PWGEM Light-flavour Cocktail Framework

Hands-on

O2 analysis tutorial
10/11/2023

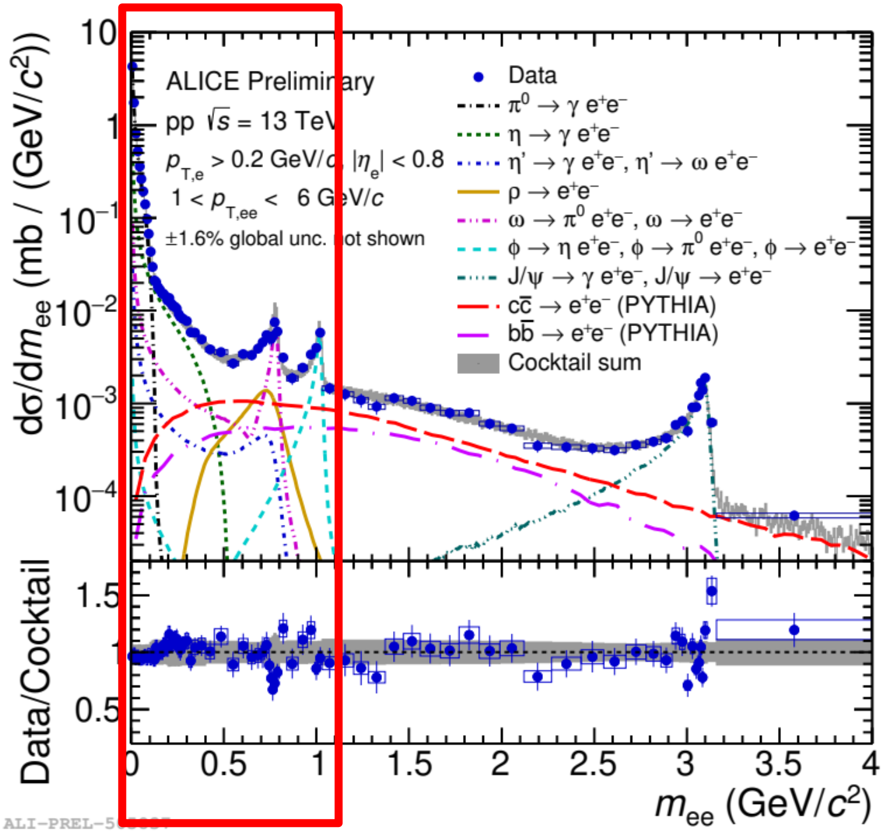
Daniel Samitz

Intro

- Hadronic Cocktail: estimate contributions from known hadronic sources two dielectron invariant mass spectrum
  - Here: light-flavour sources only $(\pi^0,\, \eta,\, \eta',\, \rho,\, \omega,\, \phi)$
- Simulation: GeneratorEMCocktail GeneratorEMCocktailV2.C
  - Generates mesons from parametrized pT-distributions
    - If not given parametrization for given type of meson, uses mT-scaling from Pi0 distribution

$$p_T \;\rightarrow\; \sqrt{p_T^2 + m^2 - m_{\pi^0}^2}$$

  - Let mesons decay into dielectron pairs
- Analysis: LF Cocktail Analysis Task lmeeLFCocktail.cxx
  - Momentum smearing for detector resolution effects
  - Applies reconstruction efficiency weights
  - Various acceptance cuts
  - Uses MC mother/daughter information to calculate spectra for each particle type



Where to find the material for this tutorial?

- O2Physics/Tutorials/PWGEM/Cocktail/
  - run_lmee_lf_cocktail.py
    - Main python script you will use today
  - configs/
    - sim_config.json
      - An example config file for the simulation
    - ana_config.json
      - An example config file for the analysis task
  - files/
    - parametrizations/
      - Two files with pT parametrizations for the simulation
    - resolution/
      - Some files for resolution, efficiency, DCA for the analysis task (are also on alien at alien:///alice/cern.ch/user/d/dsamitz/cocktail_tutorial/)
    - decaytables/
      - (optional) decay table for the simulation
  - plotLFCocktail.C
    - ROOT macro for plotting some histograms of the analysis output
- Cern Box via this link https://cernbox.cern.ch/s/aLFkmdKQzJrdWwl
  - (optional) simulation output (from 10 to 100000 events) that can be used as input for the analysis task

Load the environments

- on your computer:
  - If not done yet:
    - aliBuild init O2DPG
    - aliBuild build O2DPG
    - aliBuild build AEGIS
  - alienv enter O2Physics/latest O2DPG/latest AEGIS/latest

OR

- remotely on lxplus
  - scp -r O2Physics/Tutorials/PWGEM/Cocktail <your-cern-user-name>@lxplus.cern.ch:Cocktail
    - In case you do not have enough free space, try:
      scp -r O2Physics/Tutorials/PWGEM/Cocktail dsamitz@lxplus.cern.ch:/eos/home-d/dsamitz/Cocktail
      (with your username)
  - ssh -X <your-cern-user-name>@lxplus.cern.ch
  - alienv enter VO_ALICE@O2sim::v20231030-1

The python script

- The simulation and analysis task can be steered with the script run_lmee_lf_cocktail.py

- Three subcommands: ‘sim-only’, ‘ana-only’, ‘full’
  - Ideally we manage to try all three of them today

- For each subcommand you can use the ‘--help’ option to see which required and optional parameters it takes
  - e.g. ‘python run_lmee_lf_cocktail.py sim-only --help’

- If you are interested in what happens in the background, have a look at the output printed to screen
  - e.g. see which ‘o2-sim’ commands are called by the script

Now, do it yourself...

STEP 1: Simulation

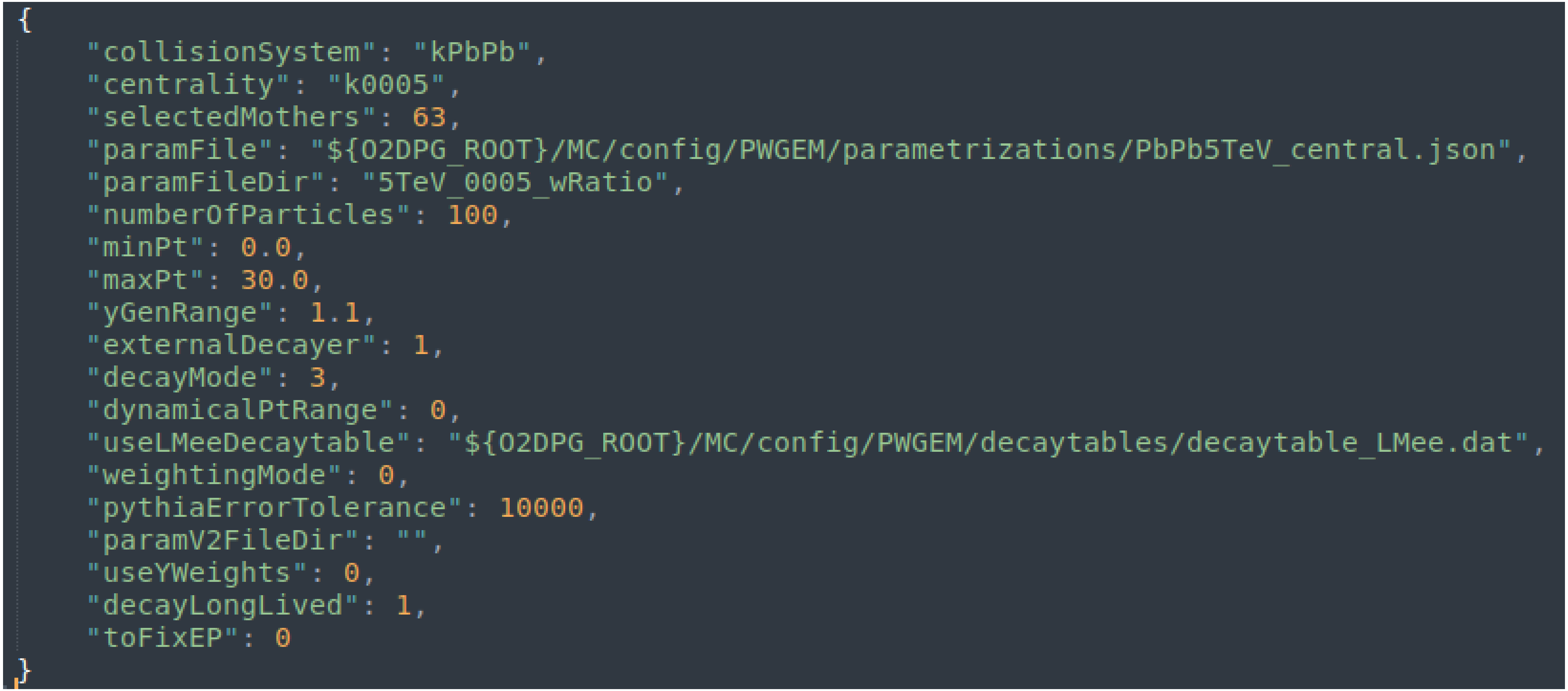
‘sim-only’ command

- Command to run the light flavour cocktail simulation
  - produces: o2sim_Kine.root with MCTracks
- With O2Physics O2DPG AEGIS (or if installed O2sim) loaded, run

```
python run_lmee_lf_cocktail.py sim-only -n <nEvents> --sim-config-file <sim_config_file> [-o <output_dir>]  [--clean]
```

  - <nEvents>: number of events to be generated
    - maybe start with ca. 10 for a first test
  - <sim_config_file>: the simulation config file that contains all relevant settings
    - see next slide for details
  - <output_dir> [optional]: output directory
    - default = current directory
  - --clean [optional]: delete files produced by the generator that are not relevant for us (recommended)
    - keeps only o2sim_Kine.root and log files

The sim-config-file (O2Physics/Tutorial/PWGEM/Cocktail/configs/sim_config.json)



```
{
    "collisionSystem": "kPbPb",
    "centrality": "k0005",
    "selectedMothers": 63,
    "paramFile": "${O2DPG_ROOT}/MC/config/PWGEM/parametrizations/PbPb5TeV_central.json",
    "paramFileDir": "5TeV_0005_wRatio",
    "numberOfParticles": 100,
    "minPt": 0.0,
    "maxPt": 30.0,
    "yGenRange": 1.1,
    "externalDecayer": 1,
    "decayMode": 3,
    "dynamicalPtRange": 0,
    "useLMeeDecaytable": "${O2DPG_ROOT}/MC/config/PWGEM/decaytables/decaytable_LMee.dat",
    "weightingMode": 0,
    "pythiaErrorTolerance": 10000,
    "paramV2FileDir": "",
    "useYWeights": 0,
    "decayLongLived": 1,
    "toFixEP": 0
}
```

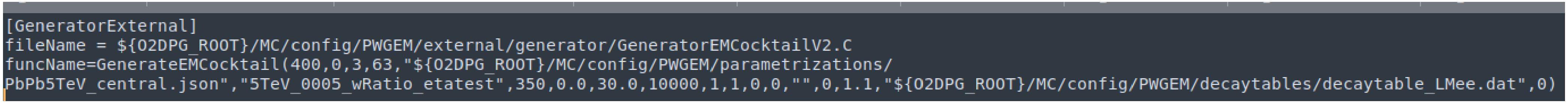
The sim-config-file

- collisionSystem {"kpp900GeV","kpp2760GeV","kpp7TeV","kpPb","kPbPb"}
- centrality {"kpp","k0005","k0510","k1020","k2030","k3040","k4050","k5060","k0010","k2040","k4060","k6080","k0020","k0040","k2080":0xE, "k4080","k2050"}
- selectedMothers switch on/off particles bitwise: order: {Pi0,Eta,Rho0,Omega,EtaPrime,Phi,…} (see AEGIS/GeneratorParam/GeneratorParamEMlibV2.h for full list)
  e.g. Pi0+Omega = 001001 = 9, e.g. all of the above = 111111 = 63
- paramFile file with pT parametrizations (from O2DPG or O2Physics/Tutorials/PWGEM/Cocktail/files/parametrizations/)
- paramFileDir ‘directory’ of parametrizations (look into the paramFile to see which are available)
- numberOfParticles: number of particles of each type per event
- minPt,maxPt minimum/maximum pT of generated particles
- yGenRange rapidity range of generated particles
- externalDecayer 0: default TPythia6Decayer, 1: EXODUS decayer
- decayMode: 0: all, 1: single photon, 2: single eletron: 3 dieelectron (leave this at 3 or you will see not much of a signal in the analysis)
- dynamicalPtRange 0: same pT range for all particles, 1: scale max. pT according to particle masses
- useLMeeDecayTable set branching ratios from decay table (“”=do not use decaytable) (from O2DPG or O2Physics/Tutorials/PWGEM/Cocktail/files/decaytables)
- weightingMode 0: analog, 1: non-analog
- pythiaErrorTolerance Pythia error tolerance
- paramFileV2Dir ‘directory’ for elliptic flow parametrizations in paramFile (we do not have this in our paramFile. Just leave it)
- useYWeights use rapidity dependent weights from paramFile (we do not have this in our paramFile. Just leave it)
- decayLongLived not implemented
- toFixEP not implemented

You can play around with different paramFiles, selectedMothers, externalDecayer, numberOfParticles…

Side note: the sim-ini-file

- The generator actually takes the inputs in form of a .ini file



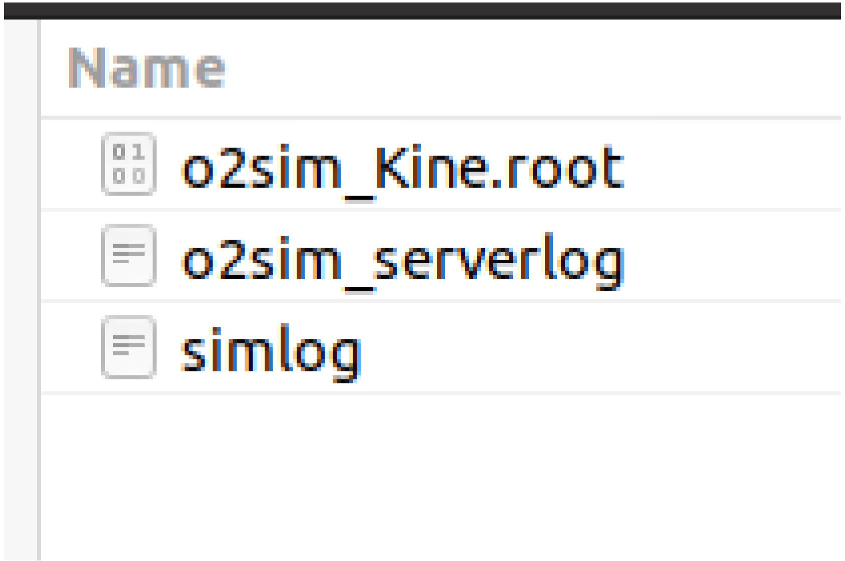
```
[GeneratorExternal]
fileName = ${O2DPG_ROOT}/MC/config/PWGEM/external/generator/GeneratorEMCocktailV2.C
funcName=GenerateEMCocktail(400,0,3,63,"${O2DPG_ROOT}/MC/config/PWGEM/parametrizations/
PbPb5TeV_central.json","5TeV_0005_wRatio_etatest",350,0.0,30.0,10000,1,1,0,0,"",0,1.1,"${O2DPG_ROOT}/MC/config/PWGEM/decaytables/decaytable_LMee.dat",0)
```

- The python script reads the sim-config-file and converts it into a .ini file
  - you find that as GeneratorEMCocktail.ini in the output directory

- If you have a .ini file of that form, you can use it directly instead of the sim-config-file
  - just use --sim-ini-file <sim_ini_file> instead of --sim-config-file <sim_config_file>
  - you find an example for a .ini file that you can run without modifications under O2Physics/Tutorial/PWGEM/Cocktail/configs/ini_example.ini

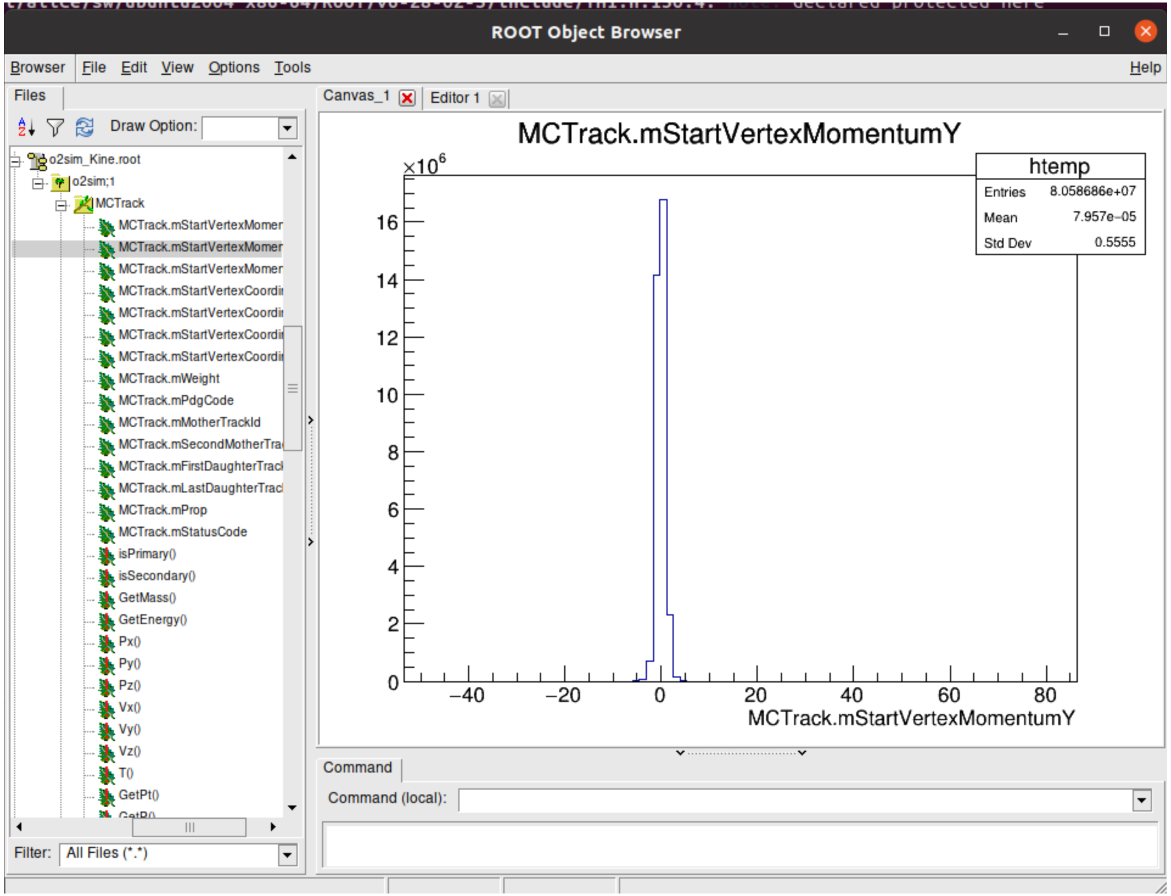
If you have problems/questions/comments/…, please speak up!

What you should have now



Name

- o2sim_Kine.root
- o2sim_serverlog
- simlog



ROOT Object Browser

```
o2sim_Kine.root
  o2sim;1
    MCTrack
      MCTrack.mStartVertexMomer
      MCTrack.mStartVertexMomer
      MCTrack.mStartVertexMomer
      MCTrack.mStartVertexCoordi
      MCTrack.mStartVertexCoordi
      MCTrack.mStartVertexCoordi
      MCTrack.mStartVertexCoordi
      MCTrack.mWeight
      MCTrack.mPdgCode
      MCTrack.mMotherTrackId
      MCTrack.mSecondMotherTra
      MCTrack.mFirstDaughterTrack
      MCTrack.mLastDaughterTrack
      MCTrack.mProp
      MCTrack.mStatusCode
      isPrimary()
      isSecondary()
      GetMass()
      GetEnergy()
      Px()
      Py()
      Pz()
      Vx()
      Vy()
      Vz()
      T()
      GetPt()
      GetP()
```

Filter: All Files (*.*)

MCTrack.mStartVertexMomentumY

htemp	
Entries	8.058686e+07
Mean	7.957e−05
Std Dev	0.5555

STEP 2: Analysis

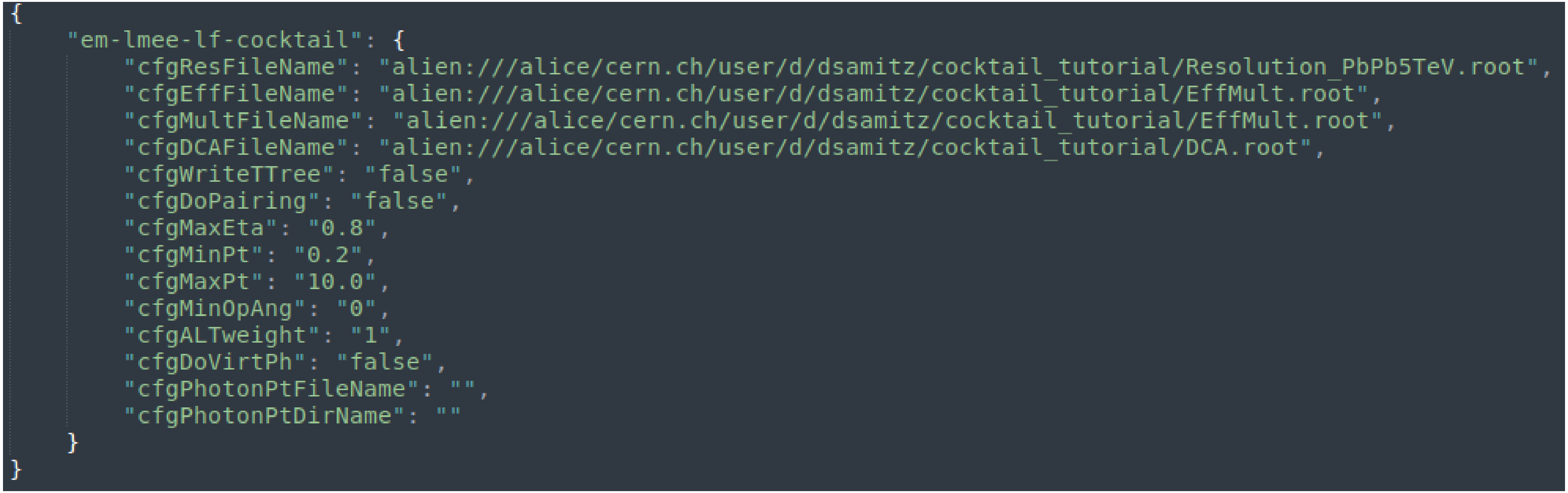
‘ana-only’ command

- Command to run the light flavour cocktail analysis task
  - reads: o2sim_Kine.root
  - produces: AnalysisResults.root with histograms and trees
- With O2Physics,O2DPG,AEGIS (or if installed O2sim) loaded, run

```
python run_lmee_lf_cocktail.py ana-only -i <input_file> --ana-config-file <ana_config_file> [-o <output_dir>]
```

  - <input_file>: o2sim_Kine.root file to be analyzed (<input_file> = filename without ‘_Kine.root’!)
    - take the one produced in previous step (or one from here: https://cernbox.cern.ch/s/aLFkmdKQzJrdWwl)
  - <ana_config_file>: the analysis config file that contains all relevant settings
    - see next slide for details
  - <output_dir> [optional]: output directory
    - default = current directory

The ana-config-file (O2Physics/Tutorial/PWGEM/Cocktail/configs/ana_config.json)



```
{
    "em-lmee-lf-cocktail": {
        "cfgResFileName": "alien:///alice/cern.ch/user/d/dsamitz/cocktail_tutorial/Resolution_PbPb5TeV.root",
        "cfgEffFileName": "alien:///alice/cern.ch/user/d/dsamitz/cocktail_tutorial/EffMult.root",
        "cfgMultFileName": "alien:///alice/cern.ch/user/d/dsamitz/cocktail_tutorial/EffMult.root",
        "cfgDCAFileName": "alien:///alice/cern.ch/user/d/dsamitz/cocktail_tutorial/DCA.root",
        "cfgWriteTTree": "false",
        "cfgDoPairing": "false",
        "cfgMaxEta": "0.8",
        "cfgMinPt": "0.2",
        "cfgMaxPt": "10.0",
        "cfgMinOpAng": "0",
        "cfgALTweight": "1",
        "cfgDoVirtPh": "false",
        "cfgPhotonPtFileName": "",
        "cfgPhotonPtDirName": ""
    }
}
```

The ana-config-file

When using files form alien, be sure to have Grid access
(do ‘alien.py’ and enter your password, then exit)

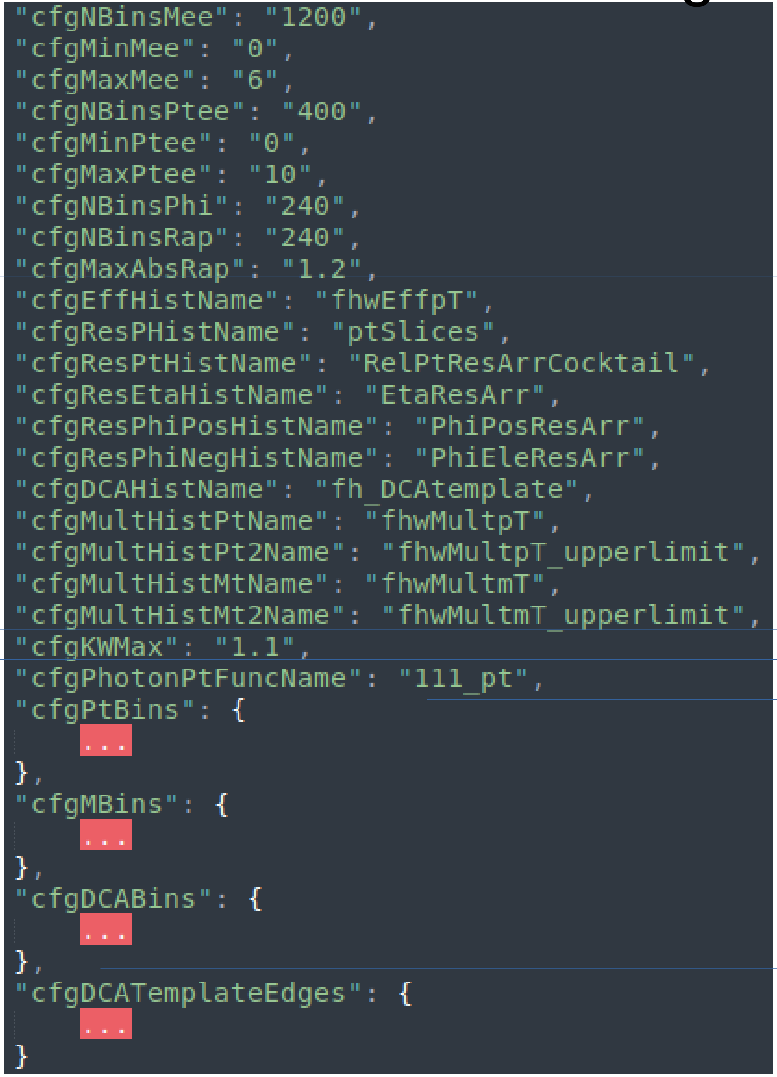
I did not manage to do this on lxplus,
here you maybe have to change the paths to the local files,
e.g..
"cfgResFileName": "/afs/cern.ch/user/...”

- cfgResFileName: file with momentum resolution maps
  - (from alien or locally under O2Physics/Tutorials/PWGEM/Cocktail/files/resolution/)
- cfgEffFileName: file with standard efficiency weights
  - (from alien or locally under O2Physics/Tutorials/PWGEM/Cocktail/files/resolution/)
- cfgMultFileName: file with multiplicity weights
  - (from alien or locally under O2Physics/Tutorials/PWGEM/Cocktail/files/resolution/)
- cfgDCAFileName: file with DCA templates
  - (from alien or locally under O2Physics/Tutorials/PWGEM/Cocktail/files/resolution/)
- cfgWriteTree: enable tree output
- cfgDoPairing: do LS&ULS same event pairing (better to set “false” today when analyzing large files, can take quite some time)
- cfgMaxEta, cfgMinPt, cfgMaxPt, cfgMinOpAng: acceptance cuts
- cfgALTweight: choose which alternative weights to use (0: standard: 1,11,2,22: multiplicity weights)
- cfgDoVirtPh: generate one virtual photon for each pion in the event
  - If switched on, you need to provide a pT parametrization file and directory via ‘cfgPhotonPtFileName’ and ‘cfgPhotonPtDirName’
    - Use the parametrization files in O2Physics/Tutorial/PWGEM/Cocktail/files/parametrizations/

(when using relative paths: ATTENTION! is relative to ‘output’ directory)

You can play around with different resolution files, tree output, acceptance cuts,...

Side note: the ana-config-file for experts (check the dumped dpl-config.json)



```
"cfgNBinsMee": "1200",
"cfgMinMee": "0",
"cfgMaxMee": "6",
"cfgNBinsPtee": "400",
"cfgMinPtee": "0",
"cfgMaxPtee": "10",
"cfgNBinsPhi": "240",
"cfgNBinsRap": "240",
"cfgMaxAbsRap": "1.2",
"cfgEffHistName": "fhwEffpT",
"cfgResPHistName": "ptSlices",
"cfgResPtHistName": "RelPtResArrCocktail",
"cfgResEtaHistName": "EtaResArr",
"cfgResPhiPosHistName": "PhiPosResArr",
"cfgResPhiNegHistName": "PhiEleResArr",
"cfgDCAHistName": "fh_DCAtemplate",
"cfgMultHistPtName": "fhwMultpT",
"cfgMultHistPt2Name": "fhwMultpT_upperlimit",
"cfgMultHistMtName": "fhwMultmT",
"cfgMultHistMt2Name": "fhwMultmT_upperlimit",
"cfgKWMax": "1.1",
"cfgPhotonPtFuncName": "111_pt",
"cfgPtBins": {
    ...
},
"cfgMBins": {
    ...
},
"cfgDCABins": {
    ...
},
"cfgDCATemplateEdges": {
    ...
}
```

nBins and ranges of histograms

names of histograms in ResFile, MultFile, EffFile, DCAfile that are loaded

max inv. mass for virtual photon from KrollWada

which pT parametrization function to use for virtual photons (111_pt is pi0)

variable bins for the DCA histograms

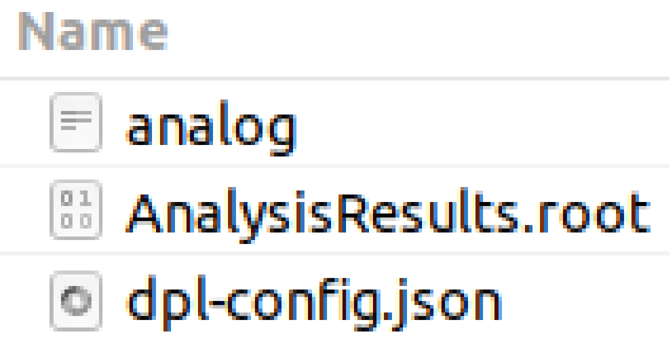
pT ranges for the DCA templates

If you have problems/questions/comments/…, please speak up!

btw, you can use O2Physics/Tutorials/PWGEM/Cocktail/plotLFCocktail.C macro to plot the invariant mass spectrum

plotLFCocktail(“your AnalysisResults.root file”, rebin (default = 1))

What you should have now

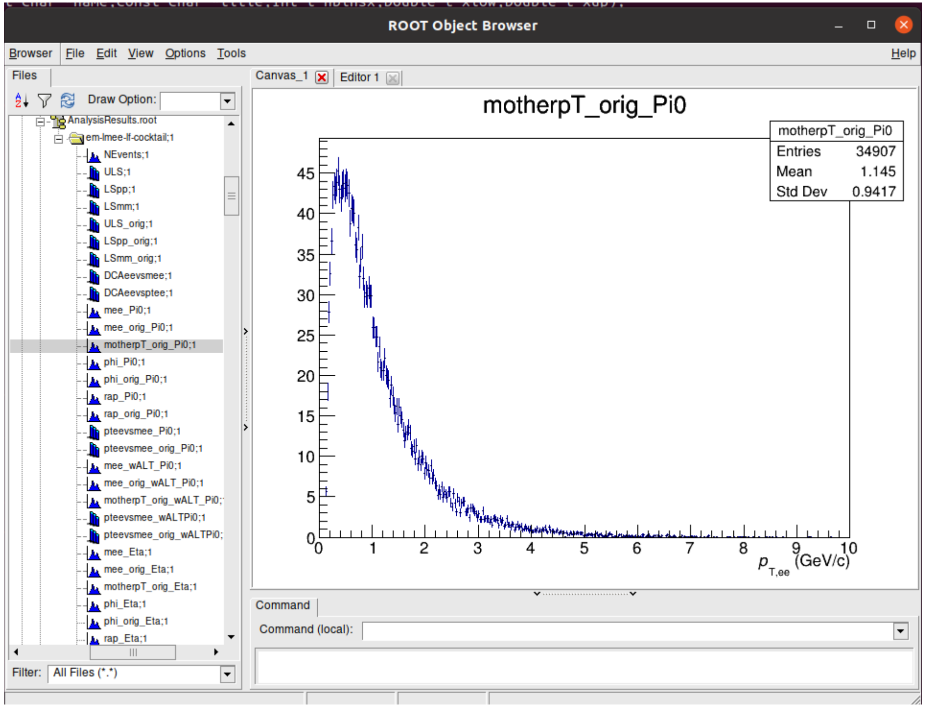


Name

analog

AnalysisResults.root

dpl-config.json

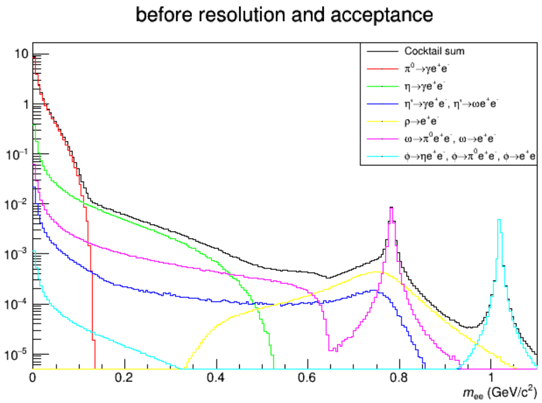


AnalysisResults.root
em-lmee-lf-cocktail;1
NEvents;1
ULS;1
LSpp;1
LSmm;1
ULS_orig;1
LSpp_orig;1
LSmm_orig;1
DCAeevsmee;1
DCAeevsptee;1
mee_Pi0;1
mee_orig_Pi0;1
motherpT_orig_Pi0;1
phi_Pi0;1
phi_orig_Pi0;1
rap_Pi0;1
rap_orig_Pi0;1
pteevsmee_Pi0;1
pteevsmee_orig_Pi0;1
mee_wALT_Pi0;1
mee_orig_wALT_Pi0;1
motherpT_orig_wALT_Pi0;
pteevsmee_wALTPi0;1
pteevsmee_orig_wALTPi0;
mee_Eta;1
mee_orig_Eta;1
motherpT_orig_Eta;1
phi_Eta;1
phi_orig_Eta;1
rap_Eta;1

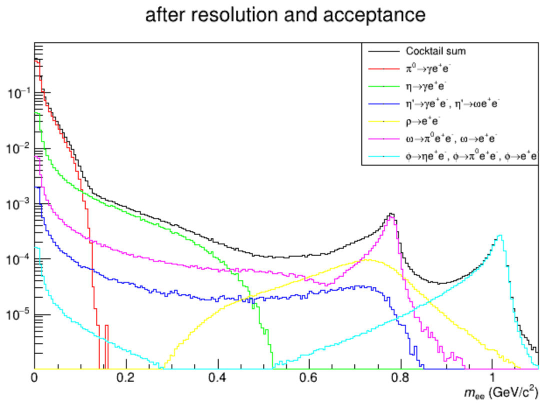
motherpT_orig_Pi0

motherpT_orig_Pi0	
Entries	34907
Mean	1.145
Std Dev	0.9417

plotLFCocktail.C ==>



before resolution and acceptance



after resolution and acceptance

STEP 3: Everything together

‘full’ command

- Run the full framework (simulation+analysis) as on-the-fly MC analysis
  - i.e. simulation output gets piped directly to analysis task
  - simulation output does not need to be written to disk at all

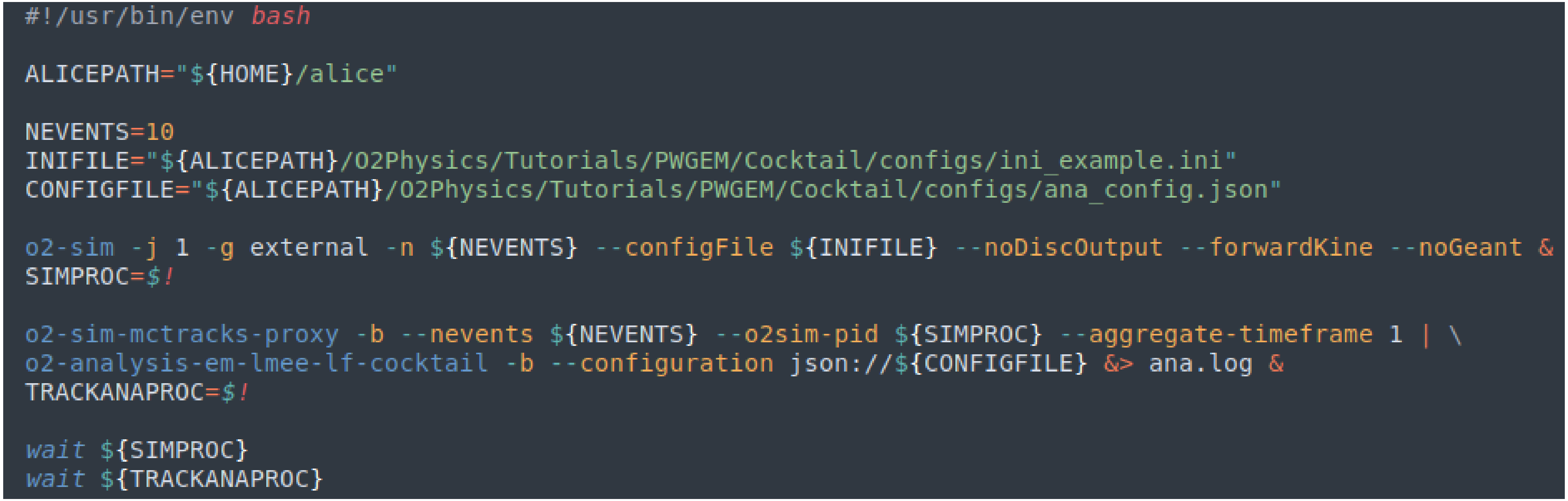
```
python run_lmee_lf_cocktail.py full -n <nEvents> --sim-config-file <sim_config_file> --ana-config-file <ana_config_file>
    [-o <output_dir>]  [--clean] [--save-kine]
```

  - <nEvents>, <sim_config_file>, <ana_config_file>, <output_dir>, [--clean]
    - as in ‘sim-only’ and ‘ana-only’ command
  - --save-kine [optional]
    - write simulation output to o2sim_Kine.root

On-the-fly MC analyses

- Here everything is steered by the python script,
  but maybe it is instructive to see the actual commands involved

- Below a bash script that executes the same on-the-fly analysis

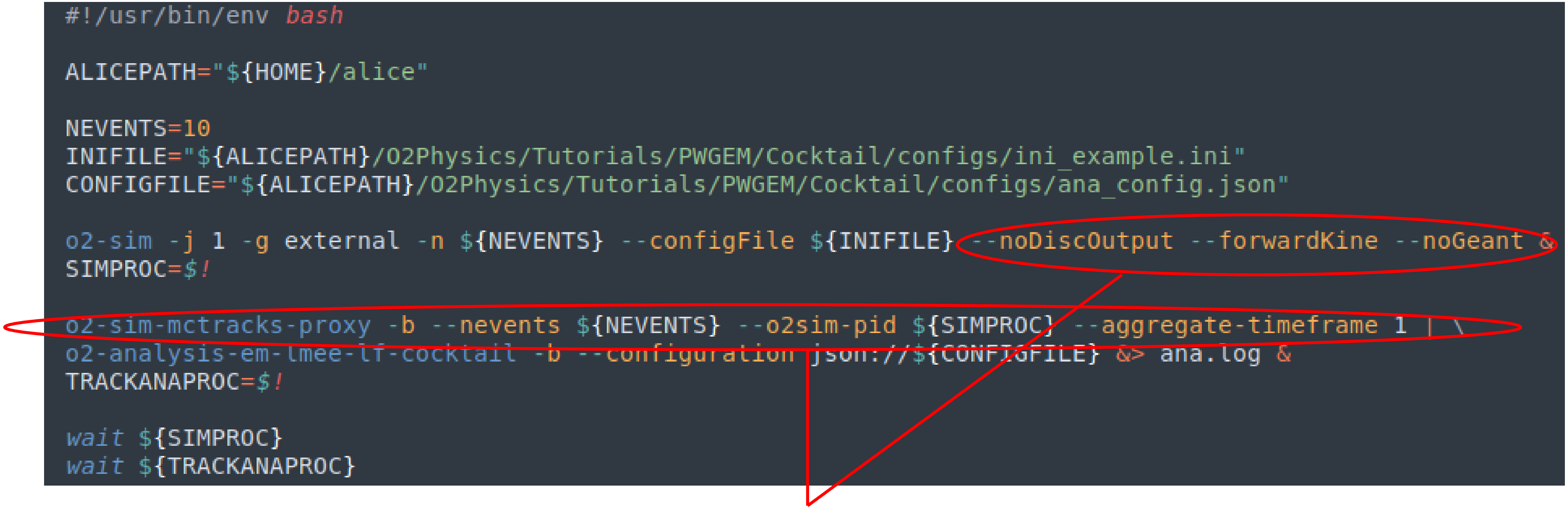


```bash
#!/usr/bin/env bash

ALICEPATH="${HOME}/alice"

NEVENTS=10
INIFILE="${ALICEPATH}/O2Physics/Tutorials/PWGEM/Cocktail/configs/ini_example.ini"
CONFIGFILE="${ALICEPATH}/O2Physics/Tutorials/PWGEM/Cocktail/configs/ana_config.json"

o2-sim -j 1 -g external -n ${NEVENTS} --configFile ${INIFILE} --noDiscOutput --forwardKine --noGeant &
SIMPROC=$!

o2-sim-mctracks-proxy -b --nevents ${NEVENTS} --o2sim-pid ${SIMPROC} --aggregate-timeframe 1 | \
o2-analysis-em-lmee-lf-cocktail -b --configuration json://${CONFIGFILE} &> ana.log &
TRACKANAPROC=$!

wait ${SIMPROC}
wait ${TRACKANAPROC}
```

On-the-fly MC analyses

- Here everything is steered by the python script,
  but maybe it is instructive to see the actual commands involved

- Below a bash script that executes the same on-the-fly analysis



```bash
#!/usr/bin/env bash

ALICEPATH="${HOME}/alice"

NEVENTS=10
INIFILE="${ALICEPATH}/O2Physics/Tutorials/PWGEM/Cocktail/configs/ini_example.ini"
CONFIGFILE="${ALICEPATH}/O2Physics/Tutorials/PWGEM/Cocktail/configs/ana_config.json"

o2-sim -j 1 -g external -n ${NEVENTS} --configFile ${INIFILE} --noDiscOutput --forwardKine --noGeant &
SIMPROC=$!

o2-sim-mctracks-proxy -b --nevents ${NEVENTS} --o2sim-pid ${SIMPROC} --aggregate-timeframe 1 | \
o2-analysis-em-lmee-lf-cocktail -b --configuration json://${CONFIGFILE} &> ana.log &
TRACKANAPROC=$!

wait ${SIMPROC}
wait ${TRACKANAPROC}
```

This is the crucial part for on-the-fly MC analyses

If you have problems/questions/comments/…, please speak up!