update on NanoAODplus jets: JEC

Armando Bermudez Martinez
09.12.2020

Comments on SL6 singularity for 2011/2012 Open Data

- Detailed steps shared by Achim:

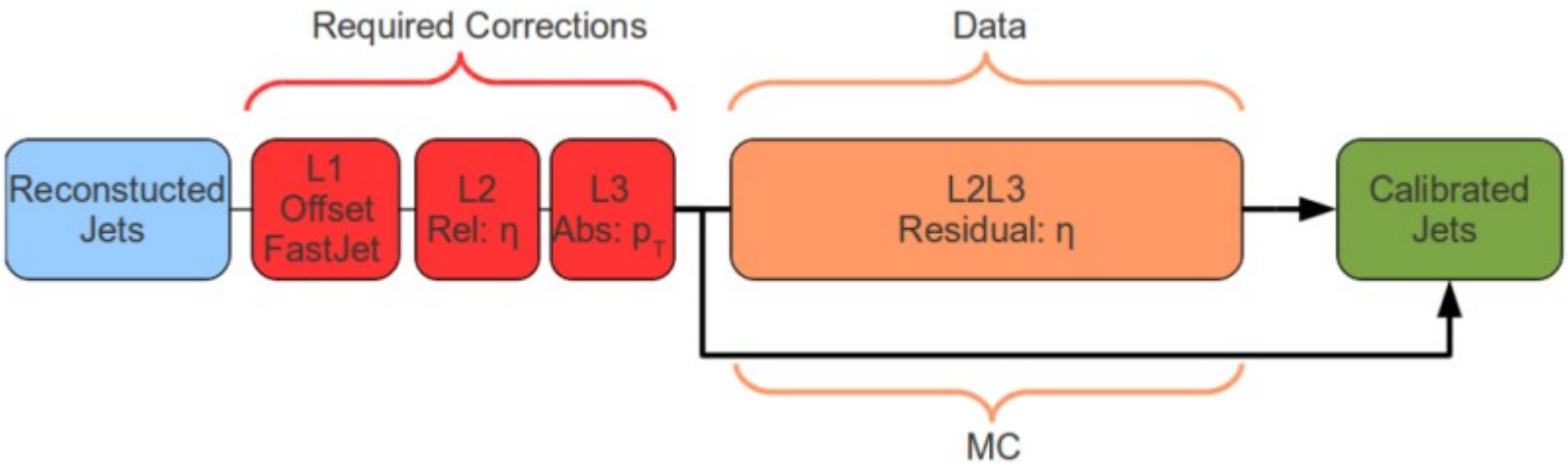
/afs/desy.de/user/g/geiser/public/CMS/NanoAODplusel7/Nanoreadme.txt

- Tried successfully in cms12

```
[bermudea@naf-cms12]~/nanoAODplus/CMSSW_5_3_32/src/NanoAOD% module use -a /afs/desy.de/group/cms/modulefiles
[bermudea@naf-cms12]~/nanoAODplus/CMSSW_5_3_32/src/NanoAOD% module load cmssw
[bermudea@naf-cms12]~/nanoAODplus/CMSSW_5_3_32/src/NanoAOD% cmssw-cc6
Singularity> source /cvmfs/cms.cern.ch/cmsset_default.sh
Singularity> voms-proxy-init -voms cms
Enter GRID pass phrase:
Your identity: /C=DE/O=GermanGrid/OU=DESY/CN=Armando Bermudez-Martinez
Creating temporary proxy ................................. Done
Contacting  voms2.cern.ch:15002 [/DC=ch/DC=cern/OU=computers/CN=voms2.cern.ch] "cms" Done
Creating proxy ............................. Done

Your proxy is valid until Tue Dec  8 23:24:29 2020
Singularity> scram b -j8
Reading cached build data
>> Local Products Rules ..... started
>> Local Products Rules ..... done
```

JEC implementation



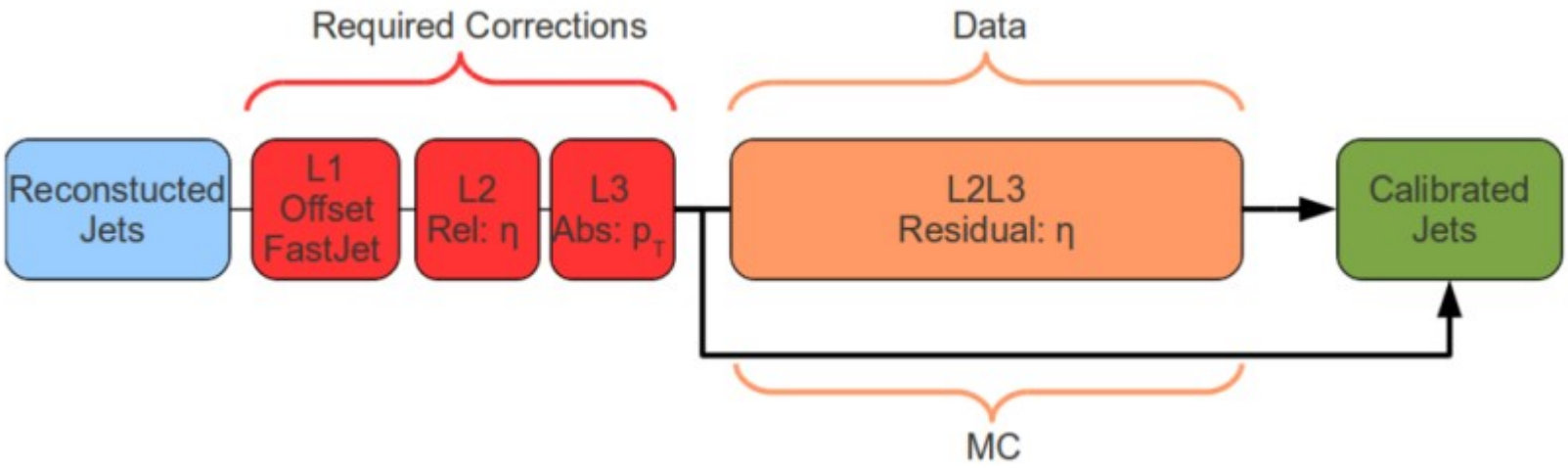
- Details on JEC in CMSSW

https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookJetEnergyCorrections

- Useful example

https://github.com/cms-opendata-validation/2011-doubleelectron-doublemu-mueg-ttbar

JEC implementation



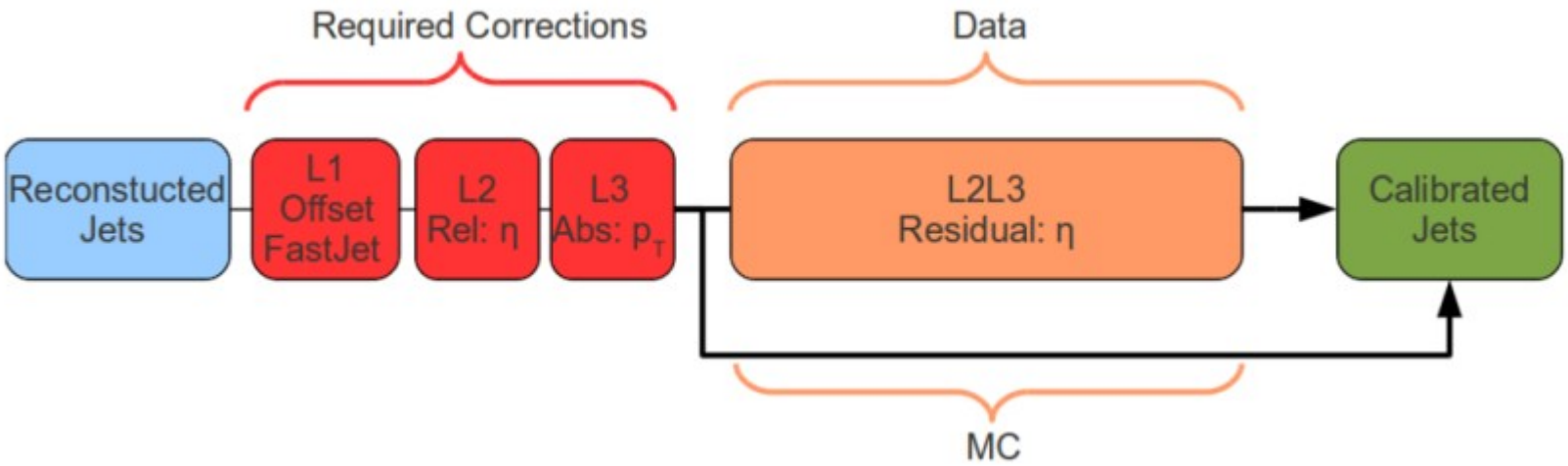
- In the following I'll use all of them:

```
  // jet correction label
  mJetCorr = "ak5PFL1FastL2L3Residual";
//  mJetCorr = "ak5CaloL2L3";
```

Many options (algorithm, level)

```
# L1 (JPT Offset) CORRECTOR
ak4L1JPTOffsetCorrector
...
# L1 (Fastjet PU Subtraction) CORRECTOR
ak4CaloL1FastjetCorrector
...
# L2 (relative eta-conformity) CORRECTORS
ak4CaloL2RelativeCorrector
ak4PFL2RelativeCorrector
...
# L3 (absolute) CORRECTORS
ak4CaloL3AbsoluteCorrector
ak4PFL3AbsoluteCorrector
ak4PFCHSL3AbsoluteCorrector
...
```

JEC implementation



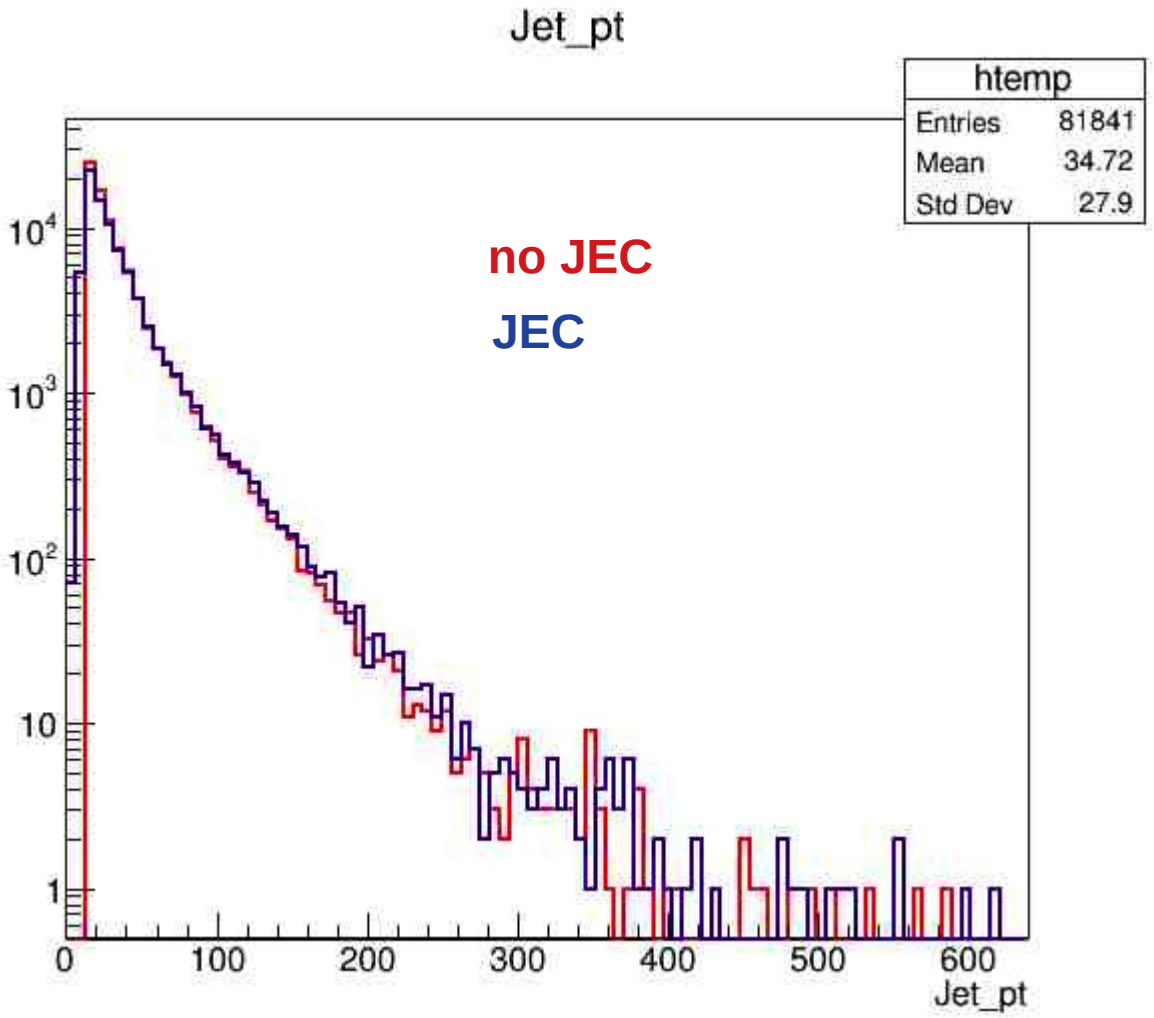

```
const JetCorrector* corrector = JetCorrector::getJetCorrector(mJetCorr, iSetup);
for (pat::JetCollection::const_iterator it = jets->begin(); it != jets->end(); ++it) {
  double jec = corrector->correction(*it, iEvent, iSetup);

  // copy original (uncorrected) jet;
  reco::PFJet corjet = *it;
  // apply JEC
  corjet.scaleEnergy(jec);

  cout<<jec<<" "<<it->pt()<<" "<<corjet.pt()<<endl;
}
```

```
0.703095 24.3894 17.1481
0.630599 22.2157 14.0092
0.607595 19.3032 11.7286
0.689357 18.118 12.4898
0.602827 17.206 10.3722
```

Jet pt: JEC vs noJEC



data set:

'root://eospublic.cern.ch//eos/opendata/cms/Run2011A/SingleMu/AOD/12Oct2013-v1/10000/00209631-9C37-E311-BACA-002590494FDE.root'

Next steps

- Extending to FatJets

- Modify the code to deal with run2 data

https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookJetEnergyCorrections

Applying the Jet Energy Corrections

UPDATED: Since CMSSW 7.6.X, the consumes interface is now mandatory. The jet corrector framework was not modular enough to enforce this, so a new one was developed. **The old interface is deprecated (using `JetCorrector::getJetCorrector` and friends) and won't work anymore on CMSSW 7.6.X and above**.