Data Reduction for protoDUNE/SP TPC

Brett Viren, Xin Qian

Physics Department

BROOKHAVEN NATIONAL LABORATORY

DRA WS

2017 May 15

Outline

Pre-requisite Processing for Data Reduction

Production Processing Schemes with Data Reduction

Status and Work Needed

Caveat and apologies:
only considering the **TPC** data here!

Initial Offline Processing Steps

Expected, high level **TPC** data processing steps, in order:

- **read** Unpack, uncompress and assemble raw *fragments*.
- **ADC** Correct for ADC nonlinearity and stuck bins.
- **noise** Remove excess noise, if any.
- **signal** Signal processing: field+electronics response deconvolution, software filter, signal-ROI selection.
- **(reduction)** **here is natural point to save reduced data**
- **all the rest** All the many reco stages. (Gaussian-hit finding or 3D binned imaging, clustering/tracking, pattern recognition, kinematics, etc, etc, etc.)

  Expected to require about 5$\times$ more CPU than prior steps.

Note: **read, ADC, noise** and **signal** are all required even if no data reduction step is applied.

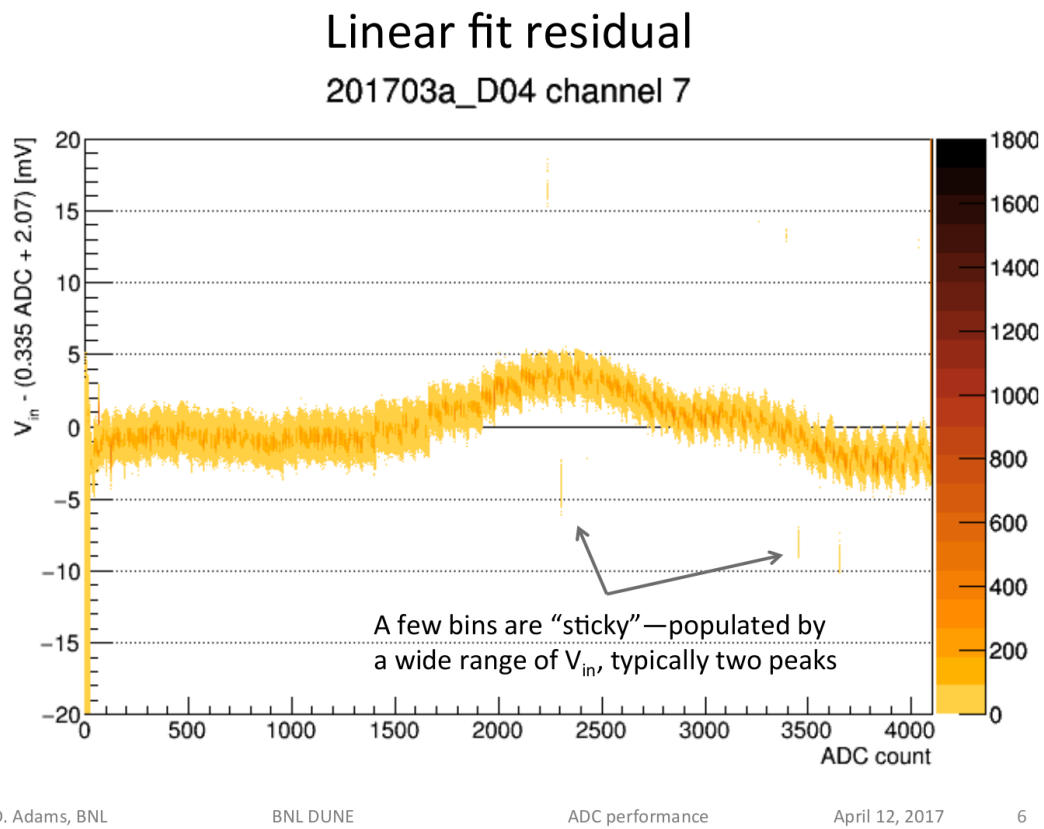
Reading Raw Data

For all protoDUNE/SP DAQ data sources:

- Raw data saved as `artdaq::Fragment` objects
  - **Packed integer arrays** saved in ROOT files.
  - `[array] = [artDAQ` **`header`**`] + [board-specific` **`payload`**`]`
- To unpack the data, need:
  - artDAQ library to unpack header and access payload.
    - → Effort by artDAQ experts (Kurt) to provide this library as "stand alone" as reasonable.
  - "Data Access Library" (DAL) to unpack each type of payload array.
    - → DAQ, "RCE + FELIX fragments use same packing schema."

Seems straight-forward once we have DAL in hand.

TPC ADC corrections



(D. Adams CE Testing work,
eg as presented to DUNE CE task force)

To mitigate ADC "sticky bins" one imagines some algorithm like:

```
if ADC[i] in sticky_list:
  ADC[i] = interpolate(i, ADC)
```

The `sticky_list` is per channel but **hopefully** not a function of time.

Because of oversampling, interpolation should be okay if it's across an isolated stuck bin.

Work ongoing to characterize ADCs, need mitigation code and database.

Noise filtering

- We won't know what excess noise protoDUNE/SP has until it is filled and turned on.
  - pD/SP designed with excess noise mitigation in mind.
  - CE testing may catch early issues.
- We have mostly understand past detectors' excess noise.

  harmonic amplitude at specific frequencies.
  coherent across atomic units of electronics.
  saturation sporadic (eg μBooNE's "chirps").
  catastrophic eg 35t "high noise state"
- We can reuse existing solutions but we must be ready for novel noise problems.

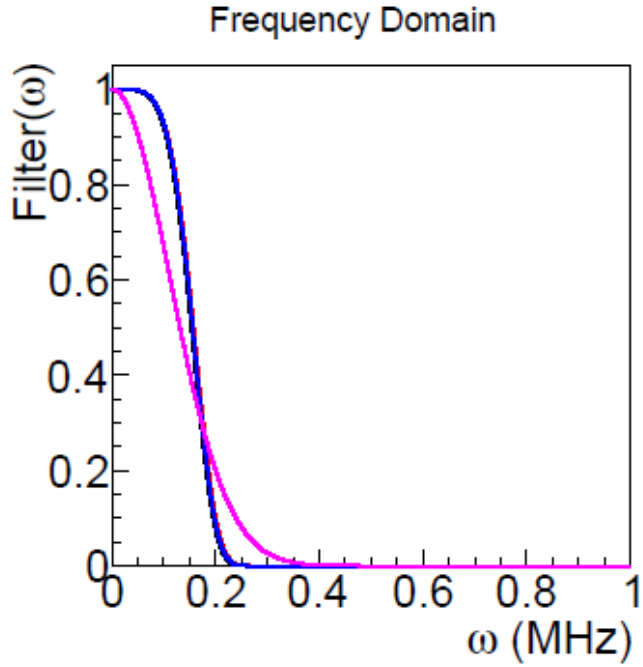
Need a few fast working, dedicated noise hunters on day one!

Signal Processing

The essence:

$$M = R \cdot S_{true} \rightarrow S_{exp} = R^{-1} M$$

Derive expected **S**ignal from **M**easured, given average **R**esponse functions.



Weiner and Gaussian filters.

- 2D deconvolution / matrix inversion with terms in frequency domain (Xin) ("2D" here means channel *vs* tick)
- H/w anti-aliasing filter replaced with two software filters:
  - Wiener maximizes S/N and just for **signal-ROI** selection.
  - Gaussian for **final signal**, preserves charge measure.
- Most of pre-reco CPU needed for the FFTs in this step.
  - Promising new technique may speed this up dramatically (Chao)

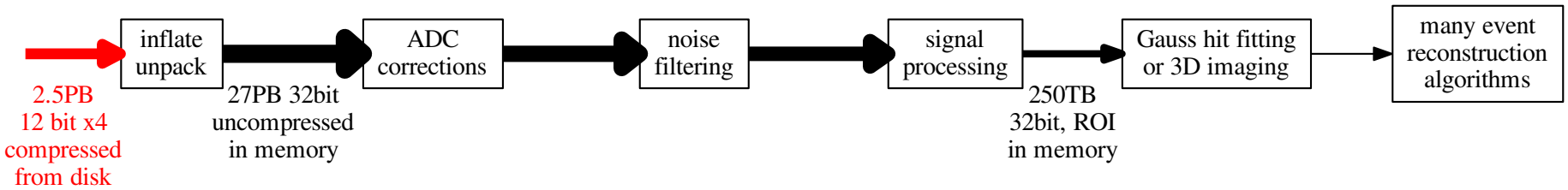
Tested/developed on μBooNE, paper in prep.

Pre-requisite Processing for Data Reduction

Production Processing Schemes with Data Reduction

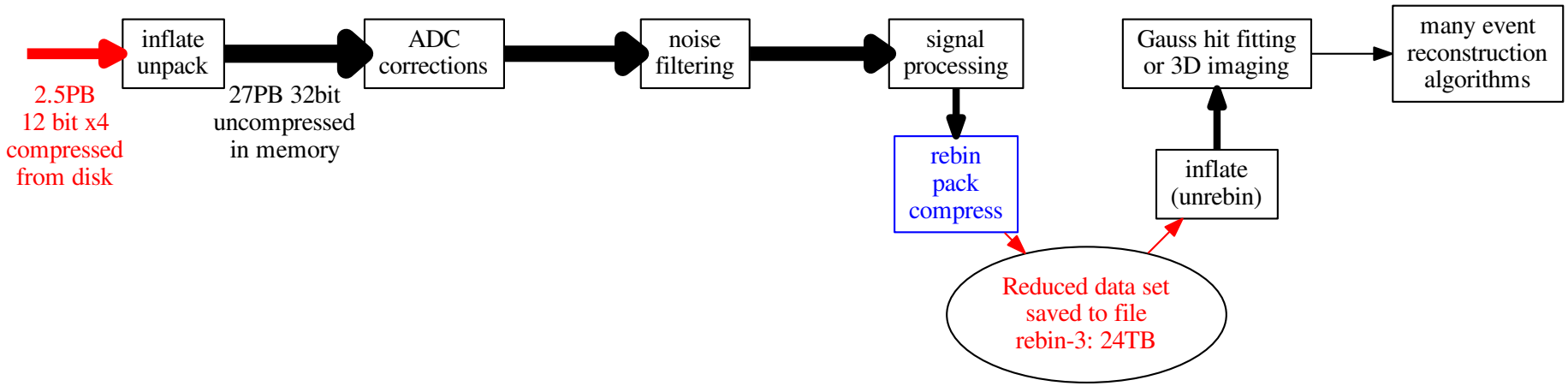
Status and Work Needed

Scheme 1: nominal, monolithic job



- Single, monolithic job. **No reduced signal data saved**
- Raw data is substantially **inflated** after read-in.
  - undoes 4× decompression, 32/12 bit sample inflation
- Note: signal-ROI selection naturally reduces data volume.
  - ROI's consist of 250 TB (in memory).
  - Recent development in ROI selection may improve on this.

Scheme 2: two-stage, reduced intermediate file



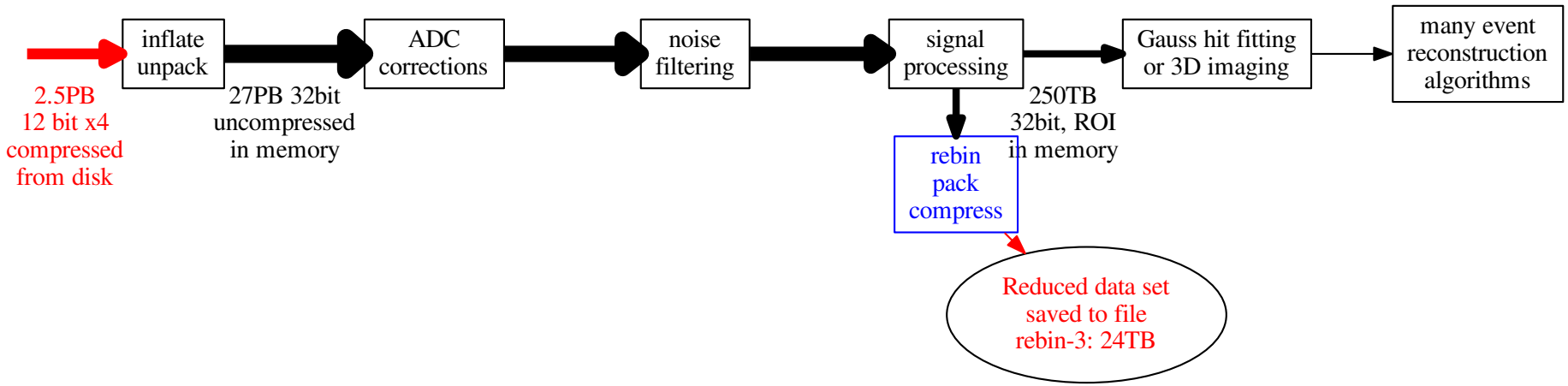
Going further: apply **lossless**[1] rebinning, save to file with 16bit packing and (ROOT) compression.

- **rebin-3: 24 TB** on file, still retains some oversampling
  - → note, a bug fix: used to quote ~~6.5 TB~~ for rebin-3.
- **rebin-4: 19TB** (Nyquist limit) or **no rebin: 70TB**.

Many benefits to having reduced signal-level data set.

[1] This reduction is **lossless** for signal information. Details in backups.

Scheme 3: reduction as side-effect



Some people doubt Mr. Nyquist (and Mr. Viren)

- To assuage them, option 3 saves reduced data only as a **side-effect**.
- Pass on **inflated and oversampled** ROI-level data to reconstruction.
- Reprocessing requires **full 2.5 TB input** and **re-running signal processing**.

No obvious (to me) benefit compared to scheme-2 with no-rebin.

Which scheme?

I propose **scheme #2** (two-stage) in two phases:

1. Begin processing during **commissioning** with **no rebin**.
   - People can use data to try to prove Mr. Nyquist wrong.
   - Immediately start development of tooling for reduced file management.
   - Still enjoy some modest reduction.
   - I think ADC/noise/signal procs will stabilize during this time period.
2. Transition into **operations** with **rebin-3** or **rebin-4**.
   - Should just entail a job configuration change.
   - Downstream reconstruction can see same inflated 32bit and even as unrebinned[2] data objects as in phase 1.

[2] Do we worry about downstream reco code with hard-coded 0.5 μs tick assumed?

Pre-requisite Processing for Data Reduction

Production Processing Schemes with Data Reduction

Status and Work Needed

What's left to do?

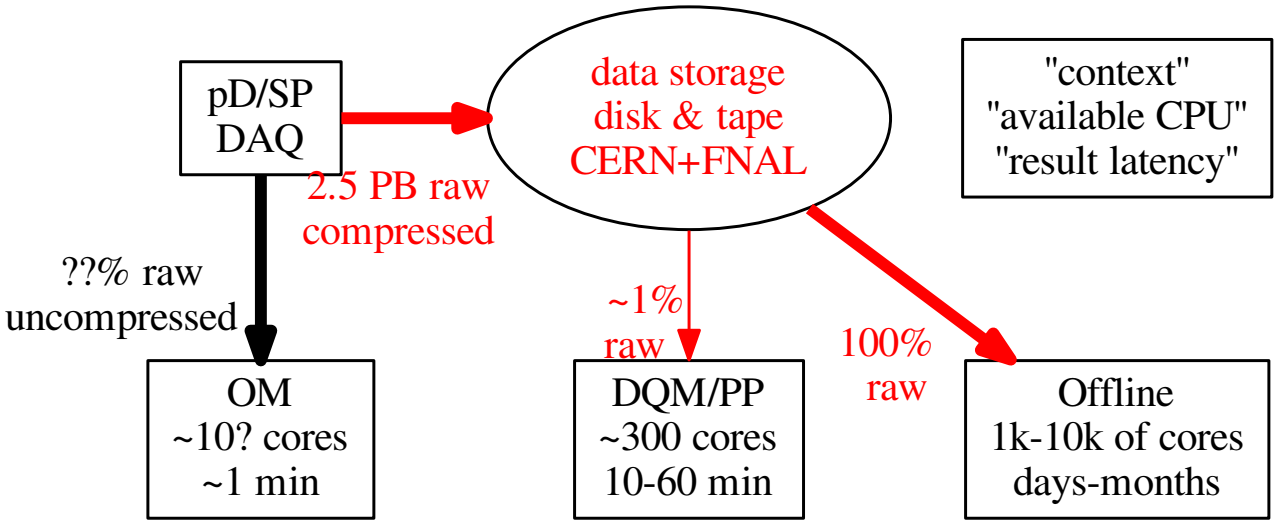
Areas where help is invited are in blue.

- Improved detector simulation for data-reduction validation:
  - Improved detector response part is **done**.
  - Proper **noise** (Jyoti/Milind/Arbin) and **drift** (Hanyu) models: **in progress**.
  - Use new sim to **validate reduction**, understand signal and noise effects
  - Integrate WCT sim components into LArSoft. **design** (Brian, bv)
    - → More in backup slides and Thursday's talk.
- Develop ADC corrections/DB (David Adams, CE testing team)
  - Lots to do, room for new people.
- Noise filtering requires dedicated experts on day 1 of pD/SP data!
- Signal processing
  - Port prototype code into Wire Cell Toolkit: **started** (bv, Xin)
  - Integrate WCT sigproc components into LArSoft. **design** (Brian, bv)
- Need code development for data I/O:
  - read TPC fragments using DAQ's Data Access Library
  - read/write non-TPC fragments untouched
  - rebin, pack and write ROI as reduced TPC fragments (bv)
    - → (the actual reduction code!)
  - read reduced fragments, populate `recob::Wire` objects
- Production batch job integration, file management, etc

FIN

Backup Slides

protoDUNE/SP Data Processing Overview



- **Data**: canonical size: 10 PB. (uncompressed, 12bit packed).
  - If expected noise: 4× compression → **2.5PB raw** (TPC).
  - **All raw data is saved** at CERN and FNAL (at least).
- **Online Monitoring**: very low latency, minimal processing, large sample.
- **Data Quality Monitoring / Prompt Processing**: low latency, moderate processing, small sample.
- **Offline**: substantial latency, full processing, full sample.

But wait, reduction is lossless? Seriously?

Yes

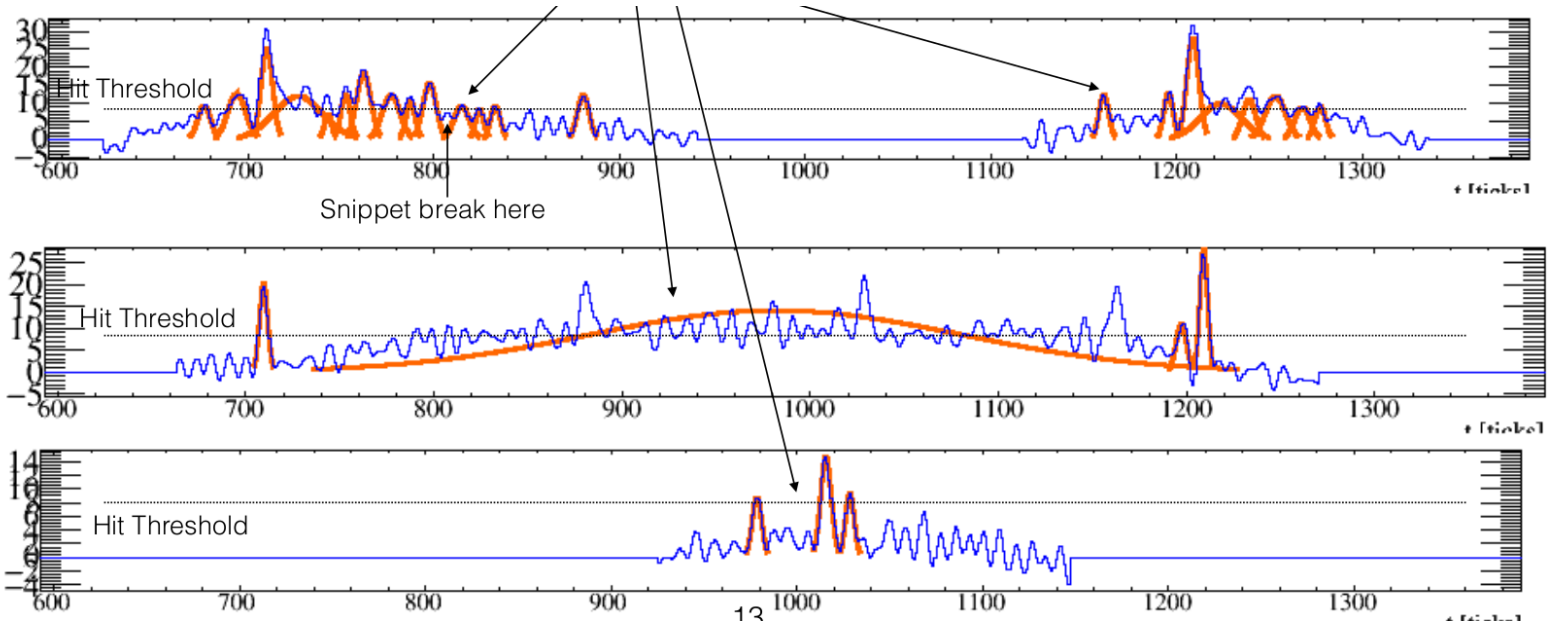
The power of the reduction comes from:

- Signal-ROI,
- Rebinning,
- Packing and
- ROOT compression (eg, ZLIB/LZMA)

The first two are not strictly information-lossless (you can not reconstruct the exact input bit-stream from the output).

But, they are lossless **for signal information** $\longrightarrow$

Signal Regions of Interest

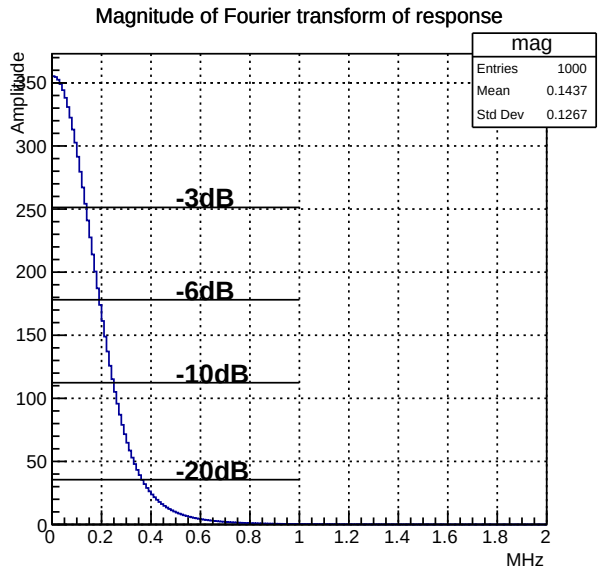


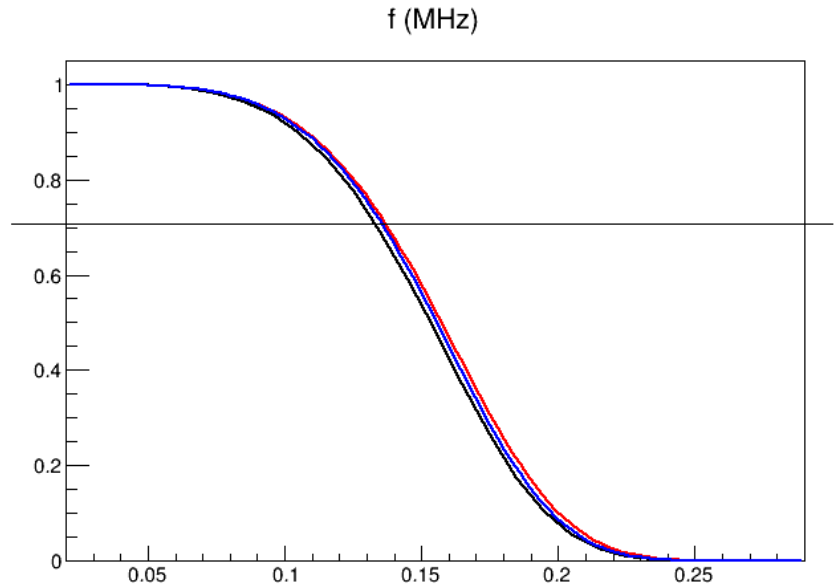
("Gauss Hit Finder Proposed Changes", Tracy Usher, Feb 2, 2016)
(for illustration, ignore orange)

- Signal-ROI concept needed to obtain highest signal efficiency and purity.
- Downstream processing (eg, Gauss Hits) only considers waveform portions inside an ROI
- **By construction**, waveforms segments outside signal-ROI are noise.

NB: special analyses that require the noise can always go back to the raw data and likely would not want any of the previous steps so they are not "clients" at this point.

Filters, Rebinning and Nyquist





2μs hardware anti-alias and software low-pass filters.

hardware signal above 1 MHz strongly suppressed
$\Rightarrow$ hardware sampling at 2MHz satisfies Nyquist theorem.

software signal above 0.25 MHz strongly suppressed to remove high frequency noise
$\Rightarrow$ software resampling at 0.5 MHz satisfies Nyquist theorem.

After software filter, re**sampling** at 0.5MHz produces the **exact same information content** as sampling at 2MHZ (rebin-4 is okay).
With an abundance of caution: propose re**binning** by 3$\times$ instead of 4$\times$.

Packing and Compression

What is the sample resolution?

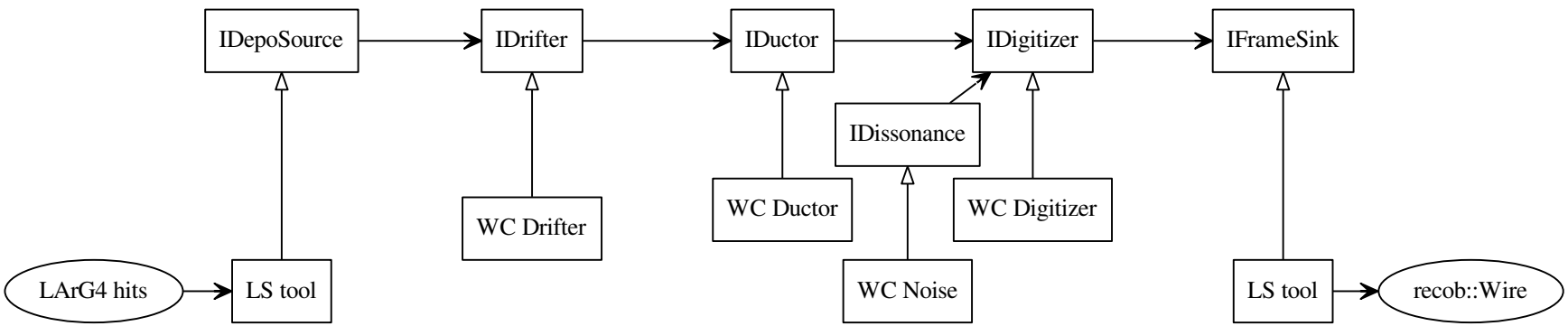
- Nominal 12 bit ADC, but has has $\sim$ 10.5 ENOB
- Intrinsic electronics noise is 1-2 ADC.
  - $\Rightarrow$ So, our samples have no more than 12 bits of information.
- During processing 32 bit `float`s are used.
- **Choose 16 bit output packing**.
  - Actually **inflated compared to input** but convenient as `short int` can be used, and less concern of memory alignment $\Rightarrow$ no info loss at this step.
  - Can go further and apply "**dynamic dynamic range**" to optimally truncate to 16 bits by first scaling samples on a per-channel or per-trigger basis.

What about compression?

- Rely on ROOT's default compression which is truly info-lossless.
- Because of the huge data reduction, can compress at "level 9" and still require **in**significant CPU (compared to, eg, signal processing)

Wire Cell Toolkit / LArSoft Integration

- `larwirecell` layer between WCT and LS
- WCT heavily uses concept of **interfaces** and **dynamic components**, similar to the new **Art Tool** idea.
- Integrate via Art Tool facade at any "touching point" between LS and WCT components.
  - → basic model: "LArSoft uses Wire Cell"
- Tool handles data conversion at input/output.
- A WCT app (eg, `Fourdee` sim below) also with Tool facade handle component assembly and configuration.



WCT Simulation App with LS Tool facades for input/output components.