

# Table of Contents

<b>BottomUpUncerts.....</b>	<b>1</b>
<b>Introduction.....</b>	<b>2</b>
<b>Inclusive Uncertainties.....</b>	<b>3</b>
Cluster Reconstruction Efficiency.....	3
Cluster Fake Rate.....	3
Cluster Energy Scale.....	3
Cluster Energy Resolution.....	5
Cluster Position Resolution.....	5
Isolated cluster splitting / merging.....	6
<b>Uncertainties Inside Dense Environments.....</b>	<b>7</b>
Cluster Energy Scale and Isolation:.....	7
Cluster Splitting/Merging.....	7
<b>Uncertainties for Hadrons that are not Pions.....</b>	<b>9</b>

# BottomUpUncerts

# Introduction

The purpose of this page is to document recommendations for bottom up uncertainties. All of these uncertainties are applied at the cluster level and then the analysis variables re-computed. This is contrast to the top-down approach where jet observables are directly calibrated. The top-down uncertainties for the trimmed jet mass and several jet shapes are described in this twiki. Reasons you may need to use the uncertainties described on this page instead of the top-down ones:

- You are very sensitive to (or want to exploit) correlations between uncertainties.
- Your observable is not on the list covered by top-down uncertainties.
- You are sensitive to more than just the scale of a jet shape (e.g. you are doing a precision jet substructure measurement).
- You want to perform a measurement which differentiates between MC generators (usually takes as a systematic uncertainty in the top-down approach).

**For analyses using AntiKt10 jets, unless you are looking at colour-flow outside of your cone (or other similar effects), these uncertainties can all be applied to the jet constituents and the sub-structure re-computed without re-doing jet clustering. If you are using smaller radius jets or looking at energy flow across the jet boundary you need to apply these variations to all clusters and re-build your jets. In all cases, the variations should be applied prior to any grooming.**

For uncertainties on track-based observables, see the recommendations from the tracking CP group.

We highlight 4 different things in the sections below:

- green: These are uncertainties that have been updated in release 21 and form recommendations.
- purple: These are uncertainties that have been updated in Run II but not in the latest release, or come from test-beam data. Therefore these can be treated as uncertainties although we hope people will volunteer to re-do these studies to reduce these uncertainties (and make them more reliable).
- blue: These are suggested checks to see if an analysis is sensitive to a given effect. These cases are the "known unknowns" where if the check results in a minor uncertainty it can be treated as such, otherwise further study is required.
- red: These are brief instructions on how we would go about updating and/or producing a new recommendation for this uncertainty. Also included are the contact details of anyone currently working on this.

A discussion of the proposals in this twiki was presented at a recent Jet/MET meeting [↗](#).

# Inclusive Uncertainties

## Cluster Reconstruction Efficiency

This uncertainty is to cover potential differences in the data-to-MC modeling of in the efficiency of particles seeding a topological cluster. It is measured by looking at tracks that have no cluster associated to them in low-mu data. This depends heavily on the clustering threshold. Current numbers, shown below (Sec. 8.4.1 [↗](#)) are from 2015 and are based on special low threshold ( $\mu = 0$ ) runs.

Procedure: Inflate the current uncertainty by 100% as this measurement was performed with  $\mu=0$  clustering threshold and in release 20.7. If you are sensitive to this uncertainty, then please contribute to re-measuring this with the 2016-17 thresholds.

```
cluster_scale = (TH2F*)cluster_file->Get("Scale");
for constituent in jet do
  double E = pt*cosh(eta)/1000.; //GeV
  int Ebin = cluster_scale->GetXaxis()->FindBin(E);
  int ebin = cluster_scale->GetYaxis()->FindBin(abs_eta);
  if (Ebin > cluster_scale->GetNbinsX()) Ebin = cluster_scale->GetNbinsX();
  if (Ebin < 1) pbin = 1;
  if (ebin > cluster_scale->GetNbinsX()) ebin = cluster_scale->GetNbinsX();
  if (ebin < 1) ebin = 1;
  double p = E;
  if (cluster_scale->GetBinContent(Ebin,ebin) > 0) p = E/cluster_scale->GetBinContent(Ebin,ebin);
  double r = 0.;
  if (abs_eta < 0.6) r = (0.12*exp(-0.51*p)+4.76*exp(-0.29*p*p))/100.;
  else if (abs_eta < 1.1) r = (0.17*exp(-1.31*p)+4.33*exp(-0.23*p*p))/100.;
  else if (abs_eta < 1.4) r = (0.17*exp(-0.95*p)+1.14*exp(-0.04*p*p))/100.;
  else if (abs_eta < 1.5) r = (0.15*exp(-1.14*p)+2768.98*exp(-4.2*p*p))/100.; //bad fit, but
  else if (abs_eta < 1.8) r = (0.16*exp(-2.77*p)+0.67*exp(-0.11*p*p))/100.;
  else if (abs_eta < 1.9) r = (0.16*exp(-1.47*p)+0.86*exp(-0.12*p*p))/100.;
  else r = (0.16*exp(-1.61*p)+4.99*exp(-0.52*p*p))/100.;
  double flip = myrand_global->Uniform(0.,1.);
  if ((flip < r) && (E/1000. < 2.5)) skip!
end do;
```

where the TFile is cluster\_uncert\_map\_LCW.root for LCW or cluster\_uncert\_map\_EM.root for EM.

This is a very straightforward measurement to repeat in release 21. We took a large min-bias data sample in the low-mu runs towards the end of 2017 and have re-processed these with the cluster threshold set to the same as that in the nominal physics running. Using this min-bias data it should be possible to update this measurement with high statistics looking at the fraction of tracks that have matched clusters as a function of pT and eta.

## Cluster Fake Rate

Spurious clusters from pure noise are negligible, but the contribution from pileup clusters can have an important impact on some observables.

Based on studies of the cluster multiplicity as a function of pileup, shift  $\mu$  up and down by 15% instead of the standard procedure that is much less than this.

## Cluster Energy Scale

The cluster energy scale is found by matching clusters to isolated tracks in low-mu data. A truncated gaussian is fit of the distribution of E over p, and the peak gives the scale in data and in simulation. The difference in

the ratio of these scales from unity, along with uncertainties on the subtraction of backgrounds gives the uncertainty. Outside of the range of E/p data, combined test beam (CTB) measurements are used, and outside of these ( $p > 350$  GeV), there is a large (10%) out of range uncertainty. These are the same uncertainties propagated to jets that are used for the current JES uncertainty beyond the in situ calibrations.

Current numbers (Sec. 8.4.2) can be found below. Note that they are also with  $\mu = 0$  thresholds. The prescriptions are only completed for LCW - if you use EM, please get in contact.

Prescription: Consider a one-component shift in the cluster energies with the above numbers. The shift up and down may lead to asymmetric results in the final observable. Apply this to all clusters coherently treating them as pions (this is thought to be conservative compared to only applying this to hadronic-like clusters).

```
TFile* cluster_file = TFile::Open("cluster_uncert_map.root");
cluster_means = (TH2F*)cluster_file->Get("Mean");
cluster_scale = (TH2F*)cluster_file->Get("Scale");
for constituent in jet do
    double E = pt*cosh(eta)/1000.; //GeV
    int Ebin = cluster_scale->GetXaxis()->FindBin(E);
    int ebin = cluster_scale->GetYaxis()->FindBin(abs_eta);
    if (Ebin > cluster_scale->GetNbinsX()) Ebin = cluster_scale->GetNbinsX();
    if (Ebin < 1) pbin = 1;
    if (ebin > cluster_scale->GetNbinsX()) ebin = cluster_scale->GetNbinsX();
    if (ebin < 1) ebin = 1;
    double p = E;
    if (cluster_scale->GetBinContent(Ebin,ebin) > 0) p = E/cluster_scale->GetBinContent(Ebin,ebin);
    int pbin = cluster_means->GetXaxis()->FindBin(p);
    if (pbin > cluster_means->GetNbinsX()) pbin = cluster_means->GetNbinsX();
    if (pbin < 1) pbin = 1;
    double myCES = fabs(cluster_means->GetBinContent(pbin,ebin)-1.);
    if (p > 350) myCES = 0.1;
    double ptcesu = pt*(1.+ myCES); //CES up
    double ptcesd = pt*(1.-myCES); //CES down
    //set px, py, pz, E like this: ptcesu*cos(phi), ptcesu*sin(phi), ptcesu*sinh(eta), ptcesu*cosh(eta)
end do;
```

where the TFile is cluster\_uncert\_map\_LCW.root for LCW or cluster\_uncert\_map\_EM.root for EM.

*Cross-checks:* - the idea of these is to check the correlation model which in the above recommended prescription is very simple. The uncertainty of each of these changes in the studied variables should be the same or smaller than the standard prescription to pass the check.

1. Apply the above uncertainty only to clusters with EM\_PROBABILITY<0.9 and 0.5% uniformly to all others - this is to check if you are sensitive to the hadronic shower scale being wrong and not the EM one.
2. Treat this uncertainty as two uncorrelated uncertainties one which is applied to clusters with EMfrac>0.5 and one which is applied to clusters with EMfrac<0.5.
3. Treat this uncertainty as two uncorrelated uncertainties one which is applied to clusters with letal>1.4 and one applied to letal<1.4
4. Treat this uncertainty as two uncorrelated uncertainties one which is applied to clusters with pT>2.5GeV and one applied to pT<2.5GeV
5. Treat the out-of-test-range (not testbeam and not E/p) uncertainty uncorrelated with the rest of the energy scale uncertainties.

The cluster energy scale may depend on many features and this dependence may be mis-modeled (akin to mis-modeling the dependence of the JES on ntrack in the GSC) and/or the distributions of the features may also be mis-modeled (akin to mis-modeling ntrack). This is a second-order effect if the inclusive response is well-constrained, but in principle this may eventually be important. The most important features to consider are those used in the calibration itself, such as the LC classification.

The Data/MC agreement of the LC classification has not been studied so we would like to see the data/MC agreement for the same set of clusters (after event and jet selection) at the EM-scale computing the relevant variables for the analysis. The level of agreement (data-to-MC) should not be worse at EM scale. (If it is then we need to study further).

We would like this measurement to be re-performed in the 2017 low-mu data. We ensured that a very large data sample of min-bias data was taken such that good statistics would be available for this measurement.

## Cluster Energy Resolution

**Method:** match clusters to tracks, fit the distribution to a truncated Gaussian, and take the standard deviation as the resolution in data and in simulation. The square root of the squared difference from the ratio of these resolutions is the uncertainty. Outside of E/p data, we have combined test beam (CTB) and outside of that ( $p > 350$  GeV), there is a large (10%) out of range uncertainty.

Current numbers (Sec. 8.4.2) can be found below. Procedure: smear all clusters randomly with the above values. Same warnings apply as for the cluster energy scale however they have a smaller impact. In particular decorrelating the uncertainty is unnecessary.

```
TFile* cluster_file = TFile::Open("cluster_uncert_map.root");
cluster_rmss = (TH2F*)cluster_file->Get("RMS");
cluster_scale = (TH2F*)cluster_file->Get("Scale");
for constituent in jet do
    double E = pt*cosh(eta)/1000.; //GeV
    int Ebin = cluster_scale->GetXaxis()->FindBin(E);
    int ebin = cluster_scale->GetYaxis()->FindBin(abs_eta);
    if (Ebin > cluster_scale->GetNbinsX()) Ebin = cluster_scale->GetNbinsX();
    if (Ebin < 1) pbin = 1;
    if (ebin > cluster_scale->GetNbinsX()) ebin = cluster_scale->GetNbinsX();
    if (ebin < 1) ebin = 1;
    double p = E;
    if (cluster_scale->GetBinContent(Ebin,ebin) > 0) p = E/cluster_scale->GetBinContent(Ebin,ebin);
    int pbin = cluster_rmss->GetXaxis()->FindBin(p);
    if (pbin > cluster_rmss->GetNbinsX()) pbin = cluster_rmss->GetNbinsX();
    if (pbin < 1) pbin = 1;
    double myCER = fabs(cluster_rmss->GetBinContent(pbin,ebin));
    if (p > 350) myCER = 0.1;
    double ptcer = pt*(1.+myrandom->Gaus(0,1)* myCER);
    //set px, py, pz, E like this: ptcer*cos(phi), ptcer*sin(phi), ptcer*sinh(eta), ptcer*cosh(eta)
end do;
```

where the TFile is cluster\_uncert\_map\_LCW.root for LCW or cluster\_uncert\_map\_EM.root for EM.

We would like this measurement to be re-performed in the 2017 low-mu data. We ensured that a very large data sample of min-bias data was taken such that good statistics would be available for this measurement. This could be parameterized in terms of eta and which calorimeter they were in etc.

## Cluster Position Resolution

Studies in 2011 found that 5 mrad was an estimate of the uncertainty on the angular resolution of cluster energy. Studies in 2012 (p46 here) saw that this was extremely conservative but did not cover all phase space. Work is ongoing to update this with Run II data (JIRA). The prescription is as follows:

```
for constituent in jet do
    double phi_smear = myrand_global->Gaus(phi,0.005); //5 mrad
    double eta_smear = myrand_global->Gaus(eta,0.005);
    double pt_smear = pt*cosh(eta)/cosh(eta_smear);
    //set px, py, pz, E like this: pt_smear*cos(phi_smear), pt_smear*sin(phi_smear), pt_smear*cosh(eta_smear)
```

end do;

## **Isolated cluster splitting / merging**

See prescription for dense environments. At the moment, same prescription is applied for both.

# Uncertainties Inside Dense Environments

## Cluster Energy Scale and Isolation:

This is something we have worried about in the past. There is out-of-cluster energy which is different if you are in a isolated and non-isolated environment. I am not sure how to go about testing this at the moment so we might need to ignore this for now. (If the scale uncertainties are large then this covers this effect).

For your observable X, make a TProfile plot of vs  $\text{Sum}\{E \cdot \text{ISOLATION}\} / \text{Sum}\{E\}$  (sums are over constituent clusters) for the measured jet in data and MC. If large trends in the data/MC agreement are seen worry.

This could be measured in 3-track tau events, and K-short resonances where the mass can be plotted as a function of isolation.

## Cluster Splitting/Merging

This is clearly something that could be important and we have no measurements of this so we need to make sure that we aren't sensitive to these effects. First, define sigma to be the effective cluster size (Eq. 20 here<sup>20</sup>):

```
sigma = atan(sqrt(cluster->auxdata<float>("SECOND_R"))/cluster->auxdata<float>("CENTER_MAG")) * cos
```

Also define:

```
E_EMB3 = cluster->auxdata<vector<float>> >("EnergyPerSampling") [3];  
E_Tile0 = cluster->auxdata<vector<float>> >("EnergyPerSampling") [12];  
E_EME3 = cluster->auxdata<vector<float>> >("EnergyPerSampling") [7];  
E_HEC0 = cluster->auxdata<vector<float>> >("EnergyPerSampling") [8];  
i_EMax = argmax cluster->auxdata<vector<float>> >("EnergyPerSampling") [i];
```

Then, we have the following procedure

(i) Splitting between HAD and EM:

```
vector merged_constituents;  
vector already_merged;  
for i in constituents do  
    already_merged[i] = false  
end do  
  
for i in constituents do  
    if (already_merged[i])  
        continue  
    end if  
    for j in constituents do  
        if i == j  
            continue  
        end if  
        if (DeltaR(constituent[i], constituent[j]) > sigma[i]+sigma[j]) #close to each other tr  
            continue  
        end if  
        if ((E_EMB3[i] !=0 && E_EMB3[j] !=0) || (E_Tile0[i] !=0 && E_Tile0[j] !=0) || (E_EME3  
            already_merged[i] = true  
            already_merged[j] = true  
            #merged_constituent has eta = (eta[i]*E[i]+eta[j]*E[j])/(E[i]+E[j]), phi = (phi[i]*  
            merged_constituents.push_back(merged_constituent)  
        end if  
    end do  
    if (!already_merged[i])  
        merged_constituents.push_back(constituents[i]);  
    end if  
end do
```

## (ii) Hadronic cluster splitting in the EM calorimeter.

```

vector merged_constituents;
vector already_merged;
for i in constituents do
    already_merged[i] = false
end do

for i in constituents do
    if (already_merged[i])
        continue
    end if
    for j in constituents do
        if i == j
            continue
        end if
        if (DeltaR(constituent[i], constituent[j]) > sigma[i]+sigma[j]) #close to each other tr
            continue
        end if
        if ((EM_PROBABILITY[i] < 0.9 && EM_PROBABILITY[j] < 0.9) && ((i_EMax[i] == 2 && (i_EMa
            already_merged[i] = true
            already_merged[j] = true
            #merged_constituent has eta = (eta[i]*E[i]+eta[j]*E[j])/(E[i]+E[j]), phi = (phi[i]*
            merged_constituents.push_back(merged_constituent)
        end if
    end do
    if (!already_merged[i])
        merged_constituents.push_back(constituents[i]);
    end if
end do

```

## (iii) Cluster splitting.

```

vector split_constituents;

for i in constituents do
    if (rand.Uniform < 0.2 && EM_PROBABILITY[i] < 0.9 && (i_EMax[i] == 2 || i_EMax[i] == 6)) t
        #split_constituent1 has eta = eta[i], phi = phi[i] + sigma[i]/2, E = E[i]/2
        #split_constituent2 has eta = eta[i], phi = phi[i] - sigma[i]/2, E = E[i]/2
        split_constituents.push_back(split_constituent1)
        split_constituents.push_back(split_constituent2)
    else
        split_constituents.push_back(constituents[i]);
    end if
end do

```

For all of a), b), c), plot the cluster multiplicity and N95 (the number of clusters required to reach 95% of total jet energy) in your jets before and after applying the change in MC, and overlay the data (without any modifications to the data distribution).

If any of the above tests shows strong dependencies on the results then we need to study this in more detail. Studying this in detail is not too difficult and we should do this soon if there are lots of sub-structure analyses which see significant effects from the above tests. Note that in fig.31/32 of <https://arxiv.org/pdf/1603.02934.pdf> we see cluster multiplicity is not well modeled (even with overlay pile-up) so we don't have great constraints on the rate of splitting/merging at the moment.

The rates of this splitting/merging can be measured at low pT looking in the low-mu data at isolated tracks and at high pT in hadronic tau events.

# Uncertainties for Hadrons that are not Pions

The only relevant contribution is for K long. The procedure is to follow the approach used for the high  $p_T$  JES uncertainty, where there is an uncertainty from neutral kaon showers resulting from Geant4 physics list variations as seen in 1203.1302 [↗](#).

For each K long found in the truth record (PDG ID = 130) add an additional pseudo-cluster with 20% of the energy of the kaon in the direction of the truth kaon. This represents a 20% (conservative) uncertainty on the energy response of this particle due to the absence of measurements in test beams.

We now have a series of new models from the GEANT4 collaboration and there is an on-going request to produce single particle samples for each of the variations to re-compute this uncertainty: JIRA [↗](#). From test-beam studies of p,K+,pi+ it is thought that this uncertainty will be significantly reduced, particularly at high energy.

---

## Major updates:

-- BenjaminNachman and ChristopherYoung - 2018-04-04

---

This topic: Main > BottomUpUncerts

Topic revision: r9 - 2018-04-21 - BenjaminNachman



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback