$ee \rightarrow qq$ Working group meeting

A. Irles, 14th January 2018

Analysis group meeting

Outline

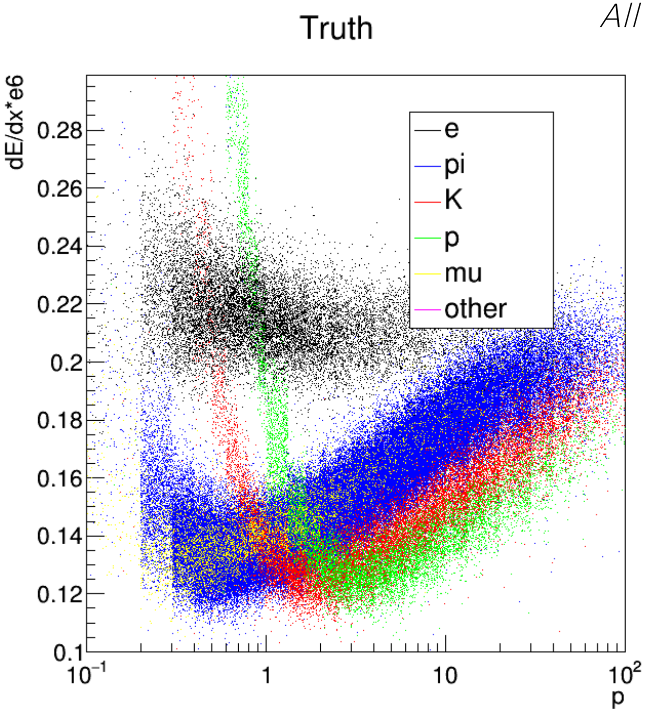
- Update on Kaon identification
- Jet algorithms

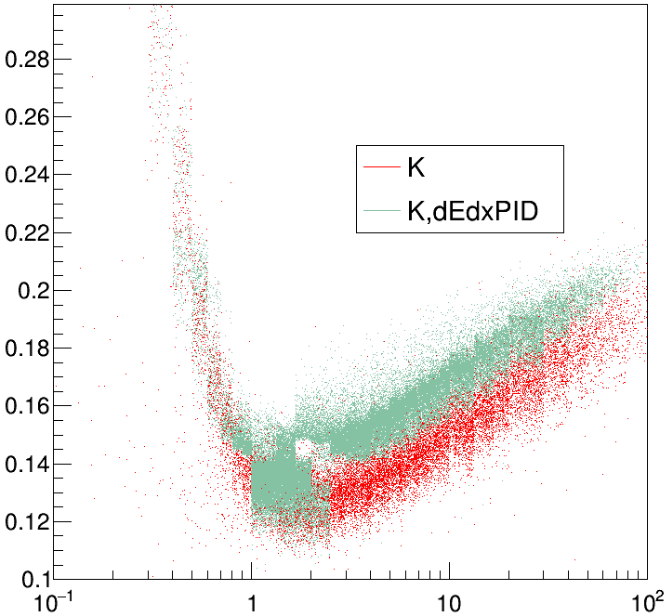
dEdx for particle ID

- Meeting of the 28th November

dE/dx (bb, 500GeV, large model)

All tracks, including primary vertexes



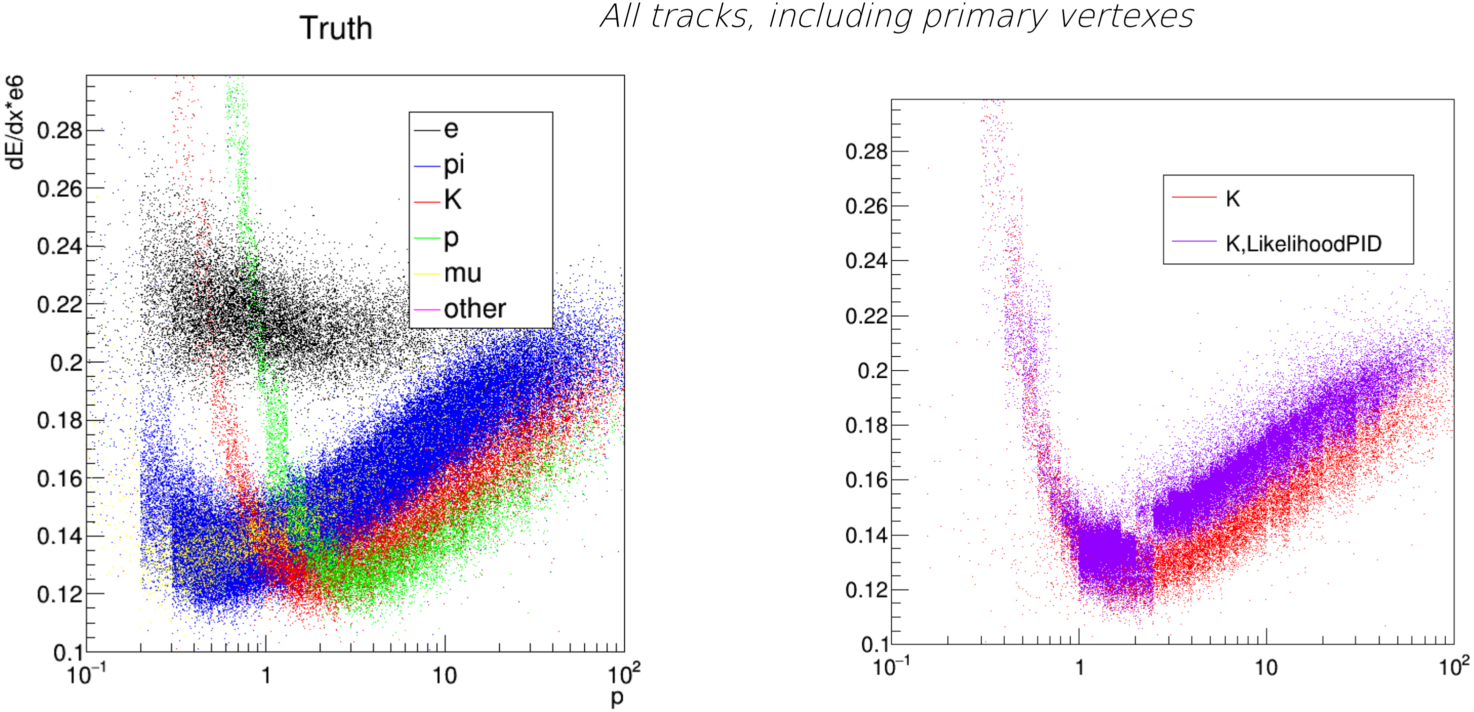


In red, truth kaons (measured dEdx)

- The dEdxPID mistakes kaons and pions at hight momentum. It looks like a simple bug or issue of the parametrization.

dE/dx (bb, 500GeV, large model)

All tracks, including primary vertexes

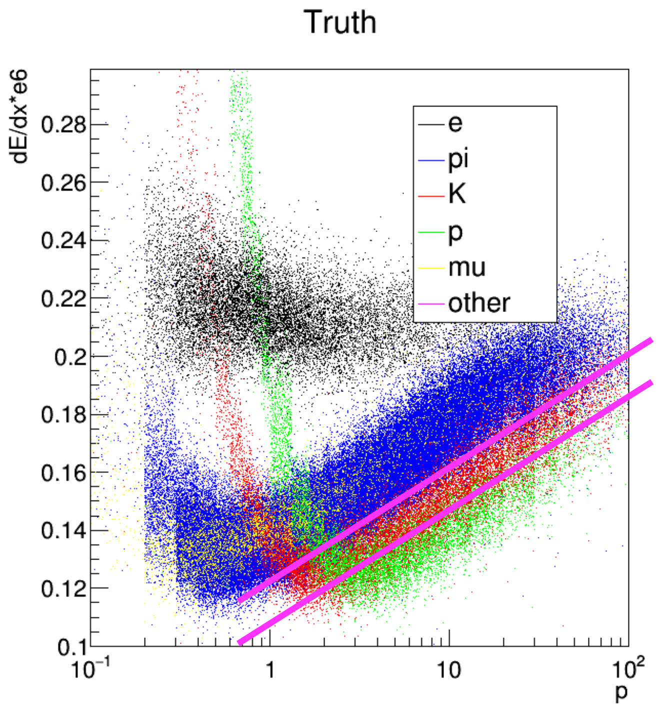


In red, truth kaons (measured dEdx)

- The dEdxPID mistakes kaons and pions at hight momentum. It looks like a simple bug or issue of the parametrization.
- Same for LikelihoodPID since it relies on same algorithm for high momentum.
- For low momentum, both perform better than the simple parametrization. Specially the LikelihoodPID.

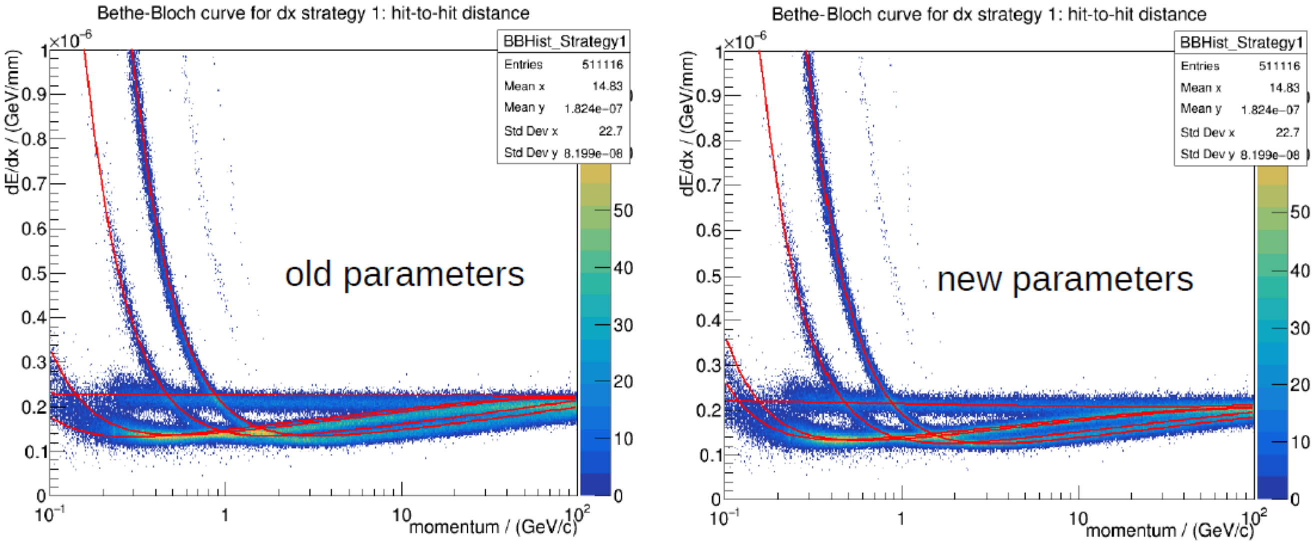
dE/dx (bb, 500GeV, large model)

- We want to recover the good kaon identification for high momentum → Use the ParticleTagger processor in our analysis as a patch.
- Basic idea: parametrize the area where the kaon density is larger and use it for particle separation.
- A **new PIDalgorithm** is created and saved in the PIDHandler of the **PandoraPFOs**
  - KaonTagger
  - Without likelihood or probability value.
- Steering file and info about the processor and the PIDHandler in the backup



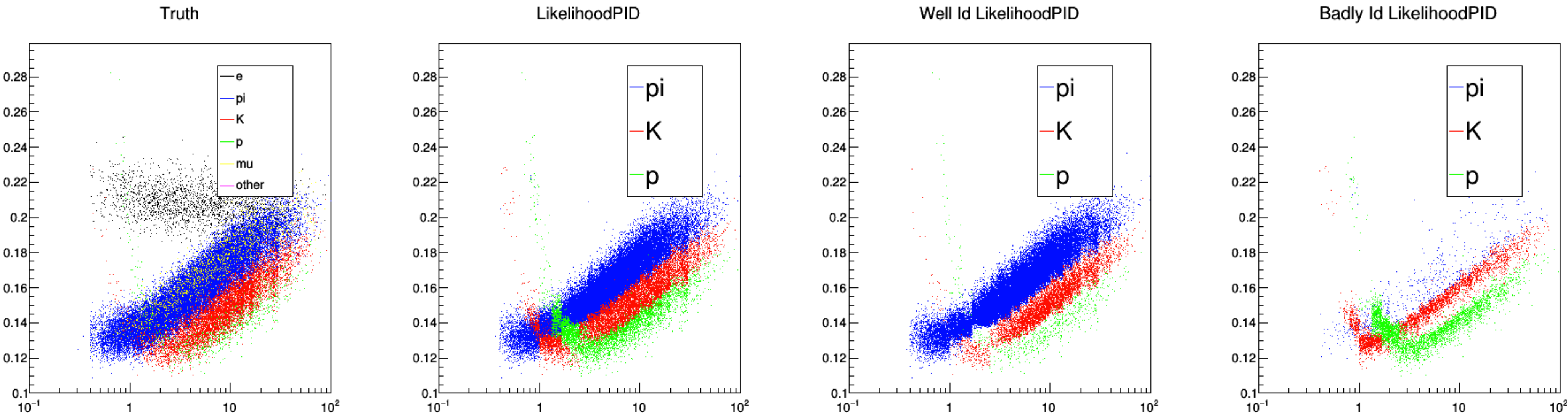
Comparison

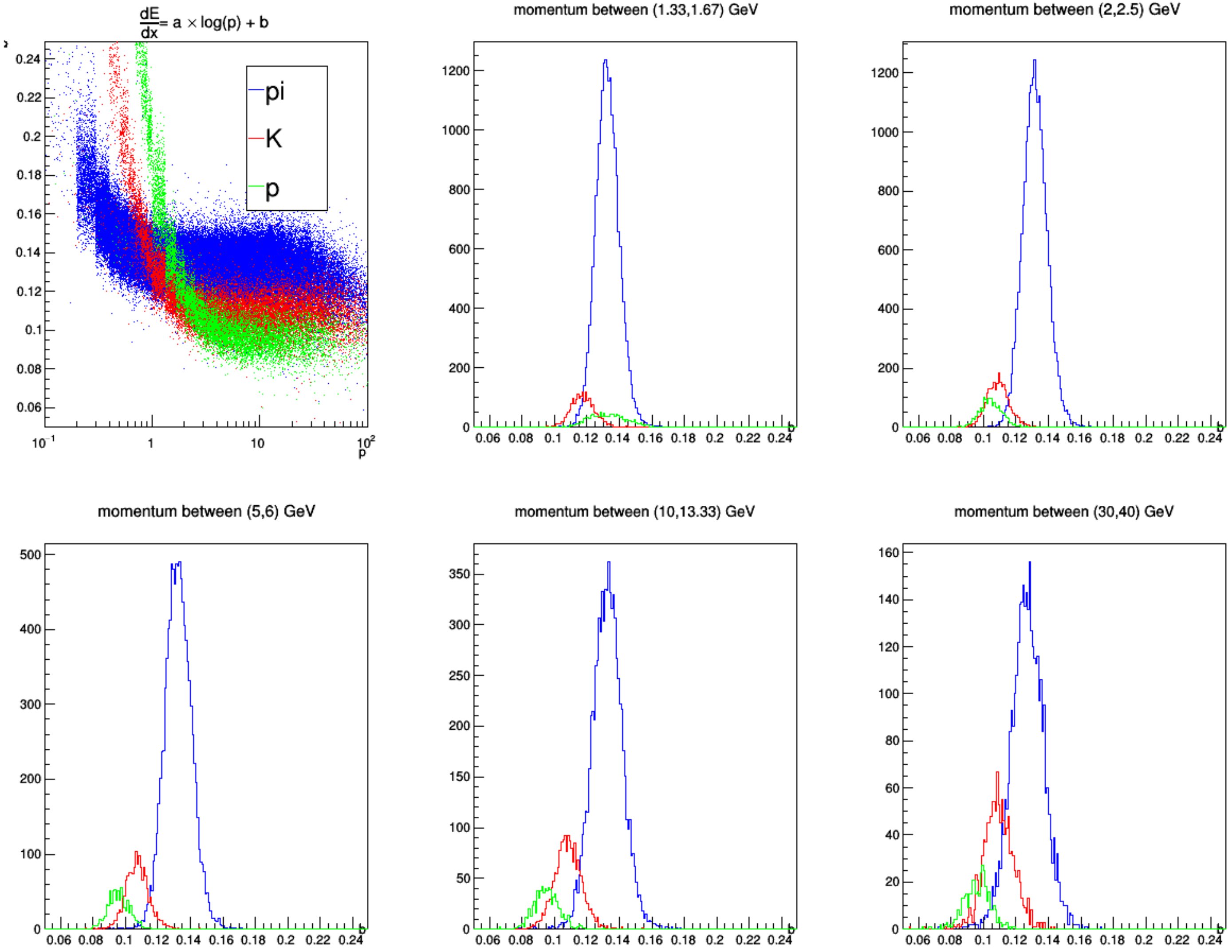
- Likely cause: not-up-to-date parametrisation of Bethe-Bloch curve of expected dE/dx values in dE/dx-PID (inside LikelihoodPIDProcessor)
- Proposed solution: Get new parameters by fitting to current MC-data
- Comparison of parameterised curves with MC-data



Uli Einhaus | Title | 01.01.2016 | Page 2

- New Parametrization… still lot of contamination from pions identified as kaons




dE/dx = a × log(p) + b
pi
K
p
momentum between (1.33,1.67) GeV
momentum between (2,2.5) GeV
momentum between (5,6) GeV
momentum between (10,13.33) GeV
momentum between (30,40) GeV

Jet algorithms (as in implemented in FastJet)

- Durham.

```
JetDefinition jet_def(ee_kt_algorithm);
```

$$d_{ij} = 2\min(E_i^2, E_j^2)(1 - \cos\theta_{ij}) \, .$$

  - A single distance.
  - All objects are clustered.

- kT.

```
JetDefinition jet_def(ee_genkt_algorithm, R, p);
```

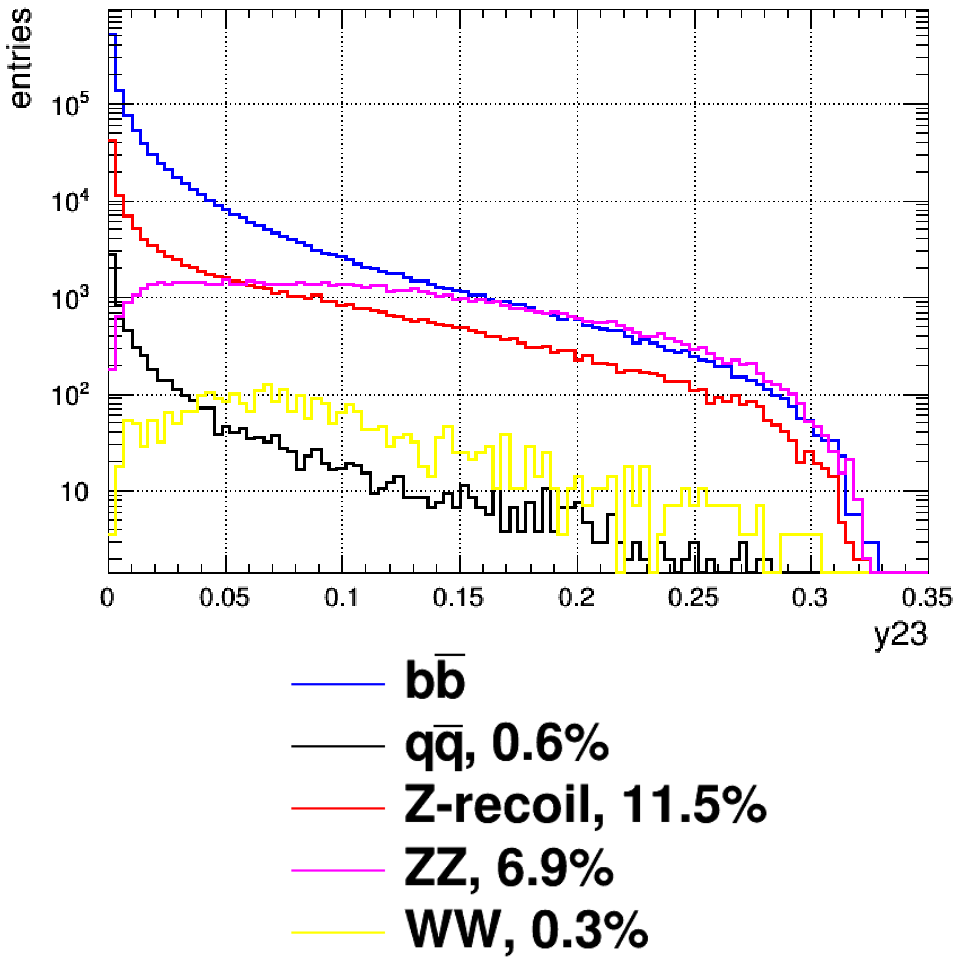
$$d_{ij} = \min(E_i^{2p}, E_j^{2p}) \frac{(1 - \cos\theta_{ij})}{(1 - \cos R)} \, ,$$

$$d_{iB} = E_i^{2p} \, ,$$

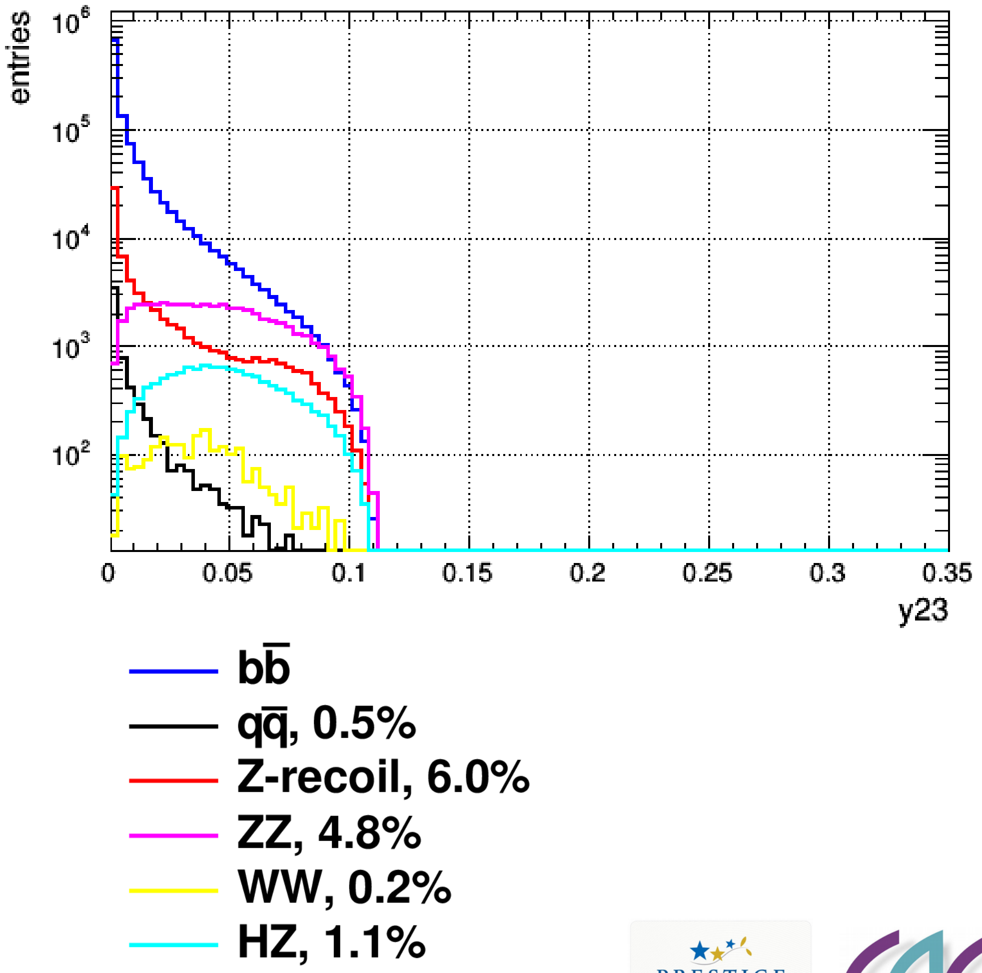
  - Two distances, if the lower is diB, the jet is removed from the list (as a beam particle or a soft radiation)
  - All objects are clustered.

- In Marlin, dij = y_{ij} (the "d" is used for the same distance but normalized by the measured energy)
- Exclusive reconstruction: we force the algorithm to cluster 2 jets, even if the topology is not "two jet like"
- y12, y23 are the typical variables to study the quality of the exclusive reconstruction
  - The distance at which two jets will recombine in only one, or two jets would be split in 3.

Durham vs kt



entries
10^5
10^4
10^3
10^2
10
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
y23
$b\bar{b}$
$q\bar{q}$, 0.6%
Z-recoil, 11.5%
ZZ, 6.9%
WW, 0.3%



entries
10^6
10^5
10^4
10^3
10^2
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
y23
$b\bar{b}$
$q\bar{q}$, 0.5%
Z-recoil, 6.0%
ZZ, 4.8%
WW, 0.2%
HZ, 1.1%