CUPS – CMS Upgrade School

Hands-on Session

DELPHES:

A modular framework for fast simulation of a generic collider experiment

Lisa Borgonovi

Outline

- Introduction to Delphes
  - What is Delphes and why do we use it?
  - Structure of the simulated detector
  - Object reconstruction
  - High-level corrections
  - Validation
  - Delphes for CMS
  - To sum up
- Analysis $H \rightarrow ZZ^* \rightarrow 4\mu$ with Delphes
- Hands-on exercise

Introduction to Delphes

Input: output of event generators (MadGraph, PYTHIA, HERWIG, ...)

DELPHES

C++ modular framework which performs a fast and parametrized multipurpose detector response simulation

- Tracking system embedded in a magnetic field
- Electromagnetic and hadronic calorimeters
- Muon identification system

Recostruction of the physics objects (γ, $e^{\pm}$, $\mu^{\pm}$, $\tau^{\pm}$, jets, b-jets, MET, ...)

Output: ROOT ntuple (generated and recontstructed objects)

Introduction to Delphes

What is Delphes and why do we use it?

Very high energy particle collisions produce a large variety of final states

Multipurpose detectors are complex and sophisticated

To tune and optimize specific analysis, a **full detector simulation** is requested, with a high level of accuracy:
GEANT package, dedicated routines, complex algorithms

Large scale computing resources and high level expertise

Introduction to Delphes

What is Delphes and why do we use it?

Phenomenological studies, comparison between different ugraded or aged scenarios, big background samples production require: less precision but a **FASTER** and **SIMPLER** tool

DELPHES

Limitations:

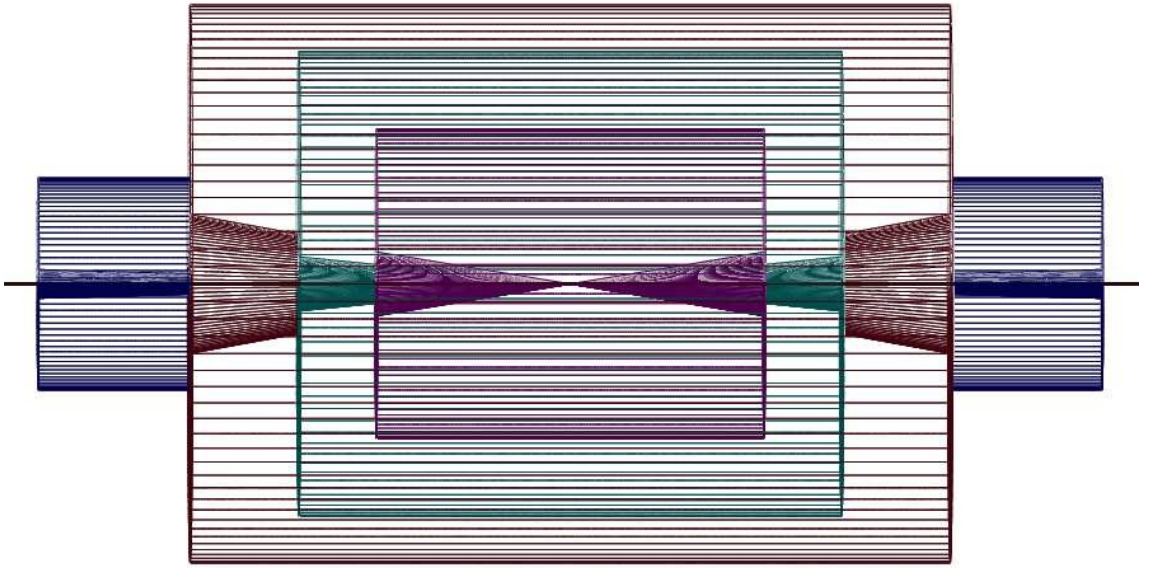
- Detector geometry is uniform, symmetric around the beam axis, without cracks or dead material
- Secondary interactions, multiple scattering, photon conversion and bremmstrahlung are not simulated

Introduction to Delphes

Structure of the simulated detector

The overall layout of a multipurpose detector is simulated thanks to a parametrization (in a DataCard*):

- Central tracking system (TRACKER)
- Electromagnetic and hadronic calorimeters (ECAL and HCAL)
- Muon system (MUON)
- Two forward calorimeters (FCAL)



* = I will talk about this in more detail later!

Introduction to Delphes

Structure of the simulated detector

- Central tracking system

Propagation of the long-lived particles in a uniform axial magnetic field along the beam direction

Neutral particles
Straight trajectory

Charged particles
Helicoidal trajectory

User-defined probability to be reconstructed as tracks

- Particles that originate outside the tracker volume are neglected
- NO smearing applied on track parameters (exept for p_T)

Introduction to Delphes

Structure of the simulated detector

- Electromagnetic and hadronic calorimeters

ECAL

Measures the energy deposit of electrons and photons

- $f_{ECAL}(e^{\pm}, \gamma) = 1$
- $f_{ECAL}(h) = 0$
- $f_{ECAL}(K, \Lambda) = 0.3$
- $f_{ECAL}(\mu^{\pm}, \nu) = 0$

HCAL

Measures the energy deposit of charged and neutral hadrons

- $f_{HCAL}(e^{\pm}, \gamma) = 0$
- $f_{HCAL}(h) = 1$
- $f_{HCAL}(K, \Lambda) = 0.7$
- $f_{HCAL}(\mu^{\pm}, \nu) = 0$

Default values, user can modify them

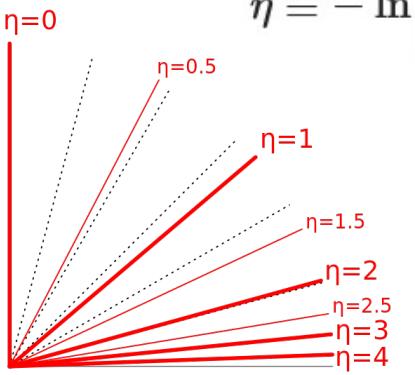
Introduction to Delphes

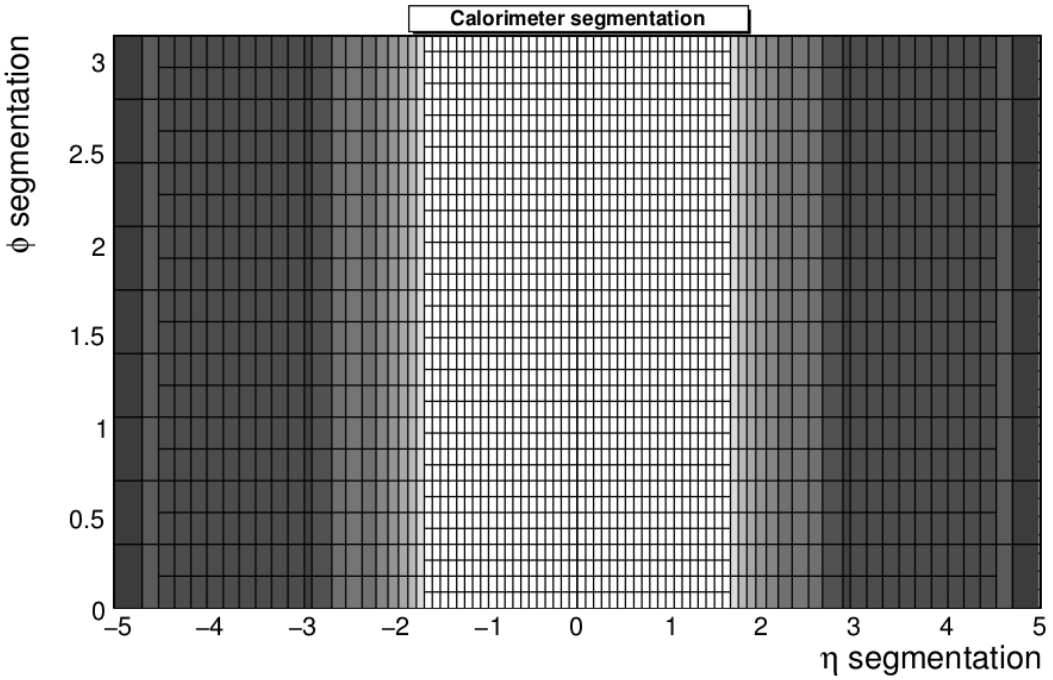
Structure of the simulated detector

- Electromagnetic and hadronic calorimeters

Segmented in the (η,φ) plane in user-defined size cells

$$\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$$





- Same granularity for ECAL and HCAL
- Uniform segmentation in φ
- Response simulated through a Gaussian smearing of the cell energy

Introduction to Delphes

Structure of the simulated detector

- Energy-Flow

Based on Tracker and Calorimeter information with the assumption that the momentum resolution of the Tracker is better than the energy resolution of the Calorimeter

The algorithm counts E_{ECAL} , E_{HCAL} , $E_{ECAL,trck}$, $E_{HCAL,trck}$ and produces:

- Energy-Flow tracks from reconstructed tracks: EF_E_{Track}
- If $EF\_E_{Tower} > 0$ ➡ $EF\_E_{Tower} = \max(0, \Delta_{ECAL}) + \max(0, \Delta_{HCAL})$

Where $\Delta_{ECAL} = E_{ECAL} - E_{ECAL,trck}$ and $\Delta_{HCAL} = E_{HCAL} - E_{HCAL,trck}$

Particle-flow track → Charged particles, good Resolution
Particle-flow tower → Charged+Neutral particles, lower Resolution

Introduction to Delphes

Object reconstruction

- $e^{\pm}$, $\mu^{\pm}$, γ ➡ quite easy to reconstruct
- Jets, Missing Transverse Energy ➡ difficult to reconstruct

Accessible in the output:

- 4-momentum and related quantities
- Additional specific properties (e.g. Isolation)

- MET $\quad \overrightarrow{E_T}^{miss} = -\sum_i \overrightarrow{p_T}(i)$

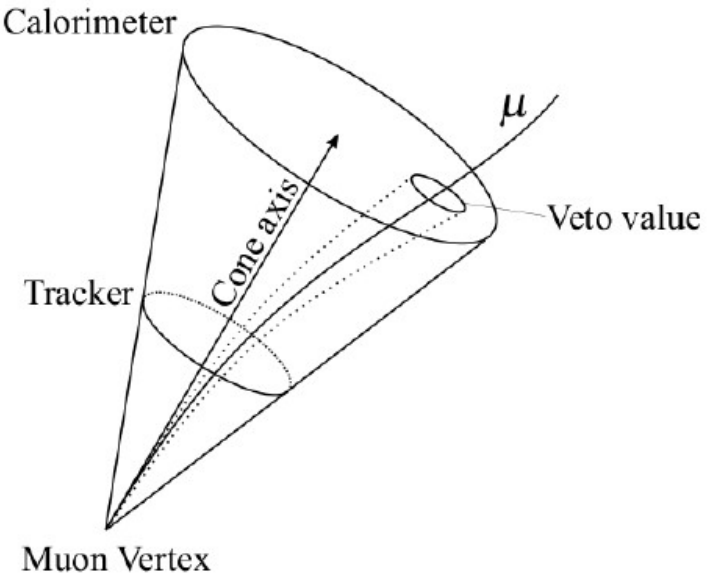
From different input collections: calorimeter, energy-flow, generator-level

Introduction to Delphes
Object reconstruction

- Photons and charged leptons (NO τ!)

Only the final-state particles identificated via generator-data and passing cuts (tracker acceptance and p_T threshold)

- Reconstructed with a user-defined Efficiency: f(η, p_T)
- **NO FAKE** candidates (they can be added in the analysis)



Isolation:

$e^{\pm}$, $\mu^{\pm}$ isolated if $I < I_{min}$ within a small cone of radius ΔR in the (η,φ) plane

Introduction to Delphes
Object reconstruction

- Jets

Final-states dominated by Jets: need an accurate reconstruction

It is possible to produce Jets starting from different input collections:

- Generated Jets: clustered from generator-level particles after parton-shower and hadronization
- Calorimeter Jets: use calorimeter towers
- Energy-flow Jets: result of clustering energy-flow tracks and towers

Many clustering algorithms (Longitudinally invariant k_t jet, Cambridge/Aachen jet, Anti k_t jet, CDF MidPoint, ...)

Introduction to Delphes

Object reconstruction

b-jets and τ-jets

Parametric approach:

A jet is a potential b/τ jet if a generated b/τ is found within some distance:

$$\Delta R = \sqrt{(\eta^{jet} - \eta^{b,\tau})^2 + (\phi^{jet} - \phi^{b,\tau})^2}$$

along the jet axis

User-defined efficiency to identify that candidate as a real b/τ jet intervenes (Mistagging efficiency can be specified)

Introduction to Delphes
High-level corrections

Resulting collections not yet ready for the analysis

- Jet Energy scale correction

Only for Jets (not for non-composite objects) → $f(\eta_{RECO\ jet}, p_{T\ RECO\ jet})$

Discrepancy between average momenta of reconstructed objects and generator-level objects

- Pile-Up Subtraction

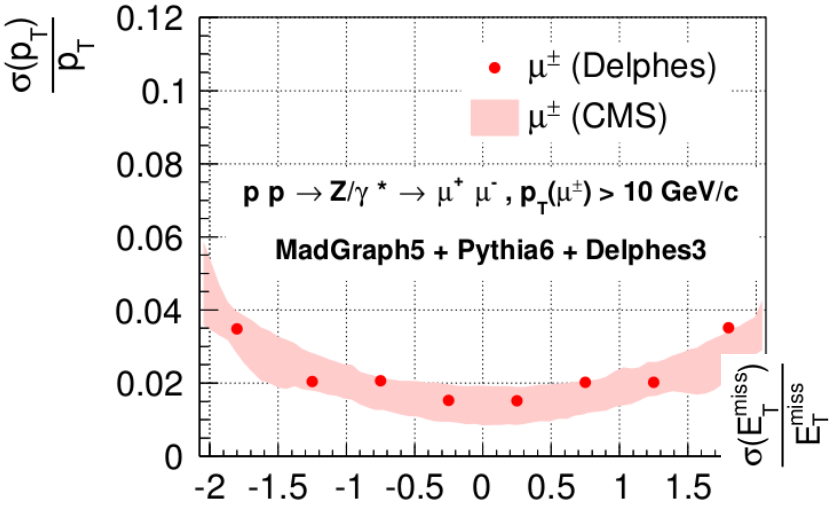
PU randomly estracted from a Minimum-Bias interactions file

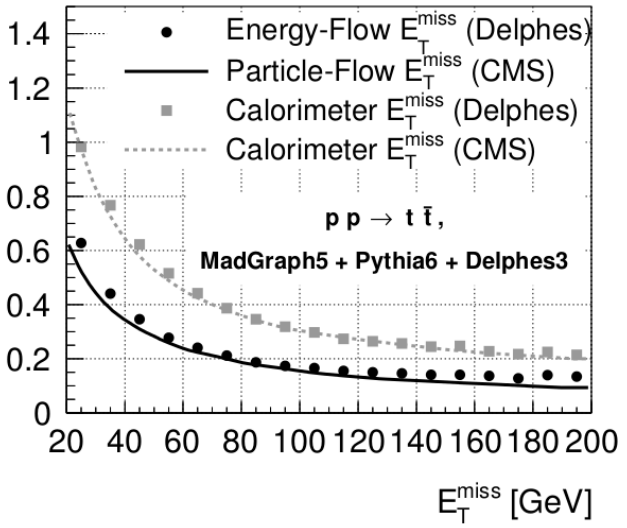
Only for Jets and Isolation using vertex reconstruction (not for MET because too difficult to account for neutral contamination)

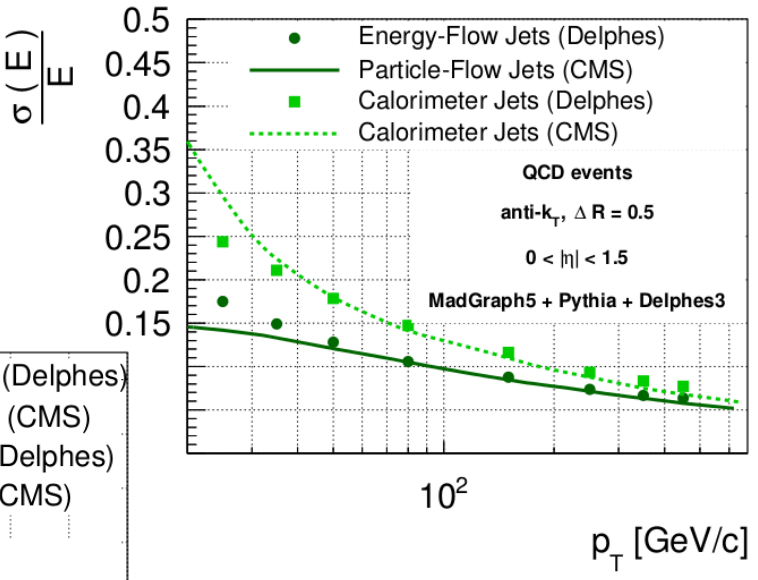
Introduction to Delphes

Validation

The simulation and reconstruction process executed by Delphes has to be validated with real experiment data (CMS and ATLAS)







Introduction to Delphes

Delphes for CMS

Many changes from the "official" Delphes:

- Timing information
- Different calorimeter configurations
- η dipendent ρ correction

In general: in CMS version there are more and different modules from the official one, to adjust the simulation to the CMS detector

This is the version we are going to use!

Reference: https://indico.cern.ch/event/294765/session/11/contribution/67

Introduction to Delphes

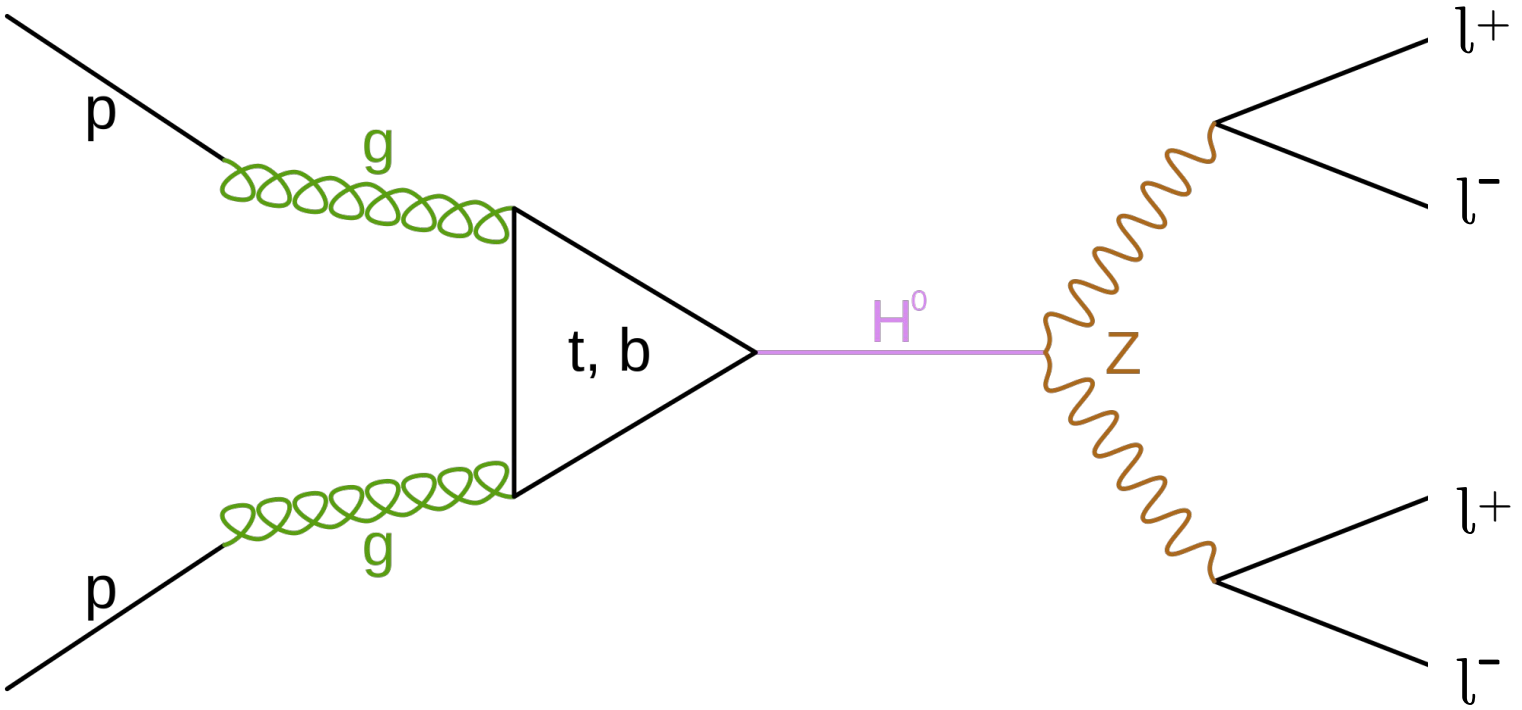
To sum up

Delphes is a modular framework that performs a fast and parametrized simulation of a multipurpose detector (e.g. CMS)

This tool has been completely validated (with CMS and ATLAS real data)

Useful in phenomenological studies and to produce big size bakground samples in which a fast simulation is needed

Analysis H → ZZ* → 4μ with Delphes



- Higgs production mechanism: gluon-gluon fusion
- Higgs decay mode: ZZ → 4l (4μ)

Analysis $H \rightarrow ZZ^* \rightarrow 4\mu$ with Delphes

Studies of the performances of the $H \rightarrow ZZ^* \rightarrow 4\mu$ analysis with the PhaseII detector upgrade

Two detector geometries:

PhaseI ("2017")

New tracker

FullSimulation samples:
Signal + Background

PhaseII ("2023")

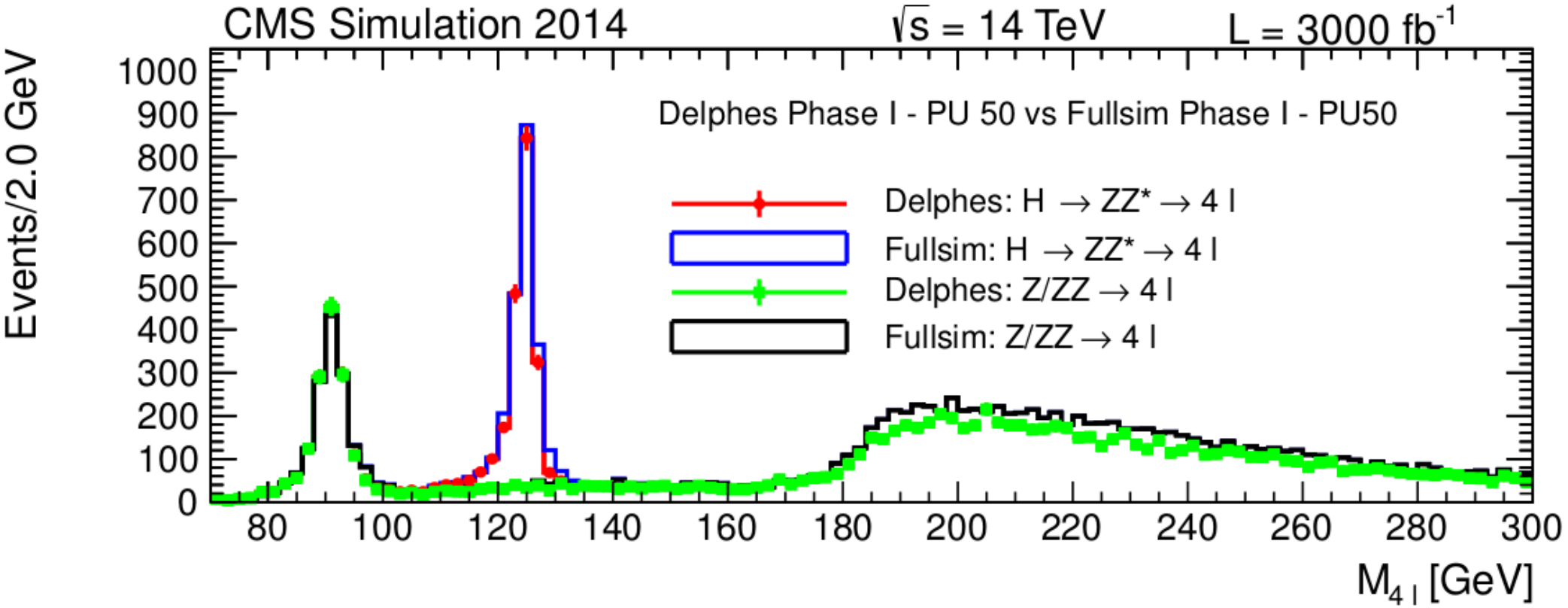
New tracker and forward ECAL
η up to 3 or 4

NO FullSimulation samples

DELPHES !

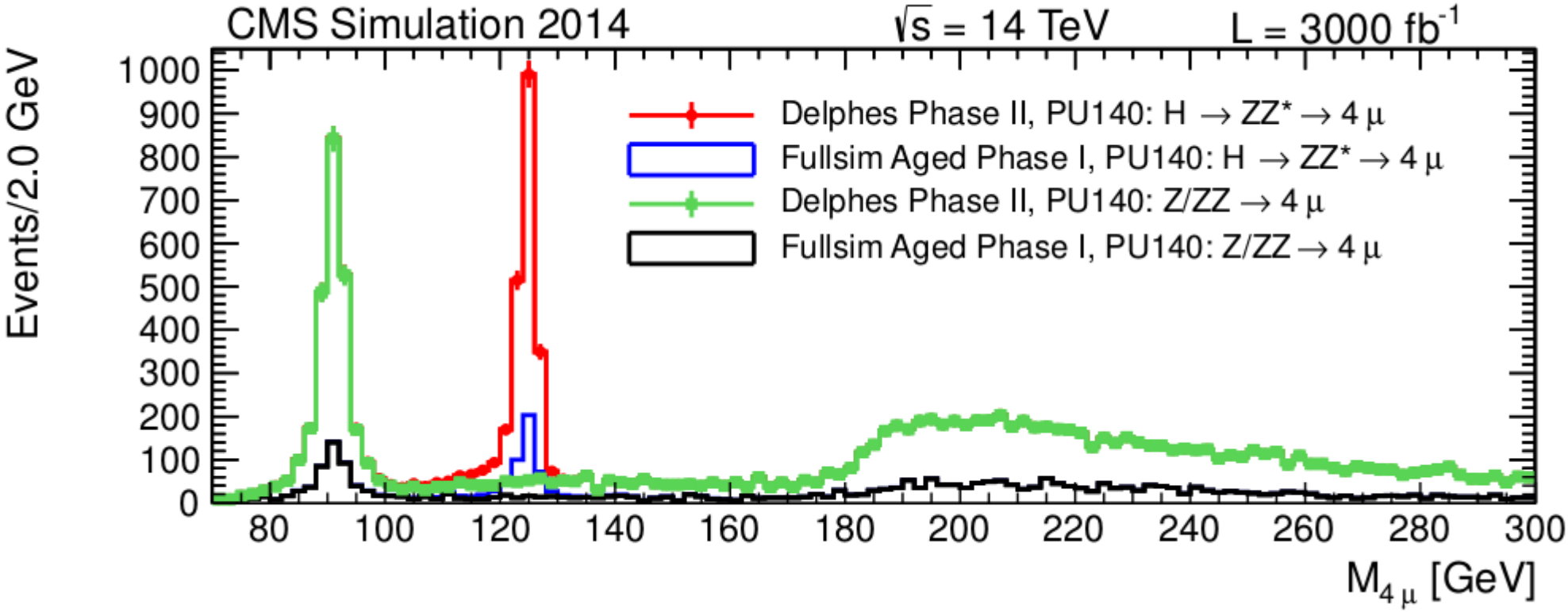
Analysis H → ZZ* → 4μ with Delphes

1st step: validate Delphes vs FullSim with PhaseI "2017" detector samples (√s = 14 TeV, PU =140)



Analysis H → ZZ* → 4μ with Delphes

2nd step: use the Delphes parametrized simulation to reconstruct a hypothetical PhaseII detector (√s= 14 TeV, PU =140)



Hands-on exercise

- Goals of this Hands-on Session

Basic knowledge about:

- the Delphes structure
- how Delphes works
- how to use Delphes to produce root samples of given events
- the ROOT samples structure
- how to analyze the samples with ROOT (for this part you will need a basic knowledge about ROOT).

Hands-on exercise

- Getting started with Delphes

1) Set the right Scientific Linux CERN version

2) Set the CMS environment

3) Download and install Delphes

4) You need one or more file(s) with Generated Events*
(in .hepMC format)

5) You need a file with Minimum Bias Events* for the PU
(MinBias.pileup)

6) Copy a customized Delphes DataCard* in your directory "Cards"

7) Copy all the macros* necessary for the analysis $H \rightarrow ZZ^* \rightarrow 4\mu$

Hands-on exercise

- Run Delphes!

  General structure of the command line:

./DelphesHepMC
Cards/cardname.tcl
somepath/outputfilename.root
somepath/hepmcfilename.hepmc

Hands-on exercise

File with Generated Events (.hepMC)

- File with all the characteristics of generated events: η, p_T, E, …
- Output of events generators

```
HepMC::Version 2.06.08
HepMC::IO_GenEvent-START_EVENT_LISTING
E 0 -1 4.6660200000000003e+01 1.4078099999999999e-01 7.7585830116479182e-03 9999 0 2024 1 2 0 1 1.0000000000000000e+00
N 1 "0"
U GEV MM
C 3.4466500000000003e+01 3.4466500000000003e+01
F -4 21 8.7436883742857147e-03 1.9855286214285714e-02 4.6660200000000003e+01 0 0 0 0
V -1 0 0 0 0 0 0 1 0
P 3 -4 0 0 6.1205818620000002e+01 6.1205818620000002e+01 0 21 0 0 -3 1 2 501
V -2 0 0 0 0 0 0 2 0
P 4 21 0 0 -1.3898700349999999e+02 1.3898700349999999e+02 0 21 0 0 -3 2 1 501 2 502
P 24 21 5.7185177904743236e+00 -6.1609436926723209e+00 -2.8704929448187880e+01 2.9910393637157195e+01 0 43 0 0 -23 2 1 507 2 501
V -3 0 0 0 0 0 0 2 0
P 5 25 4.0718400500000001e+01 -2.2785643319999998e+01 -7.5882461910000004e+01 1.5349401240000000e+02 1.2500036050000000e+02 22 0 0 -15 0
P 6 -4 -4.0718400500000001e+01 2.2785643319999998e+01 -1.8987229400000001e+00 4.6698809730000001e+01 7.3666614860000002e-07 23 0 0 -11 1 2 502
V -4 0 0 0 0 0 0 1 0
P 7 2 0 0 3.8552334991618829e+02 3.8552334991618818e+02 0 31 0 0 -6 1 1 503
V -5 0 0 0 0 0 0 2 0
P 8 21 0 0 -5.5383686130864440e-01 5.5383686130864385e-01 0 31 0 0 -6 2 1 504 2 505
```

Hands-on exercise

Delphes DataCard

File with ALL the settings of the simulation: efficiencies, resolutions, parameters for isolation, jets, …

Structure:

- Function with all the list of instructions (different modules called one after the other)
- Descriptions of all the modules (Muon Momentum Smearing, Muon efficiency, Electron Energy Resolution, Muon Isolation, …)

Last Module: TreeWriter
(choice of the Branches to write in the ROOT Tree)

Execution Path

```
#######################################
# Order of execution of various modules
#######################################

set ExecutionPath {

  PileUpMerger
  ModifyBeamSpot
  ParticlePropagator
  StatusPid
  GenBeamSpotFilter

  ChargedHadronTrackingEfficiency
  ElectronTrackingEfficiency
  MuonTrackingEfficiency

  ChargedHadronMomentumSmearing
  ElectronEnergySmearing
  MuonMomentumSmearing

  ModifyBeamSpotNoPU
  ParticlePropagatorNoPU
  ChargedHadronTrackingEfficiencyNoPU
  ElectronTrackingEfficiencyNoPU
  MuonTrackingEfficiencyNoPU
  ChargedHadronMomentumSmearingNoPU
  ElectronEnergySmearingNoPU
  MuonMomentumSmearingNoPU
  TrackMergerNoPU
  CalorimeterNoPU
  EFlowMergerNoPU
  FastJetFinderNoPU

  TrackMerger
  Calorimeter
  TrackPileUpSubtractor
  EFlowMerger

  GlobalRho
  Rho
  FastJetFinder
  GenJetFinder
  JetPileUpSubtractor

  NeutrinoFilter
  GenJetFinderNoNu
  GenMissingET

  EFlowChargedMerger
  RunPUPPI
  PuppiJetFinder
  PuppiRho
  PuppiJetPileUpSubtractor
  PuppiMissingET

  PhotonEfficiency
  PhotonIsolation

  ElectronEfficiency
  ElectronIsolation

  MuonEfficiency
  MuonIsolation

  MissingET

  BTagging
  BTaggingLoose
  TauTagging

  TrackPVSubtractor
  IsoTrackFilter

  UniqueObjectFinderGJ
  UniqueObjectFinderEJ
  UniqueObjectFinderMJ

  ScalarHT

  PileUpJetID

  PileUpJetIDMissingET

  ConstituentFilter
  TreeWriter
}
```

File Edit Options Buffers Tools Tcl Help

```
##############################
# Momentum resolution for muons
##############################

module MomentumSmearing MuonMomentumSmearing {
  set InputArray MuonTrackingEfficiency/muons
  set OutputArray muons

  # set ResolutionFormula {resolution formula as a function of eta and pt}

  # resolution formula for muons

    set ResolutionFormula {


(pt > 0.0) * ( abs(eta) > 0 && abs(eta) <= 0.1 ) * ( 0.00941022 ) + \
(pt > 0.0) * ( abs(eta) > 0.1 && abs(eta) <= 0.2 ) * ( 0.00958603 ) + \
(pt > 0.0) * ( abs(eta) > 0.2 && abs(eta) <= 0.3 ) * ( 0.0100961 ) + \
(pt > 0.0) * ( abs(eta) > 0.3 && abs(eta) <= 0.4 ) * ( 0.0108526 ) + \
(pt > 0.0) * ( abs(eta) > 0.4 && abs(eta) <= 0.5 ) * ( 0.0112063 ) + \
(pt > 0.0) * ( abs(eta) > 0.5 && abs(eta) <= 0.6 ) * ( 0.0114379 ) + \
(pt > 0.0) * ( abs(eta) > 0.6 && abs(eta) <= 0.7 ) * ( 0.0118331 ) + \
(pt > 0.0) * ( abs(eta) > 0.7 && abs(eta) <= 0.8 ) * ( 0.0124008 ) + \
(pt > 0.0) * ( abs(eta) > 0.8 && abs(eta) <= 0.9 ) * ( 0.0136308 ) + \
(pt > 0.0) * ( abs(eta) > 0.9 && abs(eta) <= 1 ) * ( 0.0159186 ) + \
(pt > 0.0) * ( abs(eta) > 1 && abs(eta) <= 1.1 ) * ( 0.0168638 ) + \
(pt > 0.0) * ( abs(eta) > 1.1 && abs(eta) <= 1.2 ) * ( 0.0177207 ) + \
(pt > 0.0) * ( abs(eta) > 1.2 && abs(eta) <= 1.3 ) * ( 0.0186257 ) + \
(pt > 0.0) * ( abs(eta) > 1.3 && abs(eta) <= 1.4 ) * ( 0.0192152 ) + \
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)


}
}
```

Resolution

```
##################################
# Energy resolution for electrons
##################################

module EnergySmearing ElectronEnergySmearing {
  set InputArray ElectronTrackingEfficiency/electrons
  set OutputArray electrons


    set ResolutionFormula {             (abs(eta) <= 2.5) * (energy > 0.1   && energy <= 2.5e1) * (energy*0.025) + \
                                        (abs(eta) <= 2.5) * (energy > 2.5e1)                     * (energy*0.035) + \
                                        (abs(eta) > 2.5 && abs(eta) <= 3.0)                      * (energy*0.035) + \
                                        (abs(eta) > 3.0 && abs(eta) <= 5.0)                      * (energy*0.07)}

}
```

Muon Efficiency

Muon Isolation

```
File  Edit  Options  Buffers  Tools  Tcl  Help
```

```
#################
# Muon efficiency
#################

module Efficiency MuonEfficiency {
  set InputArray TrackPileUpSubtractor/muons
  set OutputArray muons

  # set EfficiencyFormula {efficiency as a function of eta and pt}

  # efficiency formula for muons


    set EfficiencyFormula {

(pt < 5.0) * (0.00) + \
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 0.0 && abs(eta) <= 0.5 ) * ( 0.98
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 0.5 && abs(eta) <= 1.0 ) * ( 0.98
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 1.0 && abs(eta) <= 1.5 ) * ( 0.97
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 1.5 && abs(eta) <= 2.0 ) * ( 0.99
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 2 && abs(eta) <= 2.4 ) * ( 0.9714
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 2.4  ) * ( 0 ) + \
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 0.0 && abs(eta) <= 0.5 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 0.5 && abs(eta) <= 0.9 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 0.9 && abs(eta) <= 1.1 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 1.1 && abs(eta) <= 1.4 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 1.4 && abs(eta) <= 2.0 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 2.0 && abs(eta) <= 2.4 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 2.4 ) * ( 0 ) + \
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 0.0 && abs(eta) <= 0.5 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 0.5 && abs(eta) <= 1.0 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 1.0 && abs(eta) <= 1.5 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 1.5 && abs(eta) <= 2.0 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 2.0 && abs(eta) <= 2.4 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 2.4 ) * ( 0) + \
(pt > 40.0 ) * ( abs(eta) > 0.0 && abs(eta) <= 0.5 ) * ( 0.9949498) + \
(pt > 40.0 ) * ( abs(eta) > 0.5 && abs(eta) <= 1.0 ) * ( 0.9952686) + \
(pt > 40.0 ) * ( abs(eta) > 1.0 && abs(eta) <= 1.5 ) * ( 0.9886918) + \
(pt > 40.0 ) * ( abs(eta) > 1.5 && abs(eta) <= 2.0 ) * ( 0.9919946) + \
(pt > 40.0 ) * ( abs(eta) > 2.0 && abs(eta) <= 2.4 ) * ( 0.964623) + \
(pt > 40.0) * ( abs(eta) > 2.4 ) * ( 0)
```

```
#################
# Muon isolation
#################


module Isolation MuonIsolation {
  set CandidateInputArray MuonEfficiency/muons
  set NeutralIsolationInputArray Calorimeter/eflowTowers
  set ChargedIsolationInputArray TrackPileUpSubtractor/eflowTrac
  set RhoInputArray Rho/rho

  set OutputArray muons

  set DeltaRMax 0.4

  set PTMin 0.4

  set PTRatioMax 9999.
}

###################
```

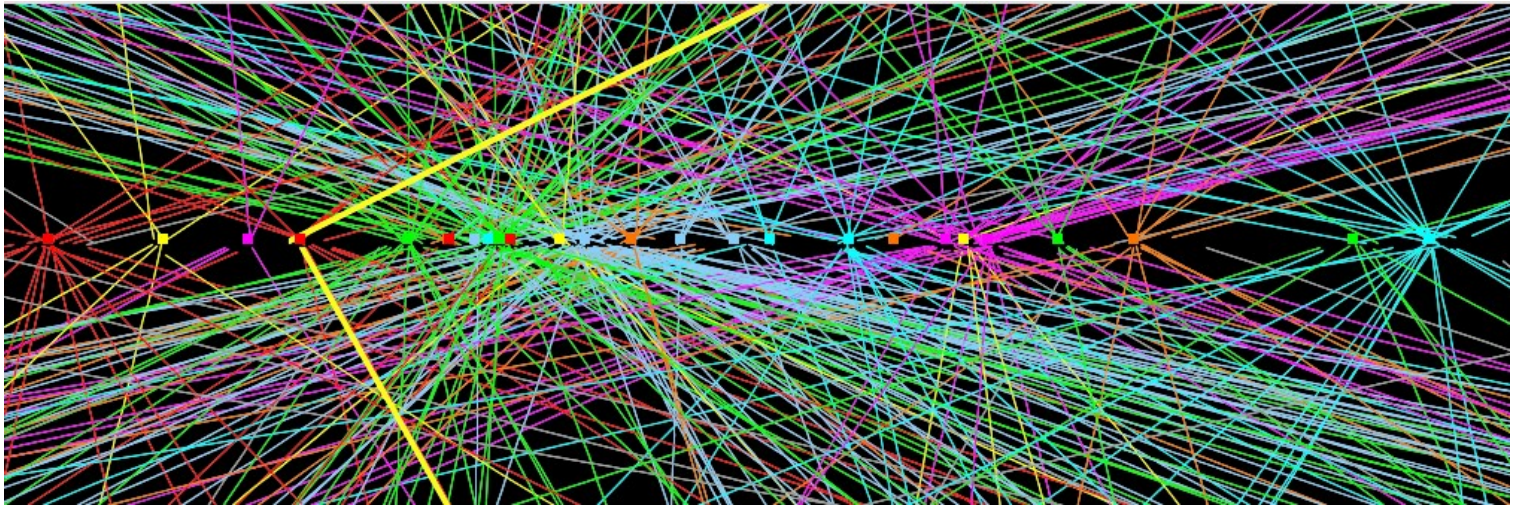
```
}

##################
# ROOT tree writer
##################

module TreeWriter TreeWriter {
 add Branch StatusPid/filteredParticles Particle GenParticle
 ### add Branch Delphes/allParticles Particle GenParticle



  add Branch GenBeamSpotFilter/beamSpotParticles BeamSpotParticle GenParticle

  add Branch FastJetFinder/jets RawJet Jet
  add Branch FastJetFinderNoPU/jets RawJetNoPU Jet

  add Branch GenJetFinder/jets GenJetWithNu Jet
  add Branch GenJetFinderNoNu/jets GenJet Jet
#  add Branch UniqueObjectFinderMJ/jets Jet Jet
#  add Branch UniqueObjectFinderEJ/electrons Electron Electron
#  add Branch UniqueObjectFinderGJ/photons Photon Photon
#  add Branch UniqueObjectFinderMJ/muons Muon Muon
  add Branch JetPileUpSubtractor/jets Jet Jet
  add Branch ElectronIsolation/electrons Electron Electron
  add Branch PhotonIsolation/photons Photon Photon
  add Branch MuonIsolation/muons Muon Muon

  add Branch PileUpJetIDMissingET/momentum PileUpJetIDMissingET MissingET
  add Branch GenMissingET/momentum GenMissingET MissingET
  add Branch PuppiMissingET/momentum PuppiMissingET MissingET


  add Branch MissingET/momentum MissingET MissingET
  add Branch ScalarHT/energy ScalarHT ScalarHT
  add Branch Rho/rho Rho Rho
  add Branch GlobalRho/rho GlobalRho Rho
  add Branch PileUpMerger/NPU NPU ScalarHT
  add Branch IsoTrackFilter/IsoTrack IsoTrack IsoTrack

  add Branch PuppiJetFinder/jets PuppiJet Jet

  set OffsetFromModifyBeamSpot 0

#  add Branch RunPUPPI/weightedparticles PuppiWeightedParticles GenParticle
#  add Branch Delphes/allParticles Particle GenParticle
--:**  CMS_PhaseI_PU50_muPOGeff.tcl      (Tcl)--L1346--95%-----------------------------------
```

Root Tree Writer

Hands-on exercise

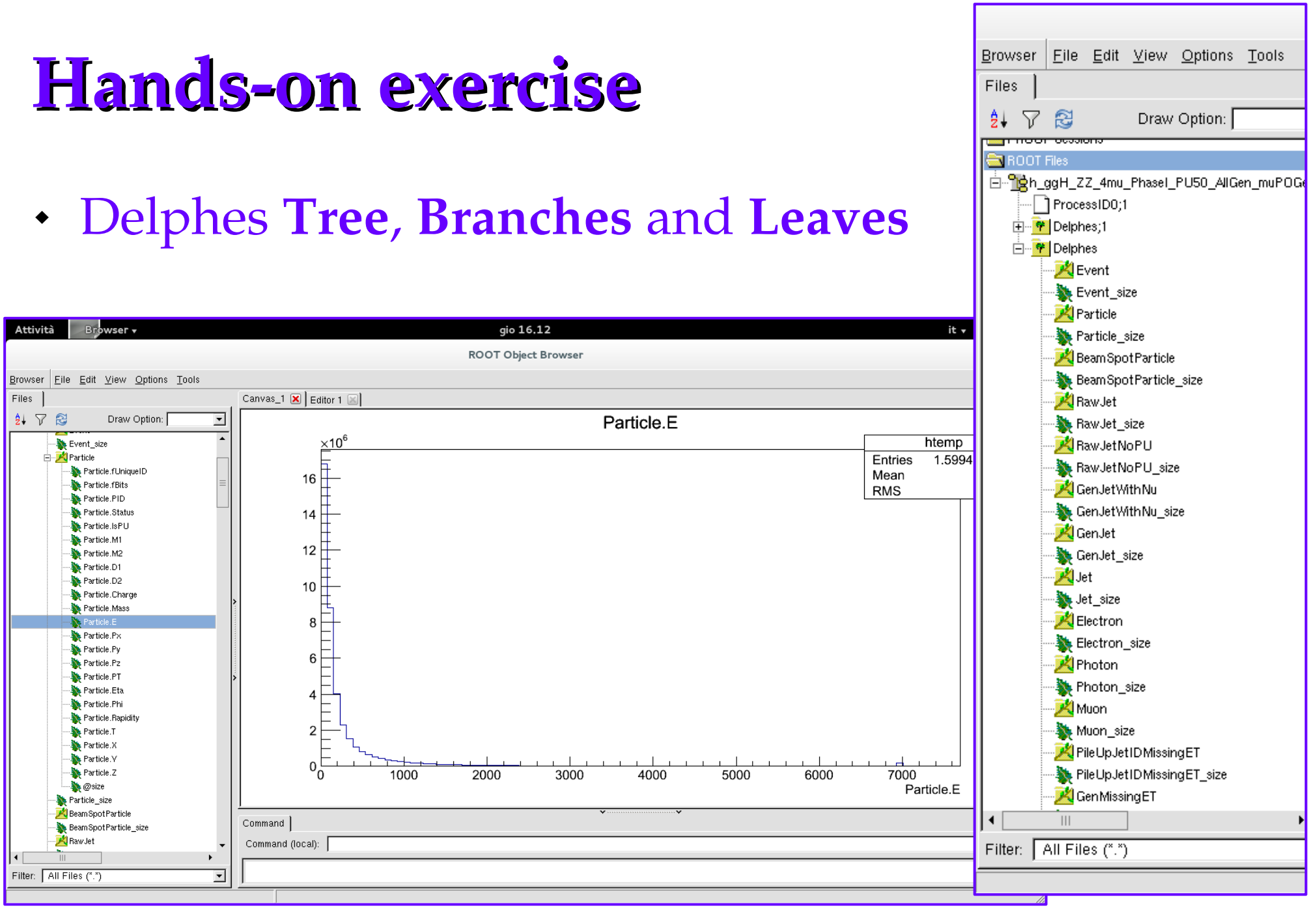
Minimum Bias Events file

- File with a collection of PU events
- Delphes accesses this file randomly for every signal event to take the selected number of PU events and superimpose them to the main one

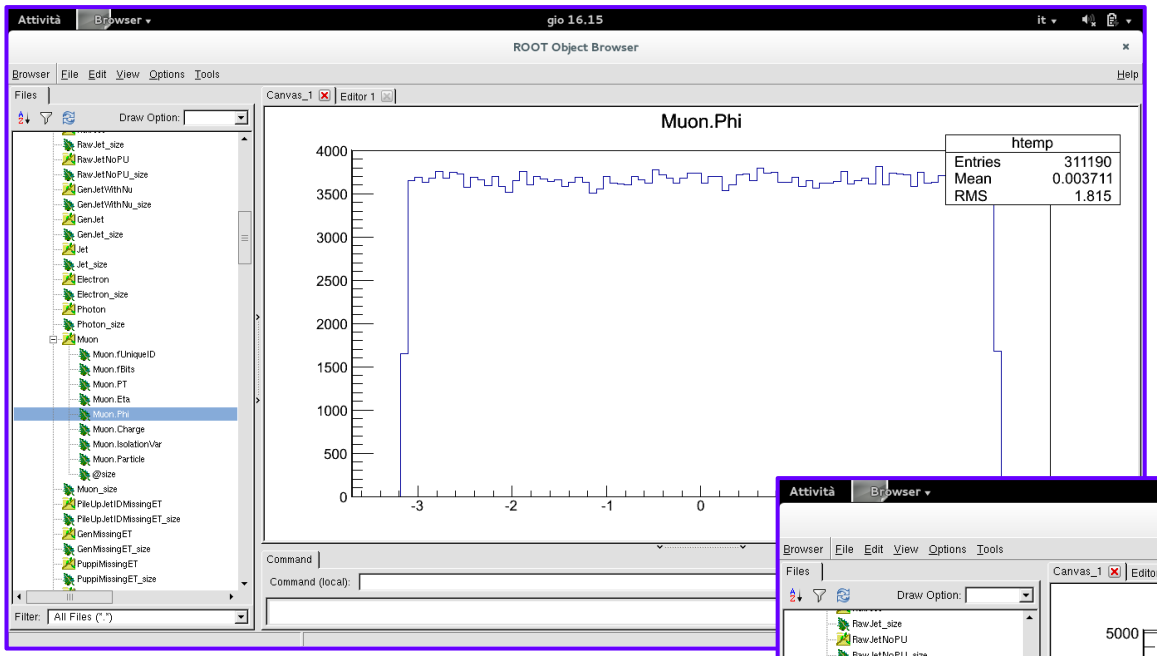


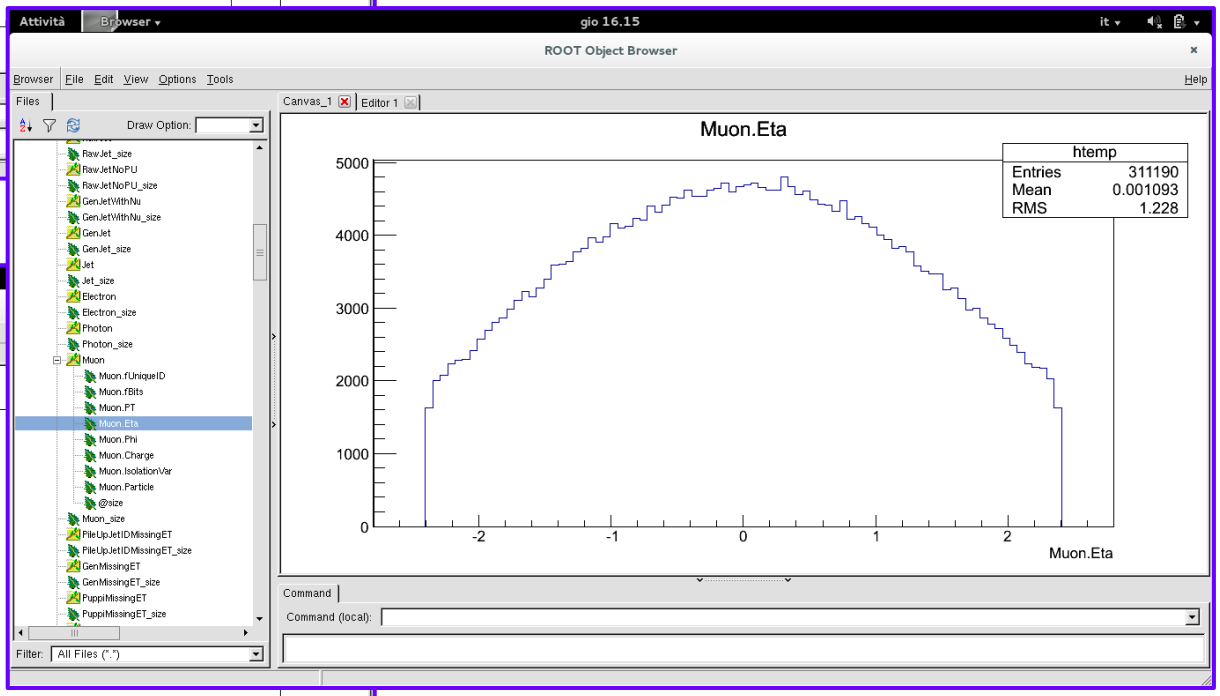
Hands-on exercise

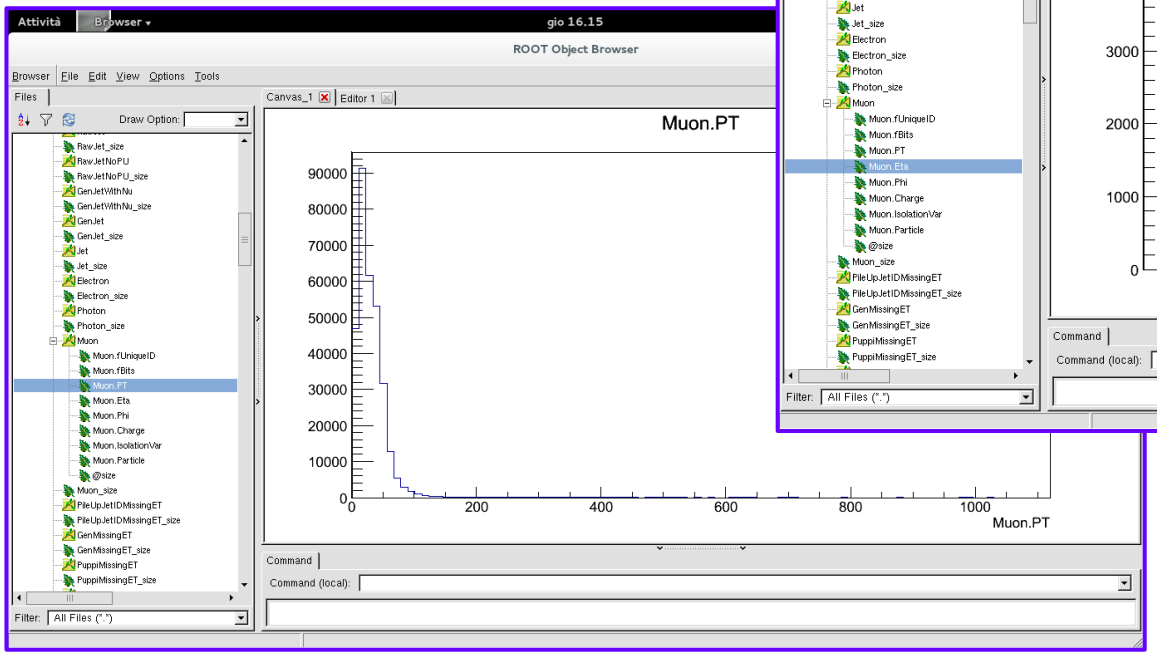
- Delphes **Tree**, **Branches** and **Leaves**

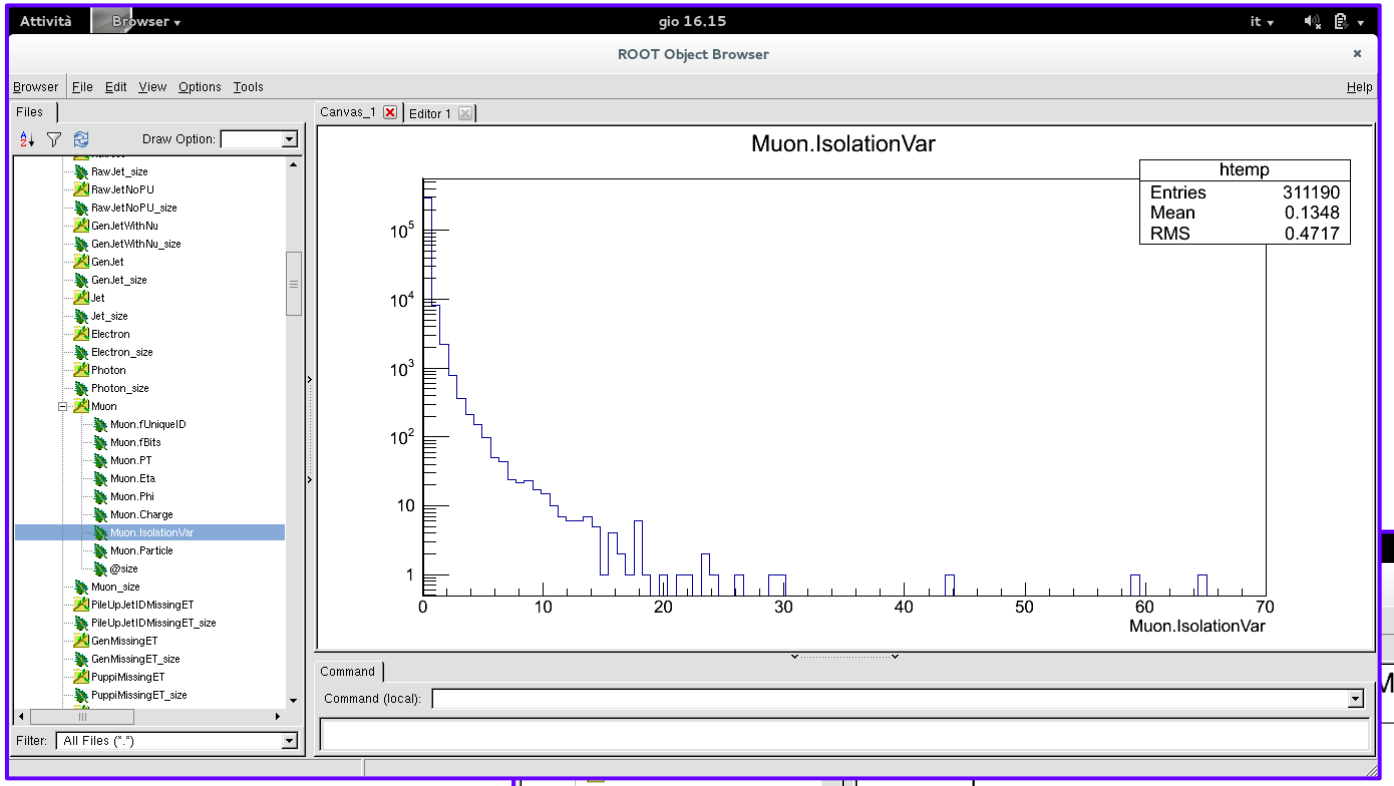


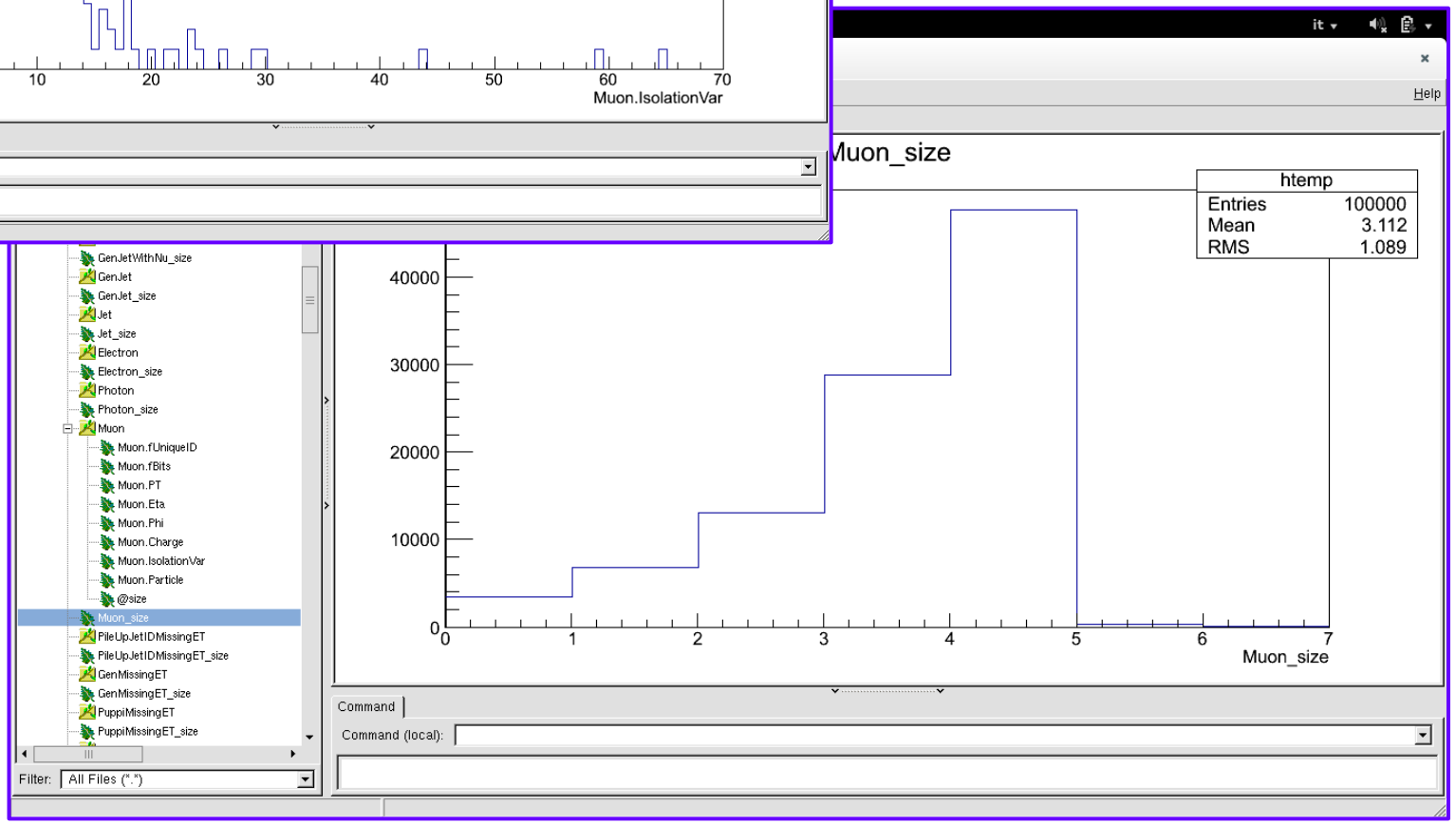
Branch Muon


Muon.Phi
htemp
Entries 311190
Mean 0.003711
RMS 1.815


Muon.Eta
htemp
Entries 311190
Mean 0.001093
RMS 1.228
Muon.Eta

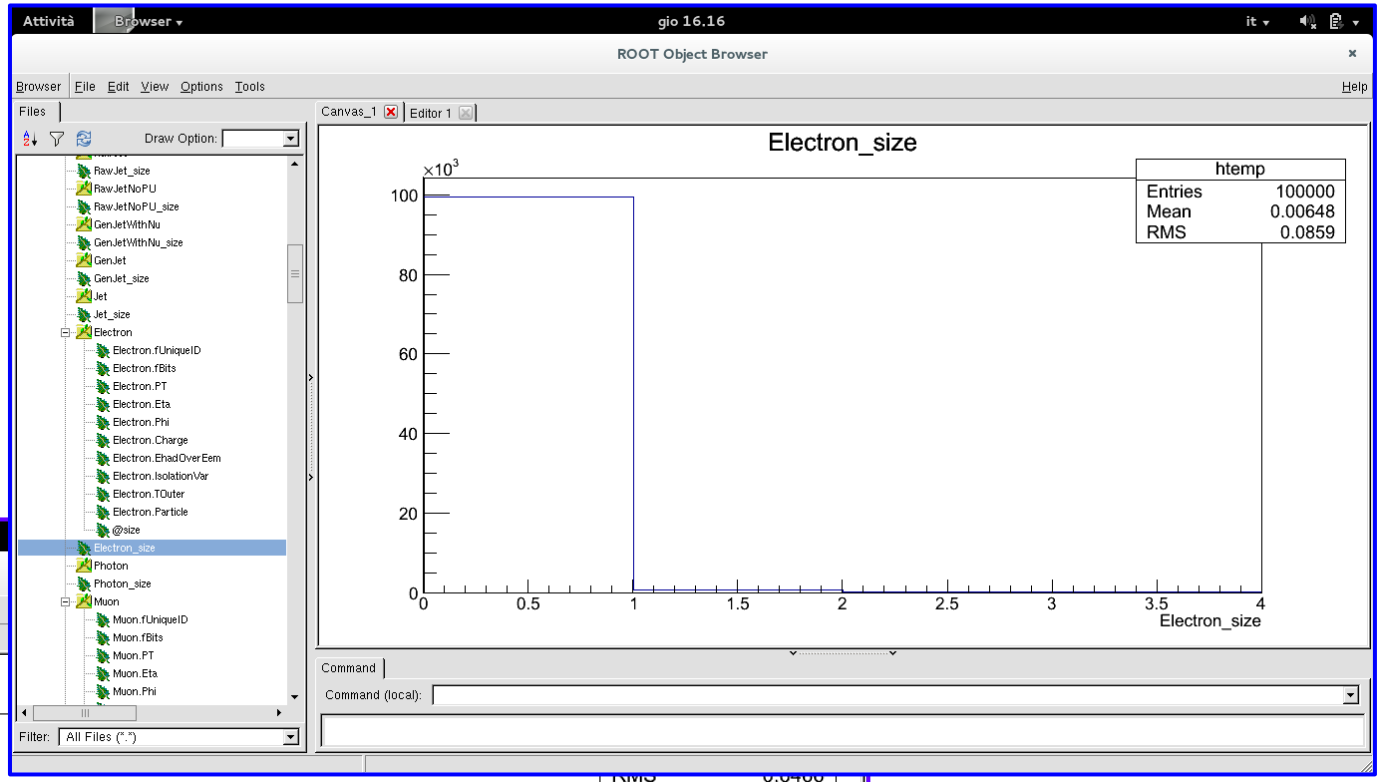

Muon.PT
Muon.PT

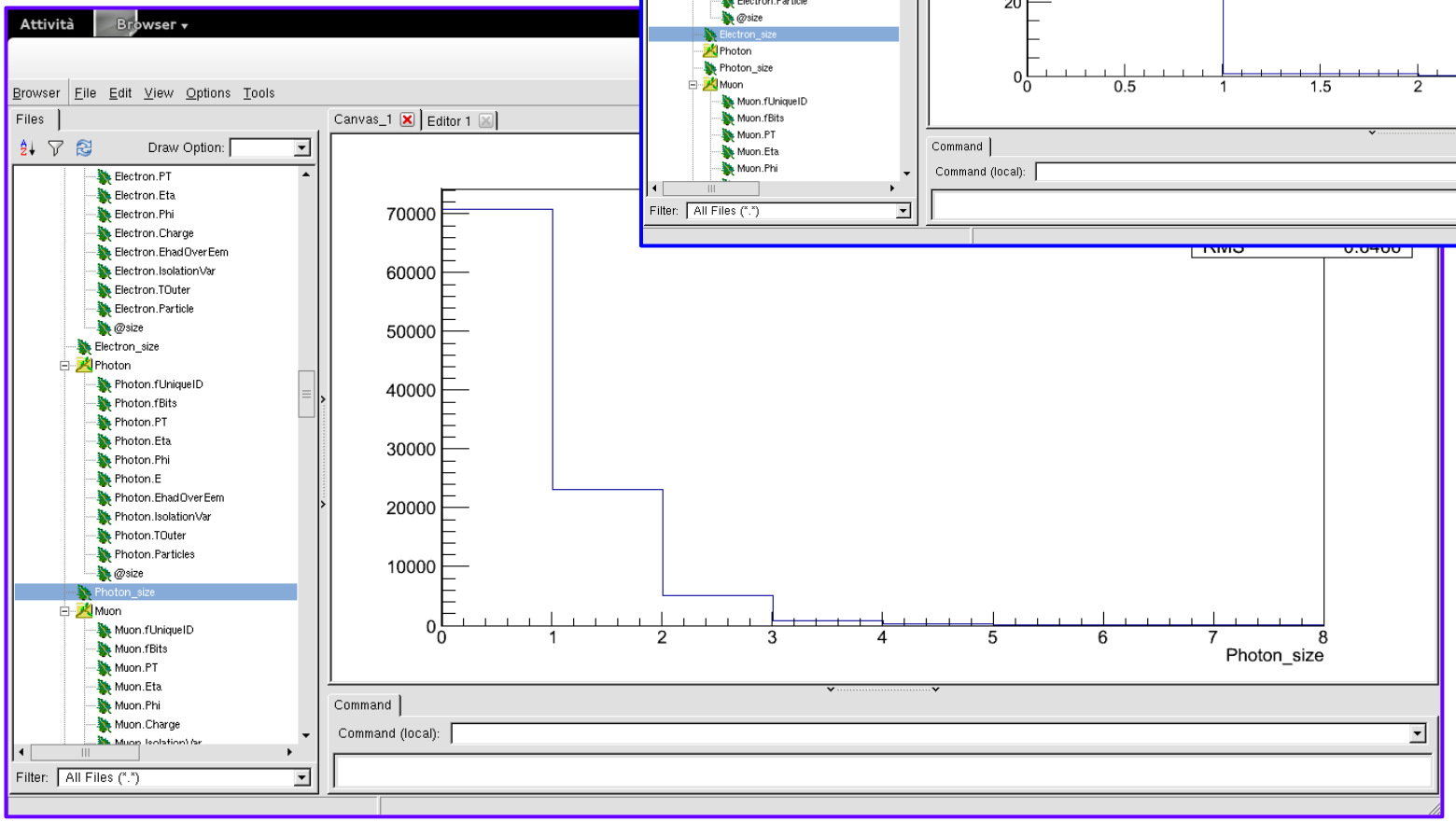

Attività
Browser
gio 16.15
it
ROOT Object Browser
Browser File Edit View Options Tools Help
Files
Draw Option:
Canvas_1
Editor 1
Muon.IsolationVar
htemp
Entries 311190
Mean 0.1348
RMS 0.4717
Muon.IsolationVar
Command
Command (local):
Filter: All Files (*.*)
RawJet_size
RawJetNoPU
RawJetNoPU_size
GenJetWithNu
GenJetWithNu_size
GenJet
GenJet_size
Jet
Jet_size
Electron
Electron_size
Photon
Photon_size
Muon
Muon.fUniqueID
Muon.fBits
Muon.PT
Muon.Eta
Muon.Phi
Muon.Charge
Muon.IsolationVar
Muon.Particle
@size
Muon_size
PileUpJetIDMissingET
PileUpJetIDMissingET_size
GenMissingET
GenMissingET_size
PuppiMissingET
PuppiMissingET_size


it
Help
Muon_size
htemp
Entries 100000
Mean 3.112
RMS 1.089
Muon_size
Command
Command (local):
Filter: All Files (*.*)
GenJetWithNu_size
GenJet
GenJet_size
Jet
Jet_size
Electron
Electron_size
Photon
Photon_size
Muon
Muon.fUniqueID
Muon.fBits
Muon.PT
Muon.Eta
Muon.Phi
Muon.Charge
Muon.IsolationVar
Muon.Particle
@size
Muon_size
PileUpJetIDMissingET
PileUpJetIDMissingET_size
GenMissingET
GenMissingET_size
PuppiMissingET
PuppiMissingET_size

Branch Muon

Branch Electron





Branch Photon

Hands-on exercise

Macros for the analysis

- Simplified anlaysis $H \rightarrow ZZ^* \rightarrow 4\mu$
- You will learn how to:
  a) read a Delphes Tree
  b) access Generated Particles
  c) access Recontructed Muon
  d) build an efficiency/resolution plot with "Delphes quantities"
  e) build an invariant mass plot and a cutflow table

Hands-on exercise

- `DelphesMU_analysis.C`

1) Bool variables to choose configuration (REMEMBER TO CHEK ON THEM!)
2) Opening of the output file
3) Reading of the Delphes Tree
4) Declarations of histograms (Higgs invariant mass and Cutflow table)
5) Loop on every event: sequential cuts to identify events $H \rightarrow ZZ^* \rightarrow 4\mu$ and filling of histrograms
6) Writing of the .histo file

Hands-on exercise

- `DelphesMU_distributions.C`

1) Bool variables to choose configuration (REMEMBER TO CHEK ON THEM!)
2) Opening of the output file
3) Reading of the Delphes Tree
4) Declarations of histograms (Efficiency as function of η and of p_T, Isolation, p_T Resolution)
5) Loop on every event: filling of histrograms
6) Writing of the .histo file

Hands-on exercise

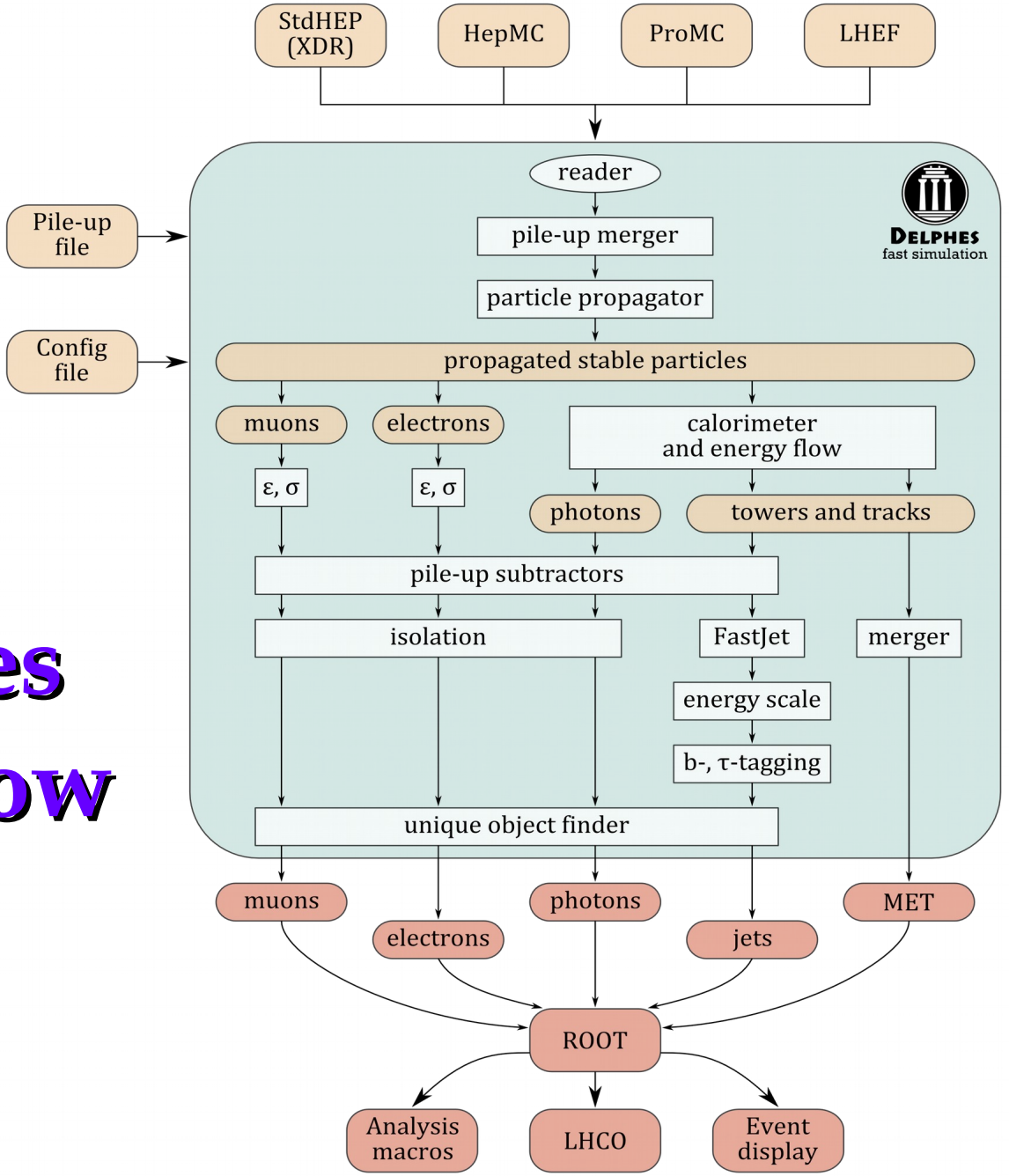
- `Plots.C`

1) Opening of all .histo files (REMEMBER TO CHECK ON THEM!)
2) Reading of the leaves of the .histo
3) Definition of the histograms options
4) Definition of the legends
5) Drawing of the histograms
6) Saving of all the plots

Thank you for the attention!

Backup

Delphes work-flow chart



• Electromagnetic and hadronic calorimeters

Calorimeter response is simulated through a Gaussian smearing of the cell energy with resolution of ECAL and HCAL independently parametrized:

$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{S(\eta)}{\sqrt{E}}\right)^2 + \left(\frac{N(\eta)}{E}\right)^2 + C(\eta)^2$$

Where S: stochastic, N: noise, C constant terms

Final energy:

$$E_{Tower} = \sum_{particles} \ln\mathcal{N}\left(f_{ECAL}\cdot E, \sigma_{ECAL}(E,\eta)\right) + \ln\mathcal{N}\left(f_{HCAL}\cdot E, \sigma_{HCAL}(E,\eta)\right)$$

Where $\ln\mathcal{N}(m,s)$ is the log-normal distribution with mean m and variance s

Too precise ➡ unique Gaussian smearing!

- Photon and charged leptons (NO !)

$\mu^{\pm}$ Reconstructed thanks to user-defined efficiencies: $f(\eta, p_T)$

p_T: Gaussian smearing of the initial 4-momentum vector with resolution as a function of eta and p_T

$e^{\pm}$ Reconstructed with user-defined efficiencies (ECAL + tracker recontruction): $f(\eta, p_T)$

Energy Resolution: f(TRACKER Res, ECAL Res)

γ Reconstructed only by the ECAL energy deposit

Pairs production neglected

$e^{\pm}$ with no reconstructed tracks that reach the ECAL