lm_access package

tool for accessing luminosity information in physics analysis of DØ data

Luminosity ID Meeting: 25^{th} Oct 2001

Alexander Kupčo

Institute of Physics, Academy of Sciences of the Czech Republic
Center for Particle Physics, Prague

Authors

- package is provided by Luminosity ID group (Michael Begel, John Hauptman)
- contribution to the code by several people
  - Alexander Kupčo (lm_access APIs)
  - Marco Verzocchi (L1 and L3 trigger map)
  - Maciej Przybycien and Heidi Schellman (LBN intervals in the data file)
  - Stephen White and Carmenita Moore (SAM code installations)

Content

- ▷ *basic concepts*
- ▷ L1 *and* L3 *trigger bit map*
- ▷ LBN *intervals in the data files*
- ▷ `lm_access` *package* APIs
- ▷ *tutorial example*
- ▷ *example of physics analysis result (jet rates)*
- ▷ *future methodology*
- ▷ *summary*

Basic concepts

- **luminosity block**
  - the smallest unit of time (about 60 seconds) in which the information about luminosity is stored
- luminosity ID group provides information about integrated luminosity per L1 trigger bit number per luminosity block
- status (based on luminosity detector status) is assigned to each luminosity block
- luminosity is provided only for good luminosity blocks right now
- each good luminosity block has its own text file that contains the L1 luminosity per trigger
- these files are located on d0mino in the directory

  /prj_root/829/c/com/begel/stage1a

- in near future, the information will be stored in luminosity database server

Basic concepts - consequences

- large fraction of the luminosity block numbers does not have the good status
- moreover, the efficiency (or the dead time) varies rapidly from luminosity block to luminosity block
- there is no way to estimate the luminosity in the bad luminosity blocks
- any serious cross section analysis must be done using the data only from good luminosity blocks
- getting luminosity per L1 trigger bit number for given run is useless
- getting luminosity per L1 trigger bit number for given file is useless

L1 and L3 trigger bit map

- L1 and L3 trigger bits are assigned by system dynamically and can change from run to run
- in physics analysis one has to know the L3 trigger bit number for the selected trigger name (like JT_HI) and corresponding L1 trigger bit number
- the information can be obtained from Run database
- Marco Verzocchi wrote script `makeL1L3TrigMap.py` that downloads the L1 and L3 trigger map from the database and store the output in text file
- the format is the following

  132417 JT_HI 13 13 CJT40 14 14 #

- run map file for all runs after August 25 in file

  /prj_root/829/c/com/begel/stage1a/L1L3TrigMap.txt

LBN intervals in the data files

- you have to now to which luminosity block intervals your data files correspond to
- information is available on SAM
- Heidi wrote scripts that can get the information from SAM

```
> getFileInfo.csh <filename>
> getManyFileInfo.csh <listoffiles>
> getManyFileInfoSAM.csh <defname>
```

- these scripts are quite new an they are not included into the package yet
- for specified root-tuple all the raw data parents are found and corresponding luminosity block intervals are listed

```
recoA_reco_all_0000133014_mrg_001-006.raw_p10.07.01.root
root 12246 12246 12246 < 133014 :  596386 - 596396
133014 :  596397 - 596401 133014 :  596402 - 596406
133014 :  596407 - 596411 133014 :  596413 - 596429
133014 :  596430 - 596431 >
```

- What to do if number of reconstructed events is not the same as number of stored events?

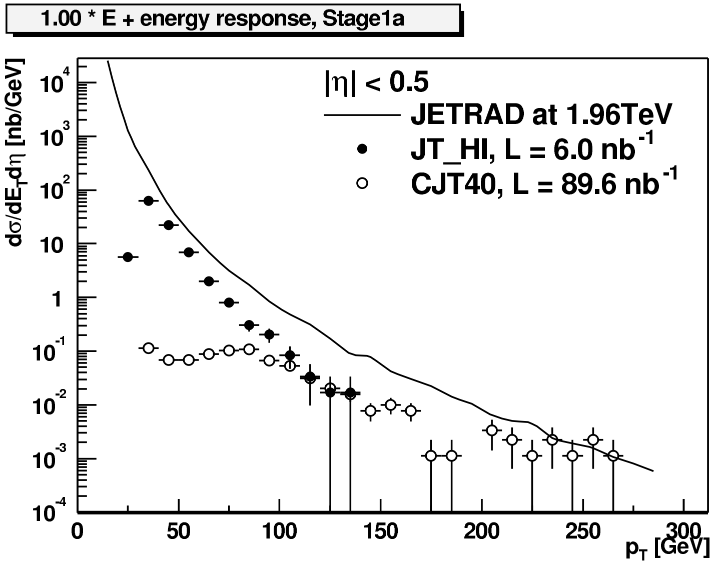
lm_access APIs

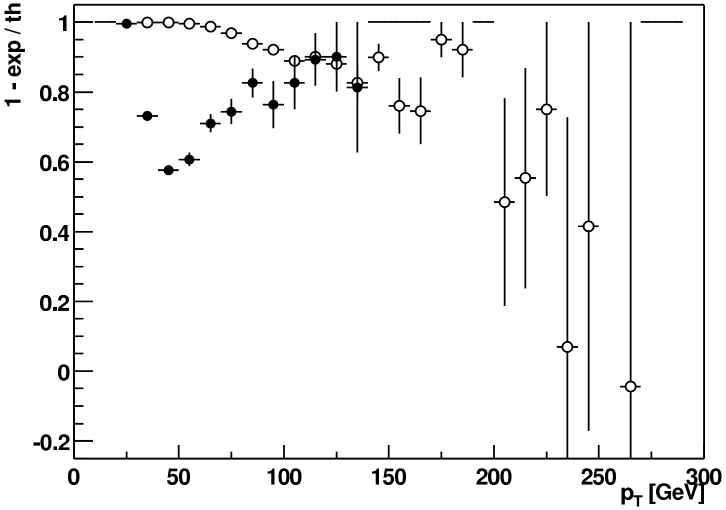
- *all documentation is available in* `lm_access/doc` subdirectory
- class LumBlkInfo
- class LumBlkMap
- type LumBlkStatus
- class RunInfo
- class RunMap

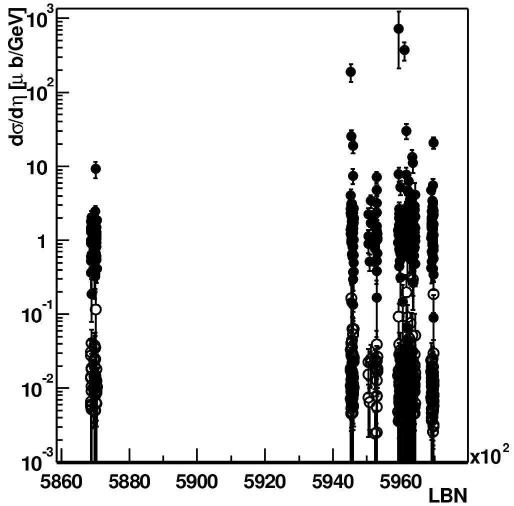
Tutorial example

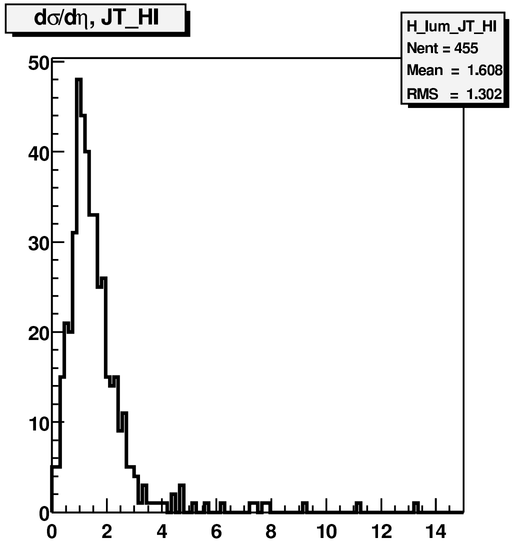
- cross section calculation for `JT_HI` trigger
- source code available in `lm_access/example` subdirectory under the name `ex_xsec1.cpp`
- the example is decribed in file `lm_access/doc/example.html`

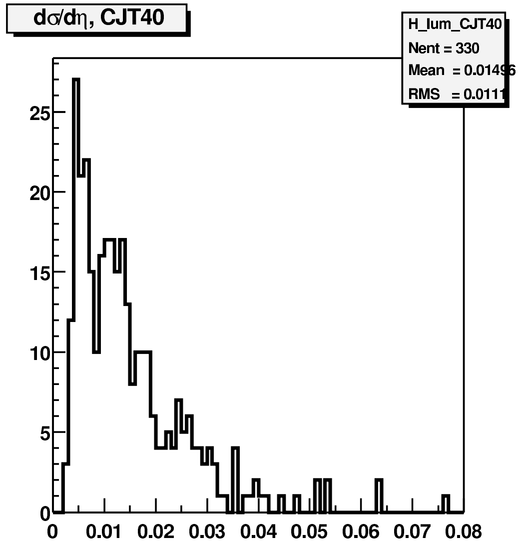
jet inclusive pT spectrum, p10.07.01 - 19Oct-21:31











Future methodology

- right now the data analysis has to be performed in few steps
  - getting the data files from SAM
  - getting the proper L1 and L3 trigger bit map
  - getting the LBN intervals that correspond to the data
  - writing `C++` code that does the analysis
- in future
  - keep the current possibility still available
  - allow user to do the cross section analysis on his SAM project
- question of luminosity provided per L3 trigger

Summary

- ▷ almost all pieces that are necessary for reasonably easy access to luminosity information in physics analysis are implemented in `lm_access` package
- ▷ hopefully, the package is designed in such a way that it will allow to work with streamed data and that it will allow to run the analysis on SAM projects in future
- ▷ due to large fraction of bad luminosity blocks and due to unstable detector efficiency the L1 luminosity per file (or run) cannot be provided right now