

# Table of Contents

<b>DQMonitoringFTAGJetSelection.....</b>	<b>1</b>
<b>Introduction.....</b>	<b>2</b>
<b>Event selection.....</b>	<b>3</b>
Event cleaning (Tile or LAr error flags).....	3
Primary Vertex.....	3
Number of tracks in Primary Vertex.....	4
Trigger pre-selection.....	4
<b>Jet Selection.....</b>	<b>5</b>
Quality cuts.....	5
Kinematic cuts.....	6
JVT cut.....	6
Taggability: definitions.....	6
Taggability (Good).....	7
Taggability (Suspect).....	7
Taggability (Bad).....	7

# DQMonitoringFTAGJetSelection

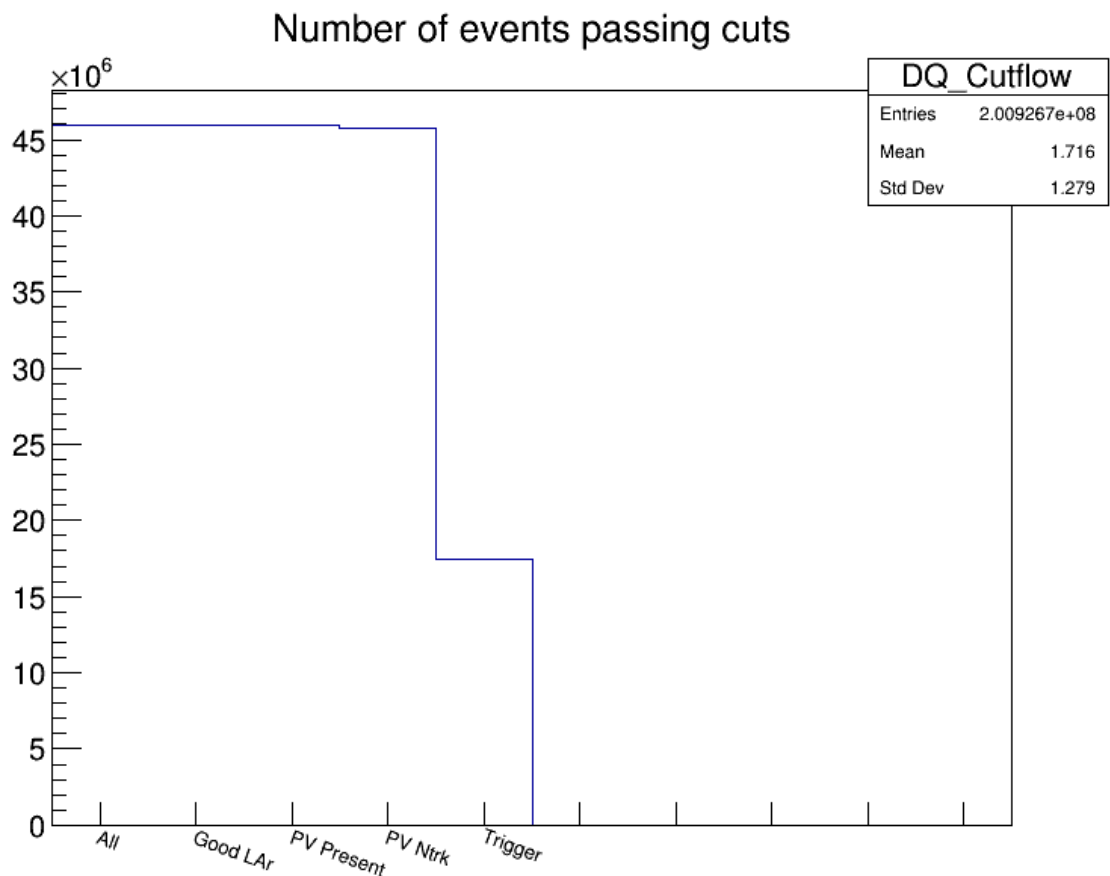
# Introduction

This page contains the different steps of selection for the jets used in Flavour Tagging Data Quality Monitoring.

# Event selection

Events are required to pass several requirements. The cut flow can be seen in the figure below:

- Event selection cut flow on 2017 data run 337705:



## Event cleaning (Tile or LAr error flags)

The event is required to not have error Flags related to Tile and LAr calorimeters:

```
if ( thisEventInfo->errorState(xAOD::EventInfo::EventFlagSubDet::Tile) == xAOD::EventInfo::ErrorState::Error ||
    thisEventInfo->errorState(xAOD::EventInfo::EventFlagSubDet::LAr) == xAOD::EventInfo::ErrorState::Error ||
    thisEventInfo->isEventFlagBitSet(xAOD::EventInfo::Core, 18) )
```

## Primary Vertex

The event is required to have at least 1 primary vertex:

```
const xAOD::VertexContainer* vxContainer(0);
bool foundPrimaryVtx = evtStore()->retrieve(vxContainer, m_primaryVertexName);
if (!foundPrimaryVtx)
```

with:

```
declareProperty("PrimaryVertexContainer", m_primaryVertexName = "PrimaryVertices");
```

The vertex container size is required to be higher than 1:

```
if (vxContainer->size() < 2)
```

## Number of tracks in Primary Vertex

The type of the vertex is required to be Primary:

```
bool foundVxTypePriVtx = false;
m_priVtx = nullptr;
for (xAOD::VertexContainer::const_iterator vtx = vxContainer->begin(); vtx != vxContainer->end();
     if ((*vtx)->vertexType() == xAOD::VxType::PriVtx && (*vtx)->numberDoF() > 0 ) {
     m_priVtx = *vtx;
     foundVxTypePriVtx = true;
     break;
   }
}
if ( ! foundVxTypePriVtx )
```

The number of tracks associated to the Primary Vertex is required to be higher than 4:

```
if (m_priVtx->nTrackParticles()>0) {
  npv_trk = m_priVtx->nTrackParticles();
}
// check first PV > 4 tracks, if not, fail
if ( npv_trk < m_pri_vtx_trk_min_cut )
```

with:

```
declareProperty("PriVtxTrkMinCut", m_pri_vtx_trk_min_cut = 4 );
```

## Trigger pre-selection

The event is required to pass an OR of single electron or single muon triggers:

```
if (m_use_trigDectool && m_trigDecTool != 0) { // only require trigger if m_use_trigDectool is true
  if (!m_trigDecTool->isPassed(m_ElectronTrigger_201X) && !m_trigDecTool->isPassed(m_MuonTrigger_201X))
    return StatusCode::SUCCESS;
}
}
```

with:

```
declareProperty( "ElectronTrigger_201X", m_ElectronTrigger_201X = "HLT_e[2-9][0-9].*" ); // e20-
declareProperty( "MuonTrigger_201X", m_MuonTrigger_201X = "HLT_mu.*" ); // all mu triggers (inclu
```

# Jet Selection

The events passing the basic event selection are used to fill histograms with jets.

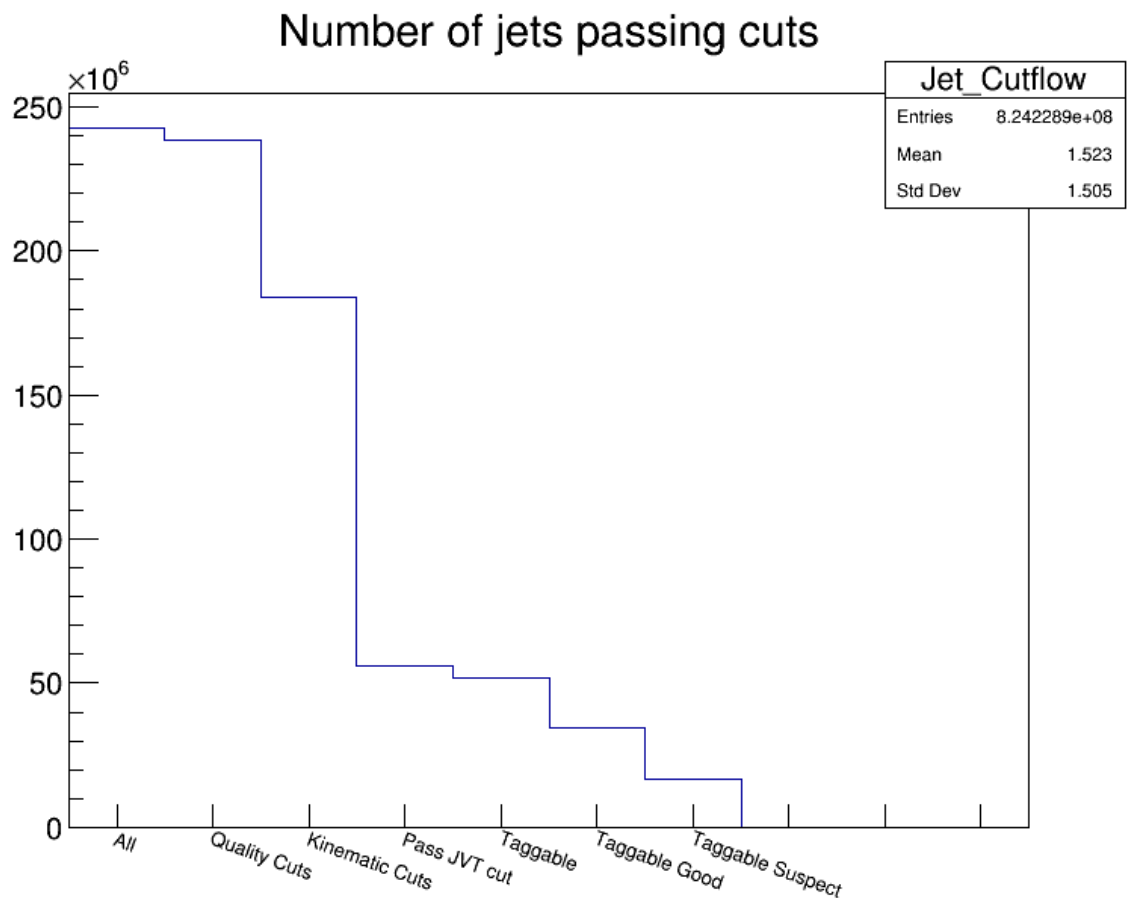
The nominal jet Collection is AntiKt4EMTopoJets.

```
declareProperty("JetContainer", m_jetName = "AntiKt4EMTopoJets");
```

In case of Heavy Ion collisions, the jet collection can be switched to "AntiKt4HIJets" in the jobOption.

The jets are required to pass several quality cuts, then are classified according to "Taggability". The cut flow can be seen in the figure below:

- Jet selection cut flow on 2017 data run 337705:



## Quality cuts

Each jet is required to pass a list of basic quality cuts:

```
float jetQuality      = jet->getAttribute<float>(xAOD::JetAttribute::LARQuality);
float jetTime         = jet->getAttribute<float>(xAOD::JetAttribute::Timing);
float hecq            = jet->getAttribute<float>(xAOD::JetAttribute::HECQuality);
float negE            = jet->getAttribute<float>(xAOD::JetAttribute::NegativeE);
std::vector<float> SumPtTrkPt1000;
jet->getAttribute(xAOD::JetAttribute::SumPtTrkPt1000, SumPtTrkPt1000);
float chf             = SumPtTrkPt1000.size() > 0 ? SumPtTrkPt1000.at(0)/jet->pt() : -1;
float emf             = jet->getAttribute<float>(xAOD::JetAttribute::EMFrac);
float hecf            = jet->getAttribute<float>(xAOD::JetAttribute::HECFrac);
float fracSamplingMax = jet->getAttribute<float>(xAOD::JetAttribute::FracSamplingMax);
```

```

if (
    !(
        (hecf>0.5 && fabs(hecq)>0.5) || (fabs(negE) > 60*CLHEP::GeV) ||
        (emf>0.95 && fabs(jetQuality)>0.8 && fabs(jet->eta()) < 2.8) ||
        (fabs(jetTime)>25) ||
        (emf<0.05 && chf<0.05 && fabs(jet->eta())<2) ||
        (emf<0.05 && fabs(jet->eta())>= 2) ||
        (fracSamplingMax>0.99 && fabs(jet->eta())<2)
    )
)

```

## Kinematic cuts

Jets are required to be with pT higher than 15 GeV and pseudorapidity lower than 2.5:

```
if ( jet->pt() / CLHEP::GeV < m_jet_pt_cut || fabs(jet->eta()) > m_jet_eta_cut )
```

with:

```
declareProperty("JetPt_cuts", m_jet_pt_cut = 15.0); // in GeV
declareProperty("JetEta_cuts", m_jet_eta_cut = 2.5);
```

## JVT cut

Jets are required to pass standard JVT cut used by physics analyses:

```
static SG::AuxElement::Accessor<float> JVT( "Jvt" ); //JVT > 0.59 (60 GeV)
double jvt = JVT(*jet);
if( !( ((jet->pt())<60000)&&(abs(jet->eta())<2.4)&&(jvt > 0.59)) || ((jet->pt())<60000)&&(abs(jet->et
```

## Taggability: definitions

The "Taggability" quality of jets is defined based on the number of tracks passing a certain number of requirements. It is performed by trackSelectorTool on ghost associated tracks:

```

m_trackSelectorTool->primaryVertex(m_priVtx->position());
m_trackSelectorTool->prepare();

std::vector<const xAOD::IParticle*> trackVector = jet->getAssociatedObjects<xAOD::IParticle>(xAOD

```

The requirement on the tracks include 16 items:

```

void JetTagMonitoring::fillBadTrackBits(const std::bitset<17> failedCuts, double eta, double phi)
    if ( failedCuts[pTMin] == 0 )
    if ( failedCuts[d0Max] == 0 )
    if ( failedCuts[z0Max] == 0 )
    if ( failedCuts[sigd0Max] == 0 )
    if ( failedCuts[sigz0Max] == 0 )
    if ( failedCuts[etaMax] == 0 )
    if ( failedCuts[nHitBLayer] == 0 )
    if ( failedCuts[deadBLayer] == 0 )
    if ( failedCuts[nHitPix] == 0 )
    if ( failedCuts[nHitSct] == 0 )
    if ( failedCuts[nHitSi] == 0 )
    if ( failedCuts[nHitTrt] == 0 )
    if ( failedCuts[nHitTrtHighE] == 0 )
    if ( failedCuts[fitChi2] == 0 )
    if ( failedCuts[fitProb] == 0 )
    if ( failedCuts[fitChi2OnNdfMax] ==0 )

```

}

The taggability is assigned as:

```

std::vector<const xAOD::IParticle*>::const_iterator trkItr;
for ( trkItr = trackVector.begin(); trkItr != trackVector.end() ; trkItr++ ) {
    const xAOD::TrackParticle* trackPart = dynamic_cast<const xAOD::TrackParticle*>(*trkItr);
    if ( ! trackPart ) continue;
    if ( m_trackSelectorTool->selectTrack(trackPart) ) {
        nTrk++;
    } else {
        // if at least one associated track does not pass the deadBLayer, nHitBLayer or nSiHit requi
        if ( !passedCuts[deadBLayer] || !passedCuts[nHitBLayer] || !passedCuts[nHitSi] )
            isSuspectJet = true;
    }
}

if ( nTrk < m_trk_n ) return badJet;

if ( isSuspectJet ) return suspectJet;

return goodJet;
}

```

## Taggability (Good)

Jets with "Taggability Good" label are required to have at least 1 track passing the trackSelectorTool requirements.

## Taggability (Suspect)

Jets with "Taggability Suspect" label are jets having at least 1 track not passing deadBLayer, nHitBLayer or nSiHit requirements from trackSelectorTool:

```

// if at least one associated track does not pass the deadBLayer, nHitBLayer or nSiHit requi
if ( !passedCuts[deadBLayer] || !passedCuts[nHitBLayer] || !passedCuts[nHitSi] )
    isSuspectJet = true;

```

## Taggability (Bad)

Jets with "Taggability Bad" label are jets having no tracks passing trackSelectorTool requirements:

```
if ( nTrk < m_trk_n ) return badJet;
```

with:

```
declareProperty("nTrk_cuts", m_trk_n = 1);
```

---

### Major updates:

-- AlessandroLapertosa - 2018-03-23

%RESPONSIBLE% AlessandroLapertosa

%REVIEW% **Never reviewed**

---

This topic: [Sandbox > DQMonitoringFTAGJetSelection](#)

Topic revision: r1 - 2018-03-23 - AlessandroLapertosa





Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)