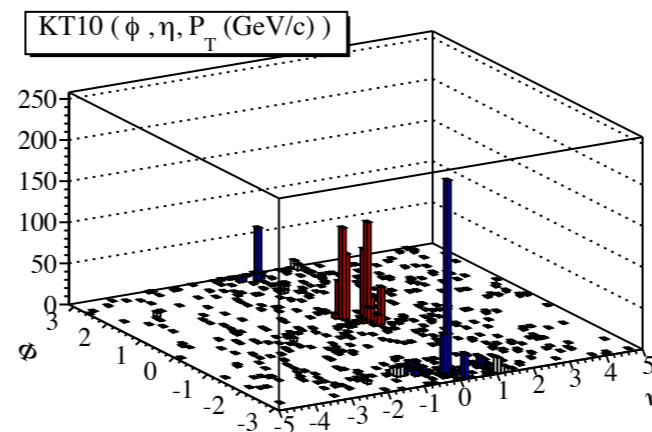
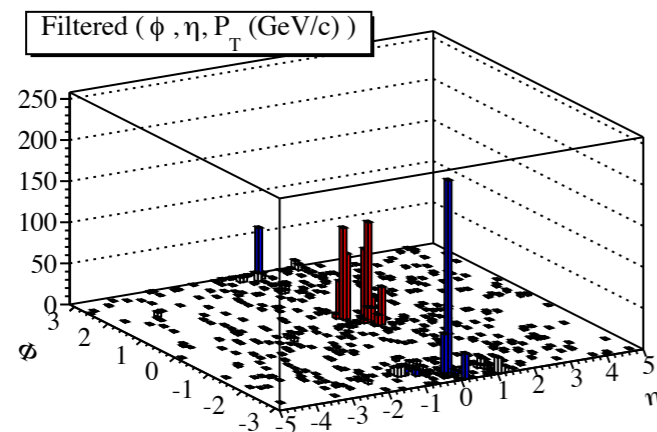
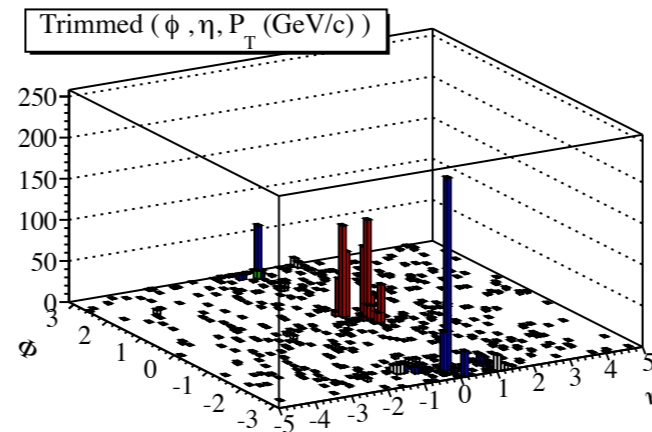
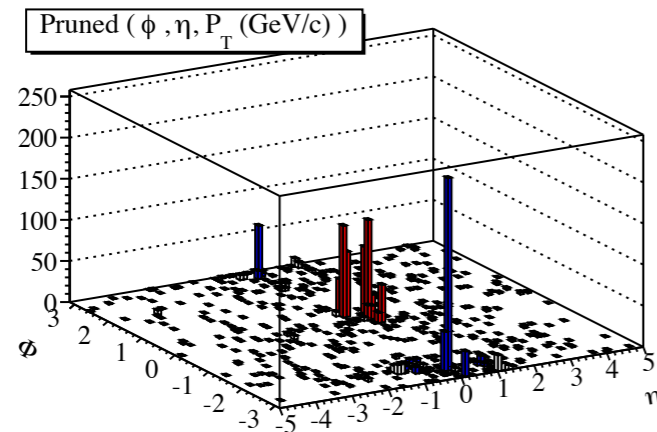


# Exploring Jet Tools with SpartyJet



Christopher Vermilion  
BOOST 2011  
5/24/11



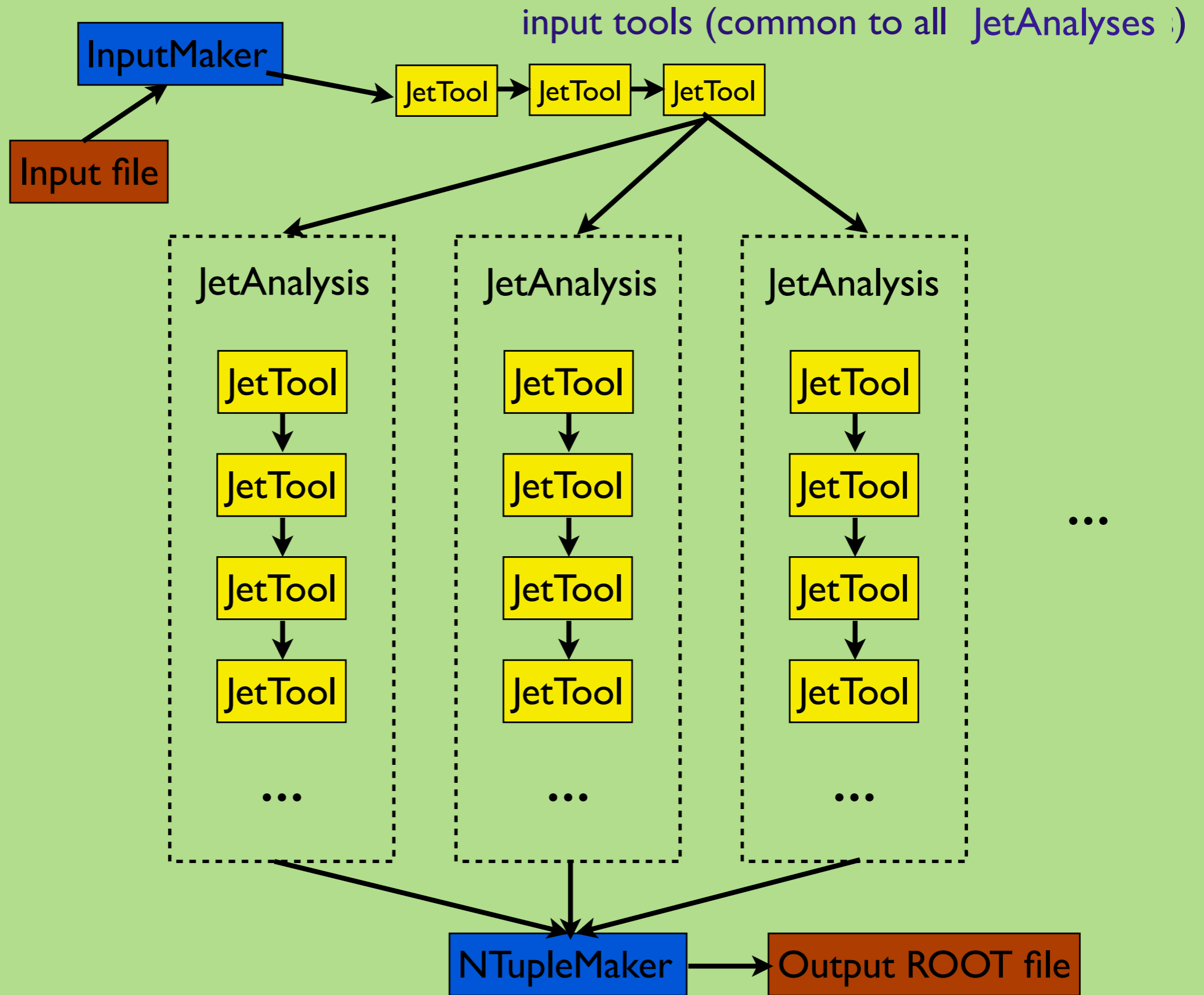
# SpartyJet

Goal is to **simplify**, **standardize**, and **explore** jet analyses

- Wraps around jet finding in FastJet
- Support for many input formats
  - StdHEP, HepMC, LHE, ROOT trees, several ASCII formats
- Output to simple ROOT files, explorable with increasingly powerful GUI
- Analyses consist of chains of “JetTool”s
  - **Very simple plug-and-play**
  - **Growing list of tools available**

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

# JetBuilder (job manager)



**A**  
**SpartyJet**  
**Analysis**

# Short example: simple.py

```
from SpartyJetConfig import *
#=====

# Create a jet builder(MessageLevel = INFO)-----
builder = SJ.JetBuilder(SJ.INFO)

# Create input object and add to builder -----
input = SJ.StdTextInput('../data/J1_Clusters.dat')
builder.configure_input(input)

# Create jet finder and add to builder -----
name = 'AntiKt4'
alg = fj.antikt_algorithm
R = 0.4
antikt4 = SJ.FastJet.FastJetFinder(name, alg, R)
builder.add_default_analysis(antikt4)

# Add a jet measurement
builder.add_jetTool(SJ.EtaPhiMomentTool())

# Configure text output (optional) -----
builder.add_text_output("../data/output/simple.dat")

# Configure ntuple output-----
outfile = "../data/output/simple.root"
builder.configure_output("SpartyJet_Tree", outfile)

# Run SpartyJet-----
builder.process_events(10)

# Save this script in the ROOT file
writeCurrentFile(outfile)
```

# Not-so-short example: Boost2010.py

```
# Create a jet builder-----
builder = SJ.JetBuilder(SJ.WARNING)
# Configure input -----
input = getInputMaker(infile)
builder.configure_input(input)

# Run Anti-kt 1.0
AntiKt10 = SJ.FastJet.FastJetFinder('AntiKt10', fj.antikt_algorithm,
1.0, False)
builder.add_analysis(SJ.JetAnalysis(AntiKt10))
# cut on pT *before* forking to save time
builder.add_jetTool(SJ.JetPtSelectorTool(200.0, 2), 'AntiKt10') # two
highest-pt jets over 200
AKTparent = SJ.ForkToolParent('AntiKt10Parent')
builder.add_jetTool(AKTparent, 'AntiKt10')

# recluster with CA, then fork again for different taggers
builder.add_analysis(SJ.JetAnalysis(SJ.ForkToolChild(AKTparent,
'CA10')))
builder.add_jetTool(SJ.FastJet.FastJetRecluster('CA10cluster',
fj.cambridge_algorithm, 1.0, False), 'CA10')
CAparent = SJ.ForkToolParent('CA10Parent')
builder.add_jetTool(CAparent, 'CA10')

# recluster with kT
builder.add_analysis(SJ.JetAnalysis(SJ.ForkToolChild(AKTparent,
'KT10')))
builder.add_jetTool(SJ.FastJet.FastJetRecluster('KT10cluster',
fj.kt_algorithm, 1.0, False), 'KT10')

# PDG id cut -- remove muons (13) and neutrinos (12, 14, 16)
ids = stdVector(-12, 12, 14, -14, 16, -16, 13, -13)
builder.add_jetTool_input(SJ.JetInputPdgIdSelectorTool(ids))
# Input eta cut
builder.add_jetTool_input(SJ.JetEtaCentralSelectorTool(-5.0, 5.0))

# some tools for use below
recluster = SJ.FastJet.FastJetRecluster('Recluster',
fj.cambridge_algorithm, pi*0.5, False)
massCut = SJ.JetMassSelectorTool(100.0)
pTcut = SJ.JetPtSelectorTool(100.0)

# ----- Analyses -----
# pruning
big_CA_def = fj.JetDefinition(fj.cambridge_algorithm, 3.14*0.5)
builder.add_analysis(SJ.JetAnalysis(SJ.ForkToolChild(AKTparent,
"Pruned")))
prune = SJ.FastJet.FastPruneTool(big_CA_def, 0.05, 0.1)
builder.add_jetTool(prune, "Pruned")

# trimming
trimPlugin = fj.QCDTrimmingFast(fj.JetDefinition(fj.antikt_algorithm,
1.0), 200.0)
trimPlugin.SetRsubPtfrac(0.35, 0.03)
trimPlugin.UseEffMass(False)
```

```
builder.add_default_analysis(SJ.FastJet.FastJetFinder(fj.JetDefinition
(trimPlugin), "Trimmed", False))
builder.add_jetTool(recluster, "Trimmed")

# filtering
builder.add_analysis(SJ.JetAnalysis(SJ.ForkToolChild(CAparent,
"Filtered")))
filter = fj.Filter(0.35, 3)
builder.add_jetTool(SJ.FastJet.FilterTool(filter, 1.0), "Filtered")
builder.add_jetTool(recluster, "Filtered")

# Set up JH top tagger
builder.add_analysis(SJ.JetAnalysis(SJ.ForkToolChild(CAparent,
"CA10JH")))
JHtagger = fj.JHTopTagger(0.04, 0.19, 81.0)
JHtool = SJ.FastJet.TopTaggerTool(fj.JHTopTagger)(JHtagger)
builder.add_jetTool(JHtool, 'CA10JH')
# add a cut on cos_theta_h, which is stored by the tagger
builder.add_jetTool(SJ.JetMomentSelectorTool(float)('cosThetaH', -0.95,
0.95), 'CA10JH')

# Set up CMS top tagger
builder.add_analysis(SJ.JetAnalysis(SJ.ForkToolChild(CAparent,
"CA10CMS")))
CMStagger = fj.CMSTopTagger()
CMStool = SJ.FastJet.TopTaggerTool(fj.CMSTopTagger)(CMStagger)
builder.add_jetTool(CMStool, 'CA10CMS')

# ATLAS z_cut variables
zcut = SJ.zcutMoment()
builder.add_jetTool(SJ.JetMomentTool("zcutTool", zcut), 'KT10')

# Thaler-Wang z_cell
zcell = SJ.zcellMoment()
builder.add_jetTool(SJ.JetMomentTool("zcellTool", zcell), 'KT10')

# -----
# Measure heavier subjet mass
subjetMoment = SJ.HeavierSubjetMass("subjetM")
builder.add_jetTool(SJ.JetMomentTool("SubjetMassTool", subjetMoment))
# Add a min-mass tool to non-top-tagging analyses (the top-taggers
already find mW)
for alg in ["Pruned", "Filtered", "Trimmed", "AntiKt10", "CA10",
"KT10"]:
    builder.add_jetTool(SJ.UnclusterTool(3), alg)
    builder.add_jetTool(SJ.MinMassTool("mW"), alg)

# Configure output-----
builder.configure_output("Jets", outfile)
# Run SpartyJet
builder.print_event_every(100)
builder.process_events()
# Save this script in the ROOT file
writeCurrentFile(outfile)
```

# Changes since Boost 2010

- New tools:
  - W-tagging (just over-all tagging, no individual variables yet...)
  - N(sub)jettiness (unvalidated)
  - CMS Top Tagger (my implementation, based on CMTopTagger)
  - HEP Top Tagger (semi-official pre-public version, to be replaced!)
  - Peter Loch's DetectorModel
  - z\_cut and z\_cell JetMoments to mimic "ATLAS" and "Thaler & Wang" taggers from BOOST 2010
- New input formats:
  - Three-vector Ntuples
  - Four-vectors from code
  - Multiple input files
- Experimental CMake build system (lots of cool features; Windows?!?)
- Non-FastJet jet finding completely removed (why duplicate?)
- Expanded FastJet access in ROOT/PyROOT
- Improvements to GUI appearance (no, really)
- "Provenance" storage
- Lots of minor bug fixes and improvements

\*requested by Sal

Let's design input to SJS

Input (class Yvec)

Vector < Yvec >

Yvec is some class

implementing

prec() ... prec(), (E)

- Multiple input files?

# SpartyJet tools: Selectors

- `JetPtSelectorTool(ptmin, nmax)`
- `JetPtOrESelectorTool(ptmin, emin)`
- `JetEtaCentralSelectorTool(eta)`
- `JetEtaForwardSelectorTool(eta)`
- `JetMassSelectorTool(mass)`
- `JetInputPdgIdSelectorTool(vector<int> ids)`
- `JetMomentSelectorTool<Momen>(name, min_val, max_val)`

# SpartyJet tools: Moments

- HullMomentTool
- EtaPhiMomentTool
- PtDensityTool
- JetAreaCorrectionTool
- YSplitterTool
- JetMomentTool:
  - HeavierSubjetMass
  - FinalDij
  - z
  - DeltaR
  - zcell
  - zcut[N]

# SpartyJet tools: Substructure

- TopDownPruneTool
  - MassDropToop
  - JHPruneTool
- SubjetCutTool
- SubjetMergeTool
- FilterTool
  - BDRSFilterTool
- N(Sub)Jettiness
- WTaggerTool
- TopTaggerTool -- templated, can take:
  - JHTopTagger
  - CMTopTagger
  - CMSTopTagger
  - HEPTopTagger

plus several miscellaneous tools...

# ~~BOOST 2011~~ To-Do List

- More interactive features
  - Pythia, tool chain control from GUI?
- More event information for crude “full analyses”
  - b-tagging, lepton isolation, pile-up tagging
- Stored cluster history (partially implemented, not tested)
- Fastjet 3 integration
- SWIG instead of PyROOT
- More tools (next slide...)
- ... Requests?

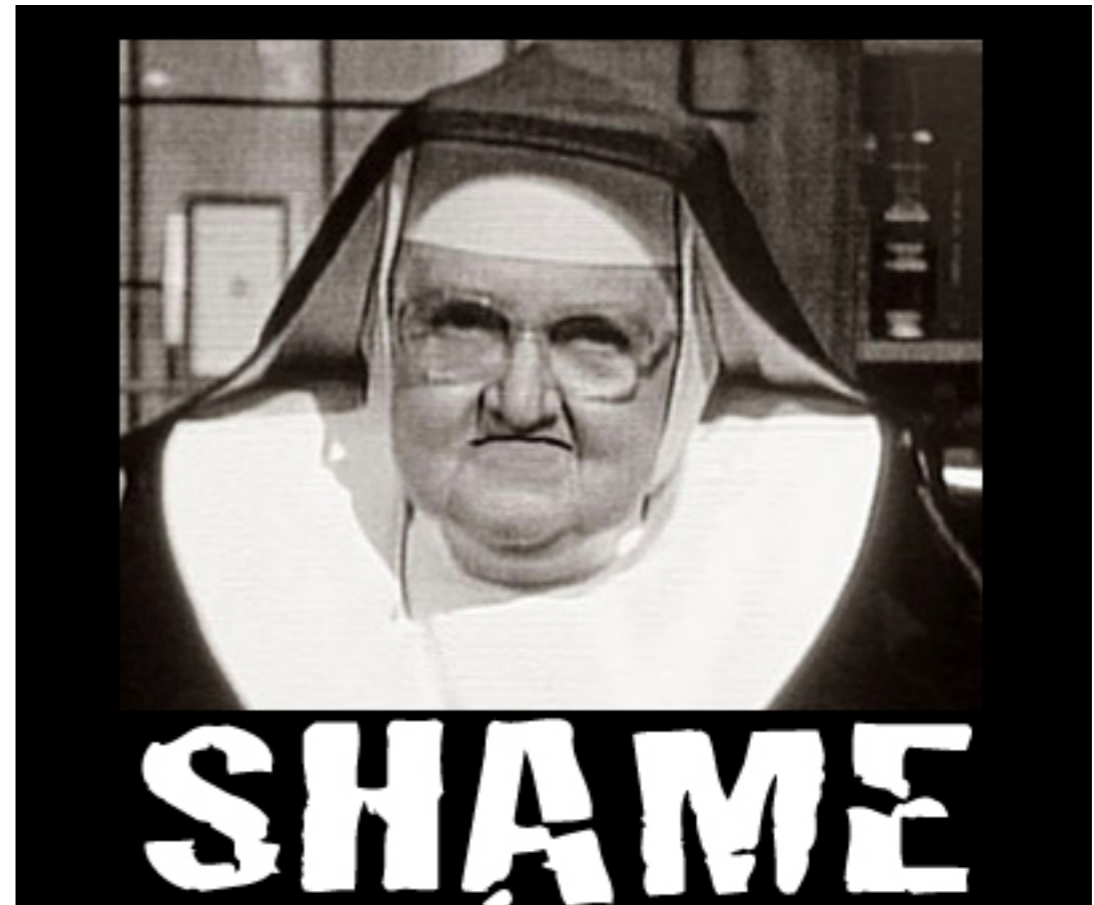
**Goal: complete jet analysis workbench**

# SLIDE OF SHAME

An incomplete list of methods that do not, to my knowledge, have public, certified code

(I would be very happy to be corrected!!)

- N(Sub)jettiness (partial credit)
- Template overlap
- Jet dipolarity
- “Substructure without trees”
- Shower deconstruction
- Quark vs. gluon suite (but see <http://jets.physics.harvard.edu/qvg/index.html>)
- ~~HEP Top tagger~~ (pseudo-public)
- Surely some I've missed...



If you build it, I will put it in SpartyJet!

# Boost 2011 Homework

- Outcome of Boost 2010:
  - Catalog of top tagger results
- Ways to go beyond this:
  - Consider different topologies?
  - Expand set of tools/taggers
  - *Catalog of code!*
    - *More comparisons, tests require more automation and standardization*

Is SpartyJet the answer?

# Get SpartyJet 3.6!

Source tarball at <http://projects.hepforge.org/spartyjet>

Checkout this branch (including any patches!):

```
svn co http://svn.hepforge.org/hepforge/svn/spartyjet/branches/  
spartyjet-3.6-patches spartyjet
```

Checkout the SVN trunk (and keep up with new features, catastrophic build failures):

```
svn co http://svn.hepforge.org/hepforge/svn/spartyjet/trunk spartyjet
```

# Useful links

**FastJet:** <http://fastjet.fr>

**Tools and plugins:** <http://www.lpthe.jussieu.fr/~salam/fastjet/tools.html>

**SpartyJet:** <http://projects.hepforge.org/spartyjet/>

**Brian's slides from Boston JW:** <http://indico.cern.ch/getFile.py/access?contribId=29&resId=0&materialId=slides&confId=113980>

**Events generated for BOOST2010 (arXiv:1012.5412):**

<http://www.lpthe.jussieu.fr/esalam/projects/boost2010-events/>

<http://tev4.phys.washington.edu/TeraScale/boost2010>

ttbar, dijet, Zj; in "UW" ASCII format

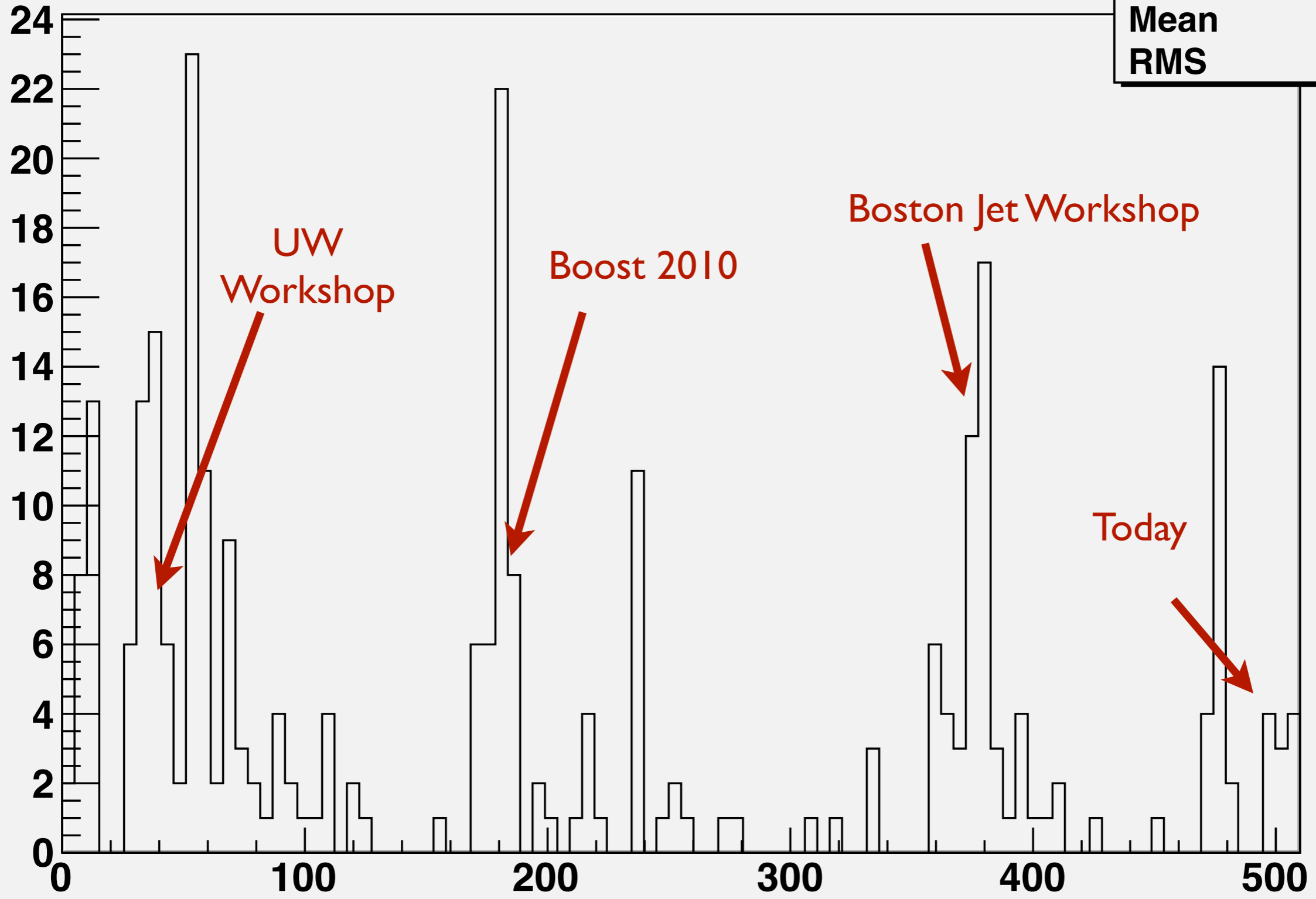
**Plots, summary, and ROOT files from UO demo:**

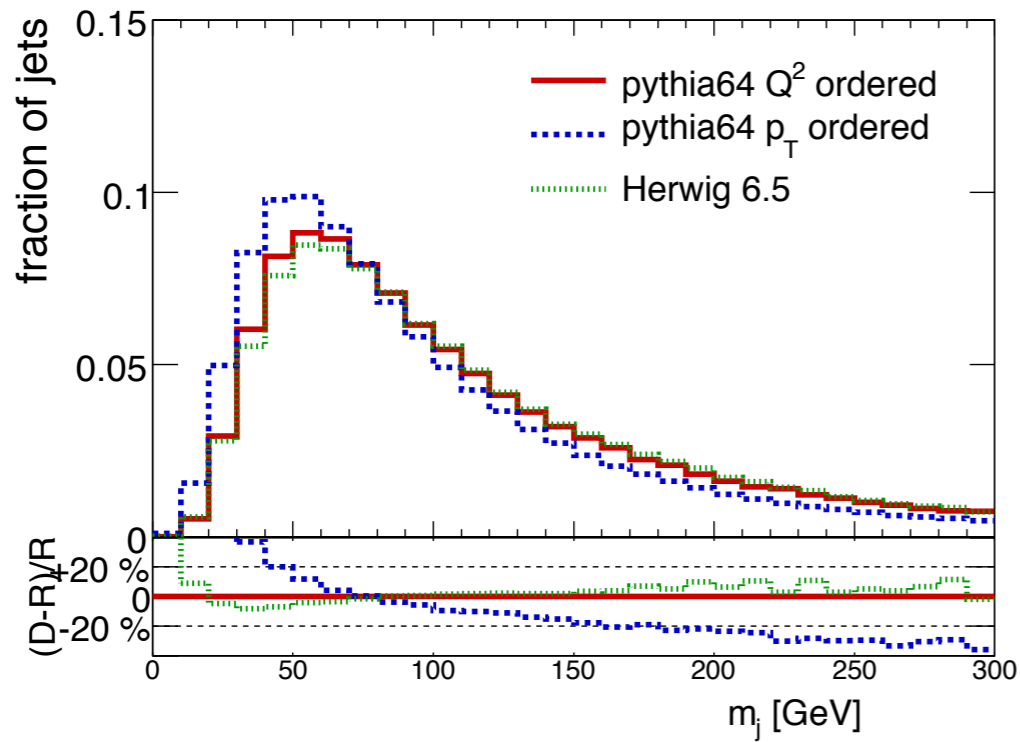
<http://tev4.phys.washington.edu/TeraScale/uoregon>

# Bonus slides

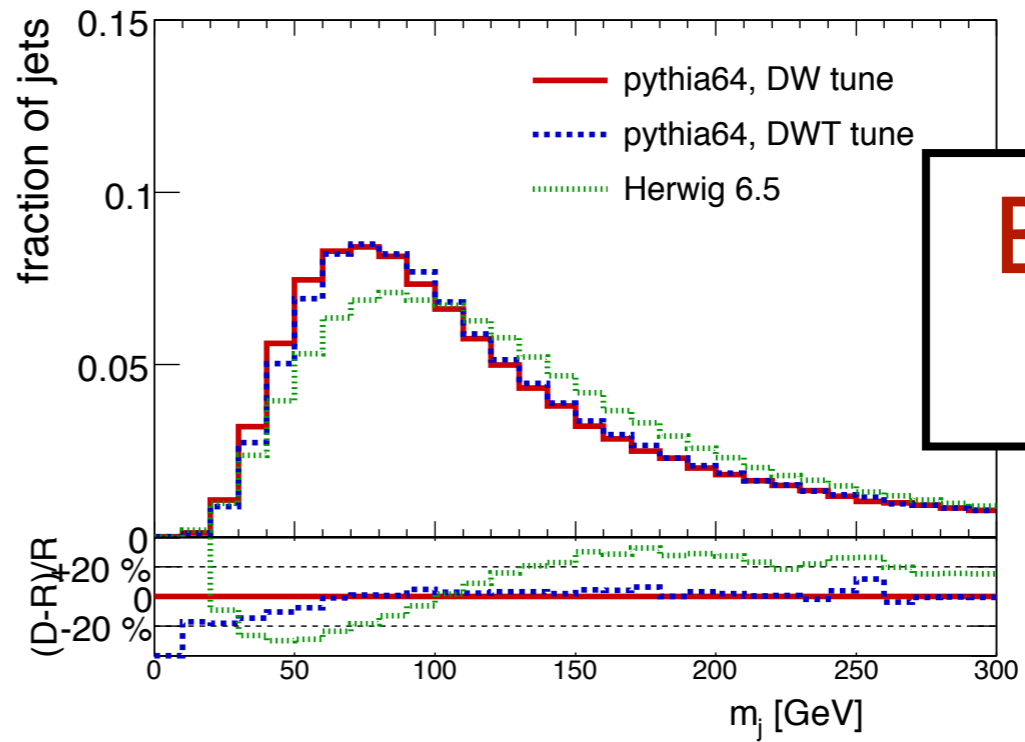
# Days since Jan. 1 of SJ commits

commits	
Entries	292
Mean	199.3
RMS	161.3





(a) jet mass - PS

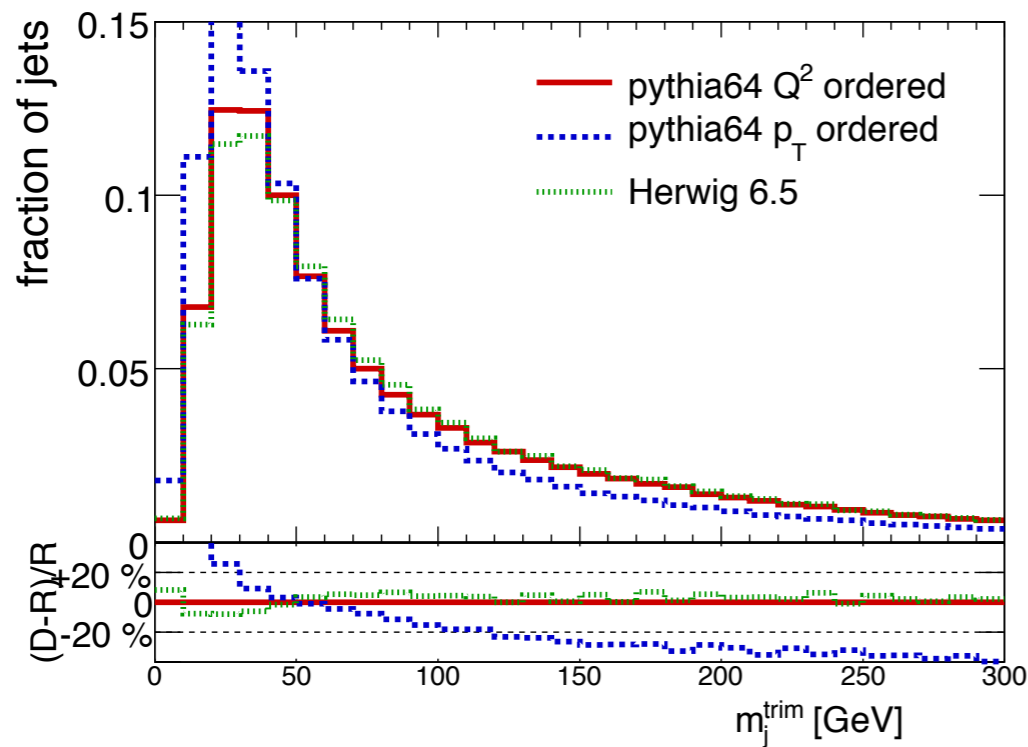


(b) jet mass - UE

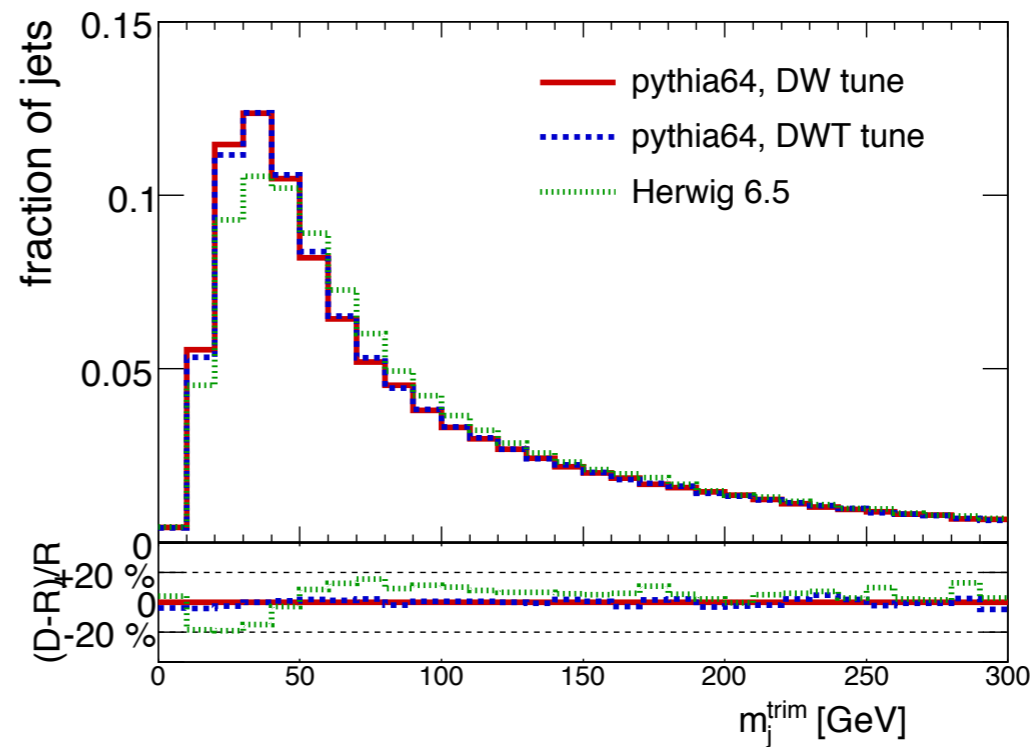
**GROOM**



(groom = trim)



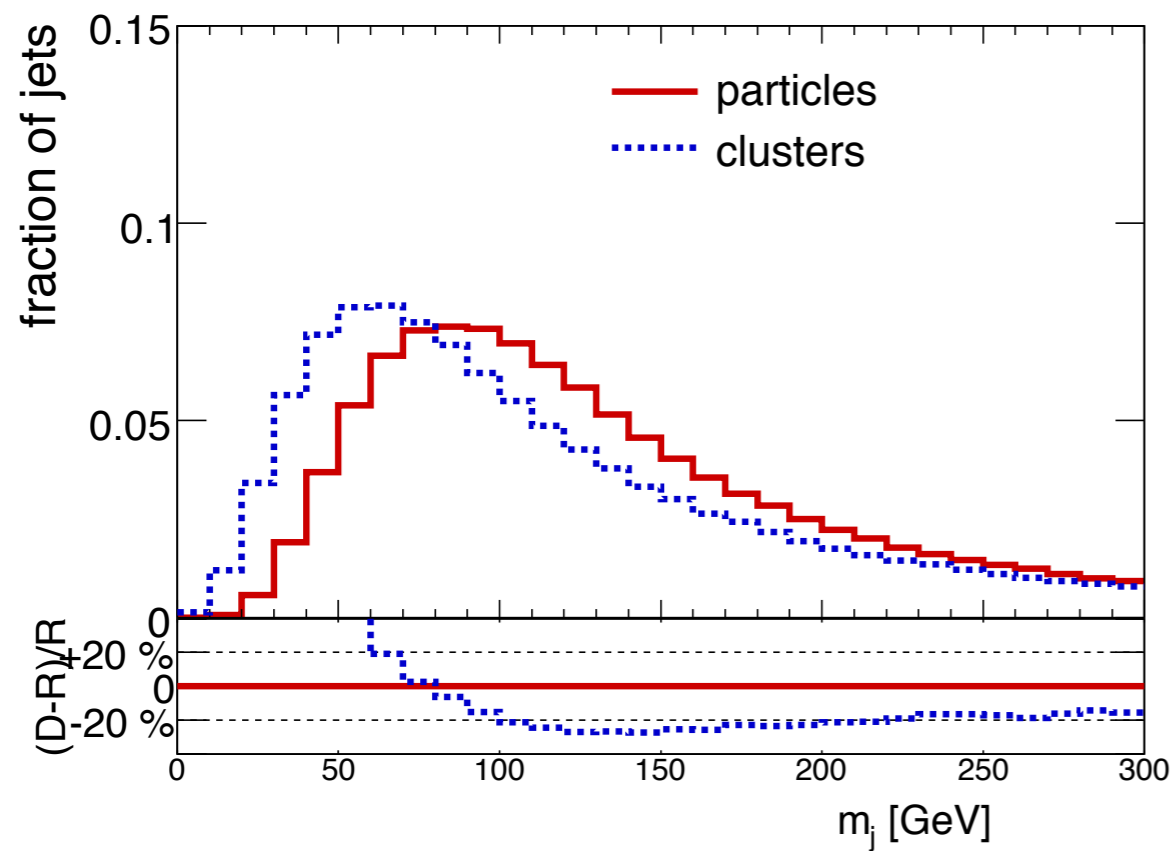
(d) groomed jets - PS



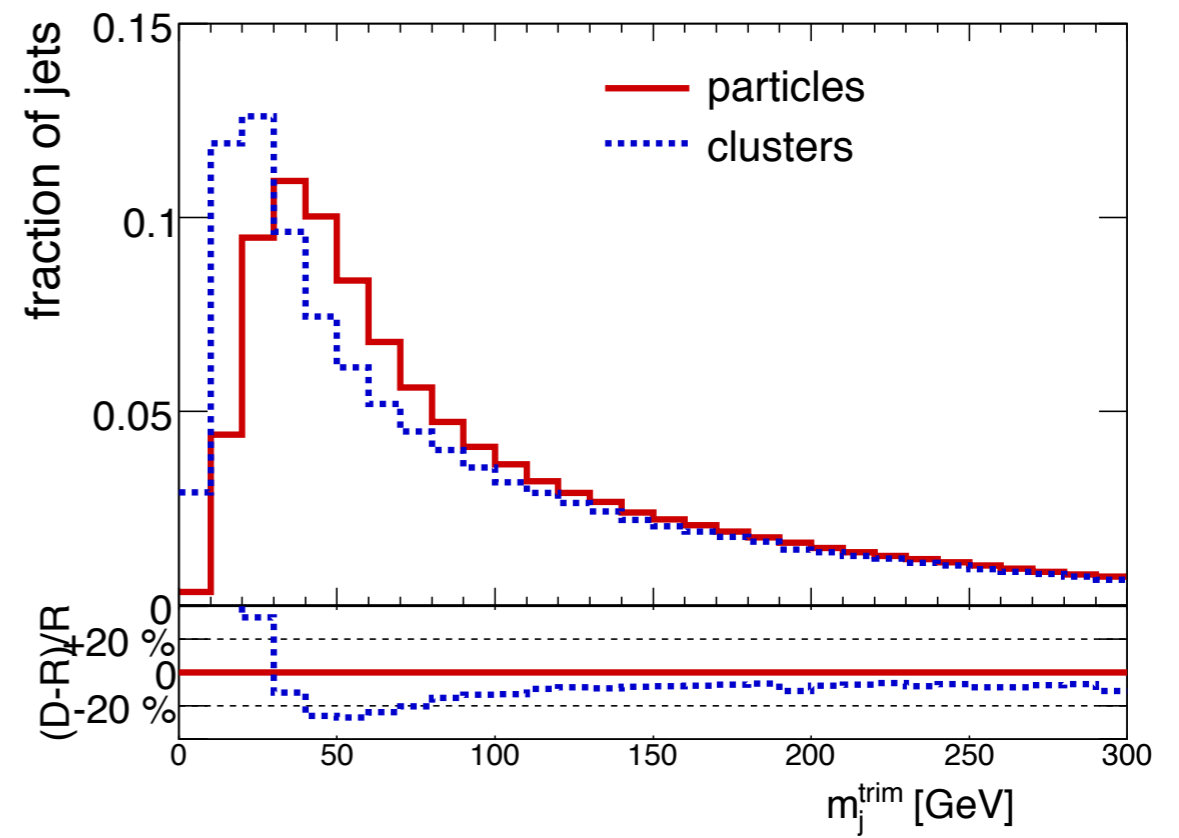
(e) groomed jets - UE

**BOOST**  
results

# “Detector” effects



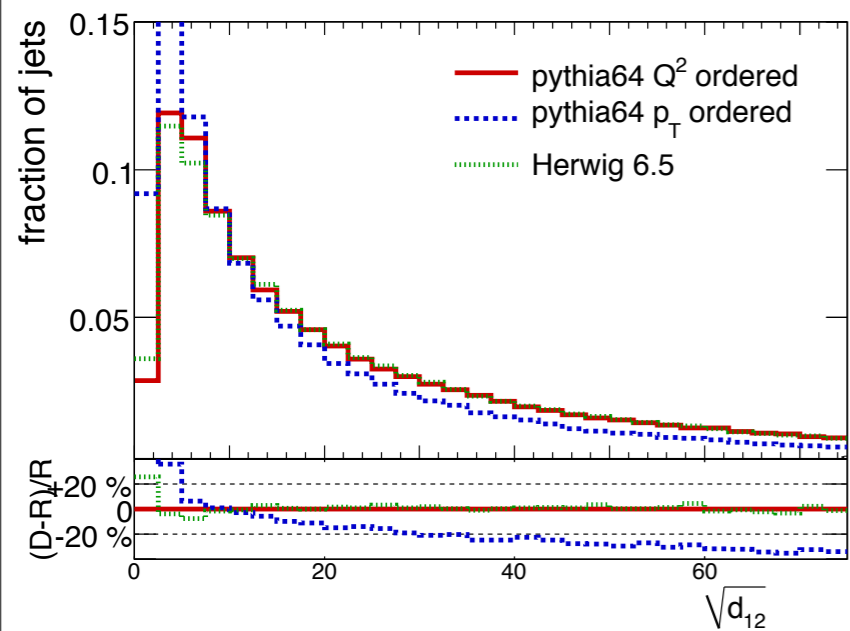
(c) jet mass - detector



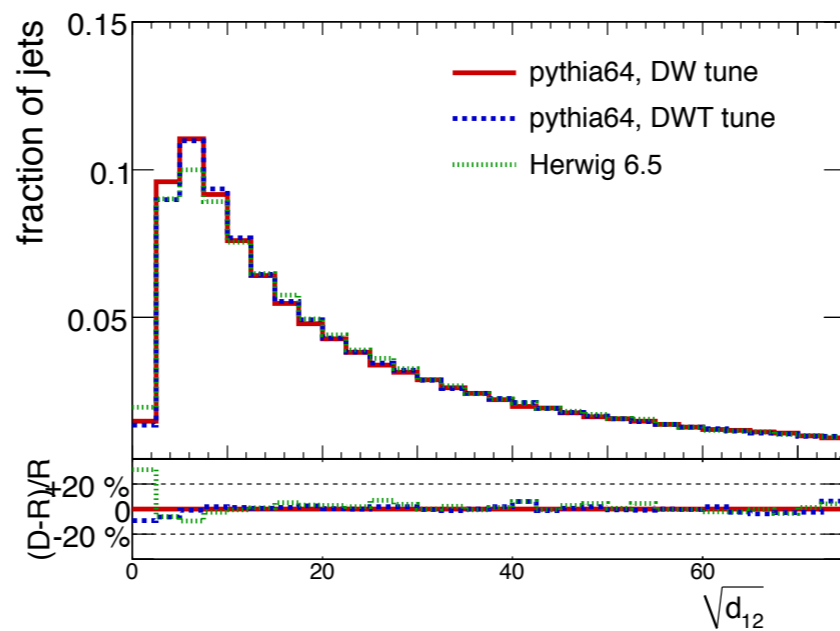
(f) groomed jets - detector

**Groomed jets: easier to calibrate?**

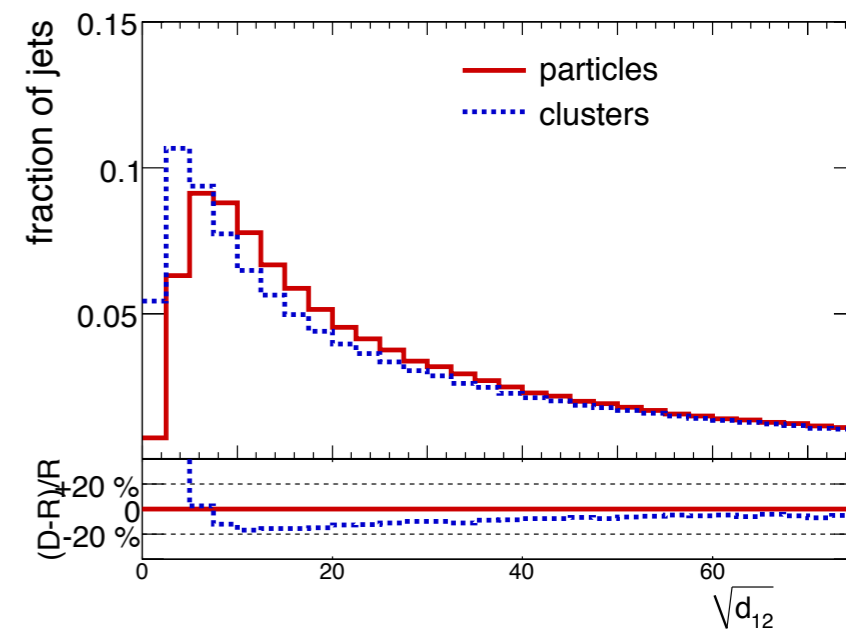
# Splitting scales in different MCs



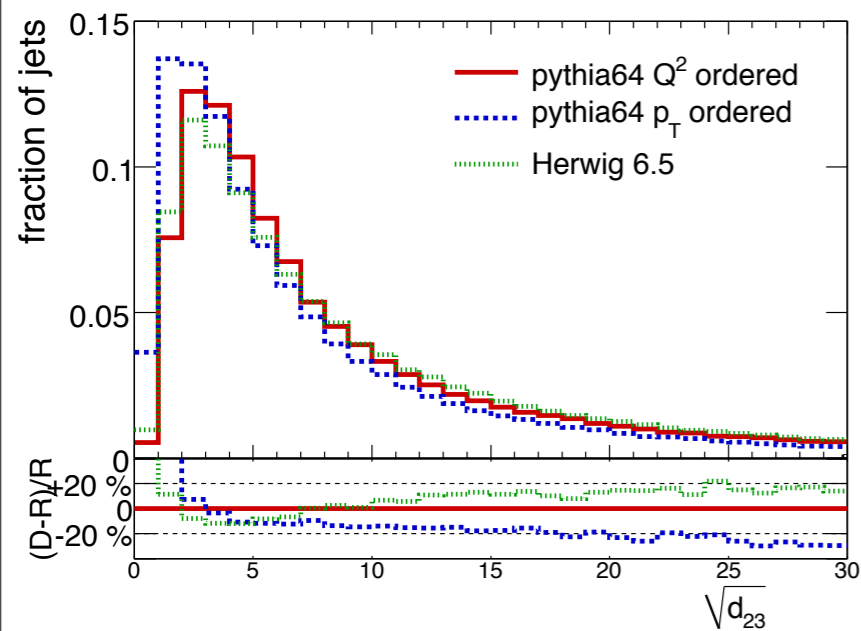
(g) 1 → 2 split - PS



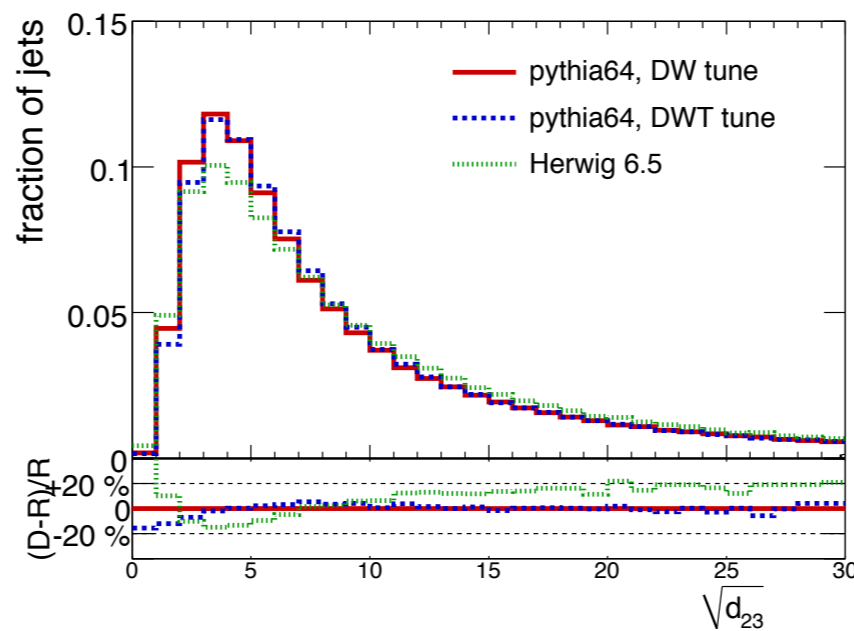
(h) 1 → 2 split - UE



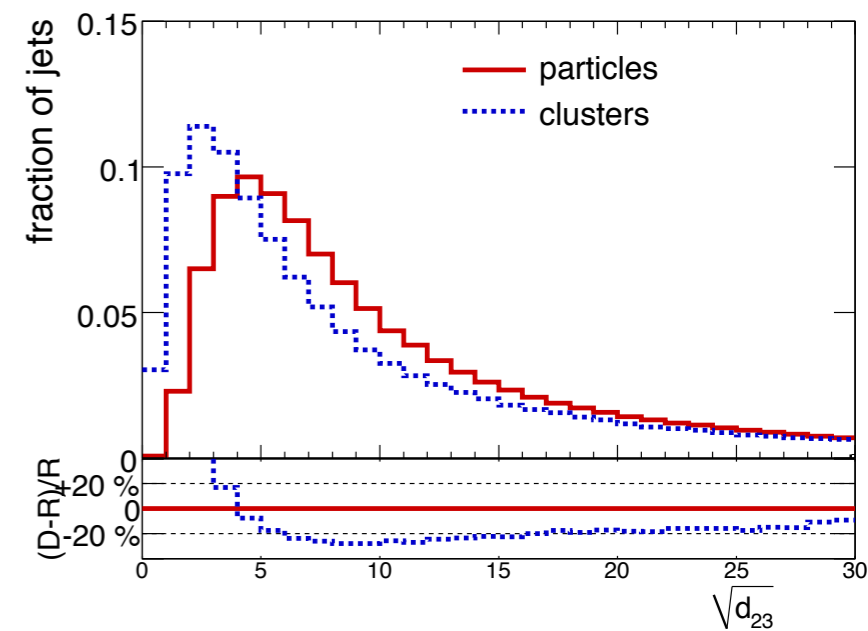
(i) 1 → 2 split - detector



(j) 2 → 3 split - PS



(k) 2 → 3 split - UE



(l) 2 → 3 split - detector