Qualification Task AFT-487

Martino Centonze

Unisalento, INFN Lecce

March 1, 2021

The workflow

My QT aims at studying the efficiency of b-tagging as a function of track reconstruction efficiency and track selection acceptance.
It needs MC truth for computing the efficiencies.
The used MC simulated samples are DAODs (Derived AODs) with b-tagging decorations.These are processed by the following three tools:

1. FlavourTagPerformanceFramework
2. Selector
3. ROC maker

FlavourTagPerformanceFramework

Extrapolate useful b-tagging related quantities of interest for a generic analysis. It runs on the DAODs as a dumper or on AODs, simulating the whole b-tagging process in athena by applying the typical b-tagging workflow and producing custom derivations.
In practice, my interest is to study official derivations, so the tool is typically run on DAODs.
Major updates:

1. made the tool capable of running on ghost collections
2. added some truth variables (d0, z0, ..)
3. developed a new track decorator, showing up the "origin" of each reconstructed track

`https://gitlab.cern.ch/mcentonz/FlavourTagPerformanceFramework/-/tree/mcentonz_UCL_norigin`

Selector

This tool has been written down from scratch.
It dumps some selected information from the FTPF output and uses it for a further b-tag oriented analysis. In particular:

1. applies a further custom selection of tracks and jets
2. dumps some useful information for subsequent analysis (e.g. hists used by the ROC maker)
3. track-association acceptance tool: computes the acceptance of the combined track association and track selection processes for the B/C decay products (so called "children")

`https://github.com/martinosal/root_selector`

ROC maker

Produces efficiency curves (ROCs) from the output of the selector for each algorithm in the sample; comparison plots show the difference in performance for different algorithms and/or for different track selection/association schemes.
The tool is written in python, it uses the UPROOT tool for converting ntuples from root.
`https://github.com/martinosal/ROCs/blob/main/multiple_roc_dyn.ipynb`

B-tagging in a nutshell

We have two different categories of taggers:

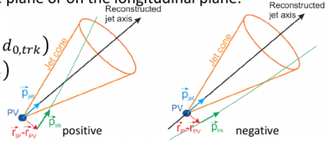
1. vertexing based: SV0, SV1, JetFitter
2. Impact Parameter (IP) based: IP2D, IP3D, RNNIP/DIPS

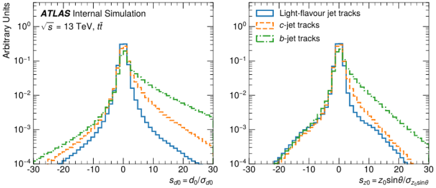
We also have high level algorithms, based on neural network architecture: DL1, DL1r and DL1rmu (not recommended).

- The lifetime sign
  - vector $\Delta\vec{r}_{IP} = \vec{r}_{\mathrm{IP}} - \vec{r}_{\mathrm{PV}}$ defines the three-dimensional impact parameter of the track with respect to the primary vertex.
  - $sign_{3D} = sign([\vec{p}_{trk} \times \vec{p}_{jet}] \cdot [\vec{p}_{trk} \times \Delta\vec{r}_{IP}])$

  can be also defined on the transverse plane or on the longitudinal plane:
  - $IP_{r\phi} = d_0, IP_z = z_0 \sin\theta$
  - $sign_{r\phi} = sign(\sin(\phi_{jet} - \phi_{trk}) \cdot d_{0,trk})$

    $sign_z = sign((\eta_{jet} - \eta_{trk}) \cdot z_{0,trk})$

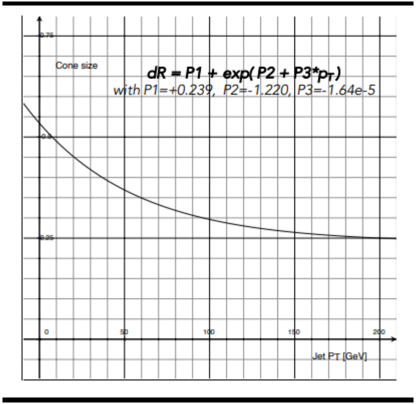




Track selection and association

Shrinking Cone Association

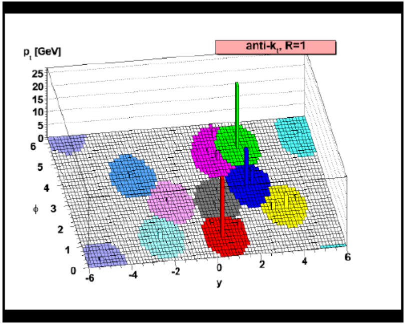
We associate tracks to jet by using a jet-pt variable cone - wider at low p_T and tighter a high p_T - which associates tracks that lie within the cone.



Ghost association

Tracks are associated to the jet by using the AntiKt4 clustering algorithm.

This is practically done by setting the track p_T to an infinitesimal value (from which the word "ghost") and after then the clustering algorithm is run.



Track selection and association

After track association, tracks are filtered out by applying different cuts optimized for different algorithms.
In particular, IP-based algorithms have a tight track selection, while vertexing-based ones have looser cuts.
These processes have the effect of further downstream cuts on tracks.

Track selection for Impact parameter based (IPxD and RNNIP), SVx and JetFitter.

Parameter	Impact parameter based	SV	JetFitter
p_T [GeV]	> 1.0	> 0.7	> 0.5
$\|d_0\|$ [mm]	< 1.0	< 5.0	< 7.0
$\|z_0 \sin\theta\|$ [mm]	< 1.5	< 25	< 10
Number of IBL hits	≥ 1	≥ 0	≥ 0
Number of pixel hits	≥ 2 (≥ 1)	≥ 1	≥ 1
Number of SCT hits	≥ 0	≥ 4	≥ 4
Number of pixel/SCT hits	≥ 7	≥ 7	≥ 7
Number of shared hits	≤ 1	≤ 1	≤ 1
Number of pixel holes	≤ 1	≤ 1	≤ 1
Number of pixel/SCT holes	≤ 2	≤ 2	≤ 2

Jet labeling: the cone based approach

Different jet labeling schemes have been studied.
There are in particular two different labelings: the cone-jet and ghost-jet. They are supposed to be used for different track association schemes: the ShrinkingCone and ghost association respectively.
The cone jet-labeling goes as follows:

1. search for truth b-hadrons in the jet with $p_T > 5$ GeV and $DR < 0.3$ from the jet axis: if no b-hadron satisfying these conditions is found, move on; otherwise label the jet as "b";
2. do the same for c hadrons - try to label the jet as "c";
3. if the previous two labeling attempts fail label the jet as "light"

HadronConeExclTruthLabelID: $0, 4, 5, 15$ for light, b, c and tau jets respectively.

HadronConeExtendedExclTruthLabelID: $0, 4, 5, 15, 44, 54, 55$ (with the $44, 54, 55$ labels referring to the presence of two c hadrons, one b and one c hadron, two b hadrons respectively).

Jet labeling: ghost

The ghost labeling searches for b-hadrons in jets by ghost-associating them: if no b-hadrons are associated the tool looks at c-hadrons. This produces b-jets and c-jets respectively. If neither b nor c hadrons are found, the jet is labeled as light. The flags available under athena are:

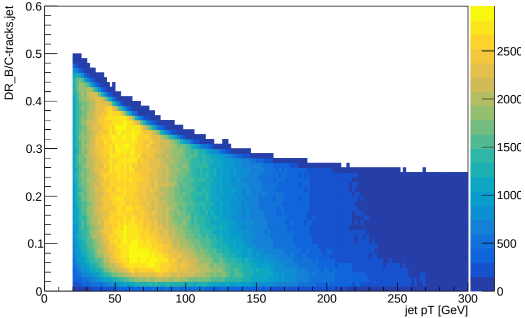
GhostBHadronFinalCount: gives the number of ghost-associated b-hadrons (as truth particles) to the jets

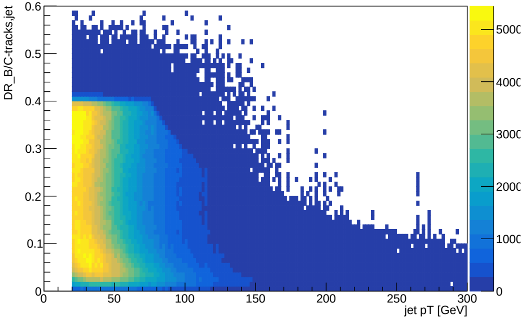
GhostCHadronFinalCount: same for c-hadrons.

It's possible to combine B and C flags for determining whether a given jet should be labeled a b, c or light:

1. b-jets: GhostBHadronFinalCount> 0;
2. c-jets: GhostBHadronFinalCount$= 0$ & GhostCHadronFinalCount> 0;
3. light-jets: GhostBHadronFinalCount$= 0$ & GhostCHadronFinalCount$= 0$;

LEFT: ShrinkingCone associated tracks (ttbar AntiKt4EMPFlowJets collection) with cone labeling.
RIGHT: Ghost associated tracks (ttbar AnitKtVR30Rmax4Rmin02TrackGhostTagJets collection) with ghost labeling.

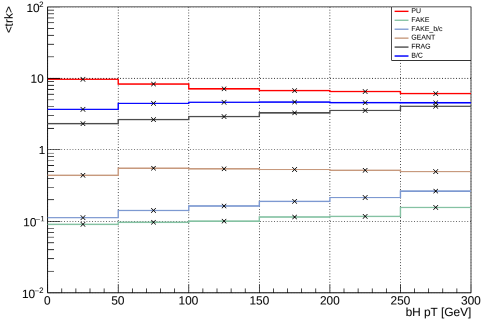


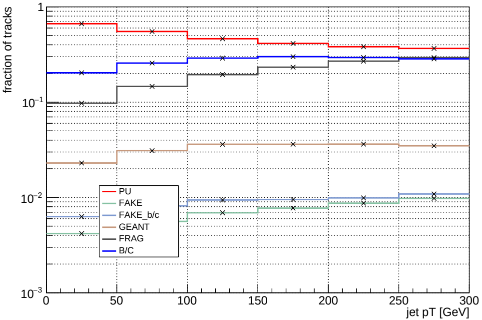


Notice: even if the two jet collections are related to the same ttbar sample, the EMPFlowJets and the VR30 jet collections are different objects with different characteristics: in particular, the pT of the jets are different, which explains the different distribution of tracks as a function of pT of these two plots.

Track composition of jets: ttbar AntiKt4EMPFlowJets

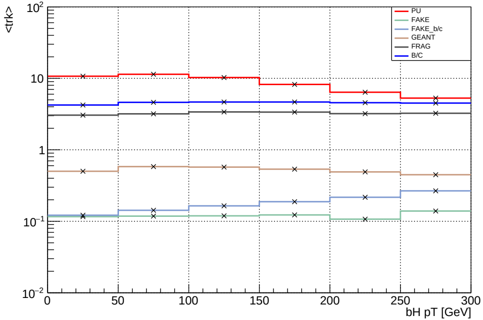
B-Jets

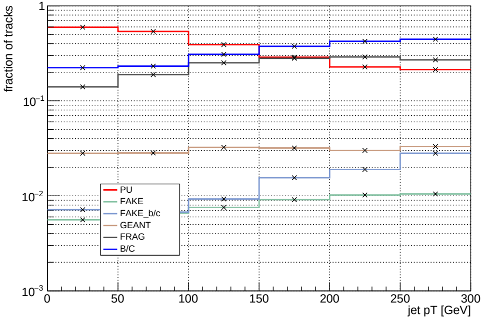




Track composition of jets: ttbar AnitKtVR30Rmax4Rmin02TrackGhostTagJets

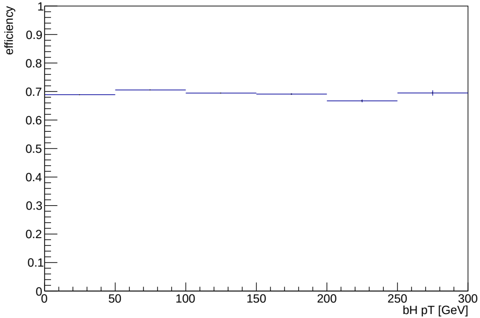
B-Jets

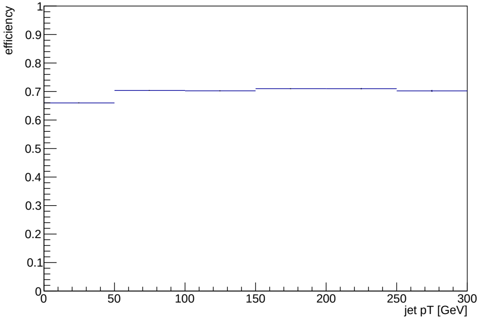




Track association efficiency: ttbar AntiKt4EMPFlowJets

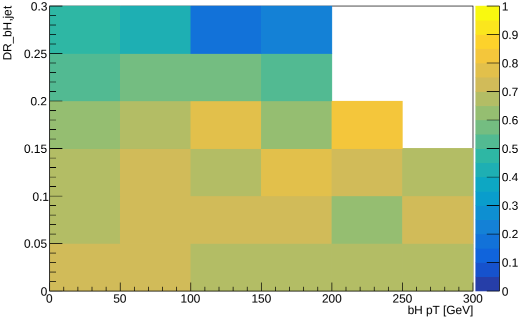
Average track association efficiency (for B/C children): 0.725

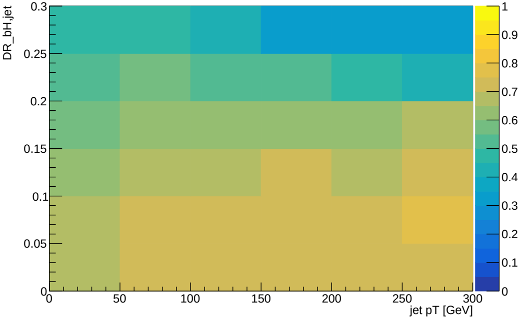




Track association efficiency: ttbar AntiKt4EMPFlowJets

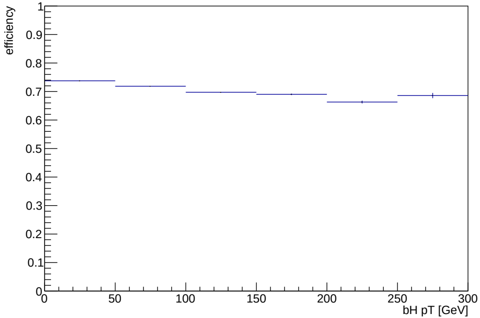
Average track association efficiency (for B/C children): 0.725

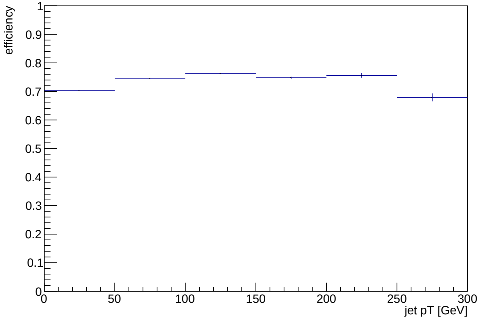




Track association efficiency: ttbar AnitKtVR30Rmax4Rmin02TrackGhostTagJets

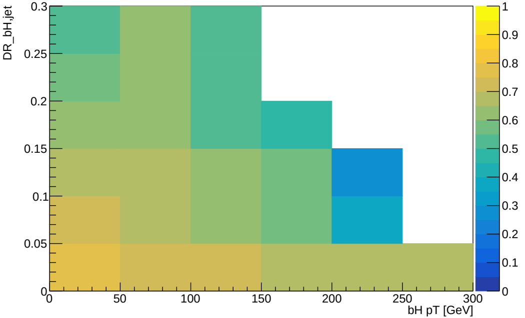
Average track association efficiency (for B/C children): 0.751

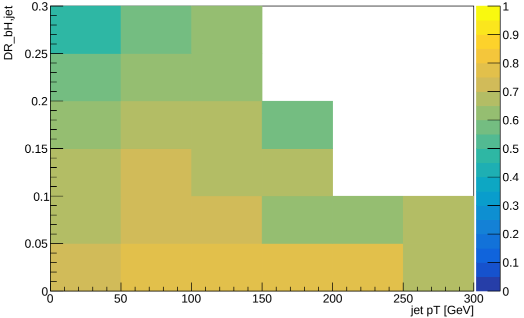


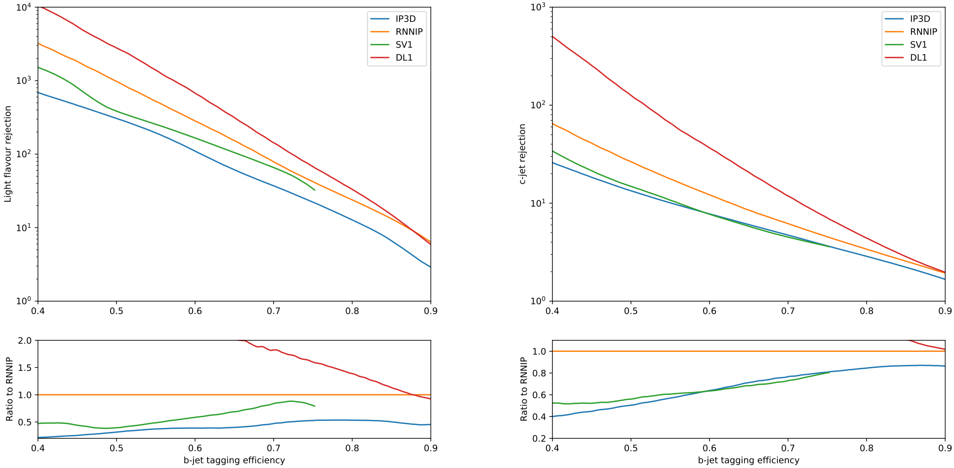


Track association efficiency: ttbar AnitKtVR30Rmax4Rmin02TrackGhostTagJets

Average track association efficiency (for B/C children): 0.751

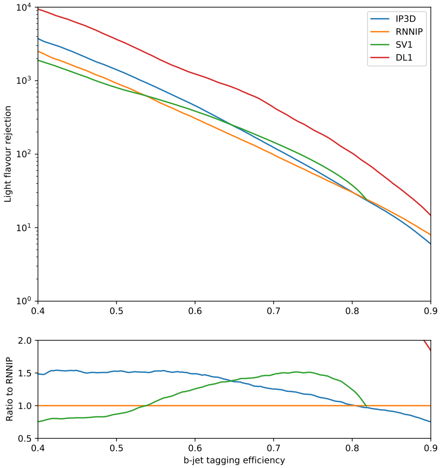


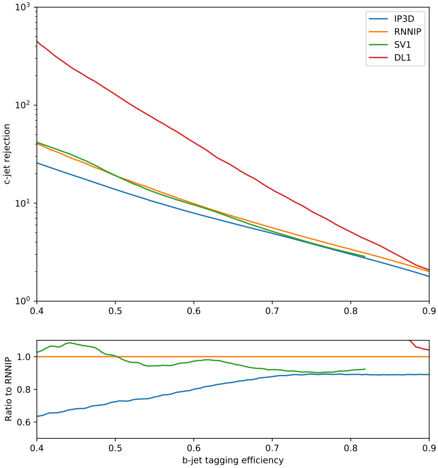




*SV1 maximum efficiency for b-jets: 0.752

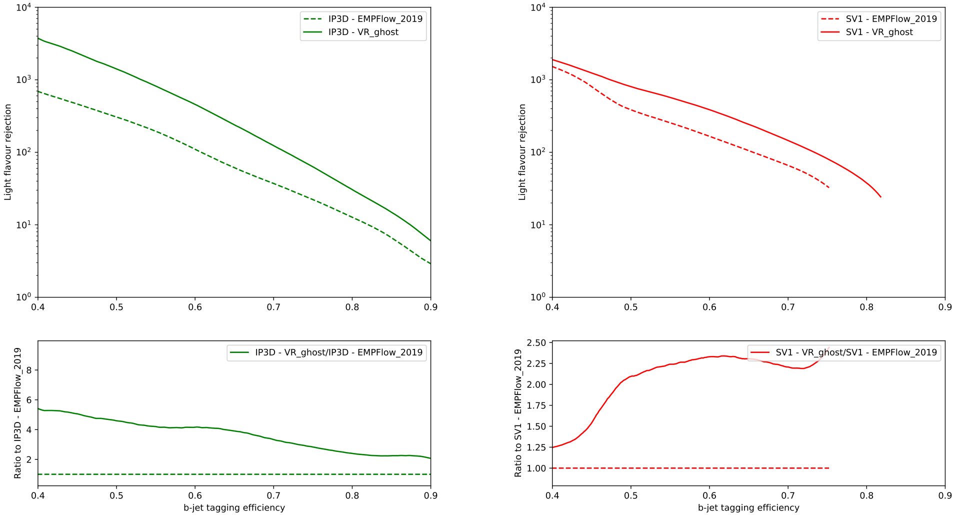
Performance: ttbar AnitKtVR30Rmax4Rmin02TrackGhostTagJets



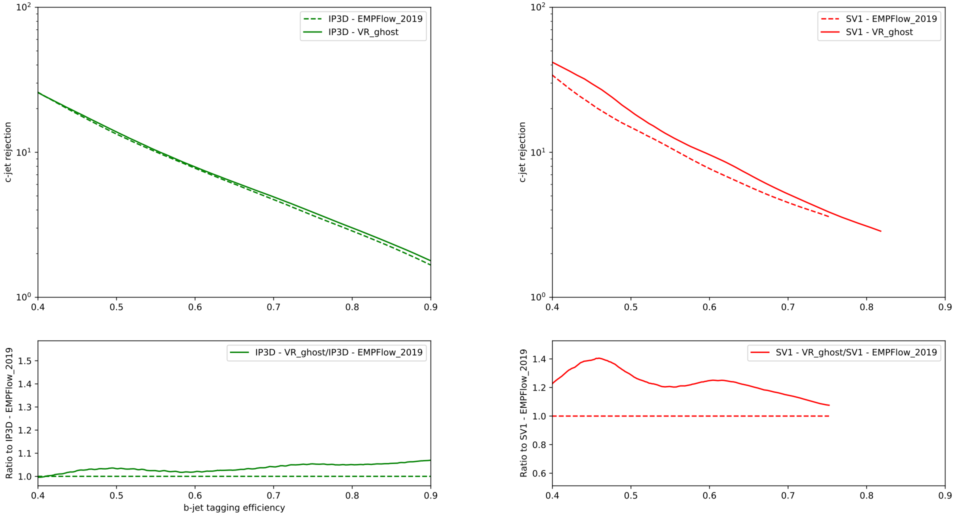


*SV1 maximum efficiency for b-jets: 0.818

Performance: comparison



Performance: comparison



Conclusions

1. The FTPF tool has been consistently integrated, allowing to study the composition of jets in terms of its associted tracks.
2. The Selector and the ROC maker use the information coming from the FTPF for producing plots in an extensive way, for a deeper inspection of jets, tracks and b-tagging itself.

In this presentation the potential of these tools is showed: an higher acceptance of B/C children (which is the case for the VR30-ghost associated collection when compared to the usual EMPFlowJets cone associated collection) clearly leads to better performance of b-tagging, for both low and high level taggers.