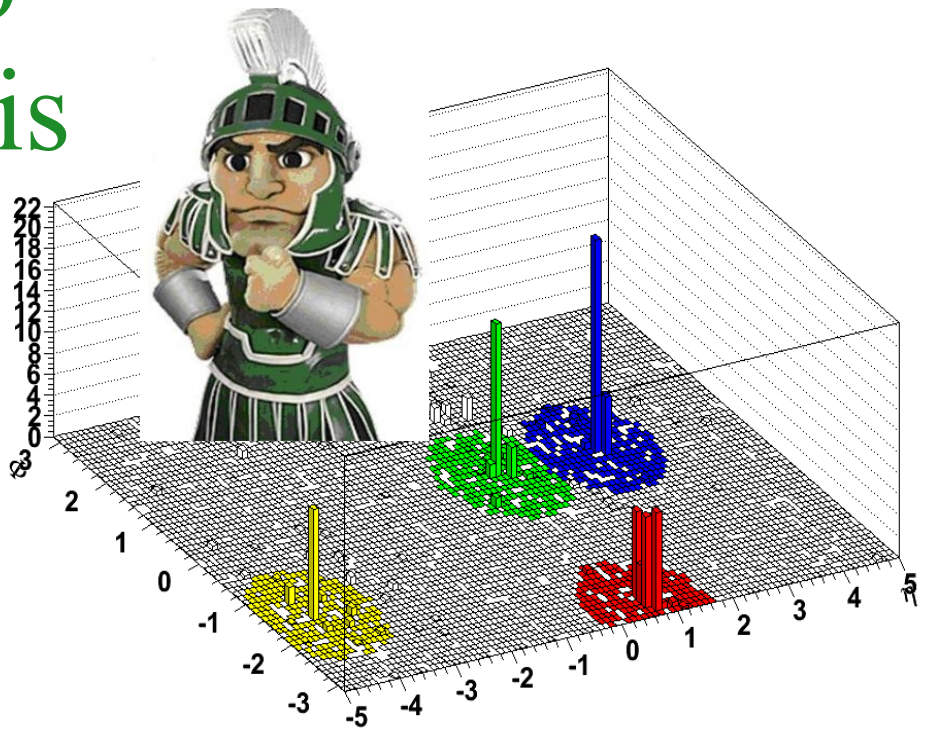
SpartyJet: A Suite of Tools for Jet Sub-Structure Analysis

Joey Huston
Brian Martin
Chris Vermillion

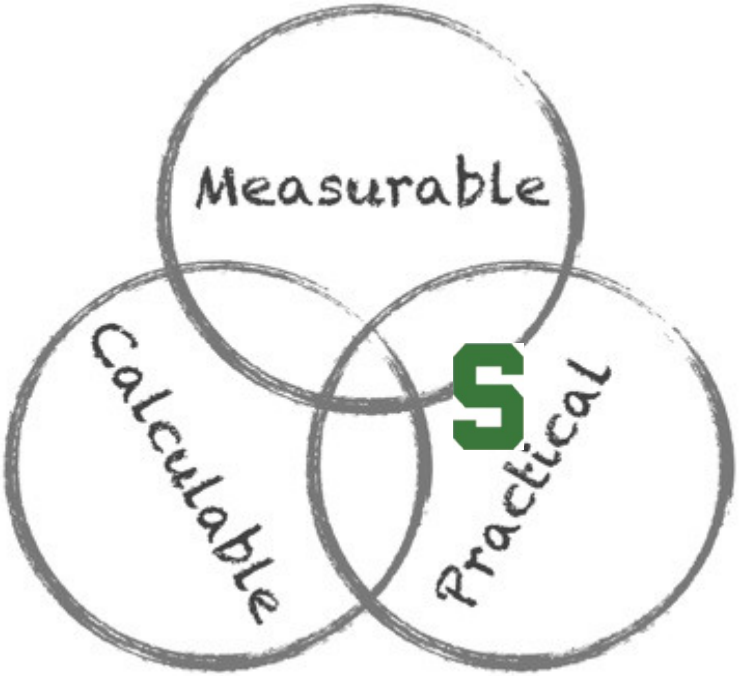


January 14th, 2011
Boston Jet Physics Workshop

Motivation

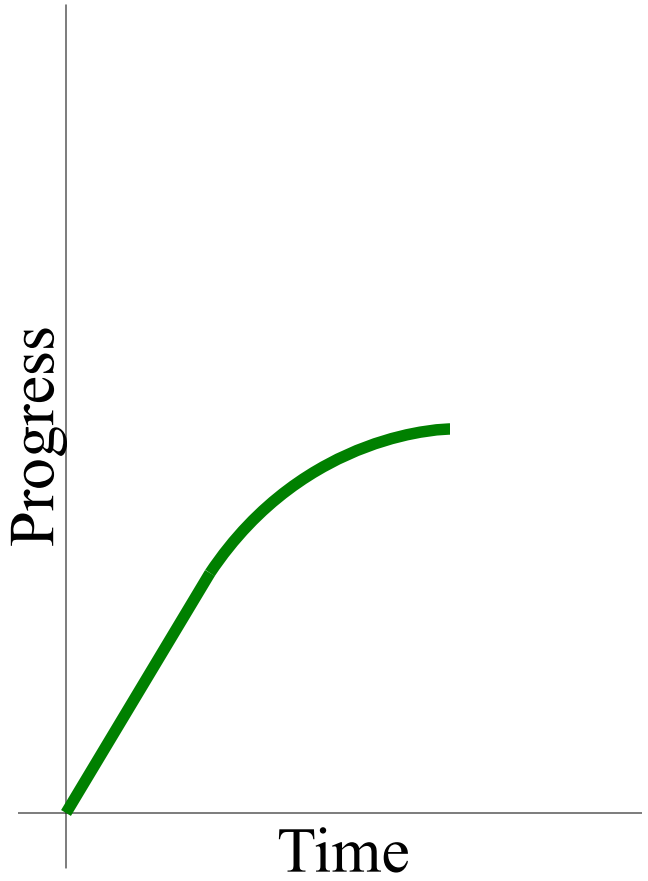
Common Interfaces for Jet Tools

- Thanks to FastJet we already have common implementation of core jet finding
- SpartyJet aims to allow Theorists and Experimentalists to both utilize the same tools
- Focus on:
  - Ease of use
  - Modular, tool-design
  - Native handling of multiple types of input
  - Interactive analysis
  - Adaptability



Evolution of SpartyJet

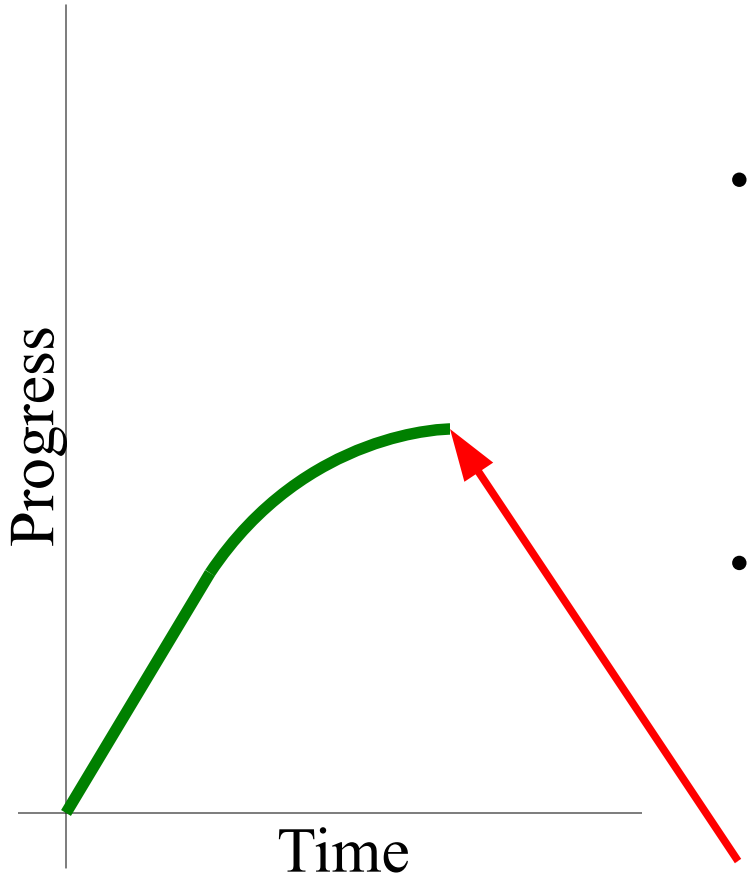
Authors: *Joey Huston, Pierre-Antoine Delsart*
Kurtis Geerlings, Brian Martin



- Originally conceived as a jet finding suite (similar to FastJet)
  - Contributed MidPoint Cone Algorithm
  - Interfaced with FastJet, Pythia, CDF, D0, and ATLAS algorithms
- Grew into a set of jet-related interfaces in addition to core jet finding
  - Jet tools, multiple inputs
  - Analysis GUI

Evolution of SpartyJet

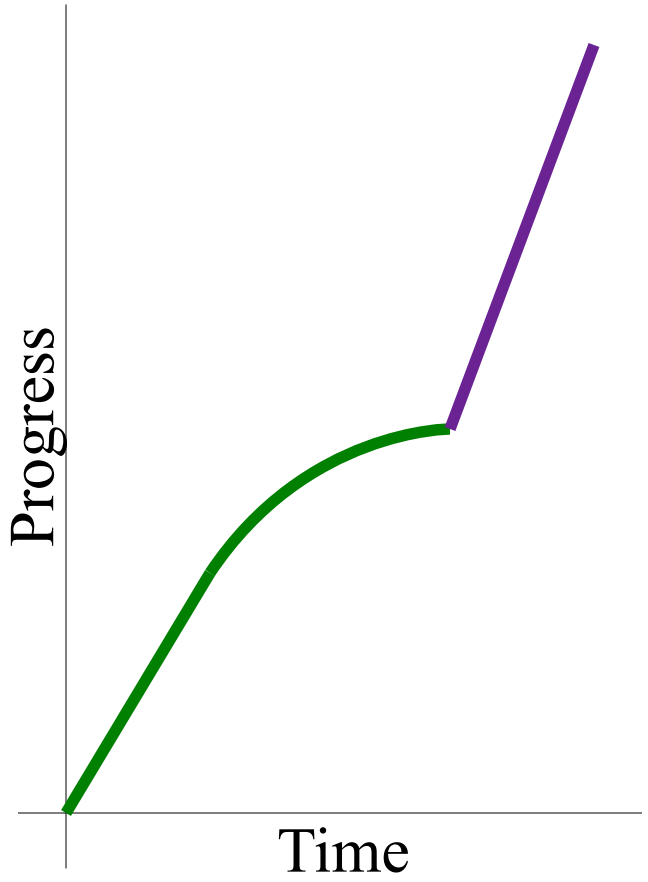
Authors: *Joey Huston, Pierre-Antoine Delsart*
Kurtis Geerlings, Brian Martin



- Originally conceived as a jet finding suite (similar to FastJet)
  - Contributed MidPoint Cone Algorithm
  - Interfaced with FastJet, Pythia, CDF, D0, and ATLAS algorithms
- Grew into a set of jet-related interfaces in addition to core jet finding
  - Jet tools, multiple inputs
  - Analysis GUI

2010 UW Jet Physics Workshop

Evolution of SpartyJet



Authors: *Joey Huston, Pierre-Antoine Delsart*
Kurtis Geerlings, Brian Martin,
Chris Vermillion

- And then there is a discontinuity:
  - Named *Chris*
- Transformed into a coherent jet analysis framework
  - Full integration with FastJet
    - No more core jet finding
  - Collection of State-of-the-art tools for substructure analysis
  - Ability to expand quickly to encompass new tools

Current SpartyJet

- Spartyjet is built on two external pieces of software
  - ROOT:
    - Provides one type of input
    - Output is in ROOT ntuples
    - Graphics/histogramming
  - FastJet:
    - Core jet finding
    - Underlying containers
  - There are additional interfaces for other pieces of external code

SpartyJet Architecture

InputMaker

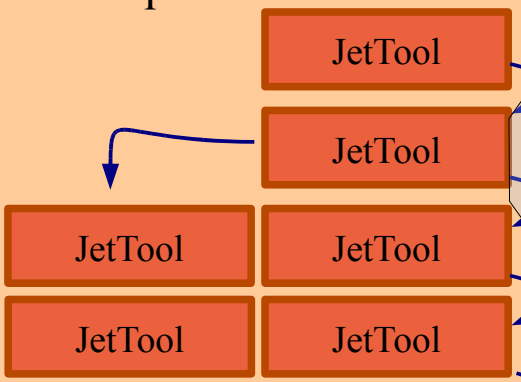
Reads an input collection of 4-vectors and converts to an **initial jets list**

- **ROOT ntuple**
- **StdHep**
- **LHE**
- **HepMC**
- **CalCHEP**
- **Pythia (directly)**
- **Text**

JetAlgorithm

Contains all the tools that can act on a set of pseudojets/jets/subjets

Feed initial jets list into sequence of JetTools



JetTool

Modifies jets in sequence.

- Jet Algs
- Cuts
- Filters
- Re/Declustering
- Jet Moments

NtupleMaker

Handles ntuple creation of jet results

SpartyGUI

For interactive analysis

MICHIGAN STATE UNIVERSITY

The SpartyJet Module: JetTool

Worker of SpartyJet:

- Performs all the actions on the Jets
- A traditional jet algorithm is just another tool in SpartyJet
- Act on the jets before/after Jet finding
- Create and store jet moments, event moments
- Examples:
  - Add minbias events (pileup); ghost (pt~0) particles
  - SelectorTools – For filtering input and output based on kinematics or pdgId
  - NegEnergyTool – handles Jets with negative energy so that they behave in jet finding
  - JetMomentTools – Jet Areas, Substructure moments
  - EventMomentTools – EventPt density, thrust
  - And....substructure tools...

JetTool

Modifies Jets in sequence

The SpartyJet Output: Ntuples

NtupleMaker

Handles ntuple creation of jet results
- ROOT TTrees

SpartyJet stores jet information in ntuples:

- Creates/Fills branches for each component of the jet four vector (including the jet mass)
- Has option to retain the original input (with particle ID) and jet matching indices, so full jet constituents for every jet is efficiently stored
  - Trivial to make jet shape functions from this information
- Creates/Fills branches for each Jet moment created by the JetTools
- User can choose whether branches are stored as arrays or vectors

Jet Substructure with SpartyJet

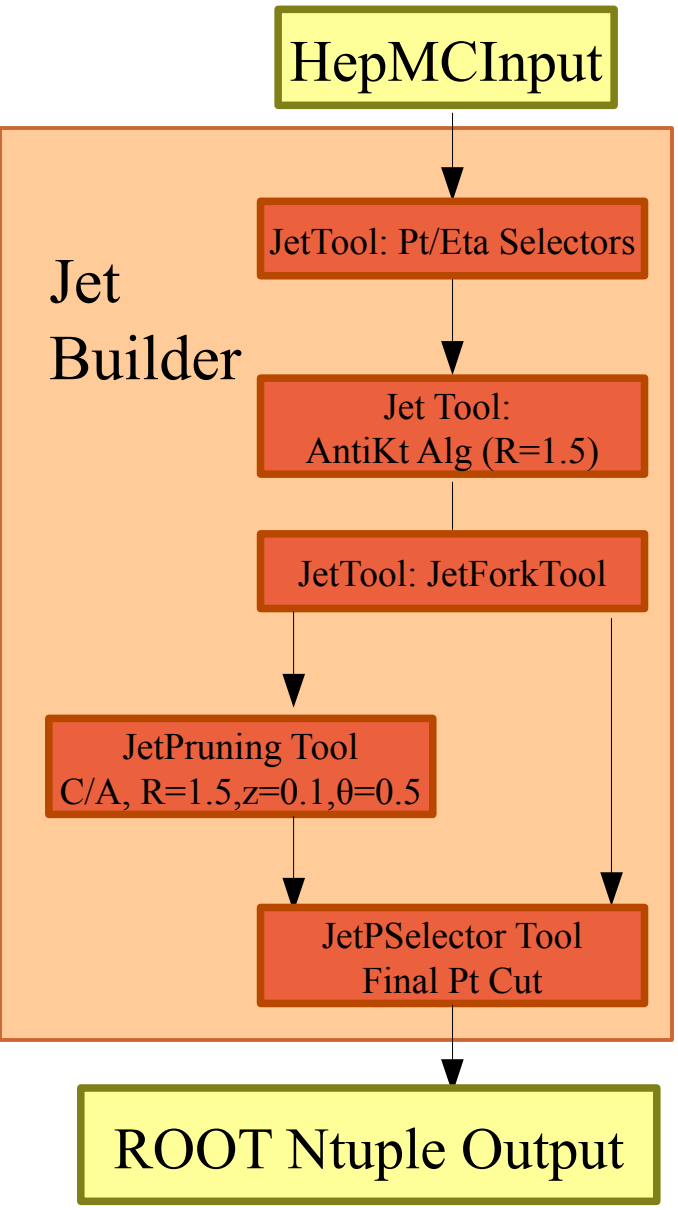
Due to its modular nature, SpartyJet is ideally suited for Jet Substructure Analysis

- Most jet substructure methods require running over input multiple times
  - In SpartyJet this is merely a sequence of two JetTools
- Many substructure tools have been directly implemented in SpartyJet
  - Others are shipped with SpartyJet and interfaced for use with SpartyJet

Current Tools (* = via plugin interface)

- Pruning – **FastPruneTool** described in arXiv:0912.0033
- Trimming – **QCDTrimmingFast** described in arXiv:0912.1342*
- Hopkins Tagger – **JHTopTagger** described in arXiv:0806.0848*
  - With the pruning and subjet mass implemented directly; the tagger interfaced
- BDRS Filter – **BDRSFilterTool** described in arXiv:0802.2470
  - With Mass Drop implemented directly; also uses G. Salam's filter class
- WTagger - **WTaggerTool** described in arXiv:1012.2077
- Ysplitter, JetAreas, PtDensity calculation – Fully-integrated with fastjet

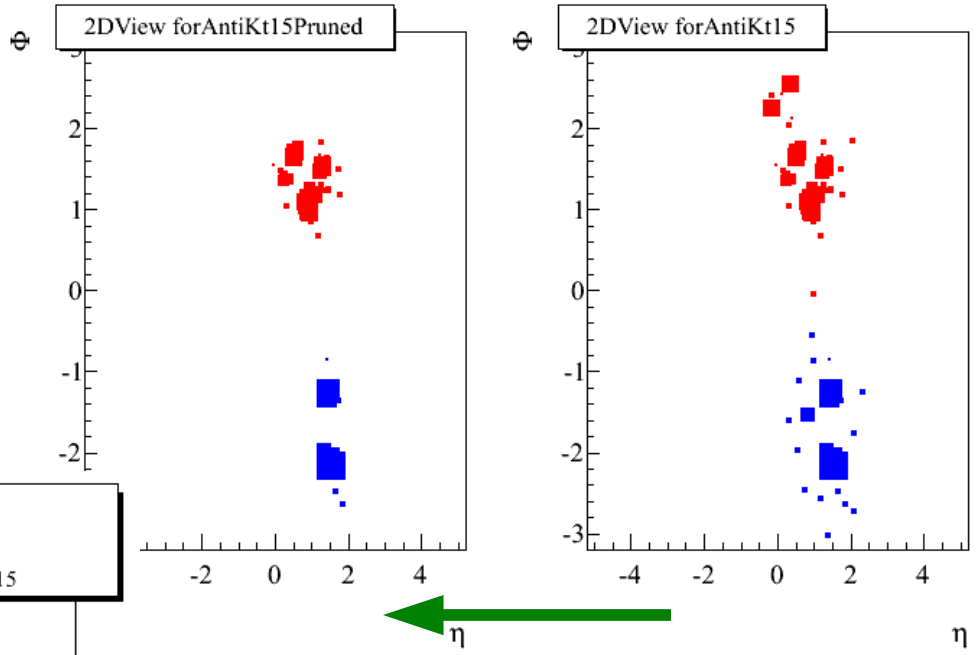
Jet Substructure Example

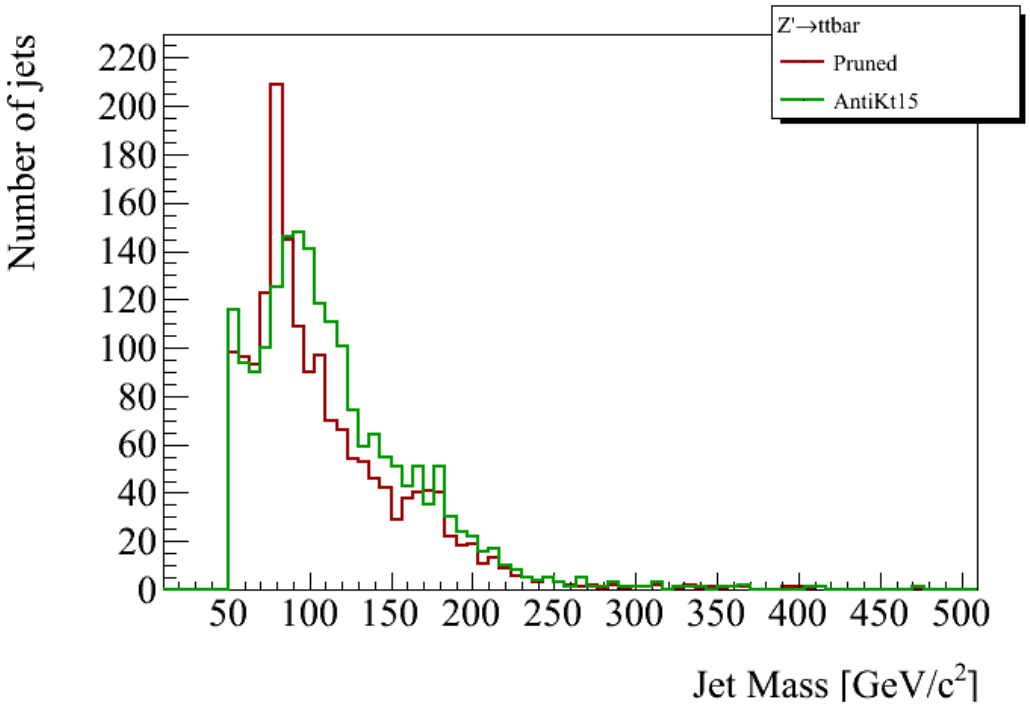



```
builder = SJ.JetBuilder()
SJ.HepMCInput('../data/Zprime_ttbar.hepmc')
builder.configure_input(input)

builder.add_default_alg(SJ.FastJet.FastJetFinder('Ant
iKt15', fj.antikt_algorithm, 1.5)

#Now the JetTools
builder.add_jetTool_front(SJ.JetEtaCentralSelectorToo
l(-4.9, 4.9))

parent = SJ.ForkToolParent('AntiKt15Parent')
builder.add_jetTool(parent, 'AntiKt15')
child = SJ.ForkToolChild(parent, 'AntiKt15Pruned')

big_CA_def =
fj.JetDefinition(fj.cambridge_algorithm, 3.14*0.5)
CAprune_tool =
SJ.FastJet.FastPruneTool(big_CA_def, 0.1, 0.5)
builder.add_jetTool(CAprune_tool, 'AntiKt15Pruned')

builder.add_jetTool(SJ.JetPtSelectorTool(10*input.get
GeV()))

builder.configure_output('SpartyJet_Tree','out.root')
builder.process_events(10)
```

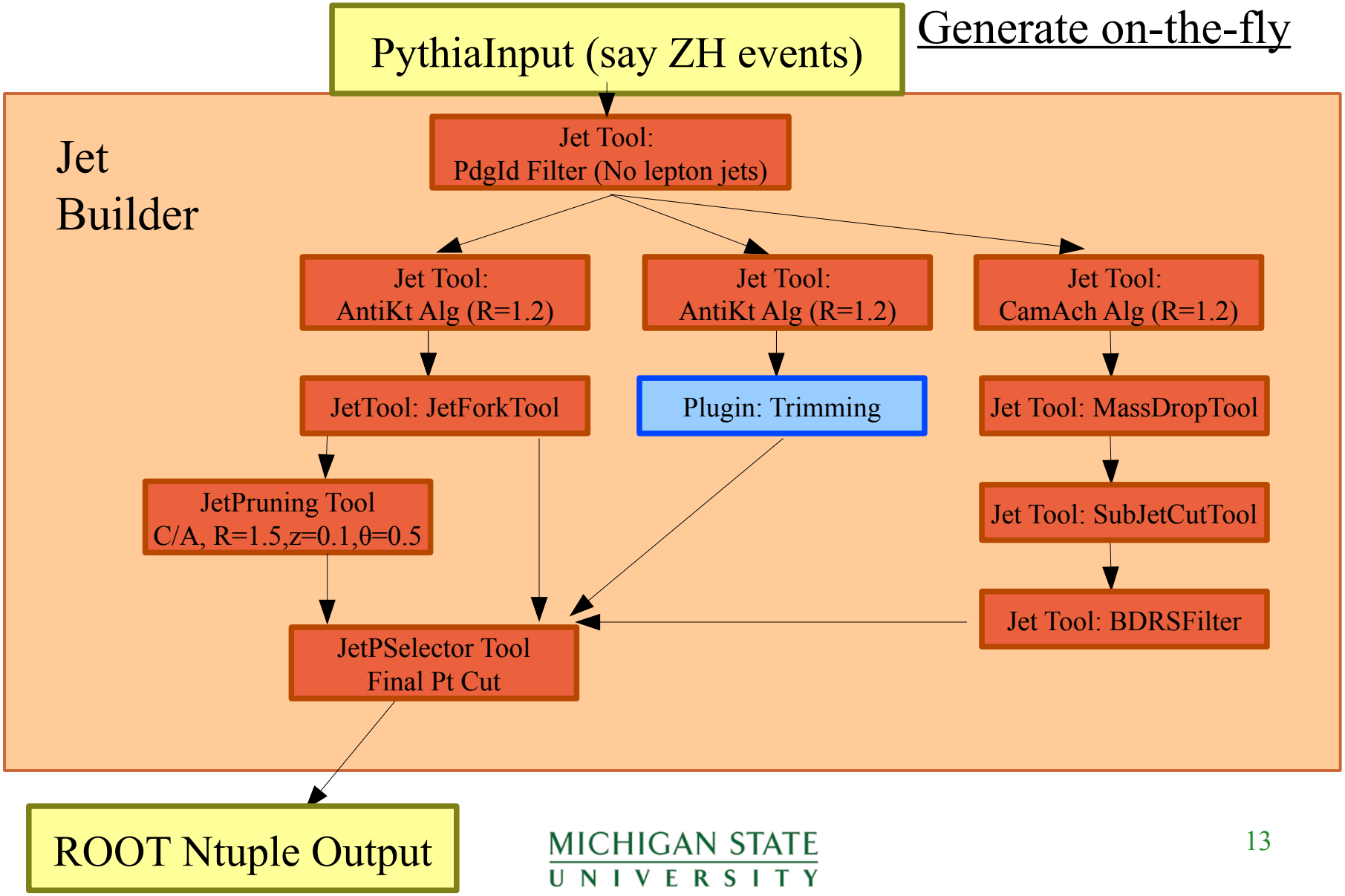
Jet Substructure Example

- The results can be viewed from the ntuple or running the GUI on the ntuple





A More Involved Jet Substructure Example

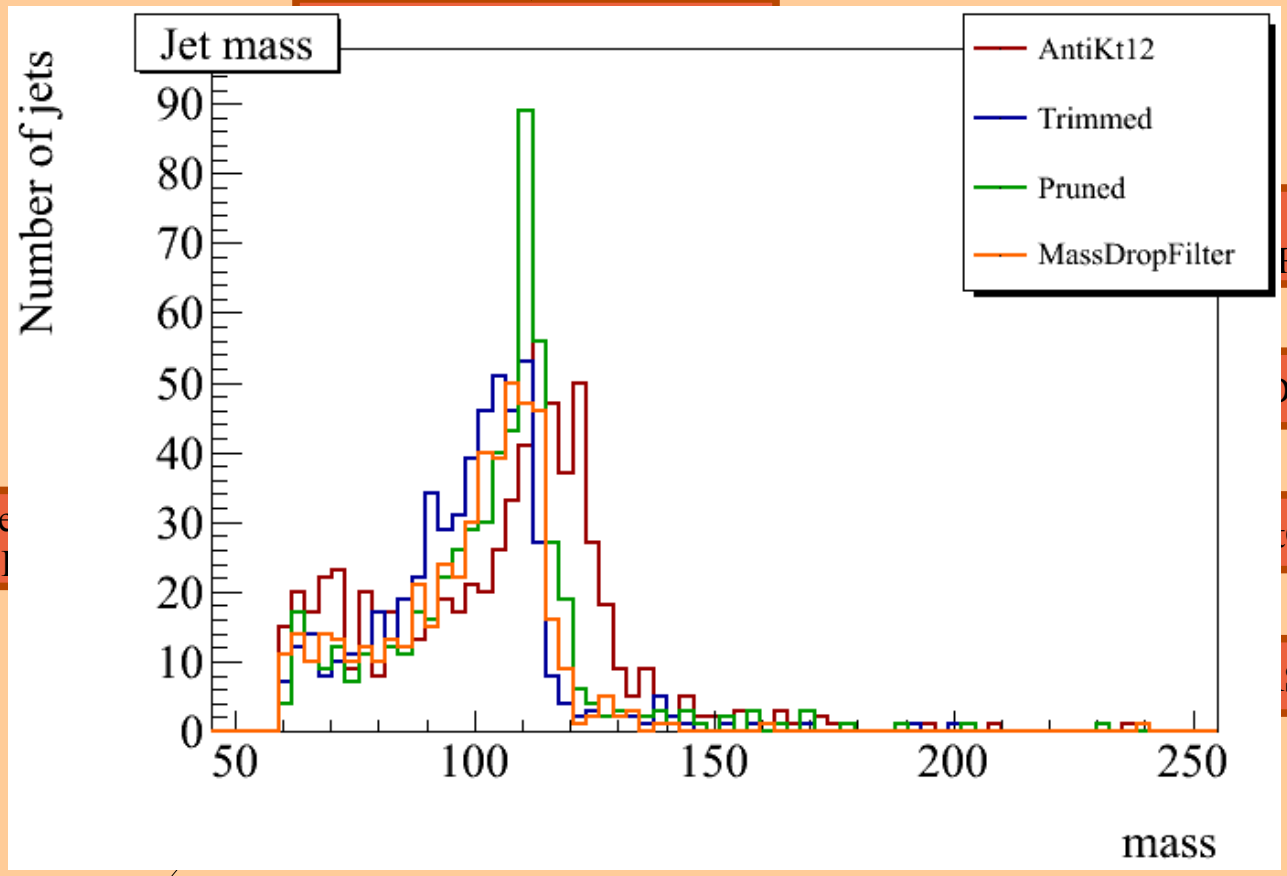


A More Involved Jet Substructure Example

PythiaInput (say ZH events)

Generate on-the-fly

Jet Builder



R=1.2)

DropTool

CutTool

SFilter

ROOT Ntuple Output

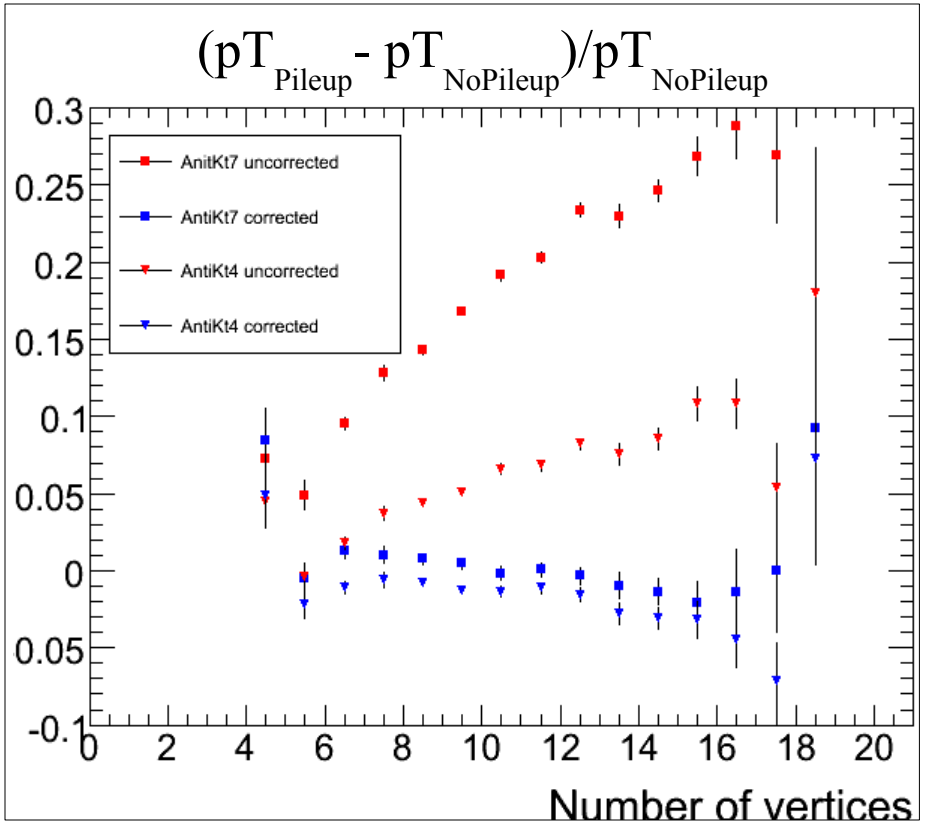
MICHIGAN STATE UNIVERSITY

SpartyGUI

Use Case in ATLAS: Jet Area Correction

- Method of removing pileup/underlying event contribution to jets
  - Measure the diffuse pT density in event
  - Correct jet pT according to the jets area
- Performed ATLAS studies exclusively with Locally Calibrated TopoClusters
- Unfortunately could not apply to Jet measurements (like W+jets cross-section)
  - Approved Jet calibration was derived from EM-scale topoclusters with no correction

- Originally proposed by Salam and Cacciari arXiv:0707.1378

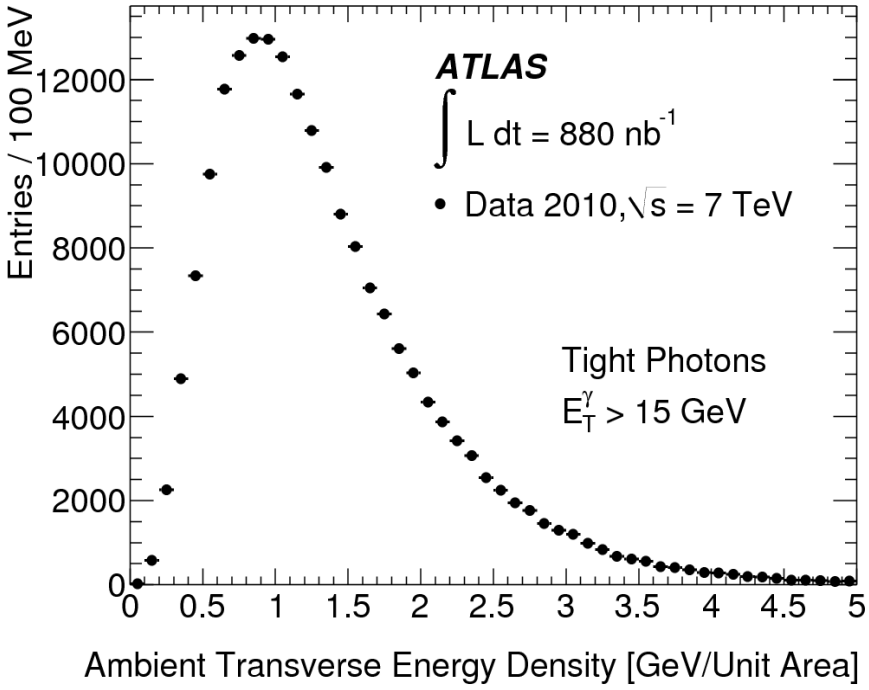


Use Case in ATLAS: Jet Area Correction

- However, method adopted by non-jet measurement
  - "Inclusive isolated prompt photon cross section"
  - Used this method to further correct the photon isolation cones for UE and pileup on an event-by-event basis

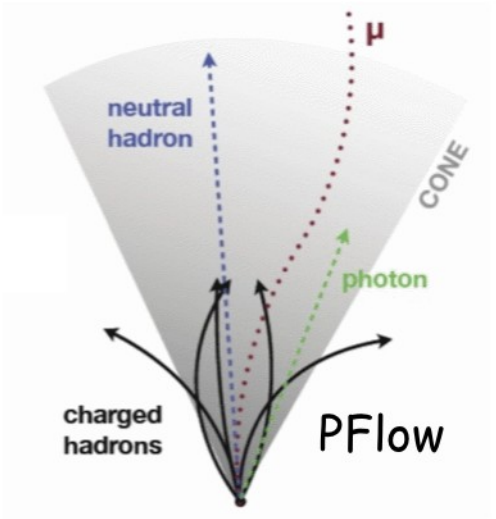
- So theoretical methods can be applied quickly in an environment like SpartyJet
  - Either for the measurement themselves or as a means of motivating adoption

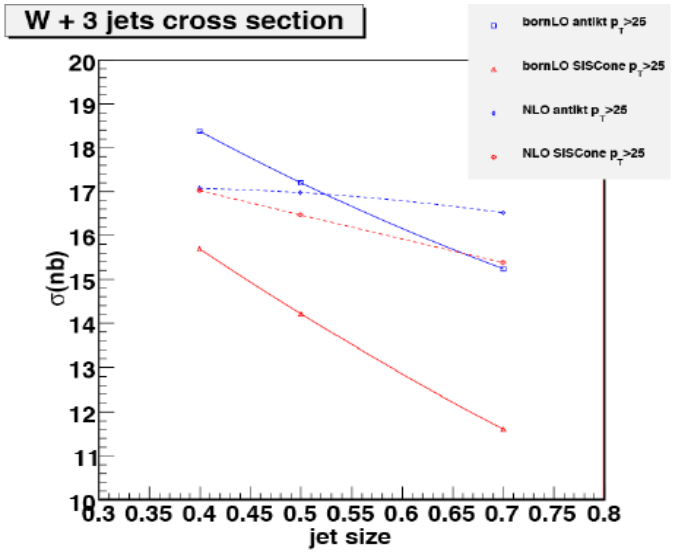
From prompt γ Cross-section arXiv:1012.4389

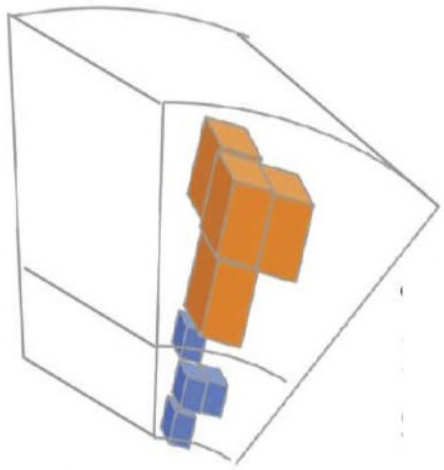


A Second Use Case for ATLAS/CMS

- From Leonard and Peter, jet-independent, calibrated jet inputs are near
  - These inputs are portable (read: ntuples) allowing factorization (sorry...) of jet finding from the monolithic detector software
- From theory cross-sections we expect large differences in jet cross-sections due to algorithmic effects
  - These predictions are increasingly available in (quasi-) jet algorithm independent format (and in ROOT ntuples)

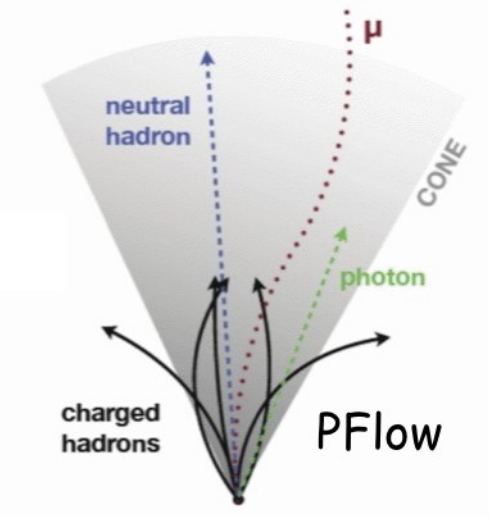


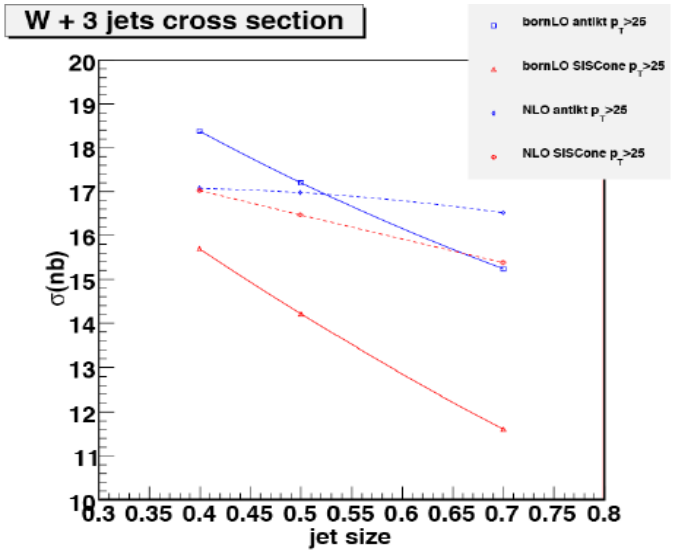


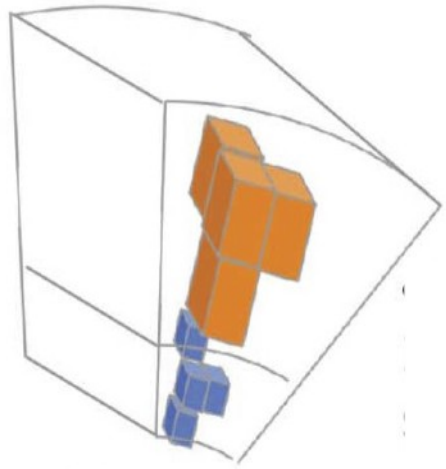


A Second Use Case for ATLAS/CMS

- These two reasons motivate a simple means of jet finding that is trivial to implement: Take ntuple, write 10 lines of python, make jets
  - On the detector jet side, the residual corrections to bring the input scale to the jet scale are ~10% and falling
  - On the MC jet side, there is no reason why multiple algorithms cannot be used regularly already to assess potential needs to investigate jet alg. dependence (SpartyJet can do even AND odd cone sizes as well)







Getting/Building SpartyJet

SpartyJet is available on HEP Forge

Requirements: linux or mac OSX, ROOT >=5.18

- Get tarball from website, setup environment and compile:

```
> wget http://projects.hepforge.org/spartyjet/spartyjet_3.5.tar.gz
> tar -xvzf spartyjet_3.5.tar.gz
> cd spartyjet_3.5
> source setup.sh

#########################################################
SpartyJet - v3.5

SpartyJet: -------- /Users/briantmartin/Work/projects/spartyjet
ROOT:      -------- /software/root_v5.26
FastJet:   -------- /software/fastjet
StdHep:    -------- enabled (using gfortran)
Pythia6:   -------- enabled (/software/pythia6)
Pythia8:   -------- enabled (/software/pythia8145)
#########################################################
```

- then

```
> make
```

Linking SpartyJet (Optional)

SpartyJet can be made to use an existing version of FastJet

- SpartyJet ships with fastjet-2.4.2
- To change to a private FastJet installation:

  put fastjet-install/bin in your path
  - SpartyJet looks for fastjet-config

```
export PATH=${PATH}/your-fastjet-build/bin
```

  - See manual for additional details.

SpartyJet can link to Pythia through ROOT

- Either Pythia6 or Pythia8
- Must follow ROOT instructions to enable Pythia support when compiling ROOT (See SpartyJet manual for explicit steps)

Running SpartyJet

SpartyJet ships with a set of examples and input data

- Examples are organized by type:
  - Python scripts: spartyjet/examples_py/ Recommended
  - Compiled C++ programs: spartyjet/examples_C
  - ROOT scripts: DEPRECATED
- All examples should be run from their directories

Python

```
source setup.sh
cd  examples_py/
./simpleExample.py
```

SpartyJet Analysis Tool

canvas_group

num cols 1 Reset

num rows 1 Clear

Reset Line Styles

```
./guiExample.py my_sparty_jet_output.root
```

JetCollections from ../data/output/SoperSpannowskyHiggs.root

AntiKt12 AntiKt12Trimmed AntiKt12Pruned CA12MDF

Event by Event plots

Previous Event 0 Next Event

Lego plot
2D view
Snowmass potential
Parameter space
Event dump

Draw

On the fly algorithms

Algo type

0.5 | main param | param 1 | param 2

Create

Run plots

Choose a variable to be drawn :

e pt eta phi mass

And/or provide the expression(s) to be draw below
(use $$ as a wildcard (ex: $$_pt) to be replaced by checked jets
Separate multiple variables with '##'.
Expression: (Ex: $$_mass)

Cuts: (Ex: $$_pt > 100 && $$_pt < 200)

$$_mass > 60

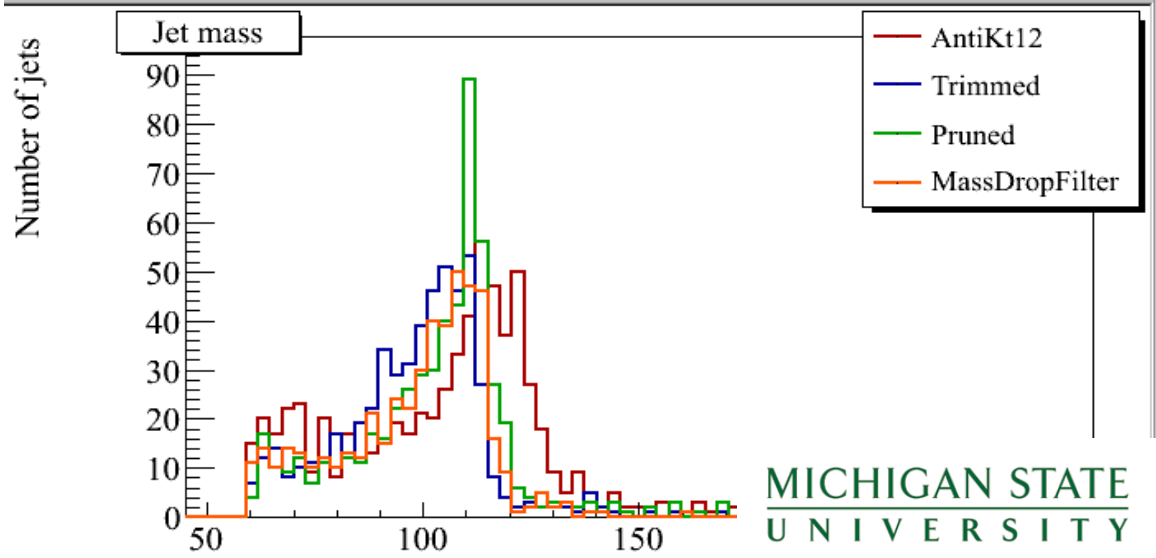
Legend entries for each curve: (Ex: ttbar##dijet)

AntiKt12##Trimmed##Pruned##MassDropFilter

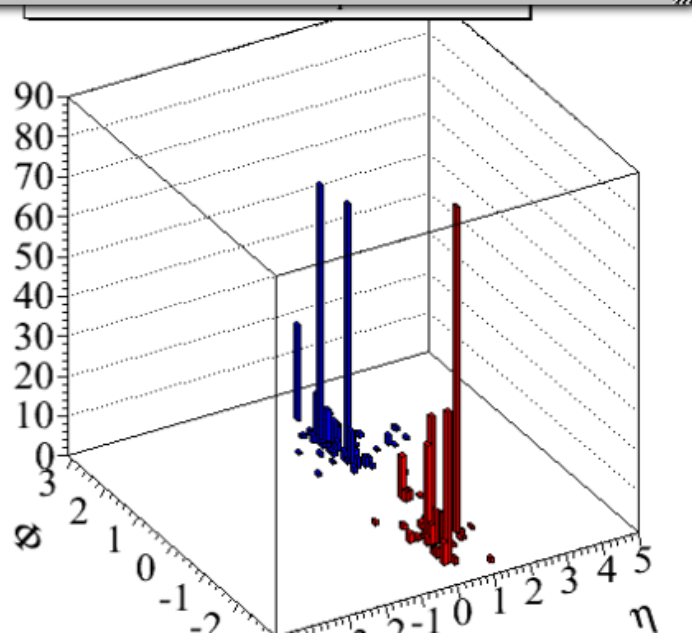
Draw reset

Normalize LogX LogY No Title Same Pad

File Edit View Options Tools Help



MICHIGAN STATE UNIVERSITY



Summary

- SpartyJet has undergone a great deal of development in 2010
  - Implemented many direct requests from Jet Physics Workshop in Washington (C. Vermillion)
- SpartyJet is now prepared to be a tool of the jet substructure trade
  - Modular nature allows quick clean implementation of sub structure analysis chains
  - Particular uses were shown in moving theoretical ideas/projects to experimental application
- We would like to add even more tools in the very near future:
  - ex. N-subjettiness
- Continues improvement/expansion of the GUI interface is now a development priority
- Goal to really make this useful, so please try it and give us feedback

Thank you

SpartyJet hosted on HEPForge:

- http://projects.hepforge.org/spartyjet/
- New Release:
  http://projects.hepforge.org/spartyjet/spartyjet_3.5.tar.gz
- Manual available:(also in release)
  http://projects.hepforge.org/spartyjet/SpartyJetDocumentation_3.5.pdf
- Support and Feature requests:
  spartyjet@projects.hepforge.org

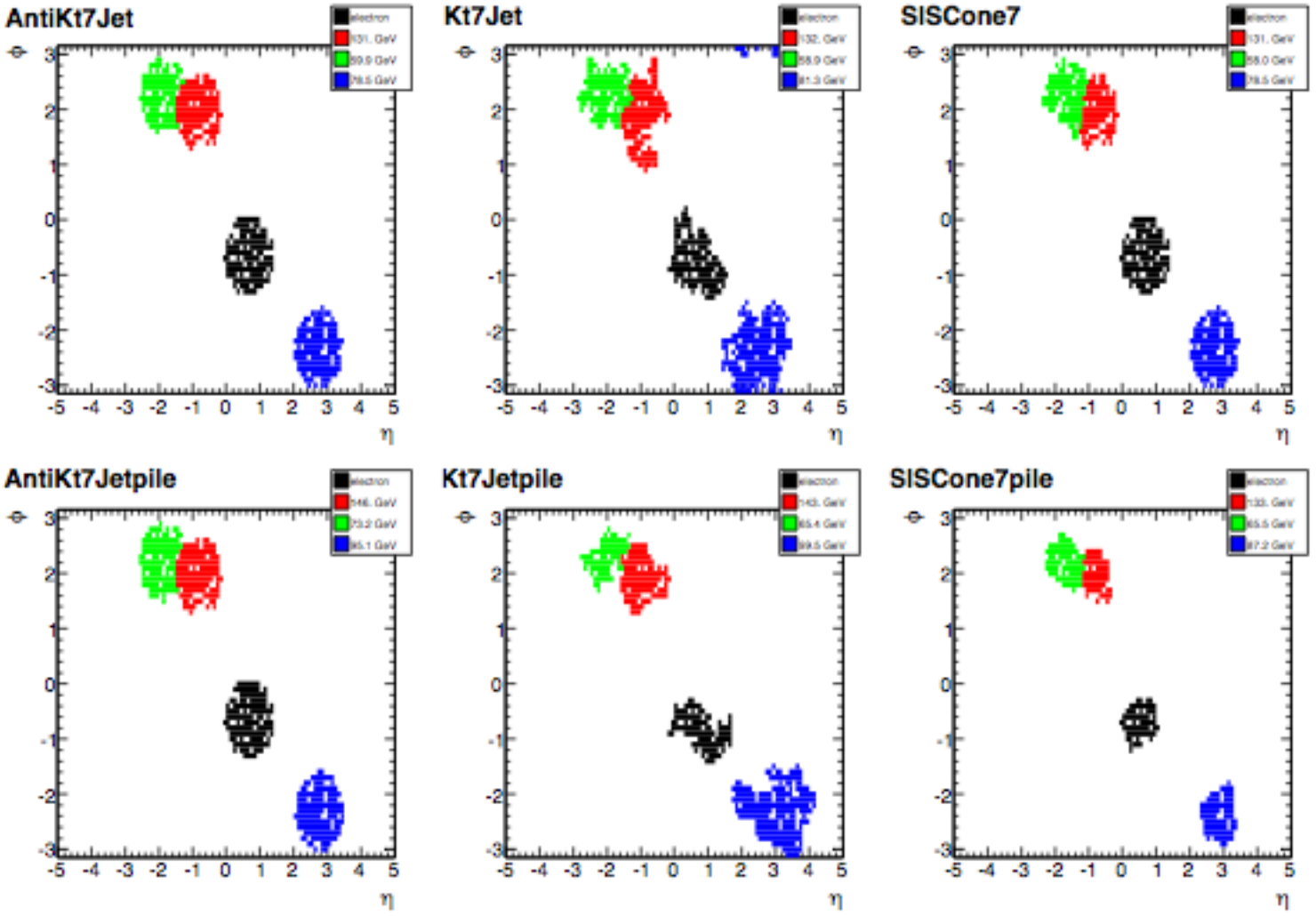
Other SpartyJet Features: Pileup Addition

SpartyJet can overlay minbias events on the signal event

- Reads a separate minbias file and adds these input 4-vectors to the list from the signal event
- This can be done in two ways:
  - Add a set number of events from the minbias file per each signal event
  - Add a number of events drawn from a Poisson distribution with a configurable most probably value (useful if you have a sample of single minbias events and want to simulate pileup)
- Additionally one can run jet reconstruction in parallel (in the same job) on events with and without pileup, allowing one to directly see the effects of pileup on an event-by-event basis

Other SpartyJet Features: Pileup Addition

Example of pileup overlay comparison (W->e+nu with 3jets)



UNIVERSITY