Secondary Vertexing Development

Shyam Kumar, Xin dong, Bishoy, Rongrong and others

Secondary Vertex Reconstruction

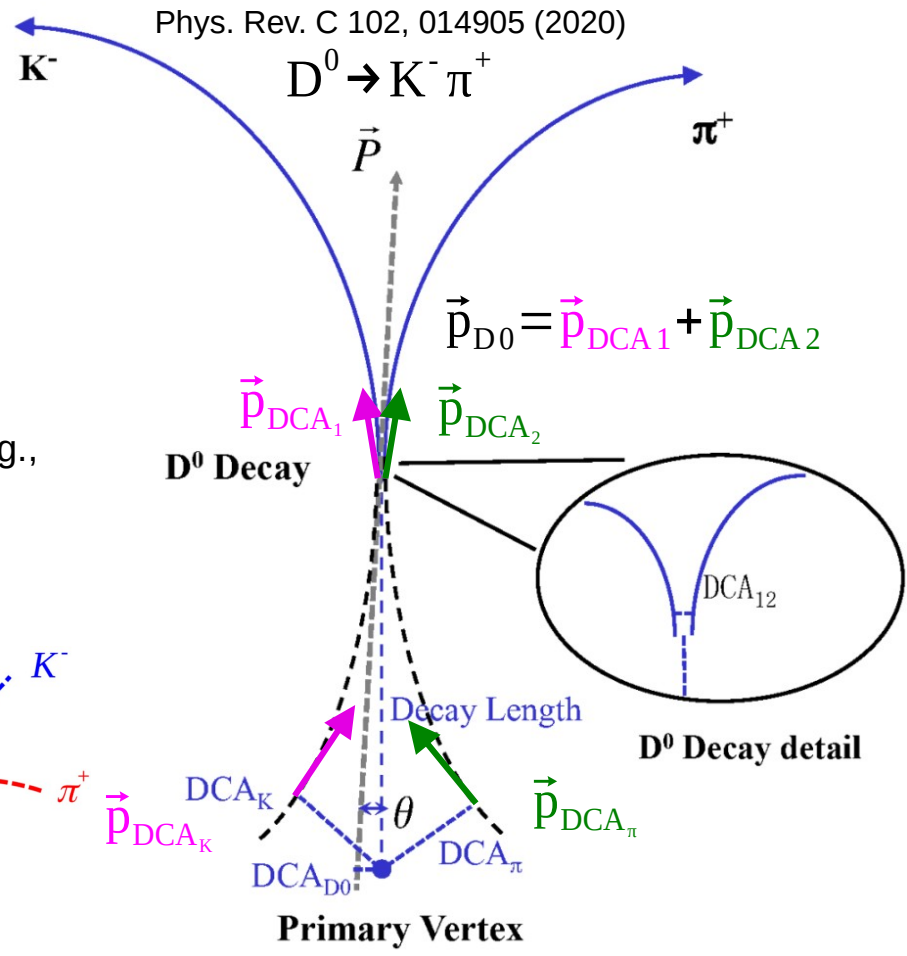
Primary Vertex Reconstruction:

- **CentralTrackVertices**

Secondary Vertex (SV) Reconstruction:

- Helix Swimming (Ignores tracking errors)**:** Xin and others
- AdaptiveMultiVertexFinder (considers track errors): Bishoy

Secondary vertex: allows access to more topological variables, e.g., decay length (dl), pointing angle (cosθ), DCA_{D0} etc.

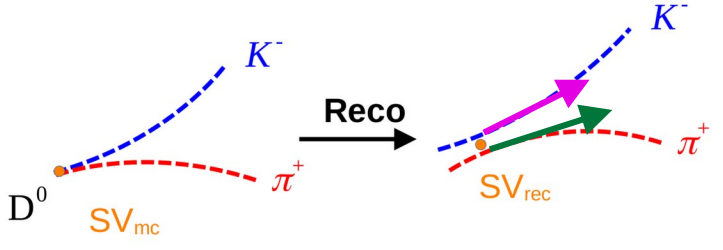


Topological Variables

$$\vec{dl} = \vec{SV} - \vec{PV}$$

$$\cos\theta = \frac{\vec{dl} \cdot \vec{p_{D0}}}{|\vec{dl}||\vec{p_{D0}}|}$$

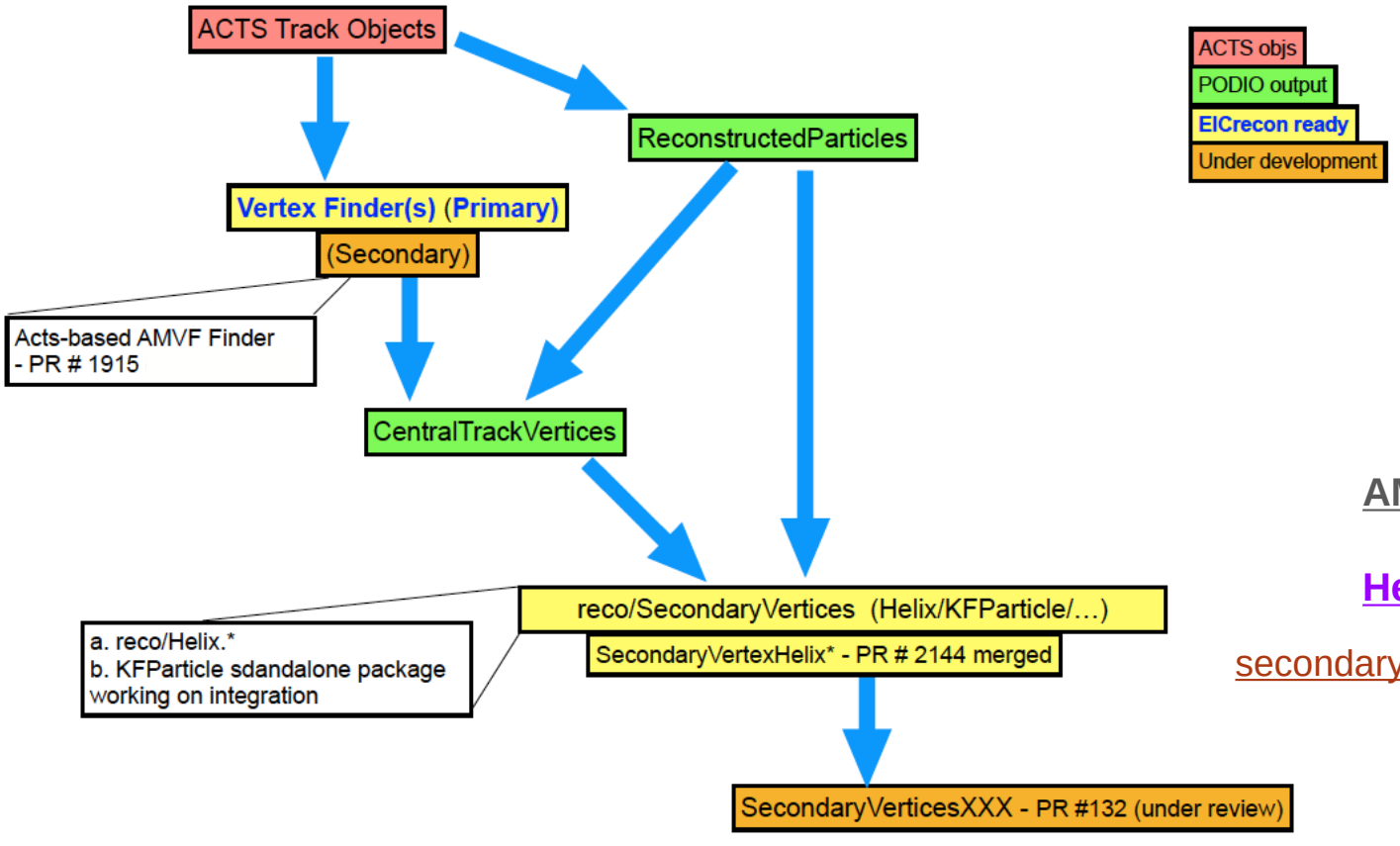
$$DCA_{D0} = |\vec{dl}| \sin\theta$$



$$m(k\pi) = \sqrt{(E_{DCA1} + E_{DCA2})^2 + (\vec{p}_{DCA1} + \vec{p}_{DCA2})^2}$$

Secondary Vertex

X. Dong (LBNL)
B. Dongwi (SBU)



AMVF – Merged (PR #1915)

Helix - merged (PR #2144)

secondary_vertexing_dis: enable for AMVF

SecondaryVertex to edm4eic: add-secondaryvertex branch

Xin dong's slides

1. Current edm4eic::vertex object doesn't contain many topological variables for secondary vertices.
2. Though one can in principle re-calculate all these based on the associated ReconstructedParticle, it will be much more convenient to save them during reconstruction and the down-stream analysis can directly use them for physics analysis and also this will avoid repeated calculations.

Propose to add edm4eic::SecondaryVertex container

https://github.com/eic/EDM4eic/issues/127

Issue 127

```
edm4eic::Vertex:
  Description: "EIC vertex"
  Author: "J. Osborn"
  Members:
    - int32_t             type          // Type flag, to identify what type of ve
    - float               chi2          // Chi-squared of the vertex fit
    - int                 ndf           // NDF of the vertex fit
    - edm4hep::Vector4f   position      // position [mm] + time t0 [ns] of the ve
    ## this is named "covMatrix" in EDM4hep, renamed for consistency with the res
    - edm4eic::Cov4f      positionError // Covariance matrix of the position+time
  OneToManyRelations:
    - edm4eic::ReconstructedParticle associatedParticles // particles associated
```

```
edm4hep::Vertex:
  Description: "Vertex"
  Author: "EDM4hep authors"
  Members:
    - uint32_t            type          // flagword that defines the type of the vertex,
    - float               chi2          // chi-squared of the vertex fit
    - int32_t             ndf           // number of degrees of freedom of the vertex fit
    - edm4hep::Vector3f   position [mm]     //  position of the verte
    - edm4hep::CovMatrix3f covMatrix [mm^2]  // covariance matrix of the position
    - int32_t               algorithmType // type code for the algorithm that has been
  VectorMembers:
    - float               parameters    // additional parameters related to this vertex
  OneToManyRelations:
    - edm4hep::ReconstructedParticle particles // particles that have been used to form t
```

```
edm4eic::SecondaryVertex:
  Description: "EIC secondary vertex"
  Author: "X. Dong"
  Members:
    - int32_t             type          // Type flag, to identify what type of vertex it is (e.g. pr
    - float               chi2          // Chi-squared of the vertex fit
    - int                 ndf           // NDF of the vertex fit
    - edm4hep::Vector4f   position      // position [mm] + time t0 [ns] of the vertex. Time is 4th c
    - edm4eic::Cov4f      positionError // Covariance matrix of the position+time. Time is 4th compo
    - edm4hep::Vector3f   parentMomentum             // parent momentum
    - float               parentInvariantMass        // parent invariant mass
    - float               parentInvariantMassError   // parent invariant mass error
    - float               parentDecayLength          // parent decay length
    - float               parentDecayLengthError     // parent decay length error
    - float               parentDca2PV               // parent dca to primary vertex
    - float               parentDca2PVError          // parent dca_error to primary vertex
  VectorMembers:
    - edm4hep::Vector3f   daughterMomentum           // daughter track momentum at the decay vertex
    - int                 daughterPDG                // daughter PDG
    - float               daughterDca2PV             // daughter dca to primary vertex
    - float               daughterDca2PVError        // daughter dca_error to primary vertex
    - int                 daughterPairIndices        // track indices for any pair
    - float               daughterPairDca            // dca between any pair of tracks
    - float               daughterPairDcaError       // dca_error between any pair of tracks
  OneToOneRelations:
    - edm4eic::Vertex     primaryVertex              // associated primary vertex
  OneToManyRelations:
    - edm4eic::ReconstructedParticle associatedParticles // particles associated to this vertex.
```

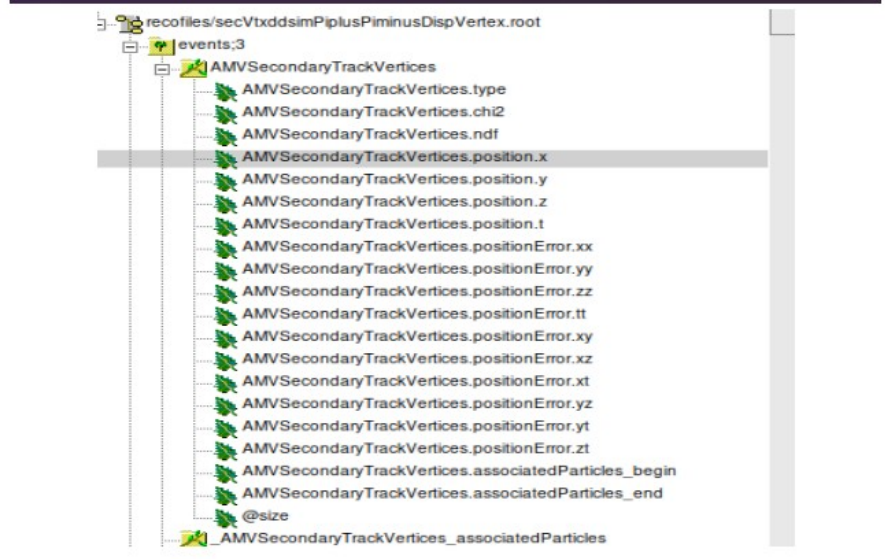
New structure has been integrated to EICrecon locally and tested to work well!
A new brunch add-secondaryvertex on edm4eic repo.

Secondary Track Vertex Factory

Bishoy's slides

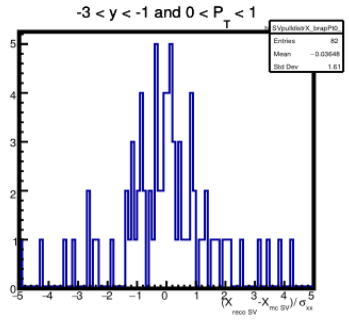
```cpp
#include <vector>

#include "algorithms/tracking/SecondaryVertexFinderConfig.h"
#include "algorithms/tracking/SecondaryVertexFinder.h"
#include "extensions/jana/JOmniFactory.h"
#include <spdlog/logger.h>

namespace eicrecon {

class SecondaryVertexFinder_factory
    : public JOmniFactory<SecondaryVertexFinder_factory, SecondaryVertexFinderConfig> {

private:
  using AlgoT = eicrecon::SecondaryVertexFinder;
  std::unique_ptr<AlgoT> m_algo;

  PodioInput<edm4eic::ReconstructedParticle> m_reco_input{this};
  Input<ActsExamples::Trajectories> m_acts_trajectories_input{this};
  PodioOutput<edm4eic::Vertex> prm_vertices_output{this};
  PodioOutput<edm4eic::Vertex> sec_vertices_output{this};

  ParameterRef<int> m_maxVertices{this, "maxVertices", config().maxVertices,
                                  "Maximum num vertices that can be found"};
  ParameterRef<bool> m_reassignTracksAfterFirstFit{
      this, "reassignTracksAfterFirstFit", config().reassignTracksAfterFirstFit,
      "Whether or not to reassign tracks after first fit"};
  ParameterRef<float> m_tracksMaxZinterval{this, "tracksMaxZinterval", config().tracksMaxZint
erval,
                                           "Max z interval for Acts::AdaptiveMultiVertexFinde
r."};
  ParameterRef<float> m_maxIterations{this, "maxIterations", config().maxIterations,
                                      "Max iterations for Acts::AdaptiveMultivertexFinder"};
  ParameterRef<float> m_maxDistToLinPoint{
      this, "maxDistToLinPoint", config().maxDistToLinPoint,
      "Max disttance to line point (pca) for Acts::AdaptiveMultivertexFinder"};

  Service<ACTSGeo_service> m_ACTSGeoSvc{this};

public:
  void Configure() {
    m_algo = std::make_unique<AlgoT>();
    m_algo->applyConfig(config());
    m_algo->init(m_ACTSGeoSvc().actsGeoProvider(), logger());
  }

  void ChangeRun(int32_t /*run_number*/) {}

  void Process(int32_t /*run_number*/, uint64_t /*event_number*/) {
    std::tie(prm_vertices_output(), sec_vertices_output()) =
        m_algo->produce(m_reco_input(), m_acts_trajectories_input());
  }
};

} // namespace eicrecon
```

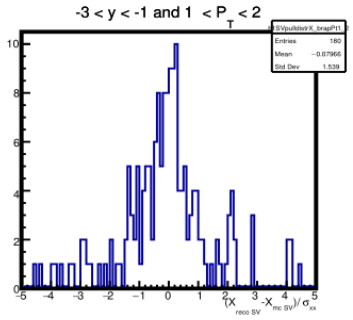
```cpp
app->Add(new JOmniFactoryGeneratorT<SecondaryVertexFinder_factory>(
    "SecondaryTrackVerticesAMVF", {"ReconstructedParticles", "CentralCKFActsTrajectories"},
    {
        "PrimaryTrackVerticesAMVF",
        "SecondaryTrackVerticesAMVF",
    },
    {}, app));
```

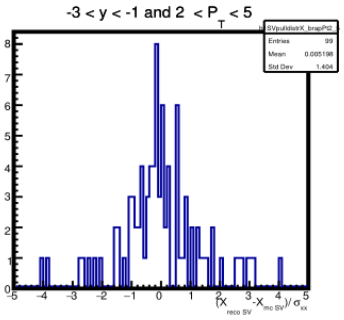


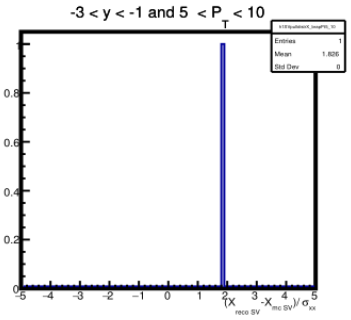
AdaptiveMultiVertexFinder (AMVF)

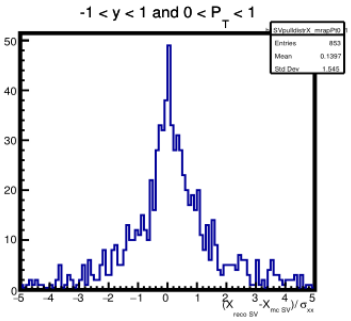
Considers track parameters and covariance (fitting approach: Kalman)

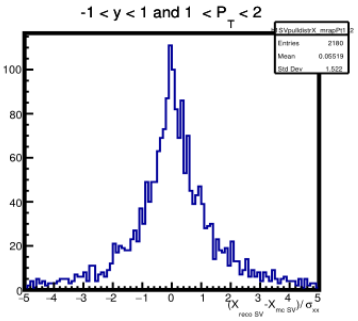


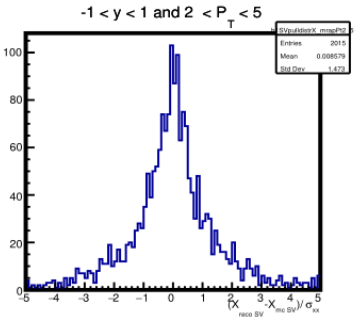


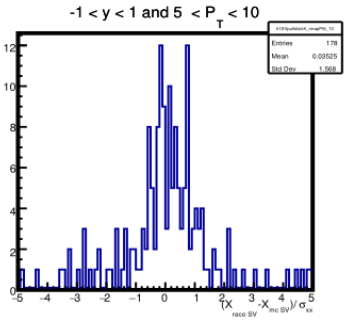


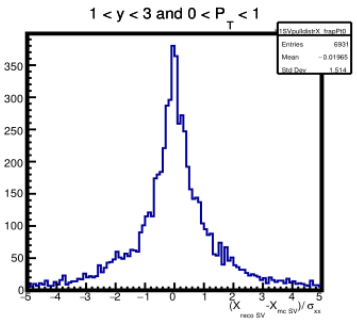


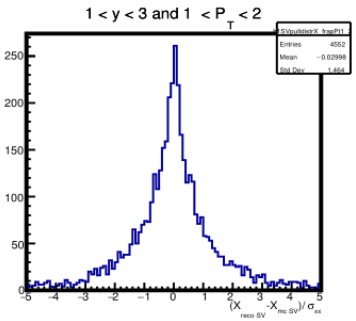


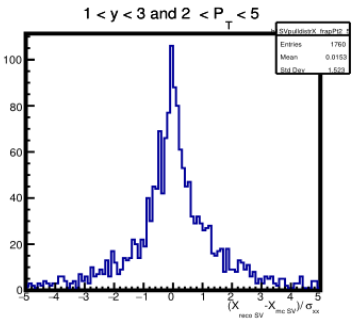


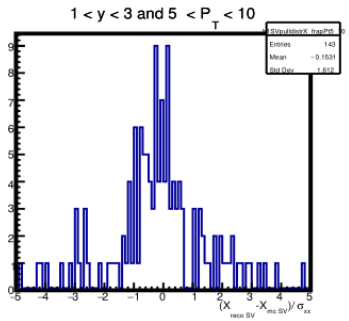












B. Dongwi

$\Lambda^0 \rightarrow p\,\pi^-$

Slide 16

Pulls are ~1.5

Apply on D^0 sample

Also presented invariant mass

Information used: **Track parameters, time, and Covariance**

Steps to be Performed

- Explicitly follow the Xin's Podio structure on slide 4
- We need following information
  - Primary vertex position and corresponding covariance in each event (Exists from CentralTrackVertices)
  - Secondary vertex position and covariance for each pair of tracks (AMVF from Bishoy: Slide 5)
  - DCA can be calculated using PV and track information in ACTS (pca_global_impactpoint.C) or helix method
  - All the topological cut on slide 4 relies on PV, SV position and it's covariances and can be calculated easily
  - Implement it in to same structure so that it can be run together on same files
- Validation:
  - Run locally on few files: activate both the helix swimming and AMVF and compare the topological distributions
  - You can also the run the existing analysis code (currently in use) on the same files
  - Later compare the topological distributions and invariant mass plots: the results must be consistent with in few % of deviations
- How to handle three/four daughters? For AMVF is easier, while Xin assign the mid point in the case of helix swimming