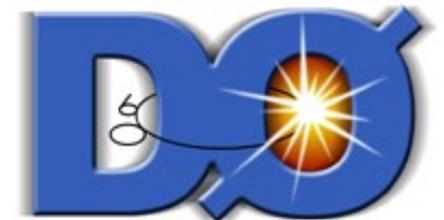


b-ID Meeting  
February 17, 2006

## BTagJetSelector

- ▶ Running b-tagging in CAFE: BTagProcessor
- ▶ Selecting tagged events: BTagJetSelector
  - Cut events based on b-tagged jets
  - Save b-tagged jets in a branch
  - Example config file
  - Example output
- ▶ Other issues



# BTagProcessor

- ▶ Current CAF Trees do not have b-tag information
- ▶ The user has to re-run tagging and store the information as a new branch
- ▶ B-ID group provides a **BTagProcessor**, inside `btags_cert_caf`, to do just that
- ▶ Uses the CAF `write()` processor to (re)fill the `TMBBTag` branches in a CAF Tree with correct information
- ▶ The `TMBJet` is modified to store a map of references (`TRef`) to the b-tag branch
- ▶ Nice feature: use the jet to access b-tag information
- ▶ See Chad's talk from Jan 6<sup>th</sup> for more details

# Example config file

```
cafe.Packages: d0root_analysis d0root_btag d0root_jlip d0root_slt  
d0root_nnbttag bc_csiptagger d0root_csip d0root_tmbtree btags_cert  
d0root_example_caf
```

```
cafe.Run:          CafeReadEventProc(event)  
+cafe.Run:        BTagProcessor(tagger) Write(myfile)
```

```
cafe.Input:       listfile:input.list
```

```
# Convert to d0root:  
event.JetBranch: JCCB  
event.JetCorr:   corr
```

```
# Tagger parameters:  
tagger.BTagAlgo: NN  
tagger.Cut:      TIGHT  
tagger.Features: DATA APV
```

```
myfile.Disable:  TrackCal TrackCalJet_JCCA TrackCalJet_JCCB  
myfile.File:    myfile_SVT.root  
myfile.Tree:    TMBTree
```

# BTagJetSelector

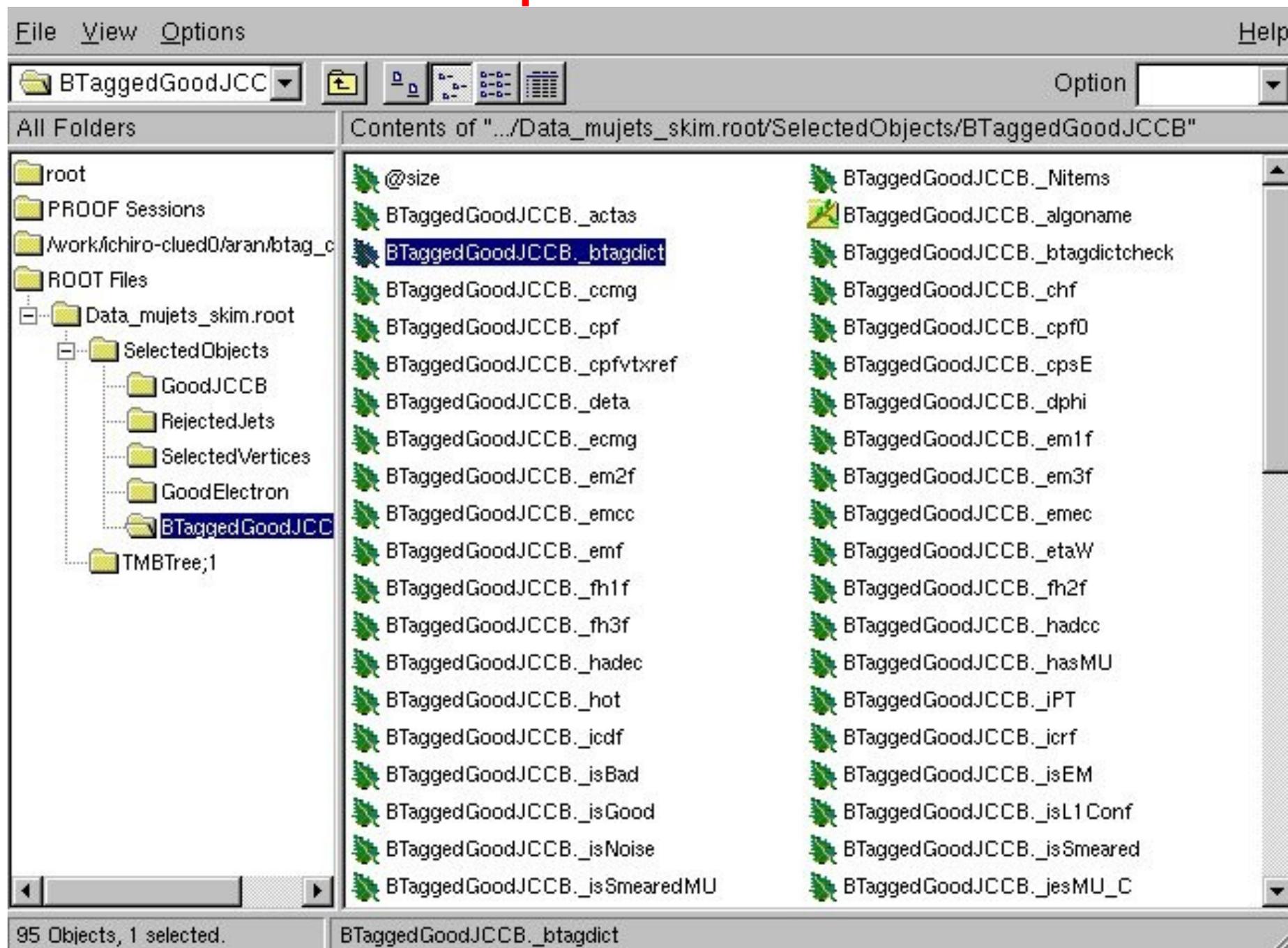
- ▶ Based on the existing `JetSelector` in `caf_util`
- ▶ It simply cuts on the number of b-tagged jets: its input is a jet branch, like `JCCB`
- ▶ Needs a branch derived from the `TMBBTag` class (obtained by running `BTagProcessor`)
- ▶ It will output the same `JCCB` branch to a skim file with only those jets that pass selection
- ▶ Later processors can access this branch in memory
- ▶ Makes full use of the `stat` information
- ▶ It is now released in **`caf_util v02-03-00!`**

# Example config file

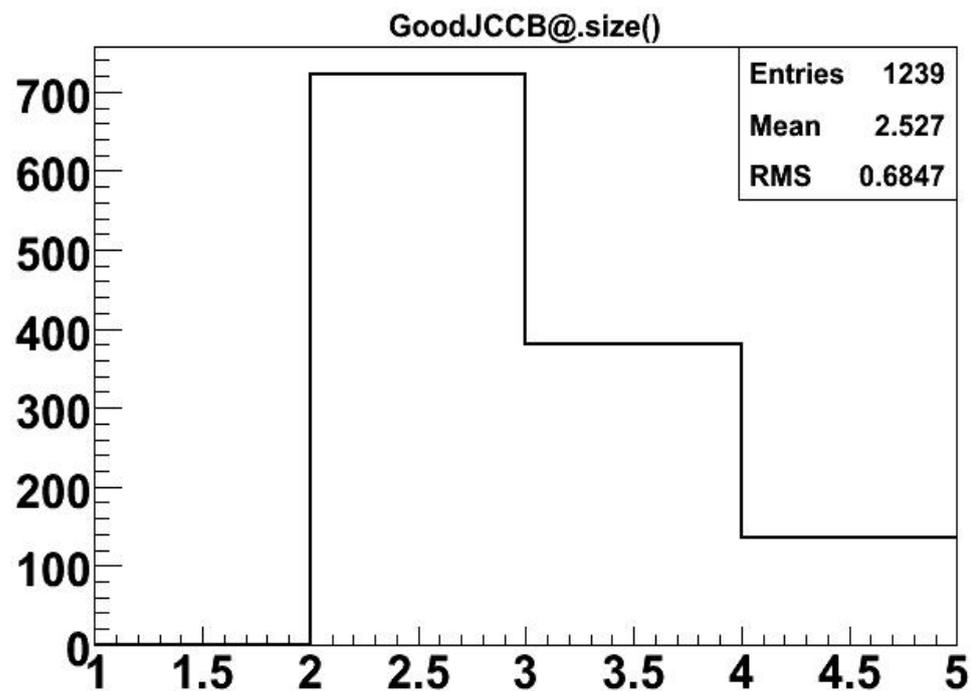
```
cafe.Packages:  caf_util
...
cafe.Run:       Stat(eff)
+cafe.Run:     JetSelector(jet)
+cafe.Run:     CafeReadEventProc(d0rootevent) BTagProcessor(tagger)
+cafe.Run:     BTagJetSelector(bjet)
+cafe.Run:     Write(selected) Write(skim) Passed
...
##### Stat Output
eff.Output:    btag_eff
eff.Sample:    DATA
##### B-TAG Selector
bjet.From:    GoodJCCB
bjet.To:      BTaggedGoodJCCB
bjet.Tree:    SelectedObjects
bjet.BTagAlgo: SVT      #
bjet.Cut:     TIGHT    #

bjet.nBJets:  1  # nBJets applies a >= cut
bjet.nBJetsMax: 1  # nBjetsMax applies a <= cut
...
##### Skim File
skim.File:    btag_skim.root
selected.File: btag_skim.root
selected.Tree: SelectedObjects
```

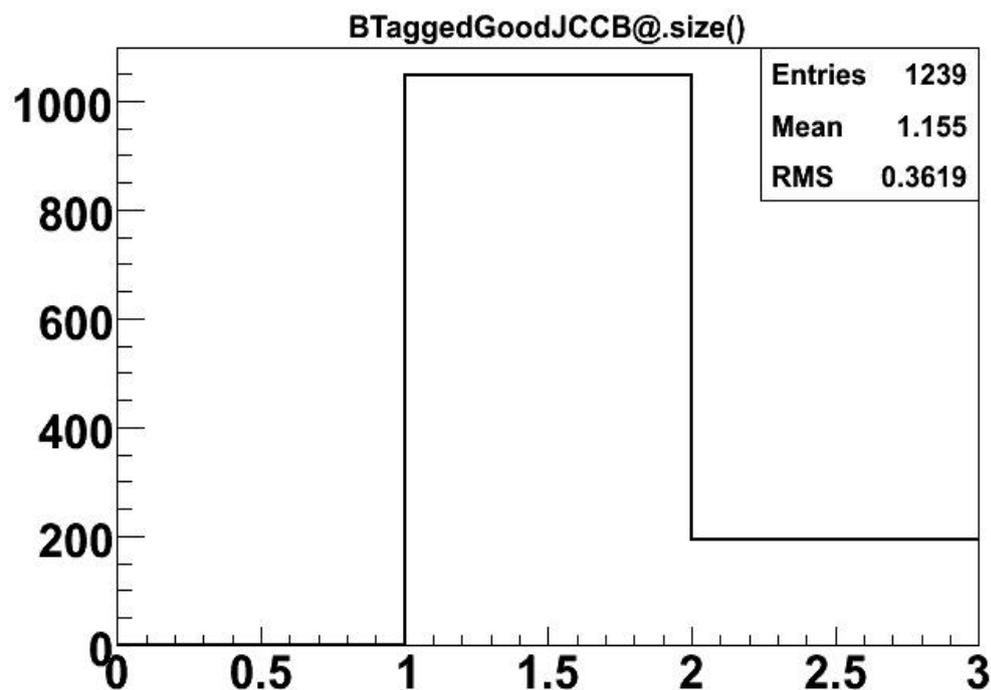
# Example skim file



# Example selected jets



JetSelector: 2-4,  $p_T > 15, \dots$



BTagJetSelector: 1-2, SVT Tight

# Example Stat output

SELECTION	DATA		
Initial	9272		
PROCESSOR "jet"	9272	100.000±0.000%	100.000±0.000%
Jet ID	8659	93.389±0.258%	93.389±0.258%
EM jets removal	8619	99.538±0.073%	92.957±0.266%
Jet pT >= 15 GeV	6973	80.903±0.423%	75.205±0.448%
Number of jets >= 1	6973	100.000±0.000%	75.205±0.448%
Number of jets >= 2	4665	66.901±0.564%	50.313±0.519%
Number of jets <= 4	4512	96.720±0.261%	48.663±0.519%
PROCESSOR "bjet"	4512	100.000±0.000%	48.663±0.519%
Taggable Jet	3992	88.475±0.475%	43.054±0.514%
SVT TIGHT Tagged Jet	367	9.193±0.457%	3.958±0.202%
Number of b-jets >= 1	367	100.000±0.000%	3.958±0.202%
Number of b-jets <= 1	344	93.733±1.265%	3.710±0.196%

# Some features

- ▶ Need to save both the jet branch and the TMBBTag branch if you want to access extra b-tag info later
- ▶ If you want to use **asymmetric tagging** (e.g.: **1 tight jet AND 1 loose jet**), you can concatenate two BTagJetSelectors. First the loose (requiring two b-tags), and then the tight (one b-tag)
- ▶ To require **1 tight jet and 0 loose (for all other jets)**, you can concatenate again: first the loosest (requiring one b-tag) and then the tight (one b-tag), such that the loose is the tight
- ▶ Should we include any special treatment for MC? How do physics groups want to treat MC b-tagged jets?
- ▶ Documentation with examples is in the [b-ID wiki](#), and in the header file: [BTagJetSelector.hpp](#)
- ▶ Currently working on a class to organize the information of all jets in the event, e.g.: return the SecondUnTaggedJet