

PMD task conversion from Run 2 to Run 3 framework

Abhi Modak^{1,*} and Sidharth Kumar Prasad²

¹INFN – Sezione di Trieste, Italy

²Bose Institute, Kolkata, India

*Email: abhi.modak@cern.ch

(Many thanks to David Chinellato and Anton Alkin)

ALICE-India Collaboration meeting
July 01, 2025

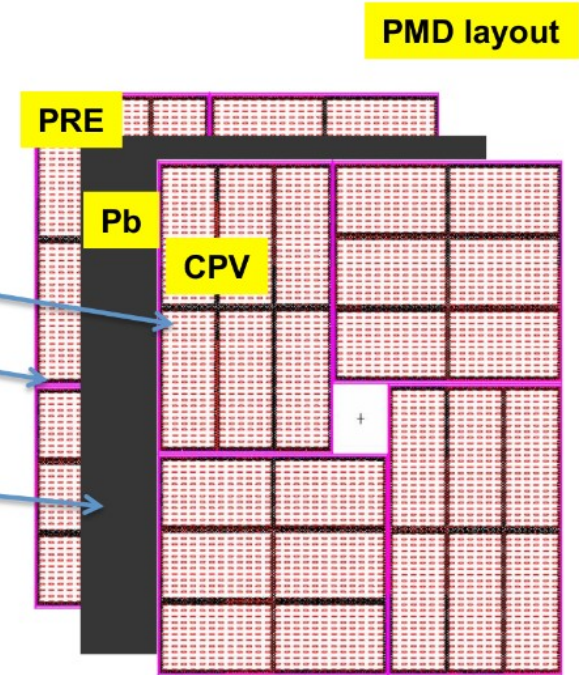
Photon Multiplicity Detector in ALICE

PMD measures photon multiplicity on an event by event basis

It consists of two planes:

CPV (Charged Particle Veto) and
PRE (Preshower)

A $3X_0$ Pb-converter
(sandwiched between the two plane
for photon conversion to showers of
electrons, positrons and photons)



PMD info in ESD

Physics analysis using PMD data relies on information stored in ESD and the information available in the OCDB files

PMD Reconstruction steps and information stored in ESD

(I) RAW DATA / Digits

|

(II) Noise Subtraction

|

(II) Calibration using files from OCDB

|

(III) Reconstruction (Clustering)

|

(IV) Cluster information stored in ESD using class AliESDPmdTrack

Clustering is only applied for pp
and p-Pb data

PMD info in ESD

```
Double_t fX = pmdtr->GetClusterX();  
// Cluster X position  
  
Double_t fY = pmdtr->GetClusterY();  
// Cluster Y position  
  
Double_t fZ = pmdtr->GetClusterZ();  
// Cluster Z position  
  
Double_t fCluADC = pmdtr->GetClusterADC();  
// Cluster Energy in ADC  
  
Double_t fCluPID = pmdtr->GetClusterPID();  
//[0.,1.,8] Cluster probability, 1: Photon, 0: Hadron  
  
UChar_t fDet = pmdtr->GetDetector();  
// Detector, 0:PRE, 1:CPV  
  
UChar_t fNcell = pmdtr->GetClusterCells();  
// Cluster cells  
  
Int_t fSmn = pmdtr->GetSmn();  
// Serial module number
```

```
Int_t fTrackNo = pmdtr->GetClusterTrackNo();  
// Track number assigned to the clus from simulation  
  
Int_t fTrackPid = pmdtr->GetClusterTrackPid();  
//Track pid assigned to clus from simulation  
  
Double_t fSigX = pmdtr->GetClusterSigmaX();  
// Cluster x-width  
  
Double_t fSigY = pmdtr->GetClusterSigmaY();  
// Cluster y-width  
  
UShort_t fCIMatching = pmdtr->GetClusterMatching();  
//Cluster of PRE matching with CPV
```

Applicable for pp and p-Pb data
analysis

[Presentation of S. Prasad](#)

PMD info in ESD

```
Int_t det = pmdtr->GetDetector();  
//detector; 0=PRE; 1=CPV
```

```
Int_t smn = pmdtr->GetSmn();  
//module number; 0 – 23 for CPV  
and 0 – 23 for PRE
```

```
Float_t clsX = pmdtr->GetClusterX();  
//cell row; 0 to 47
```

```
Float_t clsY = pmdtr->GetClusterY();  
//cell column; 0 to 95
```

```
Float_t clsZ = pmdtr->GetClusterZ();  
//cell z-position
```

```
UChar_t ncell =  
pmdtr->GetClusterCells();  
//no. of cells in cluster (in this case it  
is 1)
```

```
Float_t adc =  
pmdtr->GetClusterADC();  
//cell ADC
```

```
Float_t sigX =  
pmdtr->GetClusterSigmaX();  
//Track number
```

```
Float_t sigY =  
pmdtr->GetClusterSigmaY();  
//Track PID
```

[Presentation of S. Prasad](#)

Applicable for Pb-Pb
data analysis

Conversion into Run 3 framework

Converter task

<https://github.com/alisw/AlIPhysics/blob/master/RUN3/AlIAnalysisTaskAO2Dconverter.cxx>

The required information has been stored in the AO2D.root file per BC table

```
TTree* tPmdInfo = CreateTree(kPMD);
if (fTreeStatus[kPMD]) {
    // PMD information
    tPmdInfo->Branch("fIndexBCs", &pmdInfo.fIndexBCs, "fIndexBCs/I");
    tPmdInfo->Branch("fX", &pmdInfo.fX, "fX/F");
    tPmdInfo->Branch("fY", &pmdInfo.fY, "fY/F");
    tPmdInfo->Branch("fZ", &pmdInfo.fZ, "fZ/F");
    tPmdInfo->Branch("fCluADC", &pmdInfo.fCluADC, "fCluADC/F");
    tPmdInfo->Branch("fCluPID", &pmdInfo.fCluPID, "fCluPID/F");
    tPmdInfo->Branch("fDet", &pmdInfo.fDet, "fDet/b");
    tPmdInfo->Branch("fNcell", &pmdInfo.fNcell, "fNcell/b");
    tPmdInfo->Branch("fSmn", &pmdInfo.fSmn, "fSmn/I");
    tPmdInfo->Branch("fTrackNo", &pmdInfo.fTrackNo, "fTrackNo/I");
    tPmdInfo->Branch("fTrackPid", &pmdInfo.fTrackPid, "fTrackPid/I");
    tPmdInfo->Branch("fSigX", &pmdInfo.fSigX, "fSigX/F");
    tPmdInfo->Branch("fSigY", &pmdInfo.fSigY, "fSigY/F");
    tPmdInfo->Branch("fCIMatching", &pmdInfo.fCIMatching, "fCIMatching/I");
    tPmdInfo->SetBasketSize("", fBasketSizeTracks);
}
```

Conversion into Run 3 framework

Analysis data model

<https://github.com/AliceO2Group/AliceO2/blob/dev/Framework/Core/include/Framework/AnalysisDataModel.h>

```
namespace pmd
{
DECLARE_SOA_INDEX_COLUMN(BC, bc);           //!< BC index
DECLARE_SOA_COLUMN(X, pmdclsx, float);     //!< cluster x position
DECLARE_SOA_COLUMN(Y, pmdclsy, float);     //!< cluster y position
DECLARE_SOA_COLUMN(Z, pmdclsz, float);     //!< cluster z position0
DECLARE_SOA_COLUMN(CluADC, pmdclsadc, float);   //!< cluster energy in ADC
DECLARE_SOA_COLUMN(CluPID, pmdclspid, float);   //!< cluster probability, 1: photon, 0:hadron
DECLARE_SOA_COLUMN(Det, pmddet, uint8_t);     //!< Detector, 0:PRE, 1:CPV
DECLARE_SOA_COLUMN(Ncell, pmdncell, uint8_t);  //!< cluster cells
DECLARE_SOA_COLUMN(Smn, pmdmodule, int32_t);   //!< module number
DECLARE_SOA_COLUMN(TrackNo, pmdtrackno, int32_t);   //!< Track number assigned to clus from simulation
DECLARE_SOA_COLUMN(TrackPid, pmdtrackpid, int32_t);   //!< Track PID assigned to clus from simulation
DECLARE_SOA_COLUMN(SigX, pmdsigx, float);     //!< Cluster x-width
DECLARE_SOA_COLUMN(SigY, pmdsigy, float);     //!< Cluster y-width
DECLARE_SOA_COLUMN(CIMatching, pmdclmatching, int32_t);   //!< Cluster of PRE matching with CPV
} // namespace pmd

DECLARE_SOA_TABLE(Pmds, "AOD", "PMD", //!< Photon information from PMD detector
    o2::soa::Index<>, run2::pmd::BCId, run2::pmd::X, run2::pmd::Y,
    run2::pmd::Z, run2::pmd::CluADC, run2::pmd::CluPID, run2::pmd::Det,
    run2::pmd::Ncell, run2::pmd::Smn, run2::pmd::TrackNo, run2::pmd::TrackPid,
    run2::pmd::SigX, run2::pmd::SigY, run2::pmd::CIMatching);
```

PMD table information is defined inside
AnalysisDataModel

Conversion into Run 3 framework

Converter task

<https://github.com/AliceO2Group/AliceO2/blob/dev/Framework/Core/include/Framework/AnalysisDataModel.h>

```
namespace pmd
{
DECLARE_SOA_INDEX_COLUMN(BC, bc);           //!< BC index
DECLARE_SOA_COLUMN(X, pmdclsx, float);     //!< cluster x position
DECLARE_SOA_COLUMN(Y, pmdclsy, float);     //!< cluster y position
DECLARE_SOA_COLUMN(Z, pmdclsz, float);     //!< cluster z position0
DECLARE_SOA_COLUMN(CluADC, pmdclsadc, float); //!< cluster energy in ADC
DECLARE_SOA_COLUMN(CluPID, pmdclspid, float); //!< cluster probability, 1: photon, 0:hadron
DECLARE_SOA_COLUMN(Det, pmddet, uint8_t);  //!< Detector, 0:PRE, 1:CPV
DECLARE_SOA_COLUMN(Ncell, pmdncell, uint8_t); //!< cluster cells
DECLARE_SOA_COLUMN(Smn, pmdmodule, int32_t); //!< module number
DECLARE_SOA_COLUMN(TrackNo, pmdtrackno, int32_t); //!< Track number assigned to clus from simulation
DECLARE_SOA_COLUMN(TrackPid, pmdtrackpid, int32_t); //!< Track PID assigned to clus from simulation
DECLARE_SOA_COLUMN(SigX, pmdsigx, float);  //!< Cluster x-width
DECLARE_SOA_COLUMN(SigY, pmdsigy, float);  //!< Cluster y-width
DECLARE_SOA_COLUMN(CIMatching, pmdclmatching, int32_t); //!< Cluster of PRE matching with CPV
} // namespace pmd
```

PMD table information is defined inside
AnalysisDataModel

Name	Getter	Type
o2::run2::pmd::CluADC	pmdclsadc	float
o2::run2::pmd::X	pmdclsx	float

Analysis task in O² framework

PMD table maker

<https://github.com/AliceO2Group/O2Physics/blob/master/Common/DataModel/PmdTable.h>

```
#ifndef COMMON_DATAMODEL_PMDTABLE_H_  
#define COMMON_DATAMODEL_PMDTABLE_H_
```

```
#include "Framework/AnalysisDataModel.h"
```

```
namespace o2::aod
```

```
{  
  namespace pmdtrack
```

```
{  
  DECLARE_SOA_INDEX_COLUMN(Collision, collision);  
  DECLARE_SOA_ARRAY_INDEX_COLUMN(Collision, collisions);  
  DECLARE_SOA_INDEX_COLUMN(BC, bc);  
  DECLARE_SOA_SLICE_INDEX_COLUMN(Pmd, pmd);  
} // namespace pmdtrack
```

```
DECLARE_SOA_INDEX_TABLE_USER(PMDTracksIndex, BCs, "PMDTRKIDX", pmdtrack::CollisionId, pmdtrack::BCId, pmdtrack::PmdIdSlice);  
} // namespace o2::aod
```

Joining of PMD table information with
the collision table

Analysis task in O² framework

PMD Analysis task

<https://github.com/AliceO2Group/O2Physics/blob/master/Common/LegacyDataQA/pmdQa.cxx>

Run2 converted data

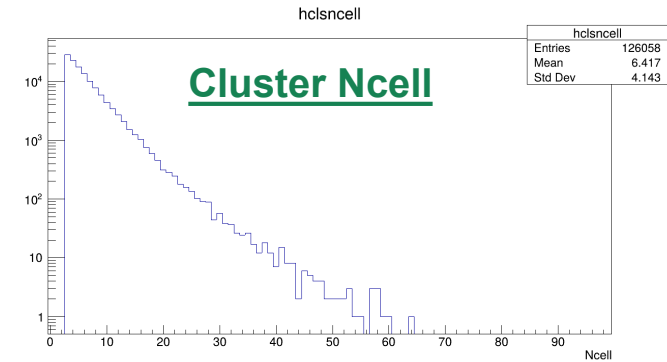
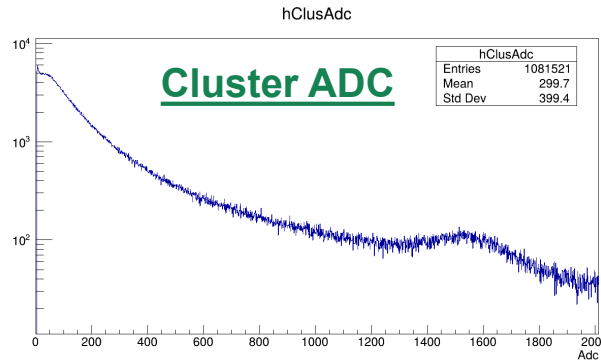
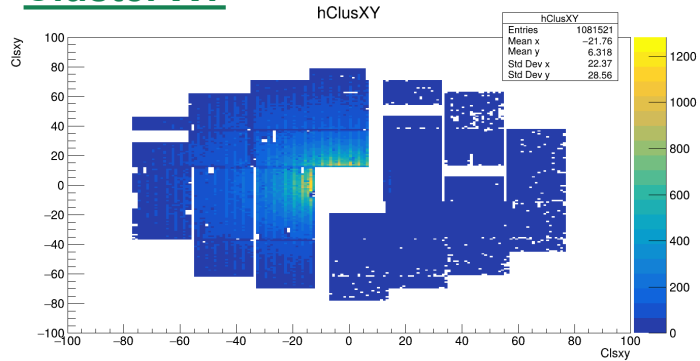
<https://alimonitor.cern.ch/hyperloop/view-dataset/2094> (only for pp)

Hyperloop run

<https://alimonitor.cern.ch/hyperloop/train-run/409058/general>

Some of the basic plots

Cluster XY



Summary

- The PMD information has been stored inside AO2D.root file
- Basic tasks have been developed to read the PMD info from AO2D.root file
- Codes are in place inside O2 framework and ready to run in hyperloop
- One has to make further QA study for pursuing the PMD analysis

Thanks for your kind attention