Details of Primary Vertexing Benchmark

14th November 2024

Khushi Singla
Lokesh Kumar
Department of Physics, Panjab University
Chandigarh, India

Overview

- The vertexing benchmark that produces the plots related to vertexing studies for DIS events was recently merged to the existing detector benchmark via PR#101, https://github.com/eic/detector_benchmarks/pull/101
- Two scripts named ‘vtx_dis_analysis.cxx’ and ‘vtx_dis_plots.cxx’ have been added at, https://github.com/eic/detector_benchmarks/tree/master/benchmarks/tracking_performances_dis/analysis
- Script ‘vtx_dis_analysis.cxx’ stores the results in a root file while ‘vtx_dis_plots.cxx’ produces the final plots.

Details of ‘vtx_dis_analysis.cxx’

- All the required collections are accessed and the histograms are defined in the beginning of the script.
- In event loop,

1. The MC event vertex is found using scattered electron vertex, https://github.com/eic/detector_benchmarks/blob/master/benchmarks/tracking_performances_dis/analysis/vtx_dis_analysis.cxx#L205-L227

```
//Finding MC vertex using scattered electron
TVector3 mcEvtVtx(-999., -999., -999.);
for(unsigned int i=0; i<mcGenStat.GetSize(); i++)
{
        if(mcGenStat[i] != 1) continue;
        if(mcPDG[i] != 11) continue;

        bool scatEfound = false;
        // mcParentBegin and mcParentEnd specify the entries from _MCParticles_parents.index
        // _MCParticles_parents.index stores the MCParticle index
        for(unsigned int j=mcParentBegin[i]; j<mcParentEnd[i]; j++)
        {
        int parentPDG = mcPDG[mcParentIndex[j]];
        if(parentPDG == 11) scatEfound = true;
        }

        if(scatEfound == false) continue;
        //Scattered electron found
        double vtx_mc_x =  mcVtxX[i];
        double vtx_mc_y =  mcVtxY[i];
        double vtx_mc_z =  mcVtxZ[i];
        mcEvtVtx = TVector3(vtx_mc_x, vtx_mc_y, vtx_mc_z);
}
```

2. The MC tracks associated with the vertex are found by selecting charged tracks with vertices within 1e-4 mm of the MC event vertex, https://github.com/eic/detector_benchmarks/blob/master/benchmarks/tracking_performances_dis/analysis/vtx_dis_analysis.cxx#L232-L247

```
232         //Filtering MC Tracks
233         int numMCTracks=0;
234         for(unsigned int i=0; i<mcGenStat.GetSize(); i++)
235         {
236                 if(mcGenStat[i] != 1) continue;
237                 if(mcCharge[i] == 0) continue;
238
239                 TVector3 mcPartVtx(mcVtxX[i], mcVtxY[i], mcVtxZ[i]);
240                 TVector3 vtx_diff = mcPartVtx - mcEvtVtx;
241                 if(vtx_diff.Mag() > 1e-4) continue;
242
243                 TVector3 mcPartMom(mcMomX[i], mcMomY[i], mcMomZ[i]);
244                 if(fabs(mcPartMom.Eta()) > 3.5) continue;
245
246                 numMCTracks++;
247         }
248         numGenTracksHist->Fill(numMCTracks);
249
```

Details of ‘vtx_dis_analysis.cxx’

3. The reconstructed event vertex is accessed. In case of 2 vertices, the vertex closer to MC vertex is selected.

4. The number of reconstructed tracks associated with the selected vertex are obtained as well, https://github.com/eic/detector_benchmarks/blob/master/benchmarks/tracking_performances_dis/analysis/vtx_dis_analysis.cxx#L256-L279

5. The 2D histograms of vertex resolution are filled. For vertexing efficiency, number of tracks for only the events with a non-zero vertex count is filled, https://github.com/eic/detector_benchmarks/blob/master/benchmarks/tracking_performances_dis/analysis/vtx_dis_analysis.cxx#L281-L298

```
//Finding Reconstructed Vertex and Vertexing Efficiency
int nVtx=0;
float diff=999.;
int nAssoPart=0;
TVector3 recoEvtVtx(-999., -999., -999.);
for(unsigned int i=0; i<recoVtxType.GetSize(); i++)
{
        nVtx++;

        TVector3 recoVtx(recoVtxX[i], recoVtxY[i], recoVtxZ[i]);

        //Finding the reconstructed vertex closer to the MC vertex
        TVector3 vtx_diff = recoVtx - mcEvtVtx;
        if(vtx_diff.Mag() < diff)
        {
        diff = vtx_diff.Mag();
        recoEvtVtx = recoVtx;

        for(unsigned int j=assoPartBegin[i]; j<assoPartEnd[i]; j++)
        {
        nAssoPart = j;
        }
        }
}
```

```
recoVtxEffHist->Fill(nVtx);
recoVtxYvsXHist->Fill(recoEvtVtx.x(), recoEvtVtx.y());
TVector3 recoRadius(recoEvtVtx.x(), recoEvtVtx.y(), 0);
recoVtxRvsZHist->Fill(recoEvtVtx.z(), recoRadius.Mag());

vtxResXvsGenTrkHist->Fill(numMCTracks, recoEvtVtx.x() - mcEvtVtx.x());
vtxResYvsGenTrkHist->Fill(numMCTracks, recoEvtVtx.y() - mcEvtVtx.y());
vtxResZvsGenTrkHist->Fill(numMCTracks, recoEvtVtx.z() - mcEvtVtx.z());

vtxResXvsRecoTrkHist->Fill(nAssoPart, recoEvtVtx.x() - mcEvtVtx.x());
vtxResYvsRecoTrkHist->Fill(nAssoPart, recoEvtVtx.y() - mcEvtVtx.y());
vtxResZvsRecoTrkHist->Fill(nAssoPart, recoEvtVtx.z() - mcEvtVtx.z());

if(nVtx !=0) {
numGenTrkswithVtxHist->Fill(numMCTracks);
numRecoTrkswithVtxHist->Fill(recoType.GetSize());

recoVsMCTracksHist->Fill(numMCTracks, nAssoPart);}
```

Details of ‘vtx_dis_plots.cxx’

- First, the histograms are read from the root file.
- The 2D histograms of vertex resolution are fitted using Student’s t-distribution which is a Gaussian function with a tail. It is defined as,

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

- Also, vertexing efficiency is plotted in each track bin.
- In the end, all the plots are drawn on canvases and stored in a pdf file.
- https://github.com/eic/detector_benchmarks/blob/master/benchmarks/tracking_performances_dis/analysis/vtx_dis_plots.cxx

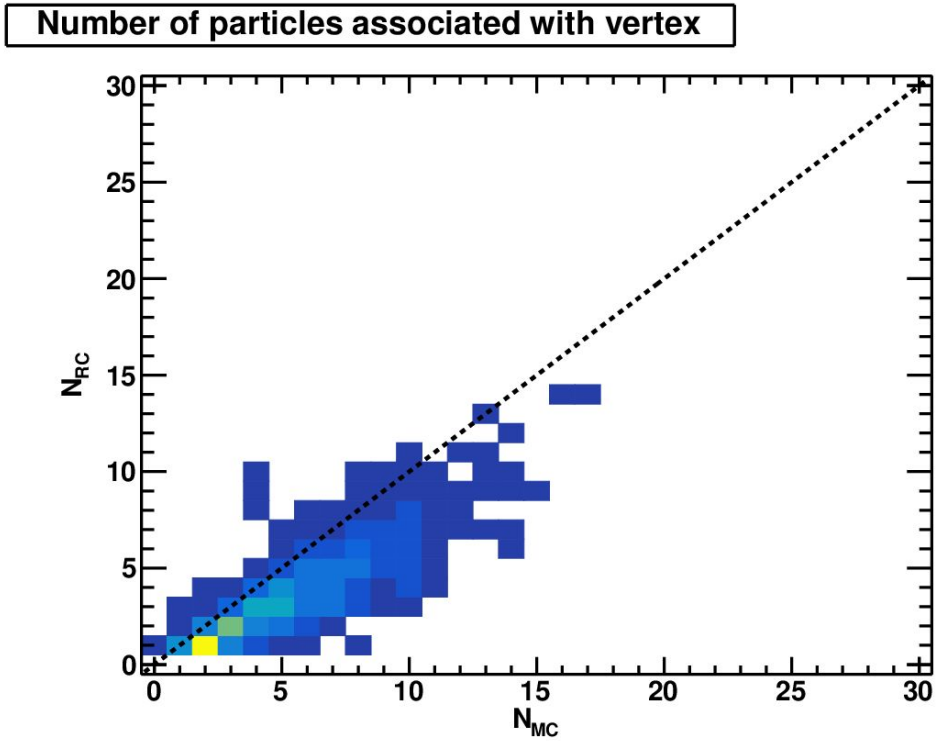
Results: Test plots

- The resulting test plots have been generated for 1000 events with:
  - Electron Beam Energy: 18 GeV
  - Proton Beam Energy: 275 GeV
  - $Q^2 > 1\ GeV^2$

- They can be viewed at, https://eicweb.phy.anl.gov/EIC/benchmarks/detector_benchmarks/-/jobs/4122425/artifacts/browse/results/vertexing_performances_dis/epic_craterlake_tracking_only/pythia8NCDIS_18x275_minQ2=1_combined_5/

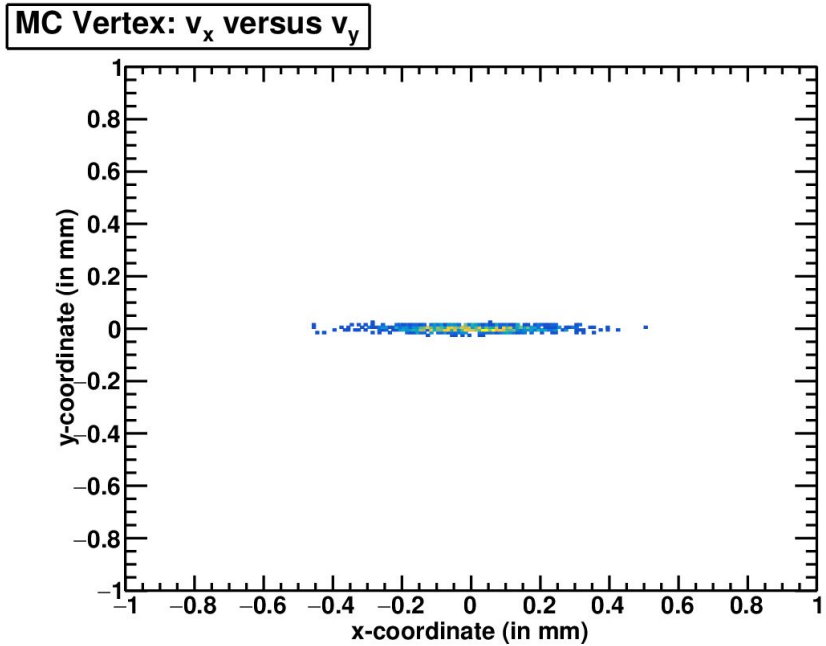
Results: plot_001.png

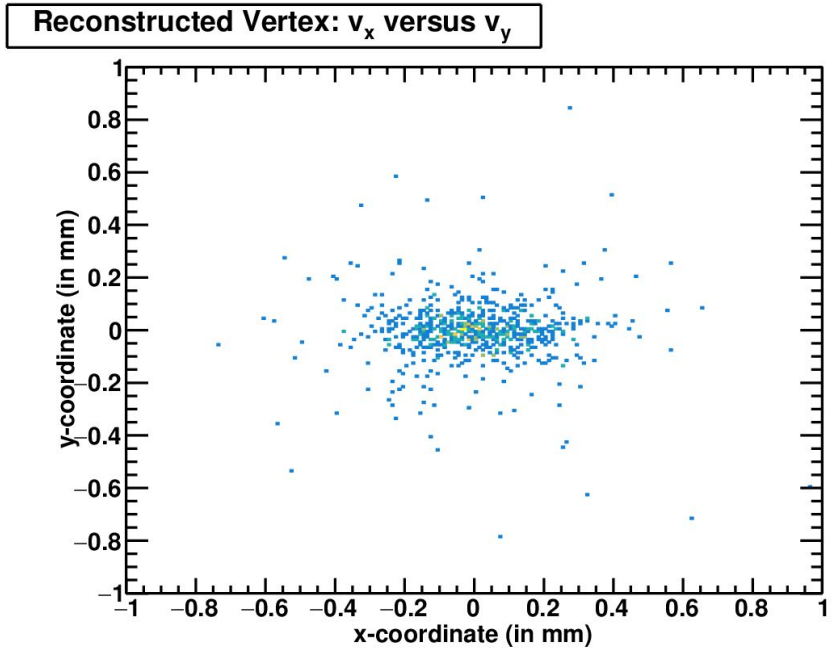
- Number of tracks associated with vertex



Results: plot_003.png and plot_005.png

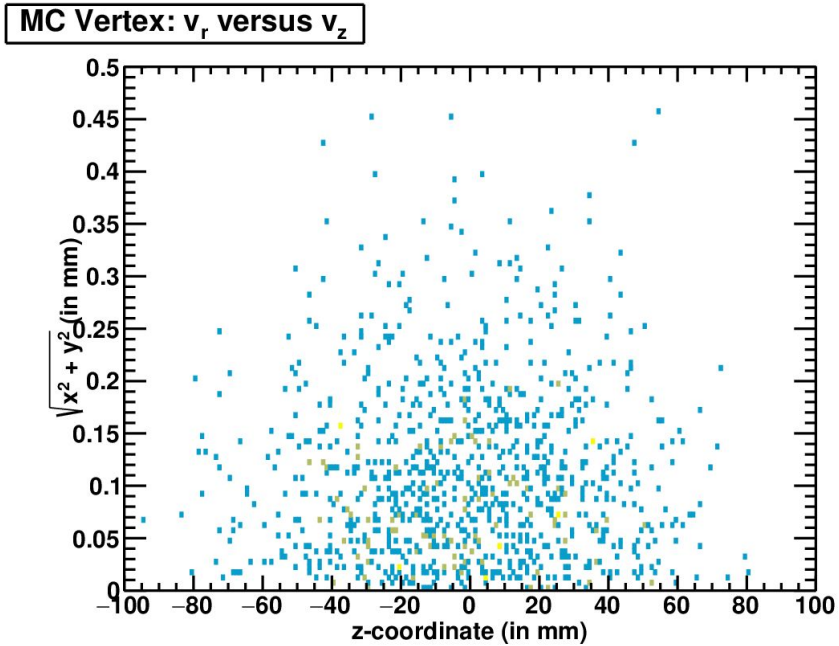
- Vertex position: v_x versus v_y

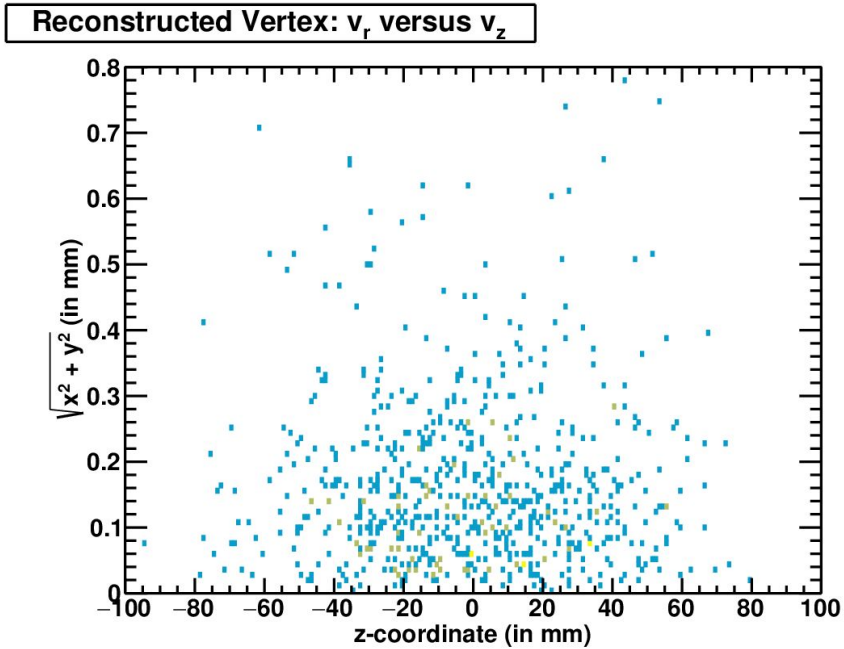




Results: plot_004.png and plot_006.png

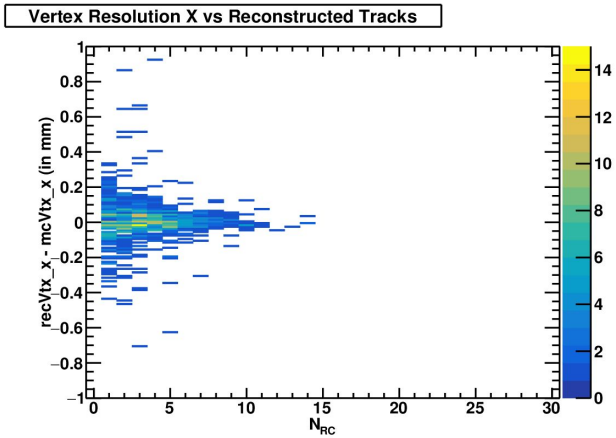
- Vertex position: v_r versus v_z

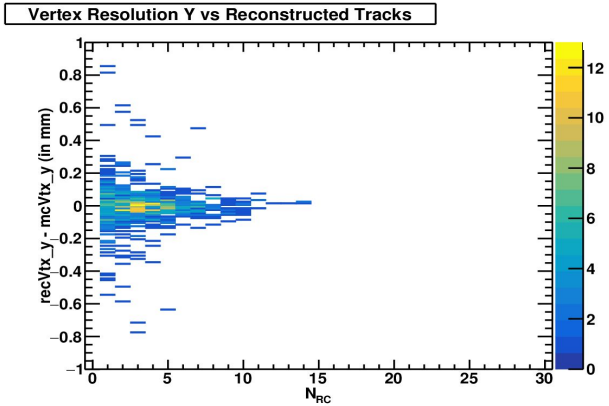


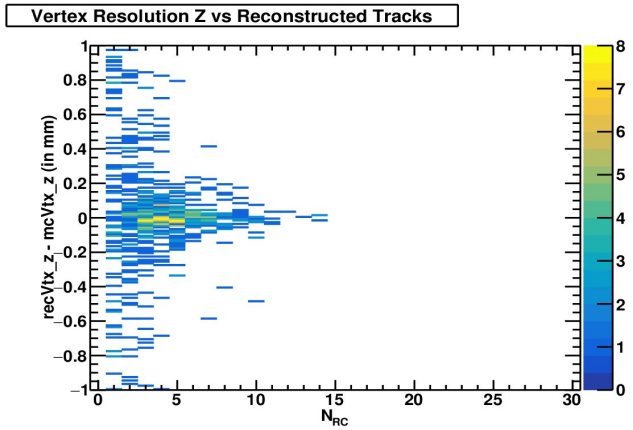


Results: plot_007.png, plot_008.png and plot_009.png

- Vertex resolution versus resconstructed particles

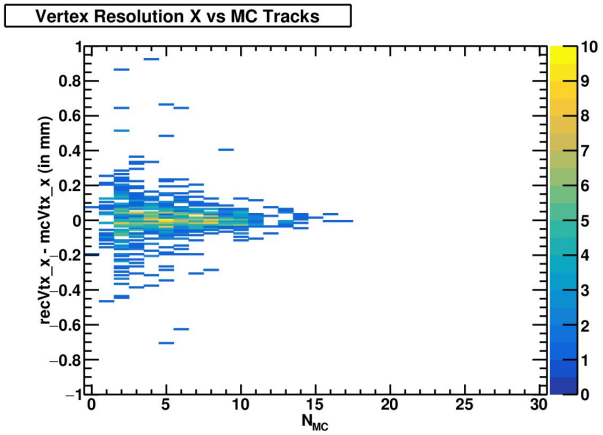


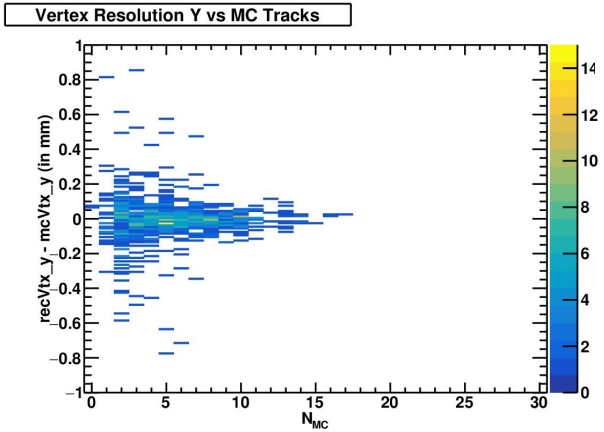


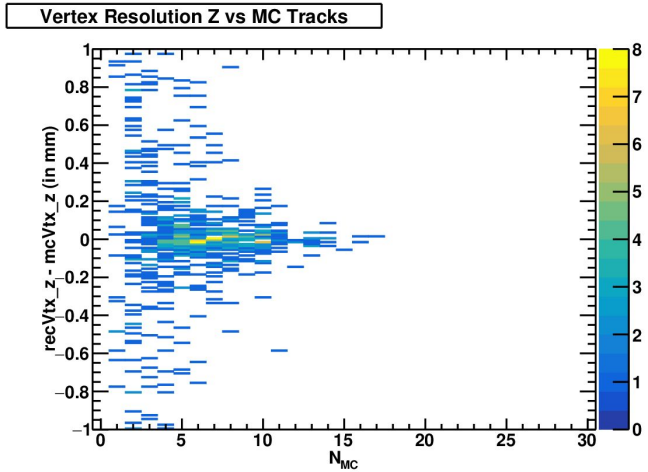


Results: plot_010.png, plot_011.png and plot_012.png

- Vertex resolution versus MC particles

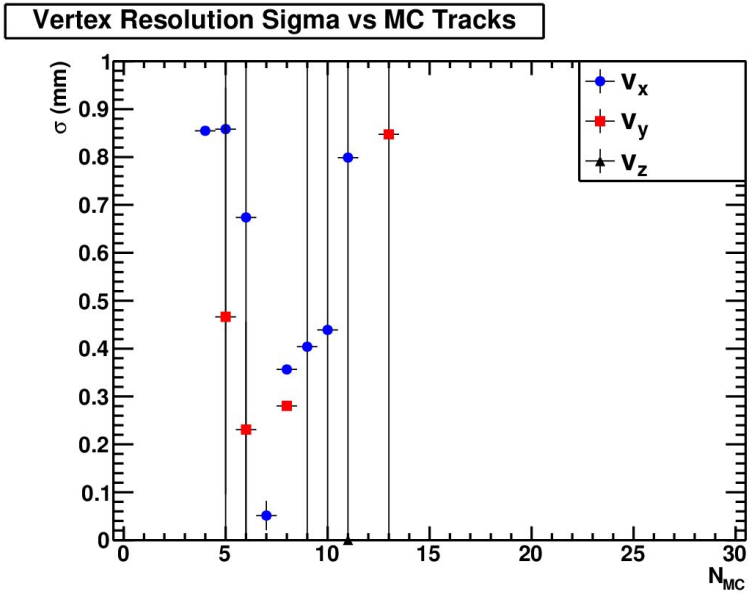


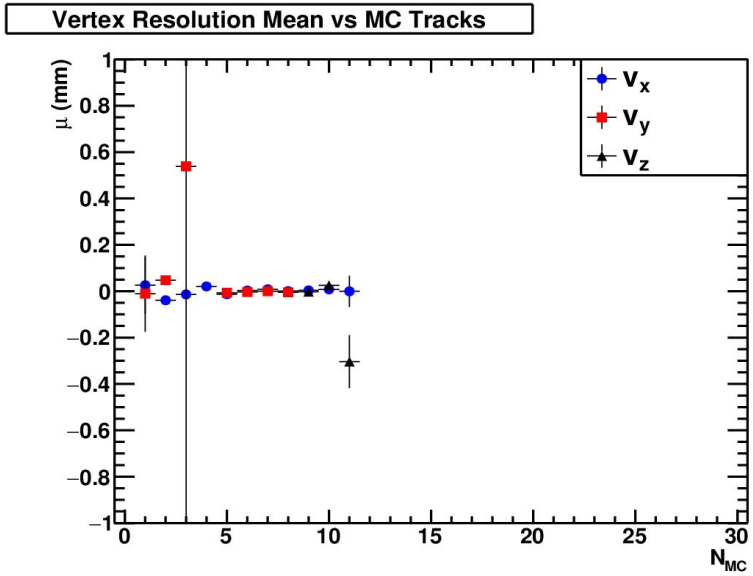




Results: plot_013.png and plot_014.png

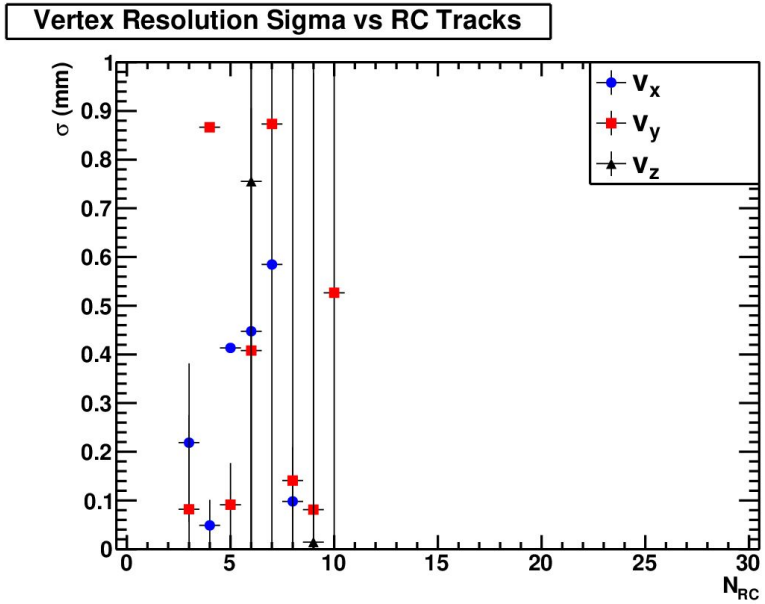
- Vertex resolution versus MC particles (Sigma and Mean)

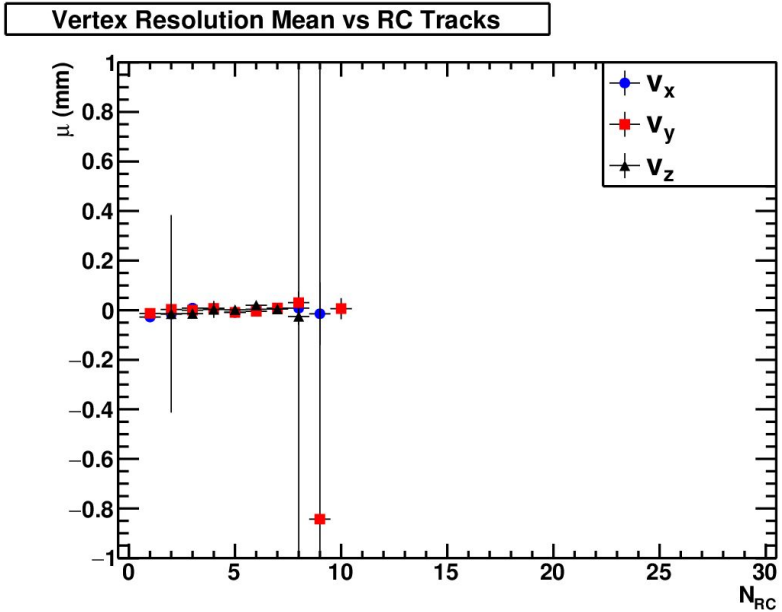




Results: plot_015.png and plot_016.png

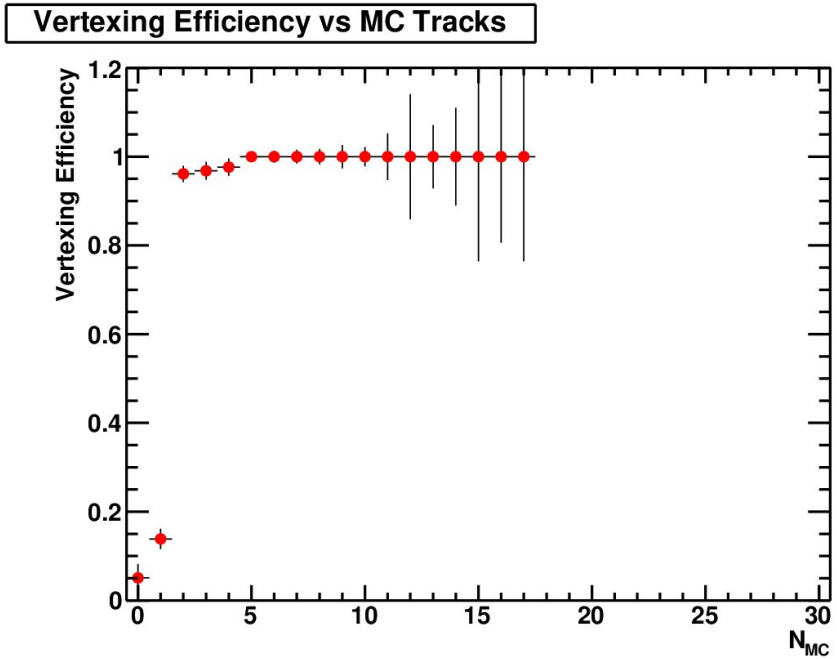
- Vertex resolution versus resconstructed particles (Sigma and Mean)

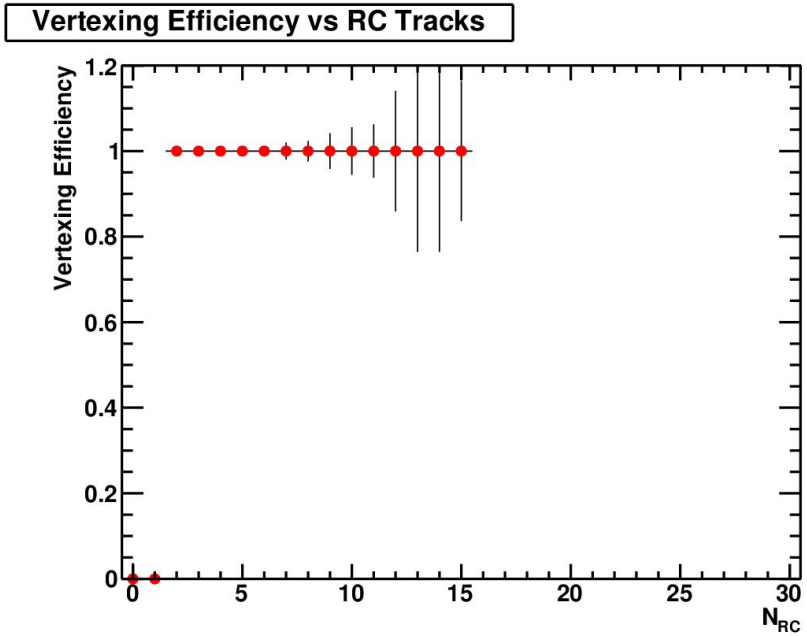




Results: plot_017.png and plot_018.png

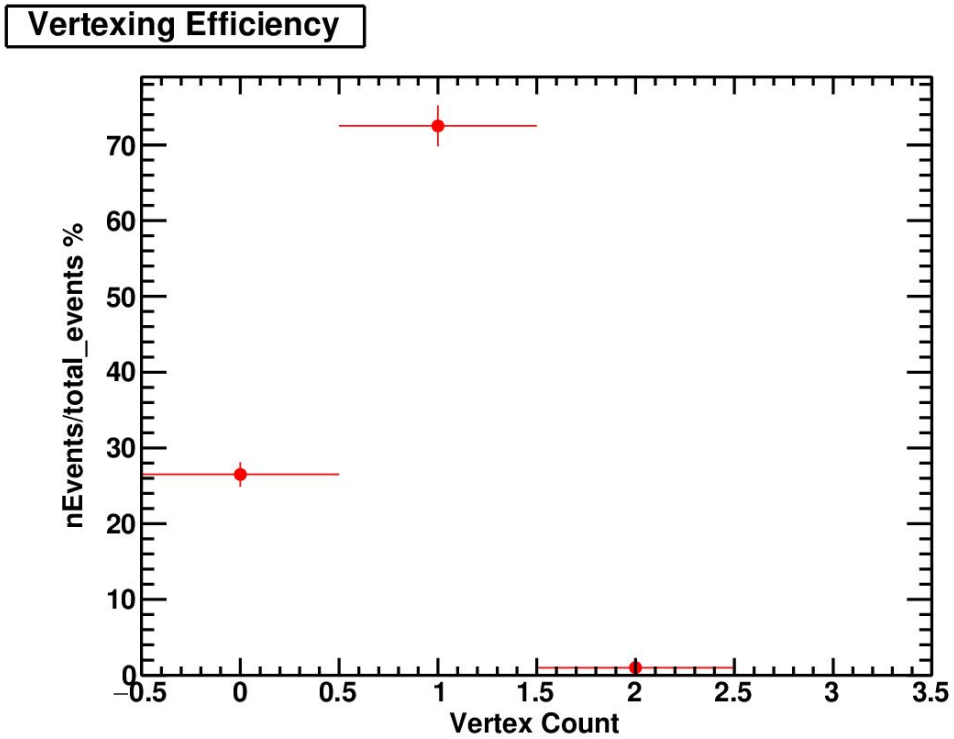
- Vertexing Efficiency versus MC and RC tracks





Results: plot_002.png

- Vertexing Efficiency



Thank You!