

EMCal Jet Framework

Salvatore Aiola, Rüdiger Haake, Constantin Loizides

Setting up an analysis in the EMCal Jet Framework

To analyze jets, you need three things:

- I The **framework** to prepare everything for jet analysis (e.g. given with `$ALICE_ROOT/PWGJE/EMCALJetTasks/macros/runEMCalJetAnalysis.C`, link can be found on slide 6)
- II A **class** for your analysis task (`AliAnalysisTaskX.cxx/.h`)
- III A **macro** to add your task and the jet finders to the analysis manager. It should return a pointer to your task, if it was created successfully.

This is can be done with the following steps:

- Create your own analysis task by using the example **class** `AliAnalysisTaskEmcalJetSample` as a template for this.
- Do the same for the **macro** `AddTaskEmcalJetSample.C`
- Modify the **framework** macro `runEMCalJetAnalysis.C` so that it calls your task in the way you want (see next slide)

Usage of the runEMCalJetAnalysis.C

- runEMCalJetAnalysis.C is a convenient method to use the framework.
- To set it up, you have to do two things in this macro:
 - 1 Call the macro that adds your task to the analysis

```
152     AliTender *alltender = dynamic_cast<AliTender*>(tender);
153   }
154 }
155
156 // ##### Now: Call jet preparation macro (picotracks, hadronic corrected caloclusters, ...)
157
158 // Jet preparation
159 gROOT->LoadMacro("$ALICE_ROOT/PWGJE/EMCALJetTasks/macros/AddTaskJetPreparation.C");
160 AddTaskJetPreparation(dataType);
161
162 // ##### Now: Add jet finders+analyzers
163
164 gROOT->LoadMacro("$ALICE_ROOT/PWGJE/EMCALJetTasks/macros/AddTaskEmcalJet.C");
165 AliEmcalJetTask* jetFinderTask = AddTaskEmcalJet("PicoTracks", "CaloClustersCorr", KANTIKT, 0.2,
166 kCHARGEDJETS, 0.150, 0.300);
167
168 // Here you can put in your AddTaskMacro for your task
169 gROOT->LoadMacro("$ALICE_ROOT/PWGJE/EMCALJetTasks/macros/AddTaskEmcalJetSample.C");
170 AliAnalysisTaskEmcalJetSample* anaTask = AddTaskEmcalJetSample("PicoTracks", "CaloClustersCorr",
171 jetFinderTask->GetName());
172
173 // Set the physics selection for all given tasks
174 TObjArray *tootasks = mar->GetTasks();
```

Jet finder

Your macro here

- 2 Choose preferences (i.e. data type, run numbers, grid dir, event selection, jet finder, etc.)

Note: If you want to run with local data, you have to specify the file locations in files_esd.txt, files_aod.txt, files_sesd.txt depending on the data type you are using

Adding Jet Finders

```
// ### Some constants to make code more readable
const Int_t kKT          = 0;
const Int_t kANTIKT     = 1;

const Int_t kFULLJETS   = 0;
const Int_t kCHARGEDJETS = 1;
const Int_t kNEUTRALJETS = 2;
// ### Settings for full jets in ESD/AOD data

// used jet finding algorithm
Int_t usedAlgo = kANTIKT;
// used jet type
Int_t usedJets = kFULLJETS;
// used track collection
const char* tracks = "PicoTracks";
// used cluster collection
const char* clusters = "CaloClustersCorr";
// radius for jet finder
Double_t radius = 0.4;

// ### Run jet finder
gROOT->LoadMacro("$ALICE_ROOT/PWGJE/EMCALJetTasks/macros/AddTaskEmcalJet.C");
AliEmcalJetTask* jetFinderTask = AddTaskEmcalJet(tracks, clusters,
                                                kANTIKT, radius,
                                                kFULLJETS, minTrackPt, minClusterPt);
TString jetCollectionName = jetFinderTask->GetName();
```

This is how to get the name of the jet collection

In principle, you just need that one line to add a jet finder

Accessing Jets, Tracks, and Clusters

If you get started in the framework...

- Derive your task from Salvatore Aiola's jet example class
AliAnalysisTaskEmcalJet
so you don't have to bother with technical details of getting the jets



Or use your existing tasks in the framework

- In your code, you can get the array pointers by
trackArr = **dynamic_cast**<TClonesArray*>(InputEvent()->FindListObject("PicoTracks"));
clusArr = **dynamic_cast**<TClonesArray*>(InputEvent()->FindListObject("CaloClustersCorr"));

Note: The same applies for the jets. The names of the jet collections are given by the jet finder tasks' names.

- All collections (i.e. PicoTracks, CaloClustersCorr, etc.) have been saved by previous tasks with the implementation of AliVEvent::AddObject(TObject* obj) as TClonesArrays in the ESD event.
- Then, to get e.g. the tracks from the TClonesArrays one can use
AliVTrack* 1stTrack = **static_cast**<AliVTrack*>(trackArr->At(0)); // get first track in array

Links

Template for an EMCal Jet Analysis

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWGJE/EMCALJetTasks/macros/runEMCalJetAnalysis.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWGJE/EMCALJetTasks/macros/AddTaskEmcalJetSample.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWGJE/EMCALJetTasks/AliAnalysisTaskJetSample.cxx?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWGJE/EMCALJetTasks/AliAnalysisTaskJetSample.h?root=AliRoot>

Basis tasks and jet preparation tasks:

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/ANALYSIS/macros/AddTaskCentrality.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWG/EMCAL/macros/AddTaskEmcalPhysicsSelection.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWG/EMCAL/macros/AddTaskEmcalSetup.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWG/EMCAL/macros/AddTaskEmcalEsdTrackFilter.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWG/EMCAL/macros/AddTaskEmcalPicoTrackMaker.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWG/EMCAL/macros/AddTaskEmcalParticleMaker.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWG/EMCAL/macros/AddTaskEmcalClusTrackMatcher.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWGJE/EMCALJetTasks/macros/AddTaskHadCorr.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWG/EMCAL/macros/AddTaskMCTrackSelector.C?root=AliRoot>

<http://svnweb.cern.ch/world/wsvn/AliRoot/trunk/PWGJE/EMCALJetTasks/macros/AddTaskEmcalJet.C?root=AliRoot>

Backup

runEMCALJetAnalysis.C – General Overview

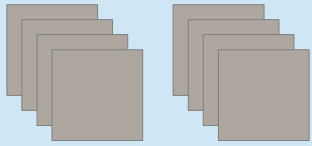
AliEmcalPhysicsSelectionTask

AliCentralitySelectionTask

AliEmcalSetupTask

[your additional basic tasks]

Basic tasks



(described on next to slides)

Jet preparation macro

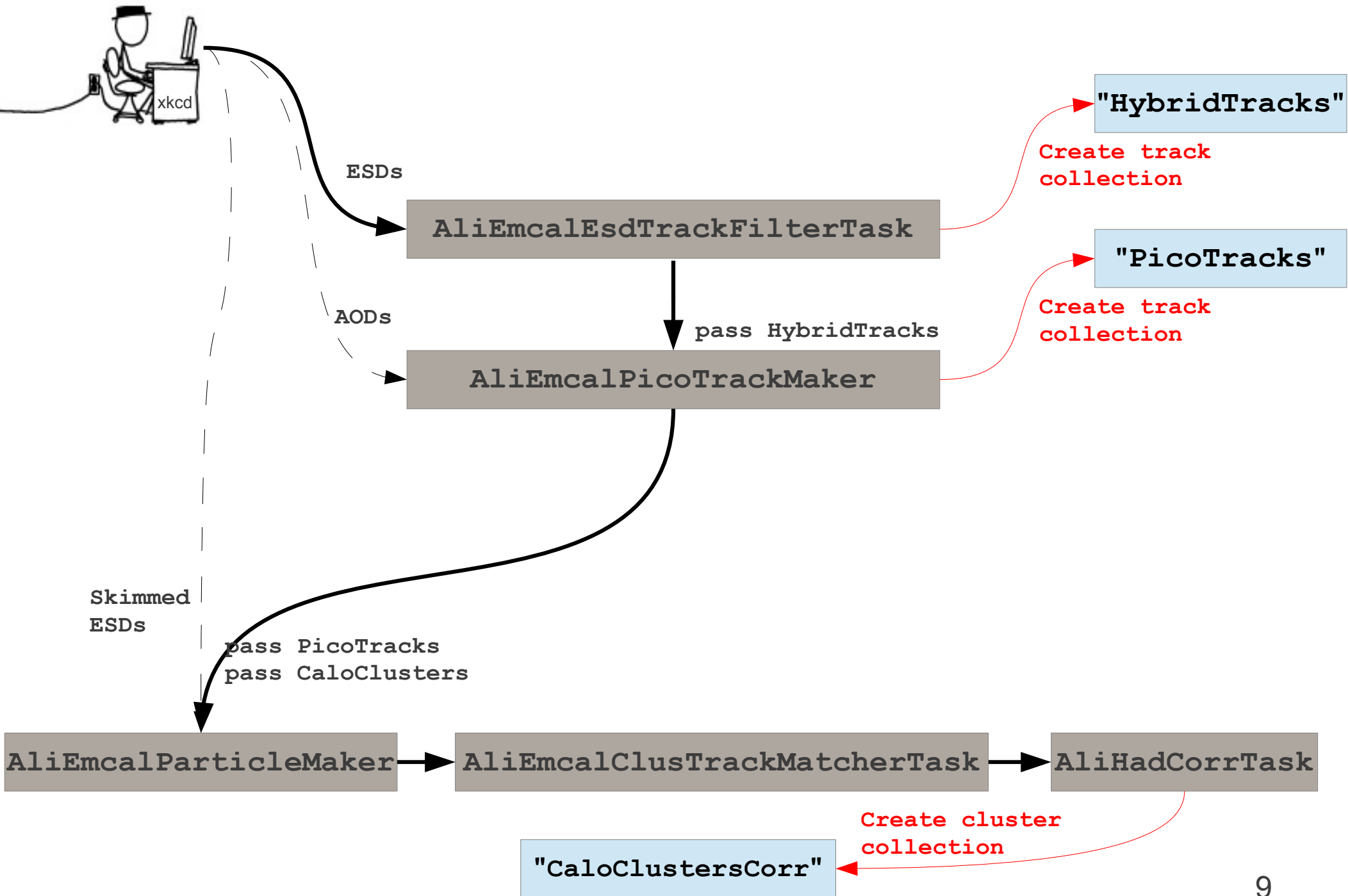
Jet Finder I
Jet Finder II

Analysis I
Analysis II
Analysis III

User defined tasks

runEMCALJetAnalysis()

Jet Preparation Macro – Details I



Jet Preparation Macro – Details II

Collections:

- "HybridTracks": Tracks (`AliESDtrack`) which obey the trackcuts of the period and which are constrained by SPD vertex
- "PicoTracks": Tracks in more compact format (`AliPicoTrack`)
- "CaloClustersCorr": EMCal clusters (`AliESDCaloCluster` or `AliAODCaloCluster`) after hadronic corrections. These corrections are done to avoid double counting of tracks that deposit energy in the EMCal

NOTE: All collections are saved as TClonesArrays in event via the implementation of `AliEvent::AddObject(TObject* obj)`

- Skimmed ESDs: Lightweight ESDs containing `AliPicoTracks` instead of `AliESDtracks` and only preselected events.
- (not necessary to run in the EMCal framework,
details can be found in `$ALICE_ROOT/PWG/EMCAL/macros/AddTaskEsdSkim.C`)