

Software Tools for Jet (and Subjet!) Analysis

Christopher Vermilion
University of Washington

with Pierre-Antoine Delsart, Joey Huston,
and Brian Martin



24/06/2010
University of Oxford



Useful links

FastJet: <http://fastjet.fr>

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

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

Software tools page on January workshop wiki:

http://librarian.phys.washington.edu/lhc-jets/index.php/Software_Tools

<http://librarian.phys.washington.edu/lhc-jets/index.php/SpartyJet> (manual linked here)

Events generated for January workshop:

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

ttbar, dijet, ZH, ZZ, Zj; in StdHEP or “UW” ASCII formats

FastJet

A tool for jet finding

- Essentially all standard algorithms implemented
 - kT , Cambridge/Aachen, anti- kT (built-in)
 - SISCone, PxCone, JetClu, MidPoint, D0 RunII Cone, ATLAS and CMS Cones (plugin)
- Easy mechanism for user-supplied plugins
 - Variable-R, pruning, trimming
- Growing set of FJ-based tools (on FJ website)
 - JHTopTagger, filtering, CA subjet finding, “CMTopTagger” (add-ons), BDRS Higgs search (built-in)

FastJet example

```
fastjet::JetDefinition jetDef(fastjet::kt_algorithm, E_scheme, Best);

vector<fastjet::PseudoJet> inputs;
// ... fill somehow

fastjet::ClusterSequence cs(inputs, jetDef);

// inclusive jets with pT > 20 GeV
vector<fastjet::PseudoJet> inc_jets = cs.inclusive_jets(20.0);

// Constituent access:
vector<fastjet::PseudoJet> consts = cs.constituents(inc_jets[0]);
fastjet::PseudoJet p1, p2;
if (cs.has_parents(inc_jets[0], p1, p2)) {
    // do something with p1, p2...
}

// dij corresponding to final 2->1 merging for a 1-jet event
double d12 = cs.exclusive_dmerge(2);

// subjets at some resolution scale (corresponding to dij in the algorithm)
double dcut = inc_jets[0].m()/inc_jets[0].perp();
vector<fastjet::PseudoJet> subjets = cs.exclusive_subjets(inc_jets[0],
dcut);
```

Tool example: Filtering in FastJet

```
// Find a ClusterSequence somehow, now get jets:  
vector<fastjet::PseudoJet> jets = cs.inclusive_jets();  
  
// Construct a Filter object (from Filter.{hh,cc}, on FastJet page)  
Filter filter(0.2, 3) // filter down to R_filt = 0.2, keep 3 subjets  
  
for (size_t i = 0; i < jets.size(); i++) {  
    fastjet::FilteredJet fjet(cs, jets[i], filter);  
    // do things with fjet -- FilteredJet derives from PseudoJet  
    vector<fastjet::PseudoJet> subjets = fjet.kept();  
    // ... other methods described in Filter.hh  
}
```

SpartyJet

A tool for jet analysis

- Wraps around jet finding in FastJet
 - (Some built-in algorithms, but only for historical reasons...)
- 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

SpartyJet walkthrough example

`examples_py/FJToolExample.py`

- Goal: Run several “grooming” analyses on ttbar events, compare mass distribution, then subject mass distribution after mass cut
- I’m going to walk through `FJToolExample.py` a few lines at a time -- I encourage you to follow along at home!

SpartyJet to-do list

- Access to all available subjet tools
 - Standardized interface for these: e.g., filtering, tagging, top-down pruning each represent a *class* of methods
- Forking JetTools for efficiency
- Expanded GUI functionality
- Re-run jet finders on jet constituents
- Visual substructure browser (partially implemented by Pierre-Antoine)
- Better documentation of available tools
- ...
- **What else?!?**

**SpartyJet is a work in progress --
check back often!**

Feedback is strongly welcomed!