Implementation of 3x1x1 detector in LArSoft

Kevin Fusshoeller
HEP Masters - ETH Zurich/Université Paris-Saclay
Science Board Meeting, June 7th 2017

Outline

1. Introduction.
2. Importing Data from 3x1x1 to LArSoft.
3. The 3x1x1 Geometry in LArSoft.
4. Example: Imported pulsing data + crosscheck with QScan

1. Introduction

Status and Goals

First steps:

- Geometry of the 3x1x1 detector is implemented in LArSoft
  - find .fcl‘s for sim and reco in dunetpc/fcl/3x1x1dp/
  - and .fcl for event display in dunetpc/dune/Utilities/evd_3x1x1dp.fcl
- It is possible to import single events from 3x1x1 raw data.

Goal: Take the data from the 3x1x1 and be able to use LArSoft for noise and pulsing analysis.

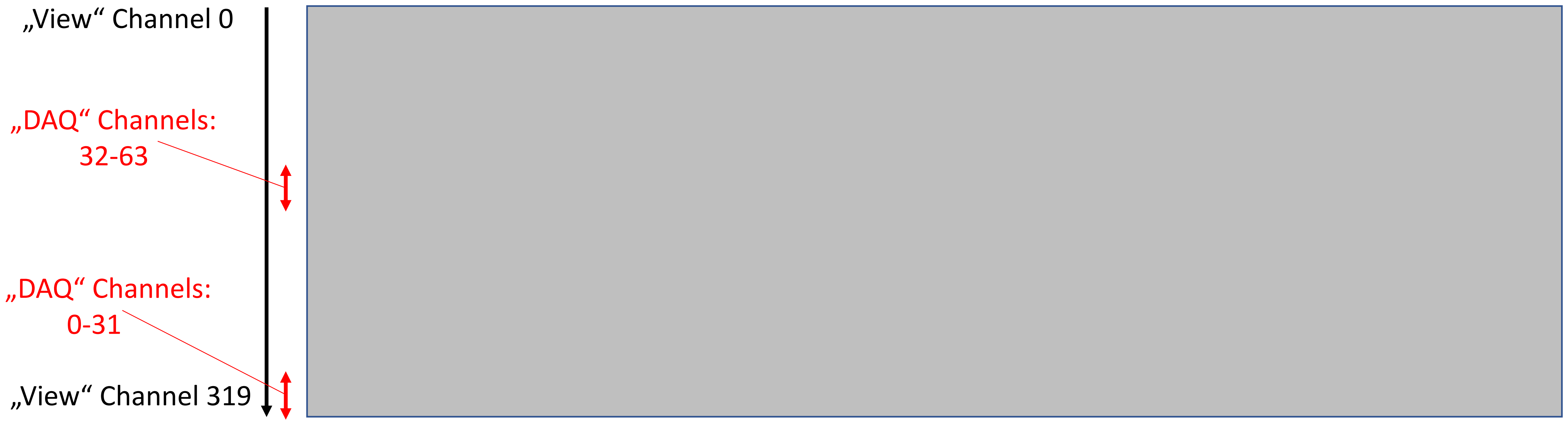
Tasks: → Import full raw data files from 3x1x1 to LArSoft.

2. Importing data from 3x1x1 to LArSoft

Data Import from 3x1x1 up till now.

Until now Qscan/WA105Soft is used to analyze the 3x1x1 data.
Now we wish to do the same in LArSoft.

Problems: -) Data structure of raw data has to be adapted to LArSoft format.
-) „Daq Channel“ is not the same as the „view Channel“. → mapping



Modus Operandi:

Create an empty event (source) and use a new module to fill it with data.

Steps: -) Read in data (Using Slavic‘s code from QScan).

-) For each „view channel“:

-) Find the corresponding „daq Channel“.

-) Extract the ADC counts for that channel.

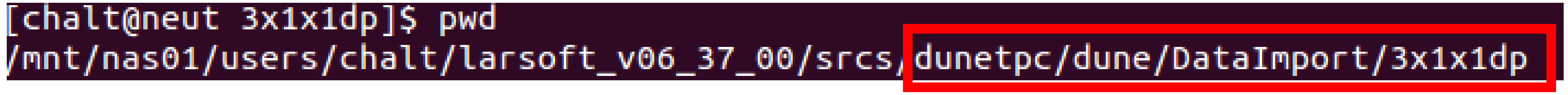
-) Create a raw::Digit and store it in the art::event.

But: only single event can be read in.

Data import from 3x1x1

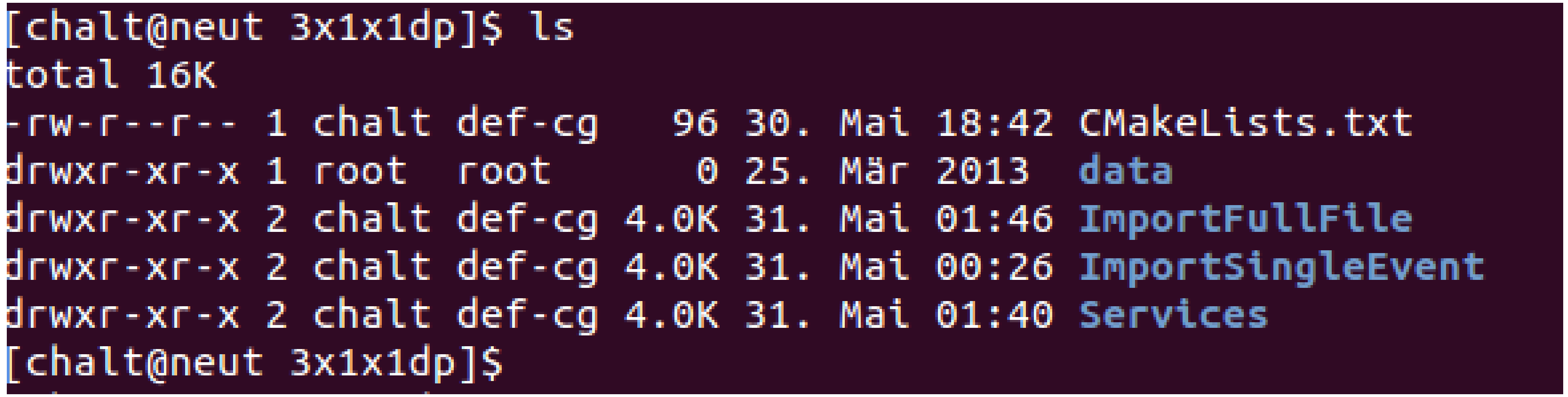
Update: Create a new source, which reads in and stores all the events in a file. The different steps are the same as before, but now repeated for each event.

Where to find the code? (pushed by Christoph)



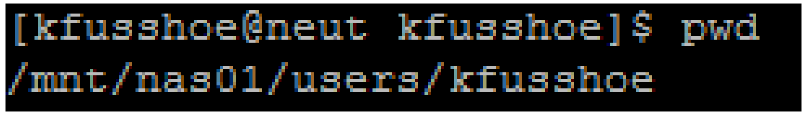
```
[chalt@neut 3x1x1dp]$ pwd
/mnt/nas01/users/chalt/larsoft_v06_37_00/srcs/dunetpc/dune/DataImport/3x1x1dp
```

What do you find there?



```
[chalt@neut 3x1x1dp]$ ls
total 16K
-rw-r--r-- 1 chalt def-cg   96 30. Mai 18:42 CMakeLists.txt
drwxr-xr-x 1 root  root      0 25. Mär 2013  data
drwxr-xr-x 2 chalt def-cg 4.0K 31. Mai 01:46 ImportFullFile
drwxr-xr-x 2 chalt def-cg 4.0K 31. Mai 00:26 ImportSingleEvent
drwxr-xr-x 2 chalt def-cg 4.0K 31. Mai 01:40 Services
[chalt@neut 3x1x1dp]$
```

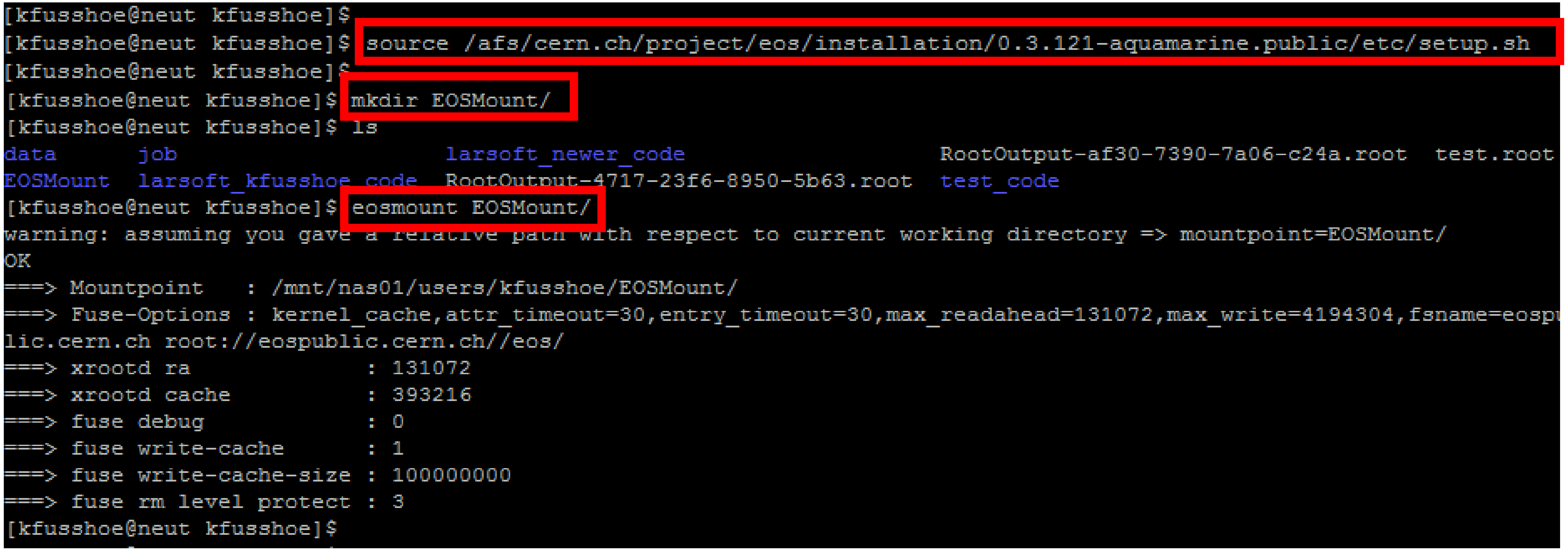
How to import data

1. Mount eos on neutrino platform to get access to the data.
   - Path: 

```
[kfusshoe@neut kfusshoe]$ pwd
/mnt/nas01/users/kfusshoe
```

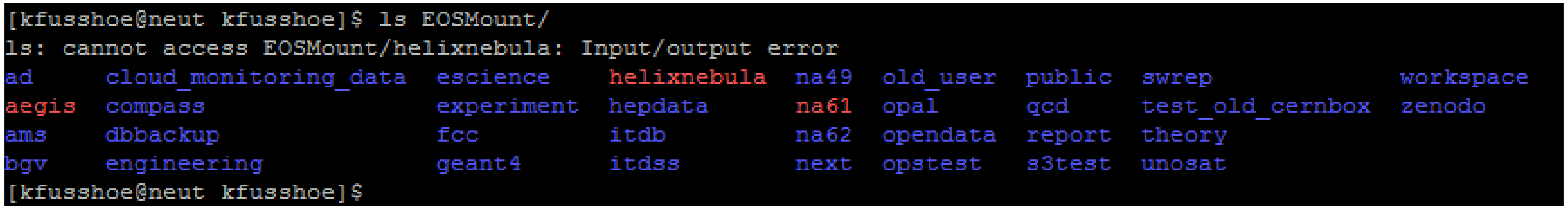
     NOT in your larsoft folder!

   - Commands:



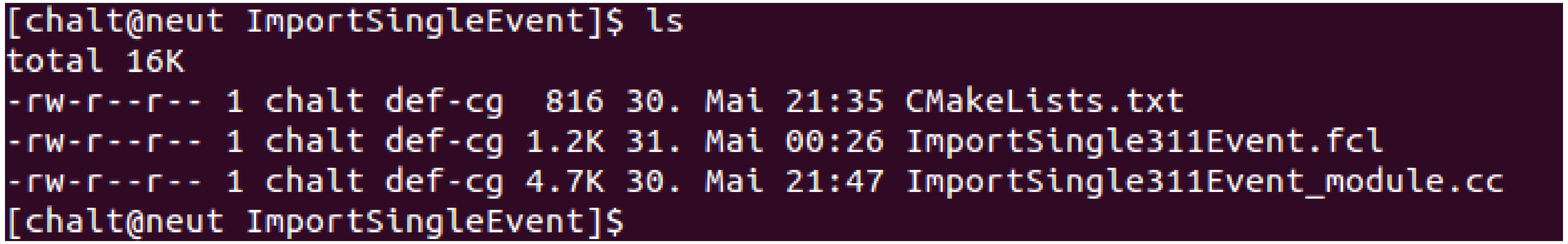
```
[kfusshoe@neut kfusshoe]$
[kfusshoe@neut kfusshoe]$ source /afs/cern.ch/project/eos/installation/0.3.121-aquamarine.public/etc/setup.sh
[kfusshoe@neut kfusshoe]$
[kfusshoe@neut kfusshoe]$ mkdir EOSMount/
[kfusshoe@neut kfusshoe]$ ls
data      job                  larsoft_newer_code                 RootOutput-af30-7390-7a06-c24a.root  test.root
EOSMount  larsoft_kfusshoe_code  RootOutput-4717-23f6-8950-5b63.root  test_code
[kfusshoe@neut kfusshoe]$ eosmount EOSMount/
warning: assuming you gave a relative path with respect to current working directory => mountpoint=EOSMount/
OK
===> Mountpoint   : /mnt/nas01/users/kfusshoe/EOSMount/
===> Fuse-Options : kernel_cache,attr_timeout=30,entry_timeout=30,max_readahead=131072,max_write=4194304,fsname=eospu
lic.cern.ch root://eospublic.cern.ch//eos/
===> xrootd ra             : 131072
===> xrootd cache          : 393216
===> fuse debug            : 0
===> fuse write-cache      : 1
===> fuse write-cache-size : 100000000
===> fuse rm level protect : 3
[kfusshoe@neut kfusshoe]$
```

How to import data



```
[kfusshoe@neut kfusshoe]$ ls EOSMount/
ls: cannot access EOSMount/helixnebula: Input/output error
ad      cloud_monitoring_data  escience    helixnebula  na49  old_user  public  swrep             workspace
aegis   compass                experiment  hepdata      na61  opal      qcd     test_old_cernbox  zenodo
ams     dbbackup               fcc         itdb         na62  opendata  report  theory
bgv     engineering            geant4      itdss        next  opstest   s3test  unosat
[kfusshoe@neut kfusshoe]$
```

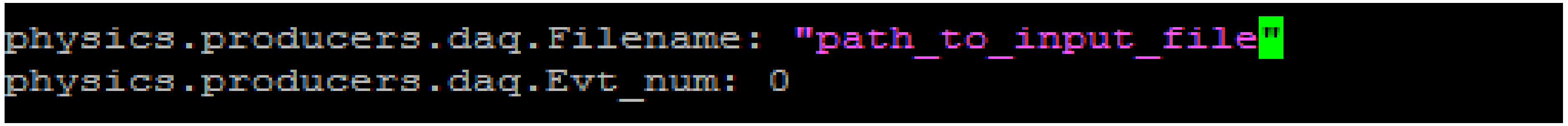
2. For single events:



```
[chalt@neut ImportSingleEvent]$ ls
total 16K
-rw-r--r-- 1 chalt def-cg  816 30. Mai 21:35 CMakeLists.txt
-rw-r--r-- 1 chalt def-cg 1.2K 31. Mai 00:26 ImportSingle311Event.fcl
-rw-r--r-- 1 chalt def-cg 4.7K 30. Mai 21:47 ImportSingle311Event_module.cc
[chalt@neut ImportSingleEvent]$
```

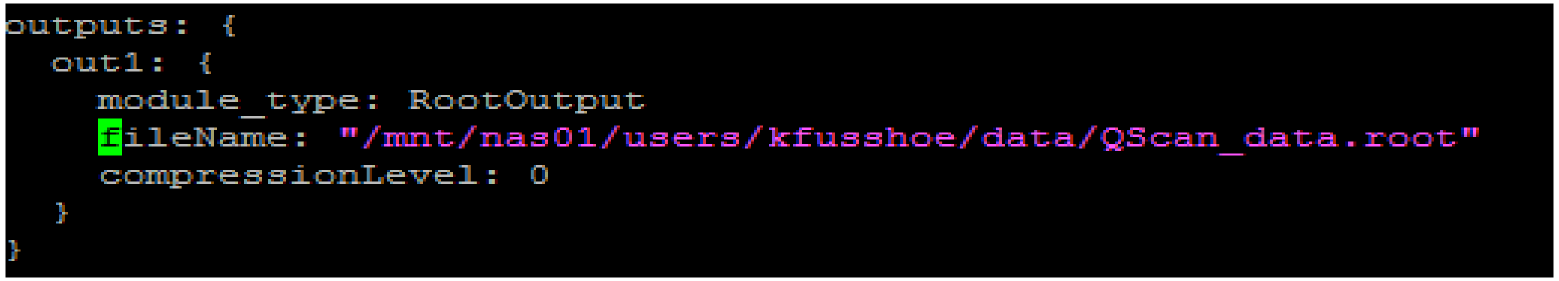
Inside ImportSingle311Event.fcl:

- change lines 43 and 44 to give your input file and event:



```
physics.producers.daq.Filename: "path_to_input_file"
physics.producers.daq.Evt_num: 0
```

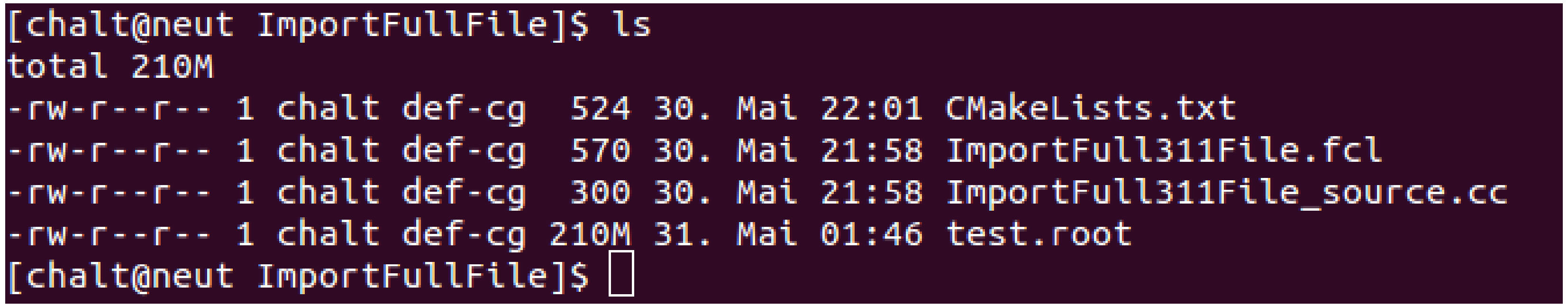
- Change line 38 to give your output file.



```
outputs: {
  out1: {
    module_type: RootOutput
    fileName: "/mnt/nas01/users/kfusshoe/data/QScan_data.root"
    compressionLevel: 0
  }
}
```

How to import data

3. For full files



```
[chalt@neut ImportFullFile]$ ls
total 210M
-rw-r--r-- 1 chalt def-cg  524 30. Mai 22:01 CMakeLists.txt
-rw-r--r-- 1 chalt def-cg  570 30. Mai 21:58 ImportFull311File.fcl
-rw-r--r-- 1 chalt def-cg  300 30. Mai 21:58 ImportFull311File_source.cc
-rw-r--r-- 1 chalt def-cg 210M 31. Mai 01:46 test.root
[chalt@neut ImportFullFile]$ 
```

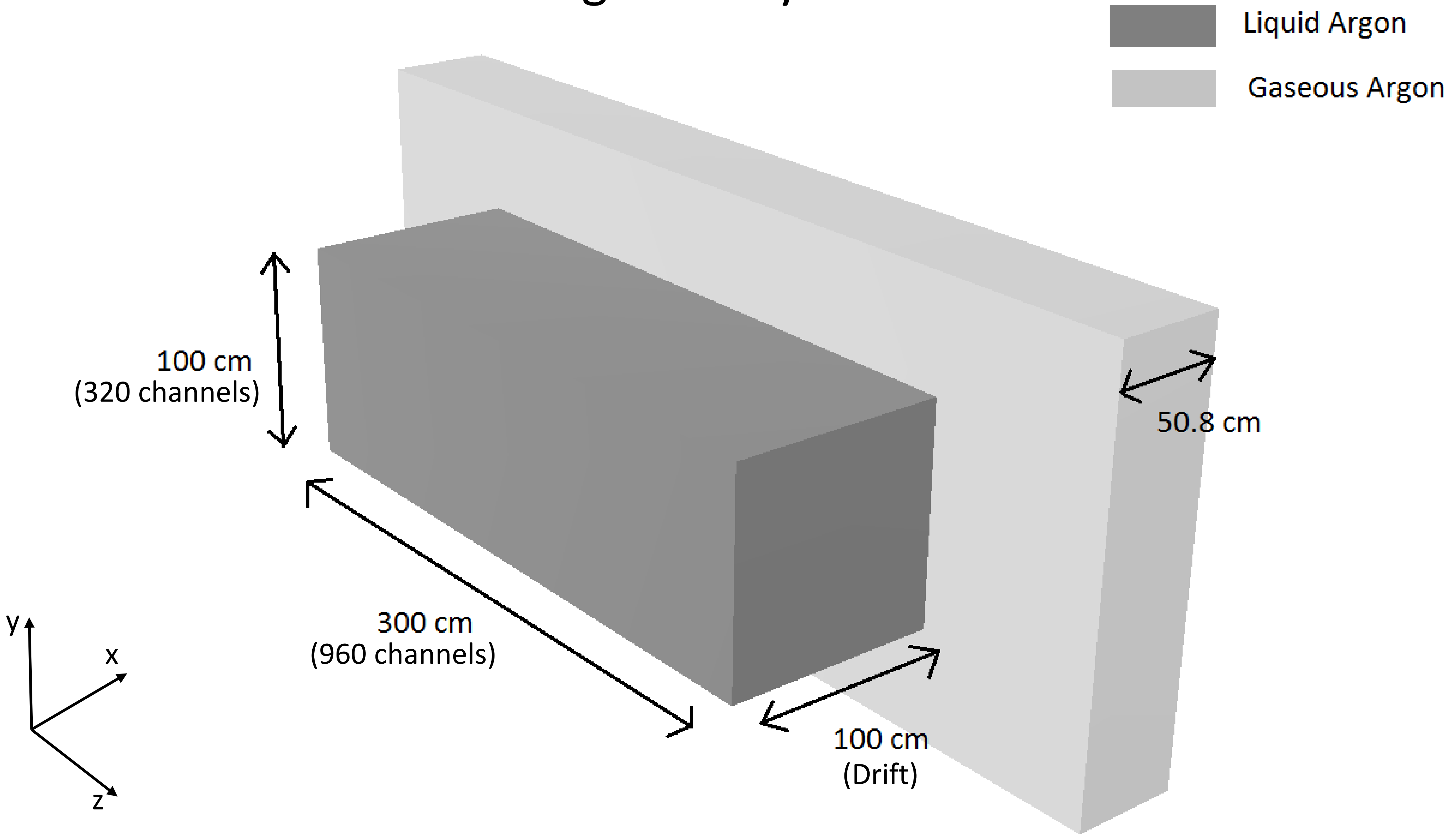
Type the command: `lar -c ImportFull311File.fcl <path_to_input_file> -o <output_file>`

or in ImportFull311File.fcl:

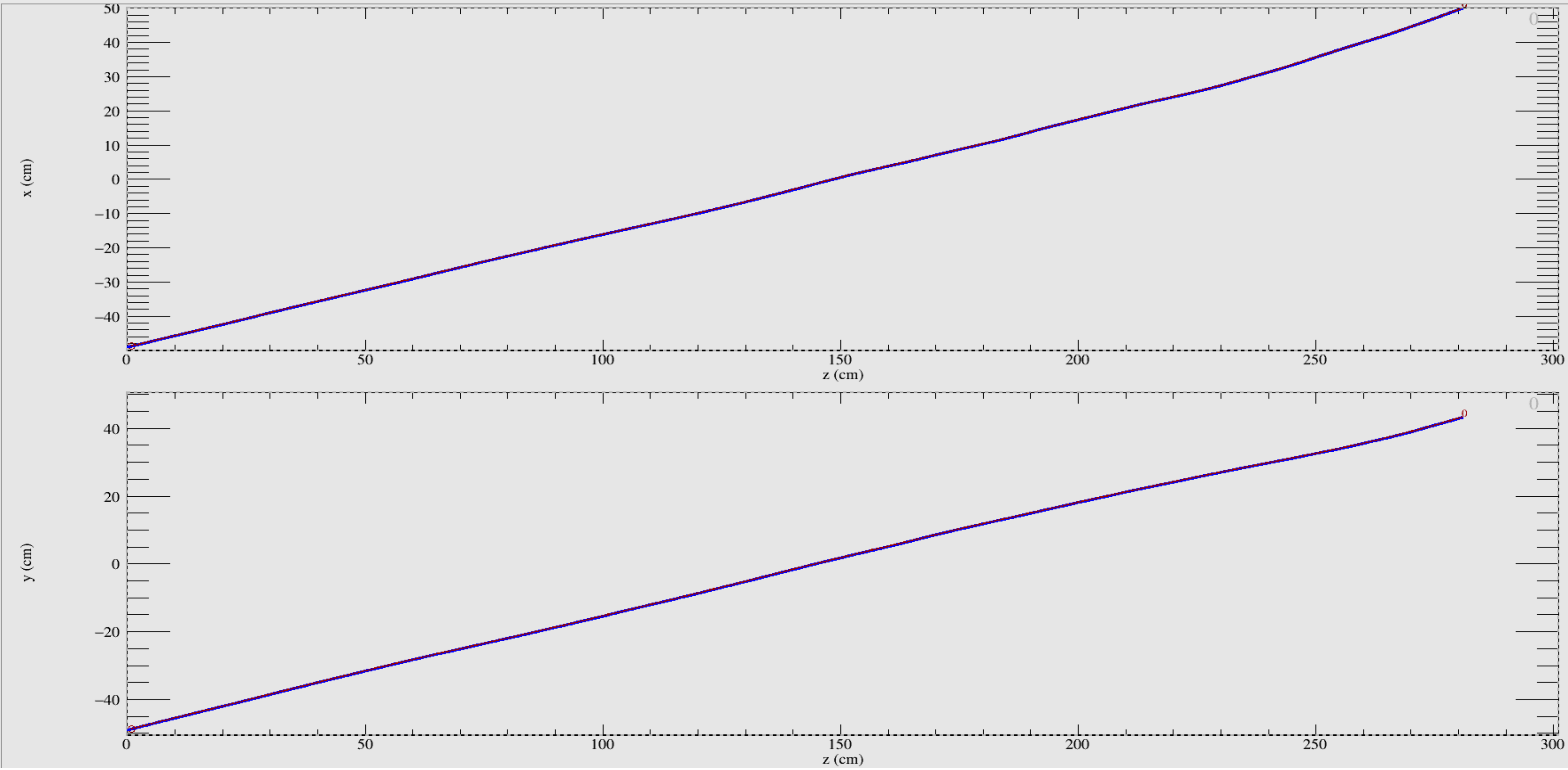
- change line 13 for the input path and line 23 for the output filename.

3. The 3x1x1 geometry in LArSoft

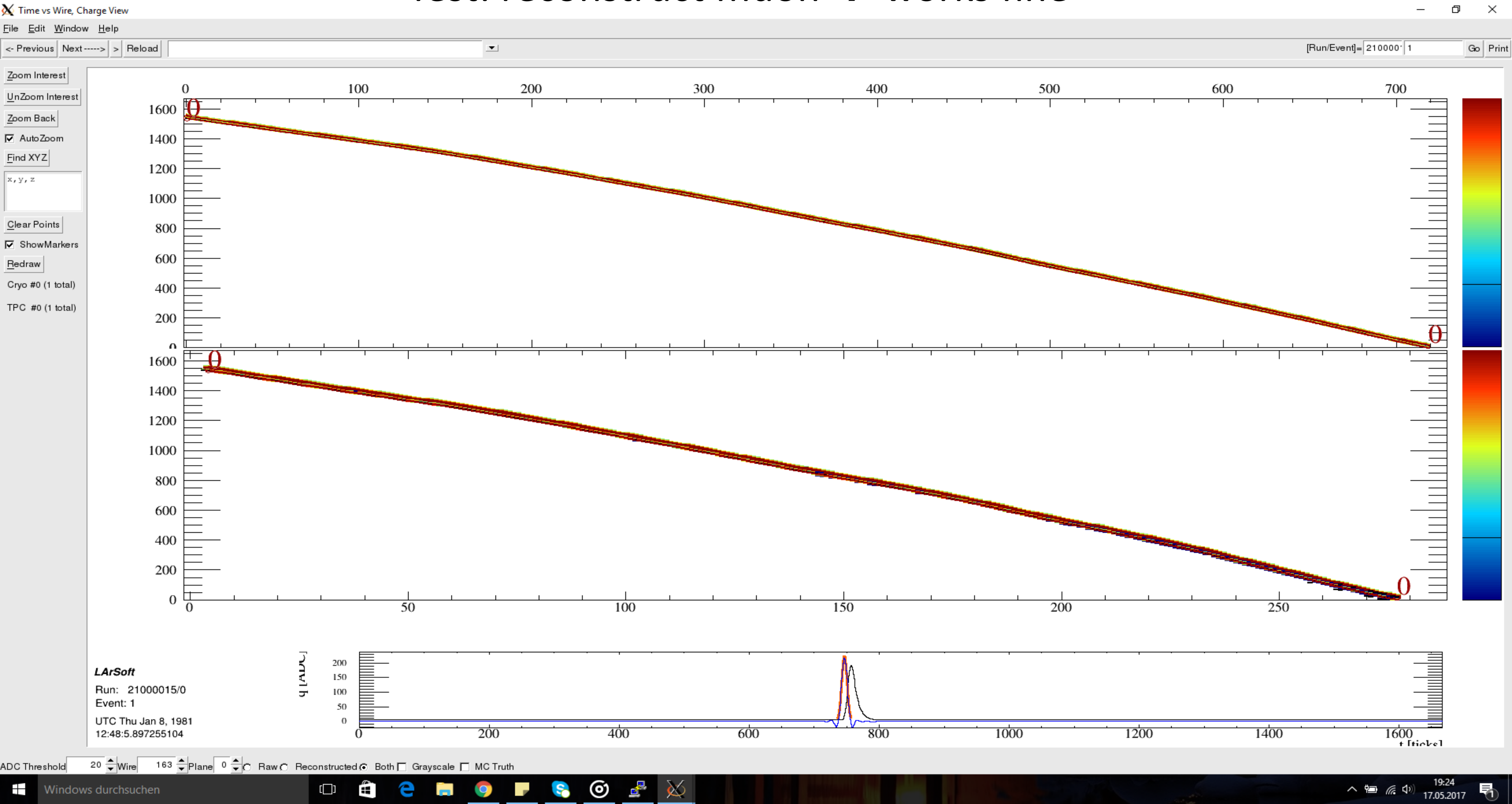
3x1x1 geometry



Test: simulate a muon → geometry accepted



Test: reconstruct muon → works fine



[Run/Event]= 210000 1

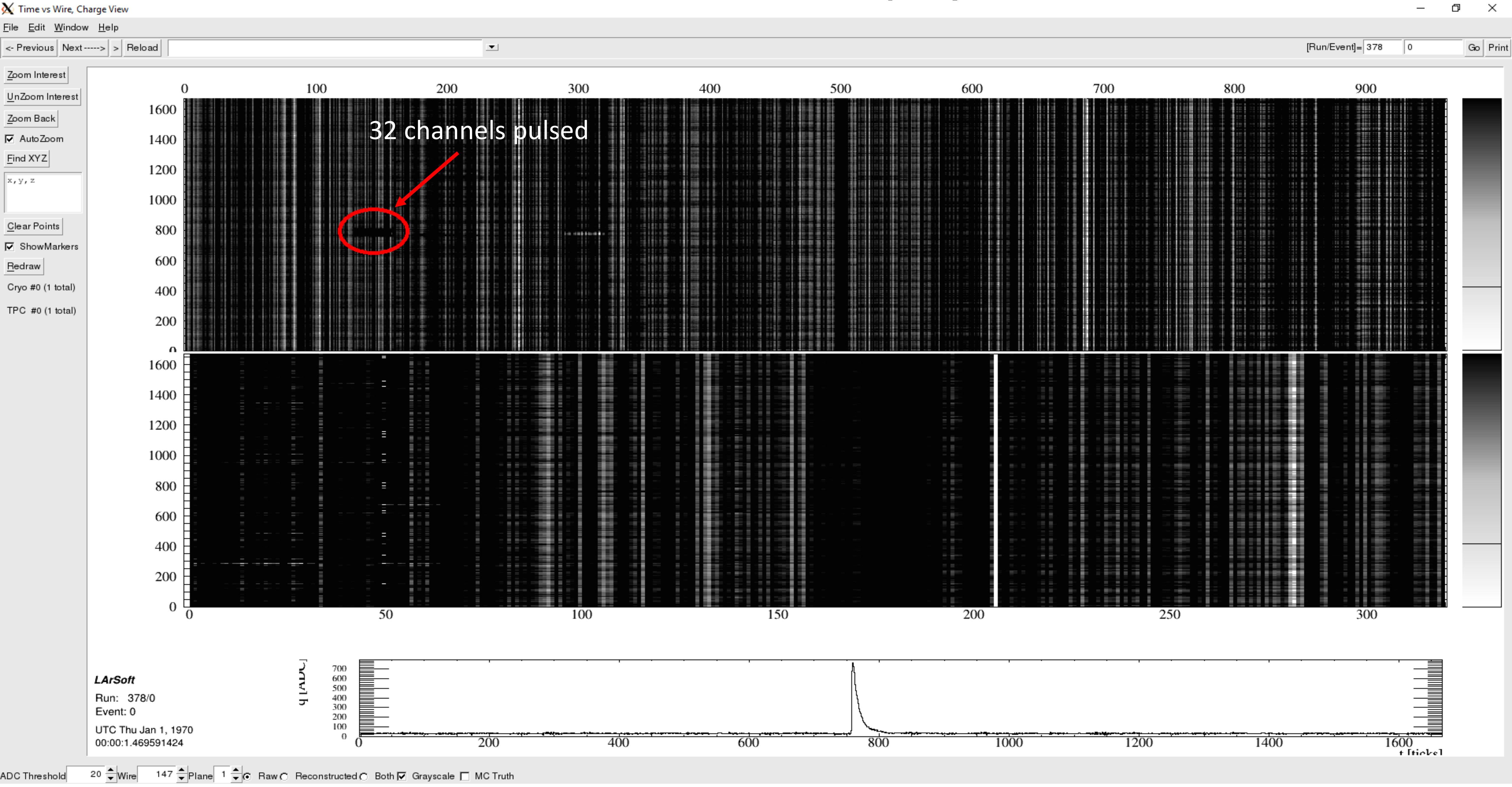
[✓] AutoZoom
[✓] ShowMarkers

LArSoft
Run: 21000015/0
Event: 1
UTC Thu Jan 8, 1981
12:48:5.897255104

ADC Threshold 20 Wire 163 Plane 0 Raw Reconstructed [✓ Both] Grayscale MC Truth

Example:
Imported pulsing data + crosscheck with QScan

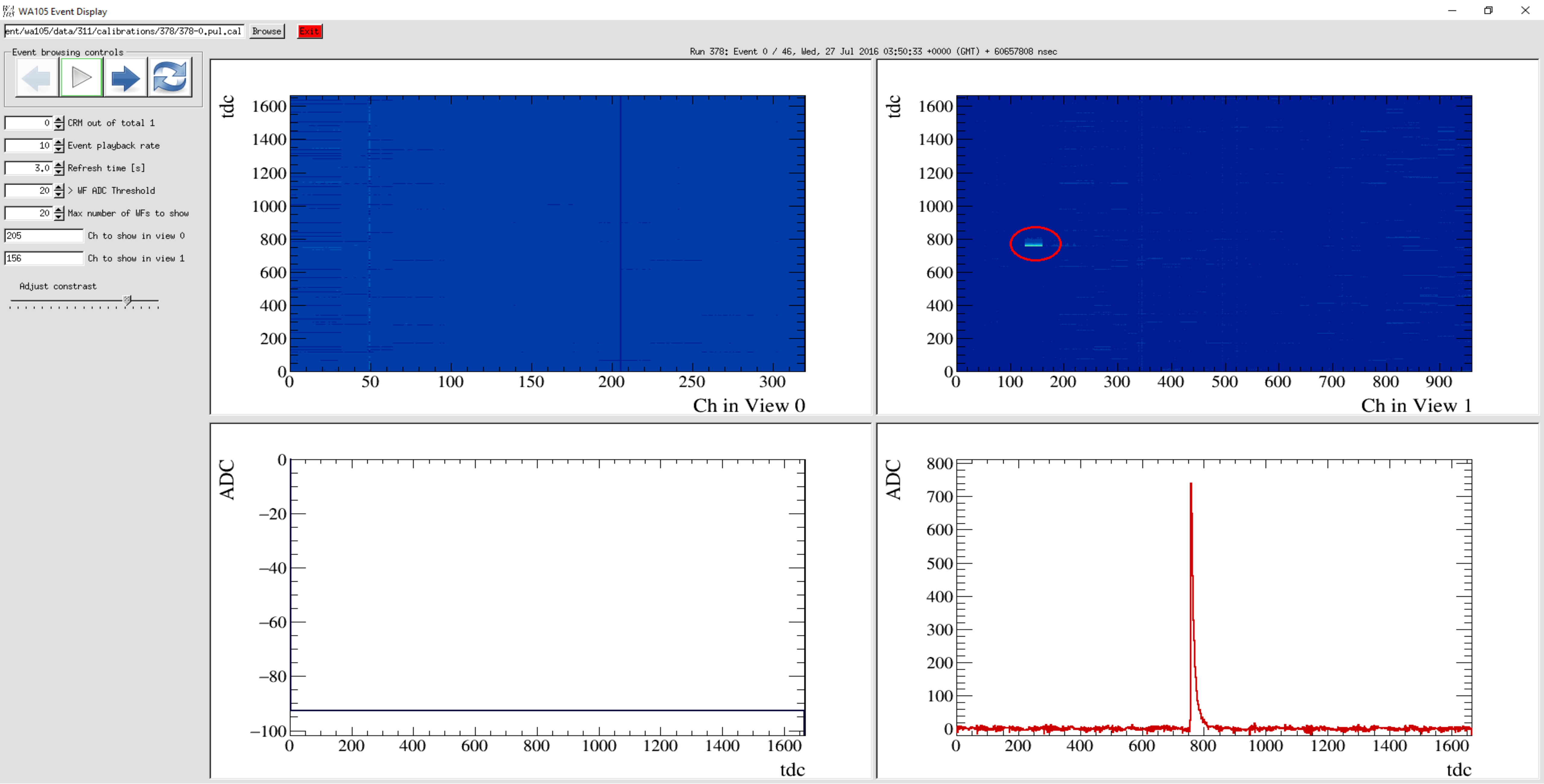
LArSoft event display



LArSoft
Run: 378/0
Event: 0
UTC Thu Jan 1, 1970
00:00:1.469591424

ADC Threshold 20 Wire 147 Plane 1 [✓ Raw] Reconstructed Both [✓ Grayscale] MC Truth

QScan event display



WA105 Event Display

ent/wa105/data/311/calibrations/378/378-0.pul.cal Browse Exit

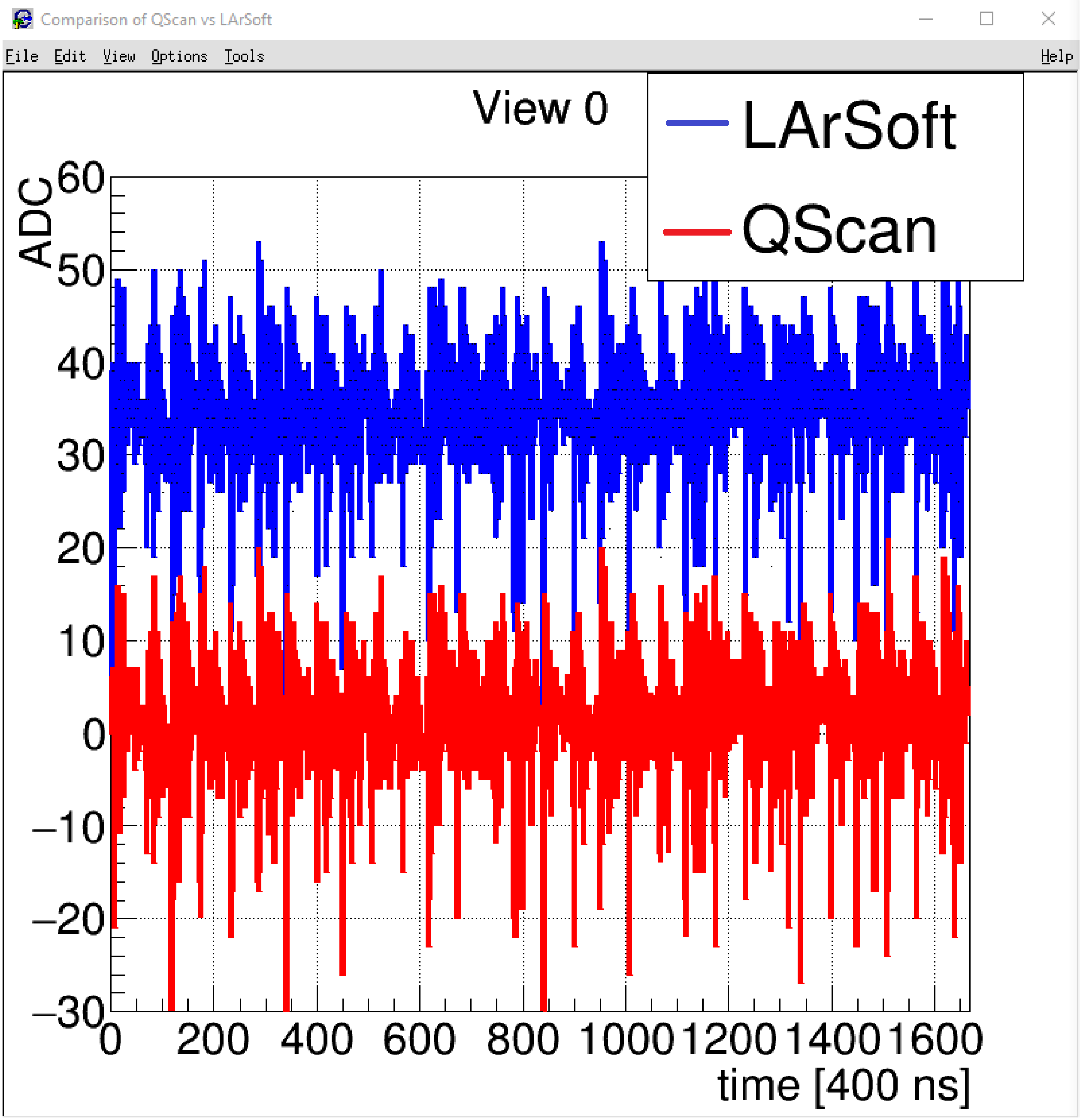
Event browsing controls

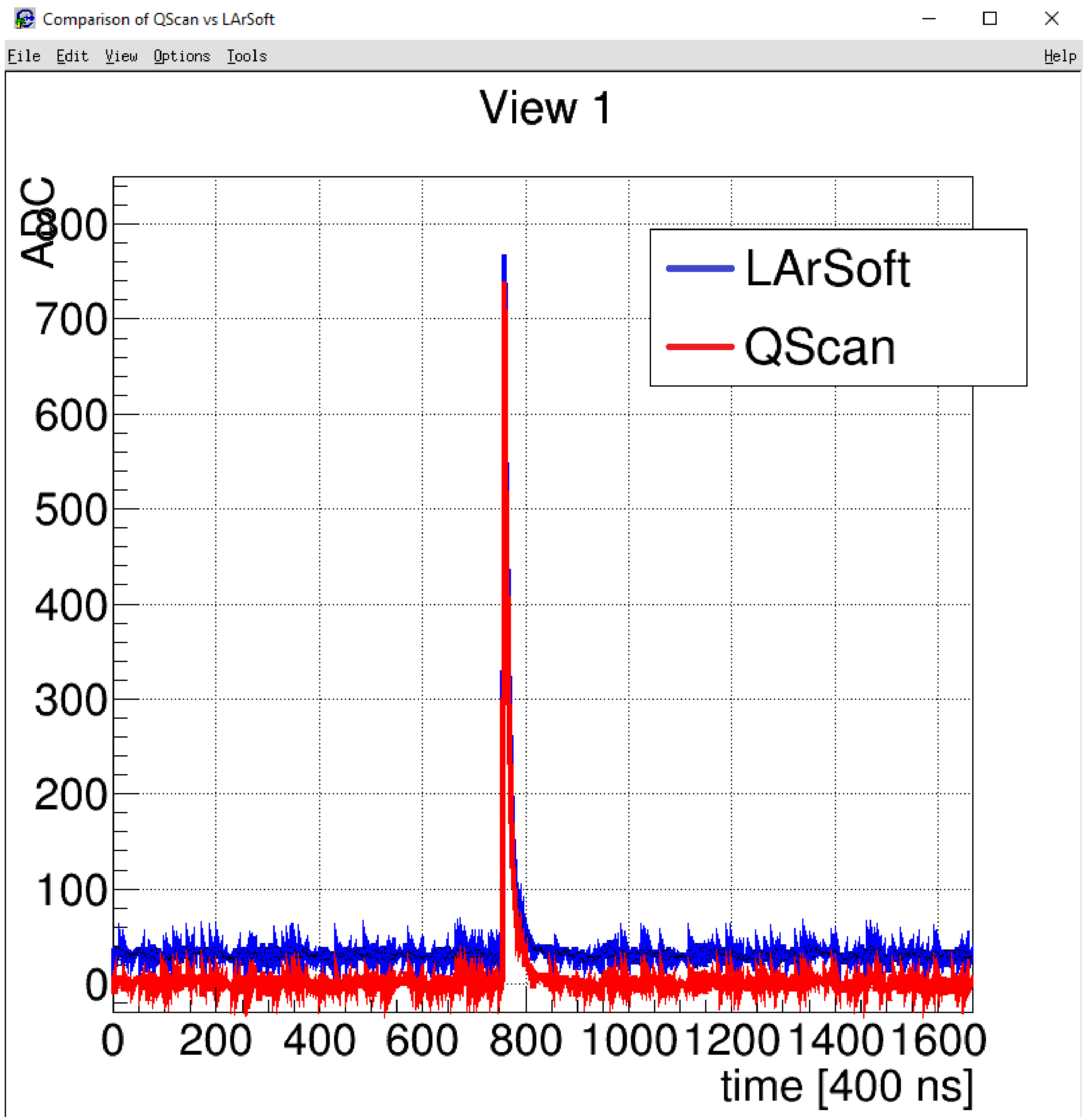
Run 378: Event 0 / 46, Wed, 27 Jul 2016 03:50:33 +0000 (GMT) + 60657808 nsec

```
0    CRM out of total 1
10   Event playback rate
3.0  Refresh time [s]
20   > WF ADC Threshold
20   Max number of WFs to show
205  Ch to show in view 0
156  Ch to show in view 1
```

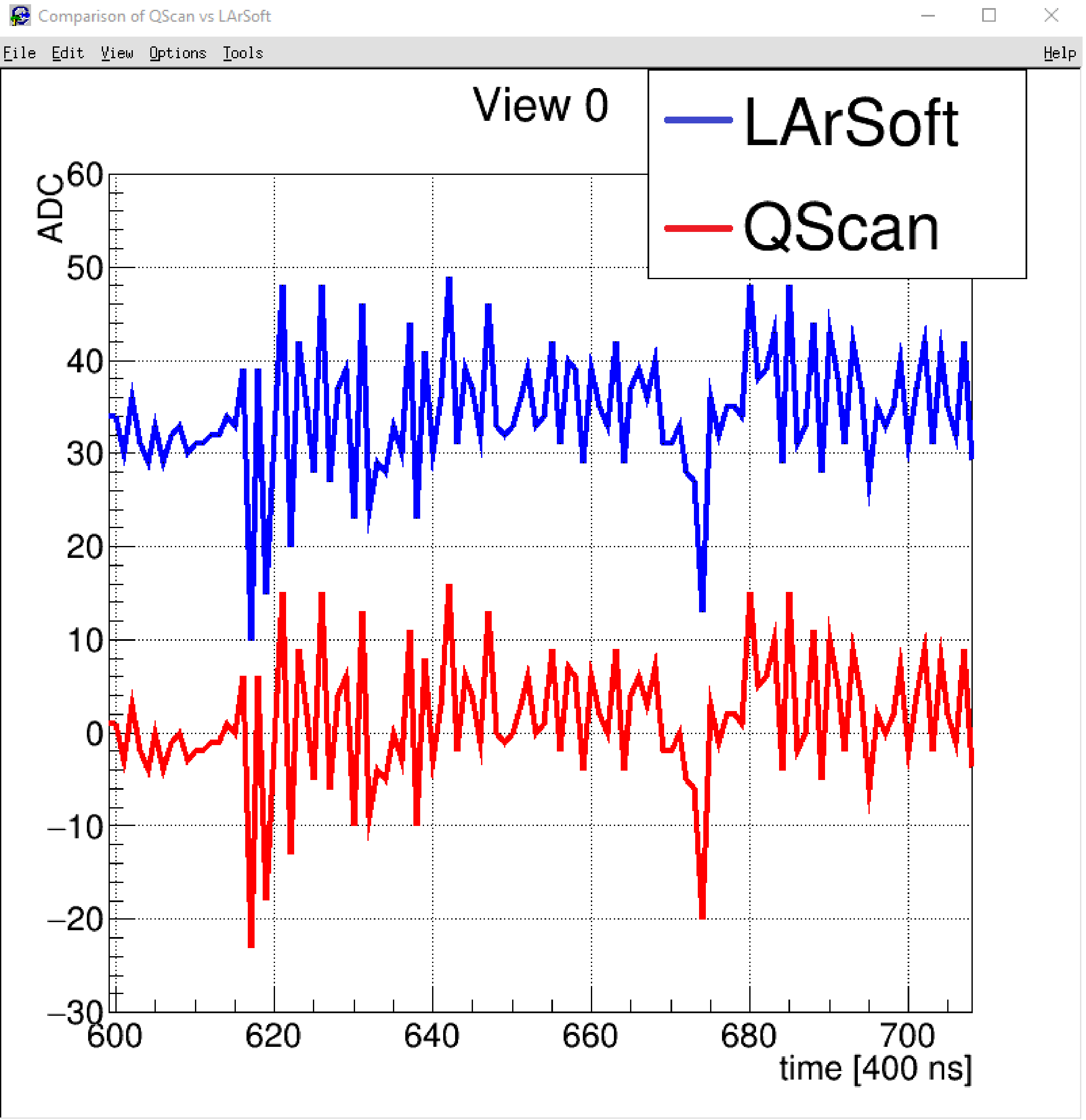
Adjust constrast

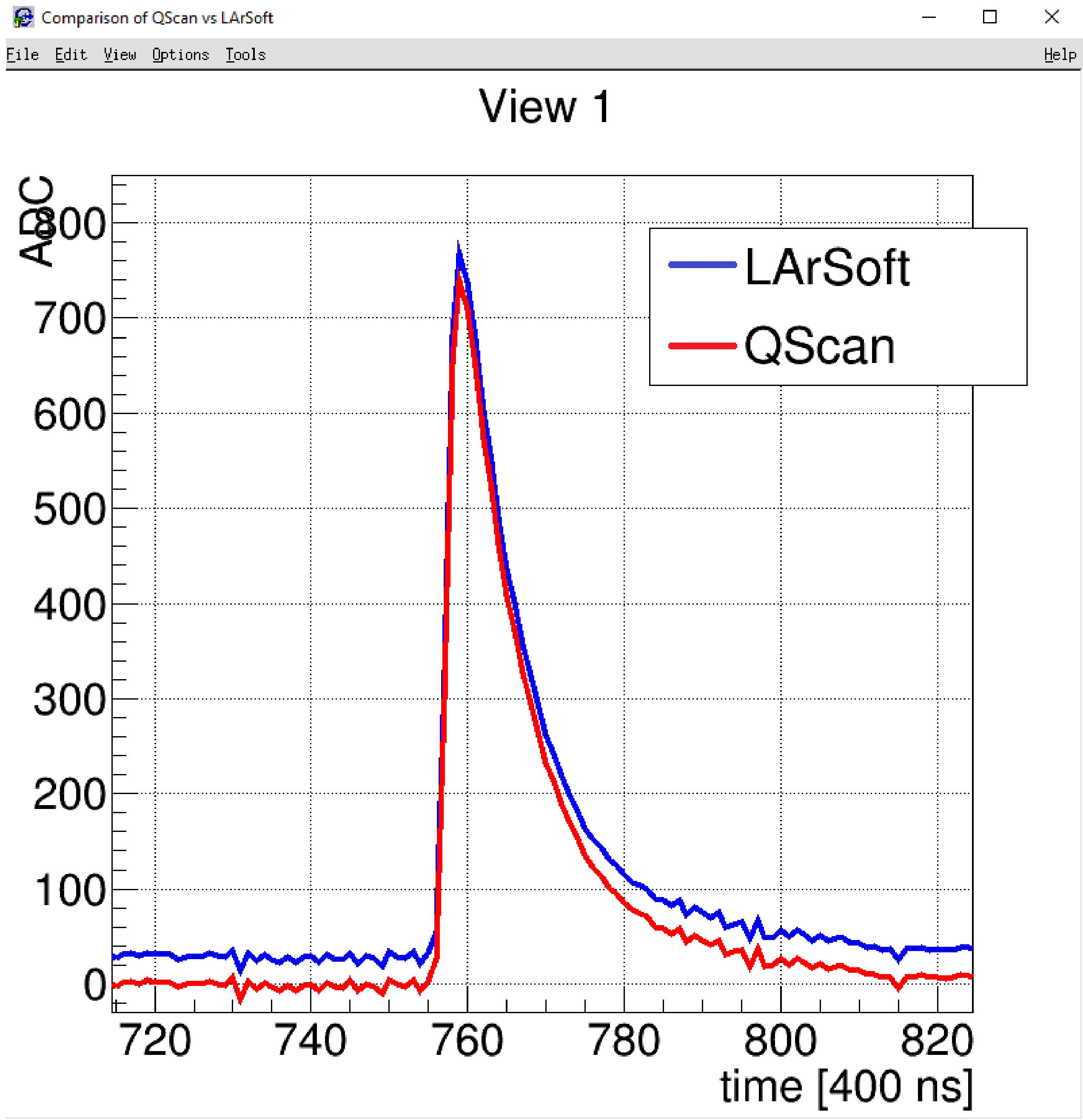
Comparison of Waveforms



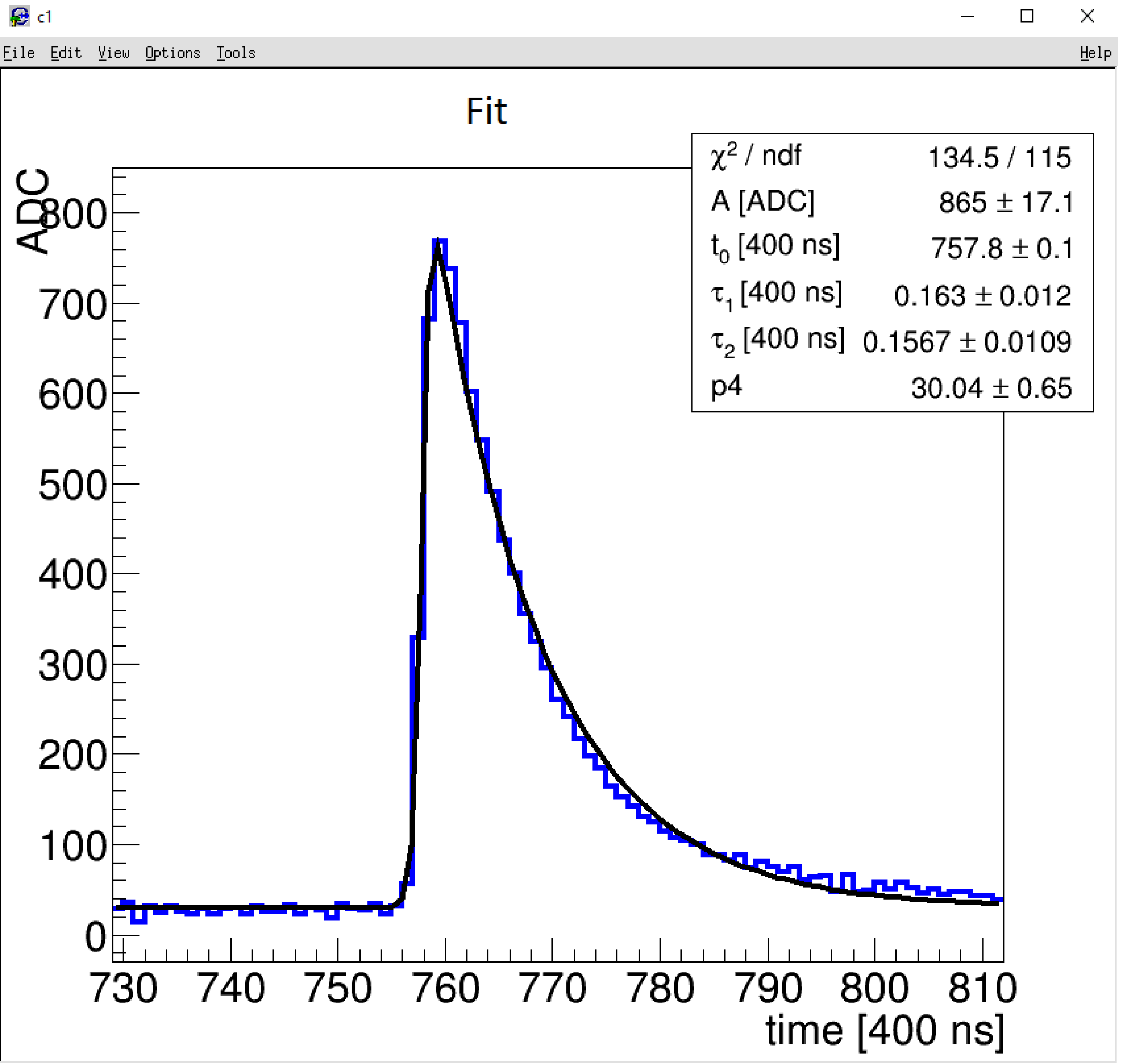


Comparison of Waveforms (Zoom-in)





Other checks.



χ^2 / ndf	134.5 / 115
A [ADC]	865 ± 17.1
t_0 [400 ns]	757.8 ± 0.1
τ_1 [400 ns]	0.163 ± 0.012
τ_2 [400 ns]	0.1567 ± 0.0109
p4	30.04 ± 0.65

Fit Function: $$f(t) = A \cdot \frac{e^{\frac{t-t_0}{\tau_1}}}{1+e^{\frac{t-t_0}{\tau_2}}} + \text{p4}$$

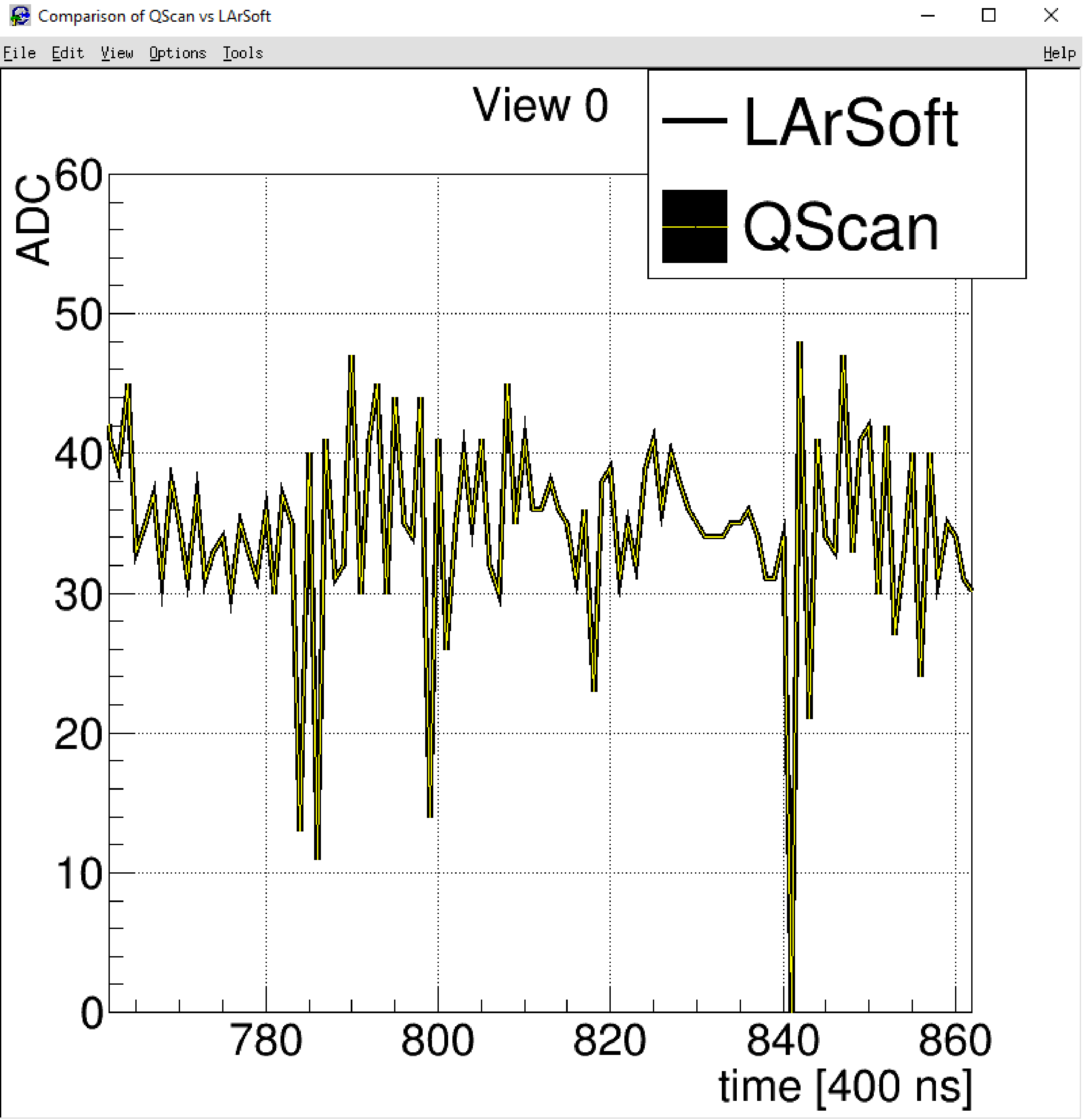
Conclusion

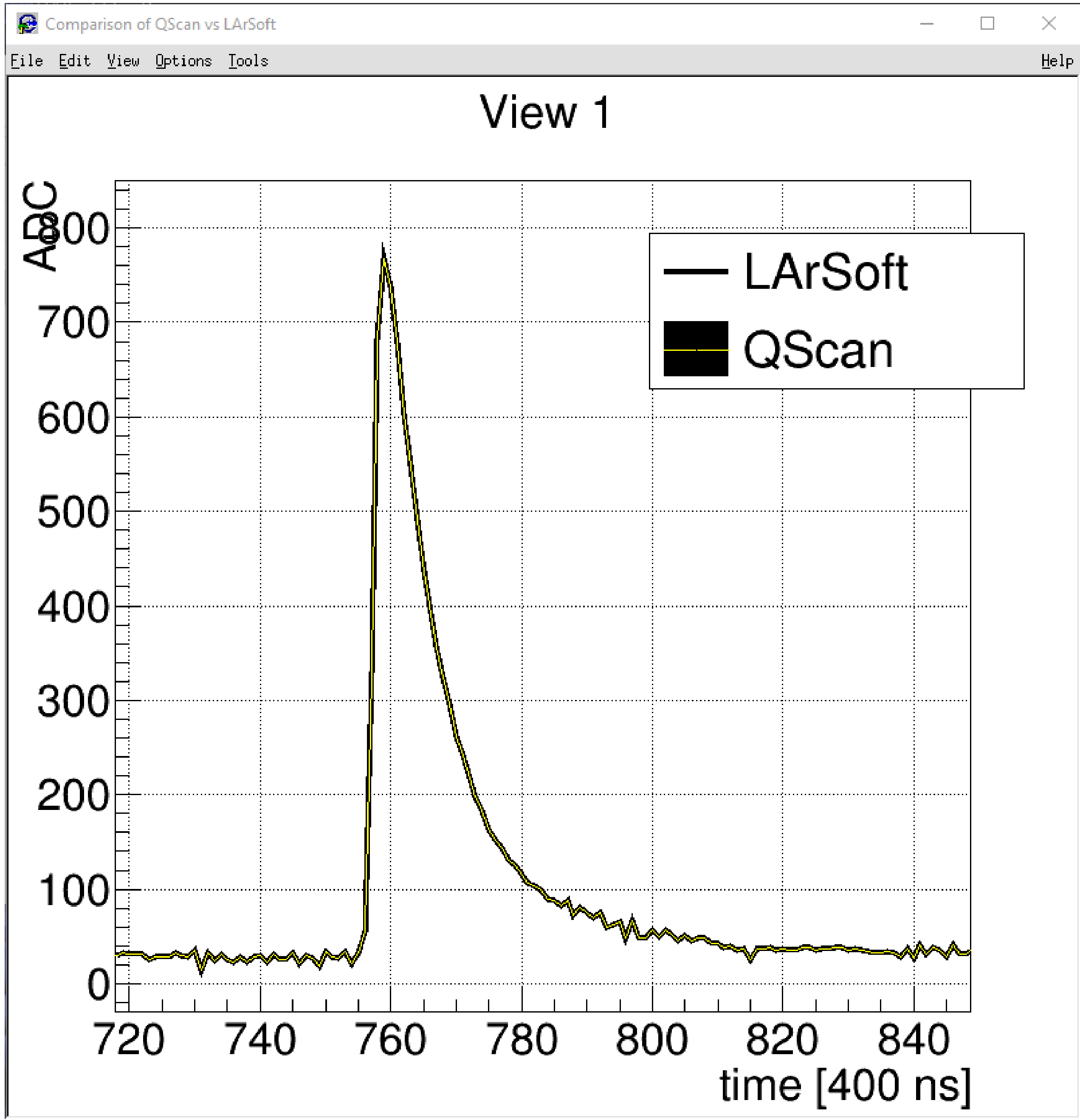
- 3x1x1 geometry is implemented in LArSoft.
- Import of data is successful.
- Code has been pushed by Christoph.
- Everything is ready to start analyzing pulsing data and noise measurements.

Thank you for your attention!

Backup Slides

Zoom-in without pedestal substraction





3x1x1 Raw Data Structure

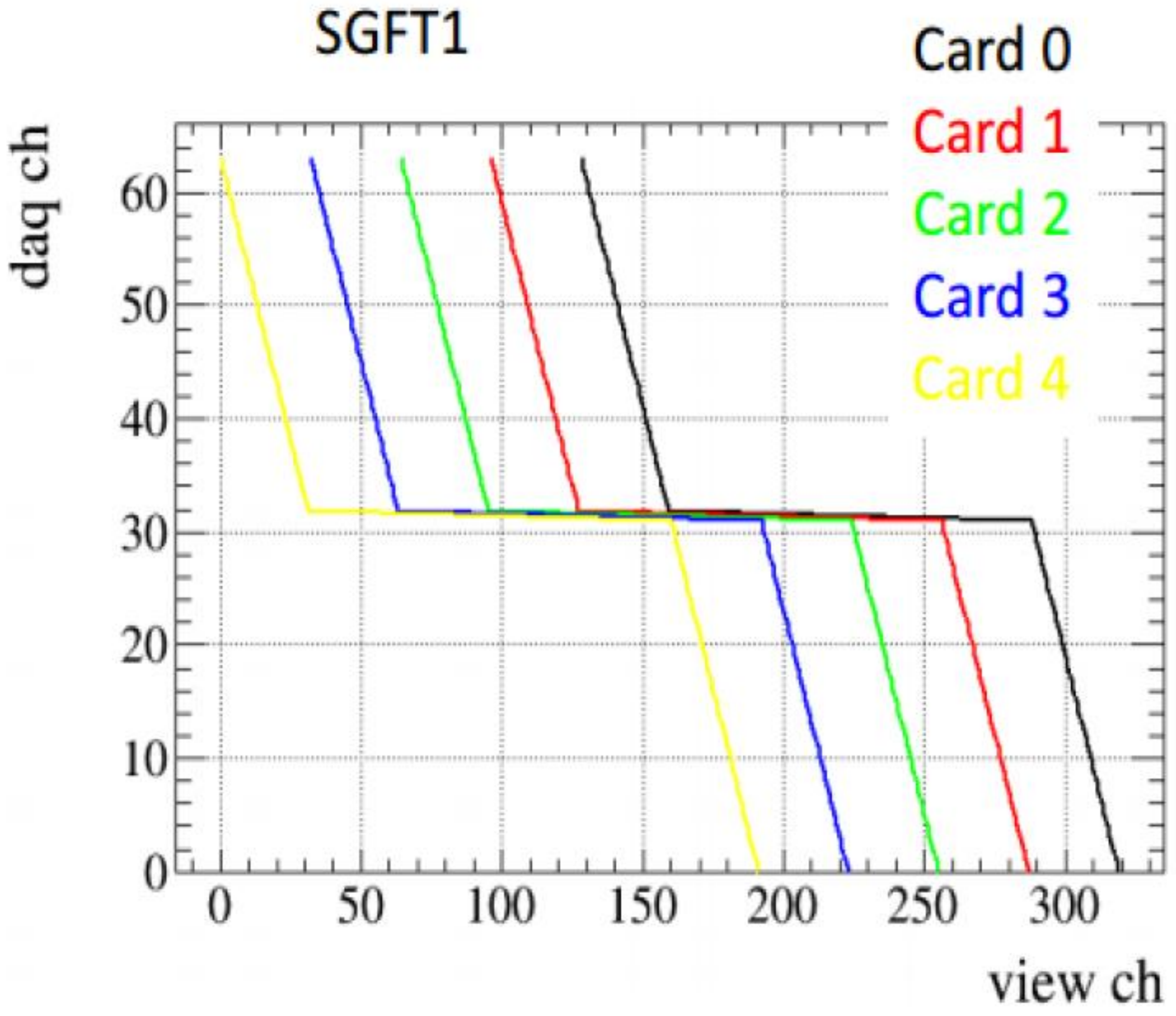
3x1x1 measurements are accessible at eos. Data structure as implemented by Elisabetta and Slavic:

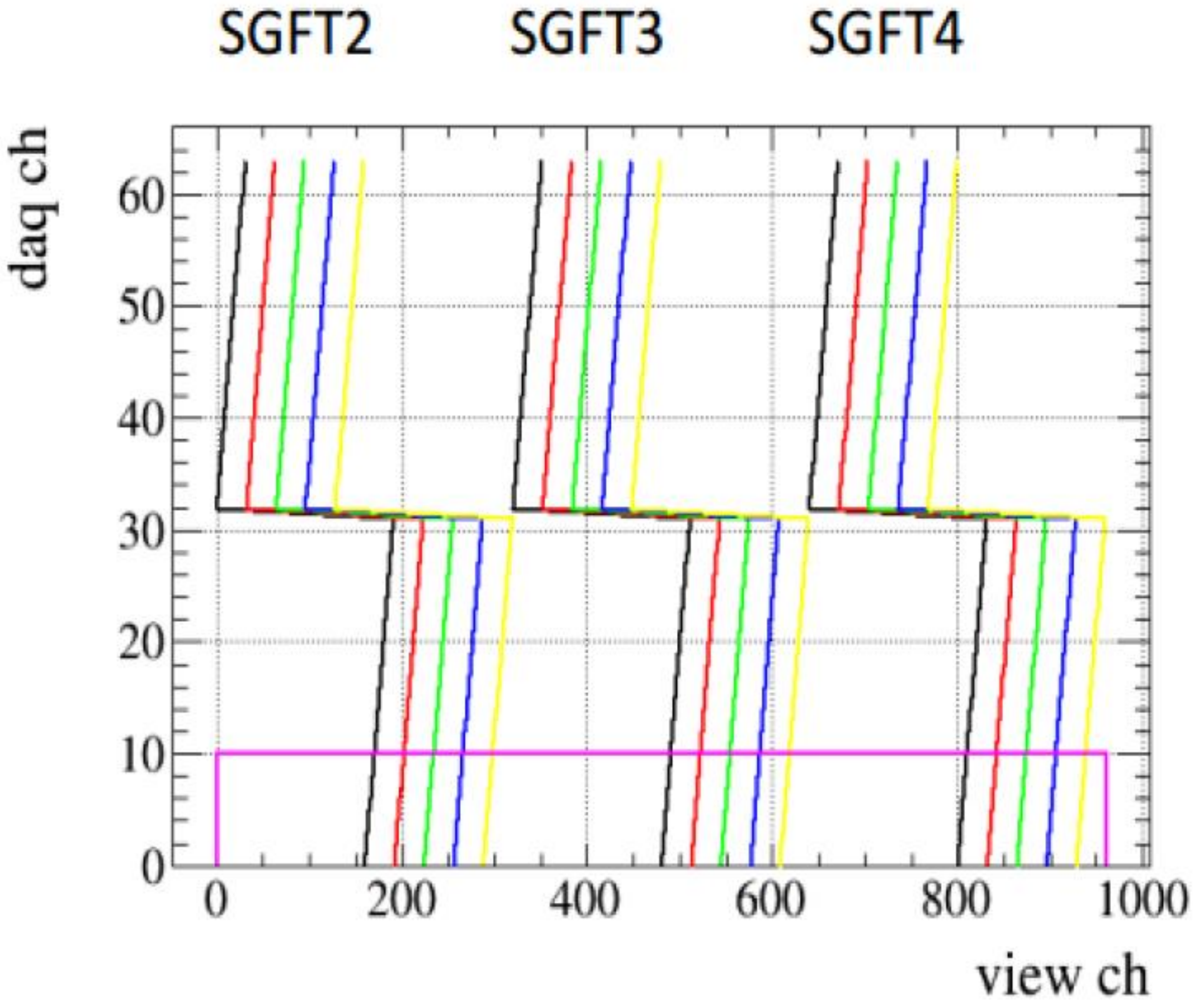
- RawData (raw or manipulated) is stored as binary file.
- Each file contains up to 335 events.
- Per event the data is stored as a single vector holding the ADC counts of all the channels.

Example: 633-0.dat:

- First 5 bytes: run header: contains the run number (4 bytes) and a flag (1 byte).
- Last 4 bytes: footer: contains keys for internal checks (2 bytes) and the number of events recorded in the file (2 bytes).
- Per event: -) Event header (35 bytes): contains keys for internal checks (2 bytes), trigger info (24 bytes), data quality flag (1 byte), event number (4 bytes) and event size (4 bytes).
  -) Then come the ADC counts: read in card by card, channel by channel.
  -) The data is stored in 12 bit format.

Order of Channels (Graphic taken from Slavic's presentation at the general collaboration meeting)





Data Import from QScan

What does LArSoft want: root file containing:

→ art::event containing a collection of raw::Digit

→ raw::Digit is a class with member elements:

-) Channel number.

-) Number of ticks for this channel.

-) ADC vector for this channel.

-) Information about the used compression.