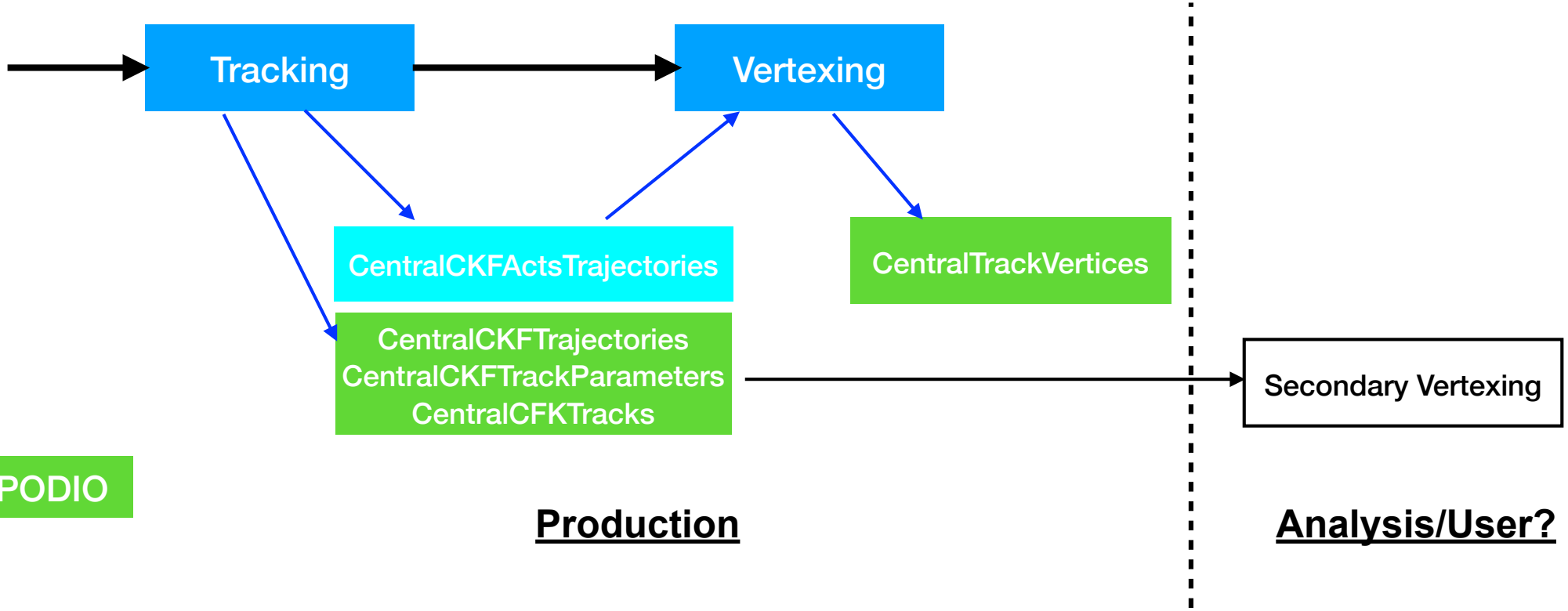
Vertexing at ePIC

Harsimran Singh, Khushi Singla, Lokesh Kumar (Panjab Univ.)

Xin Dong (LBNL)
Rongrong Ma, Joe Osborn (BNL)
Sooraj Radhakrishnan (KSU/LBNL)

Tracking/Vertexing Workflow



Overall Status:
- Basic workflow in place
- What's in place?
  > All basic components
- What's missing?
  > edm4eic::Vertex associatedParticles not filled

Workflow Inputs/Outputs
- Inputs: CentralCKF(Seeded)ActsTrajectories
- Outputs: CentralTrackVertices (edm4eic::Vertex)

Near Term Goals:
- primary-vertexing benchmark for TDR
- fill in missing associatedParticles in output

Long Term Goals:
- algorithm/parameter tuning for different classes of events
- MC/generated vertices and associations
- secondary vertexing

Vertexing Algorithm and edm4eic Vertex

IterativeVertexFinder

- Input: CentralCKFActsTrajectories / CentralCFKSeededActsTrajectories
- default 1D ZScan for vertex seeding (options to use beam line constraints, not in default)
  - logPt weight used with pT_min = 0.4 GeV/c
- output written to CentralTrackVertices (edm4eic::vertex)
  - *associatedParticles not filled at this moment*

https://github.com/eic/EICrecon/blob/main/src/global/tracking/tracking.cc

```
210         app->Add(new JOmniFactoryGeneratorT<IterativeVertexFinder_factory>(
211                 "CentralTrackVertices",
212                 {"CentralCKFActsTrajectories"},
213                 {"CentralTrackVertices"},
214                 {},
215                 app
216                 ));
```

"CentralCKFSeededActTrajectories" works well too, want to update for default in main branch

```
460     ## ============================================================================
461     ## Vertexing
462     ## ============================================================================
463
464     edm4eic::Vertex:
465       Description: "EIC vertex"
466       Author: "J. Osborn"
467       Members:
468         - int32_t             type          // Type flag, to identify what type of vertex it is (e.g. primary, secondary, generated, etc.)
469         - float               chi2          // Chi-squared of the vertex fit
470         - int                 ndf           // NDF of the vertex fit
471         - edm4hep::Vector4f   position      // position [mm] + time t0 [ns] of the vertex. Time is 4th component in vector
472         ## this is named "covMatrix" in EDM4hep, renamed for consistency with the rest of edm4eic
473         - edm4eic::Cov4f      positionError // Covariance matrix of the position+time. Time is 4th component, similarly to 4vector
474       OneToManyRelations:
475         - edm4eic::ReconstructedParticle associatedParticles // particles associated to this vertex.
```

Performance Evaluation

DIS PYTHIA events

- PYTHIA ep 18x275
- Vertex position: afterburner to apply beam effects
- $Q^2 > 10\ GeV^2$

- EIC geometry: *epic-24.06.0*
- EICrecon: branch *vertexing_group*

cloned from the main branch on June 20
realistic seeding updates + ambiguity solver included

Configurations:

- tracking input: both truth seeding and realistic seeding
- IVF (IterativeVertexFinder) parameters
  1) default
  2) |z0| < 100mm, |d0| < 3mm (PCA to (0,0) line)

Definitions:
N_{MC}: Number of MC charged tracks from collision vertex within |eta| < 3.5
N_{RC}: Number of reconstructed charged tracks associated with the reconstructed vertex

Vertexing Efficiency

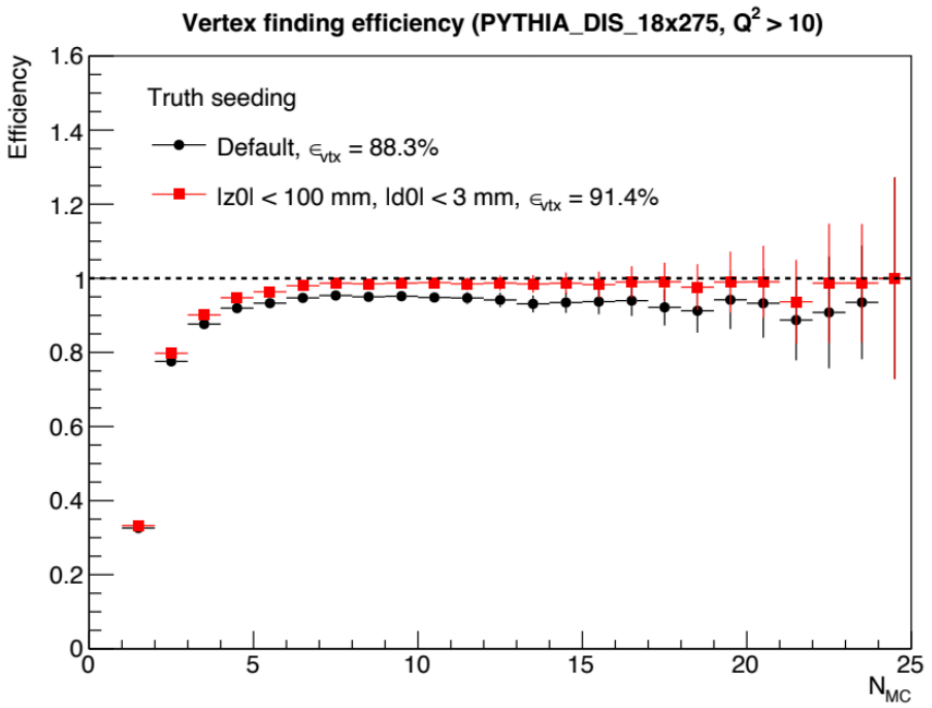
Rongrong Ma

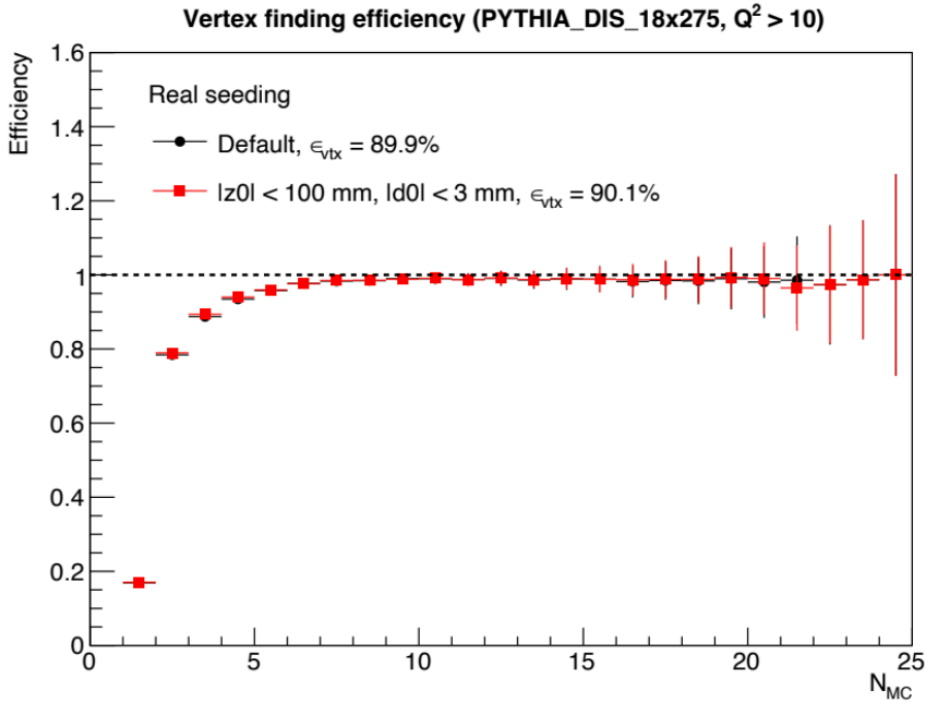
Efficiency = (Events with at least one reconstructed vertex) / (All events)

Truth seeding

$\Delta r < 1$ mm

Real seeding



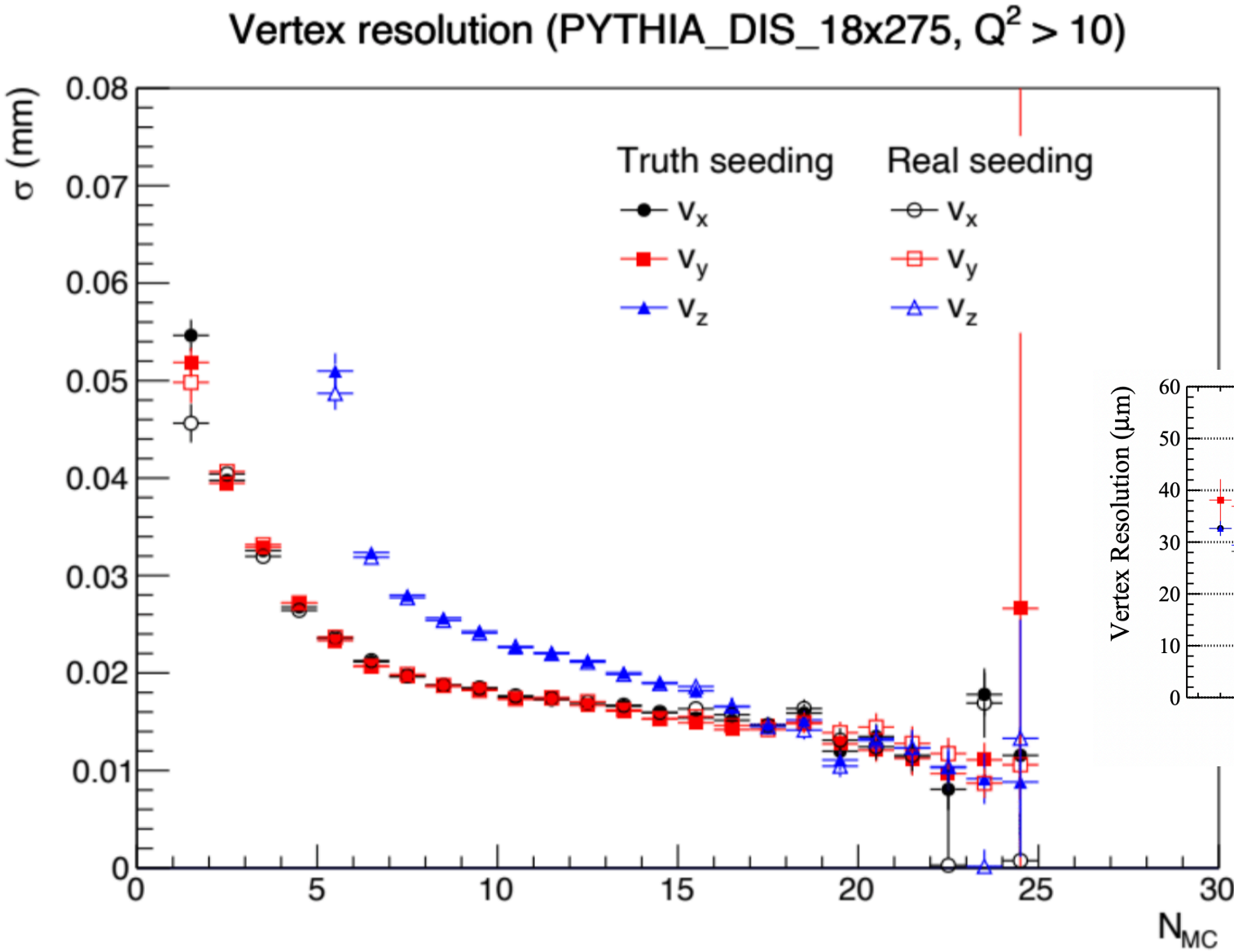


For DIS events with N_{MC} >5, vertexing efficiency is ~ 100% for truth and realistic seeded tracking

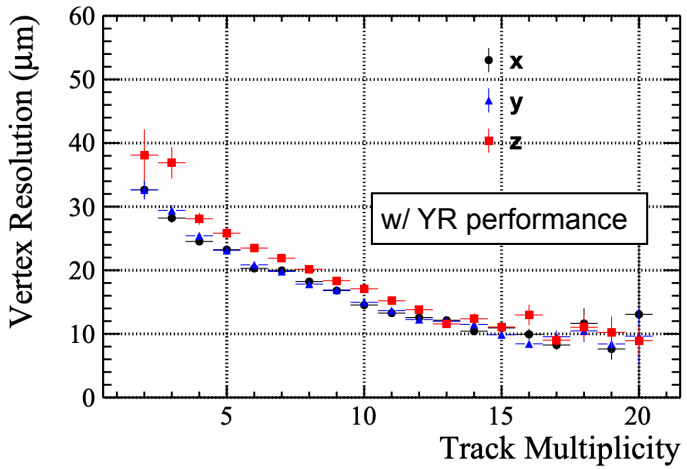
95% of reconstructed vertex within 1mm of the MC vertex

Vertex Resolution

Rongrong Ma



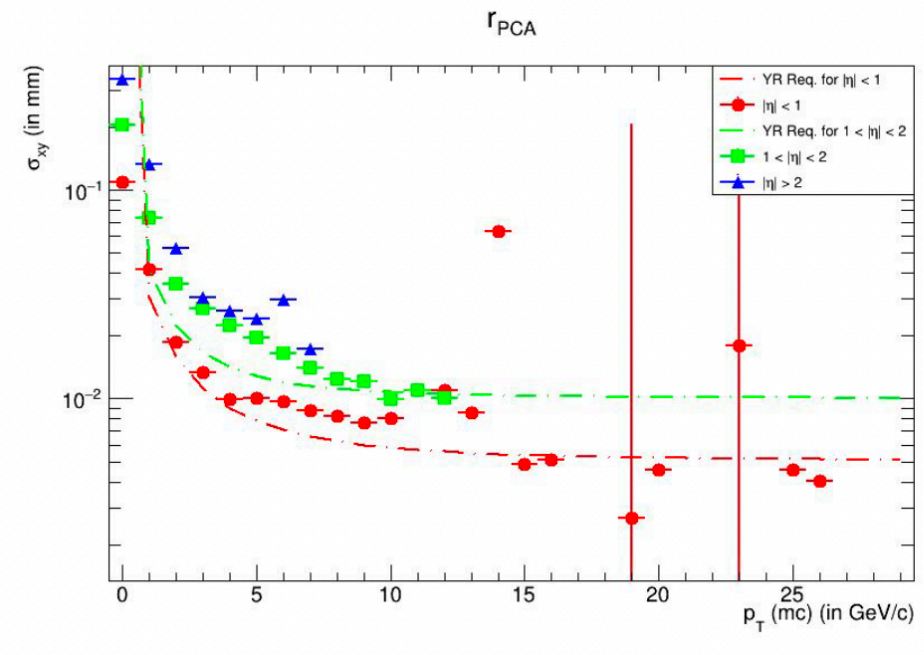
M. Kelsey et. al., PRD 104 (2021) 054002

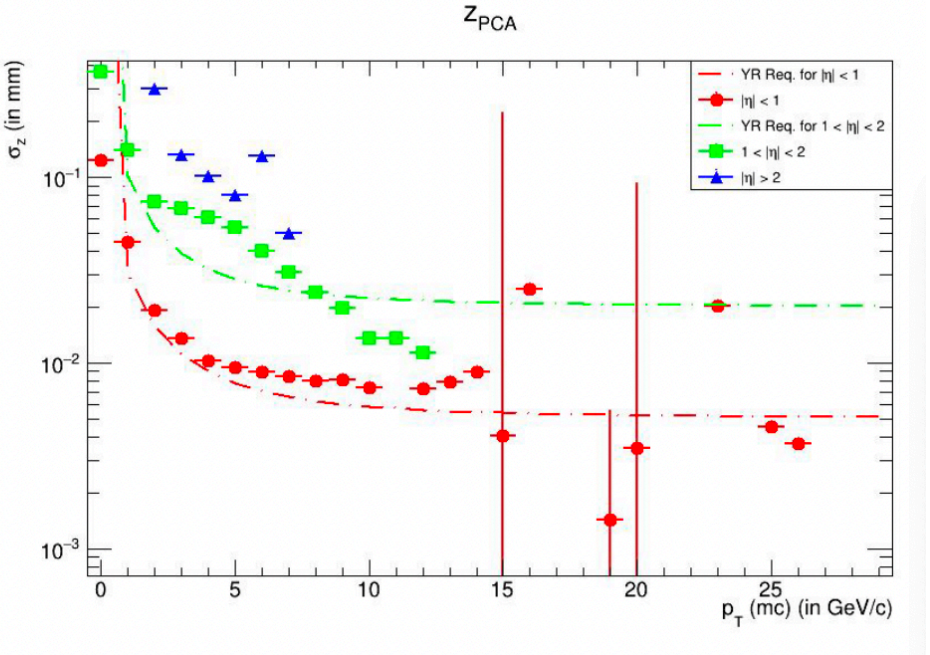


Vertex resolution reaches to < 20 μm with N_{MC} > 10

Pointing Resolution

Khushi Singla





Similar performance studied by the tracking evaluation team

DIS events from (0,0,0) with May geometry, will update with June geometry simulation

Need functionalities for helix swimming in analysis level:

- MC vertex away from (0,0)
- DCA w.r.t to reconstructed vertex position

Associated Particles in Vertex

According to Woulter, S&C team is working on a global update to the data model so the PODIO output objects keep the links to the Acts objects. This requires a new version of Acts and will need 3+ months?

In the meantime, we are working on an intermediate solution so users have the access to the associated particles from vertices.

tracking plugins

```
app->Add(new JOmniFactoryGeneratorT<TracksToParticles_factory>(
        "ChargedSeededParticlesWithAssociations",
        {"MCParticles",                                  // edm4hep::MCParticle
        "CentralCKFSeededTracks",                        // edm4eic::Track
        },
        {"ReconstructedSeededChargedWithoutPIDParticles",            //
         "ReconstructedSeededChargedWithoutPIDParticleAssociations"  // edm4eic::MCRecoParticleAssociation
        },
        link_cfg,
        app
        ));

app->Add(new JOmniFactoryGeneratorT<IterativeVertexFinder_factory>(
        "CentralTrackVertices",
        {"CentralCKFSeededActsTrajectories","ReconstructedSeededChargedWithoutPIDParticles"},
        {"CentralTrackVertices"},
        {},
        app
        ));
```

Move to the end in tracking plugins

New input added

Input arguments in IterativeVertexFinder.cc

```
std::unique_ptr<edm4eic::VertexCollection> eicrecon::IterativeVertexFinder::produce(
    std::vector<const ActsExamples::Trajectories*> trajectories,
    std::vector<const edm4eic::ReconstructedParticle*> reconParticles) {
```

New input added

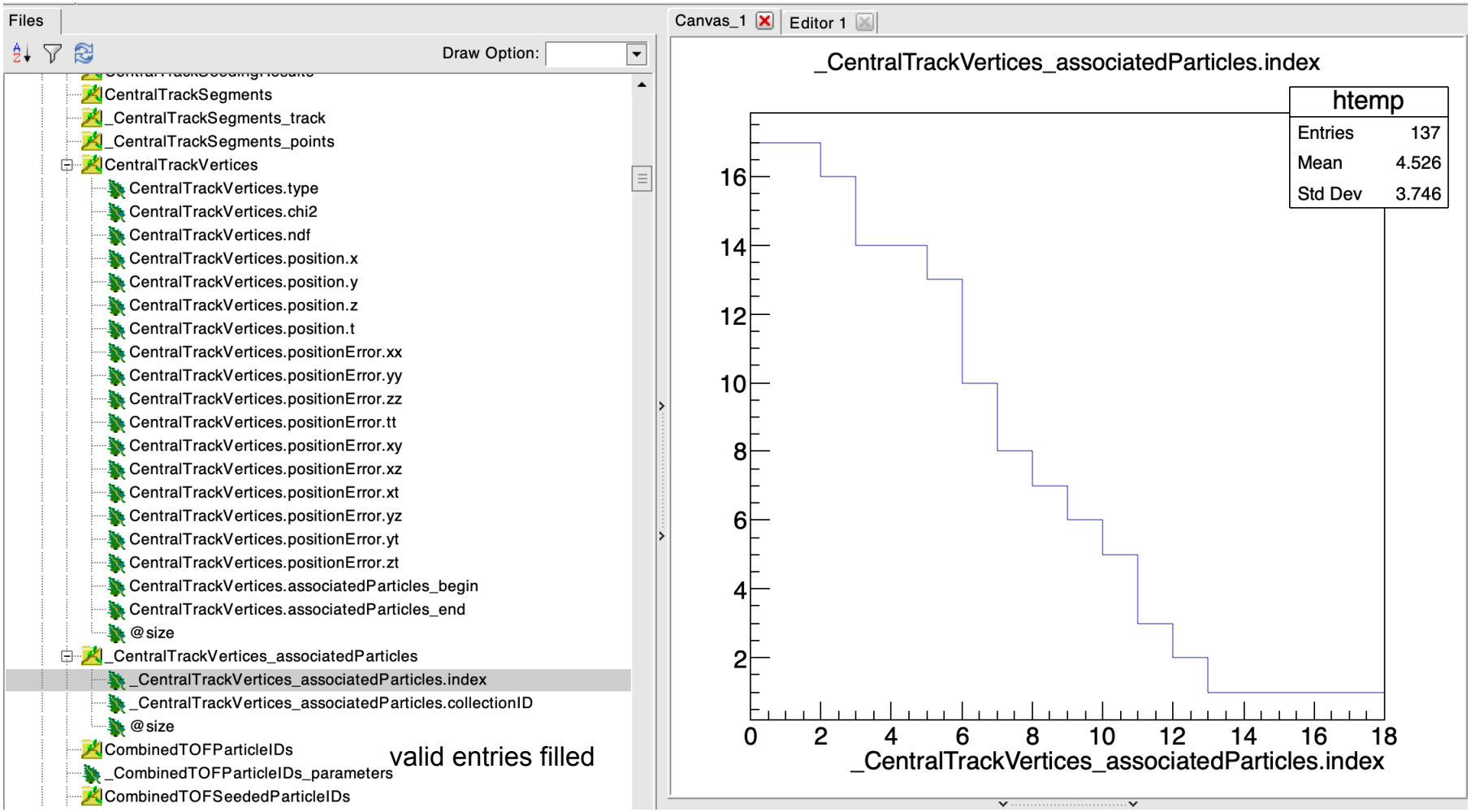
Filling part in IterativeVertexFinder.cc

```
    for (const auto& t : vtx.tracks()) {
#if Acts_VERSION_MAJOR >= 33
      const auto& trk = &t.originalParams;
      const auto& par = finderCfg.extractParameters(trk);
#else
      const auto& par = *t.originalParams;
#endif
      m_log->debug(" === track local position from vertex = {}, {}", par.localPosition().x(), par.localPosition().y());
      float loc_a = par.localPosition().x();
      float loc_b = par.localPosition().y();

      for (const auto part : reconParticles) {
        const auto& tracks = part->getTracks();
        for (const auto trk : tracks) {
          const auto& traj = trk.getTrajectory();
          const auto& trkPars = traj.getTrackParameters();
          for (const auto par : trkPars) {
            if(fabs(par.getLoc().a - loc_a) < 1.e-4 && fabs(par.getLoc().b - loc_b) < 1.e-4) {
              m_log->debug(" --- From ReconParticles, track local position = {}, {}", par.getLoc().a, par.getLoc().b);
//              std::cout << " par from ReconParticles " << par.getLoc().a << "\t" << par.getLoc().b << std::endl;
              eicvertex.addToAssociatedParticles(*part);
            } // endif
          } // end for par
        } // end for trk
      } // end for part
    } // end for t
    m_log->info(" +++ This vertex found at (x,y,z) = ({}, {}, {}) mm.", vtx.position().x(), vtx.position().y(), vtx.position().z());
```

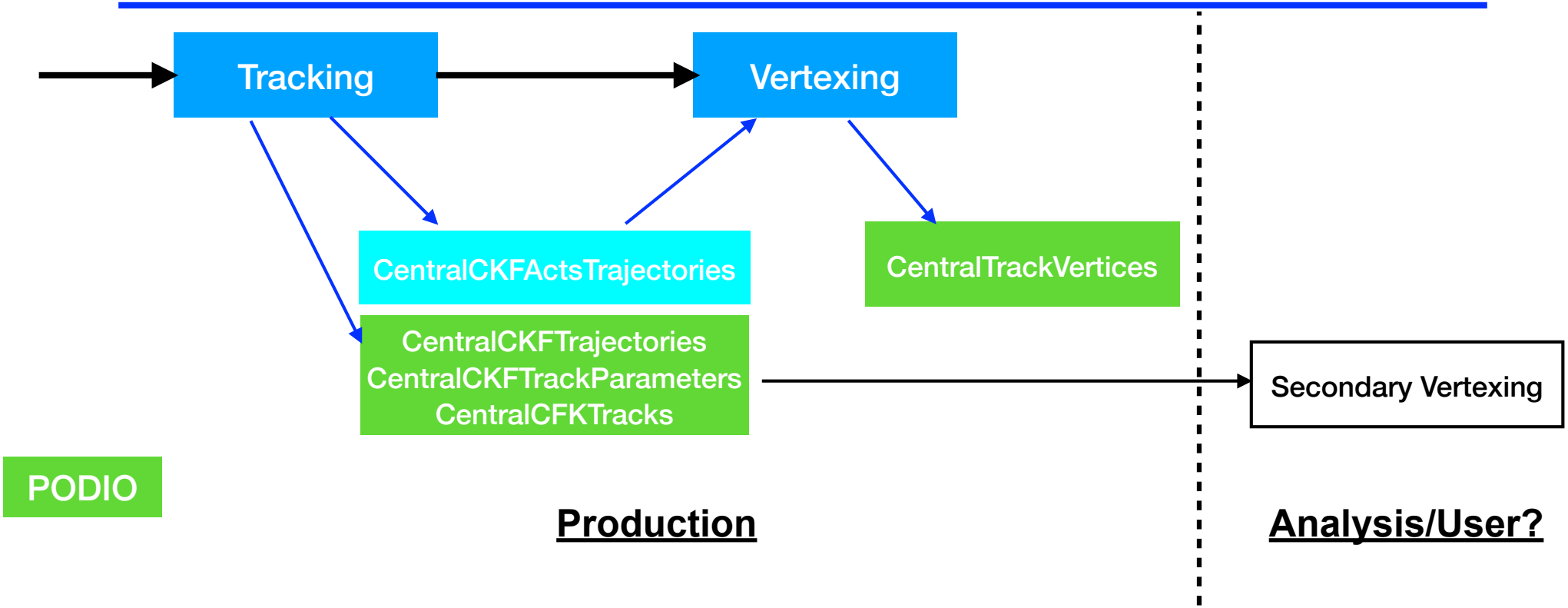
Compare track parameters

PODIO output



Compared the associatedParticle array size, consistent with the ACTS vertexing output
Working on more detailed checks on any potential issue

Secondary Vertexing



Secondary Vertexing - leave at analysis/user level
- many different kind of decays for reconstruction
- topological selection criteria better be optimized for different decays/observables

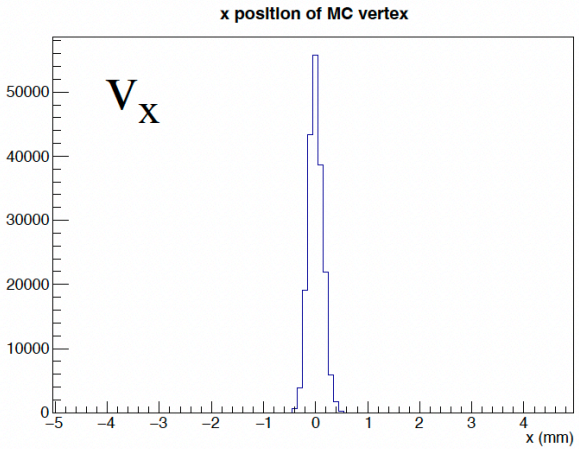
KFParticle

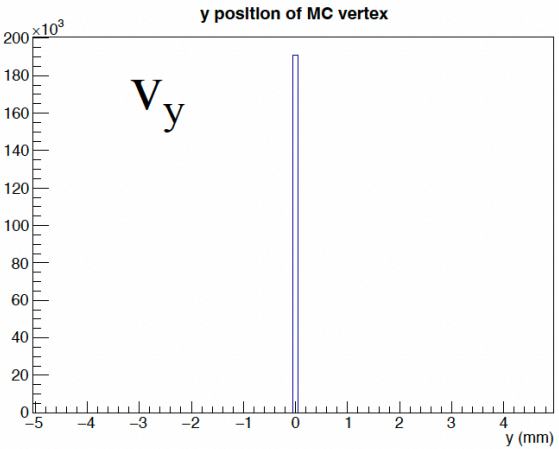
KFParticle package developed by FIAS group - deployed by STAR, sPHENIX, CBM etc. *X-Y Ju et al, NST 34 (2023) 158*
See presentations by Pavel Kisel (STAR), Cameron Dean (sPHENIX) https://indico.bnl.gov/event/24092/

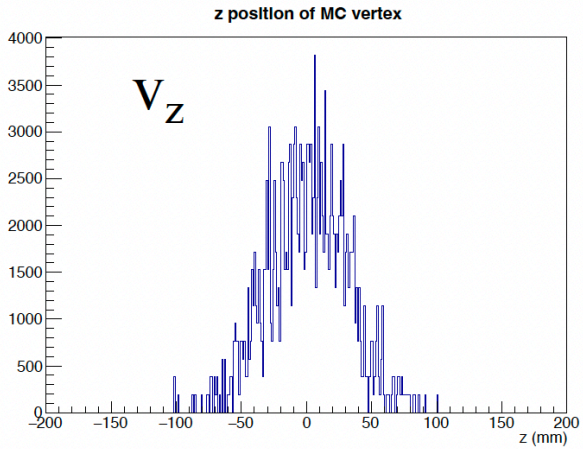
Input: track parameters and covariances (contained in PODIO output)
Many decays included and available in the package (allowing extension) - weak decays / resonance decays
Can be used for primary vertex refitting too

Backups

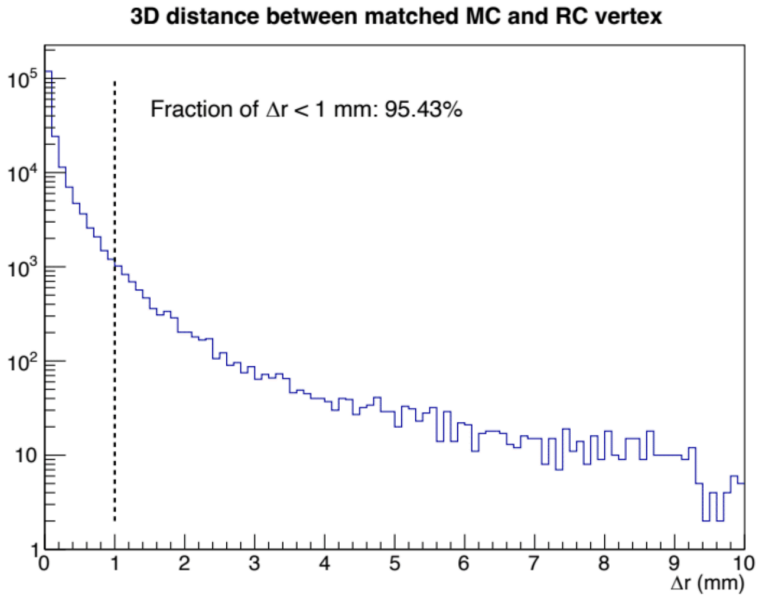
Vertex Smearing / MC-RC Vertex Matches







Real seeding: default



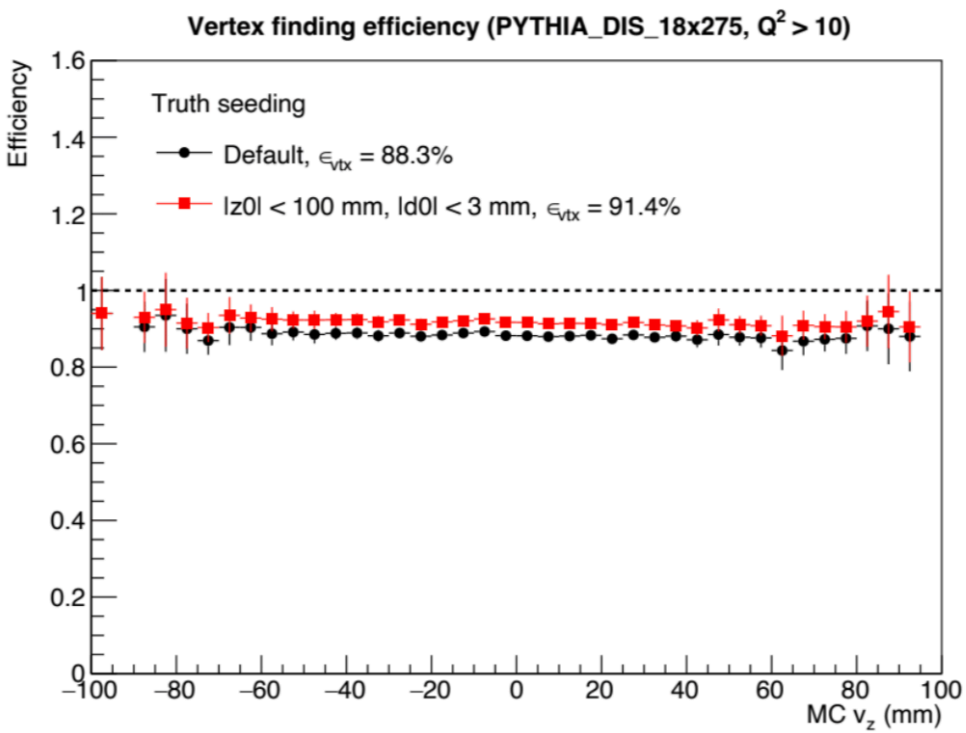
Δr : 3D distance difference

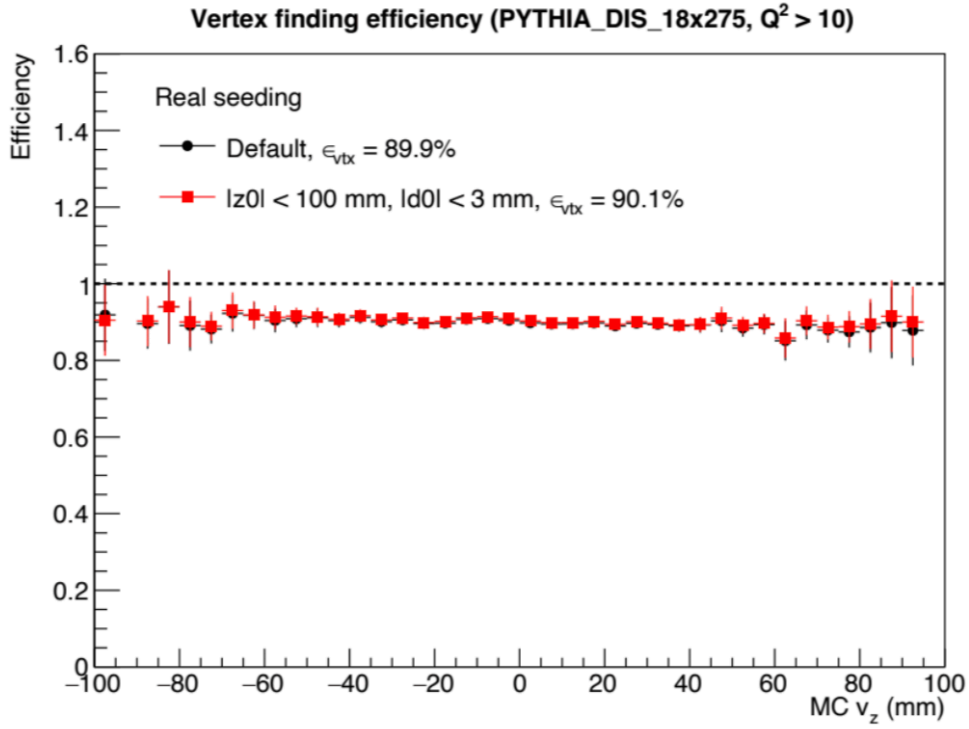
Vertex Efficiency vs. Vz

Truth seeding

$\Delta r < 1$ mm

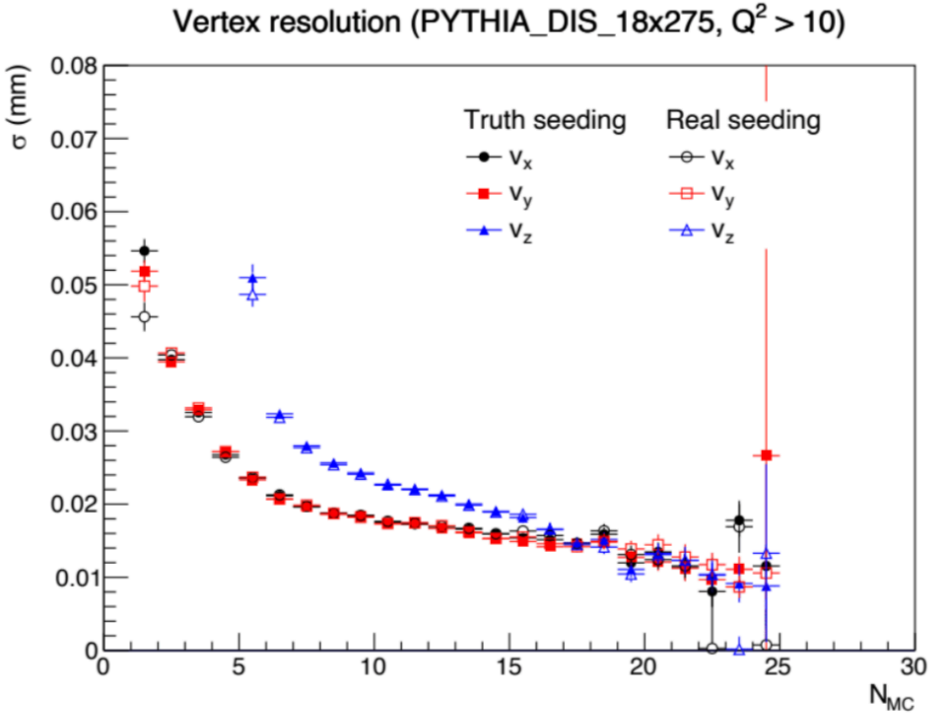
Real seeding



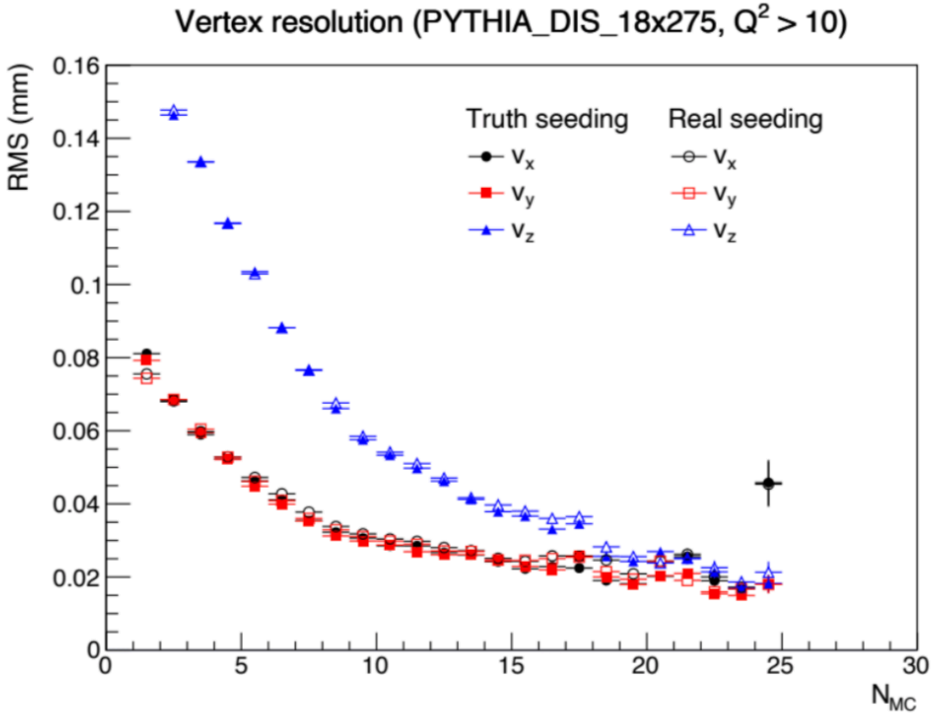


Vertex Resolution

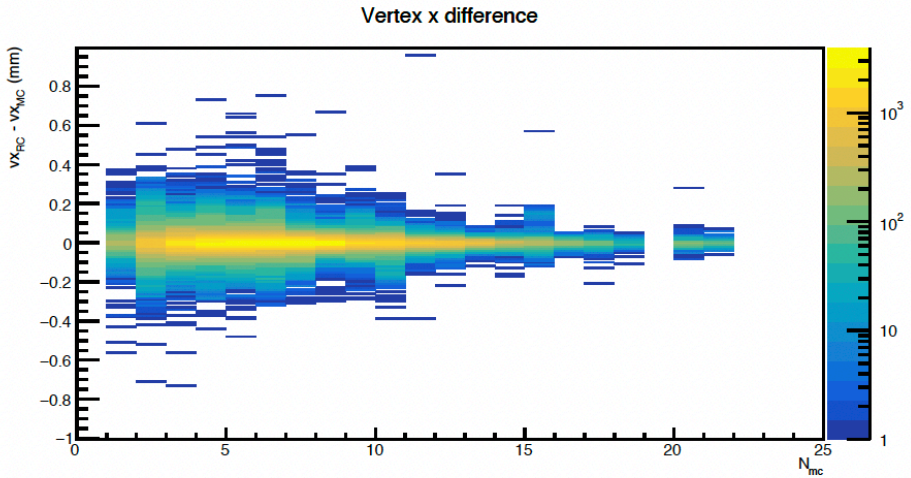
σ vs. N_{MC}

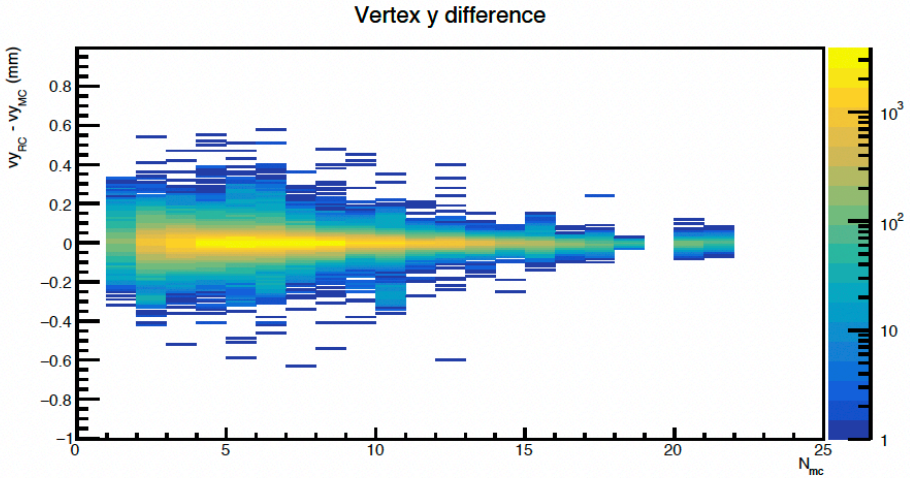


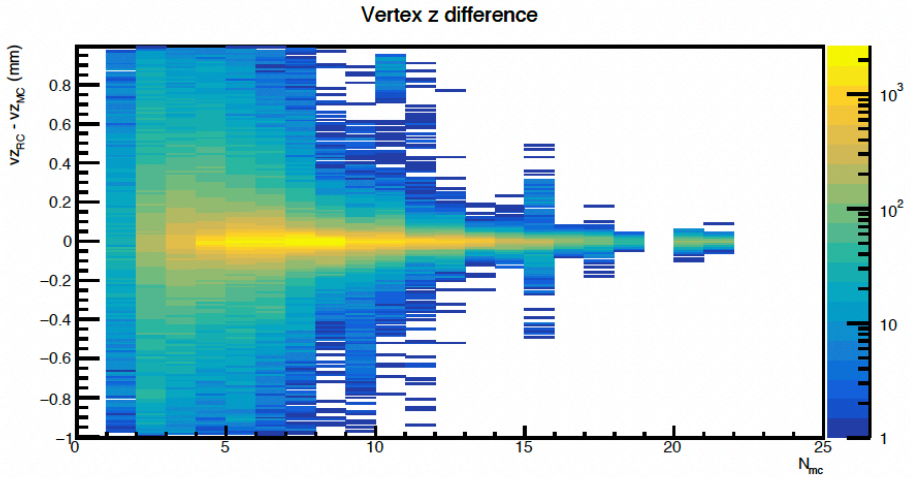
RMS vs. N_{MC}



Vertex Resolution Extraction







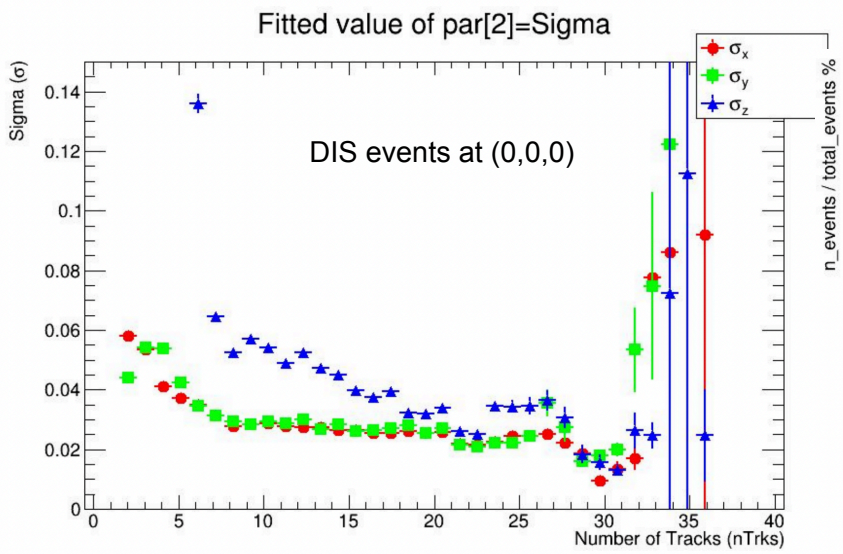
Real seeding: default

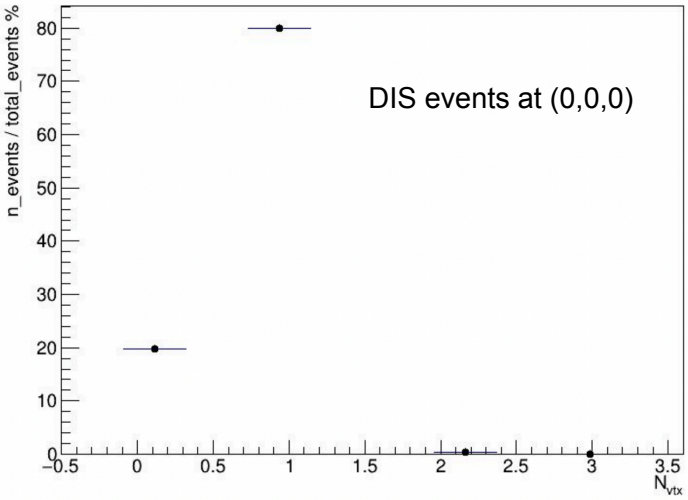
Student-t function used to fit and account for tails:
Student-t -> Gaussian when $\nu \rightarrow \inf$

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\,\nu}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-(\nu+1)/2}$$

Vertexing Status as of 04/2024

1) For DIS events at (0,0,0), vertex resolution looks good, however, efficiency is only about 80%





2) For events starting away from(0,0,0), vertex resolution degrades considerably

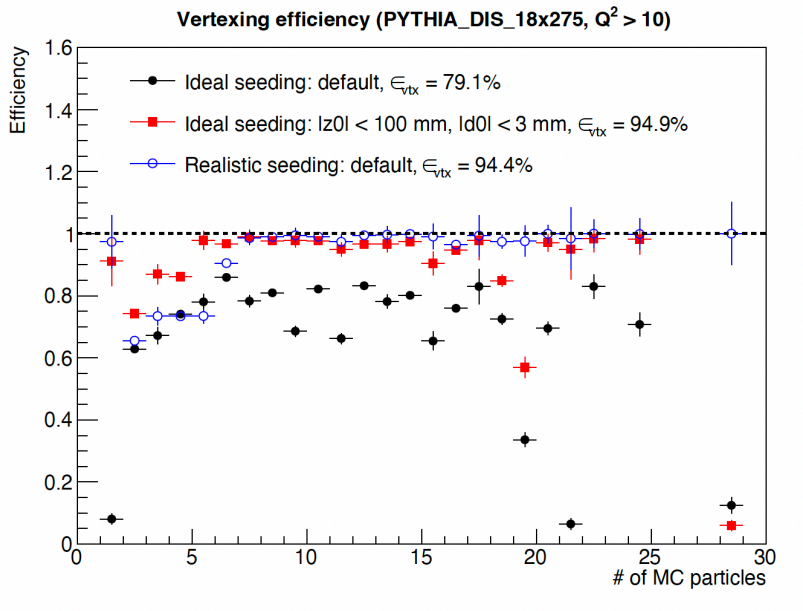
10-muon track per event

vertex pos [in mm]	vtx res: x [in μm]	vtx res: y [in μm]	vtx res: z [in μm]
(0,0,0)	11.85	11.09	10.57
(0.5,0,0)	49.89	44.91	59.33
(1,0,0)	72.17	65.80	79.88
(2,0,0)	82.43	78.21	94.38
(3,4,5)	96.12	96.55	100.7

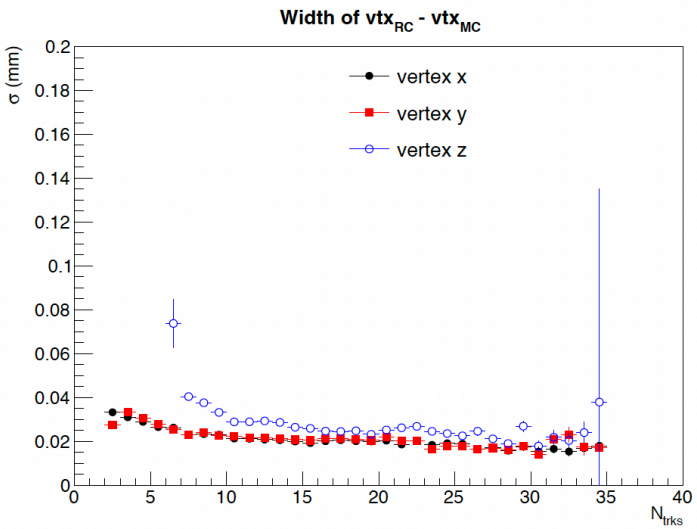
Performance with May Setup

- PYTHIA DIS 18x275
- $Q^2 > 10$ GeV^2

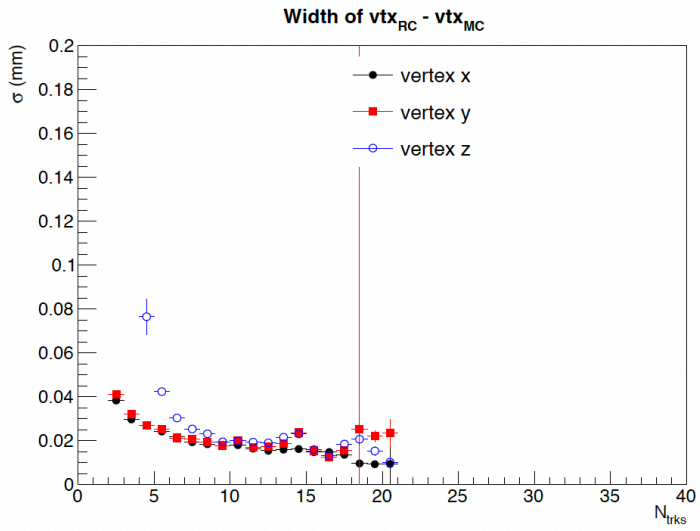
- EIC geometry: epic-24.05.0
- EICreco: May 2024



Truth seeding
$|z0| < 100$ mm, $|d0| < 3$ mm



Realistic seeding
default



Progress since Then

1) Algorithm tuning to address
   - Vertexing efficiency for DIS events from (0,0,0)
   - Vertex resolution for off-axis events
2) EICrecon update (tracking geometry update - 06.2024)
3) Realistic seeding (including AmbiguitySolver)