

Table of Contents

7.7 MET Analysis.....	1
7.7.1 Introduction.....	1
7.7.2 Computing Environment.....	1
CMSSW environment.....	1
git and github.....	2
MET Recipes.....	2
Build.....	2
Workbook files.....	2
7.7.3 Sample AOD files.....	2
7.7.4 MET Objects in CMSSW.....	3
RECO/AOD.....	3
PAT.....	4
MiniAOD.....	5
7.7.5 MET Filters.....	5
7.7.6 MET Corrections.....	6
Type-I Correction.....	6
Type-II Correction.....	7
Type-0 Correction.....	8
xy-Shift Correction.....	10
How to apply the MET corrections in CMSSW.....	11
How to store corrected MET in PAT.....	12
MET Smearing Correction (for MC only).....	12
7.7.7 MET Uncertainties.....	12
7.7.8 MET Significance.....	12
7.7.9 Other Topics.....	13
7.7.10 MET Training Sessions	13
Links.....	14

7.7 MET Analysis

Introduction to missing transverse momentum (MET) in CMSSW

DocDB 12312: MET Diagram

7.7.1 Introduction

Missing transverse momentum (MET) is the imbalance in the transverse momentum of all *visible* particles, particles which interact with the electromagnetic or strong forces, in the final state of collisions. Because momentum is conserved in each direction, MET is the transverse momentum that must have been carried by something *invisible*. Neutrinos, for example, are invisible particles; therefore, MET is an estimate of transverse momentum of neutrinos. We use MET in measurements of W bosons, top quarks, and tau leptons as these particles can decay into neutrinos. Further, many models of physics beyond the Standard Model predict the existence of particles or something else which are invisible and can carry momentum; e.g., Dark Matter models, supersymmetric models, unparticle models, and models with large extra dimensions. For this reason, we use MET to test such models.

Accurate reconstruction of MET is demanding because it entails reconstruction of all visible particles in an event with precision. This requires a *hermetic* detector which can detect all particles which electromagnetically or strongly interact with matter. With its highly granular electromagnetic calorimeters, hermetic hadronic calorimeters, redundant muon systems, and all silicon trackers in a strong magnetic field, the CMS detector meets the requirement.

MET is informally referred to as different quantities and objects related to the momentum imbalance, as in many TWiki topics, including this one. MET can mean both a vector quantity and a scalar quantity of the imbalance. Moreover, the invisible particle itself is sometimes casually called MET. In addition, classes and instances representing MET in CMSSW are also called MET or MET Objects.

~~On the other hand, in publications, different words and mathematical symbols are typically defined as follows:~~
~~**Missing transverse momentum** \vec{E}_T : the negative of the vector sum of the transverse momenta of all reconstructed objects~~
~~**Missing transverse energy** E_T : the magnitude of missing transverse momentum~~ See PubGuidelines for how to define the **missing transverse energy**, **missing transverse momentum** and their symbols in publications.

7.7.2 Computing Environment

CMSSW environment

How to set up a computing environment for CMSSW is briefly described here. The detail is described at [WorkBookSetComputerNode](#).

After logging into LXPLUS or any other machine with CMSSW installed, move to a working directory.

Then, check out CMSSW release if not done yet:

```
cmsrel CMSSW_5_3_24
```

Move down two directories and set a CMSSW runtime environment:

```
cd CMSSW_5_3_24/src
cmsenv
```

git and github

In addition to the CMSSW runtime environment, a [git](#) environment needs to be properly set up in order to be able to check out extra packages from a [github](#) repository. The instruction can be found at <http://cms-sw.github.io/cmssw/faq.html>.

MET Recipes

If you are using CMSSW_5_3 older than CMSSW_5_3_24, you need to check out extra files in order to work out this workbook. The instruction to checkout the extra files are listed on

- [SWGGuideMETRecipe53X](#)

Build

You can build with the `scram` command:

```
scram b -j 9
```

More information about `scram` can be found at

- [WorkBookScramV1Intro](#)
- [SWGGuideScram](#)

Workbook files

This workbook uses files at the [github](#) repo:

- <https://github.com/TaiSakuma/WorkBookMet>

Clone this repo to your local directory with the following command.

```
git clone git@github.com:TaiSakuma/WorkBookMet
```

7.7.3 Sample AOD files

This topic will use two sample AOD files stored in an AFS locker at CERN:

```
41M /afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET/TTJets_AODSIM_532_numEvent100.root
32M /afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET/MET_Run2012C_AOD_532_numEvent100.root
```

The 1st file is 41Mbyte and contains 100 events from [/TTJets_MassiveBinDECAY_TuneZ2star_8TeV-madgraph-tauola/Summer12_DR53X-PU_S10_START53_V7A-v1](#) (DAS).

The 2nd file is 32Mbyte and contains 100 events from [/MET/Run2012C-24Aug2012-v1/AOD](#) (DAS).



These are the commands used to create the sample files:

```

cmsrel CMSSW_5_3_8
cd CMSSW_5_3_8/src
cmsenv
edmCopyPickMerge \
  inputFiles=/store/mc/Summer12_DR53X/TTJets_MassiveBinDECAY_TuneZ2star_8TeV-madgraph-tauola/AODSIM
  outputFile=TTJets_AODSIM_532.root \
  maxEvents=100
mkdir -p /afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET
cp -a ./TTJets_AODSIM_532_numEvent100.root /afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET

curl -O http://cms-service-dqm.web.cern.ch/cms-service-dqm/CAF/certification/Collisions12/8TeV/Reco
cmsRun ./WorkBookMet/copyPickMerge_cfg.py \
  inputFiles=/store/data/Run2012C/MET/AOD/24Aug2012-v1/00000/FE9CCF41-6AFB-E111-82AF-0030486792
  certFile=Cert_198022-198523_8TeV_24Aug2012ReReco_Collisions12_JSON.txt \
  outputFile=MET_Run2012C_AOD_532.root \
  maxEvents=100
cp -a ./MET_Run2012C_AOD_532_numEvent100.root /afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET

```

7.7.4 MET Objects in CMSSW

MET Objects in CMSSW are:

- the class `reco::MET`, its subclasses, and their instances in RECO and AOD
- the class `pat::MET` and its instances in PAT.

RECO, AOD, and PAT are *data tiers*, which are explained in [WorkBookAnalysisOverviewIntroduction](#). The class hierarchy is drawn in the diagram at [Doxygen reco::MET](#).

RECO/AOD

We will show an example how to access `reco::PFMET`, a subclass of `reco::MET`, in RECO/AOD.

RECO/AOD files contain several MET objects, listed on [SWGGuideDataFormatRecoMET](#). In the example, we will access to `pfMet`, particle-flow MET.

We will use `FWLite.Python`, introduced at [WorkBookFWLitePython](#). The example Python script that we use is [printMet_AOD.py](#).

Execute by typing:

```
./WorkBookMet/printMet_AOD.py --inputPath=/afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET/TTJets_AODSIM
```

The script will print event contents as follows:

run	lumi	event	met.pt	met.px	met.py	met.phi
1	34734	10417901	36.837	18.857	-31.645	-59.21
1	34734	10417902	19.955	-16.211	-11.636	-144.33
1	34734	10417903	38.794	-9.340	-37.652	-103.93
1	34734	10417919	96.502	-24.198	-93.419	-104.52
1	34734	10417920	22.568	-14.566	-17.238	-130.20
1	34734	10417921	78.944	21.847	-75.861	-73.93

- `run`, `lumi`, `event` are the run number, the luminosity section, and the event id.
- `met.pt` is the magnitude of MET. MET is, in principle, a vector on the px-py plane. However, we often casually call its magnitude MET as well.
- `met.px`, `met.py` are the x and y components of MET respectively.
- `met.phi` is the azimuth of MET.

An example of how to fill histograms in PyROOT [↗](#) can be found at [WorkBookFWLitePython](#).

We will explain how we accessed the MET object in `printMet_AOD.py` [↗](#).

First, the handle of the PFMET collection is created as:

```
handleMETs = Handle("std::vector<reco::PFMET>")
```

The MET collection is retrieved by calling `getByLabel()` of `event`. The collection is specified by the first argument, the same strings that you would give to `cms.InputTag()` in the `cmsRun` configuration file. The second argument is the handle created above.

```
event.getByLabel("pfMet", '', 'RECO'), handleMETs)
```

The MET collection can be obtained by calling `handleMETs.product()`. However, since exactly one MET can be reconstructed per MET definition per event, the collection always contains exactly one object. We obtain the MET object and assigned to `met` at:

```
met = handleMETs.product().front()
```

In the script, we accessed several variables of MET and printed them on the screen:

```
print '%10.3f' % met.px(),
print '%10.3f' % met.py(),
print '%10.3f' % met.pt(),
print '%10.2f' % (met.phi()/math.pi*180.0),
```

All methods of `reco::PFMET` can be found at [Doxygen reco::PFMET](#) [↗](#).

PAT

We will briefly demonstrate how to create MET objects in PAT, which correspond to `pfMet`, and access them.

We will use the the configuration file `pat_rawMET_cfg.py` [↗](#) to create a pat tuple.

Execute *cmsRun* with the configuration file.

```
cmsRun ./WorkBookMet/pat_rawMET_cfg.py inputFile=file:/afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET/
```

This will create a pat tuple file which only contains one collection, `patMETs`. The following command shows the contents:

```
edmDumpEventContent pat_rawMET.root
```

The output will be:

Type	Module	Label	Process
vector<pat::MET>	"patMETs"	" "	"PAT"

The python script `printMet_PAT.py` shows how to access to the MET object in PAT which we just created.

Run the script:

```
./WorkBookMet/printMet_PAT.py --inputPath=./pat_rawMET.root
```

The output will be identical to what we saw when we accessed to `pfMet` in RECO/AOD above as it should be:

run	lumi	event	module	met.pt	met.px	met.py	met.phi
1	34734	10417901	patMETs	36.837	18.857	-31.645	-59.21
1	34734	10417902	patMETs	19.955	-16.211	-11.636	-144.33
1	34734	10417903	patMETs	38.794	-9.340	-37.652	-103.93
1	34734	10417919	patMETs	96.502	-24.198	-93.419	-104.52
1	34734	10417920	patMETs	22.568	-14.566	-17.238	-130.20
1	34734	10417921	patMETs	78.944	21.847	-75.861	-73.93

MiniAOD

See

- [WorkBookMiniAOD](#)

7.7.5 MET Filters

Large MET is caused not only by interesting physics processes in collisions such as production of invisible particles. In fact, large MET has more often uninteresting causes such as detector noise, *cosmic rays*, and *beam-halo* particles. MET with uninteresting causes is called *false* MET, *anomalous* MET, or *fake* MET. For an accurate reconstruction of MET, it is, therefore, not sufficient to reconstruct all visible particles produced in collisions.

We developed several algorithms to identify false MET. These algorithms, for example, use timing, pulse shape, and topology of signal. After the identified false MET is removed, the agreement of the MET spectrum with MC, in which causes of false MET are not explicitly simulated, will typically improve significantly.

The MET group recommends a set of MET filters to be used for physics analyses. The recommendation is documented on [MissingETOptionalFilters](#).

Here, we will describe how to apply the recommended MET filters.

We will use the python configuration file `met_filters_cfg.py`.

To run:

```
cmsRun ./WorkBookMet/met_filters_cfg.py inputFile=file:/afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET
```

As in the above configuration file, you need to change only few lines of your configuration file in order to apply the recommended MET filters. First, you need to load the python configuration fragment `metFilters_cff`.

```
process.load("RecoMET.METFilters.metFilters_cff")
```

Then, add the sequence

process.metFilters

to your path.

The example configuration file uses the global tag

FT_R_53_V21::All

You might need to find the correct tag for you configuration at SWGuideFrontierConditions.

to write:

- how to store the results of filters

7.7.6 MET Corrections

MET objects accessed above are called *raw MET*. Raw MET is the negative of the vector sum of \vec{p}_T of all reconstructed particles:

$$\vec{E}_T^{\text{raw}} = -\sum_{i \in \text{all}} \vec{p}_{Ti}$$

The raw MET is systematically different from *true MET*, i.e., the transverse momentum carried by invisible particles, for many reasons including the non-compensating nature of the calorimeters and detector misalignment. To make MET a better estimate of true MET, *MET corrections* can be applied.

Type-I Correction

The *Type-I correction* is the most popular MET correction in CMS. This correction is a propagation of the jet energy corrections (JEC) to MET. The Type-I correction replaces the vector sum of transverse momenta of particles which can be clustered as jets with the vector sum of the transverse momenta of the jets to which JEC is applied.

The particles can be classified into two disjoint sets: either clustered as jets or unclustered:

$$\vec{E}_T^{\text{raw}} = -\sum_{i \in \text{jets}} \vec{p}_{Ti} - \sum_{i \in \text{uncl}} \vec{p}_{Ti}$$

The first vector sum is the same as the vector sum of \vec{p}_T of all jets

$$\sum_{i \in \text{jets}} \vec{p}_{Ti} = \sum_{\text{jet}} \vec{p}_{T \text{jet}}^{\text{raw}}$$

The superscript "raw" here indicates that the JEC is not applied to the jets.

The Type-I correction replaces the raw jet \vec{p}_T with the corrected jet \vec{p}_T . The Type-I correction can be written as the difference between two vector sums:

$$\vec{C}_T^{\text{Type-I}} = \sum_{\text{jet}} \vec{p}_{T \text{jet}} - \sum_{\text{jet}} \vec{p}_{T \text{jet}}^{\text{raw}} - \sum_{\text{jet}} \vec{p}_{T \text{jet}}^{\text{JEC}}$$

or equivalently

$$\vec{C}_{\text{Type-I}} = \sum_{i \in \text{jets}} \vec{p}_{\text{T}i} - \sum_{\text{jet}} \vec{p}_{\text{T jet}}^{\text{JEC}}.$$

This is a vector term which can be added to the raw MET to produce a corrected MET. The first sum cancels the first sum in the raw MET and the second sum places the corrected piece.

$$\vec{E}_{\text{Type-I}} = \vec{E}_{\text{raw}} + \vec{C}_{\text{Type-I}}.$$

As a result, the Type-I corrected MET can be written as

$$\vec{E}_{\text{Type-I}} = \sum_{\text{jet}} \vec{p}_{\text{T jet}}^{\text{JEC}} - \sum_{i \in \text{uncl.}} \vec{p}_{\text{T}i}.$$

Note: The detail of JEC can be found at [IntroToJEC](#), [CMS-PAS-JME-07-002](#), and [AN-2007/055](#)

Note: The actual Type-I correction is more complicated due to a threshold and an offset. See [METType1Type2Formulae](#)

Type-II Correction

The *Type-II correction* was originally developed for CaloMET and is not recommended to use for pfMET.

The Type-II correction corrects the \vec{p}_{T} of unclustered particles by uniformly scaling it by a constant scale factor:

$$\vec{C}_{\text{Type-II}} = (1 - C^{\text{uncl.}}) \sum_{i \in \text{uncl.}} \vec{p}_{\text{T}i}.$$

If the Type-II correction is added to Type-I corrected MET

$$\vec{E}_{\text{Type-I-II}} = \vec{E}_{\text{Type-I}} + \vec{C}_{\text{Type-II}},$$

the Type-I-II corrected MET will be produced

$$\vec{E}_{\text{Type-I-II}} = \sum_{\text{jet}} \vec{p}_{\text{T jet}}^{\text{JEC}} - C^{\text{uncl.}} \sum_{i \in \text{uncl.}} \vec{p}_{\text{T}i}.$$

Note: The Type-II correction is not recommended to use for pfMET.

Note: The actual Type-II correction scales the offset of $\text{jet } \vec{p}_T$ and $\text{jet } \vec{p}_T$ below a threshold as well. See METType1Type2Formulae

Type-0 Correction

The *Type-0 correction* is a mitigation for the degradation of the MET reconstruction due to the pile-up interactions. This correction is developed for pfMET and cannot be sensibly defined for CaloMET.

Pile-up interactions have little true MET because they produce few invisible particles, e.g., neutrinos from Kaon decays. Therefore, if we were able to measure all visible particles precisely and accurately, pile-up interactions would not much degrade the MET reconstruction. However, in practice, because our measurement of visible particles is not perfect, the MET reconstruction degrades as the number of the pile-up interactions increases.

The Type-0 correction is an application of the charged hadron subtraction (CHS) to MET. CHS removes charged hadrons originating from the vertices for pile-up interactions. In addition to CHS, the Type-0 correction removes an estimate of neutral pile-up contributions. This removal of the estimate is the integral part of the Type-0 correction. Doing only CHS would make MET worse as it becomes clear below.

While, in the Type-I and Type-II corrections, the particles are classified as whether they are clustered as jets or unclustered, in the Type-0 correction, the particles are classified as whether they are produced in the hard scattering of interest (HS) or in pile-up interactions (PU):

$$\not\mathrel{\vec{E}}_T^{\text{raw}} = - \sum_{i \in \text{HS}} \vec{p}_{T,i} - \sum_{i \in \text{PU}} \vec{p}_{T,i}.$$

This equality is theoretically correct. However, we are not able to classify all reconstructed particles in this way because we do not know which particles are produced in the hard scattering. Nevertheless, we would like somehow to remove the second sum and keep only the first sum since the negative of the first sum is the MET in the hard scattering.

We further classify the pile-up particles into neutral particles (neuPU) and charged particles (chPU):

$$\not\mathrel{\vec{E}}_T^{\text{raw}} = - \sum_{i \in \text{HS}} \vec{p}_{T,i} - \sum_{i \in \text{neuPU}} \vec{p}_{T,i} - \sum_{i \in \text{chPU}} \vec{p}_{T,i}.$$

The last sum is taken over the charged particles produced in the pile-up interactions. We can identify such particles from the vertices and can measure the last sum. However, we cannot separate the first and second sums. CHS removes the last sum. It is not a good idea to remove only the last sum and keep the first and second sums. This is more likely to make the reconstructed MET more different from its true value than to make it closer to.

The second sum is for the neutral pile-up particles and the last sum is for the charged pile-up particles. These particles are produced in the same interactions, with little true MET. Therefore, the second sum and the last sum are in nearly opposite directions. At the true level, their magnitudes are close to each other and these two sums nearly cancel each other. The Type-0 correction assumes that the cancellation at the true level is exact:

$$\sum_{i \in \text{neuPU}} \vec{p}_{T,i}^{\text{true}} + \sum_{i \in \text{chPU}} \vec{p}_{T,i}^{\text{true}} = 0.$$

The Type-0 correction also assumes that we can measure charged pile-up particles perfectly:

$$\sum_{i \in \text{chPU}} \vec{p}_{T,i}^{\text{true}} = \sum_{i \in \text{chPU}} \vec{p}_{T,i}$$

We can justify the assumption as follows. CMS has an all silicon tracker in a strong magnetic field. Pile-up particles have relatively low \vec{p}_T . Low- \vec{p}_T charged particles make tight turns. And we can measure the curvatures of their tracks precisely with the CMS tracker.

Furthermore, the Type-0 correction assumes that we can measure the directions of neutral pile-up particles perfectly but measure their energies systematically off by the same factor:

$$\sum_{i \in \text{neuPU}} \vec{p}_{T,i} = R^0 \sum_{i \in \text{neuPU}} \vec{p}_{T,i}^{\text{true}}$$

The assumptions were made as follows. We can measure the directions precisely from the positions of the calorimeter cells in which we observe the energy deposits. Since we calibrate the calorimeters to measure high- \vec{p}_T particles, we measure the energies of low- \vec{p}_T particles systematically off.

With these assumptions, the second sum in raw MET, the contributions from the neutral pile-up particles, can be estimated as:

$$\sum_{i \in \text{neuPU}} \vec{p}_{T,i} = -R^0 \sum_{i \in \text{chPU}} \vec{p}_{T,i}$$

The Type-0 correction is this estimate plus the last sum in raw MET:

$$\vec{C}_T^{\text{Type-0}} = (1 - R^0) \sum_{i \in \text{chPU}} \vec{p}_{T,i}$$

This correction can be added to raw MET to remove the identified charged pile-up contributions and an estimate of neutral pile-up contributions:

$$\{\text{not}\mathrel{\vec{E}}\}_T^{\text{Type-0}} = \{\text{not}\mathrel{\vec{E}}\}_T^{\text{raw}} + \vec{C}_T^{\text{Type-0}}$$

You can add the Type-I correction to the Type-0 corrected MET to propagate JEC to MET:

$$\{\text{not}\mathrel{\vec{E}}\}_T^{\text{Type-0-I}} = \{\text{not}\mathrel{\vec{E}}\}_T^{\text{Type-0}} + \vec{C}_T^{\text{Type-I}}$$

pfMET with the Type-0 and Type-I corrections is one of the recommended METs by the MET group to use in physics analyses.

Note: Type-0RT and Type-0PC. Currently, we have two implementations of the Type-0 corrections. Type-0RT is the original implementation of Type-0. It uses `reco::Track` for the charged pile-up particles. Type-0PC, developed later, uses particle-flow charged hadrons instead. While, in Type-0RT, R^0 is a constant, in Type-0PC, R^0 is a function of the magnitude of $\sum_{i \in \text{chPU}} \vec{p}_{T,i}$.

Note: Precisely speaking, if the Type-0 and Type-I corrections are applied at the same time, a certain fraction of particle \vec{p}_T is overcorrected. The Type-0 correction is applied on a set of particles produced in the pile-up interactions. The Type-I correction is applied on a set of particles clustered as jets. These two sets are not disjoint. Therefore, the particles in both sets, the particles produced in the pile-up interactions and clustered as jets (with \vec{p}_T above the threshold of the Type-I correction), are incorrectly corrected by the both corrections. In the current implementation, this overcorrection, which is presumably not significantly large, is ignored.

Note: If the Type-0 and Type-II corrections are applied at the same time, the Type-0 correction needs to be scaled by the same scaler factor used in the Type-II correction. See METType012Formulae. However, the Type-II correction is not recommended to use for pfMET and the Type-0 correction is not defined for CaloMET. So, practically, these two corrections are never used at the same time.

xy-Shift Correction

The *xy-Shift correction* reduces the MET φ modulation. This correction is also a mitigation for the pile-up effects.

The distribution of true MET is independent of φ because of the rotational symmetry of the collisions around the beam axis. However, we observe that the reconstructed MET does depend on φ . The MET φ distribution has roughly a sinusoidal curve with the period of 2π . The possible causes of the modulation include anisotropic detector responses, inactive calorimeter cells, the detector misalignment, the displacement of the beam spot. The amplitude of the modulation increases roughly linearly with the number of the pile-up interactions.

We can reduce the amplitude of the φ modulation by shifting the origin of the coordinate in the transverse momentum plane

$$\vec{p}_T \rightarrow \vec{p}_T - \vec{c}$$

where \vec{c} is the shift. With this shift, MET is

$$\begin{aligned} \sum_i \vec{E}_T^{\text{xy}} &= \sum_i (\vec{p}_T^i - \vec{c}) \\ \vec{p}_T + \sum_i \vec{c} &= \sum_i \vec{E}_T^{\text{raw}} + n\vec{c}, \end{aligned}$$

where n is the number of the particles. The total shift $n\vec{c}$ does not depend on the value of MET. The *xy-Shift correction* is

$$\vec{C}_T^{\text{xy}} = n\vec{c}$$

Therefore, the *xy-shifted MET* is

$$\sum_i \vec{E}_T^{\text{xy}} = \sum_i \vec{E}_T^{\text{raw}} + \vec{C}_T^{\text{xy}}$$

~~As the *xy-Shift correction* does not depend on the value of MET, it can be, for example, added to the Type-0 and Type-I corrected MET, (This is probably incorrect. The above formula indicates that the *xy-Shift correction* depends on the number of the particles used in MET. For example, raw MET and Type-0 corrected MET don't use the same number of the particles, therefore the same *xy-Shift correction* cannot be added to raw MET and Type-0 corrected MET.)~~

$$\left[\left\{ \text{Type-0-I-xy} \right\} \right] = \left[\left\{ \text{Type-0-I} \right\} \right] + \left[\left\{ \text{xy} \right\} \right]$$

pfMET with the Type-0, Type-I, and xy-Shift corrections is also one of the recommended METs by the MET group to use in physics analyses.

Note: Currently, the xy-Shift correction is implemented as $\left[\left\{ \text{C} \right\} \right]_{\text{Type-0-I-xy}} = \left[\left\{ \text{c} \right\} \right]_{\text{A}} + n_{\text{vtx}} \left[\left\{ \text{c} \right\} \right]_{\text{B}}$, where $\left[\left\{ \text{c} \right\} \right]_{\text{A}}$ and $\left[\left\{ \text{c} \right\} \right]_{\text{B}}$ are constant vectors and n_{vtx} is the number of the vertices.

How to apply the MET corrections in CMSSW

We will apply several combinations of the MET corrections to pfMET and CaloMET. We will use an example configuration file `corrMet_cfg.py`.

This example file is configured to use for MC. To use for data, you need to change two places. The changes can be made by uncommenting the following two lines:

```
process.corrPfMetType1.jetCorrLabel = cms.string("ak5PFL1FastL2L3Residual")
```

and

```
process.corrPfMetShiftXY.parameter = process.pfMETSysShiftCorrParameters_2012runABCDvsNvtx_data
```

To run,

```
cmsRun ./WorkbookMet/corrMet_cfg.py inputFile=file:/afs/cern.ch/cms/Tutorials/TWIKI_DATA/MET/TTJ
```

This will produce a file `corrMet.root`, which contains various MET collections, each with different combinations of the MET corrections as summarized in the table.

module name	descriptions
pfMetT0rt	pfMet + Type-0RT
pfMetT0rtT1	pfMet + Type-0RT + Type-I
pfMetT0pc	pfMet + Type-0PC
pfMetT0pcT1	pfMet + Type-0PC + Type-I
pfMetT0rtTxy	pfMet + Type-0RT + xy-Shift
pfMetT0rtT1Txy	pfMet + Type-0RT + Type-I + xy-Shift
pfMetT0pcTxy	pfMet + Type-0PC + xy-Shift
pfMetT0pcT1Txy	pfMet + Type-0PC + Type-I + xy-Shift
pfMetT1	pfMet + Type-I
pfMetT1Txy	pfMet + Type-I + xy-Shift

The python script `printMet_corrMet.py` shows an exmple how to access to the corrected METs in `corrMet.root`.

```
./WorkbookMet/printMet_corrMet.py --inputPath=./corrMet.root
```

This will simply print the contents as follows.

WorkBookMetAnalysis < CMSPublic < TWiki

run	lumi	event	module	met.pt	met.px	met.py	met.phi
1	34734	10417901	pfMet	36.837	18.857	-31.645	-59.21
1	34734	10417901	pfMetT0rt	32.819	9.912	-31.286	-72.42
1	34734	10417901	pfMetT0rtT1	37.309	-0.433	-37.307	-90.67
1	34734	10417901	pfMetT0pc	32.296	9.728	-30.796	-72.47
1	34734	10417901	pfMetT0pcT1	36.822	-0.618	-36.816	-90.96
1	34734	10417901	pfMetT1	38.615	8.511	-37.665	-77.27
1	34734	10417901	pfMetT0rtTxy	29.873	10.417	-27.998	-69.59
1	34734	10417901	pfMetT0rtT1Txy	34.018	0.072	-34.018	-89.88
1	34734	10417901	pfMetT0pcTxy	29.349	10.233	-27.508	-69.59
1	34734	10417901	pfMetT0pcT1Txy	33.528	-0.113	-33.528	-90.19
1	34734	10417901	pfMetT1Txy	35.539	9.016	-34.377	-75.30

Note: Not all combinations of the MET corrections are sensible to use in physics analyses.

Note: The globaltag "START53_V15A::All" was used in this example.

How to store corrected MET in PAT

The above example stores the corrected METs in reco::MET or its subclass. If you would like to store the corrected MET in pat::MET, you can do this by using PATMETProducer.

For example, to store pfMetT0pcT1Txy in pat::MET, you can add the following lines in your python configuration.

```
from PhysicsTools.PatAlgos.producersLayer1.metProducer_cfi import patMETs
patPfMetT0pcT1Txy = patMETs.clone(
    metSource = cms.InputTag('pfMetT0pcT1Txy'),
    addMuonCorrections = cms.bool(False),
    addGenMET = cms.bool(False)
)
```

This will produce patPfMetT0pcT1Txy, which has the same content as pfMetT0pcT1Txy but stored in pat::MET.

MET Smearing Correction (for MC only)

Jets in MC can be smeared so as to achieve better agreement with data. Such smeared jets are available as by-product of MET Systematics Tools in PAT.

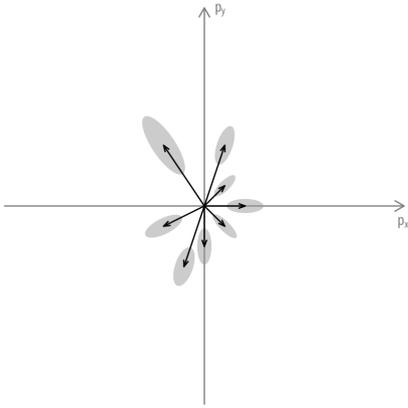
Visit SWGuidePATTools#MET_Systematics_Tools and "doSmearJets" (access limited to CMS members)

7.7.7 MET Uncertainties

coming soon.. for the moment, visit MissingETUncertaintyPrescription or SWGuidePATTools#MET_Systematics_Tools for the PAT tool (access limited to CMS members)

7.7.8 MET Significance

The SWGuide for the MET Significance is



- [SWGGuideMETSignificance](#)

How the MET Significance is defined from the reconstructed MET and the resolutions of measured particles is explained on these lecture slides:

- "Introduction to MET Significance" (pdf)

7.7.9 Other Topics

These are the topics that are important but not currently covered by this topic.

- MVA MET
- NoPU MET
- MET Performance Measurement
- MET Software Development

7.7.10 MET Training Sessions NEW

- 19-23, Jan 2015 **CMS Data Analysis School (CMSDAS)** at Bari, (Indico)[☞](#)
 - ◆ Short Exercise MET, (Indico)[☞](#), (TWiki)
- 26 Mar 2014 **Hands-on Tutorial Sessions (HATS) on MET** at Fermilab LPC (Indico)[☞](#), (TWiki)
- 8 - 11 Jan 2014 **CMS Data Analysis School (CMSDAS)** at Fermilab LPC, (Indico)[☞](#)
 - ◆ Short Exercise MET, (Indico)[☞](#), (TWiki)
- 13 - 18 Jan 2014 **CMS Data Analysis School (CMSDAS)** at CERN (Indico)[☞](#)
 - ◆ Short Exercise Jets and MET, (Twiki)
- 25-29, Jan 2011 **CMS Data Analysis School (CMSDAS)** at Fermilab LPC, (Indico)[☞](#)
 - ◆ Short Exercise MET, (Twiki)
 - ◆ highlight: exercise on MET - tails

Links

- [JetMET](#)
- [MissingET](#)
- [JetMETAlgorithmsReconstruction](#)

Responsible: TaiSakuma

This topic: CMSPublic > WorkBookMetAnalysis

Topic revision: r200 - 2019-06-28 - JakeRosenzweig



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback