Documentation and Manual for TauFinder Marlin processor

A. Muennich

CERN, CH-1211 Geneva 23, Switzerland

Abstract

The algorithm developed targets τs that produce narrow, low multiplicity jets, therefore works best on high energetic τs. This note provides an overview of the implemented algorithm, the cuts used and gives some evaluation of the performance. The last section is intended as a short user manual.

Contents

1 Documentation **2**

1.1 The Algorithm 2
1.2 The Selection Cuts 2
1.3 Evaluation 3
1.4 Results 4

2 User Manual **6**

2.1 Preparing the Input 6
2.2 The Output 7
2.3 Running the Processor 7

1. Documentation

1.1. The Algorithm

The algorithm is very much like a jet finder cone algorithm with some special criteria and cuts. The steps taken are the following:

1. Starting with the charged particle with the highest energy it is tested as a seed for the τ candidate based on transverse momentum and impact parameter.
2. Once a seed is found the remaining charged particles are checked and the ones within the search cone around the seed are added to the τ candidate adjusting it for the new combined momentum. The search cone is defined by the opening angle between the momenta of the two particles and is specified by the user.
3. After that the neutral particles are added to the τ candidate in the same fashion.
4. Starting again at step 2 with the seed for the next τ candidate the steps are repeated until no more seed is found.
5. Combine all particles associated to one τ candidate to form a τ
6. After all τ candidates in the event are found a check is performed whether one candidate was split in two. If two τ candidates lie within the search cone they are merged.

The τ is then evaluated based on a few selection cuts discussed in the following section 1.2. For the isolation criterium for example the energy of all particles outside the search cone but inside the isolation cone is summed up. A limit on that energy can be specified to reject τ candidates.

1.2. The Selection Cuts

There are a couple of cuts to influence the algorithm. Some are fixed and others can be changed by the user:

Fixed Cuts

For the seed of the τ candidate:

- A minimum transverse momentum is required: $p_t > 5$ GeV

For the acceptance of a τ candidate:

- The invariant mass should be below 2 GeV. This allows for some contamination, meaning particles withing the τ candidate that do not belong there, which is likely to happen in the case of background.

- The multiplicity of the τ-jet is low, therefore the number of charged tracks must be larger than zero but smaller than six.
- The total number of charged and neutral particles combined to a τ should be below 10 which is based on studying the τ decay products based on Monte Carlo (MC).

User Parameters
Cuts that can be set by the user with the default values given in backets, are:

- A general cut to suppress background by requiring a minimum transverse momentum for a particle to be considered (p_t=0).
- A limit on the impact parameter D0 for the seed (D0 $<$ 0.5 mm).
- The opening angle of the search cone (0.05 rad).
- The opening angle of the isolation ring given relative to the search cone (0.02 rad).
- A limit on the energy sum of all particles that is allowed within the isolation cone (5 GeV).

1.3. Evaluation

In order to evaluate the algorithm the following variables are used:

- N_τ: number of τs in the Monte Carlo Truth
- *Missed*: number of τs not recognized, e.g. seed not found, or rejected by selection cuts
- *Reconstructed*: number of τs reconstructed
- *Matched*: number of reconstructed τs where at least one of the particles used to form the τ links back to the τ in the MC truth.
- *Fake*: number of reconstructed τs where none of the particles used to form the τ links back to the τ in the MC truth.
- *Clean*: number of reconstructed τs where all the particles used to form the τ links back to the τ in the MC truth.
- *Contaminated*: Difference between Matched and Clean.
- *dE<1%*: number of reconstructed τs where the deviation of the reconstructed energy from the MC energy is less than 1%.

Figure 1 illustrates how a sample splits into the different contributions giving examplary statistics for a data sample of Charginos.

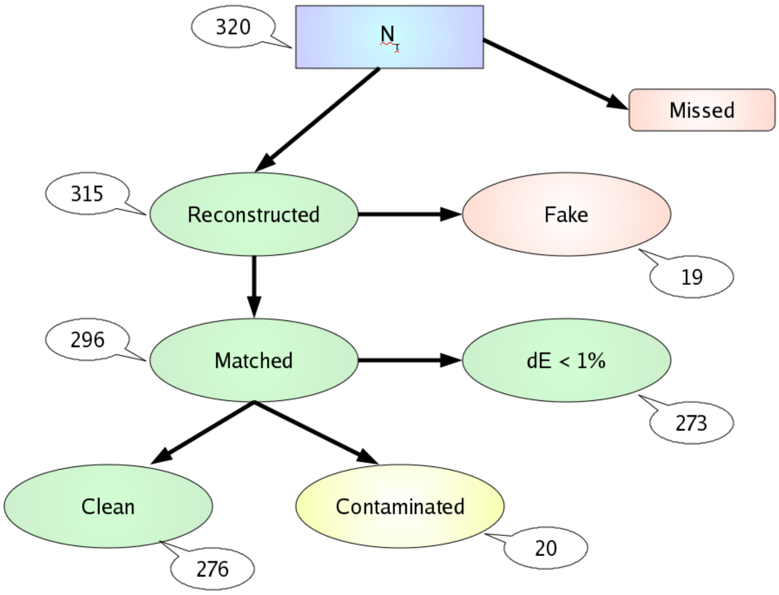

Figure 1: Illustration of nomenclature within a data sample. The numbers are an example of the statistics when running the TauFinder on a data sample of Charginos.

In order to define the efficiency and purity the important variables are *MC*, *Matched* and *Reconstructed*:

$$\begin{aligned} Efficiency &= \frac{Matched}{N_\tau} = 92.5\% \\ Purity &= \frac{Matched}{Reconstructed} = 94.9\% \\ Eff_E &= \frac{dE < 1\%}{Matched} = 92.2\% \end{aligned}$$

The numbers given correspond to the example illustrated in fig. 1.

1.4. Results

To give some results a parameter scan was carried out with the following values:

- $p_t > 0, 1$ GeV
- D0 $< 0.3, 0.4, 0.5$ mm
- Search cone: 0.03, 0.04, 0.05, 0.06, 0.07 rad
- Isolation cone: 0.01, 0.02, 0.03 rad
- Isolation energy $< 3, 5, 7, 10$ GeV

on three different data sets at different $\gamma\gamma \rightarrow$ h background levels:

1. Staus (m=900 GeV) with 0 and 20 BX
2. H (m=1194 GeV) A (m=1128 GeV) 0, 20 and 40 BX
3. Charginos (m=1064 GeV) with 0 BX

The stau samples only have τs in the event that can only be compromised by a photon radiated off earlier in the process chain which is of high energy. Therefore the efficiency is not 100% but the purity is always 100%. The chargino sample also reaches a high effciency and only slightly lower purity. In the case of HA the τs are less energetic hence the algorithm is less efficient. Fig.2 shows a comparison of the performance of the TauFinder on the different event topologies without background.

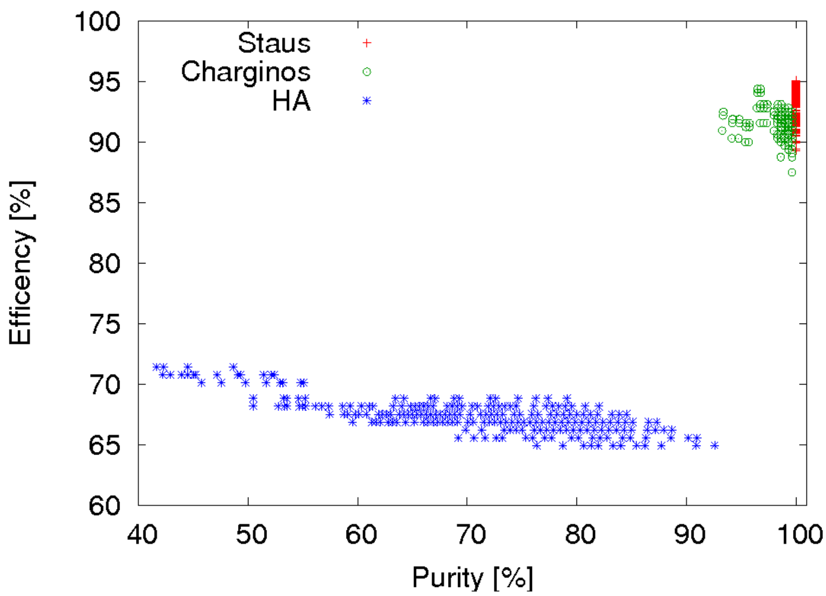

Figure 2: Performance of the TauFinder for three different data sets without background.

Fig.3 shows a comparison of the performance of the TauFinder on a data sample of HA for different background levels.

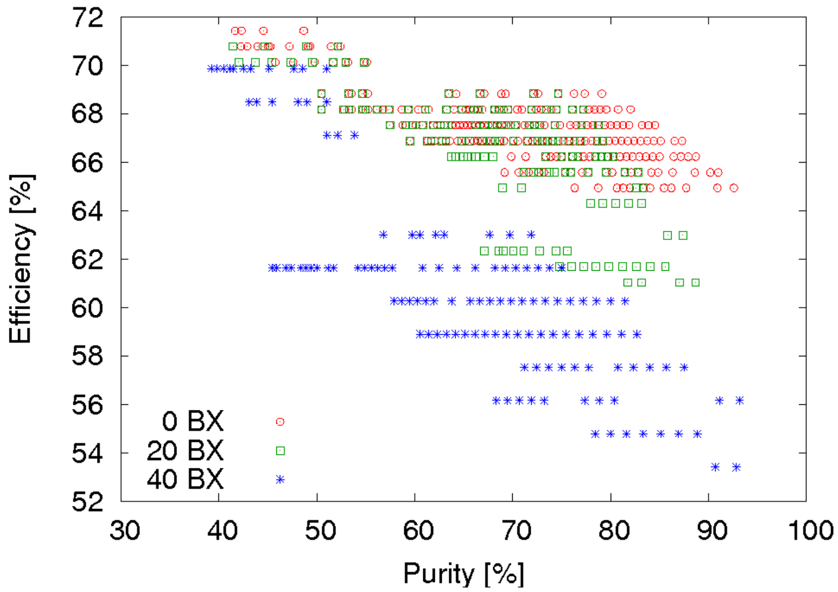

Figure 3: Performance of the TauFinder on HA for three different background levels.

2. User Manual

This is the more technical part, explaining how to set up and run the processor. The assumption is a working installation of Marlin.

2.1. Preparing the Input

The TauFinder runs on a LCIO Collection containing the LCIO objects ReconstructedParticle. In order to run the processor on Monte Carlo truth or just tracks or a combination of tracks and clusters a pre-processor has to be executed called PrepareRECParticles. Per default it fills Tracks, MCParticles and ReconstructedParticles from Pandora into a new collection of ReconstructedParticles. It is necessary to also convert the Pandora output even though these are already ReconstructedParticles because the TauFinder needs a reference point for the particle in order to compute the impact parameter. But Pandora does neither set a reference point nor does it set the StartVertex. This is done by the processor PrepareRECParcticles

using the the track with the highest momentum so set the StartVertex. This processor can be extended to convert any object or combination thereof the user wants to run the TauFinder on.
The following functions of ReconstructedParticle will be called in the TauFinder and have to be set in the conversion in order to provide the TauFinder with the necessary information:

- getMomentum()
- getCharge()
- getStartVertex()
- getEnergy()
- getType()

The last one is not necessary for the algorithm to run, but is just used to print out information, so it can be set to 0 or any other value if the information is not available. The other items are essential for the computation of the impact parameter and the angle between the seed and the particle. In addition a value for the magnetic field has to be supplied via the GEAR file. This is also needed for the computation of the impact parameter.

2.2. The Output

The TauFinder will write a new collection with the τs as ReconstructedParticles. In addition it will provide a series of LCRelations that allow to trace the τ back to the original input that was provided. The processor EvaluateTauFinder gives an example on how this is done.

2.3. Running the Processor

One complete example steering file to run the TauFinder that first uses the PrepareRECParticles to provide the input for the TauFinder based on Tracks, MC and Pandora PFOs will be available with the processor. Here just the main part to configure the TauFinder is listed:

```
<processor name="MyPrepareRECParticles" type="PrepareRECParticles">
   <parameter name="outputColMC" value="MCParticles_tau"/>
</processor>

<processor name="MyTauFinder_MC" type="TauFinder">
   <parameter name="inputCol" value="MCParticles_tau"/>
   <parameter name="outputCol" value="TauRec_MC"/>
   <parameter name="relCol" value="TauRecLink_MC"/>
   <parameter name="pt_cut" value="1"/>
   <parameter name="D0seed" value="0.5"/>
   <parameter name="searchConeAngle" value="0.07"/>
```

```
   <parameter name="isolationConeAngle" value="0.03"/>
   <parameter name="isolationEnergy" value="5.0"/>
</processor>
```