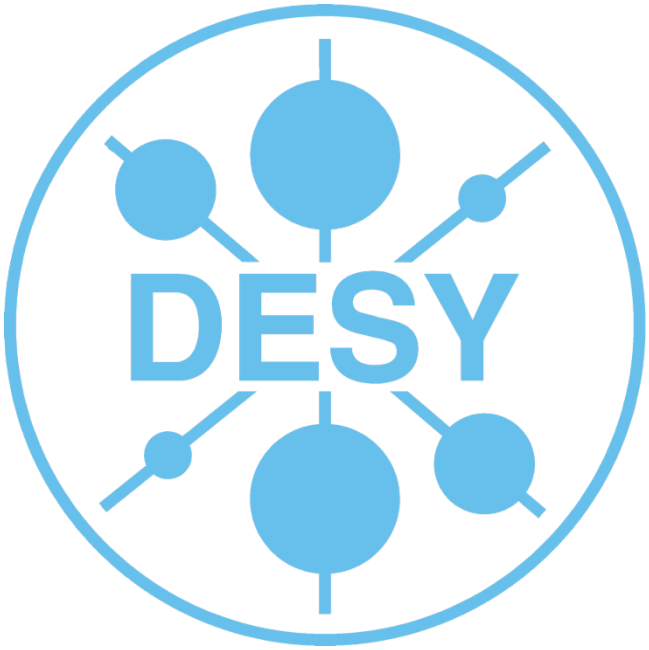
PAT::Tau Tutorial



Christian Veelken
christian.veelken@cern.ch

DESY Seminar 07/01/09

Objectives

The Aim

- Learn about the “Connection” between Stan Lai’s Talk on Tau id. and how (hadronically decaying) Taus look like in CMS software
- Learn how to access Variables used in Tau id. from CMSSW Data-Formats
- Learn where to find Documentation about Tau id. In CMS

The Path

- Read wiki Pages
- Browse cvs Repository
- Book and fill Histograms of Tau id. Variables

Taus in CMS (1/2)

- (hadronically decaying) Taus can be identified from Jets either reconstructed using Calorimeter information alone (➔ CaloJets) or from a combination of Calorimeter + Track information (➔ PFJets)
- In both cases, Tau id. proceeds by requiring all Particles within the Jet to be concentrated in a narrow Signal, surrounded by an “empty” Isolation cone
- In CMS, identifying hadronically decaying Taus means tagging Jets:

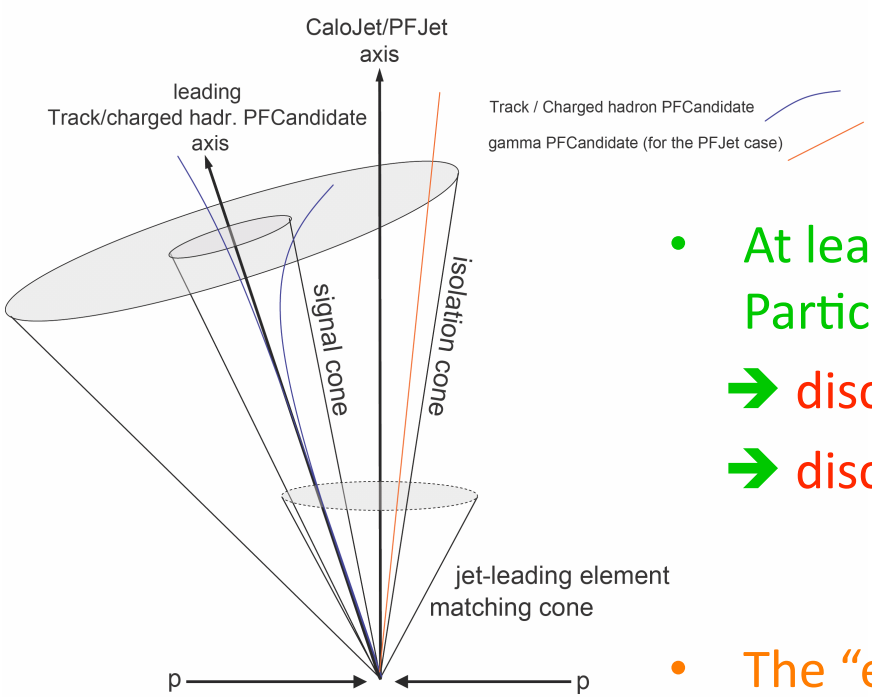
  Tau Jet : discriminator > cut

  QCD Jet : discriminator < cut

  In principle, the cut value can be tuned for your Analysis

  It turns out, however, that generally good Performance can be achieved for most Analysis using a “standard” Set of Tau id. requirements

Taus in CMS (2/2)



- At least 1 high P_T charged or neutral "leading" Particle is required within the Signal cone
  - ➔ discriminatuinByLeadingTrack
  - ➔ discriminationByLeadingPion
- The "emptyness" of the Isolation cone is quantified by either counting the Number of Tracks with $P_T > 1$ GeV or training a Neural Network
  - ➔ discriminationByIsolation
  - ➔ discriminationByTaNC ("Tau Neural Classifier")

Tau Data-Formats in CMSSW (1/2)

For knowing how to get access to Tau id. Variables, you need to know about the Inheritance Hierarchy of the PAT::Tau:

tricky Usage of C++ Templates, simplified in future PAT versions

- pat::Tau inherits from pat::Lepton<reco::BaseTau>
  DataFormats/PatCandidates/interface/Tau.h
- pat::Lepton<reco::BaseTau> inherits from pat::Object<reco::BaseTau>
  DataFormats/PatCandidates/interface/Lepton.h
- pat::Object<reco::BaseTau> inherits from reco::BaseTau
  (common base-Class for reco::CaloTau and reco::PFTau)
  DataFormats/TauReco/interface/BaseTau.h
- reco::BaseTau inherits from reco::RecoCandidate
  DataFormats/RecoCandidate/interface/RecoCandidate.h
- Reco::Candidate inherits from reco::LeafCandidate
  DataFormats/Candidate/interface/LeafCandidate.h
- Reco::LeafCandidate inherits from reco::Candidate
  DataFormats/Candidate/interface/Candidate.h
- Reco::Candidate inherits from reco::Particle
  DataFormats/Candidate/interface/Particle.h

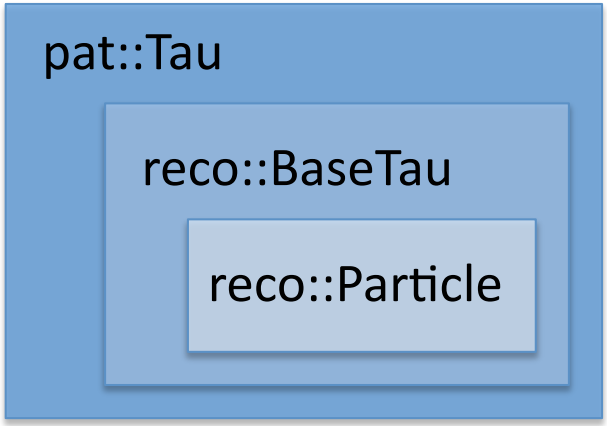
Puh!!

Tau Data-Formats in CMSSW (2/2)

Please don't get scared by all these levels of Inheritance

In practice, I find it sufficient (most of the time) to keep in mind the two Levels:

- pat::Tau
  (for Tau id. Variables and Discriminators)
- reco::Particle
  (for generic Particle information
  like Energy, P_T, η and ϕ)



An important thing to keep in mind, though, is that in case
a method is implemented in a base-Class and reimplemented in a derived Class,
the derived Class Implementation is the one that actually gets called

➔ when looking for a method to access a certain Tau id. Variable
start searching for it from the "top" of the Inheritance Hierarchy

Documentation

- Tau id. for PFJets described in the Analysis Note CMS AN-2008/043:
  "Tau reconstruction and identification with particle-flow techniques using the CMS detector at LHC"

http://cms.cern.ch/iCMS/jsp/openfile.jsp?tp=draft&files=AN2008_043_v12.pdf

- Tau id. wiki in CMSSW software Guide:

https://twiki.cern.ch/twiki/bin/view/CMS/SWGuidePFTauID

- CMSSW cvs Repository:

http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/CMSSW/

http://cmslxr.fnal.gov/lxr/search

- Particle-Flow & Tau id. Hypernews Forum:

https://hypernews.cern.ch/HyperNews/CMS/get/eflow.html

Now it is your Turn!

- Open:
  - PhysicsTools/PatExamples/plugins/PatTauAnalyzer.h
  - PhysicsTools/PatExamples/plugins/PatTauAnayzer.cc
- Add Histograms for:
  - η and ϕ of Tau Jet
  - P_T of leading Track
  - discriminatorAgainstElectron
- Compile and Test your Changes:

```
cd PhysicsTools/PatExamples
cmsenv
scramv1 b
cmsRun analyzePatTau_fromPatTuple_cfg.py

root patTau_idEfficiency.C
new TBrowser
```