Analysis wagon for the anisotropic flow measurements in MPD

Alexander Demanov, Petr Parfenov

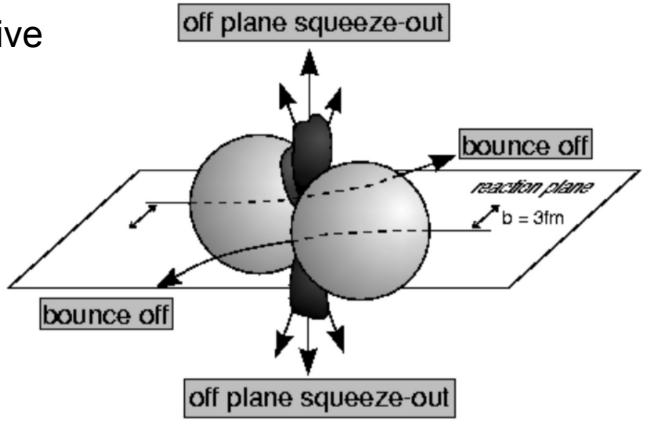
05.03.2024

Flow measurements: event plane method (EP)

The azimuthal angle distribution is decomposed in a Fourier series relative to reaction plane angle:

$$\rho(\varphi - \Psi_{RP}) = \frac{1}{2\pi}(1 + 2\sum_{n=1}^{\infty} v_n \cos n(\varphi - \Psi_{RP}))$$

Anisotropic flow: $v_n = \langle \cos\left[n(\varphi - \Psi_{RP})\right] \rangle$



Event plane method:

$$Q_{n,x} = \sum_i w_i \cos(n\phi_i)$$
$$Q_{n,y} = \sum_i w_i \sin(n\phi_i)$$

$$\Psi_n^{EP} = \frac{1}{n}\tan^{-1}\left(\frac{Q_{n,y}}{Q_{n,x}}\right)$$

$$v_n = \frac{<cos[n(\phi - \Psi_n^{EP})]>}{Res(\Psi_n)}$$

From evPlane wagon → From evFLowEP wagon

Method's details described in Phys.Rev.C 77 (2008) 034904

evFlowEP wagon: how to run

- evFlowEP wagon well be implemented in MpdRoot (mpdroot/physics/evFlowEP/)
- Example of RunAnalyses.C macro (in evFlowEP/macros/):

```
void RunAnalyses(int nEvents = -1)
{
  gSystem->Load("libZdc.so") ;
  gSystem->Load("libMpdPhysics.so") ;

  MpdAnalysisManager man("ManagerAnal", nEvents) ;
  man.InputFileList("list.txt") ;
  man.ReadBranches("*") ;
  man.SetOutput("histos_flow.root") ;

  MpdCentralityAll pCentr("pCentr","pCentr") ;
  man.AddTask(&pCentr) ;

  MpdEventPlaneAll pEP("pEP","pEP") ;
  man.AddTask(&pEP) ;

  MpdFlowEventPlane pFlowEP("pFlowEP","pFlowEP") ;
  man.AddTask(&pFlowEP) ;

  man.Process() ;

}
```

Loading needed mpdroot libraries (detectors that will be used, etc.)

List of input MpdDst files

wagon's name

Set I/O names (input: pCentr.txt, output: pCentr.root)

Add Centrality wagon to the train

Add event plane wagon to the train

Add Flow wagon to the train

Run the train (process MpdDst data)

Addition: if you measure flows for pid (default is PDG code) -> Add evPID wagon

evFlowEP wagon: input file configuration

pFlowEP.txt :

```
-------Parameters used for analysis------
# Event selection:
mZvtxCut 100 #  cut on vertex z coordinate

# Track selection:
mNofHitsCut   16  # minimal number of hits to accept track
mEtaCut       1.5  # maximal pseudorapidity accepted
mEtaGapCut  0.1  # minimal pseudorapidity accepted: abs(eta)>0.05 for mEtaGap=0.1
mPtminCut     0.1  # minimal pt used in analysis
mPtmaxCut    2.0  # maximal pt used in analysis
mDcaCut       2.0  # maximal DCA accepted
mPidPDG       1     # 1 - use PDG-code to PID; 0 - use evPID wagon
mPIDsigTPC   3.0 # n-sigma (mPidPDG=0)
mPIDsigTOF   3.0 # n-sigma (mPidPDG=0)
```

evFlowEP wagon: structure

The main class in the evFlowEP wagon is **MpdFlowEventPlane**.

MpdFlowEventPlane::UserInit() performs procedures that are needed before the event loop:

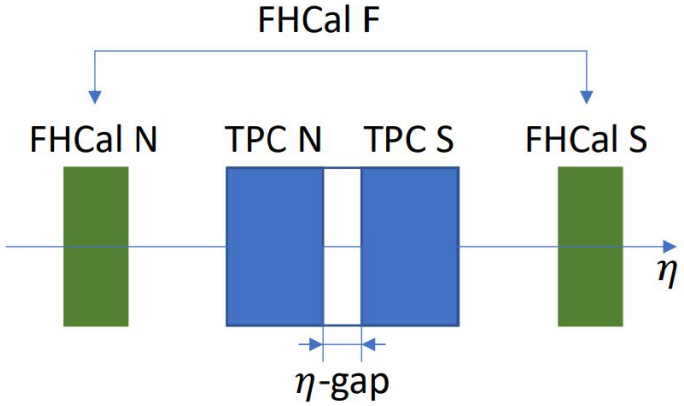
- Read input config file
- Initialize output QA histograms:
  - Basic QA (event count, vtxZ, track's parameters:p_T,η,N_{hits}, DCA)
  - TProfiles2D for <cos[n(φ-Ψ_n)]> (p_T,η(y)) for each centrality bin
  - TProfiles for Res(Ψ_n) for each centrality bin

MpdFlowEventPlane::ProcessEvent(MpdAnalysisEvent &event) performs all analysis in the event.
It consists of:

- Applying event selection (vtxZ cut from the input file)
- Getting centrality from evCentrality wagon (from event.getCentrTPC())
- Getting the event plane angle from evPlane wagon
- Applying tracks selection
- Filling TProfiles for Res(Ψ_n) and TProfiles2D for <cos[n(φ-Ψ_n)]>

evFlowEP wagon: directed flow

$$v_1 = \frac{<cos(\phi - \Psi_1^{EP})>}{Res(\Psi_1)}$$



$\Psi_1^{FHCal\ S,N}$, $\Psi_1^{FHCal\ F}$ from evPlane wagon, **Ψ^{RP}** from data

<cos[(φ-Ψ_1)]> (p_T,y) for each cent bin:

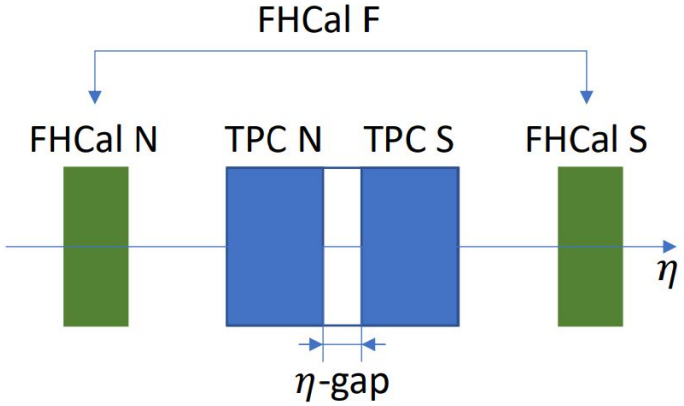
- <cos($\varphi^{N,S}$-$\Psi_1^{FHCal\ S,N}$)> (v1_FHCal_NS)
- <cos(φ-Ψ_1^{F})> (v1_FHCal_F)
- <cos(φ-Ψ^{RP})> (v1_RP)

Resolution, Res(Ψ_1) for each cent bin:

- For v1_FHCal_NS: sqrt(<cos($\Psi_1^{FHCal\ N}$-$\Psi_1^{FHCal\ S}$)>);
- For v1_FHCal_F: approximation to the full FHCal from sqrt(<cos($\Psi_1^{FHCal\ N}$-$\Psi_1^{FHCal\ S}$)>)

evFlowEP wagon: Elliptic flow

$$v_2(\Psi_2) = \frac{<cos[2(\phi - \Psi_2^{EP})]>}{Res(\Psi_2)} \qquad v_2(\Psi_1) = \frac{<cos[2(\phi - \Psi_{1,FHCal}^{EP})]>}{Res(\Psi_1)}$$



$\Psi_1^{FHCal\ S,N}$, $\Psi_1^{FHCal\ F}$, $\Psi_2^{TPC\ S,N}$ from evPlane wagon, **Ψ^{RP}** from data

<cos[2(φ-Ψ_n)]> (p_T,y) for each cent bin:

- <cos[2($\varphi^{N,S}$-$\Psi_2^{TPC\ S,N}$)]> (v2_TPC_NS)
- <cos[2($\varphi^{N,S}$-$\Psi_1^{FHCal\ S,N}$)]> (v2_FHCal_NS)

Resolution, Res(Ψ_n) for each cent bin:

- For v2_TPC_NS: sqrt(<cos($\Psi_2^{TPC\ N}$-$\Psi_2^{TPC\ S}$)>);
- For v2_FHCal_NS: sqrt(<cos($\Psi_1^{FHCal\ N}$-$\Psi_1^{FHCal\ S}$)>);

evFlowEP wagon: Output file

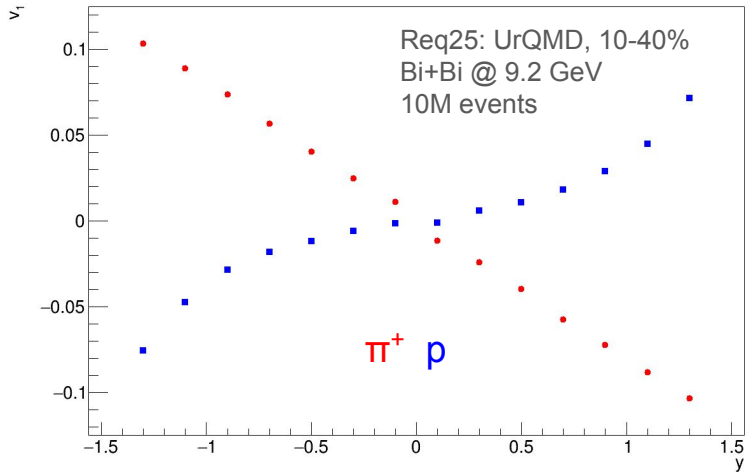
```
$ root –l pFlowEP.root
Attaching file pFlowEP.root as _file0...
(TFile *) 0x55e1bb4d2a00
root [1] .ls
TFile**         pFlowEP.root
 TFile*         pFlowEP.root

  KEY: TH1D     mhEvents;1      Number of events
  KEY: TH1D     hVertex;1       Event vertex distribution
  KEY: TH1D     mhHits;1        Number of TPC hits
  KEY: TH1D     mhEta;1         Eta
  KEY: TH1D     mhDca;1         DCA
…
 KEY: TProfile  mhCos1FHCalFPsiRP_010;1    <cos(#Psi_{1}^{FHCal F}-#Psi_{RP})> 0-10 %
 KEY: TProfile  mhCos1FHCalSFHCalN_010;1   <cos(#Psi_{1}^{FHCal N}-#Psi_{1}^{FHCal S})> 0-10 %
 KEY: TProfile  mhCos2TpcNTpcS_010;1       <cos(2(#Psi_{2}^{TPC N}-#Psi_{2}^{TPC S}))> 0-10 %
 KEY: TProfile  mhCos2FHCalSFHCalN_010;1   <cos(2(#Psi_{1}^{FHCal N}-#Psi_{1}^{FHCal S}))> 0-10 %
KEY: TProfile   mhCos2FHCalFRP_010;1       <cos(2(#Psi_{1}^{FHCal F}-#Psi_{RP}}))> 0-10 %
…
  KEY: TProfile2D       mhCos1_PhiPsiFHCal_sNnS_010_PrP;1       <cos[(#phi^{N,S}-#Psi_{1}^{FHCal S,N})]> cent 0-10 %, PrP
  KEY: TProfile2D       mhCos2_PhiPsiTPC_sNnS_010_PrP;1         <cos[2*(#phi^{N,S}-#Psi_{2}^{TPC S,N})]> cent 0-10 %, PrP
….
```

evFlowEP wagon: Macro to calculate flow

evFlowEP/macros/ has a macro getFlow.C(input_file, output_file)
that calculates flow:

- the calculated flow
- write the result to the output file



Set graph parameters:

- Method (v1_FHCal_NS, v2_TPC_NS, …)
- Set centrality range
- Set integrated range ({“pt”,{min,max}} or {“eta”,{min,max}})
- Set pid: ch, Pi, PiP, PiM, K, KP, KM, Pr, PrP, PrM
- Set the AxisX bining

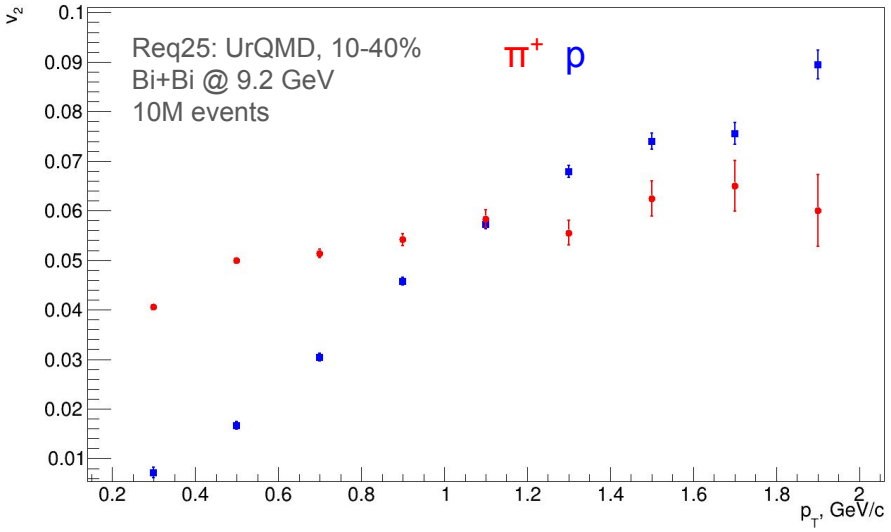
Example: v1(y) cent 10-40% for π^+ and p

```
SaveGraphInFile(fi,fout,"v1_FHCal_NS",{{10,20},{20,30},{30,40}}, { "pt",{ 0.2 ,2.0}},"PiP",EtaBin)
SaveGraphInFile(fi,fout,"v1_FHCal_NS",{{10,20},{20,30},{30,40}}, { "pt",{ 0.4 ,2.0}},"PrP",EtaBin)
where std::vector<double> EtaBin    = {-1.4,-1.2,-1.0,-0.8,-0.6,-0.4,-0.2,0.,0.2,0.4,0.6,0.8,1.0,1.2,1.4};
root -l -b -q getFlowEP.C’(“pFlowEP.root”,”out_graph.root”)’
```

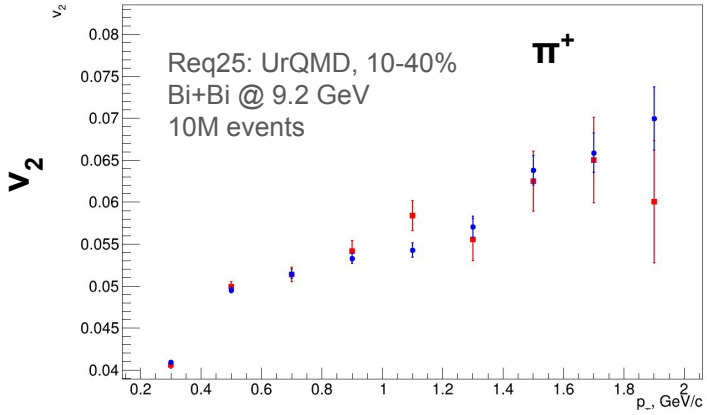
evFlowEP wagon: example (v_2)

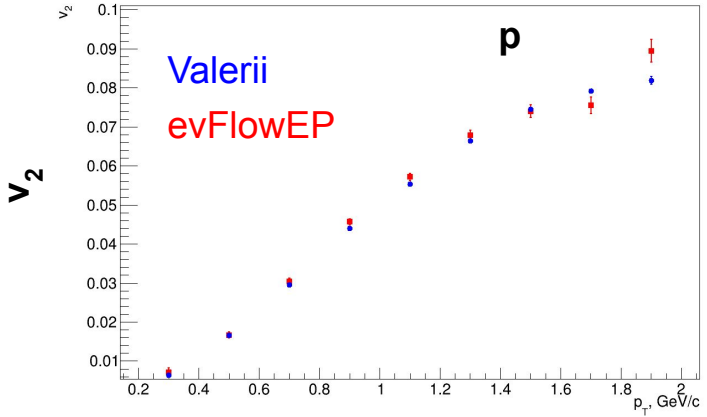
Example: v2(pT) cent 10-40% for π^+ and p

```
SaveGraphInFile(fi,fout,"v2_TPC_NS",{{10,20},{20,30},{30,40}}, { "eta",{ -1.4 ,1.4}},"PiP",PtBin)
SaveGraphInFile(fi,fout,"v2_TPC_NS",{{10,20},{20,30},{30,40}}, { "eta",{ -1.4 ,1.4}},"PrP",PtBin)
std::vector<double> PtBin     ={0.2,0.4,0.6,0.8,1.0,1.2,1.4,1.6,1.8,2.0};
```



evFlowEP wagon: Comparison with old results





From Valerii Troshin;
mr-viaro@yandex.ru

Request 25, BiBi@9.2GeV
UrQMD

cuts:

pT (0.2;2) GeV/c

rapidity (0.1;1.4)

centrality 10-40%

number of hits in TPC >16

DCA <2 cm

PID: PDG code

MotherId = -1

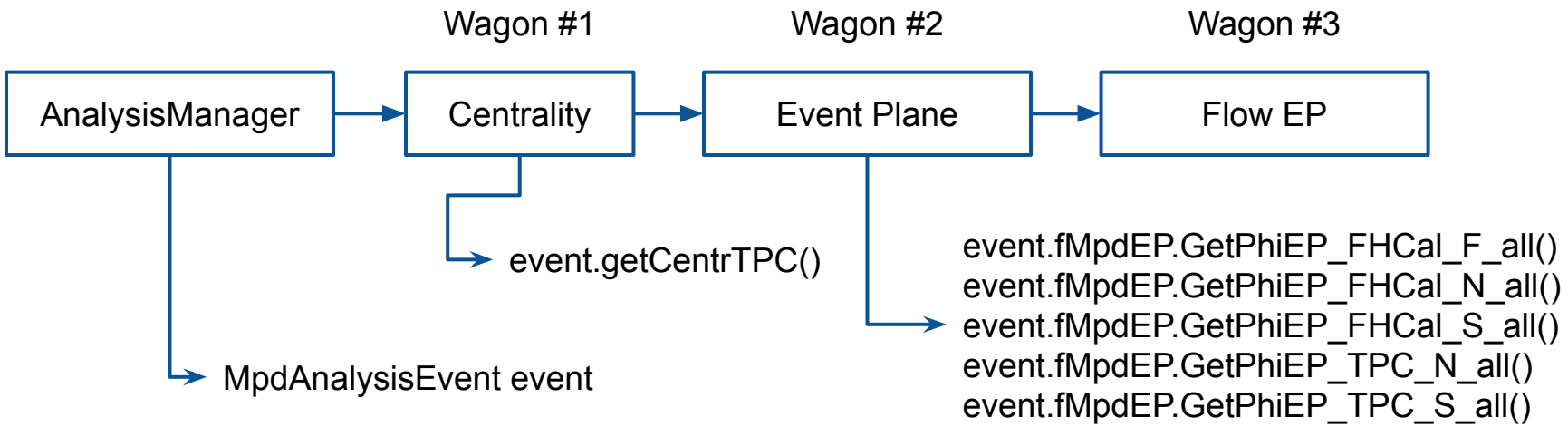
Conclusions

- The evFlowEP wagon is presented
- Comparison with previously results (Valerii results):
  - good agreement for v_2
  - not finished for v_1

backup

Main idea of the wagons

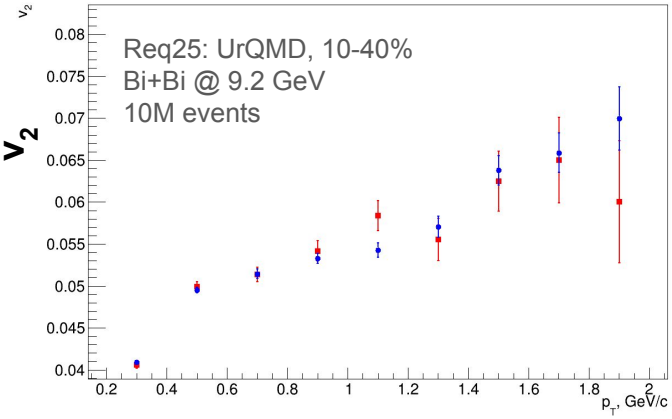
- All analyses are packed into wagons within the Analysis Framework
  - All wagons have similar structure, provide consistency among all analyses
  - All info from the wagons is being stored into the main class MpdAnalysisEvent



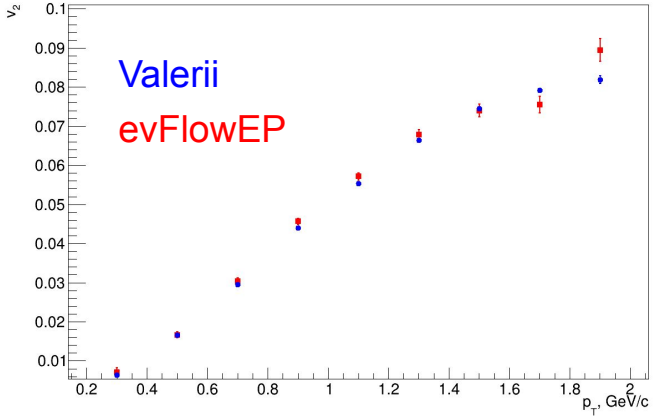
Flow EP wagon - last wagon.

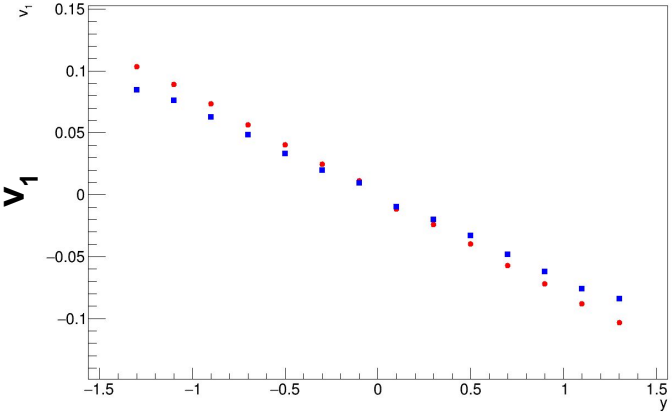
evFlowEP wagon: Comparison with old results

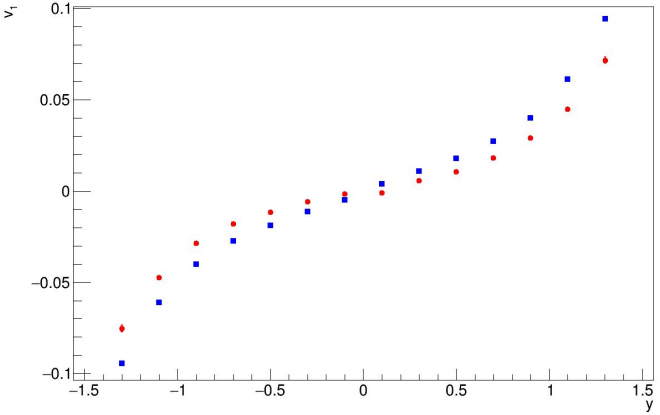
π^+



p







From Valerii Troshin; mr-viaro@yandex.ru

Request 25, BiBi@9.2GeV UrQMD

cuts:

pT (0.2;2) GeV/c

rapidity (0.1;1.4)

centrality 10-40%

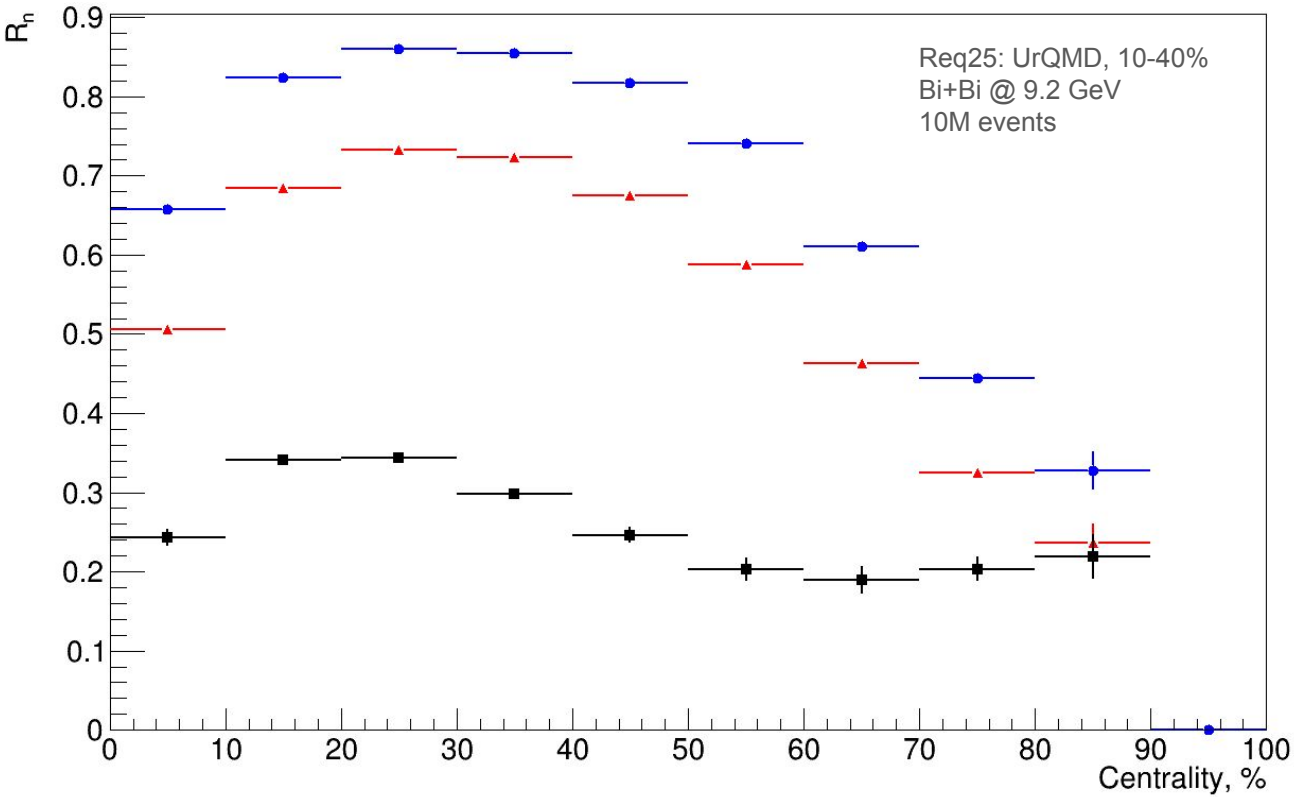
number of hits in TPC >16

DCA <2 cm

PID: PDG code

MotherId = -1

EP resolution

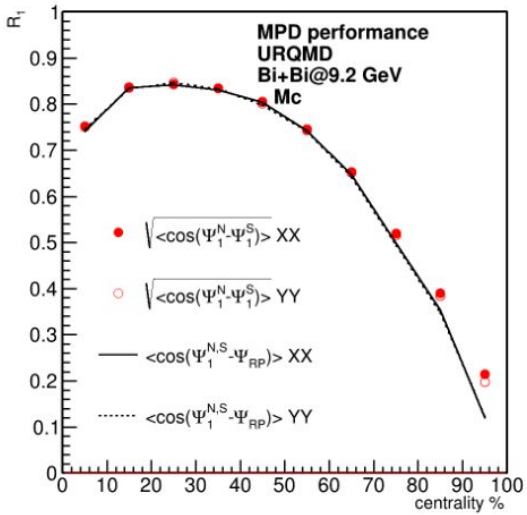


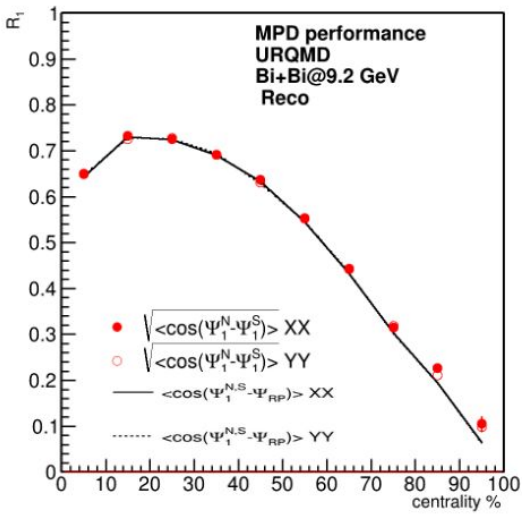
R_1 for FHCal F
R_1 for FHCal N/S
R_2 for TPC N/S

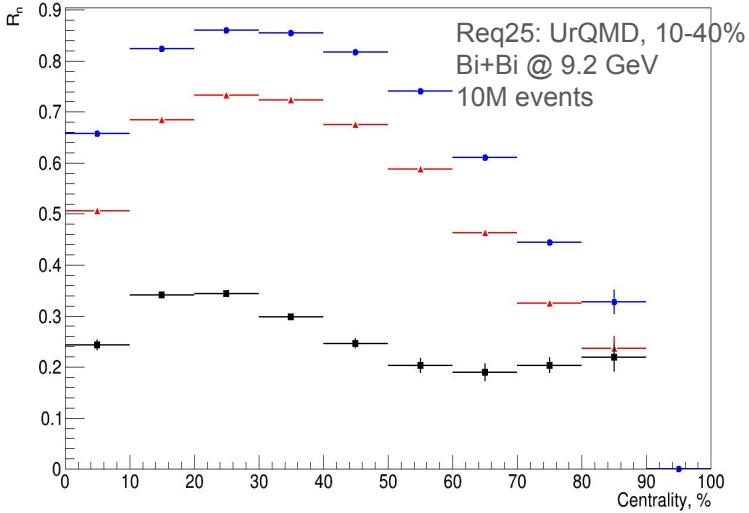
EP resolution

R_1 for FHCal F
R_1 for FHCal N/S
R_2 for TPC N/S

R_1 2 sub-event method: FHCal







Plans

- EP wagon
  - $v_{1,FHCal}(pt,y)$
  - $v_{2,TPC}(pt,y, cent)$, $v_{2,FHCal}(pt,y, cent)$
- SP wagon (in the future)
- Q-Cumulant wagon (in the future)
  - $v_n\{2\}$, $v_n\{4\}$, $v_n\{6\}$, $n>1$