PWG-UD: Plan for hands-on tutorials

09:00 - 09:30

- intro into MC data in O2
  - usage of MC data in data analysis
  - MC data structures in the O2 data model
  - relating MC truth and reconstructed information

- hands-on tutorial in two steps

09:30 - 10:30

10:30: Wrap up

1. step
   - ⋆ compare generated and reconstructed variable distributions
   - ⋆ compute efficiencies

11:00 - 12:00

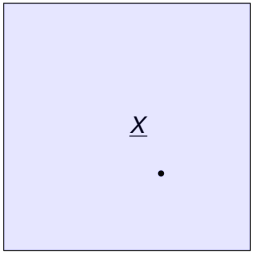
12:00: Wrap up

2. step
   - ⋆ relate MC truth and reconstructed values
   - ⋆ compare generated and reconstructed variable values

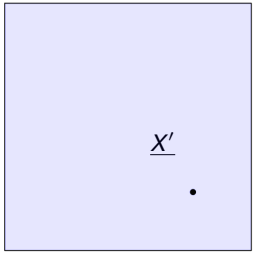
12:30: The end

Measurements with ALICE

Real world

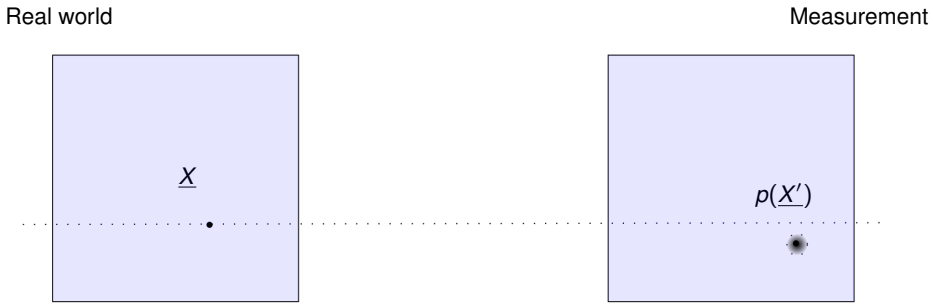


Measurement

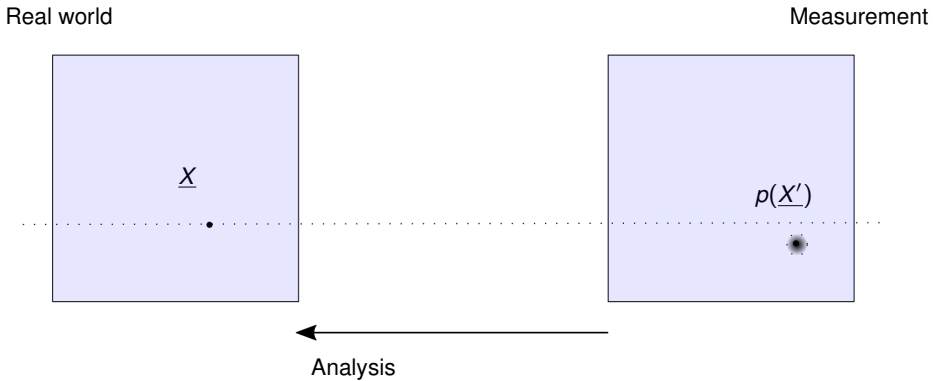


Measurements with ALICE

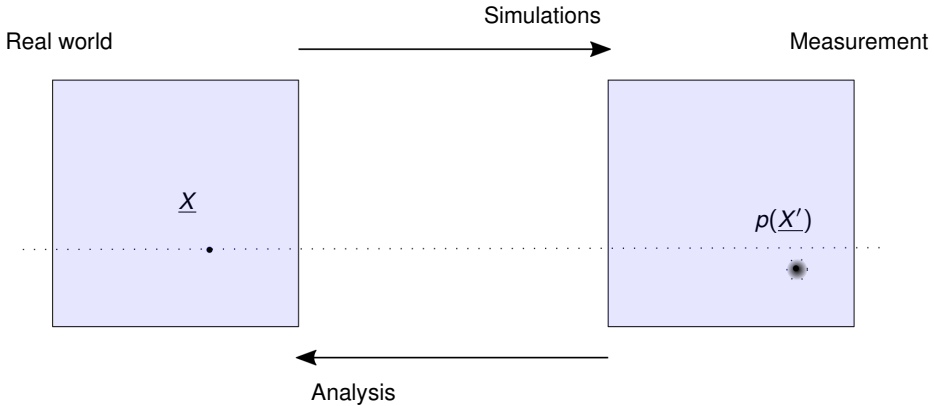
Measurements with ALICE



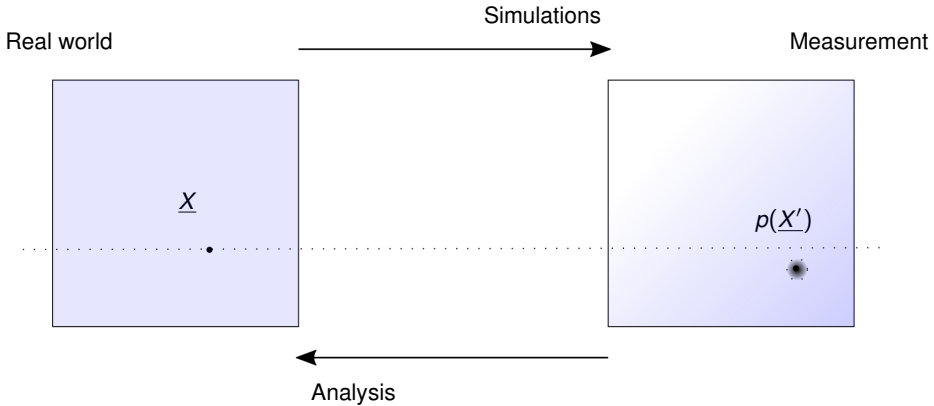
Measurements with ALICE



Measurements with ALICE

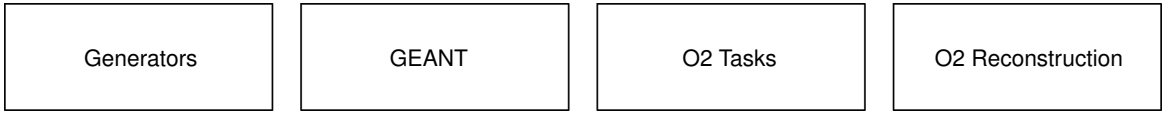


Measurements with ALICE



MC data in ALICE/O2

- Monte Carlo (MC) simulations in ALICE include
  - Generation of primary events/particles
  - Transport of particles through detector material, including energy losses
  - Simulation of detector signals
  - Event reconstruction



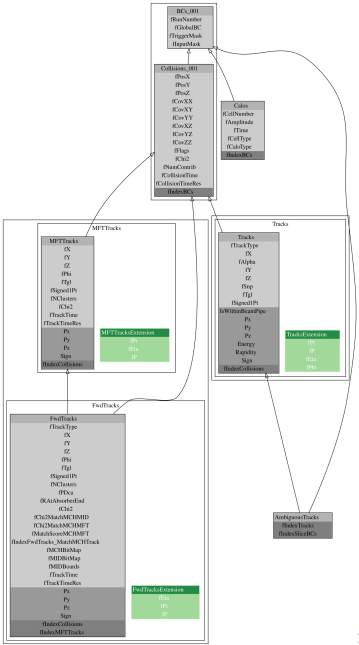
user	detector experts	detector experts	detector experts

- STARlight
- ...

MC data in ALICE/O2

- AO2Ds are the principle containers of ALICE data in O2
- includes "reconstructed" data tables

  (here only a small selection of the available tables is displayed)

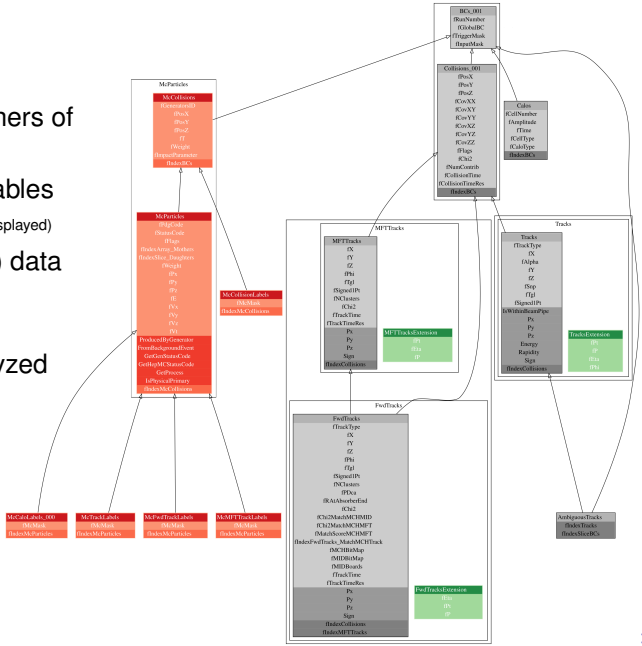


MC data in ALICE/O2

- AO2Ds are the principle containers of ALICE data in O2
- includes "reconstructed" data tables

  (here only a small selection of the available tables is displayed)
- includes "generated" (MC truth) data tables

  (see also here)
- both sets of tables can be analyzed independently

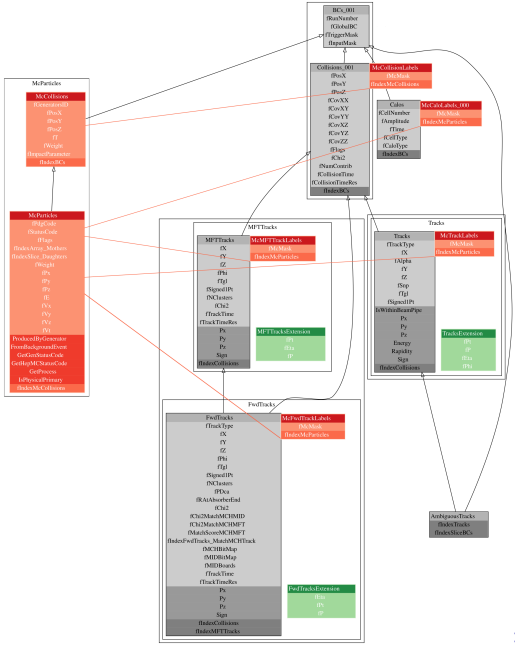


MC data in ALICE/O2

- AO2Ds are the principle containers of ALICE data in O2
- includes "reconstructed" data tables

  (here only a small selection of the available tables is displayed)
- includes "generated" (MC truth) data tables

  (see also here)
- both sets of tables can be analyzed independently
- the tables are linked by "index tables"



MC data for tutorial

- for this tutorial 2 sets of MC data were produced

1. rootfilesSignal
   - Generator: STARlight
   - Events: coherent $J/\psi \rightarrow \mu^+\mu^-$
   - Phase space limits: $|\eta_\mu| < 1.3$
   - Number of events: approx. 100 k
   - Output size: 120 MB

2. rootfilesSignalSkimmed
   - This is dataset 1. but skimmed with the DGCandProducer task $\rightarrow$ UDTables
   - Output size: 20 MB

- can be downloaded on laptop and analyzed there

Hands-on

1. Evaluate the reconstruction efficiency using dataset 1.?
   See previous slide

- use O2Physics/Tutorials/PWGUD/UDTutorials_03a.cxx as starting point
- Study the structure of the task
- Access MC truth and reconstructed values
- Compare variable distributions of MC truth and reconstructed values
- Modify UDTutorials_03a.cxx to extract the relevant information
- Try to improve - beautify - the task

UDTutorial_03a.cxx

```
struct UDTutorial03a{
  .
  .
  // define Preslices
  .
  // a few useful functions
  .
  .
  .

  void processMCTruth(aod::McCollisions const& mccollisions, CCs const& collisions,
                      aod::McParticles const& McParts, TCs const& tracks)
  {
    .
    // loop over all genererated collisions
    for (auto mccollision : mccollisions) {
      .

      // get McParticles which belong to mccollision
      auto partSlice = McParts.sliceBy(partPerMcCollision, mccollision.globalIndex());
      .

      // search for 2 muons
      auto daughterPartIds = getDaughterParts_gen(partSlice);
      .

      // compute the invariant mass
      computeIVM_gen(partSlice, daughterPartIds, lv1_gen, lv2_gen, lv_gen);
      .

      // reject if not in defined acceptance
      isInAcceptance(lv1_gen, lv2_gen, lv_gen);
  }
}
```

UDTutorial_03a.cxx

```
struct UDTutorial03a{
  .
  .
  .

  void processReco(CC const& collision, TCs const& tracks, aod::McCollisions const& mccollisions,
                   aod::McParticles const& McParts)
  {
    .

    auto daughterTrackIds = getDaughterTracks_rec(collision, tracks);
    .

    // reject if not in defined selection
    if (isSelected_rec(tracks, daughterTrackIds)) {

      // compute the invariant mass
      computeIVM_rec(tracks, daughterTrackIds, lv1_rec, lv2_rec, lv_rec);
      .

    }
  }
}
```

To get started ...

- you need a compiled and working O2/O2Physics installation
- download supporting material/instructions from https://cernbox.cern.ch/s/zOOEWNGUbK8Acsk → UDTutorials_1123.tar.gz
- prepare a working directory
  - **cp UDTutorials_1123.tar.gz "myPreferedDirectory"/.**
  - **cd "myPreferedDirectory"**
  - **tar -xvzf UDTutorials_1123.tar.gz**
  - **cd UDTutorials231108**
- you will find
  - **ls -1a**

    .getData
    plotHistograms.C
    README
    UDTutorialsConfig.json
- follow the instructions in **README**

PWG-UD: Plan for hands-on tutorials

09:00 - 09:30

- intro into MC data in O2
  - usage of MC data in data analysis
  - MC data structures in the O2 data model
  - relating MC truth and reconstructed information

- hands-on tutorial in two steps

09:30 - 10:30

10:30: Wrap up

1. step
   - compare generated and reconstructed events
   - compute efficiencies

11:00 - 12:00

12:00: Wrap up

2. step
   - relate MC truth and reconstructed values
   - assess reconstruction uncertainties

12:30: The end

Hands-on

2. Study the difference between MC truth and reconstructed values

- use O2Physics/Tutorials/PWGUD/UDTutorials_03b.cxx as starting point
- Study the structure of the task
- Access associated MC truth and reconstructed values
- Compute difference between MC truth and reconstructed values
- Correlate MC truth and reconstructed values

UDTutorial_03b.cxx

```
struct UDTutorial03b {
  .
  .
  .

  void processMCTruth(aod::McCollisions const& mccollisions, CCs const& collisions,
                      aod::McParticles const& McParts, TCs const& tracks)
  {
    .

    // loop over all genererated collisions
    for (auto mccollision : mccollisions) {
      .

      // get McParticles which belong to mccollision
      auto partSlice = McParts.sliceBy(partPerMcCollision, mccollision.globalIndex());

      for (auto McPart : partSlice) {
        // get track which corresponds to McPart
        auto trackSlice = tracks.sliceBy(trackPerMcParticle, McPart.globalIndex());

        if (trackSlice.size() > 0) {
          for (auto track : trackSlice) {
            auto pTrack = track.p();
            auto pPart = McPart.p();
            auto pDiff = pTrack - pPart;
            .

          }
        }
      }
    }
  }
}
```

UDTutorial_03b.cxx

```
struct UDTutorial03b {
  .
  .
  .

  void processReco(CC const& collision, TCs const& tracks, aod::McCollisions const& mccollisions,
                   aod::McParticles const& McParts)
  {
    .

    // get McCollision belonging to collision
    if (collision.has_mcCollision()) {
      auto mccollision = collision.mcCollision();
      .

    }
    .

    // compute the difference between generated and reconstructed momentum
    for (auto track : tracks) {
      // is there an associated McParticle?
      if (track.has_mcParticle()) {
        auto pTrack = track.p();
        auto McPart = track.mcParticle();
        auto pPart = McPart.p();
        auto pDiff = pTrack - pPart;
        .

      }
    }
  }
}
```

PWG-UD: Plan for hands-on tutorials

09:00 - 09:30

- intro into MC data in O2
  - usage of MC data in data analysis
  - MC data structures in the O2 data model
  - relating MC truth and reconstructed information

- hands-on tutorial in two steps

09:30 - 10:30

10:30: Wrap up

1. step
   - compare generated and reconstructed events
   - compute efficiencies

11:00 - 12:00

12:00: Wrap up

2. step
   - relate MC truth and reconstructed values
   - assess reconstruction uncertainties

12:30: The end