Introduction to the HF O^2 analysis framework and general information

Mattia Faggin, on behalf of the HF O^2 team
INFN Padova (Italy)

O^2 analysis tutorial 5.0
Wednesday 12th November 2025

Previous tutorials

mfaggin@cern.ch

O^2 Analysis Tutorials

- [2021, December] HF O^2 software hackathon: https://indico.cern.ch/event/1101005/
- [2022, October] O^2 analysis tutorial: https://indico.cern.ch/event/1200252/timetable/#20221013.detailed
- [2023, April] O^2 analysis tutorial 2.0: https://indico.cern.ch/event/1267433/timetable/#20230417.detailed
- [2023, November] O^2 analysis tutorial 3.0: https://indico.cern.ch/event/1326201/timetable/
- [2024, October] O^2 analysis tutorial 4.0: https://indico.cern.ch/event/1425820/
- [2025, October] O^2 analysis tutorial 5.0: https://indico.cern.ch/event/1574136/timetable/

MONDAY 10 NOVEMBER

09:00 → 13:10 Lectures — 53/R-044
Convener: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))

09:00	Introduction to O2/O2Physics analysis Speaker: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))	1h 15m
10:15	Coffee break	15m
10:30	Analysis with Hyperloop Speaker: Nicolas Poffley (CERN)	1h
11:30	DPG: Event and track selection recommendations	40m
12:10	DPG: A brief introduction to O2 reconstruction	40m

14:00 → 17:30 General hands-on: Monday Hands-on — 53/R-044
Convener: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))

→ the basics of O2/O2Physics framework

TUESDAY 11 NOVEMBER

09:00 → 12:45 Lectures: Tuesday Lectures — 53/R-044
Convener: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))

09:00	Primary particle identification and analysis	25m
09:30	Secondary particle identification and analysis	25m
10:00	Using derived data in O2/O2Physics Speaker: Victor Gonzalez (Wayne State University (US))	25m
10:30	Coffee break	30m
11:00	Run 2 analysis using O2/O2Physics Speaker: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))	25m
11:30	Machine learning in O2/O2Physics Speaker: Francesco Mazzaschi (CERN)	25m
12:00	Software triggers in ALICE	25m

14:00 → 17:30 General hands-on: Tuesday Hands-on — 53/R-044
Convener: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))

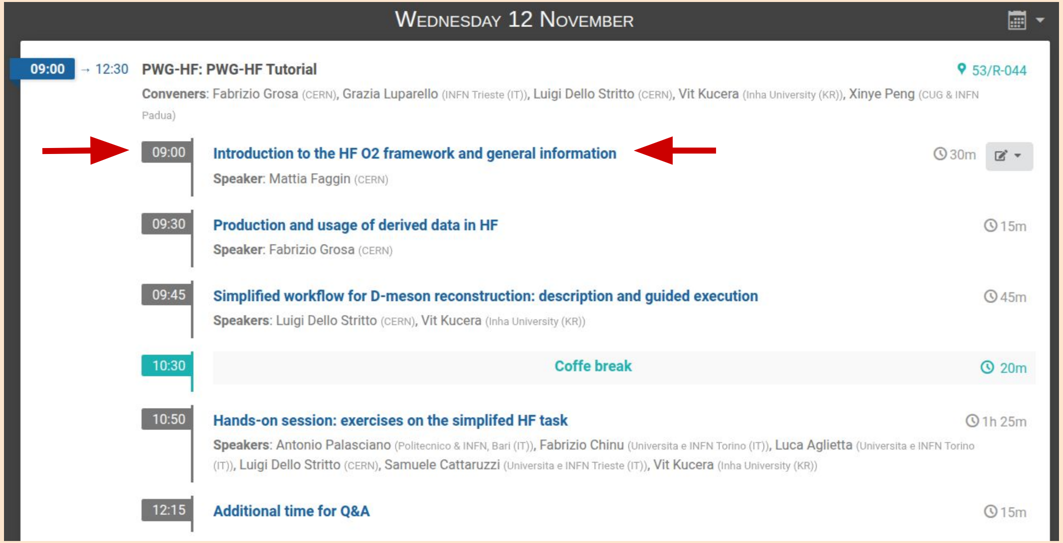
Previous tutorials

mfaggin@cern.ch

ALICE

O^2 Analysis Tutorials

- [2021, December] HF O^2 software hackathon: https://indico.cern.ch/event/1101005/
- [2022, October] O^2 analysis tutorial: https://indico.cern.ch/event/1200252/timetable/#20221013.detailed
- [2023, April] O^2 analysis tutorial 2.0: https://indico.cern.ch/event/1267433/timetable/#20230417.detailed
- [2023, November] O^2 analysis tutorial 3.0: https://indico.cern.ch/event/1326201/timetable/
- [2024, October] O^2 analysis tutorial 4.0: https://indico.cern.ch/event/1425820/
- [2025, October] O^2 analysis tutorial 5.0: https://indico.cern.ch/event/1574136/timetable/


Wednesday 12 November

09:00 → 12:30 **PWG-HF: PWG-HF Tutorial** — 53/R-044

Conveners: Fabrizio Grosa (CERN), Grazia Luparello (INFN Trieste (IT)), Luigi Dello Stritto (CERN), Vit Kucera (Inha University (KR)), Xinye Peng (CUG & INFN Padua)

09:00	**Introduction to the HF O2 framework and general information** Speaker: Mattia Faggin (CERN)	30m
09:30	**Production and usage of derived data in HF** Speaker: Fabrizio Grosa (CERN)	15m
09:45	**Simplified workflow for D-meson reconstruction: description and guided execution** Speakers: Luigi Dello Stritto (CERN), Vit Kucera (Inha University (KR))	45m
10:30	Coffe break	20m
10:50	**Hands-on session: exercises on the simplifed HF task** Speakers: Antonio Palasciano (Politecnico & INFN, Bari (IT)), Fabrizio Chinu (Universita e INFN Torino (IT)), Luca Aglietta (Universita e INFN Torino (IT)), Luigi Dello Stritto (CERN), Samuele Cattaruzzi (Universita e INFN Trieste (IT)), Vit Kucera (Inha University (KR))	1h 25m
12:15	**Additional time for Q&A**	15m

Today: ***HF analysis framework in O2Physics***

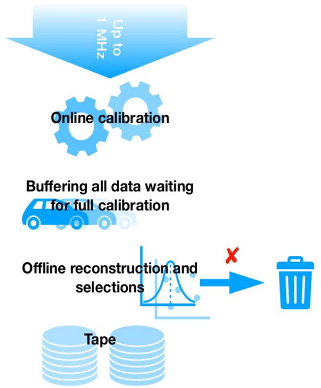
Upgrades and challenges in Run 3

mfaggin@cern.ch

HF goal: **precise charm- and beauty- hadron measurements down to p_T = 0**

- Large combinatorial background
- Small S/B ratio, difficult triggering

→ HF reconstruction and selection as the most challenging process in Run 3



Experimental upgrades:

- new ITS → based on ALPIDE technology
  - improved low p_T tracking efficiency (> 90% for p_T > 200 MeV/c)
  - improved pointing resolution to the PV (factor 2 (4) in rφ (z))
- upgraded TPC readout and frontend electronics
  - MWPC → GEM: similar performance as in Run 2 for dE/dx and tracking, but lower ion backflow (no more gating grid)
- **continuous readout up to 1 MHz (50 kHz) in pp (Pb-Pb) collisions**
  - 100 times more Pb–Pb data than in Run 2

Framework-design requirements:

- Minimize disk space occupied by derived analysis objects
- Maximize CPU performance, minimize running time

2.

Monday 10^{th} November

09:00 Introduction to O2/O2Physics analysis

Speaker: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))

1.

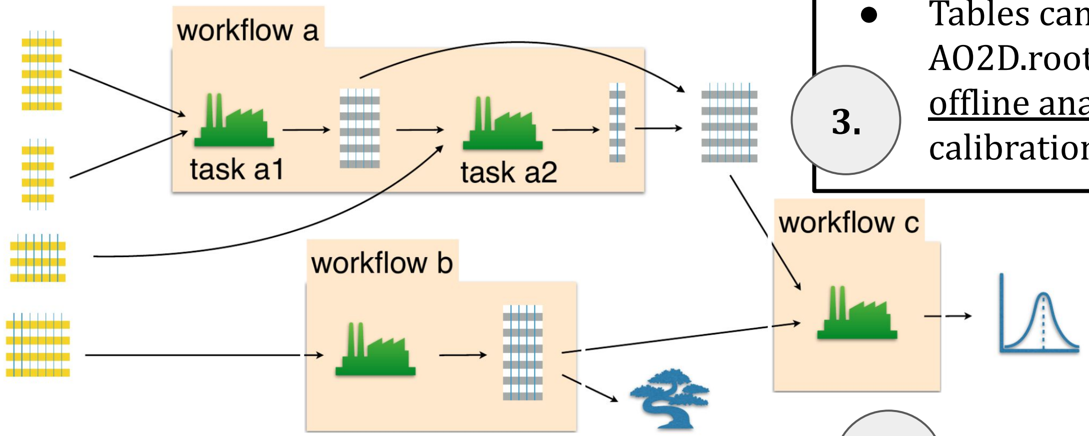
Data stored in flat tables interlinked via indices
<u>Example</u>: collisions, tracks in AO2D

2.

- Analysis task: `struct`
- Tasks with different purposes run together in a **workflow** (i.e. your "analysis")

3.

- Tables can be saved in AO2D.root as `TTree` on disk for <u>offline analysis</u> (e.g.: QA, calibrations, ML training, …)



4.

- A task can
  - read an existing table
  - create and fill a new table → available as input for the next tasks
  - extend an existing table → available as input for the next tasks

5.

End of the chain: final objects (`TH1`, `TH2`, `THnSparse`, `TEfficiency`, …) stored in `AnalysisResults.root`

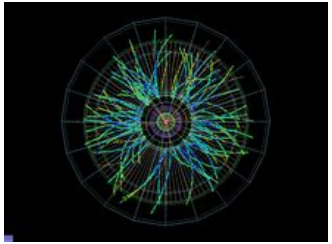
HF signal reconstruction - the strategy

- **Step 0: Event selection**

1.

Tracks

- Quality selections
- Particle Identification (PID) with central barrel detectors (e.g. TPC, TOF)



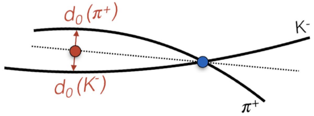
2.

Candidate reconstruction and filtering

- 2/3 track loop to reconstruct candidate from 2/3 - body decays
- Secondary-vertex determination
- Filtering with "loose" selections
- (MC) matching to generated particles

- On disk: only indices of tracks belonging to reconstructed secondary vertices
- Secondary-vertex information recomputed at the analysis level

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



HF derived data
"Production and usage of derived data in HF"
F. Grosa, today, 09:30

3.

Candidate selection

- Application of analysis selections (topological variables, track PID)

4.

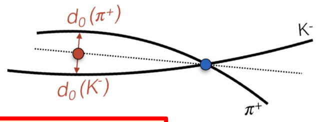
Physics analysis

- Store information into analysis objects (`TH1`, `TH2`, `THnSparse`, `TTree`, …)
Perform the analysis:
  - M_{inv} fits → signal
  - Corrections (eff × acc, f_{prompt})
  - Systematic uncertainties
  - Results

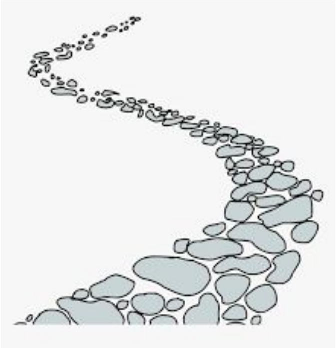
Tracks

Track index skim creator → **Candidate creator** → **Candidate selector** → **Analysis task**

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



Let's go through all the steps…



PWGHF/TableProducer/trackIndexSkimCreator.cxx

mfaggin@cern.ch

ALICE

1.

Track index skim creator

Doing a $D^0 \to K^- \pi^+$ analysis...

Here we flag the candidate $K\pi$ pairs, with loose preselections

Input: **tracks, collisions**

Event and track selections

- p_T, η, DCA, quality

Labelling for skimming (filtering)

- Double/triple loop over tracks
- *Loose* candidate preselection (invariant mass, p_T, $\cos\theta_p$,…)

Output: pairs of track indices for candidates (`Hf2Prongs`)

HF derived data

PWGHF/DataModel/TrackIndexSkimmingTables.h

```
DECLARE_SOA_TABLE_VERSIONED(Hf2Prongs_001, "AOD", "HF2PRONG", 1, //! Table for HF 2 prong candidates (Run 3 format)
                            o2::soa::Index<>,
                            hf_track_index::CollisionId,
                            hf_track_index::Prong0Id,
                            hf_track_index::Prong1Id,
                            hf_track_index::HFflag);
```

global index of the collision the candidate is associated to

1st track global index

2nd track global index

flag to identify the 2-prong candidate surviving the pre-selections

PWGHF/TableProducer/candidateCreator2Prong.cxx

mfaggin@cern.ch

2.

Candidate creator

Doing a $D^0 \to K^- \pi^+$ analysis…

Here we reconstruct the $D^0 \to K\pi$ candidates and their secondary vertex

Input: pairs/triplets of track indices for candidates (`Hf2Prongs`)

Candidate creation

- Secondary-vertex reconstruction and candidate building
- Full info for candidate selection and analysis

MC matching

- Rec. level (candidate)
- Gen. level (MC particle)
- MC origin tracing (non-)prompt (from b/c quark)

Output:

- Reconstructed HF candidates (`HfCandProng2Base`)
- MC flags

PWGHF/DataModel/CandidateReconstructionTables.h

```
// 2-prong decay candidate table
DECLARE_SOA_TABLE(HfCand2ProngBase, "AOD", "HFCAND2PBASE", //!
                  o2::soa::Index<>,
                  // general columns
                  HFCAND_COLUMNS,
                  // 2-prong specific columns
                  hf_cand::PxProng0, hf_cand::PyProng0, hf_cand::PzProng0,
                  hf_cand::PxProng1, hf_cand::PyProng1, hf_cand::PzProng1,
                  hf_cand::ImpactParameter0, hf_cand::ImpactParameter1,
                  hf_cand::ErrorImpactParameter0, hf_cand::ErrorImpactParameter1,
                  hf_cand::ImpactParameterZ0, hf_cand::ImpactParameterZ1,
                  hf_cand::ErrorImpactParameterZ0, hf_cand::ErrorImpactParameterZ1,
                  hf_track_index::Prong0Id, hf_track_index::Prong1Id, hf_cand::NProngsContributorsPV, hf_cand::BitmapProngsContributorsPV,
                  hf_track_index::HFflag,
```

```
// general columns
#define HFCAND_COLUMNS
  hf_cand::CollisionId,
    collision::PosX, collision::PosY, collision::PosZ,
    hf_cand::XSecondaryVertex, hf_cand::YSecondaryVertex, hf_cand::ZSecondaryVertex,
    hf_cand::ErrorDecayLength, hf_cand::ErrorDecayLengthXY,
    hf_cand::Chi2PCA,
```

… plus many other **dynamic columns**, for which the values are derived from those shown here

mfaggin@cern.ch

2.

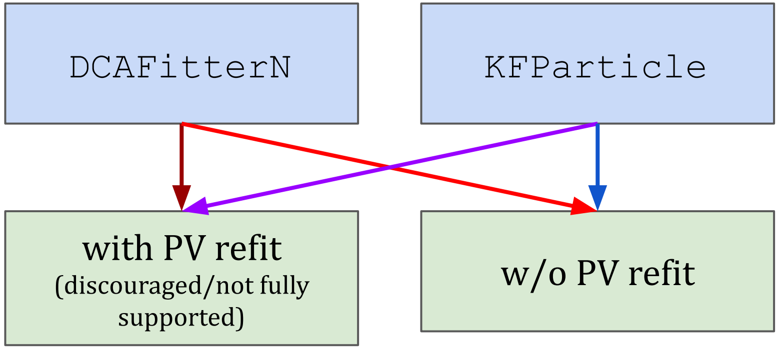
Candidate creator

Doing a $D^0 \rightarrow K^- \pi^+$ analysis…

Here we reconstruct the $D^0 \rightarrow K\pi$ candidates and their secondary vertex

secondary-vertex fitting

candidate-daughter removal from PV determination



Possibility to build HF-candidate **secondary vertex** with **two** different **algorithms**: DCAFitterN or KFParticle

⚠️⚠️⚠️ **Different tables created!** ⚠️⚠️⚠️

Be consistent with it during next steps (i.e. `candidate-selector`, `task`)

`processPvRefitWithDCAFitterN`
`processNoPvRefitWithDCAFitterN` (default)
`processPvRefitWithKFParticle`
`processNoPvRefitWithKFParticle`

… + other versions for analysis vs. centrality/UPC
(e.g `processNoPvRefitWithDCAFitterN`**`CentFT0C`**)

PWGHF/TableProducer/candidateSelectorD0.cxx

mfaggin@cern.ch

ALICE

3.

Candidate selector

Doing a $D^0 \rightarrow K^- \pi^+$ analysis...

*Here we **define** the selections on topological variables and PID to $D^0 \rightarrow K\pi$ candidate daughters*

Input:

- Reconstructed HF candidates (`HfCand2ProngBase` joined with other tables → `HfCand2Prong`)

Candidate selection definition

- Topological cuts
- Daughter PID cuts

Output: selection flags (`HfSelD0`)

NB: just flagging! Selections applied at the next step!

```
DECLARE_SOA_TABLE(HfSelD0, "AOD", "HFSELD0", //!
                  hf_sel_candidate_d0::IsSelD0,
                  hf_sel_candidate_d0::IsSelD0bar,
                  hf_sel_candidate_d0::IsRecoHfFlag,
                  hf_sel_candidate_d0::IsRecoTopol,
                  hf_sel_candidate_d0::IsRecoCand,
                  hf_sel_candidate_d0::IsRecoPid);
```

PWGHF/DataModel/CandidateSelectionTables.h

Is this a candidate $D^0 \rightarrow K^- \pi^+$ from the *creator* ?

Does it survive the topological selections?

Does it survive also the PID selections?

Then we can flag it as D^0 or anti-D^0

mfaggin@cern.ch

4.

Analysis task

Doing a $D^0 \to K^- \pi^+$ analysis...
Here we apply the selections and fill the histograms with $D^0 \to K\pi$ candidate information: M_{inv}, L_{xy}, $\cos\theta_p$, …

Input:

- ~~Selected~~ Flagged candidates (`soa::Join<HfCand2Prong, HfSelD0>`)
- MC particles
- MC flags

Analysis task

- Histogram filling for selected candidates

Output: **histograms** (kinematic properties, signal vs. background, efficiency, …), in stored .root file

4.

Analysis task

Doing a $D^0 \rightarrow K^- \pi^+$ analysis...
Here we apply the selections and fill the histograms with $D^0 \rightarrow K\pi$ candidate information: M_{inv}, L_{xy}, $\cos\theta_p$, ...

- Take into account only the 2-prong candidates flagged as D^0 or anti-D^0 in the `candidateSelectorD0`

```
using D0Candidates = soa::Join<aod::HfCand2Prong, aod::HfSelD0>;
[...]
Partition<D0Candidates> selectedD0Candidates = aod::hf_sel_candidate_d0::isSelD0 >= selectionFlagD0
|| aod::hf_sel_candidate_d0::isSelD0bar >= selectionFlagD0bar;
[...]
if (candidate.isSelD0() >= selectionFlagD0) {
    registry.fill(HIST("hMass"), massD0, ptCandidate);
    registry.fill(HIST("hMassFinerBinning"), massD0, ptCandidate);
    registry.fill(HIST("hMassVsPhi"), massD0, ptCandidate, candidate.phi());
}
```

4.

Analysis task

Doing a $D^0 \to K^- \pi^+$ analysis...
Here we apply the selections and fill the histograms with $D^0 \to K\pi$ candidate information: M_{inv}, L_{xy}, $\cos\theta_p$, …

- Take into account only the 2-prong candidates flagged as D^0 or anti-D^0 in the `candidateSelectorD0`
- Fill the histograms for your analysis!

⚠️

Beware of the **secondary-vertex fitter** you used in the `candidate-creator`, to use the correct `process` function!

⚠️

`processDataWithDCAFitterN` (default)

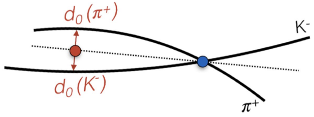
`processDataWithKFParticle` … and equivalent for MC

```
using D0Candidates = soa::Join<aod::HfCand2Prong, aod::HfSelD0>;
[...]
Partition<D0Candidates> selectedD0Candidates = aod::hf_sel_candidate_d0::isSelD0 >= selectionFlagD0
|| aod::hf_sel_candidate_d0::isSelD0bar >= selectionFlagD0bar;
[...]
if (candidate.isSelD0() >= selectionFlagD0) {
    registry.fill(HIST("hMass"), massD0, ptCandidate);
    registry.fill(HIST("hMassFinerBinning"), massD0, ptCandidate);
    registry.fill(HIST("hMassVsPhi"), massD0, ptCandidate, candidate.phi());
}
```

Recap of the D^0 reconstruction and analysis

mfaggin@cern.ch

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



Tracks
Collisions

`Hf2Prongs`

`HfCand2Prong`

`HfSelD0`

`soa::Join<HfCand2Prong, HfSelD0>`

Track index skim creator → **Candidate creator** → **Candidate selector** → **Analysis task**

Doing a $D^0 \to K^- \pi^+$ analysis…
Here we flag the candidate $K\pi$ pairs, with loose preselections

1.

Doing a $D^0 \to K^- \pi^+$ analysis…
Here we reconstruct the $D^0 \to K\pi$ candidates and their secondary vertex

2.

Doing a $D^0 \to K^- \pi^+$ analysis…
Here we define the selections on topological variables and PID to $D^0 \to K\pi$ candidate daughters

3.

Doing a $D^0 \to K^- \pi^+$ analysis…
Here we apply the selections and fill the histograms with $D^0 \to K\pi$ candidate information: M_{inv}, L_{xy}, $\cos\theta_p$, …

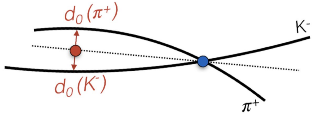
4.

The modularity of O^2 workflows allows to build analyses of multi-stage decays on top of analyses of direct ones

Tracks
Collisions

1. Track index skim creator → **2. Candidate creator** → **3. Candidate selector** → **Analysis task**

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



PS: possible to work with dedicated skimmed data (later)

4. B^+ candidate creator → **5. B^+ candidate selector** → **6. B^+ analysis task**

$B^+ \to D^{0bar} \pi^+$

B^+ candidates built from:

- **tracks**, from input AO2Ds
- **candidate $D^{0(bar)}$**, i.e. 2-prong candidates selected in the `candidateSelectorD0`

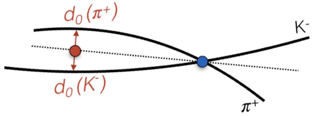
```
Filter filterSelectCandidates = (aod::hf_sel_candidate_d0::isSelD0 >= selectionFlagD0 ||
aod::hf_sel_candidate_d0::isSelD0bar >= selectionFlagD0bar);
```

PWGHF/TableProducer/candidateCreatorBplus.cxx

The modularity of O^2 workflows allows to build analyses of multi-stage decays on top of analyses of direct ones

Tracks
Collisions

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



Track index skim creator → **Candidate creator** → **Candidate selector** → **Analysis task**

Candidate selector → **B^+ candidate creator** → **B^+ candidate selector** → **B^+ analysis task**

$B^+ \rightarrow D^{0bar} \pi^+$

… many more analyses are possible!

2-prong

D^0 → B^+

3-prong

$D^+, D^{*+}, D^+_s, \Lambda^+_c, \Xi^+_c$ → B^0, D_{s1}^{+}, D_{s2}^{*+}, $\Sigma^{0,++}_c$, Ξ^{++}_{cc}

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

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

$B^+ \rightarrow D^{0bar} \pi^+$

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

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

$D_{s1}^{+} \rightarrow D^{*+} K^0_s$

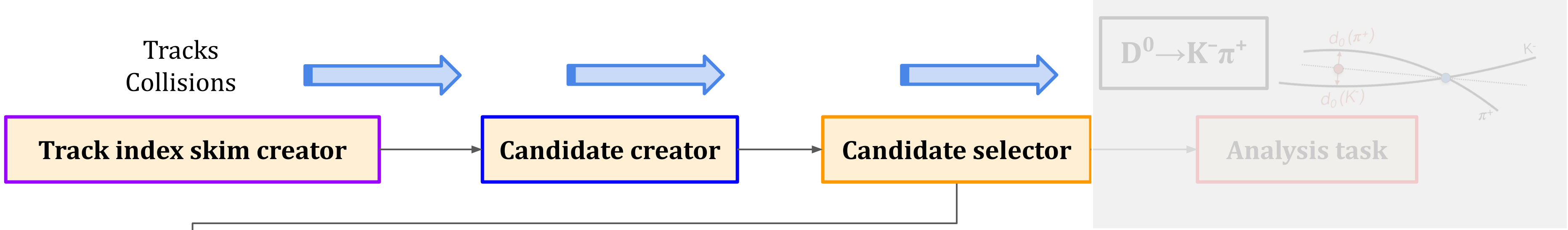
$D_{s2}^{*+} \rightarrow D^+ K^0_s$

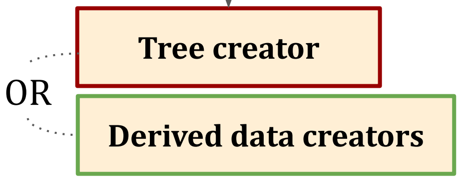
$\Lambda^+_c \rightarrow p\, K^- \pi^+, p\, K^0_S$

$\Sigma^{0,++}_c \rightarrow \Lambda^+_c \pi^{-,+}$

$\Xi^+_c \rightarrow p\, K^- \pi^+$

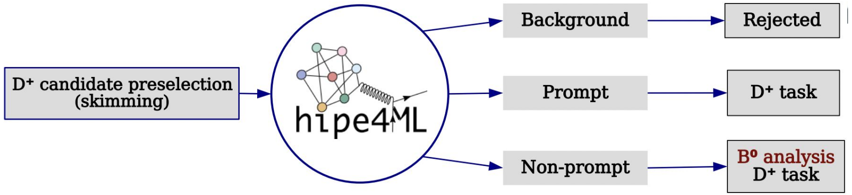
$\Xi^{++}_{cc} \rightarrow \Xi^+_c \pi^+$





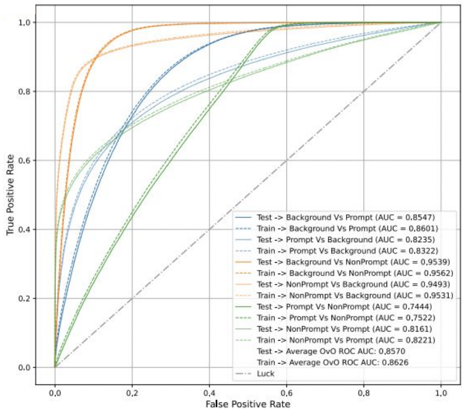
$B^0 \to D^- \pi^+$
$B^+ \to D^{0bar} \pi^+$
$B_s^0 \to D_s^- \pi^+$
$\Lambda_b^0 \to \Lambda_c^+ \pi^-$,

$D^0 \to K^- \pi^+$
$D^+ \to \pi^+ K^- \pi^+$
$D_s^+ \to K^- K^+ \pi^+$
$\Lambda_c^+ \to p\, K^- \pi^+$, $p\, K^0_S$
$\Xi_c^+ \to p\, K^- \pi^+$, $\Xi^- \pi^+ \pi^+$
$\Xi_c^0 \to \Xi^- \pi^+$
$\Xi_{cc}^{++} \to \Xi_c^+ \pi^+$



- `tree-creator`s: custom reduced tables on output AO2Ds.root files
- `derived-data-creator`s: lightweight self-contained interlinked tables
  → entry-point for jet framework (HFJ analyses)

→ Input for Machine Learning-based analyses
- model training doable offline (e.g. scikit-learn, hipe4ML)
- model application doable on GRID via ONNX
  - Tools/ML/MlResponse.h
  - PWGHF/Core/HfMlResponse.h



Tracks
Collisions

Track index skim creator → **Candidate creator** → **Candidate selector** → Analysis task

Tree creator

OR

Derived data creators

$B^0 \rightarrow D^- \pi^+$
$B^+ \rightarrow D^{0bar} \pi^+$
$B_s^0 \rightarrow D_s^- \pi^+$
$\Lambda_b^0 \rightarrow \Lambda_c^+ \pi^-$,

$D^0 \rightarrow K^- \pi^+$
$D^+ \rightarrow \pi^+ K^- \pi^+$
$D_s^+ \rightarrow K^- K^+ \pi^+$
$\Lambda_c^+ \rightarrow p\, K^- \pi^+$, $p\, K_S^0$
$\Xi_c^+ \rightarrow p\, K^- \pi^+$, $\Xi^- \pi^+ \pi^+$
$\Xi_c^0 \rightarrow \Xi^- \pi^+$
$\Xi_{cc}^{++} \rightarrow \Xi_c^+ \pi^+$

D⁺ candidate preselection (skimming) → hipe4ML

hipe4ML → Background → Rejected

hipe4ML → Prompt → D⁺ task

hipe4ML → Non-prompt → B⁰ analysis D⁺ task

Tuesday 11th November

11:30 **Machine learning in O2/O2Physics**

Speaker: Francesco Mazzaschi (CERN)

Wednesday 12th November

09:30 **Production and usage of derived data in HF**

Speaker: Fabrizio Grosa (CERN)

HF O^2 analysis framework Dependencies

Code not under the PWGHF responsibility, but fundamental to setup HF-related analyses

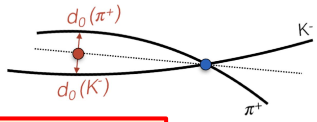
HF O^2 analysis framework - dependencies

mfaggin@cern.ch

Tracks
Collisions

4. Track index skim creator → **5. Candidate creator** → **6. Candidate selector** → **7. Analysis task**

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



Workflows specific of HF framework

Monday 10^{th} November

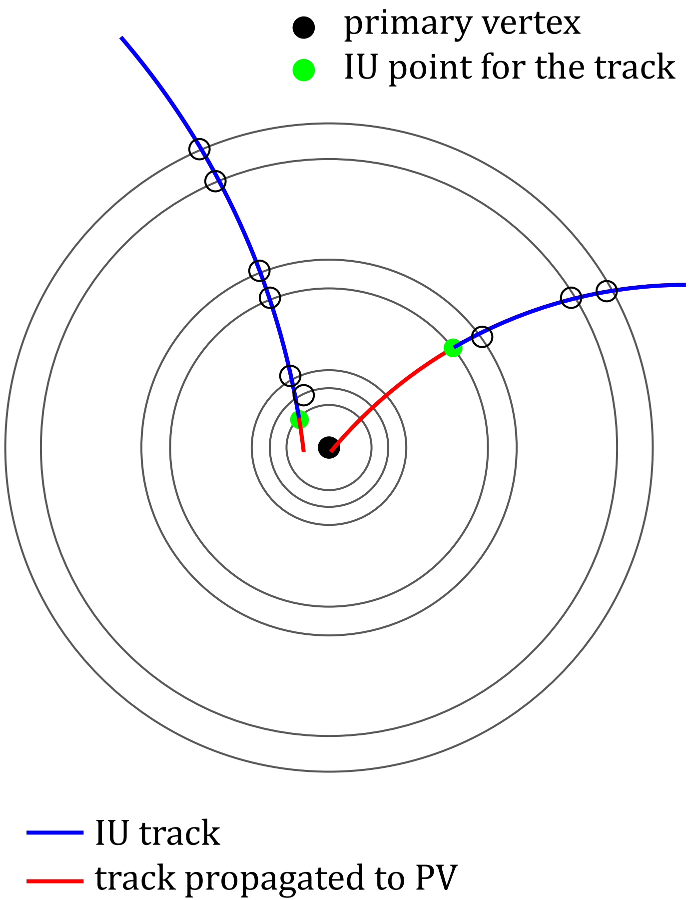
11:30 DPG: Event and track selection recommendations

Dependencies

1. Track propagation to the primary vertex → **2. Track selection** → **3. Track-To-Collision association**

Track propagation to the PV – the concept

mfaggin@cern.ch



The *track parametrization* at *different radii changes*
- more (or less) points to be used in the Kalman-Filter estimate
- material budget

- In AO2Ds: `TracksIU`
  - **IU: Innermost Update** point
  - Track-parametrization at the IU written in AO2Ds → not the same radius ↔ `track.x()`
  - Table written in the AO2Ds

- In analysis: `Tracks`
  - Track-parametrization after the propagation to the distance of closest-approach (DCA) to the primary vertex
  - In the workflow: dca_{XY} and dca_{Z} calculated as well
  - Table **not** written in the AO2Ds, but created on-the-fly by the analysis task

Common/TableProducer/trackPropagation.cxx

OBSOLETE

processStandard[XXX]
- track parameters
- dca_{XY}, dca_Z values

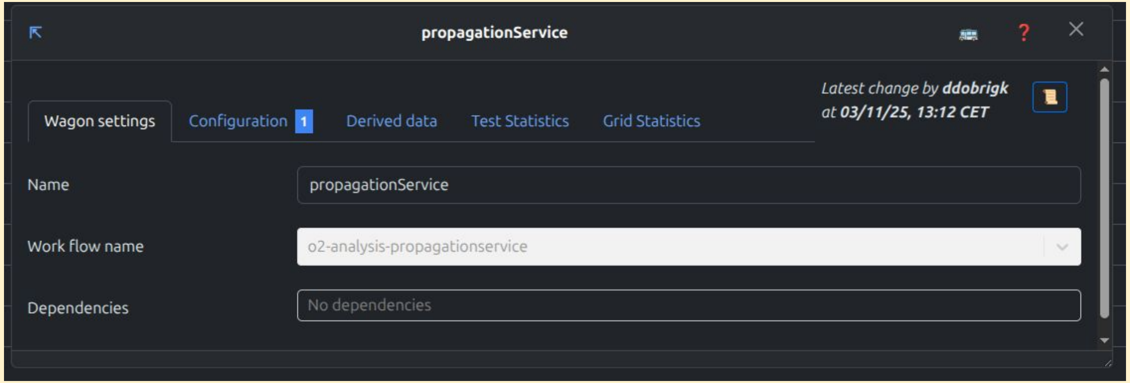
processCovariance[XXX]
- track parameters and covariance matrix
- dca_{XY}, dca_Z values and uncertainties

→ much more resources consumed!

Common/TableProducer/propagationService.cxx

- Code reorganized in such a way that the track propagation to PV and the strangeness builder (K_s^0, V^0) compose a unique analysis task
- Auto-detection of which of the two "tasks" to be executed, as well as of which process function, based on the tables later needed in the workflow

Wagon for *data*: Core_Service_Wagons/propagationService

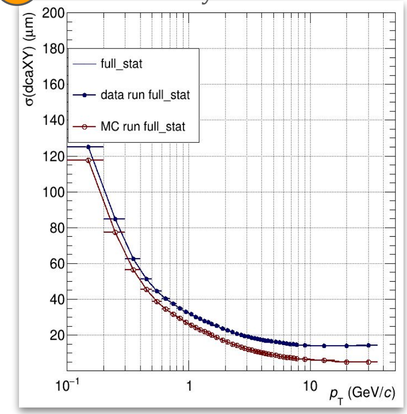


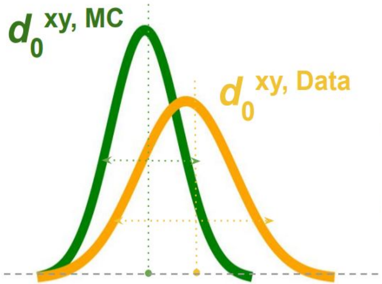
Track smearing in MC - the TrackTuner

mfaggin@cern.ch

1

dca$_{xy}$ resolution vs. p_T in data and MC



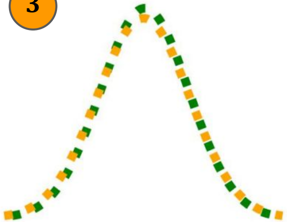


- TrackTuner: utility (*__not a task__*) to smear track parameters in MC in order to better reproduce the performance in data
- Variables adjustable with the TrackTuner:
  - track impact parameter $r\varphi$, z
    - → smearing based on data-vs-MC comparison, calibrations provided centrally in CCDB
    - → ❗ approach valid only for primary particles
  - track p_T
    - → ❗ based on custom inputs, i.e. no recipe provided centrally (single scaling factor, or input scaling factors vs. p_T)

2

TrackTuner at work

3

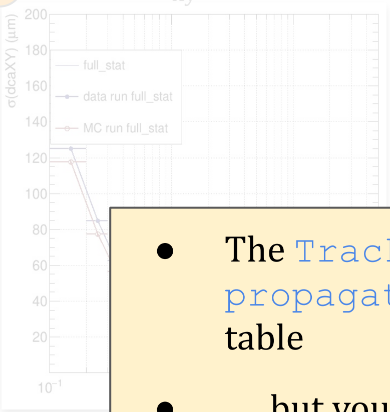


dca$_{xy}$ resolution in data and MC after the TrackTuner

mfaggin@cern.ch

1

dca$_{xy}$ resolution vs. p_T in data and MC



- TrackTuner: utility $\boxed{(\textit{not a task})}$ to smear track parameters in MC in order to better reproduce the performance in data
- Variables adjustable with the TrackTuner:
  - track impact parameter $r\varphi$, z
    → smearing based on data-vs-MC comparison,

- The TrackTuner is instantiated in the track-propagation and propagation-service tasks to handle the track smearing when creating the Tracks table
- … but you can instantiate by yourself the TrackTuner in your code, if needed!

  → example: DPG/Tasks/AOTTrack/V0Cascades/perfK0sResolution.cxx

TrackTuner at work

dca$_{xy}$ resolution in data and MC after the TrackTuner

Code: Common/Tools/TrackTuner.h

📚 Documentation here

📚 More info in this presentation

```
HfPropagationService_trackTuner_MC_vsPhi_autodetect
HfPropagationService_trackTuner_PbPb_MC
HfPropagationService_trackTuner_PbPb_MC_vsPhi
HfPropagationService_trackTuner_pp_MC
HfPropagationService_trackTuner_pp_MC_vsPhi
HfPropagationService_trackTuner_ptSmearing1p5_MC_vsPhi_autodetect
HfPropagationService_trackTuner_ptSmearing1p5_PbPb_MC
HfPropagationService_trackTuner_ptSmearing1p5_PbPb_MC_vsPhi
HfPropagationService_trackTuner_ptSmearing1p5_pp_MC
HfPropagationService_trackTuner_ptSmearing1p5_pp_MC_vsPhi
HfPropagationService_trackTuner_ptSmearing1p5_pp_worseResoPullsData1p1_pp_MC
```

Wagon for ***MC*** in : Service Analyses / Service Wagons HF

```
HfPropagationService_trackTuner_MC_vsPhi_autodetect
HfPropagationService_trackTuner_PbPb_MC
HfPropagationService_trackTuner_PbPb_MC_vsPhi
HfPropagationService_trackTuner_pp_MC
HfPropagationService_trackTuner_pp_MC_vsPhi
HfPropagationService_trackTuner_ptSmearing1p5_MC_vsPhi_autodetect
HfPropagationService_trackTuner_ptSmearing1p5_PbPb_MC
HfPropagationService_trackTuner_ptSmearing1p5_PbPb_MC_vsPhi
HfPropagationService_trackTuner_ptSmearing1p5_pp_MC
HfPropagationService_trackTuner_ptSmearing1p5_pp_MC_vsPhi
HfPropagationService_trackTuner_ptSmearing1p5_pp_worseResoPullsData1p1_pp_MC
```

	Calibrations in CCDB
pp, $\sqrt{s}$ = 13.6 TeV, 2022 and 2023	✅ Users/m/mfaggin/test/inputsTrackTuner/pp2023/pass4/vsPhi
pp, $\sqrt{s}$ = 13.6 TeV, 2024	✅ Users/m/mfaggin/test/inputsTrackTuner/pp2024/pass1_minBias/vsPhi
pp, $\sqrt{s}$ = 5.36 TeV, 2024	👷 work in progress
Pb–Pb, $\sqrt{s_{NN}}$ = 5.36 TeV	✅ Users/m/mfaggin/test/inputsTrackTuner/PbPb2023/apass4/vsPhi
O–O, Ne–Ne	❌ to be provided

Blue: settings for pp 2022, 2023 (not 2024, 2025)

Green: settings for Pb–Pb

Red: settings "auto-detected"

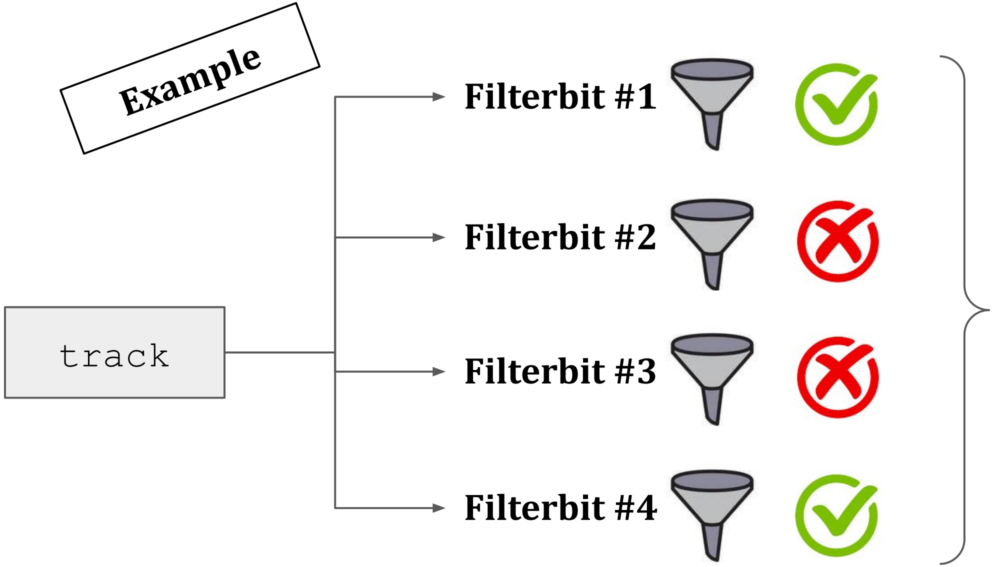
→ pp or Pb–Pb settings based on the analysed run number

Track selector (1/2)

mfaggin@cern.ch

- For each (<u>propagated!</u>) track the following check is done: does it satisfy the selections defined in the *i*-th predefined set (filterbit)?
- Track-by-track filling of `aod::TrackSelection` table, according to the responses

- This table contains for each track the flag for each filterbit
- To have a flag for each single cut: use the `aod::TrackSelectionExtension` table (not filled by default!)



`aod::TrackSelection`

	FB #1	FB #2	FB #3	FB #4
track0	✓	✗	✗	✗
track1	✗	✓	✗	✗
track2	✓	✓	✓	✗
track3	✓	✓	✓	✓
track4	✗	✗	✓	✗

Predefined sets of <u>pre-selections</u>: [Common/Core/TrackSelectionDefaults.cxx]

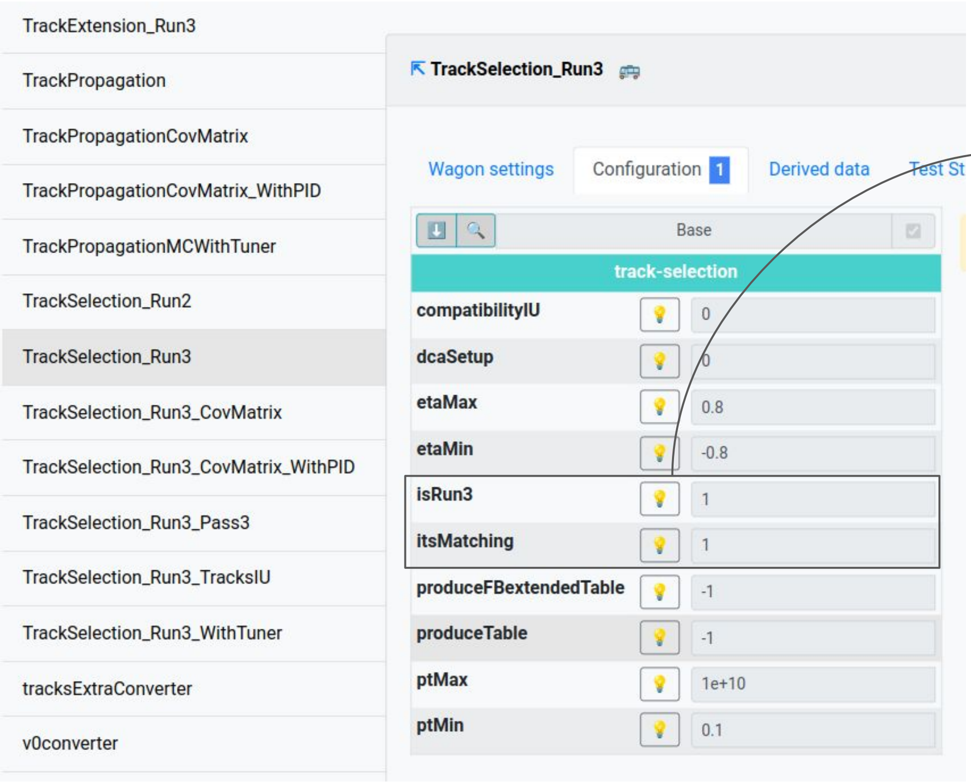
Track selector (2/2)

mfaggin@cern.ch

Code: [Common/TableProducer/trackSelection.cxx]

📚 Documentation [here]

[Core Service Wagons/TrackSelection_Run3]



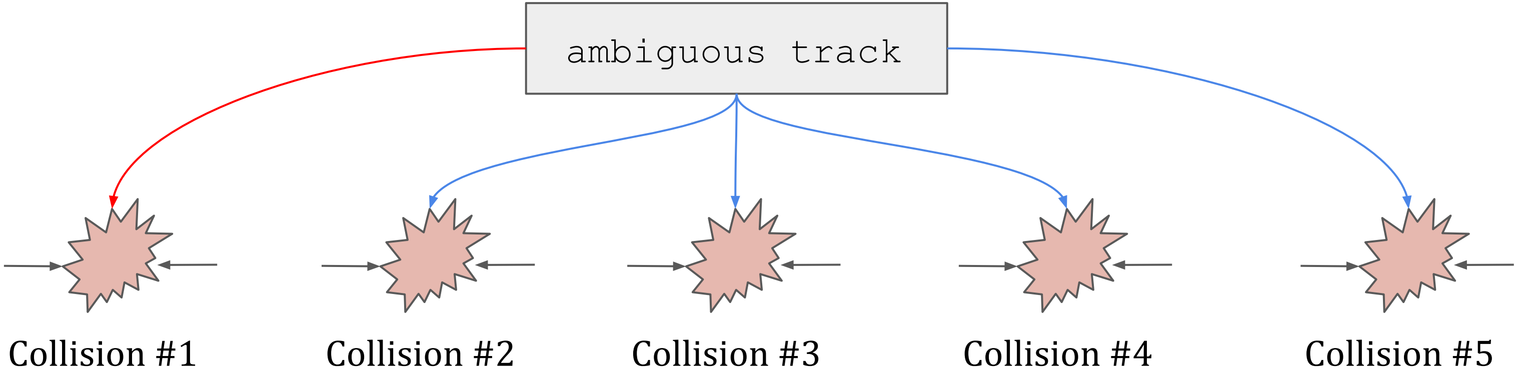
[Common/TableProducer/trackselection.cxx#L85-L91]

```
case 1:
        // Run 3 kAny on 3 IB layers of ITS
        if (isRun3) {
          [...]
          globalTracks =
getGlobalTrackSelectionRun3ITSMatch(TrackSelecti
on::GlobalTrackRun3ITSMatching::Run3ITSibAny,
dcaSetup.value);
          break;
        }
```

- By default, global track selections defined in [Common/Core/TrackSelectionDefaults.cxx#L27-L45] are enabled (see the documentation for ITS matching)
- Possibility to enable <u>subsets</u> of such cuts via "masks"
- Example of application in [DPG/Tasks/AOTTrack/qaEventTrack.cxx#L134-L141]

Track-to-collision associator (1/2)

mfaggin@cern.ch



- Continuous readout → **ambiguous tracks**: tracks with more than 1 collision possible
- By default, in the AO2D the `track.collisionId()` is that of the first compatible collision
- `track-to-collision-associator` : duplication of the track to each collisions compatible in time
  - recovery of 2,3-prong decay topologies!
  - possible signal duplication if all the daughters are ambiguous and are duplicated in many collisions → negligible with analysis selections on topological variables

→ Used already for the derived-data creation of 2,3-prong candidates
→ To be added explicitly in your analysis only for further bachelor tracks, if desired (e.g. $B^0 \rightarrow D^-\pi^+$)

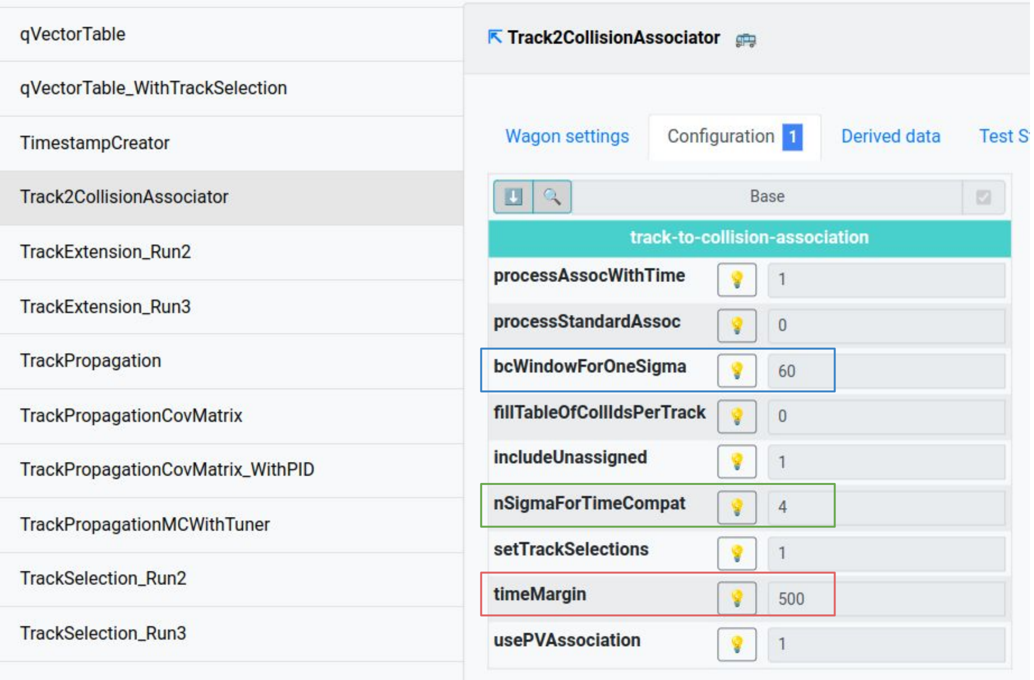
Track-to-collision associator (2/2)

mfaggin@cern.ch

Code:

- utility: Common/Core/CollisionAssociation.h
- workflow: Common/TableProducer/trackToCollisionAssociator.cxx

Core Service Wagons/TrackSelection_Run3



Common/Core/CollisionAssociation.h#L190

```
int64_t bcOffsetMax =
mBcWindowForOneSigma *
mNumSigmaForTimeCompat + mTimeMargin /
o2::constants::lhc::LHCBunchSpacingNS;
```

❗ arbitrary ❗ arbitrary ❗ arbitrary ❗

- *"Collisions compatible in time with a track"* == within a time window equal to ITS integration time

  60 bc ✕ 4 (✕ 25ns) = 240 bc (~6 μs)

- Further margin of 500 ns (20 bc) to account for possible imperfections in TPC calibrations

Linked derived data

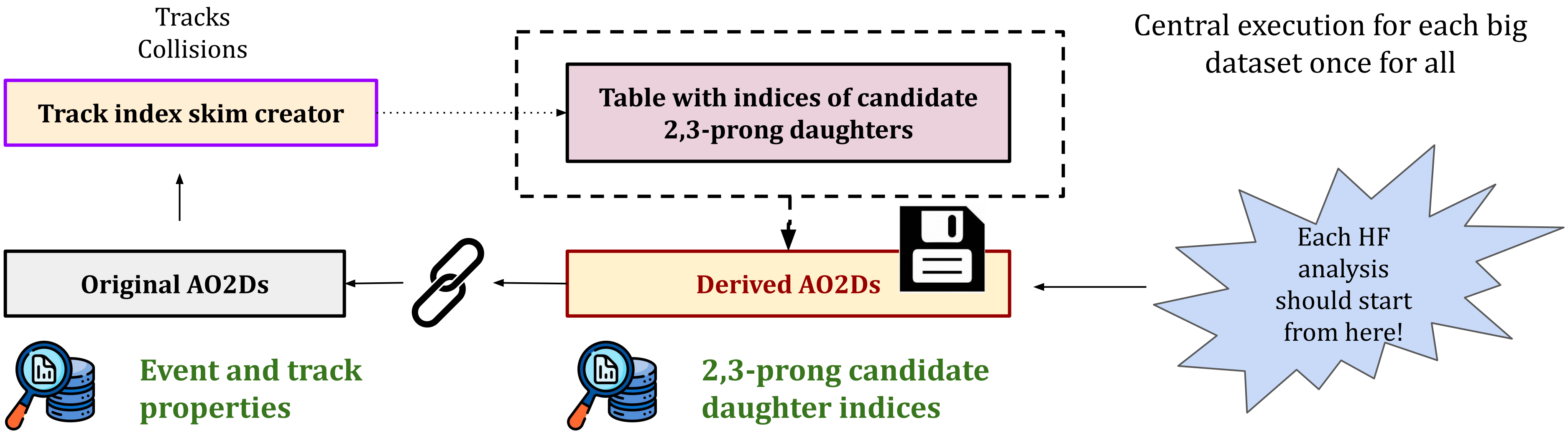
mfaggin@cern.ch

Tracks
Collisions

Track index skim creator → **Table with indices of candidate 2,3-prong daughters**

- Large resource consumption in the combinatorics
- Not possible to run on large datasets: only `_small` ones allowed w/o PB approval

Linked derived data



- **Derived AO2Ds**: .root files saved on disk containing the **table with (only!) indices of candidate 2,3-prong daughters**
- Derived AO2Ds linked to the original AO2Ds.root from the data reconstruction in Hyperloop
  → **query of original or derived AO2Ds depending on the requested info**
- **Self-contained** derived data creation possible for multi-staged analyses (e.g. B mesons)
  → *to be produced by the analysers*

Linked derived data

mfaggin@cern.ch

Tracks
Collisions

Track index skim creator

Table with indices of candidate 2,3-prong daughters

Central execution for each big dataset once for all

Original AO2Ds

Derived AO2Ds

Each HF analysis should start from here!

Event and track properties

2,3-prong candidate daughter indices

Wednesday 12th November

09:30

Production and usage of derived data in HF

Speaker: Fabrizio Grosa (CERN)

References and useful information

mfaggin@cern.ch

Mattermost: https://mattermost.web.cern.ch/alice/channels/hf-o2-analysis

Documentation:
https://aliceo2group.github.io/analysis-framework/docs/advanced-specifics/pwghf.html

O^{2}Physics code:
https://github.com/AliceO2Group/O2Physics/tree/master/PWGHF

Validation framework & postprocessing analysis tools:
https://github.com/AliceO2Group/Run3AnalysisValidation