Online monitoring for ProtoDUNE-SP

Jingbo Wang, Antonino Sergi, Marco Adinolfi

ProtoDUNE-SP DAQ Commissioning Meeting

UCDAVIS
UNIVERSITY OF CALIFORNIA

UNIVERSITY OF BIRMINGHAM

CERN

January 8, 2018, CERN

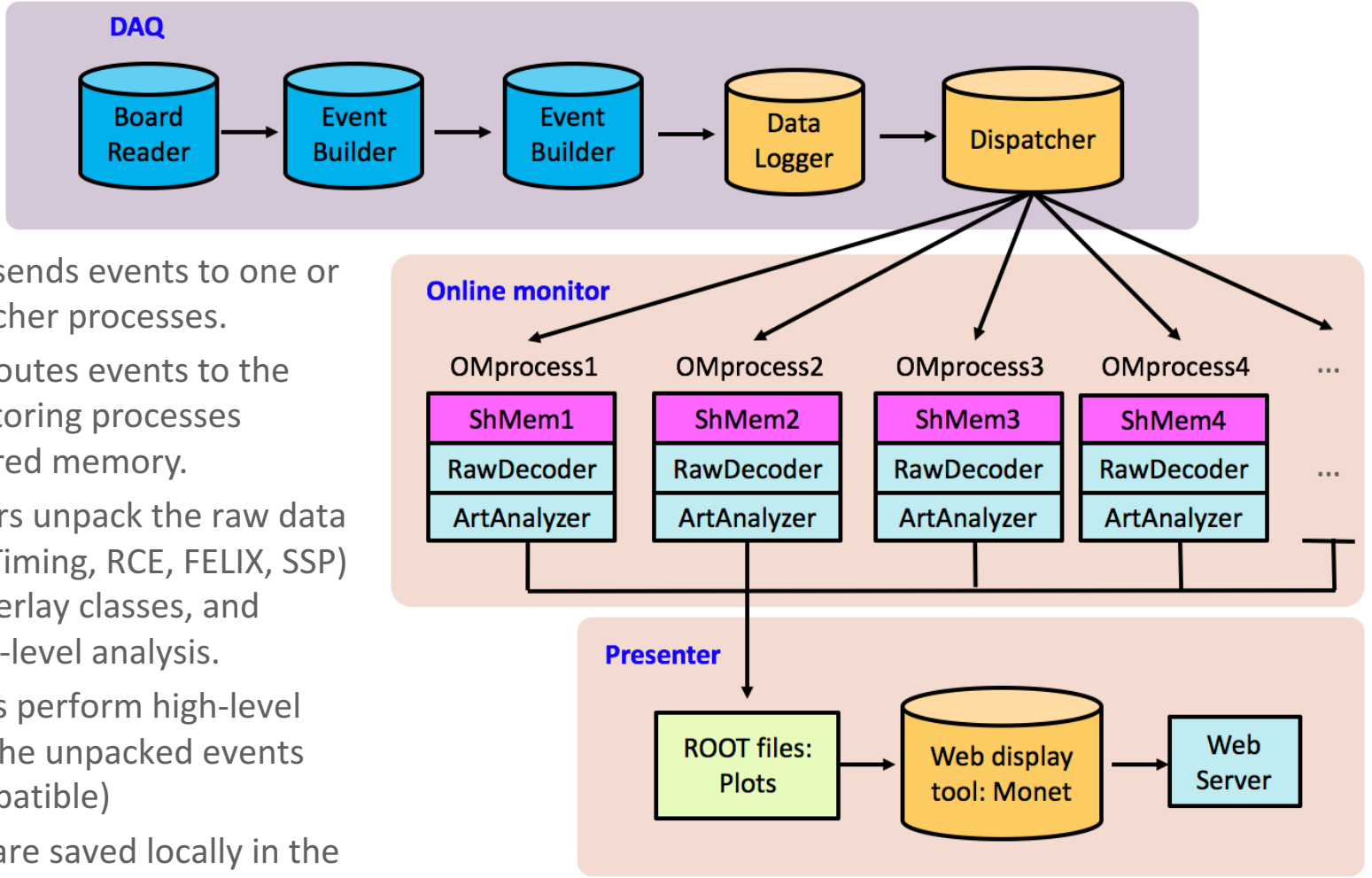
Online monitoring

- During the run it will be essential to monitor the performances of the detectors and make sure everything is functioning as expected.
- The online monitoring runs with the DAQ during data taking and updates at regular intervals to indicate any possible problems.
- The monitoring modules are written in art and will unpack the raw data and produce the monitoring plots during the run.
- At the moment, the basic online monitoring framework has been built and is being tested.

Candidate Monitoring Metrics

- **General monitoring**
  - Number of subdetectors with data
  - Event size vs “time”
  - Trigger rate vs “time”
- **Event monitoring**
  - Raw event display in three views for each APA
  - Mean/RMS of the ADCs vs channel number
  - FFT of waveforms vs channel number
  - Stuck bits fraction
  - Trigger type
- **Plots with basic reconstruction** (depending on the computational capability)
  - Number of hits per view in the TPC
  - Number of hits/average amplitude per view vs time
  - Distribution of cluster length and number of hits per cluster
- **Hardware diagnostic information (for electronics experts)**
  - Seeking suggestions

Architecture of OM software



- DataLogger sends events to one or more dispatcher processes.
- Dispatcher routes events to the online monitoring processes through shared memory.
- Raw decoders unpack the raw data fragments (Timing, RCE, FELIX, SSP) using the overlay classes, and perform low-level analysis.
- Art analyzers perform high-level analysis of the unpacked events (offline compatible)
- Histograms are saved locally in the ROOT files.
- Web display tool (Monet/LHCb) copies the histograms to the web server

Installation of the online monitor

- Installation of the online monitor (John Freeman):
  - https://cdcvs.fnal.gov/redmine/projects/dunetpc/wiki/Guide_for_protoDUNE_online_monitoring_developers
- Installation of the web-display tool Monet (Marco Adinolfi):
  - https://twiki.cern.ch/twiki/bin/view/CENF/PdspMonet

- The online monitor is installed in the CERN computer np04-srv-014:/nfs/sw/om
  - Using a dunetpc feature branch "feature/online_monitoring_artdaq" (depends on both lbne-raw-data and dune-raw-data) with the raw decoders
  - Using the develop branch of dune_raw_data that provides the overlay classes and the channel mapping service.

- Note: the RCE overlay class depends on an external library that is installed in the Proto-DUNE computer at CERN. At the moment, the online monitoring software is only functional on the CERN machines. Fermilab (Tom Junk) is working to merge this library to dunetpc for offline use.

Running the online monitor

- Basic operation: https://twiki.cern.ch/twiki/bin/view/CENF/PdspOMOp (to be updated)

- Run from the DAQ interface
  - The configuration files (fcl files) are stored in np04-srv-014:/nfs/sw/om/fcl
  - RawDecoder.fcl defines the parameters for all the decoding and analyzing modules
  - RunOnlineMonitor_ShMem.fcl is called by the DAQ interface to analyze the live data during a run
  - StartOM_lite.sh starts a single OM process and saves the process ID to a txt file.
  - StopOM_lite.sh reads the process ID and kills the OM process
  - The output root file and log message file are saved to a local path: /scatch/OMoutput to be accessed by Monet.

- Run from the terminal
  - source /nfs/sw/om/ProtoDUNE_OM/setupONLINEMONITORING
  - Online with live data: art –c RunOnlineMonitor_ShMem.fcl
  - Offline with an existing file: art –c RunRawDecoder.fcl RawRootFile.root

- Multi-process script was prepared and will be tested soon.

Example fcl file

np04-srv-014:/nfs/sw/om/fcl/RunOnlineMonitorShMem.fcl

```
#include "services_dune.fcl"
#include "RawDecoder.fcl"
services:
{
   TimeTracker:        {}
   RandomNumberGenerator: {}
   #message:              @local::dune_message_services_prod_debug
   FileCatalogMetadata:  @local::art_file_catalog_mc
   ExptGeoHelperInterface:        @local::dune_geometry_helper
   Geometry:                      @local::protodune_geo
   DetectorClocksService:         @local::dunefd_detectorclocks
   DetectorPropertiesService:     @local::protodune_detproperties
   LArPropertiesService:          @local::dunefd_properties
   LArFFT:                        @local::dunefd_larfft
   DatabaseUtil:                  @local::dunefd_database
   ChannelStatusService:          @local::dunefd_channel_status
   ChannelGroupService:           @local::changroup_apa
   PdChannelMapService: {
     FileName:          "protoDUNETPCChannelMap_v1.txt"
     LogLevel:          1
   }
   TFileServiceOM: {
     closeFileFast: true  # default
     fileName:  "/scratch/OMoutput/np04_hist_run%06r_%#.root"
     tmpDir: "<parent-path-of-filename>"  # default
     fileProperties: {
      maxEvents: 100
     }
   }
}
# RCE and FELIX waveform has a size of 10000
services.DetectorPropertiesService.NumberTimeSamples: 10000
services.DetectorPropertiesService.ReadOutWindowSize: 10000
```

Channel mapping service

Online monitoring specific service allows for file switching during the run (per 100 events)

```
physics:
{
  producers:
  {
    timingrawdecoder: @local::timing_raw_decoder
    ssprawdecoder: @local::ssp_raw_decoder
    rcerawdecoder: @local::rce_raw_decoder
    felixrawdecoder: @local::felix_raw_decoder
  }

  analyzers:
  {
    tpcmonitor: @local::tpc_monitor
    tpcfft: @local::tpc_fft
    sspmonitor: @local::ssp_monitor
  }

  #Create path to run module
  produce: [ timingrawdecoder, ssprawdecoder, rcerawdecoder, felixrawdecoder ]
  analyze: [ tpcmonitor, tpcfft, sspmonitor]
  trigger_paths : [ produce ]
  end_paths: [ analyze ]
}

source:
{
  module_type: TransferInput
  timeoutInUsecs: 10000000
  dispatcherHost: localhost
  dispatcherPort: 5266
  transfer_plugin: {
    transferPluginType: Shmem
    unique_label: "shmem0"
    shm_key: 0x40471450
    max_fragment_size_words: 40971520
    first_event_builder_rank: 0
  }
}

process_name: Onlinemonitor
```

Unpack the raw data and write to offline format

Analyze the event with offline format

Producers and analyzers are ran in sequence and no intermediate file needs to be saved to the disk

Data transferred through shared memory

RCE/FELIX Channel Map

Online channel	RCE, FELIX ID	RCE, FELIX channel	Regulator	Regulator pin	Asic	Asic channel	APA	APA plane	Offline channel
0	100	0	1	32	0	0	3	1	0
1	100	1	1	33	0	1	3	0	1
2	100	2	1	34	0	2	3	1	2
3	100	3	1	35	0	3	3	0	3
4	100	4	1	36	0	4	3	1	4
5	100	5	1	37	0	5	3	0	5
6	100	6	1	38	0	6	3	1	6
7	100	7	1	39	0	7	3	0	7
8	100	8	1	40	0	8	3	1	8
9	100	9	1	41	0	9	3	0	9
10	100	10	1	42	0	10	3	1	10
11	100	11	1	43	0	11	3	0	11
12	100	12	1	44	0	12	3	1	12
13	100	13	1	45	0	13	3	0	13
14	100	14	1	46	0	14	3	1	14
15	100	15	1	47	0	15	3	1	15
16	100	16	1	48	1	0	3	1	16
17	100	17	1	49	1	1	3	1	17
18	100	18	1	50	1	2	3	0	18
19	100	19	1	51	1	3	3	1	19
20	100	20	1	52	1	4	3	0	20
21	100	21	1	53	1	5	3	1	21
22	100	22	1	54	1	6	3	0	22
23	100	23	1	55	1	7	3	1	23
24	100	24	1	56	1	8	3	0	24
25	100	25	1	57	1	9	3	1	25
26	100	26	1	58	1	10	3	0	26
27	100	27	1	59	1	11	3	1	27
28	100	28	1	60	1	12	3	0	28
29	100	29	1	61	1	13	3	1	29
30	100	30	1	62	1	14	3	0	30
31	100	31	1	63	1	15	3	1	31
32	100	32	1	15	2	0	3	0	32

```
int onlineChannel = channelMap->OnlineFromRCE(frag.fragmentID(), i_ch);
ch_counter++;
_h_online_channels->Fill(onlineChannel);
int offlineChannel = -1;
offlineChannel = channelMap->Offline(onlineChannel); // to get offline from online
_h_offline_channels->Fill(offlineChannel);
raw::RawDigit raw_digit(offlineChannel, n_ticks, v_adc);
raw_digits.push_back(raw_digit);
```

- Modified the 35-ton channel map service and the map file
- RCE/FELIX fragment ID and internal channel ID are known parameters obtained from the DAQ
- Mapping the RCE/FELIX parameters to oneline/offline channel numbers
- Regulator and asic information are not used at this moment, but will be useful for coherent noise removal
- APA and plane columns may be not needed because they can be obtained from the offline channel number with the geometry service

RCE and FELIX Monitoring

- Raw decoder (art producer module)
  - Number of ticks
  - All ADC values
  - Online and offline channel IDs
  - Expect more electronics diagnostic input: asic ID, asic channel?
- Monitoring analyzers (art analyzer modules)
  - Raw event display in three views
  - ADC Mean/RMS VS channel ID for each view in each APA (2D histogram)
  - **Profiled histogram of the ADC Mean/RMS**
  - FFT spectrum VS channel ID (2D histogram)

✓ FELIX is assigned to the last APA.

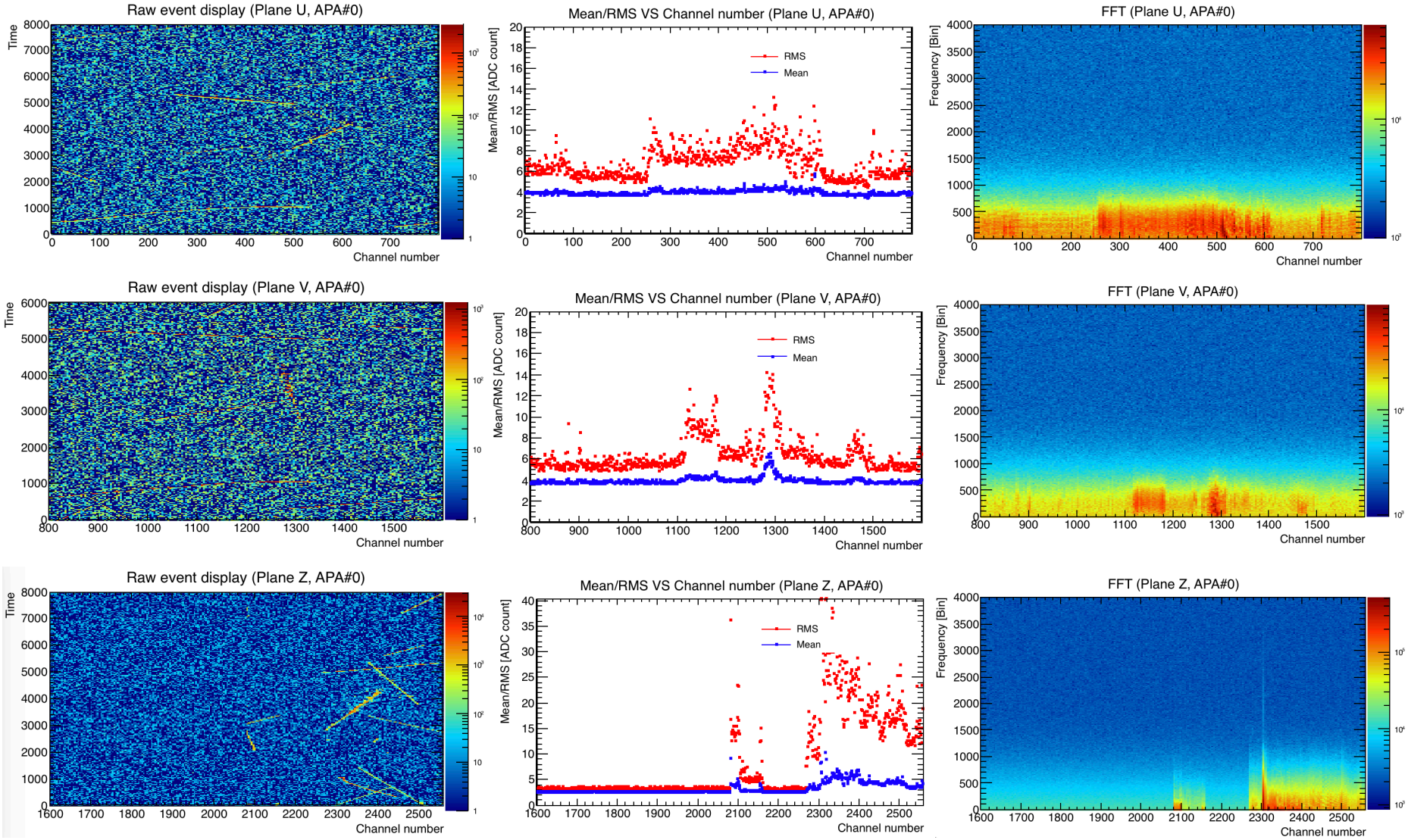
✓ In the analyzer module, raw digits from RCE and FELIX are merged to a single "TPC" vector

✓ Analyzer modules work for offline analysis with simulated file as well

✓ Channel map still needs to be settled

For more available offline modules, see https://twiki.cern.ch/twiki/bin/view/CENF/DUNEProtSPOM

Test with simulation file

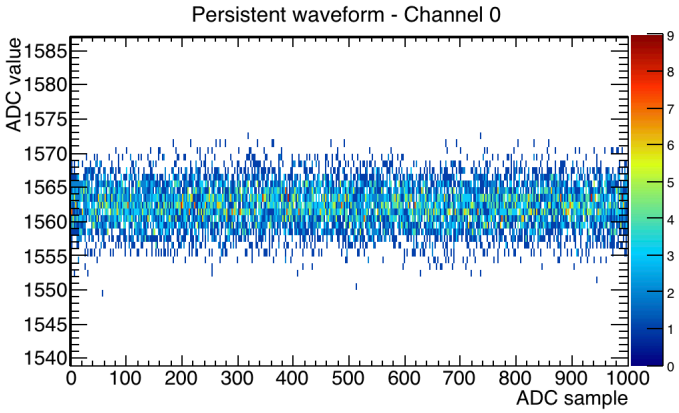


SSP Monitoring

- Raw decoder (art producer module)
  - All ADC values
  - Number of packets per event
  - Peak (from metadata) spectrum per channel
  - Area (from metadata) spectrum per channel
  - Area (from metadata) spectrum per channel
  - One waveform per channel per file
  - FFT per channel
  - Trigger type histogram (internal, external)
  - Persistent waveforms over many events
- Monitoring analyzers (art analyzer modules)
  - ADC Mean/RMS VS channel ID (2D histogram)
  - Profiled histogram of the ADC Mean/RMS
  - More suggestions?

✓ Channel map needs to be settled
✓ High-level analysis (FFT) can be moved to the analyzer

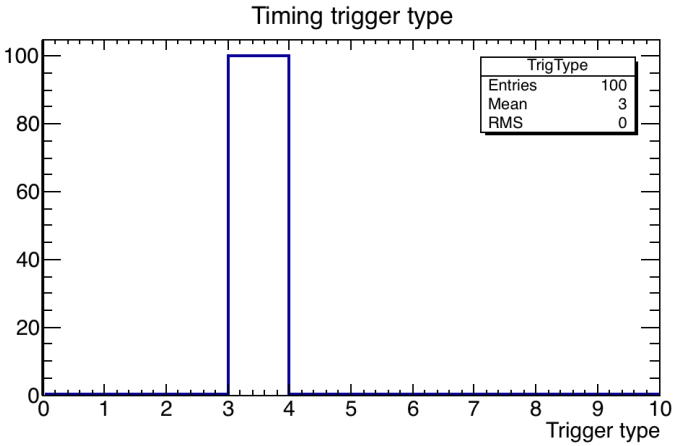
→ Plots added by Paolo Franchini

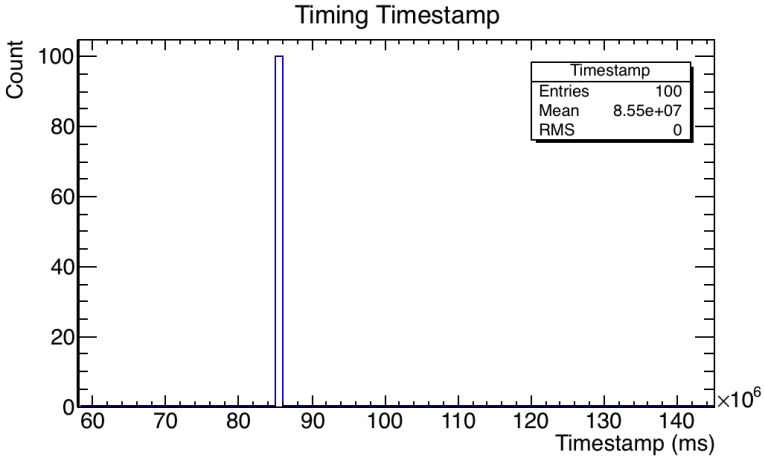


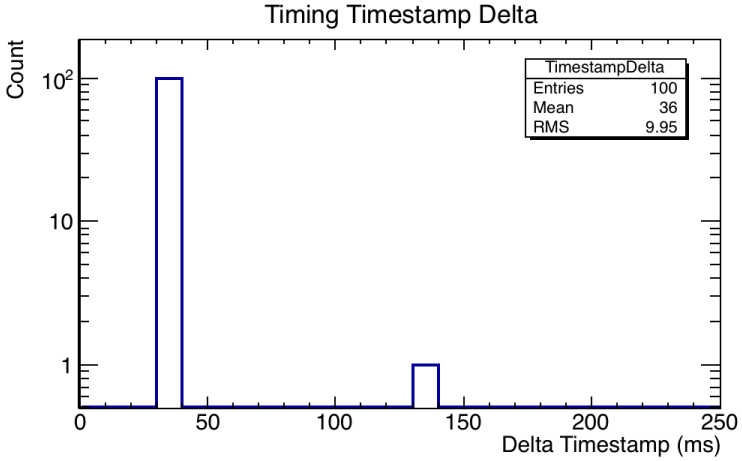
Timing Board Monitoring

- Raw decoder (art producer modules)
  - Time stamp of the timing fragment
  - Time difference between two consecutive fragments
  - Trigger type
- Monitoring analyzers (art analyzer modules)
  - Suggestion?

Plots added by Philip Rodrigues



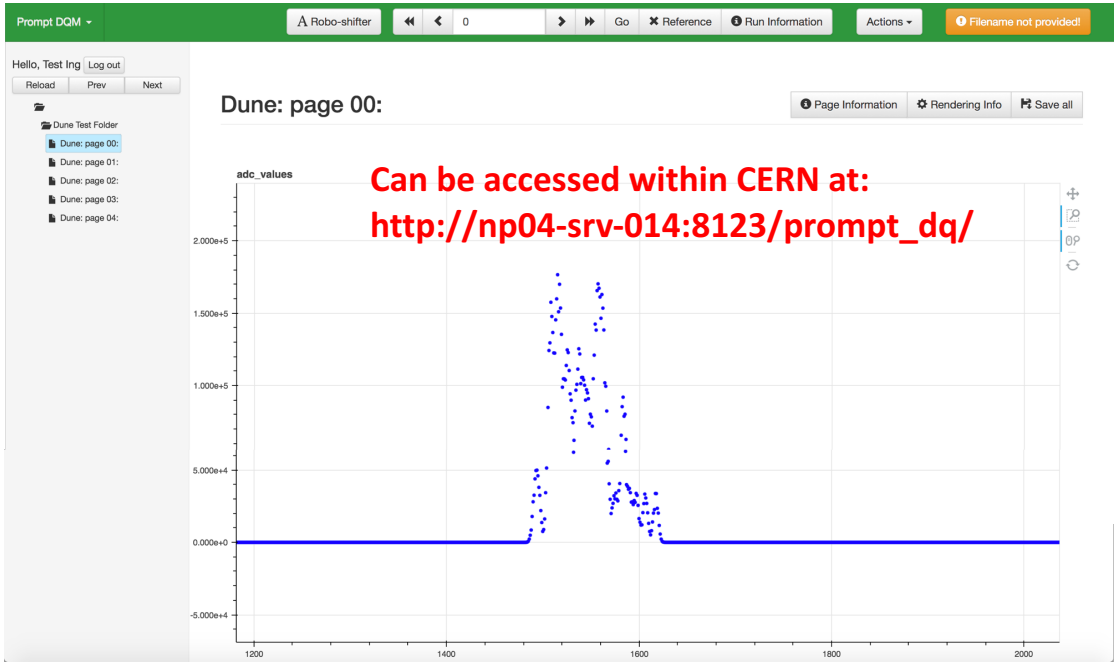




Web Display Tool: Monet

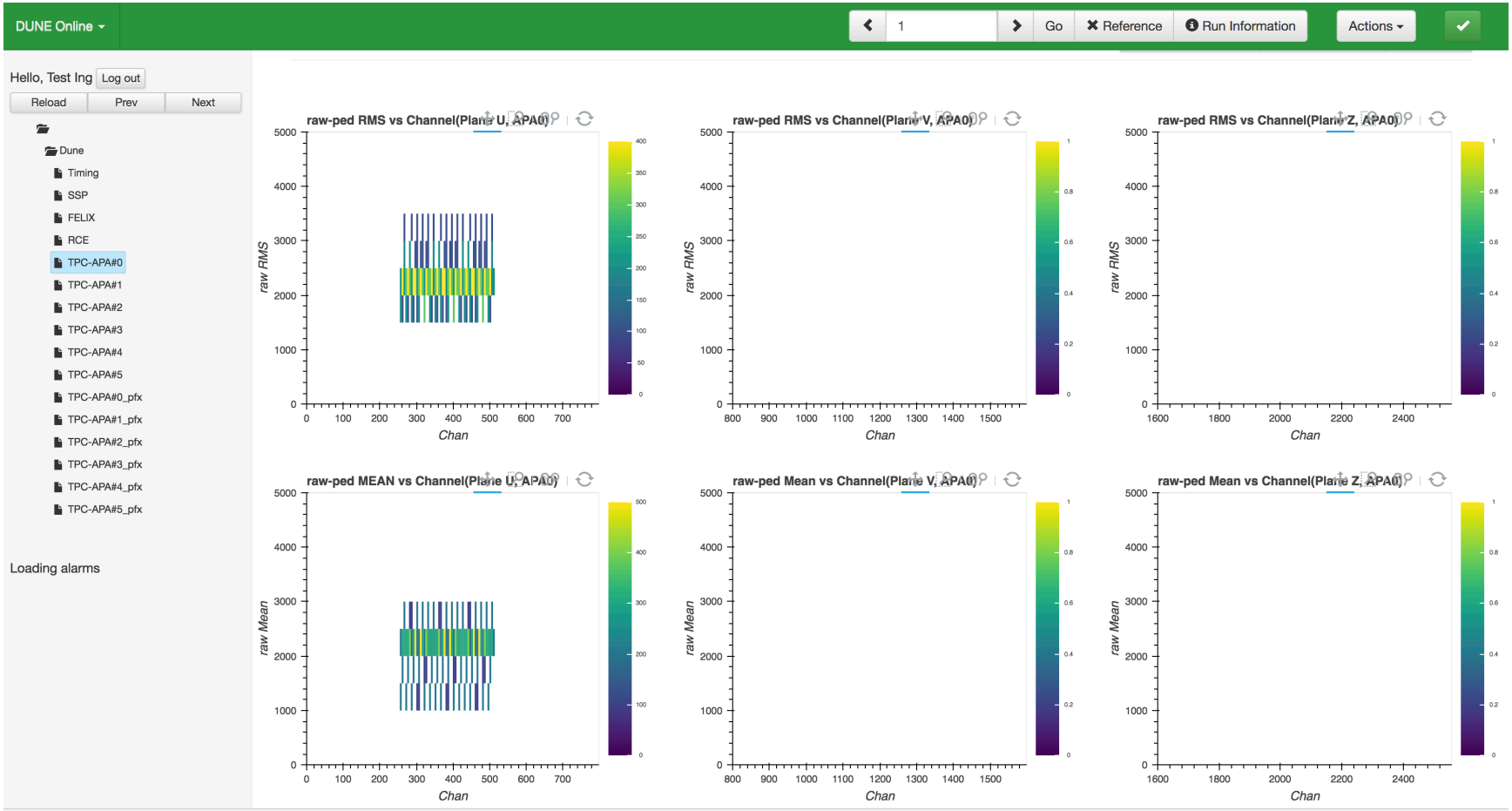
- Web allows for easy monitoring from everywhere without connecting to remote machines
- Proto-DUNE is using a modified version of Monet that was originally written by LHCb.
- Monet is written in python and has several user cases. The PromptDQM case is hacked for Proto-DUNE, as that is the one that reads ROOT files (done by Marco Adinolfi).

- ✓ Folder/page/histogram structure allows to organize plots for different categories of users: shifter, expert, special studies...
- ✓ /nfs/sw/om/Monet/Dune_page_file defines the pages
- ✓ /nfs/sw/om/Monet/Dune_histo_file defines which histogram to present for each page.



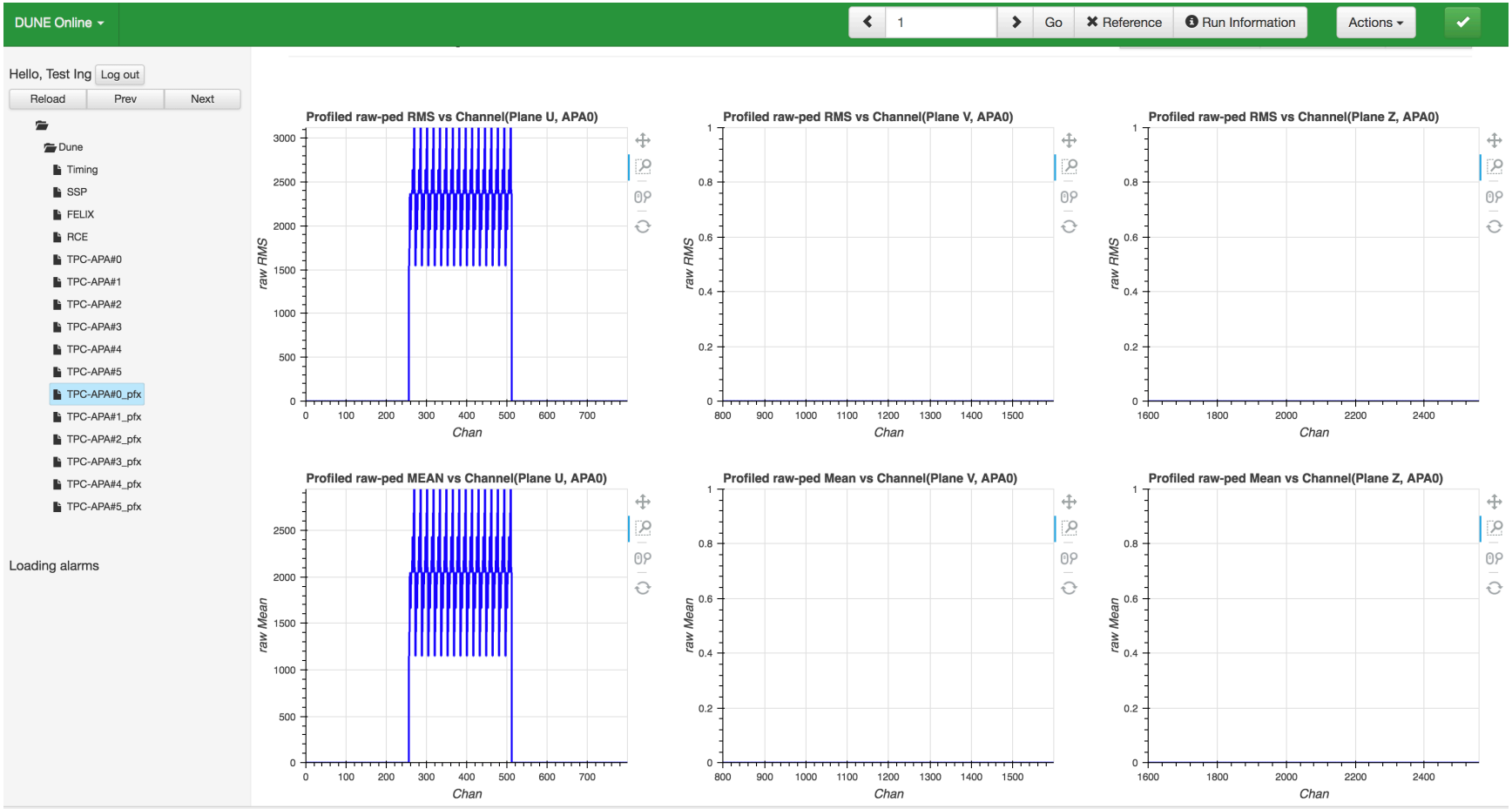
Preliminary plots in Monet: TPC

ADC Mean/RMS vs Channel#



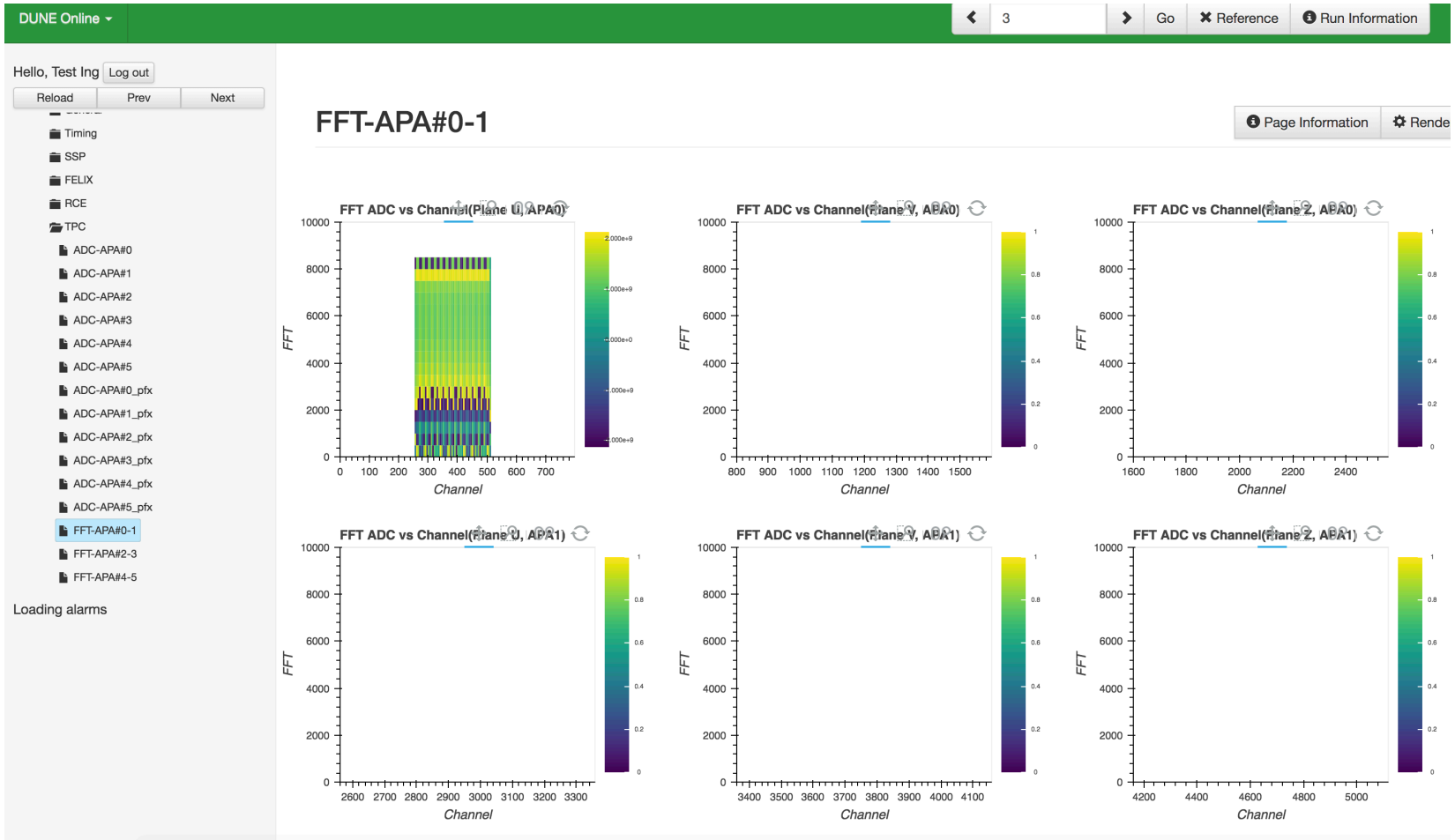
Preliminary plots in Monet: TPC

ADC Mean/RMS vs Channel# (profiled)



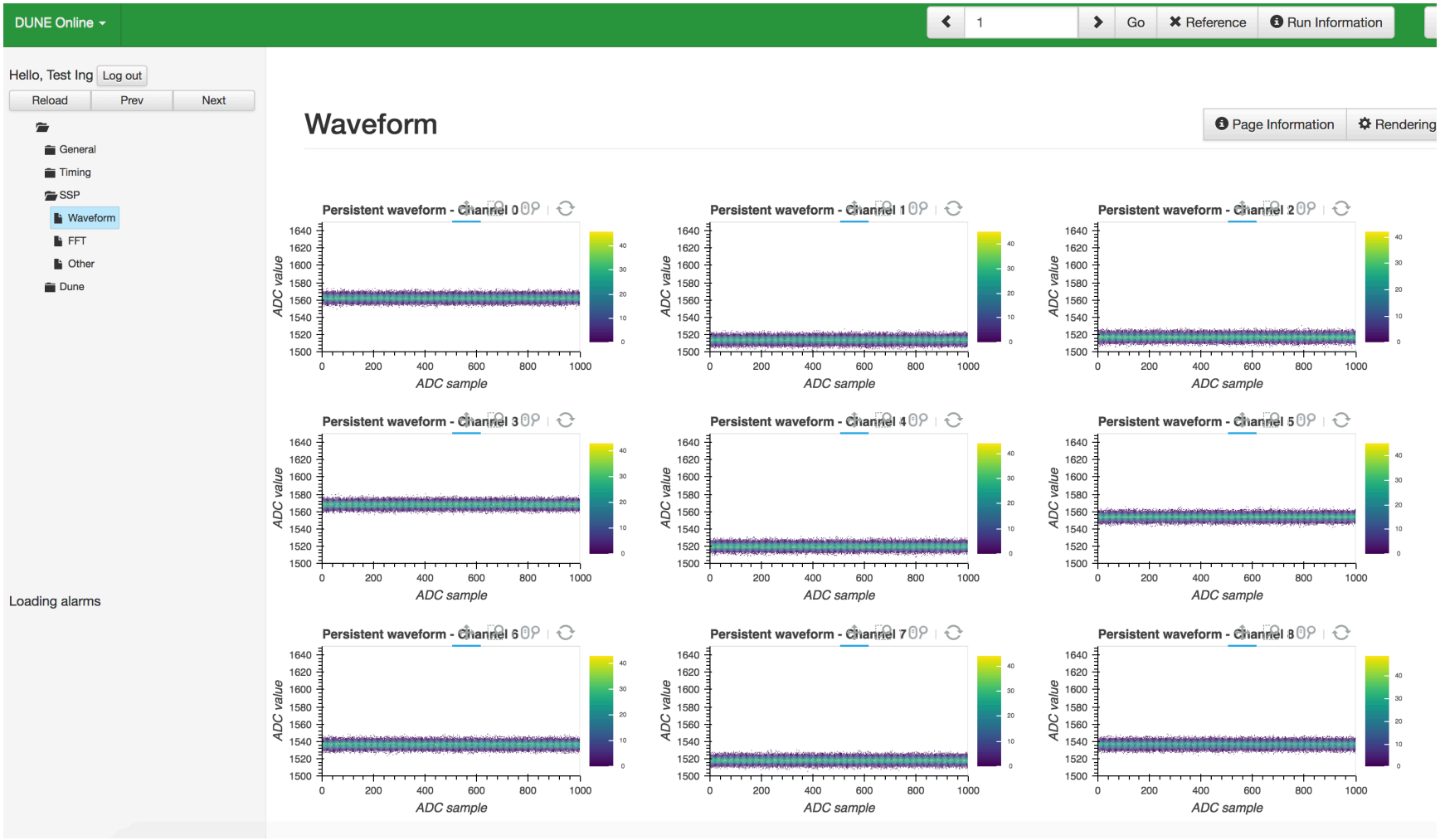
Preliminary plots in Monet: TPC

FFT vs Channel#



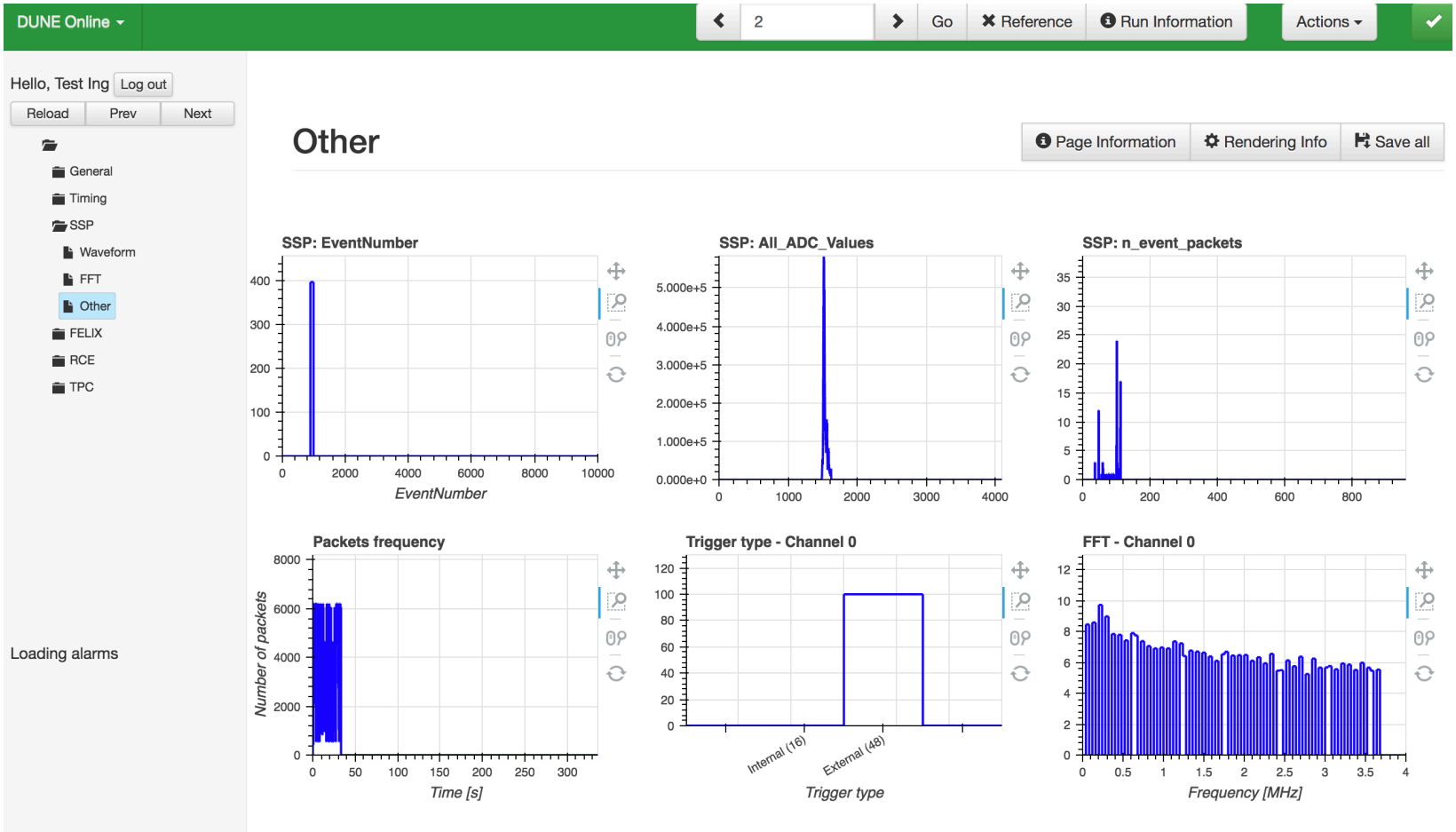
Preliminary plots in Monet: SSP

Persistent waveform

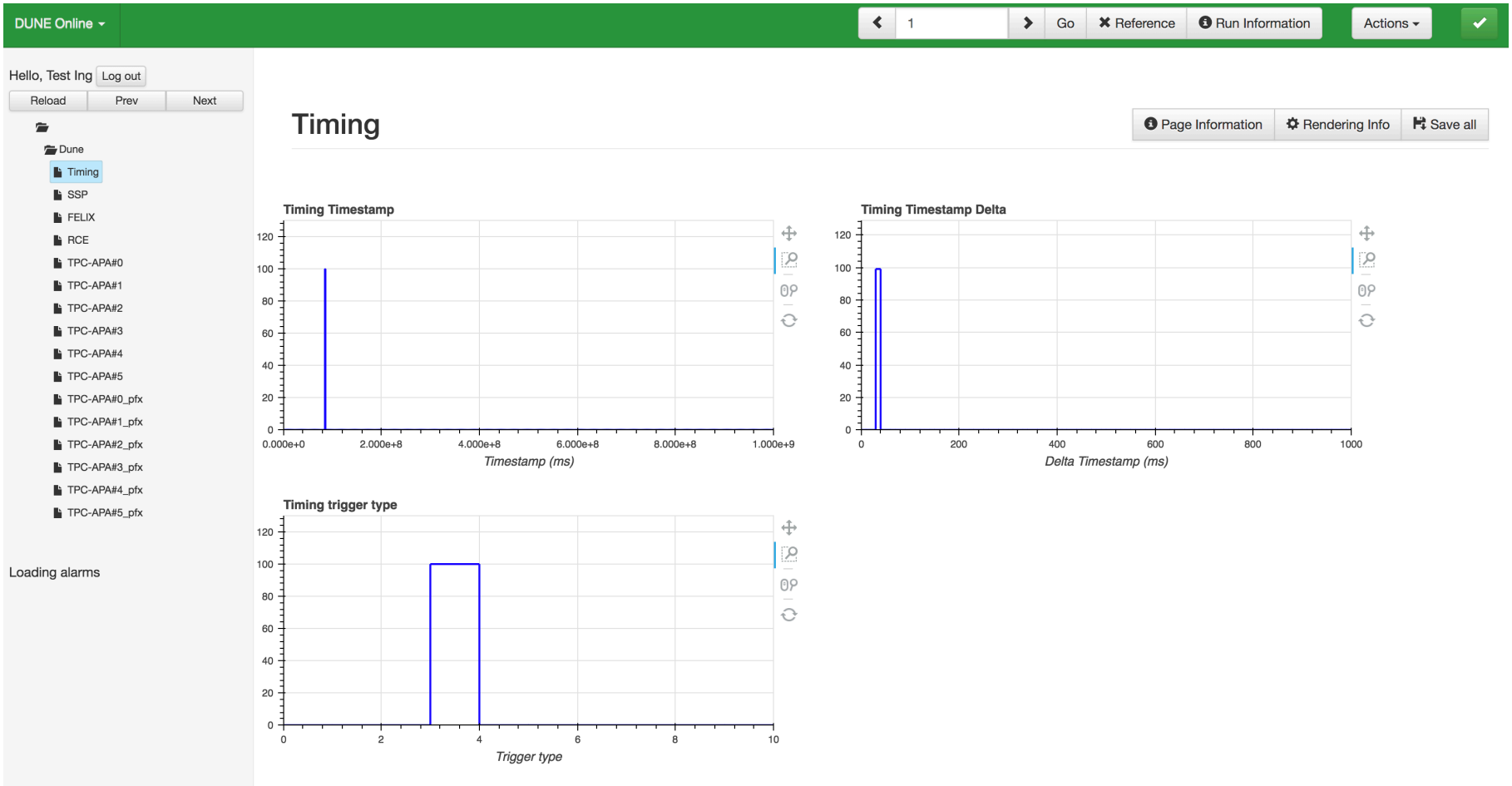


Preliminary plots in Monet: SSP

Event number, ADCs, Npackets, frequency, Trigger type, FFT



Preliminary plots in Monet: Timing



Multi-process running

- The module must be very efficient since the monitoring needs to allow supervision of the run
- Multi-process capability allows faster running and scalability to larger detectors
- Multi-process running was tested with the old set of monitoring modules. We'll update the scripts soon.
- Art doesn't support multi-process within a module, so we chose to implement the multi-process feature in the terminal
  - ./Makefclconfig.sh creates a number of fcl files to be processed
  - ./StartOM.sh launches all the monitoring processes and saves the PIDs.
  - ./StopOM.sh reads the PIDs and kill all the processes.
  - Lastly, the root files of different process have to be merged into a single file.

```
filters:
{
  Nthevt:{
     module_type: Prescaler
     prescaleFactor: 5
     prescaleOffset: 2
    }
}
```

The Prescaler module allows to assign events to different processes.

Things to do

- Will clean up the existing code and add all the produced histograms to Monet.
- Will keep adding more histograms as they are requested or required.
- The channel maps for the RCEs/FELEX and SSPs have to be settled.
- Monet is slow, especially when presenting 2-D histograms. Need to select appropriate bin size to increase the uploading speed. Bugs in Monet? (*Philip Rodrigues* fixed a bug and made Monet faster)
- Need to update the multi-process scripts and add the feature to the DAQ interface
- The shared memory plugin doesn't allow to transfer data across nodes. Need to switch to multi-cast plugin (John Freeman)
- The online monitoring modules can be directly used by offline analysis with very little modification (Fermilab).

Conclusion

- We have a monitoring framework set up for reading the raw data and producing data quality plots during the run.
- We are developing and improving the modules. We'll add more plots as they are required.
- The web display tool is under development and will be the next focus.
- The online monitor is currently being tested alongside the DAQ. Potential issues could be found. Please report bugs by E-mail to the online group.
- Any suggestions are very welcome!