Table of Contents

Combined Jet Mass

This TWiki summarises the combined jet mass definition, available for analysers in the ATLAS experiment. This new jet mass definition has been developed by the Jet Substructure and Jet-by-Jet tagging subgroup of the JetEtMiss CP group, and is available in the latest tag of JetCalibTools.

1. What's a *Combined* Jet Mass ?

In ATLAS, the jet mass is one of the most important substructure variables we have when we want to describe and study hadronic jets. The standard jet mass definition has, for some time, been the *calorimeter* jet mass, m_{calo}. This is the invariant mass of the sum of the four-momenta of the individual calorimeter topo-clusters which are associated to the jet via a jet reclustering algorithm (by default the Anti-k_t algorithm, in ATLAS). This calorimeter mass is calculated at both the EM and LC scales, and for different jet sizes (where the jet size is quantified in terms of a *jet radius*, *R*).

In order to take advantage of regions of phase space where the calorimeter resolution is sub-optimal, a second jet mass definition has recently been added to ATLAS derivations: the *track-assisted* jet mass, m_{TA}. In the track-assisted approach, tracks from the inner detector are first *ghost-associated* to jets in the calorimeter, and the sum of the masses of the individual tracks associated to a calorimeter jet yields the track mass, m_{track}, for that jet. The track-assisted mass is then calculated by multiplying the track mass with the "charged/neutral" fraction $p_{T,calo}/p_{T,track}$: $m_{TA} = m_{track} \times p_{T,calo}/p_{T,track}$.

The calorimeter and track-assisted jet mass definitions will give the smallest jet mass resolution at different values of mass and transverse momentum, and the exact behaviour of their resolutions will vary from jet topology to jet topology. Is it possible to find a way to optimise the jet mass, by taking a linear combination of the m_{calo} and m_{TA}, such that the final jet mass has a lower jet mass resolution than the individual calorimeter and track-assisted masses? Yes! Hence the combined jet mass.

The combined jet mass, m_{comb}, is the *linear* combination of m_{calo} and m_{TA} which minimises the jet mass resolution: $m_{comb} = a \times m_{calo} + b \times m_{TA}$, where the weights *a* and *b* are to be found.

2. Determining the *a* and *b* Weights

2.1 Neglible Response Correlations

Using the constraint $a + b = 1$, and minimising the combined mass resolution, the master equations for *a* and *b* follow immediately. The exact form of the master equations depends on the *correlation*, , between the calorimeter and track-assisted jet mass response, m_{calo}/m_{truth} and m_{TA}/m_{truth} respectively. Assuming negatigible correlation, the master equations become:

$a = \sigma_{calo}^{-2}/(\sigma_{calo}^{-2} + \sigma_{TA}^{-2})$

$b = \sigma_{TA}^{-2}/(\sigma_{calo}^{-2} + \sigma_{TA}^{-2})$

Here the -values refer to the different *jet mass resolutions*. The mass resolution is defined to be 68 % confidence interval of the interquantile range of the jet mass response distrbution. Therefore, in order to determine the weights, one must first calculate the jet mass resolution. The jet mass resolutions for the calorimeter and the track-assisted masses are determined as a functon of the $p_{T,calo}$, and $m_{reco}/p_{T,calo}$ (a single | | bin is used). Two resolution maps are required: the calorimeter resolution map ($m_{reco = calo}$) and the track-assisted resolution map ($m_{reco = TA}$). The weights, binned in $p_{T,calo}$, $m_{reco}/p_{T,calo}$, then follow immediately from these maps.

2.2 Non-neglible Response Correlations

If the correlation is non-negligible (a working definition of non-negligible is a | | > 0.3 between the calorimeter and track-assisted mass responses), then the *a* and *b* weights must be calculated using three maps: the two resolution maps, and a correlation map. The correlation map is binned in $p_{T,calo}, m_{TA}/p_{T,calo}$. Since the correlations are a second-order effect, the map has coarser binning compared to the resolution maps. The final, *correlated a* and *b* are then given by:

a = (_{TA}^{2} -   _{calo} _{TA})/(_{calo}^{2} +   _{TA}^{2} - 2   _{calo} _{TA})

b = 1 - *a*

Adding in correlations can give rise to *negative* weights. The sum of the two weights (for a given $p_{T}, m/p_{T}$ bin) remains unity. It can be shown that for correlations   > _{TA}/ _{calo}, a negative calorimeter weight is obtained.

3. Recommended Resolution and Correlation Maps

3.1 Current Recommendations

The recommended resolution and correlation maps have bee derived using the JETM8 derivation. A QCD dijets sample was used. The final recommended maps have the following properties:

1. Binning is in $p_{T,calo}$: 0.0 - 3000.0 GeV
2. Binning is in $m_{reco}/p_{T,calo}$: 0.0 - 1.0
3. TA resolution map: $m_{reco} = m_{TA}$
4. Calorimeter resolution map: $m_{reco} = m_{calo}$
5. Correlation map: $m_{reco} = m_{TA}$
6. No cut on the jet mass response is applied when calculating the jet mass resolutions
7. A single | | bin: 0.0 - 2.0
8. If correlations are not being considered, the weights are calculated from the two resolution maps
9. If correlations are being considered, the weights are calculated from the two resolution maps and the correlation map
10. Gaussian kernel smoothing is applied to the three maps, with 50 x-bins and 50 y-bins.

[INSERT LINKS TO THE THREE SMOOTHED MAPS HERE]

3.2 Future Recommendations

Ongoing work focuses on making new resolution and correlation maps which incorporate *different sample topologies*. In addition to the QCD dijets sample, the JETM8 derivation is also being used to develop resolution and correlaton maps for three additional topologies:

1. W' -> W/Z jets
2. Z' -> t tbar jets
3. Randall-Sundrum graviton -> HH -> b bbar b bbar jets

4. The Framework

The CombinedJetMass framework can be obtained directly from the JSS git repository using the commands:

[GIT COMMANDS HERE]

The framework consists of three central packages: NTupleMaker, JetInspector, and BinHandler. These have been adapted from the infrastructure used to perform Monte Carlo jet mass calibration in ATLAS.

4.1

The NTupleMaker is an EventLoop -based n-tuple generator. Is can generate n-tuples locally and on the grid. It is recommeneded that n-tuple generation is performed on the grid. To do this, run the command:

source NTupleMaker /scripts/runGrid.sh

In runGrid.sh the particular samples to run on can be specified by the $SAMPLES = "" bash command, where the pathway to the sample is provided in the quotations. Samples are stored as text files in the NTupleMaker /scripts/Samples/ directory. The $JETCOLLECTION bash command can be modified on different jet collections. The framework currently supports n-tuple generation of the AntiKtLCTopoTrimmedPtFrac5SmallR20, AntiKt4EMTopo, and particle flow jet collections. The n-tuples, once processed on the grid, can then be requested using rucio download:

rucio download <n-tuple name>

It is recommended that the n-tuple outputs are stored in the following way:

<path to output>/OutputDirectory/<SampleName>/<n-tuples go here>

4.2

This is based on the JetInspector package used in Monte Carlo jet mass calibration. The type of binning and size of the variable bins in specified in the script BinHandler /Root/BinHandler.cxx. The central source code is JetInspector /Root/JetInspector.cxx. This code loops over the reconstructed jets in the output n-tuple, applying standard jet cleaning cuts, truth-matching, jet flavour-matching and isolation conditions between reconstructed jets and truth jets. This framework is designed to run on Monte Carlo only. JetInspector then calculates the jet mass responses, and the correlations between the calorimeter and track-assisted mass responses. The running scripts are stored in the JetInspector /python directory. To run on a batch system:

python JetInspector /python/launch_bsub.py (or launch_local.py to run locally)

Within thee lunch scirpts the top directory <path to directory>/OutputDirectory/<SampleName> must be specified to that SampleHandler may run over the n-tuples correctly. To run over several subdirectories of <SampleName>, specify the directory name in the relvant subdirectory python lists in launch*. The particular binning of the configuration is set using the string:

<sample name>_<jet collection>_<m/p_{T} bin string>_mOpt_<p_{T} bin string>-pt_<   bin string>_abseta_fidcut0

In order to produce the inputs for *two* resolution maps, the JetInspector script has to be run twice: once for $m_{reco} = m_{calo}$ and once for $m_{reco} = m_{TA}$. To bin in $m_{calo}/p_{T,calo}$ (for making the calorimeter mass resolution map) uncomment the settings:

if (theMvar == "m") {

theMval = thetruthjet_m;

theRecoMval = therecojet_m;

}

4. The CombinedJetMass Framework   3

```
if (theMvar == "mOpt") {

theMval = thetruthjet_m/thetruthjet_pt;

theRecoMval = therecojet_m/therecojet_pt;

}
```

To bin in m_{TA}/p_{T,calo} (for making the track-assisted mass resolution map and the correlation map) instead uncomment the settings:

```
if (theMvar == "m") {

theMval = thetruthjet_m;

theRecoMval = therecojet_trkassistmass;

}

if (theMvar == "mOpt") {

theMval = thetruthjet_m/thetruthjet_pt;

theRecoMval = the recojet_trkassistmass/therecojet_pt;

}
```

The outputs are saved in JetInspector /outputs. The output contains, for each p_{T}, m/p_{T}, and | | bin, the correlation factor between the calorimeter and track-assisted jet mass response. This will be used as the input to create the final correlation map (see 4.2.2). However, in order to first perform the resolution calculation, a few additional directories are required: JetInspector /Fit, JetInspector /rootfiles, and JetInspector /Summary. We shall now look at each of them in turn.

4.2.1 /Fit

The JetInspector /Fit/RunFit.cxx code is responsible for making ATLAS-standard plots of the calorimeter and track-assisted jet mass responses (used for jet mass calibration) and calculating the jet mass resolution in each p_{T}, m/p_{T}, and | | bin. For each response distribution (where a single response distribution is produced for each p_{T}, m/p_{T} (or just m, depending on the user's inputs), and | | bin) the 68 % confidence interval interquantile range is calculated. The jet mass resolution is then defined to be this 0.5 of the interquantile range, divided by the median of the response. The interquantile range and median are then stored as sets of TH1s in the directory JetInspector /rootfiles. The exact output destination can be directly modified by the user in RunFit.cxx.

4.2.2 /rootfiles

The JetInspector /rootfiles directory contains the outputs for making the final jet mass resolution maps. The output file is of the form <user-defined inputs>_OUTPUT.root.

4.2.3 /Summary

The JetInspector /Summary directory contains several importsnt sumary scripts which are designed to create the final resolution and correlation maps. Let's summarise each important plotting script in turn. The user is encouraged to read through the code. In each script, the only modifications required of the user are the

pathways to the input and output files. By default, the input pathways are set to the JetInspector /rootfiles directory. The recommendation is that the user specifies the outputs to exist in (or be a subdirectory of) JetInspector /Summary/Plots.

The important plotting scripts:

1. SummaryMaker.cxx: this makes a set of summary histograms where the *truth-binned* variables are used. This piece of code is not relevent to the final resolution and correlation map calculation, and is included here for completeness.
2. RecoSummaryMaker.cxx: this makes the TH2s which correspond to the jet mass resolutions, binned in $p_{\{T,calo\}}$, $m_{\{reco\}}/p_{\{T,calo\}}$, and | |. For each | | bin, two resolution maps are produced (calorimeter and track-assisted resolution maps, respectively).
3. getIndivdualPlots.py: this takes the output of RecoSummaryMaker.cxx and saves each TH2 resolution map to a separate root file. For two resolution maps, it produces two outputs: Calo_Res.root and TA_Res.root.
4. makeWeights.py: this takes the two (for a single | | bin we have only two) resolution maps and calculates the final weights, assuming that correlations are negligible (see 2.1). The relevant command to run is python runWeights.py Calo_Res.root TA_Res.root.
5. makeCorrelationMap.py: this takes the initial correlations stored in the JetInspector /output directory, and produces a single ATLAS-style correlation map (saved as a pdf).
6. makeResolutionPlots.py: this takes the TH2 resolution maps and converts them into ATLAS-style pdf plots.
7. makeFinalWeightPlots.py: this takes the TH2 resolution maps (not the correlation map) and makes ATLAS-style calorimeter and track-assisted weight plots, saved as pdfs.
8. makeCorrelatedWeights.py: this takes the resolution map produced by 2. and the correlations in JetInspector /output, and produces weight maps where the correlations are taken into account. They sare saved as both ROOT TH2s, and as ATLAS-style pdfs.

The JetInspector /Summary/ComparisonStudies directory contains a few additional python scripts which allow the user to make comparisons between different resolution and jet mass response plots. They are not integral to the calculation of the correlation and resolution maps, and are included here for completeness.

This topic: Sandbox > MichaelEdwardNelsonSandbox
Topic revision: r4 - 2016-10-17 - MichaelEdwardNelson