.5T

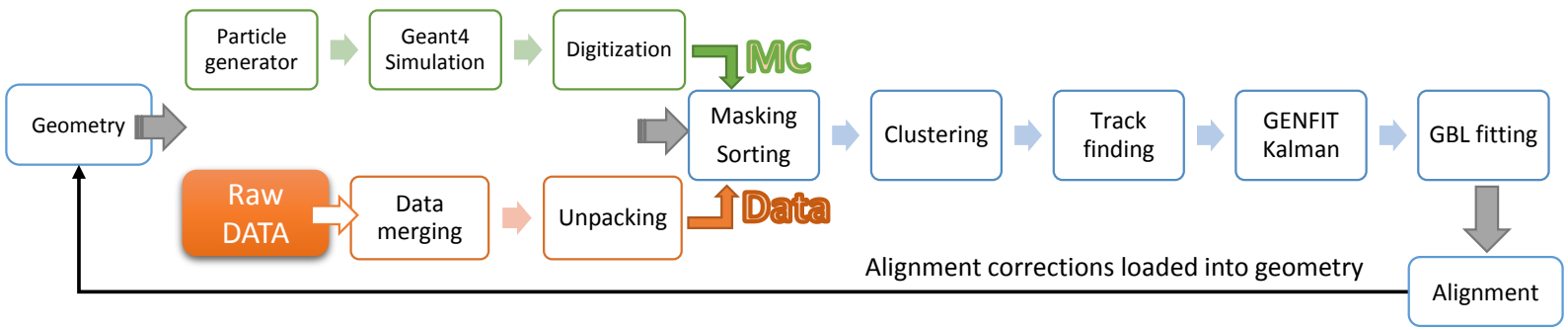
Ideal geometry | misaligned | after alignment



Current status | VXD testbeam example

- Simulate digits or get real data in form of merged digits
  - *SimulateDigits.py* (eutel/MergerTest3.py)
- Clusterize, mask, find tracks, fit with Kalman, output alignment data
  - *Digits2Tracks.py*
- Plug in computed alignment
  - *FitForAlignment.py* (2 step script)

- Not very nice (but automatic) manipulation with xml files for changing alignment parameters using python *alignment_tools* (in TB package) … will be gone after misalignment stuff is added



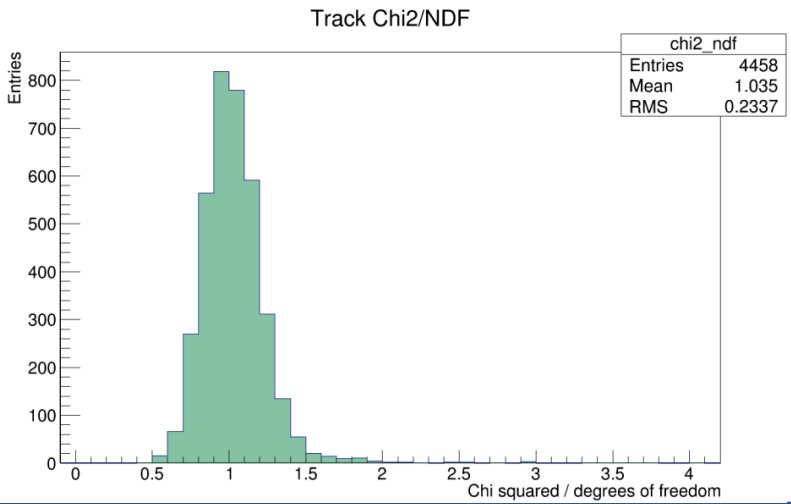
Ongoing development | New features

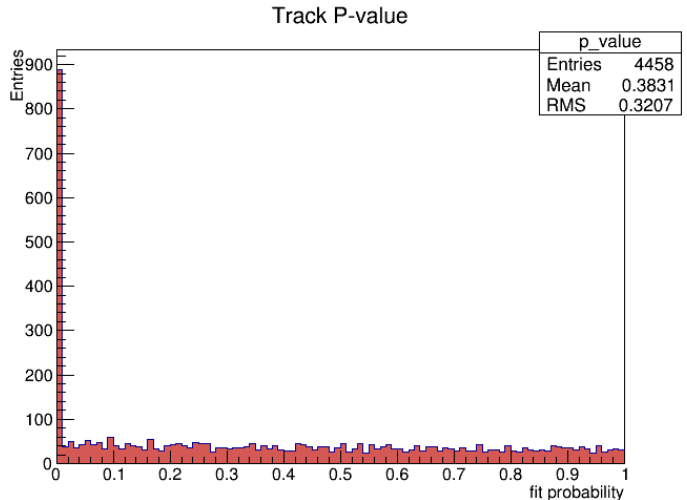
- Combination of SVD clusters in genfit::Track (correct errors for slanted SVDs)
- Trajectories with arbitrary combination of 1D and 2D hits supported

- Hierarchical alignment (experimental)
  - = production of derivatives w.r.t. parameters of composite structures (ladders, layers) and corresponding constraints
  - First testing version privately available
  - Highly depends on misalignment stuff – it needs matrices for transformation between subcomponents
  - Numbering scheme? Currently structures identified by string, but for Millepede, we need 1:1 map between this string and its integer label (for each parameter)

Ongoing development | GBL fit for VXD + CDC

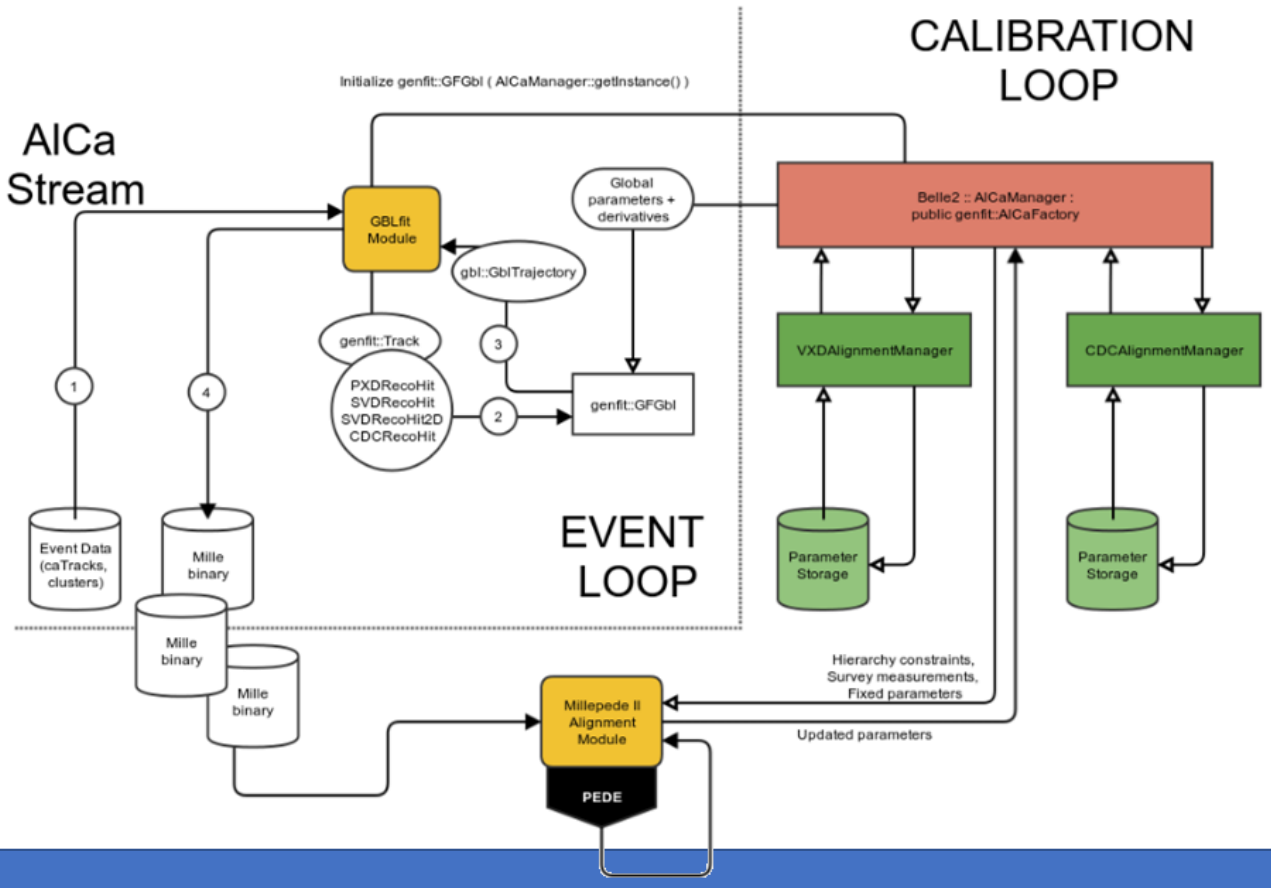
- First attempt to add CDC to GBL trajectory
  - Extended MCFitting.py (uses TrueHits)
  - Seg. Fault in standard MCFitting.py for TrueHits & RootOutput | clusters OK
    - No comparison to GENFIT
  - CDC measurement with highest weight taken; *thin* scatterers used
- Problems with hit sorting in long tracks (over 100 hits)
  - I cut the track before the point extrapolation steps back
- Alignment / calibration for CDC ?
  - „Just“ add derivatives and their labels





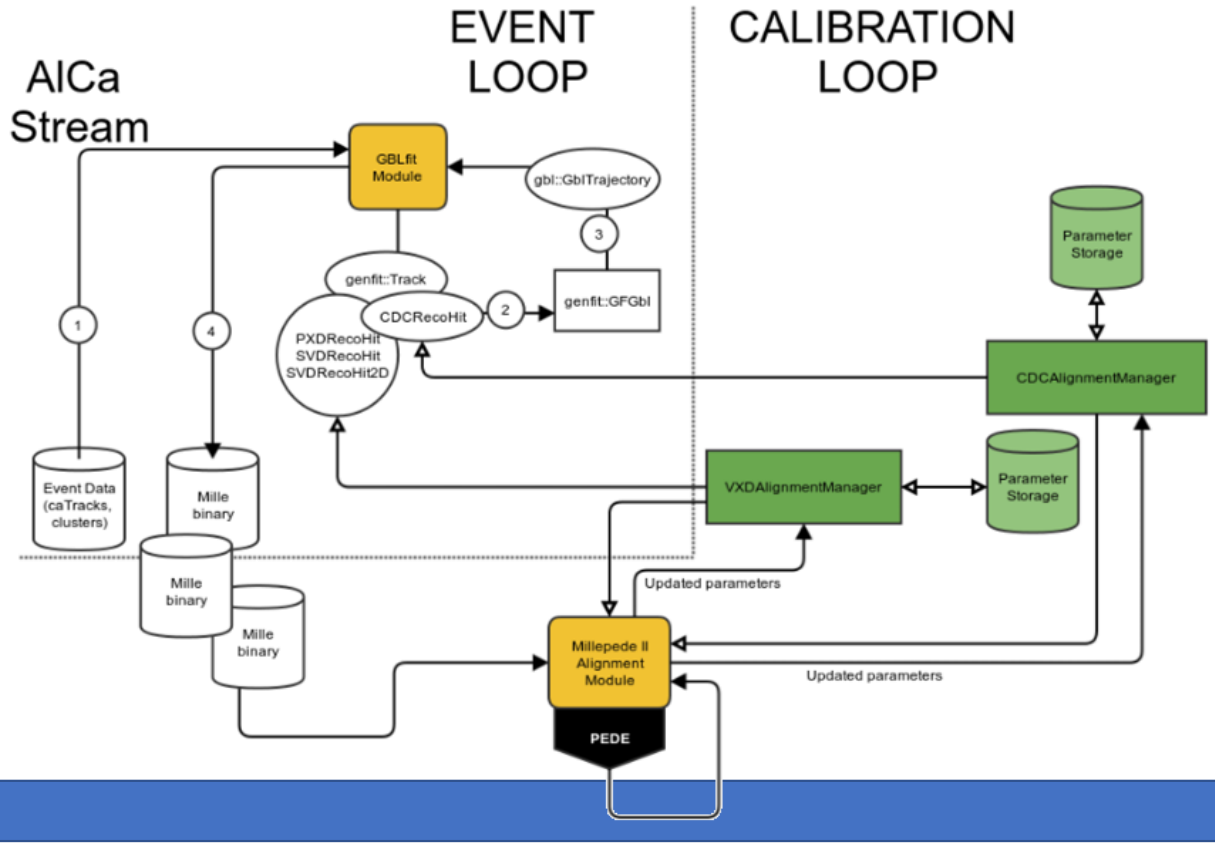
Implementation issues | Interfaces: Option 1

- All in one: Interface between GENFIT and derivatives through single class
  - Only GBL part of GENFIT affected
  - Construct GFGbl with pointer to AlCaManager



Implementation issues | Interfaces: Option 2

- Extending RecoHits: Each RecoHit has to provide derivatives for itself
  - Makes whole GENFIT alignment – friendly
  - 2 possible solutions:
    - XXXRecoHit: public *AbsMeasurementWithDerivatives* | *AbsMeasurementWithDerivatives* : public AbsMeasurement
    - XXXRecoHit: public AbsMeasurement, public *IAlignmentDerivatives* (interface only) … multiple inheritance (!?)



Implementation issues | GBL integration into GENFIT

- Current status
  - GBL inside GENFIT
  - GFGbl (AbsFitter) takes genfit::Track, propagates it to account for material and creates gbl::Trajectory
  - Output to Mille Binary hidden in GFGbl
  - GBLfit module: starts from genfit::TrackCand, but also possible to start from genfit::Track with reference state (previously fitted by KalmanRefTrack)

- Not neccessary to fit GBL trajectory for alignment
  - GBL used to construct linear equation system
  - Millepede performs last (in fact only) iteration of trajectory fit itself
- Full GBL integration ?
  - Non – measurement points … ThinScatterer should only have variance + plane (position on arc-length)
  - GblFitterInfo ? Or re-use Kalman stuff?
  - How to attach constructed trajectory to the track (GblFitStatus ?)
  - Iterations? Re-propagation of track or only update the states (→derivatives) ?

Conclusion

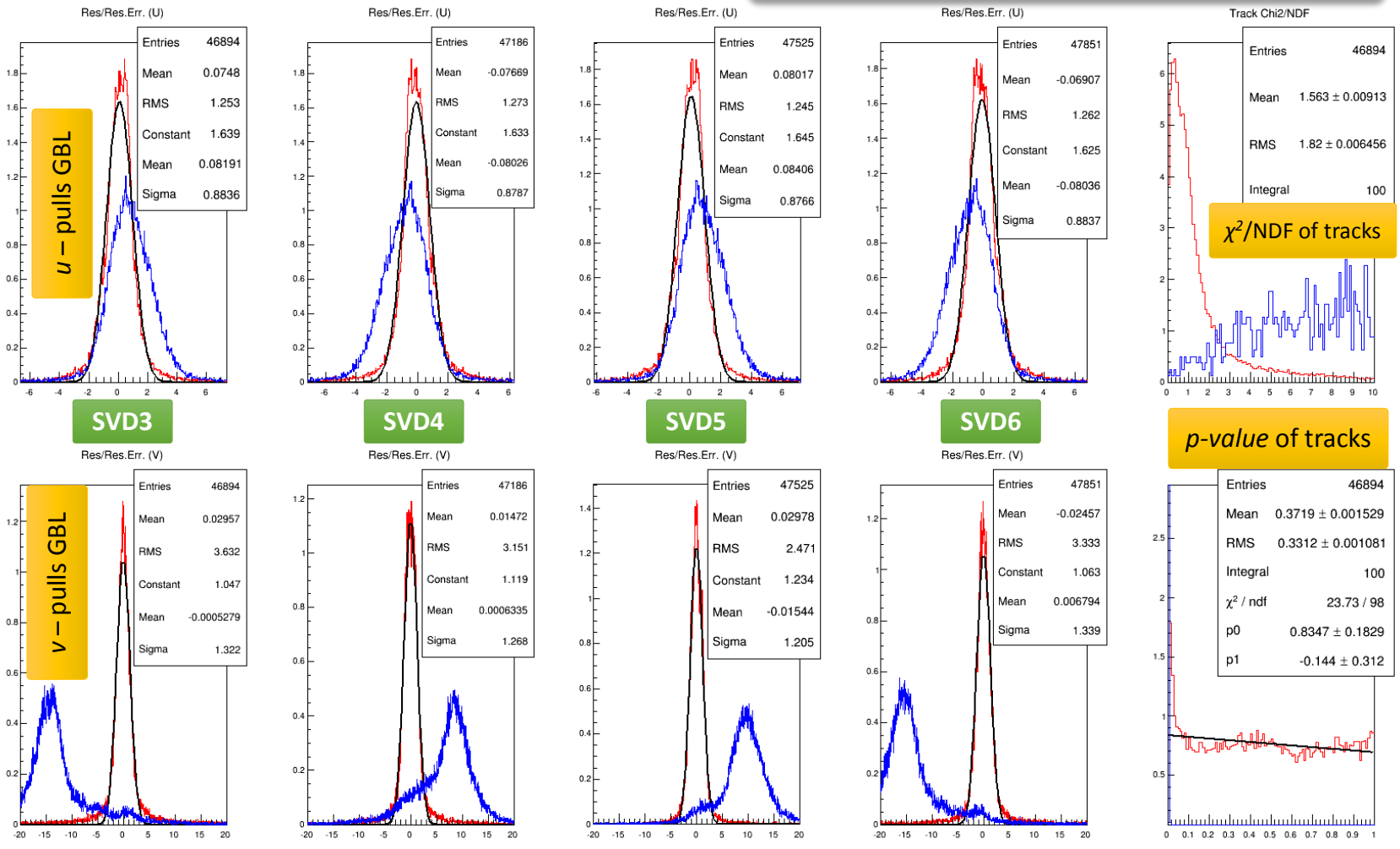
- Working example with Belle II VXD alignment in release
  - More studies needed, but the procedure works
- Working example for TB
  - Fully working in MC
  - Real data processing requires to take care of masking, track candidates…
  - Attempt to add telescopes without VXDTF using cluster combiner from Peter Kodys → additional studies needed
- Additional interface for global derivatives and labels needed
  - Especially because of hierarchical alignment (for each hit needs to somehow figure out all its parent structures and corresponding transformations)
  - Option with AlCaManager partly implemented including hierarchical alignment
- CDC experimentally added to GBL trajectory
  - I am still not sure about some details in CDC, but fitting performs „well“
- Testbeam package dependency
  - Not possible to make just a clone of Display module for testbeam (dataobjects added in a library, not in the module)

Testbeam alignment

TB alignment. Pulls

RUN 507 | Full tracks with 4 x 2 x 1D measurements
B=**1T** E=**4GeV**
Combined alignment data: **500**(4GeV)+**507**+**508**(5GeV)
SVD3 fixed, SVD6 fixed shifts. SVD5 fixed v.
Shifts and in-plane rotations only.

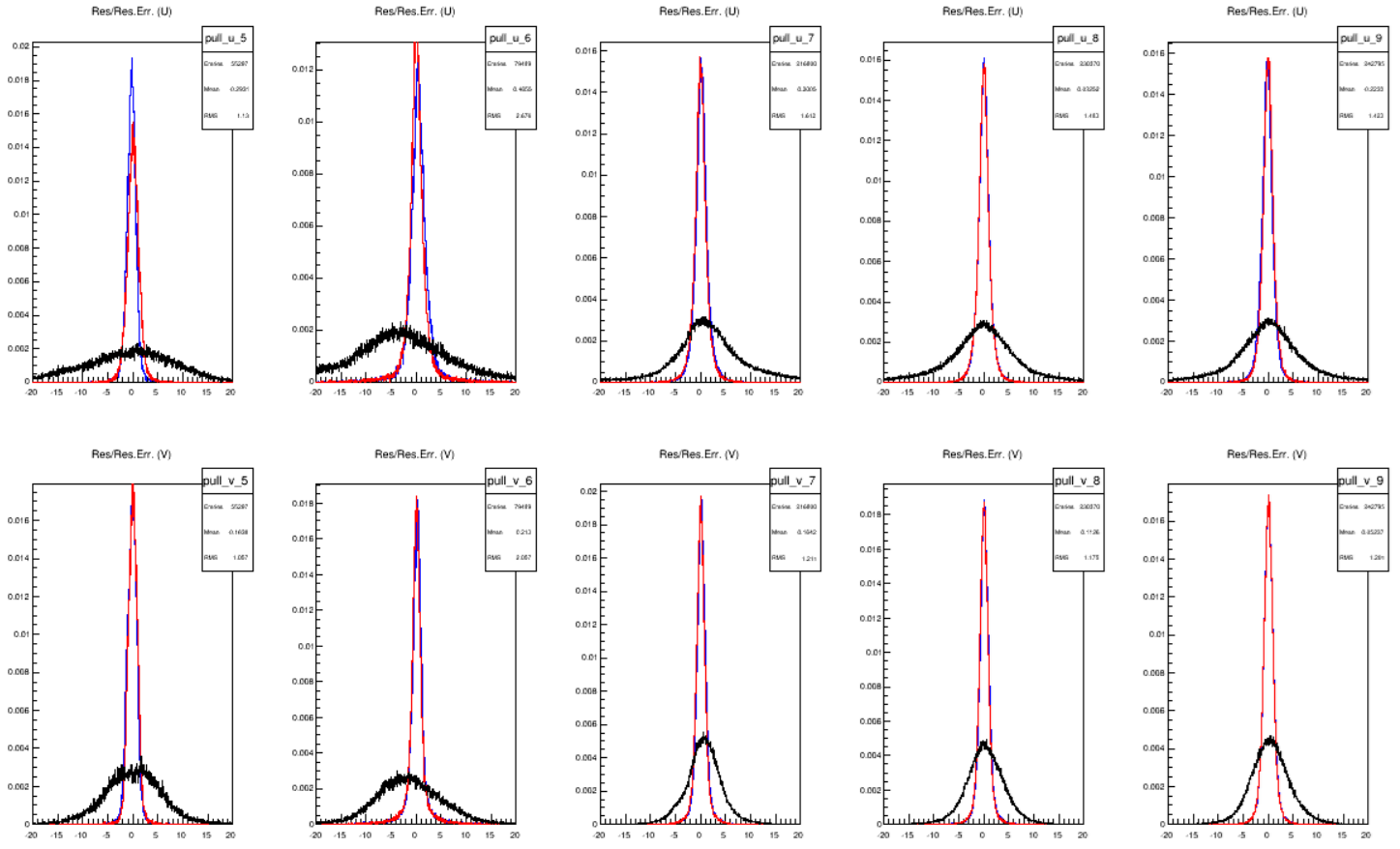
❑ **Before** and **after** Millepede alignment



Backup: Belle 2 VXD alignment

- Pulls per layer. Plots for B=0T

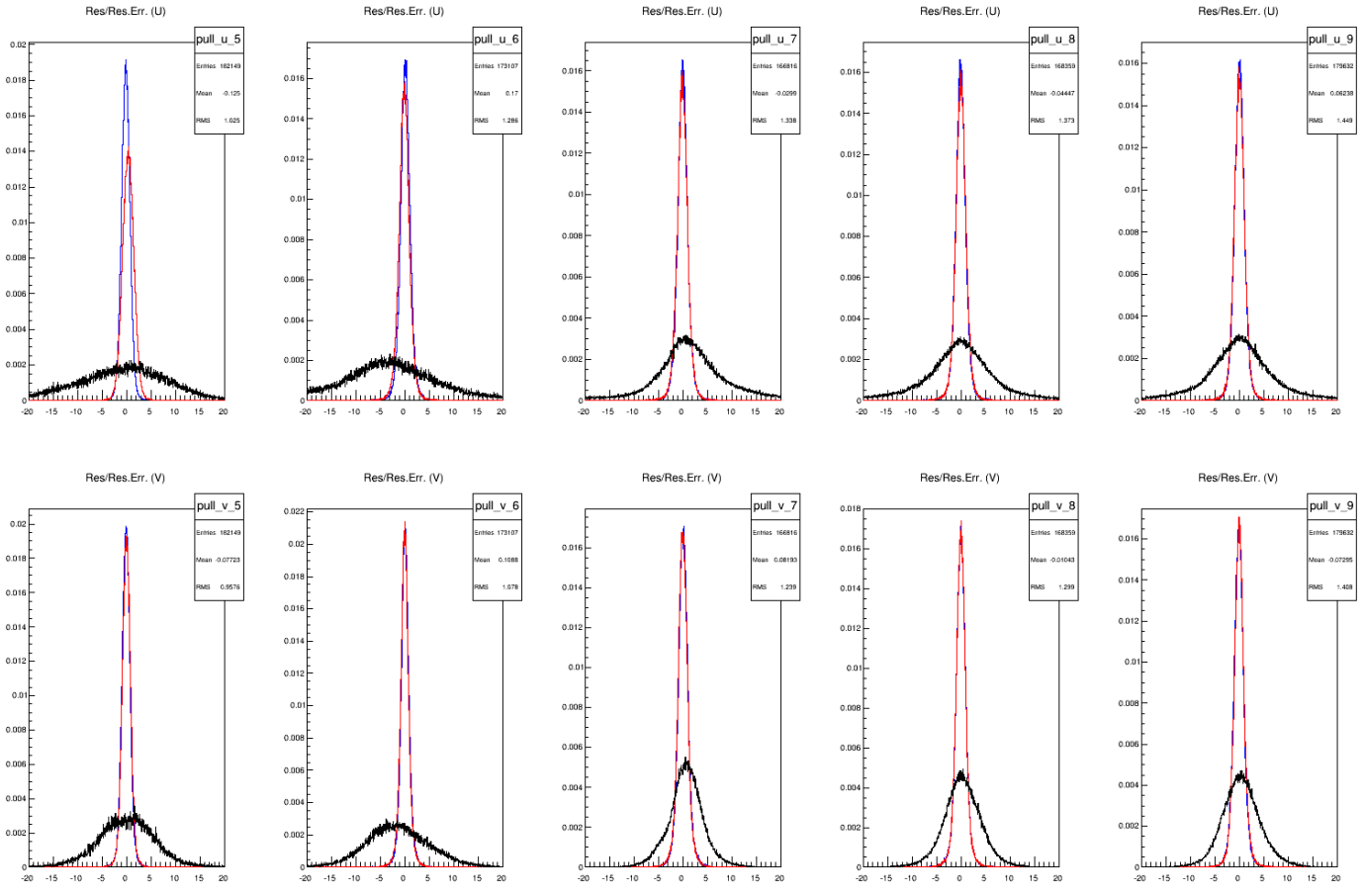
Ideal geometry | misaligned | after alignment



Backup: Belle 2 VXD alignment

- Pulls per layer. Plots for B=1.5T

Ideal geometry | misaligned | after alignment



Low TB track yield

Are we losing tracks?

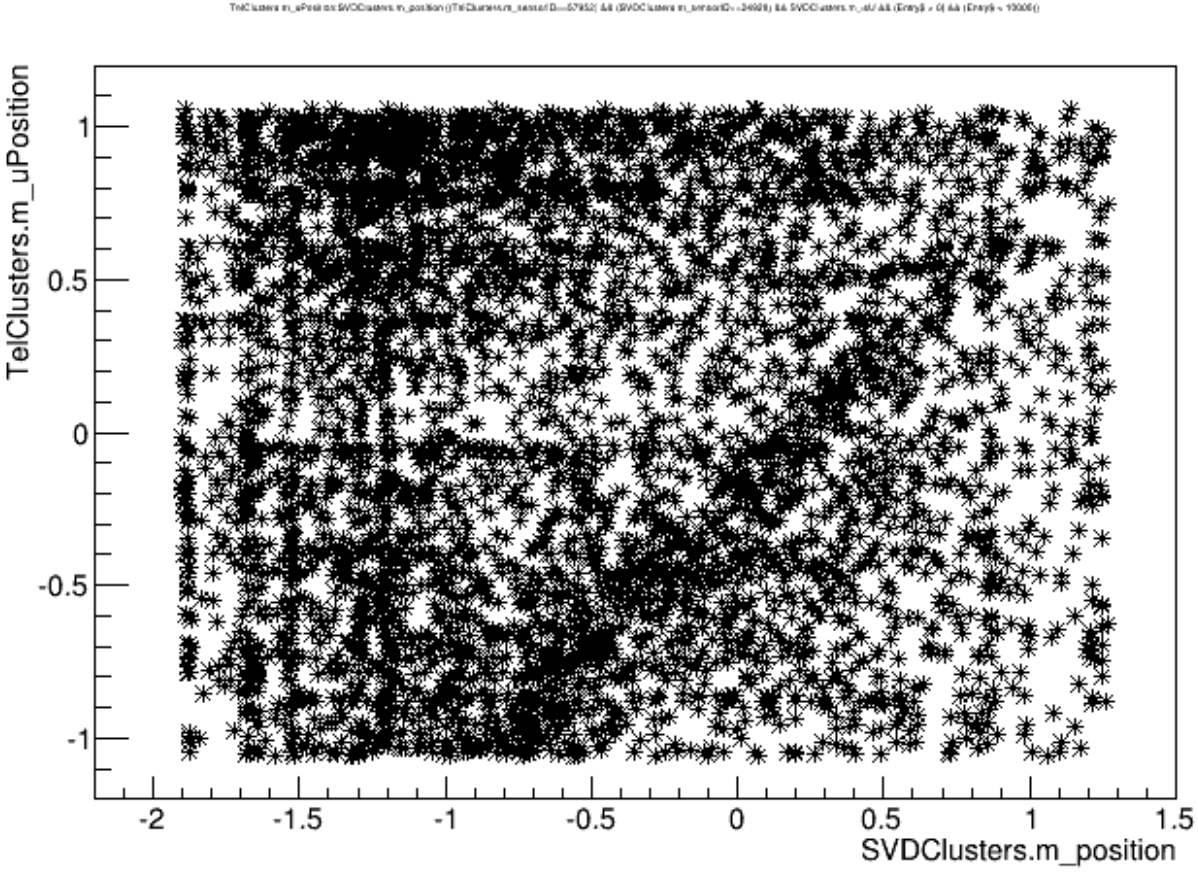
- We are able to find a usable track passing first telescopes and several VXD planes in about 1 of 10^4 events.
- Track finding algorithm doesn't make substantial difference.
- Suspect: Telescope merging.

Telescope merging

- Telescope data are synchronized using event counters and time stamps.
- TLU tags are not handled correctly by EUDAQ.
- Timestamps come from different clocks, so we only look at patterns of event times.
- The merging software detects sync problems and does not merge if out of sync.
- Nevertheless, it seems we can be losing sync in a way we don't see. We're curently after it.

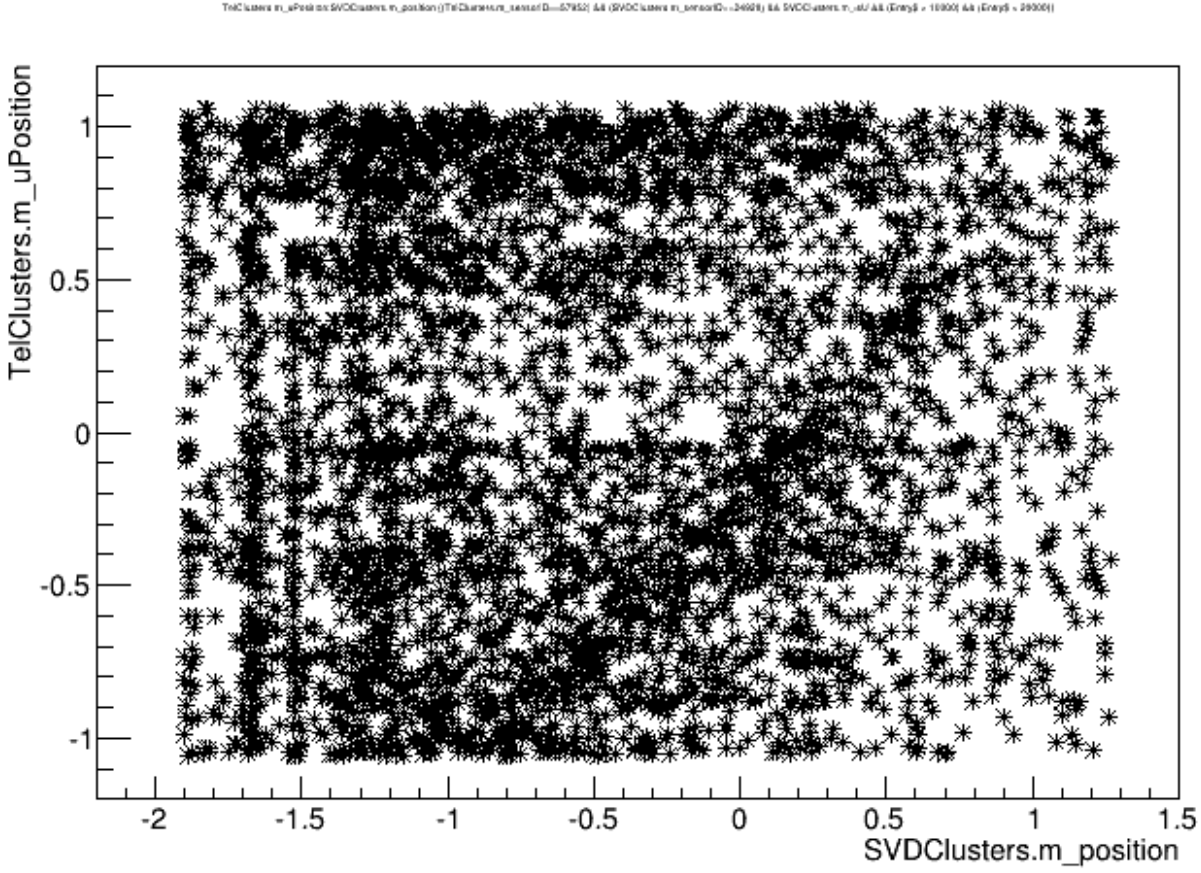
Telescope synchronization during a run

VXD run 470, telescope run 131, events 0 - 10000, correlation between tel3 and SVD3.



Telescope synchronization during a run

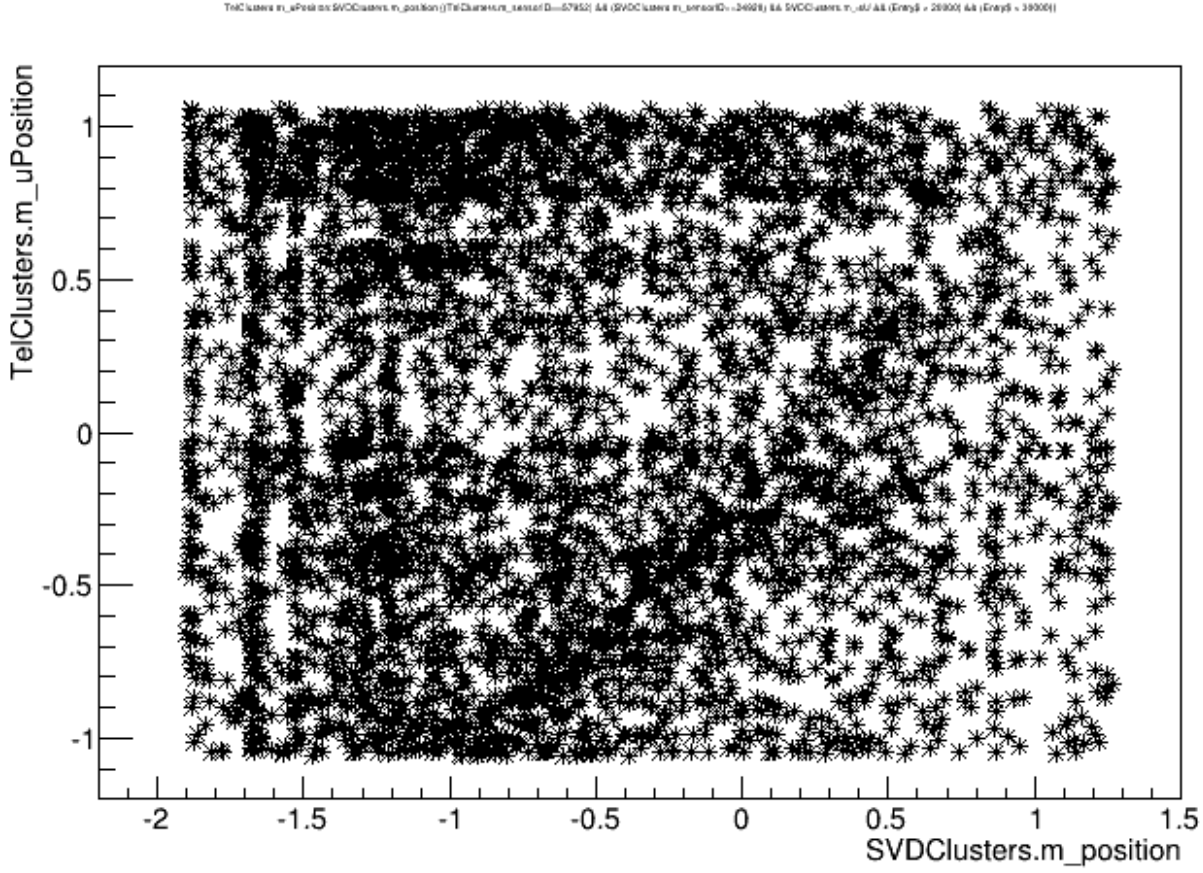
VXD run 470, telescope run 131, events 10000 - 20000, correlation between tel3 and



SVD3.

Telescope synchronization during a run

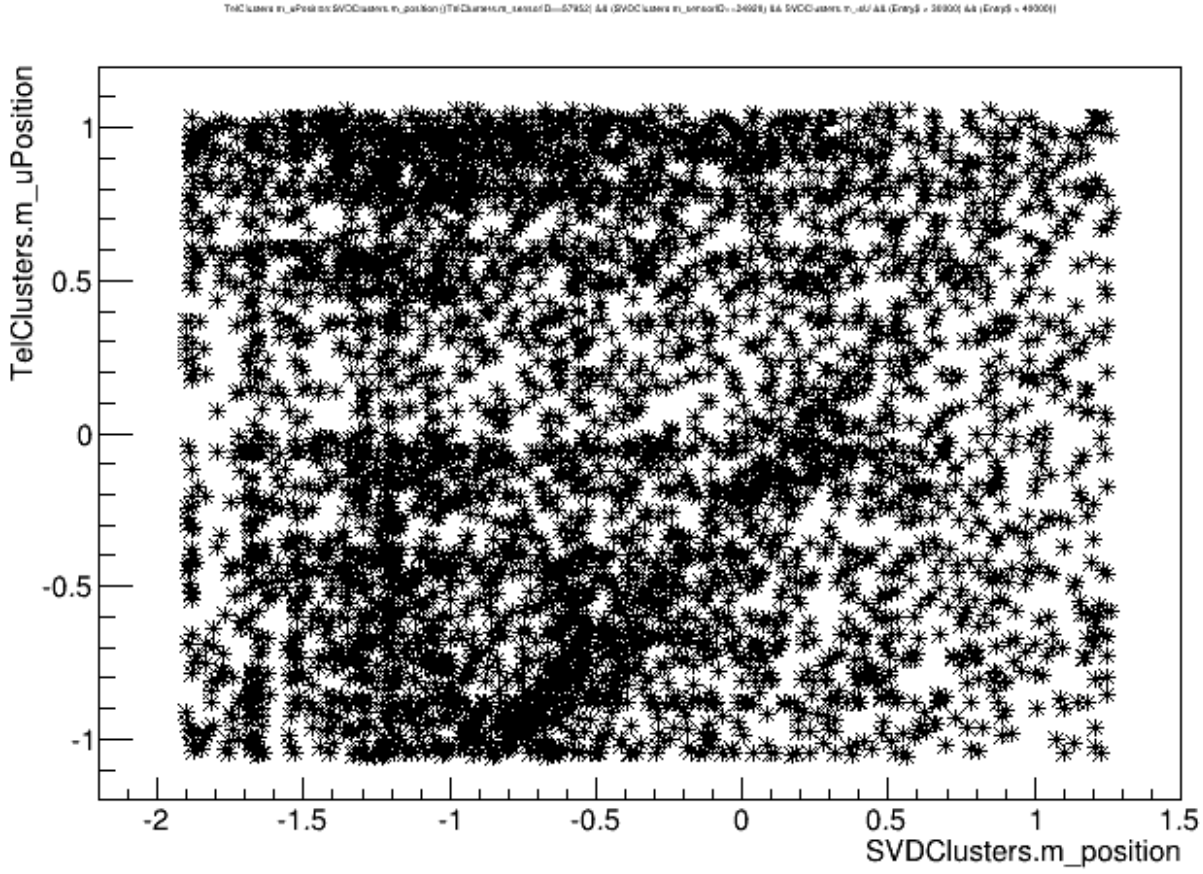
VXD run 470, telescope run 131, events 20000 - 30000, correlation between tel3 and



SVD3.

Telescope synchronization during a run

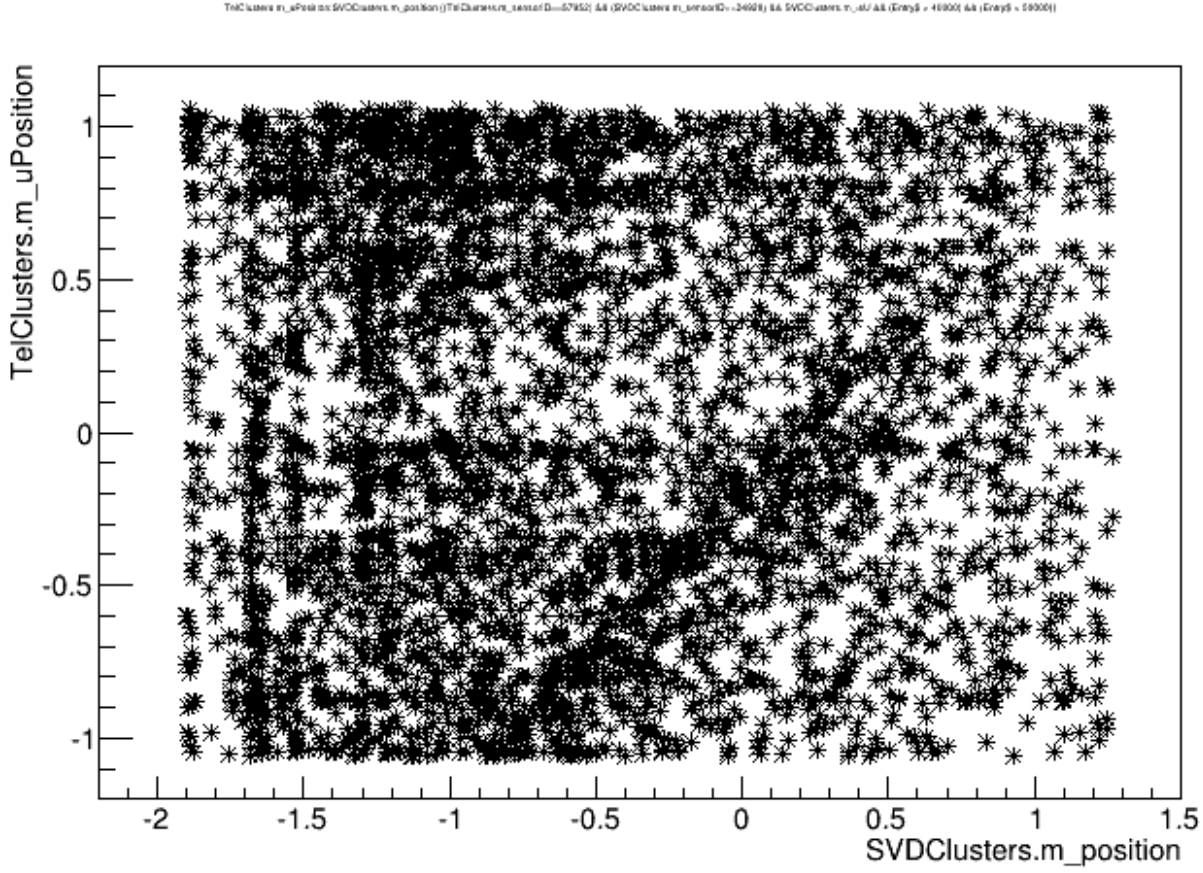
VXD run 470, telescope run 131, events 30000 - 40000, correlation between tel3 and



SVD3.

Telescope synchronization during a run

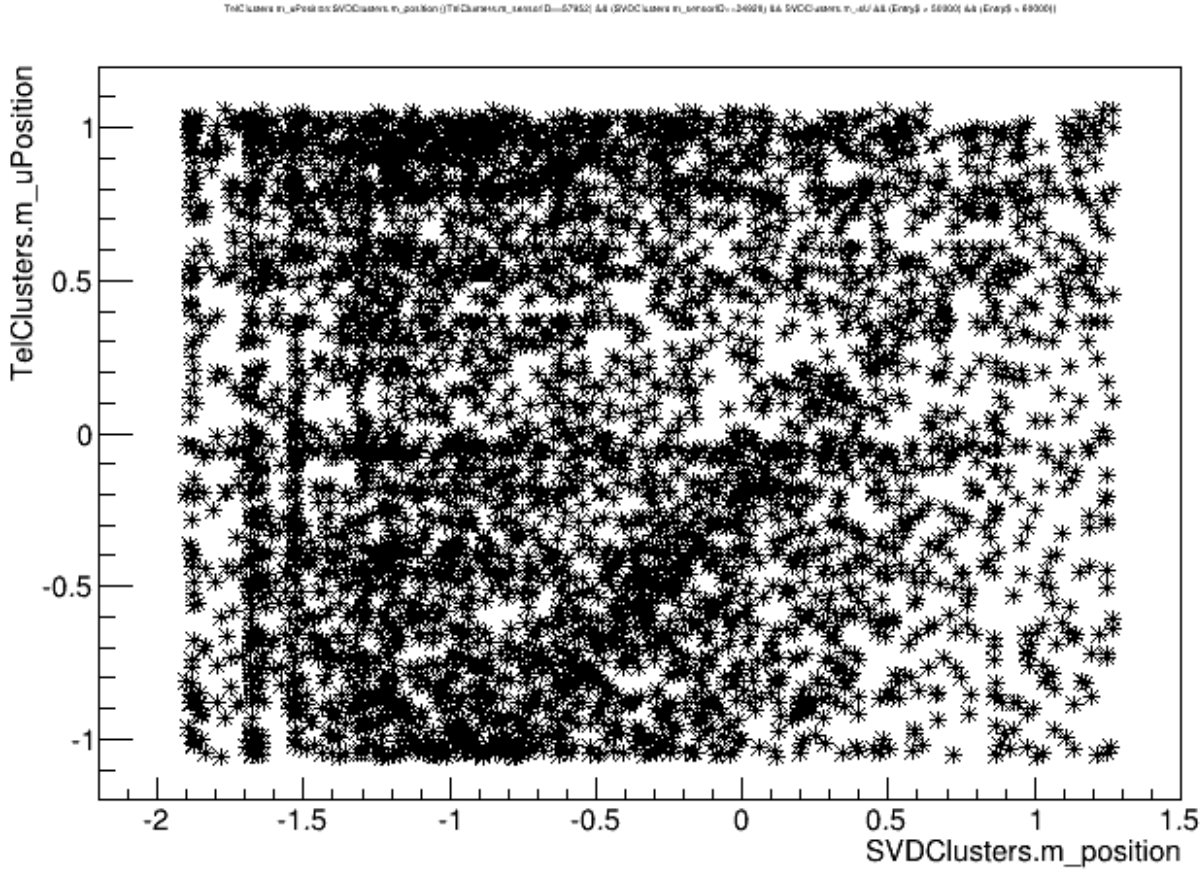
VXD run 470, telescope run 131, events 40000 - 50000, correlation between tel3 and



SVD3.

Telescope synchronization during a run

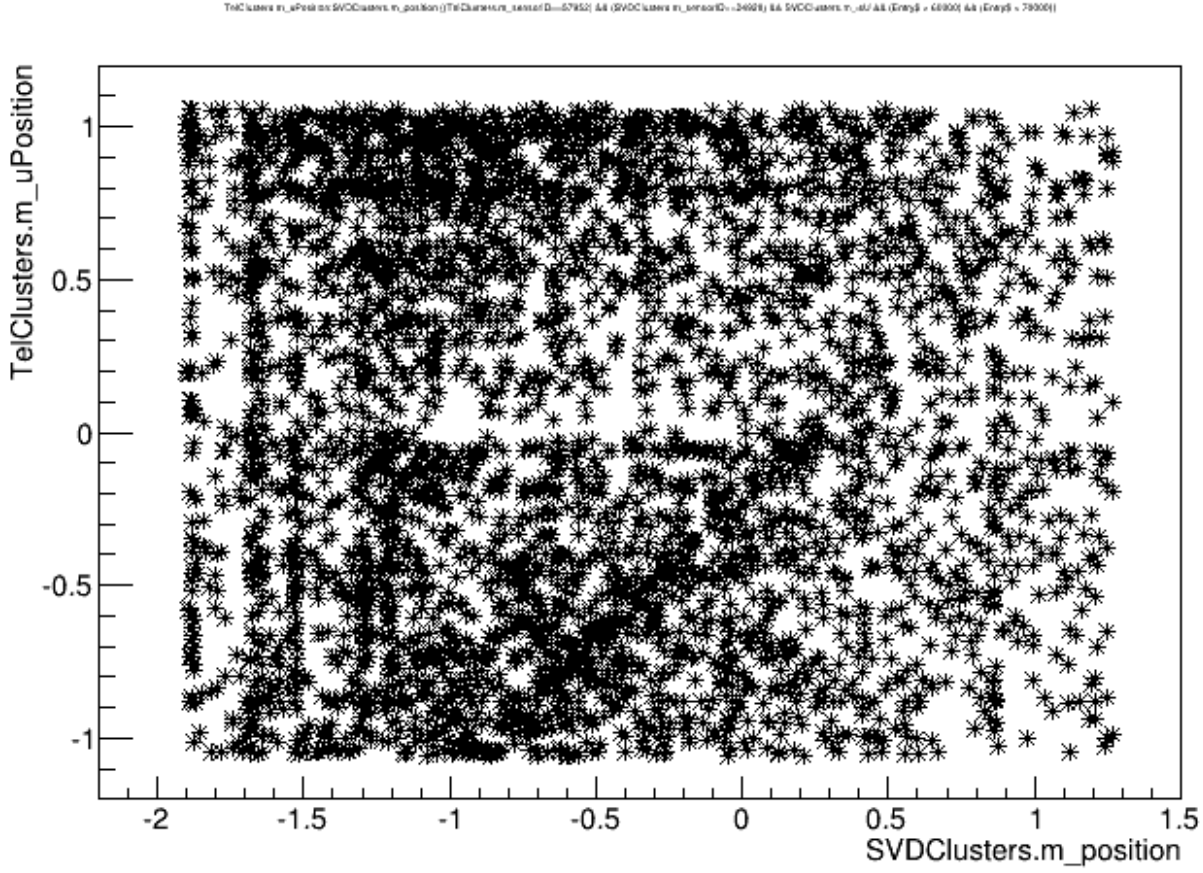
VXD run 470, telescope run 131, events 50000 - 60000, correlation between tel3 and



SVD3.

Telescope synchronization during a run

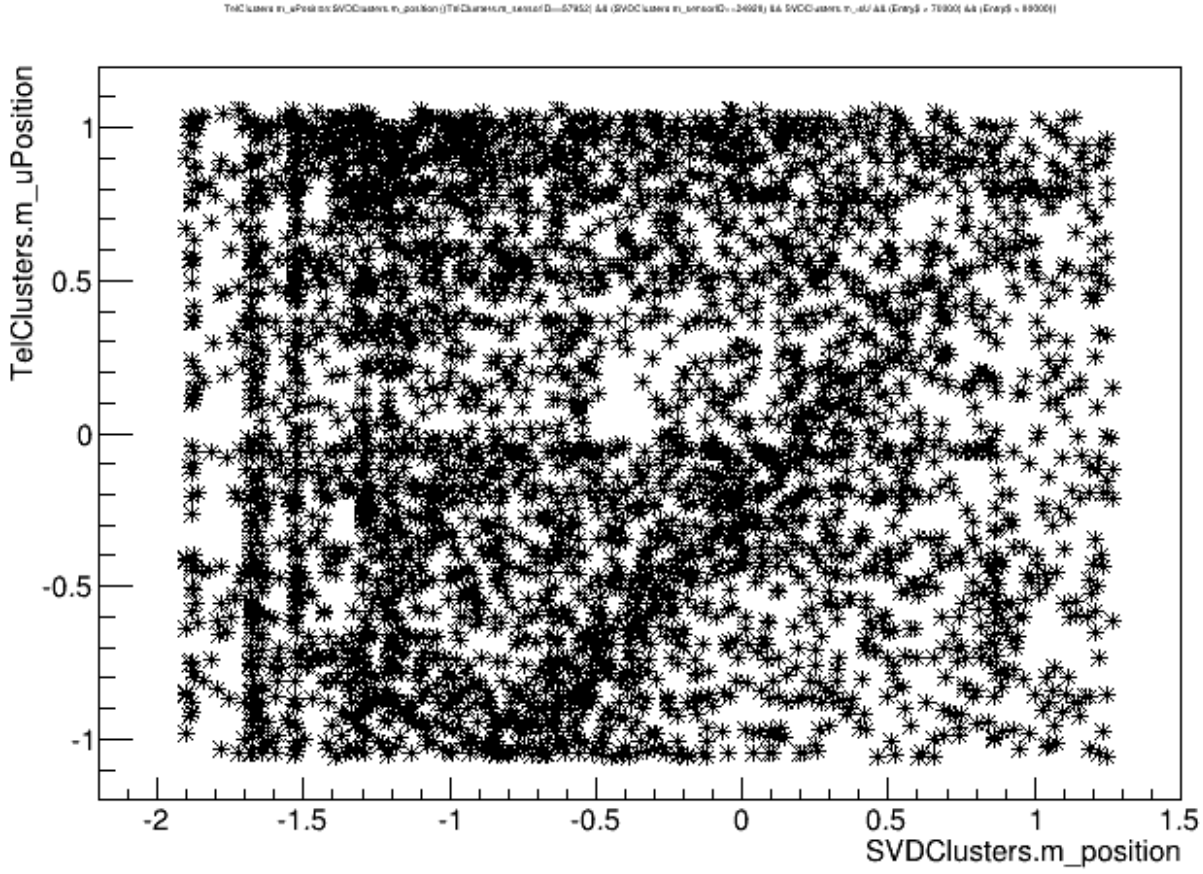
VXD run 470, telescope run 131, events 60000 - 70000, correlation between tel3 and



SVD3.

Telescope synchronization during a run

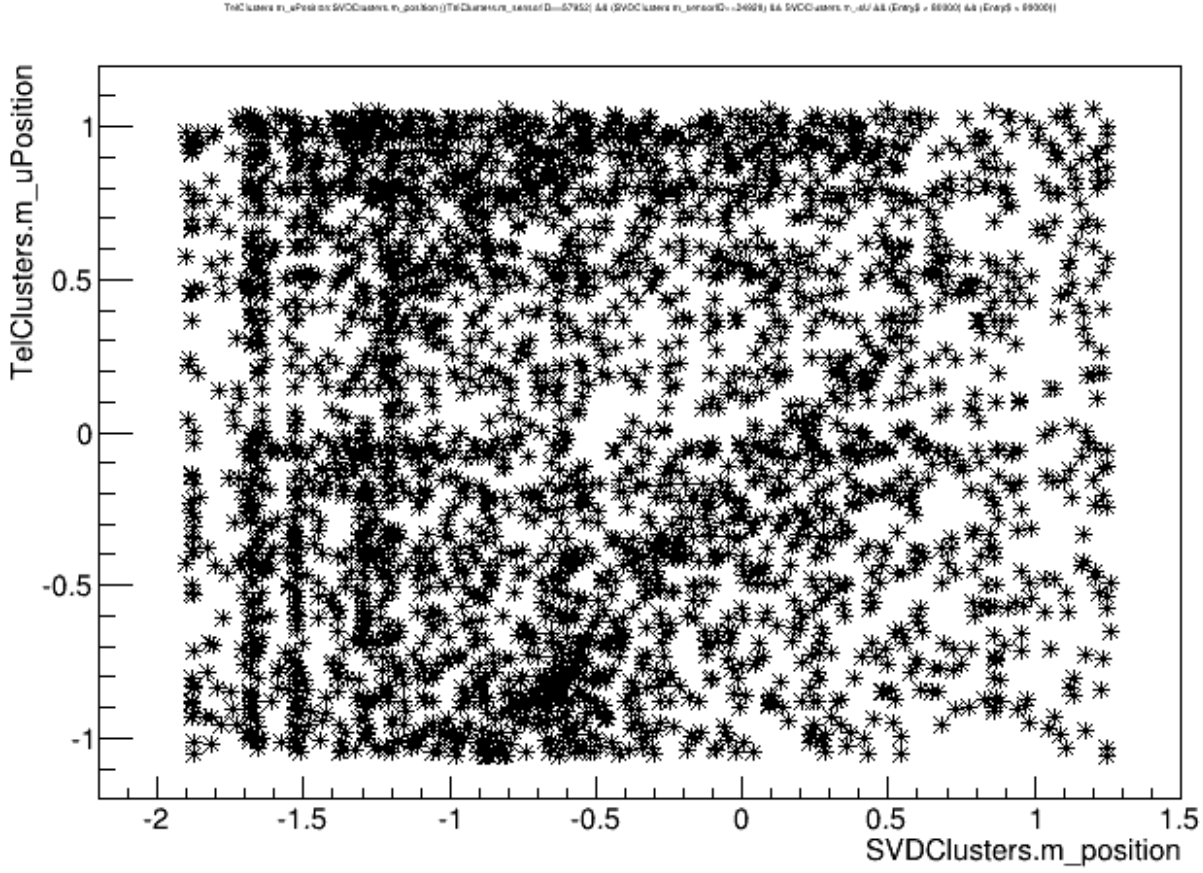
VXD run 470, telescope run 131, events 70000 - 80000, correlation between tel3 and



SVD3.

Telescope synchronization during a run

VXD run 470, telescope run 131, events 80000 - 90000, correlation between tel3 and



SVD3.

VXD (mis-)alignment

Alignment implementation

- Hierarchy of alignable structures, corresponding to subdetector - (half-)layer - ladder - sensor structure. linked with transformation matrices
- Properties of alignable structures, such as whether they are rigid bodies or deformable structures, and how they calculate adjustments to spatial transformations based on alignment parameters (and position).
- Code to read/write (mis-)alignment data files.
- System to shift coordinates of clusters based on misalignment information
- System to apply geometry modificatons to coordinate transforms to reflect alignment information.

The `AlignmentGeometry` class

- The `AlignmentGeometry` class provides access to data on alignment hierarchy and on individual alignables.
- It is implemented as a singleton and is built together with VXD geometry.

Alignment Hierarchy

- The alignment hierarchy is implemented as a `std::map<AlignableID, AlignmentLink>`
- Alignable IDs *are not* the VxdIDs, since the structure may deviate from the layer/ladder/sensor scheme. Strings are used as IDs instead, and only for sensors the IDs are guaranteed to be string representations of the VxdIDs.
- The alignment link contains data on mother of the alignable, including the baseline transform, and IDs of daughters, if any.
- The structure is easily browsable in both directions.
- Initialization takes place in the geometry creator.

Alignables

- The list of alignables is a `std::map<AlignableID, Alignable>`.
- The `Alignable` class holds current displacement/alignment/misalignment parameters, and methods to construct a 3D tranform based on alignable type.
- The class features transforms from local to global coordinates (and v.v.), so that position-dependent transformations for deformable structures can be supported.
- The list is initialized in the geometry creator based on the displacement xml file.
- Alignment or misalignment modules fill the corresponding data.

Misalignment

I will spend a second on misalignment implementation.

The Misalignment module

- The module's basic task is to provide input of misalignment data, either from an xml file or randomly simulated.
- These data are input to Alignables.
- A method providing a (possibly position-dependent) 3D transform) will be provided for each sensitive plane and retrieved by sensor VxdID via GeoCache.

RecoHit construtor

- The RecoHit constructor will adapt the position of the generating cluster based on the misalignment information in the GeoCache and direction of the generating MC track (taken from a TrueHit).

Conclusions

- VXD alignment effort is progressing, current work focuses on covering CDC alignment.
- VXD testbeam alignment works, but limited by low track yield.
- Basic implementation of VXD support for (mis-)alignment is ready.

Thanks

Thank you for attention.