

O2 DQ hands-on session: Edition 3

Ionut Arsene, Fiorella Fionda, Laure Massacrier, Michael
Winn, Luca Micheletti

28th of April 2023

Instructions

Step-by-step

I) environment

Make support easier:

- standardize the version of O2Physics & simulation & scripts

Enlarge your space on Ixplus

With cern logged-in in your browser open the link:

<https://account.cern.ch/account/>

- click on web-page (under the title 'Authorization, Resources and CERN applications') on 'Resources and Services', click on 'List services', click on 'AFS Workspaces', click on Settings tab on the left hand side

- > Increase workspace to 5 GB: sufficient space

Installation of O2

Follow the instructions given during the preparatory day of the 17th of April, prior to the O2/DQ tutorial:

<https://indico.cern.ch/event/1267433/contributions/5359315/attachments/2629584/4547945/Git,%20GitHub%20and%20aliBuild%20intro.pdf>

If your local O2 installation failed, it is also possible to follow this tutorial using a recent O2/Physics version installed on lxplus (see instructions below)

Lxplus working environment and folder structure

Connect via ssh to lxplus:

use lxplus: `ssh -X name@lxplus.cern.ch`

Name and password: your cern account

Make sure that you have a valid grid certificate: instructions to be found here:

<https://alice-doc.github.io/alice-analysis-tutorial/start/cert.html>

To make paths the same:

`cd /afs/cern.ch/work/n/name/`

n: first letter of account name, name your account name

`mkdir O2ana_april2023`

`cd O2ana_april2023`

`mkdir configs`

Retrieve scripts for DQ workflows

In order to steer the work flows, there are python scripts provided

O2-DQ scripts: <https://github.com/iarsene/O2DQworkflows>

please clone them inside your O2ana_april2023 folder:

git clone <https://github.com/iarsene/O2DQworkflows>.git

*copy into your personal /**configs** directory, modify config files contained there*

```
cp O2DQworkflows/*.json configs/
```

You find a copy of the scripts also in the O2Physics/Tutorial/PWGDQ repository

Example files for work and command lines

Create inside O2ana_april2023 a folder structure

```
mkdir barrel_files
```

```
mkdir muon_files
```

```
mkdir CentralBarrel_2023Apr11
```

```
mkdir filterPP_file
```

```
mkdir interface_aods
```

You find AO2D files and reduced tables for testing in:

<https://cernbox.cern.ch/s/OSoYqMXWwaHGeZA>

pw: DQframework

This folder also contains example commands.

On lxplus, you can copy the file (assuming that you are in O2ana_april2023):

```
cp -r /afs/cern.ch/work/l/lmassacr/public/O2ana_april2023/barrel_files/* .
```

(copy then also muon_files, CentralBarrel_2023Apr11, filterPP_file and interface_aods repositories)

Or if you are on your local machine:

```
scp -r username@lxplus.cern.ch:/afs/cern.ch/work/l/lmassacr/public/O2ana_april2023/barrel_files/* .
```

I) running Table Maker

Workflow for data skimming
no MC truth information used (even if we run over MC data)

Scripts and configurations

Steering macro in O2DQworkflow directly

`runTableMaker.py`

Configuration file for workflow in O2DQworkflows

`configTableMakerDataRun3.json`

Options related to:

- Input data file
- ccdb input
- Information on additional tasks required
 - Input data type (Run2 converted/Run3)
 - Event and track selections for reduced data
 - variables to be written in output

Table-maker config adaptation: input file

Open config file for Table Maker with your preferred text-editor for running on real data:

these config files should be in the config folder:

e.g. in: configs/configTableMakerDataRun3.json

Adapt to your config file line 10, e.g.:

For prompt jpsi forward in muons (MC)

```
"aod_file": "muon_filesVprompt_MCVAO2D_MCLHC21id_04_2023.root"
```

For prompt jpsi / non-prompt jpsi / B⁺ at midrapidity (MC)

```
"aod_file": "barrel_filesVLHC21i3b2_PromptVAO2D.root"
```

```
"aod_file": "barrel_filesVLHC21i3f3_NonPromptVAO2D.root"
```

```
"aod_file": "barrel_filesVLHC21i3i2_BplusToJpsiKaon/AO2D.root"
```

For data (muon)

```
"aod_file": "muon_filesVFW_Real_April23VAO2D.root"
```

Table-maker config adaptation: adapt process function

Adapt to your config file line 259-305, i.e.:

For jpsi forward in muons several options possible, e.g.:

Modify from `"processMuonOnlyWithCov": "false",`

To `"processMuonOnlyWithCov": "true",`

For electrons, you could put:

`"processBarrelOnly": "false",`

To `"processBarrelOnly": "true",`

You can also process all inputs by:

`"processFull": "false",`

To `"processFull": "true",`

Alternatively, you put all processes to fall and control the option via the command line, see next slides, for example

`--arg table-maker:processMuonOnlyWithCov:true`

NOTE: - All tasks need to have at least one activated process function. Thus, to switch off one task from the workflow, one has to enable the "processDummy" and disable all the other process functions.

Table-maker config adaptation: cut configuration

Line 1 and following lines of configTableMakerDataRun3.json:

- Configure track cuts applied to data set

Please play around yourself!

The alias meanings can be found in:

<https://github.com/AliceO2Group/O2Physics/blob/master/PWGDQ/Core/CutsLibrary.cxx>

E.g : Useful cuts for muons :

“mchTrack” : MCH standalone tracks,

“matchedGlobal” : matched MFT-MCH-MID tracks, *“matchedFwd”* : matched MFT-MCH tracks,

“muonQualityCuts” : apply quality cuts (Rabs, pdca) for all track types

Table-maker command

Enter **alienv O2** inside your **O2ana_april2023** folder

```
source /cvmfs/alice-nightlies.cern.ch/bin/alienv enter VO_ALICE@VO_ALICE@O2Physics::nightly-20230419-1
```

Enter into the alien environment:

```
alien-token-init
```

Execute:

```
python3 O2DQworkflows/runTableMaker.py configs/configTableMakerDataRun3.json -runData --arg table-maker:processMuonOnlyWithCov:true --add_track_prop
```

```
[--arg table-maker:processBarrelOnlyWithCov:true --add_track_prop for electrons]
```

runTableMaker.py: python script to steer process, avoid to retype complicated command line

configTableMakerDataRun3.json: script to pass different options

Analogue with other provided input files, see files and commands in files provided in

<https://cernbox.cern.ch/s/OSoYqMXWwaHGeZA>

pw: DQframework

Need to adapt everytime your json file for specific input file and for data type

Output: 2 files

AnalysisResults.root

- Histogram results for quality control purposes and fast checks
 - At the moment variable choice fixed by track selection

reducedAod.root

- Reduced event and track level output
- At the moment variable choice fixed by track selection
 - Could be made configured to add specific variables

All variables are defined in VarManager within DQ framework

- Common code basis to avoid duplication bugs

All produced histograms are dealt with Histogram manager

Tables in reducedAod.root used as input for Table-Reader

II) running Table Maker MC

Workflow for MC-data skimming

MC truth information used and related to reconstructed objects

Table-maker-MC config adaptation: input file & process function

Analogue to Table Maker

Open config file for Table Maker with your preferred text-editor for running on real data:

these config files should be in the config folder, e.g. in: configs/configTableMakerMCRun3.json

Input file: Adapt to your config file line 10, e.g.

For prompt jpsi forward in muons

```
"aod_file": "aod_file": "muon_files\vprompt_MCVAO2D_MCLHC21id_04_2023.root"
```

For prompt jpsi / non-prompt jpsi / B+ at midrapidity

```
"aod_file": "barrel_files\VLHC21i3b2_PromptVAO2D.root"
```

```
"aod_file": "barrel_files\VLHC21i3f3_NonPromptVAO2D.root"
```

```
"aod_file": "barrel_files\VLHC21i3i2_BplusToJpsiKaon/AO2D.root"
```

Process function: Adapt to your config file line 185-212, analogue to Table Maker instructions i.e.

For jpsi forward in muons several options possible, e.g.

Modify from "processMuonOnlyWithCov": "false",

To "processMuonOnlyWithCov": "true",

Table-maker-MC config adaptation: cut configuration and MCsignals

Cut configuration analogue to Table Maker

MCsignal definition:

- Identifies specific MC truth signals in reconstructed data and produces specific output

See presentation by Ionut on logic and implementation:

https://indico.cern.ch/event/1098200/contributions/4620964/attachments/2359981/4028498/IArsene_DQO2analysis_2021Dec07.pdf

Table-maker config adaptation: MC signal configuration

Line 197 of configTableMakerMCRun3.json

- "cfgMCsignals": "electronPrimary,eFromJpsi,muFromJpsi,eFromLMeeLF,LMeeLF,Jpsi",

Please play around yourself with different signal definitions!

The alias meanings can be found in

<https://github.com/AliceO2Group/O2Physics/blob/master/PWGDQ/Core/MCSignalLibrary.cxx>

Table-maker-MC command

Enter alienv O2 inside your O2ana_april2023 folder

```
source /cvmfs/alice-nightlies.cern.ch/bin/alienv enter VO_ALICE@VO_ALICE@O2Physics::nightly-20230419-1
```

Execute:

```
python3 O2DQworkflows/runTableMaker.py configs/configTableMakerMCRun3.json -runMC --arg  
table-maker-m-c:processMuonOnlyWithCov:true --add_track_prop
```

```
[--arg table-maker-m-c:processMuonOnlyWithCov:true --add_track_prop for electrons]
```

runTableMaker.py: python script to steer process, avoid to retype complicated command line

configTableMakerMCRun3.json: script to pass different options, needs to be adapted depending on the input and the desired output

Input file treatment: either <-runData|-runMC>

See files and commands in files provided in

<https://cernbox.cern.ch/s/OSoYqMXWwaHGeZA>

pw: DQframework

Need to adapt everytime your json file for specific input file and for data type

Output

Analogue to Table-Maker with additional MC-truth and MC-matched folders

AnalysisResults.root

- Histogram results for QC purposes and fast checks
 - At the moment variable choice fixed by track selection
 - Could be made configured to add specific variables

reducedAod.root

- Reduced event and track level output
- At the moment variable choice fixed by track selection
 - Could be made configured to add specific variables

All variables are defined in VarManager within DQ framework

- Common code basis to avoid duplication bugs

All produced histograms are dealt with Histogram manager

Instructions

Step-by-step

IV) Running Table Reader

Workflow for real-data histogram and tree production from skimmed data

no MC truth information used

Table-reader prelude

We use our own reducedAod.root to run via the table maker
assume that you already produced reducedAod.root
(see instructions from Table-Maker)

Alternatively, you can use a larger file from data (running on hyperloop) the lego-train t
Muons:

“muon_files/FW_Real_April23/reducedAod.root”

Barrel:

“CentralBarrel_2023Apr11/inputfile/ReducedAOD_LHC22o_526712_data.root”

Table-reader config file modification: muons & event mixing

Open configAnalysisData.json with your preferred editor and modify the configuration to run (default should be fine for barrel dielectron analysis without event mixing).

Explanation: you need to modify the configAnalysisData.json file at two places: select either track-selection or muon-selection and select only one process for the same-event-pairing, changes in configAnalysisData.json between line 22 and 36, line 49 and 64

For muons, we need to make the following modifications:

Line 10: input file name: should be already correct, if you want to use the already produced files: "aod-file": "reducedAod.root",

Line 31: "processSkimmed": "false",

Line 33: "processDummy": "true"

Line 53: "processMuonSkimmed": "true",

Line 57: "processDummy": "false"

Line 74: "processDecayToEESkimmed": "false",

NOTE: - All tasks need to have at least one activated process function. Thus, to switch off one task from the workflow, one has to enable the "processDummy" and disable all the other process functions

- "analysis-track-selection": barrel tracks, "analysis-muon-selection": muon

Table-reader command

```
python3 O2DQworkflows/runAnalysis.py configs/configAnalysisData.json -runData --arg  
analysis-same-event-pairing:processDecayToMuMuSkimmed:true
```

```
[--arg analysis-same-event-pairing:pprocessDecayToEESkimmed:true for electrons]
```

If using the option "VertexingSkimmed" (either processDecayToMuMuVertexingSkimmed or processDecayToEEVertexingSkimmed) remember to set "cfgPropToPCA" option true

Dpl-workflow in read

configAnalysisData.json: contains all configurations

This will produce by default an output: AnalysisResults.root*

With all kind of pair-level information and other workflows of Table-reader (e.g. mixed events and dilepton + 1track etc. pp)

(*will overwrite the tablemaker output)

Table-reader output

AnalysisResults.root

- **Contains Histograms on event, track and pair level including:**
 - event-mixing
 - Like-sign
 - mumu, ee, e-mu

Table-reader: produce tree output

Execute with Python Script:

```
python3 O2DQworkflows/runAnalysis.py configs/configAnalysisData.json -runData --aod-writer-json  
configs/writerConfiguration_dileptons.json
```

writerConfiguration_dileptons.json: additional config file to produce pairs

This will produce a file dileptonAOD.root in addition

- This is the starting point to produce tuples/trees for further applications
 - Unbinned log-likelihood fits, machine learning applications with need for unbinned data.

Instructions

Step-by-step

IV) running DQ efficiency task

Workflow to produce MC-data histograms and tree production from skimmed with MC truth information used
Main purpose: efficiency and simulation background studies

DQ efficiency prelude and config modification

We use our own reducedAod.root to run via the table maker

assume that you already produced reducedAod.root

(see instructions from Table-Maker-**MC**)

You can find some reducedAod.root obtained from MC (running on hyperloop) for barrel analyses (i.e. including only central barrel tables) available in the cernbox folder: "CentralBarrel_2023Apr11/inputfile/Reduced_AO2D01_MC"

Need to adapt the the file configs/configAnalysisMC.json

Analogous change as for Table-reader config: configAnalysisData.json, see slide 27

Additional required configuration for efficiency functionality MCSignals:

"cfgTrackMCSignals": "electronPrimary", (line 22)

"cfgMuonMCSignals": "muFromJpsi", (line 31)

See alias meanings in <https://github.com/AliceO2Group/O2Physics/blob/master/PWGDQ/Core/MCSignalLibrary.cxx>

DQ efficiency task configuration

Dilepton (2-prong) candidate process :

```
"analysis-same-event-pairing": {
```

```
...
```

```
"cfgBarrelMCREcSignals": "mumuFromJpsi",
```

```
"cfgBarrelMCGenSignals": "Jpsi,dielectron",
```

```
"cfgFlatTables": "true",
```

```
"processJpsiToMuMuSkimmed": "false",
```

```
"processJpsiToMuMuVertexingSkimmed": "true",
```

```
...
```

} MCREc : reconstructed MC signals, MCGen :MC truth signals
concerns **all tracks**, not only Barrel (contrary to the name...)

—————→ Put this to *true* to produce a flat table, do be used in analysis, ML, ...
(only available for muons for now)

} Run dimuon pairing without or with secondary vertexing (will produce a
table *DileptonsExtra*)

Dilepton-track (3-prong) candidate process :

```
"analysis-dilepton-track": {
```

```
"cfgBarrelMCREcSignals": "mumumuFromBc",
```

```
"cfgBarrelMCGenSignals": "Jpsi",
```

```
"processDimuonMuonSkimmed": "false",
```

```
"processDummy": "true"
```

```
},
```

} Same as for previous process

DQ efficiency task command

Execute with Python script:

```
python3 O2DQworkflows/runAnalysis.py configs/configAnalysisMC.json -runMC  
--arg analysis-same-event-pairing:processDecayToMuMuSkimmed:true
```

```
[--arg analysis-same-event-pairing:pprocessDecayToEESkimmed:true for electrons]
```

If using the option "VertexingSkimmed" (either processDecayToMuMuVertexingSkimmed or processDecayToEEVertexingSkimmed) remember to set "cfgPropToPCA" option true

dpl-workflow

configAnalysisMC.json: contains all configurations

This will produce by default an output: AnalysisResults.root*

With pair-level information

(*will overwrite the tablemakerMC output)

DQ efficiency task command with tree output

Execute:

```
o2-analysis-dq-efficiency --configuration json://configs/configAnalysisMC.json --aod-writer-json  
configs/writerConfiguration_dileptonMC.json -b
```

Execute with Python script:

```
python3 O2DQWorkflows/runAnalysis.py configs/configAnalysisMC.json -runMC  
--arg analysis-same-event-pairing:processDecayToMuMuSkimmed:true [--arg  
analysis-same-event-pairing:processDecayToEESkimmed:true for electrons]  
--aod-writer-json configs/writerConfiguration_dileptonMC.json
```

dpl-workflow

configAnalysisMC.json: contains all configurations

writerConfiguration_dileptonMC.json: specifies to be written output

This will produce two outputs: AnalysisResults.root* and dileptonAOD.root

With all kind of pair-level information

(*will overwrite the tablemakerMC output if no renaming)

Instructions

Step-by-step

V) running Filter PP

Task for software trigger in pp data

Need to run with earlier O2Physics version, since scripts need to be adapted: `O2Physics/nightly-20221212-1`

filter-pp config adaptation: input file

Open config file for Filter pp with your preferred text-editor for running on real data:
these config files should be in the config folder:

e.g. in: configs/configFilterPPDataRun3.json

Change the name of the AO2D.root:

Line 10 : "aod-file": "filterPP_file\AO2D_filterppTest.root"

This AO2D.root can be found in the filterPP_file folder in
<https://cernbox.cern.ch/s/OSoYqMXWwaHGeZA>

filter pp config adaptation: cut configuration

Line 150-154 : Event Selection subtask :

```
"d-q-event-selection-task": {  
    "cfgEventCuts": "eventStandardNoINT7",    ->Defines the event selection  
    "cfgWithQA": "false",                    ->Plot control histograms (or not)  
    "processEventSelection": "true",  
    "processDummy": "false"  
},
```

The list of available cuts can be found in :

<https://github.com/AliceO2Group/O2Physics/blob/master/PWGDQ/Core/CutsLibrary.cxx>

filter pp config adaptation: cut configuration

Line 156-164 : barrel track selection subtask :

```
"d-q-barrel-track-selection": {
```

```
  "cfgBarrelTrackCuts":
```

```
  "jpsiO2MCdebugCuts2,jpsiO2MCdebugCuts2,jpsiO2MCdebugCuts3,jpsiO2MCdebugCuts3,jpsiO2MCdebugCuts,jpsiO2MCdebugCuts",
```

->Defines the cuts for barrel track selection

```
  "cfgWithQA": "false",
```

->Plot control histograms (or not)

```
  "processSelection": "true",
```

```
  "processDummy": "false"
```

```
}
```

The list of available cuts can be found in :

<https://github.com/AliceO2Group/O2Physics/blob/master/PWGDQ/Core/CutsLibrary.cxx>

filter pp config adaptation: cut configuration

Line 165-170 : forward track selection subtask :

```
"d-q-muons-selection": {
```

```
    "cfgMuonsCuts": "muonHighPt",
```

->Defines the cuts for forward track selection

```
    "cfgWithQA": "false",
```

->Plot control histograms (or not)

```
    "processSelection": "true",
```

```
    "processDummy": "false"
```

```
},
```

The list of available cuts can be found in :

<https://github.com/AliceO2Group/O2Physics/blob/master/PWGDQ/Core/CutsLibrary.h>

filter pp config adaptation: cut configuration

Line 171-176 : trigger subtask :

```
"d-q-filter-p-p-task": {  
  
    "cfgBarrelSels":  
    "jpsiO2MCdebugCuts2::1,jpsiO2MCdebugCuts2::2,jpsiO2MCdebugCuts3::1,jpsiO2MCdebugCuts3::2,jpsiO2MCdebugCuts::1,jpsiO2MCdebugCuts::2",  
  
    "cfgMuonSels": "muonHighPt::1",  
  
    "cfgWithQA": "false",  
  
    "processFilterPP": "true",  
  
    "processDummy": "false"  
  
},
```

The format for the configuration is "SingleTrackCut:PairCut:1" for a dilepton trigger
And for a single lepton trigger "SingleTrackCut::1"

This allows to not look for leptons pairs with cuts that are not used in the dilepton triggers.

The triggers are : Single Electron, DiElectron, Single Muon Low pT, SingleMuon High pT, DiMuon (in that order)

Filter pp task command

Needs to jump back to earlier version, scripts need to be adapted to run with current version O2Physics/nightly-20221212-1

Execute:

```
python3 O2DQworkflows/runFilterPP.py configs/configFilterPPDataRun3.json
```

A few things to be careful of :

- The filter **always** run over both the barrel and forward tracks. If you only require one, don't deactivate the task but leave the corresponding selection empty.
For instance : **"cfgMuonsCuts": ""** and **"cfgMuonSels": ""** in the corresponding subtasks
- The number of selections in the track selections subtasks must match the one in the filter subtask. For instance, if your triggers are **"muonLowPt::1,muonLowPt:pairNoCut:1"** (which corresponds to a single muon trigger and a dimuon trigger), then the selections in the *d-q-muons-selection* subtask must be **"muonLowPt,muonLowPt"** to match

filterPP output

Monitoring histograms in: AnalysisResults.root

- Histogram showing the counts for each trigger
- Histograms on event and track level before and after applied cuts (if the QA option is activated)
- A list of 5 booleans that indicates if the event passes the defined trigger conditions