# LightJetAnalysis framework

## LightJetAnalysis: a lightweight framework to analyse flat root ntuples

We setup a lightweight framework to analyse flat root ntuples. These ntuples can (but don't need to) be derived from ATLAS D3PDs. The purpose of this framework is to make it easy for (undergraduate) students to analyse events, without having to deal with the complication (and potential overhead) of using ProofAna.

### Getting Started & Compile

### (1) Install FastJet (only if you're not running this at SLAC but on your laptop!)

Follow instructions here: http://fastjet.fr/quickstart.html

### (2) Check out the code from GitHub

```
git clone https://github.com/SLACJetWorkingGroup/LightJetAnalysis.git
```

If you run this on your local machine (and not on atlint) you need to adjust the file `setup.sh` to point to your local installation of fastjet and root.

- Note: if you have the C++ boost library installed, you can set it up in the setup.sh file. This allows you set arguments in the command line.

```
$source setup.sh
```

```
$make
```

(3) Copy input test file:

```
scp <userid>@atlint03.slac.stanford.edu:/atlas/output/pnef/skimmed.20140609.17.13_ClustersAndTruth.PythJ1to3mc12aJETMET.jetmet2012.root .
```

(1 GB in size)

A larger sample is available at:

/atlas/output/pnef//20140609.17.13_ClustersAndTruth.PythJ1to3mc12aJETMET.jetmet2012.root

### Running the code

run the executable:

```
$./runLightJetAnalysis.exe
```

if you compiled the code against the boost library, you can use

```
$./runLightJetAnalysis.exe --h
```

to see the available run options / arguments.

### Changing the MakeClass

The files `LightJetAnalysisBase.C` and `LightJetAnalysisBase.h` are generated using the `MakeClass()` method of a `TTree.`

If the branches of the `TTree` of the input file is changed, these two files need to be regenerated. For this, do the following:

```
$root -l myfile.root
```

```
root [1] myTree->MakeClass("LightJetAnalysisBase");
```

The file `LightJetAnalysisBase.h` now contains the declaration of the branches of the TTree `myTree` .

One annoying feature of MakeClass is that the arraw size is taken from the first entry of the tree. For instance, you may find lines like this:

```
Float_t Jpt[17]; //[NJetsFilled]
```

which needs to be changed into

```
Float_t Jpt[NJetsFilled];
```