University of Zurich

Short Exercise I - Observation of top quark and measurement of its production cross section

Annapaola de Cosa

(Exercise based on CMS Data Analysis Tutorial 2014 by Prof.C.Sender and Dr. A.Schmidt)

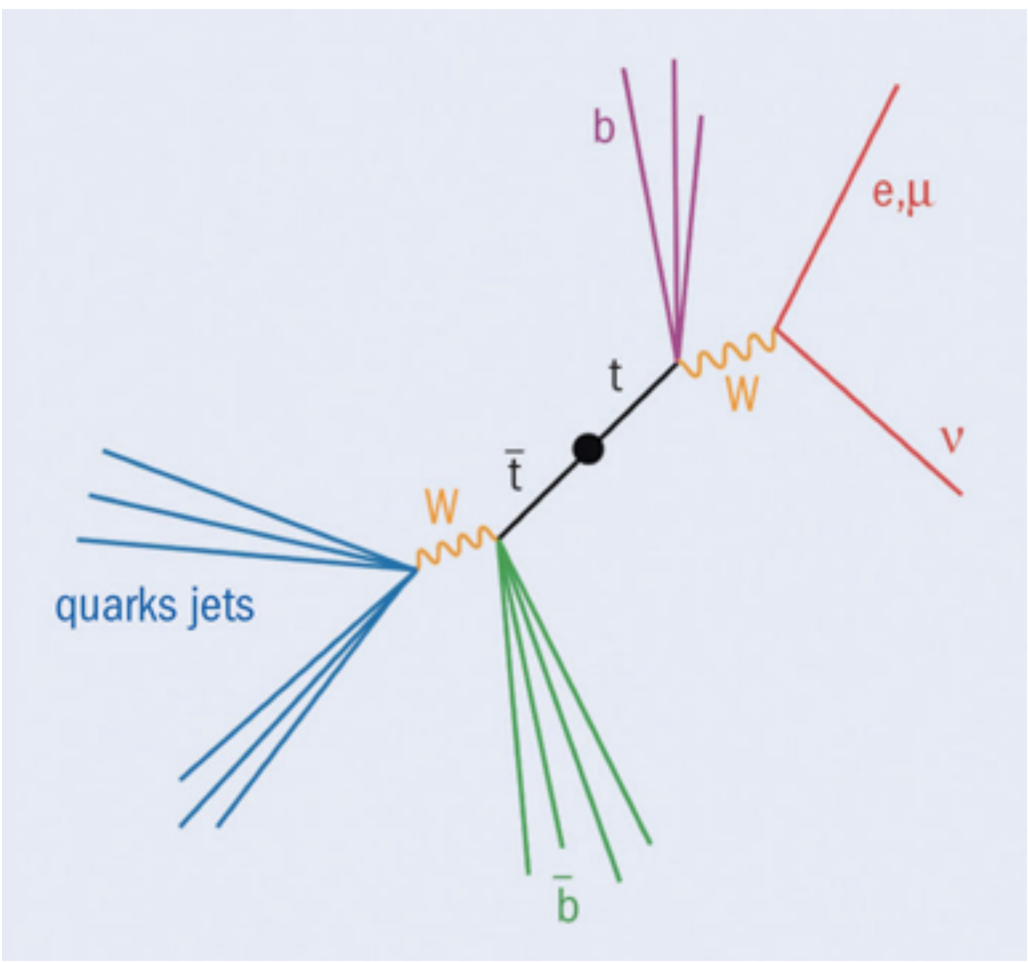
PHYS451 - Experimental Particle Physics

Exercise class 8

08th November 2016

Introduction

The short exercise consists in measuring the production cross section of top quark pairs performing a physics analysis of 50 fb-1 of data collected by the CMS experiment at the proton collider LHC



- **The top quark mean lifetime is about 5×10^{-25} s**
  - Too short to enable direct top quark reconstruction
  - The alternative is to measure final state decay products of top quark from energy deposits in the detector and indirectly reconstruct the top quark

Hunt for tt production

- **For this exercise only semileptonic decay of top-quark pair will be considered (in the muon channel)**
  - Expected signal final state: 4 jets out of which 3 from b quarks, 1 lepton and missing momentum from non-reconstructed neutrino

- **The hunt for ttbar production (signal) consists in correctly identifying the final state products to distinguish the reconstructed final state from other processes which have similar signature (called background processes)**
  - Examples: QCD jet production, electroweak W or Z boson production in association to jets
  - Moreover, additional objects such as jets from radiation or from overlapping collisions can be produced complicating the top reconstruction

Physics analysis of experimental data

Physics analysis workflow

Study of kinematical properties distinguishing signal events against background using Monte Carlo simulation

↓

Define a selection strategy based on these studies to suppress background and select signal-like data

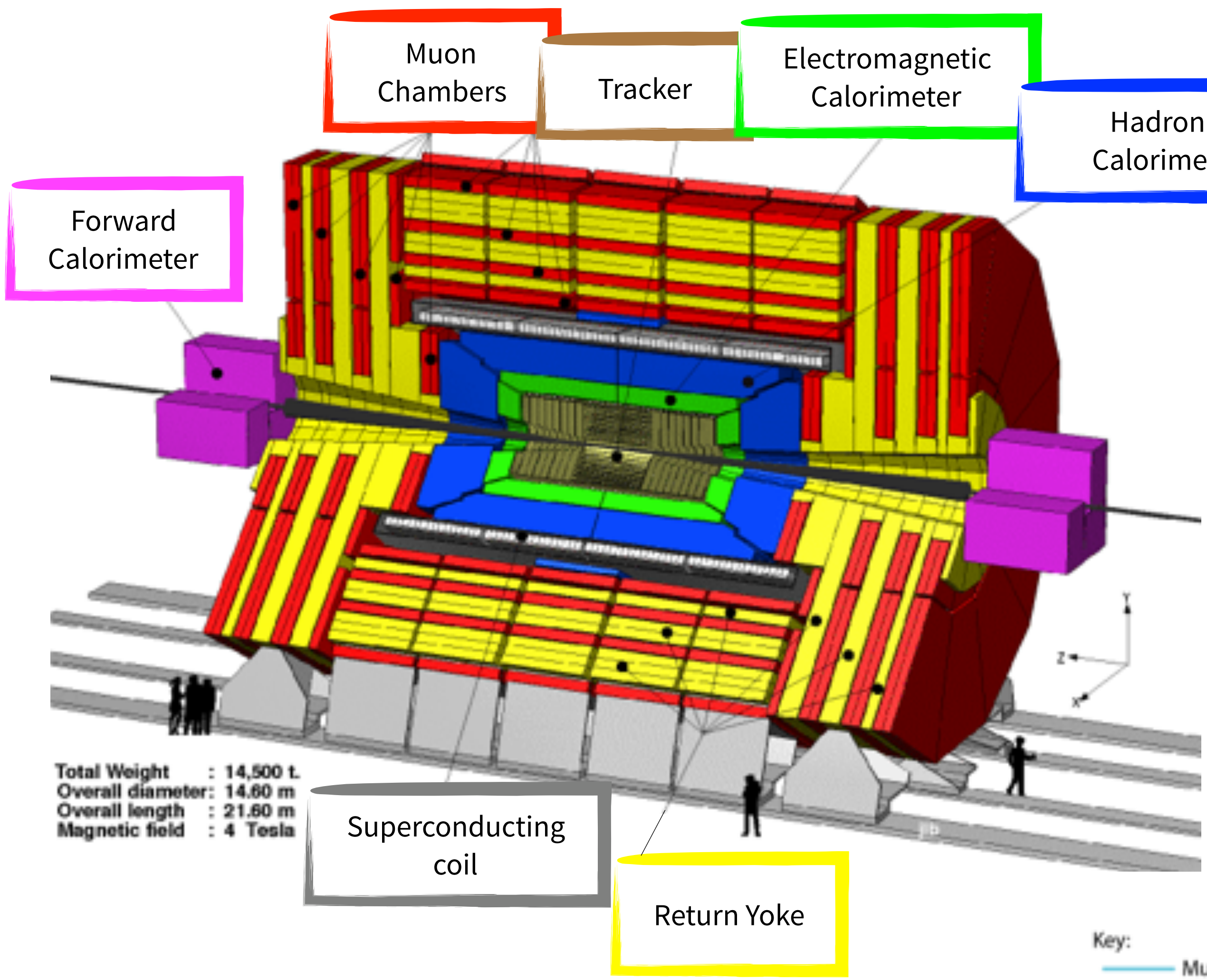
↓

Apply this selection on experimental data

↓

Estimate ttbar production cross section

The CMS experiment

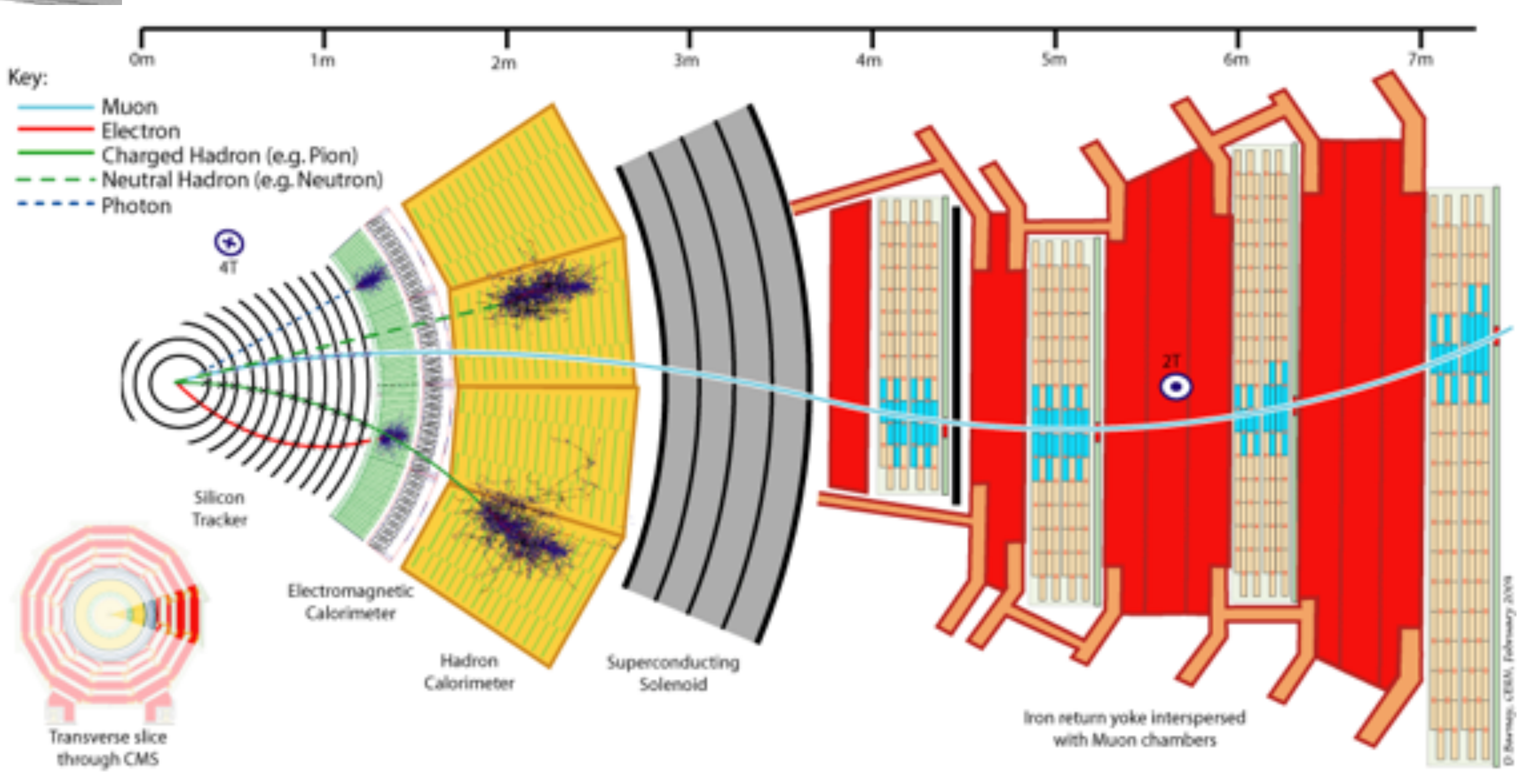


The Compact Muon Solenoid, CMS

- Total weight: 14.5 t.
- Overall diameter: 14.8 m
- Overall length: 21.6 m
- Internal magnetic field: 3.8 T
- External magnetic field: 1.8 T

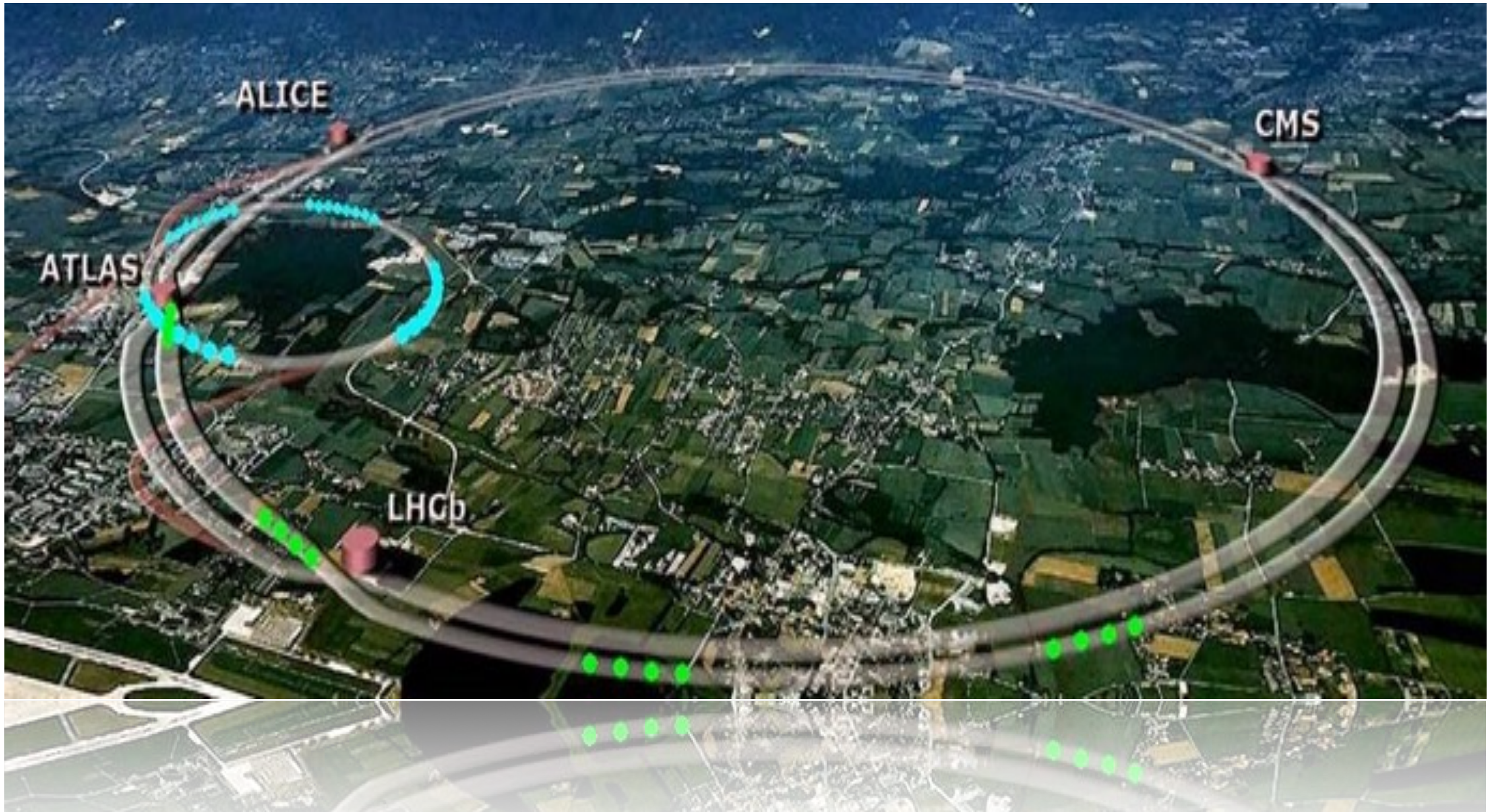
CMS main features

- Superconducting solenoidal magnet
- High-quality tracker system:
- High-performance electromagnetic calorimeter
- Highly hermetic hadron calorimeter



LHC

- **The Large Hadron Collider is built at CERN (Geneva), on the swiss-french border**
  - Accelerate bunch of protons through a circumference of 27 km up to 13 TeV c.m.e
  - This exercises makes use of 50 fb^{-1} of p—p collision data delivered by LHC at a $\sqrt{s}$ = 7 TeV



Exercise pantry!

Data and MC simulation samples together with a basic analysis framework are available at the following link:

- https://decosa.web.cern.ch/decosa/PHYS451_Fall2016/ShortExercise

- **The data samples for this exercise are stored in root trees, you can find them in the folder *files***
  - File names indicate which kind of simulated data are stored in the file (ttbar.root, qcd.root, etc.) or whether it contains real data (data.root)
- **The analysis framework will serve as base for your own analysis**

Data and MC simulation samples

- **A sample of real data collected by the CMS experiment in 2011 is stored in data.root**
  - Corresponding to **50 fb^{-1}** of proton-proton collisions at **$\sqrt{s}$ = 7 TeV**
- **Given the very large collision rate, not all produced data are stored**
  - A selection is performed by the trigger system to store only “interesting” data
  - Only events containing a reconstructed, isolated muon are saved
- **Data are stored in files as ROOT trees**
  - Each tree contains a selection of variables useful for the analysis
  - A list of all variables stored in the trees is included in the backup

Tree structure - Jets

Variable	Type	Content
NJet	Integer	Number of jets in the event
Jet_Px	vector of floats	x-component of jet momentum (only jets with pT> 30GeV are stored)
Jet_Py	vector of floats	y-component of jet momentum (only jets with pT> 30GeV are stored)
Jet_Pz	vector of floats	z-component of jet momentum (only jets with pT> 30GeV are stored)
Jet_E	vector of floats	Energy of the jet (only jets with pT> 30GeV are stored)
Jet_btag	vector of floats	b tag discriminator value
Jet_ID	vector of bools	Jet quality identifier - Used to reject jets from detector noise. A good jet has true value for this variable

Tree structure - Muons

Variable	Type	Content
NMuon	Integer	Number of muons in the event
Muon_Px	vector of floats	x-component of muon momentum
Muon_Py	vector of floats	y-component of muon momentum
Muon_Pz	vector of floats	z-component of muon momentum
Muon_E	vector of floats	Muon energy
Muon_Charge	vector of integer	Muon charge
Muon_Iso	vector of floats	Muon isolation value

Tree structure - MET

Variable	Type	Content
Met_Px	vector of floats	x-component of the missing energy
Met_Py	vector of floats	y-component of the missing energy
EventWeight	vector of floats	Weight factor to be applied to simulated events to account for different sample cross sections
triggerIsoMu24	vector of bools	Trigger bit, it is true if the event is triggered, and false if it is not

Analysis framework

- **You can develop your own analysis using the framework available on the web page**
- **The framework consists of different modules**
  - **MyAnalysis.py (*)**
  - Samples.py
  - Plotter.py
  - **example.py (*)**

(*) To modify

Analysis framework

- **MyAnalysis.py**
  - It's the core of the analysis implementation, here the selection strategy is implemented and histograms are booked and filled with selected events
  - Two main functions:
    - bookHistos: where histograms are defined
    - processEvent: where histograms are filled according to analyst criteria

```
def bookHistos(self):
   h_nJet = ROOT.TH1F("NJet","#of jets", 6, -0.5, 6.5)
   h_nJet.SetXTitle("%# of jets")
   self.histograms["NJet"] = h_nJet
    ....
```

Analysis framework

- **MyAnalysis.py**
  - It’s the core of the analysis implementation, here the selection strategy is implemented and histograms are booked and filled with selected events
  - Two main functions:
    - bookHistos: where histograms are defined
    - processEvent: where histograms are filled according to analyst criteria

```
def processEvent(self, entry):
   tree = self.getTree()
   tree.GetEntry(entry)
   w = tree.EventWeight

    for m in xrange(tree.NMuon):
     muon =ROOT.TLorentzVector(tree.Muon_Px[m],tree.Muon_Py[m],tree.Muon_Pz[m],tree.Muon_E[m])
     self.histograms["Muon_Iso"].Fill(tree.Muon_Iso[m], w)
```

Analysis framework

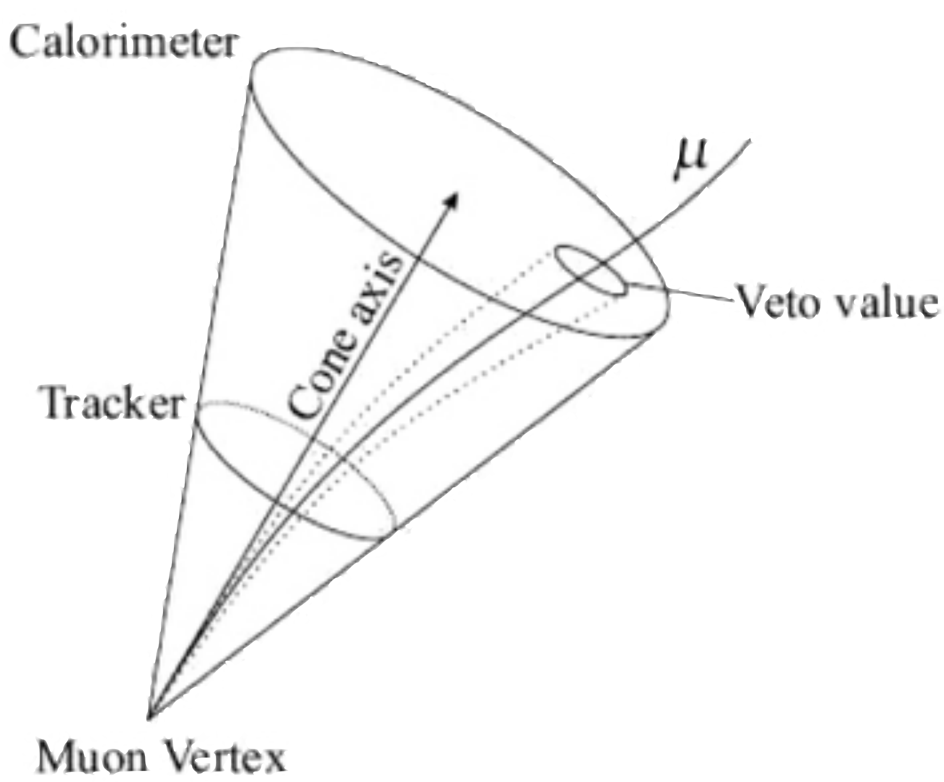
- **example.py**
  - It's the main python file. It calls the other modules and is used to define the actions, which samples should be processed and which plots should be created.
  - It returns .pdf files with the plot for all indicated variables.
  - Two kind of plots are produced:
    - Signal vs Background distribution normalised to one
    - A full plot representing the sum of background plus signal and comparison to data

```
for v in vars:
    print "Variable: ", v
    ### plotShapes (variable, samples,logScale )
    plotShapes(v, samples,  True)
    ### plotVar(variable, samples,isData, logScale )
    plotVar(v, samples,  True, True)
```

Isolation

Muons originating from W decay manifest themselves in the detector as a clean trajectory traversing the inner tracker up to the muon system

- Muons from b quark decay are instead produced within a jet

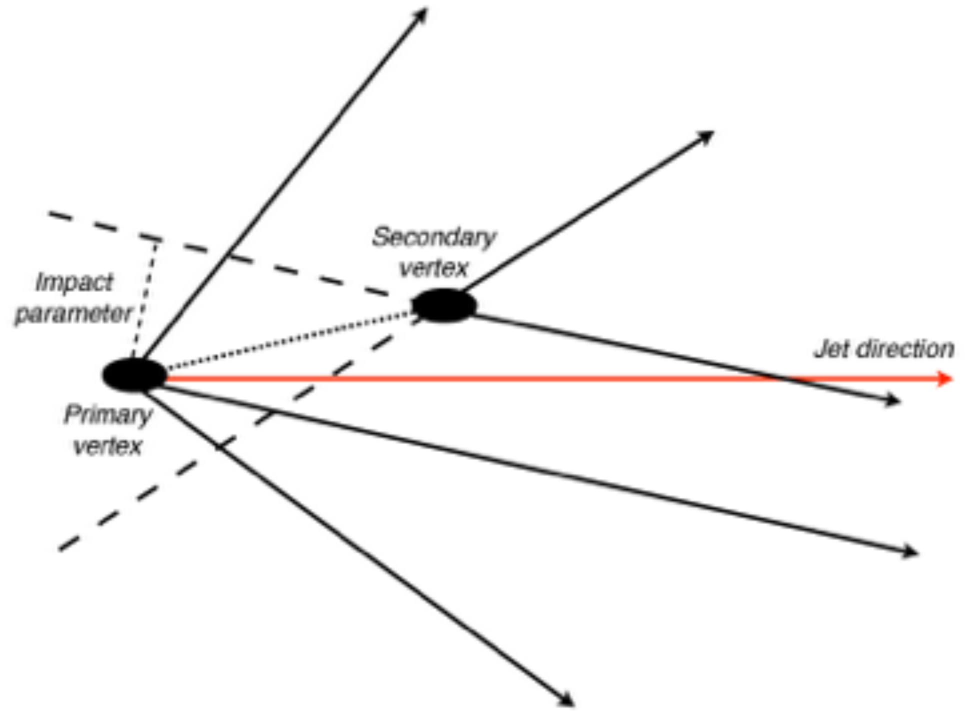


The isolation, I (Muon_Iso), is defined as **the sum of p_T of reconstructed particles** within a cone of size ΔR ($\Delta R = \sqrt{(\Delta\eta^2 + \Delta\vartheta^2)}$) around the muon. It is used to discriminate muons originated from W decay from those produced in jets, which are surrounded by a multitude of particles.

Isolated Muon: $I_\mu/p_T < 0.12$

b-jet identification

- **The long lifetime of B hadrons can be exploited to identify jets originating from b quarks**
  - The reconstruction of secondary vertex due to the decay of B mesons indicates the jet is a b-jet
- **A b-tagging discriminator (Jet_btag) is obtained from an algorithm that identifies secondary vertices**
  - Higher values of btagging discriminator indicates an higher probability that the jet comes from a b quark



A jet is tagged as b jet if b tag discriminant is greater than 2.0 (Jet_btag > 2.0)

Short Exercise - Part I

1) **Open the files data.root and ttbar.root and inspect them**

- Look interactively at the distributions of variables:
  - NJet, NMuon, Jet_Px/Py/Pz, Muon_Px/Py/Pz, Muon_Iso, Met_px/py
  - Which is the most probable value for the number of Jets (NJet) and for the number of Muons (NMuon) in data? And in simulated tear events?
- Pay attention to the variable triggerIsoMu24
  - Which values can you observe for data? And for MC?

2) **Using the analysis framework selects events with at least one isolated muon:**

- Relative isolation (Muon_Iso/ Muon_P_T) < 0.05
- Muon_P_T > 25 GeV

Short Exercise - Part I

3) **Compare the shapes of several distributions for the signal (ttbar) and the background (e.g number of jets, number of tagged jets)**

- Which ones provide best discrimination between signal and background?

4) **Find the most discriminating variables and choose a cut to apply in order to reject the background. Optimise the selection chasing the cut that gives the highest signal over background ratio**

5) **Produce the histograms of these variables comparing MC simulation and data**

- Use the function plotVar(v, samples, True, True) in example.py

6) **Apply this selection to MC and data**

- What is the highest purity that you can get (based on simulation)?