Drift Signal Simulation and Data Reduction

(double feature)

Brett Viren

Physics Department

BROOKHAVEN NATIONAL LABORATORY

FD Sim/Reco
DUNE Collaboration Meeting
24 Jan 2017

Outline

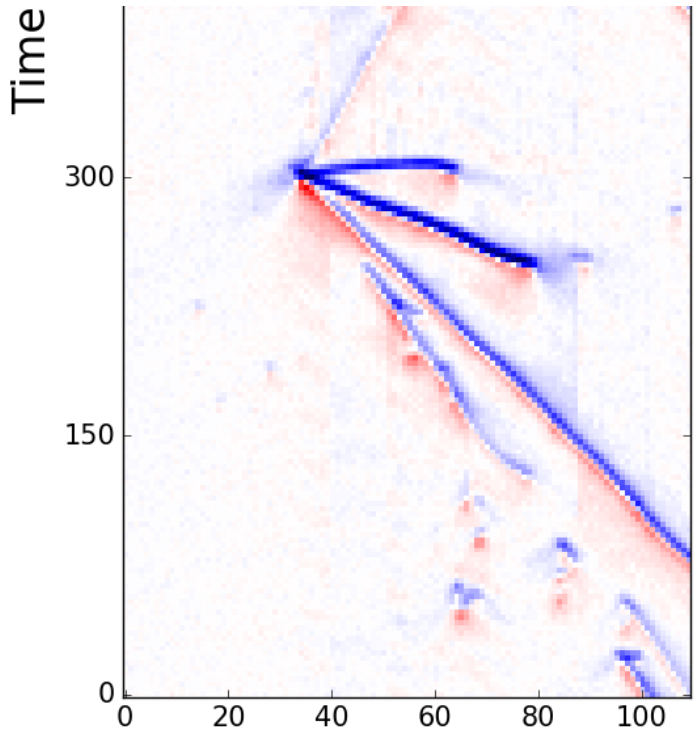
Full TPC Drift and Signal Simulation

Proposal for Initial Data Reduction Process

Motivation for a Better Drift Signal Simulation

- MicroBooNE observes long range induction effects and other potentially due to intra-wire field variations.
- Standard simulation (and deconvolution) does not take these into effect.
- Drift signal sim very tied to nature of field calculations.
- RAM issues exacerbated going to higher-rez fields

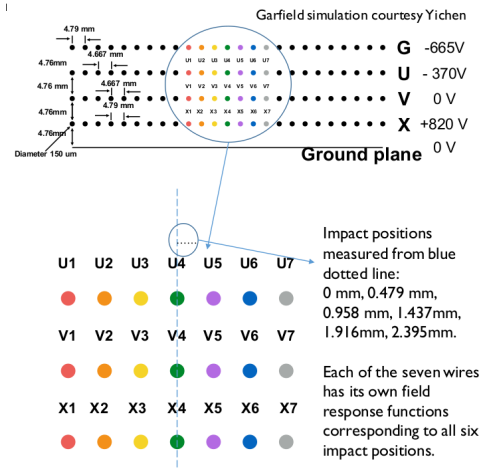
$\rightarrow$ need a fresh start

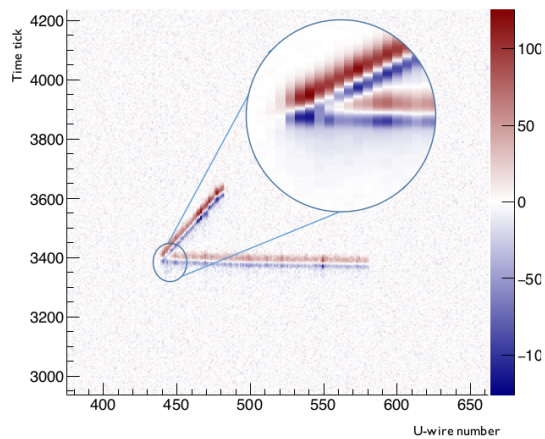


Zoom in on raw MicroBooNE event, induction plane.
(figure from Brooke and Chao.)

Initial Simulation Improvements - Xiaoyue

- Uses (2D) field responses which are calculated:
  - across **7 nearest wires** (instead of 1)
  - at paths spaced by **1/10th the wire pitch** (instead of full pitch)
- Truncated 3σ Gaussian diffusion "cloud" after drifting.
  - Longitudinal & transverse diffusion coefficients.
  - Charge elements sampled to weight field response drift paths.
- Some long-range induction effects start to be reproduced.



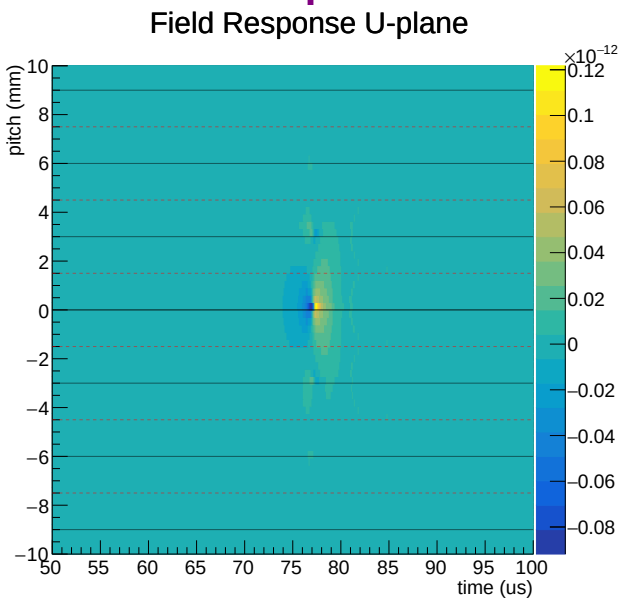


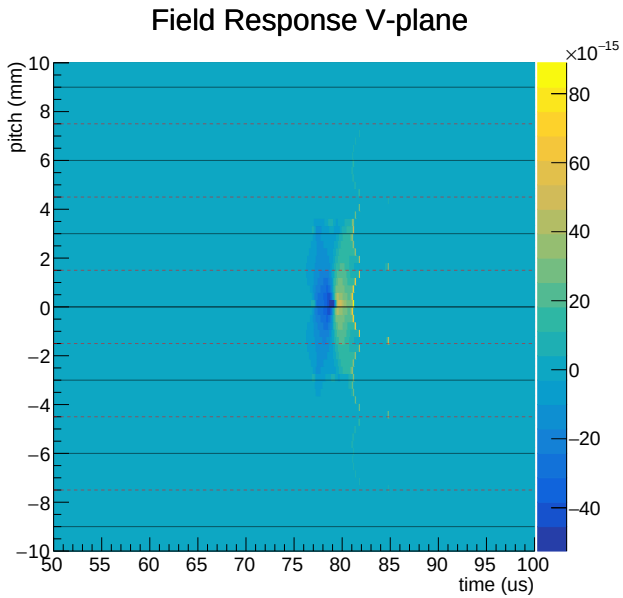
From Xiaoyue's report last collab meeting.

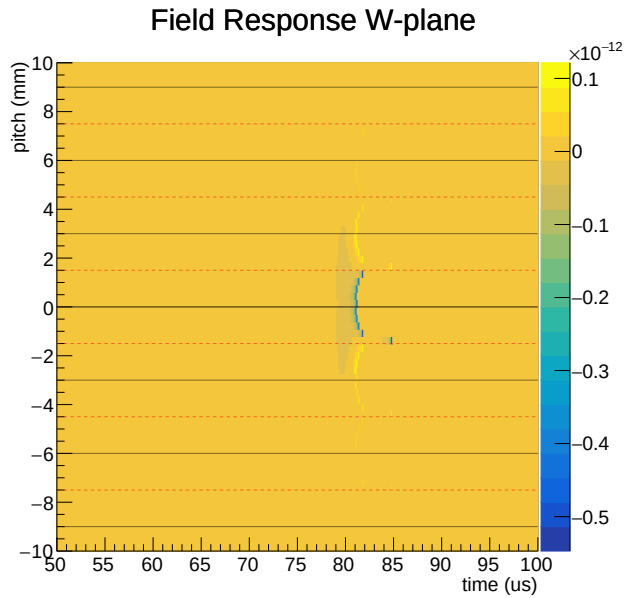
Next Iteration

- Conceptually the same approach as Xiaoyue.
- Retain 1/10th pitch path spacing
- Support "arbitrary" input fields response functions
- New convolution algorithm to minimize RAM usage.
  - Otherwise same basic optimization strategy as Xiaoyue.
- Implemented in Wire Cell Toolkit (instead of prototype) to facilitate integration into LArSoft.

Field Responses



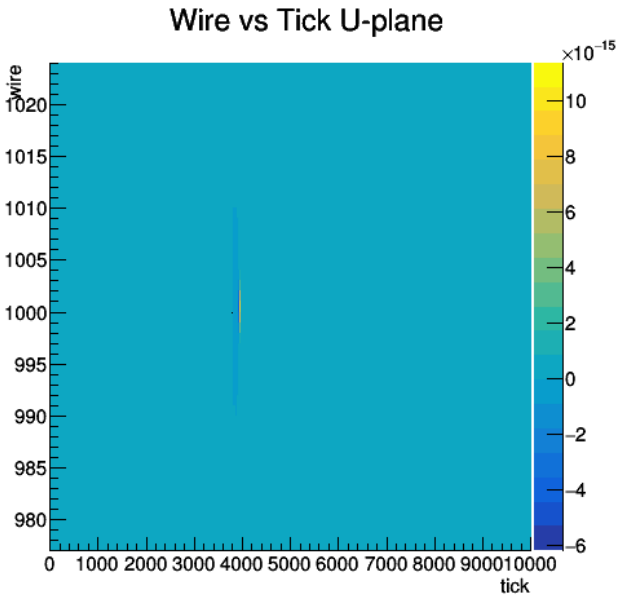


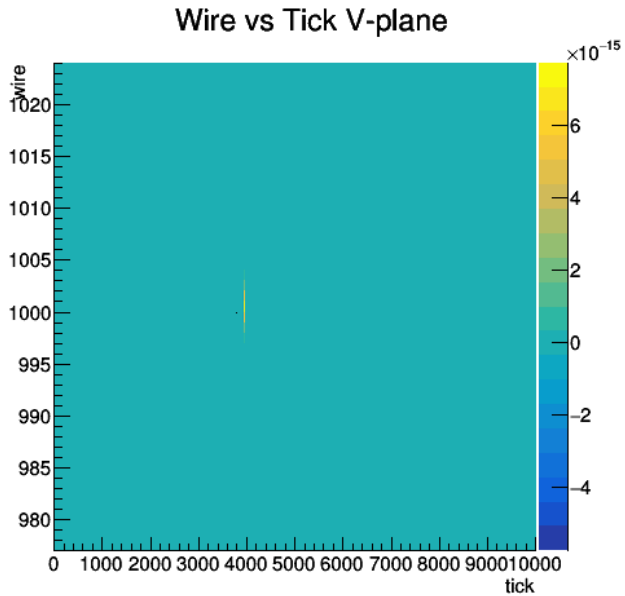


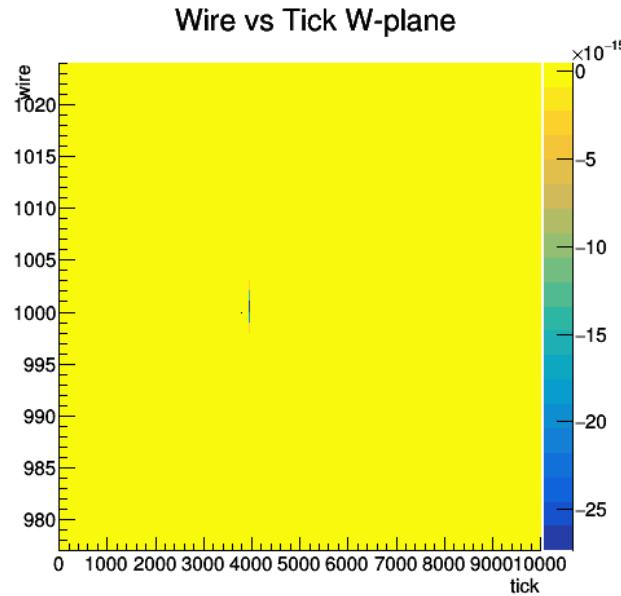
- Garfield 2D fields (Yichen)
  - Will apply BEM/LARF 3D fields in future
- Data-driven
  - Different input field data can be accommodated
  - Some restrictions (2D, certain impact position symmetries).
- Currently:
  - 21 wire regions/plane (instead of 7)
  - 0.1 μs sampling, 1/10th pitch impact positions.

Example Event - Point Source

Preliminary, no electronics response, simple drift.



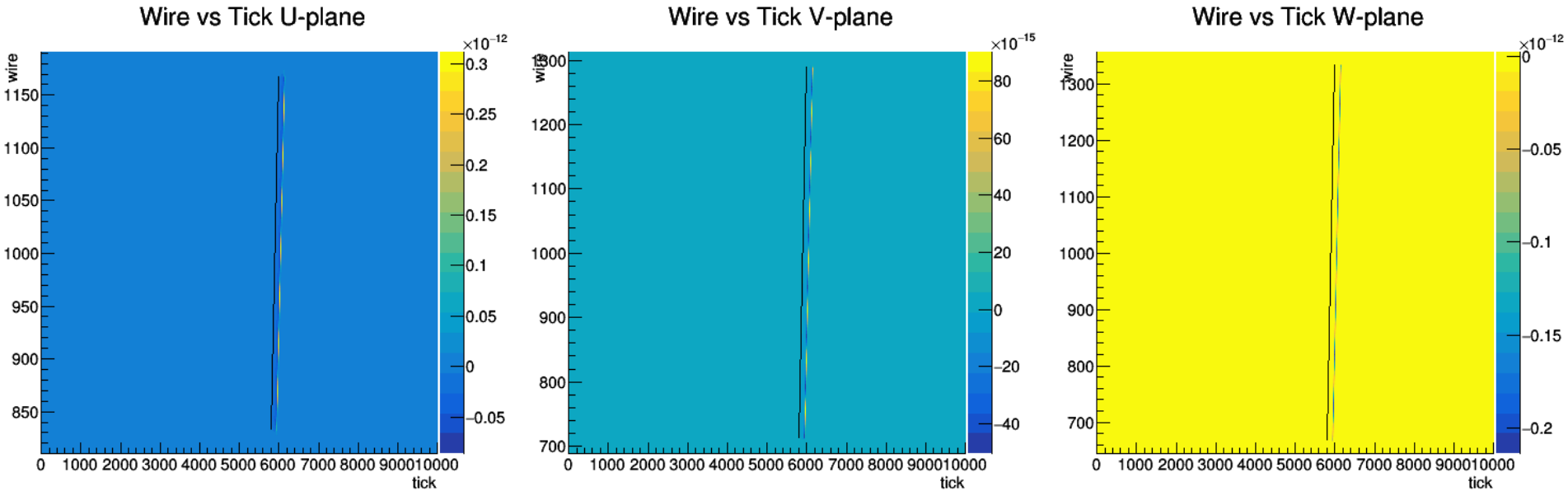




- Short line 1.5 m from plane.

Example Event - Isochronous Line Source

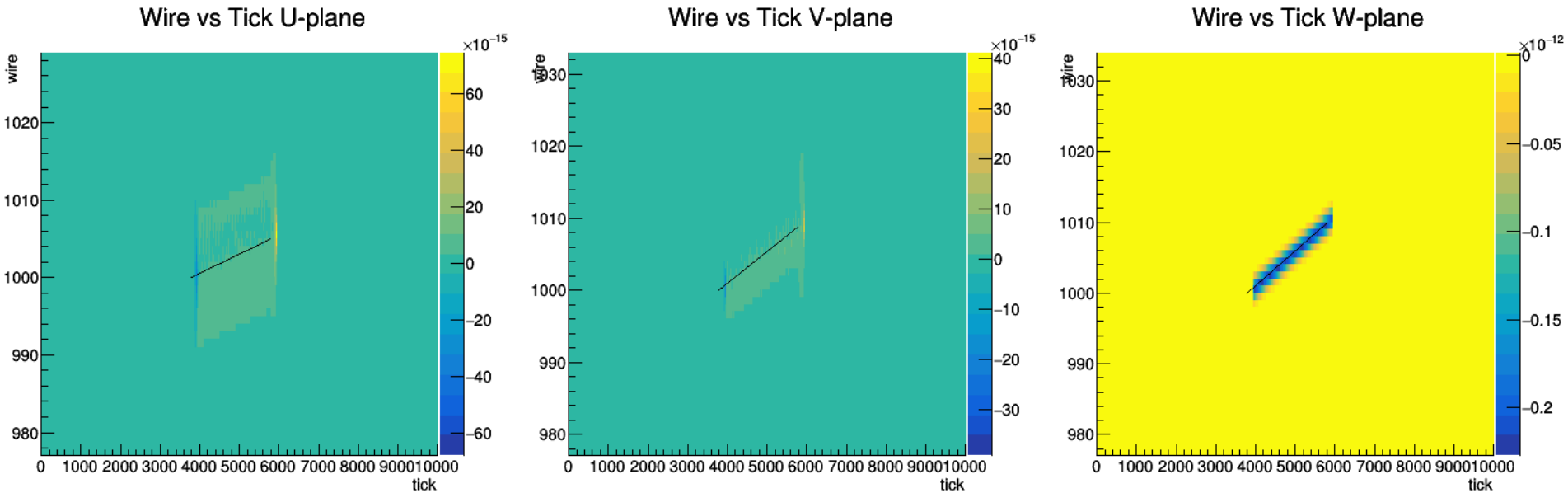
Preliminary, no electronics response, simple drift.



- 100 μs time span, 2 m Z-span.

Example Event - Prolonged Line Source

Preliminary, no electronics response, simple drift.



- Line spans 10 wires in Z, 1 meter in X (drift axis)

Remaining Work for Drift Signal Simulation

- More basic validation still to do:
  - Fold in electronics response.
  - Deal with FFT wrap-around.
- Some performance issues need tuning:
  - 25s per example event including plots.
  - Using $<$ 2 GB RAM, maybe can do better.
- Low level functions are done but need:
  - proper Wire Cell Interface layer packaging,
  - include some batteries:
    - drift/diffusion/absorption (uniform and with space charge)
    - noise models (uBoone model and M.Diwan calculation),
  - user configuration layer.
- LArSoft integration via Art Service Interface
  - Work with D. Adams and other LS experts.

Full TPC Drift and Signal Simulation

Proposal for Initial Data Reduction Process

Reduction Proposal

- Run a **data reduction process** soon after raw data is saved.
  - Algorithms (noise & signal processing) must run somewhere in any case.
  - Strategy: "do no harm to information content".
- Save reduced data and **still save full raw data to tape!**
  - Eg, for understanding detector during commissioning.
  - Any special analysis can also still run on full raw data.
- Later, improve reduction? $\Rightarrow$ re-run on full raw data.

Some uses for the **highly reduced** data set:

- **Efficiently feed production** jobs ("keep up" and re-re-re-processing).
- **Easy sharing of fresh data** to look for/at **emerging detector issues**.
- Tidy samples **facilitate reco development**.
- **Everyone can have full copy** of all signal waveforms on their workstation!
- Maybe facilitate some **public outreach / data sharing**?

Many technical details are in DocDB 2089

Reduction Methods, Options and Evaluation

evaluation	use a few uBoone raw events (`ADC` tier)
scaling	protoDUNE has $\sim 2\times$ channels of uBoone
noise filter	reduces entropy, helps compression (`SUB` tier)
	$\rightarrow$ if all goes well, emulates expected protoDUNE data
ROI selection	keep only **signal regions of interest** (`ROI` tier)
resample	sum N neighboring bins to **remove oversampling** (`RBn` tiers)
	$\rightarrow$ but do not violate Nyquist theorem!
truncation	32 bit float working samples to 16 bit int (for `ROI` or `RBn`)
packing	12 (for `ADC`/`SUB`), 16 or 32 bit packing into 32 bit output arrays.
compression	apply ROOT compression (Z level of 0-9)

Look at **reduction factor** and **compression time** as a function of all meaningful combinations of above options $\longrightarrow$

Reduction Options and Efficacy

tier	bits	Z-lvl	Z-time	factor	2.5 PB
ADC	12	5	6.8 sec	**4.2**	0.6 PB
ADC	12	9	**4.3 min**	5.5	0.5 PB
SUB	12	5	5.6 sec	6.0	0.4 PB
SUB	12	9	**3.6 min**	8.0	0.3 PB
ROI	16	9	0.5 sec	150	17 TB
RB2	16	9	0.3 sec	260	9.6 TB
RB3	16	9	0.2 sec	415	**6.0 TB**
RB4	16	9	0.1 sec	540	4.6 TB

(just a summary, see note for more)

- Noise filtering and signal processing takes **4.5 min/trig** (6 APA)
  - → Z-time above is scaled to 6 APA protoDUNE channel count.
- 12 bits means fully packing 8 samples across 3 words (32 bit int).
- Can **achieve expected factor of 4 raw data compression** at uBoone noise level.
  - → expect factor 6-8 raw data compression if low-noise, but Z-time grows fast!
- Z-time not significant once some reduction is included.
- This **only counts the TPC data**, other data sizes may become significant!
- RB4 rebins to exactly the Nyquist limit, **RB3 is the safe recomendation**.

Implications for Reco

Validate Gaussian Hit Finder is okay with rebinning.

- **Must be okay or Nyquist is wrong!**
- We do better than Nyquist by summing instead of resampling.
- Simple Gaussian fit check (thanks Xin) $\longrightarrow$
  - $\mu = 10, \sigma = 2$, rebin: 1, 2, 4, 8$\times$
  - **Recommend RB3** as safe.
- Gaussian hit finder "erases" any signs of reduction for downstream algorithms.

RBn	const	μ	σ
1	10.0	10.0	2.00
2	19.8	10.0	2.02
4	38.5	10.0	2.08
8	68.9	10.0	2.31

Need modest development for:

- **Art input module** to read in reduced data
  - $\rightarrow$ needed for raw data in any case, reduced data will be similarly packed
- **Art service interface** to the reduction code
  - $\rightarrow$ "keep up" process uses either plain-art or stand-alone to conserve RAM.
  - $\rightarrow$ will run in LArSoft for applying to simulated data.

Summary

simulation Technical progress on implementation for an **improved drift signal simulation** which allows for long-range induction effects and intra-wire field variations has been made

reduction A **data reduction process** can give us a much smaller data set while keeping all useful signal. It can be used by the rest of the processing chain, enable easy data sharing and can allow all collaborators their own copy of the entire set of protoDUNE signals.

$\rightarrow$ later, need to understand how these ideas might apply to DUNE FD.