PWG4
PartCorr, GammaConv, Pi0 Calibration code Status

Gustavo Conesa Balbastre
INFN-Frascati

PartCorr: Reminder

- Responsible Gustavo Conesa Balbastre
- Particle identification (photons, pi0, eta, electrons …) and correlation (with jets, hadrons …) package.
- Subdivided in 2 directories Base and Dep:
  - PartCorrBase: Manager classes are here and common pieces of code to be used for for any kind of analysis
  - PartCorrDep: All classes doing the different analysis
- Quite complete documentation can be found in the offline analysis pages:
  - http://aliceinfo.cern.ch/Offline/Activities/Analysis/PWGDocumentation/PWG4/PartCorr.html
  - Not up-to-date, changes listed in this presentation and in the last offline week presentation.

Partcorr: Issues from last report

- Provide AddTaskPartCorr.C macro to run in the train:
  - Available since 20th April (last version from 14th May).
  - It executes: Photon + Pi0 analysis analysis and Photon + Isolation (photon) + Photon-Jet Finder Correlation + Photon-Hadron correlation
    - Part of these analysis are good for the MB productions (LHC09xxx) but the others should be executed with Jet or EMCAL prodution. Where does the train run?
  - It is executed with the train twice, one for PHOS and other for EMCAL.
  - Can only be used in the train to read AODs produced by the filter or directly AOD events.
  - No problems reported with the execution in the train.
  - To add Neutral Meson (pi0, eta omega) and electron analysis (see next slides)
- Recreation of objects (tracks, clusters) only in memory.
  - I was filling arrays of new TClonesArrays with copies of selected ESDs or AODs or TParticles into AOD objects
  - Now I use TRefArrays instead of TClonesArrays
    - ESD Reader: To be used only if the official ESD filter is not executed before. I do my own filtering, creating AODs (tracks, clusters) and putting them in the output AOD. Then I do references to these objects which I use during the analysis.
    - AOD Reader: Fill the TRefArray with selected AODs
    - MC Reader: Same as ESD reader, transform TParticles into AODs, put them in output AOD.
- Use of official mixing frame in pi0 analysis and other analysis that require this.
  - Still open issue to be discussed with the analysis authors.
  - Next slide for possible solution.

PartCorr and mixing frame

- PartCorr executes all from the analysis task AliAnalysisTaskPartCorr
  - It derives from AliAnalysisTaskSE
- Right now 2 classes using their own mixing, AliAnaPi0 and AliAnaNeutralMeson:
  - They fill a buffer with TClonesArrays, each for different events, number to be set by the user.
  - Each TClonesArray contains AliAODPWG4ParticleCorrelation objects, photons.
- After brief discussion with Andreas I had a month ago, I propose to:
  - create new analysis task to be used when the mixing is needed AliAnalysisTaskPartCorrMix deriving from AliAnalysisTaskME
  - do analysis in 2 steps, not in the same train:
    - Step 1: Execute Single Event analysis (AliAnalysisTaskPartCorr) that will produce the AOD output with the particle (photon, pi0 (in case of omega analysis)) for all the events.
    - Step 2: Execute Mixed Event analysis (AliAnalysisTaskPartCorrMix)that will access from the buffer of events the previously generated particles (photons, pi0)
  - First step could go in the normal train and the second maybe single user analysis or another train that would run after the first one?
  - Need to discuss with the authors of the analysis if they like this approach.

PartCorr: Frame changes

- Selected list of tracks/clusters are now TRefArrays (slide 2).
- Terminate():
  - It was not working in case of use of the alien plugin or PROOF.
    - Histograms pointers are null.
  - Each analysis must refill the histogram pointers, from the output list.
  - Example in AliAnaPi0, AliAnaExample and AliAnaCalorimeterQA
- Histograms naming:
  - All histograms produced in the frame, executed by different analysis are kept in the same folder.
  - Different analysis types might have similar histogram name, and the ordering when browsing can be chaotic.
  - Even the same analysis but with different settings can be executed.
  - A string is added at the beginning of the histogram name.
    - In the configuration file, for each analysis do `AddToHistogramsName("AnaExample_");`
- Proposed change:
  - Christian suggested to create a new module in PWG4 that would contain AOD classes from the jet studies for background.
  - We could move there also the classes AliAODPWG4Particle and AliAODParticleCorrelation, so that if somebody want to use them and don't want to enter in the frame could do it.
  - Also AliAnaScale, class for normalization of output histograms could go there since is a quite useful, general utility.

PartCorr: Frame changes

- Change in AliAODPWG4ParticleCorrelation
  - There are several data member very analysis specific:

```
Bool_t fIsolated ;            //Particle is isolated or not
TString fLeadingDetector;// Detector where leading particle was measured.
TLorentzVector fLeading;  // Leading Particle 4-momentum vector
TLorentzVector fCorrJet;  // Jet  4-momentum vector
TLorentzVector fCorrBkg;  // Background  4-momentum vector
TRef  fRefJet;        // Reference to jet found with JETAN and correlated with particle
TRefArray *fRefTracks (Clusters); //For correlated hadrons or jet FF storage
TRefArray *fRefIsolationConeTracks (Clusters); //For particle isolation studies
TRefArray *fRefBackgroundTracks(Clusters); // For correlated hadron or jet FF backg.
```

  - I want to reduce the dependence on specific analysis.
  - I have replaced the TRefArrays by a TList *fListOfRefArrays
    - Each analysis will create and store there if needed the list of references.
    - This will allow to store in the same AOD, the correlated lists of particles of different analysis.
  - There are still a few more data members that depend on specific analysis that I would like to deal in a similar way if possible, specially the TLorentzVector and TRef data members. To be done in future somehow.
  - The current modification affects the correlation analysis:
    - AliAnaParticlePartonCorrelation
    - AliAnaParticleHadronCorrelation
    - AliAnaParticeJetLeadingConeCorrelation

PartCorr: Frame changes

- Change in AliAODPWG4ParticleCorrelation
  - How to use it in the analysis:
    - When selecting the correlated AODs, typically in MakeAnalysisFillAOD():
      - Before track/cluster loop:

```
TRefArray * refarray  = new TRefArray; BoolI_t first = kTRUE;
```

      - In Loop after selection:

```
if(first) {new (refarray) TRefArray(TProcessID::GetProcessWithUID(track));
          first=kFALSE;}
refarray->Add(track);
```

      - After loop:

```
refarray->SetName(GetAODRefArrayName());
if(refarray->GetEntriesFast() > 0) particle->AddRefArray(refarray);
```

    - When filling the final histograms, typically in MakeAnalysisFillHistograms(), to recover the array:

```
TRefArray * refarray = particle->GetRefArray(GetAODRefArrayName());
//check first if array exists
```

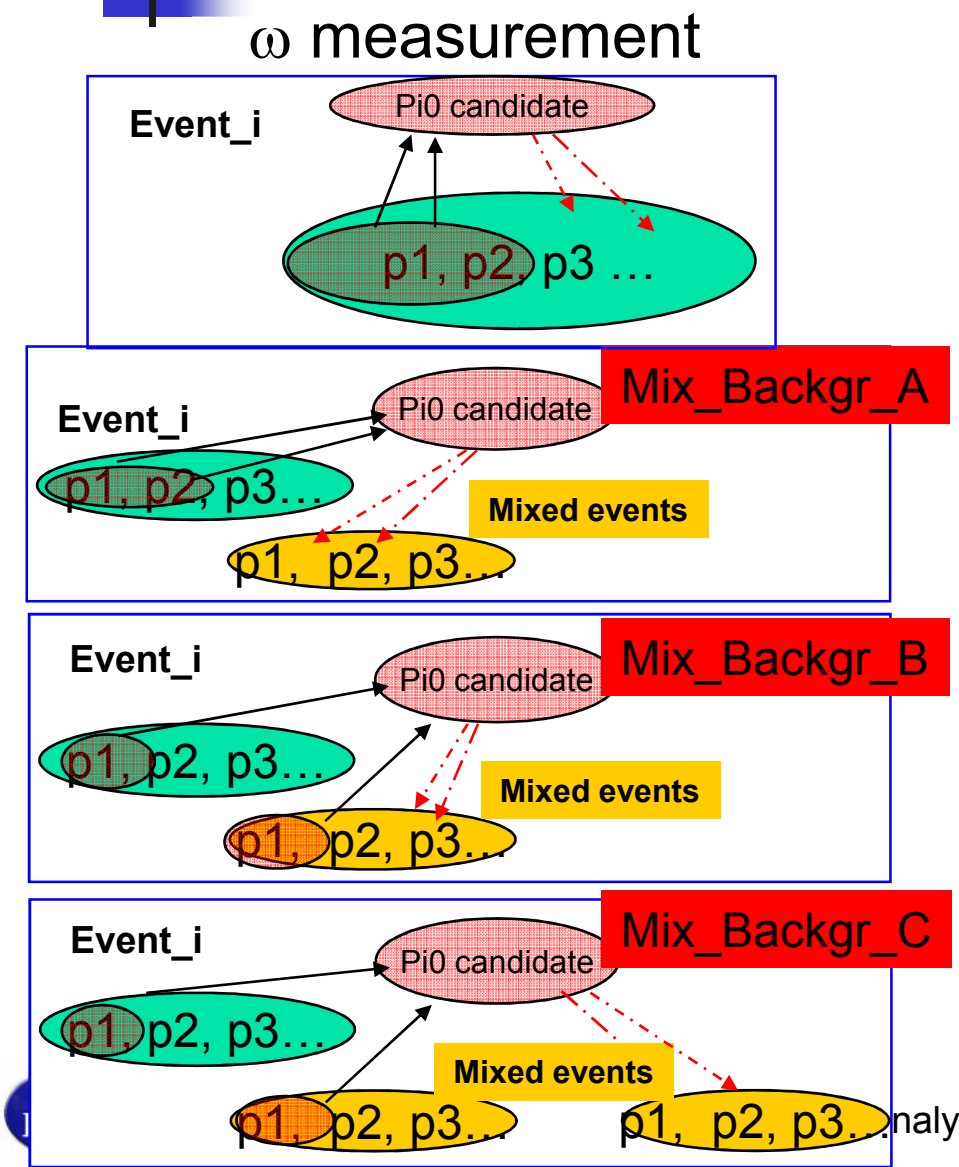
    - In InitParameters() or configuration file:

```
SetAODRefArrayName("CorrelatedParticles");
```

PartCorr: New analysis class AliAnaCalorimeterQA

- Class for calorimeter data checking after productions.
- Produces and plots several useful histograms to check possible problems in the calorimeters.
  - Some of the plots can be seen in the weekly meeting agenda: http://indico.cern.ch/conferenceDisplay.py?confId=59405
  - More plots and corrections added since these meeting.
- After discussion with Yves, this code must be moved to the EMCAL/PHOS QA code when checking the ESDs.
  - At this point I will remove this class or leave it there as an analysis example. Not decided yet.
- Most of the histograms are dependent on access to stack information, so a switch on the kind of data MC or Real should be used.
- In case a production was run without QA, or corrections are done in the calorimeters QA and we need to reprocess data we have to create a new task that should be able to execute the QA at the ESD level, and do the corresponding plots.

PartCorr: New analysis class AliAnaNeutralMeson



- Class for the pi0, eta mesons reconstruction via invariant mass analysis of 2 photons and omega via 3 photons.
- Still under test, it will replace the existing AliAnaPi0.
- Work for PHOS and EMCAL.
- It contains its own mergin procedure, to be changed.
- In Charge Renzhuo Wan.
- For more details:
  - http://indico.cern.ch/conferenceDisplay.py?confId=60213

PartCorr: New analysis class AliAnaElectron

- Why?
  - For historical reasons, EMCAL physics (jets, high p_T spectra) has been part of PWG4
    - We want to tag b-jets via electron decays
    - We want to do high p_T HF electron spectra/R_{AA}
- How?
  - Created AliAnaElectron (parallel to AliAnaPhoton) to perform EMCAL-centered electron analysis/cuts on ESD info**
    - Should be available in svn by next tagged release (When is that?)
- What?
  - Output is histograms and AOD branch with EMCAL-identified electron candidates in AliAODPWG4Particle objects
- Then what?
  - New class AliAnaElectronJetFinderCorrelation will combine AOD jet info with AOD electron info to tag b-jets with different tagging methods (soft lepton, DVM, etc.)
  - New classes for electron spectra, R_{AA} will be developed using electron candidates
- Who?
  - Jenn Klay, Ken Read, Mark Heinz et al.

PartCorr: New analysis class AliAnaElectron

- Why ESDs?
  - We need ESDs to get AliExternalTrackParams** to propagate tracks to EMCAL for further refined track-cluster matching
    - Necessary because not all electron analyses will want the same cuts (efficiency vs. purity trade-off)
    - Only "loose" cuts used in reconstruction track-cluster matching
  - AliAnaElectron uses the PWG4 Configuration code to specify cut ranges, histogram binning, etc.

**Is there a way to get this from the AOD?

PartCorr: New analysis class AliAnaElectron

- PWG4/PWG3/PWGxx Connection
  - We are aware that there is significant overlap between HFE group of PWG3 and HFE group of PWG4 and probably even other groups (i.e. Anyone working on HFE flow in PWG2?)
  - Recently initiated joint discussions with PWG3 to try to combine efforts, merge common algorithms to one code**
    - Should develop global ALICE electron PID using CTS, TRD, EMCAL, etc. in combination
    - However, individual analyses will have different needs and one single global PID will probably never be sufficient
    - Flexibility for users to make specialized cuts within a standard framework should be goal
  - Planning a PWG3/4 HFE meeting during ALICE week, future collaboration
  - Expect more news at next offline week

**Should there be an über-PWG manager to help identify overlaps and encourage inter-PWG collaboration? Does one already exist? Is there a formal structure (i.e. not just someone telling someone else to "talk to" so-and-so in PWGxx)

GammaConv: Status

- Photon reconstruction via electron pairs measured in central tracking system
  - In charge: Katrin, Kenneth and Ana
- Recent developments
  - Write AODs with reconstructed photons
  - Created file to be added to the train
    - in contact with Mihaela to make it work
- Many coding violations were found and are being solved.

Calorimeters calibration

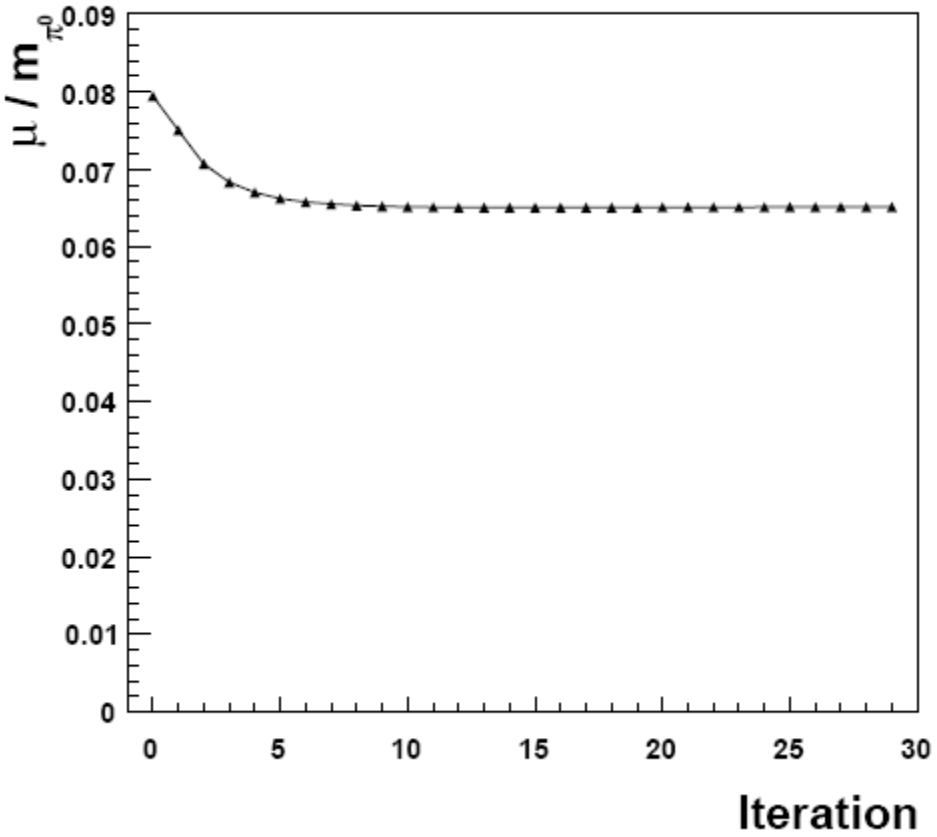
- AliAnalysisTaskPi0CalibSelection.cxx
  - Inherits from AliAnalysisTaskSE
  - Input: collection of AliESDs.root files
  - Energy and coordinate of AliESDCaloCluster's is recalculated by applying the corrections to the calibration coefficiencts calculated in the previous iteration
  - Adapted only for PHOS
  - Output: 5x64x56 TH1 histograms with a 2-cluster invariant mass in each cell stored in a .root file
- pi0Calib.C
  - Input: merged outputs of AliAnalysisTaskPi0CalibSelection
  - Output: corrections (~1) to the calibration coefficients
- The code is in the aliroot trunk: $ALICE_ROOT/PHOS/macro/pi0Calib

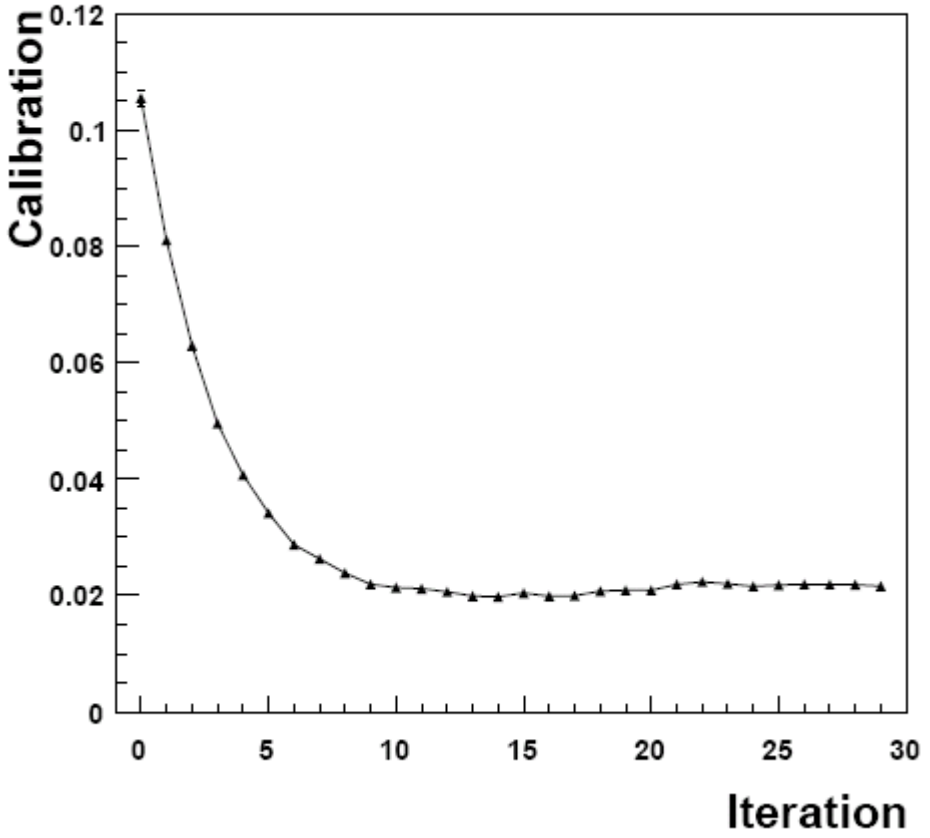
Calorimeters calibration: Status and To Do

- Boris P. is in charge of its implementation in PHOS
- Tested in AliEn with LHC09a4 pp min. bias production
- Size of produced root file : 10 Mb, does not scale with statistics
- Needs further development:
  - To work with AOD instead of ESD (AliAODCluster and AliAODCells)
  - To compile into a shared library
  - To take the PHOS geometry from OCDB GRP using ESD information, **or**
  - To store rotation matrices in ESD (where, how?)
- EMCAL: I have to work in its adaptation. It won't take me much time to adapt, but to test …
- Where do we keep this code, in PHOS/EMCAL directories or in PWG4/CaloCalib directory for example?

Convergence of calibration procedure (H.Qvigstad)

Validated with single-pi0 events at pT=10 GeV/c





Relation to offline calibration framework

- The first reconstruction pass should produce "Calo-wise AOD": AOD with only PHOS (EMCAL) clusters and cells, and the primary vertex.
- The first iteration can work at the end of the first reconstruction pass
- Any further iterations will work apart from the calibration framework
- New calibration objects should be submitted to OCDB manually.

To Do

- PartCorr
  - Produce delta AODs, modify example macros.
  - Move AOD classes to new library.
  - How to handle the common PWG3/PWG4 code for electron identification.
  - Implement mixing analysis in PartCorr frame
  - Put AliAnaNeutralMeson in the train
  - Commit AliAnaElectron and put it on the train.
    - How to work with AODs
  - Update documentation
  - …
- GammaConv
  - Put it on the train
  - How to connect GammaConv output with PartCorr analysis.
  - Correct coding violations
  - …
- Calorimeters calibration
  - PHOS: Make it work with AODs
  - EMCAL: Write it.

Backup

PartCorr: Particle identification/selection analysis

- Photon identification (shower shape): AliAnaPhoton
  - Input: ESD CaloClusters
  - Output: aod (AliPWG4Particle(Correlation) objects) and histograms
- Pi0 identification via invariant mass: AliAnaPi0
  - Input: aod (only AliPWG4Particle objects) from AliAnaPhoton
  - Output: histograms
  - Uses mixing event techniques.
  - Under study the implementation of the eta and omega identification cases.
  - Access to PHOS geometry possible when the PHOSutils library is loaded.
- Pi0 identification on Event-by-Event basis: AliAnaPi0EbE, 3 cases:
  - Calorimeters invariant mass and other selections cuts
    - Input: aod from AliAnaPhoton
    - Output: aod (AliPWG4Particle(Correlation) objects), histograms
  - Calorimeter + Gamma Conversion invariant mass and other selection cuts
    - Input: aod from AliAnaPhoton and aod from GammaConv package (when available)
    - Output: aod (AliPWG4Particle(Correlation) objects) , histograms
  - Pi0 is a single cluster, decay photons overlap, shower shape identification of pi0
    - Input: ESD CaloClusters
    - Output: aod (AliPWG4Particle(Correlation) objects) and histograms
- Charged Particles selection:new, not committed, AliAnaChargedParticles
  - Input: ESD Tracks
  - Output: aod (AliPWG4Particle(Correlation) objects) and histograms

PartCorr: Particle selection/correlation analysis

- Particle Isolation: AliAnaParticleIsolation
  - Input : aod (AliPWG4ParticleCorrelation objects) from AliAnaPhoton, AliAnaPi0EbE, AliAnaChargedParticles …
  - Output: histograms, modifies aod adding an isolation label and adding a list with references to tracks/caloclusters falling in isolation cone.
- Particle correlation with hadrons: AliAnaParticleHadronCorrelation
  - Hadrons in a wide angular window opposite to the trigger particle, and with a selected pt cut.
  - Input : aod (AliPWG4ParticleCorrelation objects) from AliAnaPhoton, AliAnaPi0EbE, AliAnaChargedParticles …
  - Output: histograms, modifies aod adding a list with references to the selected hadrons.
- Particle correlation with jets:
  - AliAnaParticleJetFinderCorrelation: correlation with aod jet from JETAN
  - AliAnaParticleJetLeadingConeCorrelation: jet reconstruction algorithm depending on trigger particle.
  - Input : aod (AliPWG4ParticleCorrelation objects) from AliAnaPhoton, AliAnaPi0EbE, AliAnaChargedParticles … and aod from JETAN (only for AliAnaParticleJetFinderCorrelation)
  - Output: histograms, modifies aod adding a list with references to the jet hadrons and a reference to the aod jet.

PartCorr: Output AOD objects

- AliAODPWG4Particle: Derives from AliVParticle, suitable for particle identification studies, quite light object. Its data members are:
  - Particle kinematics:
    - TLorentzVector fMomentum:
  - Monte Carlo:
    - Int_t fLabel: Id of original MC particle
    - Int_t fTag: Tag particle as Decay, Fragmentation, Prompt, Conversion … defined in class AliAnalysisMCUtils
  - Connect with original ESD/AOD Id number, necessary for isolation studies.
    - Int_t fTrackLabel[2]: if original particle is a track put here the index, if is reconstructed from 2 tracks (conversion gamma), put 2 indexes.
    - Int_t fCaloLabel[2]: if original particle is a calocluster put here the index, if is reconstructed from 2 caloclusters (pi0), put 2 indexes.
  - PID:
    - Int_t fPdg: Assigned identification label after PID
    - bits for PID selection of caloclusters in later stages of the analysis
      - Int_t fTof: Time of Flight
      - Int_t fDisp: Shower shape dispersion
      - Int_t fCharged: cluster is charged.
  - Quality of the cluster:
    - Int_t fBadChannel: Distance to bad channel
  - Detector of origin:
    - TString fDetector : PHOS, EMCAL, CTS

PartCorr: Output AOD objects

- AliAODPWG4ParticleCorrelation: Derives from AliAODPWG4Particle, suitable for correlation studies, data members are:
  - Isolation studies:
    - Bool_t fIsolated: Flag for Isolation
    - TRefArray* fRefIsolationTracks(Clusters): Reference array with tracks (caloclusters) found in the trigger particle isolation cone.
  - Trigger particle - Jet/Hadron correlation stdudies
    - TList *fListOfRefArrays: List of Reference arrays of correlated tracks (clusters) with the trigger particle for different porpouses.
    - TRef fRefJet: Reference to jet found with JETAN
    - TLorentzVector * fCorrJet(Bkg): Kinematics of jet (background) found with gamma-tagging techniques
    - TLorentzVector * fLeading: Kinematics of Jet core / leading particle

PartCorr: AliMCAnalysisUtils

- Class devoted for analysis utils relative to information stored stack, headers, etc.
- Two methods for the moment:
  - Int_t CheckOrigin(): Given a particle it assigns a label to know its origin, right now concentrated on photons:
    - enum mcTypes {kMCPrompt, kMCFragmentation, kMCISR, kMCPi0Decay, kMCEtaDecay, kMCOtherDecay, kMCPi0, kMCEta, kMCElectron, kMCConversion, kMCUnknown};
    - Depends on the generator used, right now only works for PYTHIA and HERWIG, if needed other generators could be considered.
  - TList *GetJets(): Returns the lists of generated jets (TParticles)
    - Depends on the Generator, only 2 options
      - PYTHIA: Returns jets found with Pycell
      - HERWIG: Generated objects with PDG 94, (CMShowers), not real jets but close.