# ALICE project: Introduction and hands-on

# Outline of the proposed study

Reconstruct the decay of two charm hadrons:

- $D^0 \rightarrow K^-\pi^+$
- $\Lambda_c^+ \rightarrow pK^-\pi^+$

analysing proton-proton collision data from the ALICE experiment

- real data
- data from Monte Carlo simulations with enhanced signal

These particles are relatively rare and without selections the signal-to-background ratio is quite low

I will give you a series of root files containing root trees

After some first explanations and instructions, and a first ~hands-on session together,
I will give you some practical goals (mainly plots you should produce) and let you free to play with the data

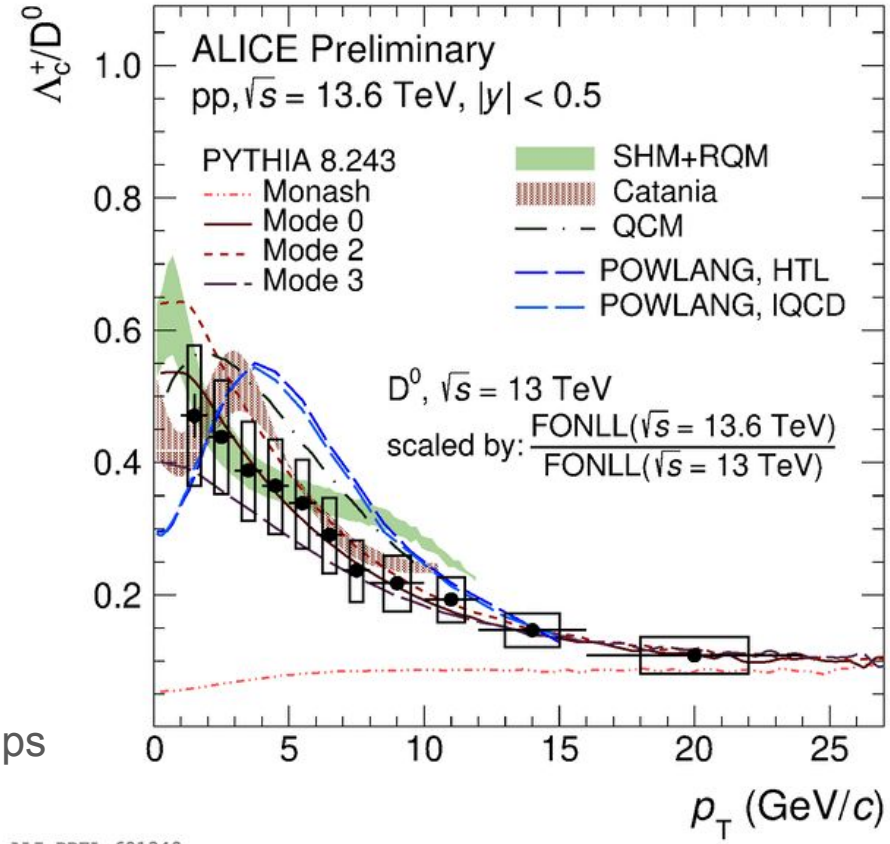
# Outline of the proposed study

Ultimate goal would be to measure
the production cross section of the two particles
and compare them like in this result

Ratio of the production cross sections of these
two particle is a sensitive observable to investigate
the hadronisation process, i.e. the transition from
a system of quarks to one of hadrons

However, we cannot repeat the full analysis
- time limitation
- data-size needed too large

We will study and reproduce (some of) the main steps



# Objectives

1) Become familiar with data-analysis techniques typically used in particle and (high-energy) nuclear physics
    - few concepts of data reconstruction
    - PID, vertexing, invariant mass analysis
2) Apply Machine Learning classification to reject background
3) Touch concepts and problems typical of the measurements done in this (and not only) context:
    - selection efficiency
    - statistical significance
    - systematic uncertainties

# Data and software tools

Data files are root files containing trees (flat tables)
You will find them in the Cloud Veneto space at this directory:
/home/ubuntu/ALICEphyisicsOfData/

Right now, just one file in: /home/ubuntu/ALICEphyisicsOfData/FirstFile/AO2Dtree.root
For today, you can access the file we need, also via this google link:
https://drive.google.com/file/d/1CCHq1R24JuJqKFkmelCuJgnrQEcXzAYK/view?usp=drive_link

You can analyse them via ROOT(*) or with whatever programme/language you prefer: use UPROOT, transform in pandas dataframe, use python.

If you like to use Jupyter notebook, that's more than welcome!

(*) it might help to look at Root documentation and tutorial, https://root.cern/doc/master/group__Tutorials.html
but my advice is to do it when you need specific information. Learn by doing things.

# What we do together

1) Introduction:
    i) main goals of ALICE experiment related to this project
    ii) basic concepts of data reconstruction
    iii) quick introduction to the physics case that you will explore
2) Hands-on session:
    i) data structure
    ii) first steps:
        - look at main single-track variables used in analysis
        - produce figures related to PID and tracking performance
        - build by hand candidates of decay particles ($K^0_s$, $D^0$)
3) Second appointment:
    i) look at files with already produced and filtered candidates
    ii) introduction to the main variables used to distinguish signal and background
    iii) discuss concepts of signal extraction, selection efficiency, statistical significance
    iv) systematic uncertainties

# Brief introduction to what ALICE looks at in pp and Pb-Pb collisions

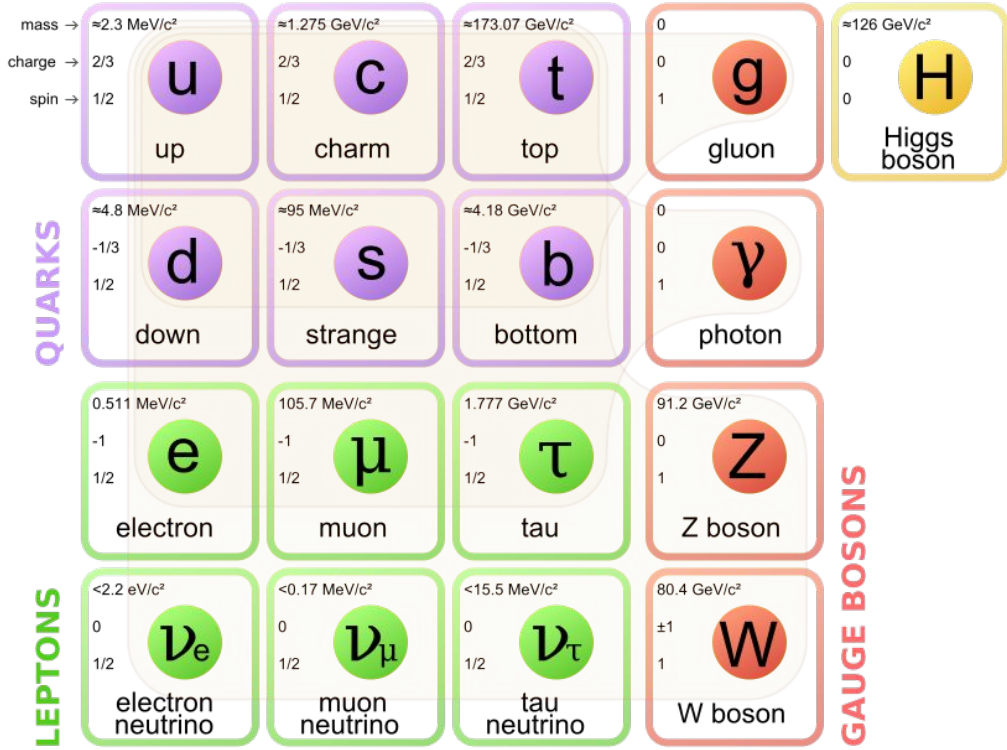
# Recall: Elementary particles in the Standard Model

Elementary = no internal structure, pointlike (no dimension) opposite of “composite”
→ fundamental building blocks of all known matter

6 quarks
6 leptons
1 Higgs boson
4 gauge bosons (interaction messengers)

3+1 interactions:

- strong → g
- electromagnetic → $\gamma$
- weak → $W^{\pm}$ ,$Z^0$
- (gravitation → graviton?)

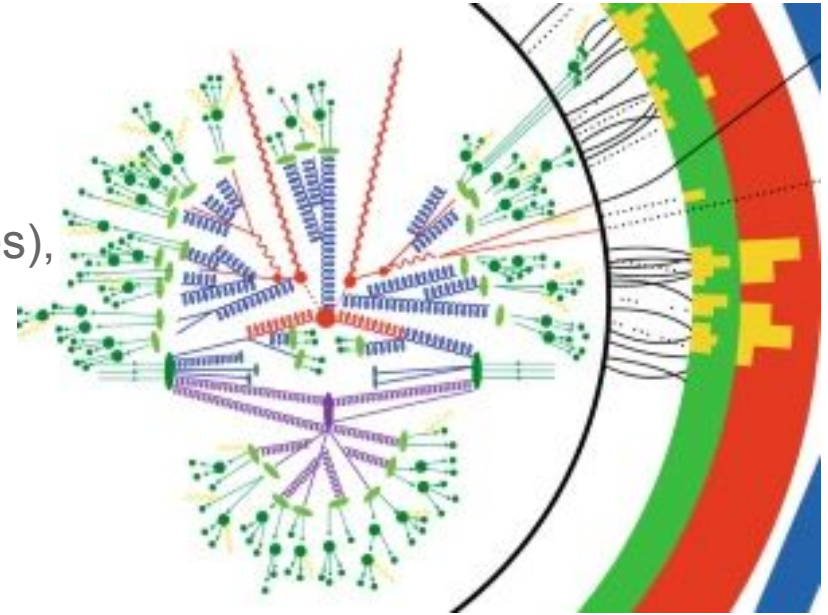



[https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg]

# Investigating the strong force at high-energy at colliders

What happens in **proton-proton collisions at high energies** like those accessible at the **Large Hadron Collider at CERN**?

Quarks and gluons inside protons
interact and produce
**several quarks and gluons** (rarely other particles),
i.e. a system of interacting partons,
which **evolves until hadrons are formed**



ALICE goal:
**study the properties and evolution of the partonic and hadronic systems**
→ learn properties of strong force and of some fundamental processes in nature

Picture from: Georgescu, I. Simulating parton showers. *Nat Rev Phys* 3, 73 (2021)

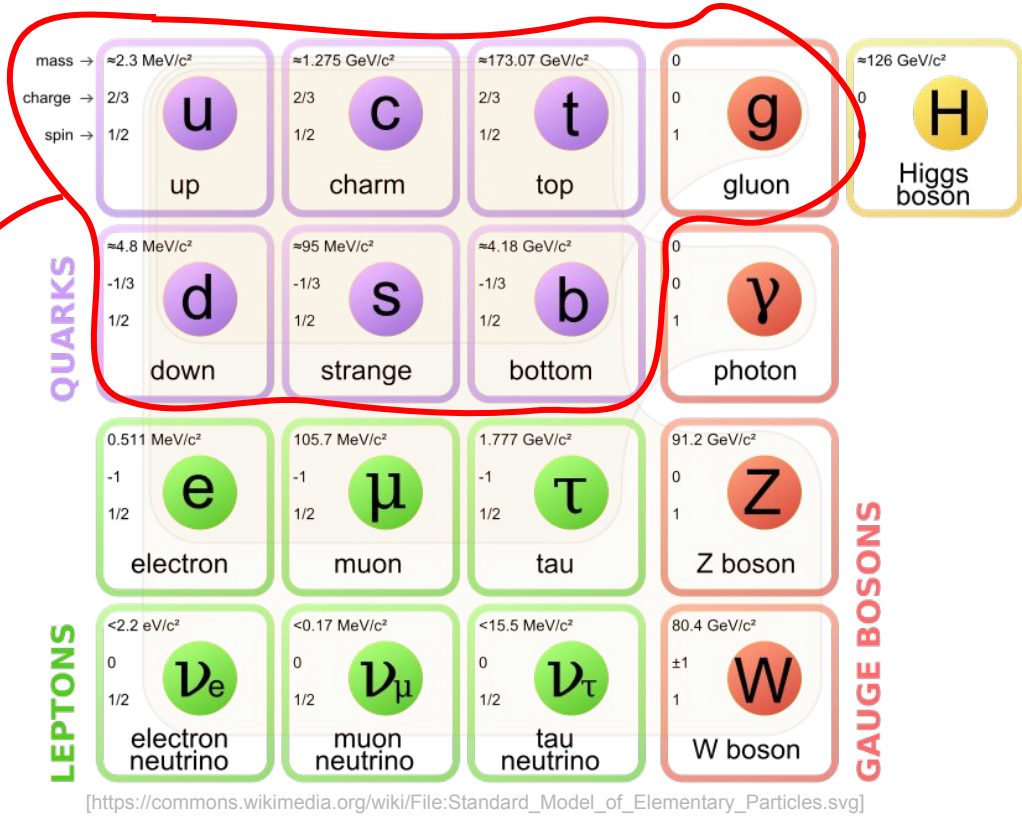
# Recall: Elementary particles in the Standard Model

Elementary = no internal structure, pointlike (no dimension) opposite of “composite”
→ fundamental building blocks of all known matter

6 quarks
6 leptons
1 Higgs boson
4 gauge bosons (interaction messengers)

3+1 interactions:

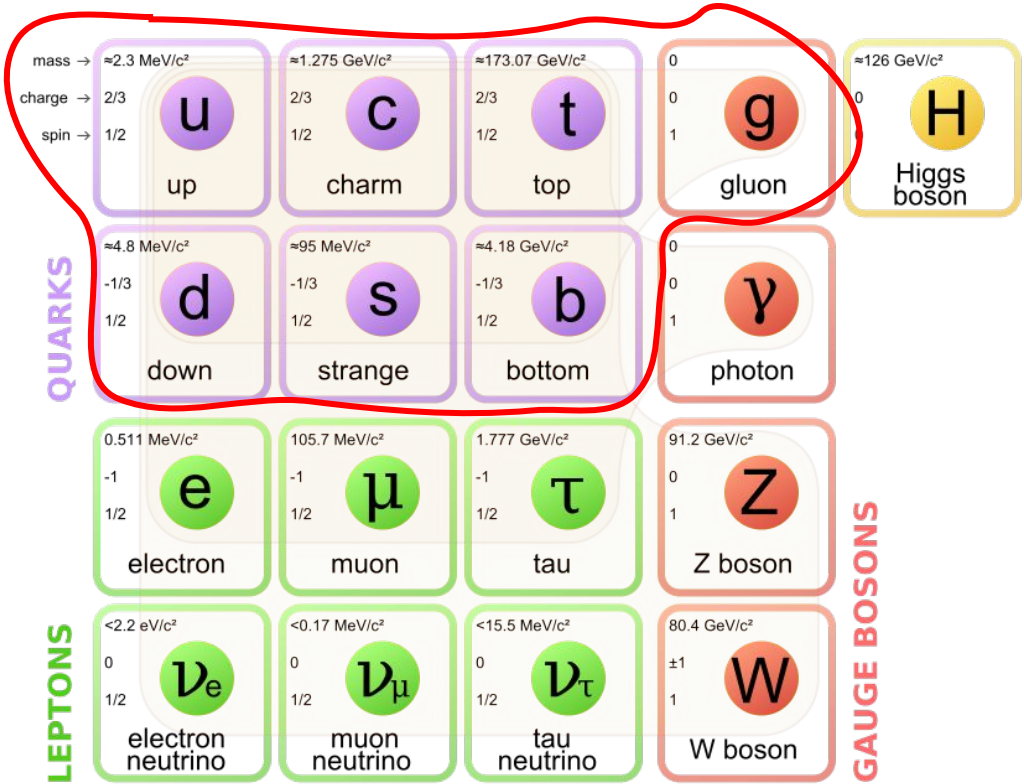
- strong → g
- electromagnetic → γ
- weak → $W^{\pm}$ ,$Z^0$
- (gravitation → graviton?)

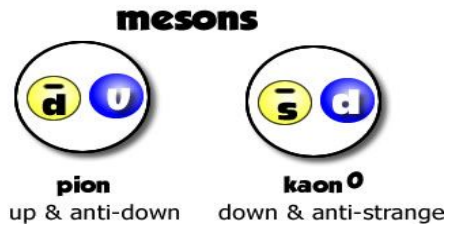


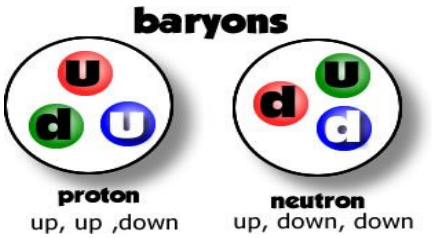

[https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg]

# (Elementary particles and hadrons)

We never observe free quarks, only composite objects called hadons, in which quarks are bound and confined to stay by the strong nuclear force

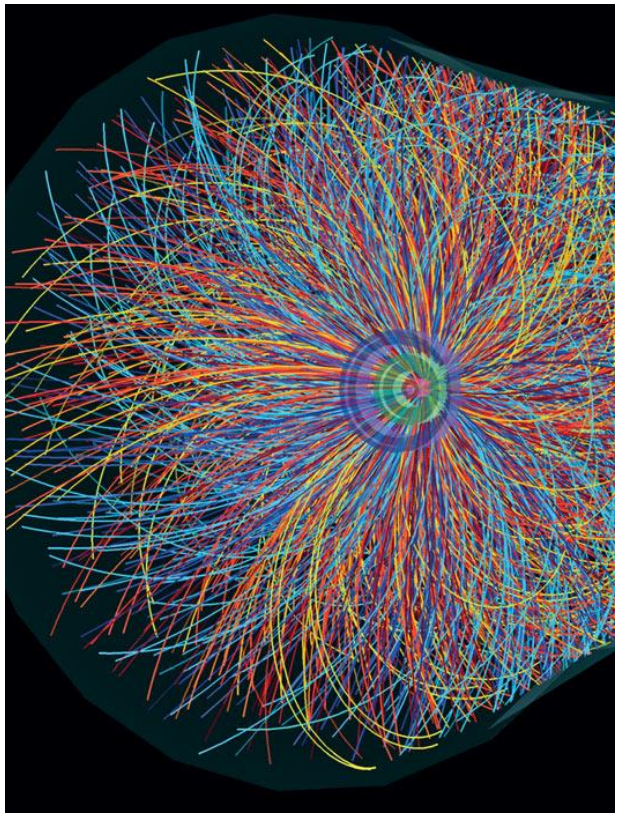


# Investigating the strong force at high-energy at colliders

**What happens in Pb-Pb collisions** at high energies like those accessible at the Large Hadron Collider at CERN?



Something similar to proton-proton collisions

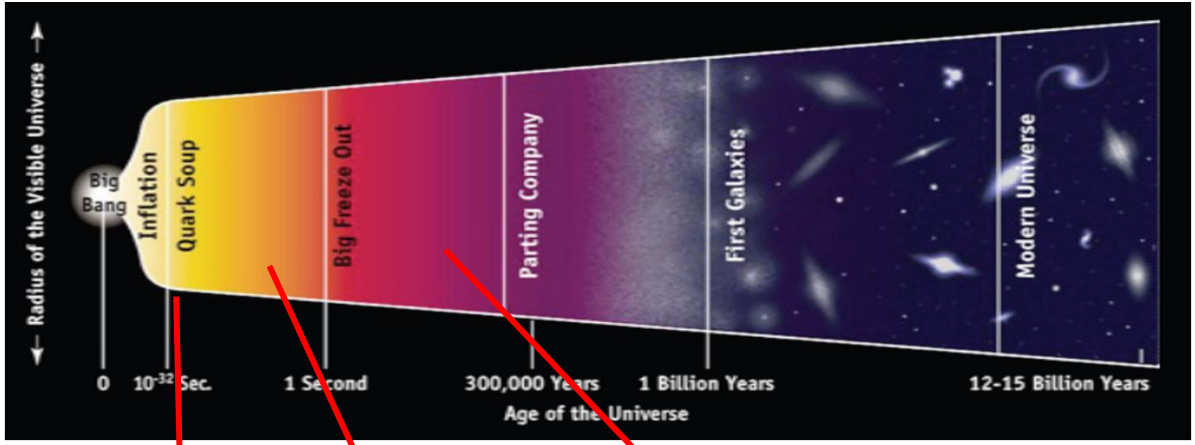
but the **number of produced quarks and gluons (and then hadrons) is much higher** (about x1000)

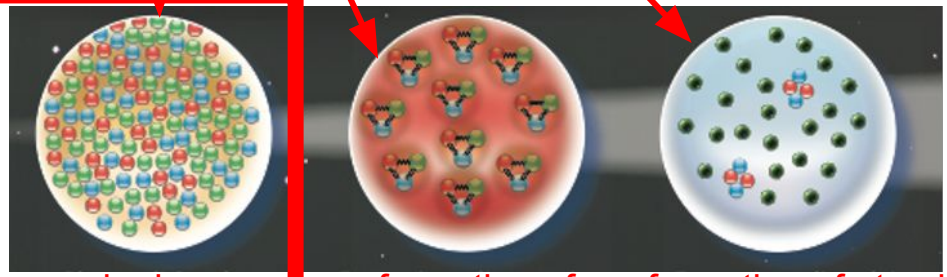
Partonic system with very high energy density

which behaves for a very short time (few $10^{-23}$ s) as a **quark-gluon plasma (QGP) state**

in which quarks and gluons behave and interact as free particles

# Quark-gluon matter in the Early Universe …



quark-gluon plasma

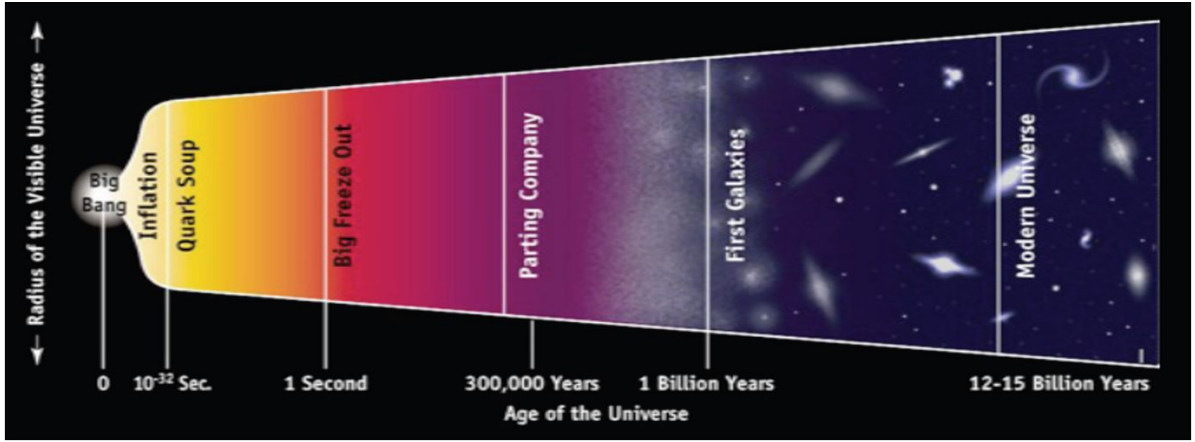
formation of protons, neutrons

formation of atomic nuclei

The transition from quarks to hadrons occurred in the expanding & cooling early Universe
**~10 µs after the Big Bang**

→ Before: **Quark Gluon Plasma**

# Quark-gluon matter in Neutron Stars…




The transition from quarks to hadrons occurred in the expanding & cooling early Universe
**~10 μs after the Big Bang**

→ Before: **Quark Gluon Plasma**

QGP **may** characterise also the core of neutron stars.

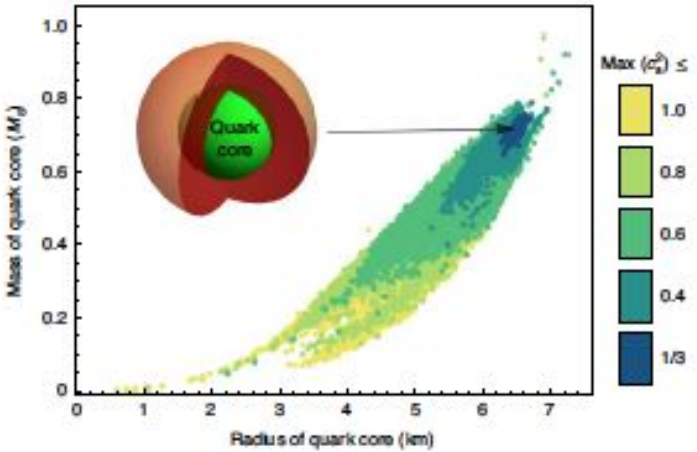
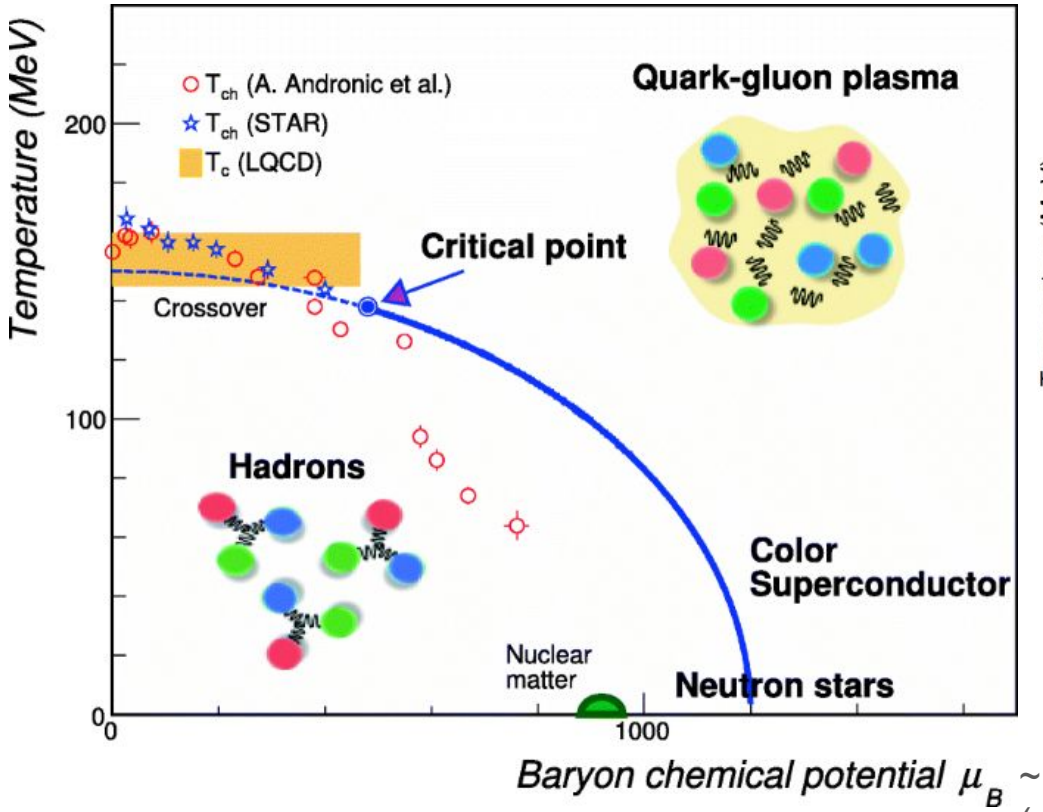
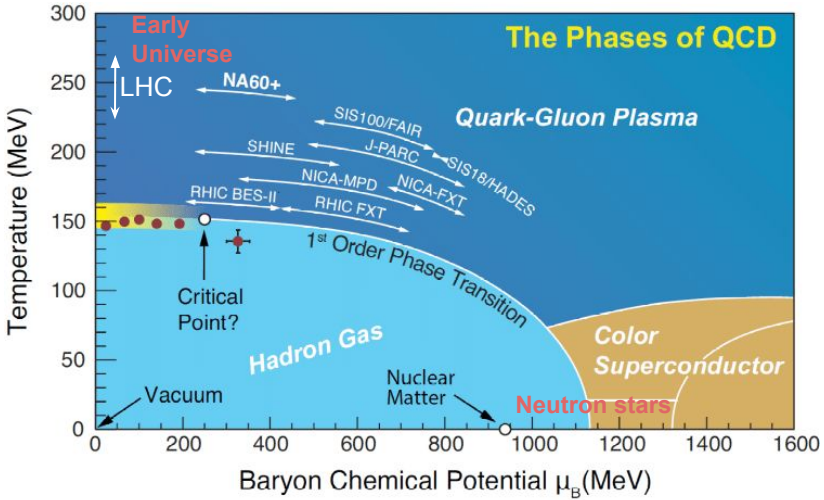



figure from Nature Physics, 16, 907-910 (2020),
https://www.nature.com/articles/s41567-020-0914-9

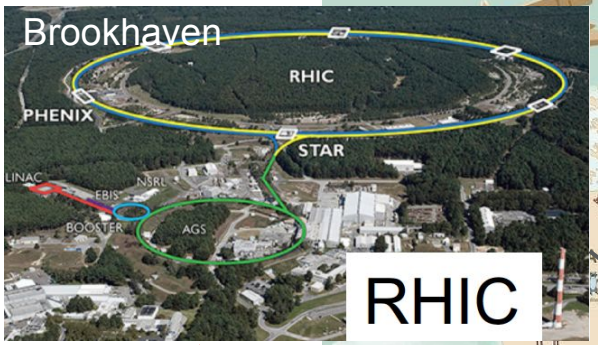
# Exploring the strong-interaction phase diagram

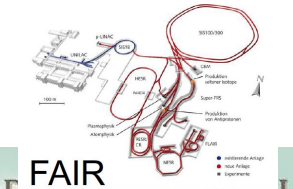



~ net-baryon density
(=density of protons - density of antiprotons)

# Quark-gluon matter on Earth







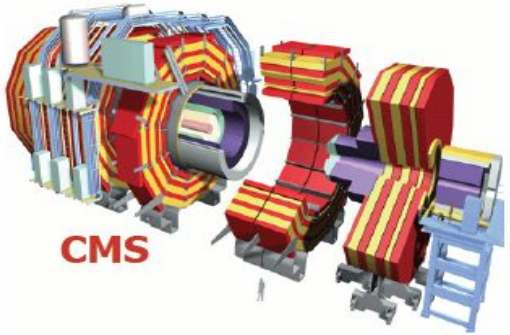


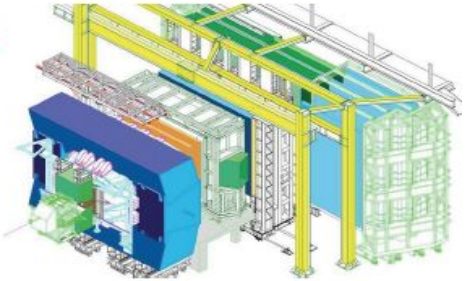


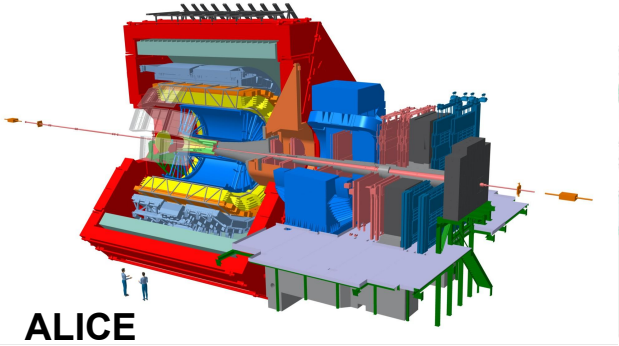

**High Energy Frontier**

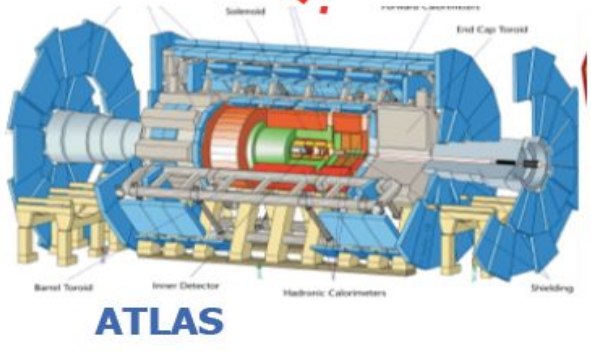
ALICE

# The ALICE experiment at the Large Hadron Collider



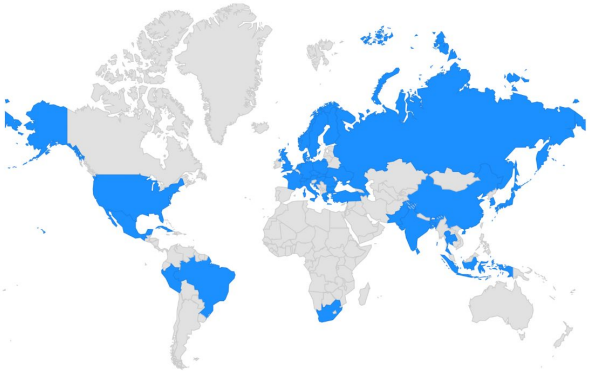



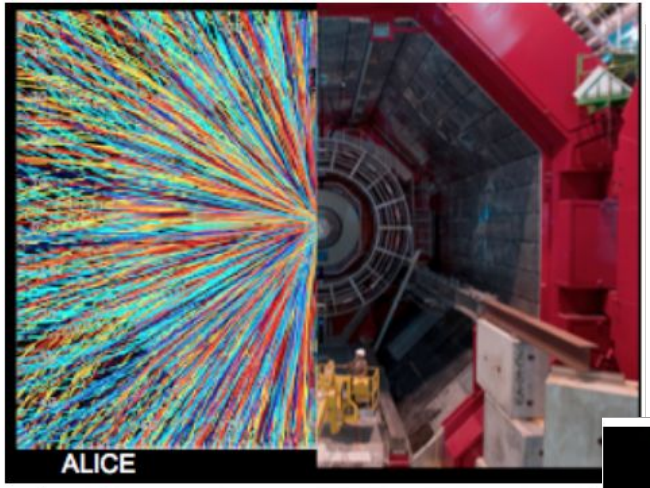



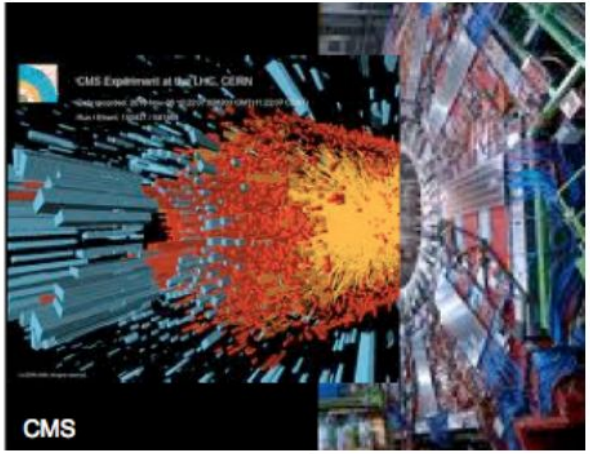
# The ALICE experiment at the Large Hadron Collider

**40 countries, 170 institutes, ~2000 members**



# What we see in Pb-Pb collisions

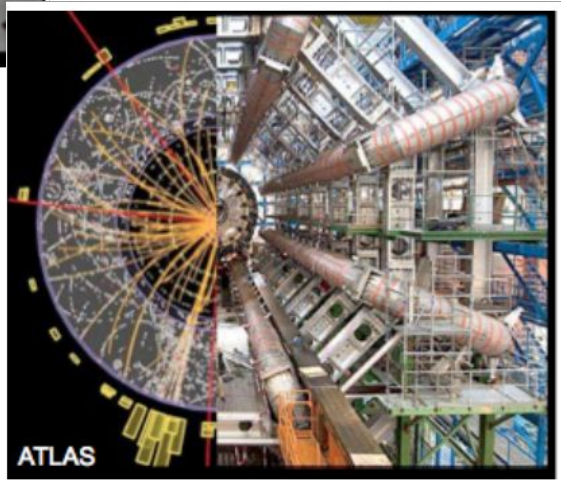


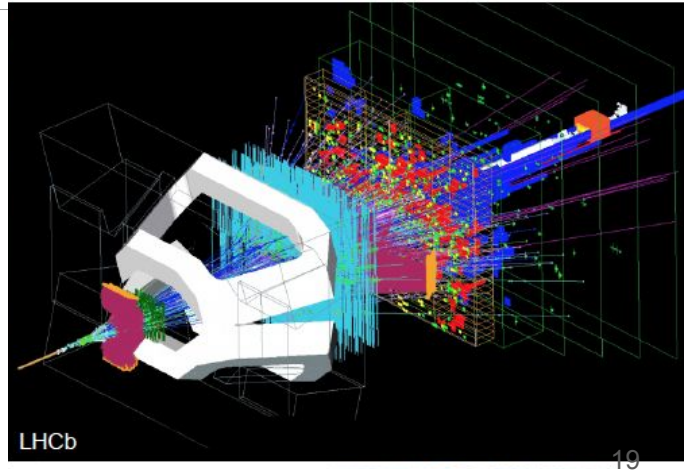




Coloured lines and boxes: visualisation of particle tracks or calorimeter energy deposits

From “event display” (so from real data)

Thousands of particles produced in a head-on Pb-Pb collision at the LHC
A lot of energy involved




“Artistic representations” of various features: MADAI webpage: https://madai.phy.duke.edu/indexaae2.html?page_id=503

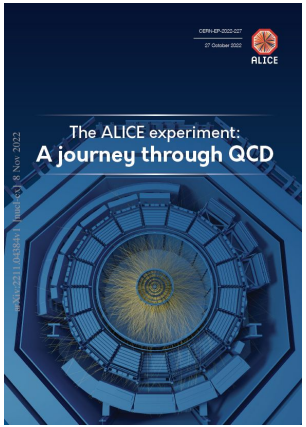
# What do we want to know?

*What are the **global properties (e.g. temperature, density, volume, viscosity)** of the system?*

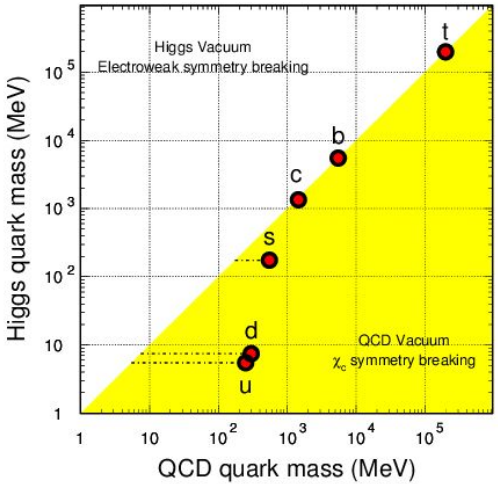
*Can we model the evolution of the quark-gluon system **from proton-proton to Pb-Pb collisions**?*

*How do **hadrons form out of a system of quarks** (hadronisation process)?*

… many more questions




https://arxiv.org/pdf/2211.04384.pdf




N.B.

- Higgs boson accounts only for a few % of the matter mass:
  M(proton) ~ 938 MeV/$c^2$
  M(up, down) ~ few MeV/$c^2$

- Most of matter mass is generated dynamically during the transition from quarks to hadrons

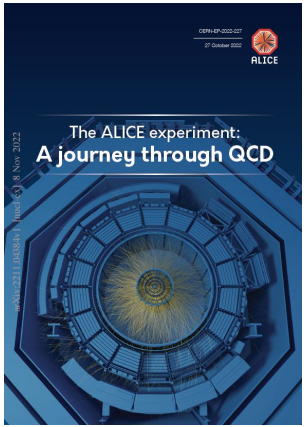
# What do we want to know?

*What are the **global properties (e.g. temperature, density, volume, viscosity)** of the system?*

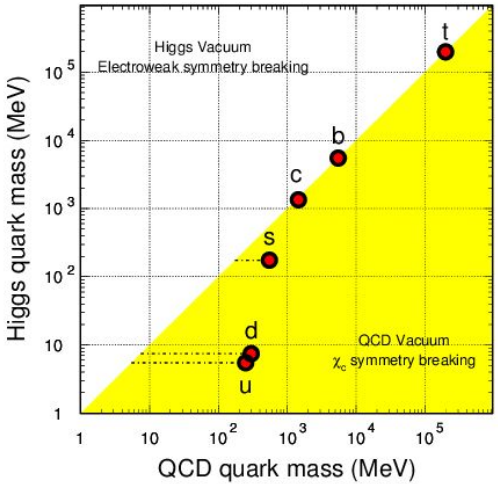
*Can we model the evolution of the quark-gluon system **from proton-proton to Pb-Pb collisions**?*

*How do **hadrons form out of a system of quarks** (hadronisation process)?*

… many more questions




https://arxiv.org/pdf/2211.04384.pdf

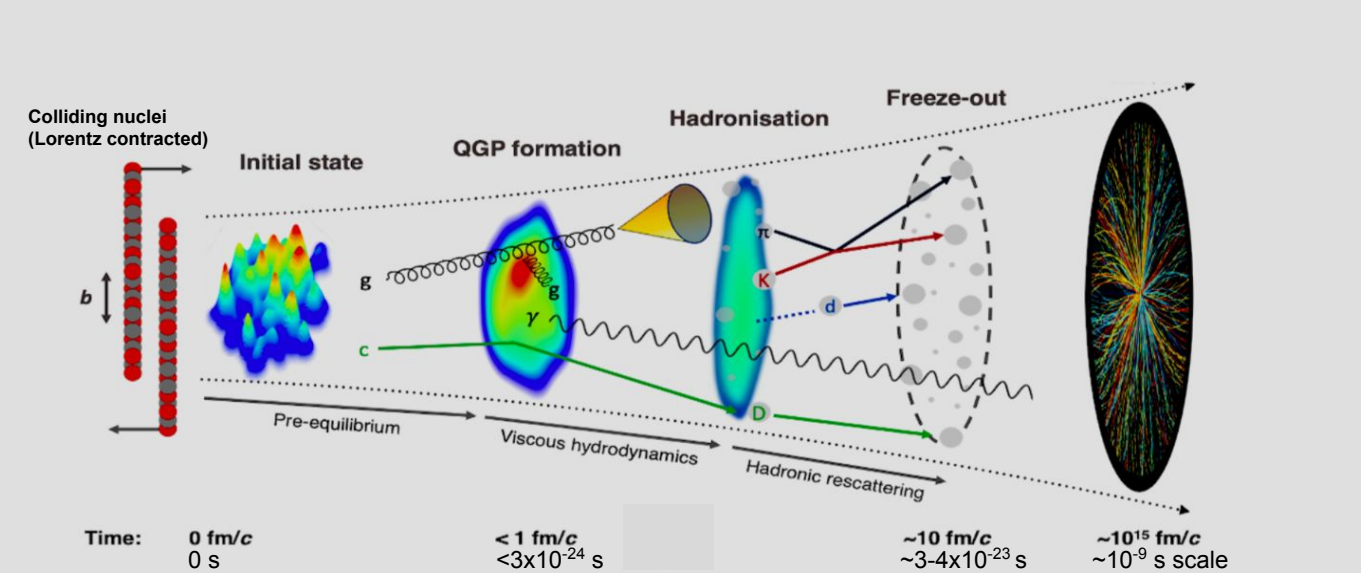



N.B.

- Higgs boson accounts only for a few % of the matter mass:
  M(proton) ~ 938 MeV/$c^2$
  M(up, down) ~ few MeV/$c^2$

- Most of matter mass is generated dynamically during the transition from quarks to hadrons

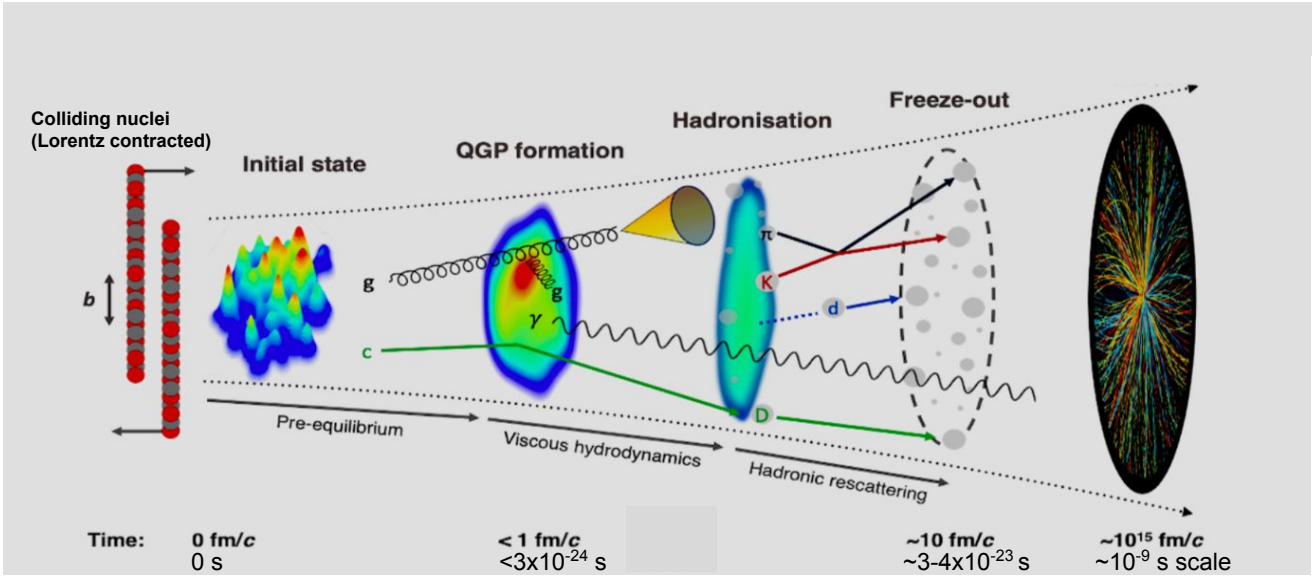
# Back to nuclei collisions: system evolution and phases

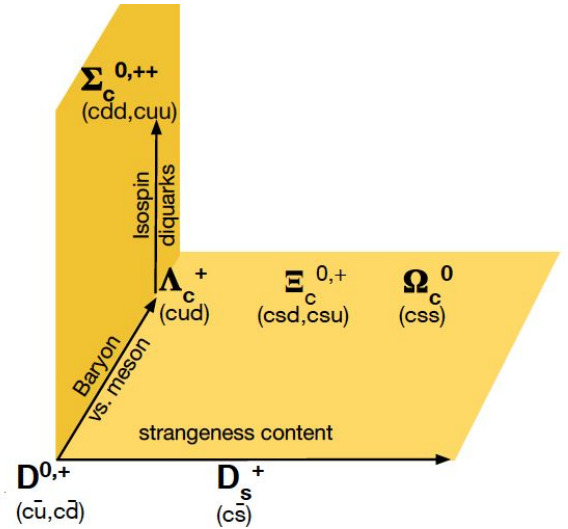



## Temperature

Phase transition critical temperature: $T_c$=156 MeV ~ $1.8 \cdot 10^{12}$ K
Sun core: $1.5 \cdot 10^7$ K
Sun surface: 5778 K

# The role of charm quarks to study hadronisation




**Massive quarks (mass >> temperature)** as charm and beauty are produced only in hard-scattering processes occurring in the **very first instants** (before QGP formation, before hadronisation)

The production rate and quark kinematics can be calculated with perturbative QCD techniques

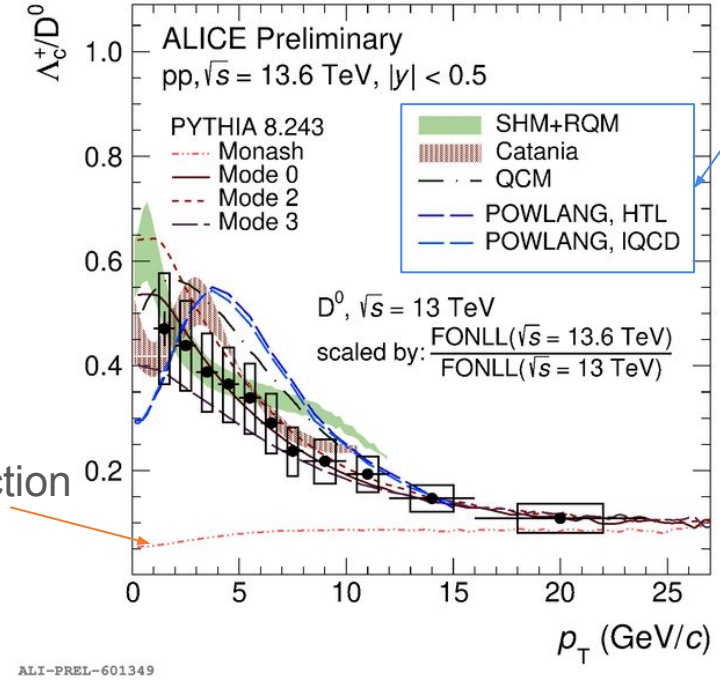
→ theoretically under control

→ we can use these quarks as probes to investigate the medium and the hadronisation process

In Pb-Pb collisions, but also in proton-proton collisions

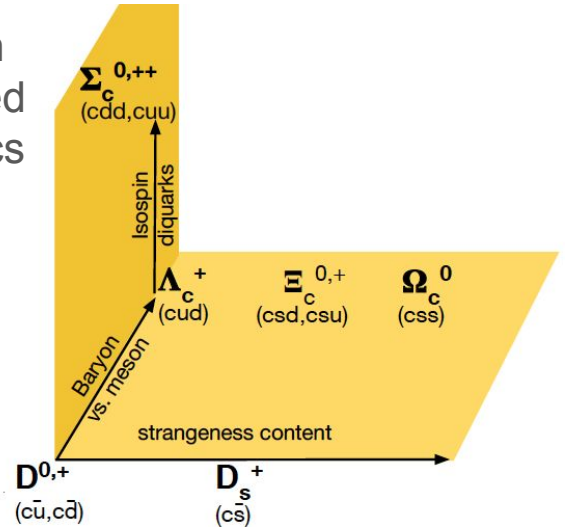
**How? by looking at hadrons with different quark composition**

# The role of charm quarks to study hadronisation



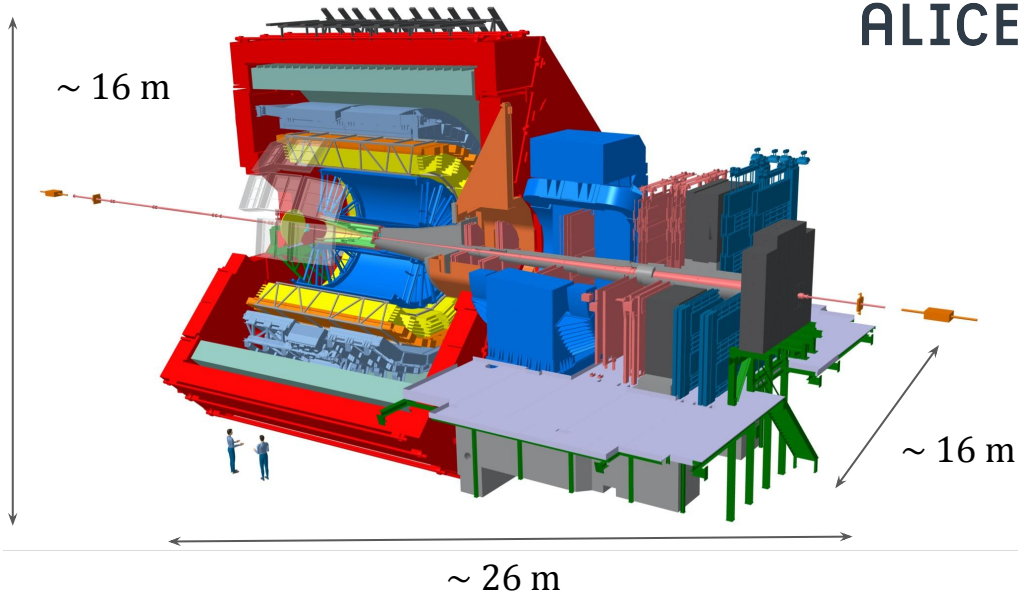

initial prediction

Models based on concepts inherited from QGP physics




We started looking at proton-proton collisions as the needed reference to interpret Pb-Pb results … but it turned out that the modelling of hadronisation in pp was naïve

# The ALICE experiment at the Large Hadron Collider



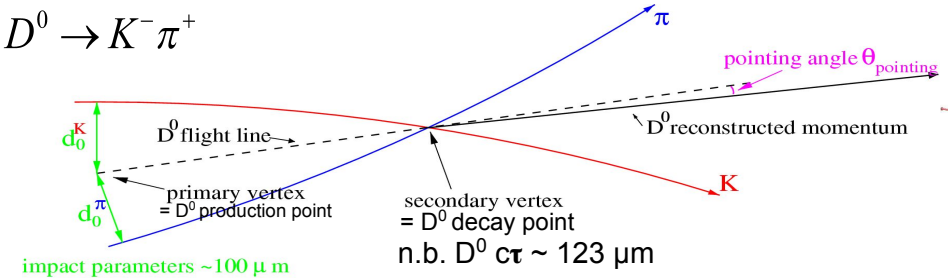

“Gigantic” particle accelerator (LHC) with gigantic apparatus

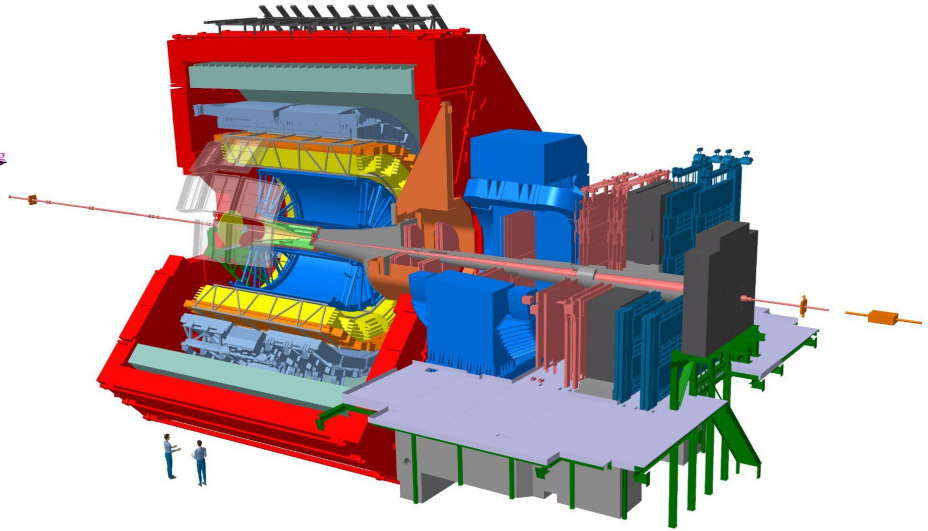
Magnets, trackers, muon-chambers, calorimeters… and several other detectors

ALICE public webpages (just look for “ALICE CERN”): https://alice-collaboration.web.cern.ch/ https://alice.cern/alice-physics

# $D^0$ meson reconstruction

$D^0 \rightarrow K^- \pi^+$



**How much this signal is rare and the signal-to-background ratio (S/B) low?**

Rough rough estimate:

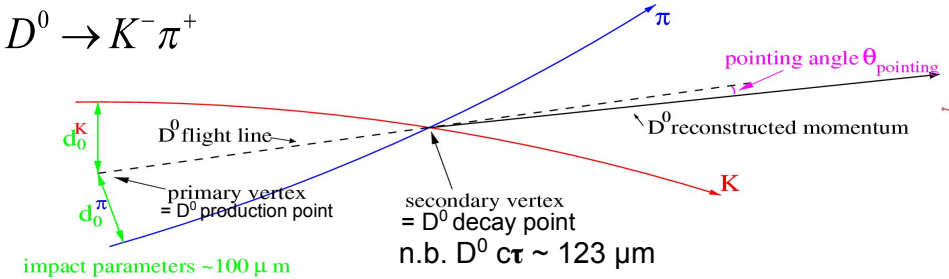
- Number of $D^0$/collision in detector acceptance ~1/100
- Branching ratio (fraction of decays in the specific decay channel)~4% → **S/collision~4/10000**
- Number of charged particles/collision in pp at 13 TeV in relevant detector acceptance: ~12, half positive, half negative → **B/collision ~ 36 → S/B ~ $10^{-5}$**

A similar estimate, for the $\Lambda_c^+ \rightarrow pK^-\pi^+$ gives S/B~ $10^{-6}$

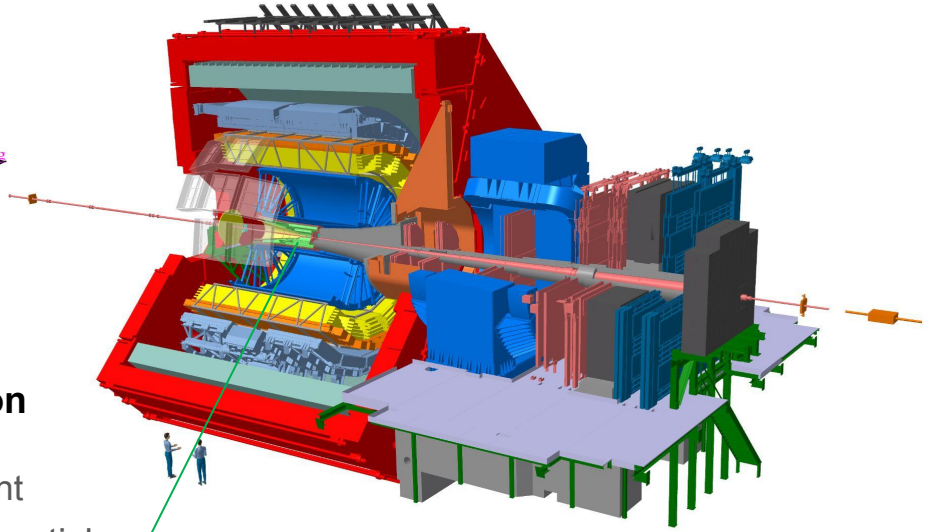
**How do we improve it?**

# $D^0$ meson reconstruction

$D^0 \rightarrow K^- \pi^+$




n.b. $D^0$ $c\tau$ ~ 123 μm

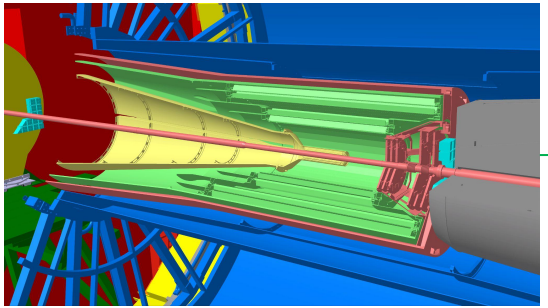


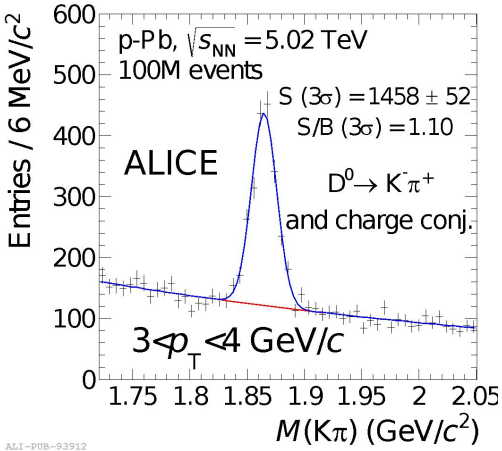
Exploit **invariant mass**, **secondary vertex reconstruction** and **particle ID (PID)**

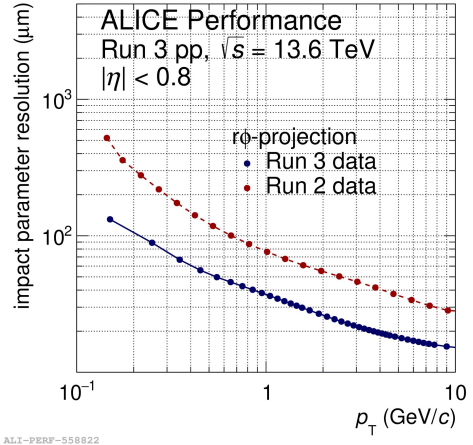
- $d_0$ = impact parameter ~ distance of a track from collision point
- decay length ~ hundreds micron

vs. most background particles produced at collision point

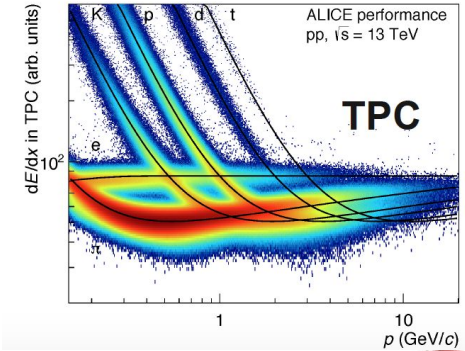
ITS (pixel detector with resolution ~ 5-10 μm

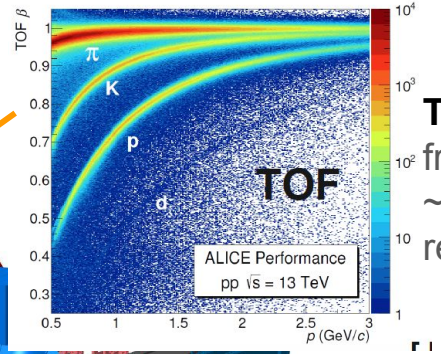




# Particle IDentification (PID) in the ALICE experiment




**Time Of Flight**
from collision point to ~4m away
resolution ~80 ps



**Time Projection Chamber**
PID signal: specific energy loss in detector gas

Described by Bethe-Block formula (black lines in above figure)

# Particle IDentification (PID) in the ALICE experiment

**PID with Time Of Flight (TOF)**

Measurement of arrival time $t_{track}$ w.r.t. collision "start time" $t_{0\ event}$

Distance covered ($L$) is known

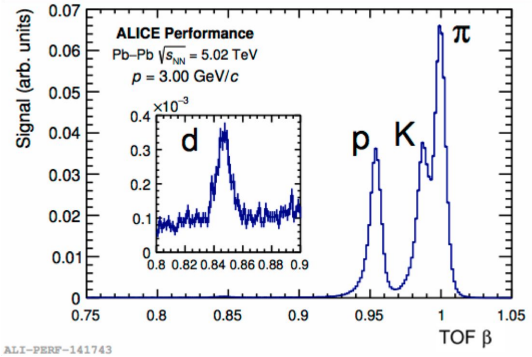
→ measure particle velocity ($\beta = v/c$)
$\beta = L_{track} / (t_{track} - t_{0\ event})c$

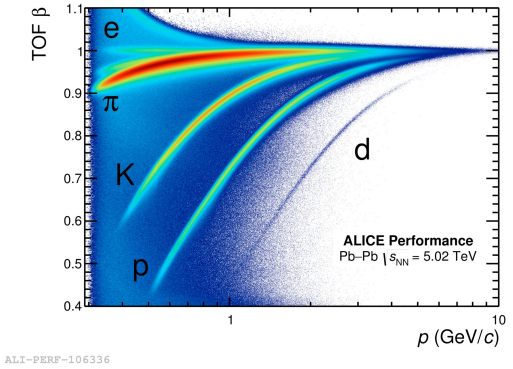
$L \sim 4$ m ; $\Delta t\ (\beta = 1) = c/L = 13.3$ ns

Detector + $t_{0\ event}$ resolution ~ 100 ps

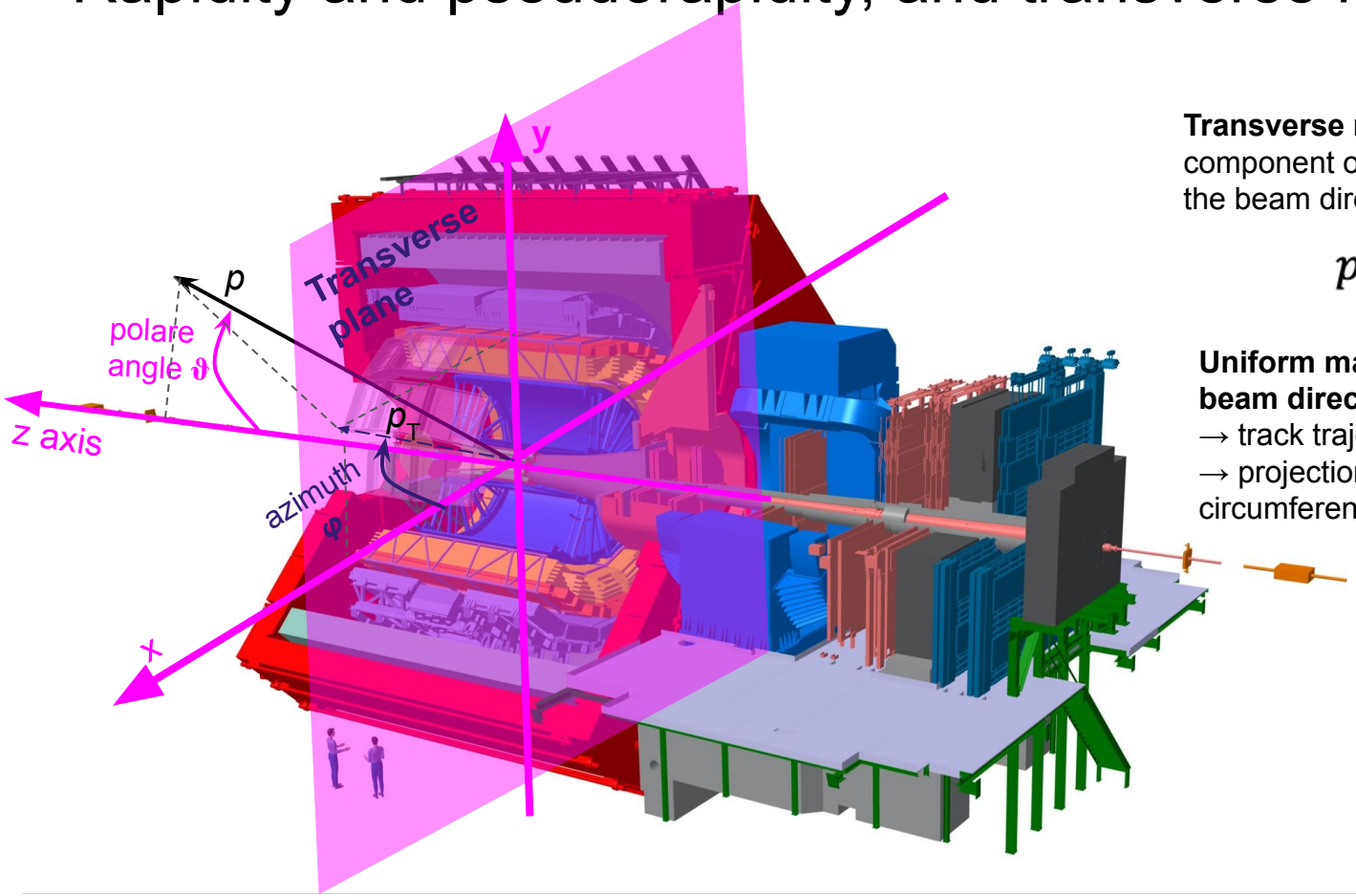
Knowing momentum (TPC) → mass ID

E.g. pion with p= 0.600 [2] GeV/c → $\Delta t$ = *13.691 ns [13.366 ns]*
E.g. kaon with p= 0.600 [2] GeV/c → $\Delta t$ = *17.3 ns [13.734 ns]*



# Rapidity and pseudorapidity, and transverse momentum

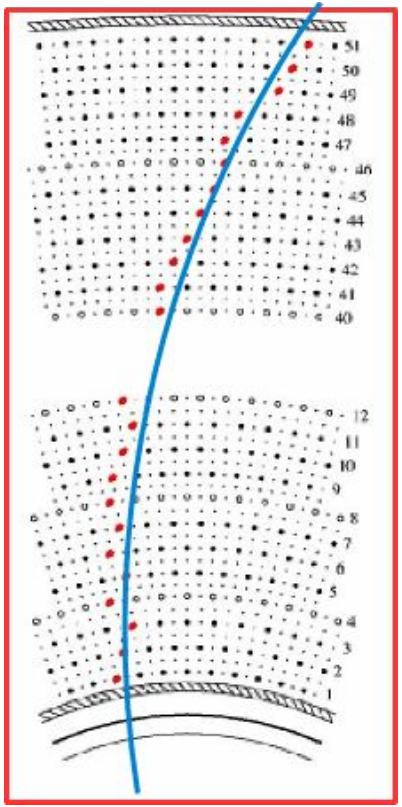



**Transverse momentum ($p_T$)**: component of momentum transverse to the beam direction

$$p_T = \sqrt{p_x^2 + p_y^2}$$

**Uniform magnetic field parallel to beam direction**, B=0.5 T)
→ track trajectories are helix
→ projection on transverse plane is a circumference

# Measuring particle trajectories and momentum



A series of **experimental "points"** that must be identified as belong to a given particle ( → "**track finding**") and **fitted** to reconstruct the particle trajectory ("**tracking**").

**Curvature in magnetic field B and transverse momentum ($p_T$ )** related by:

$$p_T\,[\mathrm{GeV/c}] = 0.3\ R\,[\mathrm{m}]\ \ B\,[\mathrm{T}]\ z\ [\text{charge, in proton-charge units}]$$

→ measurement of radius R (actually what is measured is the sagitta) → $p_T$

**Examples**

1) what is the radius of the track of a particle with $p_T$=1 GeV/*c* (assume z=1, B=0.5 T)?

R=1/(0.3*0.5) = 6.67 m

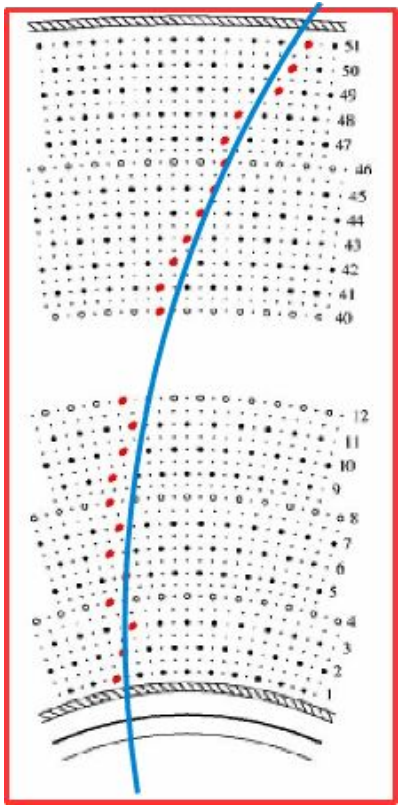
2) what is the minimum $p_T$ a track must have to reach a detector located 4 m away from the collision point, assuming z=1 and B=0.5 T?

diameter =4 m → R=2 m → $p_T$ = 0.3 * 2 *0.5 = 300 MeV/*c*

**Ambiguity for particles with different charges (what we get from *R* is $p_T/z$ )** :

- charge sign from curvature sign
- ambiguity remains but particles with Z>1 (e.g. He nuclei) very rare
  - PID information helps removing ambiguity

# Measuring particle trajectories and momentum



A series of **experimental "points"** that must be identified as belong to a given particle ( → "**track finding**") and **fitted** to reconstruct the particle trajectory ("**tracking**").

**Curva** … related by:

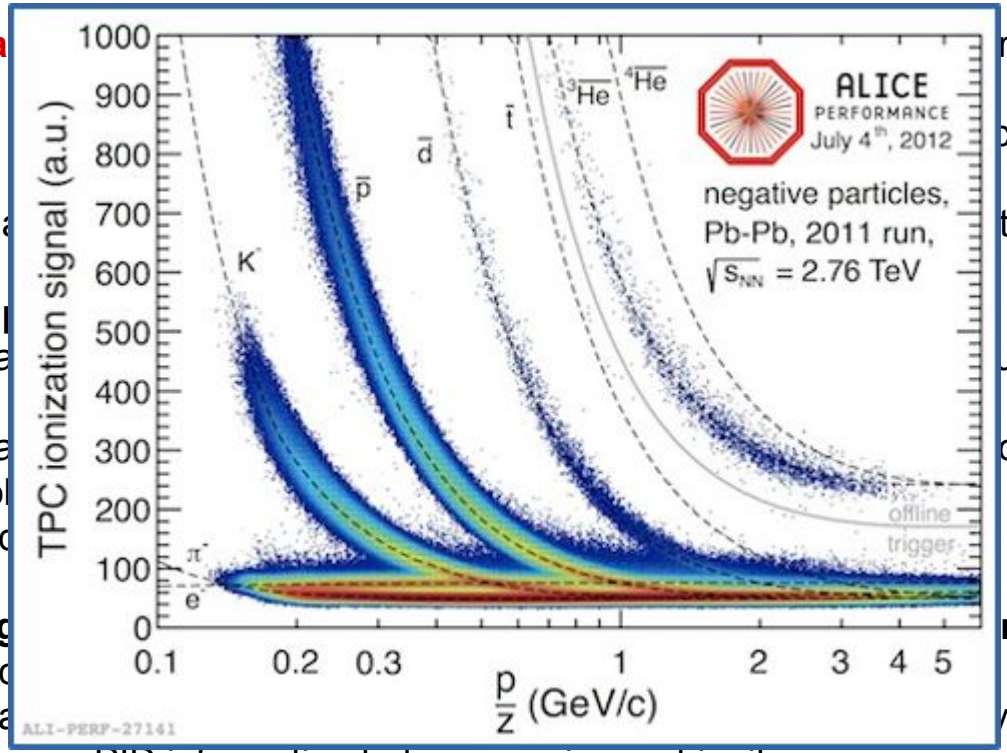
… charge units]

→ mea… tta) → $p_T$

**Exam**…
1) wha… ume z=1, B=0.5 T)?

2) wha… cated 4 m away from the col…

**Ambig**… **m *R* is $p_T$/z )** :
- …
- a… y rare
  - PID information helps removing ambiguity



# Introductory concepts: Rapidity and pseudorapidity

Velocities in different reference systems:

non-relativistic case: $v = v_1 + v_2$ relativistic case: $\beta = \frac{\beta_1 + \beta_2}{1 + \beta_1 \beta_2}$

Rapidity: $y = \tanh^{-1}\beta = \frac{1}{2}\log\left(\frac{1+\beta}{1-\beta}\right)$ relativistic and non-relativistic cases: $y = y_1 + y_2$

(in the non-relativistic limit: $y = \beta$ )

Just need to remember what rapidity and pseudorapidity refer to

In accelerator physics, one usually defines rapidity along the beam direction (z axis)

$$y = \frac{1}{2}\log\frac{E + p_z}{E - p_z}$$

**it depends on particle mass**, in the sense that you need to know both momentum and energy or one of the two and the particle species (→ mass) to calculated it
**But can be used to define *y* for any 4-vector, even that of a system of particles**

And a connected variable, pseudorapidity:

does not depend on particle mass, it has a direct interpretation in terms of "direction" in the apparatus

$$\eta = \frac{1}{2}\log\frac{p + p_z}{p - p_z} = -\log\tan\frac{\vartheta}{2}$$

$\eta \approx y$ when E>>m

# Introductory concepts: Rapidity and pseudorapidity

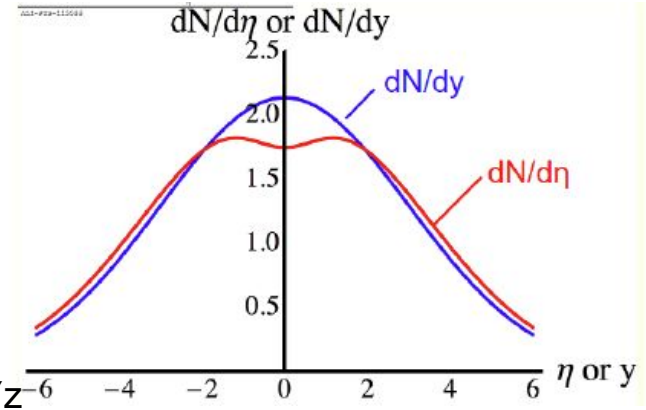
Velocities in different reference systems:

non-relativistic case: $v = v_1 + v_2$ relativistic case: $\beta = \frac{\beta_1 + \beta_2}{1 + \beta_1 \beta_2}$

Rapidity: $y = \tanh^{-1}\beta = \frac{1}{2}\log\left(\frac{1+\beta}{1-\beta}\right)$

relativistic and non-relativistic cases: $y = y_1 + y_2$

(in the non-relativistic limit: $y = \beta$ )




In accelerator physics, one usually defines rapidity along the beam direction (z-axis)

$$y = \frac{1}{2}\log\frac{E + p_z}{E - p_z}$$

**it depends on particle mass**, in the sense that you need to know both momentum and energy or one of the two and the particle species (→ mass) to calculated it
**But can be used to define *y* for any 4-vector, even that of a system of particles**

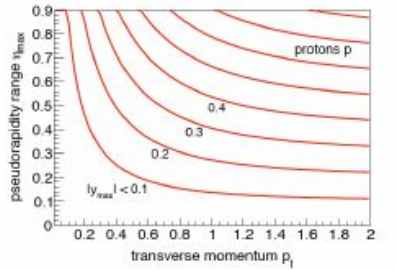
→ Always keep in mind: Rapidity and pseudo-rapidity are not the same, especially at low transverse momenta!

And a connected variable, pseudorapidity:

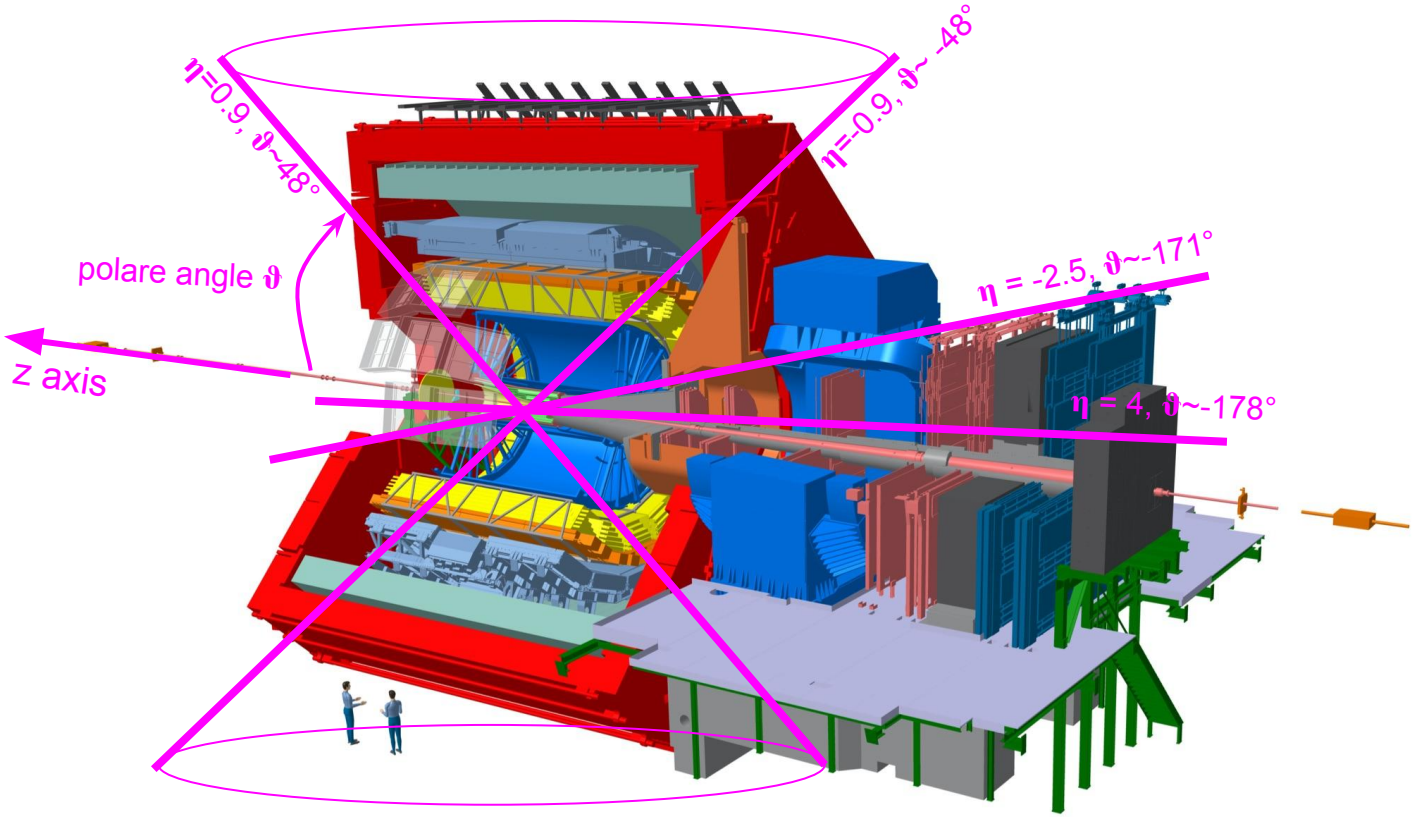
$$\eta = \frac{1}{2}\log\frac{p + p_z}{p - p_z} = -\log\tan\frac{\vartheta}{2}$$

does not depend on particle mass, it has a direct interpretation in terms of "direction" in the apparatus

$\eta \approx y$ when E>>m



# Rapidity and pseudorapidity, and transverse momentum



# First part: understand single track variables

# Single track, output data format

ALICE has a dedicated software based on C++, ROOT, and other packages, among which Apache Arrow (https://arrow.apache.org/)

Main need: most signal under study are rare but with limited possibility of being selected via “online” triggers
→ large data samples to be analysed

Data output based on flat tables
- fast analysis
- modular handling of information: more info needed → new table added

Typically:
- tables with track infos
- tables with collision infos

# Single track, output data format

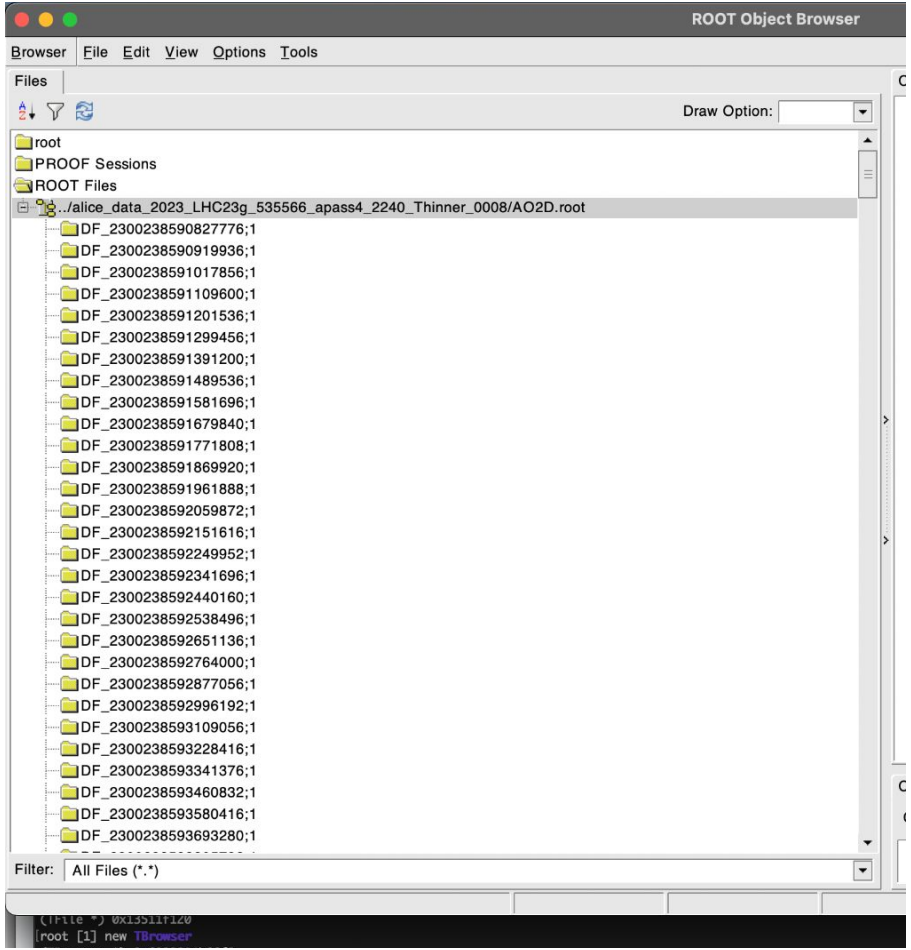
Data taken in “continuous readout” (peculiar of ALICE and few other experiments):

- no central event trigger sent by a dedicated detector to all subdetectors for starting readout.
- data continuously streamed
- events reconstructed in a second step
  - track reconstruction → primary vertices (~collision points) identified (position, time) by grouping of tracks in space and time

Time beat for data reconstruction: dataframe
**Dataframe (DF)**: bunch of data readout contiguously in time that get reconstructed together (ALICE specific concept, you need to know it only for practical purposes!)

Subsequent procedures of “thinning” for data reduction (including track selections).
Final structures: files (called AO2D.root) with flat trees (tables) organised in DF.

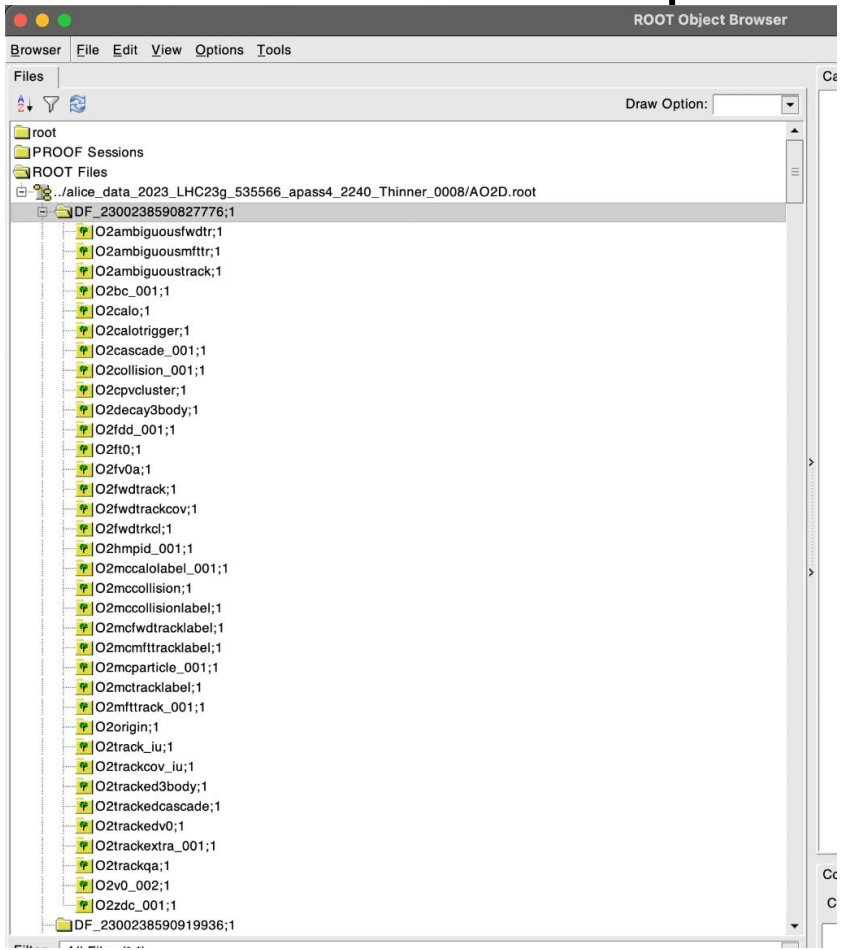
# ALICE data format in practice




Let's open a AO2D.root file with ROOT and look at it with a TBrowser

We see several directories, each corresponding to a different Data Frame

# ALICE data format in practice



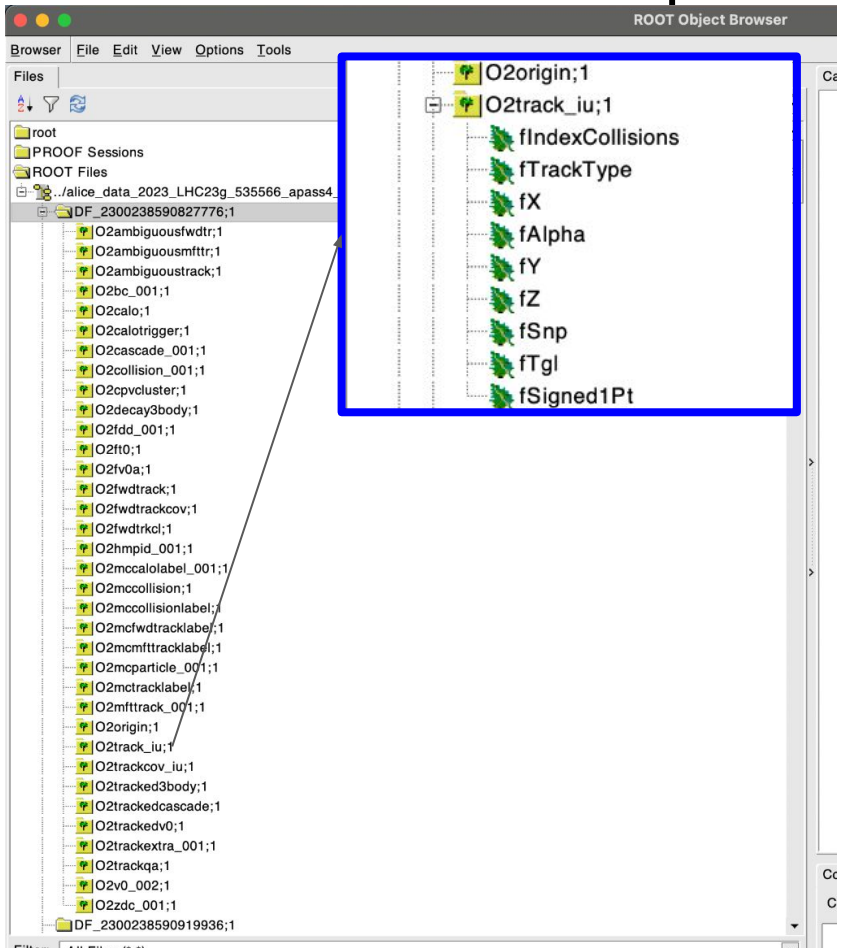

Inside a AO2D.root file ( I open one with ROOT and look at it with a TBrowser)

We see several directories, each corresponding to a different Data Frame

If we open one, we see the list of tables

Their names start with O2: name of ALICE software, irrelevant for you…

# ALICE data format in practice




Inside a AO2D.root file ( I open one with ROOT and look at it with a TBrowser)

We see several directories, each corresponding to a different Data Frame

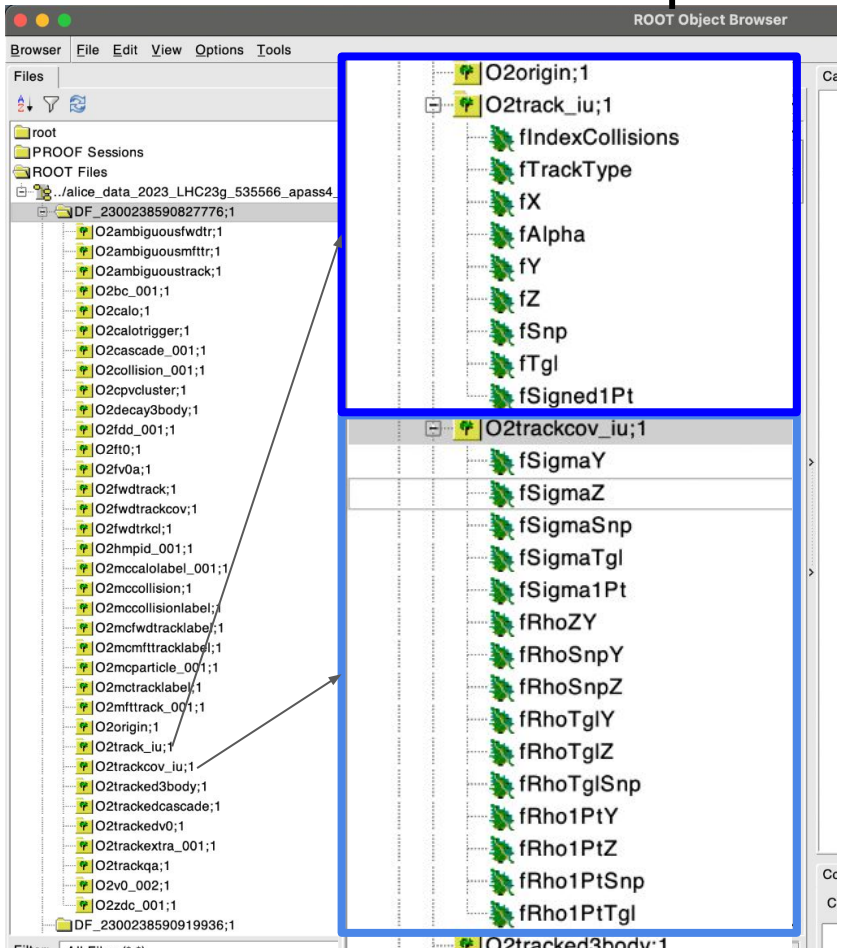
If we open one, we see the list of tables

Their names start with O2: name of ALICE software, irrelevant for you…

Main tables:

- table with track parameters

# ALICE data format in practice




Inside a AO2D.root file ( I open one with ROOT and look at it with a TBrowser)

We see several directories, each corresponding to a different Data Frame

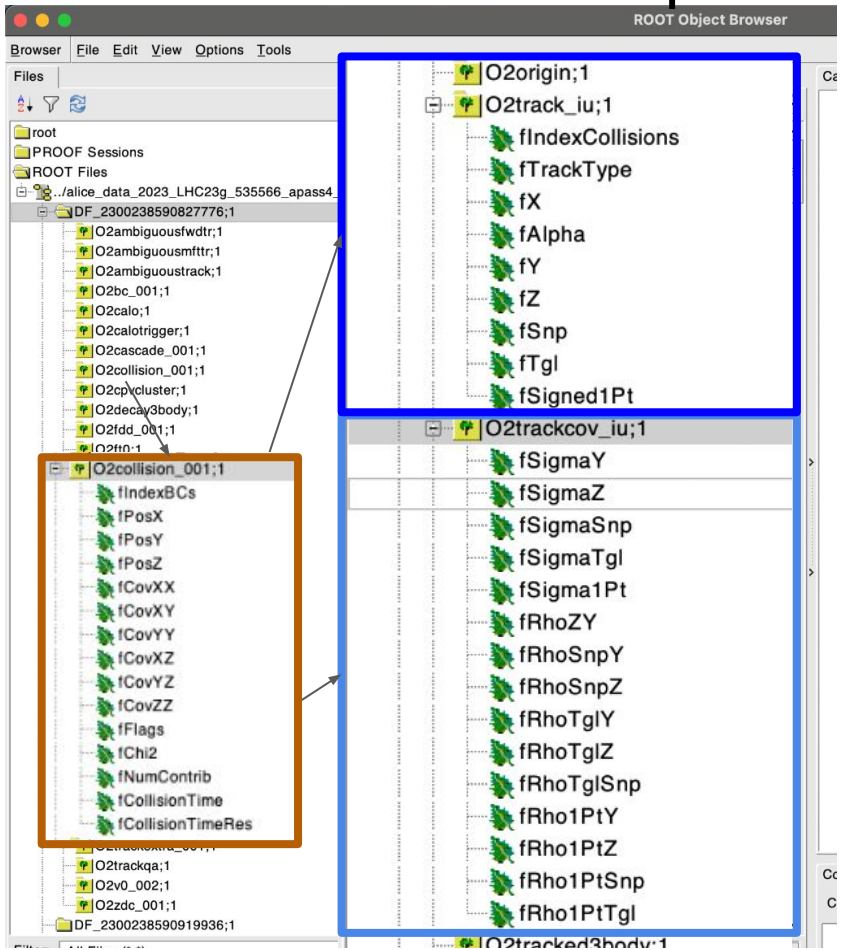
If we open one, we see the list of tables

Their names start with O2: name of ALICE software, irrelevant for you…

Main tables:

- table with track parameters
- related table with covariance matrix of track parameters

# ALICE data format in practice




Inside a AO2D.root file ( I open one with ROOT and look at it with a TBrowser)

We see several directories, each corresponding to a different Data Frame

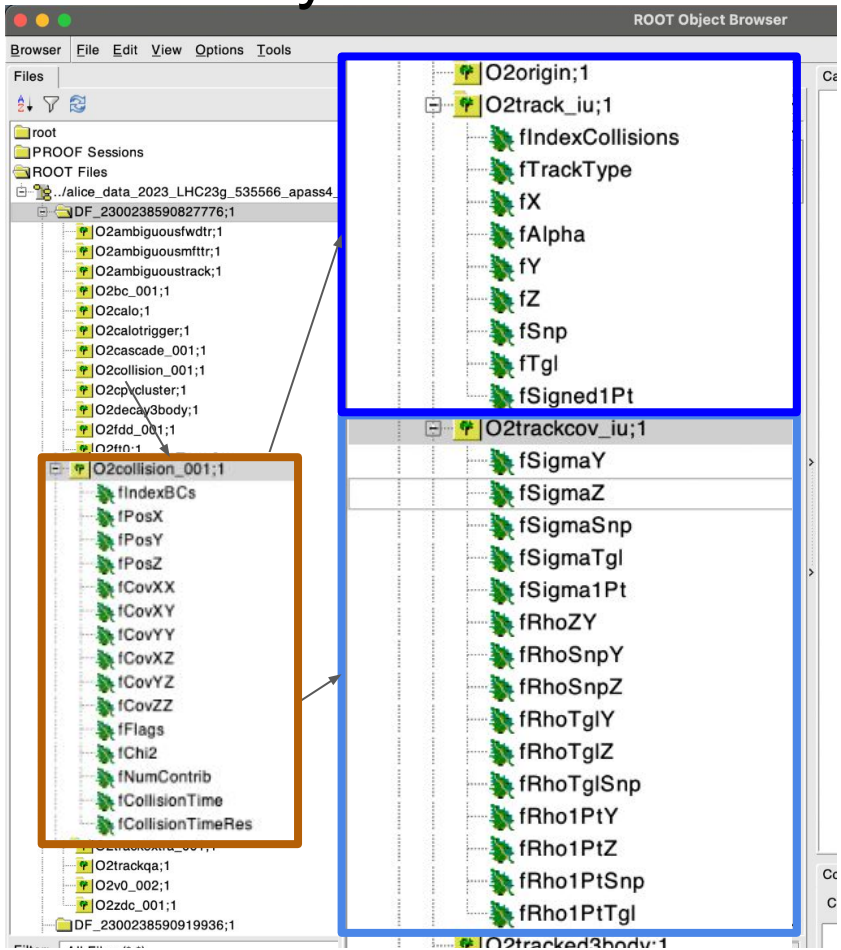
If we open one, we see the list of tables

Their names start with O2: name of ALICE software, irrelevant for you…

Main tables for us:

- table with track parameters
- related table with covariance matrix of track parameters
- table with collision and primary vertex information

# Practically…




Inside a AO2D.root file ( I open one with ROOT and look at it with a TBrowser)

We see several directories, each corresponding to a different Data Frame

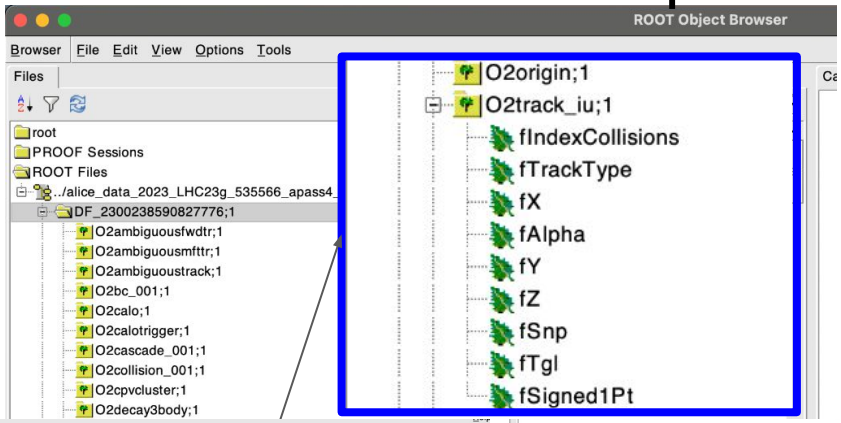
If we open one, we see the list of tables

Their names start with O2: name of ALICE software, irrelevant for you…

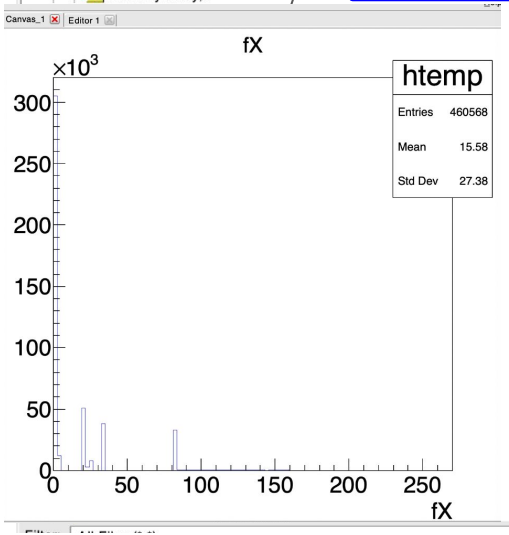
Main tables for us:

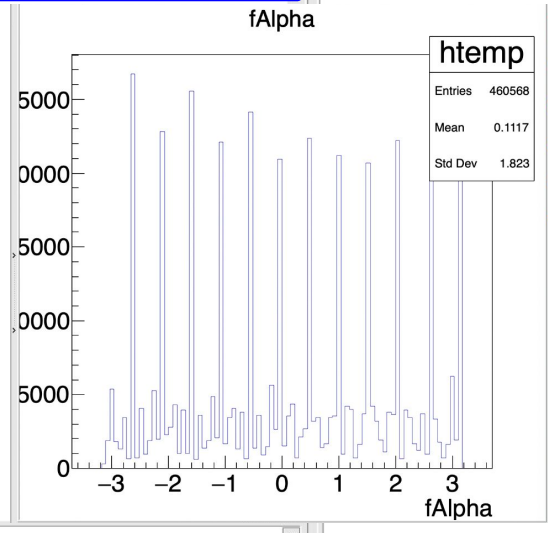
- table with track parameters
- related table with covariance matrix of track parameters
- table with collision and primary vertex information
- ... tables with PID information (later)

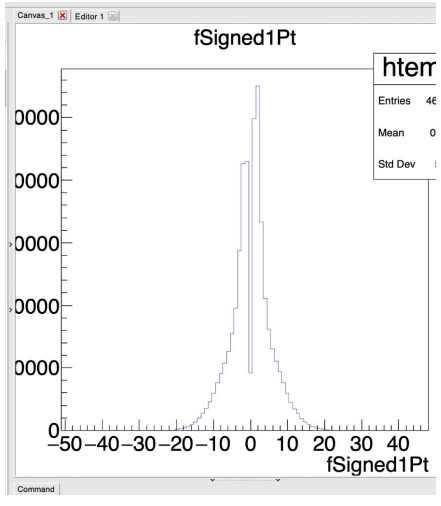
# ALICE data format in practice



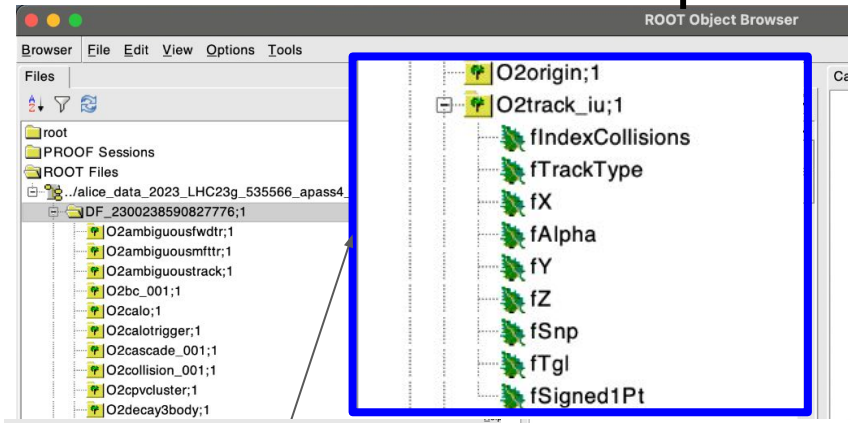

We can look at the variables, even by just clicking on them in the TBrowser or in whatever way you prefer



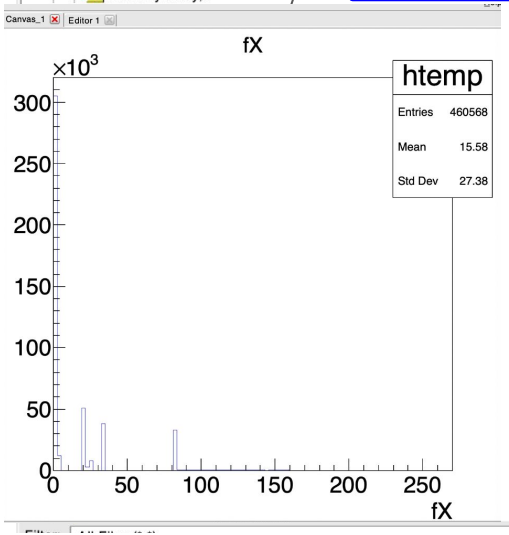



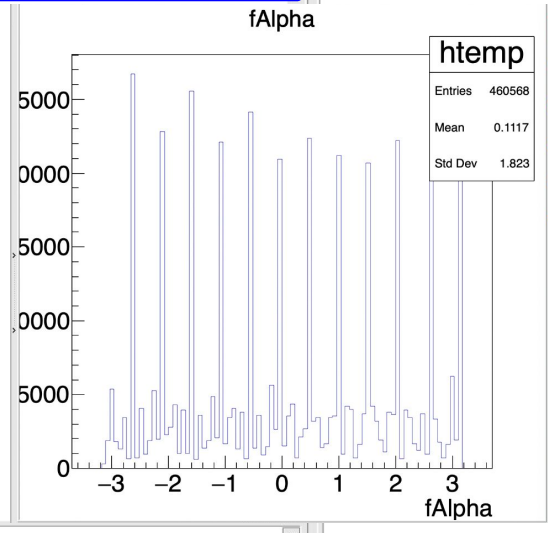
# ALICE data format in practice

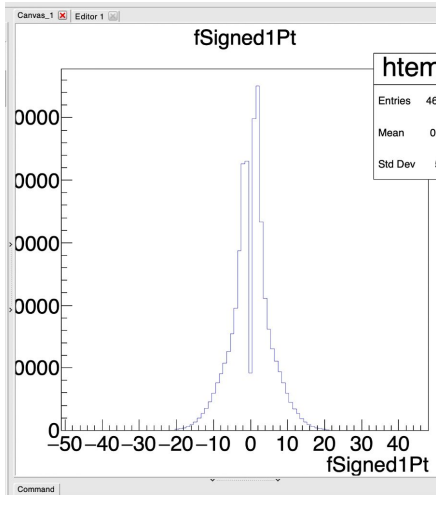



Note the name of the tree: O2track_iu
"iu" stands for "innermost update" point →

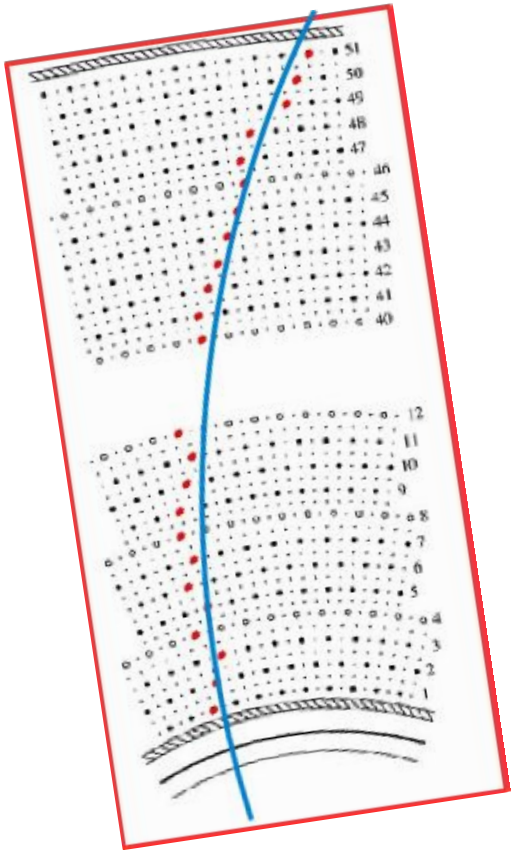
- these are the track parameters at the latest experimental point: this is not what we need (discussed later)
- in the **tracking reference system**



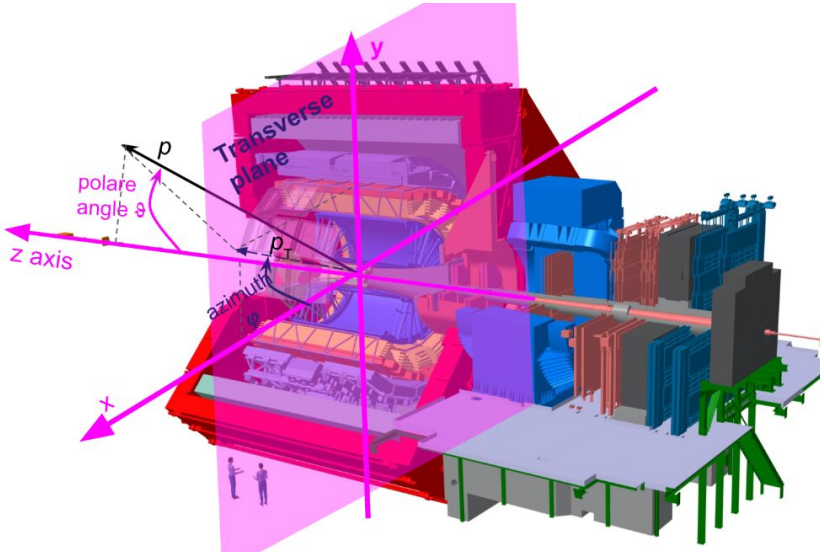



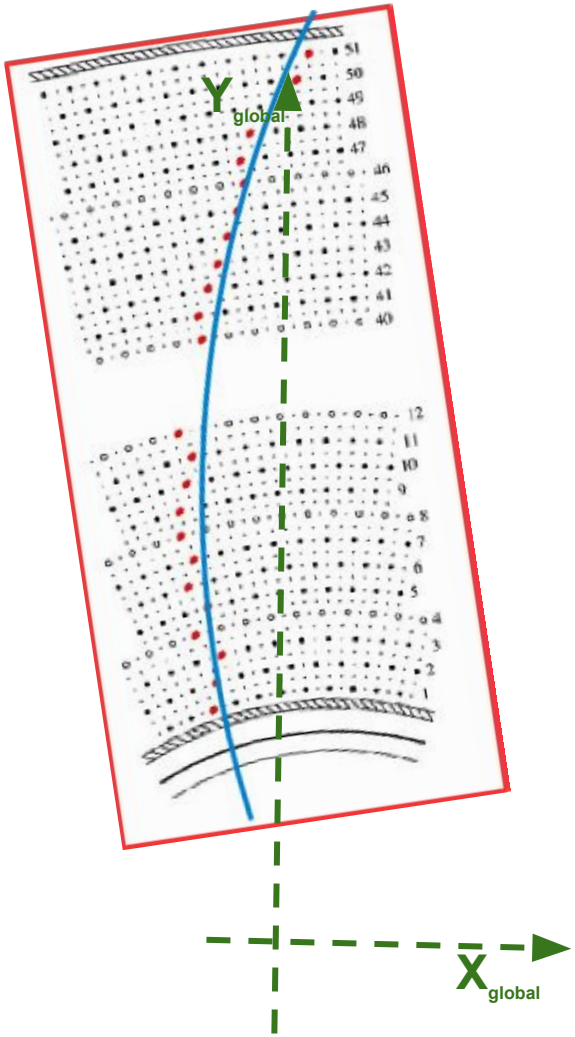



# Tracking and global reference systems

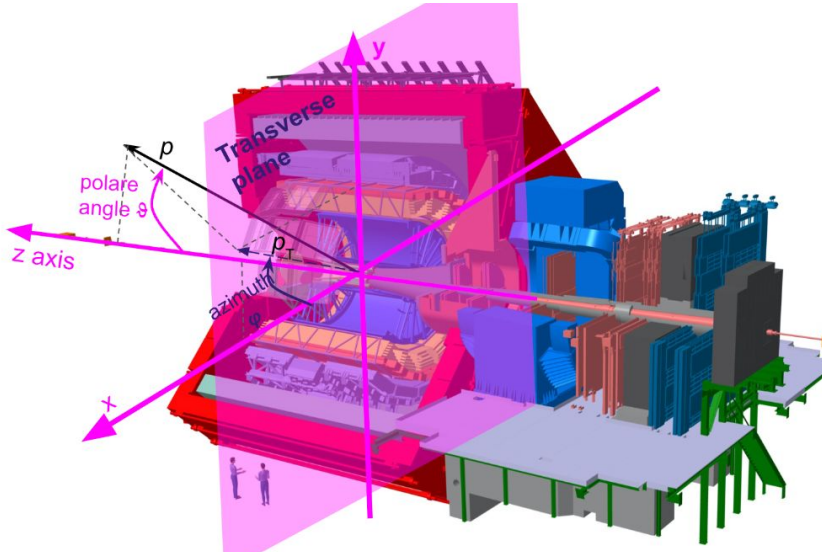


# Tracking and global reference systems

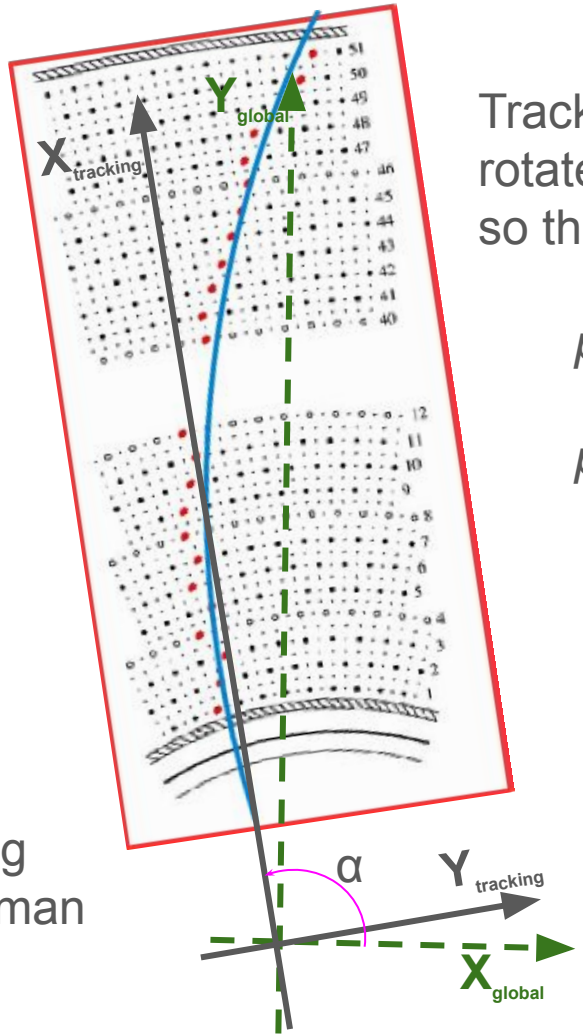


# Tracking and global reference systems




Why is it useful to use a track-dependent reference system?
Just because calculations related to tracking are easier (allows for approximation in Kalman filter used)

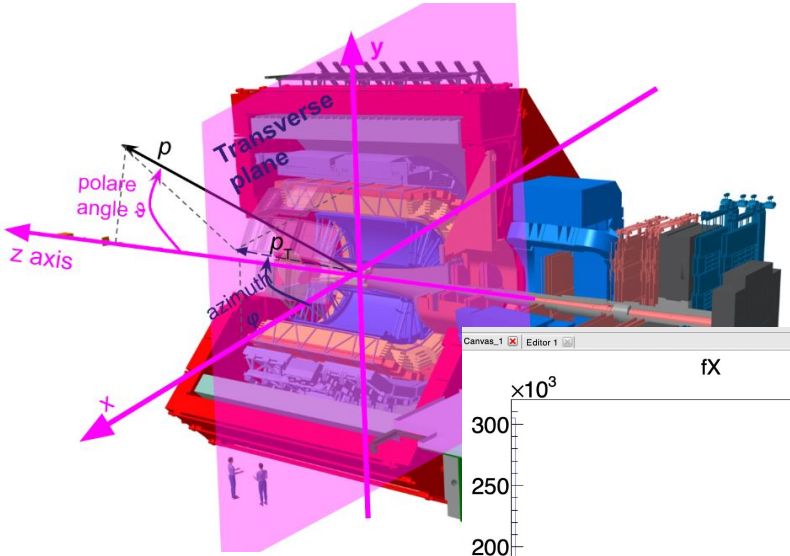


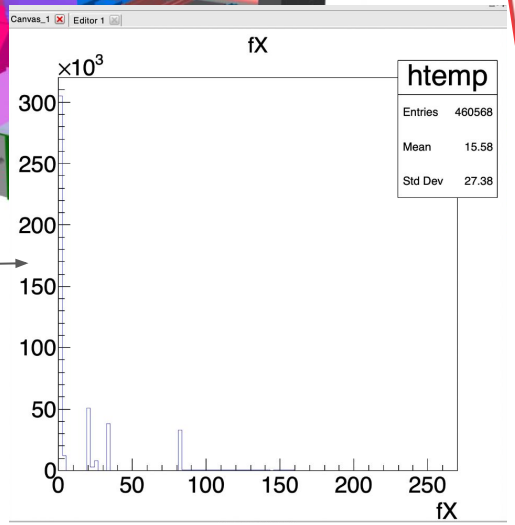

Tracking ref. system: rotated by an angle $\alpha$ so that

$$p_{x\ global} = p_T \cos\alpha$$

$$p_{y\ global} = p_T \sin\alpha$$

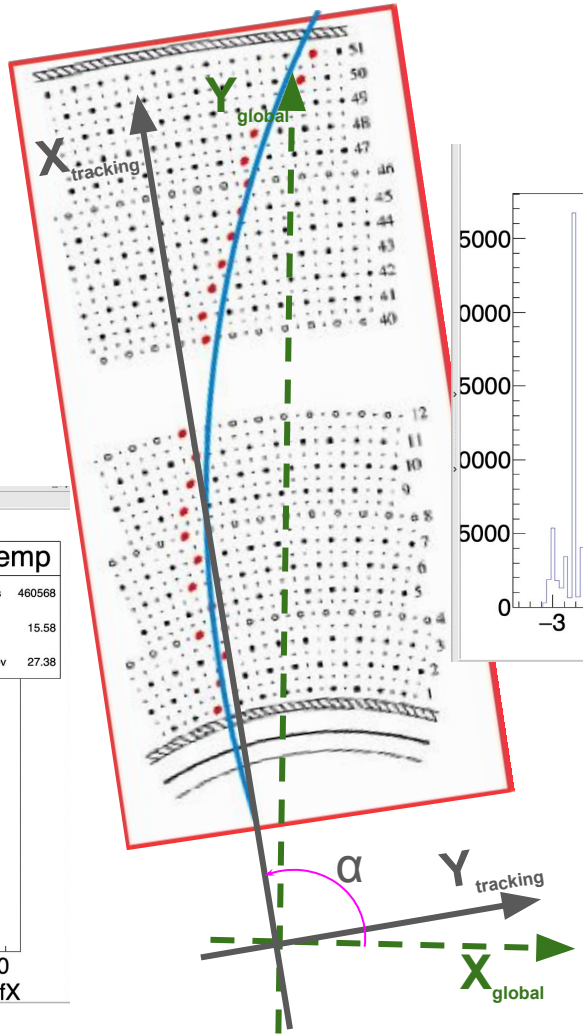
# Tracking and global reference systems

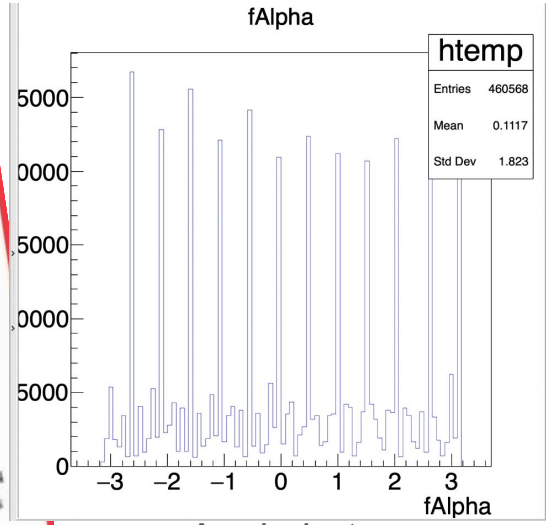



**fX = $X_{tracking}$**

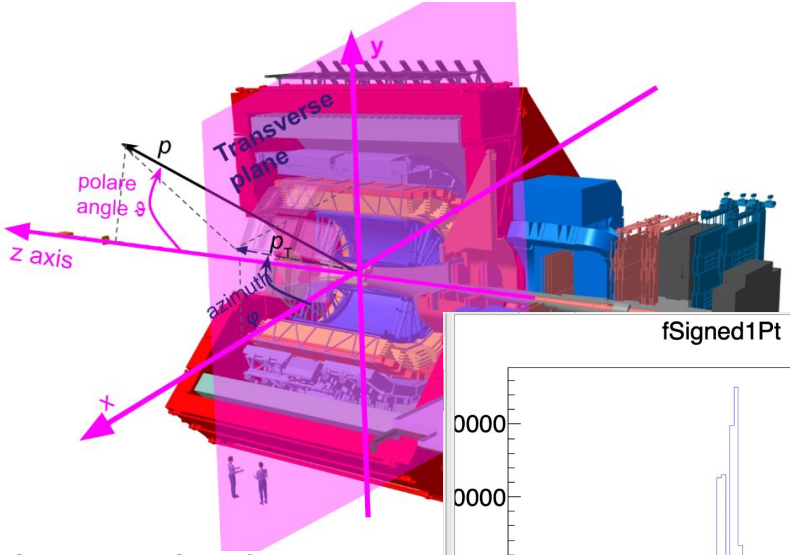
not really a track parameter, is the “free” reference coordinate at which the track parameters are reported




Angle between global and tracking system

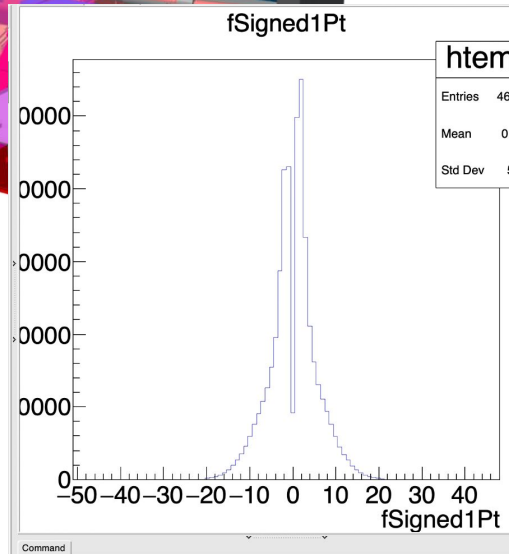
# Tracking and global reference systems

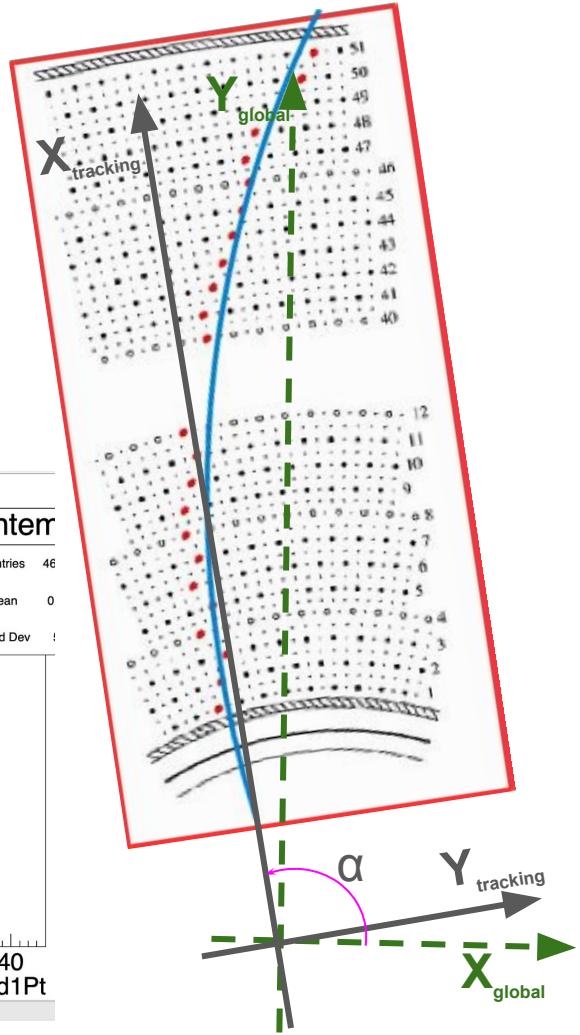



charge sign/$p_T$
N.B. $1/p_T$

$\propto$ curvature

same in global and tracking system



# Tracks at primary vertex

In a AO2D.root file there is too much information for our needs.
Moreover, having the tracks at the “innermost update” (IU) point is not that useful for us.

Usually, for most analyses and physics observables, **the parameters (e.g. the momentum components)** of a given track **must be evaluated where the original particle is produced**:

- **collision point** ( → “**primary vertex**”): vast majority of produced particles
- **decay point of parent particle**, for particles from weak decays, like daughters of
  - strange hadrons: $K^0_s$ ($c\tau$ ~ 2.68 cm), $\Lambda$ ($c\tau$ ~ 7.8 cm)
  - charm hadrons: $D^+$ ($c\tau$ ~ 310 μm), $\Lambda_c^+$ ($c\tau$ ~ 60 μm)
  - beauty hadrons: $B^+$ ($c\tau$ ~ 491 μm), $\Lambda_b^+$ ($c\tau$ ~ 441 μm)

N.B. timescale of strong and electromagnetic decay of elementary particles: $10^{-24}$ - $10^{-20}$ s
($\tau$~$8\times10^{-17}$ s for $\pi^0$, $c\tau$ ~ 25 nm) → decay point basically coincides with primary vertex

We save (in ALICE) tracks at the IU point because of possible ambiguities in the association of tracks to events (primary vertices).

# Tracks at primary vertex

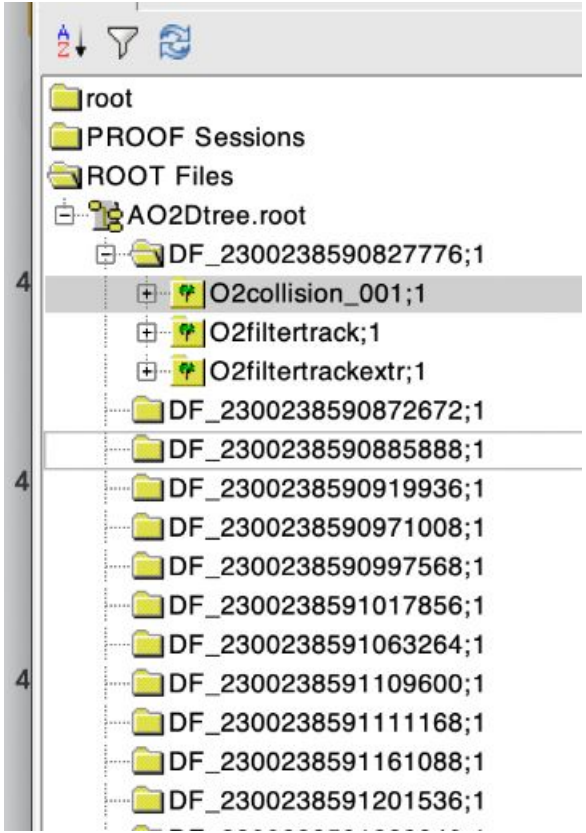
What we primarily need are the track parameters at the primary vertex (PV).

**Propagation from IU to PV is not as trivial as it may seem**: for best description of trajectories the material crossed must be taken into account: average energy loss (deterministic correction), multiple coulomb scattering (accounted for in track covariance matrix)
→ requires detailed knowledge of detector geometry + accurate modeling of interaction with material.
→ we can't do this without ALICE specific software

**Data reduction/filtering operation**
I prepared you files with

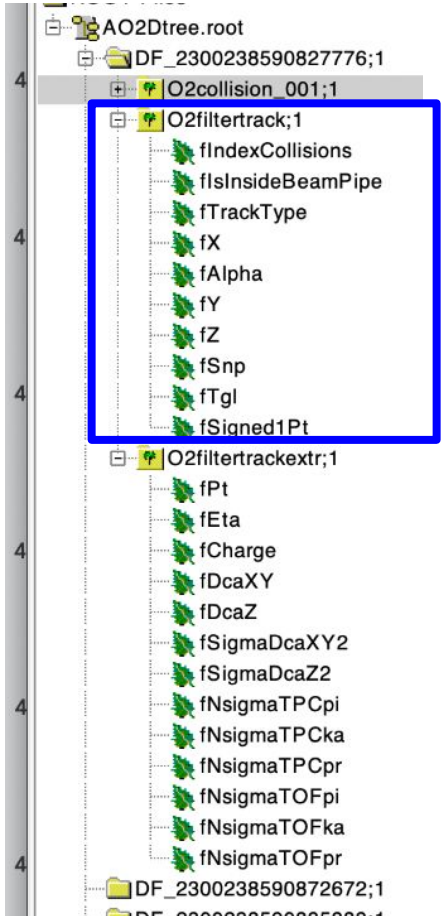
- tracks already propagated at the PV, i.e. at the point of closest approach (PCA) to the PV
  - tracking reference system, with X axis parallel to track direction (momentum vector) at PV
- selection of tracks (e.g. discaring tracks without points in the ITS, $p_T$>0.3 GeV/c)
- reduced info stored

# The data files you will use




Same data-frame based structure than original files but only three tables inside:

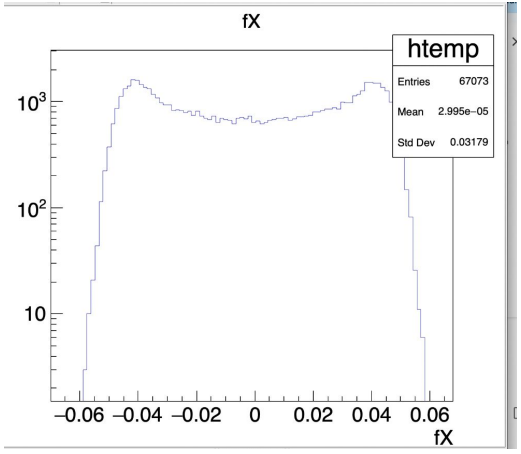
# The data files you will use

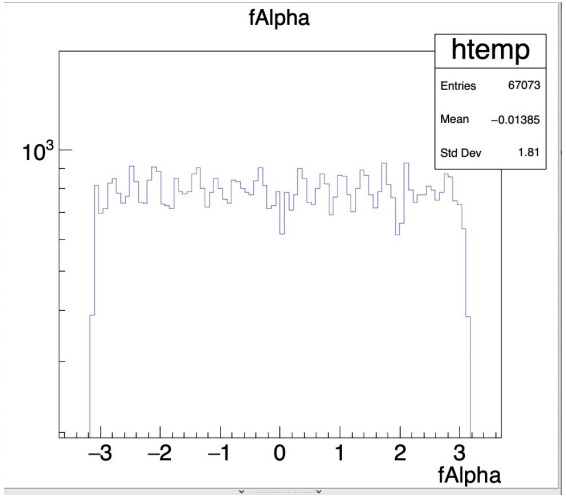



Same data-frame based structure than original files but only three tables inside:

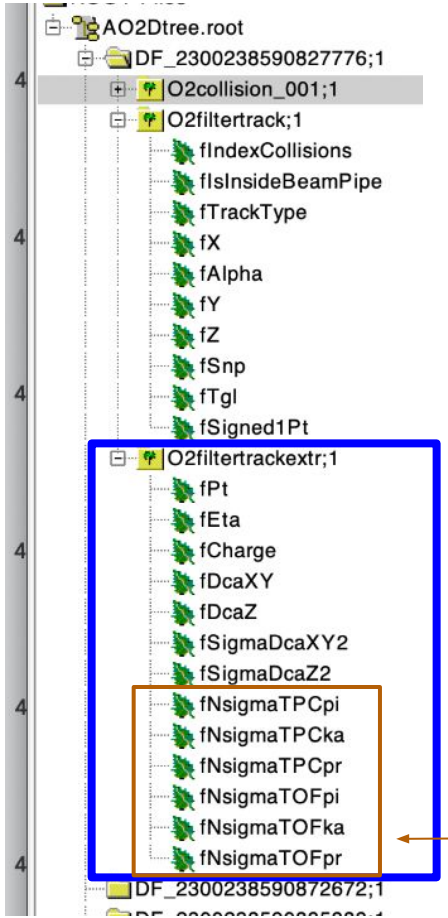
- **O2filtertrack**: same track table presented before but now track parameters are at the PCA to the PV

  Still in tracking reference system!



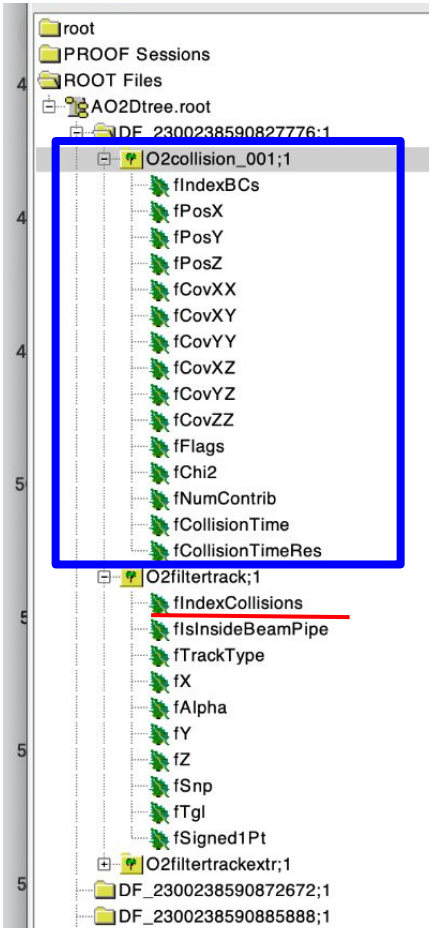
# The data files you will use




Same data-frame based structure than original files but only three tables inside:

- O2filtertrack: same track table presented before but now track parameters are at the PCA to the PV
- **O2filtertrackextr**: table with further track parameters you need
  - fEta = $\eta$ (pseudorapidity)
  - fDcaXY = track impact parameter (w.r.t. the primary vertex) in the XY plane
  - fDcaZ = track distance to the the PV at the XY DCA pont (~ impact parameter along z)
  - fSigmaDcaXY2 = variance of fDCAxy
  - TPC and TOF PID variables

# The data files you will use




Same data-frame based structure than original files but only three tables inside:

- O2filtertrack: same track table presented before but now track parameters are at the PCA to the PV
- O2filtertrackextr: table with further track parameters you need
- **O2collision_001**: table with PV information
  - coordinates (in global ref. system)
  - covariance matrix, chi2 of PV fit, ….
    (no selection/data reduction of this table since number of collisions << number of tracks)

**Most important**: **fIndexCollisions** in track table is the index of the collision to which the track is associated

→ **variable which allows to link the two tables**


let’s start playing with the data….

# Let’s get familiar with the data

Take 1 file, access the trees and obtain what follows:

**Goal 1:**

- produce histogram of track pt distribution, first from a single DF,  than using all DF in the file. Check that $p_T$ and abs(fSigned1Pt) give the same.

**Goal 2:**

- produce plots with TPC and TOF PID information

**Goal 3:**

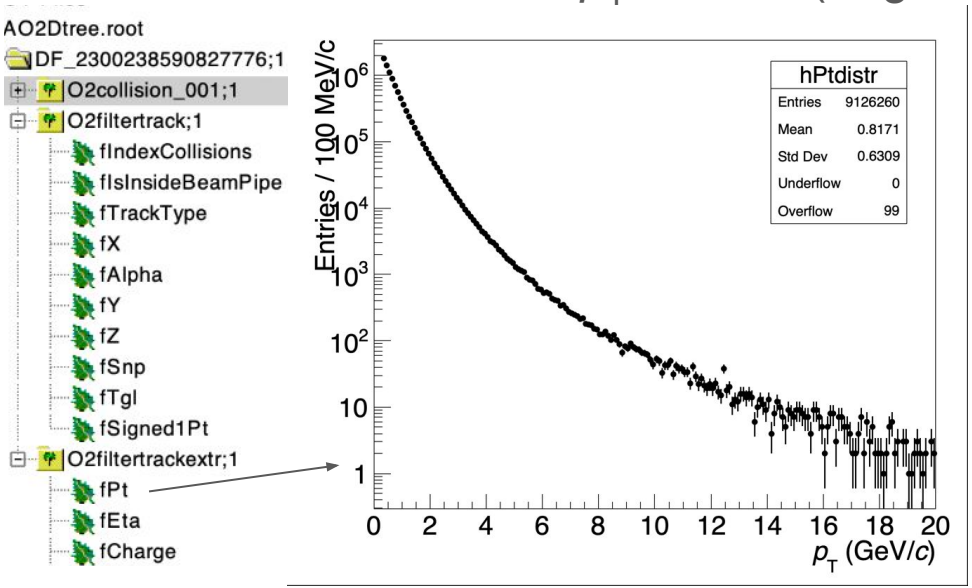
- calculate dcaXY variable from “local coordinates” (aka those of tracking reference system) and check your calculation comparing what you get with the available dcaXY
  - need to handle primary vertex coordinates (in global reference system) stored in collision table

# Let’s get familiar with the data

Take 1 file, access the trees and obtain what follows:

**Goal 1:**

- produce histogram of track pt distribution, first from a single DF,  than using all DF in the file. Check that $p_T$ and abs(fSigned1Pt) give the same.




I used Root but you can use what you prefer!

Jupyter notebook is more than welcome!

Check that you get the same counts!

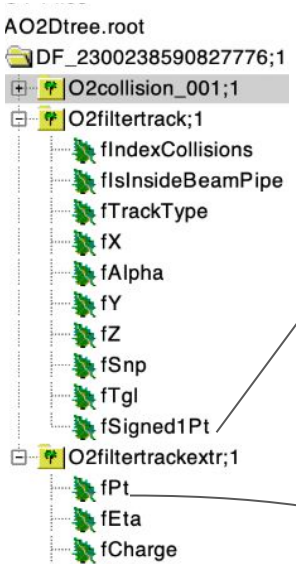
# Let’s get familiar with the data

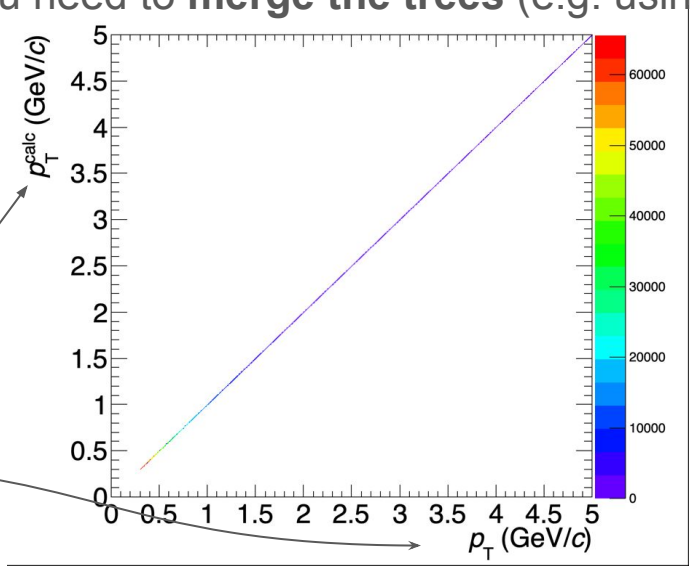
Take 1 file, access the trees and obtain what follows:

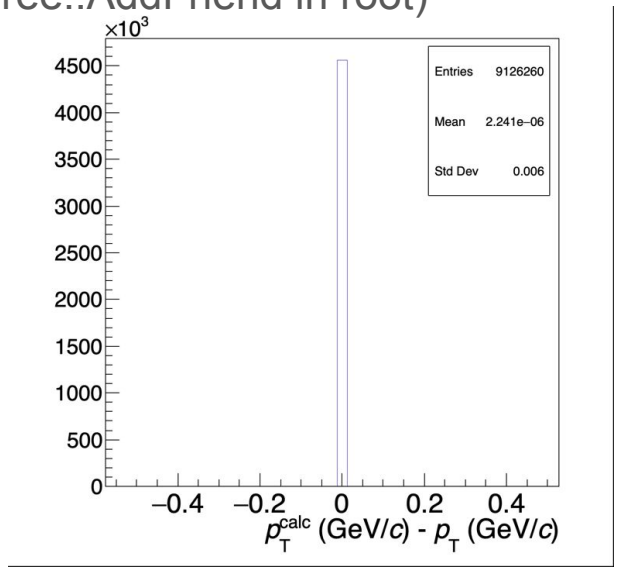
**Goal 1:**

- produce histogram of track pt distribution, first from a single DF, than using all DF in the file. Check that $p_T$ and abs(fSigned1Pt) give the same.

You need to **merge the trees** (e.g. using TTree::AddFriend in root)



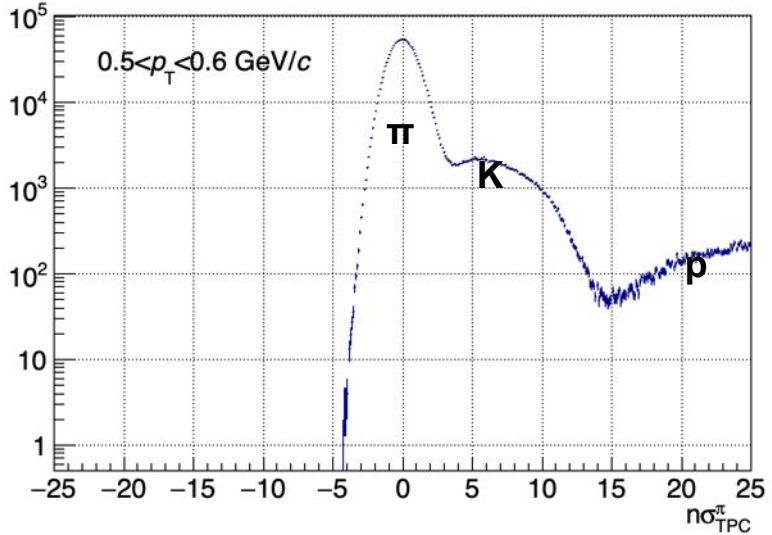



# Let's get familiar with the data

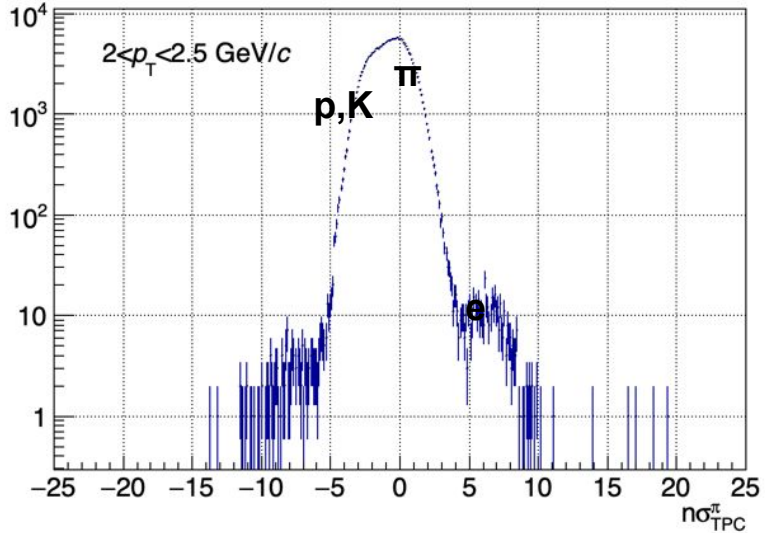
Take 1 file, access the trees and obtain what follows:

**Goal 2:**

- produce plots with TPC and TOF PID information

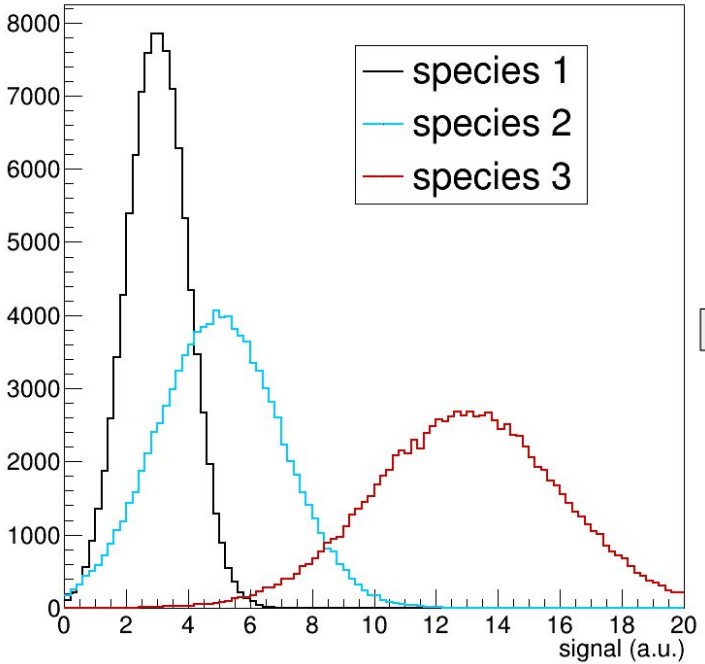



**nσ = (measured signal - expected signal)/σ**




**$n\sigma^{\pi}$ ($p_T$) → expected signal = expected signal for a pion with given $p_T$**

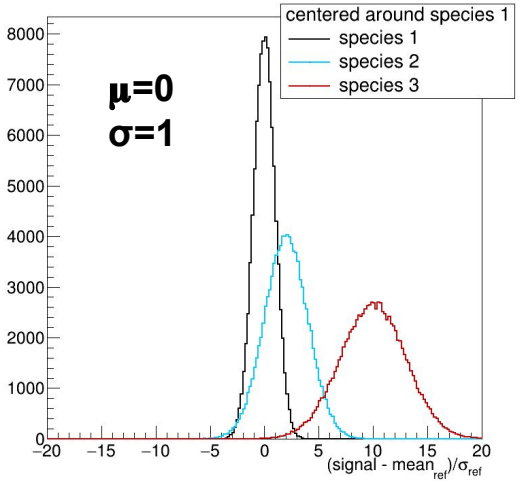
# Parenthesis: from signals to nσ

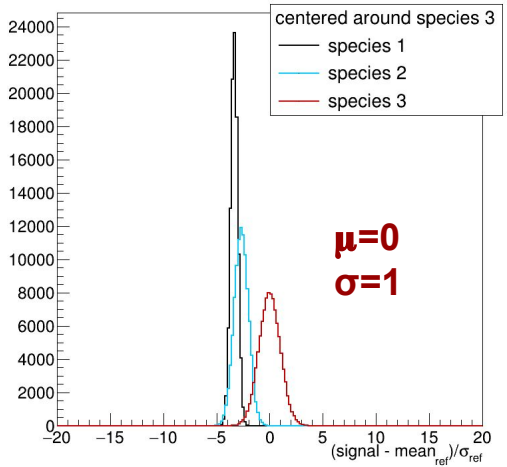



mean = mean1
sigma = sigma1

**(signal - expected)/sigma**

mean = mean3
sigma = sigma3



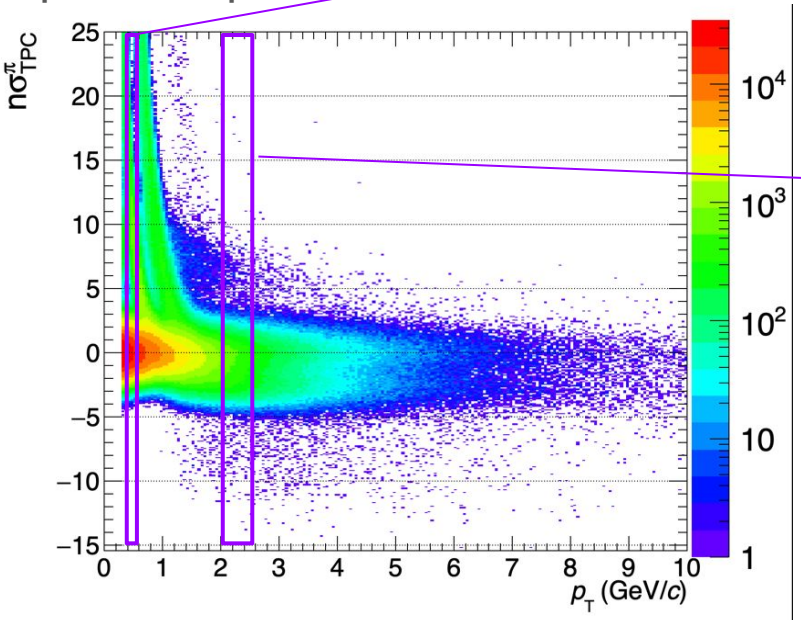



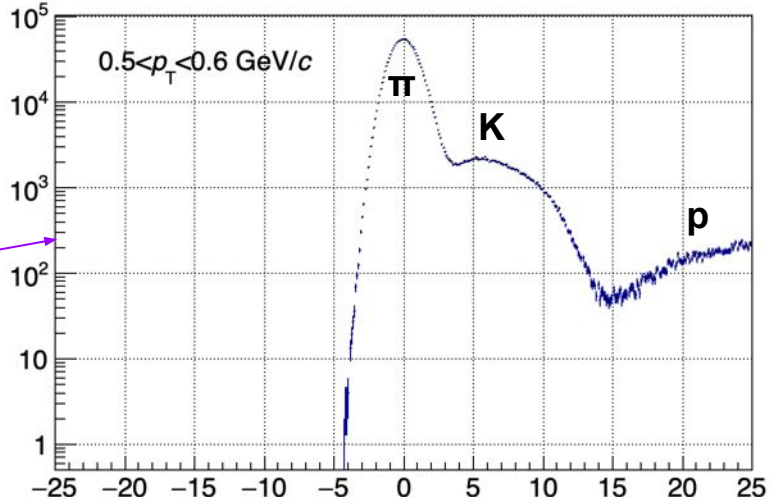
# Let's get familiar with the data

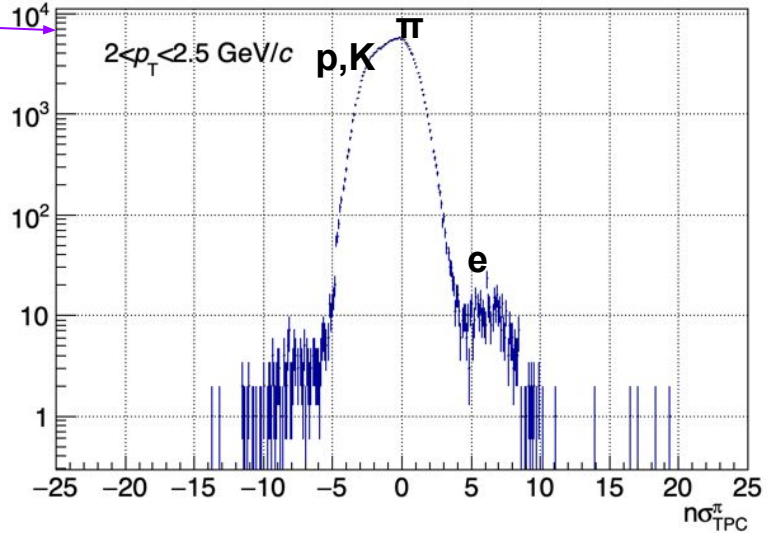
Take 1 file, access the trees and obtain what t

**Goal 2:**

- produce plots with TPC and TOF PID infc



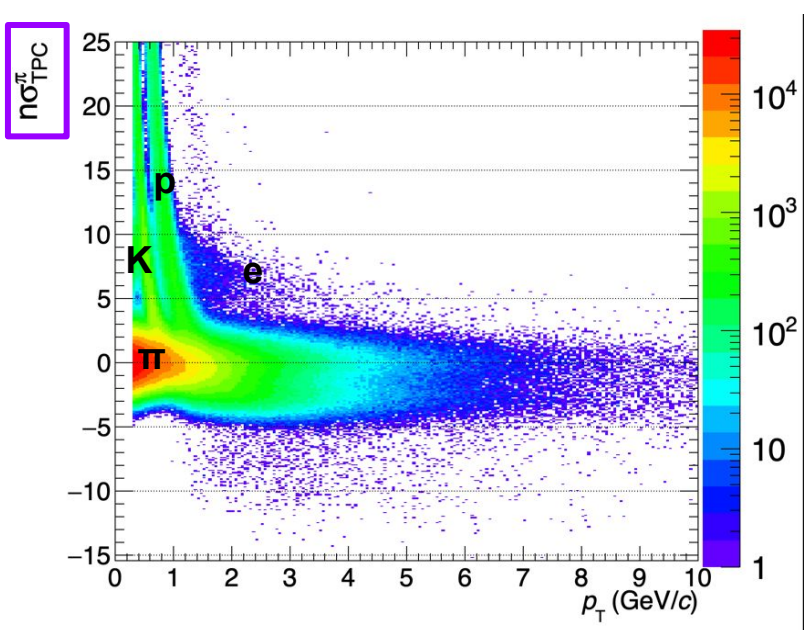



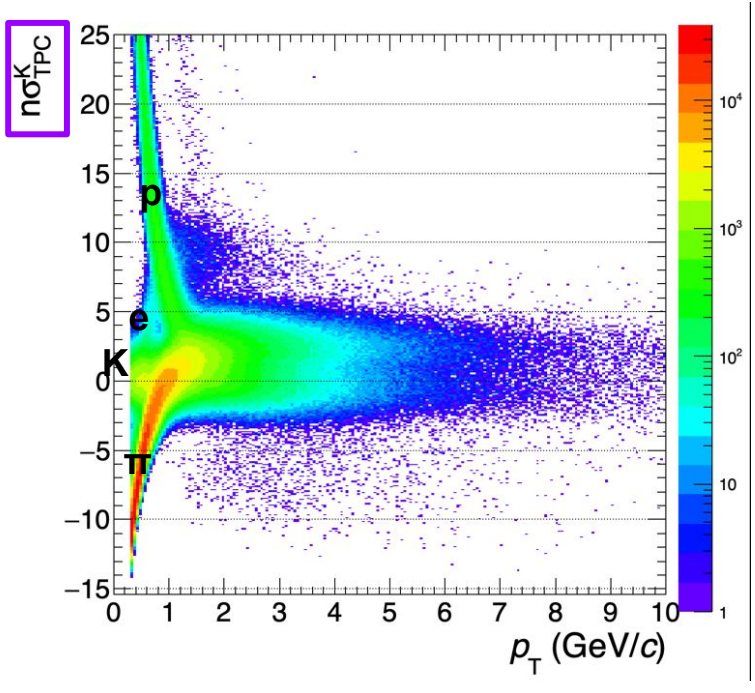
# Let’s get familiar with the data

Take 1 file, access the trees and obtain what follows:

**Goal 2:**

- produce plots with TPC and TOF PID information



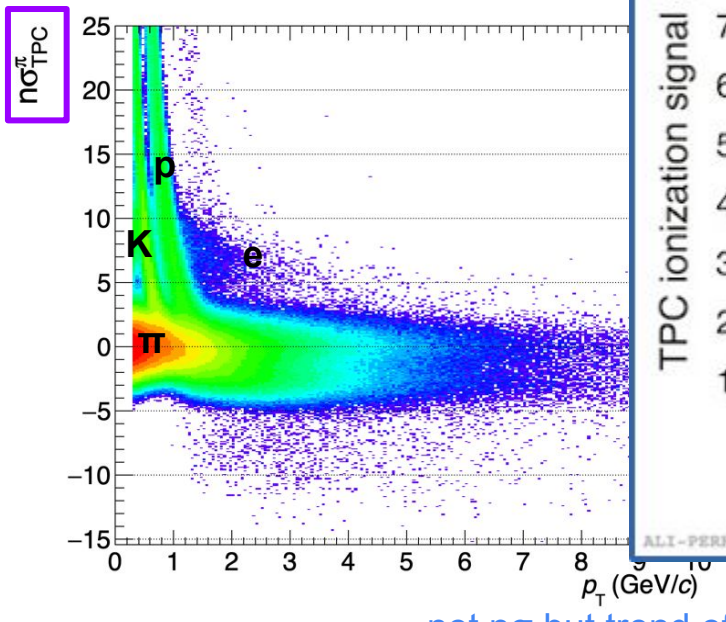



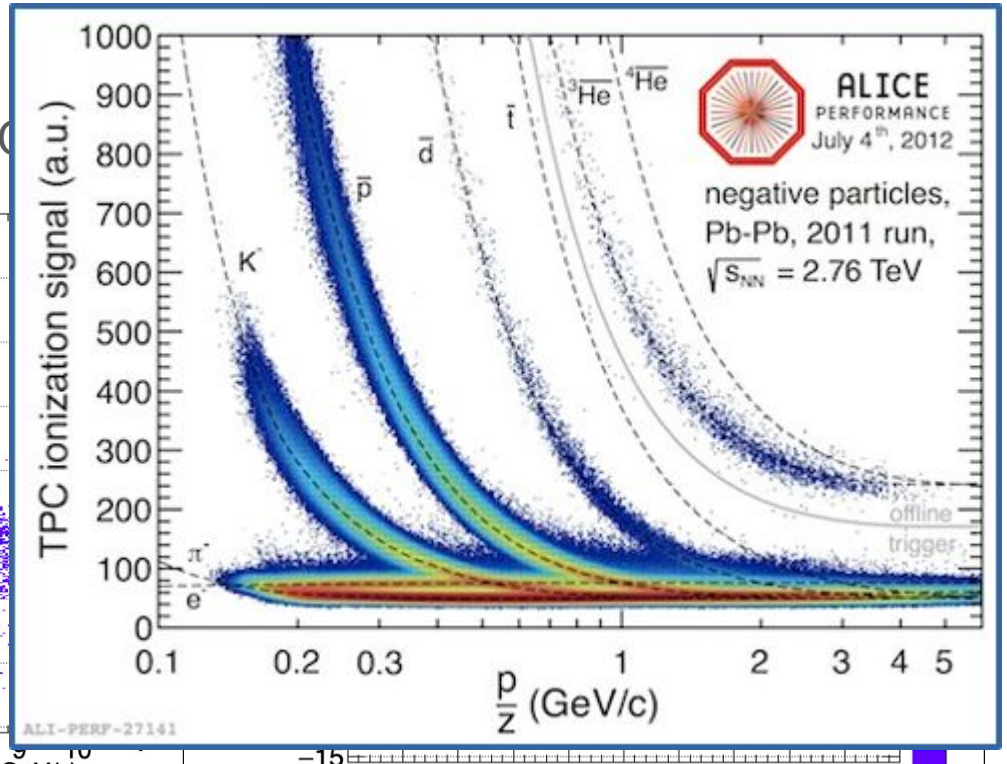
# Let's get familiar with the data

Take 1 file, access the trees and obtain what follows:

**Goal 2:**

- produce plots with TPC and TO




not nσ but trend of bands distribution similar to $n\sigma^{\pi}$ ($n\sigma^{e}$) $p_T$ (GeV/c)

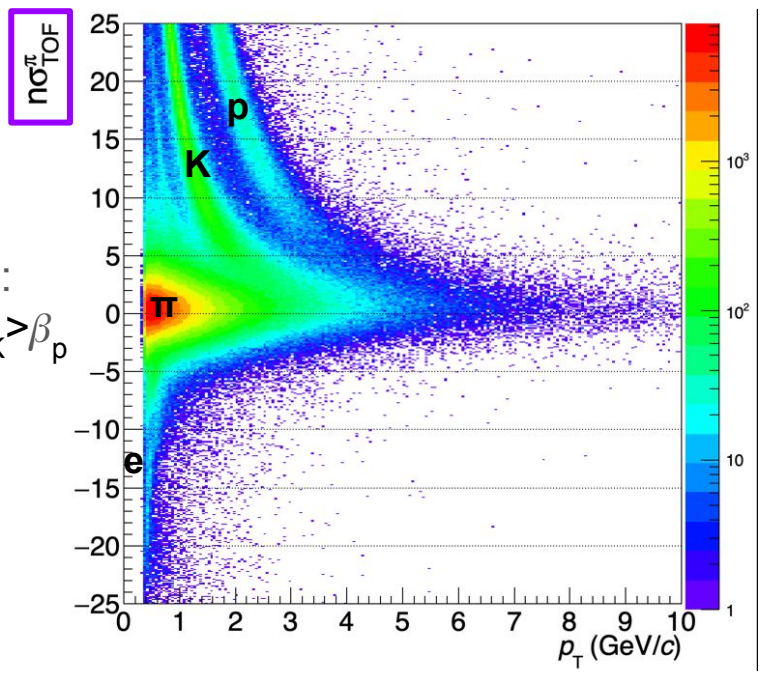
# Let's get familiar with the data

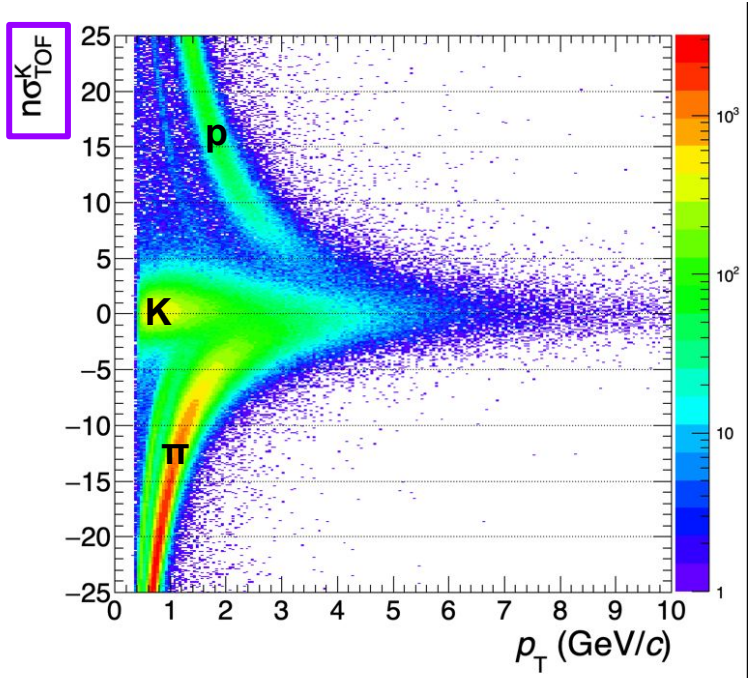
Take 1 file, access the trees and obtain what follows:

**Goal 2:**

- produce plots with TPC and TOF PID information

For a given momentum:
$1 \sim \beta_e > \beta_\pi > \beta_K > \beta_p$



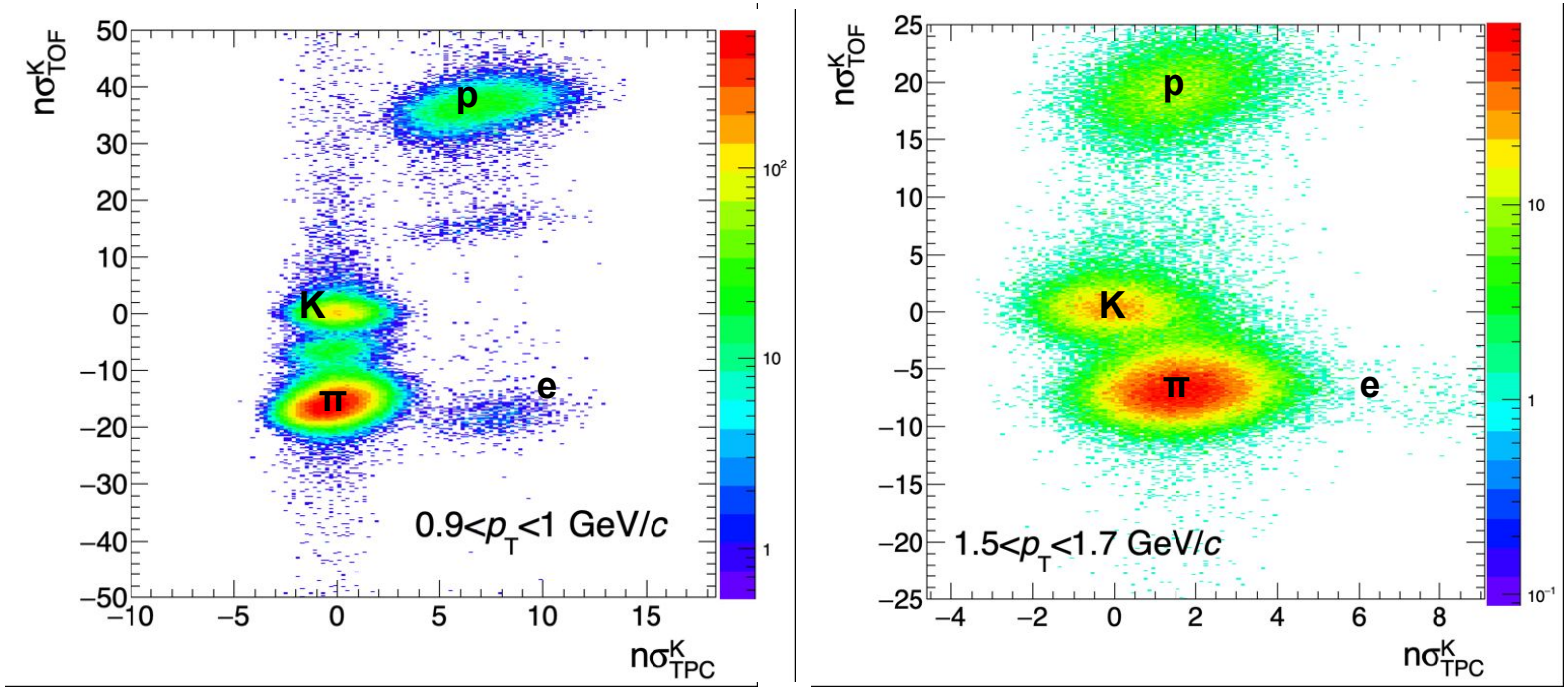



# Let’s get familiar with the data

Take 1 file, access the trees and obtain what follows:

**Goal 2:**

- produce plots with TPC and TOF PID information

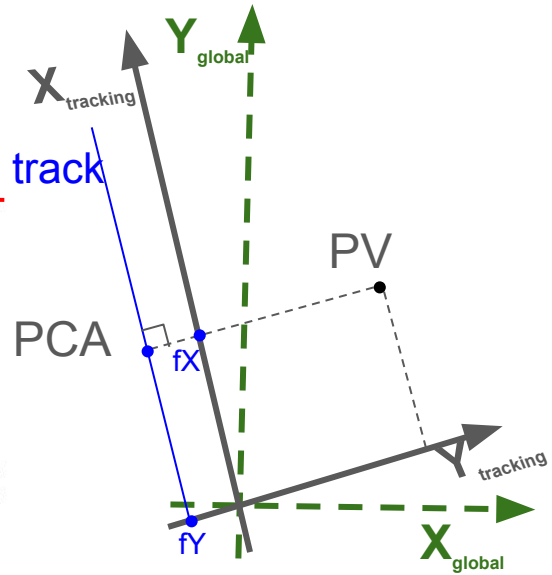


# Let's get familiar with the data

Take 1 file, access the trees and obtain what follows:

**Goal 3:**

- calculate dcaXY variable from "local coordinates" (those of tracking reference system) and check your calculation comparing what you get with the available dcaXY variable
  - in tracking system: dcaXY ~ $Y_{track}$ - $Y_{PV\ (in\ tracking\ system)}$
  - need to handle primary vertex coordinates stored (in global reference system) in collision table:


**dcaXY= (signed) distance of closest approach in XY plane**
= track impact parameter ($d_0$, see before)
= distance between PCA and PV points in XY plane
Relation dcaXY = $Y_{track}$ - $Y_{PV}$ with $Y_{PV}$ in tracking system gives proper sign convention

# Let’s get familiar with the data

Take 1 file, access the trees and obtain what follows:

**Goal 3:**

- calculate dcaXY variable from “local coordinates” (those of tracking reference system) and check your calculation comparing what you get with the available dcaXY variable
  - in tracking system: dcaXY ~ $Y_{track}$ - $Y_{PV}$
  - need to handle primary vertex coordinates stored (in global reference system) in collision table:


**Remember**:
fIndexCollisions gives the position in the O2collision tree of the primary vertex to which the track is associated

# Let’s get familiar with the data

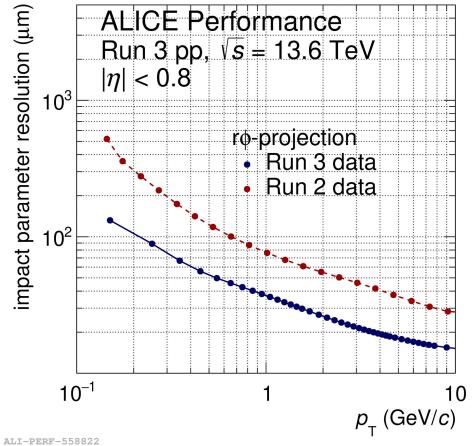
Goal 4:

- look at the dcaXY distribution in narrow pt intervals:

- take the rms of the distribution around the peak, i.e. calculate the rms’ that you get when you restrict the distribution to (approximately) the range [mean -2.5 rms, mean +2.5 rms]
  - alternatively: fit the distribution in a wider range (~5 rms) with a function composed of a Gaussian term (for the peak) and symmetric exponential tails:

- plot rms’ (or the Gaussian sigma) as a function of pt. You should get something like the figure on the right.

- repeat isolating pions, kaons, and protons



# Let's get familiar with the data

Goal 5:

- let's find $K^0_s$ signal! (mass = 497.614 MeV/$c^2$)
- $K^0_s$ decays with a BR~69.2% (see PDG) in two opposite charge pions, $K^0_s \rightarrow \pi^+\pi^-$
- relatively large lifetime, $\tau \sim 0.896 \times 10^{-10}$ s (~90 ps) $\rightarrow c\tau$ is large (2.6844 cm)
  - exponential distribution of decay time in particle rest frame

    $$\text{prob}(t) = 1/N \, dN/dt \, (t) = 1/\tau \, e^{-t/\tau}$$

  - decay length ($L$) in lab frame is distributed as:

    $$\text{prob}(L) = 1/(\beta\gamma c\tau) \, e^{-L/\beta c\gamma\tau}$$

    where $\beta c$: $K^0_s$ velocity in lab frame ($\beta = v/c = p/E$)

    $\gamma\tau$: decay time in lab frame; $\gamma = 1/\sqrt{(1-\beta^2)} = E/m$, relativistic boost factor determining time-dilation

    $\rightarrow$ average decay lenght, $\langle L \rangle = \beta\gamma c\tau$

Similarly to single-particles, $K^0_s$ are formed with a given momentum distribution $dN/dp$
(we checked $dN/dp_T$ for single particles but at midrapidity $dN/dp$ and $dN/dp_T$ are similar)
$\rightarrow$ decay length distribution determined by the convolution of $dN/dp$ and $dN/dt$

# Let’s get familiar with the data

Without refining too much the analysis selections, let’s try to find them

Combine positive and negative tracks to build candidates of $K^0_s$ decay

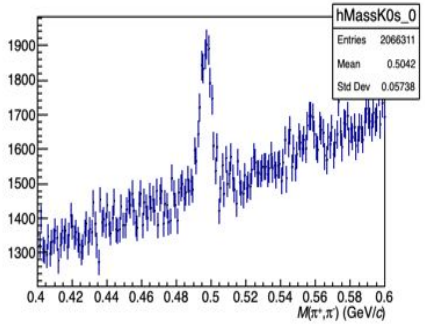
- take tracks from the same event!
- calculate invariant mass and plot it

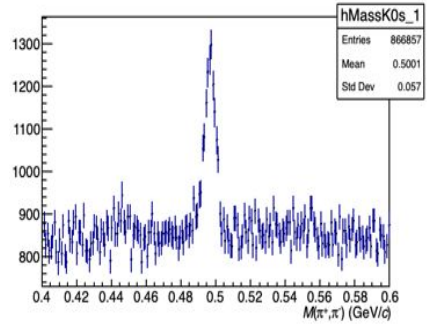
- you can play with some selections (PID: do you expect it to help a lot? dcaXY, …) and plot the invariant mass distribution in different pt intervals (e.g. 0-1,1-1.5,1.5-2,2-3,3-5).

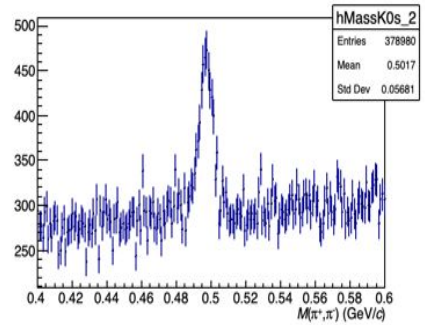
- train yourself in counting the yield (i.e. the number of K0s): fit the invariant mass distribution with a Gaussian term for the signal and another function (straight line, parabola, exponential… whatever you like and makes sense) for the background

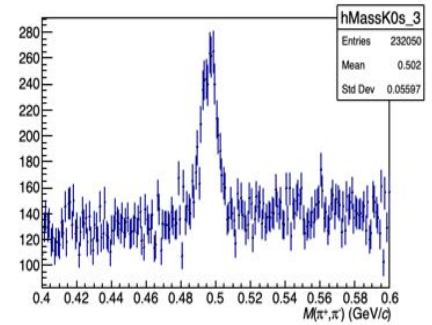
# Let’s get familiar with the data

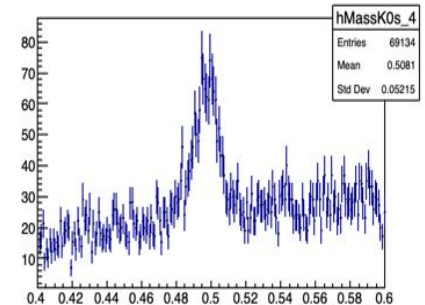
You should get something like this



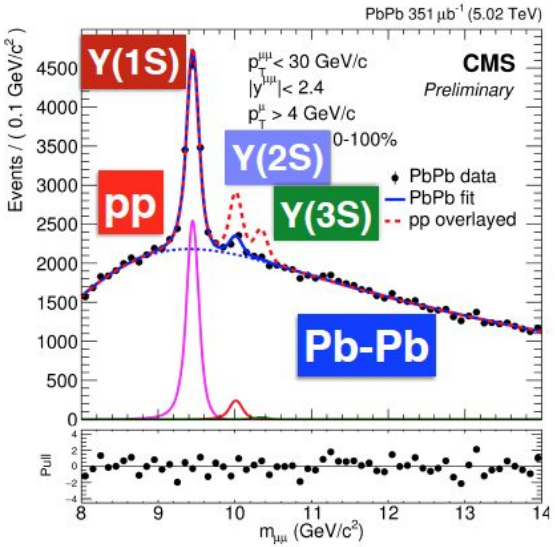



# Invariant mass




Invariant mass analysis: a procedure to count particles that decay.

Mass + quantum numbers (spin, quark content) = particle identity card

$p^{\mu}_{\text{Mother}} = p^{\mu}_{\text{Daughter 1}} + p^{\mu}_{\text{Daughter 2}}$ (in any frame, for a N-body decay the sum is over the N daughters)

$p^{\mu}_{\text{Mother}} p_{\mu,\text{Mother}}$ is a Lorentz-invariant quantity

In Mother rest frame:

$p^{\mu}_{\text{Mother}} = (M,0,0,0), \quad p^{\mu}_{\text{Daughter 1}} = (E_1, \vec{p_1}) \quad , \quad p^{\mu}_{\text{Daughter 2}} = (E_2, \vec{p_2})$ N. B.: $\vec{p_2} + \vec{p_1} = 0$

$\sqrt{p^{\mu}_{\text{Mother}} p_{\mu,\text{Mother}}} = M$ in any frame

$\sqrt{\left(p^{\mu}_{\text{Daughter 1}} + p^{\mu}_{\text{Daughter 2}}\right)\left(p_{\mu,\text{Daughter 1}} + p_{\mu,\text{Daughter 2}}\right)} = M$ in any frame

# End of first lecture

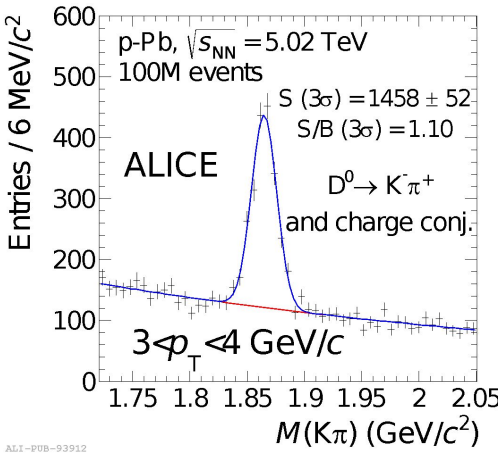
# Try to find $D^0$ decays

- Two-body decay, combinatorial very similar to $K^0_s$
- S/event much lower, S/B much lower

→ need:

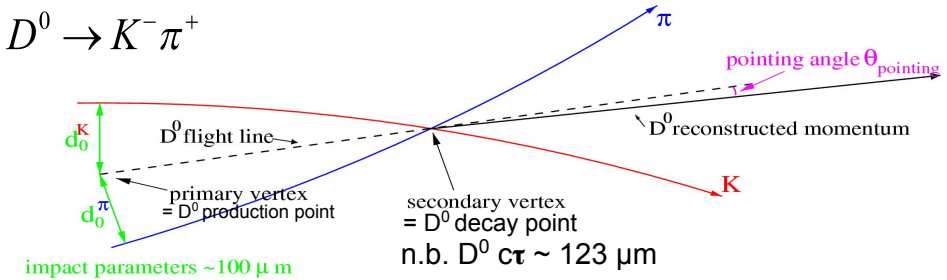
files not yet uploaded! I might update the pathname

1) **more events** → analyse sample in directory: /home/ubuntu/ALICEphyisicsOfData/data/tracks/alice_data_2023_LHC23f_5/
   Several files, in directories with only partly regular names
2) **need selections:**
   - exploit pion and kaon PID
   - exploit decay geometry, which impacts the distribution of single-track and pair variables



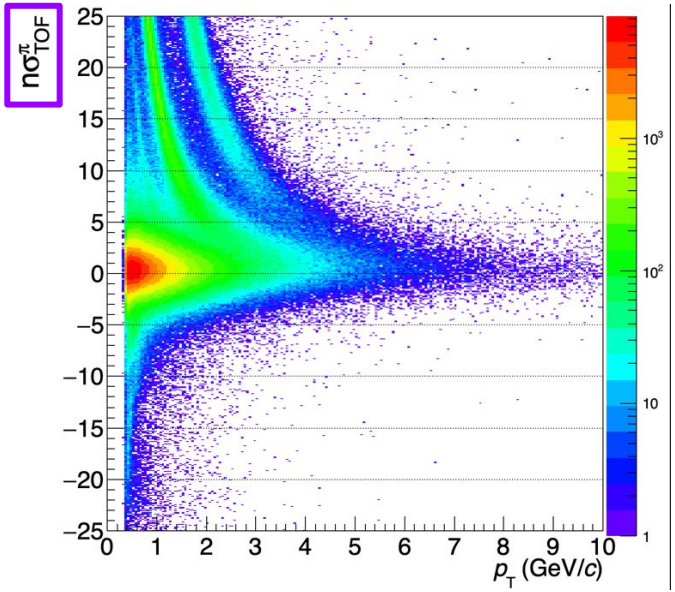

**Two "classes" of variables:**

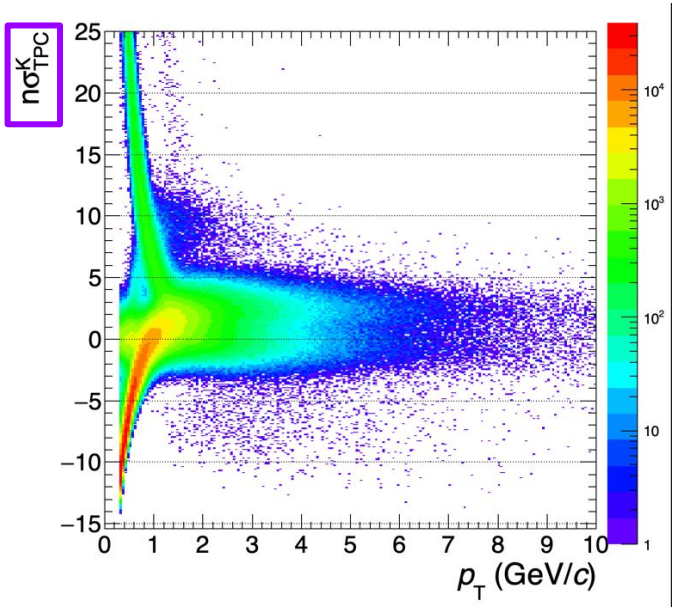
- single-track variables: can be exploited also to reduce inspected combinatorial ( → big impact on CPU time). E.g. PID, dcaXY
- pair variables: usually higher S-to-B discrimination power. E.g. decay length, PID



# Which selections? PID

- you should have become familiar with the TPC and TOF nsigma variables (n.b. sometimes you find very high values (~999) for TOF nsigma → tracks for which the TOF information is missing)
- why PID helps? Look at the plots: number of π >> number of K → (π,π) pairs is the dominant background component



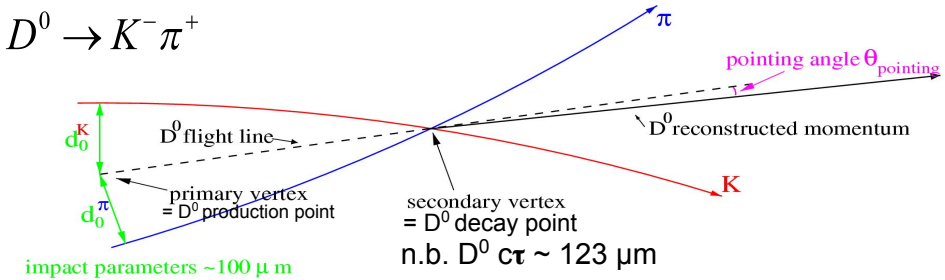



# Which selections? Decay geometry

**Signal**: decay displaced from primary vertex (PV, i.e. our estimate of collision-point position)
**Background**: mostly particles produced at the primary vertex

Useful variables, single tracks:

- **dcaXY of each track**:
  - background: for particles produced at the PV the real value would be zero but one has a distribution of values due to the finite resolution on the track trajectory and on the PV position → “resolution term”
  - signal: distribution is the result of the convolution of a real offset and the same resolution term which describes background
- **track $p_T$**: the average $p_T$ of the particles produced in the collision is lower than that of the D-meson daughters. E.g. if you require $p_T$>300 or 400 MeV/c you reduced the inspected combinatorial background.



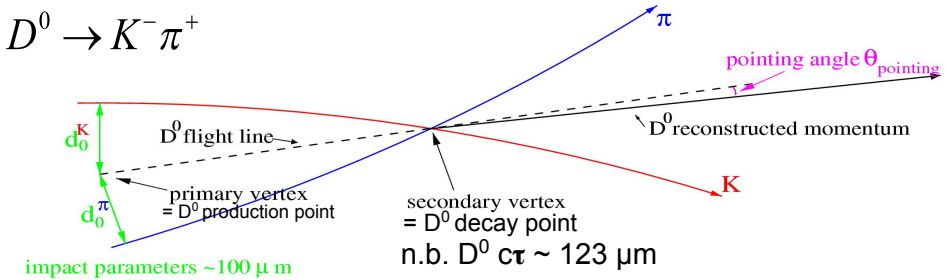
# Which selections? Decay geometry

**Signal**: decay displaced from primary vertex (PV, i.e. our estimate of collision-point position)
**Background**: mostly particles produced at the primary vertex

Useful variables, pairs:

- **dcaXY x dcaXY**: distribution (quite) symmetric around 0 for background, while asymmetric with tail to negative values for signal due to displacement from PV and convention of the dcaXY sign
- **decay length**: distance between the primary vertex and the decay point (also called “secondary vertex”, SV).
- **pointing angle**: angle between the $D^0$ momentum vector and the flight line, i.e. the line joining the PV and the decay vertex. Usually we use the cosine of this angle.
- we typically use also variables normalized by the uncertainty: that’s helpful but I do not advise you to explore this option

Decay length and pointing angle are very powerful but to calculate them you need the SV, whose proper calculation is not one of your main goals.
I suggest you to initially skip it and on a second stage give a quick try with what I describe in next slide, but go ahead if you do not manage.



# Which selections? Decay geometry

**Signal**: decay displaced from primary vertex (PV, i.e. our estimate of collision-point position)
**Background**: mostly particles produced at the primary vertex

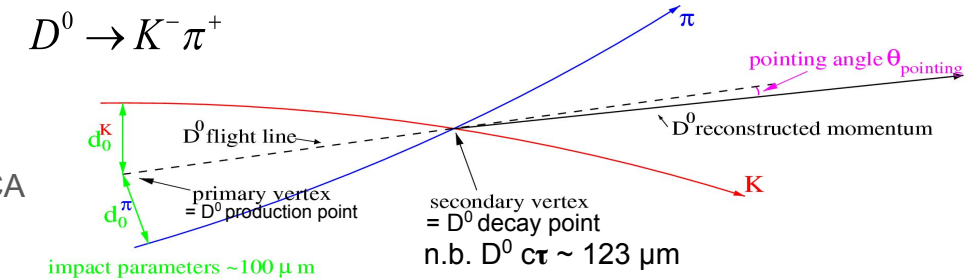
**Calculation of the SV**
A proper way to compute the SV would be to calculate the point-of-closest approach of the two tracks, using their uncertainty from the track covariance matrix, which I have not saved you in the trees.
Though you could retrieve the main cov. matrix term from dcaXY and dcaZ uncertainties, my advice is to **initially skip SV calculation** and then try to **estimate the SV in the following less accurate way**.

- assume tracks can be approximate to straight lines in the vicinity of PV and SV (this is a very reasonable assumption, curvature effects are negligible over few millimeters)
  - you can parametrize them using the momentum vector and the coordinates
- calculate the intersection of the track projections in the transverse plane (XY plane) → SV(x,y)
- calculate the SV z coordinate as the average of the z track coordinates after propagating the tracks at the found SV

$$z_{\text{track at SV}} = p_z/p_x \cdot x_{SV} + z_0$$

with $z_0$ to be determined using the z track coordinate at PCA

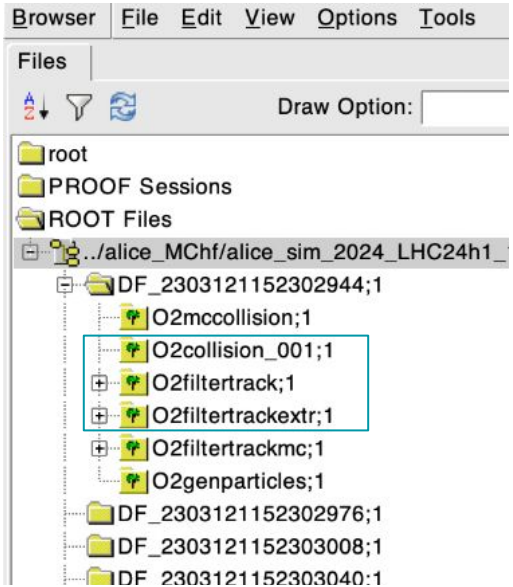
**Important**: you must work with global coordinates!



# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced

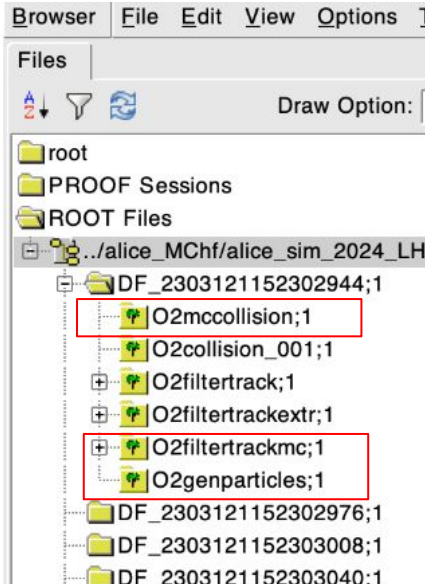



In each file you find the same DF structure than in the real data files

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced

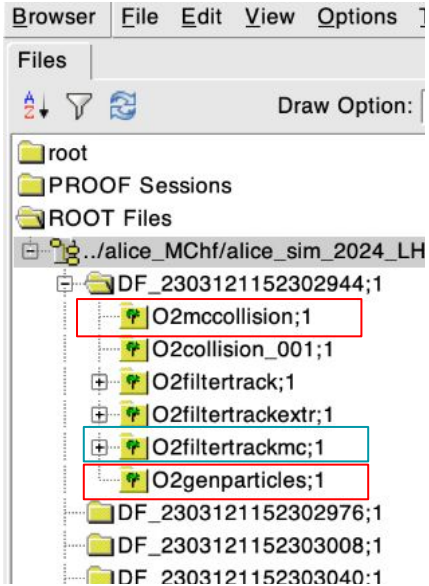



In each file you find the same DF structure than in the real data files
but you find also additional trees

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




## Monte Carlo simulation

### Generation step

- particles (quarks, gluons, ɣ, hadrons,...) produced in the simulation of the collision
- interaction of particles with detector ( → detector response, electrical signals)

quark1
W boson
$D^0$
charged hadrons out of acceptance
…

unreconstructable

hadron 1 in acceptance
hadron 2 in acceptance
hadron 3 partly out-of-acceptance
hadron 4 in acceptance
hadron 5 K in acceptance but decayed after 1 m

### Reconstruction step

- start from electrical signal
- recover “point-by-point” information (e.g. fired pixels, energy deposit)
  - clustering
- track finding and fitting

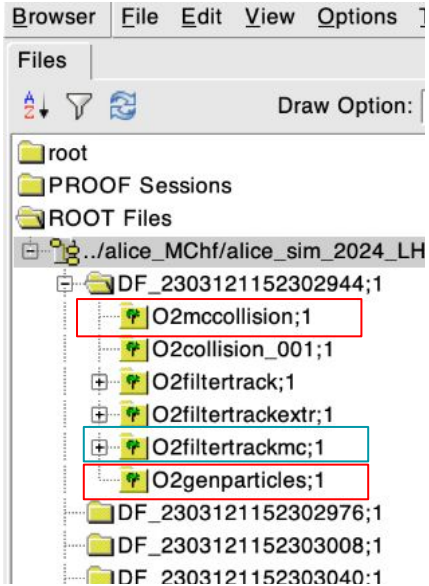
→ matching to MC information

track 1 (majority)
track 2
track 3 (fake! quite rare)
track 4 short track, discarded by basic selections
track 5
…

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




**Monte Carlo simulation**

Generation step → **O2genparticles, O2mccollision**
Information from gen. step

Reconstruction step → **O2filtertrackmc**
MC information of reconstructed objects (tracks)

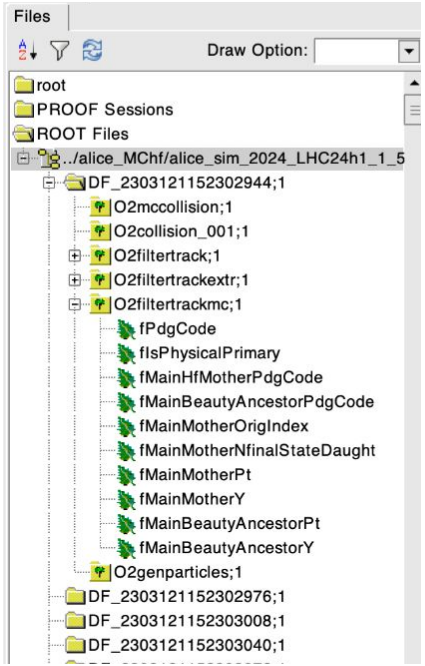
Information from both steps are needed in analysis:
e.g. suppose you want to know how many particles of a given kind you reconstructed and detected of those produced. You need to know

- **acceptance**: how many particles were generated in acceptance / number of generated particles
- **efficiency**: how many particles were reconstructed (reconstruction efficiency) and selected (selection efficiency)

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




**O2filtertrackmc**: table of the same size of O2filtertrack
→ **one entry per reconstructed track** containing simulation information of the particle associated to the reconstructed track

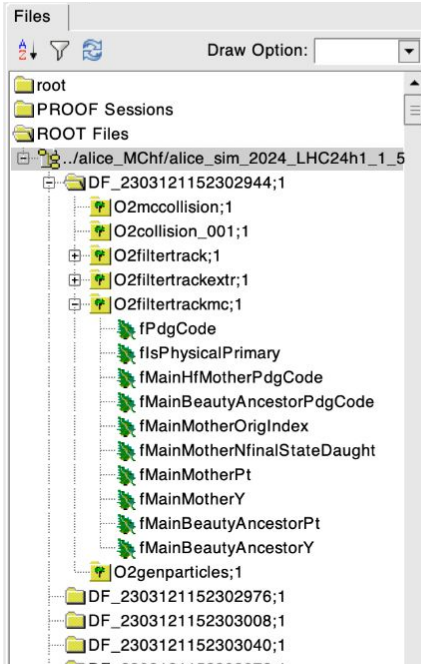
- **fPdgCode**: identifies particle species according to convention reported in PDG ("Particle Data Group"): https://pdg.lbl.gov/2007/reviews/montecarlorpp.pdf
e.g. $\pi^{+(-)}$: (-)211; $K^{+(-)}$:(-)321; p (p) : (-)2212;

- **fMainHfMotherPdgCode**: pdg code of the mother when relevant, 0 otherwise
  - HF particles we want to study: $D^0$ ($\overline{D^0}$): (-)421; $\Lambda_c^+$ ($\Lambda_c^-$): (-)4122
  - $K^0_S$: 310

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




**O2filtertrackmc**: table of the same size of O2filtertrack

- **fMainMotherOrigIndex**: index of mother particle in original simulation tree

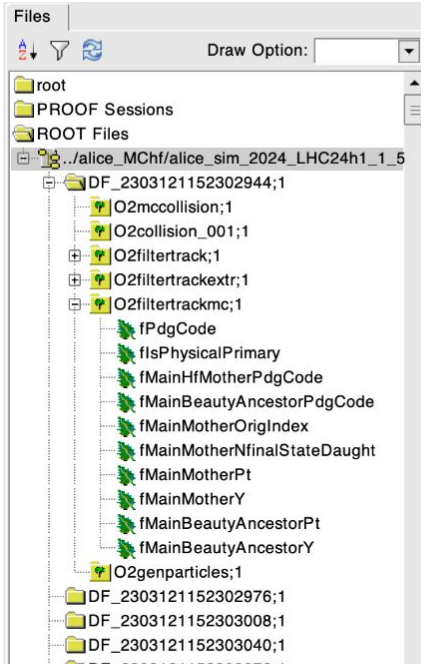
In simulation many particles (quarks, gluons, hadrons) are produced. Some particles may even appear more than once (e.g. think to a quark radiated a gluon, or to a particle which interacts with the material). Book keeping them all, makes the tree size very large. I saved you only the index and a filtered tree (O2genparticles), which contains the information of only the particles we are interested in.

You can use it to check that two tracks (particles) come from the same mother. E.g. you find two particles with fMainHfMotherPdgCode = 421, you know that they come from a $D^0$ decay, but you do not know whether it's the same $D^0$ → you must require that the fMainMotherOrigIndex is the same

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




**O2filtertrackmc**: table of the same size of O2filtertrack

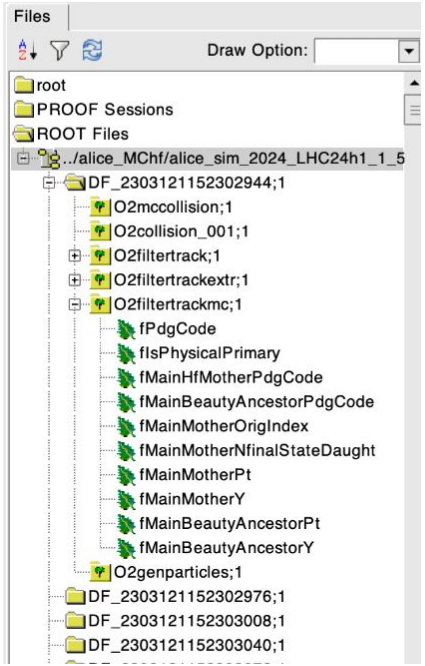
- **fMainMotherNfinalStateDaught**: number of final-state daughters, negative in case the final state does not match one in which we are interested (e.g. $D^0 \rightarrow K^-K^+$, $\Lambda_c^+ \rightarrow p(K^0_s\text{->})\pi^-\pi^+$)

Usage: it's not enough that two or three particles come from the same $D^0$ or $\Lambda_c^+$ to identify signal, you must be sure that these $D^0$ and $\Lambda_c^+$ decayed in the right channel.

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




**O2filtertrackmc**: table of the same size of O2filtertrack

In summary: a pair of tracks corresponds to a $D^0 \rightarrow K^-\pi^+$ decay if:

- fPdgCode are -321 and 211
- fMainHfMotherPdgCode = 421 for both
- fMainMotherOrigIndex is the same for the two tracks
- fMainMotherNfinalStateDaught = 2 (for both)

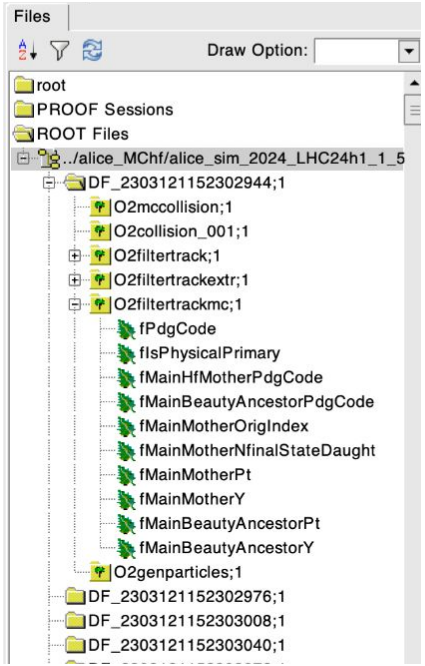
Similar condition would apply to the three daughters of $\Lambda_c^+ \rightarrow pK^-\pi^+$ decay

**fMainMotherPt, fMainMotherY**: $p_T$ and rapidity ($y$) of mother particle

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




**O2filtertrackmc**: table of the same size of O2filtertrack

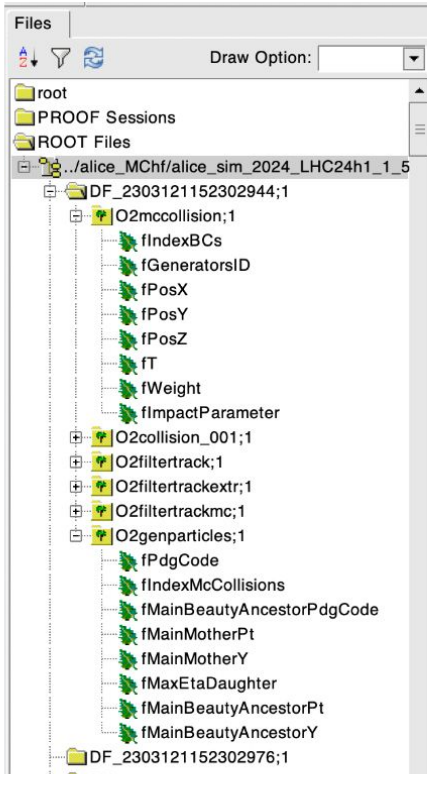
**fMainBeautyAncestorPdgCode**: pdg code of beauty particles for cases in which the charm hadrons derive from beauty decay, 0 otherwise.
**fMainBeautyAncestorPt, fMainBeautyAncestorY**: $p_T$ and rapidity ($y$) of beauty ancestor particle.

Charm hadrons from beauty decay are a source of background if one wants to measure directly produced charm (“prompt charm hadrons”).
**Exclude them when calculating efficiency.**
The computation of the fraction of charm hadrons from beauty decay in data is not simple. The method we currently use is reported here (https://inspirehep.net/literature/1848990 ) but it is beyond what you should do.

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




**O2genparticles**: table of *filtered* generated particles

> As previously mentioned, in simulation many particles (quarks, gluons, hadrons) are produced. Some particles may even appear more than once (e.g. think to a quark radiated a gluon, or to a particle which interacts with the material). Book keeping them all, makes the tree size very large.

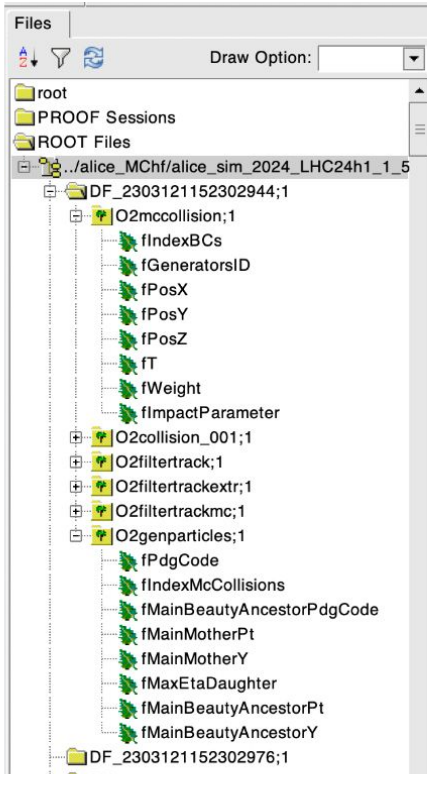
→ I saved you only a filtered tree, which contains the information of
**only the particles we are interested in: $D^0$ and $\Lambda_c^+$ (and $K^0_s$)**

**fPdgCode:** pdg code of generated particle (same convention as in O2filtertrackmc)
**fMainMotherPt, fMainMotherY**: particle $p_T$ and rapidity. Sorry for possibly misleading name: “mother” here it is used in correspondence to “main mother” in O2filtertrackmc
**fMaxEtaDaughter:** max abs(pseudorapidity, $\eta$) of daughter particles. Needed to identify reconstructable decays, for which daughters must have $|\eta|<0.8$
**fMainBeautyAncestor [PdgCode,Pt,Y]:** same as in O2filtertrackmc
**fIndexMcCollisions**: match to generated collision index of tree entry in O2mccollision table → needed only to select collision with |fPosz| <10 cm

# Files with data from Monte Carlo simulations

files not yet uploaded! I might update the pathname

in the directory: alice_sim_2024_LHC24h1_1_536237_AOD/
you will find a series of directories with AO2Dtree.root files inside.
These files were produced with a Monte Carlo simulation in which the $D^0$ and $\Lambda_c^+$ signals are enhanced




**O2mccollision**: table of generated collisions

**fPosX, fPosY, fPosZ:** global coordinates of collision point
n.b. primary vertex in O2collision_001 table gives coordinates of primary vertex, i.e. of the reconstructed position of what is assumed to be a collision

You will need only the fPosZ variable to reject collisions generated with at **|fPosZ| <10 cm**

**Apply same condition in reconstruction:** i.e. reject tracks from collision whose primary vertex has |fPosZ| <10 cm

# Practical instruction summary (example for $D^0$)

**Monte Carlo simulation** **Data**

## Generation step

reject collisions generated with at **|fPosZ| <10 cm**

**Count generated prompt $D^0 \rightarrow K^-\pi^+$ :**
**→ efficiency denominator**

fPdgCode = 421
fMainMotherPt
abs(fMainMotherY) <0.8 (or 0.5)
fMainBeautyAncestor =0

## Reconstruction step

reject PV with at **|fPosZ| <10 cm**
reject tracks with $|\eta|$ <0.8
apply your desired selections, build your pairs and select them as you wish
→ defined $p_T$ interval in which you can/want to "make the measurement"
**→ (data only): count reconstructed and selected $D^0$ from invariant mass fit → Raw yield**
**→ count reconstructed and selected $D^0$ by matching to Monte Carlo info → efficiency numerator**

Check that a pair of tracks actually corresponds to a $D^0 \rightarrow K^-\pi^+$ decay:

- fPdgCode are -321 and 211
- fMainHfMotherPdgCode = 421 for both
- fMainMotherOrigIndex is the same for the two tracks
- fMainMotherNfinalStateDaught = 2 (for both)
- fMainBeautyAncestorPDGcode = 0

# Efficiency, corrected yield, statistical uncertainty

$$\textbf{Efficiency} = \frac{\text{number of particles reconstructed and selected}}{\text{number of generated particles}}$$

**“Corrected yield” = “raw yield” / efficiency x further corrective factors**

Measured counts after selections

Usually from MC simulations
e.g. contamination from beauty decay

With proper normalization (number of events)
→ yield/event (*)
→ **Physical quantity, independent on the detector and analysis details**
For ratio ($\Lambda_c^+/D^0$) of particle yields measured in the same dataset, normalization cancels

**Statistical uncertainty of the measurement determined by statistical uncertainty of raw yield**

(*) More frequently, we report cross section (σ)

$$\Re = \mathcal{L}\sigma$$

$\Re$=Rate ($s^{-1}$) $\mathcal{L}$ = Luminosity ($cm^{-2}\ s^{-1}$)

# Statistical uncertainty and signifiance

Number of produced particles as well as that of measured particles fluctuate according to Poisson statistics

If you repeat the same experiment and data analysis N times counting the raw yield R of a given particle you get a distribution of values centered around a mean $\mu = \langle R \rangle$ with a variance $\sigma^2 = \mu$.

If you analyse $N_{events}$ you get an average raw yield $\langle R \rangle = r \times N_{events}$
where r is a parameter determined by nature and the detector/analysis details (=generated yield x efficiency)

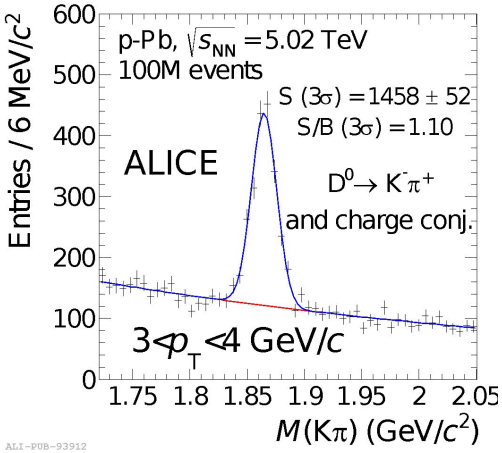
→ uncertainty on yield : $\sigma(\langle R \rangle) = \mathrm{sqrt}(\langle R \rangle) = \mathrm{sqrt}( r \times N_{events} )$
scales with sqrt of number of events
→ if I collect x4 more data I reduced the (relative) uncertainty by a factor of 2

# Statistical significance

From a given number of signal (S) and background (B) counts you can calculate the statistical significance

$$\frac{S}{\sqrt{S+B}}$$

← amplitude of fluctuations according to Poisson statistics




Remember:
the statistical significance is approximately the inverse of the relative statistical uncertainty:

T = S + B → $\sigma(T)=\sqrt{T}$ (both S and B fluctuate following Poisson statistics);

S = T - $\langle B \rangle_{fit}$ → $\sigma^2(S) = \sigma^2(T) + \sigma^2(\langle B \rangle_{fit})$ ; but B is constrained from sidebands → $\sigma^2(\langle B \rangle_{fit}) < \sigma^2(T)$

→ $\sigma(S)=\sqrt{T}$ → $\sigma(S)/S$ = 1/significance

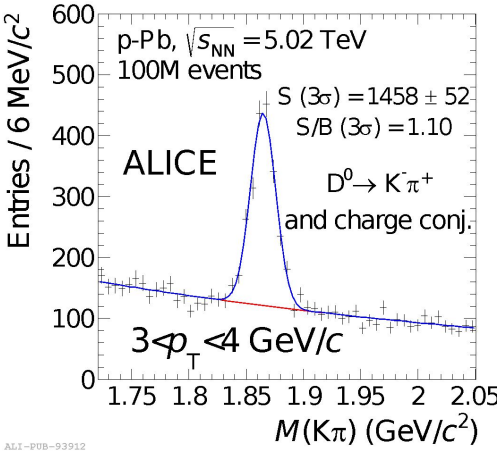
# Raw yield with two different background functions

The determination of the background function is often a delicate point in the analyses. Why? Test it yourself

Take a D0 (or K0s) invariant mass distribution in few selected pt intervals and implement three fit functions, using always a Gaussian for the signal:
1- Linear function for background
2- Exponential function
3- Parabolic (2nd order polynomial) function
If you change the fit range (try a couple of times), what does it happen?
Fill a table with the numbers you get and derive some conclusions

| Pt interval | Linear | Exp … |
|---|---|---|
| 1 | yield, chi2 | |
| 2 | | |
| 3 | | |
| 4 | | |

N.B. some fits/background models can be ruled out (e.g. bad chi2), for others is more difficult
You can apply statistical tests
But it's not always easy to conclude
"gray areas" → source of systematic uncertainty



# Tasks

1) **Look and plot (some of) the distributions of the variables which can be used to identify signal pairs:**
   - on data (mostly background) and on Monte Carlo simulation selecting signal tracks.
   - for pair variables: before applying selections your pairs will be mostly background
     Anyhow, if you select a region far from the $D^0$ mass peak in data (e.g. mass>1.94 GeV/$c^2$) you are sure to have only background pairs.
     - there could be also other ways to compare signal and background

Check the scales (e.g. 10, 100, 1000, 10000 μm for dcaXY or decay length, cosine of pointing angle can be very close to unity for signal), think to what can be more helpful and what is not.

2) **Find a reasonable set of selections to count signal**
   - the selections are typically varied/optimized as a function of $p_T(D^0)$: however, focus first to find a good set of selections for the interval $3<p_T(D^0)<8$ GeV/c
   - **calculate the efficiency** of the selection in the $p_T$ intervals: 0-1,1-2,2-3,3-5,5-8,8-16 GeV/c
   - **plot efficiency vs. $p_T(D^0)$** (binned)

You can also train a Machine Learning model (e.g. using Scikit learn) but it is not fundamental at this stage

# Tasks

3) **try to optimize the selections in a blind way to maximize expected statistical significance**
   for a bunch of selections that you consider reasonable, compute
      Signal / ev = efficiency x assumed generated yield/ev
         just as a guidance, take the $D^0$ $p_T$ distribution from the MC table and scale it so that the integral gives you a (pt-integrated) yield/ev = $2.5 \times 10^{-4}$
         → this will not give you too realistic values
      Background/ev from data: in a reasonable mass window (e.g. 50 MeV, close to the $D^0$ peak) count the background and divide it by the number of collisions inspected).

→ for each selection you can calculate S/B and the significance, i.e. S/sqrt(S+B)

4) Calculate, with your selection, what is the expected statistical significance (thus statistical uncertainty) for $10^7$, $10^8$, $10^9$,$10^{10}$ events
5) Estimate what would be the CPU time needed to run your code over the mentioned statistics
6) Study of invariant mass fit

You could in principle repeat all steps for the $\Lambda_c^+$ case, but we can better analyse $\Lambda_c^+$ at next step

# Last part (next meeting)

Files with already filtered $D^0$ and Lc candidates

- prompt, fd separation from MC
- data candidate file

Build a ML model to separate signal and background
- draw ROC curve
- study feature importance (and try to understand it)
- calculate efficiency for what you consider a good selection

Apply to data tree

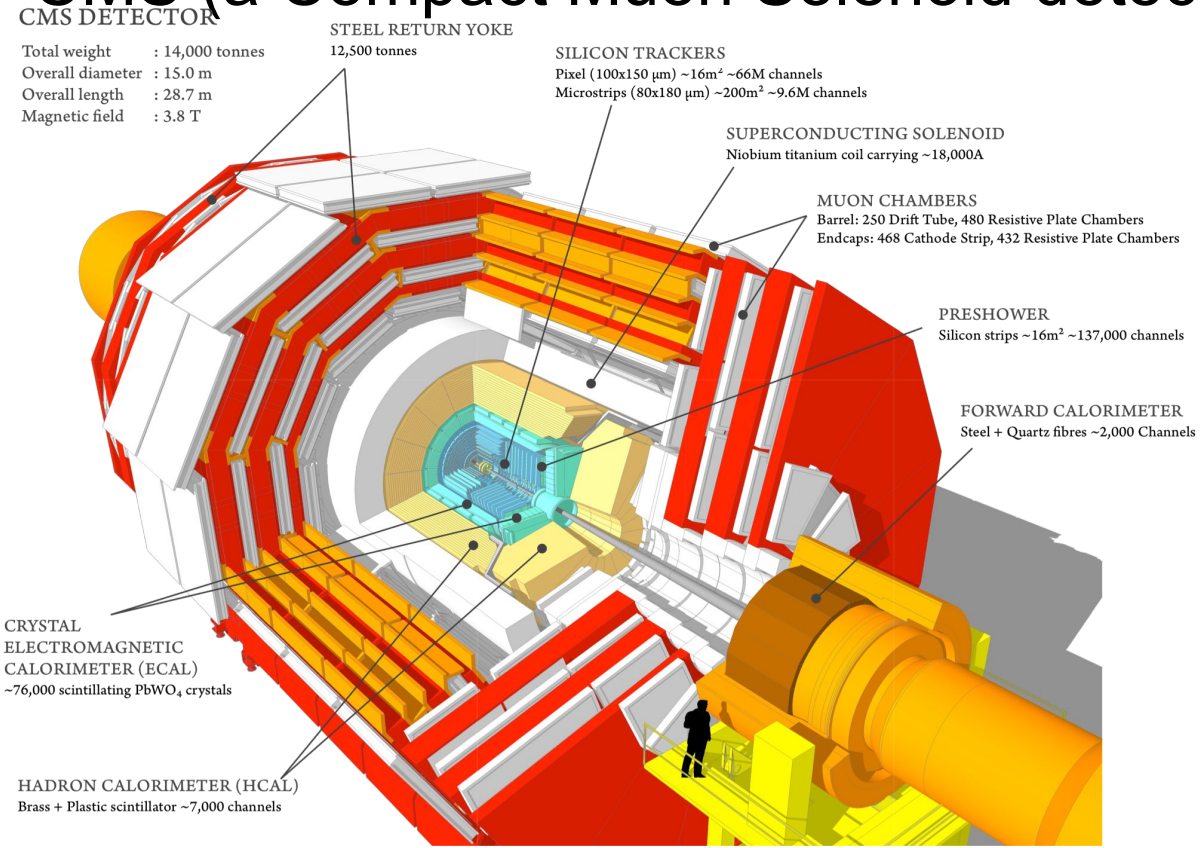
# Work in team

E.g. splitting the work in few tasks

1) study of variables
2) invariant mass fit
3) machine learning model
4) efficiency calculation
5) (if you wish) secondary vertex calculation

extra

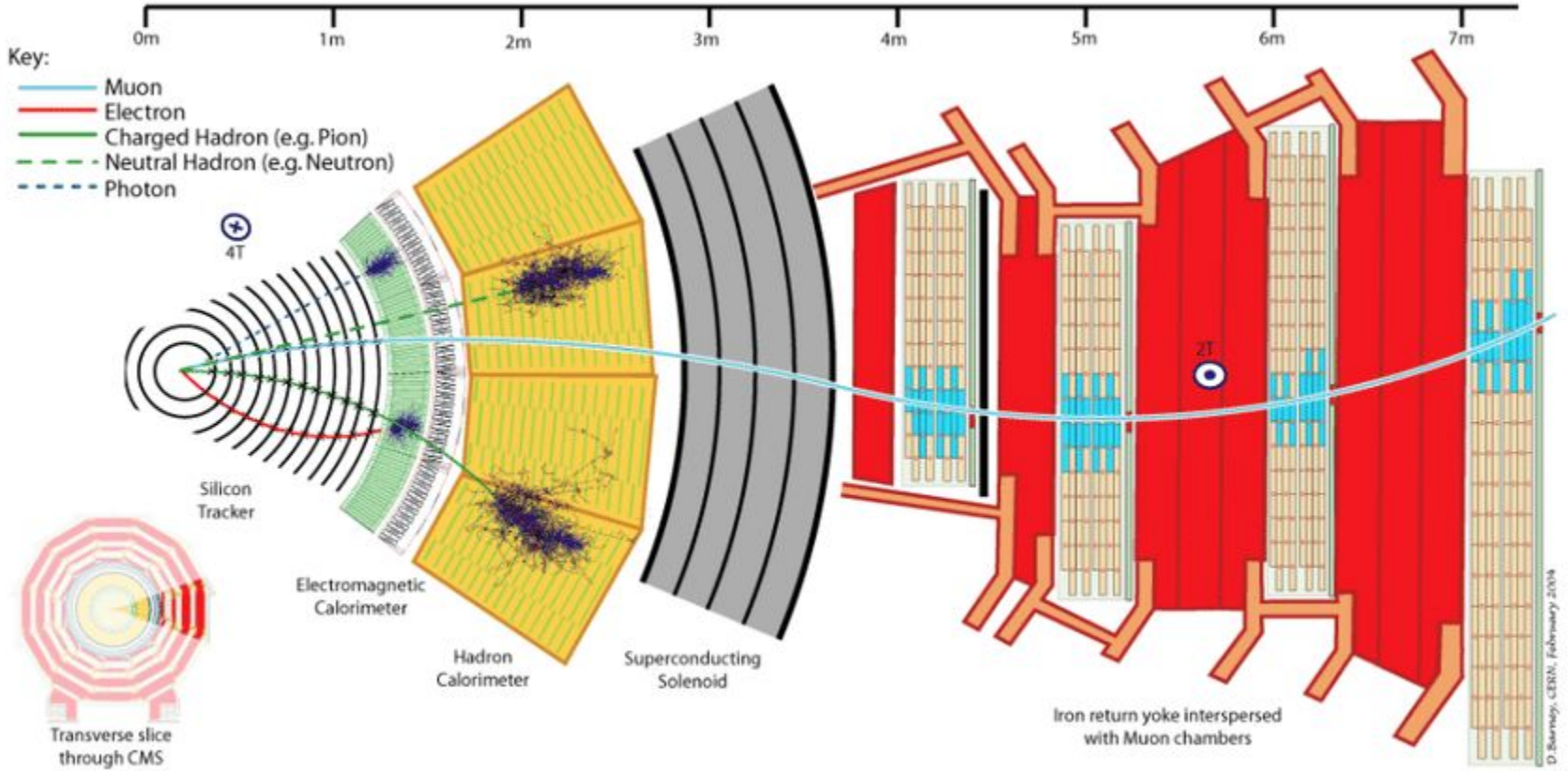
# CMS (a Compact Muon Solenoid detector)




Muon detectors are always “external” because **muon identification exploits their penetrating power**. Muons do not interact much with material:

- they do lose energy and suffer from multiple Coulomb scattering, but these processes do not destruct them/not lead to absorption.
- they do not interact strongly
- cannot annihilate (there are no muons/antimuons in the material)
- Bremsstrahlung is a rather minor effect (differently from electrons, due to the x200 larger mass)

C


0m
1m
2m
3m
4m
5m
6m
7m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
Transverse slice through CMS