Update: Analysis of $ee \rightarrow t\bar{t}$ and jet studies

Julie Munch Torndal

Supervisors:
Jørgen Beck Hansen

Niels Bohr Institute

May 17, 2021

FCC-ee: Analysis of $ee \to t\bar{t}$

Analysis: anomalous EW couplings of the top quark to the photon and the Z boson.

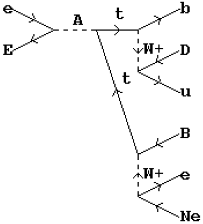
Signal:

Semileptonic channel

$$t\bar{t} \to b\bar{b}W^+W^- \to b\bar{b}q\bar{q}\ell\nu_\ell$$

Planned phase of FCC-ee @ $\sqrt{s} = 365\,GeV$

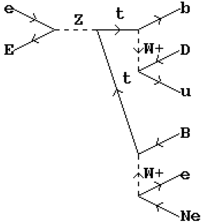
Signature: 1 lepton + $\not{E}$ + 4 jets



Backgrounds:

$$\begin{array}{c|c|c} \begin{array}{c}\mu\mu\\ b\bar{b}\\ \sum q\bar{q}\end{array} & \begin{array}{c}\gamma Z\\ W^+W^-\\ ZZ\end{array} & \underbrace{\begin{array}{c}ZW^+W^-\\ ZZZ\\ \text{single top}\end{array}}_{\substack{\text{LHE files}\\ \text{from MadGraph}}}\end{array}$$

$$\swarrow \quad q = u, d, c, s$$



- All MC files are generated in the FCCSW framework witn DelphesPythia8_EDM4HEP and IDEA Delphes Card

Event Pre-Selection

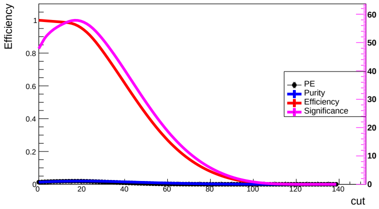
Significance-Optimised Selection Cut Strategy:

Iterative procedure based on maximum significance on a set of variables

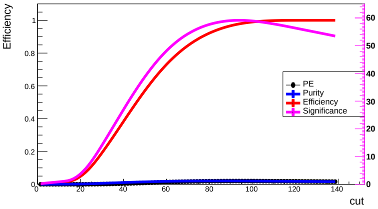
→ choose variable with highest maximum significance

→ make cut and reiterate on remaining set

Lower cut on Highest Energy Lepton



Upper cut on Highest Energy Lepton



Pre-selection cuts

- Exclude events with zero leptons
- Thrust for entire event < 0.85
- 15 GeV $<$Highest energy lepton $<$ 100 GeV
- 2nd highest energy lepton $<$ 40 GeV
- 60 GeV $<$ Invariant mass of lepton-neutrino pair $<$ 110 GeV
- 160 GeV $<$ Invariant mass of event excluding highest energy lepton $<$ 300 GeV
- Jet specific cuts to be determined

→ Jet studies!

Jet Definition

Jet Algorithms
k_t
Anti-k_t
Cambridge/Aachen
Generalised-k_t
Durham
Generalised-k_t for e^+e^-
Valencia
Jade

$\otimes$

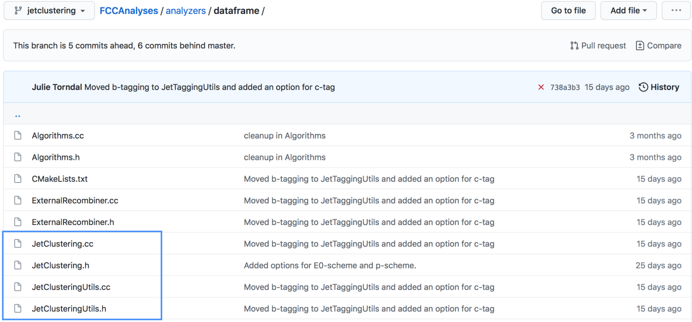
Recombination Schemes
E-scheme
p_t-scheme
p_t^2-scheme
E_t-scheme
E_t^2-scheme
Boost-invariant p_t-scheme
Boost-invariant p_t^2-scheme
$E0$-scheme
p-scheme

The jet definition depends on

- *which* partons are chosen to be combined into the jet
- *how* they are combined into the jet

Jet Clustering Interface

Thank you to Clement Helsens for helping setting up the interface



FCCAnalyses / analyzers / dataframe /

This branch is 5 commits ahead, 6 commits behind master.

Julie Torndal Moved b-tagging to JetTaggingUtils and added an option for c-tag ✕ 738a3b3 15 days ago **History**

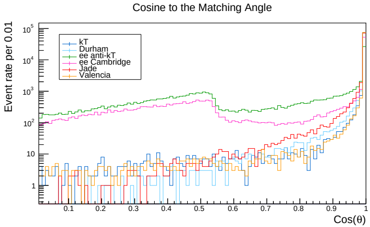
..		
Algorithms.cc	cleanup in Algorithms	3 months ago
Algorithms.h	cleanup in Algorithms	3 months ago
CMakeLists.txt	Moved b-tagging to JetTaggingUtils and added an option for c-tag	15 days ago
ExternalRecombiner.cc	Moved b-tagging to JetTaggingUtils and added an option for c-tag	15 days ago
ExternalRecombiner.h	Moved b-tagging to JetTaggingUtils and added an option for c-tag	15 days ago
JetClustering.cc	Moved b-tagging to JetTaggingUtils and added an option for c-tag	15 days ago
JetClustering.h	Added options for E0-scheme and p-scheme.	25 days ago
JetClusteringUtils.cc	Moved b-tagging to JetTaggingUtils and added an option for c-tag	15 days ago
JetClusteringUtils.h	Moved b-tagging to JetTaggingUtils and added an option for c-tag	15 days ago

Jet reconstruction with Jet Clustering Interface in FCCAnalyses:

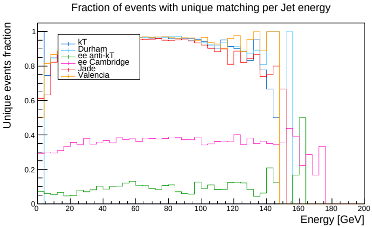
- Interface for later stage process adaptive jet clustering using FastJet
- **Jet Algorithms:** k_t, Durham, e^+e^- anti-k_t, e^+e^- Cambridge, Jade Plugin, Valencia Plugin
- **Recombination schemes:** E-scheme, E0-scheme, p-scheme
- Jet reconstruction with **exclusive clustering up to exactly 4 jets**. Highest energy lepton is excluded from the clustering.

Jet Algorithms

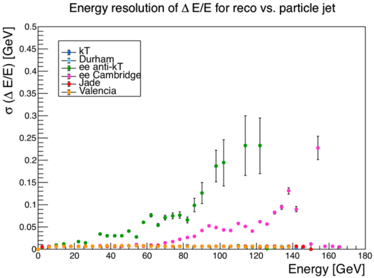
Matching angle between reco and particle jets

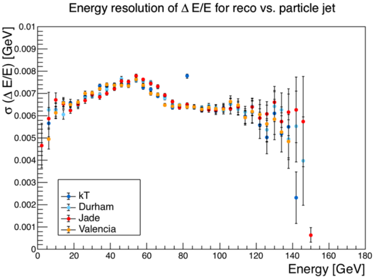


Unique matching between reco and particle jets



Energy resolution of jets (only events w/ unique matching)





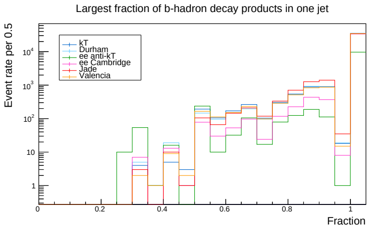
Jet Algorithms

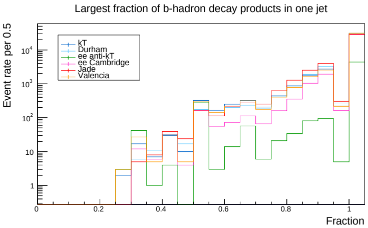
b-hadron distribution in jets

Reco jets

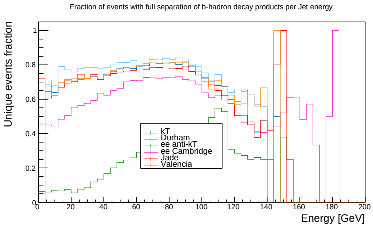
Particle jets

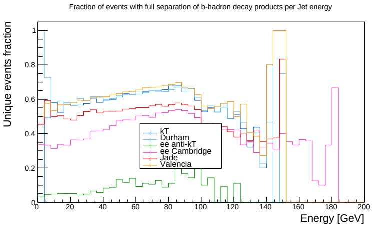
Largest fraction of b-hadron decay products in one jet





Fraction of events with full separation of b-hadron decay products





Recombination schemes

E-scheme: Parton *i* and *j* are replaced by a pseudojet *k* with four-momentum

$$\mathbf{p}_k = \mathbf{p}_i + \mathbf{p}_j$$

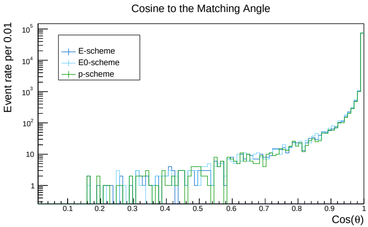
E0-scheme: The four-momentum of pseudojet *k* is rescaled to have zero invariant mass

$$E_k = E_i + E_j \quad , \quad \vec{p}_k = \frac{E_k}{|\vec{p}_i + \vec{p}_j|} \cdot (\vec{p}_i + \vec{p}_j)$$

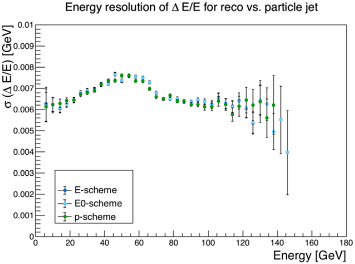
p-scheme: The four-momentum is constructed to have zero invariant mass

$$\vec{p}_k = \vec{p}_i + \vec{p}_j \quad , \quad E_k = |\vec{p}_k|$$

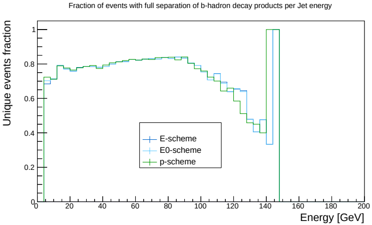
Matching angle



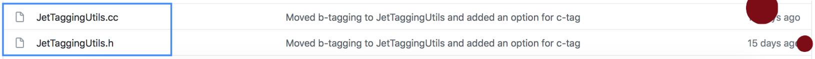
Energy resolution



Full separations of b-hadron decay products



b-tagging a la Delphes



JetTaggingUtils.cc	Moved b-tagging to JetTaggingUtils and added an option for c-tag	[redacted]s ago
JetTaggingUtils.h	Moved b-tagging to JetTaggingUtils and added an option for c-tag	15 days ag[redacted]

Delphes: $\Delta R(\text{jet,b-parton}) < 0.5$

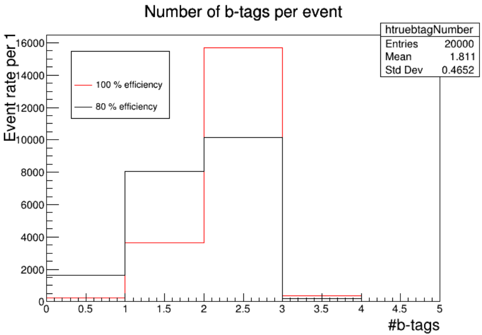
Efficiency formula is applied e.g. in `delphes/cards/delphes_card_IDEA.tcl`:

```
# efficiency formula for b-jets
add EfficiencyFormula {5} {0.80}
```

FCCAnalysis: $\alpha(\text{jet,b-parton}) < 0.3$

It takes a flat efficiency (for now):

```
#Get flavour of jets
.Define("jets_flavour",     "JetTaggingUtils::get_flavour(jets, Particle)")
#Apply efficiency to b-tagged jets
.Define("jets_btag",        "JetTaggingUtils::get_btag(jets_flavour, 0.80)")
```



OBS: Only works for samples generated with Pythia since b-partons are selected from their status code (71-79) and PDG.

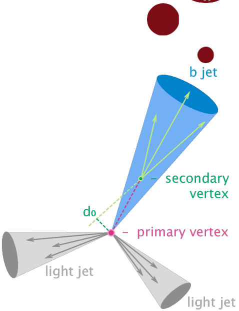
b-tagging with vertexting

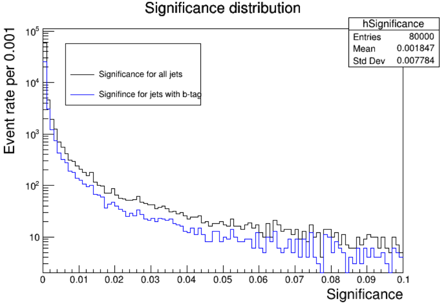
Apply vertex fitter to constituents of a jet and use distance to the IP (0,0,0):

$$\text{significance} = \frac{d}{\sigma_d^2}$$

where σ_d^2 is found using error propagation on the covarince matrix returned by the fitter.

- Assumes good separation of decay products in the jets and does not account for potential tertiary vertex.





Next Steps

Investigate b-tagging with vertexing further

- Primary vs. secondary tracks in a jet
- Look for other correlations

Finalise event selection

- Number of b-tags
- Jet mass
- Jet Energy
- $d_{n,n+1}$
- Significance of fitted vertex to the IP

- Invariant mass for jet systems
  - $\Delta(m_i - m_{W/2})$
  - $\Delta(m_{ij} - m_W)$
  - $\Delta(m_{ijk} - m_t)$
  - $\Delta(m_{l,\nu,i} - m_t)$

Kinematic fit

- Aiming to write software compatible with FCCAnalyses
- Development has recently started
- Inspired by ABC-fit

Backup

Significance-Optimised Cut Strategy for Pre-Selection

List of variables:

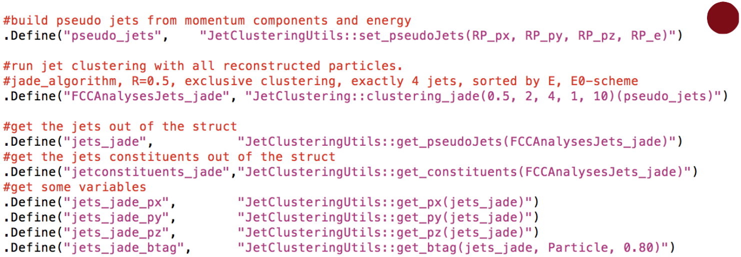
- Highest energy lepton
- 2nd highest energy lepton
- Lepton momentum
- Lepton momentum excluding highest energy lepton
- Momentum difference between highest and second highest energy lepton
- Missing momentum
- Invariant mass of lepton-neutrino pair
- Invariant mass of 1st and 2nd highest energy leptons
- Invariant mass of event excluding highest energy lepton
- Thrust of event excluding highest energy lepton
- Thrust of entire event

Significance of signal for each background before and after Pre-Selection

Background	Sanity check	Pre-Selection
Total	49	315
$\mu\mu$	162	390
$\sum q\bar{q}$	165	389
$b\bar{b}$	196	389
γZ	150	388
WW	60	323
ZZ	233	387
ZWW	405	385
ZZZ	421	390
single top	420	389

$$\textbf{Significance} = \frac{sig}{\sqrt{sig + bkg}} \quad , \quad \textbf{Efficiency} = \frac{sig}{sig_{\text{tot}}} \quad , \quad \textbf{Purity} = \frac{sig}{sig + bkg}$$

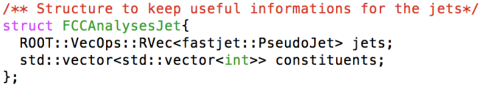
Example code



```
#build pseudo jets from momentum components and energy
.Define("pseudo_jets",    "JetClusteringUtils::set_pseudoJets(RP_px, RP_py, RP_pz, RP_e)")

#run jet clustering with all reconstructed particles.
#jade_algorithm, R=0.5, exclusive clustering, exactly 4 jets, sorted by E, E0-scheme
.Define("FCCAnalysesJets_jade", "JetClustering::clustering_jade(0.5, 2, 4, 1, 10)(pseudo_jets)")

#get the jets out of the struct
.Define("jets_jade",            "JetClusteringUtils::get_pseudoJets(FCCAnalysesJets_jade)")
#get the jets constituents out of the struct
.Define("jetconstituents_jade","JetClusteringUtils::get_constituents(FCCAnalysesJets_jade)")
#get some variables
.Define("jets_jade_px",         "JetClusteringUtils::get_px(jets_jade)")
.Define("jets_jade_py",         "JetClusteringUtils::get_py(jets_jade)")
.Define("jets_jade_pz",         "JetClusteringUtils::get_pz(jets_jade)")
.Define("jets_jade_btag",       "JetClusteringUtils::get_btag(jets_jade, Particle, 0.80)")
```



```
/** Structure to keep useful informations for the jets*/
struct FCCAnalysesJet{
  ROOT::VecOps::RVec<fastjet::PseudoJet> jets;
  std::vector<std::vector<int>> constituents;
};
```

Arguments for jet definition:

1. Jet cone radius
2. Clustering
   - 0=inclusive clustering,
   - 1=exclusive clustering with dcut,
   - 2=exclusive clustering to exactly njets,
   - 3=exclusive clustering up to exactly njets,
   - 4=exclusive clustering with ycut.
3. Cut-value depending on clustering
4. Ordering of returned jets
   - 0=sorted by p_t,
   - 1=sorted by E.
5. Recombination scheme

+. Additional input parameters specific to jet algorithm

- see JetClustering.h