Status of Reading HDF5 Files into LArSoft Jobs

Tom Junk, Jake Calcutt, Barnali Chowdhury, David Adams

DUNE Core Computing Meeting

November 1, 2021

Where we are

- Vertical Drift Coldbox Vertical Slice Test is the source of current HDF5 data
- We had been given software test data earlier in the year in HDF5 format
  - File format was proposed – attributes, group names, and header information were defined
  - Data were placeholders
- Three repositories for defining data formats, in GitHub under DUNE-DAQ:
  - daqdataformats
  - detdataformats
  - detchannelmaps

Where we are

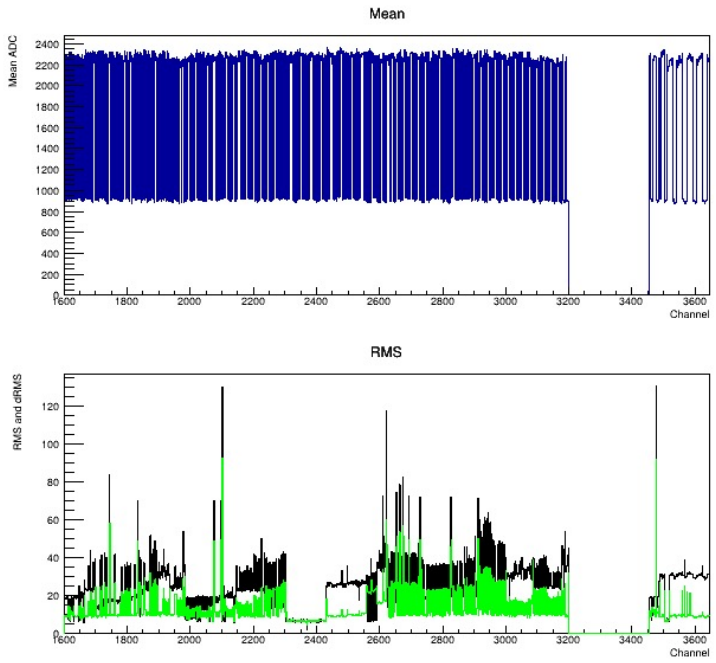
- UPS products made of two of the three repositories
  - dunedaqdataformats
  - dunedetdataformats
- There are technical details with how to handle channel maps, as they depend on both the hardware and the offline channel numbering. What does the version depend on? Who maintains them? (detector folks, but the repo is in DUNE-DAQ).
- Current VD channel maps put in dunetpc
- dune_raw_data now a legacy product, for ProtoDUNE-1, Iceberg, 35t data readin tools.

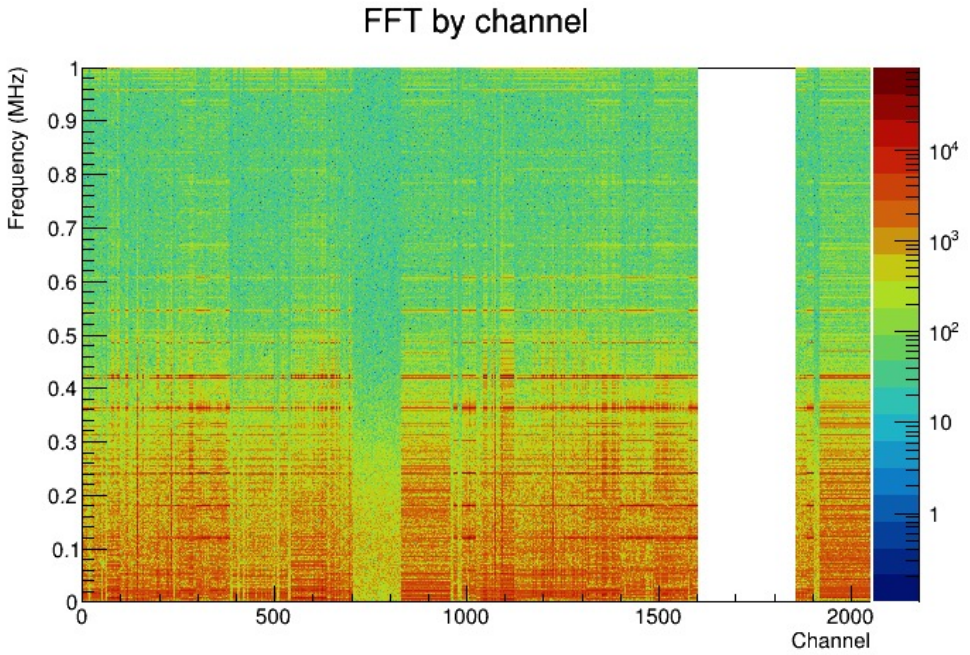
Readin Status

- Jake and Barnali have written:
  - an input source for HDF5 files that reads header info like the run number and the list of trigger records, and leaves the file handle and file name in the art event memory.
  - an input tool that does the actual reading of the data (H5Dread) and deserialization into raw::RawDigits, raw::RDTimeStamps, and raw::RDStatus data products in memory, not in the *art* event. We're doing our own delayed reading functionality.
  - This tool is specific to VD Coldbox Bottom Drift Electronics data
  - Jake hooked up a fcl configuration chain to run dataprep with the new tool and output the three sets of products to an artROOT file.
- Tom made a channel map service which is needed by the deseriliazation as raw::RawDigits and raw::RDTimeStamps have the offline channel number in them.
- Tom ported some gallery scripts to make basic plots for the VD Coldbox people

Success: Software delivered to VD Coldbox team

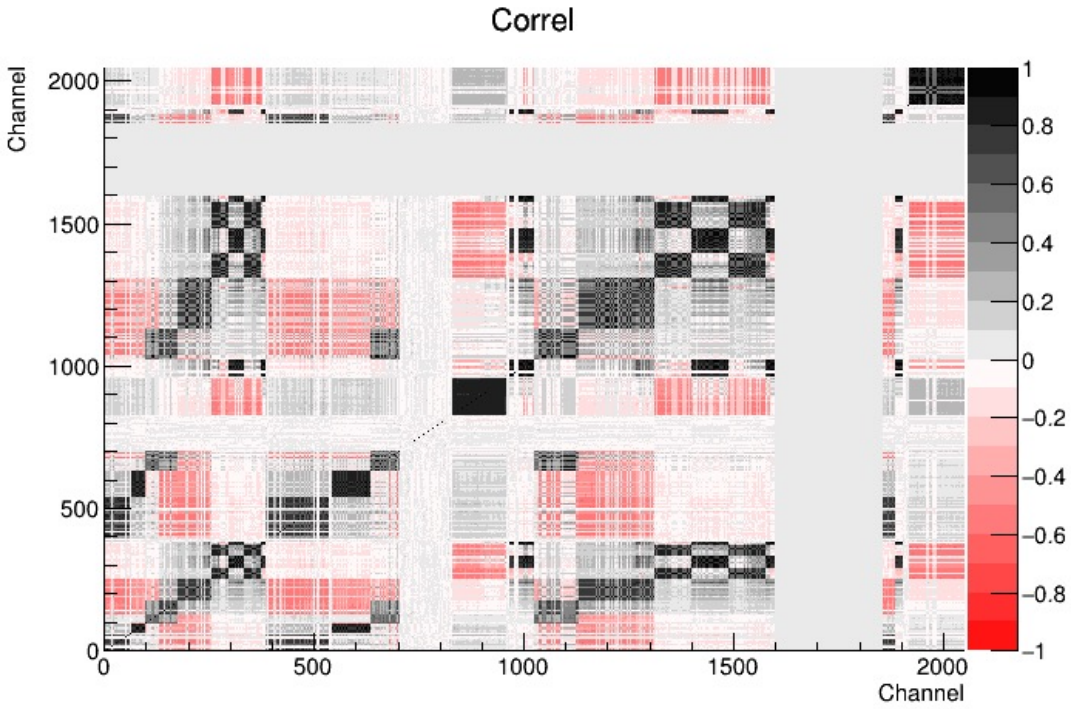
- We missed the weekly release of dunetpc, so we tarred up a test release and gave that, along with scripts, to the detector team.

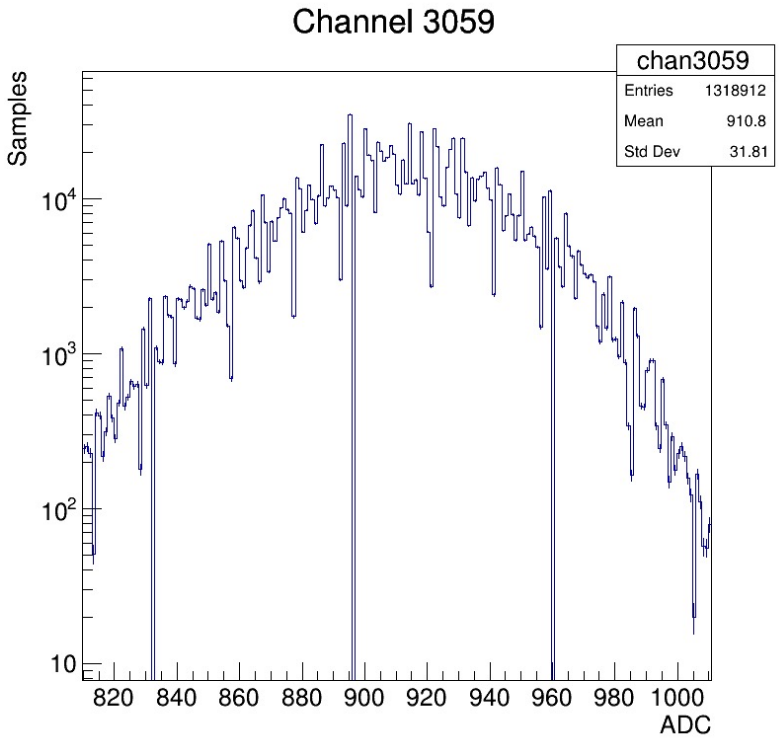




Channel mapping needs some testing and verification

More VD Coldbox plots





Just one of 1792 channels' ADC histograms

Fermilab DUNE

Still to do

- Clean up code – it's messy and doesn't use all the features of daqdataformats / detdataformats
- Make it read in just one APA's worth of data at a time, or for a specified list of APAs, which usually will have just one, but maybe not.
- Currently it reads in all the data on a trigger record into memory
  - fine for cold box data
  - not impossible to handle for ProtoDUNE-2, but we could do better
  - Uses way too much memory for DUNE FD.

More Things to Do

- Photon Detector data readin module
  - For ProtoDUNE-SP, we just had a separate module to decode this stuff.
  - It can use the same HDF5 file handle left in the event by the source
  - We unpacked it all at once – data were much smaller than drift electronics data
- Top Drift Electronics (TDE) – Slavic is working on an unpacker here. Separate trigger and DAQ.
- Unpacking other things
  - trigger
  - CRT
  - beam (this came from an external database for ProtoDUNE-SP)

More Things to Do

- Currently, the *art* input source reads in the entire input file, one Trigger Record at a time
- art's –n option seems to be ignored. Would like it back, especially for testing purposes.
- Deserializer tool should not be opening files – should open the hdf5 file just once, and leave it open while we read it.
- Make sure files get closed.
- Make sure run registry gets populated with things we need, and have tools to read it back offline
- Lots more info:

https://indico.fnal.gov/event/51611/

Detector-Related Things to Do

- Online monitoring
  - separate, standalone effort
- Detector performance plots
- Event display -- we're using the ProtoDUNE-SP one at the moment
- David's dataprep monitoring tools are quite extensive
- They have some ProtoDUNE-SP-specific features. Hopefully easy to re-use a lot of this code. It was re-used for Iceberg successfully.