The PAT jet cleaner
Christian Autermann  (Universität Hamburg)

• The PAT concept
• Jet Cleaner
• External jet ID
• Custom jet selections
The 'PAT' is:
Roberto Tenchini, Petar Maksimovic, Steven Lowette, Giovanni Petrucciani, Frédéric Ronga, Wolfgang Adam, Volker Adler, Christophe Delaere, Luca Lista, Slava Krutelyov, Jeremy Andrea, ...

More information:
https://twiki.cern.ch/twiki/bin/view/CMS/PatLayer1
http://indico.cern.ch/categoryDisplay.py?categId=1402

The Code:
http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/CMSSW/DataFormats/PatCandidates/
http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/CMSSW/PhysicsTools/PatAlgos/
http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/CMSSW/PhysicsTools/PatUtils/
The marriage of flexibility and user friendliness

- a multi-layered approach was needed to provide both maximal flexibility and user-friendliness within the constraints of the EDM

- event interpretation
  - “Layer 0”: cleaning and disambiguation -> event interpretation + additional analysis-level tasks (e.g. MC matching)
  - “Layer 1”: creation of big objects that collapse externally associated information -> no algorithmic tasks

- event hypothesis
  - “Layer 2”: event hypothesis dependent tasks -> provide possibility for re-tuning event interpretation

- analysis
  - “Starter Kit”: for data exploration and plotting
  - “paste-your-analysis-here”
The PAT Layered Approach

RECO/AOD

PAT Layer-0
  cleaning
  disambiguation
  analysis level associated info

PAT Layer-1
  de-associate into PAT objects

user analysis

Starter Kit

Layer-2 (event hypothesis)

Steven Lowette
Overview PAT Level 0

**PAT Level 0:**

- Association management
- **Object ID and cleaning, duplicate removal**
- Disambiguation
- additional associations
Object-dependent ID and cleaning

- duplicates removal
  - choose electron in case of shared supercluster (or track)
    - should move to/come from the egamma POG
  - `GenericDuplicateRemover` tool (Giovanni Petrucciani)
    - not PAT specific, so move to PhysicsTools?

- object ID variables
  - produce if not on AOD yet
  - try to work towards similar output formats (e.g. isolation)
  - interface multiple algorithms, using POG recommendations as defaults

- cleaning
  - technically: a cleaner (EDProducer) calls a FW-independent selector
    - this factorizes algorithmic code from the rest
    - requirement for framework independence not there yet though...
  - use the ID variables to choose “clean objects”
  - to be used in the next step of event-wide disambiguation
  - need to provide possibility to also keep “non-clean”, e.g. electrons inside jet
Jet Cleaner

- Templated in order to use CaloJets and PFJets
- Duplicate and Overlap removal similar to other PAT cleaners
- Selection of generic jets by external packages possible, if the results are provided by ValueMaps:
  1. JetRejectorTool provides a jet likelihood ([twiki.cern.ch/twiki/bin/view/CMS/JetRejectorTool](twiki.cern.ch/twiki/bin/view/CMS/JetRejectorTool))
  2. Other jet ID tools can be implemented similarly
- Custom selection by standalone C++ algorithms, e.g. CaloJetSelector ([PhysicsTools/PatUtils/interface/CaloJetSelector.h](PhysicsTools/PatUtils/interface/CaloJetSelector.h)):

```cpp
caloJet = dynamic_cast<const reco::CaloJet*>(&Jet);
if (caloJet) result = CaloJetSelector_->filter(*caloJet);
```
Jet Cleaner usage

- No jet cleaning
  
  ```
  PSet selection = { string type = "None" }
  ```

- Selection of generic jets by external packages, e.g.:
  
  ```
  PSet selection = {
    string type = "JetMETLikelihood"
    double value = 0.8
  }
  ```

- Custom user CaloJet (or PFJet) selections:
  
  ```
  PSet selection = {
    string type = "custom"
    double EMFmin = 0.0
    double EMFmax = 1.0
    double Etamax = 2.5
    double N90min = 0.0
    ...
  }
  ```

Implemented cleaning vars under study:

- EMF, HadF, Eta, N90, Ntower,
- $\frac{E_{\text{max}}}{E_{\text{jet}}}$, $\frac{\sum E_{\text{tower},i} \cdot \Delta R_{\text{jet},i}}{\sum E_{\text{tower},i}}$, $E_{T, \text{jet}}$, area
Electrons use: AssociationMap<ElectronCollection, ElectronIDCollection>

```cpp
class ElectronID
{
public:
    ElectronID(bool cutBasedDecision=-1,
               double likelihood=-1,
               double neuralNetOutput=-1.);
    cutBasedDecision_(cutBasedDecision),
    likelihood_(likelihood),
    neuralNetOutput_(neuralNetOutput) {}

    bool cutBasedDecision() const {return cutBasedDecision_;}
    double likelihood() const {return likelihood_;}
    double neuralNetOutput() const {return neuralNetOutput_;}

private:
    bool cutBasedDecision_;
    double likelihood_;   
    double neuralNetOutput_; 
};  (AnalysisDataFormats/Egamma/interface/ElectronID.h)
```

- Doesn't exist for jets (yet?)
- Is it necessary?
- Best: general selector? (Twiki: SWGuideGenericSelectors)
My jet-ID study plans

Fourier transformation:

\[ \hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int f(t) e^{-i\omega t} dt \]

- Is it possible to identify noise in the fourier-transformed energy-deposition distribution?
- What effect has noise/zero-suppression?
- Will look different than in these pictures!

- Could serve as additional variable in a cut based jet-ID
Summary

- PAT will be the analysis framework
- First version is operational and ready to test
- Many tools still under development
- Ideal testbed to develop the Jet-ID, together with the PAT
- I will integrate the jet-ID into PAT

→ POG must provide interface to jet-ID that is flexible enough also for future developments!