O2 DQ hands-on session: Edition 4

Ionut Arsene, Fiorella Fionda, Cristiane Jahnke, Laure Massacrier, Michael Winn, Luca Micheletti, Pengzhong Lu, Xiaozhi Bai

8th of November 2023

Support during tutorial

Please use the mattermost channel:

O2AT-PWG-DQ

The convenors and several experts will try to help you as much as possible

Instructions
Step-by-step

I) environment

Make support easier:

- standardize the version of O2Physics & simulation & scripts

Enlarge your space on lxplus

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 at:

https://aliceo2group.github.io/analysis-framework/docs/gettingstarted/installing.html

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_nov2023

cd O2ana_nov2023

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_nov2023 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

You find AO2D files and reduced tables for testing in:

https://cernbox.cern.ch/s/OSoYgMXWwaHGeZA

pw: DQframework

This folder also contains example commands.

On lxplus, you can copy the file (assuming that you are in O2ana_nov023):

```
cp -r /afs/cern.ch/work/m/mwinn/public/DQinstructions/*  .
```

(or better only a subset - faster and smaller file size)

```
cp -r /afs/cern.ch/work/m/mwinn/public/DQinstructions/barrel_files_Nov2023/ .
cp -r /afs/cern.ch/work/m/mwinn/public/DQinstructions/muon_files_Nov2023/ .
cp /afs/cern.ch/work/m/mwinn/public/DQinstructions/*.txt .
```

Or if you are on your local machine (same files as in the cernbox above):

```
scp -r username@lxplus.cern.ch://afs/cern.ch/work/m/mwinn/public/DQinstructions/*files_Nov2023/ .
scp username@lxplus.cern.ch://afs/cern.ch/work/m/mwinn/public/DQinstructions/*.txt .
```

Instructions
Step-by-step

I) Running Table Reader

Workflow for real-data histogram and tree production from skimmed data
no MC truth information used

Table-reader prelude: input file

We use reducedAod.root that were produced by central hyperloop running

the instructions work also if you produce yourself the reduced data via the TableMaker as explained in the back-up.

We will go through it for

Muons:

“muon_files_Nov2023/pp/*.root” (pp, still with phi bug) (on lxplus need to use only a single file due to memory limits on lxplus machines)

¨muon_files_Nov2023/PbPb/*.root” (PbPb 2023) (file with 1 GeV pt cut on daughters, file without cut)

The folder MC prompt MC AO2Ds (not reduced AO2Ds!) after phi-bug fix, one larger and one smaller file for download. We will come to it later.

Barrel:

“ input/SkimmedTree_midy-pp13.6TeV-528531_AO2D.root” was produced from ppTriggerTask and TableMaker by the PWG-DQ

This is the pp skimmed tree, the J/psi signal was filtered during the skimming

The Kaons are also saved in this skimmed tree.

Table-reader prelude: O2-environment

Enter alienv O2 inside your O2ana_nov2023 folder

source /cvmfs/alice-nightlies.cern.ch/bin/alienv enter VO_ALICE@O2Physics::daily-20231027-0200-1

Or

bash source /cvmfs/alice-nightlies.cern.ch/bin/alienv enter VO_ALICE@O2Physics::daily-20231027-0200-1

Or if you are on your local machine: alienv enter O2Physics/latest-master-o2

Enter the alien environment (access ccdb etc.):

alien-token-init

If you want to check for other versions, you can find all available tags via:

source /cvmfs/alice-nightlies.cern.ch/bin/alienv q | grep O2Physics::daily

Table-reader prelude merging

If you open the reduced AO2D-files, you will realise that there is a subfolder structure that is inherited from the original AO2D-files.

For some simple operations with root outside of O2 or within O2, it might be useful to merge a list of files provided in a text file with the following command within o2:

o2-aod-merger input.txt

Table-reader examples MUONS

Table-reader config: muons

The two data subfolders with the muon files (pp, PbPb) contain a file ´commands.txt´ that contains a command line that runs the Table-Reader

o2-analysis-dq-table-reader -b --configuration json://configuration.json --aod-file @input_data.txt

(pp)

o2-analysis-dq-table-reader -b --configuration json://configuration_table_reader.json --aod-file AO2D_small_no_cuts.root

(PbPb)

In case that you work on lxplus, you will probably encounter memory issues if you want to run over all files provided in the input_data.txt-data. For pp, on the lxplus-copy folder of the cernbox files, there is input_data_1.txt

If you want to produce the dilepton-tree, you can add the following command:

Please copy the file writerConfiguration_dileptons.json into the corresponding folder and run

o2-analysis-dq-table-reader -b --configuration json://configuration.json --aod-file @input_data.txt --aod-writer-json writerConfiguration_dileptons.json

Table-reader config file modification via python scripts

Alternatively, a python script is provided in the configs-Folder to run the analysis.

This works schematically like this:

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

Dpl-workflow in read

configAnalysisData.json: contains all configurations, needs to be adapted, input file name change, e.g. (for pp) in line 10: @muon_files_Nov2023/pp/input_data_2.txt

Please be aware:

If using the option "VertexingSkimmed" (either processDecayToMuMuVertexingSkimmed or processDecayToEEVertexingSkimmed) two things should be considered:

1) **The covariance matrix table of tracks (O2rtbarrelcov/O2rtmuoncov) should be filled in the reducedAod.root, i.e. the table-maker that produces this file needs to be run with the option "process*WithCov"=true**
2) **"cfgPropToPCA" option should be true in the configAnalysisData.json**

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.

Table-reader examples ELECTRONS

Table-reader config file modification

Go to barrel_files_Nov2023/data/pp

Open configs/SkimmedTreePPReader.json

1. How to modify the input of TableReader
   line 10 "aod-file": "@input_DataSkimmed_pp.txt"

2. How to modify the cuts of event and track selection in TableReader
   line 14 "cfgEventCuts": "eventStandardNoINT7"
   line 29 "cfgTrackCuts": "Jpsi_TPCPost_calib_debug4"

3. How to config the histograms of the event and track
   line 17 "cfgAddEventHistogram": "vtxpp"
   line 31 "cfgAddTrackHistogram": "tpcpid,dca,its"
4. Select the process
   Line 76 "processDecayToEESkimmed": "true"

Table-reader command

1. The version of O2Physics used is VO_ALICE@O2Physics::daily-20231027-0200-1 (same as slide 11)
   /cvmfs/alice.cern.ch/bin/alienv enter VO_ALICE@O2Physics::daily-20231027-0200-1

2. You need CCDB access for the calibration file
   alien.py
   To provide your password for the alien.py connection
   exit

3. Access to the tutorial of the central barrel file fold for pp skimmed tree
   cd barrel_files_Nov2023/data/pp

4. How to run the skimmed tree
   First option:
   ./runSkimmedTree.sh
   (need before: chmod 777 runSkimmedTree.sh)
   Second option:
   python3 runAnalysis.py configs/SkimmedTreePPReader.json -runData --arg analysis-same-event-pairing:processDecayToEESkimmed:true

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

II) running Table-Maker-MC

Workflow to produce MC-data histograms and derived data with MC truth information used
Main purpose: efficiency and simulation background studies

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

Open config file for Table Maker MC 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_Nov2023\MC\AO2D_MC_1.root”

Process function: Adapt to your config file line 202-216, i.e.

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

Modify from "processMuonOnlyWithCov": "false",

To "processMuonOnlyWithCov": "true",

And you need to modify ´processOnlyBCs:true´ to ´processOnlyBCs:false´

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_nov2023 folder

source /cvmfs/alice-nightlies.cern.ch/bin/alienv enter VO_ALICE@O2Physics::daily-20231027-0200-1

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>

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

Output

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
  - variable choice fixed by track selection

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

III) 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 mc

assume that you already produced reducedAod.root

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

Need to adapt the file configs/configAnalysisMC.json, see next slide for details

Additional required configuration for efficiency functionality MCSignals:

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

"cfgMuonMCSignals": "muFromJpsi", (line 29)

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

DQ efficiency prelude and config modification

Changes in configuration:

"aod-file": "reducedAod_mc.root", -> "aod-file": "reducedAod.root", line 10

"cfgQA": "true", "processSkimmed": "true", "processDummy": "false"

-> "cfgQA": "false", "processSkimmed": "false", "processDummy": "true", lines 23 -25

"processSkimmed": "false", "processDummy": "true"

-> "processSkimmed": "true", "processDummy": "false" line 31,32

"cfgFlatTables": "false", -> "cfgFlatTables": "true", line 39

"cfgUseAbsDCA": "false", "cfgPropToPCA": "false" , "processDecayToEESkimmed": "true",

-> "cfgUseAbsDCA": "true", "cfgPropToPCA": "true", "processDecayToEESkimmed": "false", lines 45-47

"processDecayToMuMuVertexingSkimmed": "false" -> "processDecayToMuMuVertexingSkimmed": "true", line 50

DQ efficiency task configuration

Dilepton (2-prong) candidate process :

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

 …
    "cfgBarrelMCRecSignals": "mumuFromJpsi",
    "cfgBarrelMCGenSignals": "Jpsi,dielectron",
    "cfgFlatTables": "true",
 "cfgUseAbsDCA": "true",
    "cfgPropToPCA": "true",
    "processDecayToMuMuSkimmed": "false",
    "processDecayToMuMuVertexingSkimmed": "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, …

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

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)
two things should be considered:

 1)   The covariance matrix table of tracks (O2rtbarrelcov/O2rtmuoncov) should be filled in the reducedAod.root, i.e. the
      table-maker-m-c should be run with the option "process*WithCov"=true
 2)   "cfgPropToPCA" option should be true in the configAnalysisMC.json
```

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
```

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)

Back-up:
running Table Maker and

Workflow for data skimming
no MC truth information used (even if we run over MC data)
By default this part of the workflow should be done centrally by the DQ working group with the hyperloop system
The given local tests are provided to give an idea of how it works and can be useful for testing

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_files\/prompt_MC\/AO2D_MCLHC21id_04_2023.root”

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

“aod_file”: ”barrel_files\/LHC21i3b2_Prompt\/AO2D.root”

“aod_file”: ”barrel_files\/LHC21i3f3_NonPrompt\/AO2D.root”

“aod_file”: ”barrel_files\/LHC21i3i2_BplusToJpsiKaon/AO2D.root”

For data (muon)

“aod_file”: ”muon_files\/FW_Real_April23\/AO2D.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

Instructions
Step-by-step

VI) running V0 Selector

Task for selecting V0 particles

Select barrel tracks which belong to clean V0 and cascade topologies. This code loops over a V0Data table and produces some standard analysis output for select clean V0 sample.

V0 selector config adaptation: input file

Copy the table-maker config file to be as the v0-selector config file, because they have many common dependencies

cp configs/configTableMakerDataRun3.json configs/configV0DataRun3.json

Open config file for V0 selector with your preferred text-editor for running on real data:

these config files should be in the configs folder:

e.g. in: configs/configV0DataRun3.json

Change the name of the AO2D.root:

Line 10 :

“aod_file”: ”barrel_files\LHC21i3b2_Prompt\AO2D.root”

“aod_file”: ”barrel_files\LHC21i3f3_NonPrompt\AO2D.root”

“aod_file”: ”barrel_files\LHC21i3i2_BplusToJpsiKaon/AO2D.root”

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

V0 selector config adaptation: add V0 configuration

Add configurations to the config files

```
"lambdakzero-builder": {
    "createV0CovMats": "-1",
    "d_UseAutodetectMode": "false",
    "dcanegtopv": "0.100000001",
    "dcapostopv": "0.100000001",
    "v0cospa": "0.995",
    "dcav0dau": "1",
    "v0radius": "0.899999976",
    "tpcrefit": "0",
    "d_bz": "-999",
    "d_UseAbsDCA": "true",
    "d_UseWeightedPCA": "false",
    "useMatCorrType": "2",
    "rejDiffCollTracks": "0",
    "d_doTrackQA": "false",
    "ccdb-url": "http:\/\/alice-ccdb.cern.ch",
    "grpPath": "GLO\/GRP\/GRP",
    "grpmagPath": "GLO\/Config\/GRPMagField",
    "lutPath": "GLO\/Param\/MatLUT",
    "geoPath": "GLO\/Config\/GeometryAligned",
    "d_doQA": "false",
    "dQANBinsRadius": "500",
    "dQANBinsPtCoarse": "10",
    "dQANBinsMass": "400",
    "dQAMaxPt": "5",
    "dQAK0ShortMassWindow": "0.00499999989",
    "dQALambdaMassWindow": "0.00499999989",
    "processRun2": "false",
    "processRun3": "true"
},
```

Cuts for selecting V0s

->Plot track QA histograms (or not)

->Plot basic QA histograms (or not)

QA histograms setting

This task loops over a set of V0 indices and creates the corresponding analysis tables that contain the typical information required for analysis.

V0 selector config adaptation: add V0 configuration

Add configurations to the config files

```
"v0-selector": {
    "v0max_mee": "0.0599999987",
    "maxpsipair": "0.300000012",
    "v0cospa": "0.998000026",
    "dcav0dau": "0.300000012",
    "v0Rmin": "0",
    "v0Rmax": "90",
    "dcamin": "0",
    "dcamax": "1e+10",
    "mincrossedrows": "70",
    "maxchi2tpc": "4",
    "fillhisto": "false"
},
"track-pid-qa": {
    "dcamin": "0",
    "dcamax": "1e+10",
    "mincrossedrows": "70",
    "maxchi2tpc": "4",
    "processQA": "true",
    "processDummy": "false"
}
```

} Cuts for selecting V0s

->Plot V0 QA histograms (or not)

} Cuts for the barrel tracks from V0s

->Plot basic QA histograms (or not)

This workflow loops over a V0Data table (produced by the lambdakzero-builder) and produces some standard analysis output for select clean V0 sample.

V0 selector task command

Execute:

```
o2-analysis-dq-v0-selector -b --configuration json://configs/configV0DataRun3.json | o2-analysis-timestamp -b
--configuration json://configs/configV0DataRun3.json | o2-analysis-event-selection -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-pid-tof-full -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-pid-tof-beta -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-pid-tof-base -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-multiplicity-table -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-trackselection -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-lf-lambdakzerobuilder -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-pid-tpc-full -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-zdc-converter -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-track-propagation -b --configuration
json://configs/configV0DataRun3.json | o2-analysis-pid-tpc-base -b --configuration
json://configs/configV0DataRun3.json
```

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

With V0 and track-pid-qa information