Tag and Probe Tutorial
CMSSW 3 1 2

N. Adam, J. Berryhill, V. Halyo,
A. Hunt, K. Mishra

Outline

- Tag and Probe Overview
  - Definition of tag and probe efficiency
  - Examples of tag and probe results
- Tag and probe work flow
  - Creating Tag and Probe nTuples
  - Creating Fit files
- Tutorial Instructions
- Additional information

Tag And Probe Overview

- Tag and probe is a data driven technique used to calculate efficiencies.
- In order to calculate the efficiency one needs a mass resonance (i.e. J/psi, upsilon or Z), or a well known PDF.
- The Tag is a muon or electron that has very tight selection criteria and a very low fake rate.
- The Probe has looser criteria.
- The Passing Probe has tighter criteria than the probe, but not tighter than the Tag (unless the Tag and Probe sets are mutually exclusive).

Definition of Efficiency

Efficiency of the probe is the number of passing probes divided by the total number of probes.

$$E(Probe) = \frac{N_{passing\ probes}}{N_{all\ probes}}$$

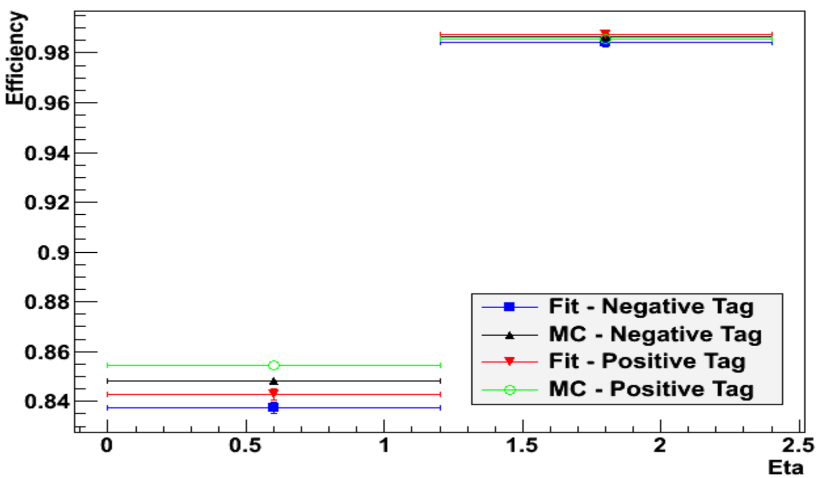
Ideal

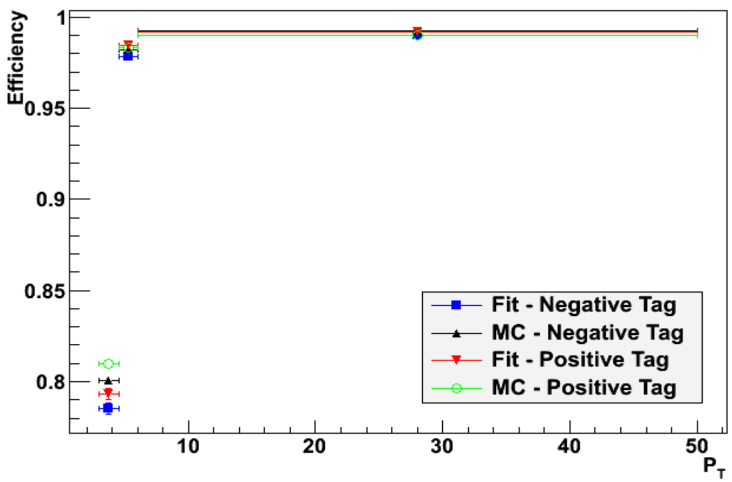
$$E(Probe) = \frac{2N_{TT} + N_{TP}}{2N_{TT} + N_{TP} + N_{TF}}$$

Tag and Probe

The probe efficiency calculated by tag and probe does depend depend on the definition of the tag.

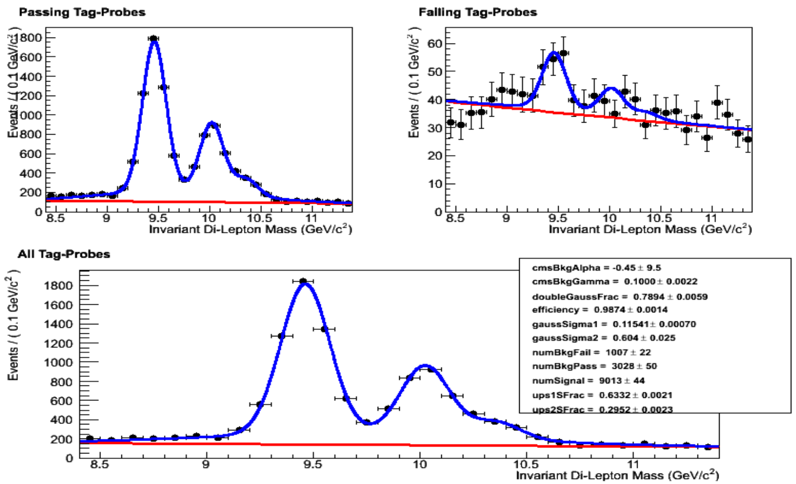
Upsilon Tracker Muon Efficiency



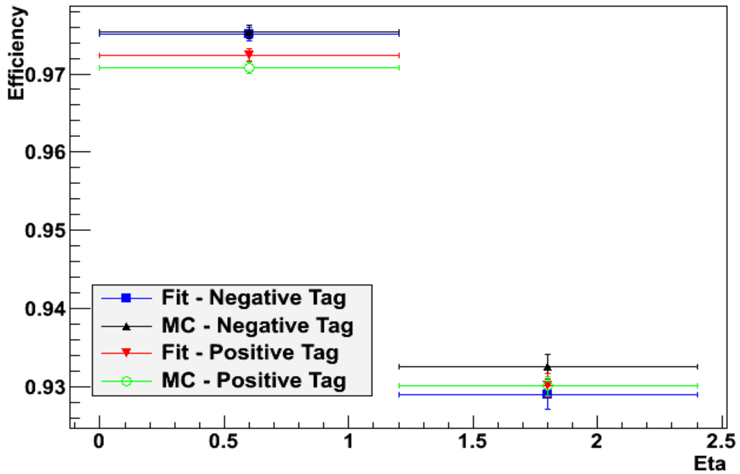


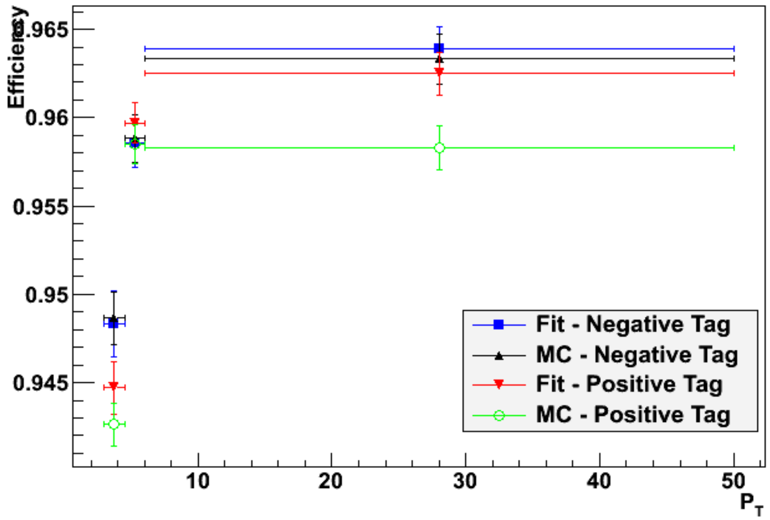
Probe – General tracks with muon hypothesis and pt > 3.0 GeV.

Passing Probe – Tracker Muon



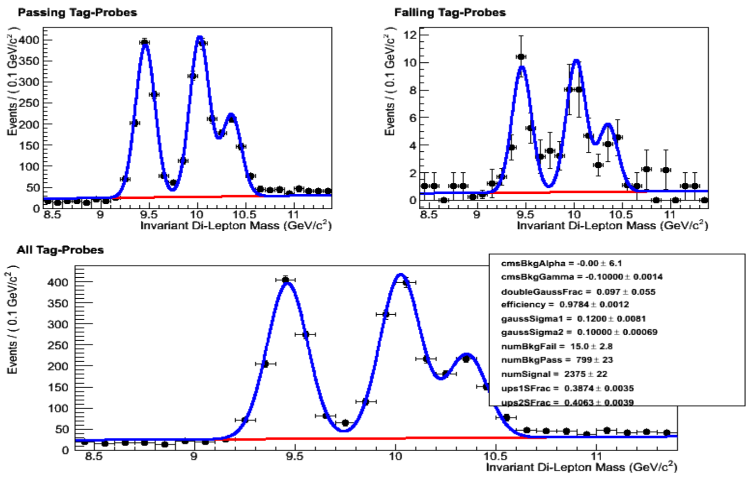
Identification Efficiency



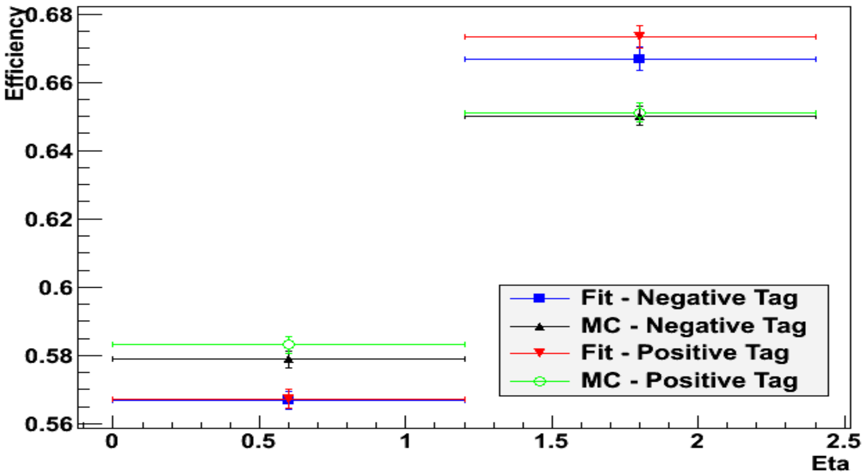


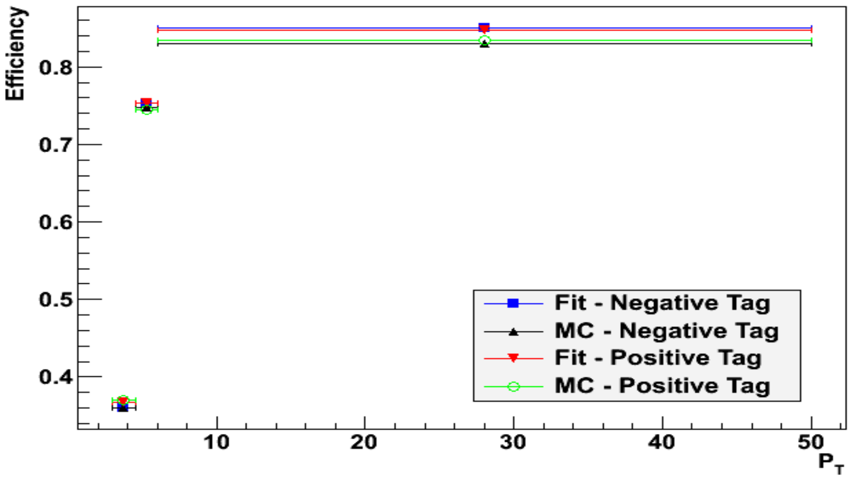
Probe – Tracker muon with pt > 3.0 GeV

Passing Probe – Probe + TMLastStationOptimizedLowPtTight



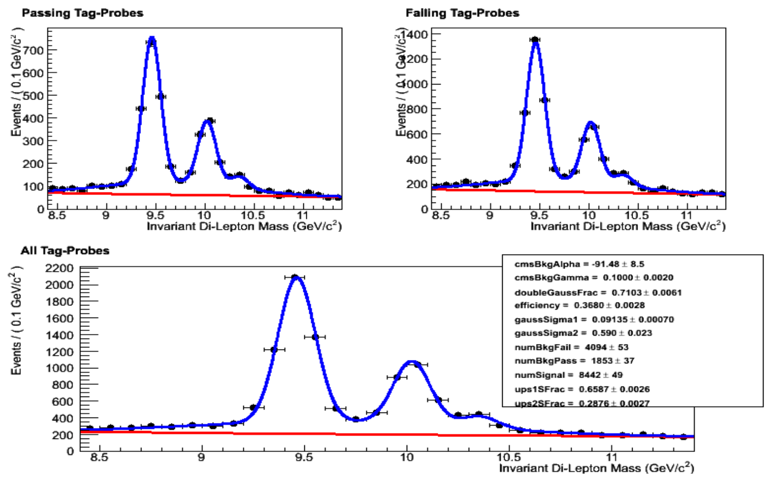
HLT Efficiency





Probe – Tracker muon + TMLastStationOptimizedLowPtTight

Passing Probe – Probe matched to HLT_Mu3

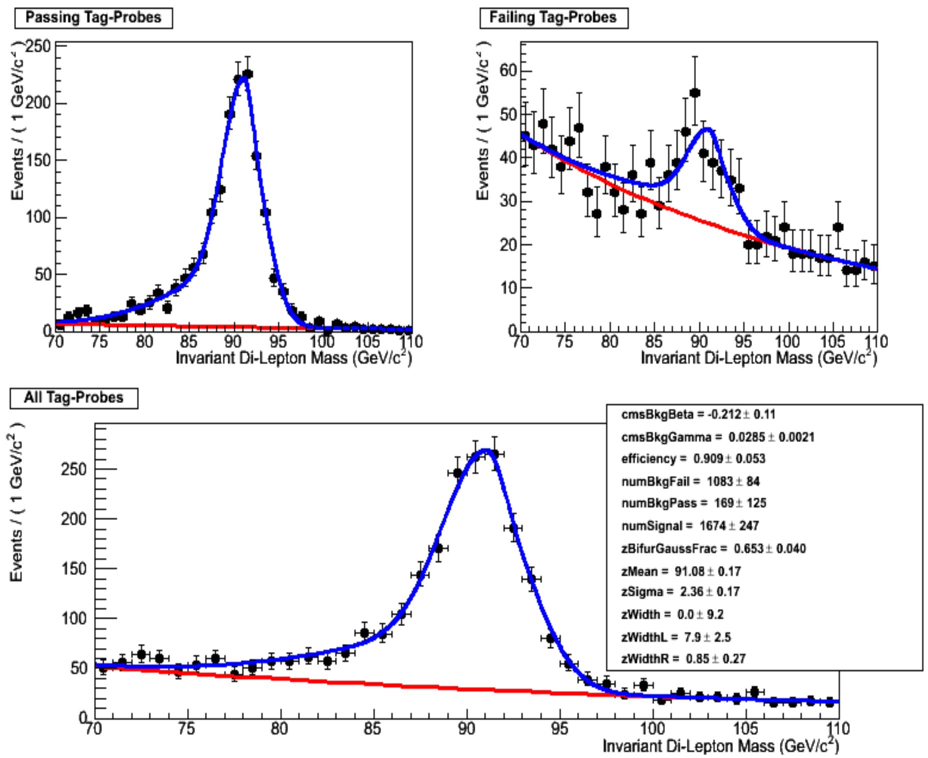


Efficiency Table

Muon-Track Efficiency			
Positive Muons	$p_T \in (3.0, 4.5)$	$p_T \in (4.5, 6.0)$	$p_T \in (6, \infty)$
$\|\eta\| \in (0.0, 1.2)$	0.6865 ± 0.0043	0.9709 ± 0.0025	0.9953 ± 0.0021
$\|\eta\| \in (1.2, 2.4)$	0.9777 ± 0.0041	0.9882 ± 0.0026	0.9904 ± 0.0023
Negative Muons	$p_T \in (3.0, 4.5)$	$p_T \in (4.5, 6.0)$	$p_T \in (6, \infty)$
$\|\eta\| \in (0.0, 1.2)$	0.7021 ± 0.0043	0.9813 ± 0.0025	0.9894 ± 0.0022
$\|\eta\| \in (1.2, 2.4)$	0.9752 ± 0.0039	0.9948 ± 0.0027	0.9876 ± 0.0023
Muon-Id Efficiency			
Positive Muons	$p_T \in (3.0, 4.5)$	$p_T \in (4.5, 6.0)$	$p_T \in (6, \infty)$
$\|\eta\| \in (0.0, 1.2)$	0.9765 ± 0.0013	0.9747 ± 0.0013	0.9779 ± 0.0014
$\|\eta\| \in (1.2, 2.4)$	0.9140 ± 0.0027	0.9347 ± 0.0024	0.9537 ± 0.0022
Negative Muons	$p_T \in (3.0, 4.5)$	$p_T \in (4.5, 6.0)$	$p_T \in (6, \infty)$
$\|\eta\| \in (0.0, 1.2)$	0.9674 ± 0.0015	0.9782 ± 0.0012	0.9703 ± 0.0016
$\|\eta\| \in (1.2, 2.4)$	0.9145 ± 0.0027	0.9308 ± 0.0024	0.9421 ± 0.0024
Muon-HLT Efficiency			
Positive Muons	$p_T \in (3.0, 4.5)$	$p_T \in (4.5, 6.0)$	$p_T \in (6, \infty)$
$\|\eta\| \in (0.0, 1.2)$	0.2506 ± 0.0032	0.7589 ± 0.0033	0.8812 ± 0.0031
$\|\eta\| \in (1.2, 2.4)$	0.5287 ± 0.0046	0.7391 ± 0.0044	0.7849 ± 0.0045
Negative Muons	$p_T \in (3.0, 4.5)$	$p_T \in (4.5, 6.0)$	$p_T \in (6, \infty)$
$\|\eta\| \in (0.0, 1.2)$	0.2563 ± 0.0032	0.7573 ± 0.0034	0.9000 ± 0.0029
$\|\eta\| \in (1.2, 2.4)$	0.5362 ± 0.0045	0.7383 ± 0.0043	0.7686 ± 0.0047

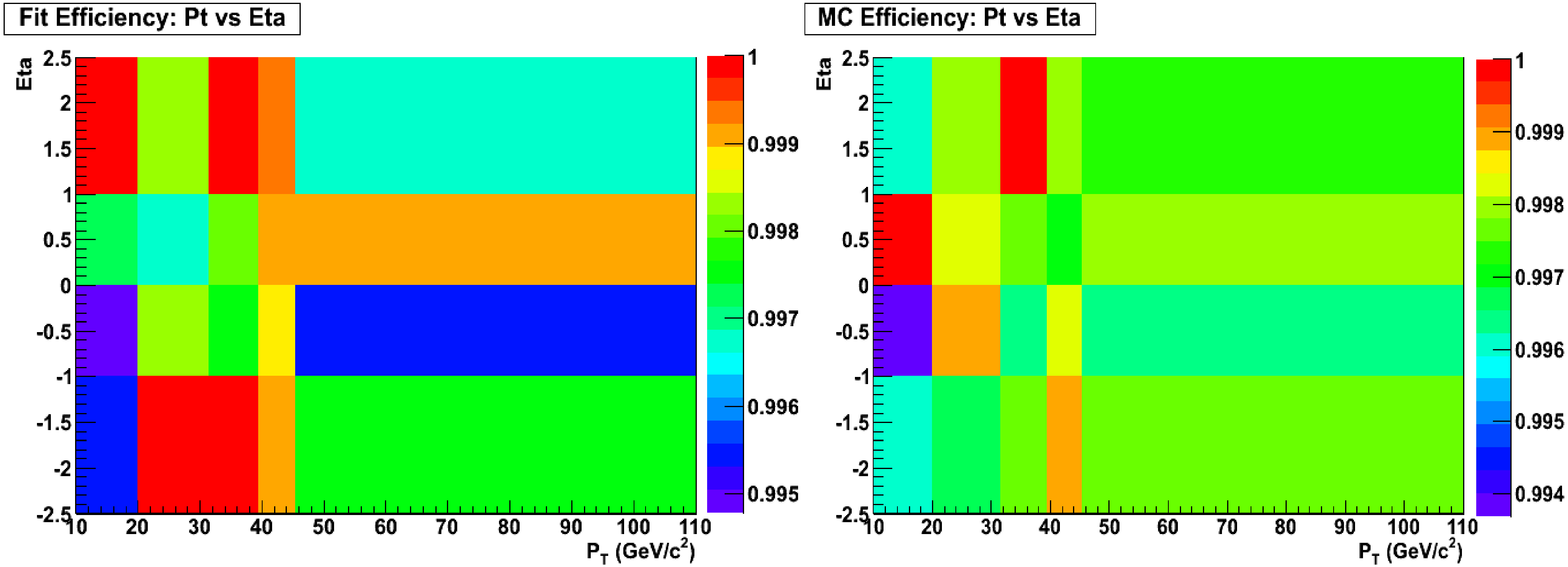
Table 4: Muon efficiency tables for low p_T muons.

Fit example



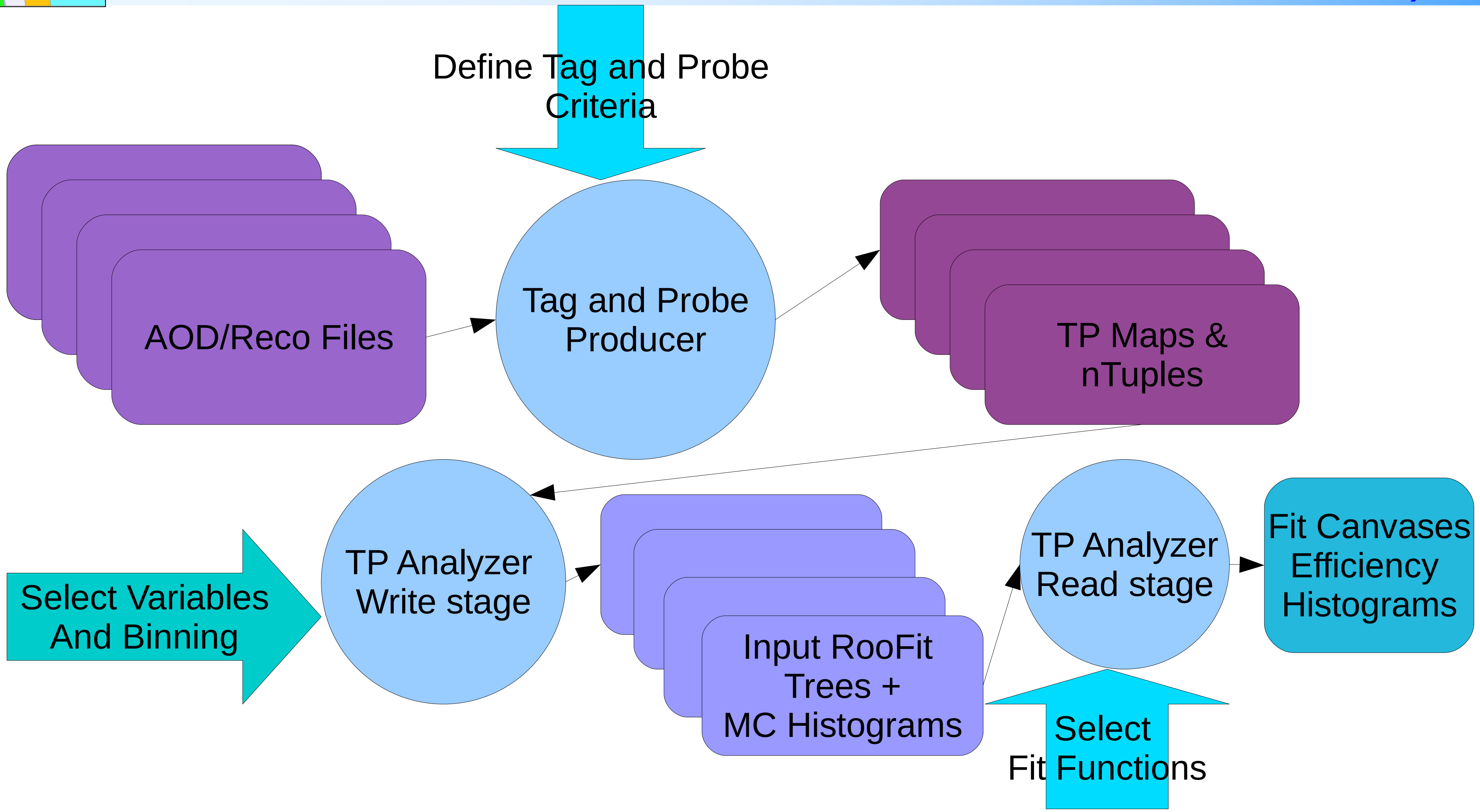
Z ->Mu Mu Reco Efficiency

2D Efficiency Example



Reco muon efficiency for Z -> MuMu

Tag And Probe Work Flow



Step 1- Creating TP ntuples

1) Define selection criteria

- Tag
- Probe
- Passing probe

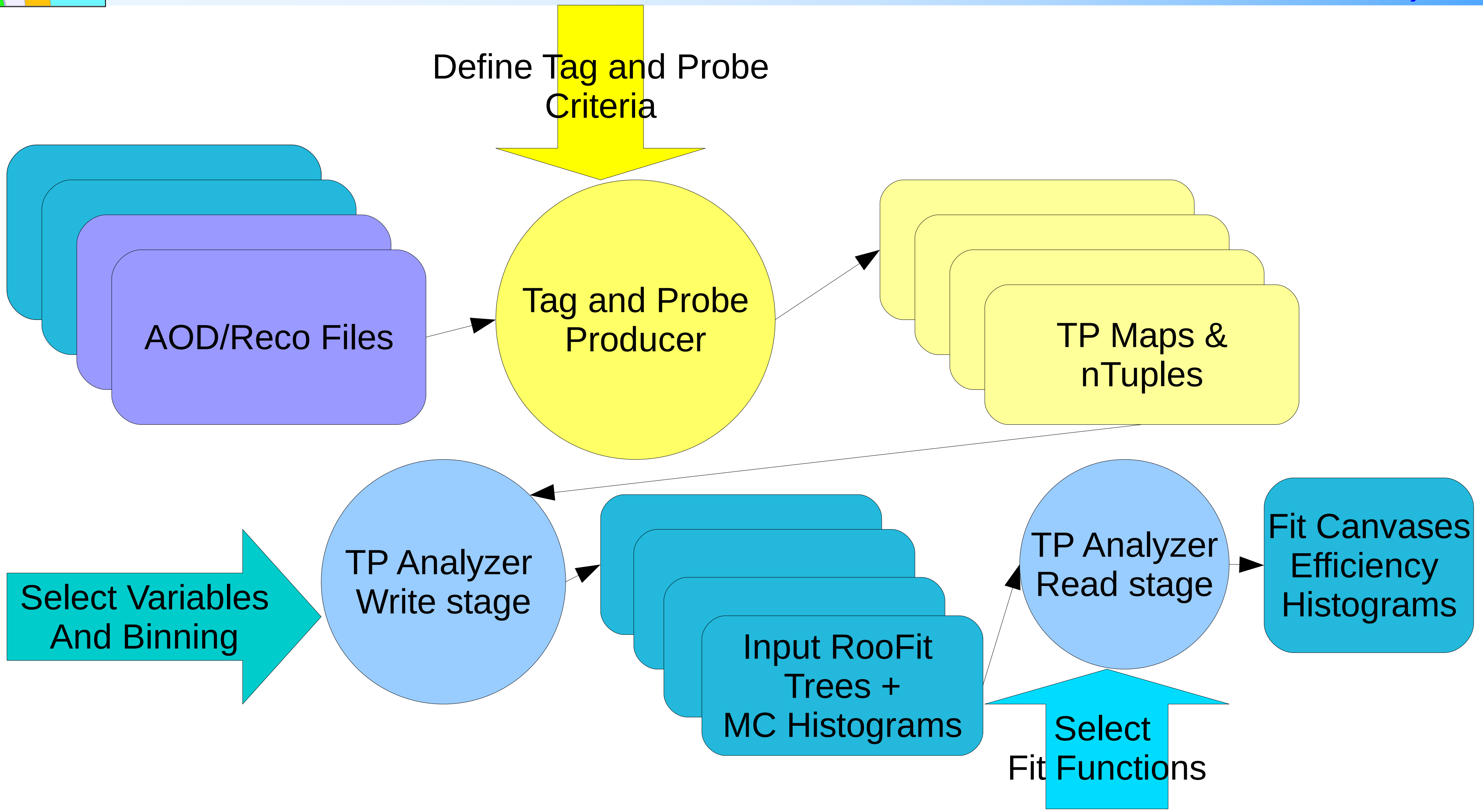
2) Create Tag Probe maps

3) Perform Monte Carlo truth matching

4) Create Tag Probe nTuples

5) To reduce the size of the output files, filter out events that do not have TP Maps

Step 1 – Creating TP ntuples



1.1 Tag and Probe criteria

Example using MuonRefSelector.

```
tagCands = cms.EDFilter("MuonRefSelector",
    src = cms.InputTag( "muons" ),
    cut = cms.string( "isGlobalMuon &  pt > 20.0 & isolationR05.sumPt < 5.0" )
    )

probeCands = cms.EDFilter ("MuonRefSelector",
    src = cms.InputTag( "muons" ),
    cut = cms.string( "isTrackerMuon && pt > 20.0" )
    )

passProbeCands = cms.EDFilter ("MuonRefSelector",
    src = cms.InputTag( "muons" ),
    cut = cms.string( "isTrackerMuon && isStandAloneMuon && pt > 20.0" )
    )
```

1.2 Tag and Probe Producer

Options

- **TagCollection** – The name of the EDFilter or EDProducer used to generate your tag collection
- **ProbeCollection** – The name of the EDFilter or EDProducer used to generate you probe collection
- **MassMinCut** – Minimum invariant mass calculated from the tag-probe pair.
- **MassMaxCut** – Maximum invariant mass.
- **DelRMinCut** – Minimum delta R between the tag and probe. Default: 0
- **DelRMaxCut** – Maximum delta R. Default: 10^6
- **RequireOS** – Require the tag and probe to have opposite charge. Default: True

Output

A map containing references to the original objects.

TP Producer Example

Tag-probe pair map with

- No delta R restriction
- With opposite sign requirement

```
ZTagProbeMap = cms.EDProducer("TagProbeProducer",
    MassMaxCut = cms.untracked.double (120.0),
    MassMinCut = cms.untracked.double (50.0),
    TagCollection = cms.InputTag ("tagCands"),
    ProbeCollection = cms.InputTag ("probeCands")
)
```

1.3 Monte Carlo Matching

We use the standard matching tools in CMSSW.

Three maps are needed for tag, probe and passing probe.

```
tagMuonMatch = cms.EDFilter ( "MCTruthDeltaRMatcherNew",
    pdgId   = cms.vint32( 13 ),
    src     = cms.InputTag( "tagCands" ),
    matched = cms.InputTag( "genParticles" ),
    distMin = cms.double( 0.15 )
    )
```

1.3 Monte Carlo Matching

```
allProbeMuonMatch = cms.EDFilter("MCTruthDeltaRMatcherNew",
   pdgId   = cms.vint32( 13 ),
   src     = cms.InputTag( "probeCands" ),
   matched = cms.InputTag( "genParticles" ),
   distMin = cms.double( 0.15 )
   )

passProbeMuonMatch = cms.EDFilter("MCTruthDeltaRMatcherNew",
   pdgId   =  cms.vint32( 13 ),
   src     = cms.InputTag( "passProbeCands" ),
   matched = cms.InputTag( "genParticles" ),
   distMin = cms.double( 0.15 )
)
```

1.4 Tag and Probe nTuple

Options

- **tagProbeType** – Muon or Electron
- **checkExactOverlap** – If true, delta R must be less than 1E-3 and delta Pt must be less than 1E-3.
- **isMC** – Fill Monte Carlo information. Default: True.
- **mcParticles** – The ntuple will contain information on all pdgids listed.
- **mcParents** – If pdgid from mcParticle must come from a specific particle, list it here. Otherwise, enter 0.
- **tagProbeMapTags** – Name of the TagProbeProducer modules.

Output

- A flat tree containing select particle information (pt, eta, etc.).

1.5 Filter Events

The Tag and Probe filter passes the events that contain nTuples. This reduces the size of the background files significantly.

```
process.TPFilter = cms.EDFilter("TagProbeEDMFilter")
```

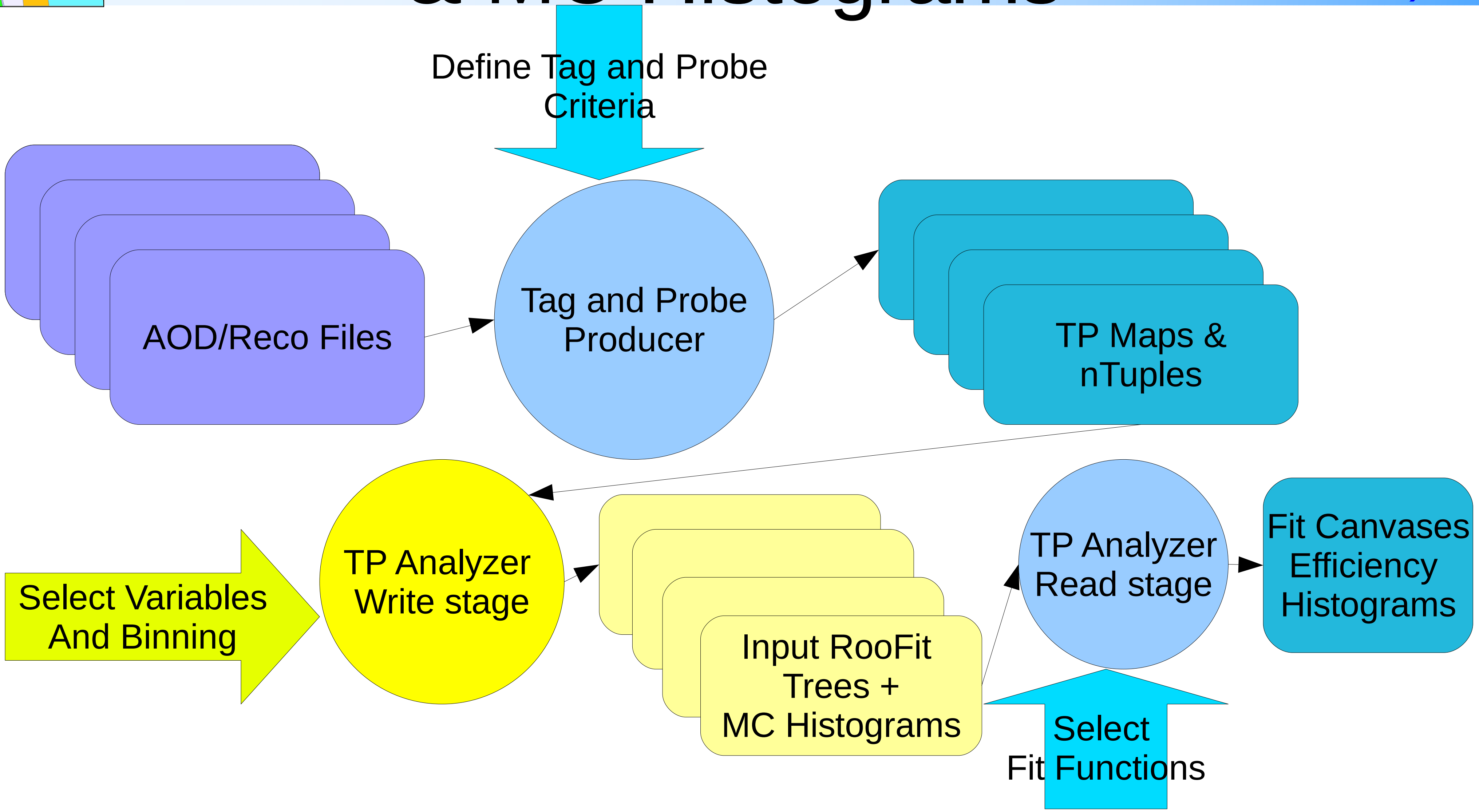
Step 2 – Create Input RooFit Trees & MC Histograms

1) Choose your independent variables on which efficiency will depend.

2) Choose your independent variable binning.

3) Choose the number of mass bins for chi^2 fitting.

Step 2 – Create Input RooFit Trees & MC Histograms



2.1 – Choose your efficiency variables.

Possible variables include:

pt, p, px, py, pz,

e, et, eta, phi,

jetDeltaR, and totJets.

Appending this list can be done in the TagProbeEDMNTuple Producer.

2.2 – Choose your variable binning

Variable or constant width binning is available.

```
NameVar1             = cms.untracked.string( "pt" ),

Var1BinBoundaries   = cms.untracked.vdouble( 0.0,45.0,100.0 ),


NameVar2             = cms.untracked.string( "eta" ),

Var2BinBoundaries   = cms.untracked.vdouble( -2.4,0.0,2.4 ),
```

2.3 – Choose the mass bins

Choose the number of mass bins, the maxium mass and the minimum mass.

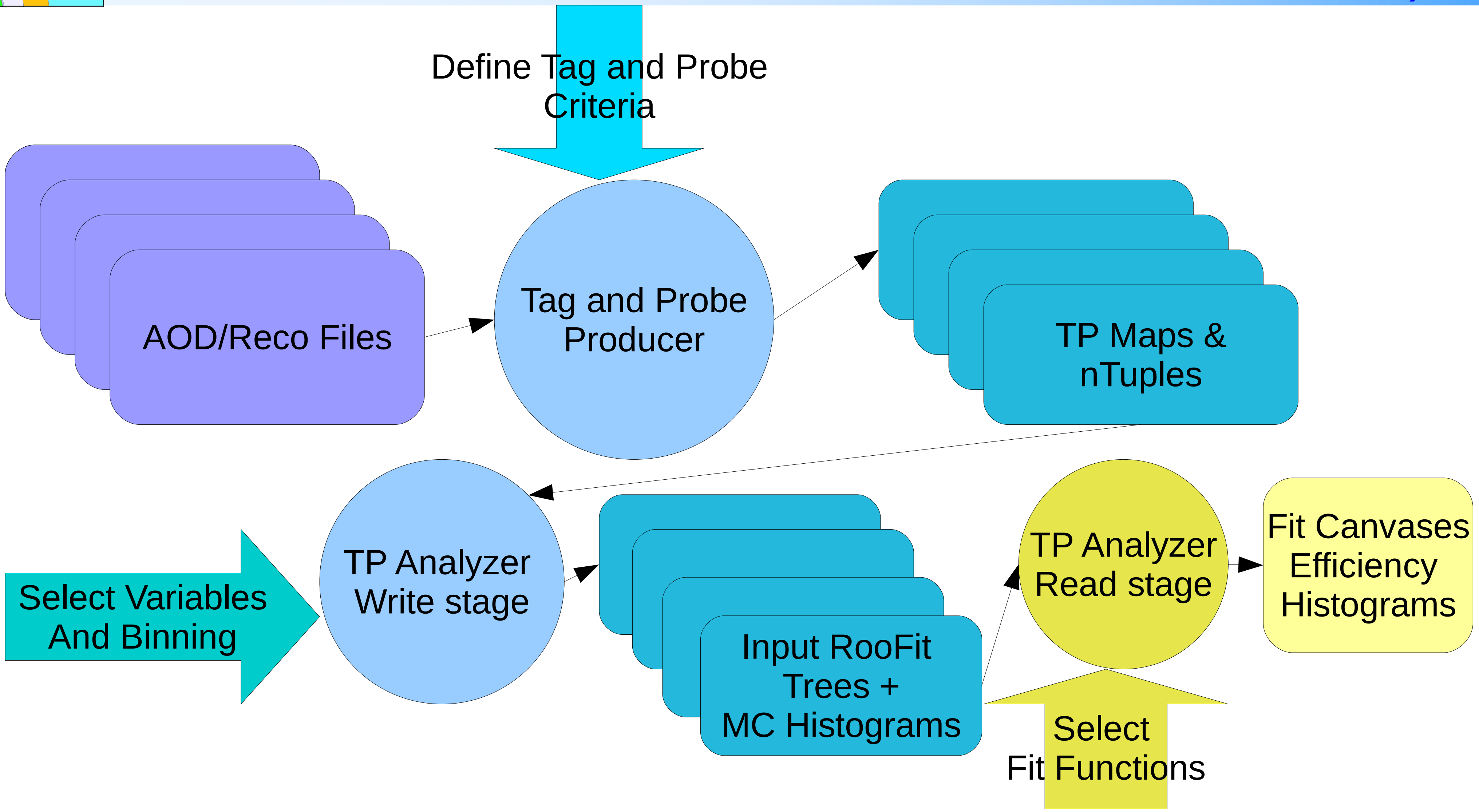
```
NumBinsMass       = cms.untracked.int32 ( 10 ),
MassLow           = cms.untracked.double ( 70.0 ),
MassHigh          = cms.untracked.double ( 110.0 )
```

Step 3 – Calculate Efficiencies

1) Determine what method(s) you would like to perform to calculate efficiencies (Side band subtraction, log likelihood, or chi^2 fit).

2) Choose your line shapes for signal and background.

Step 3 – Calculate Efficiencies



3.1 Tag and Probe Analyzer

Options

- **CalculateEffSideBand** – Use side band subtraction to calculate efficiencies.
- **CalculateEffFitter** – Use the RooFit simulatneous fitter to calculate efficiencies.
- **CalculateEffTruth** – When Monte Carlo is available, calculate the Monte Carlo efficiencies.
- **UnbinnedFit** – If true, perform a log likelihood fit. Otherwise, perform a binned chi^2 fit.

3.2 RooFit Line Shapes

Signal

- Z line shape
- Crystal Ball
- Gaussian

Background

- CMS background
- Polynomial

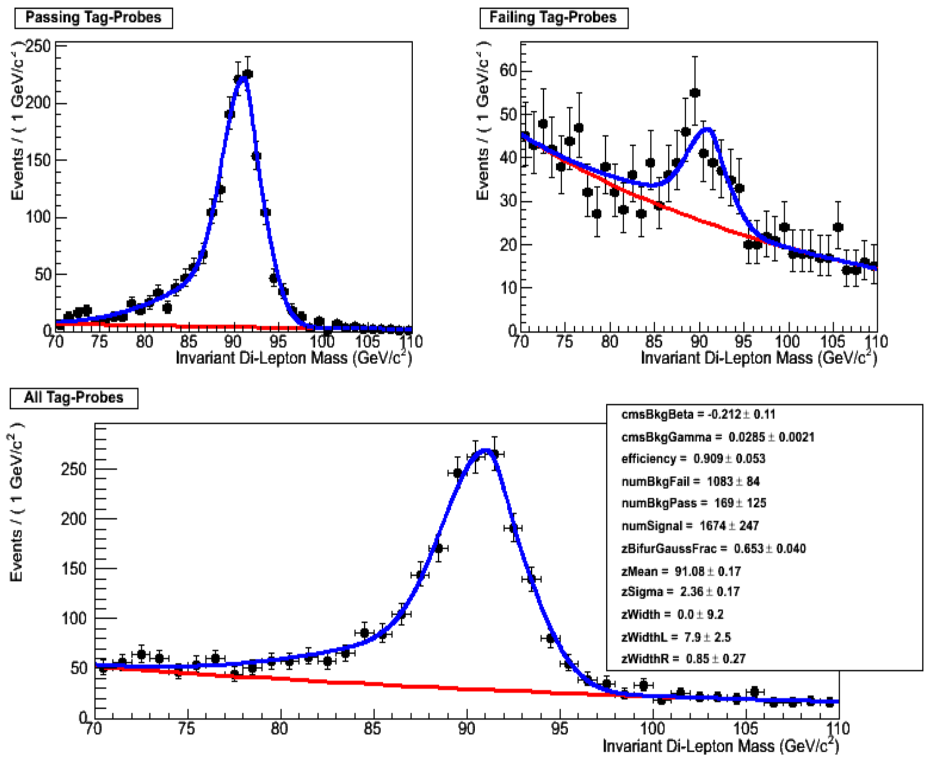
Additional line shapes can be added easily.

Example Line Shape Config

Z line shape with exponential background

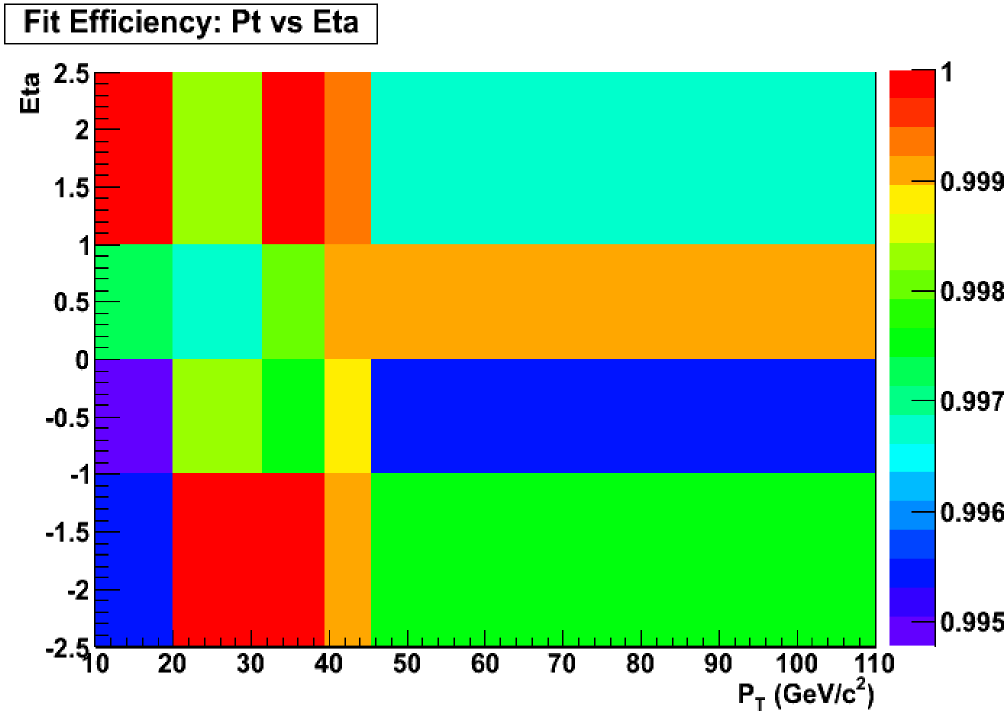
```
ZLineShape = cms.untracked.PSet(
   ZMean       = cms.untracked.vdouble( 91.1876, 89.0, 93.0 ),
   ZWidth      = cms.untracked.vdouble( 2.5, 0.1, 10.0 ),
   ZSigma      = cms.untracked.vdouble( 0.75, 0.00, 10.0 ),
   ZWidthL     = cms.untracked.vdouble( 0.0 ),
   ZWidthR     = cms.untracked.vdouble( 0.0 ),
   ZBifurGaussFrac    = cms.untracked.vdouble( 1.0 )
 ),

 CMSBkgLineShape = cms.untracked.PSet(
   CMSBkgAlpha          = cms.untracked.vdouble( 0.0 ),
   CMSBkgBeta           = cms.untracked.vdouble( 100.0 ),
   CMSBkgPeak           = cms.untracked.vdouble( 91.1876 ),
   CMSBkgGamma             = cms.untracked.vdouble( 0.04, 0.0, 0.4 )
 ),
```

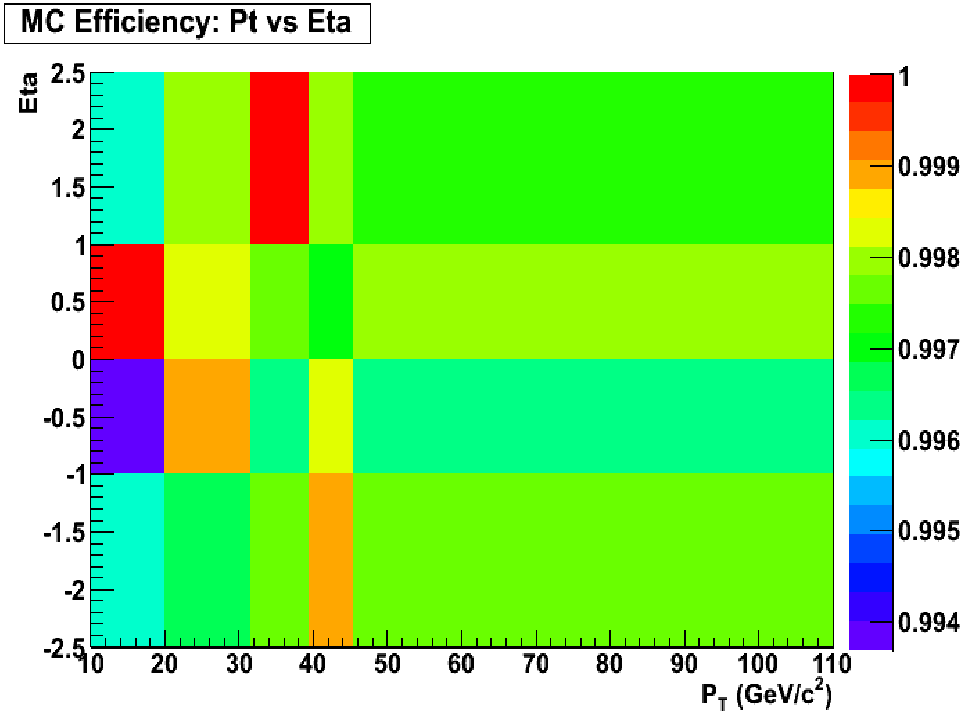
Example Fit Results



Z->MuMu Reconstruction Efficiency

2D Efficiency Example





Z->MuMu Reconstruction Efficiency

Setting up the tutorial

Download and install the tutorial.

```
$ scram p CMSSW CMSSW_3_1_2
$ cd CMSSW_3_1_2/src/
$ cmsenv
$ cvs co UserCode/ahunt/TPTutorial.tgz
$ tar xvf UserCode/ahunt/TPTutorial.tgz
$ cvs -R Tutorial-V01 co PhsyicsTools/TagAndProbe
$ scram b
```

Tutorial Instructions

Go into the TPTutorial/ZMuMuTutorial directory

Run the two configurations to generate the nTuple files.

$ cmsRun Muon_EDMNtuple_RecoEff_InclusiveMuMu_cfg.py

$ cmsRun Muon_EDMNtuple_RecoEff_ZMuMu_cfg.py

Once your EDM files are complete, you can run the analyzer to perform the fits.

$ cmsRun Muon_Analyzer_RecoEff_cfg.py

Summary

Very few files depend on CMSSW_3_1_X. Most of the package can be used in CMSSW_2_2_X.

A note has been uploaded to the document server (CMS AN 2009/111). It will be updated as the package changes.

Developers: Nadia Adam, Jeff Berryhill, Valerie Halyo, Adam Hunt, Kalanand Mishra

Additional comments from users are welcome and encouraged. Please post them on the twiki

A new generic tag and probe twiki has been created https://twiki.cern.ch/twiki/bin/view/CMS/TagAndProbe