WZ D3PD production

M Bellomo[(1)], C. Gatti[(2)]

[(1)]INFN Pavia

[(2)]LNF

W,Z muon analysis

January 29, 2009

D3PDs

- **D3PD** are ROOT plain ntuple used as final step in analysis
  - ★ Remembering that each analysis must be able to reproduce results from POOL (AOD D12PD)
  - ★ In early period ntuples useful for fast analysis development
  - ★ scalability to high luminosity to be verified
- **D3PD** produced in **EWPA** using **NtupleMaker** athena tool
  - ★ ad-hoc blocks for truth trigger track muon electron photon jet tauJet MET cluster
  - ★ general blocks for UserData (int float string bool)
  - ★ centralized tool allowing for common bug-fix cycles

D3PD common format

- **D3PD** are ROOT plain ntuple used as final step in analysis
  - ★ Remembering that each analysis must be able to reproduce results from POOL (AOD D12PD)
  - ★ In early period ntuples useful for fast analysis development
  - ★ scalability to high luminosity to be verified
- **D3PD** produced in **EWPA** using **NtupleMaker** athena tool
  - ★ ad-hoc blocks for truth trigger track muon electron photon jet tauJet MET cluster
  - ★ general blocks for UserData (int float string bool)
  - ★ centralized tool allowing for common bug-fix cycles

Common block	N eta pt px py pz e m charge
Truth block + common	pdgId motherPdgId prodVx prodVy prodVz endVx endVy endVz prodVbarcode endVbarcode d z phi theta qOverp barcode status
Track block + common	vertexType d z phi theta qOverp chi2 ndf numberOfBLayerHits numberOfPixelHits numberOfSCTHits numberOfTRTHits numberOfTRTHighThresholdHits numberOfMDTHits d_cov z_cov phi_cov theta_cov qOverp_cov z_d_cov z_phi_cov z_theta_cov z_qOverp_cov d_phi_cov d_theta_cov d_qOverp_cov phi_theta_cov

D3PD common format

Muon block + common	author d z phi theta qOverp matchChi2 matchNumberDoF fitChi2 fitNumberDoF etCone20 etCone30 etCone40 nuCone20 nuCone30 nuCone40 ptCone20 ptCone30 ptCone40 energyLoss bestMatch isStandAloneMuon isCombinedMuon isLowPtReconstructedMuon d_cov z_cov phi_cov theta_cov qOverp_cov z_d_cov z_phi_cov z_theta_cov z_qOverp_cov d_phi_cov d_theta_cov d_qOverp_cov phi_theta_cov phi_qOverp_cov theta_qOverp_cov
Electron block + common	d z phi theta qOverp author hasTrack isEM etcone etcone20 etcone30 etcone40 e233 e237 e277 weight bgweight eOverp d_cov z_cov phi_cov theta_cov qOverp_cov z_d_cov z_phi_cov z_theta_cov z_qOverp_cov d_phi_cov d_theta_cov d_qOverp_cov phi_theta_cov phi_qOverp_cov theta_qOverp_cov
Photon block + common	author isEM etcone etcone20 etcone30 etcone40 e233 e237 e277 weight bgweight CnvAngleMatch CnvTrackMatch CnvVtx_x CnvVtx_y CnvVtx_z
Jet block + common	tag truthTag truePdg deltaRMinToB deltaRMinToC deltaRMinToT btag_ntag btag_w_cmb btag_w_ip2d btag_w_ip3d btag_w_sv1 btag_w_sv2 btag_w_lf2d btag_w_svbu btag_w_lhsig

D3PD common format

Tau Jet block + common	(to be extended)			
MET block	pt px py phi sumet			
Cluster block + common	phi			
Trigger Features block + common	phi			
Trigger Tool block (int value = 0,1)	L1_MU4 L1_MU6 L1_MU10 L1_MU11 L1_MU20 L1_MU40 L1_2MU4 L1_2MU6 L1_2MU10 L1_2MU20	L2_mu4 L2_mu6 L2_mu10 L2_mu11 L2_mu20 L2_mu40 L2_2mu4 L2_2mu6 L2_2mu10 L2_2mu20	EF_mu4 EF_mu6 EF_mu10 EF_mu11 EF_mu20 EF_mu40 EF_2mu4 EF_2mu6 EF_2mu10 EF_2mu20	For each trigger item: isPhysicsPassed, isPassed, isRawPassed, isPrescaled, isPassThrough (N.B. isPassed includes pass-through and prescale while isPhysicsPassed only prescale)

D3PD physics content

- The physics content of the D3PD is fully configurable from the jopOption: EWJobOptions/share/DPDMakingConfig/make_D3PD.py
- Particle reading from AOD through StoreGate in Athena is done with a set of tools for each type of reconstructed object : see **EWInserters** package

```
if loadTruth:
    insertTruth("TrueMuon",        PdgID=13, PtCut=1.0*GeV, EtaCut=3.0)
    insertTruth("TrueNeutrinoMu", PdgID=14)
    insertTruth("TrueElectron",    PdgID=11)
    insertTruth("TrueNeutrinoEl", PdgID=12)
    insertTruth("TruePhoton",      PdgID=22)
    insertTruth("TrueZ",           PdgID=23, Status=3, PtCut=0.*GeV, EtaCut=10.0, DoBarcode=False)
    insertTruth("TrueW",           PdgID=24, Status=3, PtCut=0.*GeV, EtaCut=10.0, DoBarcode=False)

if loadTrigger:
    #============================================================
    # Muon Trigger to be read
    #============================================================
    insertTrigger("L1MuonTriggerInserter", TriggerSlice="Muon", TriggerLevel="L1")
    insertTrigger("HLTMuonTriggerInserter", TriggerSlice="Muon", TriggerLevel="HLT")

if loadTrack:
    #============================================================
    # Track(s) to be read
    #============================================================
    insertTrack("TrackIDInserter",
                { "TrackID" : "TrackParticleCandidate" },
                PtCut=4.0*GeV, EtaCut=3.0, doThin=False)

    insertTrack("TrackMSInserter",
                { "MuidSAMuon"    : "MuidExtrTrackParticles",
                  "MuonBoySAMuon" : "MuonboyTrackParticles" ,
                  "MuTagMuon"     : "MuTagTrackParticles"   },
                PtCut=1.0*GeV, EtaCut=3.0, doThin=False)
```

cuts to select MC particles

“label” : “StoreGate container key”

CUTS TO SELECT TRACKS ARE ACTIVE ONLY IF doThin = TRUE

D3PD physics content

```
if loadMuon:
    #============================================================
    # Muon(s) to be read
    #============================================================
    insertMuon("MuidCBInserter",
               { "MuidCBMuon" : "MuidMuonCollection" },
               AuthorCut="MuidCo", StatusCut=1, doThin=True)

    insertMuon("StacoInserter",
               { "StacoMuon"  : "StacoMuonCollection" },
               AuthorCut="STACO", StatusCut=1, doThin=True)

    insertMuon("CaloMuonIdInserter",
               { "CaloMuon"  : "CaloMuonCollection" },
               StatusCut=4, doThin=True)

if loadElectron:
    #============================================================
    # Electron(s) to be read
    #============================================================
    insertElectron("ElectronInserter",
                   { "Electron" : "ElectronAODCollection" },
                   PtCut=1.0*GeV, EtaCut=3.0, doThin=False)

if loadPhoton:
    #============================================================
    # Photon(s) to be read
    #============================================================
    insertPhoton("PhotonInserter",
                 { "Photon" : "PhotonAODCollection" },
                 PtCut=1.0*GeV, EtaCut=5.0, doThin=False)

if loadJet:
    #============================================================
    # Jet(s) to be read
    #============================================================
    insertJet("JetInserter",
              { "JetCone4H1TW" : "Cone4H1TowerJets" },
              PtCut=1.0*GeV, EtaCut=5.0, doThin=False)
```

```
if loadTauJet:
    #============================================================
    # TauJet(s) to be read
    #============================================================
    insertTauJet("TauJetInserter",
                 { "TauRec" : "TauRecContainer" },
                 PtCut=1.0*GeV, EtaCut=5.0, doThin=False)

if loadCluster:
    #============================================================
    # Cluster(s) to be read
    #============================================================
    insertCluster("ClusterInserter",
                  { "TopoCluster" : "CaloCalTopoCluster" },
                  PtCut=1.0*GeV, EtaCut=5.0, doSkim=False, doThin=False)
if loadBJet:
    #============================================================
    # BJet(s) to be read
    #============================================================
    insertBJet("BJetInserter",
               { "BJet" : "Cone4H1TowerJets" },
               PtCut=1.0*GeV, EtaCut=5.0, doThin=False)

if loadMet:
    #============================================================
    # add missing energy
    #============================================================
    insertMet("MetInserter",
              { "ObjMETFinal" : "ObjMET_Final" ,
                "METLocHadTopoObj" : "MET_LocHadTopoObj" ,
                "METMuonBoy" : "MET_MuonBoy" ,
                "METFinal" : "MET_Final" })

    insertMet("MetInserterRefFinal",
              { "METRefFinal" : "MET_RefFinal" },
              EtCut=10.0*GeV, doSkim=False)

    if EWDataType == "SIMUL":
        insertMet("TruthMetInserter",
                  { "METTruth" : "MET_Truth" })
```

test with Wmunu dataset with <u>all selections disabled</u> on lxplus
D3PD size about 10-15 kb/event - D3PD making time about 100 ms/event

D3PD user defined format

- Clearly there can be the need for an additional user-defined block
  - ★ this can be added as UserData to the existing set of information
  - ★ you can easily create a new tool that computes this UserData and add it to the D3PD, test it, then if you feel it can be useful to all we can include it in the common production

User Data

User Data is added to the ntuple in 2 steps:

- (1) Book the ntuple branch: a branch in the ntuple has to be reserved to hold the data. This is done at the jobOption level calling a dedicated python function.

Here are the use cases ("Int" type is used as example, other possibilities are "Float", "Bool", "String":

- Event-level Scalar User Data :

```
insertUD("Event", UDIntList=["NtupleVersion"])
insertUD("Event", UDIntList=["RunNumber","EventNumber","TimeStamp","TimeStampNsOffset","LumiBlock","BunchCrossingId"])
```

- Event-level Vectorial User Data :

```
insertUD("Event", UDVecIntList=["TagsOfTheEvent"])
```

- Particle-level Scalar User Data :

```
insertUD("TrigMuonL1", UDIntList=["NumOfIDTrackInCone"])
```

Note that the first parameter is the label used to associate User Data to objects: "Event" for event-level user data and the particle label in the other case. Adding label to truth objects is special: you can decide to add User Data to all truth objects using as label "Truth" or only to specific ones using its label (as "TrueMuon" to give User Data only to MC muons).

D3PD user defined format - example

- An example of already implemented UserData is the association information between tracks and MC objects
  - ★ a set of associations are defined in the jobOption EWJobOptions/share/DPDMakingConfig/make_D3PD.py

```
        addAssociation("TrigMuonL1",   "StacoMuon", "EtaPhi", 0.3)
        addAssociation("TrigMuonL2SA", "StacoMuon", "EtaPhi", 0.15)
        addAssociation("TrigMuonL2CB", "StacoMuon", "EtaPhi", 0.1)
        addAssociation("TrigMuonEFSA", "StacoMuon", "EtaPhi", 0.3)
        addAssociation("TrigMuonEFCB", "StacoMuon", "EtaPhi", 0.05)

    if loadTrack :
        addAssociation("MuidSAMuon",    "TrackID", "EtaPhi", 0.2)
        addAssociation("MuonBoySAMuon", "TrackID", "EtaPhi", 0.2)
        addAssociation("MuTagMuon",     "TrackID", "EtaPhi", 0.05)

    if loadTrack and loadMuon:
        addAssociation("MuidCBMuon",    "TrackID", "EtaPhi", 0.05)
        addAssociation("StacoMuon",     "TrackID", "EtaPhi", 0.05)
        addAssociation("CaloMuon",      "TrackID", "EtaPhi", 0.05)

    if loadTruth and EWDataType == "SIMUL" and loadTrigger:
        addAssociation("TrigMuonL1",   "TrueMuon", "EtaPhi", 0.3)
        addAssociation("TrigMuonL2SA", "TrueMuon", "EtaPhi", 0.15)
        addAssociation("TrigMuonL2CB", "TrueMuon", "EtaPhi", 0.1)
        addAssociation("TrigMuonEFSA", "TrueMuon", "EtaPhi", 0.3)
        addAssociation("TrigMuonEFCB", "TrueMuon", "EtaPhi", 0.05)

    if loadTruth and EWDataType == "SIMUL" and loadTrack:
        addAssociation("MuidSAMuon",    "TrueMuon", "EtaPhi", 0.4)
        addAssociation("MuidSAMuon",    "TrueMuon", "Barcode", 10.)
        addAssociation("MuonBoySAMuon", "TrueMuon", "EtaPhi", 0.4)
        addAssociation("MuonBoySAMuon", "TrueMuon", "Barcode", 10.)
        addAssociation("TrackID",       "TrueMuon", "EtaPhi", 0.05)
        addAssociation("TrackID",       "TrueMuon", "Barcode", 10.)
        addAssociation("MuTagMuon",     "TrueMuon", "EtaPhi", 0.05)

    if loadTruth and EWDataType == "SIMUL" and loadMuon:
        addAssociation("MuidCBMuon",    "TrueMuon", "EtaPhi", 0.05)
        addAssociation("StacoMuon",     "TrueMuon", "EtaPhi", 0.05)
        addAssociation("CaloMuon",      "TrueMuon", "EtaPhi", 0.05)
```

D3PD user defined format - example

- An example of already implemented UserData is the association information between tracks and MC objects

`MuidCBMuon_AssoId_TrueMuon_EtaPhi`

This for instance has the positional index, in the ntuple, of the associated "TrueMuon". Giving this command line to ROOT:

```
ewd3pd.Scan("TrueMuon_pt[MuidCBMuon_AssoId_TrueMuon_EtaPhi]:MuidCBMuon_pt","MuidCBMuon_AssoId_TrueMuon_EtaPhi>-1","",20,0)
```

and you will get something like:

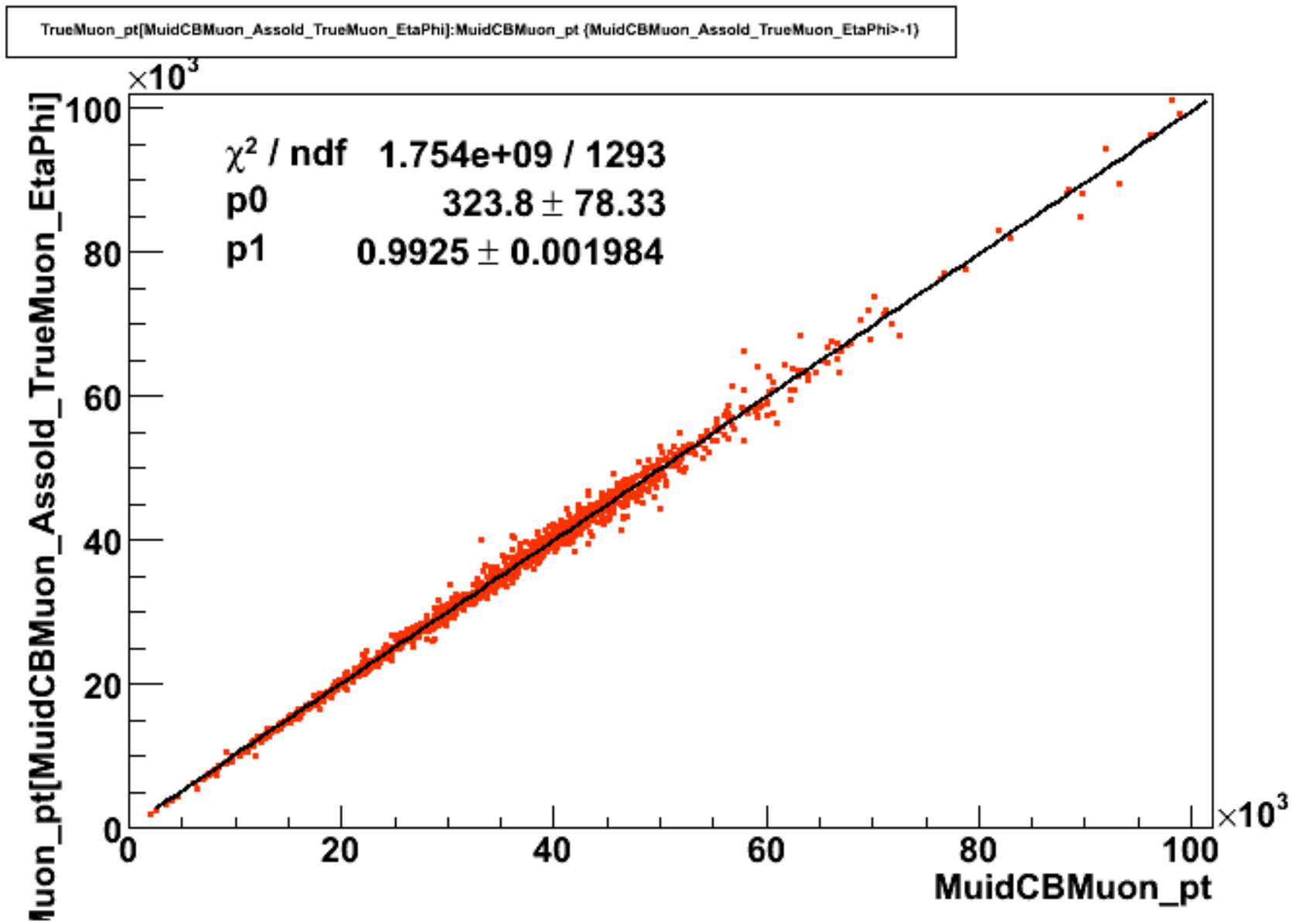
```
root [10] ewd3pd.Scan("TrueMuon_pt[MuidCBMuon_AssoId_TrueMuon_EtaPhi]:MuidCBMuon_pt","MuidCBMuon_AssoId_TrueMuon_EtaPhi>-1","",20,0)
************************************************
*    Row   * Instance * TrueMuon_ * MuidCBMuo *
************************************************
*        0 *        0 * 42637.289 * 43385.609 *
*        2 *        0 * 36299.414 * 36259.824 *
*        2 *        1 * 34636.246 * 35900.179 *
*        4 *        0 * 10858.209 * 11043.053 *
*        6 *        0 * 70463.585 * 68838.921 *
*        6 *        1 * 59280.457 * 60117.105 *
*        6 *        2 * 2010.1121 * 2012.2266 *
*        8 *        0 * 13325.550 * 13296.592 *
*        9 *        0 * 27497.753 * 26942.640 *
*       10 *        0 * 46284.164 * 46268.519 *
*       10 *        1 * 36516.773 * 36347.089 *
```

D3PD user defined format - example

- An example of already implemented UserData is the association information between tracks and MC objects

A simple 1-line plot as pT correlation can be done as:

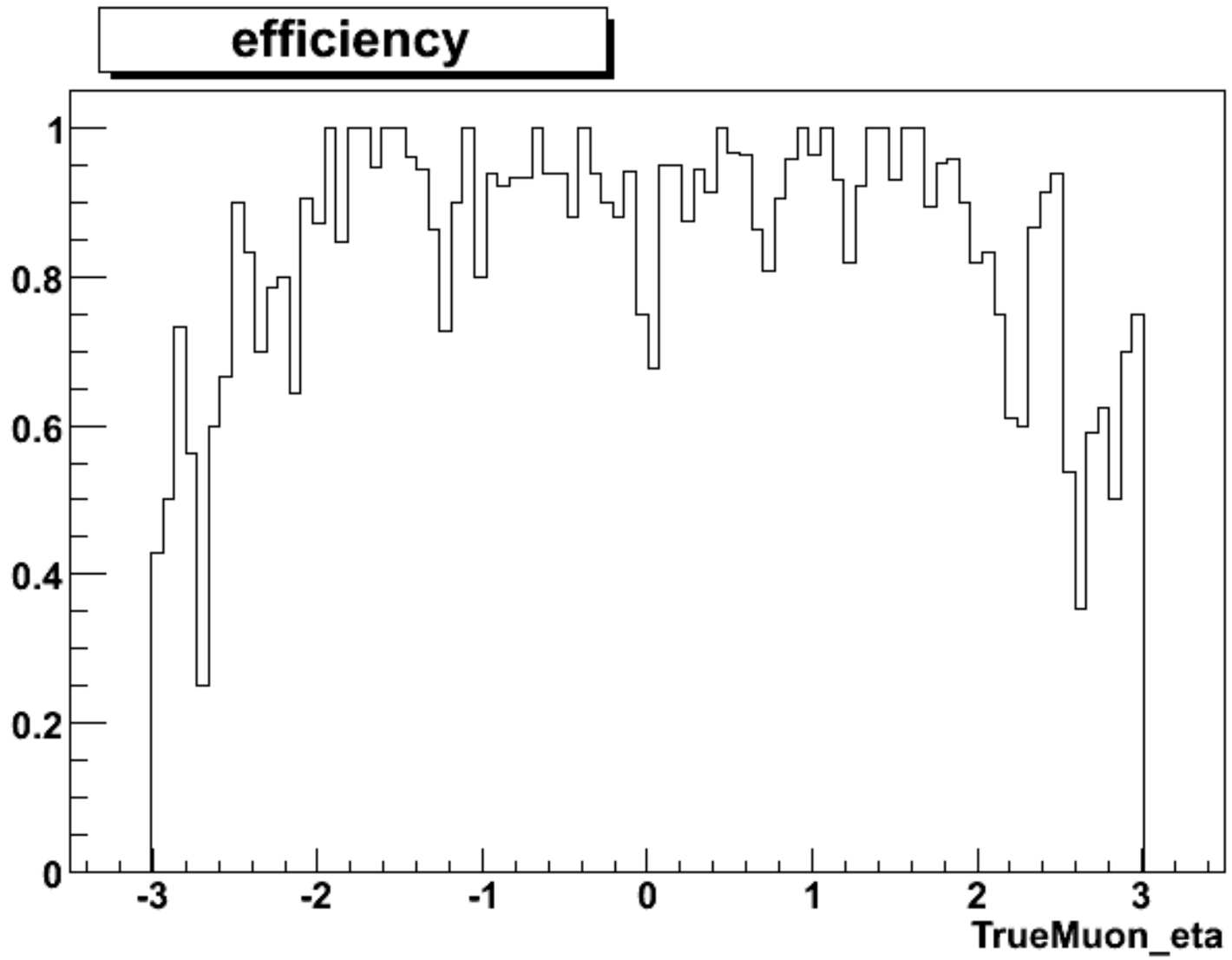
```
root[0] ewd3pd.Draw("TrueMuon_pt[MuidCBMuon_AssoId_TrueMuon_EtaPhi]:MuidCBMuon_pt","MuidCBMuon_AssoId_TrueMuon_EtaPhi>-1")
```



D3PD user defined format - example

- An example of already implemented UserData is the association information between tracks and MC objects

```
root[0] ewd3pd.Draw("TrueMuon_eta","MuidCBMuon_AssoId_TrueMuon_EtaPhi>-1")
root[1] TH1F* hpassed = (TH1F*)htemp->Clone("hpassed")
root[2] ewd3pd.Draw("TrueMuon_eta")
root[3] hpassed->Divide(htemp)
root[4] hpassed->Draw()
```



D3PD reading

- The main advantage of ntuples is of course that you can read them in the way you like, directly from ROOT command line, with simple ROOT macros, ...
- The draw-back of this is that you can end-up with a not so simple macro of really thousands lines of code ...
- Remember also that *by now* ATLAS demands that the analysis code must be able to run in Athena from POOL data

D3PD reading

- The main advantage of ntuples is of course that you can read them in the way you like, directly from ROOT command line, with simple ROOT macros, ...
- The draw-back of this is that you can end-up with a not so simple macro of really thousands lines of code ...
- Remember also that *by now* ATLAS demands that the analysis code must be able to run in Athena from POOL data

- An alternative approach wrt to ROOT macros is to re-read back in EWPA the ntuple
  - ★ EWPA can run in “standalone” mode independent from Athena (fitting nicely in you laptop!)
  - ★ a dedicated NtupleInserter do this job and fills back the EventLibrary
  - ★ you can define which branch block has to be read in the jobOption
  - ★ the analysis code is invariant for input and running environment: **you can run it from AOD in Athena without changes**
  - ★ the job configuration is done with the same python jobOption: **you can use the same configuration parameters for your analysis tool also in Athena**

- Find out more @ https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EWPARunStandalone

production details

- You can monitor the status of D3PD production from
  https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EWAnalysisWZMeasurements#Samples_Output
- File name policy
  AOD ----> mc08.106051.PythiaZmumu_1Lepton.**recon**.e347_s462_r541_r581
  D2PD ---> mc08.106051.PythiaZmumu_1Lepton.**EWPA_D2PD_N**.e347_s462_r541_r581
  D3PD ---> mc08.106051.PythiaZmumu_1Lepton.**EWPA_D3PD_N**.e347_s462_r541_r581
  recon -> EWPA_DnPD_N
- "N" stands for a production id (1,2,3,4, ...)
  - ★ especially at the beginning it will be necessary to re-produce D3PD to add variables and/or fix bugs
- Datasets are registered in GRID and you can list/retrieve with dq2 tools
  - ★ a copy available also in CASTOR (temporary), see InputCollections.py listed in the twiki
- Currently listed D3PDs are from kick-off production (buggy trigger tool info)
  - ★ Some N3 production are ongoing with fixes but only for isPassed info
- N4 production with all values reported in previous table is planned shortly
  - ★ all other information is already there by the kick-off production
    - if you are interested in these D3PDs: please test them to help us spot new bugs
    - if you feel some info is missing for your analysis please tell us to get it included for N4

production details

Full Simulation

Using tags eXXX_s462_r541 with ATLAS-GEO-02-01-00.
D3PD datasets are registered on the grid with DQ2, you can get the dataset name from the InputCollections.py file linked in the table.

processed dataset AOD	sample statistic	InputCollection
mc08.106051.PythiaZmumu_1Lepton.recon.e347_s462_r541_r581	300000	InputCollections.py
mc08.106021.PythiaWmunu_1Lepton		
mc08.106052.PythiaZtautau		
mc08.106022.PythiaWtaunu_1Lepton		
mc08.106005.PythiaDrellYanLowM_mu2p5mu2p5		
mc08.105200.T1_McAtNlo_Jimmy		
mc08.108405.PythiaB_bbmu15X (o simili)		
mc08.105013.j4_pythia_jetjet		
mc08.105014.j5_pythia_jetjet		
mc08.105015.J6_pythia_jetjet.recon.AOD.e344_s479_r541	390000 ca	InputCollections.py
mc08.105016.j7_pythia_jetjet		
mc08.095001.pythia_minbias.recon.AOD.e354_s462_d122_r517_tid025303	100000	InputCollections.py

link to castor files

Fast Simulation

processed dataset AOD	sample statistic	InputCollection
mc08.106051.PythiaZmumu_1Lepton.recon.AOD.e347_a68		

docs

- Starting point is EWPA twiki page + specific pages linked from there

main page
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EWPAMainPage

D3PD making/reading
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EWPADPDMakingReading
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EWPATutorialPersistency

HowTo make new tools
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EWPATutorialTool

W,Z measurements page
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EWAnalysisWZMeasurements