



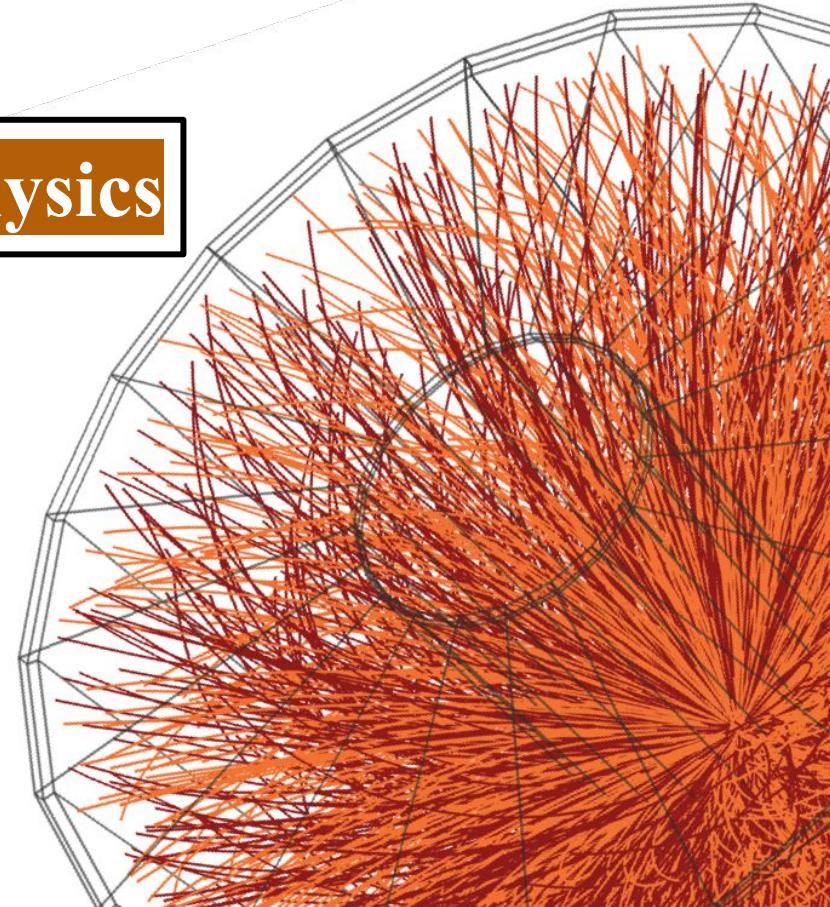
ALICE

Jadavpur University

Particle Identification in O²Physics

Hirak Kumar Koley
(hirak.koley@cern.ch)
Jadavpur University

O² Analysis Tutorial v4.0
<https://indico.cern.ch/event/1425820/>
15th October 2024

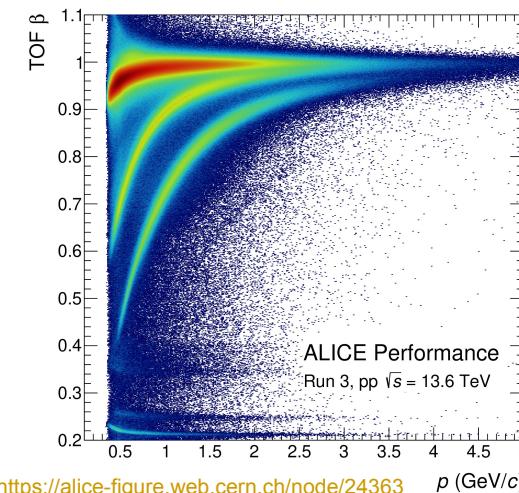
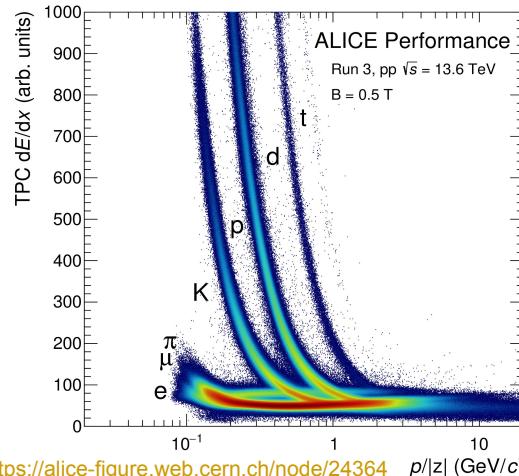


Goal of this session



Get you acquainted with:

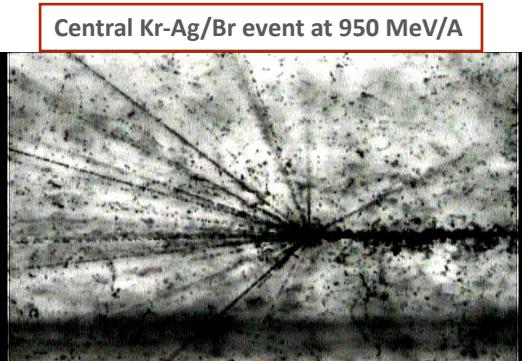
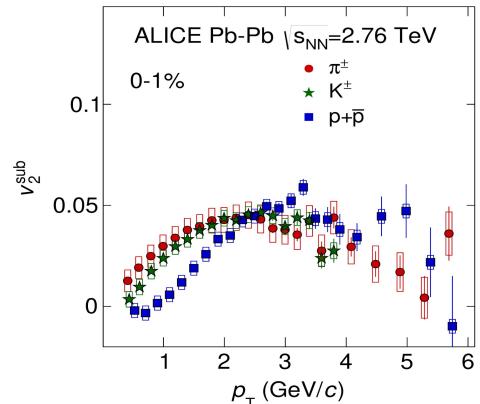
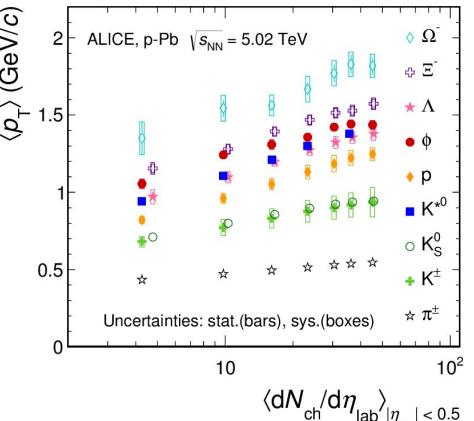
- What is **Particle IDentification (PID)** ?
- **Detectors** for PID
- **Software tools** for PID in **O²Physics**
- Work with **additional helper tasks**
- Use detector responses to perform **PID in analysis**
- **Assess quality** of particle identification with helper tasks
- Produce **beautiful plots** for PID performance
- Get help if things don't work as expected



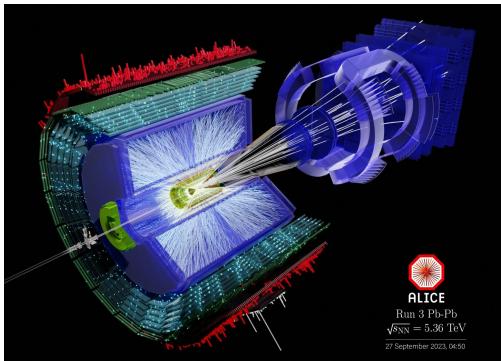
Particle Identification



- From the thousands of particle crowd, we need to identify concerned particles
- Introducing PID, we get wealth of information and new insights i.e. more information towards our goal



Few particles only



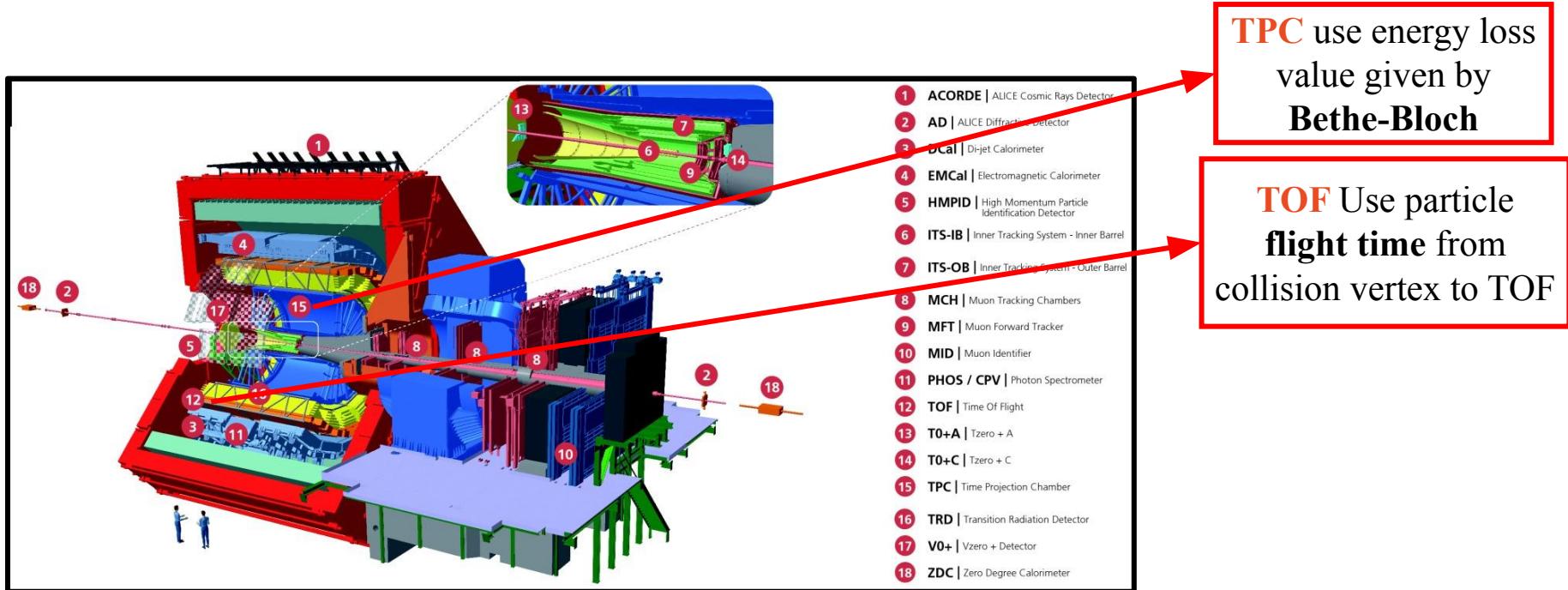
Thousands of particles



How to do PID:

- **Stable Particles:** Detector response that depends on particle velocity
 - Energy loss via Ionization ($dE/dx \sim 1/\beta^2$)
 - Time-of-flight, $\tau \sim 1/\beta$
 - Emitted transition radiation $\sim 1/\gamma$
 - Cherenkov emission, $\cos(\theta) = 1/n\beta$
- **Calorimetry:** full energy deposition, e-photon-hadron discrimination from shower profile
- **Unstable particles:** Invariant mass reconstruction / topological decay

ALICE In Run 3



PID using TPC-TOF: "nσ-cut" method



Specific energy loss (dE/dx) of the track compared with the expected energy loss (Bethe-Bloch formula) for different particle species.

$$n\sigma = \frac{dE/dx_{measured} - dE/dx_{expected}}{\sigma_{PID(TPC)}}$$

- $dE/dx_{measured}$: measured energy loss in the TPC gas
- $dE/dx_{expected}$: expected energy loss for particular particle hypothesis
- $\sigma_{PID(TPC)}$: TPC dE/dx resolution on the measurement which depends on detector performance

Time of flight of the particle compared with the expected arrival time of flight for a given particle species hypothesis.

$$n\sigma = \frac{t_{measured} - t_{expected}}{\sigma_{PID(TOF)}}$$

$$t_{measured} = t_{hit} - start_{Time}$$

- t_{hit} : particle arrival time measured in the TOF detector
- $start_{Time}$: event collisions time (namely the particle start time)
- $t_{expected}$ is computed by the ALICE reconstruction taking into account track length, momentum and energy loss in the material
- $\sigma_{PID(TOF)}$ depends on
 - intrinsic resolution of the TOF detector
 - tracking capabilities of the ALICE
 - the precision in the determination of the collision time of the events → depends on the "TOF" detector resolution, number of tracks reaching the TOF and the spread of the vertex.

$$\sigma_{PID(TOF)} = \sqrt{\sigma_{TOF}^2 + \sigma_{starttime}^2 + \sigma_{tracking}^2}$$

Data model and helper tasks

- Adding pieces to the data model

<https://indico.cern.ch/event/1267433/timetable/#3-introduction-to-framework>

Running analysis in practice and assembling a workflow

Helper tasks

- Various helper tasks, like [event](#) and [track selection](#), [PID](#), etc., define corresponding tables
- These can be easily accessed in your task by including [corresponding headers](#)
- PWGs maintain their own [data model extensions](#) also defined in headers
- To use the outputs of the helpers tasks, [add them to the workflow](#)
- This is done by [adding the task's binary to the command](#) with | (called "pipe")
- Several tasks can be piped together (note that full configuration needs to be provided to all entries)

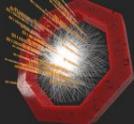
```
o2-analysis-a --configuration=json://file.json |\
o2-analysis-b --configuration=json://file.json |\
o2-analysis-c --configuration=json://file.json --other-option
```

Definitely more in the hands-on!



A guide to the helper tasks

- Advanced analyses need more helper tasks



ALICE O2 Analysis framework
DOCUMENTATION

Search docs...

DOCUMENTATION HOME

- Getting started
- Writing an analysis task
- Running an analysis
 1. Downloading AO2D samples
 2. Setting up an execution
 3. Using json configuration files
- 4. Helper tasks**
 - Timestamp
 - Event selection
 - Multiplicity and centrality selection in O2
 - Particle identification (PID)
 - Track Selection
 - Track Propagation
- Saving tables to a file
- Reading tables from files

<https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html>

Helper tasks

Table of contents:

- [Timestamp](#)
- [Event selection](#)
- [Multiplicity and centrality selection in O2](#)
- [Particle identification \(PID\)](#)
- [Track Selection](#)
- [Track Propagation](#)

Timestamp

The timestamp task is needed to fill the table [Timestamps](#). Timestamp contains the time of a bunch crossing since the start of the run. This time is often needed to retrieve objects in the CCDB (see Tutorial [CCDB](#)).

Since the Timestamps table has an entry per bunch crossing it can be joined with table BC. The join is defined by `o2::aod::BCsWithTimestamps` (see list of defined [joins and iterators](#)).

Event selection

Table of contents:

- [Concept](#)
- [Basic usage in user tasks](#)
- [Trigger aliases](#)
- [Event selection criteria](#)
- [Event selection decisions](#)
- [Found bunch crossings](#)
- [Configurables](#)
- [Remarks](#)

A guide to the helper tasks

- Advanced analyses need more helper tasks

<https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html>

Helper tasks

Table of contents:

- [Timestamp](#)
- [Event selection](#)
- [Multiplicity and centrality selection in O2](#)
- [Particle identification \(PID\)](#)
- [Track Selection](#)
- [Track Propagation](#)

4. Helper tasks

Timestamp

The Timestamp task is used to get the start of the run. This time is often needed to retrieve objects in the CCDB (see Tutorial o2:aod::BCsWithTimestamps (see list of defined [joins and iterators](#))).

Event selection

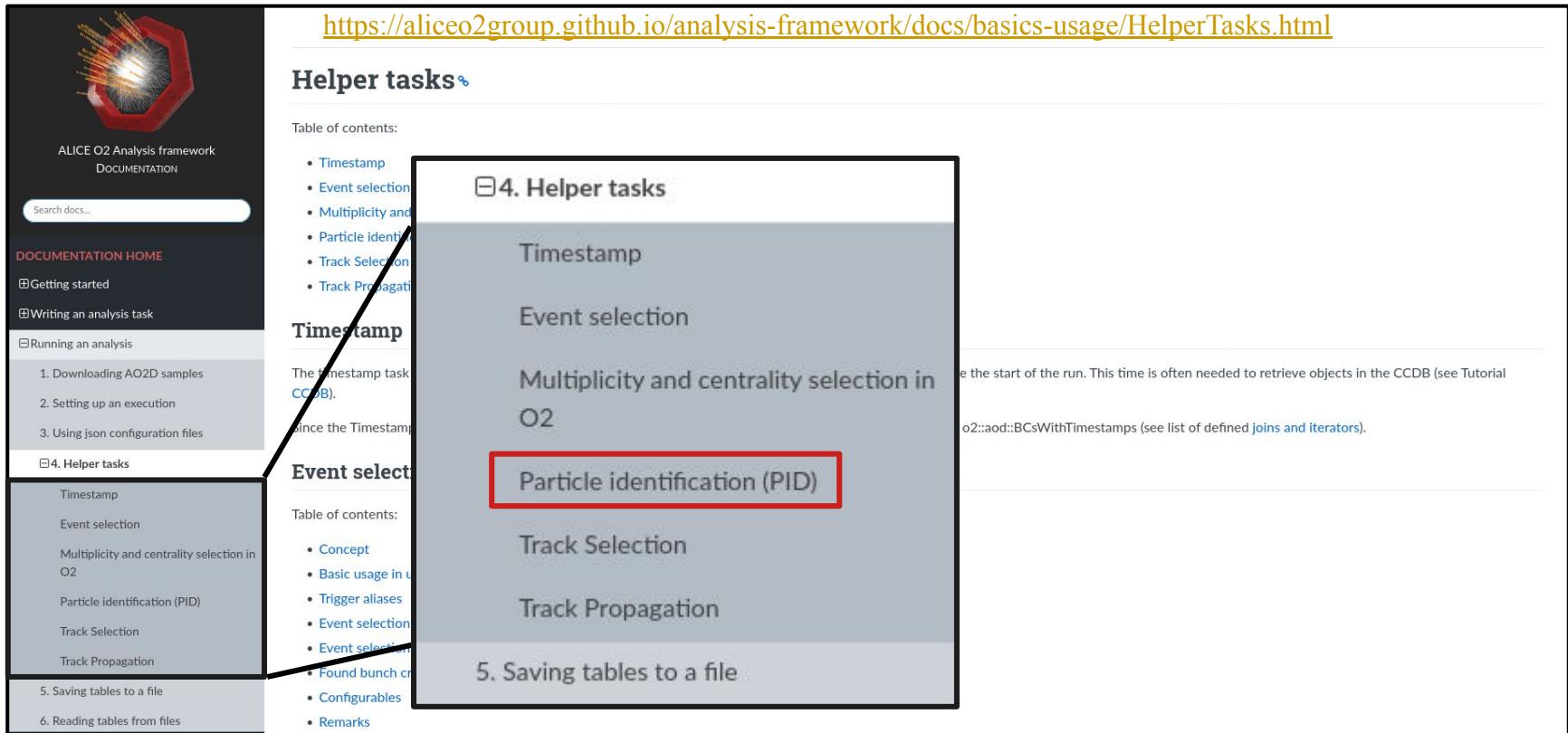
Multiplicity and centrality selection in O2

Particle identification (PID)

Track Selection

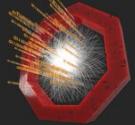
Track Propagation

5. Saving tables to a file



A guide to the PID helper tasks

- Guide to use the PID information, should cover all cases


ALICE O2 Analysis framework
DOCUMENTATION

Search docs... ▾

DOCUMENTATION HOME

- ▢ Getting started
- ▢ Writing an analysis task
- ▢ Running an analysis
 - 1. Downloading AO2D samples
 - 2. Setting up an execution
 - 3. Using json configuration files
- ▢ 4. Helper tasks
 - Timestamp
 - Event selection
 - Multiplicity and centrality selection in O2
- ▢ Particle identification (PID)
 - Track Selection
 - Track Propagation
- 5. Saving tables to a file
- 6. Reading tables from files

ALICE O2 documentation master ▾

Particle identification (PID)

Table of contents: <https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html#particle-identification-pid>

- [Introduction](#)
- [Usage in user tasks](#)
- [Task for TOF and TPC PID](#)
- [Example of tasks that use the PID tables \(and how to run them\)](#)

Here are described the working principles of Particle Identification (PID) in O2 and how to get PID information (expected values, nSigma separation et cetera) in your analysis tasks if you plan to identify particles.

Introduction

PID is handled in analysis by filling helper tables that can be joined to tracks (propagated or not). The parameterization of the expected detector response (e.g. signal, resolution, separation) is used in the PID tasks to fill the PID tables. These parameterizations are detector specific and handled by the detector experts; usually, they are shipped to the PID helper tasks from the CCDB to match the data-taking conditions. The interface between the detector and the Analysis Framework (i.e. your tracks) is fully enclosed in [PIDResponse.h](#). Here are the defined tables for the PID information for all the detectors.

The filling of the PID tables is delegated to dedicated tasks in [Common/TableProducer/PID/](#). Examples of these tasks can be found in [pidTOF.cxx](#) and [pidTPC.cxx](#) for TOF and TPC tables, respectively.

Usage in user tasks

Tables for PID values in O2 are defined in [PIDResponse.h](#). You can include it in your task with:

```
#include "Common/DataModel/PIDResponse.h"
...
```

In the process functions, you can join the table to add the PID (per particle mass hypothesis) information to the track. In this case, we are using the mass hypothesis of the electron, but tables for nine (9) stable particle species are produced ([El](#), [Mu](#), [Pi](#), [Ka](#), [Pr](#), [De](#), [Tr](#), [He](#), [A1](#)).

- For the TOF PID as:

```
void process(soa::Join<aod::Tracks, aod::pidTOFEl>::iterator const& track) {
    track.tofNSigmaEl();
}
```

- For the TPC PID as:

A guide to the PID helper tasks

- Guide to use the PID information, should cover all cases



**ALICE O2 Analysis framework
DOCUMENTATION**

Search docs...

DOCUMENTATION HOME

- Getting started
- Writing an analysis task
- Running an analysis
 - 1. Downloading AO2D samples
 - 2. Setting up an execution
 - 3. Using json configuration files
- 4. Helper tasks
 - Timestamp
 - Event selection
 - Multiplicity and centrality selection in O2
- Particle identification (PID)
 - Track Selection
 - Track Propagation
 - 5. Saving tables to a file
 - 6. Reading tables from files

ALICE O2 documentation master ▾

Particle identification (PID)

<https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html#particle-identification-pid>

Table of contents:

- Introduction
- Usage in user tasks
- Task for TOF and TPC PID
- Example of tasks that use the PID tables

Here are described the working principles of PID.

Introduction

PID is handled in analysis by filling helper table tasks to fill the PID tables. These parameterize conditions. The interface between the detector and the analysis is defined by helper table tasks. The filling of the PID tables is delegated to detector specific helper table tasks.

Usage in user tasks

Tables for PID values in O2 are defined in `PID.h`:

```
#include "Common/DataModel/PIDResponse.h"
...
```

In the process functions, you can join the table to add the PID (per particle mass hypothesis) information to the track. In this case, we are using the mass hypothesis of the electron, but tables for nine (9) stable particle species are produced (`E1`, `Mu`, `Pi`, `Ka`, `Pr`, `De`, `Tr`, `He`, `A1`).

- For the TOF PID as:

```
void process(soa::Join

- For the TPC PID as:

```

You're invited to contact to improve the docs



The bearer of this license is allowed to: Complain, whine, nag, pick on, poke fun at, or in general just be annoying. To **Anyone**.

CERTIFIED COMPLAINER

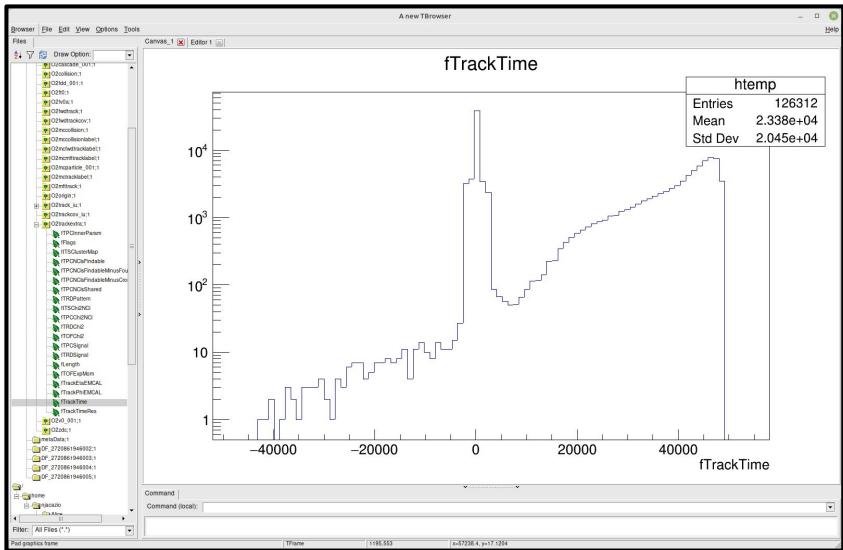
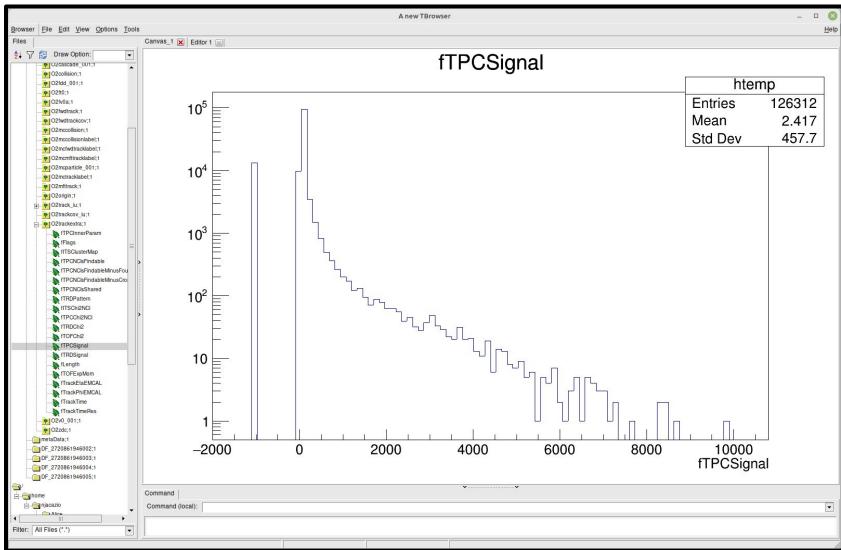
Name _____
E-Mail _____
Signature _____

U.S. Dept. of annoyance Sillyhumor.com

Example: select pions

AO2D.root

“Raw” detector data: $1/p_T$, TPCSignal, TrackTime, ...

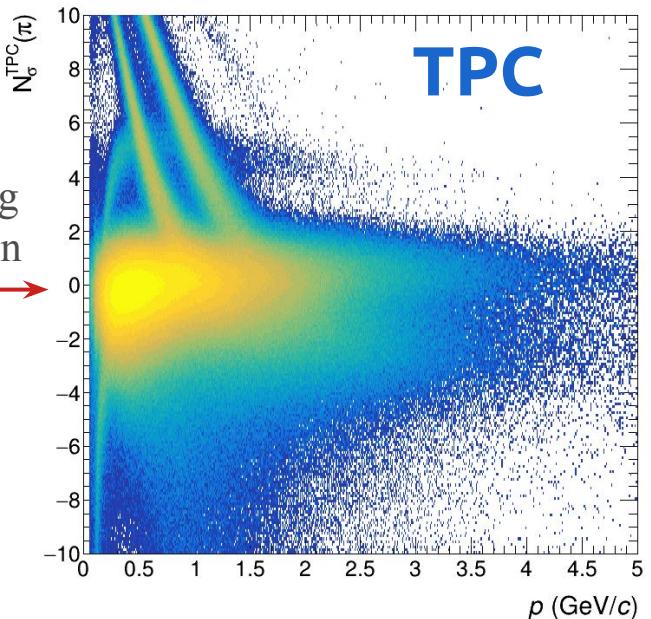


AnalysisResults.root

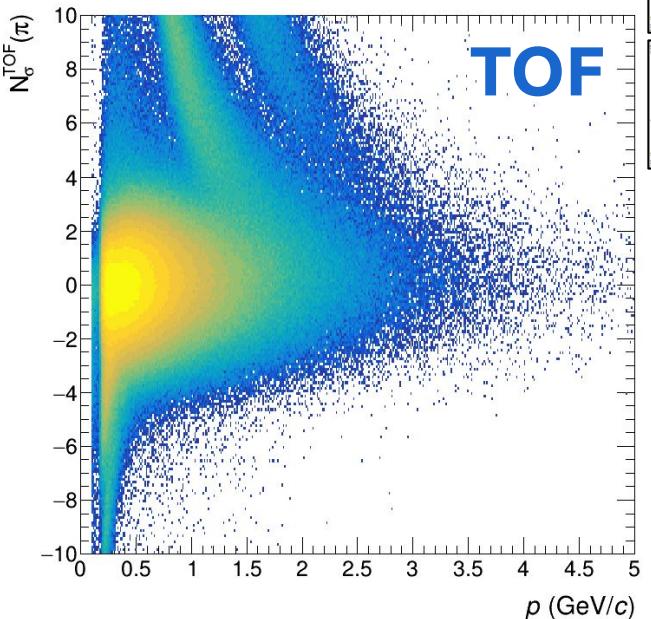
Example: select pions

AO2D.root

“Raw” detector data: $1/p_T$, TPCSignal, trackTime, ...



Processing: building
pion $N\sigma$ information



AnalysisResults.root

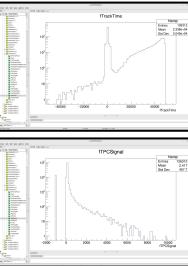
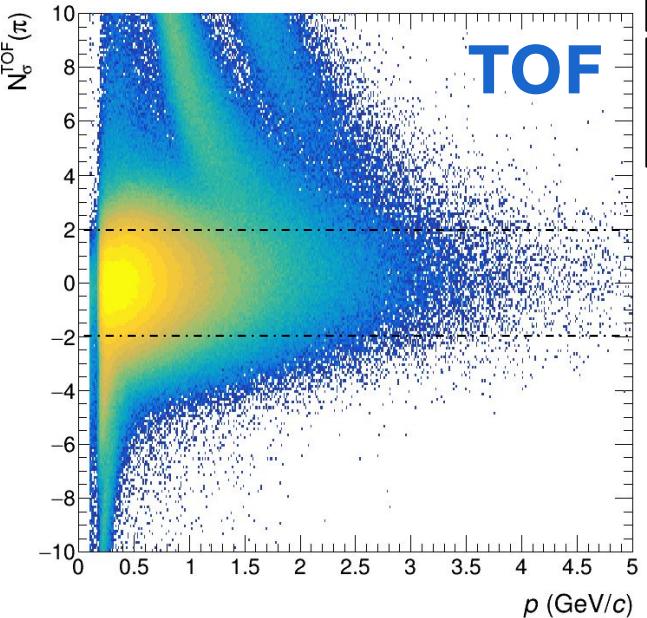
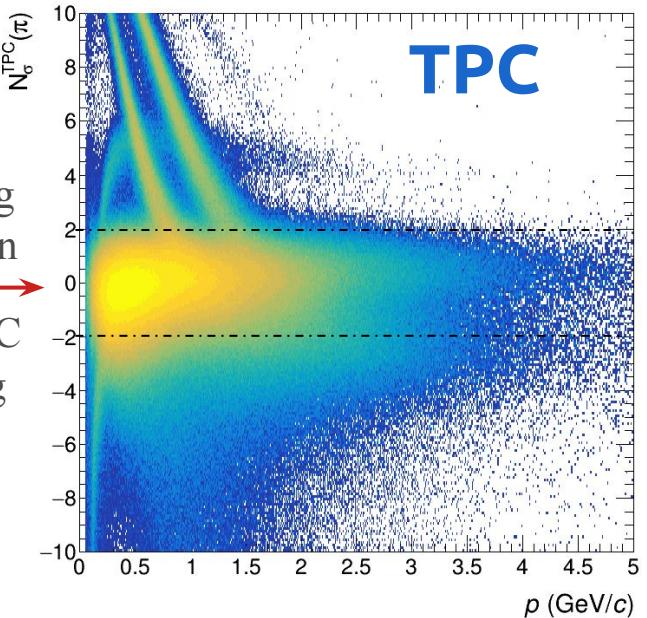
Example: select pions

AO2D.root

“Raw” detector data: $1/p_T$, TPCSignal, trackTime, ...

Processing: building
pion $N\sigma$ information

Cutting at 2σ in TPC
and TOF (e.g. using
partitions or filters)



AnalysisResults.root

Filter & Partitions: <https://indico.cern.ch/event/1425820/timetable/#1-introduction-to-o2o2physics>

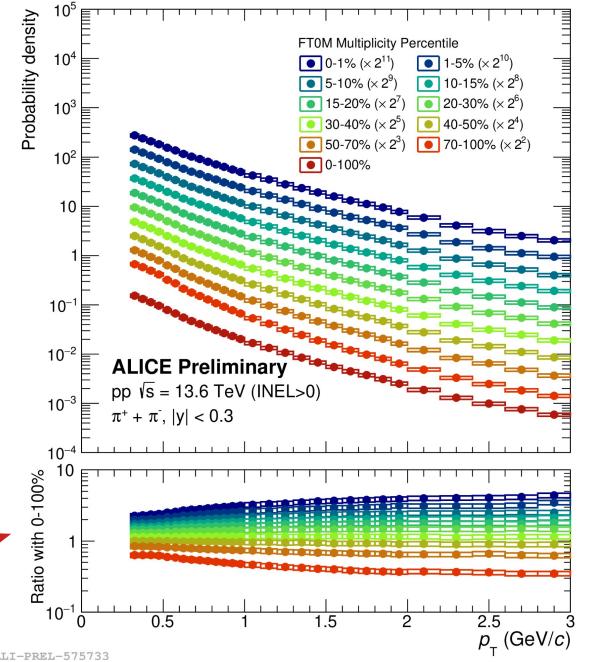
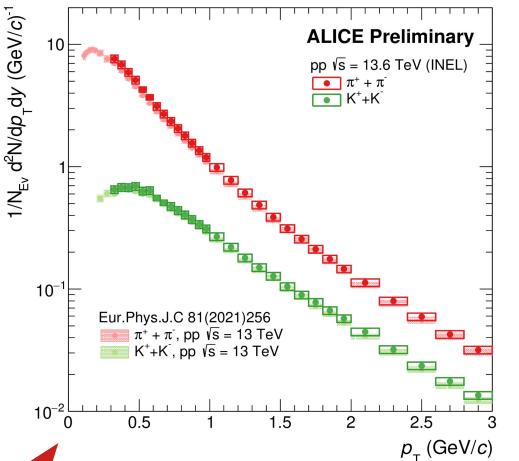
Example: select pions

AO2D.root

Processing: building
pion $N\sigma$ information

Cutting at 2σ in TPC
and TOF (e.g. using
partitions or filters)

“Raw” detector data: $1/p_T$, TPCSignal, trackTime, ...



Your physics results, *OK but how?*

Filter & Partitions: <https://indico.cern.ch/event/1425820/timetable/#1-introduction-to-o2o2physics>

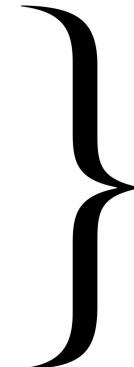
Including PID data formats

- How to access PID data formats, include **PIDResponse.h** header!

```
#include "Common/DataModel/PIDResponse.h"  
...
```

El, Mu, Pi, Ka, Pr, De,
Tr, He(³He), Al(⁴He)

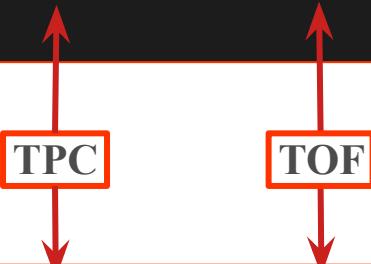
- Tables available for PID
 - Expected values (e.g. Bethe Bloch, expected TOF)
 - Expected resolutions
 - Separation
 - $N\sigma$
 - Extra PID information e.g. TOF mass, TOF beta, ...



Available for all particle mass hypotheses from electrons to α

Subscribing to PID tables

```
void process(soa::Join<aod::Tracks, aod::pidTPCPi, aod::pidTOFPi> const& tracks)
{ ... }
```



```
void process(soa::Join<aod::Tracks, aod::pidTPCKa, aod::pidTOFKa> const& tracks)
{ ... }
```

Pion PID

Kaon PID

- For more than one particle - join the table for each particle.

```
void process(soa::Join<aod::Tracks, aod::pidTPCPi, aod::pidTPCKa,
            aod::pidTOFPi, aod::pidTOFKa> const& tracks) { ... }
```

Pion and Kaon PID

NB: El, Mu, Pi, Ka, Pr, De, Tr, He(³He), Al(⁴He) → are the 9 PID tables available per detector

Documentation → <https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html#usage-in-user-tasks>

Accessing PID info

- Accessing to table content

```
void process(soa::Join<aod::Tracks, aod::pidTPCPi, aod::pidTPCKa,
            aod::pidTOFPi, aod::pidTOFKa> const& tracks)
{
    for (auto& track : tracks) {
        track.tpcNSigmaPi(); // TPC Nsigma of pions
        track.tofNSigmaPi(); // TOF Nsigma of pions
        track.tpcNSigmaKa(); // TPC Nsigma of kaons
        track.tofNSigmaKa(); // TOF Nsigma of kaons
    }
}
```

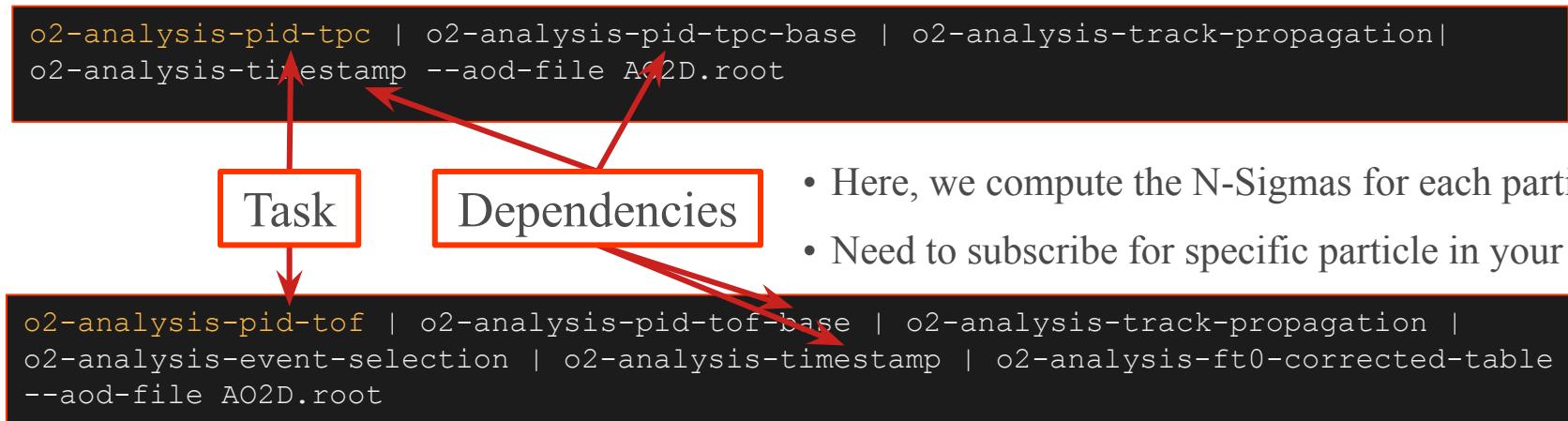
- Can be done by using helper functions

```
void process(soa::Join<aod::Tracks, aod::pidTPCPi, aod::pidTPCKa,
            aod::pidTOFPi, aod::pidTOFKa> const& tracks)
{
    for (auto& track : tracks) {
        o2::aod::pidutils::tpcNSigma<2>(track); // for pions, index = 2 & index runs from 0 to 8
        o2::aod::pidutils::tofNSigma<2>(track); // for pions, index = 2 & index runs from 0 to 8
    }
}
```

Documentation → <https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html#usage-in-user-tasks>

PID tasks in the workflow

- Tasks to include in the workflow

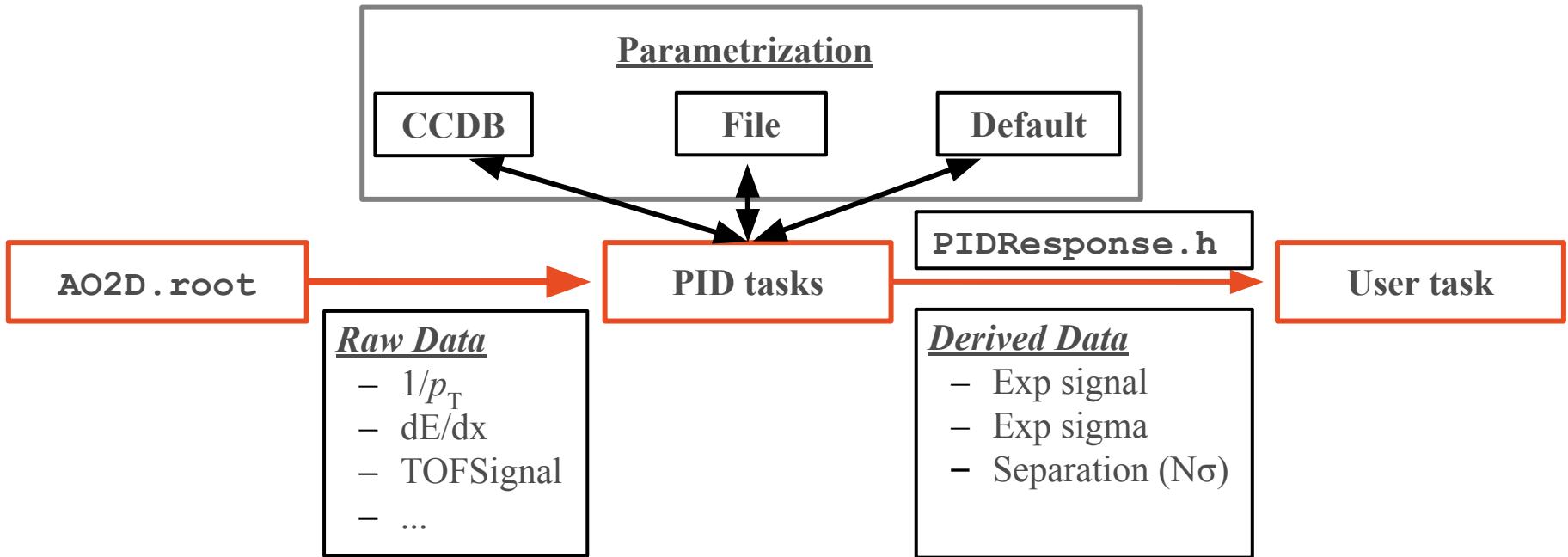


- These tasks produce tables for all particles
- Only tables that are subscribed to are produced
- Configuration is flexible

Documentation → <https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html#task-for-tof-and-tpc-pid>

Advanced features

The complete PID workflow



NB: access to CCDB is needed for the PID calibration
→ timestamp task is a dependency of the PID tasks

Include PID QA output!

- PID tasks come with QA output included!

`o2-analysis-pid-tpc-qa`

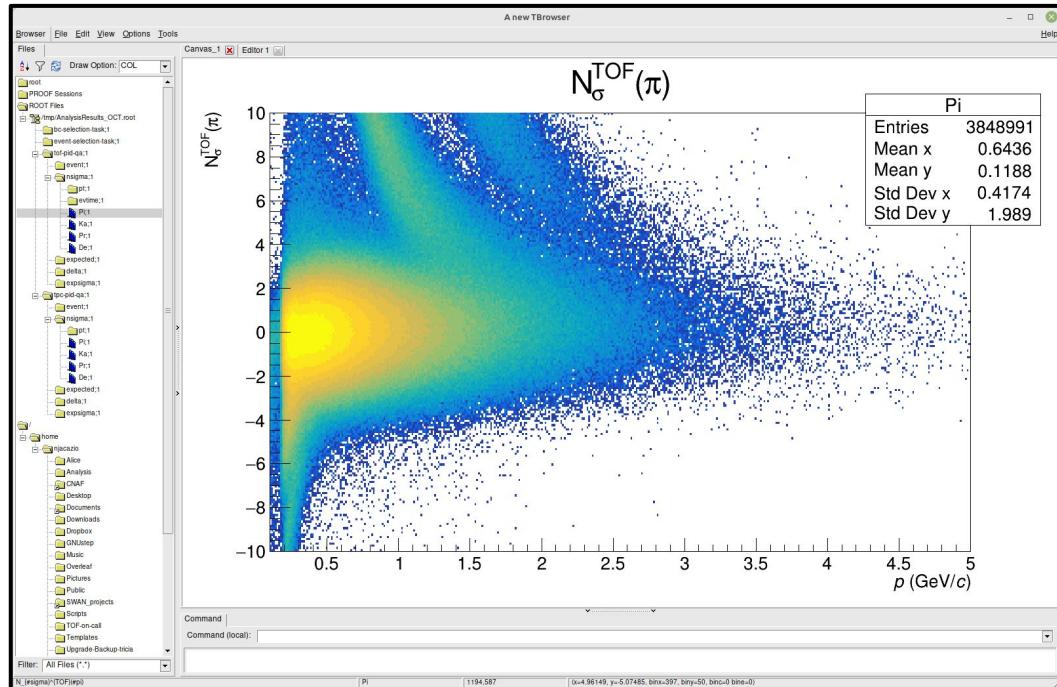
`o2-analysis-pid-tof-qa`

- Monitor No
- Produce nice performance plots e.g. TPC/TOF signals

`o2-analysis-pid-tpc-qa-mc`
`o2-analysis-pid-tpc-qa-sig`

`o2-analysis-pid-tof-qa-evtime`
`o2-analysis-pid-tof-qa-beta`
`o2-analysis-pid-tof-qa-mc`

- QA for MC (e.g. purity) and more (e.g. mass)!



Documentation → <https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html#enabling-qa-histograms>

Setup wagons for PID QA in Hyperloop



- Monitor $N\sigma$
- Include nice performance plots e.g. TPC/TOF signals

AliHyperloop

Train Submission

My Analyses All Analyses Dashboard

Search wagons by name...

Wagon LHC22o_pass7_minBias_small LHC23lb Last run

TOF_BASE

TOF-BETA_QA

TOF-PID

TOF-QA

TPC_PID

TPC-QA

TPC-QA-SIG

+ Add new wagon (or clone wagon from other)

TPC-QA-SIG

Wagon settings Configuration 1 Derived data Test Statistics Grid Statistics

Latest change by hkoley at 14/10/24, 13:32 GMT+5:30

Name TPC-QA-SIG

Work flow name o2-analysis-pid-tpc-qc-sig

Dependencies Core Service Wagons/TrackPropagation

When enabling a wagon, there is no need to enable its dependencies. This is done automatically on train submission.

Train Runs

tpcsignalPos

Entries	5469278
Mean x	0.6663
Mean y	57.89
Std Dev x	0.5206
Std Dev y	29.48

dE/dx Arb. units

tpcsignalNeg

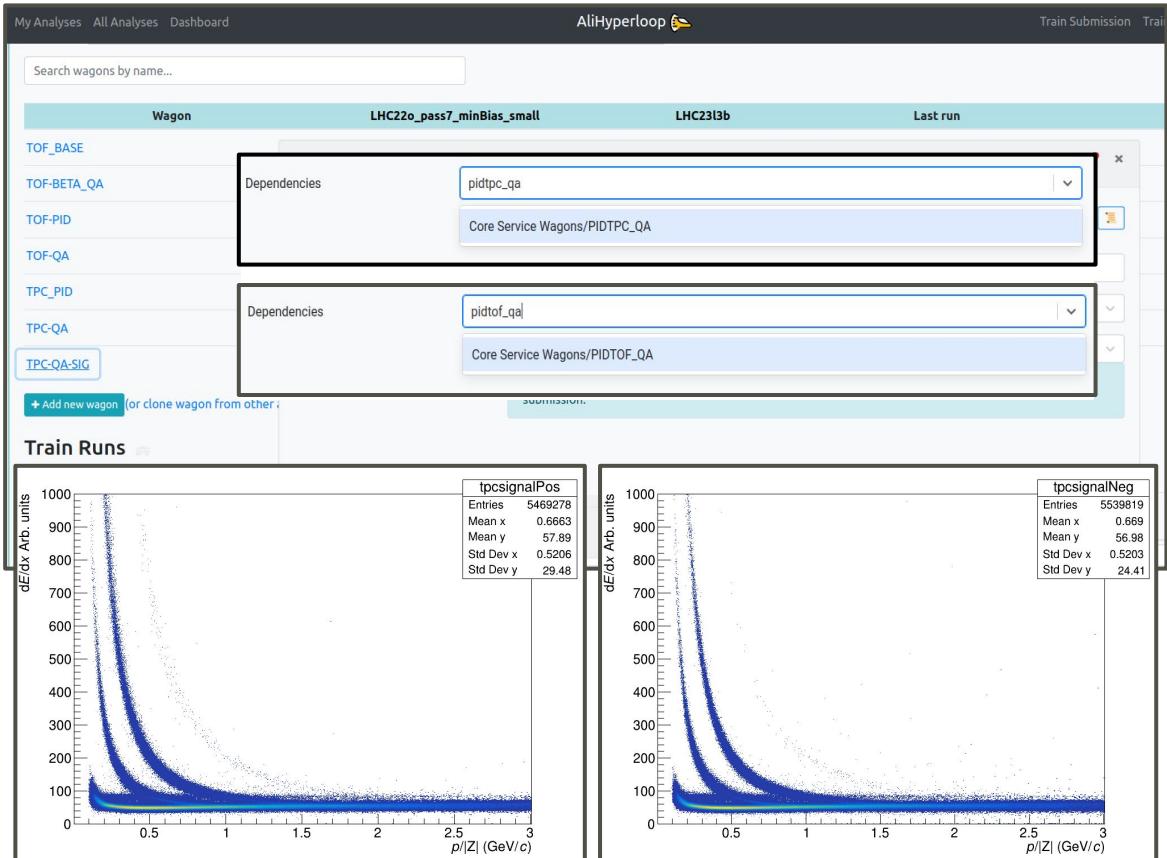
Entries	5539819
Mean x	0.669
Mean y	56.98
Std Dev x	0.5203
Std Dev y	24.41

dE/dx Arb. units

Setup wagons for PID QA in Hyperloop



- Monitor $N\sigma$
- Include nice performance plots e.g. TPC/TOF signals
- QA wagons are also available centrally to be subscribed to instead of duplicated



TOF beta and TOF mass

- TOF specific are the beta and mass

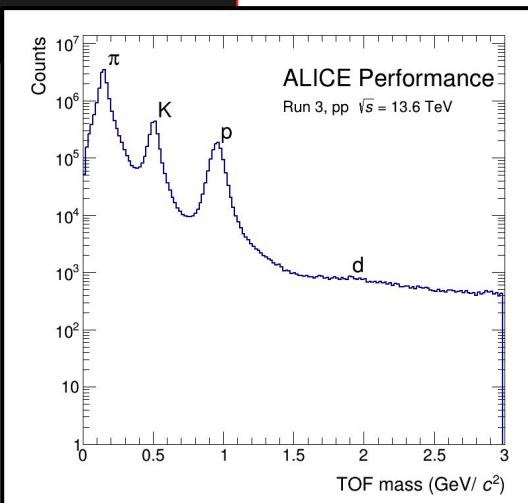
```
void process(soa::Join<aod::Tracks, aod::pidTOFmass, aod::pidTOFBeta> const& tracks)
{
    for (auto& track : tracks) {
        track.beta(); // TOF beta
        track.mass(); // TOF mass
    }
}
```

- Access by the following task: Executable

`o2-analysis-pid-tof-beta`

- Comes with QA as well

`o2-analysis-pid-tof-beta-qa`



Event time for TOF

- Multiple event times can be used for TOF measurement
- With respect to Run2 we now ask the track which event it used

```
void process(soa::Join<aod::Tracks, aod::pidEvTimeFlags> const& tracks)
{
    for (auto& track : tracks) {
        track.isEvTimeTOF(); // TOF Event time
        track.isEvTimeT0AC(); // T0AC Event time
    }
}
```

- Executable that handles event time

`o2-analysis-pid-tof-base`

- Comes with QA as well

`o2-analysis-pid-tof-qa-evttime`

How these values are computed: TPC

- The expected values and expected resolution are computed from the Aleph Bethe Bloch (5 parameters)



```

table(response.GetExpectedSigma(collisions.iteratorAt(trk.collisionId()), trk, pid),
      response.GetNumberOfSigma(collisions.iteratorAt(trk.collisionId()), trk, pid));
  
```

- Fine-tuned neural-network based calibration can be used (see with experts)

```

Configurable<bool> useNetworkCorrection{"useNetworkCorrection", 0, "(bool) Wether or not to use the
network correction for the TPC dE/dx signal"};
  
```

- Refer to the Hyperloop configurations (explained later)

How these values are computed: TOF

- The ingredients for the expected values are already in the AO2D
- Need to compute the expected resolution + corrections



```

table(responsePID.GetExpectedSigma(mRespParamsV2, track),
      responsePID.GetSeparation(mRespParamsV2, track));
  
```

- Fine-tuned pass dependent calibration of the charge bias on momentum + resolution

```

Configurable<bool> loadResponseFromCCDB{"loadResponseFromCCDB", false, "Flag to load the response
from the CCDB"};
  
```

- Refer to the Hyperloop configurations (explained later)

TPC PID helper tasks on Hyperloop

- Dependencies on Hyperloop
 - **Internal dependency**
 - **Run 2**
 - **Run 3**
 - **Neural network corrections**
 - **MC tune on Data**
 - **keep TPC only tracks**
- Include the one according to your needs and in doubt ask!!
- Take them as reference for local running!!

AliHyperloop	Core wagons
PIDTPC	*
PIDTPC_KeepTPCOnlyTracks	*
PIDTPC_MCTuneOnData	*
PIDTPC_MCTuneOnData_KeepTPCOnlyTracks	*
PIDTPC_NN	* *
PIDTPC_NN_KeepTPCOnlyTracks	* *
PIDTPC_NN_MCTuneOnData	* *
PIDTPC_NN_MCTuneOnData_KeepTPCOnlyTracks	* *
PIDTPC_Run2	*
PIDTPC_Run2_MCTuneOnData	*
PIDTPC_Run2_NN	* *
PIDTPC_Run2_NN_MCTuneOnData	* *
PIDTPCBase	*
PIDTPCBase_Run2	*

Description on the specific tasks

For the training, samples of electrons, pions and protons are obtained from a V0 sample,
Kaons from TPC+TOF

Neural network corrections:

- BB curve is not precise enough for a "real life" detector
- Difference between the BB fits and real function is done as a correction using neural networks
- Input to the network: p , $\tan(\lambda)$, $\text{sign}(q)/p_T$, mass hypothesis, normalized multiplicity, normalized number of clusters
- Network learns correction factor (in 6D) to the Bethe-Bloch Parameterization
- Output: dE/dx of identified particles as a ratio to the Bethe-Bloch Parameterization

MCTuneOnData:

- Expected BB fits and sigmas are used to sample the dE/dx signal in MC

TOF PID helper tasks on Hyperloop



- Dependencies on Hyperloop
 - **Internal dependency (not to be included)**
 - **Run 2**
 - **Run 3**
- Include the one according to your needs and in doubt ask the experts!!
- Take them as reference for local running!!

AliHyperloop	Core wagons
PIDTOFBaseRun3	*
PIDTOFBaseRun3_PbPb	*
PIDTOFBetaRun3	*
PIDTOFBetaRun3_PbPb	*
PIDTOFFullRun3	*
PIDTOFFullRun3_PbPb	*
PIDTOFGeneralPurpose_Run2	*
PIDTOFGeneralPurpose_Run3	*
PIDTOFRun3	*
PIDTOFRun3_PbPb	*

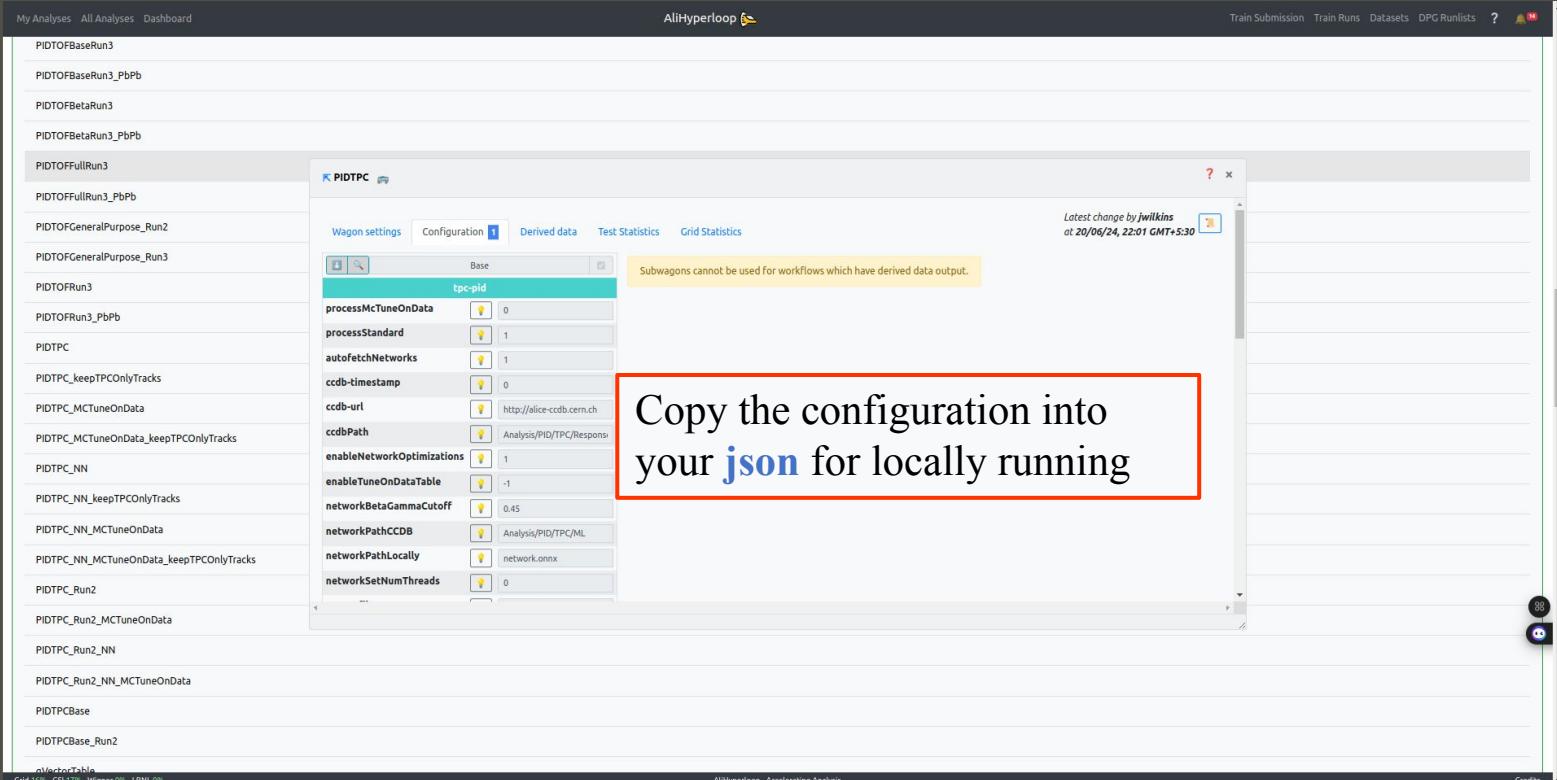
TOF PID helper tasks on Hyperloop

- Dependencies on Hyperloop
 - **Internal dependency (not to be included)**
 - **Run 2**
 - **Run 3**
- Include the one according to your needs and in doubt ask the experts!!
- Take them as reference for local running!!
- Only general purpose tasks are needed (no other TOF dependencies). Setup of collision system is automatic.
Other tasks will be deprecated soon.

Self sufficient ←

AliHyperloop 🎤		Core wagons
PIDTOFBaseRun3	*	TOF signal
PIDTOFBaseRun3_PbPb	*	Ev. time
PIDTOFBetaRun3	*	TOF beta
PIDTOFBetaRun3_PbPb	*	TOF mass
PIDTOFFullRun3	*	
PIDTOFFullRun3_PbPb	*	
PIDTOFGeneralPurpose_Run2	*	General Purpose
PIDTOFGeneralPurpose_Run3	*	
PIDTOFRun3	*	
PIDTOFRun3_PbPb	*	

Taking Hyperloop as a reference



The screenshot shows the ALICE Hyperloop analysis interface. On the left, a sidebar lists various analysis configurations. The main area is a configuration editor for 'PIDTPC' under 'Wagon settings'. A yellow box highlights a note: 'Subwagons cannot be used for workflows which have derived data output.' A timestamp indicates the latest change was made by jwilkins at 20/06/24, 22:01 GMT+5:30. A red box highlights the text: 'Copy the configuration into your json for locally running'.

Copy the configuration into your **json** for locally running

Setting	Value
processMcTuneOnData	0
processStandard	1
autoFetchNetworks	1
ccdb-timestamp	0
ccdb-url	http://alice-ccdb.cern.ch
ccdbPath	Analysis/PID/TPC/Responses
enableNetworkOptimizations	1
enableTuneOnDataTable	-1
networkBetaGammaCutoff	0.45
networkPathCCDB	Analysis/PID/TPC/ML
networkPathLocally	network.onnx
networkSetNumThreads	0

- Inspect the task configuration that suits your case, in case of doubt ask !!

Where to orient yourself and report your findings



Plethora of occasions to report your (good or bad) findings

- Async QC Weekly meeting (Tuesdays 15h00) → <https://indico.cern.ch/event/1464688/>
- DPG AOT-track (Thursdays 9h30) → <https://indico.cern.ch/event/1464946/>
- Detector and QA experts
- **TPC** → ana.marin@cern.ch
- **TOF** → nicolo.jacazio@cern.ch
- **PID QA** → banajit.barman@cern.ch + DPG-AOT coord.

Suggestions are welcome, eager to improve!

Get updated

- Mattermost channel → <https://mattermost.web.cern.ch/alice/channels/o2-analysis>
- Subscribe to announcements:
<https://mattermost.web.cern.ch/alice/channels/o2-analysis-announcements>



Nicolo Jacazio 12:02 PM

Commented on Nicolo Jacazio's message: Hi @channel in the effort of streamlining the dependencies on the HL wagon chains we merged the Run2 vs R...
Hi @all we will proceed to further remove deprecated wagons. This time mostly for the Run2 MC, however verify as suggested by Jan Flete that you are not using deprecated service wagons in your analysis. Thanks!

Nicolo Jacazio 1:14 PM

For reference, here are the wagons to be removed

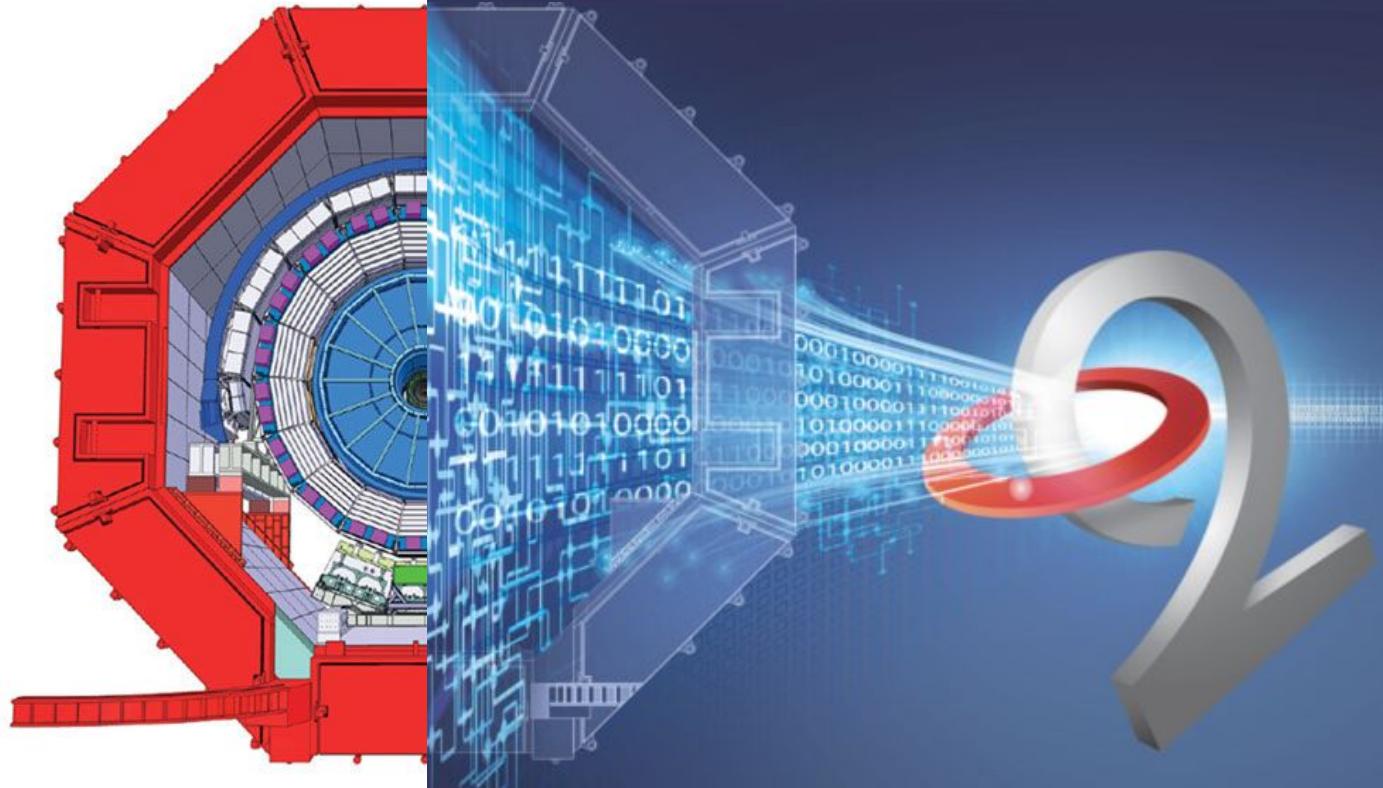
Wagon
deprecated
PIDTOFBaseRun2_MC_deprecated
PIDTOFBetaRun2_MC_deprecated
PIDTOFFullRun2_MC_deprecated
PIDTOFGeneralPurpose_Run2_MC_deprecated
PIDTOFRun2_MC_deprecated
TrackExtension_Run2_MC_deprecated
TrackSelection_Run2_MC_deprecated
+ Add new wagon (or clone wagon from other analysis)

Nicolo Jacazio 11:49 AM

Hi @channel the saga of simplifying deprecated core wagons continues, now that the timestamp between run2 and run3 was merged we can further merge the workflows. Also in the event selection there is no difference between pp and Pb-Pb. In the TOF there is a single Run2 workflow needed now. Please check if you are using the wagons in the list, they will be removed starting from next Monday. Any feedback is welcome, in case you need directions let us know! @Evgeny Kryshen

depreec
Centrality_Run2_MC_deprecated
EventSelection_Run2_pp_deprecated
EventSelection_Run2_pp_MC_deprecated
EventSelection_Run2_pp_p_deprecated
EventSelection_Run2_PbPb_Data_and_MC_deprecated
EventSelection_Run2_pp_deprecated
PIDTOFBaseRun2_deprecated
PIDTOFBetaRun2_deprecated
PIDTOFFullRun2_deprecated
PIDTOFGeneralPurpose_Run2_pp_deprecated
PIDTOFRun2_deprecated
TimestampCreator_Run2_deprecated

Issues, developments, and calibration releases are announced here (for PID but not only!)



Thank you for your attention