Universidade do Estado do Rio de Janeiro - UERJ

Project:

Tag and probe for CMS Open Data

Raphael Gomes de Souza

Advisor:
Doc. Kati Lassila-Perini

Contents

1 Introduction

This is a report about my stay in CERN as a summer student in the 2019 Summer Student Programme. The project that I was working ON is about Tag and Probe methods using datasets of CMS Open Data. The main goal of this project is to make a simple example code applying the Tag and Probe methods, since the official **TagAndProbe** tool is complex. In this project, we did a more easy-to-understand code for CMS Open Data users.

2 About the Tag and Probe

The Tag and Probe method consists in calculating the efficiency of the data itself without THE dependency on the simulations, and with a small or easily subtracted, fake rate [2]. For that, this method uses known resonances (e.g J/ψ, Z, Υ) to select particle of a desired type and probe the efficiency of this particle in a determined selection criteria.

2.1 Tags and Probes objects

In the **Tag and Probe** method, one needs two objects:

- **Tag** - It is an object that passes a set of very tight selection criteria designed to isolate the required particle type (usually an electron or muon). And the fake rate for passing tag selection criteria should be very small ($< 1\%$).
- **Probes**. It is a generic set of the desired particle type (with very loose selection criteria). It is selected by pairing these objects with tags, such as the invariant mass of the combination that must be consistent with the mass of the resonance.

To eliminate the background, one can use any of a variety of background subtraction methods such as fitting.

2.2 Tight and Loose Criteria

As **tight** criteria for the **Tags**, we use **isTightMuon()** function which has many functions that can determine a muon with very small fake rate. It contains:

```
isPFMuon()
isGlobalMuon()
isGoodMuon()
numberOfMatchedStations() > 1
muon.innerTrack()->hitPattern().trackerLayersWithMeasurement() > 5
muon.innerTrack()->hitPattern().numberOfValidPixelHits() > 0
```

```
muon.muonBestTrack()->dxy(vtx.position())) < 0.2
muon.muonBestTrack()->dz(vtx.position())) < 0.5
```

We choose Probes, by a loose criteria, only requiring them to be in the **MuonCollection**. We can calculate the invariant mass by pairing those two and see if it is consistent with a resonance like it was told above. For this code, we use the **TLorentzvector**, a ROOT Class. We compute some kinematics variables like pT (Tranverse Momentum), η(Pseudo-Rapidity) and Φ and extract what we need and record the values in a ROOT tree.

2.3 Efficiencies

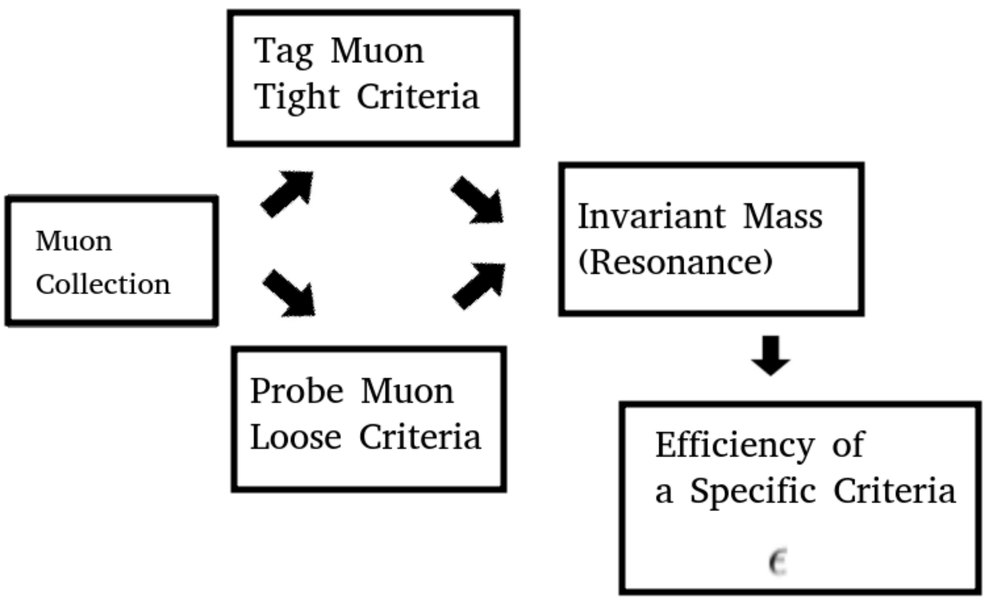

Figure 1: Workflow.

The definition of the probe object depends on the specification of the selection criteron being examined. The efficiency itself is measured by counting the number of “probe” particles that pass the desired selection criteria:

$$\epsilon = \frac{P_{passes}}{P_{all}} \tag{1}$$

For muons the total efficiency measurement can be broken down (factorized) into sequential measurements. The steps can be, for example: muon tracking efficiency **muon tracking**

efficiency, **stand-alone muon reconstruction efficiency**, **identification efficiency**, **isolation efficiency** and **the online or total trigger efficiency**. The total efficiency is given by the product:

$$\epsilon = \epsilon_{trk} \times \epsilon_{sta} \times \epsilon_{id} \times \epsilon_{iso} \times \epsilon_{online} \tag{2}$$

The exact definition of each efficiency component is not relevant for this first work, but one have to to know that the total efficiency is the product of the the sequential efficiencies. The estimate of this total efficiency (2) is important, with the luminosity, to calculate the cross section of a determined process:

$$\sigma = \frac{N}{\int Ldt} = \frac{N_{measured}}{\epsilon \int Ldt} \tag{3}$$

In this first work, we will focus on the first step of the process: extracting the sample for the efficiency computation. The workflow representing this first step can be seen in the figure 1.

3 Results

We use the J/ψ resonance in this example and make an output tree (root file) with the same content as the CMSSW **TagAndProbe** method as you can see in the figure 2.

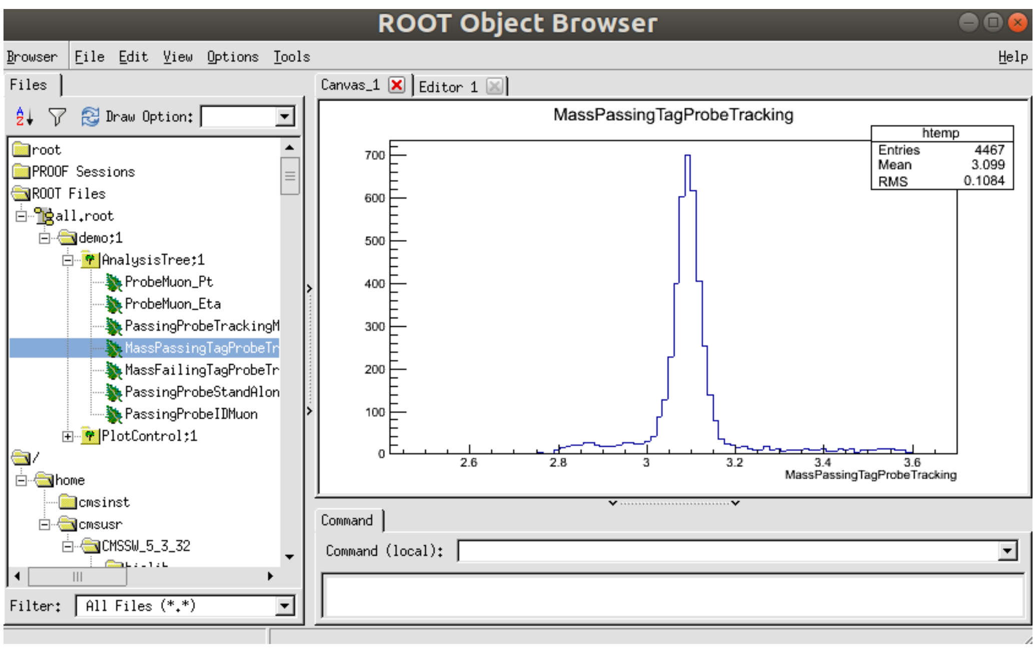

Figure 2: Invariant Mass.

In this tree, we probe the efficiency of the tags using track muon criteria, i.e., how often probe muons pass in the track muon criteria. This rate can be seen in figure 3.

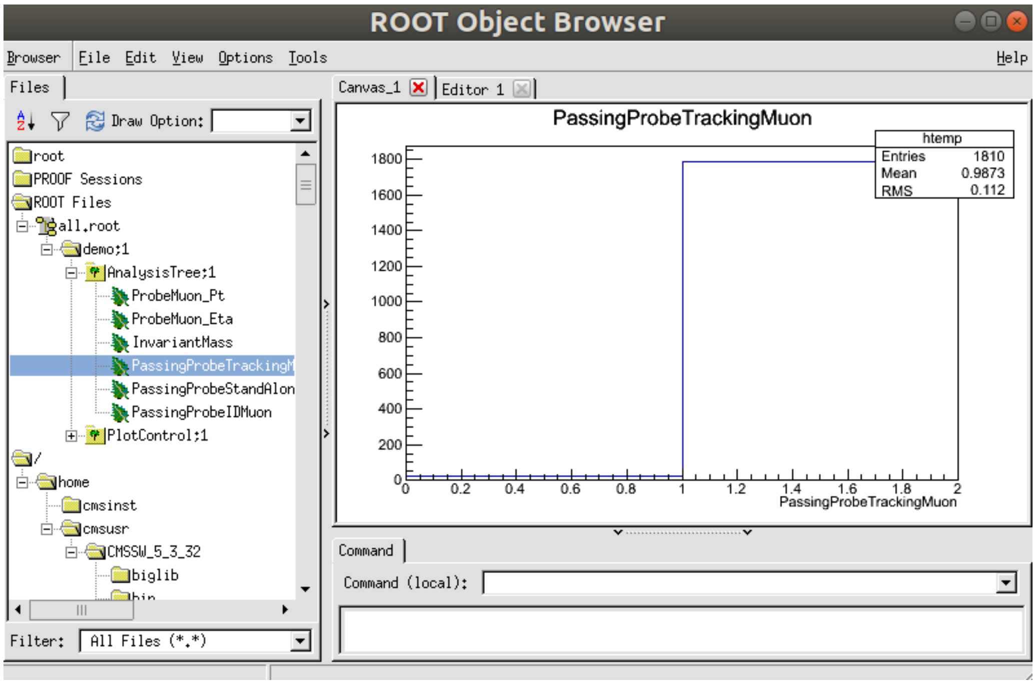

Figure 3: Efficiency of Tracking Muons.

4 Conclusions and next steps.

We implement a simplified Tag and Probe sample extraction using CMS Open Data. The current implementation uses the J/ψ resonance but it can be changed to other particles (e.g. Z) by changing the trigger and selecting a dataset with more statistic around this resonances as singleMu [3]. The source code can be found in this repository:

https://github.com/cms-legacydata-analyses/TagAndProbe

The instruction are documented in a Twiki page, which, after a review, can be made available for others CMS Open Data users as a simple example of the Tag and Probe process.
The next step of this work is to run the Tag and Probe code on the entire dataset for more statistic. After that, a fit can be made of the Tags and Probes that passes the selection to estimate the efficiency of the selection.

5 References.

[1] Tag and Probe. https://twiki.cern.ch/twiki/bin/view/CMSPublic/TagAndProbe

[2] Generic Tag and Probe Tool for Measuring Efficiency at CMS with Early Data. N. Adam,

J. Berryhill, V. Halyo, A. Hunt, K. Mishra. **CMS AN -2009/111**.

[3] http://opendata.cern.ch/record/32.

[4] Repository of the source code. https://github.com/cms-sw/cmssw/blob/CMSSW_5_3_X/DataFormats/MuonReco/src/MuonSelectors.cc#L709