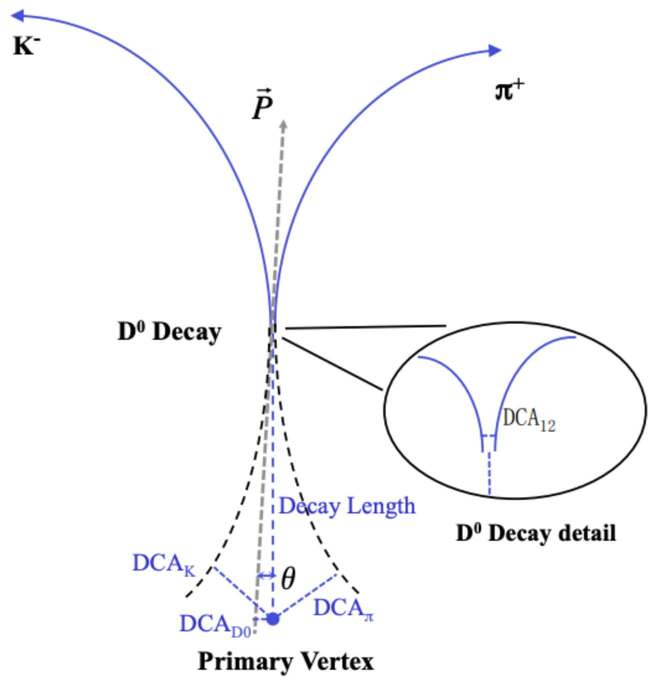
Helix Utils to edm4eic

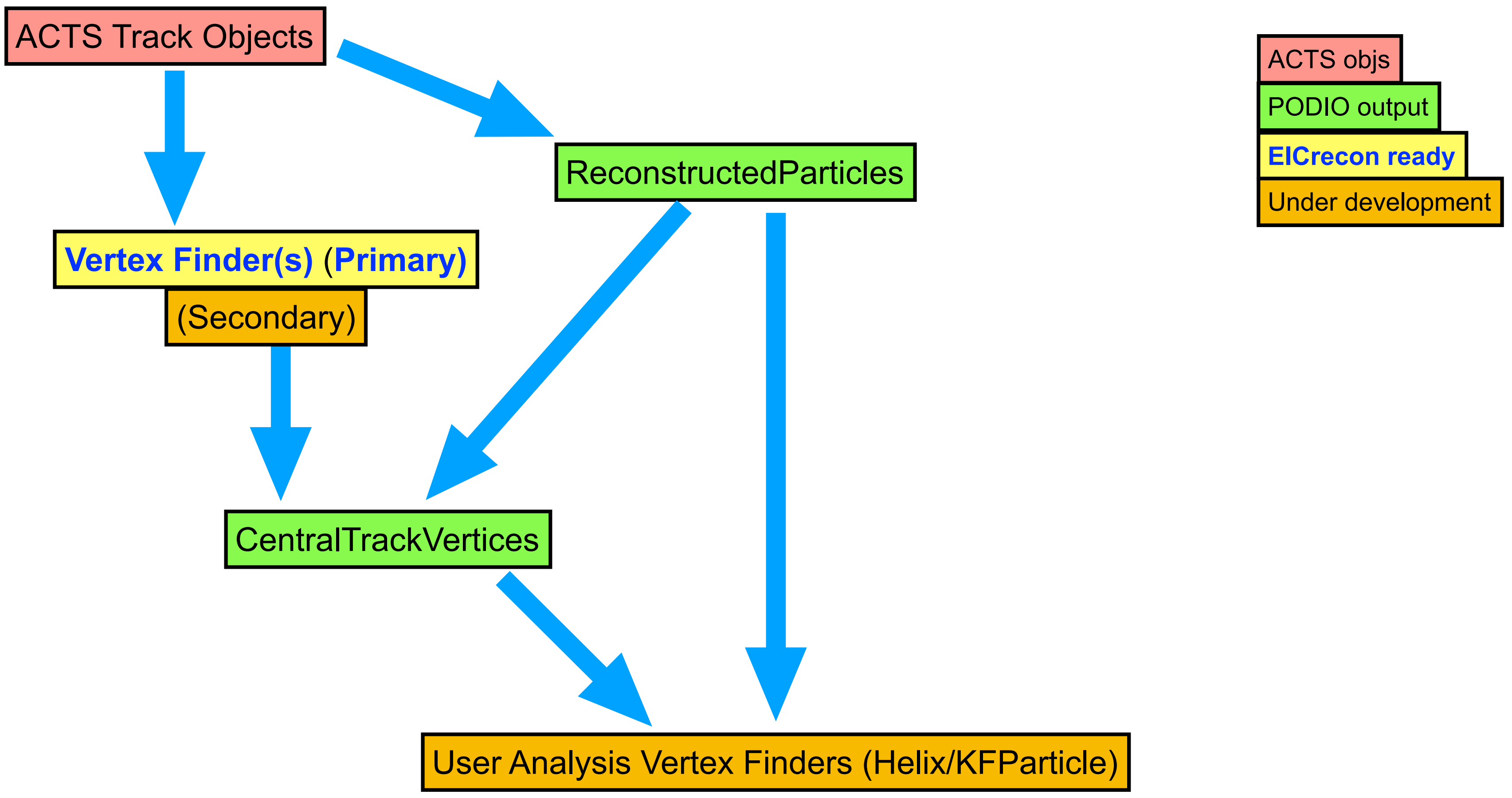
Xin Dong



Thanks to:

Bishoy Dongwi, Lokesh Kumar, Rongrong Ma, Joe Osborn, Ashish Pandav, *Harsimran Singh*, *Khushi Singla*, Deepa Thomas, Connie Yang etc.

Vertex Finders

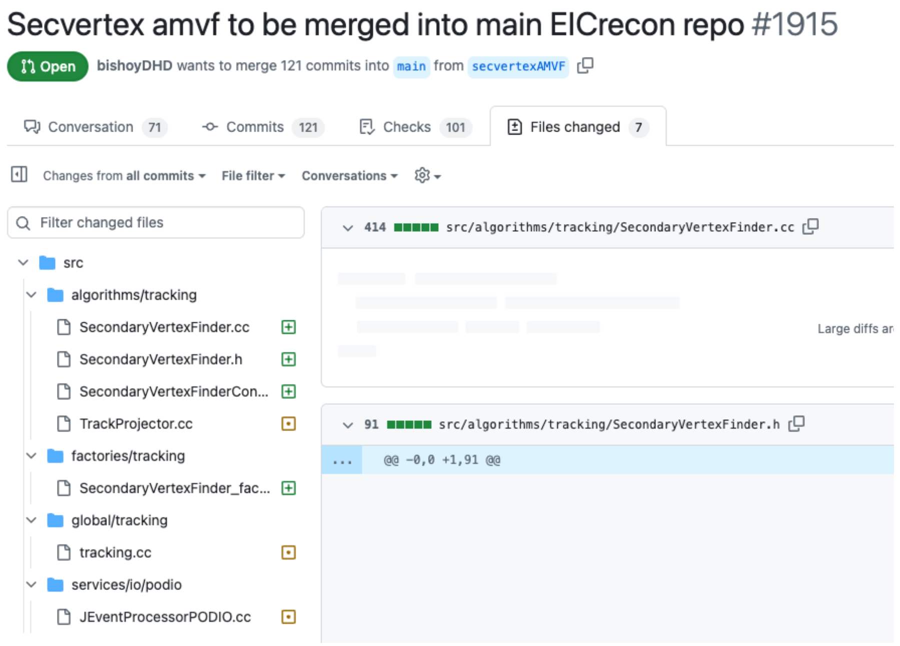


1) Primary Vertex Finder: IterativeVertexFinder in EICrecon production
2) All vertices stored in CentralTrackVertices in PODIO, ranked in PrimaryVertices
3) QA performance plots included in detector_benchmarks repo

Secondary Vertex Finder(s)

1. <u>Production level:</u>
SecondaryVertexFinder factory in EICrecon - PR # 1915
- based on ACTS AVF finder
- performance check for selected particles during production

Bishoy Dongwi


Secvertex amvf to be merged into main EICrecon repo #1915

Open bishoyDHD wants to merge 121 commits into main from secvertexAMVF

Conversation 71 · Commits 121 · Checks 101 · Files changed 7

Changes from all commits · File filter · Conversations

Filter changed files

- src
  - algorithms/tracking
    - SecondaryVertexFinder.cc
    - SecondaryVertexFinder.h
    - SecondaryVertexFinderCon...
    - TrackProjector.cc
  - factories/tracking
    - SecondaryVertexFinder_fac...
  - global/tracking
    - tracking.cc
  - services/io/podio
    - JEventProcessorPODIO.cc

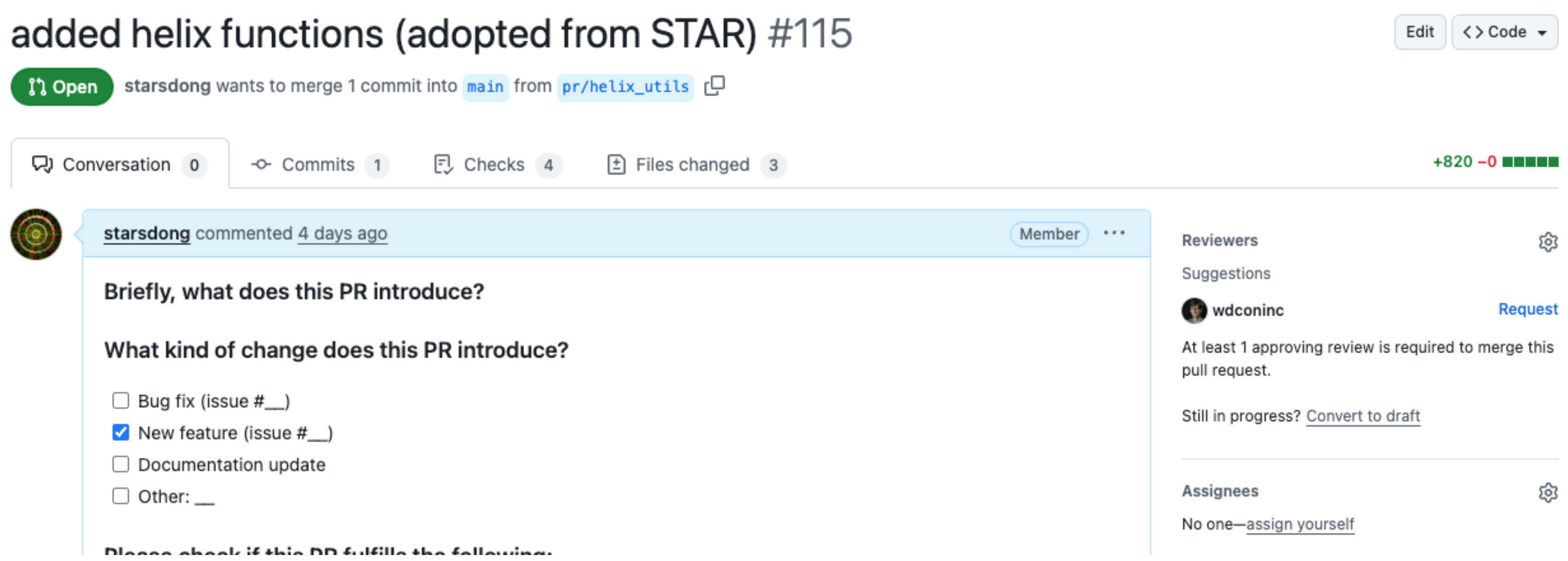
```
414 src/algorithms/tracking/SecondaryVertexFinder.cc
Large diffs ar
91 src/algorithms/tracking/SecondaryVertexFinder.h
... @@ -0,0 +1,91 @@
```

2. <u>User level:</u>
Work with PODIO input
Flexible and can handle various different secondary vertices

a. Helix swimming (adopted from STAR) *Rongrong Ma, Connie Yang etc.*
   - helix utils to edm4eic - PR # 115 (today's discussion)

b. KFParticle *Ashish Pandav*
   - used by STAR/sPHENIX/ALICE/CBM etc.
   - under testing now, to be deployed later

Adding Helix Functions in EDM4eic



added helix functions (adopted from STAR) #115

Open starsdong wants to merge 1 commit into main from pr/helix_utils

starsdong commented 4 days ago

Briefly, what does this PR introduce?

What kind of change does this PR introduce?

- [ ] Bug fix (issue #__)
- [x] New feature (issue #__)
- [ ] Documentation update
- [ ] Other: __

Reviewers

Suggestions

wdconinc Request

At least 1 approving review is required to merge this pull request.

Still in progress? Convert to draft

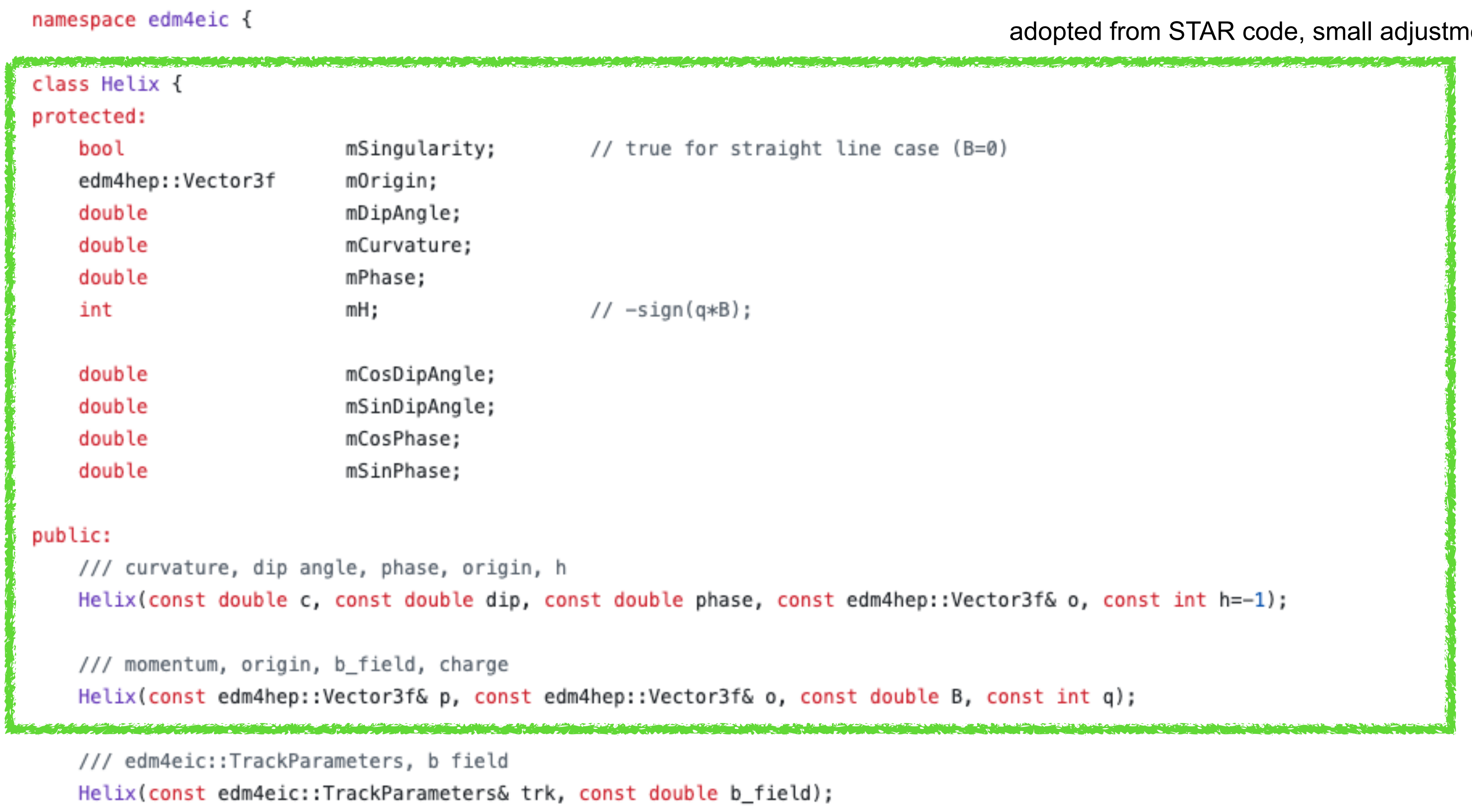
Assignees

No one—assign yourself

1) Helix afterburner reconstruction is used in D^0 reconstruction (later part), targeted to be used for updated physics projection plots.
2) Constructor includes using EICrecon TrackParameters as input.
3) Handling constant z-magnetic field (or zero field - straight-line)
    - can be extended to handle varying B-field for track projection
4) Iterative varying-step-scan to find DCA positions between helices numerically.

Helix

adopted from STAR code, small adjustments



```
namespace edm4eic {

class Helix {
protected:
    bool                    mSingularity;       // true for straight line case (B=0)
    edm4hep::Vector3f       mOrigin;
    double                  mDipAngle;
    double                  mCurvature;
    double                  mPhase;
    int                     mH;                 // -sign(q*B);

    double                  mCosDipAngle;
    double                  mSinDipAngle;
    double                  mCosPhase;
    double                  mSinPhase;

public:
    /// curvature, dip angle, phase, origin, h
    Helix(const double c, const double dip, const double phase, const edm4hep::Vector3f& o, const int h=-1);

    /// momentum, origin, b_field, charge
    Helix(const edm4hep::Vector3f& p, const edm4hep::Vector3f& o, const double B, const int q);

    /// edm4eic::TrackParameters, b field
    Helix(const edm4eic::TrackParameters& trk, const double b_field);
```

added constructor, taking edm4eic objects

Useful Functions in Helix

adopted from STAR code, small adjustments

```
/// path length at given r (cylindrical r)
std::pair<double, double> pathLength(double r)   const;

/// path length at given r (cylindrical r, cylinder axis at x,y)
std::pair<double, double> pathLength(double r, double x, double y);

/// path length at distance of closest approach to a given point
double       pathLength(const edm4hep::Vector3f& p, bool scanPeriods = true) const;

/// path length at intersection with plane
double       pathLength(const edm4hep::Vector3f& r,
               const edm4hep::Vector3f& n) const;

/// path length at distance of closest approach in the xy-plane to a given point
double       pathLength(double x, double y) const;
```

Track DCA to point/plane/perigee surface etc.

```
/// path lengths at dca between two helices
std::pair<double, double> pathLengths(const Helix&,
                                      double minStepSize = 10*edm4eic::unit::um,
                                      double minRange = 10*edm4eic::unit::cm) const;
```

Two-track crossing / DCA calculation

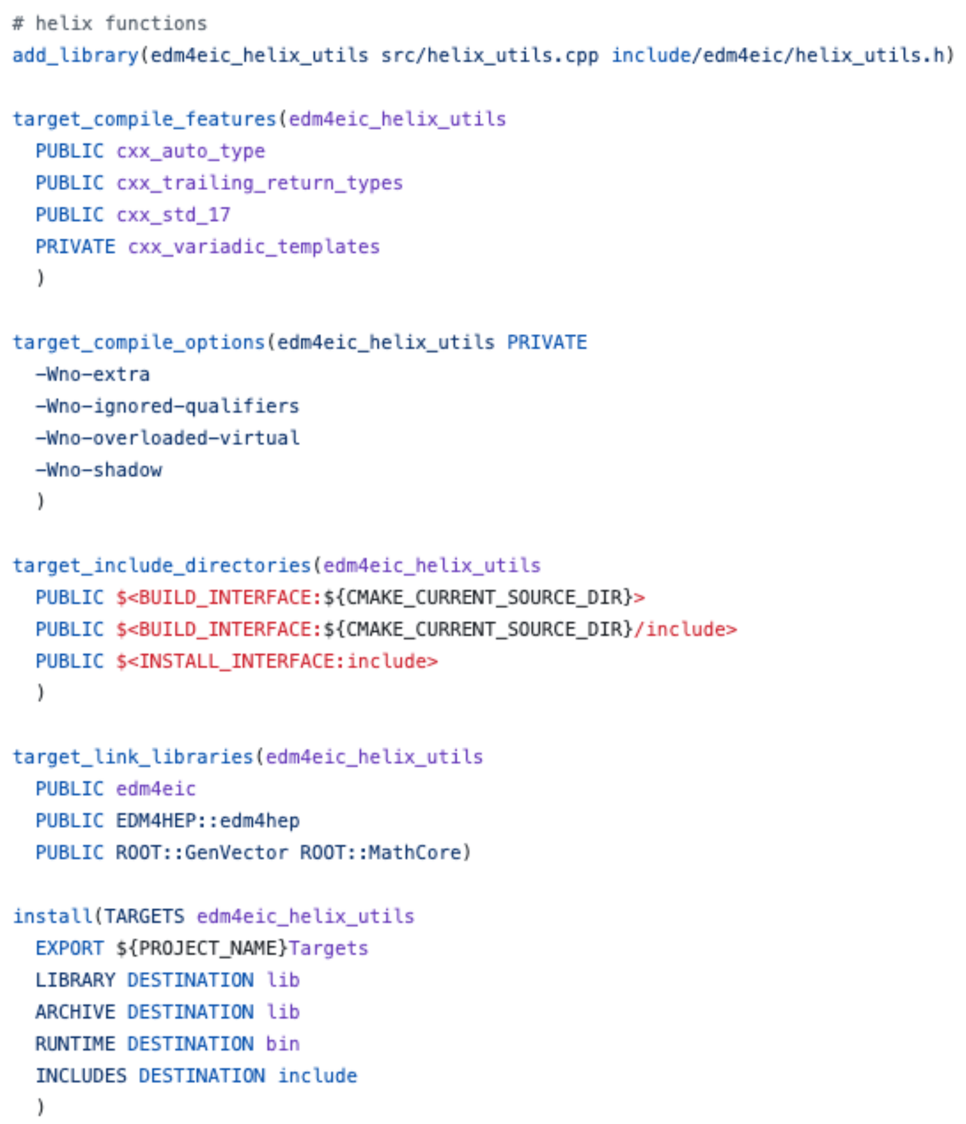
```
/// minimal distance between point and helix
double       distance(const edm4hep::Vector3f& p, bool scanPeriods = true) const;

/// checks for valid parametrization
bool         valid(double world = 1.e+5) const {return !bad(world);}
int            bad(double world = 1.e+5) const;

/// move the origin along the helix to s which becomes then s=0
virtual void moveOrigin(double s);
```

Question: How to build and install

Current implementation in CMakeLists.txt, need experts' advice and help to define the strategy



```
# helix functions
add_library(edm4eic_helix_utils src/helix_utils.cpp include/edm4eic/helix_utils.h)

target_compile_features(edm4eic_helix_utils
  PUBLIC cxx_auto_type
  PUBLIC cxx_trailing_return_types
  PUBLIC cxx_std_17
  PRIVATE cxx_variadic_templates
  )

target_compile_options(edm4eic_helix_utils PRIVATE
  -Wno-extra
  -Wno-ignored-qualifiers
  -Wno-overloaded-virtual
  -Wno-shadow
  )

target_include_directories(edm4eic_helix_utils
  PUBLIC $<BUILD_INTERFACE:${CMAKE_CURRENT_SOURCE_DIR}>
  PUBLIC $<BUILD_INTERFACE:${CMAKE_CURRENT_SOURCE_DIR}/include>
  PUBLIC $<INSTALL_INTERFACE:include>
  )

target_link_libraries(edm4eic_helix_utils
  PUBLIC edm4eic
  PUBLIC EDM4HEP::edm4hep
  PUBLIC ROOT::GenVector ROOT::MathCore)

install(TARGETS edm4eic_helix_utils
  EXPORT ${PROJECT_NAME}Targets
  LIBRARY DESTINATION lib
  ARCHIVE DESTINATION lib
  RUNTIME DESTINATION bin
  INCLUDES DESTINATION include
  )
```

Usage of Helix Method

Helix method has been used in many heavy flavor hadron analysis (Rongrong/Shyam/Connie etc.)

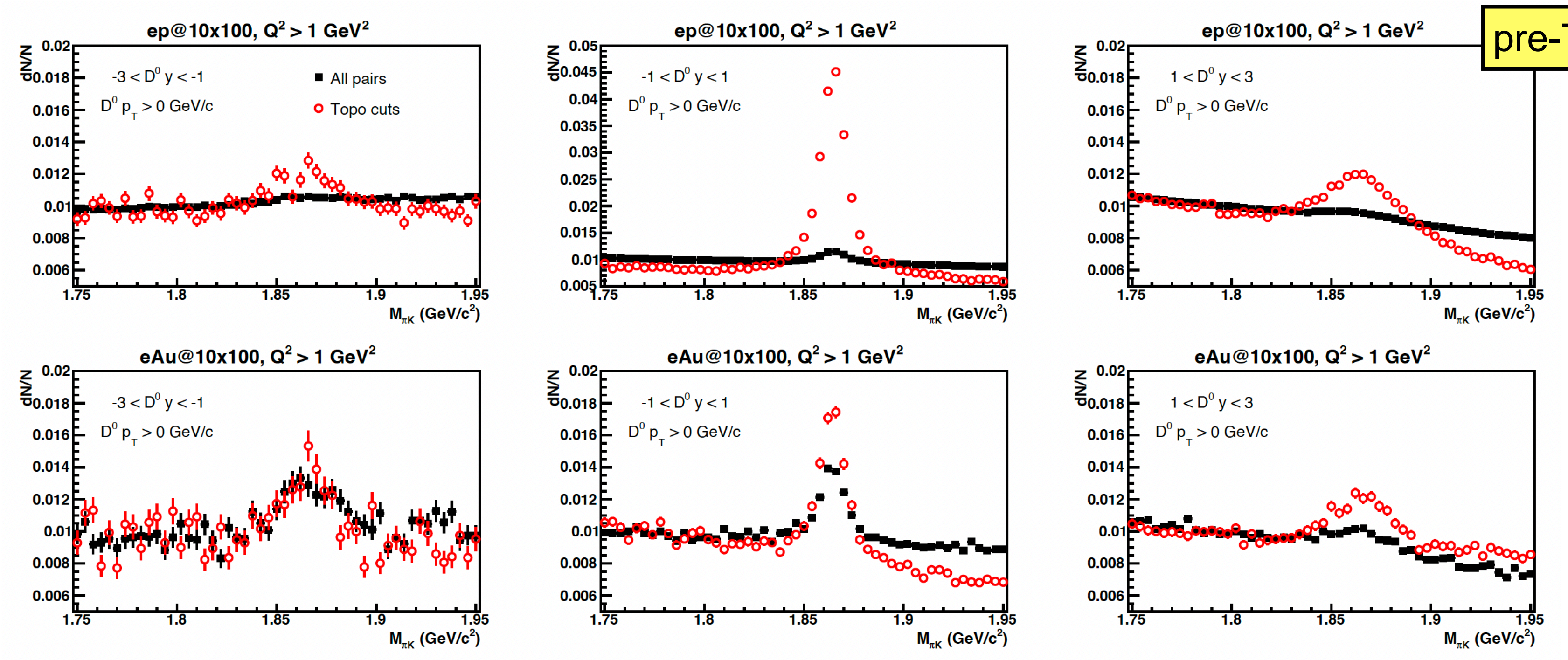


Figure 2.22: Invariant mass distributions of $\pi + K$ pairs with (red circles) and without (black squares) topological selections in 10×100 GeV e+p (top) and e+Au (bottom) collisions with a minimum Q^2 of 1 GeV^2. Different panels from left to right correspond to different D^0 rapidity intervals: $-3 < y < -1$ (left), $-1 < y < 1$ (middle) and $1 < y < 3$ (right).

Example of using Helix method:
https://github.com/marrbnl/ePIC/tree/main/HF_reco/helix

Summary

1) My opinion for secondary vertexing strategy: Need both developments in the EICrecon/production-level and the user level.

2) At user level, adopted helix method from STAR, PR #115 to edm4eic, need experts' help on build and install before merging.
   - KFParticle (broader including covariance) under early testing stage

3) At production level, EICrecon PR #1915 - AMVF under review too.

SecondaryVertexFinder in EICrecon

repo: EICrecon branch: displaced vertex

Bishoy Dongwi (Stonybrook)

src/algorithms/tracking/SecondaryVertexFinder.h and .cc

1) Utilize ACTS AdapterVertexFinder for secondary vertex seed and fitting (can be used for primary vertex finder too)

2) Initial code setup, working on comparison on primary vertex w.r.t the default VF (IterativeVertexFinder)

Goal: Integrate a few selected secondary particle (e.g. Ks, Lambda, D0 etc.) reconstruction in EICrecon production chain for QA/performance check

A more comprehensive secondary vertex tool - KFParticle, more flexible and targeted at the analysis level.
Ashish is working on local test with the KFParticle package with PODIO output, then work on deployment to EICrecon