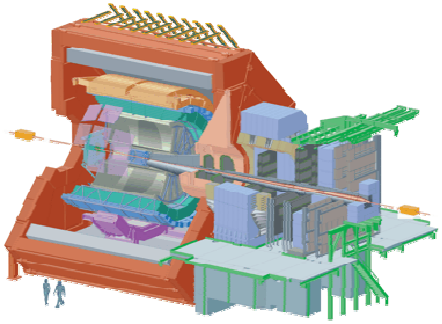
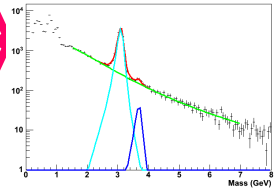
# PWG3-MUON: analysis status

**Roberta Arnaldi**
**INFN Torino (Italy)**



**Raw Data** — **ESD** — **AOD** — **Muon/Dimuon AOD** ⟷ **Event Mixing**

**Analysis**



**MC AOD** — **Correction Framework**

# AOD production

Due to the small fraction of events containing muons in the Dimuon Spectrometer, we foresee the creation of specific AODs:

**Muon/Dimuon AODs**

Advantage: their size will be smaller than the one of the standard AODs

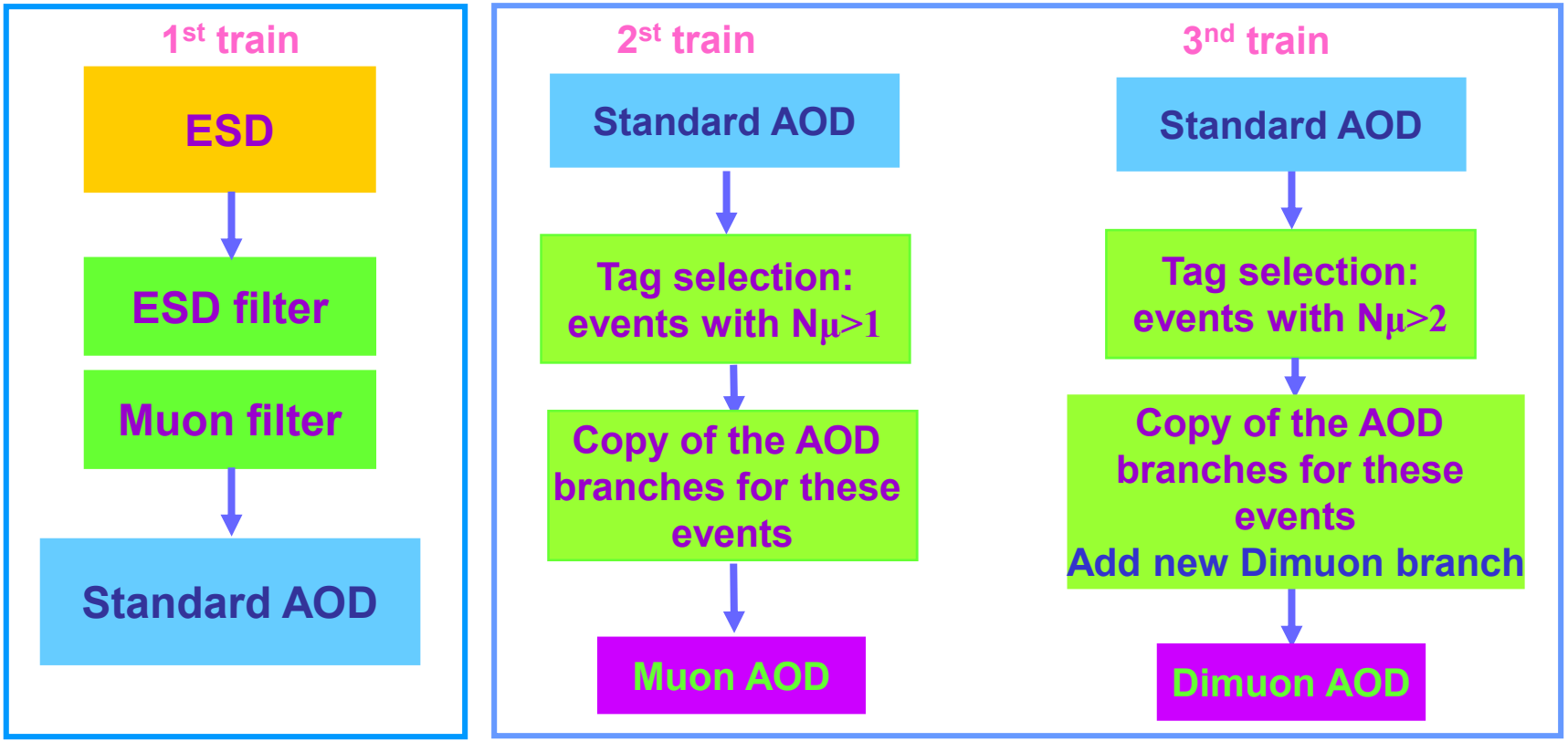
→ possible to store them locally on a PC
→ possible to merge several AODs together

Selection of events containing at least 1 (2) muons in the dimuon spectrometer is based on the **Event Tags**

Example: AliEventTagCuts::SetNFWMuonRange(nMin,nMax)

Muon/Dimuon AOD should be produced running a different analysis train with respect to the one producing the standard AOD, since with the tag selection only muon events will be processed

# Analysis train for AOD production




- If there are muons in the dimuon spectrometer
  - → the full AOD event is replicated
- If at least two muons exist
  - → a dimuon object is created and stored in an additional AOD branch
  - → size of the dimuon branch: ~ 0.02 KB/event negligible with respect to standard AOD event size (~3 KB/event)

# Analysis macros for AOD production

Analysis tasks for the Muon/Dimuon AOD production were already committed in PWG3/muon:

`AnalysisTrainMuonLocal.C`
`AliAnalysisTaskESDMuonFilter.cxx (.h)`
`AnalysisTrainFromStandardToMuonAODLocal.C`
`AliAnalysisTaskFromStandardToMuonAOD.cxx (.h)`

Code has now been updated to be compliant with the new analysis train configuration:

→ To produce the standard AOD:

`AnalysisTrainFromESDToAOD.C` → simplified version of the official analysis train macro

`+ AddTaskESDfilter.C` → to filter barrel/muon tracks

`+ AddTaskTagCreation.C` → to create AOD tags

→ To produce the muon AOD:

`AnalysisTrainFromAODToMuonAOD.C` → selection of muon events

`+ AddTaskFromStandardToMuonAOD.C` → replica of branches for events containing muons

# Test on Cosmic Run March 2009

- The analysis chain has been tested on the GRID on the Cosmic Run data (March 09)
  - 1st Step: ESD → Standard AOD+AOD tag files
  - 2nd Step: Standard AOD+AOD tag files → Muon AOD

- 2nd step has been performed locally (for the moment)
- code to be optimize in order to foresee merging of the files (at the AOD or Muon AOD level)

# Event mixing for the Muon Spectrometer

To perform dimuon studies, the combinatorial background has to be subtracted

This is done using the event mixing technique

→ developments based on the ALICE **Event Mixing Framework**

As an example, the code consists of:

- User macro: `AnalysisTrainCreateMixedDimuons.C`
- Analysis task: `AliAnalysisTaskCreateMixedDimuons.(h,cxx)`
- Pool manager: `AliEventPoolMuon.(h,cxx)`

Since muon should be mixed only if they belong to events with similar characteristics, pools are created according to

- z of the vertex
- charged particle multiplicity
- forward muons multiplicity

Pool definition based on tags

# Event mixing for the Muon Spectrometer (2)

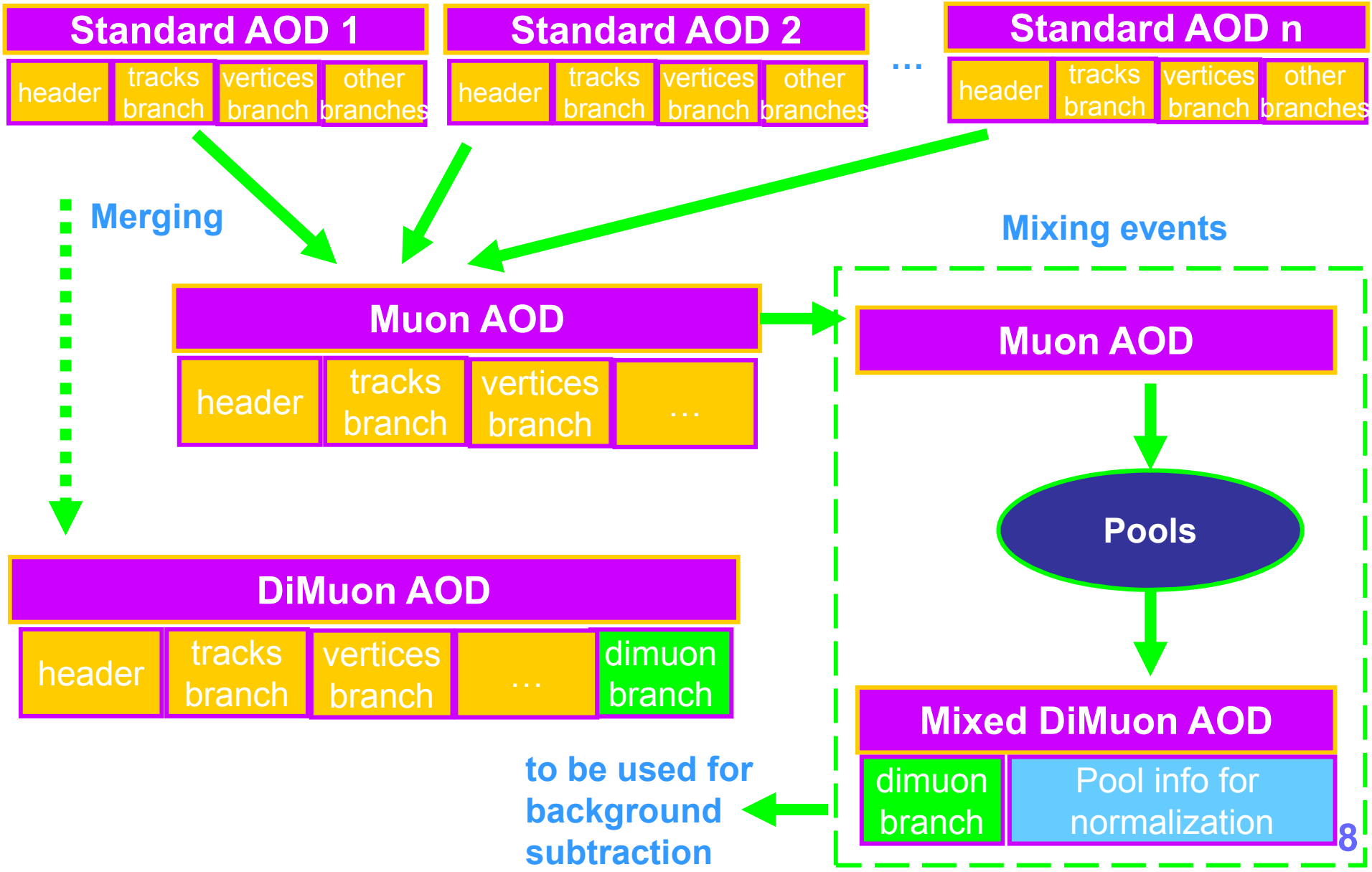
- The Standard AOD or Muon AOD (since it contains single muon tracks) can be used as input for the mixing, together with the tag files

- Output of the mixing framework can be a standard AOD, a muon AOD or even a Mixed Dimuon-AOD directly containing dimuon infos

- Output should have the same format as the real data

  → real and mixed events will be treated in the same way (same macros)

# Schema of the analysis train for MUON analysis



# MC information in the AOD

For simulation studies, information on generated and reconstructed events should be accessible also from AOD.

**MC truth information** is stored as an **additional branch in the AOD**, together with the reconstructed data
(based on K. Klein-Bosing implementation)

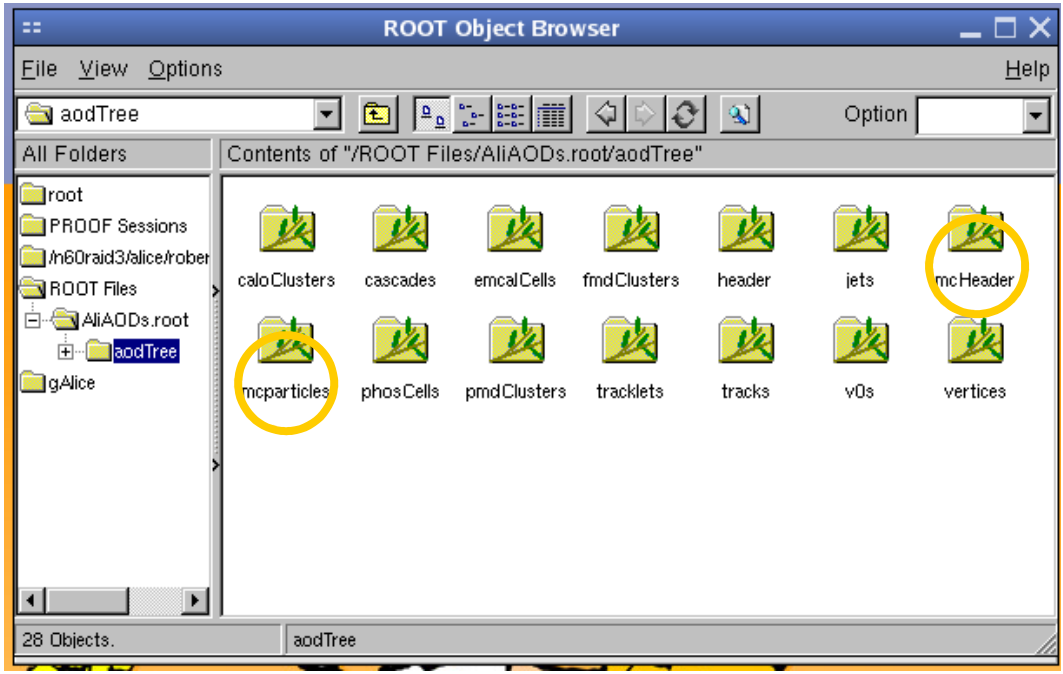
- MC labels have been defined for muon tracks
  → allow to keep the correspondence between MC and reconstructed tracks
- MC info added in
  **PWG3/muon/AliAnalysisTaskESDMuonFilter.cxx (.h)**

```
AliMCEventHandler *mcH = 0;
if(MCEvent()){
   pStack = MCEvent()->Stack();
   mcH = (AliMCEventHandler*) ((AliAnalysisManager::GetAnalysisManager())
   ->GetMCtruthEventHandler());
}
....
if(mcH)mcH->SelectParticle(esdMuonTrack->GetLabel());
```

# MC information in the AOD (2)

**AnalysisTrainMuon_MC.C**

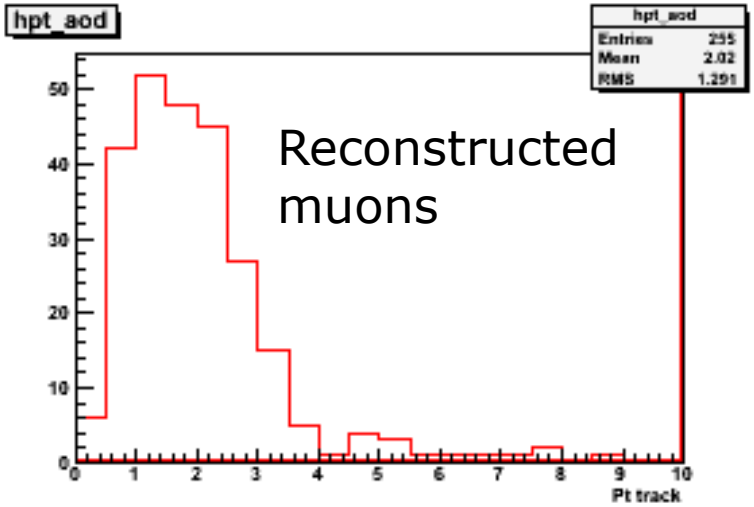
Macro to produce AOD+MC branch, starting from ESD+Kinematics

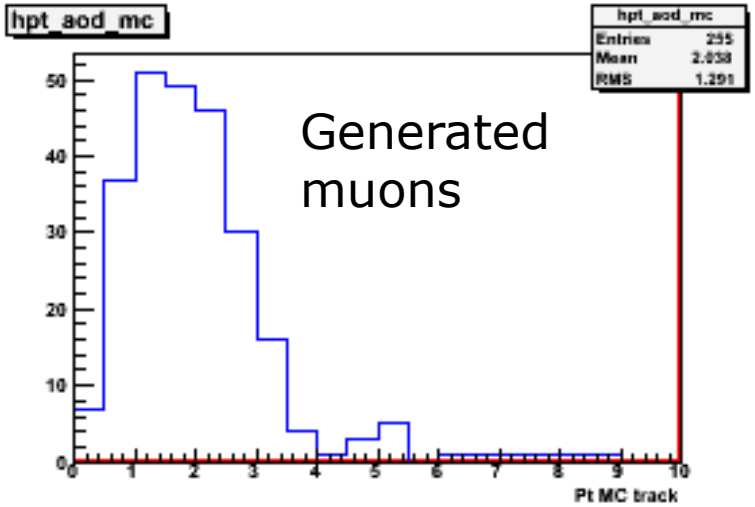


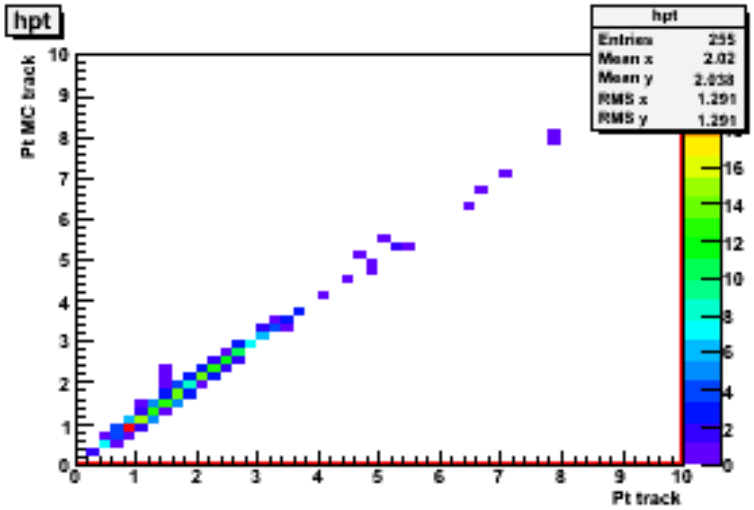

```
// ESD input handler
    AliESDInputHandler *esdHandler = new AliESDInputHandler();
// AOD output handler
    AliAODHandler* aodHandler   = new AliAODHandler();
// MC Truth handler
    AliMCEventHandler* mcHandler = new AliMCEventHandler();
    mgr->SetMCtruthEventHandler(mcHandler);
AliAnalysisTaskMCParticleFilter *kinefilter = new AliAnalysisTaskMCParticleFilter("Filter");
mgr->AddTask(kinefilter);
```

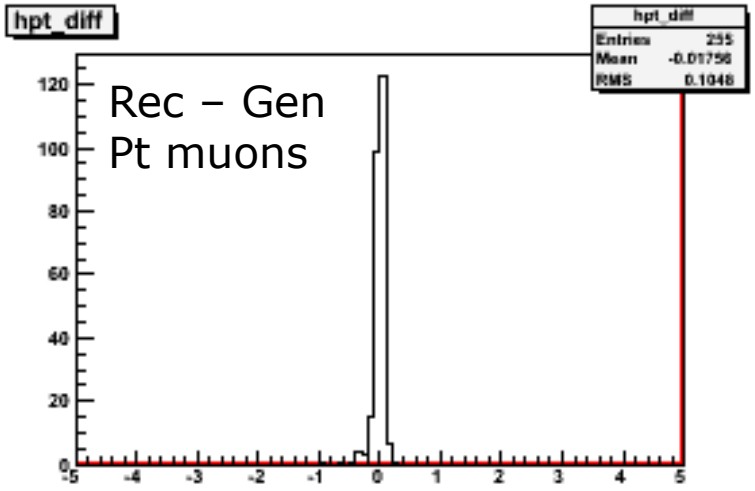
# MC information in the AOD (2)

AOD obtained from a J/ψ generation (`ReadAOD_MC.C`)



# Correction Framework

- To compute acceptances and efficiencies and to correct the reconstructed data
  - → the correction framework (CORRFW)
- The CORRFW has been adapted to muon/dimuon studies
  - → CORRFW for single muon analysis
  - → CORRFW for dimuon analysis
  - → CORRFW for continuum analysis
- Optimization of the number of variables needed for the different analysis is ongoing
  - e.g.: Quarkonium polarization analysis requires containers based on 4 variables:

$$y,\ p_T,\ \cos\theta,\ \phi$$

# Example of use of Correction Framework

Example of CORRFW code developed for single muon analysis

**AliCFMuonSingleTask1.C**

- set the number of steps for the container (2 steps: MC and ESD)
- set the number of variables
- set their range and bins and cuts

**AliCFMuonSingleTask1.cxx (.h)**

- fill the container (both steps)

**MuonSingleGrid.C**

- read the containers and compute the efficiency matrix
- data correction

Similar code has been prepared for other muon-related analysis

# Conclusions

- The tools available in the analysis framework have been adapted to the PWG3-muon requirements
  - → Analysis train
  - → Event Mixing framework
  - → Correction framework
- Next steps:
  - → commit in PWG3/muon all the available code
  - → foresee the inclusion of analysis wagons in a PWG3-muon analysis train

# Backup

# Files dimension

Some numbers on files dimension

**Typical ESD event (from PDC08/LHC08x pp@ 14TeV) :**

~ 16 KB/event

**Standard AOD :**

~ 3.1 KB/event

} compression factor ~ 5

**Events with $N_\mu \geq 1$ ~ 1%**

**Fraction of events with $N_\mu \geq 2$ / $N_\mu \geq 1$ ~ 1%**

**Size of the dimuon branch:**

~ 0.02 KB/event negligible with respect to standard AOD event size (~3 KB)

Because of the small size of the files, we should create the Muon/Dimuon AODs merging several files