Jet Software: From the User's Point of View

Kerstin Perez

ATLAS Hadronic Calibration Workshop

23 June 2009

COLUMBIA UNIVERSITY

With many, many thanks to Pier-Olivier Deviveiros, Pierre-Antoine Delsart, Peter Loch, Tancredi Carli, …

Introduction

- This talk will serve as an intro to the detailed jet software tutorial that will take place on Saturday
- Will focus on *what* you can do, not necessarily *how* you can do it
- Outline:

Jet Building

- Input to jet finding → Constituents
- Jet finding → Algorithms
- Signal states
- Associated data

Jet Calibration

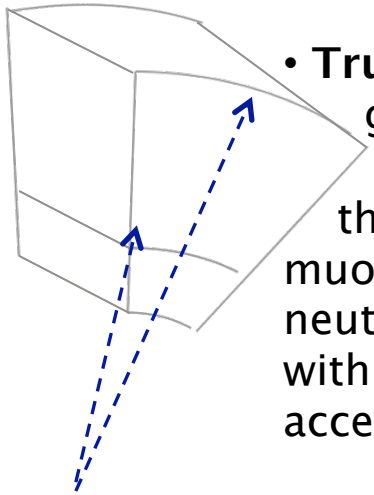
- JetCalibTool → basics for developers and users

Data Formats

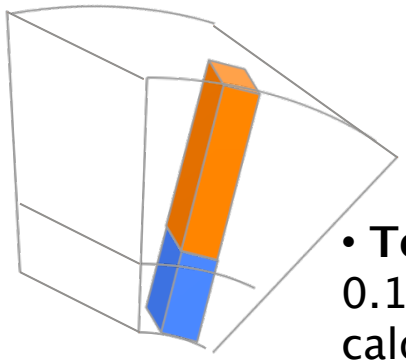
- AOD and ESD
- D1PD and D2PD

Jet Building: Constituents

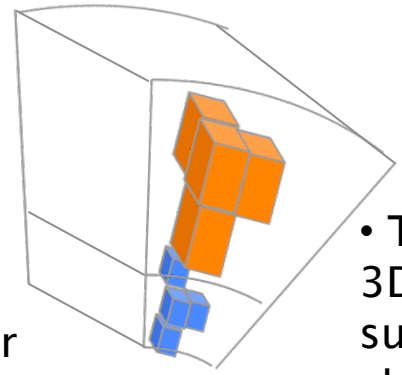
These are the inputs to jet finding:



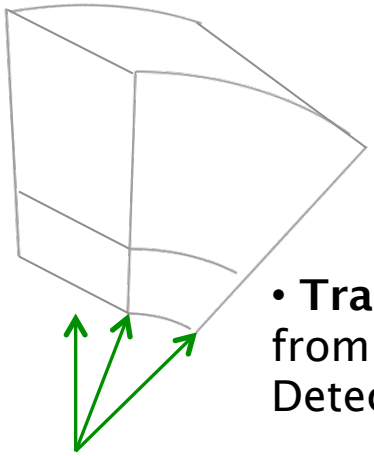
• **Truth**: generator particles that aren't muons or neutrinos, within η acceptance



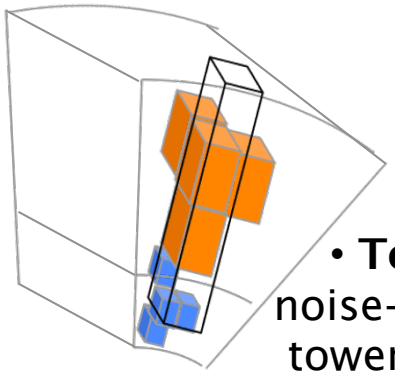
• **Towers**: 0.1x0.1 calorimeter towers



• **TopoClusters**: 3D noise-suppressed clusters of cells



• **Tracks**: from Inner Detector



• **TopoTowers**: noise-suppressed towers built from topoclusters

Can use *JetConstituentIterator* to access constituents, all the way back to *CaloCells*

Jet Building: Algorithms

- In the jet software, a jet algorithm is a sequence of tools, containing a jet finder
- FastJet Library
  - standard definitions of algorithms
  - contains lists of tools for ATLAS Cone, SISCone, Anti-kt, and Kt jet finders
- For example, configuring ***make_StandardJetGetter('Cone',0.4,'H1Topo')*** will execute:

JetInitialEtCut	Select input with minimum E_T
JetConeFinder	Actual jet finding algorithm, with R=0.4
JetConeSplitMerge	Split/merge procedure
CellCalibrator	H1 Calibration procedure
JetFinalEtCut	Discard jets below minimum E_T
JetSorter	Sort jets in decreasing E_T
EinSampling	Compute energy per EM sampling

- Output is a collection of jet objects with 4-momenta written to StoreGate

Jet Building: Signal States

3 Possible jet **SIGNAL STATES**:
4-momentum of reconstructed jet at different scales.

- **JETEMSCALE** ➔ at the raw EM calorimeter scale
- **JETCONSTITUENTSCALE** ➔ right after jet-finding
- **JETFINALSCALE** ➔ after final jet energy scale corrections

By default, the 4-momentum is at the **FINAL** scale. Can switch between signal states using the same jet object. For example:

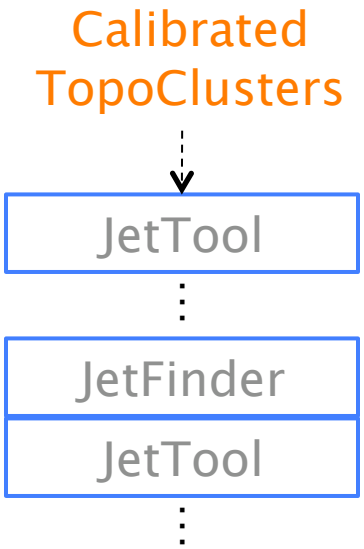
- *myJet→setSignalState(P4SignalState::JETEMSCALE)*
- Now *myJet→e()*, or any other 4-momentum component, will now return value at EM scale

Jet Associated Data

What else can you get from a jet?

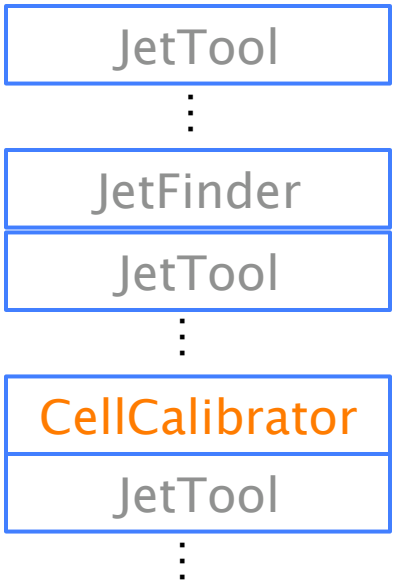
- Jet Moments (Shapes)
  - jet moment = a string name, a float value
  - Accessed by *myJet->setMoment(string, float)* and *myJet->getMoment(string)*
  - For example: energy per sampling, Cone split/merge ratio, width, b-tagging likelihood
- Jet Association
  - Associate a muon, electron, track, etc. to a jet
- Jet Tag Info
  - Associate complex info to a jet, for example b-tagging, τ-tagging, etc.

Jet Calibrations



Local Calibration:
use calibrated topoclusters as input (LC)

Global Calibration:
jets calibrated after jet finding (H1)



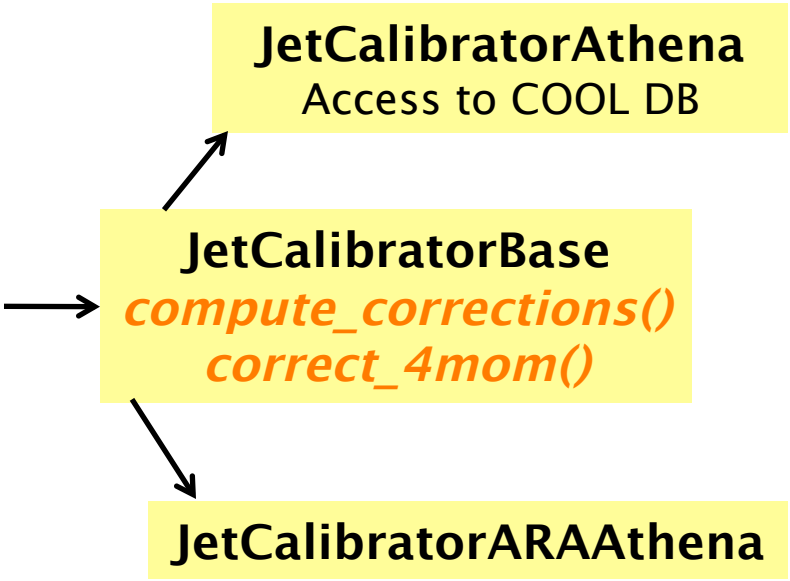
- Derivation of corrections handled by *JetCalib* (soon to be *JetCalibBuilder*)
- Calibration constants stored in offline COOL database
- But how do we handle the application of multiple calibrations? Custom calibrations? Versioning? Store and retrieve systematic uncertainties?...

JetCalibTool

- Common framework for applying calibrations to jets
- Calibration history is stored in *Jet::m_jetAuthor*
- Protection to prevent applying the same calibration twice!

IJetCalibrator
initialize()
finalize()
initializeEvent()
finalizeEvent()
calibrate()
get_calibrated_copy()
store_calibration_correction()
get_calibration_correction()
get_calibration_corrections()
get_calibration_error()
get_calibration_errors()
get_calibration_tag()
set_calibration_tag()

- Calibrate jet or store corrections as moments
- Retrieve systematic uncertainties
- Append a tag for each calibration to the jet author string



JetCalibTool: Developer

Many correction already available:

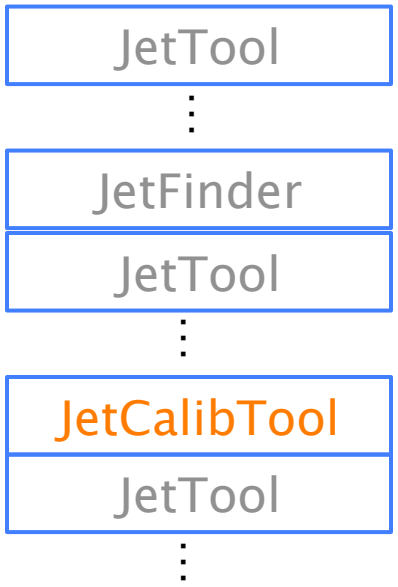
Scale
- H1 Cell Weighting
- Numerical Inversion from MC
- JetMoments from specific jet moment
- Offset Correction for pile-up
- Gamma+Jet from γ-jet balance
- MET Projection Fraction from MET projection in γ-jet events
- Layer Weighting weighting for each calorimeter layer

Resolution
- CaloFractions from fraction of energy in different layers
- Ntrk from track multiplicity
- JetVertexFraction for pile-up

To write your own calibration, just implement:

- ***compute_corrections()***
  Given a Jet, calculate all the correction factor(s) needed and store them in a vector
- ***correct_4mom()***
  Given the vector of correction factors, how would the Jet 4-momentum be changed

JetCalibTool: User



JetCalibTool can be used in the reconstruction sequence just like other tools

- *InplaceCorrection* property determines whether the jet is corrected, or the calibration constants are only stored as JetMoments

How to handle configuring many different calibrations?

- *JetCalibSequence* = a *JetCalibTool* containing many *JetCalibTool*s
  - Stores all calibration constants in moments, and can either record the name of the sequence ('StandardH1Calib') or the whole sequence of names ('H1Cell_H1f')
- Given a collection type ('Cone', 0.4, 'Tower'), the *JetSequenceManager* will gather the set of officially supported calibration schemes (or a custom one) and allow them to be applied at analysis level
- Official calib sequences are configured and built in dedicated functions

Data Formats

- ESD's are for special taks, will have restricted access
- AOD's used for physics analyses
- Jets are stored in jet collections, named according to their jet-finding algorithm, size parameter, and input

Data Type	Appx. Size	Location	Contains	Can Reconstruct
Event **S**ummary **D**ata	0.8MB/ event	2 Tier1's	CaloCell, CaloCalTopoCluster, Cone4(H1Topo/H1Tower/Truth), Cone7H1Tower **Planned: include, then switch to, AntiKt**	Any jet algorithm!
Analysis **O**bject **D**ata	0.2MB/ event	All Tier1's	CaloCalTopoCluster, Cone4(H1Topo/H1Tower/Truth), Cone7H1Tower	Locally-calibrated cluster jets, track jets

Data Formats

- **Jet performance DPD** : same calorimeter information as ESD, but with reduced number of events
  - ~10% of AOD size, ~2% of ESD size
  - All calorimeter cells
  - Distributed to Tier2's
- These will be the **main** data sets for jet and MET performance studies

DPD Name	Trigger Stream	Purpose
SingleEl	Egamma	MET performance, W/Z, top pairs
SingleMu	Muon	MET performance, W/Z, top pairs
PhotonJet	Egamma	Jet calibration
MinBias	MinBias	Jet, MET, single track performance
CaloJET	Jet	Di-jet performance and calibration
LargeMET	Jet	MET performance

- Can also produce specialized **D2PD**'s with several jet collections and several calibration schemes for easy analysis and early feedback from physics groups

Summary

- Jet Building is handled by a series of tools configured during reconstruction
- The FastJet Library contains a standard list of tools for ATLAS Cone, SISCone, Anti-kt, and Kt jet finders
- Jet Calibrations can be handled during reconstruction or analysis by the JetCalibTool
- JetCalibSequence and JetSequenceManager help configure standard calibrations
- Performance and calibration studies will be performed on DPDs
- More information during the tutorial on Saturday!