Latest reference on Muon Colliders: arXiv:2303.08533

Tracking Using ACTS For Muon Collider Detector

Karol Krizka

July 11, 2023

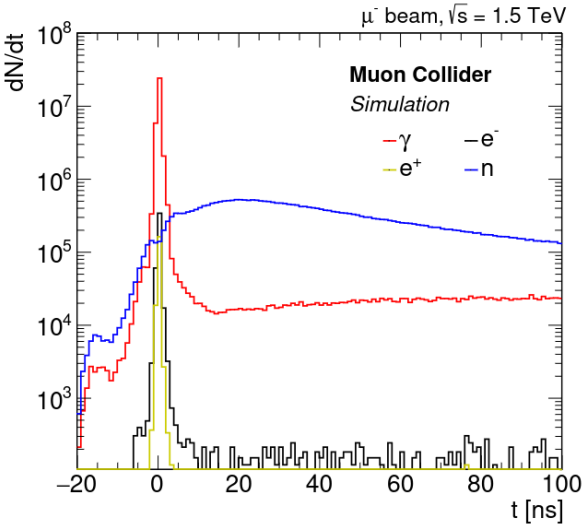
UNIVERSITY OF BIRMINGHAM

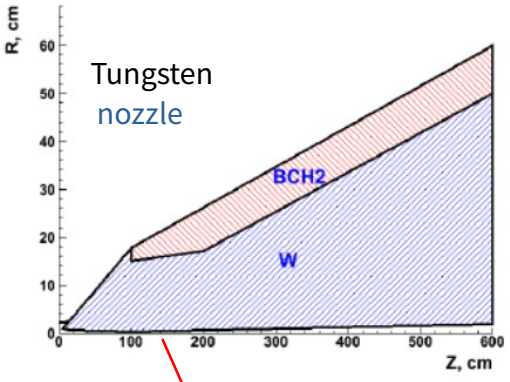
ECFA Obj Reco

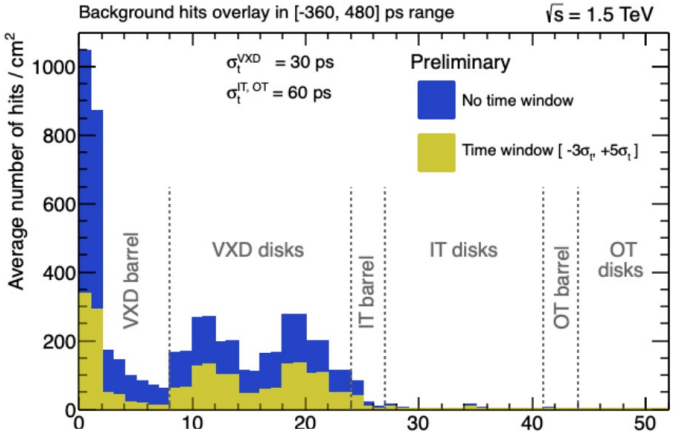
Beam Induced Background

arxiv:2105.09116

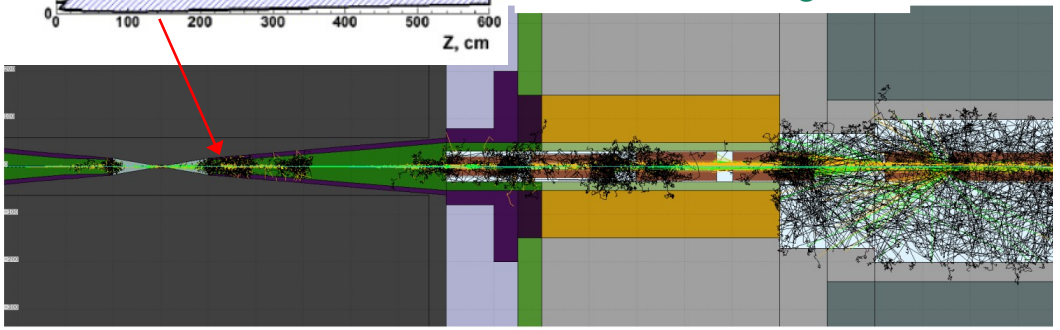
- **BIB = muon beam decay and strike the detector**
- **Several main mitigation**
  - 10° tungsten nozzle to shield from beam decay products
  - Precision timing information from detectors



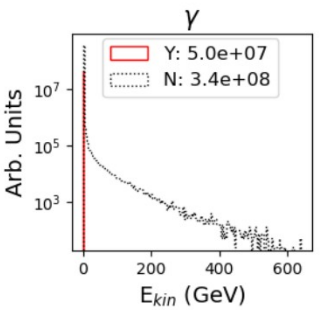


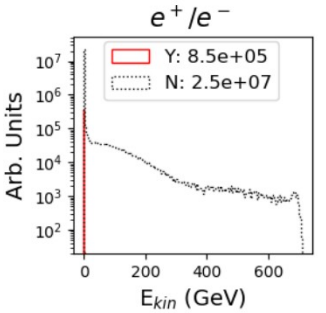


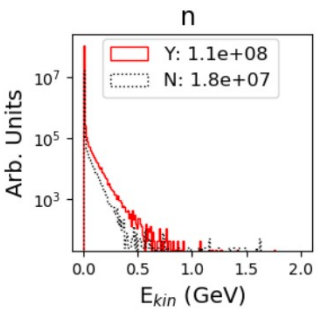
Tracker after timing cuts.

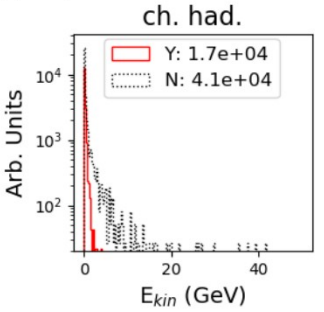


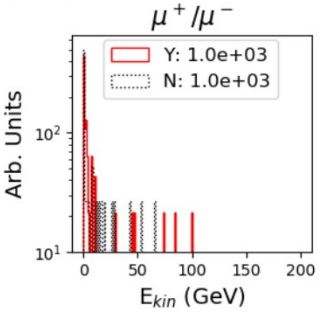
FLUKA simulation of BIB before reaching the detector.





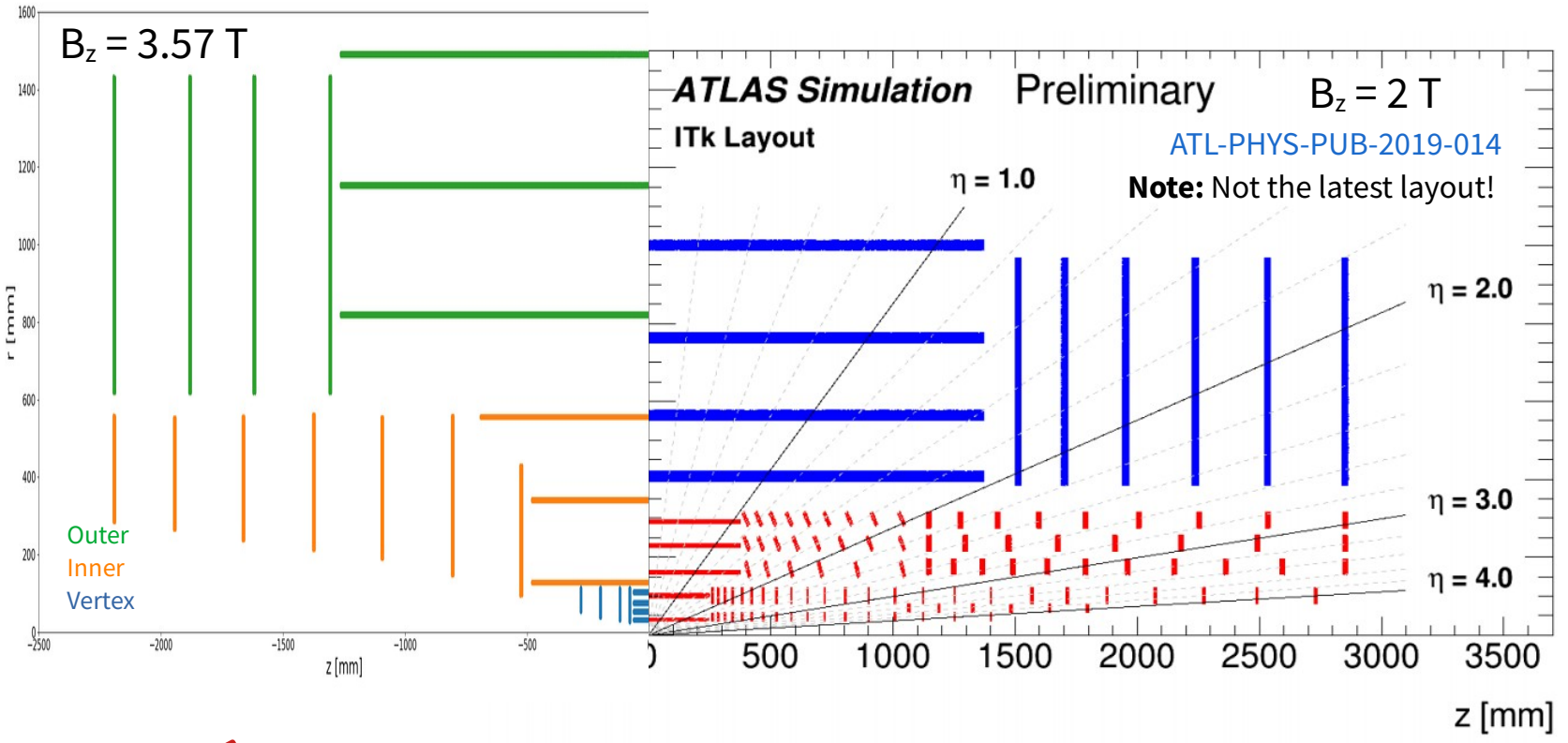






Particle energy spectra with (Y) and without (N) nozzle.

The Scale of BIB



Hit density **after timing cuts** **10x HL-LHC**

	ITk Hit Density [mm^{-2}]	MCC Equiv. Hit Density [mm^{-2}]
Pix Lay 0	0.643	3.68
Pix Lay 1	0.022	0.51
Str Lay 1	0.003	0.03

ITk Pixels TDR, ITk Strips TDR

All-Silicon Tracking Detector

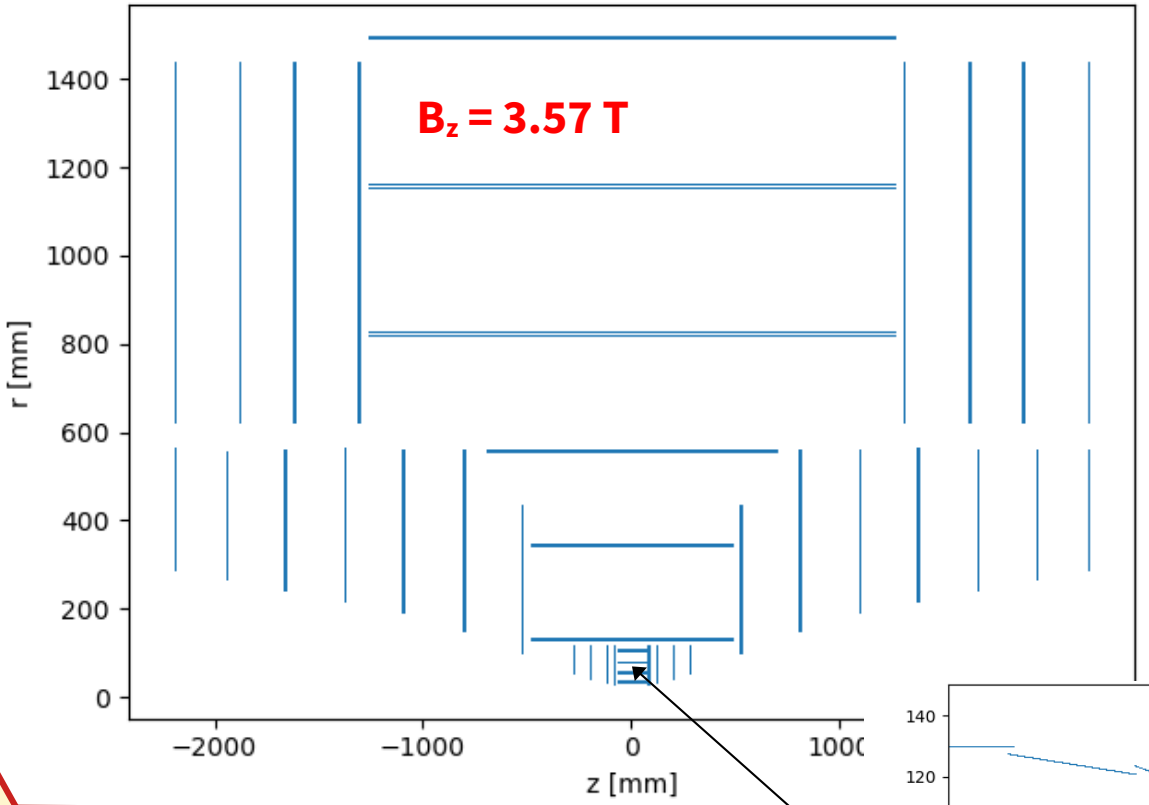
Details

Outer Tracker (OT)

- micro-strips
- 50 µm x 10 mm
- σ_t = 60 ps

Inner Tracker (IT)

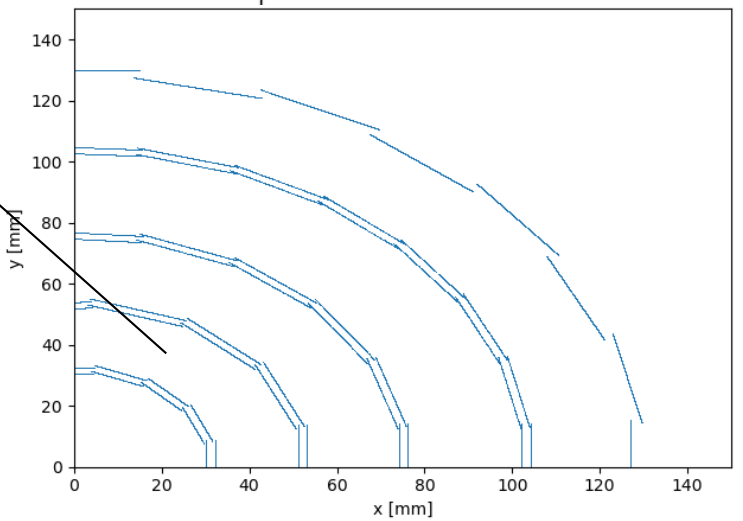
- macro-pixels
- 50 µm x 1 mm
- σ_t = 60 ps



4D tracking critical

Vertex Detector (VXD)

- pixels
- 25 µm x 25 µm
- σ_t = 30 ps
- double layers



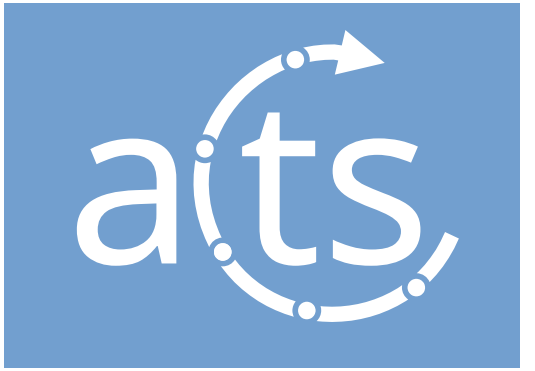
Software Stack

https://github.com/MuonColliderSoft/

- **Original software based on iLCSoft.**
  - Many packages forked and modified .
- **Ongoing migration to Key4HEP (details).**
  - Centrally supported software for turnkey HEP with many users.
  - Still using iLCSoft processors (backwards compatibility in key4hep).
  - Package management done via Spack.
- **Tutorial: https://mcdwiki.docs.cern.ch/**

A Common Tracking Software

- **ACTS is a standalone library for tracking algorithms**
- **Dedicated team working on advancing tracking algorithms**
  - Tracking is hard!
- **Allows us explore alternate algorithms**
  - Triplet-based seeding optimized for high multiplicity environments
  - Ongoing work to incorporate ML-based algorithms
- **Code optimization come for free**
  - Good software is even harder than tracking!
  - Also explores modern computing architectures (ie: GPU's)



https://github.com/acts-project/acts

Fit Library	Kalman Filter Execution Time
ACTS	0.5 ms / track
iLCsoft	100 ms / track

ACTS Versions

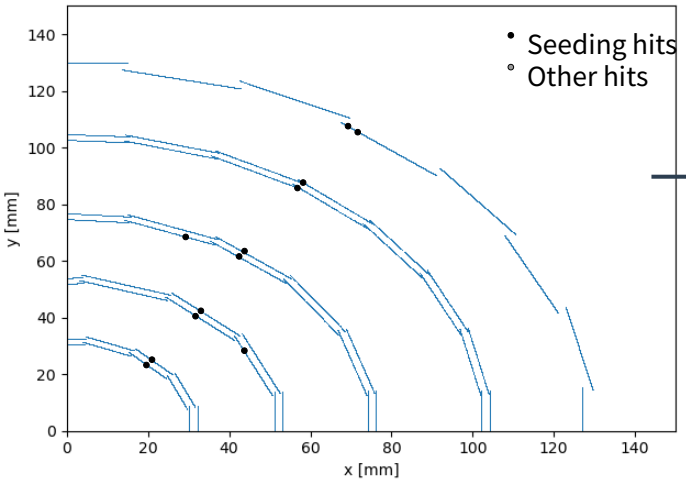
- **ACTS v13.0.0: Latest when IMCC implemented ACTS tracking.**
- **ACTS v27.1.0: Latest today.**
  - Improved material map tuning and validation.
  - CKF settings auto-tuning is now part of their tools.
  - Support for 4D tracking (TBC).
  - GraphNet-based pattern recognition.

Why are you so far behind?

- ACTS keeps breaking API every release… hard to keep up.

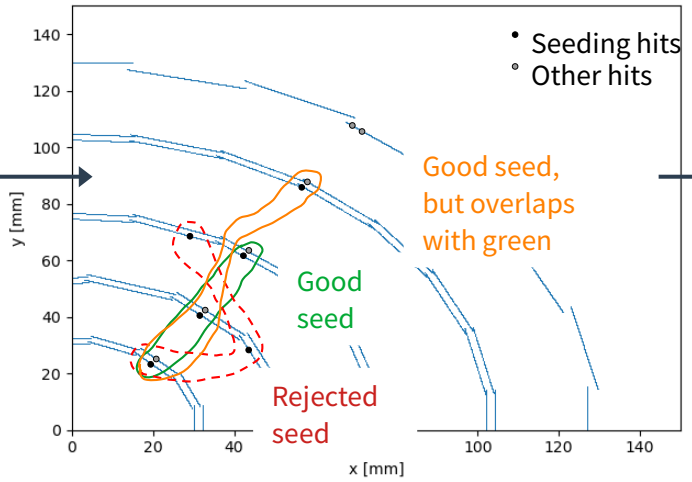
Triplet Seeded CKF

Global Hit Selection
ie: timing, *

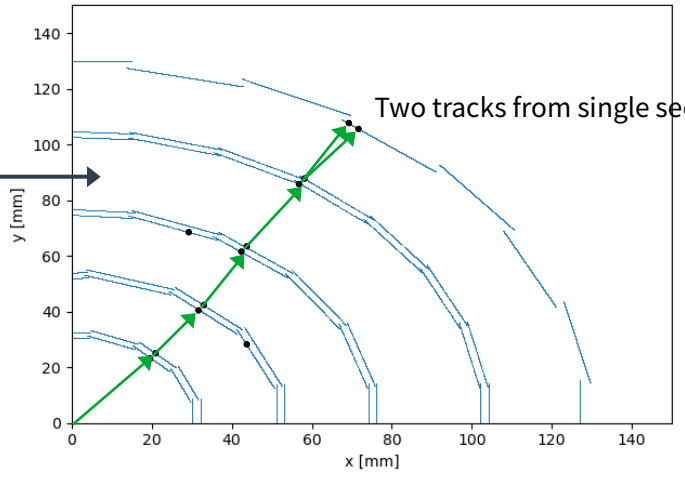


* Currently not leveraging double layers.

Seed Finding
Initial parameters for CKF



Combinatorial Kalman filter
Track fit



Remove BIB hits

Pattern Recognition

Track Fit

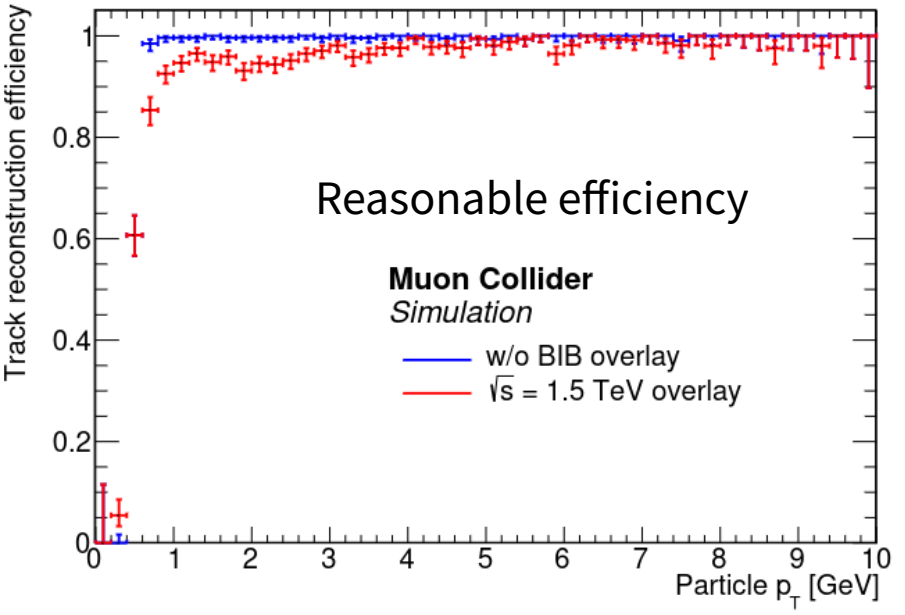
Similar algorithm used by ATLAS.

aka optimized for high hit multiplicity

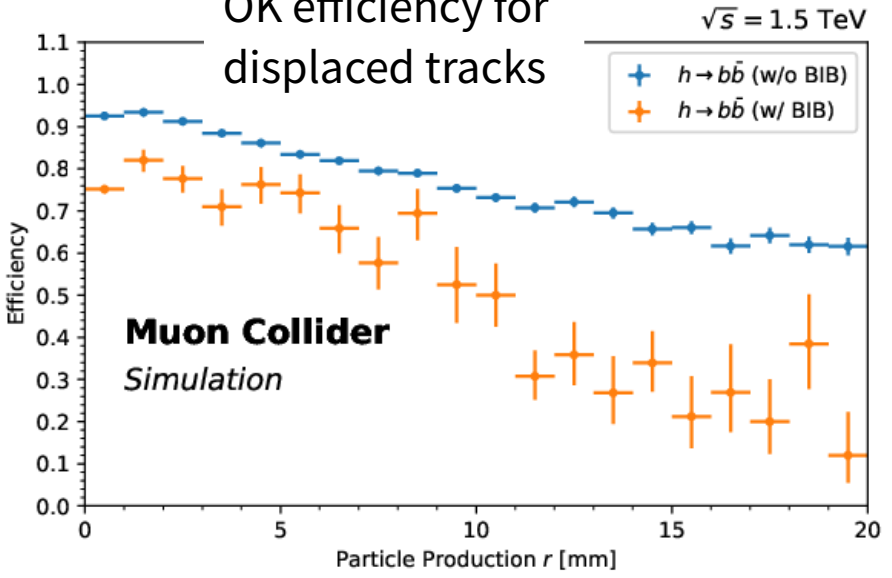
CKF (ACTS) Tracking Performance

Details

- **Seeded CKF runs in ~4 min / event.**
- **Parameters need to be optimized.**
  - Seeding: *very narrow collision region*
  - CKF: No branching allowed



OK efficiency for displaced tracks



Fake track removal
(optimized with evolutionary algorithms)

Eff WP	Fakes / event
90%	3900
80%	0.13
70%	0.06

ACTS Components

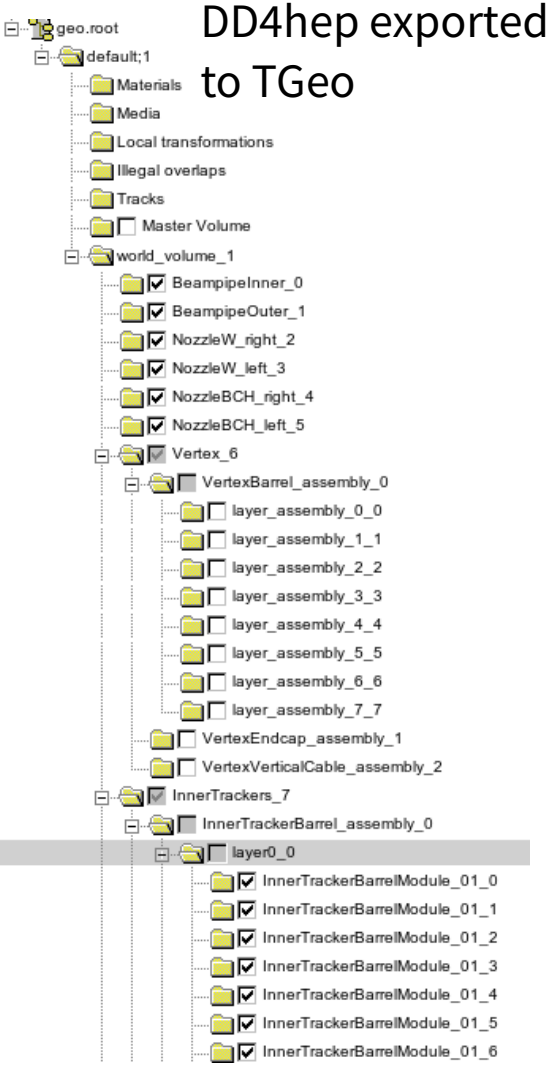
- **Core library**
  - The core track EDM and reconstruction classes as a library
  - To be used by *experiments* inside their *framework*.
- **Plugin libraries**
  - Optional libraries that supplement the core library.
- **Applications**
  - Fast simulation, running algorithms, material map, auto-tuning…
  - Your detector can be loaded if supported via official plugin.

Detector Geometry Descriptions

ACTS supports several ways to load detector geometry.

- **Manually describe the geometry**
  - Very tedious for a real detector
  - Duplicates work from simulation detector description
- **DD4hep plugin**
  - Can load existing description for detectors that use DD4hep
  - *Need a specific detector tree structure* to extract logical information
- **ROOT's TGeo plugin**
  - Common, but simple, format (ie: DD4hep can export to TGeo)
  - Logical information is extracted from geometry
    - Slow (*enable compiler optimizations*)

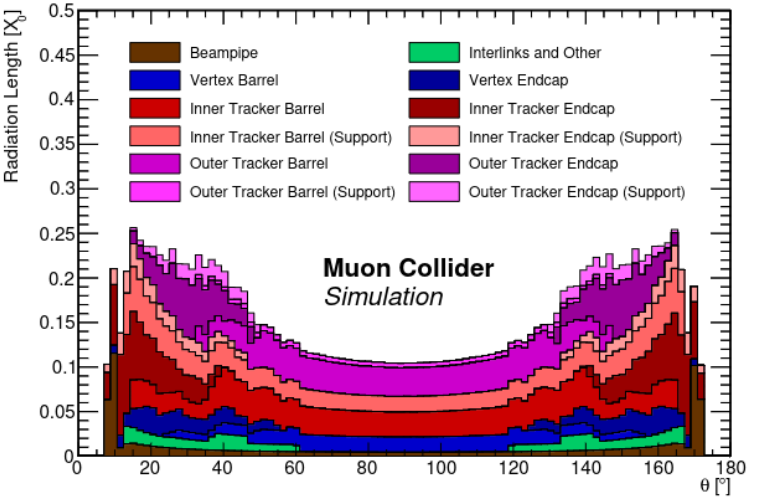
DD4hep exported to TGeo



Two Tracking Detector Geometries

1) Loading geometry for geantino scan to determine material.

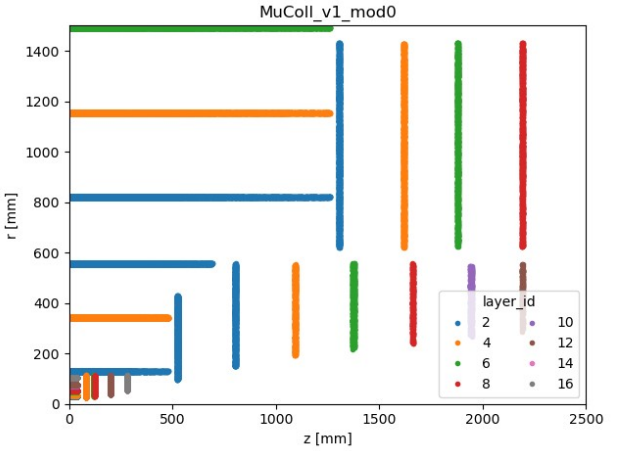
- Load using DD4hep ACTS plugin.
- Includes all the details on supporting structures.
- Geantino scan can be performed using ACTS tools.



* separation into components is manual.

2)Loading geometry to determine sensitive layers for tracks.

- Load using TGeo ACTS plugin.
- A lot of manual work (and validation) required
  - Need to specify bounding boxes for layers.
  - Need to create a static map from sim detector ID to ACTS detector ID
  - Validation can be partially performed using ACTS tools.



Material Map

ACTS caches the detector material as a histogram.

- **Three step approach to create the material map:**

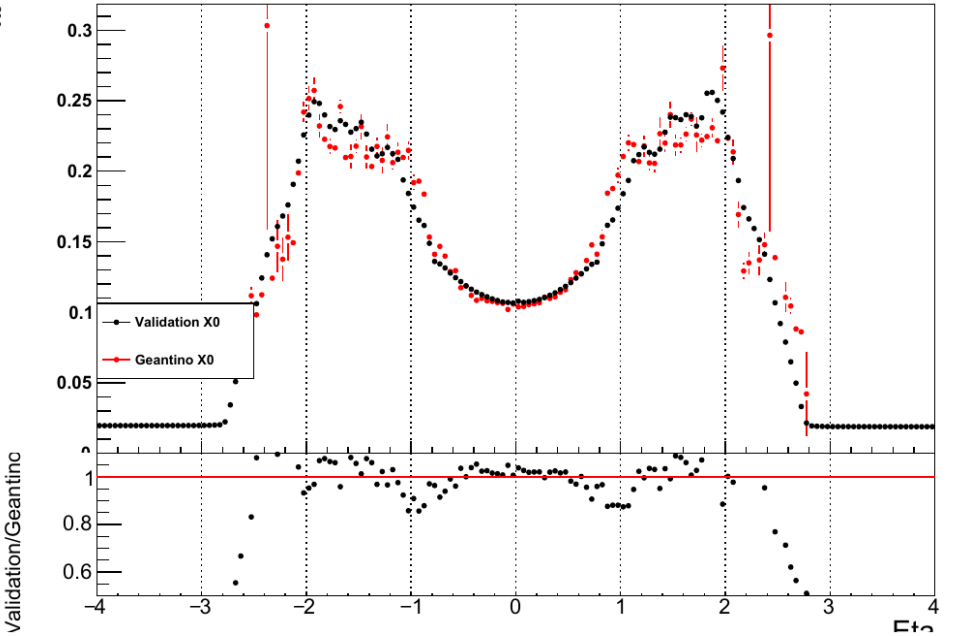
  1) Run a geantino scan to determine material (ACTS tool).

  2) Run a bunch of ACTS scripts to create the map.

  - Blindly using defaults for most settings (ie: bin sizes).

  3) Validate (ACTS tools)

- **Plenty of documentation**
  - Generic ACTS documentation.
  - Muon Collider Detector specific instructions.
- **Rather annoying to run due to *TGeo*.**
  - Makes *detector layout studies tedious*.

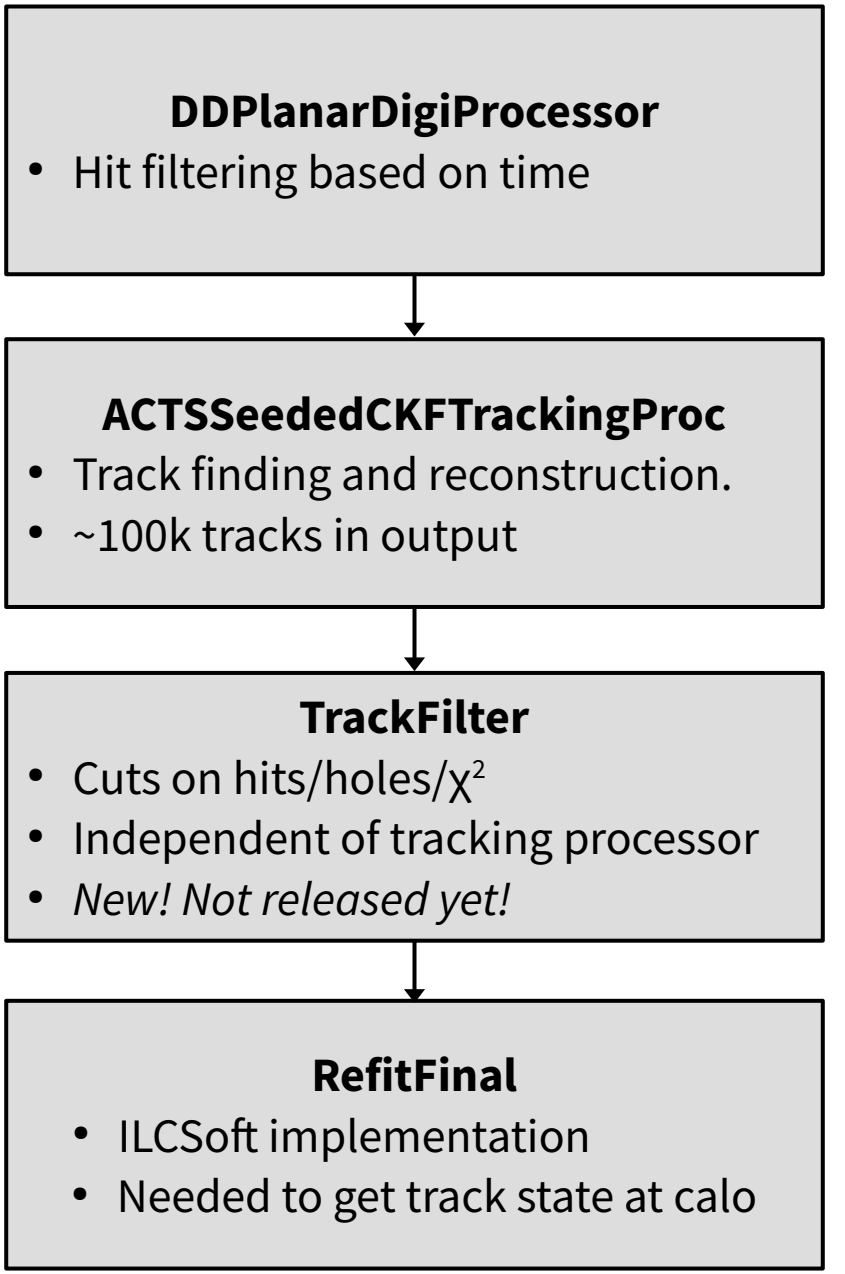


Tracking Using Marlin

- **ACTS as a library to create a generic Marlin processor**
  - Pass LCIO objects. Drop-in for existing tracking processors.
- **Key steps to code:**
  - Geometry loading*.
  - LCIO → ACTS hit objects.
  - Running of algorithms*.
  - ACTS → LCIO track objects.

* Copy-paste code from ACTS applications.

https://github.com/MuonColliderSoft/ACTSTracking



TrackPerf: Package for Common Tracking Plots

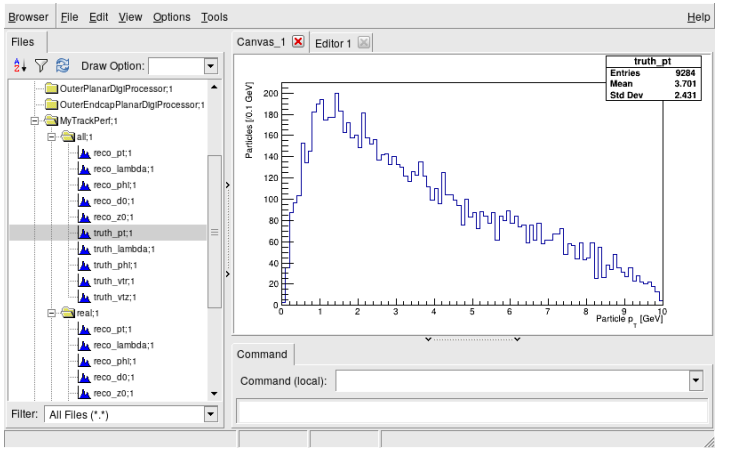
- **Common way to compare the different tracking approaches**
  - Started a new package TrackPerf (unreleased)
- **Functionality**
  - Input: EVENT::Track collection
  - Output: all the histogram you would want
    - Parameters of truth particles matched/not-matched/all
    - Parameters of tracks matched/not-matched/all
    - Resolution plots of all parameters
  - Configurable selection on truth particles
    - Default: charged, decay in tracker, left tracker
    - Option to filter for particles from b-meson decay (TODO)
  - ROOT Ttree for custom studies (TODO)

```
registerProcessorParameter("MatchProb",
                           "Minimum matching probabilty to be considered a good track-mc match.",
                           _matchProb,
                           _matchProb);

registerInputCollection( LCIO::MCPARTICLE,
                         "MCParticleCollection" ,
                         "Name of the MCParticle collection"  ,
                         _mcpColName,
                         _mcpColName
                         );

registerInputCollection( LCIO::TRACK,
                         "TrackCollection" ,
                         "Name of the Track collection" ,
                         _trkColName,
                         _trkColName
                         );

registerInputCollection( LCIO::LCRELATION,
                         "MCTrackRelationCollection" ,
                         "Name of LCRelation collection with track to MC matching",
                         _trkMatchColName,
                         _trkMatchColName
                         );
```



Missing plotting scripts!

Tracking Core TODO

- **Update to latest version of ACTS.**
  - Just too far behind now.
- **Run CKF in both directions and merge tracks.**
  - Currently missing the inner most layer as we only go “inside-out”.
- **Extrapolate tracks to calorimeter.**
  - Needed for particle flow. Currently refit using “slow” iLCSoft tracking.
- **Convert DD4hep geometry tree to follow ACTS assumptions.**
  - Easier material maps. Can use ACTS tools. Faster geometry loading.
- **Port to key4hep…**
  - Very interested in collaborating! No reason to be detector specific.

Final Thoughts

Tracking using ACTS is awesome.

- Easy*: *Basics* implemented by following ACTS examples.
- Fast: Same algorithms are orders of magnitude faster than iLCSoft.
- Works: See plots above. Also being used in other experiments.

A few downsides…

- API breaks every release. Needs work to keep up.
- * Easy = once you know how it works.

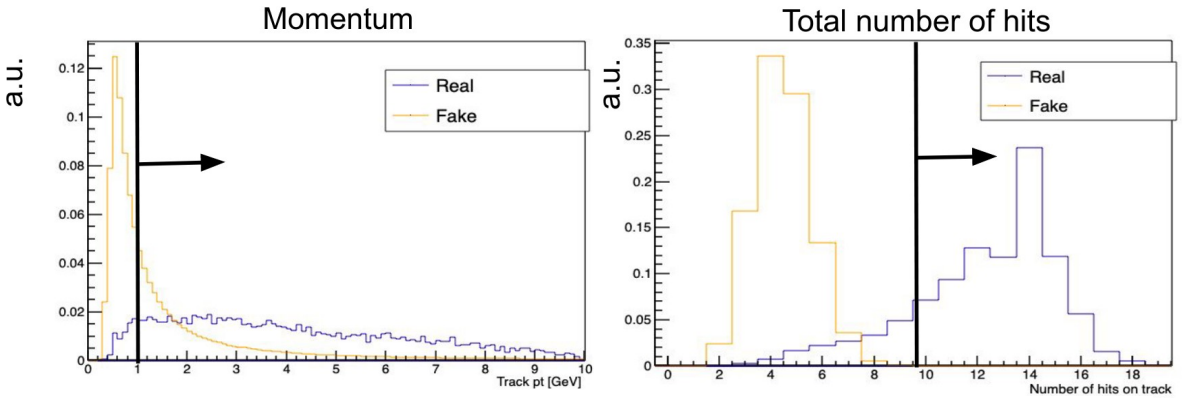
Interested in a common key4hep ACTS processor.

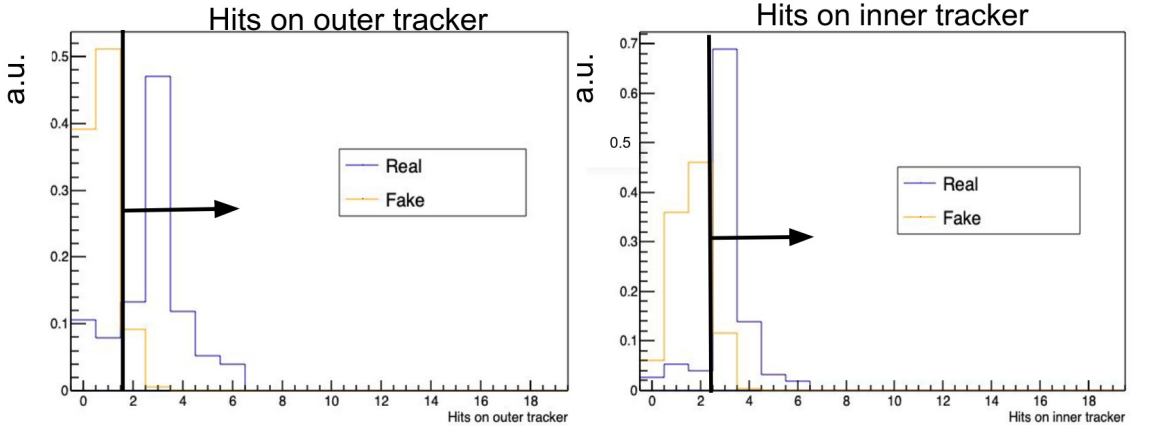
BACKUP

Rejecting Fakes

Details

- 100k fake tracks / event
- reduce to < 1 fake / event
- Still missing a few handles
  - χ^2, N_{holes}, timing
- Implemented as an (unreleased) processor





Efficiencies