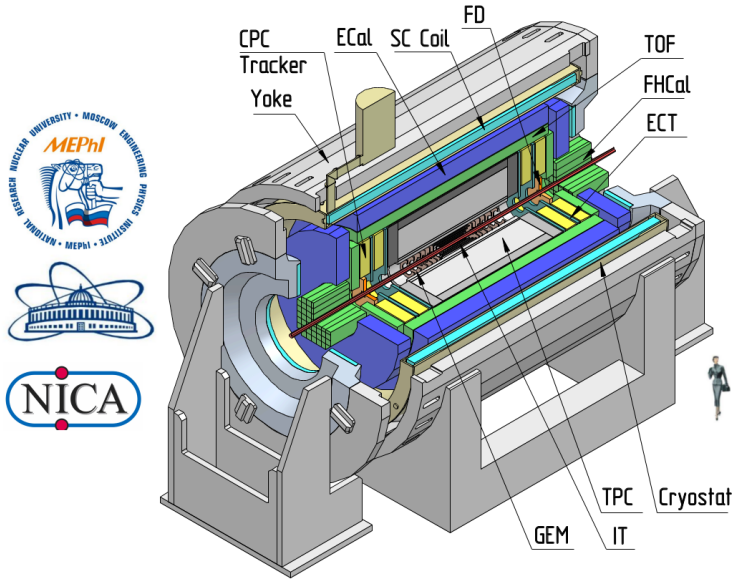


Data format for physics analysis (MiniDst) and package for femtoscopic studies

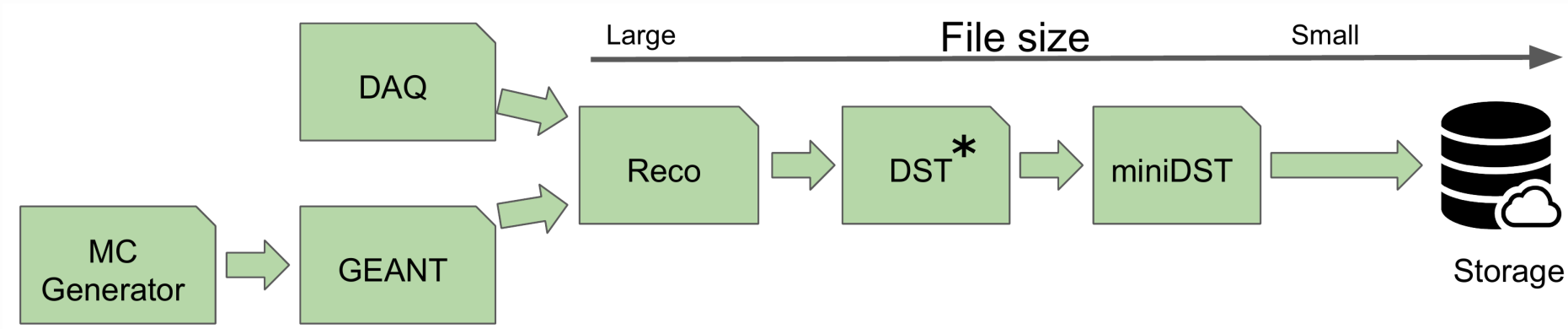
P. Batyuk[1], *G. Nigmatkulov*[2]

[1]Joint Institute for Nuclear Research, Dubna, Russia
[2]National Research Nuclear University MEPhI, Moscow, Russia

October 29, 2020

Data formats in relativistic heavy-ion collisions

Event processing scheme:



DST:

Event, trigger, hit, track, detector information:
Large size, hard to use for physics analysis

***The file size of the current MpdDst is 1.3 MB/event**

miniDST:

Only basic event, trigger, track, and detector information:
Small size, easy to use for physics analysis

miniDST: requirements

It contains info requested by physicists and only vital variables for all analyses.

MUST be implemented in MpdRoot

- Done

Easy / fast to produce

- During the whole production chain (DAQ → DST → miniDst)
- Or reproduced for the larger format (DST → miniDst)

Possibility to work independently from the MPD software

- Must work on any computer farm or laptop with ROOT 6
- Works on various OS's (Linux, MacOS, Windows)
- Only simple (native) data types (int, float, ...)
- Should be easily compiled with Makefile or CMake

MpdMiniDst: what is currently in?

List of currently implemented classes (MpdMiniClassName):

- Event - information about general event properties
- Track - reconstructed (both global and primary) track parameters
- TrackCovMatrix - covariance matrix of the global track
- BTofHit - barrel Time-Of-Flight hit information
- BTofPidTraits - information about TOF-matched track
- BECalCluster - information about ECal clusters
- FHCalHit - information about FHCal hit
- McEvent - Monte Carlo event properties
- McTrack - Monte Carlo track information

- DstReader - does all routing jobs and allows one to read miniDst
- Helix and PhysicalHelix - to work with helix trajectories

- Makefile - to compile in a standalone mode minidst_env.sh - shell script to setup the environment for standalone mode
- PhysicalConstants.h and SystemOfUnits.h - helper headers

MpdMiniDst: what is currently in? (examples of main classes)

MpdMiniEvent

```
/// Run number (or runId)
Int_t fRunId;
/// Event ID
Int_t fEventId;
/// Fill number
UShort_t fFillId;
/// Magnetic field strength (kG)
Float_t fBField;
/// Primary vertex position X
Float_t fPrimaryVertexX;
/// Primary vertex position Y
Float_t fPrimaryVertexY;
/// Primary vertex position Z
Float_t fPrimaryVertexZ;
/// Number of ECal-matched tracks
UShort_t fNBECalMatch;
/// Number of TOF-matched tracks
UShort_t fNBTOFMatch;
/// List of triggers that were fired in the current event
std::vector<unsigned int> fTriggerIds;
```

MpdMiniTrack

```
/// Px momentum (GeV/c) of the primary track
Float_t fPMomentumX;
/// Py momentum (GeV/c) of the primary track
Float_t fPMomentumY;
/// Pz momentum (GeV/c) of the primary track
Float_t fPMomentumZ;
/// Px component of the momentum (GeV/c)
Float_t fGMomentumX;
/// Py component of the momentum (GeV/c)
Float_t fGMomentumY;
/// Pz component of the momentum (GeV/c)
Float_t fGMomentumZ;
/// Index of the barrel ECal cluster in the event (-1 if not matched)
Short_t fBECalClusterIndex;
/// Index of the BTOF pidTratis in the event (-1 if not matched)
Short_t fBTofPidTraitsIndex;
/// Index of miniMcTrack that corresponds to miniTrack (aka IdTruth). (-1 if no match)
Short_t fMcTrackId;
```

MpdMiniBTofHit

```
/// Detector ID
Int_t fDetectorID;
/// Hit position projected on X plane
Short_t fBTofHitPosX;
/// Hit position projected on Y plane
Short_t fBTofHitPosY;
/// Hit position projected on Z plane
Short_t fBTofHitPosZ;
/// Time since the event start [ns]
Float_t fTime;
```

MpdMiniBTofPidTraits

```
/// Index to the associated track in the event (-1 if no matching)
Short_t fTrackIndex;
/// Index to the associated hit in the event (-1 if no matching)
Short_t fHitIndex;
/// Beta
UShort_t fBTofBeta;
```

MpdMiniDst: already done and how to ...

- Event information: primary vertex, multiplicity ...
- Information on primary tracks (tracks refitted to the primary vertex)
- Possibility to look at MC generator level information
- Access to PID information
- Barrel ECal with track-matching info
- FHCal - possibility to reconstruct event plane and determine centrality
- TOF-matching information

Available datasets for testing:

UrQMD, BiBi @ 9GeV, GEANT3 (it includes TPC + TOF + FHCAL, no correct track cov. matrix yet)
`/eos/nica/mpd/sim/data/MiniDst/dst-BiBi-09GeV-mp07-20-pwg3-250ev/BiBi/09.0GeV-0-14fm/UrQMD/BiBi-09GeV-mp07-20-pwg3-250ev-1,2,3,4`

```
// 1. Instantiate reader
MpdMiniDstReader* miniDstReader
= new MpdMiniDstReader(inFileName);
miniDstReader->Init();

// 2. One can specify branches to be read
miniDstReader->SetStatus("Track*", 1);
miniDstReader->SetStatus("TrackCovMatrix*", 0);

// 3. Loop over events
for (Long64_t i = 0; i < events2read; i++) {
  // Read next event
  miniDstReader->readMiniEvent(i);

  // Retrieve current miniDst
  MpdMiniDst *dst = miniDstReader->miniDst();

  // Reco track loop
  for (Int_t j = 0; j < nGTracks; j++) {
      // Retrieve j-th mini track
      MpdMiniTrack *miniTrack = dst->track(j);
  }

  // Monte Carlo track loop
  for (Int_t j = 0; j < dst->nMCTracks(); j++) {
      // Retrieve j-th MC track
      MpdMiniMcTrack *mcTrack = dst->mcTrack(j);
  }
}
```

See $VMCWORKDIR/macro/physical_analysis/miniDst/miniDstProcExample.C

... And to be done

- **Prepare a correct implementation of MpdTrackCovMatrix ...**
- **Develop a system of QA tools to be sure whether the filled miniDST looks correctly ...**

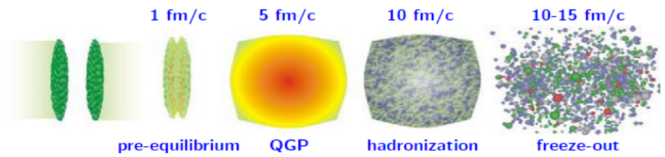
If any question occurred ...

pavel.batyuk@jinr.ru (JINR)
ganigmatkulov@mephi.ru,nigmatkulov@gmail.com (NRNU MEPhI)

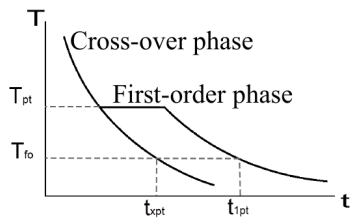
Femtoscopic measurements in MPD

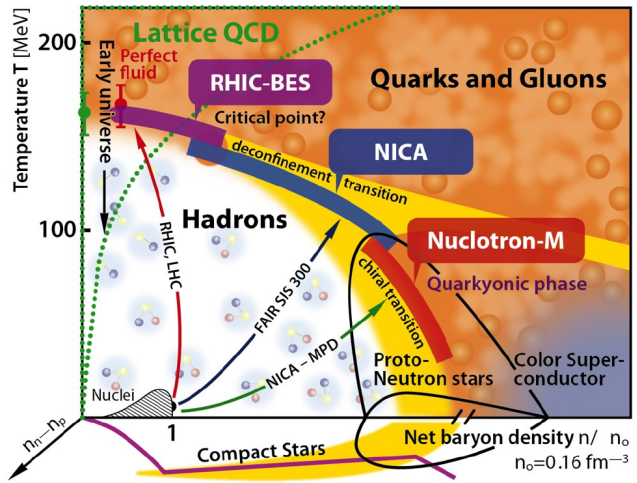
Femtoscopy:

- Allows one to study the dynamic properties of the medium produced in heavy-ion collisions looking at the two-particle momentum correlations



- Crossover transition to QGP occurs at RHIC & LHC
- 1st order phase transition to QGP occurs at lower energies (?)





- Provides information about spatial and temporal properties of the particle emission process, final-state interaction between particles, allows one to put constraints on the Equation of State and various models, sensitive observable for the first-order phase transition

See PWG3 Summary II on October 30, 15.00 (K. Mikhaylov)

MpdFemtoMaker: what is it?

MpdFemtoMaker is a package to perform femtoscopic analysis

Features:

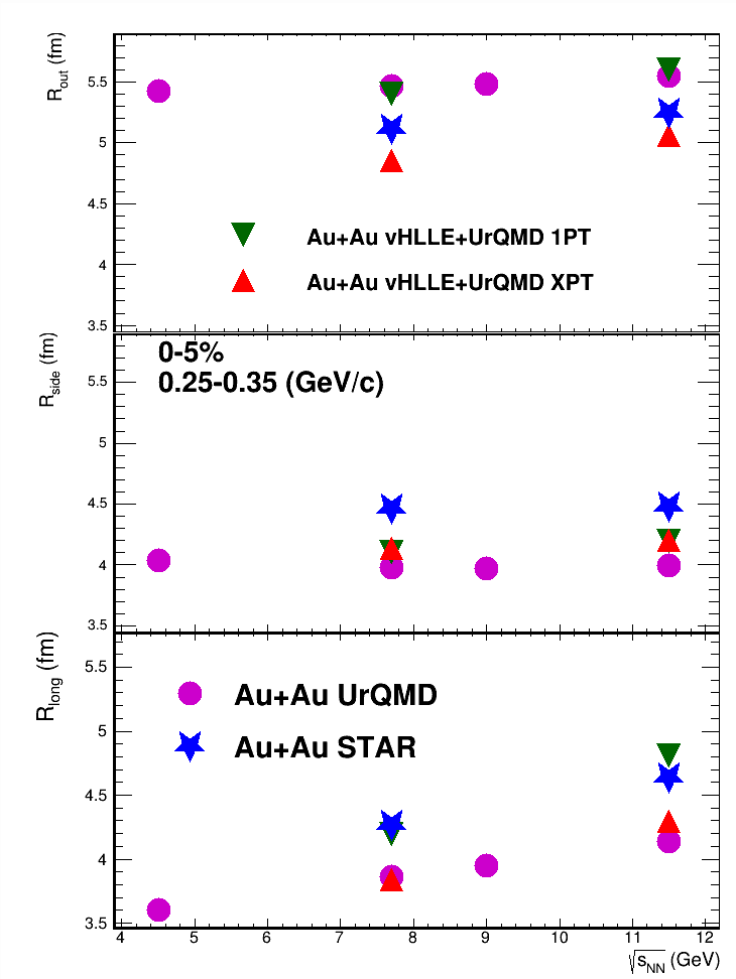
- Inherited from STAR (StHbtMaker) and ALICE (AliFemto) and incorporates best practices
- Works with ROOT 5 and 6 (may work standalone) and already implemented in the MpdRoot framework
- Physicist can use either existing OR add personal reader to his favorite input data format ...
- Allows one to apply the whole variety of the event/track/pair/... cuts, fill and plot histograms, use event-mixing technique ...

Supported input data formats:

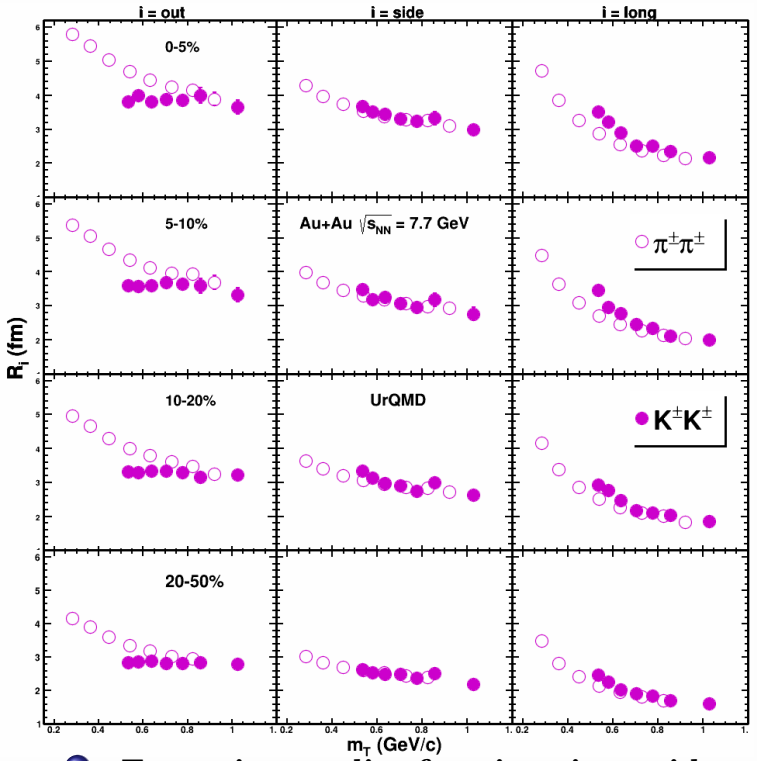
- mcDst is used for femtoscopy studies with Monte Carlo generators (UrQMD, vHLLE ...)
- miniDst is used for analysis with reconstructed data (full MPD simulation)

MpdFemtoMaker: studies with mcDst

Pion femtoscopy in the NICA energy range (NER)



Kaon femtoscopy with UrQMD (EoS - pure hadron gas)



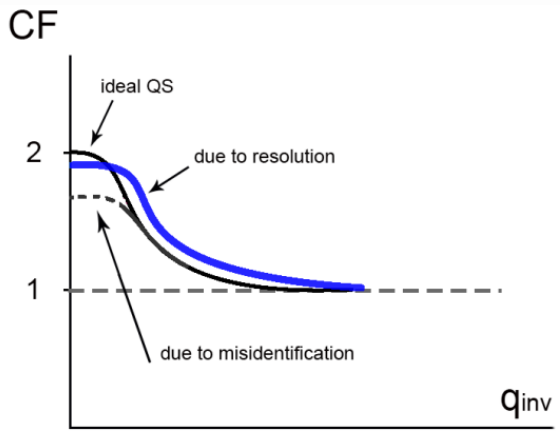
- Extensive studies for pions in a wide centrality range with two Monte Carlo generators
- Kaons calculations for four energies with UrQMD (4.5, 7.7, 9 and 11.5 GeV) and two energies with vHLLE (7.7 and 11.5 GeV)

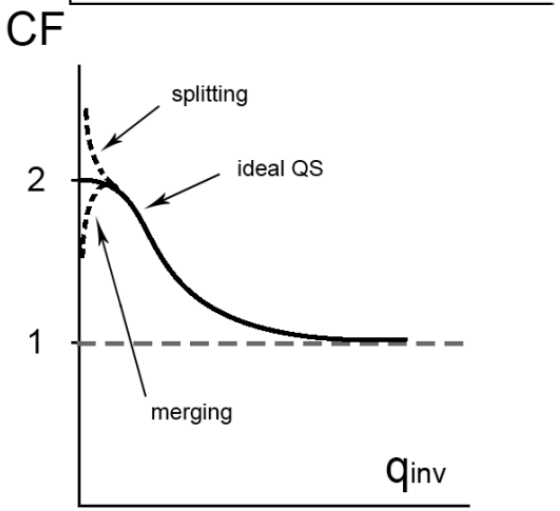
Kaon femtoscopy is in progress (different colliding systems, centralities, MC generators ...) to do energy scan in NER.

MpdFemtoMaker: studies with MpdMiniDst

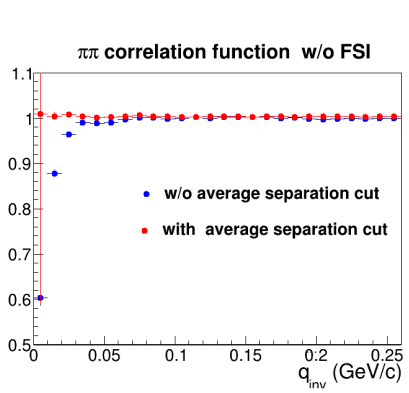
The methods to suppress two-track inefficiencies have been developed

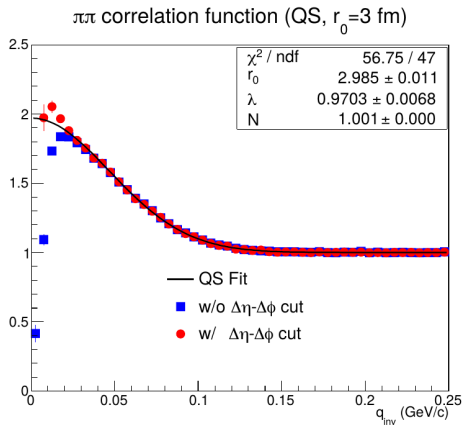
Two-track effects:





Average separation, $\Delta\eta\Delta\phi^*$





Correlation function can be reconstructed correctly applying two-track cuts

Detector effects:

- Study the effects of ionization cluster sharing in TPC
- Single- and pair-track purity
- Track-merging and track-splitting

Conclusion:

MpdMiniDst

- Satisfies most MPD physics needs
- Implemented in MpdRoot (and standalone)
- Contains most of the detector subsystems
- Easy to use with the given example on how to
- Already used in several analyses

MpdFemtoMaker

- Implemented in MpdRoot (and standalone)
- Applied to study identical pion and kaon correlations with Monte Carlo generators
- Allowed us to study the influence of two-track effects on correlation function and develop two-track selection criteria to suppress these parasitic effects

Thank you for your attention!