Mark Sutton
Tania McMahon
Ricardo Gonçalo

The PESAsim analysis framework

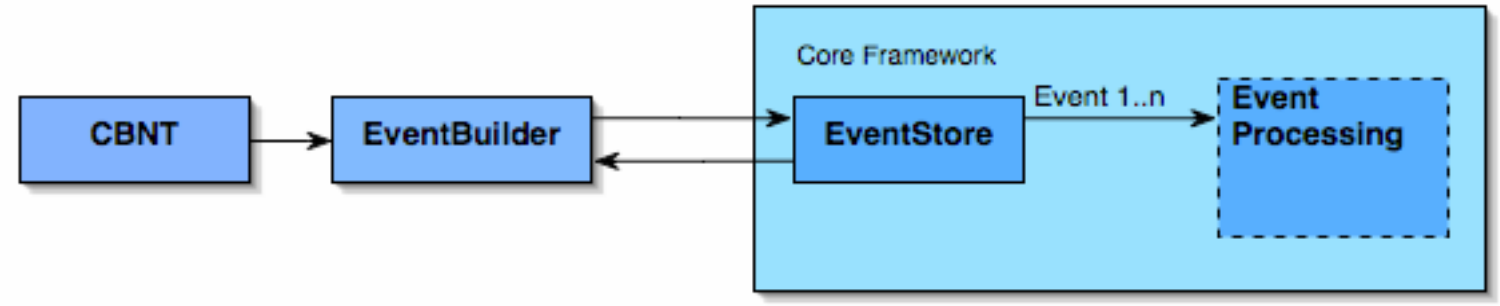
- What it is
- How it works
- What it can do
- How to get it and use it

What it is

- PESAsim is a framework for:
  - fast simulation of event selection algorithms
  - fast prototyping of algorithms
  - algorithm validation
  - automatic, iterative tuning strategies.
- Internally, the event data structure closely resembles that used online in the trigger to enable fast transfer of algorithms without too much recoding.
- The framework itself, simulates the behaviour of the online steering, and can be used for complete trigger menus, and multiple object triggers

How it works: code structure

- EventStore



  - Responsible for accessing the persistent storage (i.e. CBNT) and building each FrameworkEvent
  - Factorises event storage from event processing.
  - Can be used to read events sequentially from the file for processing, or buffer all events in memory for speed (automated tuning strategies).

Trigger classes

- Data objects grouped by RoI
- LVL1, LVL2 and EventFilter
  - Each contains a collection of TrigSignatures, each with it's own prescale.
  - The output of all TrigSignatures are OR'ed together to get overall decision at each trigger level.
  - Does their own book keeping, number of events passed, failed etc.
  - Monitors event correlations between TrigSignatures, overlap of numbers of events of each TrigSignature with all others etc.

TrigSignature and TrigSequence

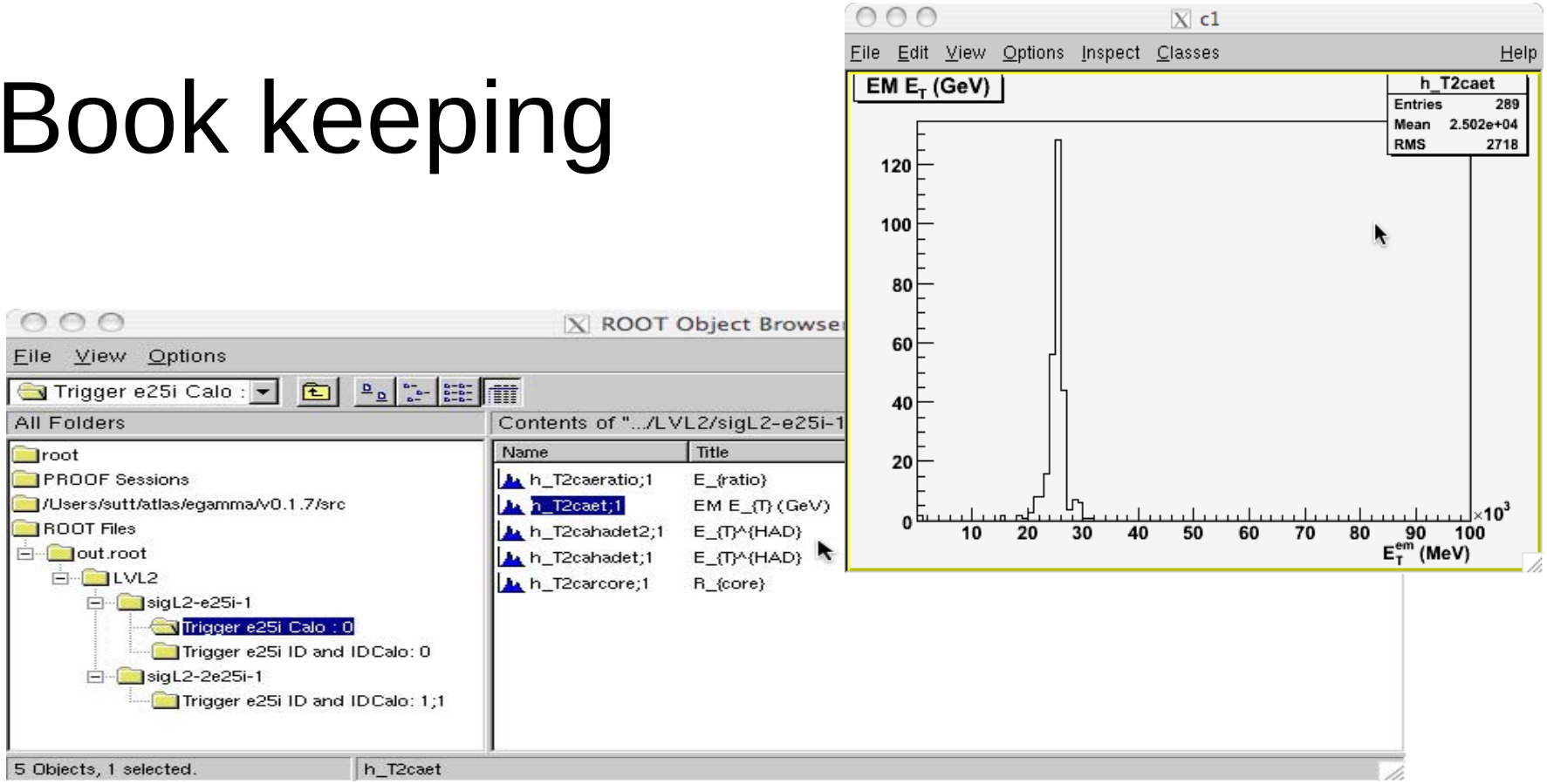
- TrigSignature
  - Contains set of TrigSequences (corresponding to HLTAlgos in Athena…should probably change class names to make it more transparent).
  - All book keeping (number of events passed, before/after presecale etc) done automatically by the TrigSignature.
  - Users write their own derived signature classes or use standard ones with new data

- TrigSequence
  - Main part of user intervention is on writing his/her own TrigSequence derived classes
  - Run on RoIs
  - Facilities for automatic book keeping in place
    - Counting of number of events passed/failed per event or RoI automatically done in base class
    - Directories in ROOT file for histogram booking created automatically

Some additional features

- TruthEvent
  - Accessible through a method of the trigger elements: `TrigElement::truth()`
- Level 2 and Event Filter track/truth association:
  - Encoded in classes `L2IDTrackTruth` and `EFTrackParticleTuth`
- PreFilter :
  - Selects events to be put into EventStore
- Has a set of PreSelections:
  - Similar to TrigSequences
  - Run on full FrameworkEvent instead of TrigElement/RoI
  - PreFilter accepts AND of PreSelections (instead of OR as in TrigLevel)
- TrigLevel Forced Accept: take a certain fraction of events (from 0 to 1)
- Print sequence efficiency per RoI and per event at end of run

Book keeping



- Example ROOT file:
  - Directories created automatically for each sequence to have its histograms
  - Histograms booked and filled in user TrigSequence derived classes

What it can do

- get the framework from:

https://uimon.cern.ch/twiki/bin/view/Main/PESAsim

Demonstration

- Get the framework from:

https://uimon.cern.ch/twiki/bin/view/Main/PESAsim

- Running with one signature
- Making it a double-object signature
- Applying preselection cuts
- Optimizing trigger cuts
- Building a menu: Z->ee

How to find out more

- Wiki page at

https://uimon.cern.ch/twiki/bin/view/Main/PESAsim

- In CVS, temporarily at
  `Trigger/TrigAnalysis/TrigEgammaAnalysis/PESAsim`
- Writeup in preparation (linked from Wiki)
- Feedback very welcome!