HF O^2 analysis framework

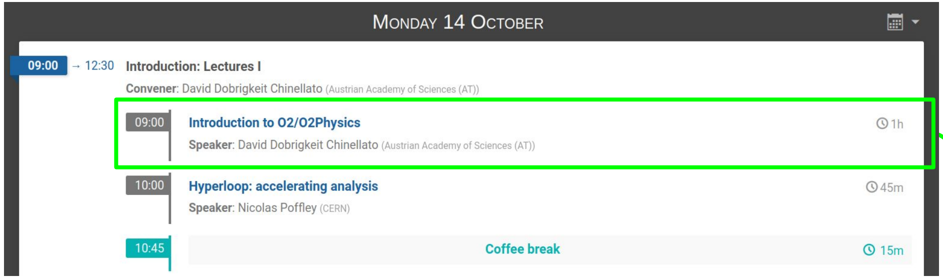
Mattia Faggin, on behalf of the HF O^2 team
CERN

O^2 analysis tutorial 4.0
Wednesday 16th October 2024

Previous tutorials

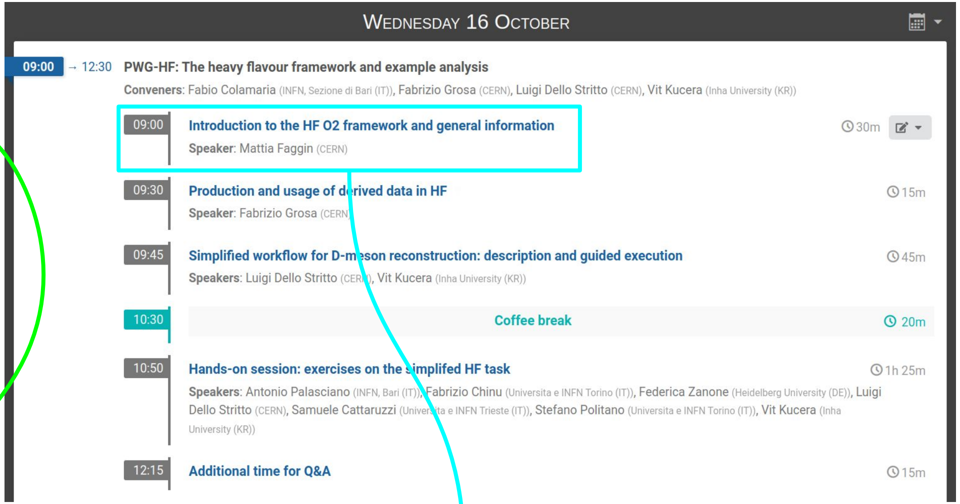
mfaggin@cern.ch

- [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/


MONDAY 14 OCTOBER

09:00 → 12:30 Introduction: Lectures I
Convener: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))

09:00	Introduction to O2/O2Physics Speaker: David Dobrigkeit Chinellato (Austrian Academy of Sciences (AT))	1h
10:00	Hyperloop: accelerating analysis Speaker: Nicolas Poffley (CERN)	45m
10:45	Coffee break	15m


WEDNESDAY 16 OCTOBER

09:00 → 12:30 PWG-HF: The heavy flavour framework and example analysis
Conveners: Fabio Colamaria (INFN, Sezione di Bari (IT)), Fabrizio Grosa (CERN), Luigi Dello Stritto (CERN), Vit Kucera (Inha University (KR))

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	Coffee break	20m
10:50	Hands-on session: exercises on the simplifed HF task Speakers: Antonio Palasciano (INFN, Bari (IT)), Fabrizio Chinu (Universita e INFN Torino (IT)), Federica Zanone (Heidelberg University (DE)), Luigi Dello Stritto (CERN), Samuele Cattaruzzi (Universita e INFN Trieste (IT)), Stefano Politano (Universita e INFN Torino (IT)), Vit Kucera (Inha University (KR))	1h 25m
12:15	Additional time for Q&A	15m

- Basics of O^2 framework

- Today: introduction to HF framework in O^{2}Physics

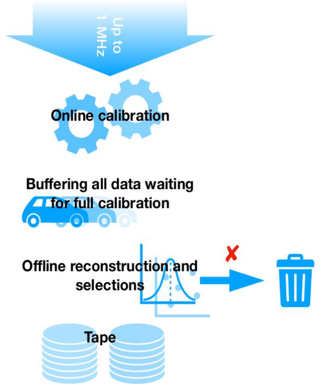
Upgrades and challenges in Run 3

mfaggin@cern.ch

<u>HF goal</u>: **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



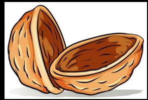
<u>Experimental upgrades:</u>

- 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

<u>Framework-design requirements</u>:

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

The O^2 framework in a nutshell



mfaggin@cern.ch

1.

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

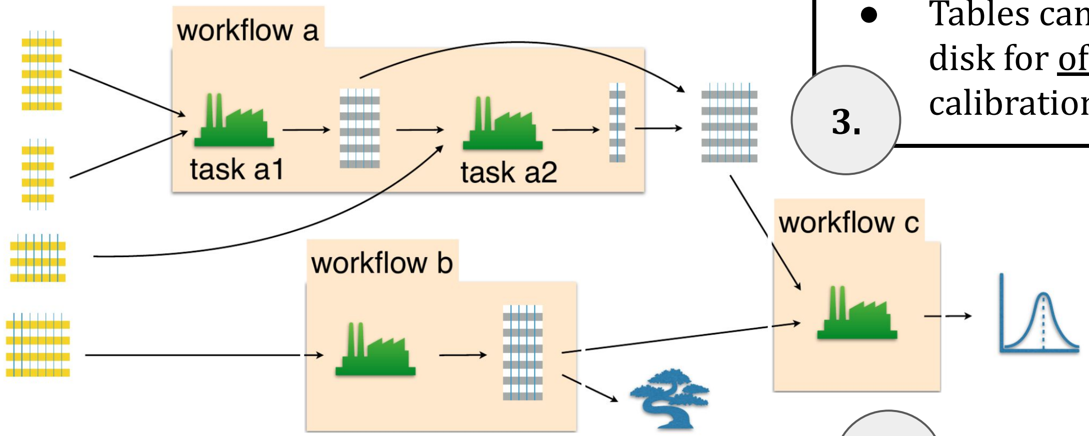
2.

- Analysis task: **struct**
- Tasks organised in **workflows**
- 1 or more tasks in a single workflow

“[Introduction to O2/O2Physics](#)”
14th October

3.

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



4.

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

5.

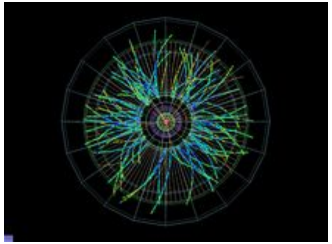
End of the chain: finale objects (TH1, TH2, THnSparse, TEfficiency, …) stored in **AnalysisResults.root**

- **Step 0: Event selection**

1.

Tracks

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

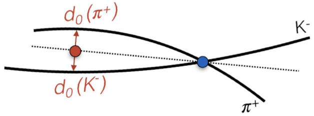


2.

Candidate reconstruction and filtering

- 2/3 track loop to reconstruct candidate from 2/3 - body decays
- Secondary vertex determination
- Filtering with “loose” selections

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



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

3.

Candidate selection

- Application of analysis selections (topological variables, track PID)
- (MC) matching to generated particles

4.

Physics analysis

- Store information into analysis objects (TH1, TH2, THnSparse, TTree, …)

Perform the analysis:

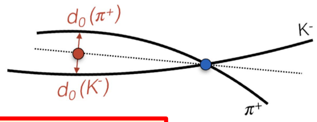
- M_{inv} fits → signal
- Corrections (eff × acc, f_{prompt})
- Systematic uncertainties
- Results

HF O^2 analysis framework

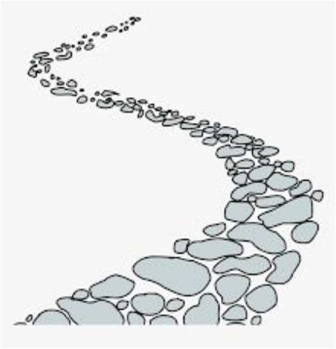
Tracks

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

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



Let's go through all the steps…



mfaggin@cern.ch

1.

Track index skim creator

Doing a $D^0 \rightarrow K^- \pi^+$ 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, $\cos\theta_p$,...)

HF derived data

Output: pairs/triplets of track indices for candidates (**Hf2Prongs**)

PWGHF/DataModel/CandidateReconstructionTables.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

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

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 c/b 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_track_index::Prong0Id, hf_track_index::Prong1Id,
                  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.

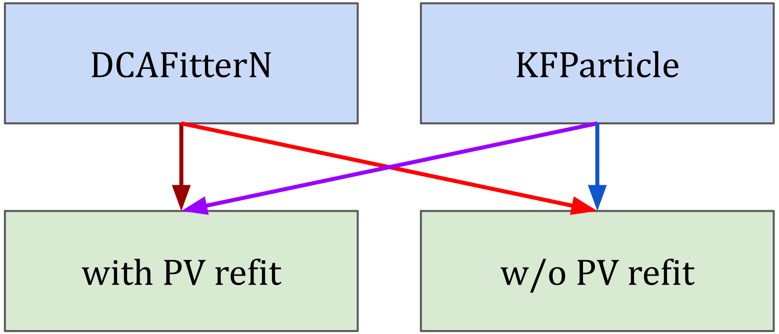
Candidate creator

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**

Beware

- **Different tables** created! Binary choice, be consistent with it during next steps (i.e. `candidate-selector,task`)
- Currently available **only** for **2-prong** candidates

`processPvRefitWithDCAFitterN`

`processNoPvRefitWithDCAFitterN` (default)

`processPvRefitWithKFParticle`

`processNoPvRefitWithKFParticle`

PWGHF/TableProducer/candidateCreator2Prong.cxx

mfaggin@cern.ch

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 (**Hf2Prongs**)
- Track PID

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 either as D^0 or D^0bar

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}, $\cos\theta_p$, …

<u>Input</u>:
- Selected candidates (**HfSelD0**)
- MC particles
- MC flags

Analysis task
- Histogram filling for selected candidates

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

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 selected as D^0(bar) 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 $D^0 \rightarrow K^- \pi^+$ 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 selected as D^0(bar) 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`

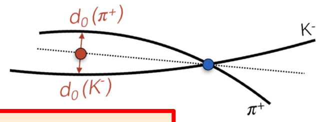
… and equivalent for MC

```
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

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.

A more complex analysis: $B^+ \to D^{0bar}\pi$

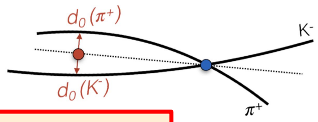
mfaggin@cern.ch

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

Tracks

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

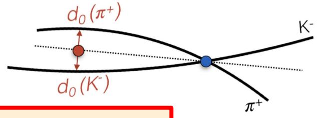
More complex analyses

mfaggin@cern.ch

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

Tracks

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



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

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

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

… many more analyses are possible!

2-prong: D^0 → B^+

3-prong: D^+, D^{*+}, D^+_s, Λ^+_c, Ξ^+_c → B^0, D_{s1}^+, D_{s2}^{*+}, $\Sigma^{0,++}_c$, Ξ^{++}_{cc}

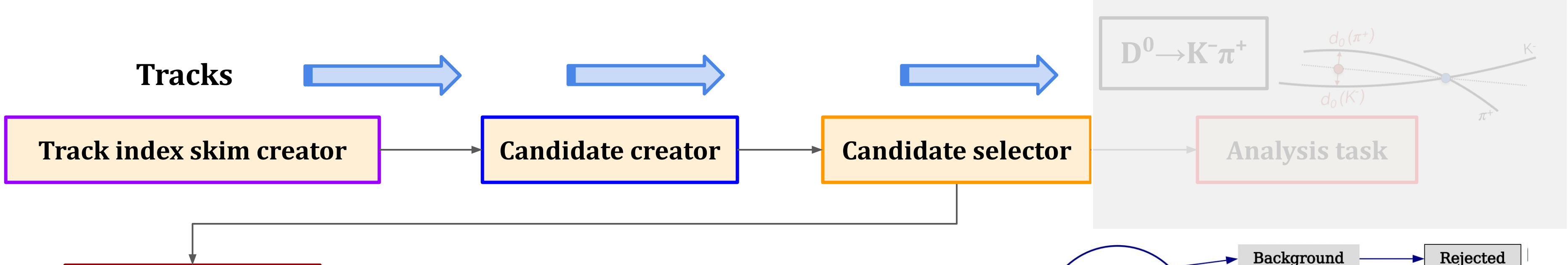
$D^0 \to \pi^+ K^-$
$B^0 \to D^- \pi^+$
$B^+ \to D^{0bar} \pi^+$
$D^+ \to \pi^+ K^- \pi^+$
$D^+_s \to \pi^+ K^- K^+$

$D_{s1}^+ \to D^{*+} K^0_s$
$D_{s2}^{*+} \to D^+ K^0_s$

$\Lambda^+_c \to p K^- \pi^+, p K^0_S$
$\Sigma^{0,++}_c \to \Lambda^+_c \pi^{-,+}$

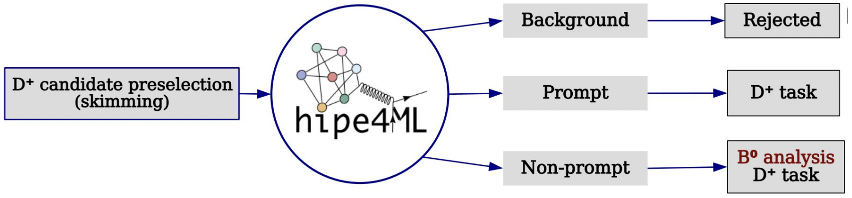
$\Xi^+_c \to p K^- \pi^+$
$\Xi^{++}_{cc} \to \Xi^+_c \pi^+$

Tree creators and Machine Learning

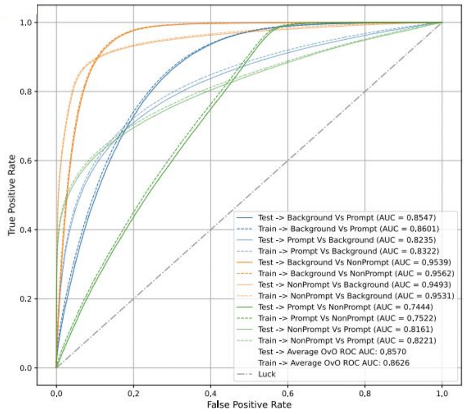


$D^0 \to K^- \pi^+$
$D^+ \to \pi^+ K^- \pi^+$
$D_s^+ \to K^- K^+ \pi^+$
$\Lambda_c^+ \to p\, K^- \pi^+$, $p\, K^0_S$
$\Xi_c^+ \to p\, K^- \pi^+$
$\Xi_c^0 \to \Xi^- \pi^+$
$\Xi_{cc}^{++} \to \Xi_c^+ \pi^+$

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



- Tree creator: reduced tables on output AO2Ds.root files
- 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



Tree creators and Machine Learning

mfaggin@cern.ch

ALICE

Tracks

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

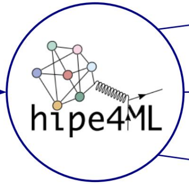
$D^0 \to K^- \pi^+$

Candidate selector → Tree creator

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

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

D⁺ candidate preselection (skimming) →



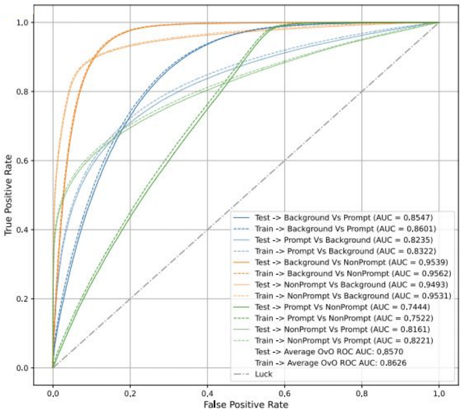
→ Background → Rejected
→ Prompt → D⁺ task
→ Non-prompt → B⁰ analysis D⁺ task

Tuesday 15th October

11:30

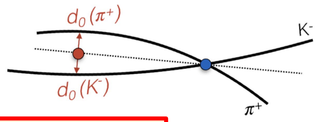
Machine learning in O2/O2Physics: general introduction

Speakers: Fabio Catalano (CERN), Francesco Mazzaschi (CERN)



HF O^2 analysis framework - dependencies

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



Tracks

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

Workflows specific of HF framework

Monday 14th October

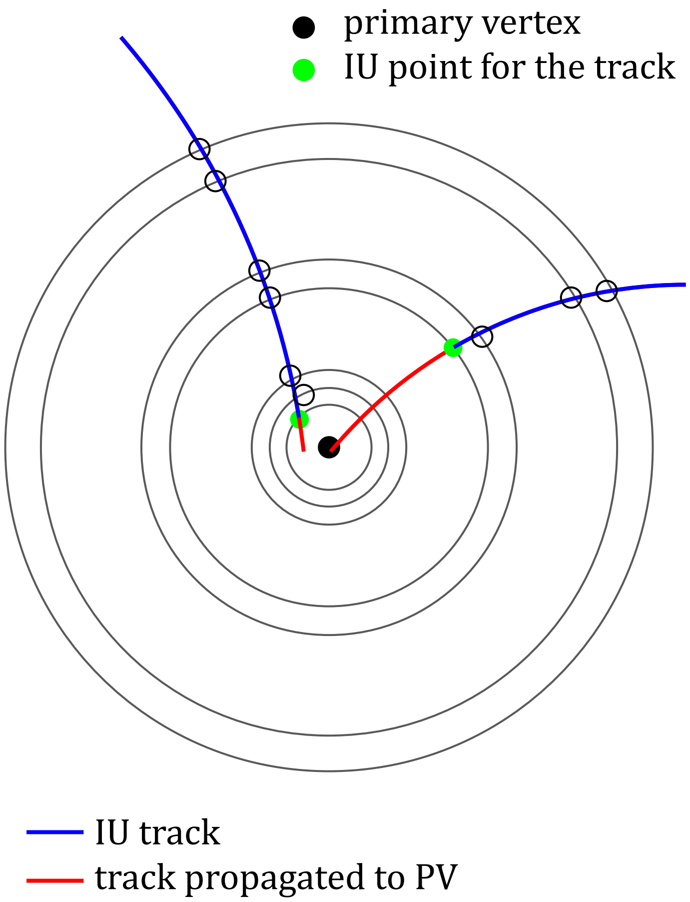
11:45 DPG Talk: Analysis Objects and Tools (AOT)

Dependencies

Track-To-Collision-Associator **3.** ← Track-Selector **2.** ← Track-Propagator **1.**

Track propagator

mfaggin@cern.ch



- 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 by the track-propagation workflow

`processStandard`

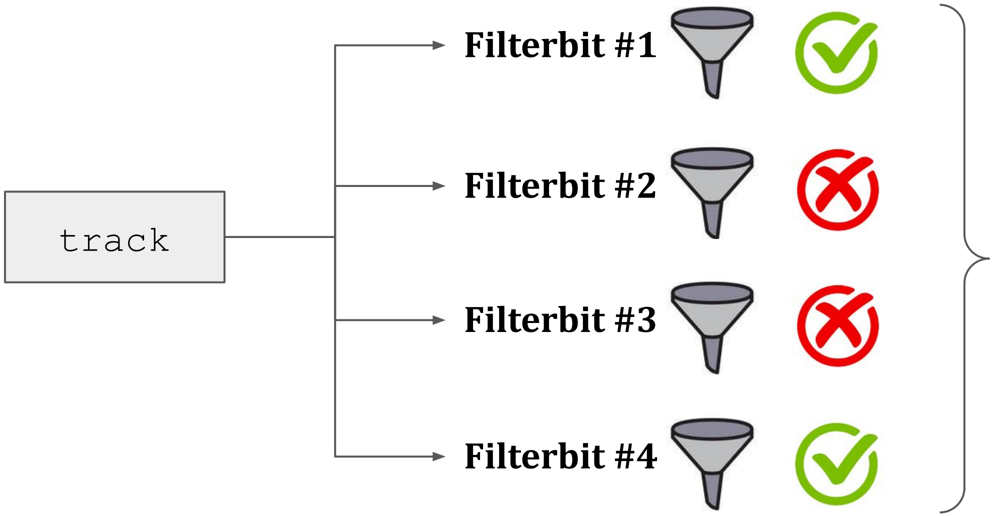
- track parameters
- dca_{XY}, dca_{Z} values

`processCovariance`

- track parameters and covariance matrix
- dca_{XY}, dca_{Z} values and uncertainties

→ much more resources consumed!

Predefined sets of selections: Common/Core/TrackSelectionDefaults.cxx



`aod::TrackSelection`

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

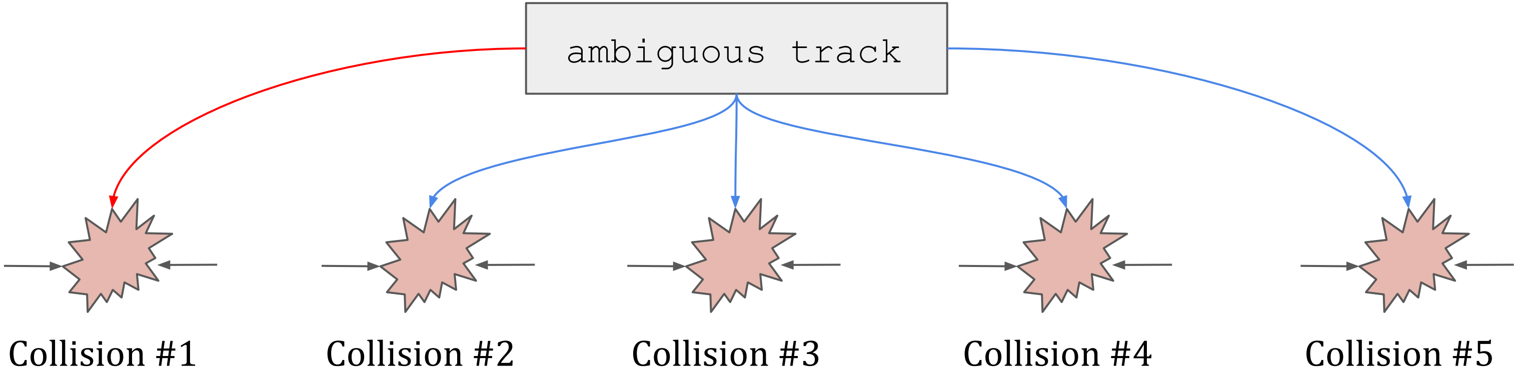
- For each (propagated!) track the following check is done: does it satisfy the selections defined in the *i*-th predefined set (filterbit)?
- Track-by-track filling of `aod::TrackSelection` table, according to the responses

- 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!)

Track-to-collision associator

Common/TableProducer/trackToCollisionAssociator.cxx

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

Linked derived data

mfaggin@cern.ch

Tracks

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

Tracks

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

↓

Original AO2Ds ← 🔗 ← **Derived AO2Ds**

Event and track properties

2,3-prong candidate daughter indices

Execution for each big dataset once for all by PWG conveners

Start (shorter) analysis workflow from here!

- Preparation ongoing on different data and MC datasets
- **Self-contained** derived data creation possible for multi-staged analyses (e.g. B mesons) → **to be produced by the analysers**

- **Derived AO2Ds**: .root files saved on disk containing the **table with indices of candidate 2,3-prong daughters**
- Each analysis runs on these derived AO2Ds, w/o the need to rerun the combinatorics anymore!
- 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**

DO NOT MISS!

09:30 Production and usage of derived data in HF

Speaker: Fabrizio Grosa (CERN)

References and useful information

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

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

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

Validation framework & postprocessing analysis tools:
https://github.com/AliceO2Group/Run3Analysisvalidation

Thanks a lot for your attention …

… and enjoy the hands-on session!

Backup