SpartyJet

Generic routines for Jet analysis

(this is Sparty!)

Pierre-Antoine Delsart
LAPP (IN2P3, Annecy)

Kurtis L. Geerlings & Joey Huston
Michigan State University

Introduction

Goal :

Give physicists a common set of **c++** routines to ease :

- analysis with jets
- Jet algorithm comparison
- Jet algorithm development

We created it trying to make it

Fast to run (no need of heavy framework, compiled code)

Easy to use (driven through ROOT scripts)

Flexible (adapt to any input – allow in-depth analysis)

How it works

Design inspired by Atlas Jet software

SpartyJet Objects

Jet ~HepLorentzVector with constituents
JetMomentMap : associate any quantity to a jet
JetCollection

Algorithm flow

Input Classes
Interfaces to read any kind of input into SpartyJet format
Jet Sequence classes
Each jet algo is a sequence of operation (preselection, jet finding, moment calculation, etc...) on a jet list
Output classes
ROOT Ntuple

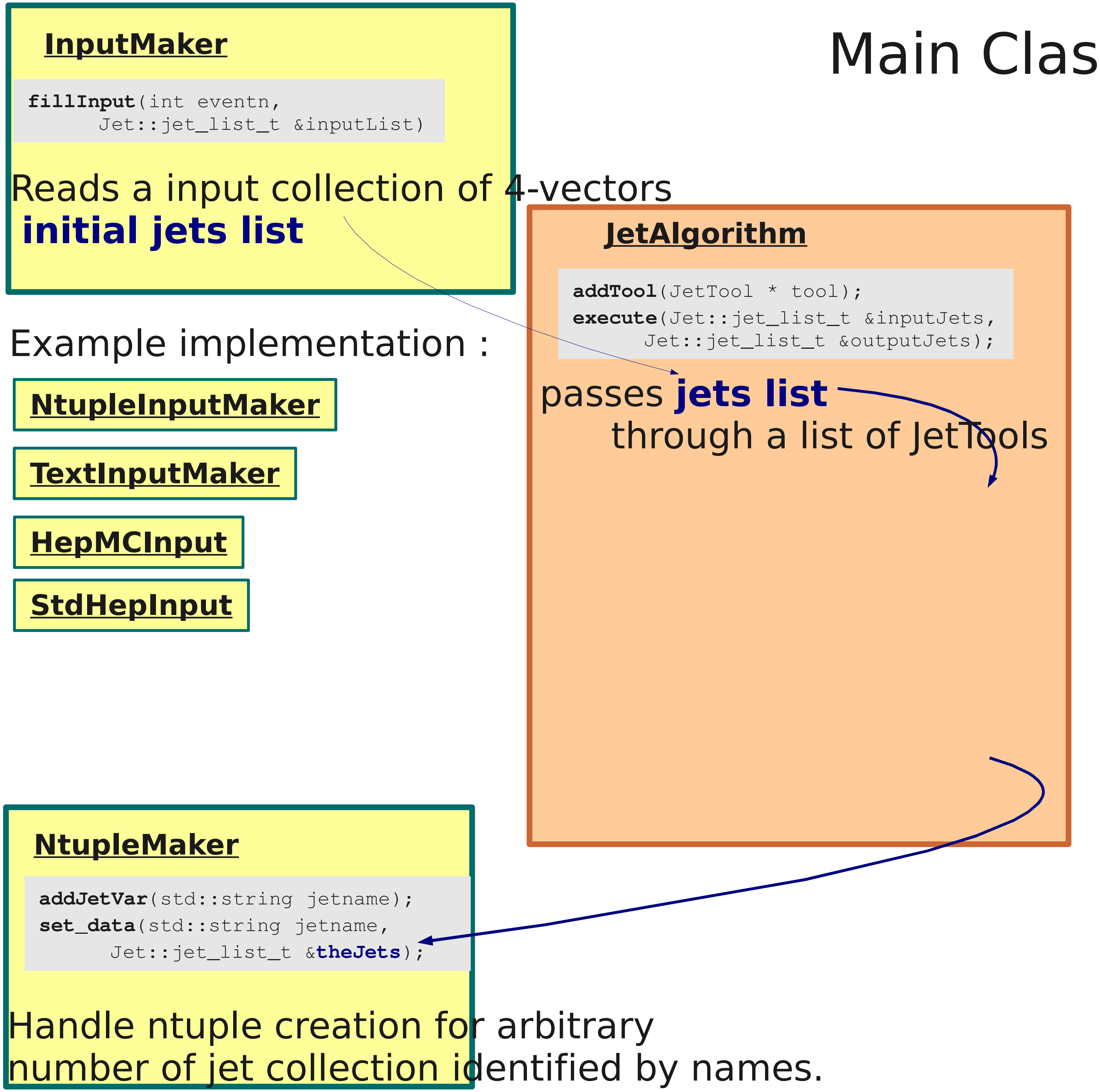
Everything is very modular

⇒ flexible

We provide wrapper class/functions (**JetBuilder ,getJets()**)

⇒ easy to use

Main Classes



InputMaker

```
fillInput(int eventn,
      Jet::jet_list_t &inputList)
```

Reads a input collection of 4-vectors

initial jets list

Example implementation :

NtupleInputMaker

TextInputMaker

HepMCInput

StdHepInput

JetAlgorithm

```
addTool(JetTool * tool);
execute(Jet::jet_list_t &inputJets,
      Jet::jet_list_t &outputJets);
```

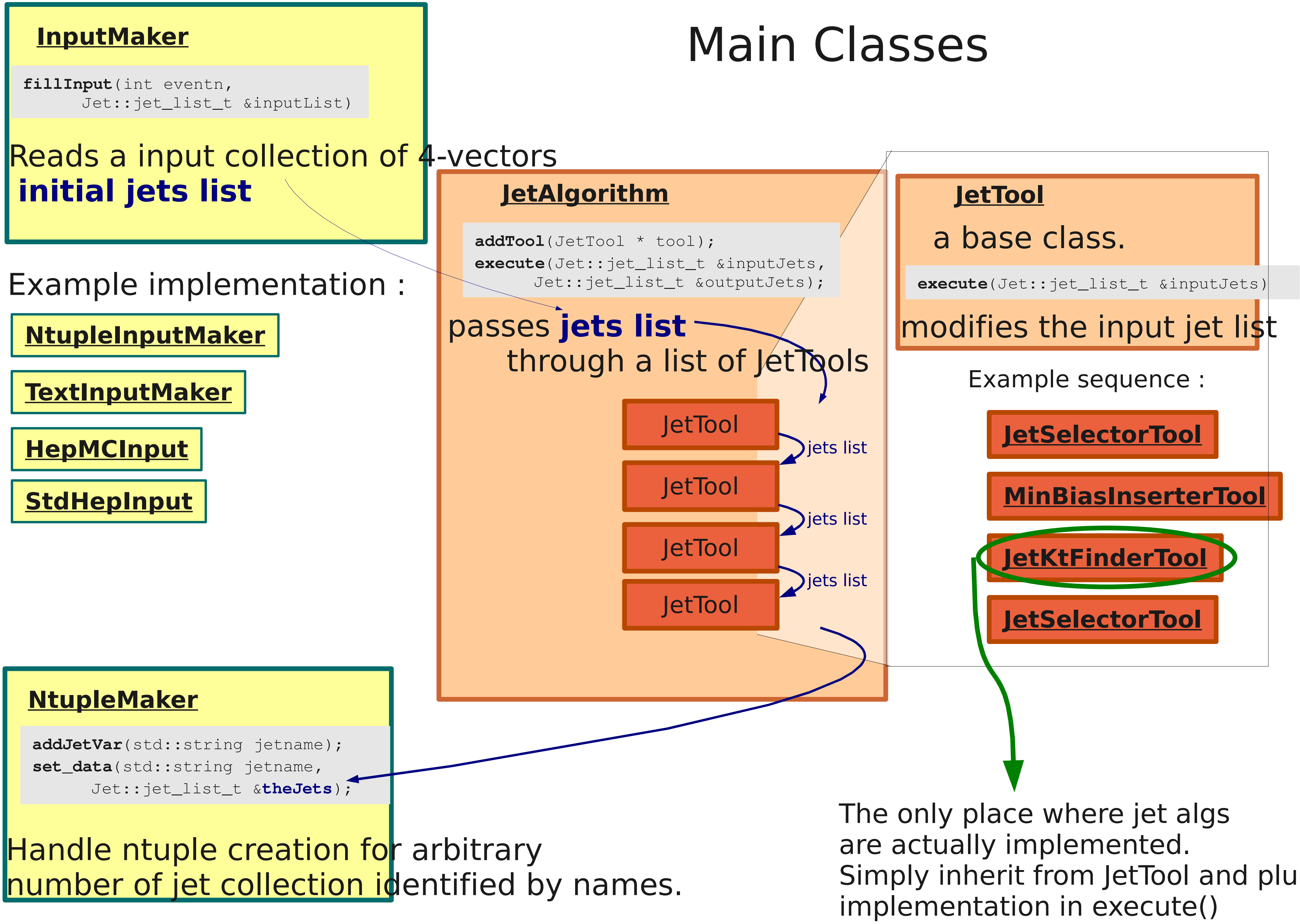
passes **jets list** through a list of JetTools

NtupleMaker

```
addJetVar(std::string jetname);
set_data(std::string jetname,
      Jet::jet_list_t &theJets);
```

Handle ntuple creation for arbitrary number of jet collection identified by names.

Main Classes



InputMaker

```
fillInput(int eventn,
      Jet::jet_list_t &inputList)
```

Reads a input collection of 4-vectors

initial jets list

Example implementation :

NtupleInputMaker

TextInputMaker

HepMCInput

StdHepInput

JetAlgorithm

```
addTool(JetTool * tool);
execute(Jet::jet_list_t &inputJets,
       Jet::jet_list_t &outputJets);
```

passes **jets list** through a list of JetTools

JetTool

a base class.

```
execute(Jet::jet_list_t &inputJets)
```

modifies the input jet list

Example sequence :

JetSelectorTool

MinBiasInserterTool

JetKtFinderTool

JetSelectorTool

The only place where jet algs are actually implemented.
Simply inherit from JetTool and plu
implementation in execute()

NtupleMaker

```
addJetVar(std::string jetname);
set_data(std::string jetname,
      Jet::jet_list_t &theJets);
```

Handle ntuple creation for arbitrary number of jet collection identified by names.

Example of use

- All code is **compiled** into libraries loadable into ROOT
- Easiest way of running : through ROOT scripts

1st step : configure input

```
// This is the main class
JetBuilder builder;

// configure an interface to the tree --------------
NtupleInputMaker input(NtupleInputMaker::EtaPhiPtE_vector_float);
// give variable names
input.set_prefix("cl_");
input.set_suffix("_caltopo");
input.set_variables("eta","phi","pt","e");
input.set_n_name("nc");
input.set_masslessMode(true); // consider clusters as massless
// Give a name to input :
input.set_name("InputJet");
// give input file and tree names
input.setFileTree("data/AtlasClustersJ5.root", "CollectionTree");

// Finally assign input class to the builder
builder.configure_input((InputMaker*)&input);
// ------------------------------------------------
```

Instanciate a builder class

Configure an input clas (set variable names, TTree and file name)

Associate input class to JetBuilder

Example of use II

2nd step : schedule jet algorithms

```
 // Schedule 2 algorithms --------------------------------

 // instanciate a jet finder tool, configure it and add it to the builder
 atlas::FastKtTool * fastkt = new atlas::FastKtTool("FastKt");
 fastkt->simple_config("Standard",0.7) ;
 builder.add_default_alg(fastkt);

 atlas::ConeFinderTool * coneF = new atlas::ConeFinderTool("ConeFinder");
 coneF->set_config(0.7,2*GeV,0.5);
 builder.add_default_alg(coneF);
 // ------------------------------------------------
```

3rd step : Run

```
// Other settings ---------------------------------
 // configure min Pt cuts for input and output jets
 builder.set_default_cut(0,2*GeV);

 // Give final file and tree name
 builder.configure_output("myTree","out.root");
 // ------------------------------------------------
 // Run !! ------------------------------------------
 builder.process_events(10);
 // ------------------------------------------------
```

What’s available now

List of Algorithms :

CDF

JetClu MidPoint

ATLAS standard algorithms:

Cone Kt (fast version)

D0

Cone

Pythia’s CellJet

+ all variants,
fully parameterizable

FastJet algorithms (from G. Salam, and M. Cacciari)

FastKt Seedless Cone (SISCone)

- *Will be the core software for jet algorithms of CMS, CDF, Atlas (before 2008) … (maybe D0?)*
- *SpartyJet has full support for algorithms and Jet areas*

What’s available now

List of Algorithms :

+ all variants,
fully parameterizable

CDF

JetClu MidPoint

ATLAS standard algorithms:

Cone Kt (fast version)

D0

Cone

Pythia’s CellJet

FastJet algorithms (from G. Salam, and M. Cacciari)

FastKt Seedless Cone (SISCone)

- *Will be the core software for jet algorithms of CMS, CDF, Atlas (before 2008) … (maybe D0?)*
- *SpartyJet has full support for algorithms and Jet areas*

Also Available :

Jet Moment framework

jet Areas, angular moment
Ysplitter
eventshape framework

Jets constituents

Timing

Working with SpartyJet

SpartyJet Interactive (recommended)

We provide :

ROOT scripts examples

PyROOT scripts example

ROOT dictionaries : access to any class from ROOT session

SpartyJet standalone

We provide Makefiles to compile standalone executables

Analyzing SpartyJet results

Results are simple ROOT ntuple

We provide analysis/visualization scripts

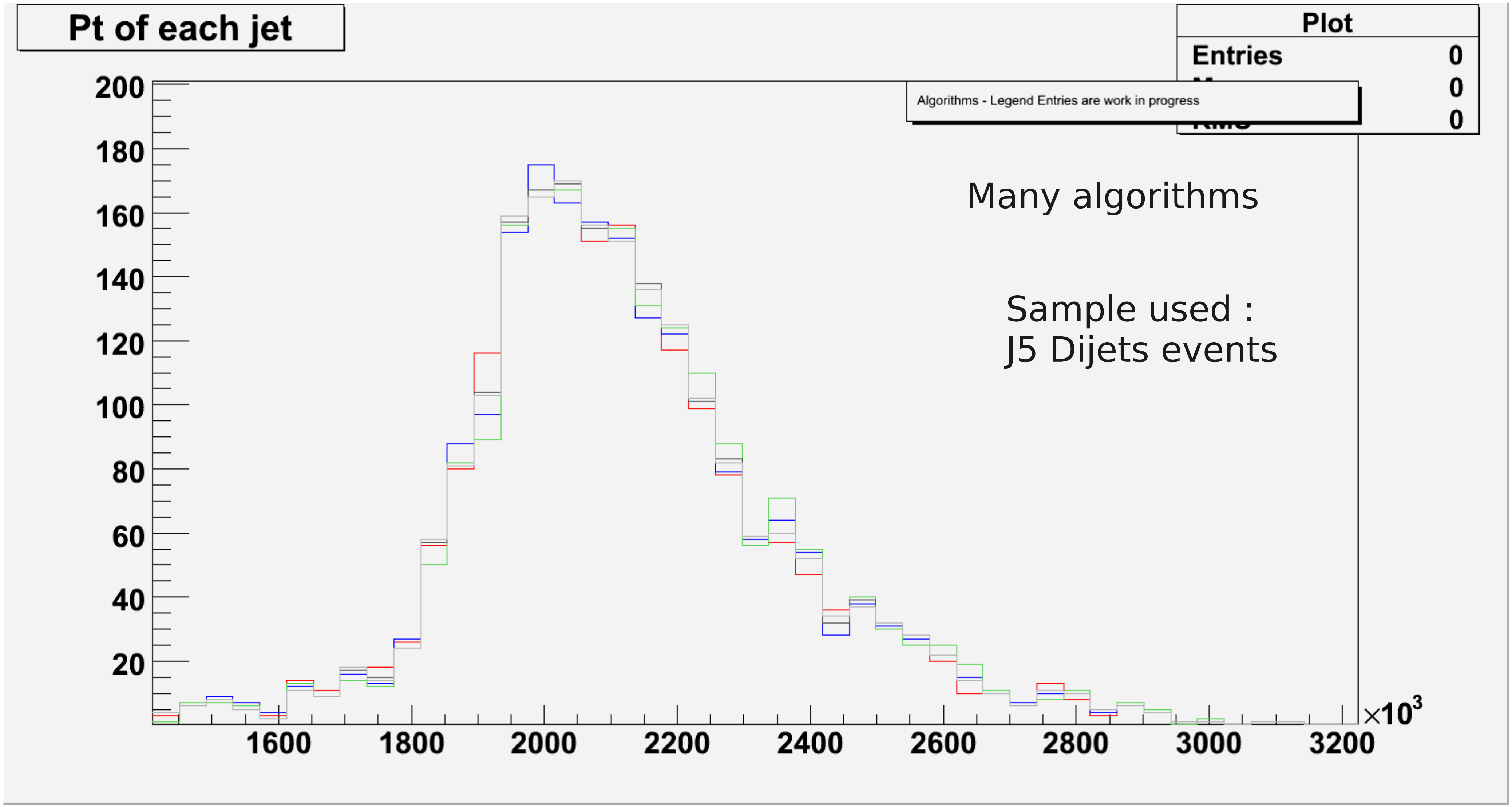
Developing with SpartyJet

Skeleton class for adding one’s own jet algorithm

Source code is provided, hopefully readable ... feel free to hack...

Illustrations

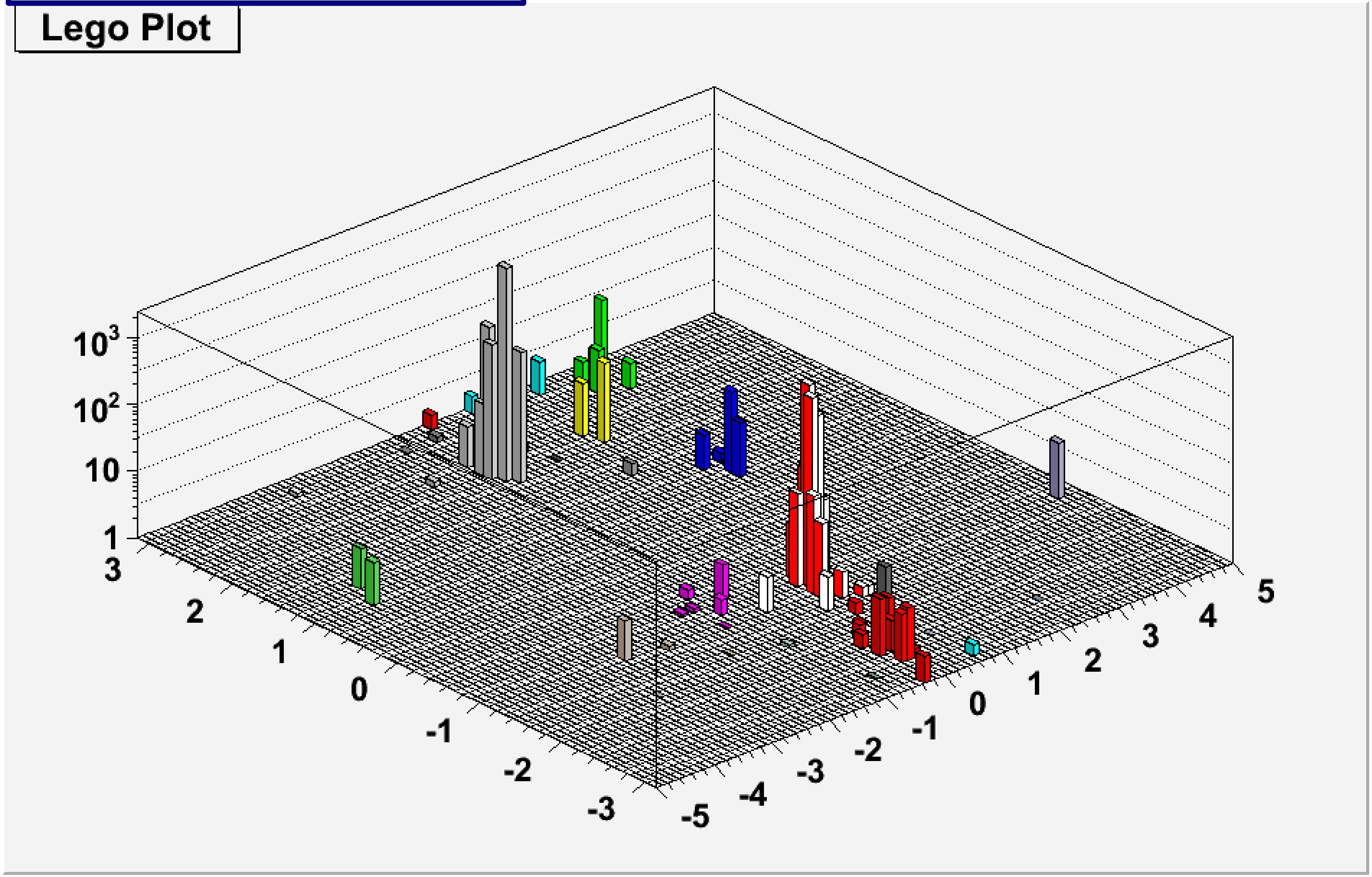
Multiple Pt distribution



Many algorithms

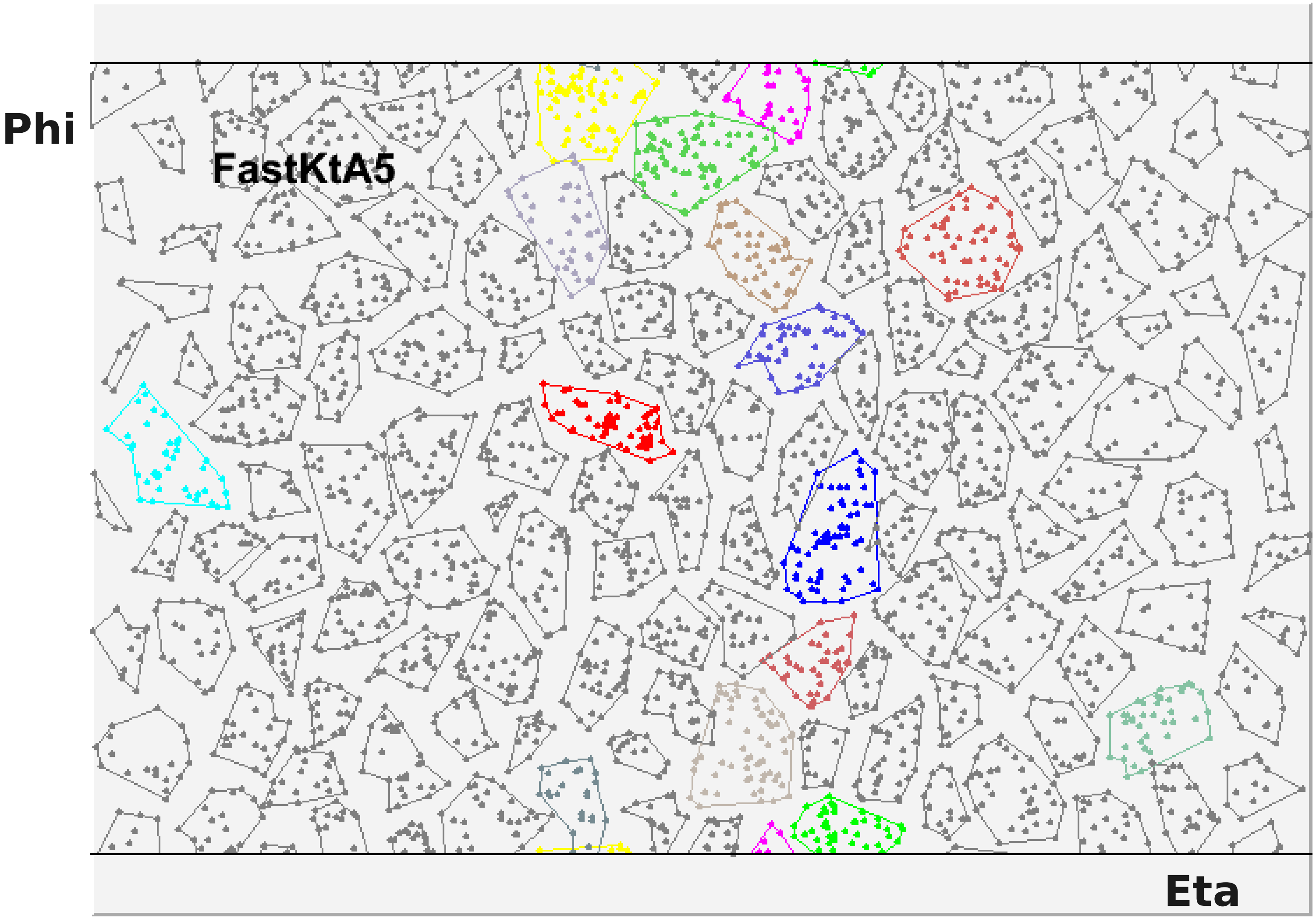
Sample used :
J5 Dijets events

Event visualization : jets positions

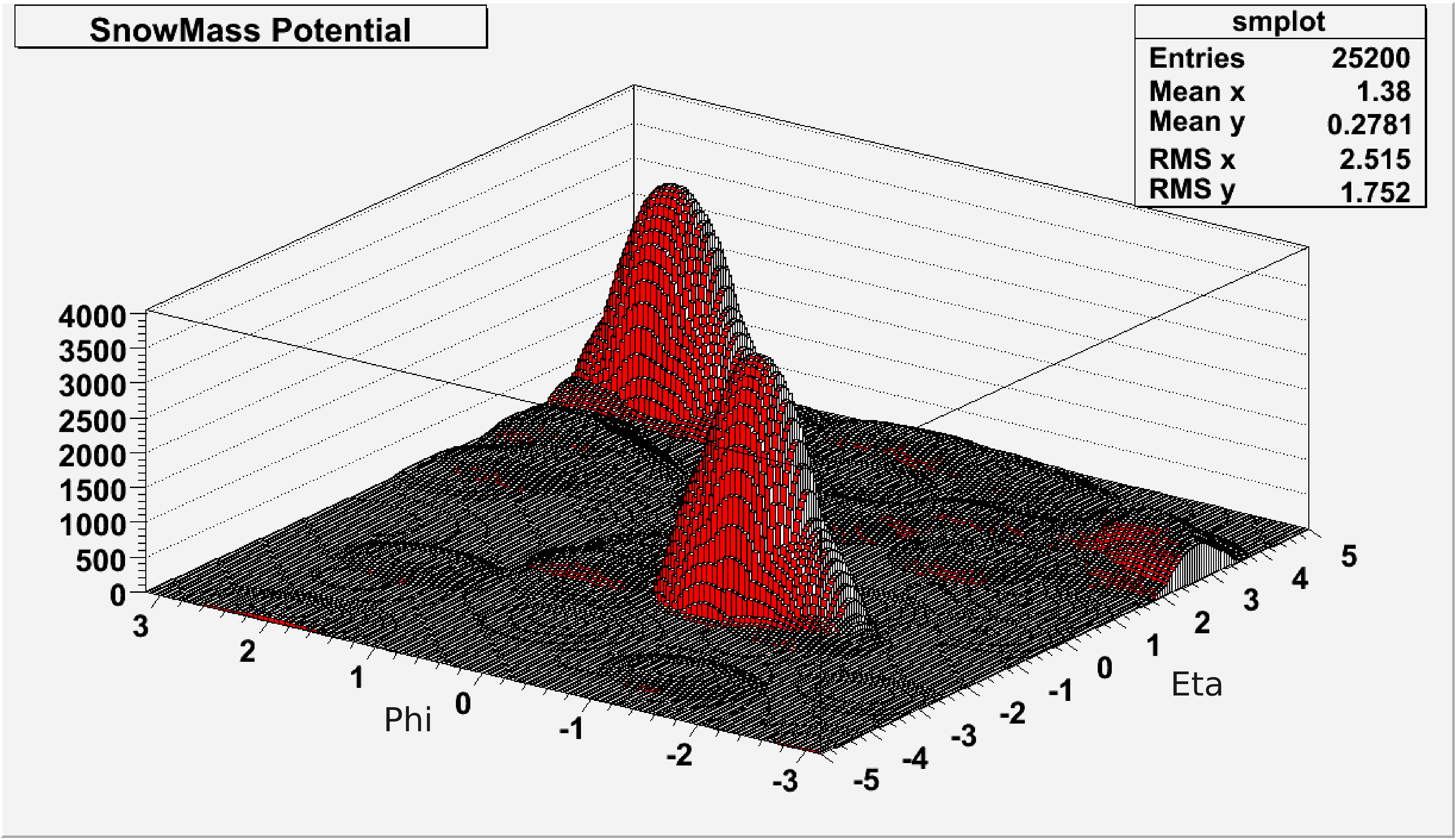


Jet constituents and jets positions

J5 sample with min-bias



Snowmass Jet Potential



Future plans

SpartyJet is already used in Atlas and CDF

We plan to have it accepted as

- Official analysis tool in Atlas
- Official tool in CDF

The core software is stable

We're working improving some aspects :

- Adding moments, event shapes
- Developing Jet algorithms
- Visualization tools : working on a simple GUI
- 'Live' jet finding

Try it out !

Jets exploration is made easy thanks to SpartyJet

Documentation, downloads:

www.pa.msu.edu/~huston/SpartyJet/SpartyJet.html

Questions/Comments are very welcome !
contact the authors