ALICE

CERN

YEARS / ANS CERN
1954-2024

DPG/Analysis Objects and Tools (AOT)

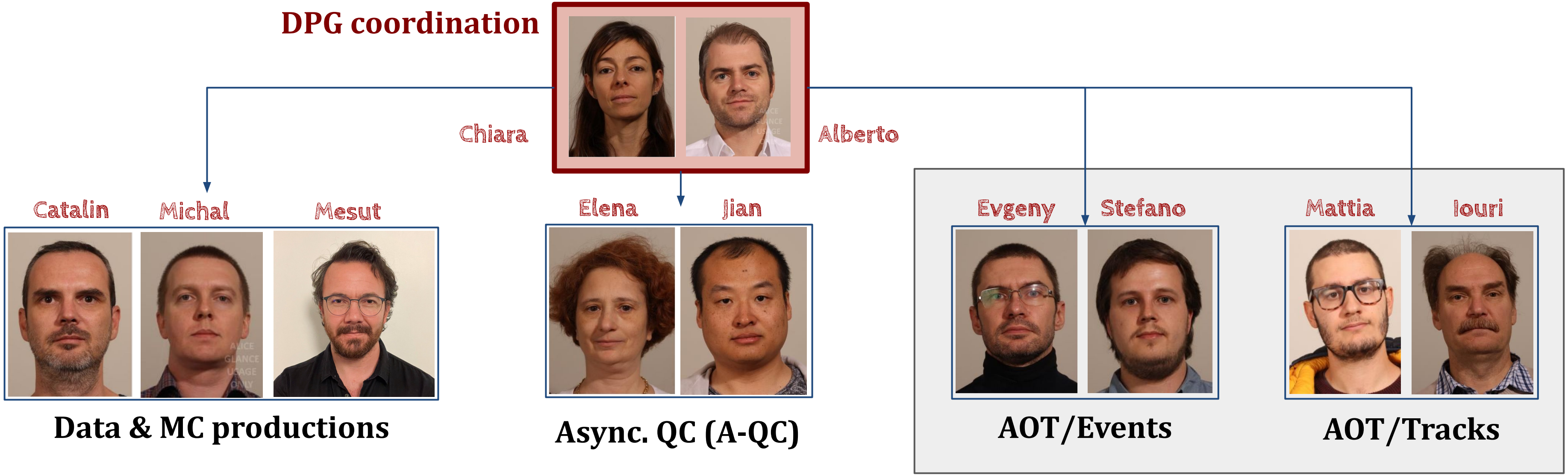
Iouri Belikov, University of Strasbourg (FR)
Evgeny Kryshen, NRC Kurchatov Institute PNPI (RU)
Mattia Faggin, CERN (CH)
Stefano Trogolo, University and INFN Torino (IT)

O^2 analysis tutorial 4.0
14th October 2024

Data Preparation Group (DPG): who we are

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

ALICE



- Responsible for steering and coordinating the reconstruction of the data collected by ALICE and the preparation and the execution of the Monte Carlo simulations, and of organizing the Quality Assurance of the reconstructed and simulated data

Analysis Objects and Tools (AOT)

- In charge of the Analysis Objects and Tools for events and tracks characterization (AOT/Events, AOT/Tracks), which includes the production, maintenance, quality assurance and bookkeeping of the AOD (Analysis Object Data) files, as well as the coordination of the groups working on event selections and properties and track selections and properties

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

AOT = Analysis Objects and Tools

Focus of this talk: common **tools**/utilities for your analysis with O2Physics

AOT/Events

- Event selection
- Event-plane determination
- Multiplicity and centrality calibration

Extra

- Particle Identification (PID)

AOT/Tracks

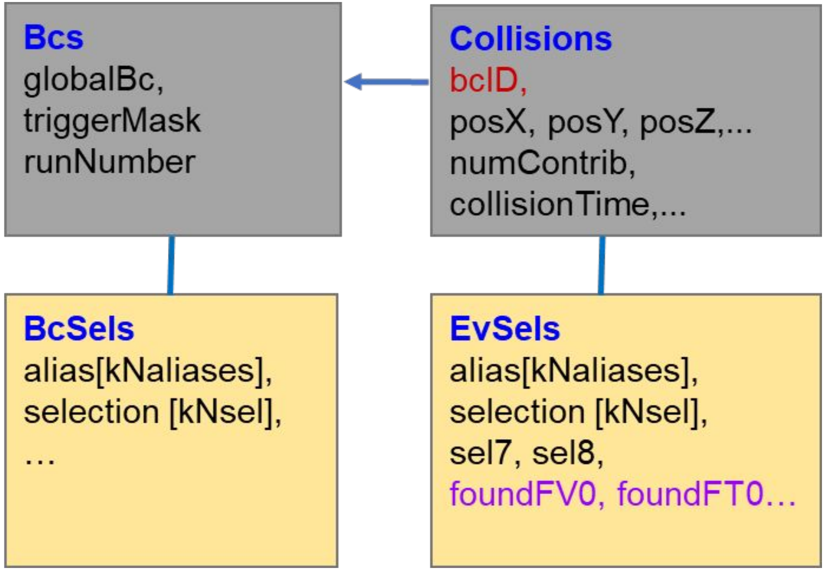
- Track propagation to the primary vertex
- Track selections
- Track-to-collision association
- Primary-track DCA track smearing in MC (trackTuner)

DPG/AOT-Event

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

- **`EvSels`** table joinable with **`Collisions`** table. To be used in analyses based on loops over Collisions (primary vertices), i.e. majority of ALICE analyses.
- **`BcSels`** table joinable with **`BCs`** table. To be used in analyses based on loops over BCs table such as muon arm UPCs, luminosity monitoring etc.
- Main contents:
  - `aliases[kNaliases]` : fired trigger aliases (trigger classes)
  - `selection[kNsel]` : decisions on single selection criteria
  - `sel7`, `sel8` (historical names): selection decisions = logical AND of several selection criteria
  - `foundFV0`, `foundFT0`, `foundBC`: indices to FV0, FT0 and BC entries matched to current collision

In **`EvSels`** table only

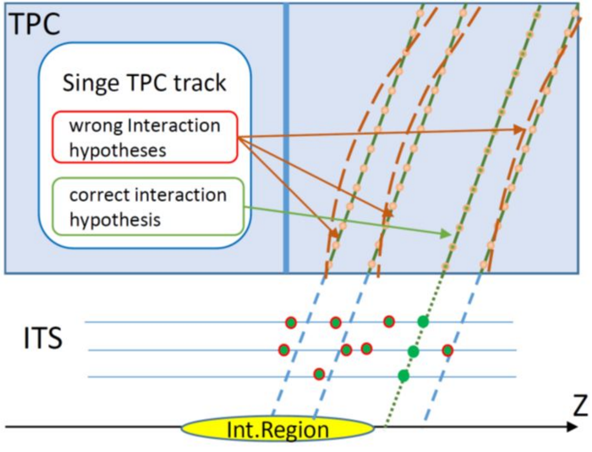


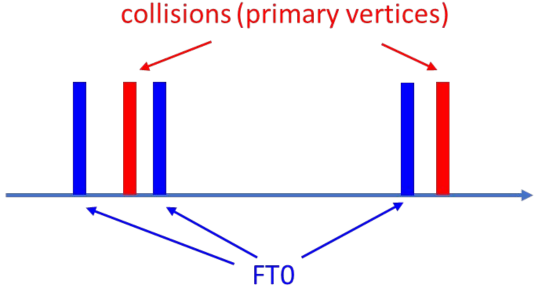
Code (and more details): O2Physics/Common/DataModel/EventSelection.h

Event selection - main challenge in Run 3

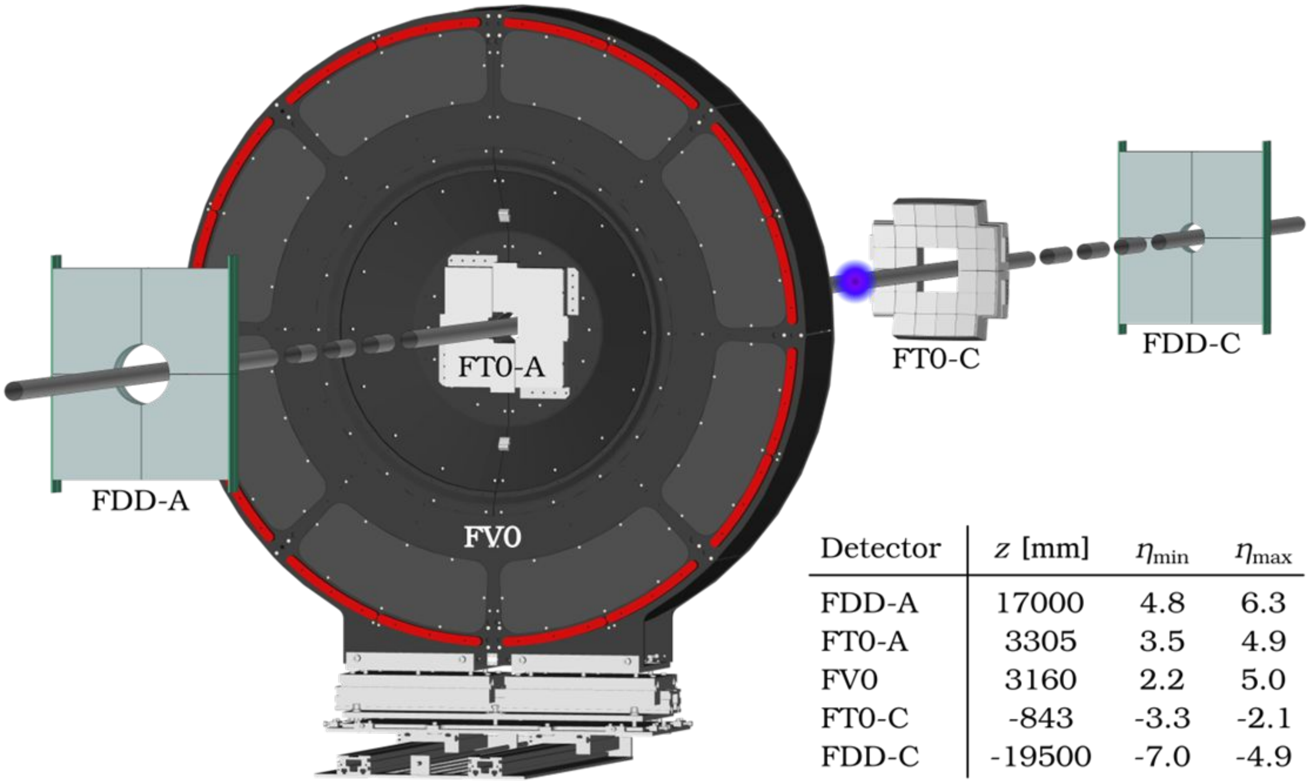
jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

- TPC: tracks are drifting towards endcaps (~100 μs drift time)
  - z-time ambiguity for TPC-only tracks
- ITS integration time ~5 μs in pp (15 μs in Pb–Pb)
  - Several overlapping events (in 650 kHz INEL pp runs)
  - No precise timestamp
- z-time ambiguity for TPC tracks can be resolved via:
  - ITS-TPC matching → ~100 ns resolution
  - TOF matching → precise timing (resolution < 1 ns).
- Collision time uncertainty depends on
  - time resolution of single tracks
  - number of contributors
- Event selection challenge: most probable collision bc is not precise and might be shifted wrt bc with corresponding FIT signals
  - Might be a problem in high-rate environment
  - (e.g. typical distance between collisions in high-rate pp ~40 bcs)
- Solution: search for FIT info in neighbouring bcs and provide `foundBC`, `foundFT0`, `foundFV0` indices + flags, trigger aliases and decisions corresponding to foundBC





jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch



Detector	z [mm]	η_{min}	η_{max}
FDD-A	17000	4.8	6.3
FT0-A	3305	3.5	4.9
FV0	3160	2.2	5.0
FT0-C	-843	-3.3	-2.1
FDD-C	-19500	-7.0	-4.9

Using T0-vertex signal (TVX) - coincidence of FT0A and FT0C signals + good timing:

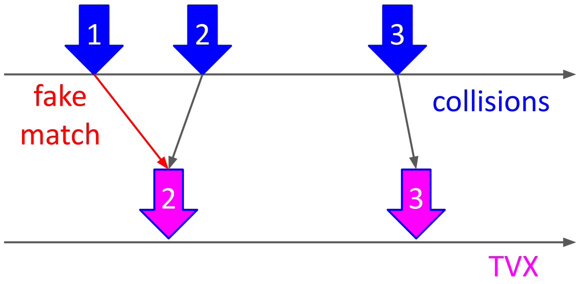
TVX ≈ FT0A & FT0C

Event selection - collision-to-FT0 matching

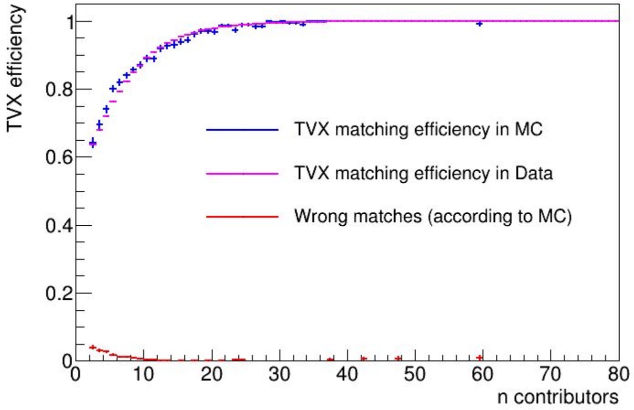
jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

`sel8 = kIsTriggerTVX & kNoTimeFrameBorder & kNoITSROFrameBorder` since April

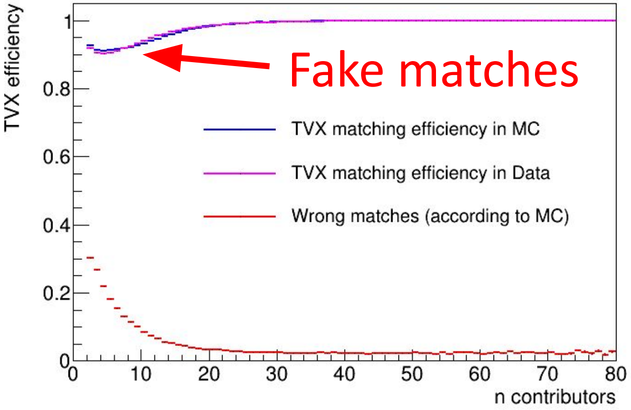
- Collision time is not known precisely (up to ~100 bc uncertainties)
- Event selection tries to find closest bc with TVX (FT0-vertex activity)
  - Works well at low IR ~ 10 kHz (average TVX efficiency ~ 90%)
- BUT: large fraction of fake matches at high IR, especially at low mult…
- Use low IR to cross check results/normalization at high IR!



Low IR: LHC22q_apass4



High IR: LHC24ag_apass1 (551504)



$$\varepsilon = \frac{\#\,\mathrm{colls}^{\mathrm{matched\ to\ TVX}}_{\mathrm{kNoTF\ \&\ kNoITSROF}}}{\#\,\mathrm{colls}_{\mathrm{kNoTF\ \&\ kNoITSROF}}}$$

Event selection - kNoTimeFrameBorder

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

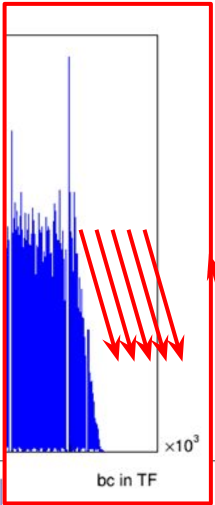
ALICE

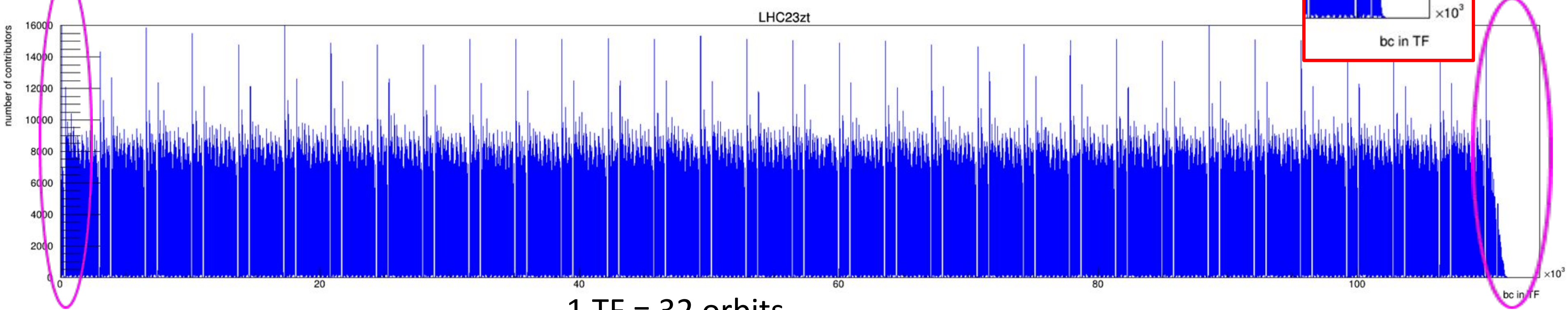
`sel8 = kIsTriggerTVX & kNoTimeFrameBorder & kNoITSROFrameBorder` since April

- Incomplete information in TPC at the borders of time frame
- `kNoTimeFrameBorder` cut rejects:
  - 300 bcs in the beginning of TF
  - 4000 bcs at the end of TF
- Time frame duration:
  - 2022: TF = 128 orbits = 128 x 3564 bcs → ~1.1% rejected
  - 2023-24: TF = 32 orbits = 32 x 3564 bcs → ~3.7% rejected

Technical: 32 spikes ↔ 1 per orbit

- different bunch-by-bunch lumi → spike in bunches with higher probability to have a collision
- features of collision-to-bc association, e.g. all ITS-only vertices appear exactly in the middle of ITS ROF → spike.



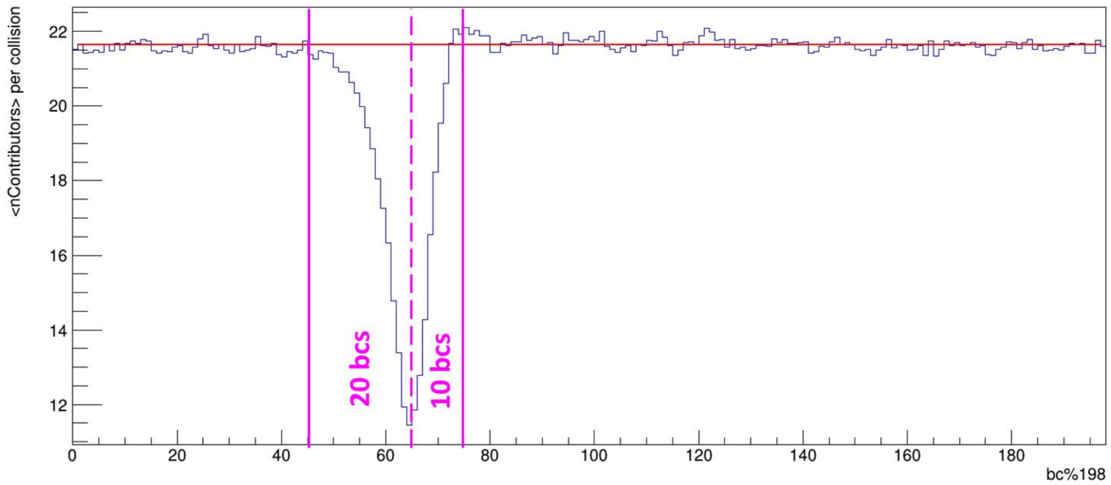


1 TF = 32 orbits

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

`sel8 = kIsTriggerTVX & kNoTimeFrameBorder & kNoITSROFrameBorder` since April

- ITS cluster loss on the ROF boundary due to the ALPIDE time walk
- `kNoITSROFrameBorder` cut rejects:
  - 20 bcs at the end of ITS RO frame
  - 10 bcs in the beginning of ITS RO frame
- ITS RO frame duration:
  - pp: 198 bcs → ~15% rejected
  - PbPb: 594 bcs → ~5% rejected



jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

- add `EventSelection.h` header:

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

- join `Collisions` and `EvSels` tables and use corresponding iterator as an argument of the process function:

```
void process(soa::Join<aod::Collisions, aod::EvSels>::iterator const& col, ...)
```

- check trigger aliases for Run2 data or triggered Run3 data (EMCAL, PHOS, TRD, HMPID):

```
if (!col.alias()[kINT7]) {
  return;
}
```

Bypass this check for MC or continuous Run3 data

- apply offline selection criteria:

Run 2:

```
if (!col.sel7()) {
  return;
}
```

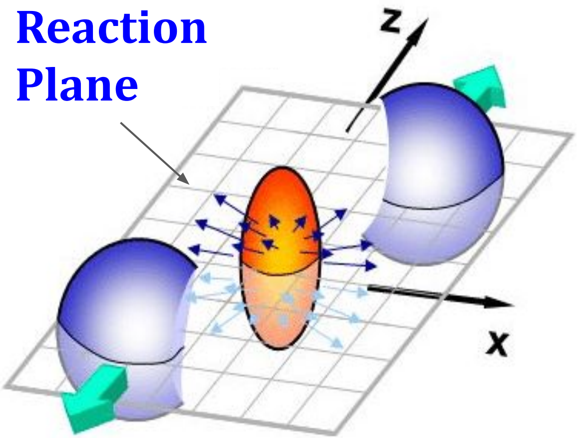
Run 3:

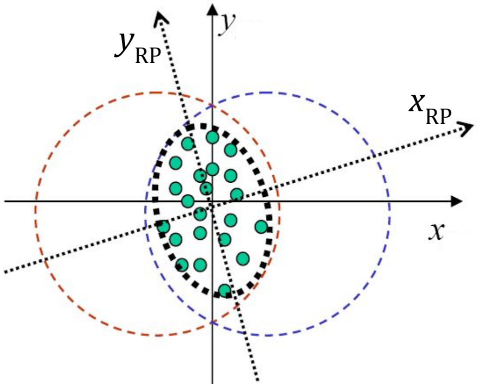
```
if (!col.sel8()) {
  return;
}
```

- run your tasks in stack with timestamp and event-selection tasks:

```
o2-analysis-timestamp --aod-file AO2D.root -b | o2-analysis-event-selection -b | o2-analysis-user-task -b
```

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch





- In heavy-ion collisions:
  - **colliding Pb nuclei** can overlap partially
  - geometrical anisotropy → momentum anisotropy
  - origin of azimuthal anisotropy

- In **anisotropic flow** analyses:
  - **Reaction plane** → plane formed by the impact parameter vector b and the collision axis
  - In practice we cannot find RP but can only approximate and this approximation is known as **Event Plane**
  - **Event Flow Vector "Q_n"** and **Event Plane Angle ψ_n** from the n^{th} harmonics are defined as:

$$Q_{n,x} = \sum_i \omega_i \cos(n\varphi_i) \quad Q_{n,y} = \sum_i \omega_i \sin(n\varphi_i)$$

$$\Psi_n = (1/n) \arctan\left(Q_{n,y}/Q_{n,x}\right)$$

where ω_i is weights and φ_i is particle's azimuthal angle

Event plane (Q-vector) framework (2/3)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

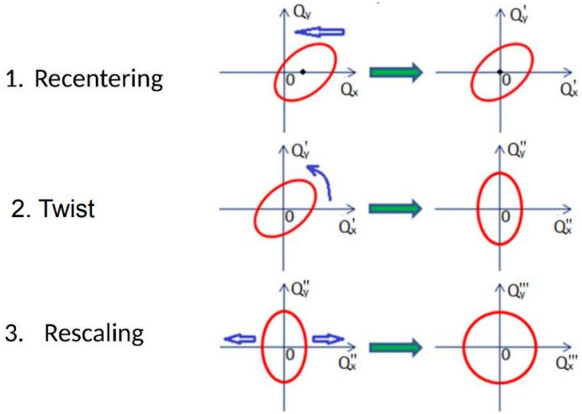
How do we determine EP (Q-vector) in Run 3?

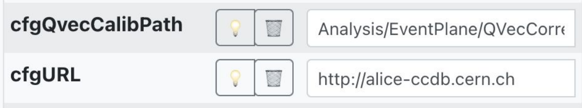
- Use FV0A, FT0A, FT0C, TPCneg, TPCpos, TPCall detectors
- 1st step correction: Gain equalization
- 2nd step correction: Recentering, twisting, rescaling

Provided by DPG-AOT/Event group

How do we handle EP (Q-vector) in Run 3?

- Code: `Common/TableProducer/qVectorsTable.cxx`
- Workflow: `o2-analysis-qvector-table`
- Available in **Core Service Wagons** → **use it!**
- A few tips in case you want to configure a personal wagon
  - **cfgCentEsti** → set the centrality estimator and must be the same used in all your analysis
  - **cfgCorrLevel** → set the correction steps to be applied (i.e. 4 = GainEq, Rec., Twist., Rescal.)
  - **cfgnMods** → set the harmonics to be computed at the same time (available up to 4^{th} harmonic)
  - **cfgURL** → http://alice-ccdb.cern.ch always use central ALICE CCDB
  - **cfgQvecCalibPath, cfgGainEqPath** → set the path to the correction uploaded in the central ALICE CCDB





Event plane (Q-vector) framework (3/3)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

ALICE

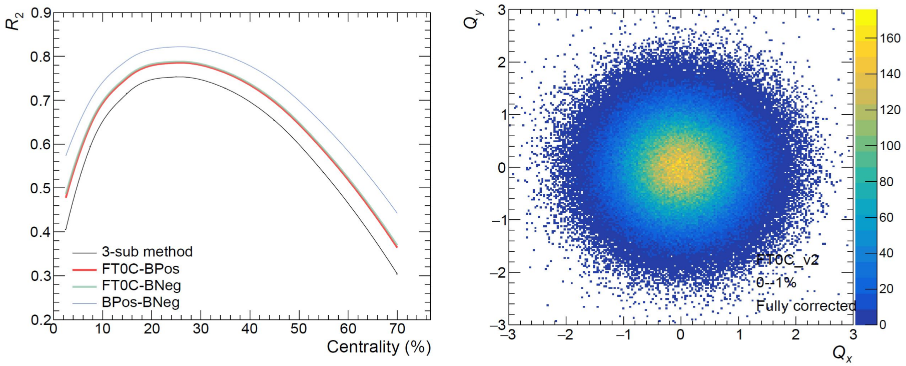
How do we handle EP (Q-vector) in Run 3?

- few tips in case you want to configure a personal wagon
  - **cfgUseXXXX** → set detectors used for Q-vector/EP determination (i.e. 1= use; 0= do not use)
    XXXX = FV0A, FT0A, FT0C, TPCneg, TPCpos, TPCall

If you write a new analysis task don't use Bpos, Bneg, Btot (to be removed soon) → use TPCneg, TPCpos, TPCall instead

In summary

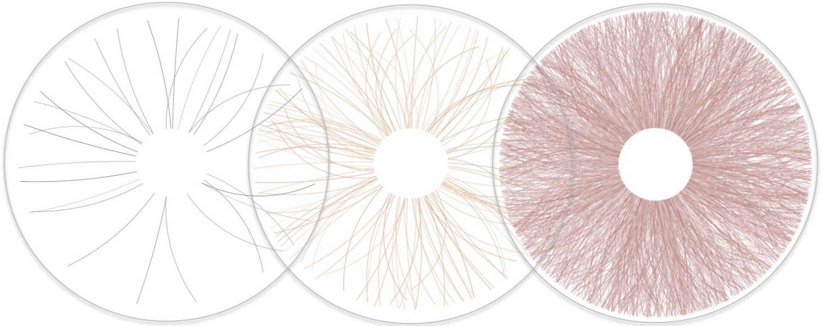
- add the `o2-analysis-qvector-table`workflow
- subscribe to qVectorXXXXVec tables in your analysis task (Common/DataModel/Qvectors.h)
  - bear in mind that each Q-vector has a real (Re) and an imaginary (Im) component
- Up to now: Q-vector calibration up to Ψ_4 with 1%-centrality
  intervals available run-by-run for PbPb apass3
  - apass4 soon available on CCDB→ it will be announced

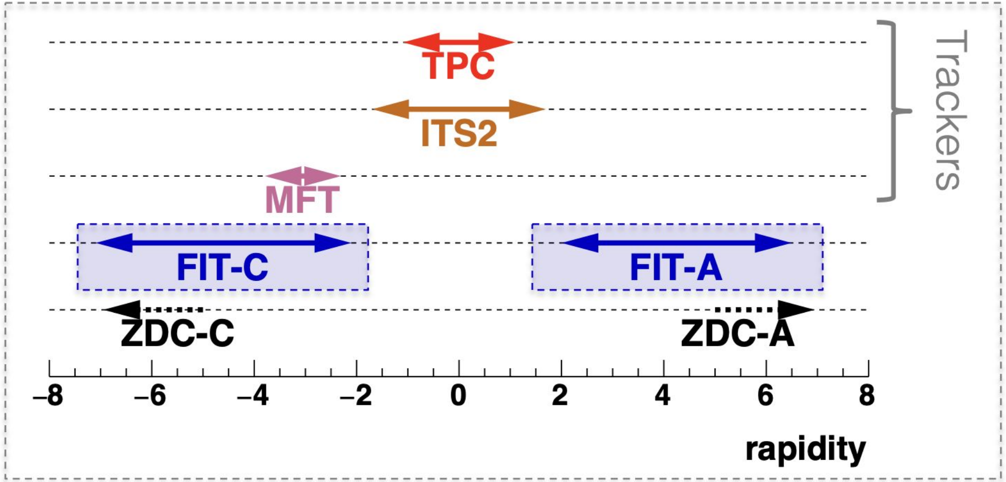


Example of R2 calculation in the
Scalar Product method with Q-vector

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

- Many analyses need the **event multiplicity/centrality**:
  - Study an **observable** as a function of multiplicity/centrality
  - Other **tasks depends on multiplicity/centrality** selection (e.g. PID, Q-vector, …)



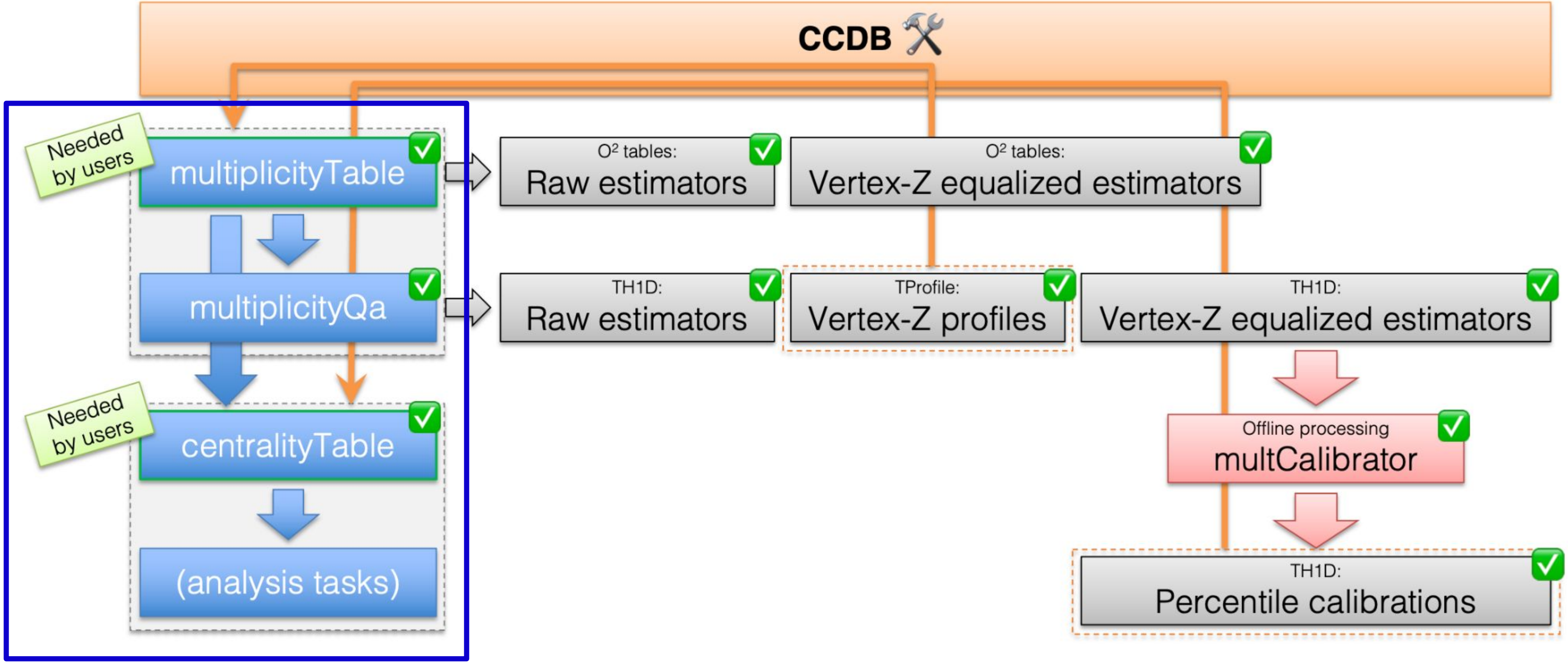


- Detectors used for multiplicity/centrality:
  - FT0: $-3.3 < \eta < -2.1$, $3.5 < \eta < 4.9$
  - FV0: $2.2 < \eta < 5.0$
  - FDD: $-6.9 < \eta < -4.9$, $4.7 < \eta < 6.3$
  - Central barrel detectors → number of tracks used to fit the primary vertex (N_PV_{tracks})

Multiplicity/Centrality (2/3)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

- A complex procedure to get the calibration but only a **small part is for analysers**
  - calibrations are provided centrally by DPG-AOT/Events-group via central ALICE CCDB



jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

Code: Common/TableProducer/multiplicityTable.cxx
Common/TableProducer/centralityTable.cxx

- If you need the **multiplicity**
  - add the `o2-analysis-multiplicity-table` workflow
  - subscribe to MultiplicityTable → it is possible to subscribe a table with all estimators
  - as an example

```
using CollisionCandidates = soa::Join<aod::Collisions, aod::EvSels,
aod::FT0MultZeqs, aod::MultZeqs>;

const float multiplicity = collision.multFT0C();
```

Multiplicity equalized for the vertex position with the FT0 detector

multiplicity equalized for all estimators

- If you need the **centrality**
  - add the `o2-analysis-centrality-table` workflow
  - subscribe to CentralityTable → need to specify the estimators in the table subscription
  - as an example

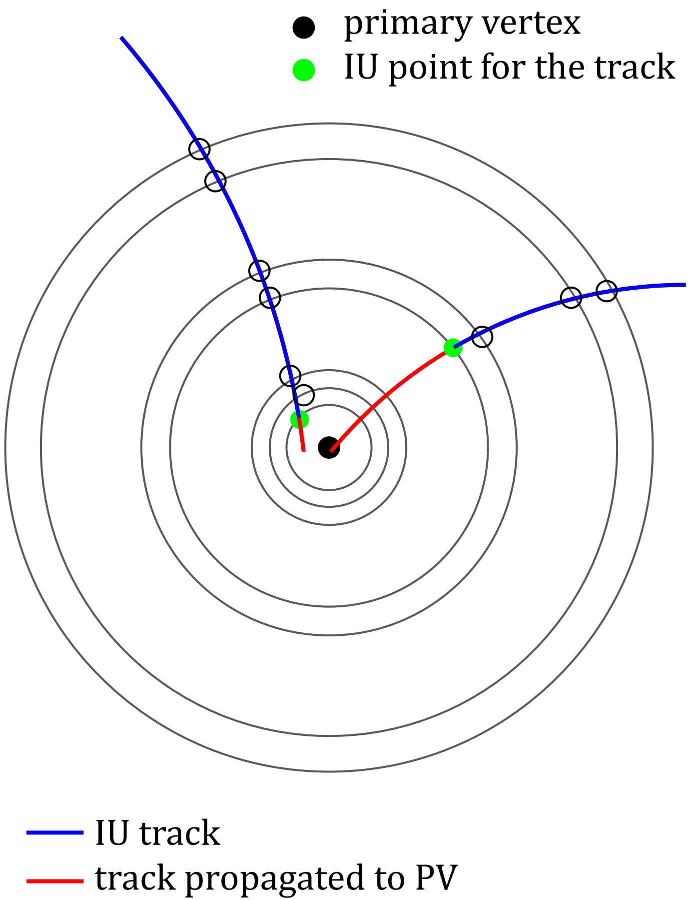
```
using CollisionCandidates = soa::Join<aod::Collisions, aod::EvSels,
aod::CentFV0As, aod::CentFT0Ms, aod::CentFT0As, aod::CentFT0Cs>;

const float centrality = collision.centFT0M();
```

DPG/AOT-Tracks

Track propagation to the PV (1/2)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch



- In AO2Ds: **TracksIU**
  - **IU: Innermost Update** point
  - Track-parametrization at the IU written in AO2Ds → not the same radius for all tracks ↔ `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 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!

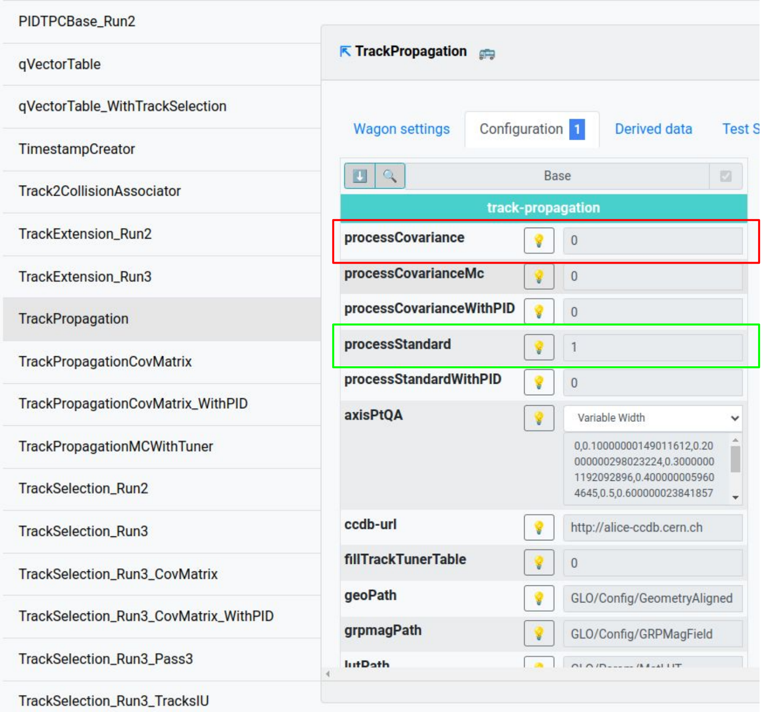
`process` functions available for analysis

Track propagation to the PV (2/2)

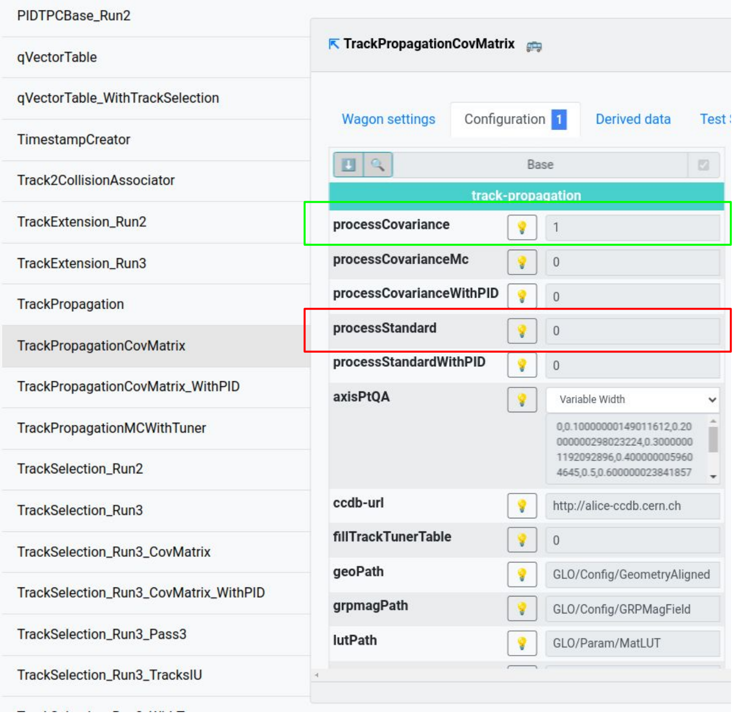
jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

Code: Common/TableProducer/trackPropagation.cxx
📚 Documentation here

Core Service Wagons/TrackPropagation



Core Service Wagons/TrackPropagationCovMatrix

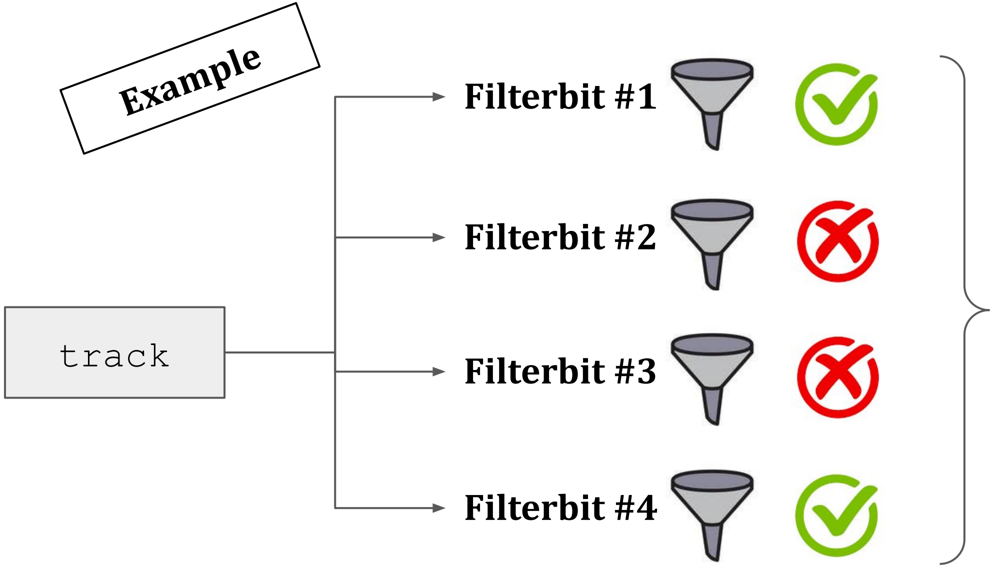


Other (nested) possible configurations: (1) use PID in tracking; (2) use DCA/p_T smearing in MC with the trackTuner

Track selector (1/2)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

- 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!)

`aod::TrackSelection`

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

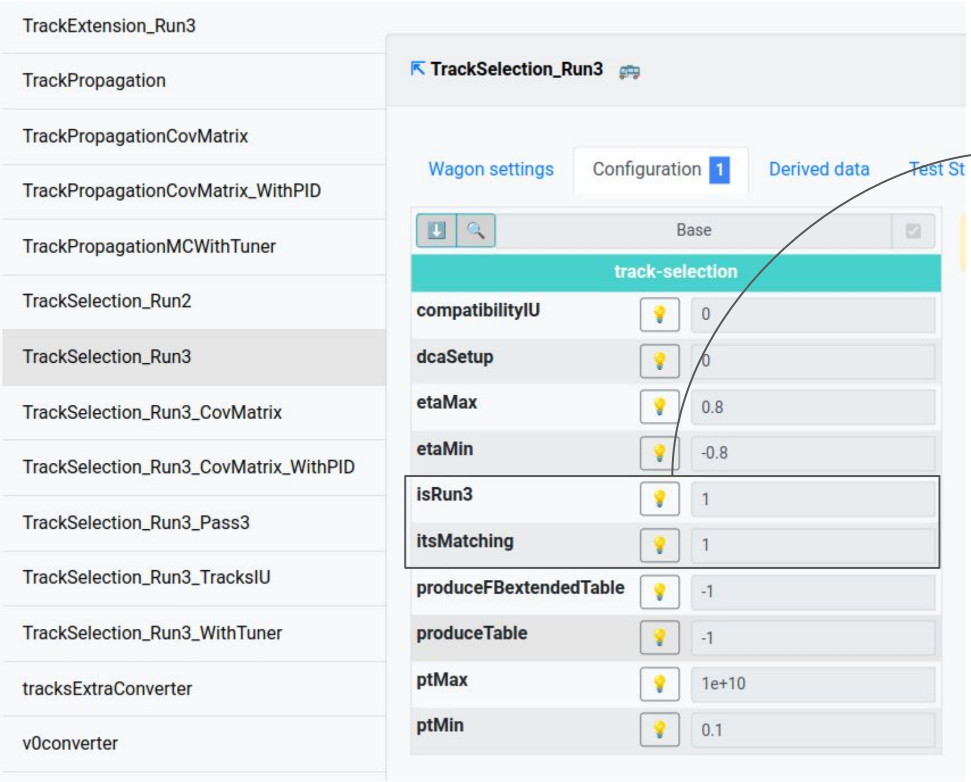
Predefined sets of pre-selections: `Common/Core/TrackSelectionDefaults.cxx`

Track selector (2/2)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

Code: Common/TableProducer/trackSelection.cxx
📚 Documentation here

Core Service Wagons/TrackSelection_Run3



Common/TableProducer/trackselection.cxx#L78-L84

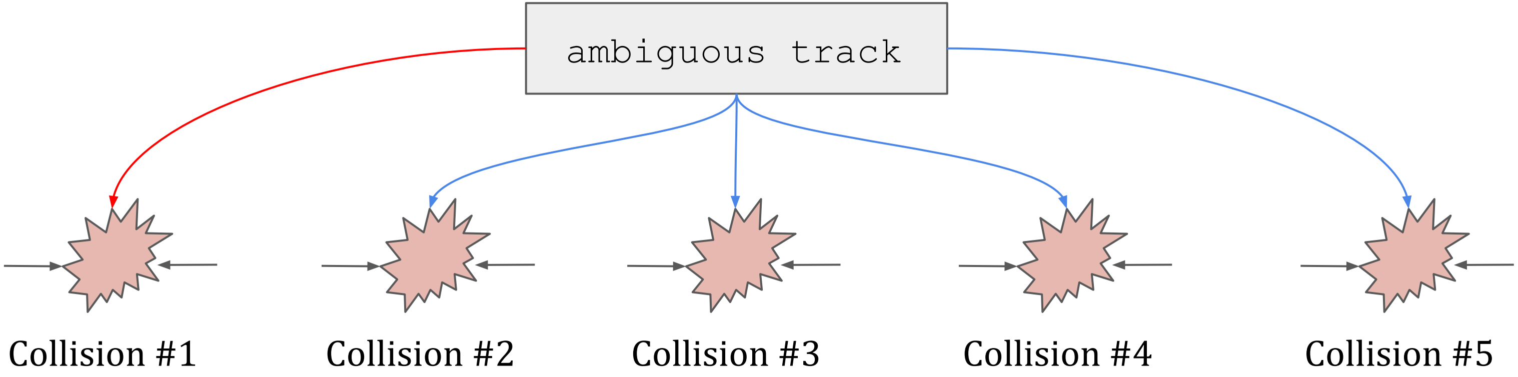
```
case 1:
        // Run 3 kAny on 3 IB layers of ITS
        if (isRun3) {
          [...]
          globalTracks =
getGlobalTrackSelectionRun3ITSMatch(TrackSelection::GlobalTrackRun3ITSMatching::Run3ITSibAny, dcaSetup.value);
          break;
        }
```

- By default, global track selections defined in Common/Core/TrackSelectionDefaults.cxx#L22-L40 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#L116-L123

Track-to-collision associator (1/2)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch



- Continuous readout → **ambiguous tracks**: tracks with more than 1 collision possible
- By <u>default</u>, in the <u>AO2D</u> the `track.collisionId()` is that of the <u>first compatible collisi</u> <u>in time</u>

- `track-to-collision-associator`: duplication of the track to each collisions compatible in time

😀 decay-reconstruction analyses: recovery of multi-prong candidates!
→ duplicates removable with analysis selections (topology)

🙁 single-track analyses: signal duplication
→ duplicates removed with specific analysis selections? To be studied in analysis

Shall I use the track-to-collision associator? And how?

It depends on your analysis!

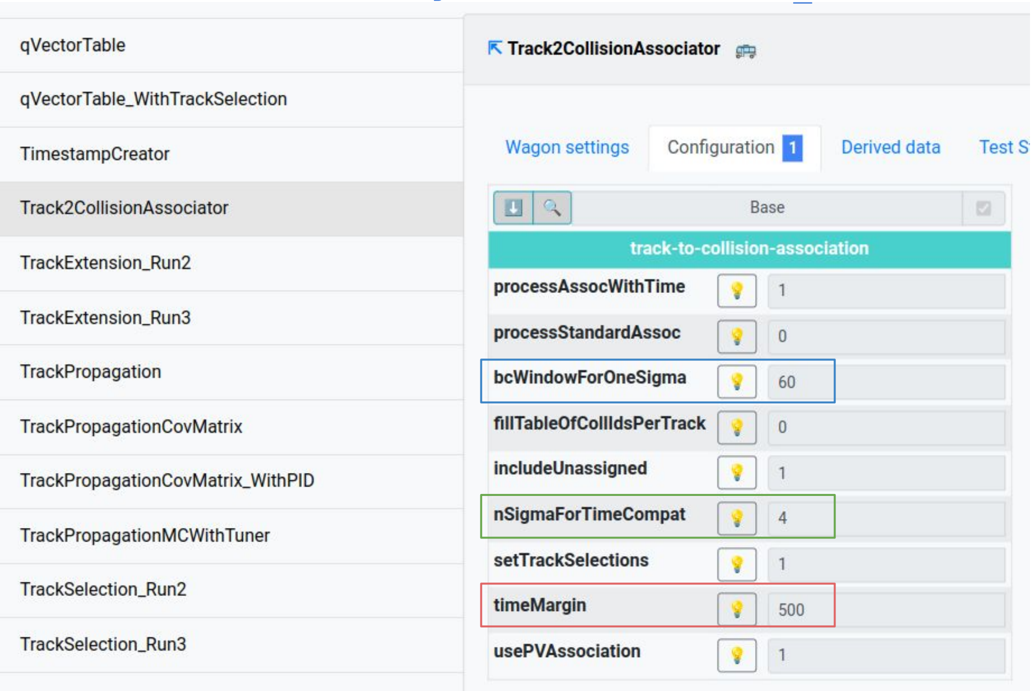
Track-to-collision associator (2/2)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

Code:

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

Core_Service_Wagons/TrackSelection_Run3



Common/Core/CollisionAssociation.h#L184

```
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)

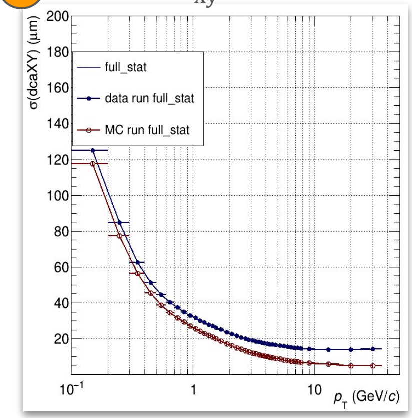
- Further margin of 500 ns (20 bc) to account for possible imperfections in TPC calibrations

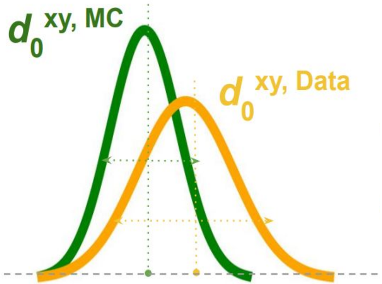
Track smearing in MC: the `TrackTuner` (1/2)

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

1

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



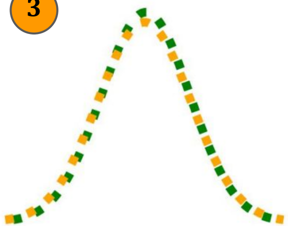


- `TrackTuner`: utility to smear track parameters in MC in order to better reproduce the performance in data
- Variables adjustable with the `TrackTuner`:
  - track impact parameter $r\varphi$, z
    - → smearing based on data-vs-MC comparison, calibrations provided centrally
    - → ❗ approach valid 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



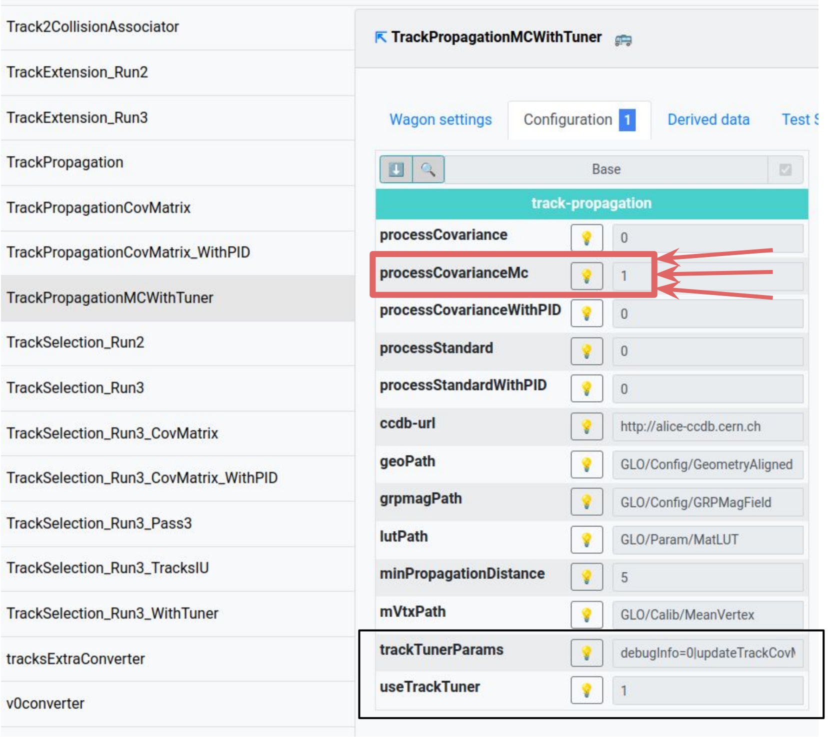
dca$_{xy}$ resolution in data and MC after the `TrackTuner`

Track smearing in MC: the `TrackTuner` (2/2)

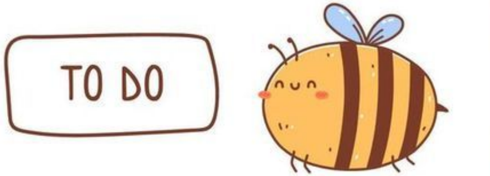
jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

Code: `Common/Tools/TrackTuner.h`
📚 Documentation here
📚 More info in this presentation

`Core Service Wagons/TrackPropagationMCWithTuner`



- `trackTunerParams`: string to configure the `TrackTuner` object
- See the documentation for the details



- [x] Configuration via long string
- [ ] Make it more user-friendly

Available dca_{xy}, dca_{z} parametrizations:

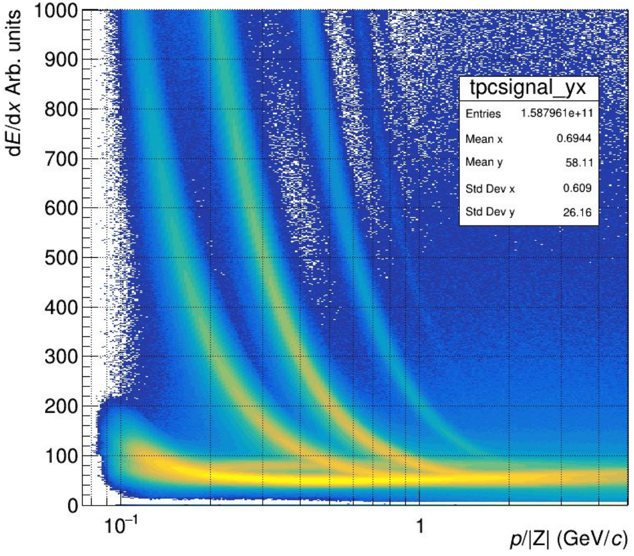
- proxy for pp: `trackTuner_DataLHC23fPass1_McLHC23k4b_run535085.root`
- proxy for PbPb: `trackTuner_DataLHC23zziPass3_McLHC24d2b_run544184.root`

NB: default `TrackTuner` in `track-propagation` workflow, but you can instantiate and customize it in your own analysis task!

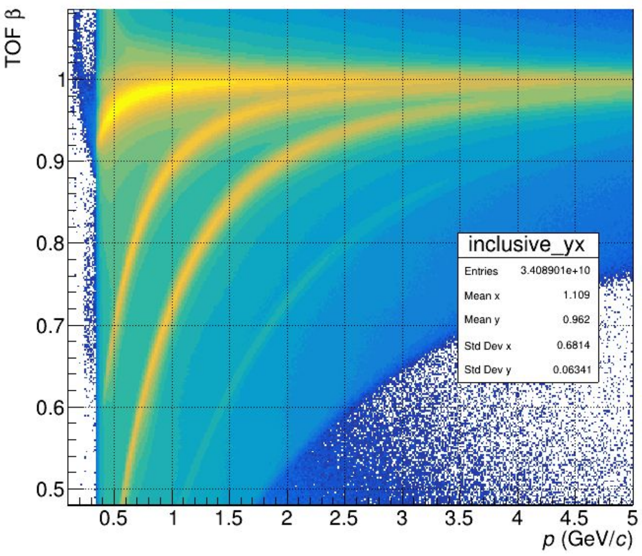
→ example: `PWGLF/TableProducer/Nuspex/nucleiSpectra.cxx#L458-L463`

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

PID in TPC via dE/dx



PID in TOF via time-of-flight



- Different particle species have separate distributions of dE/dx (TPC) and time-of-flight (TOF)
- In analysis: n_σ values are used

Tools/parameterizations provided by TPC, TOF experts (not by DPG!)

TPC

- Calibrations provided via Bethe-Block (BB) parametrizations and via Neural Network (use "NN" when available, list here)
- In MC the dE/dx observed in data is reproduced by random-sampling the dE/dx parametrizations from data → *"tune-on-data"*
  - n_σ centered at 0 and with σ=1

Extra - Particle identification (PID) in TPC and TOF (2/2)

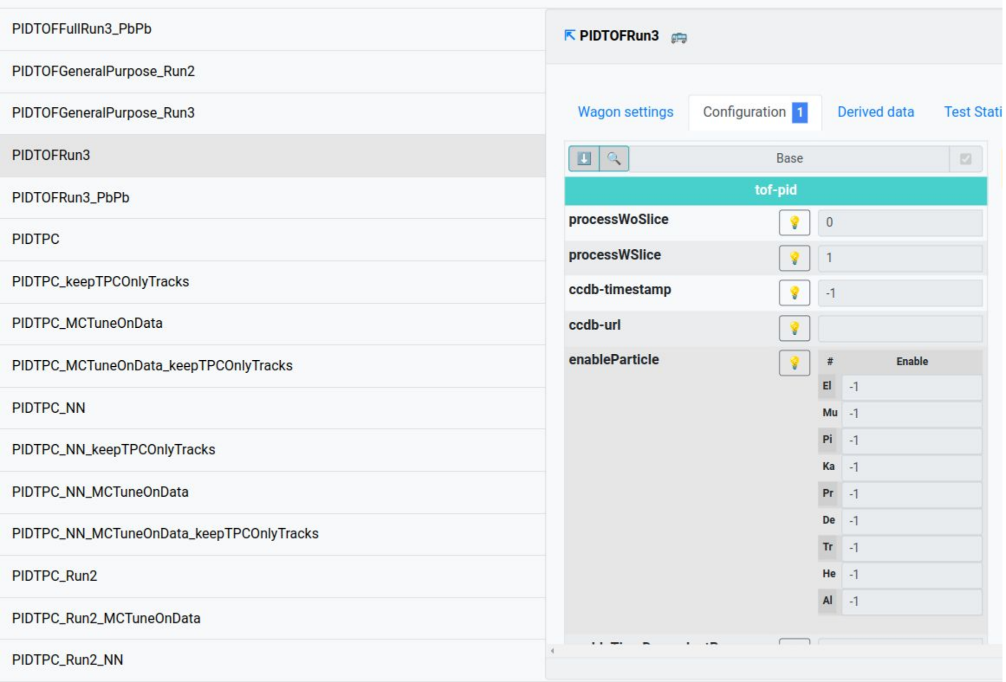
jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

Code:

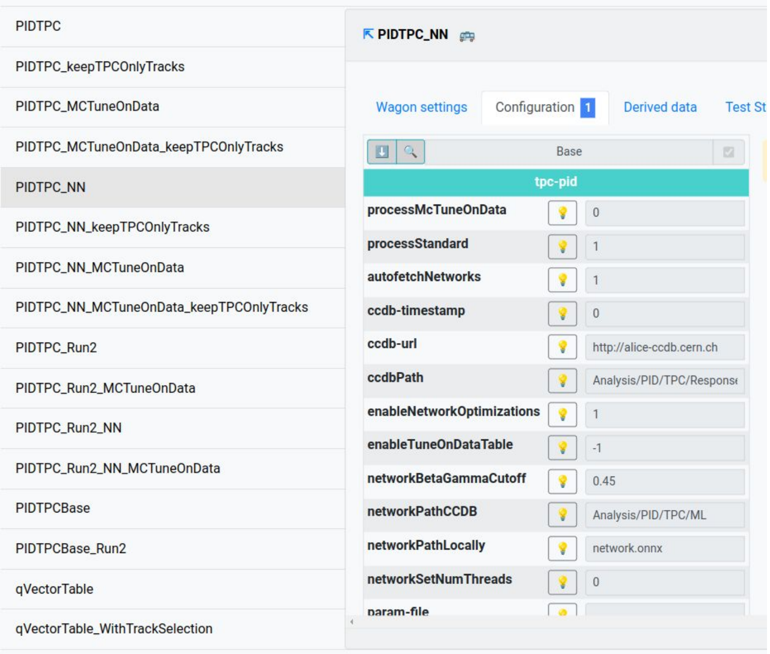
- Common/TableProducer/PID/pidTPC.cxx
- Common/TableProducer/PID/pidTOF.cxx

📚 Documentation here

Core Service Wagons/PIDTOFRun3



Core Service Wagons/PIDTPC_NN



- Version with "standard" parametrizations (no NN): Core Service Wagons/PIDTPC
- For MC: search for "tune-on-data" versions

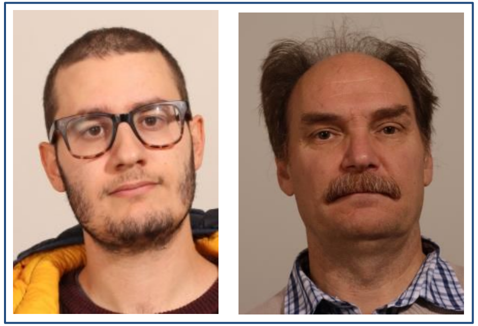
Contacts

- alice-dpg-aot-event-props@cern.ch → subscribe!
- alice-dpg-aot-track-props@cern.ch → subscribe!
- jouri.belikov@cern.ch, mfaggin@cern.ch, stefano.trogolo@cern.ch, evgeny.kryshen@cern.ch

Evgeny Stefano

AOT/Events

Mattia Iouri



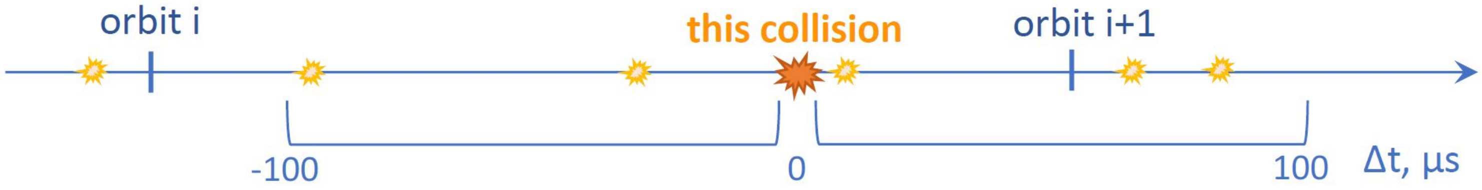
AOT/Tracks

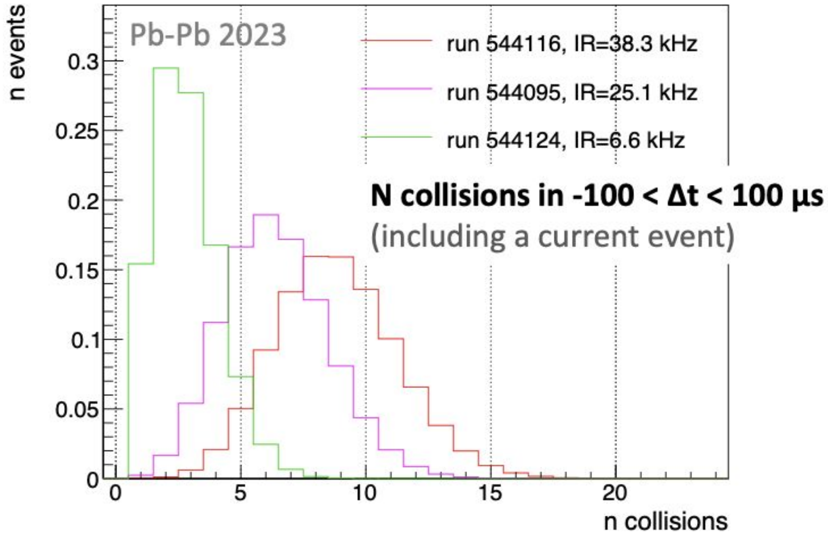
Thank you for your attention… and enjoy the tutorial! 😄

Backup

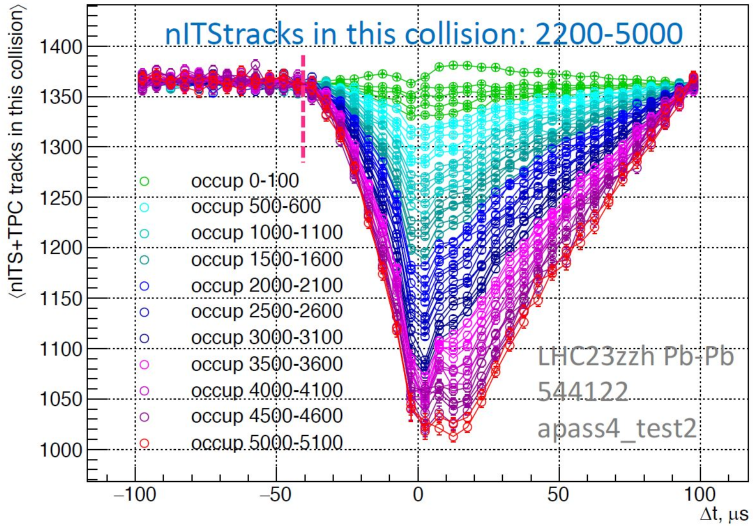
Example of advanced usage: occupancy effects

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch





Distributions of the number of collisions accumulated in a time interval for different IRs



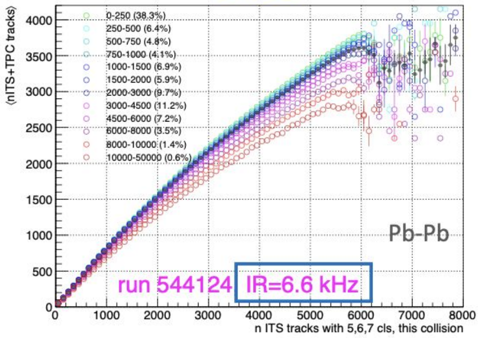
Using number of ITS tracks as a proxy for occupancy

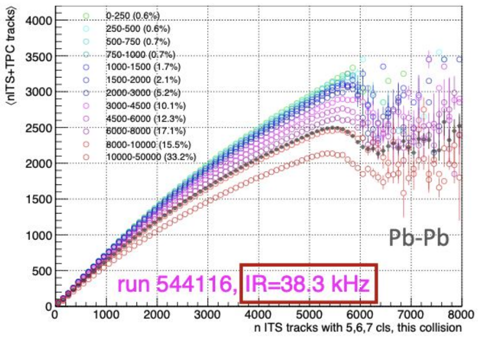
Example of advanced usage: occupancy effects

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

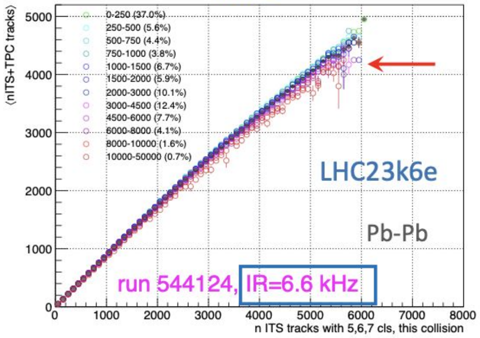
ALICE

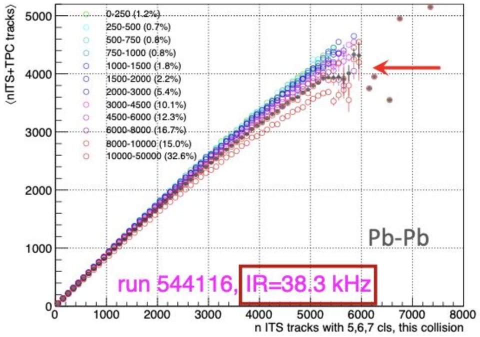
Data





MC

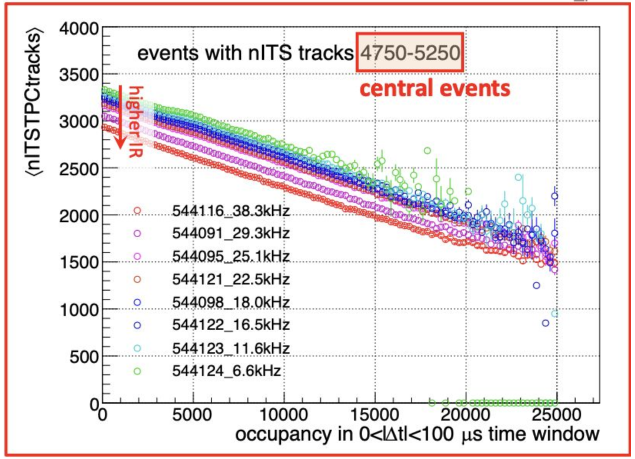




TPC-ITS matching degrades with occupancy

Currently, these occupancy effects are not well reproduced by MC

Pb-Pb LHC23zzh_pass3



- Selection bits available to test sensitivity of your analyses to occupancy effects, e.g.:

  col.selection_bit(kNoCollInTimeRangeStandard)

- More details in Igor's talk

Trigger aliases for triggered detectors in Run 3

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch

ALICE

Trigger alias: association of trigger class names (defined by CTP) to bits in the alias bit array (note: strings are not supported in the O2 data model)

Aliases for pp in Run 3:

```
mAliases[kEMC7] = "CTVXEMC-B-NOPF-EMC";
mAliases[kDMC7] = "CTVXDMC-B-NOPF-EMC";
mAliases[kTVXinTRD] = "CMTVX-B-NOPF-TRD,minbias_TVX";
mAliases[kTVXinEMC] = "C0TVX-B-NOPF-EMC,minbias_TVX_L0,CMTVXTSC-B-NOPF-EMC,CMTVXTCE-B-NOPF-EMC";
mAliases[kTVXinPHOS] = "C0TVX-B-NOPF-PHSCPV,minbias_TVX_L0,CMTVXTSC-B-NOPF-PHSCPV,CMTVXTSC-B-NOPF-PHSCPV";
mAliases[kTVXinHMP] = "C0TVX-B-NOPF-HMP,minbias_TVX_L0,CMTVXTSC-B-NOPF-HMP";
mAliases[kPHOS] = "CTVXPH0-B-NOPF-PHSCPV,mb_PH0_TVX,CPH0SC-B-NOPF-PHSCPV,CPH0CE-B-NOPF-PHSCPV";
```

Aliases for Pb-Pb in Run 3:

```
mAliases[kTVXinTRD] = "CMTVXTSC-B-NOPF-TRD,CMTVXTCE-B-NOPF-TRD";
mAliases[kTVXinEMC] = "CMTVXTSC-B-NOPF-EMC,CMTVXTCE-B-NOPF-EMC,C0TVXTSC-B-NOPF-EMC,C0TVXTCE-B-NOPF-EMC";
mAliases[kTVXinPHOS] = "CMTVXTSC-B-NOPF-PHSCPV,CMTVXTCE-B-NOPF-PHSCPV,C0TVXTSC-B-NOPF-PHSCPV,C0TVXTCE-B-NOPF-PHSCPV";
mAliases[kTVXinHMP] = "CMTVXTSC-B-NOPF-HMP,CMTVXTCE-B-NOPF-HMP";
mAliases[kPHOS] = "CPH0SC-B-NOPF-PHSCPV,CPH0CE-B-NOPF-PHSCPV";
```

https://github.com/AliceO2Group/O2Physics/blob/master/Common/CCDB/macros/upload_trigger_aliases_run3.C

Analysed luminosity in pp

- Integrated luminosity of the analysed sample can be calculated using TVX trigger counts in AO2Ds and corresponding visible cross section
- run-by-run TVX trigger counts can be extracted from FT0 entries: ft0.triggerMask() & BIT(o2::ft0::Triggers::bitVertex)
- Run-by-run TVX trigger counts are available in the output of o2-analysis-event-selection
  - Stored in the bc-selection-task/hCounterTVX histogram
- Luminosity is monitored run-by-run:
  - Stored in the bc-selection-task/hLumiTVX histogram:

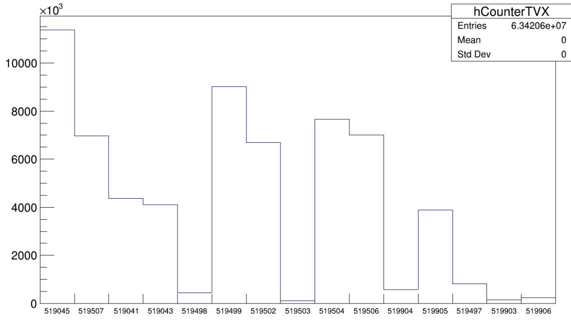
```
histos.get<TH1>(HIST("hLumiTVX"))->Fill(srun, 1. / csTVX);
```

  - Using visible cross sections estimated in MC: σ(TVX, 13 TeV) = 59.4 mb and σ(TVX, 0.9 TeV) = 35.5 mb
  - Next: extract cross sections from vdM scans, account for ageing
- TODO: apply pileup correction (typical correction ~ 2%)

L = N(TVX)/σ(TVX)*P(μ)

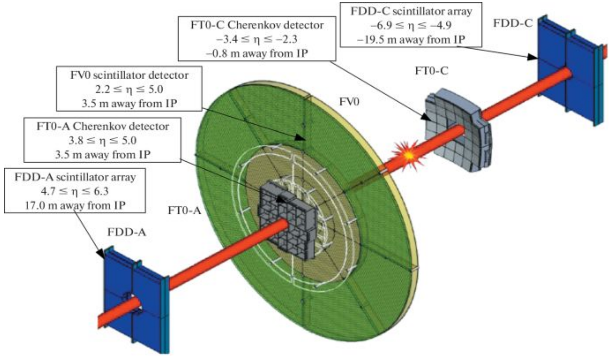
μ = average number of TVX-like collisions per bunch crossing

P(μ) = μ/(1-exp(-μ))



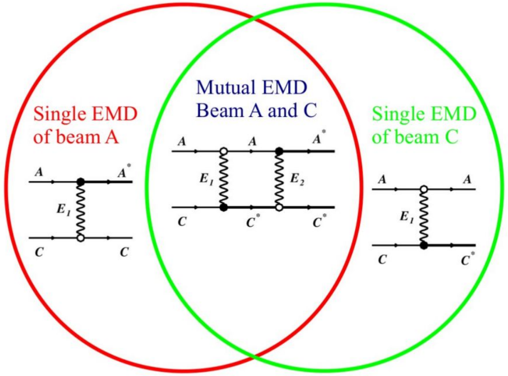
Luminosity estimators in Pb-Pb

- TVX counters are not good for lumi in Pb-Pb:
  - Fired in all hadronic Pb-Pb events with ~100% efficiency
  - BUT also sensitive to EM e+e- pair production pileup:
    - coincidences of single-sideA and single-sideC events
  - Non-linear scaling with luminosity



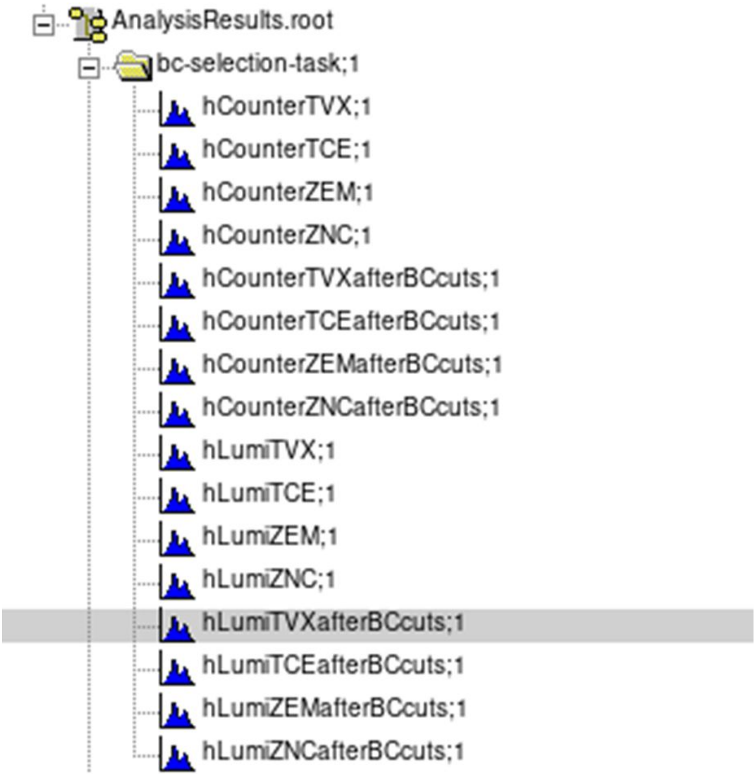
- Possible estimators from FT0:
  - TCE (FT0 central - mainly hadronic) → 10.36 b
  - TVX + tunable FT0 multiplicity threshold (to be checked)

- Possible estimators from ZDC:
  - Based on estimates from I. Pshenichnov (RELDIS generator) https://indico.cern.ch/event/1215416/#5-total-hadronic-and-emd-cross
  - ZNA (A-side neutron ZDC, mainly EMD) → 214.5 b
  - ZNC (C-side neutron ZDC, mainly EMD) → 214.5 b
  - ZAC (A & C-side neutron ZDC, hadronic+mututal EMD) → 14.1 b
  - ZEM (A-side or C-side neutron ZDC, mainly EMD) → 415.2 b



Luminosity monitoring tools

jouri.belikov@cern.ch
mfaggin@cern.ch
stefano.trogolo@cern.ch
evgeny.kryshen@cern.ch



- Luminosity is filled with negative values if estimators are not reliable:
  - TVX not good in Pb-Pb
  - TCE/ZEM/ZNC not available in pp
- Potentially possible to compare different lumi estimators and assign systematic uncertainties on lumi
- kNoITSROFrameBorder and kNoTimeFrameBorder cuts effectively reject a fraction of analysed luminosity which is monitored with forward detectors
  - Luminosity plots “after BC cuts” added in the output of bc-selection-task

Useful links

- Review on Event selection in Run2:
  https://indico.cern.ch/event/760954/contributions/3172100/
- Documentation on Event Selection in O2:
  https://aliceo2group.github.io/analysis-framework/docs/basics-usage/HelperTasks.html#event-selection
- Event selection tables:
  https://github.com/AliceO2Group/O2Physics/blob/master/Common/DataModel/EventSelection.h
- Trigger aliases:
  https://github.com/AliceO2Group/O2Physics/blob/master/Common/CCDB/TriggerAliases.h
  https://github.com/AliceO2Group/O2Physics/blob/master/Common/CCDB/macros/upload_trigger_aliases.C
- Event selection criteria:
  https://github.com/AliceO2Group/O2Physics/blob/master/Common/CCDB/EventSelectionParams.h#L21
- Event selection parameters:
  https://github.com/AliceO2Group/O2Physics/blob/master/Common/CCDB/EventSelectionParams.cxx
  https://github.com/AliceO2Group/O2Physics/blob/master/Common/CCDB/macros/upload_event_selection_params.C
- Luminosity normalization tools:
  https://indico.cern.ch/event/1305271/#10-tools-for-luminosity-monito
- Event selection QA repository:
  https://evsel-qa.web.cern.ch/