`MuX` user guide

Keith Pedersen (kpeders1@hawk.iit.edu)

November 30, 2015

Abstract

This user guide explains how to use the `MuX` module for DELPHES, along with some of its internal kinematic expressions. This guide assumes that the reader has read the paper, and understands the basics of the μ_x tag.

1 Using the module

The `MuXboostedBTagger` module (henceforth frequently abbreviated as `MuX`) is a relatively straightforward extension to the standard DELPHES functionality. Like DELPHES' native `BTagging` module, `MuX` draws from `JetInputArray`, a list of fully reconstructed/calibrated jets (e.g. from `JetEnergyScale`), and only alters their `Candidate::BTag` field. Additionally, to allow `MuX` to be used in conjunction with `BTagging`, `MuX` allows the user to define *which* bit in `BTag` is set when a jet is tagged (the default is bit-3 for `MuX`, versus bit-0 for `BTagging`).

However, unlike `BTagging`, `MuX` must actually *alter* the jest it tags. This is because the μ_x tag relies on jets undergoing semi-leptonic decay, which systematically lack neutrino energy. This requires an additional level of jet energy correction, in the form of **neutrino estimation**. Currently, only the simplest estimation is implemented: the associated muon is used as an exact proxy for the neutrino ($p_{\nu_\mu} = p_\mu$), since most of the momentum of both comes from their shared boost.

In order to accomadate this neutrino estimation in a **sensible** way, the only field which `MuX` can alter on its `JetInputArray` is `Candidate::BTag`. These jets are then cloned, and the neutrino estimated jets are sent to the `JetOutputArray`. Thus, both input/output jets contain the same jets, in the same order, with the same `Candidate::BTag` field, and only the difference is that jets in `JetOutputArray` which are *tagged* by μ_x have their `Candidate::Momentum` field altered to estimate their neutrino(s).

Parallel lists of tagged jets can be iterated through in parallel during post-detector analysis code. For exmample, consider an alysis quite useful in an analyses which requires only 1 b-tag. *If* only one jet is μ_x tagged, then the event $\not{E}_T$ can be used for neutrino estimation instead. However, since the event could also involve other sources of $\not{E}_T$ (e.g.), this decision should only only be made in post-production — not inside DELPHES.

```
module HighPtBTagger HighPtBTagger {
set JetInputArray JetEnergyScale/jets

set BitNumber             3
set MaxJetRank            4
set MinJetPt              135.
set MinMuonPt             10
set MinTowerPtRatio       0.05
set CoreAntiktR           0.04
set CorePtRatioMin        0.5
set CoreMassHypothesis    2.0
set SubjetMassHypothesis  5.3
set MinFinalMass          0.
set MaxFinalMass          12.
set MaxX                  3.0

set BCoreMinBoost         1.0
}
```

2 Code notes

2.1 Robust subjet mass

We define the subjet of B hadron decay

$$p_{\text{subjet}} = p_{\text{core}} + p_{\mu} + p_{\nu_\mu}, \tag{1}$$

where the "core" is ostensibly a charm hadron. Since the neutrino is not observable, we estimate it using the simplest choice ($p_{\nu_\mu} = p_\mu$, since most of the momentum comes from their shared boost). This gives us an approximate subjet

$$p_{\text{subjet}} \approx p_{\text{core}} + 2\,p_\mu. \tag{2}$$

The core is found by reclustering the jet using a much smaller radius parameter. This produces a list of *candidate* cores, of which only one is the "correct" core; this is found by using the hardest muon in the jet to find the core which produces $\sqrt{p^2_{\text{subjet}}}$ closest to $m_B = 5.3$ GeV (the mass of a B hadron admixture). However, since m_{core} is not accurately measured (from the granularity of the Calorimeter), we must first constrain it to its hypothesized mass under the B hadron decay hypothesis ($m_{\text{core}} = 2$ GeV).

In order to find the core quickly, we don't want to correct the mass of each core candidate before adding the muon. Instead, we can use our large cut on muon p_T to treat the muon as massless, then calculate the mass of the subjet analytically

$$p^2_{\text{subjet}} = m^2_{\text{core}} + 2\,p_{\text{core}} \cdot p_\mu = m^2_{\text{core}} + 4\,E_\mu E_{\text{core}}(1 - \cos(\xi)\sqrt{1-g}), \tag{3}$$

where ξ is the angle between the muon and the core and $g \equiv \gamma_{\mathrm{core}}^{-2}$. Using $y = \tan^2(\xi) = \left(\frac{\vec{p}_{\mathrm{core}} \times \vec{p}_\mu}{\vec{p}_{\mathrm{core}} \cdot \vec{p}_\mu}\right)^2$ and $\cos(\xi) = \frac{1}{\sqrt{1+y}}$ (since $\xi < \pi/2$), this becomes

$$p_{\mathrm{subjet}}^2 \approx m_{\mathrm{core}}^2 + 4\, E_\mu E_{\mathrm{core}} (1 - \frac{\sqrt{1-g}}{\sqrt{1+y}}). \tag{4}$$

This expression can be optimized to minimize floating point cancellation, (which should be miniscule, but eliminating unnecessary cancellation is a good habit)

$$p_{\mathrm{subjet}}^2 \approx m_{\mathrm{core}}^2 + 4\, E_\mu E_{\mathrm{core}} \frac{g+y}{1+y+\sqrt{1-((g-y)+g\,y)}}. \tag{5}$$