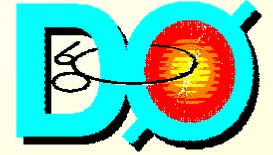


BID & Thumbnail Tree Status

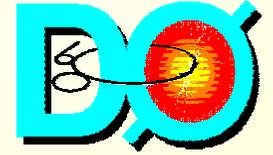


Overview

- Design
- Coding
- Testing
- Efficiency



Design



Goals:

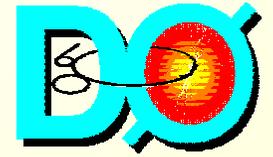
- Allow for custom taggers
- Efficient in storage space
- Easy to use

Non-standard design:

- TMBBCJetsMaker is a regular maker, TMBBCTaggersMarker is not (just adds data to jet)
- Only one branch for bc jet id containing all information



Coding



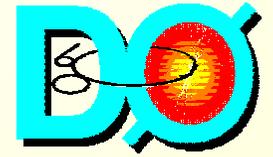
Framework, BID's jets and taggers fully implemented

TMB tree's dynamic library needs to have customization options for ROOT I/O (linkdef.h), work in progress

Code sent to Elemer – could be put into cvs



Testing

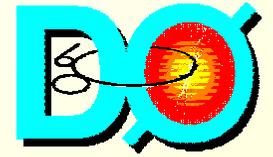


Draft version!

It works: transferred 500 events of tagging data from reco into thumbnail tree containing all taggers

Output from `tmb_tree.root` not okay – needs to be debugged

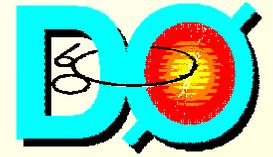
Efficiency



Need to optimize storage space used by taggers (500 events with all 7 tags use 28.5kB = approx 57 Bytes / event)

Data storage of tagging information handled by base class TMBBCTagger, so change in storage layout invisible for users

Customization



Too add your custom tagger:

- Derive it from `TMBBCTagger`
- Fill it using information available in the TMB tree
- Say `jet.AddTagger(myCustomTagger)`
- Save the new tree
- Read your tagger back like all the others