



ALICE



Jadavpur University

PID: Electrons, Hadrons, and Light Nuclei

Hirak Kumar Koley¹, Nicolo Jacazio²

¹Jadavpur University, ²UPO & INFN

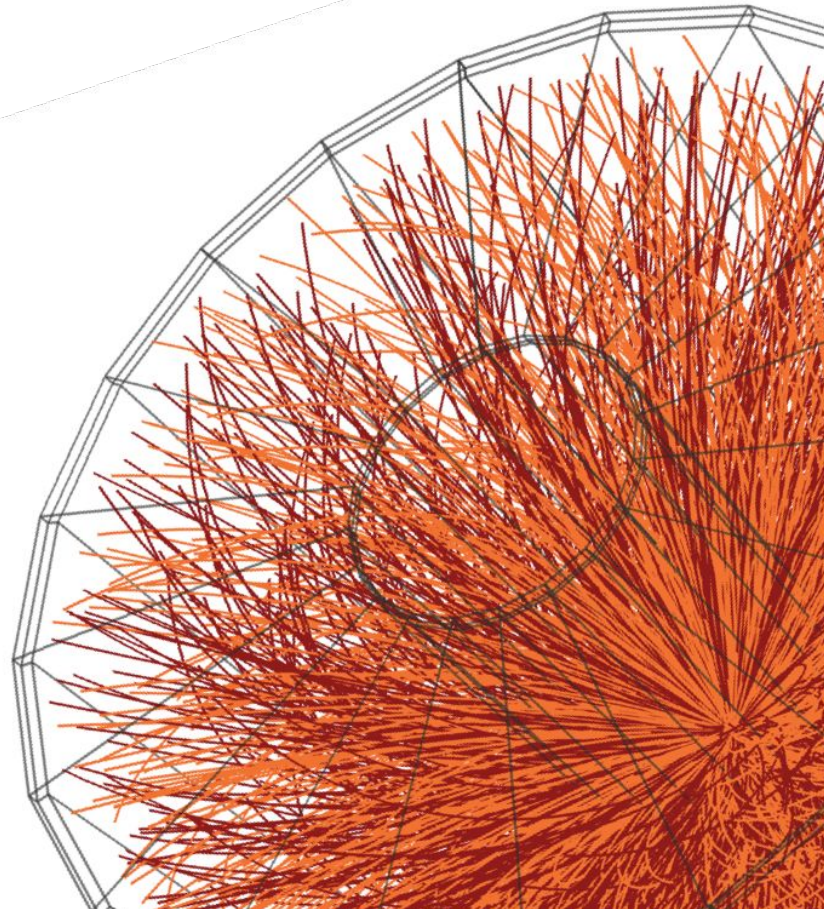
hirak.koley@cern.ch

nicolo.jacazio@cern.ch

O² Analysis Tutorial v5.0

[*https://indico.cern.ch/event/1574136/*](https://indico.cern.ch/event/1574136/)

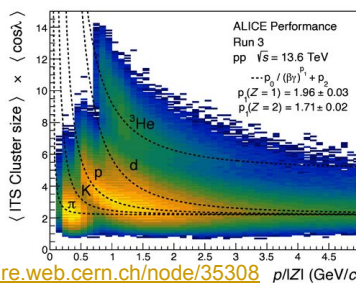
11th November 2025



Goal of this session

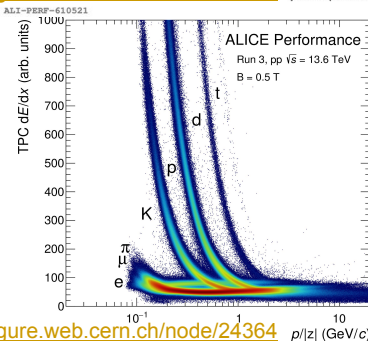
Get you acquainted with:

- What is **P**article **I**dentification (**PID**) ?
- **D**etectors that can be used for PID
- **S**oftware tools for PID in **O**²**P**hysics
- Work with **a**dditional helper tasks
- Use detector responses to perform **PID** in **a**nalysis
- **A**ssess **q**uality of particle identification with helper tasks
- Produce **b**eautiful **p**lots for PID performance
- Get help if things don't work as expected



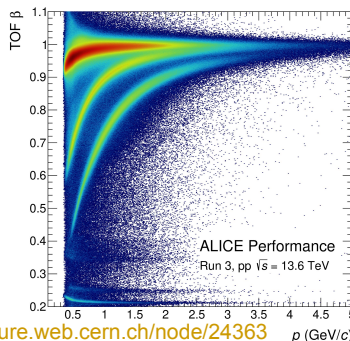
<https://alice-figure.web.cern.ch/node/35308>

ITS



<https://alice-figure.web.cern.ch/node/24364>

TPC

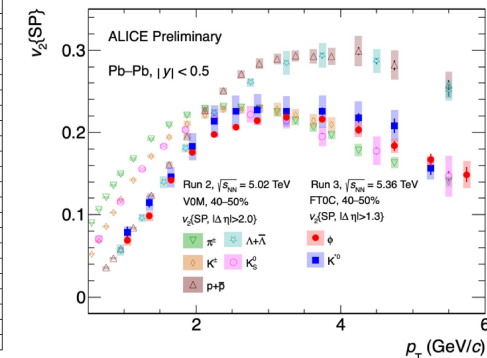
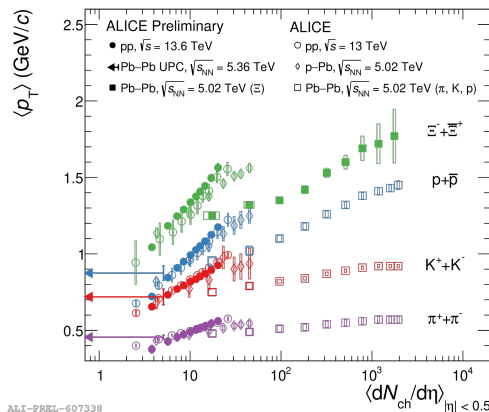


<https://alice-figure.web.cern.ch/node/24363>

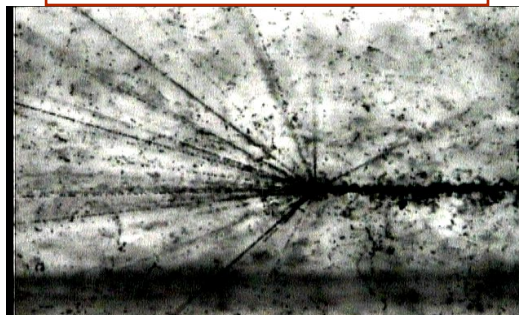
TOF

Particle Identification

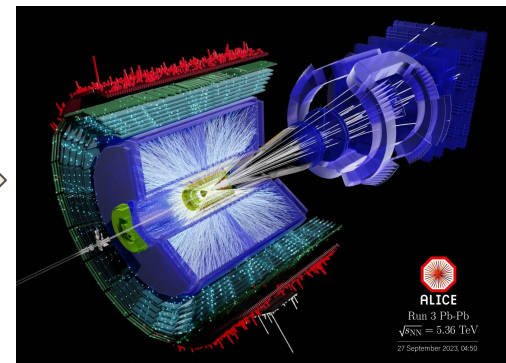
- From the thousands of particle crowd, we need to identify particles
- Introducing **PID**, we get wealth of information and new insights i.e. more information towards our goal



Central Kr-Ag/Br event at 950 MeV/A



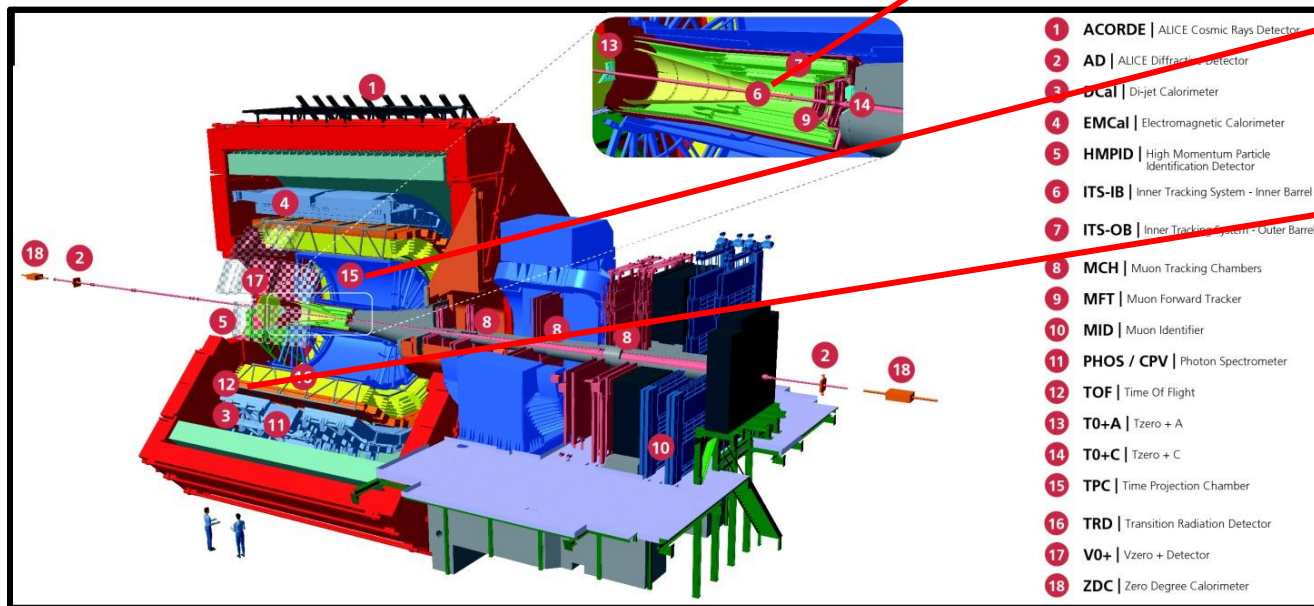
Few particles only



Thousands of particles



ALICE PID in Run 3



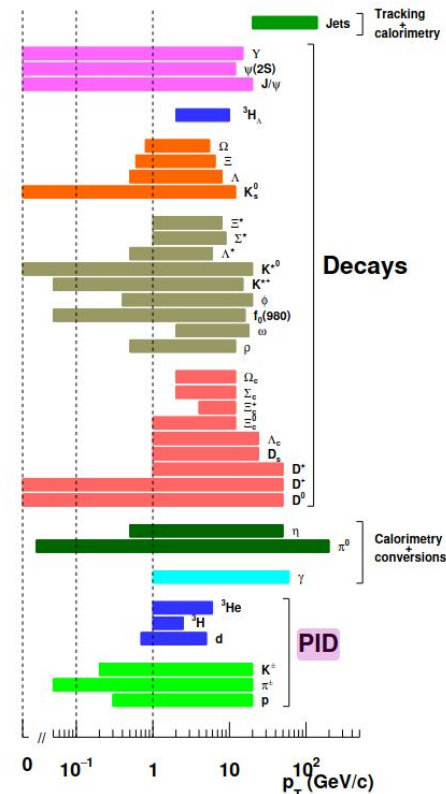
ITS: energy loss
proportional to
Cluster size

TPC: energy loss
value given by
Bethe-Bloch

TOF: particle **flight**
time from collision
vertex to TOF

How to do PID:

- **Unstable particles:** Invariant mass reconstruction / topological decay
- **Calorimetry:** full energy deposition, e-photon-hadron discrimination from shower profile
- **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, $\cos(\theta) = 1/n\beta$



<https://arxiv.org/abs/2211.04384>

PID using "n σ -cut" method

ITS/TPC

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

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

- 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

TOF

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

$$n\sigma = \frac{t_{\text{measured}} - t_{\text{expected}}}{\sigma_{PID(TOF)}} \quad t_{\text{measured}} = t_{\text{hit}} - \text{start}_{\text{Time}}$$

- t_{hit} : particle arrival time measured in the TOF detector
- $\text{start}_{\text{Time}}$: 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 $\rightarrow$ depends on the "T0" 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/1574136/timetable/#1-introduction-to-o2o2physics>

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)

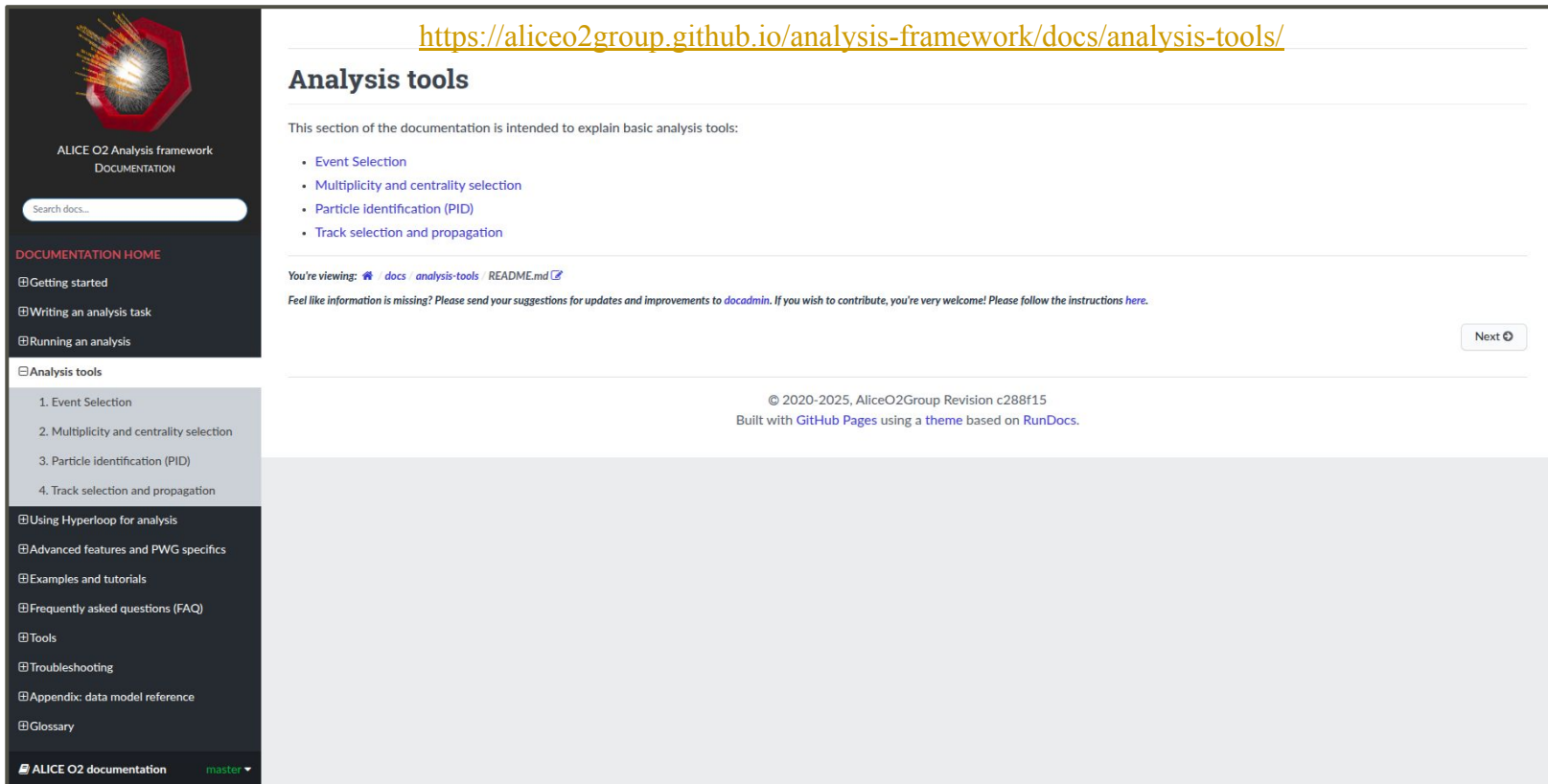
```
o2-analysis-a --configuration json://file.json |\n o2-analysis-b --configuration json://file.json |\n o2-analysis-c --configuration json://file.json --other-option
```

→ More in the hands-on!

- Extended info (e.g. $N\sigma$) is not directly available in the AO2D
→ **computed on the fly**
How?
- New headers will need to be included to extend the subscribed data model
- New PID tasks will be needed in pipe to compute additional information

A guide to the helper tasks

- Advanced analyses need more helper tasks



<https://aliceo2group.github.io/analysis-framework/docs/analysis-tools/>

Analysis tools

This section of the documentation is intended to explain basic analysis tools:

- [Event Selection](#)
- [Multiplicity and centrality selection](#)
- [Particle Identification \(PID\)](#)
- [Track selection and propagation](#)

You're viewing: [docs](#) / [analysis-tools](#) / [README.md](#)

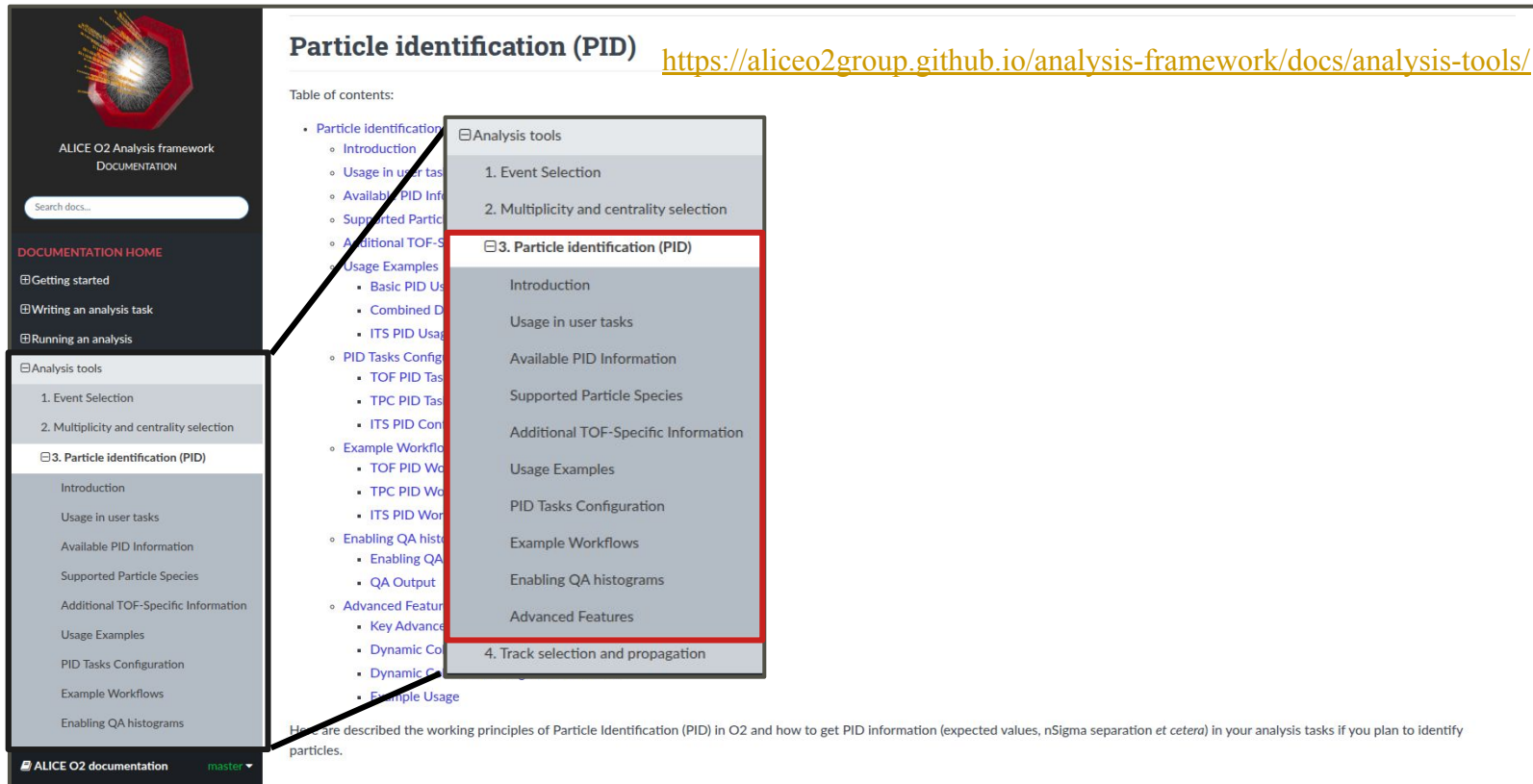
Feel like information is missing? Please send your suggestions for updates and improvements to [docadmin](#). If you wish to contribute, you're very welcome! Please follow the instructions [here](#).

Next

© 2020-2025, AliceO2Group Revision c288f15
Built with [GitHub Pages](#) using a [theme](#) based on [RunDocs](#).

A guide to the helper tasks

- Advanced analyses need more helper tasks



Particle identification (PID) <https://aliceo2group.github.io/analysis-framework/docs/analysis-tools/>

Table of contents:

- Particle identification
 - Introduction
 - Usage in user tasks
 - Available PID Information
 - Supported Particle Species
 - Additional TOF-Specific Information
 - Usage Examples
 - Basic PID Usage
 - Combined D
 - ITS PID Usage
 - PID Tasks Configuration
 - TOF PID Task
 - TPC PID Task
 - ITS PID Configuration
 - Example Workflows
 - TOF PID Workflow
 - TPC PID Workflow
 - ITS PID Workflow
 - Enabling QA histograms
 - Enabling QA
 - QA Output
 - Advanced Features
 - Key Advancements
 - Dynamic Configuration
 - Dynamic C
 - Example Usage

Analysis tools

1. Event Selection
2. Multiplicity and centrality selection
3. Particle identification (PID)
4. Track selection and propagation

Introduction

Usage in user tasks

Available PID Information

Supported Particle Species

Additional TOF-Specific Information

Usage Examples

PID Tasks Configuration

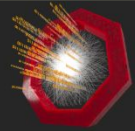
Example Workflows

Enabling QA histograms

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.

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
- Analysis tools
 - 1. Event Selection
 - 2. Multiplicity and centrality selection
 - 3. Particle identification (PID)
 - Introduction
 - Usage in user tasks
 - Available PID Information
 - Supported Particle Species
 - Additional TOF-Specific Information
 - Usage Examples
 - PID Tasks Configuration
 - Example Workflows
 - Enabling QA histograms

ALICE O2 documentation master

Particle identification (PID)

Table of contents:

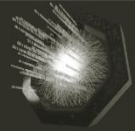
- Particle identification (PID)
 - Introduction
 - Usage in user tasks
 - Available PID Information
 - Supported Particle Species
 - Additional TOF-Specific Information
 - Usage Examples
 - Basic PID Usage
 - Combined Detector PID
 - ITS PID Usage
 - PID Tasks Configuration
 - TOF PID Task
 - TPC PID Tasks
 - ITS PID Configuration
 - Example Workflows
 - TOF PID Workflow
 - TPC PID Workflow
 - ITS PID Workflow
 - Enabling QA histograms
 - Enabling QA Tasks
 - QA Output
 - Advanced Features
 - Key Advanced Features:
 - Dynamic Columns for TOF Beta and Mass
 - Dynamic Columns for nSigma Calculations
 - Example Usage

<https://aliceo2group.github.io/analysis-framework/docs/analysis-tools/PID.html>

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.

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
- Analysis tools
 - 1. Event Selection
 - 2. Multiplicity and centrality selection
 - 3. Particle identification (PID)
 - Introduction
 - Usage in user tasks
 - Available PID Information
 - Supported Particle Species
 - Additional TOF-Specific Information
 - Usage Examples
 - PID Tasks Configuration
 - Example Workflows
 - Enabling QA histograms

ALICE O2 documentation master

Particle identification (PID)

Table of contents:

- Particle Identification (PID)
 - Introduction
 - Usage in user tasks
 - Available PID Information
 - Supported Particle Species
 - Additional TOF-Specific Information
 - Usage Examples
 - Basic PID Usage
 - Combined Detector PID
 - ITS PID Usage
 - PID Tasks Configuration
 - TOF PID Task
 - TPC PID Tasks
 - ITS PID Configuration
 - Example Workflows
 - TOF PID Workflow
 - TPC PID Workflow
 - ITS PID Workflow
 - Enabling QA histograms
 - Enabling QA Tasks
 - QA Output
 - Advanced Features
 - Key Advanced Features:
 - Dynamic Columns for TOF Beta and Mass
 - Dynamic Columns for nSigma Calculations
 - Example Usage

<https://aliceo2group.github.io/analysis-framework/docs/analysis-tools/PID.html>

You're invited to contact experts to improve the docs



LICENSE TO COMPLAIN

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

 Name _____
E-Mail _____
Signature _____

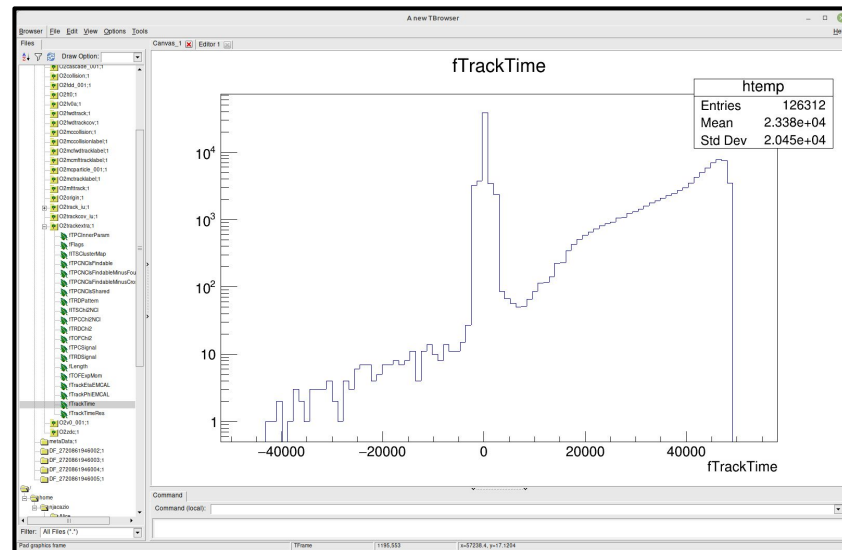
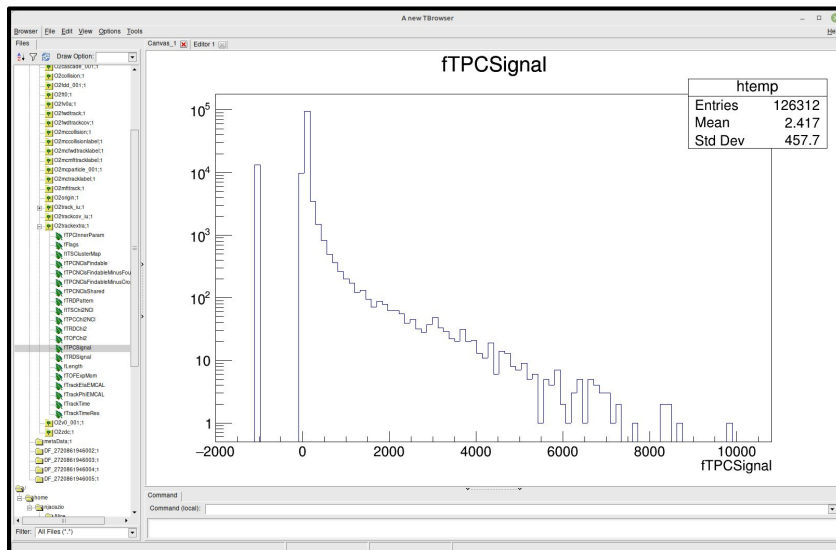
U.S. Dept. of annoyance [Sillyhumor.com](https://www.sillyhumor.com)

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.

Example: select pions

AO2D.root

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



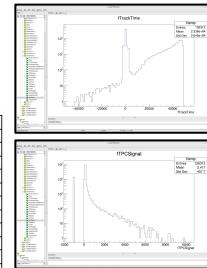
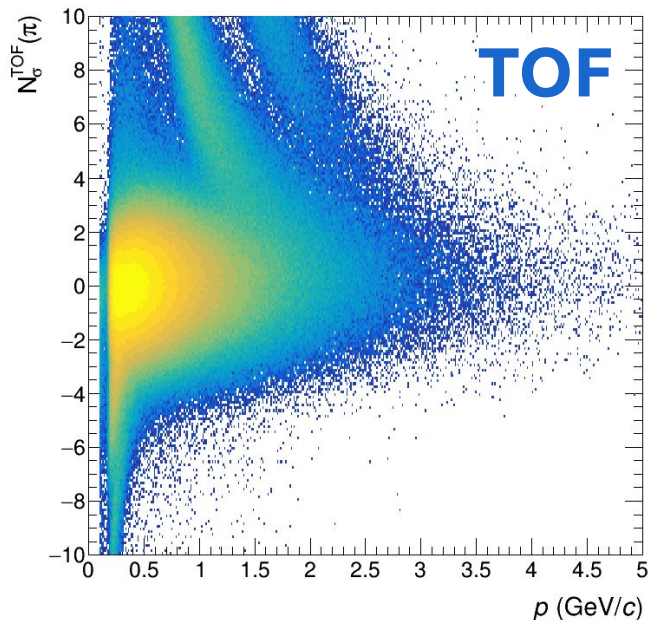
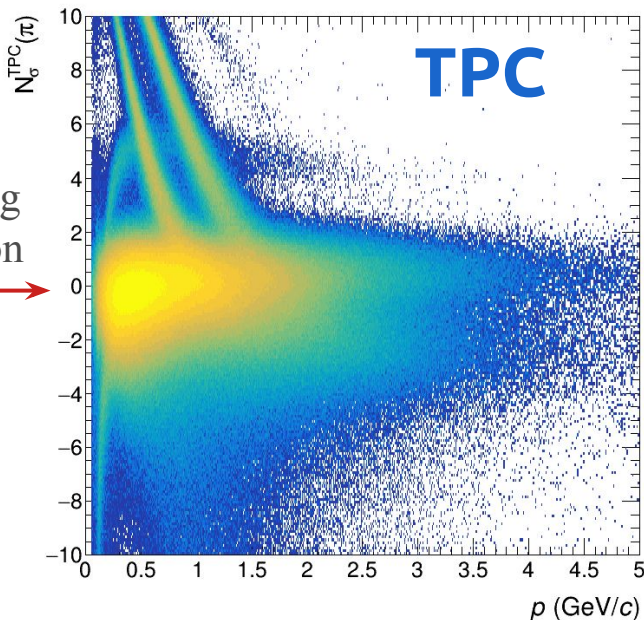
AnalysisResults.root

Example: select pions

AO2D.root

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

Processing: building
pion N_σ information



AnalysisResults.root

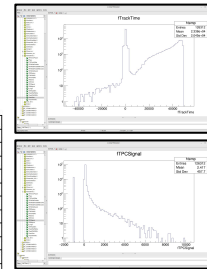
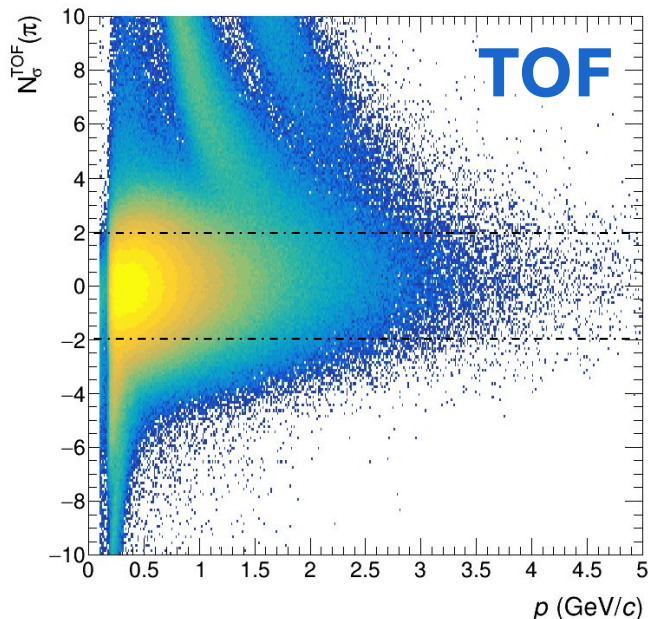
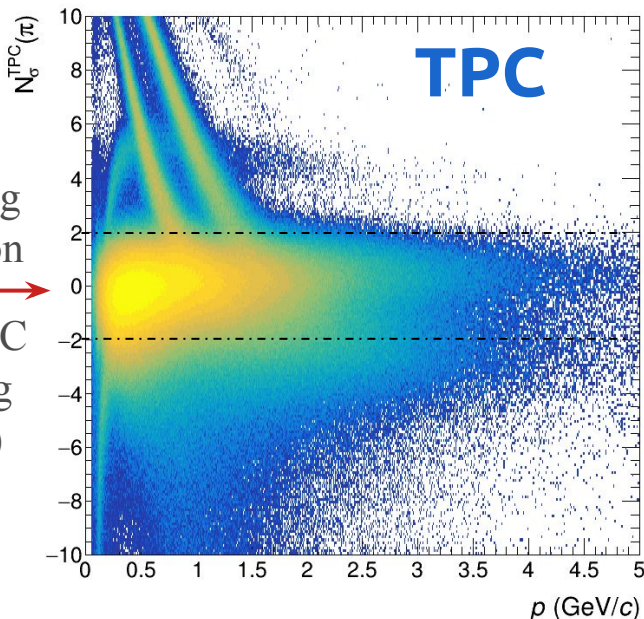
Example: select pions

AO2D.root

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

Processing: building
pion N_σ information

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



AnalysisResults.root

Filter & Partitions: <https://indico.cern.ch/event/1574136/timetable/#1-introduction-to-o2o2physics>

Example: select pions

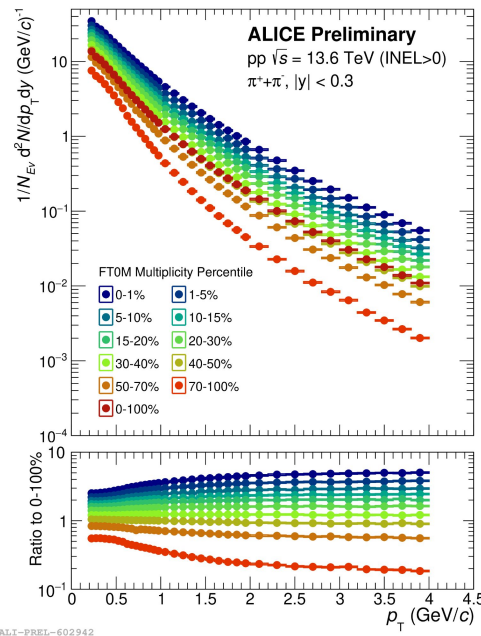
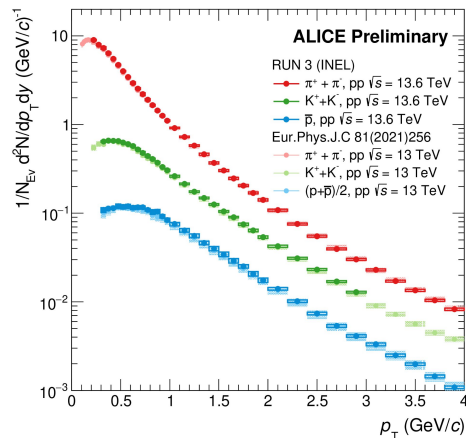
AO2D.root

“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



Your physics results, OK but how?

Filter & Partitions: <https://indico.cern.ch/event/1574136/timetable/#1-introduction-to-o2o2physics>

Example: select pions

AO2D.root

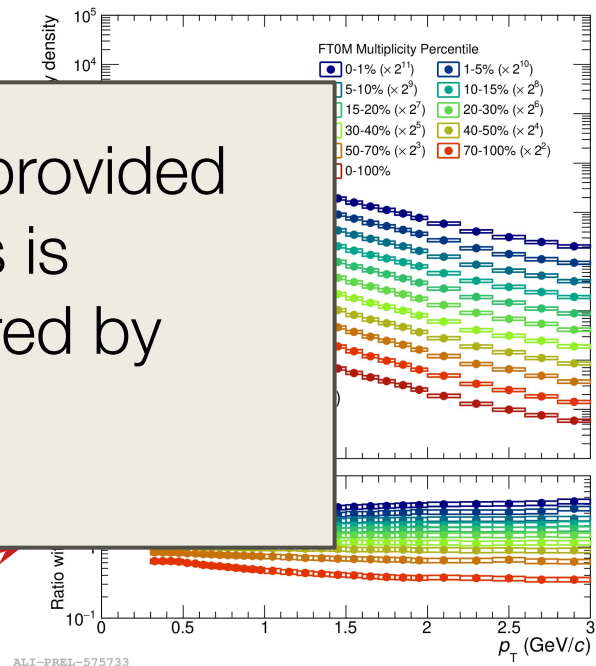
“Raw” detector data: $1/p_T$, TPCSignal, trackTime, ...

Processing: building
pion $N\sigma$ information

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

AnalysisResults.root

PID performance provided
by central systems is
constantly monitored by
DPG experts!



Your physics results, *OK but how?*

Filter & Partitions: <https://indico.cern.ch/event/1574136/timetable/#1-introduction-to-o2o2physics>

Including PID data formats

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

```
#include "Common/DataModel/PIDResponseITS.h"  
#include "Common/DataModel/PIDResponseTPC.h"  
#include "Common/DataModel/PIDResponseTOF.h"
```

- ITS nSigma values
- TPC nSigma values
- TOF nSigma values

...

- Tables available for PID

- Expected values (e.g. Bethe Bloch, expected TOF)
- Expected resolutions
- Separation (i.e. measured signal - expected signal)
- $N\sigma$ separation
- Extra PID information e.g. TOF mass, TOF beta, ...

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

Available for all particle
mass hypotheses from
electrons to α

Subscribing to PID tables

```
void process(soa::Join
```

Pion PID

TPC

TOF

```
void process(soa::Join
```

Kaon PID

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

```
void process(soa::Join
```

**Pion and
Kaon PID**

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

Accessing PID info for TPC and TOF

- 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/analysis-tools/PID.html#usage-in-user-tasks>

TPC/TOF PID tasks in the workflow

- Tasks to include in the workflow

```
o2-analysis-pid-tpc-service | o2-analysis-pid-tpc-base |  
o2-analysis-track-propagation | o2-analysis-timestamp --aod-file AO2D.root
```

PID Tasks

Dependencies

- Here, we compute the N-Sigmas for each particle.
- Need to subscribe for specific particle in your task.

```
o2-analysis-pid-tof-merge | o2-analysis-track-propagation |  
o2-analysis-event-selection | o2-analysis-timestamp | o2-analysis-ft0-corrected-table  
--aod-file AO2D.root
```

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

Accessing PID info for ITS

- ITS information is only dynamic (computed on the fly).
- Parametrization is taken from the **pidITS.cxx** task with the workflow *o2-analysis-pid-its*

```
void init(o2::framework::InitContext& context)
{
    o2::aod::ITSResponse::setParameters(context); // Load Parameters from pidITS.cxx task
}

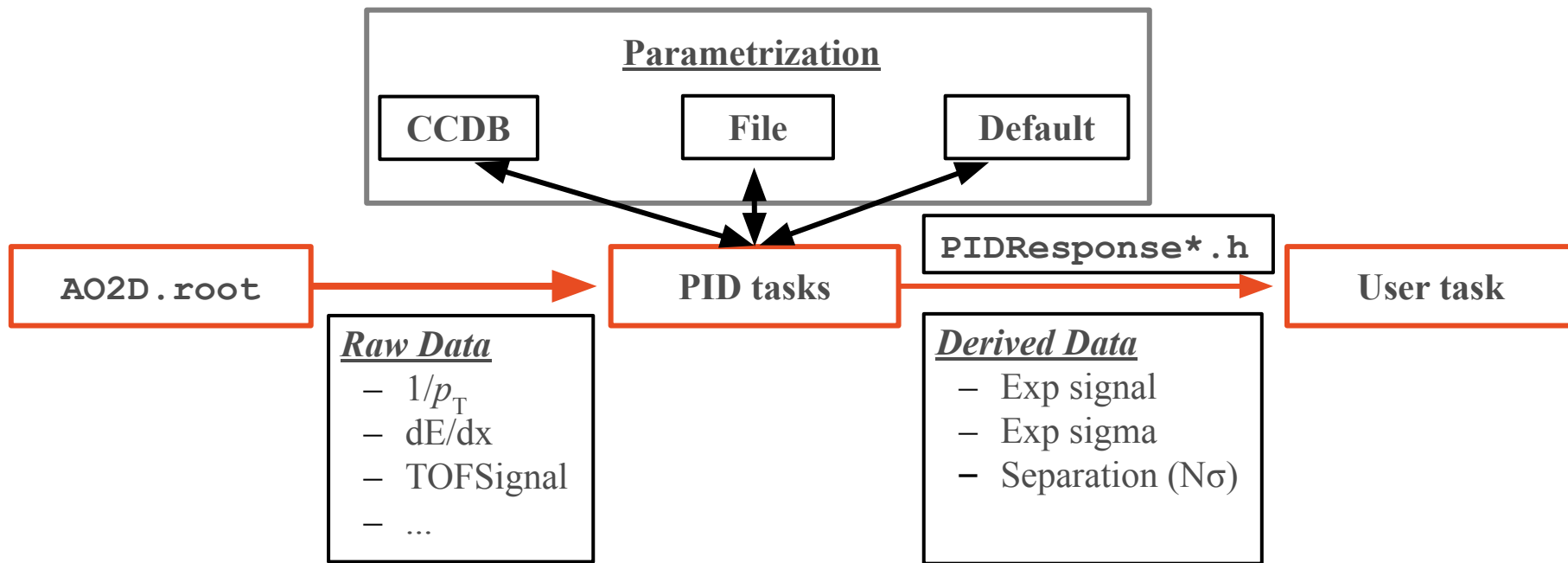
void process(aod::Tracks const& tracks)
{
    auto tracksWithPid = soa::Attach<aod::Tracks,
                                    aod::pidits::ITSNSigmaPi,
                                    aod::pidits::ITSNSigmaKa>(tracks);

    for (auto& track : tracksWithPid) {
        track.itsNSigmaPi(); // ITS Nsigma of pions
        track.itsNSigmaKa(); // ITS Nsigma of kaons
    }
}
```

Documentation → <https://aliceo2group.github.io/analysis-framework/docs/analysis-tools/PID.html#its-pid-usage>

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

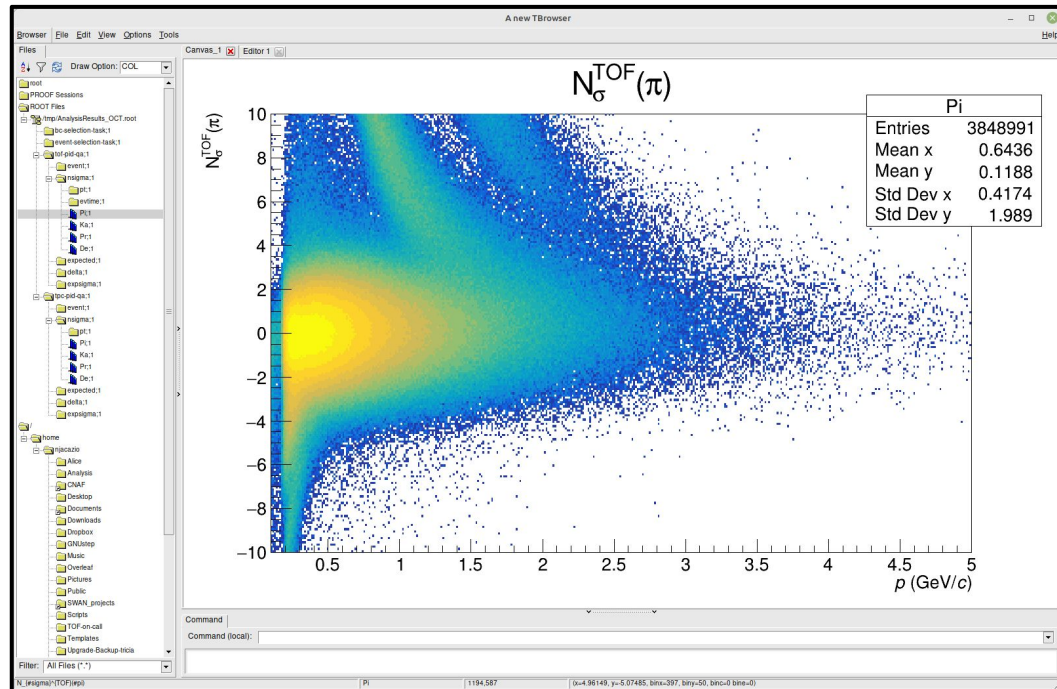
Include PID QA output!

- PID tasks come with QA output included!

```
o2-analysis-pid-its-qa  
o2-analysis-pid-tpc-qa  
o2-analysis-pid-tof-qa
```

- Monitor N_{σ}
- Produce nice performance plots e.g. ITS/TPC/TOF signals

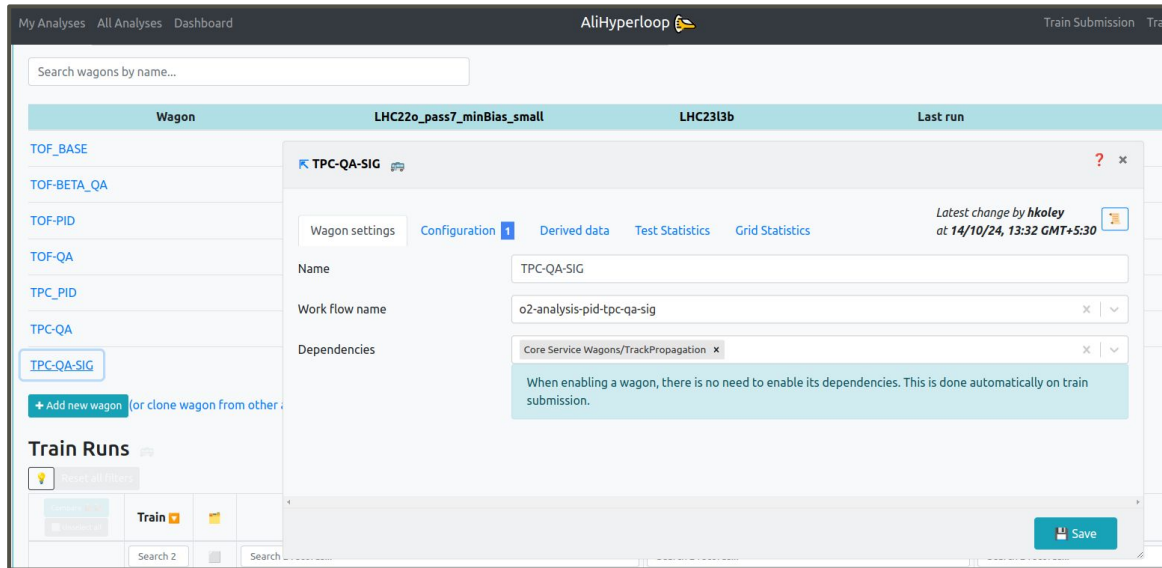
```
o2-analysis-pid-tpc-qa-mc  
o2-analysis-pid-tpc-qa-sig  
  
o2-analysis-pid-tof-qa-evtime  
o2-analysis-pid-tof-qa-beta  
o2-analysis-pid-tof-qa-mc
```



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

Setup wagons for PID QA in Hyperloop

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

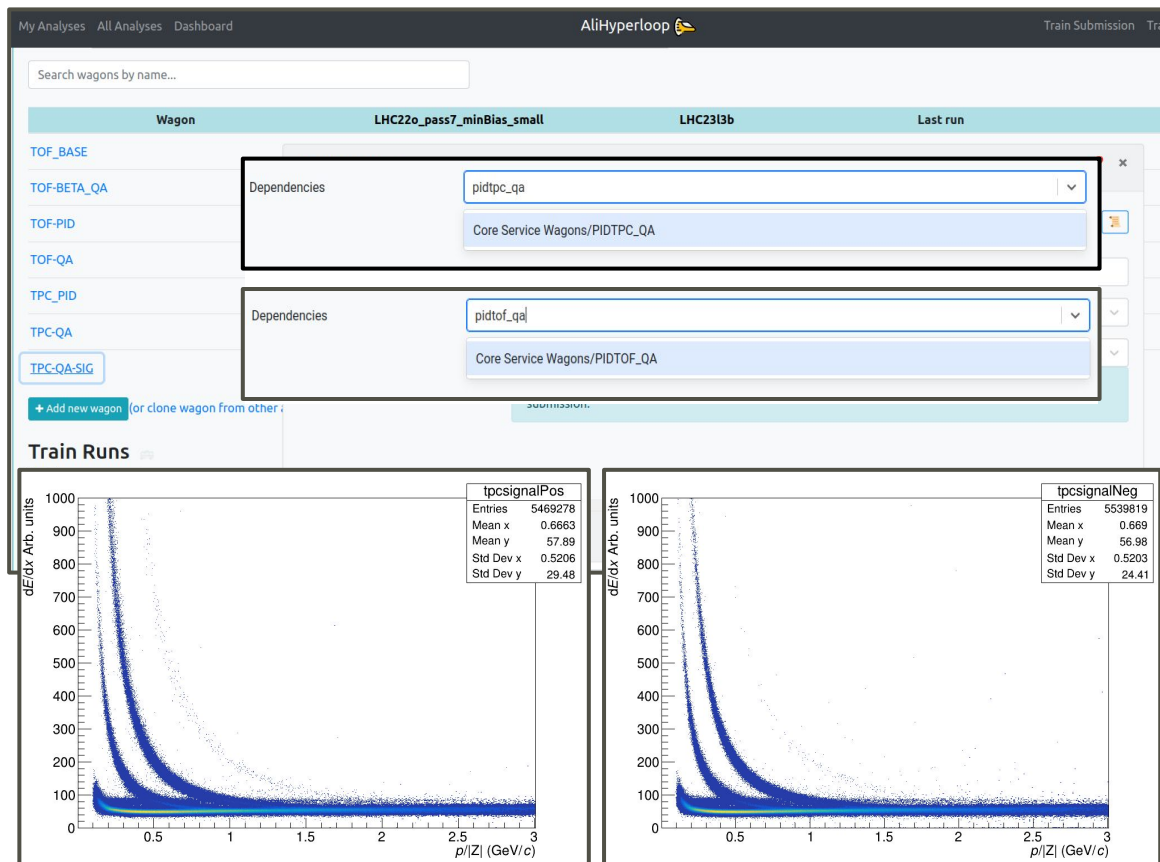


The screenshot displays the AliHyperloop web interface. At the top, there's a navigation bar with 'My Analyses', 'All Analyses', and 'Dashboard'. The main header shows 'AliHyperloop' and 'Train Submission'. A search bar for wagons is at the top left. Below it, a table lists existing wagons: 'LHC22o_pass7_minBias_small' and 'LHC23l3b'. On the left sidebar, a list of analysis types includes TOF_BASE, TOF-BETA_QA, TOF-PID, TOF-QA, TPC_PID, TPC-QA, and TPC-QA-SIG (which is highlighted). A modal window titled 'TPC-QA-SIG' is open, showing configuration details. It includes tabs for 'Wagon settings', 'Configuration' (active), 'Derived data', 'Test Statistics', and 'Grid Statistics'. The configuration fields show: Name: 'TPC-QA-SIG', Work flow name: 'o2-analysis-pid-tpc-qa-sig', and Dependencies: 'Core Service Wagons/TrackPropagation'. A note states: 'When enabling a wagon, there is no need to enable its dependencies. This is done automatically on train submission.' The 'Latest change by hkoley at 14/10/24, 13:32 GMT+5:30' is noted. At the bottom right of the modal is a 'Save' button. Below the modal, there's a 'Train Runs' section with a 'Train' button and search filters.

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

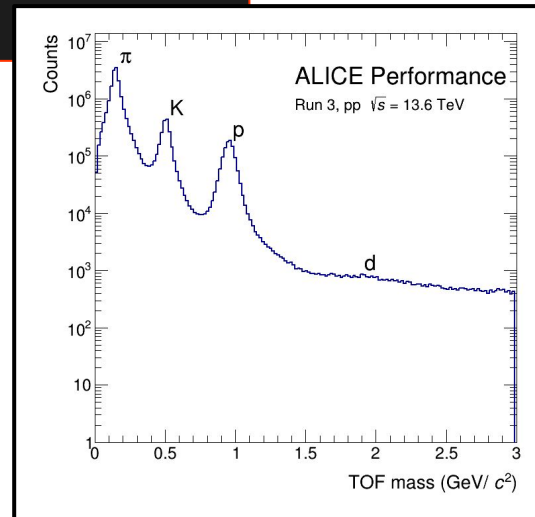
```
void process(soa::Join<aod::Tracks, aod::pidTOFmass, aod::pidTOFbeta> const& tracks)
{
    for (auto& track : tracks) {
        track.beta(); // TOF beta
        track.mass(); // TOF mass
    }
}
```

- Access by the following task: Executable

```
o2-analysis-pid-tof-merge
```

- Comes with QA as well

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



Event time for TOF

- Multiple event times can be used for TOF measurement
- With respect to Run2, we now ask the track which event time 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
        track.tofEvTime(); // Event Collision time used for the track
    }
}
```

- Executable that handles event time

```
o2-analysis-pid-tof-merge
```

- Comes with QA as well

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

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) - recommended

```
Configurable<bool> useNetworkCorrection{"useNetworkCorrection", 0, "(bool) Weather 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));
```

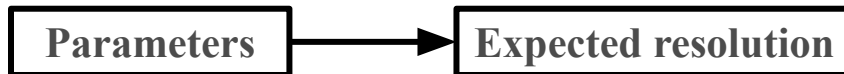
- Fine-tuned pass dependent calibration of the charge bias on momentum + resolution

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

- Refer to the Hyperloop configurations (explained later)

How these values are computed: ITS

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



```
DECLARE_SOA_DYNAMIC_COLUMN (ITSNSigmaElImp, itsNSigmaEl, //!< Nsigma separation with the ITS detector for electrons
    []( uint32_t itsClusterSizes, float momentum, float eta) -> float {
        return ITSResponse::nSigmaITS<o2::track::PID::Electron>(itsClusterSizes, momentum,
eta);
    });
```

- ITS PID is provided with dynamic columns only. Parametrization for MC and Data is slightly different and has to be provided to the dynamic column before being able to provide a calibrated signal. Resolution parametrization is taken from *o2-analysis-pid-its*
- **N.B.** it is fundamental to perform the `o2::aod::ITSResponse::setParameters(context);` call in your init function!

ITS PID helper tasks on Hyperloop

- Dependencies on Hyperloop
 - **Run 3**
- in doubt ? ask!!
- Take them as reference for local running!!



TPC PID helper tasks on Hyperloop

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

AliHyperloop 	Core wagons		
PIDTPCService	* *		
PIDTPCService-MonteCarloTuneOnData	* *		*
PIDTPCService-NeuralNetwork	*	*	
PIDTPCService-NeuralNetwork-MonteCarloTuneOnData	*	*	*
PIDTPCService-NeuralNetwork-MonteCarloTuneOnData-Run2	*	*	*
PIDTPCService-NeuralNetwork-Run2	*	*	

Description on the specific tasks

Neural network corrections:

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

- BB curve is not precise enough for a "real life" detector
- Difference between the Bethe-Bloch fits, and real function is done as a correction using neural networks
- Network learns correction factor (in 7D) to the Bethe-Bloch Parameterization
- **Input to the network:** p , $\tan(\lambda)$, $\text{sign}(q)/p_T$, mass hypothesis, normalized multiplicity, normalized number of clusters, FT0 Occupancy
- **Output:** dE/dx of identified particles as a ratio to the Bethe-Bloch Parameterization

MCTuneOnData:

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

TOF PID helper tasks on Hyperloop

- Dependencies on Hyperloop
 - **Run 2**
 - **Run 3**
 - **Data**
 - **MC**
- Include the one according to your needs and in doubt ask the experts!!

AliHyperloop 

Core wagons

PIDTOFGeneralPurpose_Run2



PIDTOFGeneralPurpose_Run3

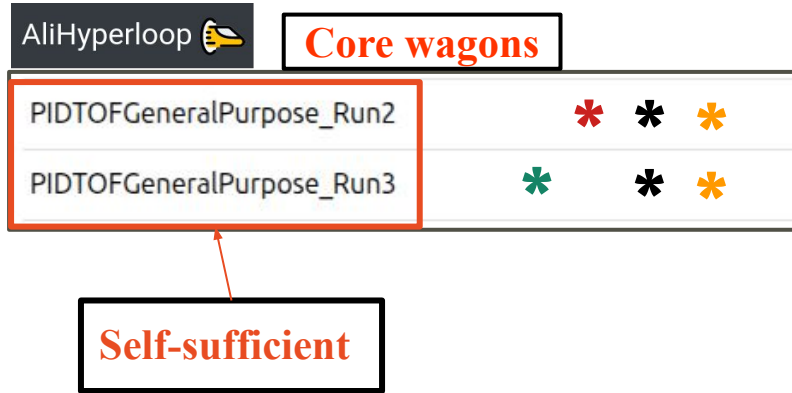


TOF PID helper tasks on Hyperloop

- Dependencies on Hyperloop

- **Run 2**
- **Run 3**
- **Data**
- **MC**

- Include the one according to your needs and in doubt ask the experts!!
- Only general purpose tasks are needed (no other TOF dependencies). Setup of collision system/MC is automatic.
- Take them as reference for local running!!



- N.B. legacy wagons are marked as Deprecated are not actively maintained and will be removed
- Test the new TOF dynamic columns for PID (similar to ITS approach) for better performance. See backup for example.

Taking Hyperloop as a reference

My Analyses All Analyses Dashboard AliHyperloop Train Submission Train Runs Datasets DPG Runlists ? 24

PIDTOF_QA
PIDTOFBaseRun3_deprecated
PIDTOFBaseRun3_PbPb_deprecated
PIDTOFBaseRun3_PbPb_toEvTime_deprecated
PIDTOFBetaRun3_deprecated
PIDTOFBetaRun3_PbPb_deprecated
PIDTOFFullRun3_deprecated
PIDTOFFullRun3_PbPb_deprecated
PIDTOFFullRun3_PbPb_toEvTime_deprecated
PIDTOFGeneralPurpose_Run2
PIDTOFGeneralPurpose_Run3
PIDTOFGeneralPurpose_Run3_MC_OverrideFT0
PIDTOFRun3_deprecated
PIDTOFRun3_PbPb_deprecated
PIDTPC_QA
PIDTPCService
PIDTPCService-MonteCarloTuneOnData
PIDTPCService-NeuralNetwork
PIDTPCService-NeuralNetwork-MonteCarloTuneOnData
PIDTPCService-NeuralNetwork-MonteCarloTuneOnData-Run2

PIDTOFGeneralPurpose_Run2

Wagon settings Configuration 1 Derived data Test Statistics Grid Statistics

Latest change by **ddobrigk** at 08/09/25, 15:01 GMT+5:30

Subwagons cannot be used for workflows which have derived data output.

Base

tof-event-time		
processRun2	💡	1
processRun3	💡	0
computeEvTimeWithFT0	💡	-1
computeEvTimeWithTOF	💡	-1
maxEvTimeTOF	💡	100000
maxMomentum	💡	2

Copy the configuration into your **json** to run locally

– 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/1607965/>
- DPG AOT-track (Thursdays 9h30) → <https://indico.cern.ch/event/1568266/>
- Detector and QA experts
- ITS → francesco.mazzaschi@cern.ch
- TPC → ana.marin@cern.ch
- TOF → nicolo.jacazio@cern.ch
- PID QA → DPG-AOT coord.

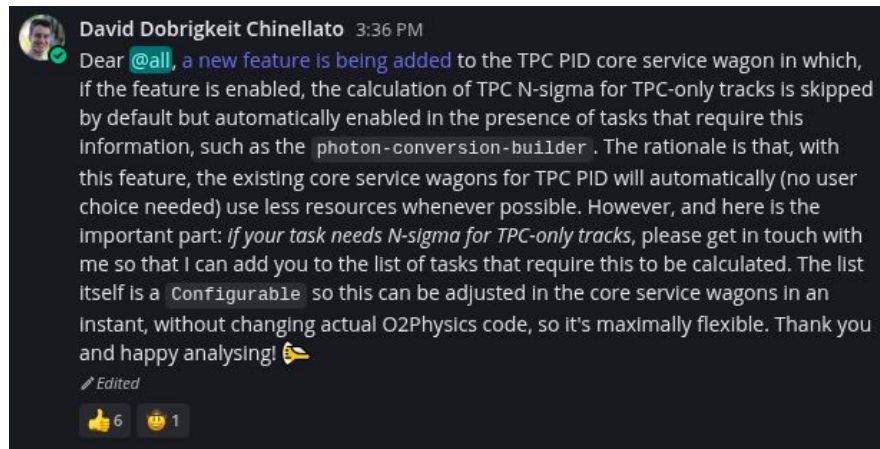
Suggestions are welcome, eager to improve!

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>

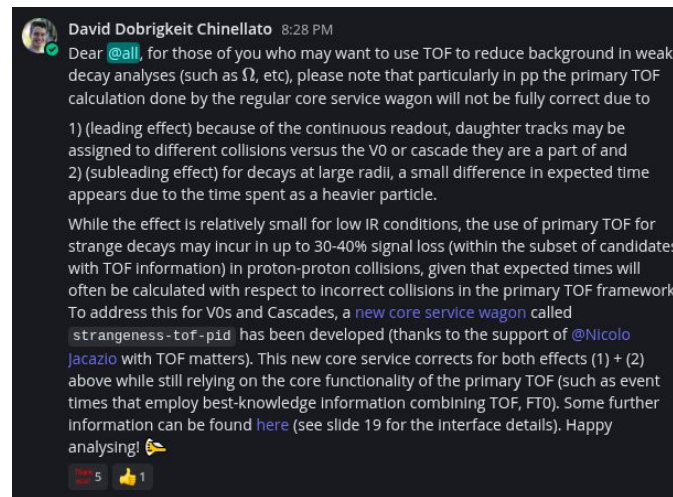


David Dobrigkeit Chinellato 3:36 PM

Dear @all, a new feature is being added to the TPC PID core service wagon in which, if the feature is enabled, the calculation of TPC N-sigma for TPC-only tracks is skipped by default but automatically enabled in the presence of tasks that require this information, such as the `photon-conversion-builder`. The rationale is that, with this feature, the existing core service wagons for TPC PID will automatically (no user choice needed) use less resources whenever possible. However, and here is the important part: *if your task needs N-sigma for TPC-only tracks*, please get in touch with me so that I can add you to the list of tasks that require this to be calculated. The list itself is a `Configurable` so this can be adjusted in the core service wagons in an instant, without changing actual O2Physics code, so it's maximally flexible. Thank you and happy analysing! 🙌

Edited

👍 6 🧐 1



David Dobrigkeit Chinellato 8:28 PM

Dear @all, for those of you who may want to use TOF to reduce background in weak decay analyses (such as Ω , etc), please note that particularly in pp the primary TOF calculation done by the regular core service wagon will not be fully correct due to

- 1) (leading effect) because of the continuous readout, daughter tracks may be assigned to different collisions versus the V0 or cascade they are a part of and
- 2) (subleading effect) for decays at large rad_{ll} , a small difference in expected time appears due to the time spent as a heavier particle.

While the effect is relatively small for low IR conditions, the use of primary TOF for strange decays may incur in up to 30-40% signal loss (within the subset of candidates with TOF information) in proton-proton collisions, given that expected times will often be calculated with respect to incorrect collisions in the primary TOF framework. To address this for V0s and Cascades, a new core service wagon called `strangeness-tof-pid` has been developed (thanks to the support of @Nicolo Jacazio with TOF matters). This new core service corrects for both effects (1) + (2) above while still relying on the core functionality of the primary TOF (such as event times that employ best-knowledge information combining TOF, FTO). Some further information can be found [here](#) (see slide 19 for the interface details). Happy analysing! 🙌

👍 5 🧐 1

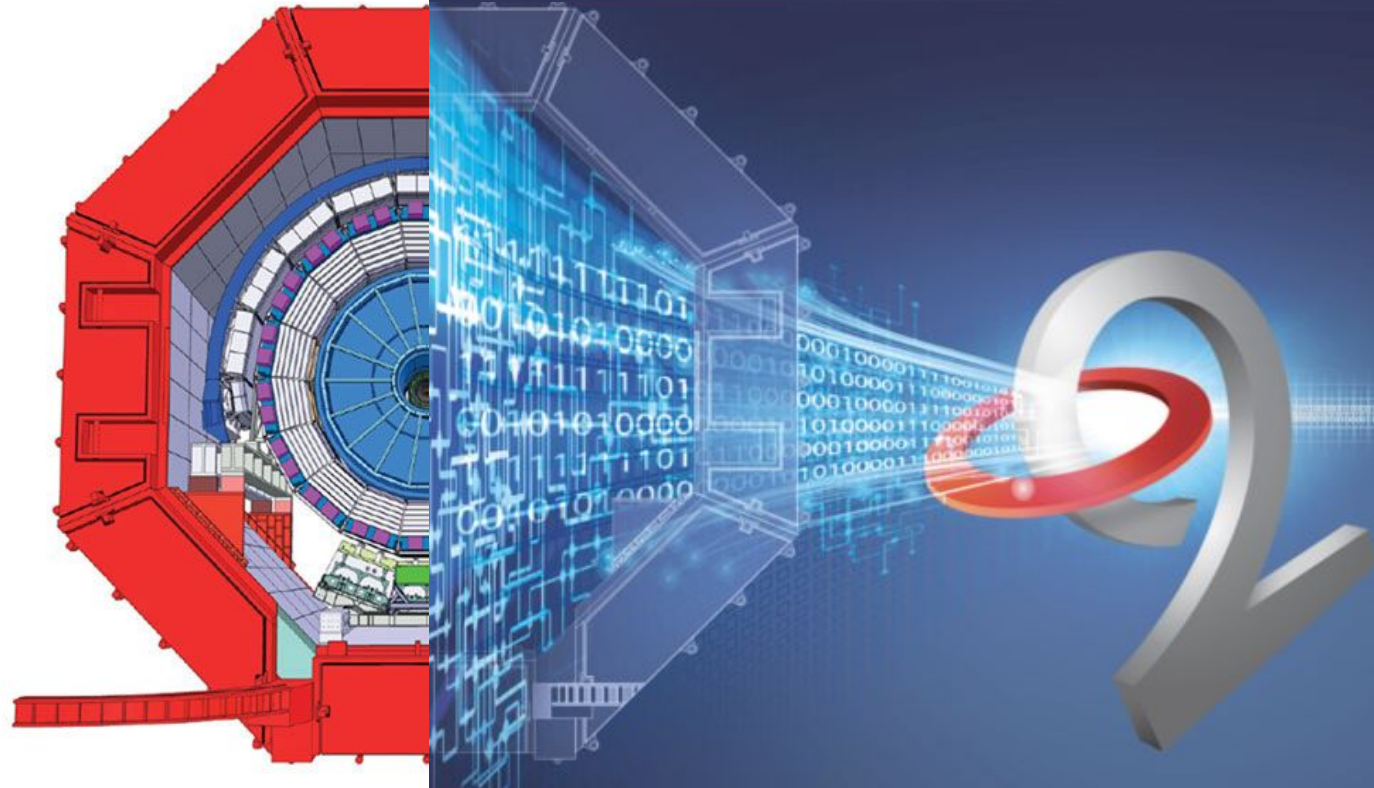
Issues, developments, and calibration releases are announced here (for PID but not only!)

Take home message

- PID will be likely useful in your analysis
 - Provided by central tasks, calibrated and checked by DPG
- Refer to the documentation for usage
- In case of doubts or unexpected features ask away!
- Test it yourself right now!



```
wget -L --content-disposition "https://cernbox.cern.ch/s/rZTWHWSGHRHz8iB/download" -O /tmp/setup.sh
bash /tmp/setup.sh
```



Thank you for your attention

TOF dynamic columns PID response

```
struct tofPidQaImp {
    Service<o2::pid::tof::TOFResponse> tofResponse;
    Service<o2::ccdb::BasicCCDBManager> ccdb;
    void init(o2::framework::InitContext& context)
    {
        tofResponse->initSetup(ccdb, context);
    }

    using TrackCandidates = soa::Join<aod::Tracks, aod::TracksExtra, aod::TrackSelection,
                                     aod::pidEvTimeFlags, aod::TOFSignal, aod::TOFEvTime,
                                     aod::pidTOFFlags>;

    void process(CollisionCandidate const& collision,
                 TrackCandidates const& tracks,
                 o2::aod::BCsWithTimestamps const&)
    {
        tofResponse->processSetup(collision.bc_as<o2::aod::BCsWithTimestamps>());
        auto tracksWithPid = soa::Attach<TrackCandidates, o2::aod::TOFNSigmaDynEl>;
        for (const auto& t : tracksWithPid) {
            t.tofNSigmaDynEl();
        }
    }
}
```