A. Sidoti
INFN - Bologna

July 2019-07-24

ROOT Tutorial HASCO 2019

Table of Contents

Introduction

This a set of instruction for the HASCO 2019 Root tutorial; they cover both the Thursday afternoon session and the Friday morning hands on.

Root Tutorial (Thursday afternoon)

During the tutorial on Thursday afternoon, we will go trough some slides to present few important ROOT key aspects. During the session, I will show and run some code examples. Code will NOT run on my laptop (except few things) but it will run on the swan notebook at CERN. For those who have an account at CERN I will share with you the notebooks (write your user account on mentimeter during the first slide). You can go though them during the tutorial.

There are two notebbooks:

1) dedicated to CINT/CLING (C++) https://cernbox.cern.ch/index.php/s/QAKJP1PaCrqRGjG

2) dedicated to PyROOT (Python 2.7) https://cernbox.cern.ch/index.php/s/YPuGLHyXlrg5K9Z

Password for both is hasco2019 and you can access them until 31st July 2019

People without CERN account can access the files through the indico agenda.

Root Hands On (Friday Morning)

The goal is to go through a single exercize on your PC. It can be useful to revise the jupyter notebooks shown on previous session. The jupyter notebook installation in Goettingen seems to prefer the CINT notebook, so don’t worry if the pyroot notebook doesn’t work as expected.

Together with the notebooks, the ROOT files you need to perform the exercise have already been downloaded to the PC you will work on. They are part of the ATLAS Open data project. Cf here for more information http://cds.cern.ch/record/2203649/files/ATL-OREACH-PUB-2016-001.pdf

Important: Please create single user work directories since you have to share the same account

Datasets to download

They have been already downloaded (Thanks Tomas!)

MonteCarlo (Simulation)	
Z → **μμ**	http://opendata.atlas.cern/release/samples/MC/mc_147771.Zmumu.root
Drell-Yan 8<M(**μμ**)<15 GeV	http://opendata.atlas.cern/release/samples/MC/mc_173043.DYmumuM08to15.root
Drell-Yan 15<M(**μμ**<40 GeV	http://opendata.atlas.cern/release/samples/MC/mc_173044.DYmumuM15to40.root
Zprime 400 GeV	http://opendata.atlas.cern/release/samples/MC/mc_110899.ZPrime400.root
Data	
Muon Stream	http://opendata.atlas.cern/release/samples/Data/DataMuons.root

Instruction for Hands On

You are free to execute the Hands On in C++ (access the Ttree variables, do a MakeClass or a MakeSelector) or in Python (if you want you can do both).
You should follow the order

Suggestion: Implement Steps 1, 2, 3 and 6with CINT/CLINT/Aclic

Steps 4 and 5 with PyRoot

Step 0

Download from Thursday agenda the tutorial_nicegraphics.tgz and undat it in your working directory.

Step 1

It can take a long time to run on all the events. So the first thing you will do is to reduce the size of the data and Z→mumu sample to 10k Events each.

You can just pick the first 10k events, or you can sample randomly the files such to have ~approx 10k events. (Hint: Look for "Create a subsample of event file" in notebooks)

Step 2

Implement a selection of events:

Global events should have:

- fired the muon trigger trigM = true
- passGRL=true (detector in good conditions)
- hasGoodVertex=true (primary vertex of the interaction should have been reconstructed correctly)
- with exactly 2 muons of good quality (defined below) , oppiste sign (lep_charge) and at least one of them shoud have been trigger matched lep_trigMatched>=1
- Good Muons should have:
  - lep_type=13
  - lep_pt>25 GeV (remind units of energies in the ntuple are MeV!)
  - isolated. Differently from leptons coming from mesons or baryons decay, leptons from ewk bosons are "isolated" (i.e. there are no other particles around except ????). The request is on the "relative" isolation i.e. the sum of the energy/momentum around a cone of radius 0.3 (in $\Delta R=\sqrt{\Delta\phi^2+\Delta\eta^2}$) divided by the momentum of the lepton. In the ntuple you have absolute values of isolation lep_ptcone30 and lep_etconel30. For isolated leptons, the relative isolation should be <0.15

Hint: Look at "Plot Non Standard Variable"

Hint: Use MakeClass, MakeSelector

Hint: Define and write a bool isGoodMuon(int index), bool isGoodEvent(), int NumGoodMuons(int &ilep1, int &ilep2) methods (if you are using C++ or equivalent in python)

Step 3

Using the selection defined above, calculate and plot the dimuon invariant mass for data in muon stream (Hint: “Plot non standard variable in the notebook”. Hint2: Use TLorentzVector)

When running on MC simulated datasets each event should be weighted by scale factors which represent how well the simulation agrees with real data. A scale factor of 1 means that you have perfect agreement between Data and MC.

The weight is an event variable: should be the product of:

scaleFactor_PILEUP x scaleFactor_MUON x scaleFactor_TRIGGER x scaleFactor_ZVERTEX

Since it is MC you should apply (multiply) the mcWeight as well (may be you have seen that at the QCD Lecture, don’t worry if mcWeight is negative, ask QCD lecturer why ….). But there are things I don’t fully understand for this sample so don’t ue it.

To understand if you are not applying odd weights, plot also the histograms of the weights you are applying.

Save the plot in an output file (Hint: Saving histograms in output file). Do the same for all the MC datasets.

Remember, develop and give a try on the filtered ntuple to save time, than, when you are sure your code is not buggy, launch once on the full dataset!

Step 4

Retrieve the histograms from output files, normalize the ones from data and from Z→mumu simulated sample to unity and plot them superimposed. Divide your canvas in two and plot on the bottom pad the ratio between data and MC. (Hint: use TH1::SetNormFactor(1) or TH1::Scale(1/histo→Integral())).

Before normalizing use the method TH1::Sumw2() (This is to get statistic uncertainties calculated correctly for the histograms. I am not sure this is still needed now for the root version you are using, but it doesn’t harm applying it)

Step 5

Now normalize the MC simulated (not only Z→mumu, but also Drell-Yan) to the data integrated luminosity.

Data integrated luminosity is reported here http://cds.cern.ch/record/2203649/files/ATL-OREACH-PUB-2016-001.pdf tab.4 p. 47 (you can assume it is 1fb^{-1})

MC simulated sample integrated luminosity is reported on 6^{th} column of table 5 and 6 of the above doument (p. 48,49)

Draw the data histogram superimposed with the different contributions from MC (Hint: Use THStack)

Compare Data with MC plotting the ratio (Data/Total MC)

Fit the distribution (the dimuon invariant mass) you get with a gaussian

Fit the same distribution with a Breit Wigner

Comment

Exercise 6

With the Tag & Probe method measure from data the efficiency as a function of Pt f the probe muon

Tag&Probe is a method to measure efficiencies with data. The strategy is the following:

1) Select a physical process where there is very little background background contamination

2) This process should be selected using only one object in the final state (the **Tag**) and minimum bias on the other object (the **Probe**). For example, the trigger request should be done only on the Tag object etc.

3) This defines your denominator (you might want to subtract background here event if it is small)

4) Implement the selection on your probe and measure how many objects pass your selection (this is your numerator)

5) Divide numerator / denominator to get your efficiency (note it is a binomial so statistical uncertainties are correlated etc… → Use TEfficiency as shown in CINT notebook “Graphs”)

To be more specific in the example:

1) The process we want to select is $Z \rightarrow \mu\mu$

2) The request on the tag is: to be a muon, pt>25, isolation, trigger match.

3) The request on the probe is minimal : to be a muon, and the invariant mass of the dimuon system is such that it should be within 5 GeV of the Z mass peak

4) probe selection is on the isolation

5) At the end, what the Tefficiency plot represents is the efficiency of the isolation selection as a function of momentum