Centrality questions & answers - III

V. Riabov

Implementation of centrality wagon

Centrality wagon

- Centrality wagon: mpdroot/physics/evCentrality
- Currently works with Request25_UrQMD, Request26_DCM-SMM and Request31_PHSD
- Other productions/event generators can be added by request
- Calculates TPC centrality for all accepted events
- Returns centrality '-1' for rejected events (not to be included in the analyses) :
  - ✓ empty events (UrQMD, PHSD)
  - ✓ events with no vertex by TPC
  - ✓ events with reconstructed vertex |z-vertex-TPC| > 130 cm
  - ✓ events that failed to fire the FFD||FHCL trigger (assessed based on event track multiplicity)
- Centrality is provided as a float in the range [0-91] for accepted events
- All conversion tables and service information is saved in the output of the wagon
- Event centrality is available for all other wagons in the train: event.getCentrTPC();
- Example on how centrality variable is used in the analysis: mpdroot/physics/pairKK

Example

- How to run:
  - ✓ cd mpdroot/physics/pairKK/macros
  - ✓ root -b -q RunAnalyses.C

```
void RunAnalyses(){

  gROOT->LoadMacro("mpdloadlibs.C");
  gROOT->ProcessLine("mpdloadlibs()");

   MpdAnalysisManager man("ManagerAnal") ;
   man.InputFileList("list.txt") ;  // List of input DST files
   man.ReadBranches("*") ;
   man.SetOutput("histos.root") ;

   MpdCentralityAll pCentr("pCentr","pCentr") ; //Wagon #1 – Centrality: input file pCentr.txt, output pCentr.root
   man.AddTask(&pCentr) ;

//   MpdConvPi0 pDef("pi0Def","ConvDef") ;
//   man.AddTask(&pDef) ;

   MpdPairKK pKK("pKK","pKK") ; //Wagon #2 – Phi->KK: input file pKK.txt, output pKK.root
   man.AddTask(&pKK) ;

   man.Process() ;
}
```

Input file

- Input file pCentr.txt:

```
#-------Parameters used for analysis------
# Event selection:
mZvtxCut 130 //  cut on vertex z coordinate

# Track selection:
mNofHitsCut   10  // minimal number of hits to accept track
mEtaCut      0.5  // maximal pseudorapidity accepted
mPtminCut    0.1  // minimal pt used in analysis
mDcaCut      2.0  // maximal pseudorapidity accepted

# Production selection:
//mProdGenerator Req25-UrQMD  // Production-Generator
//mInFileConvert nTr_Centr_Req25-UrQMD.root  // input file with track-to-centrality converter

//mProdGenerator Req26-DCM-QGSM-SMM  // Production-Generator
//mInFileConvert nTr_Centr_Req26-DCM-QGSM-SMM.root  // input file with track-to-centrality converter

mProdGenerator Req30-PHSD  // Production-Generator
mInFileConvert nTr_Centr_Req30-PHSD.root  // input file with track-to-centrality converter

# Track efficiecny corrections:
mInFileTrEff TrackRecEff.root  // input file with track reconstruction efficiecnies
```

Output file

- Output file pCentr.root:

```
$ root -l pCentr.root
root [0]
Attaching file pCentr.root as _file0...
(TFile *) 0x3bfde40
root [1] .ls
TFile**         pCentr.root
 TFile*         pCentr.root
  KEY: TH1F     hEvents;1       Number of events → number of events after different selections
  KEY: TH1F     hVertex;1       Event vertex distribution → z-vertex-TPC for non-empty events with reconstructed vertex
  KEY: TH1F     hVertexAcc;1    Accepted event vertex distribution → z-vertex-TPC for accepted events
  KEY: TH1F     hHits;1 Number of TPC hits → nHits for accepted tracks
  KEY: TH1F     hEta;1  Eta → eta of accepted tracks
  KEY: TH1F     hPt;1   Pt → transverse momentum of accepted tracks
  KEY: TH1F     hDca;1  DCA → DCA of accepted tracks
  KEY: TH1F     hMultiplicity;1 Multiplicity distribution → multiplicity distribution (no efficiency corrections)
  KEY: TH1F     hMultiplicityEff;1      Weighted multiplicity distribution → multiplicity distribution after corrections
  KEY: TH1F     hCentrality;1   Centrality distribution  → centrality distribution for accepted events
  KEY: TH1F     hCentConvert;1  nTr-Centrality converter → number of tracks to centrality convertion table
  KEY: TH2F     hTrEff;1        Track Efficiency → track reconstruction efficiency vs. z-vertex and eta
```

How to get centrality in other wagons

- Example of pairKK wagon, mpdroot-dev/physics/pairKK/MpdPairKK.cxx:

```
float cen = event.getCentrTPC();

 if (cen < 0 || cen >= 100) { //TPC centrality not defined
    return false;
 }
```

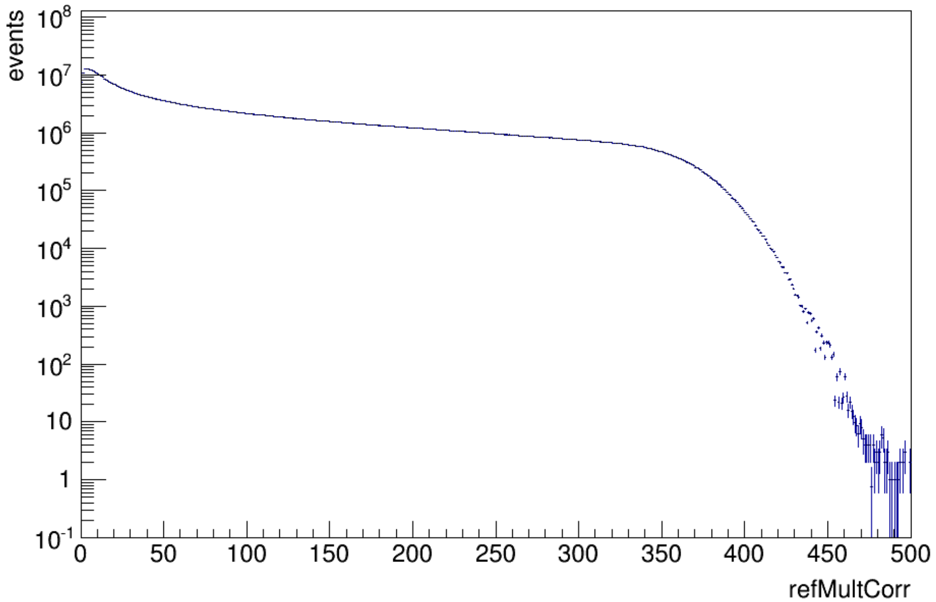
Conclusions

- TPC centrality wagon is now available in the mpdroot
- Please report any problems

Tests of the Glauber fitting procedures

Multiplicity distribution

- In fits to UrQMD/PHSD/DCM-SMM data get unphysical values of ‘f’ parameter for N_a
- Test our fitting machinery using the real data multiplicity distribution
- Multiplicity distribution by STAR for AuAu@19.6 GeV (for internal use only !!!):

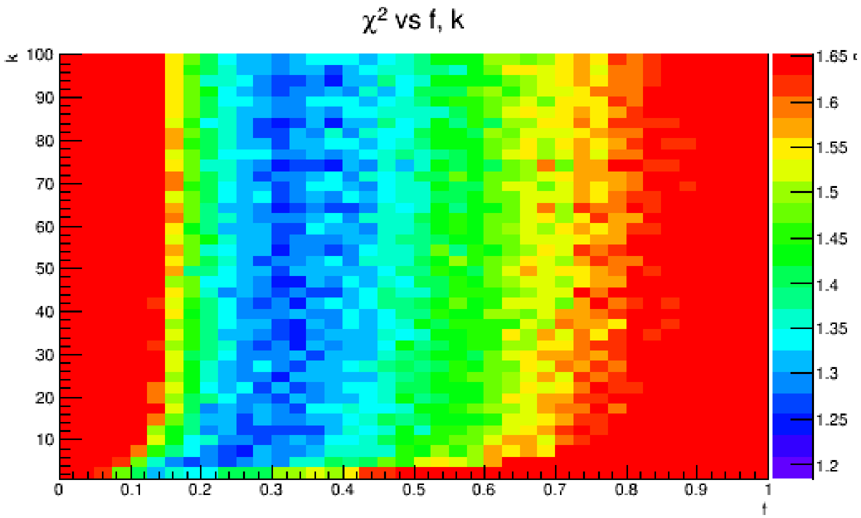


- Task:
  - ✓ run Glauber model with the same parameters as used by STAR (R, d, sigma_NN, beta2, beta4)
  - ✓ fit ‘STAR distribution’ using our standard Glauber fit machinery provided by MEPhI team (“STAR” option for N_a)
  - ✓ look at distribution of Chi2/NDF vs. (k, f) → are they physical?
  - ✓ compare the best fit parameters with those from STAR

Chi2/NDF vs (f,k)

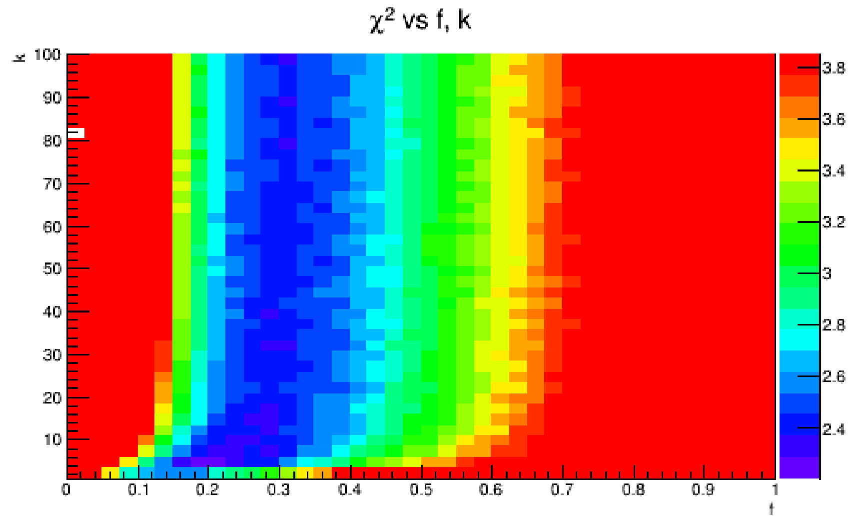
- Distributions of Chi2/NDF vs (f,k) parameters:

5 M Glauber events



f = 0.31 mu = 0.88 k = 54 chi2 = 1.18

20 M Glauber events

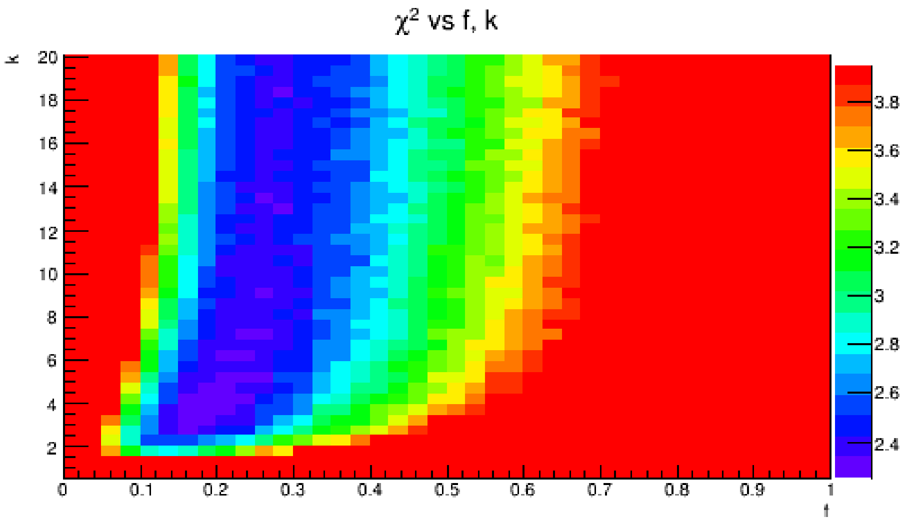


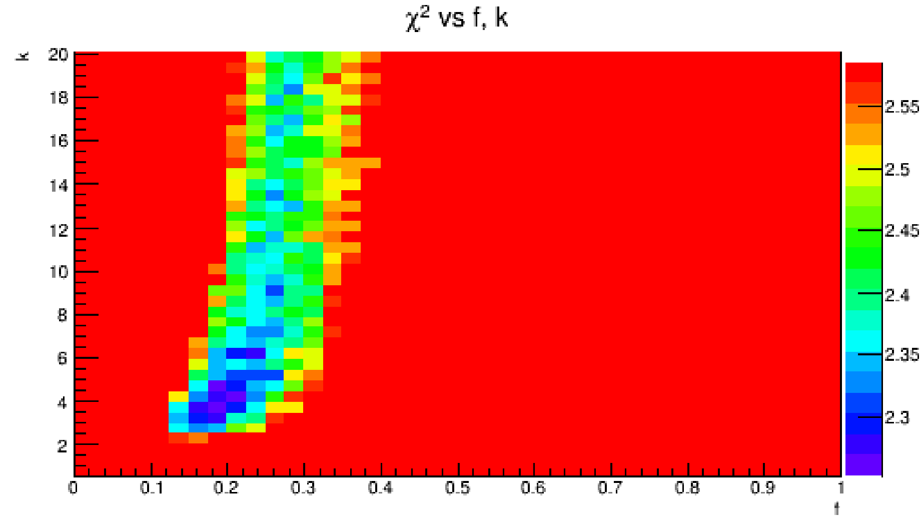
f = 0.19 mu = 1.12 k = 3 chi2 = 2.20

- Smaller values of f are preferred by data
- Fit results with 5 M and 20 M events are consistent (larger statistics → larger Chi2/NDF)
- Fits with 5 M Glauber events are prone to fluctuations
- Fit with 20 M Glauber events are more stable, distinct minimum is observed

A closer look at the region of minimum

- Distributions of Chi2/NDF vs (f,k) parameters (20 M Glauber events):





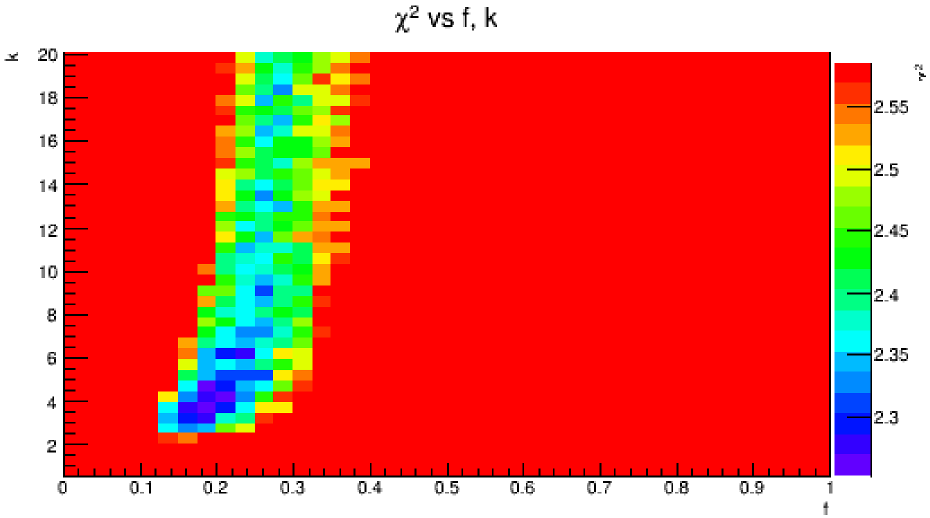
f = 0.19 mu = 1.12 k = 3.9 chi2 = 2.13

- A minimum is observed and confirmed with finer steps

Even larger statistics

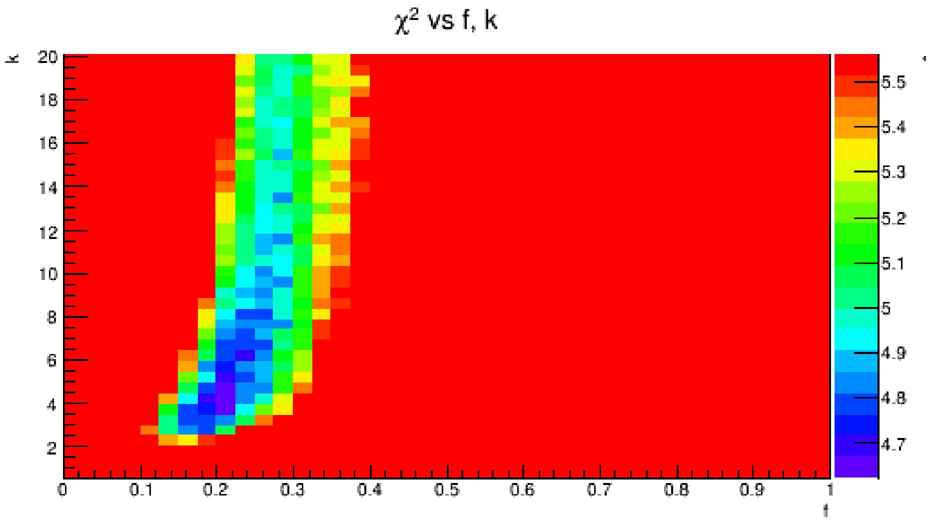
- Distributions of Chi2/NDF vs (f,k) parameters (20 M Glauber events):

20 M Glauber events



f = 0.19 mu = 1.12 k = 3.9 chi2 = 2.13

50 M Glauber events

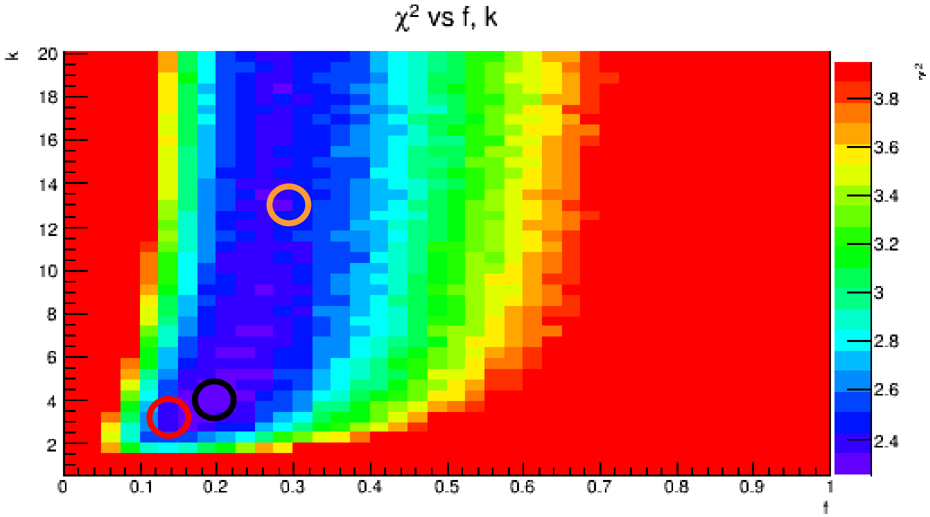


f = 0.21 mu = 1.07 k = 4.9 chi2 = 4.49

- Fluctuations become slightly smaller, but nothing new is observed
- STAR results: chi2/ndf = 1.2, Npp=1.23, k=9.33, x=0.156 → not identical, but close
- STAR ranges: Npp: [1.1,1.5] in 10 steps; k : [4,10] in 10 steps; x : [0.1,0.2] in 10 steps

How solid are the constraints from the fits

- Distributions of Chi2/NDF vs (f,k) parameters (20 M Glauber events): from slide 5

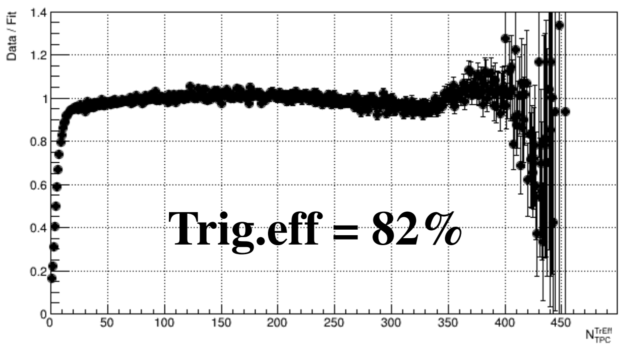


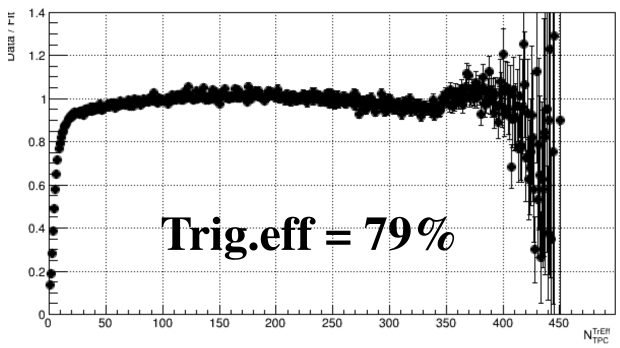
Default best fit: f = 0.19 mu = 1.12 k = 3.9 chi2 = 2.13
Alternative 1: f = 0.30 mu = 0.90 k = 13.7 chi2 = 2.28
Alternative 2: f = 0.13 mu = 1.29 k = 2.9 chi2 = 2.23

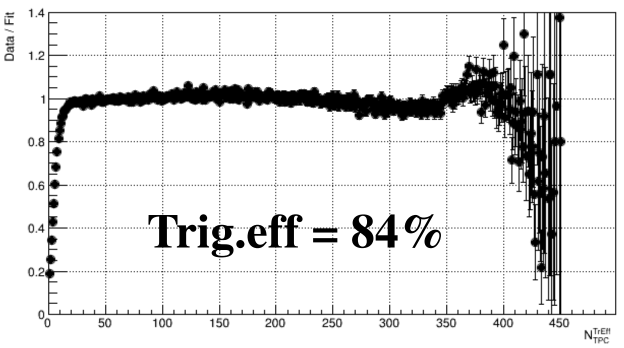
These variations correspond to +1.8σ and +1.2σ variation of the best fit Chi2/NDF

- Data-to-fit ratios and trigger efficiencies:

Best fit | **Alternative 1** | **Alternative 2**







- Trigger efficiencies are constrained within ~ 2-3 % → OK

Conclusions

- Fitting produces physically meaningful results with real data multiplicity distributions
- Extracted best fit parameters are close, but not identical to those reported by STAR
- Observed minimum for (k,f) parameters is quite shallow (not as shallow as in simulations)
- Variation of (k,f) parameters within ~ 1.5σ of Chi2/NDF changes trigger efficiency by ~ 2%