DUNE Analysis Workflows

Tom Junk

ROOT Workshop

May 11, 2022

About DUNE

- 1150 authors; 1392 collaborators (constantly changing)
- Far Detector (4 modules x 10 kt fiducial each) – first FD module will come first, circa 2029. Beam and near detectors come later.
- Near Detectors (ND-LAr, TMS, ND-GAr, SAND)
- Prototypes
  - ProtoDUNE-SP
  - ProtoDUNE-DP
  - ProtoDUNE-2-HD
  - ProtoDUNE-2-VD
  - Coldboxes for each ProtoDUNE
  - ICEBERG
  - Beam Interfaces/Simulation

DUNE Software: *art* and LArSoft

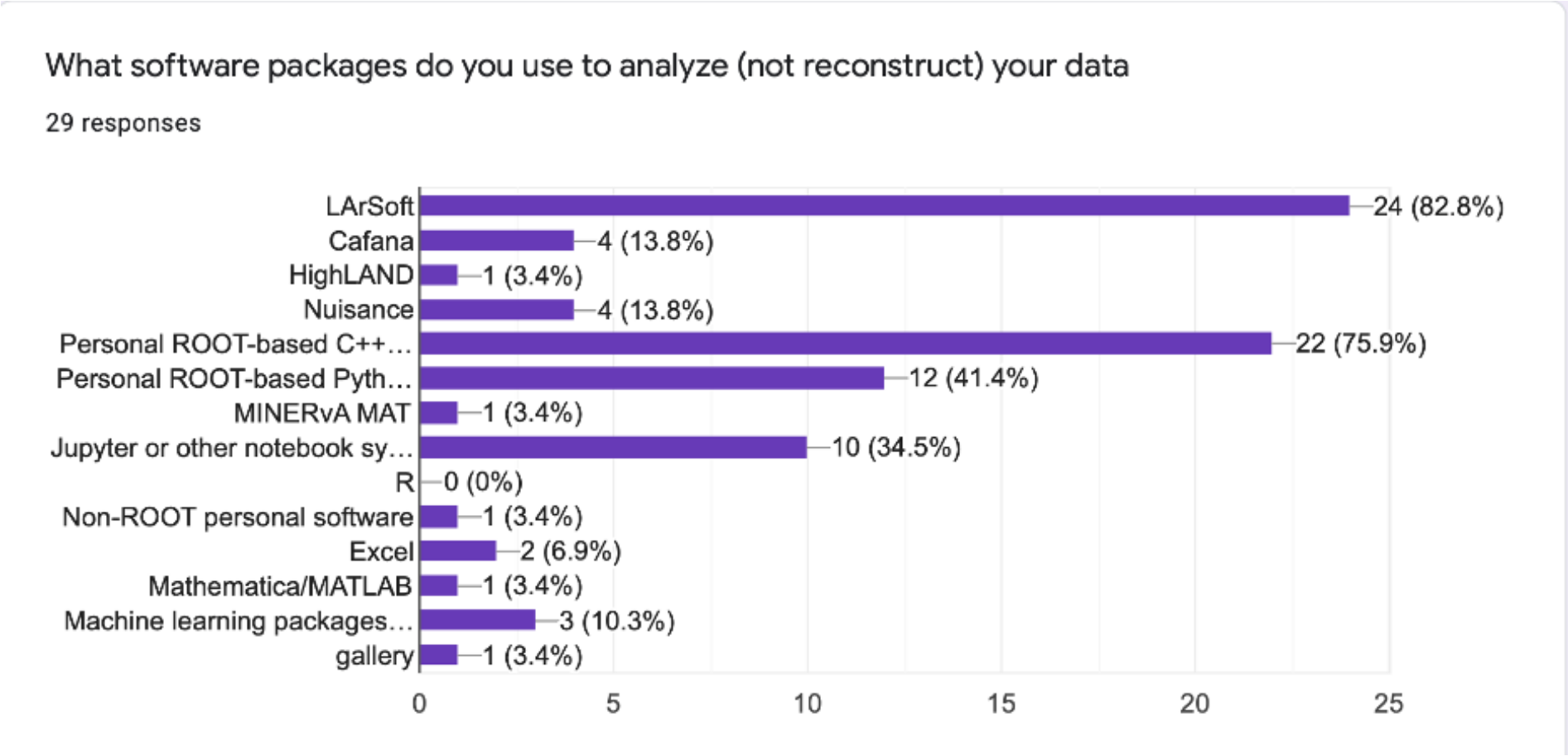
- For the Far Detector, the ProtoDUNEs and ICEBERG, we use *art* and LArSoft-based software, which makes extensive use of ROOT functionality.
- The *art* team has refactored its software stack to isolate ROOT I/O to specific packages.
- *art* was forked from CMSSW as a "simpler" framework about a decade ago. C. Green et al., *J.Phys.Conf.Ser.* 396 (2012) 022020
- artdaq as a separate product which makes *art*-formatted ROOT files. artdaq was in heavy use on DUNE up to 2021
- Experiment software makes free use of ROOT functions.
- Common workflow: *art* jobs simulate and reconstruct data, and make analysis trees. Users analyze trees to make plots for publication.

Not Everthing on DUNE is *art*-based

- Near Detector uses edep-sim (see Clark McGrew's GitHub page), which is an interface to GEANT4 that uses ROOT TTrees as output. And python code for detector response simulation and reconstruction
- Beam simulations have a direct interface to GEANT4 and use ROOT TTrees.
- CAF (from NOvA) files also have ROOT TTrees in them.
- Some ML training work uses HDF5 and custom formats (more on this later).

DUNE Analysis User Survey

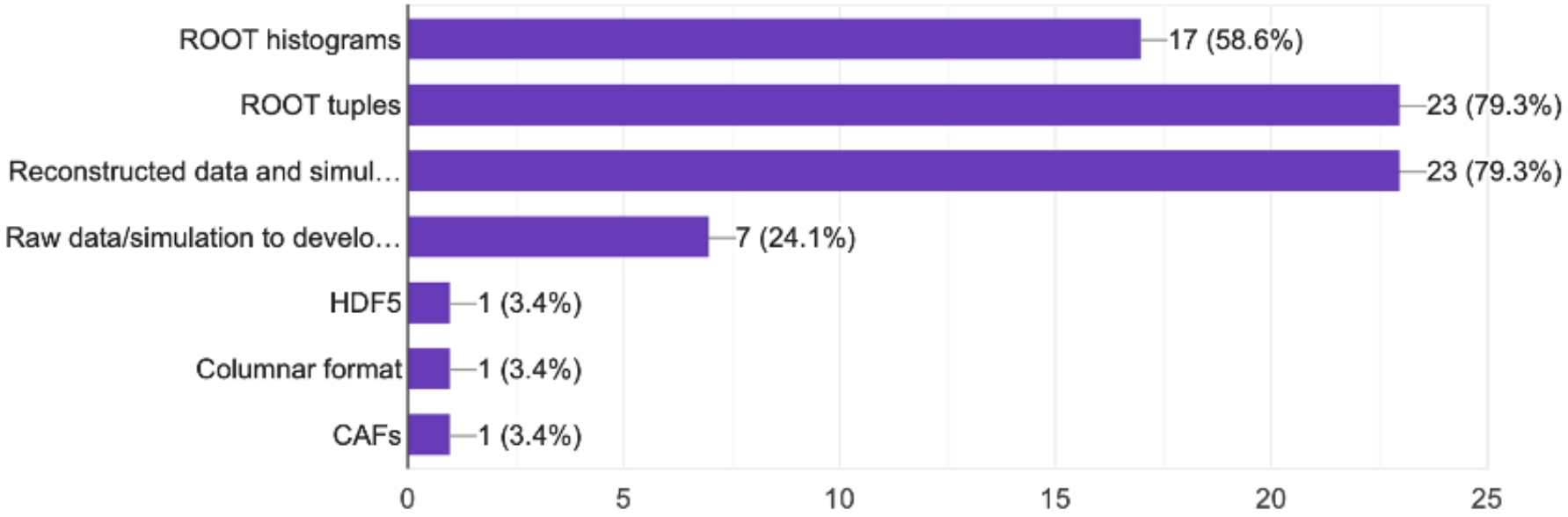
Survey conducted February, 2022. Many thanks to Heidi Schellman for running it!



DUNE Analysis User Survey

What data format do you use when running your analysis code (not sample creation)

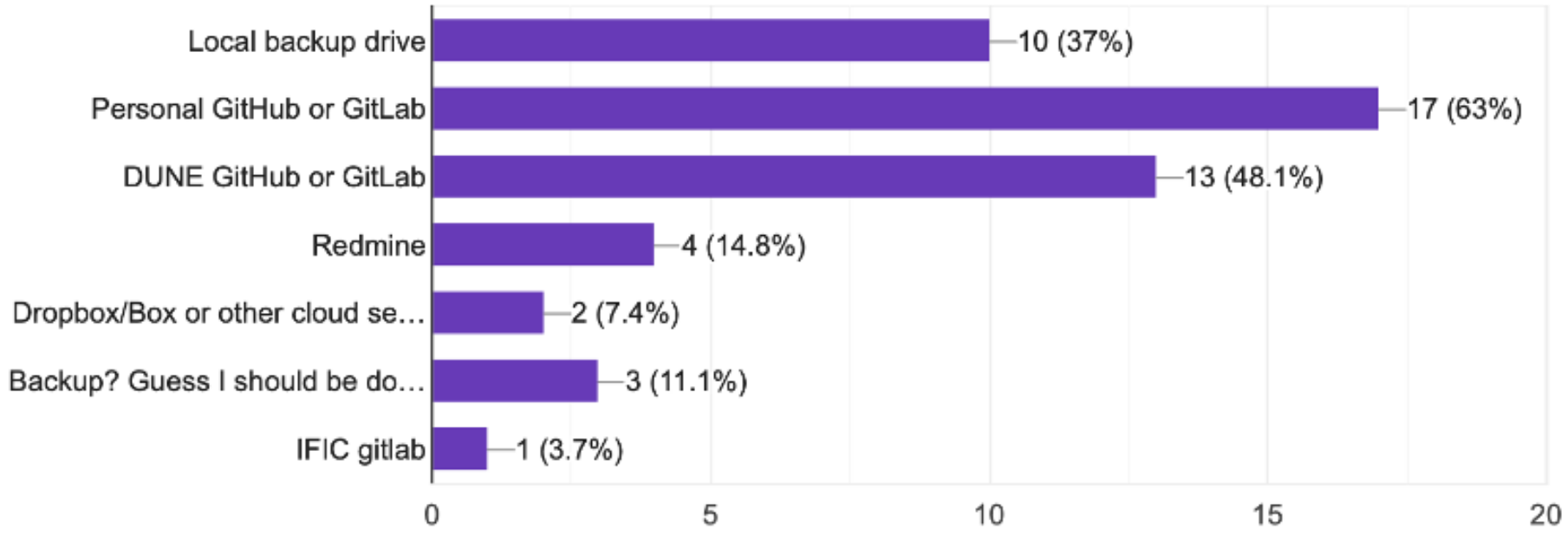
29 responses



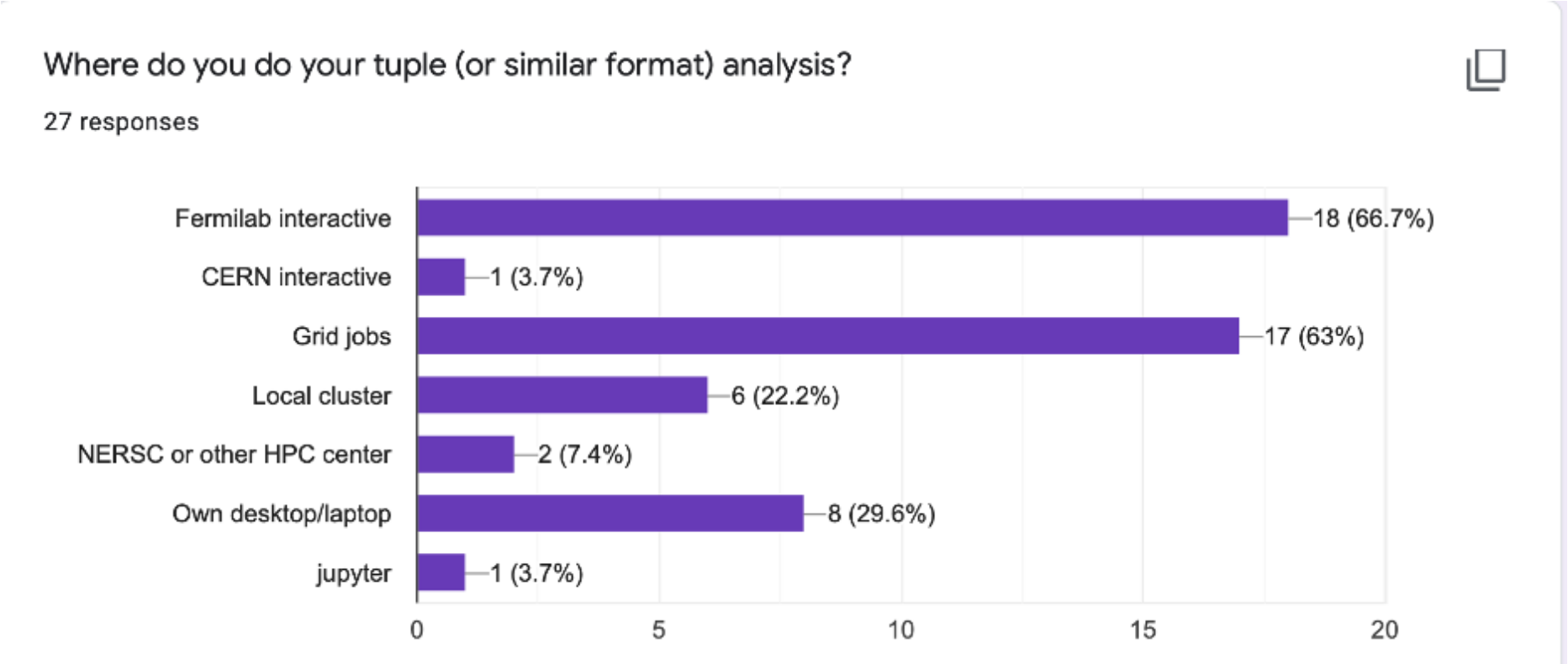
DUNE Analysis User Survey

Where do you back up your code?

27 responses



DUNE Analysis User Survey



HDF5-Formatted Data

- DUNE always had niche use of HDF5-formatted data. Exporting data from an *art* job to external ML package often involves picking a non-ROOT format.
- DUNE's DAQ has switched over to writing data in HDF5 format.
  - Coldboxes are already writing data in this format
  - Petabytes of data are coming in 2022 from ProtoDUNE-2 and VD
- No more artdaq for DUNE – dune-daq is a newly engineered DAQ system

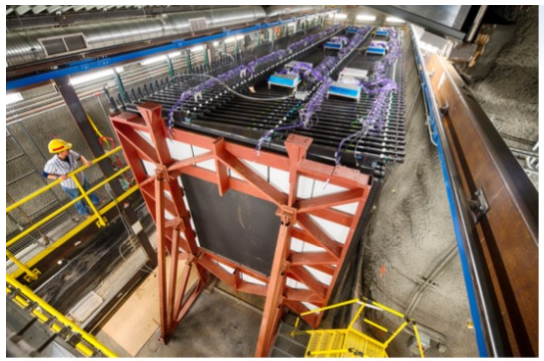
HDF5 Issues

- Benefits:
  - DAQ group prefers to write data in this format
  - File transfers within HPC centers optimized for HDF5
  - Common format used in a broad community
  - allows users to read only select subsets of a file, just like ROOT.
- Challenges:
  - XRootD works with HDF5 using the POSIX preload library interface. Prefer the static wrapper over the dynamic wrapper so as not to intercept all i/o with XRootD but need to get it to work
  - Persisting C++ objects is not automatic – have to do it ourselves.
  - File browsing is not as easy. ROOT's TBrowser with zero code is very convenient
  - We had to re-invent a delayed HDF5 reader for *art* jobs.

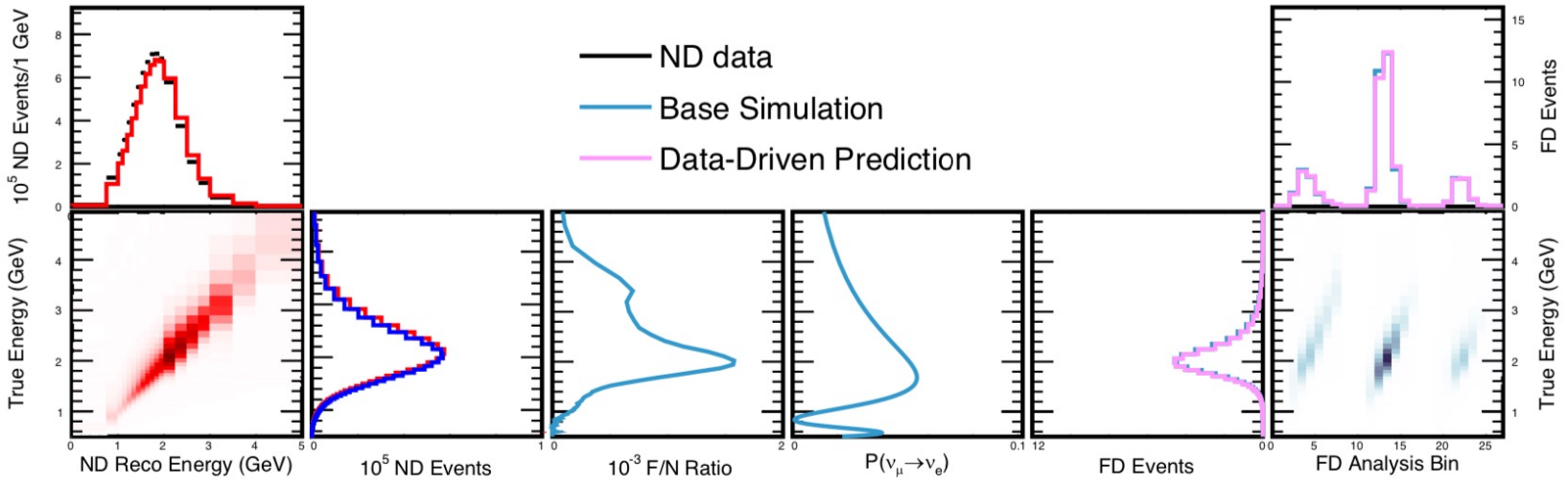
Large DUNE FD Trigger Records

- One trigger record from a far-detector module with non-zero-suppressed waveforms takes of order 4 GB of memory/storage
- Calibrated, de-noised, deconvoluted etc. waveforms take more space.
- Some analyses do not work on high-level objects – CNN analyses take waveforms ("images") and output event classification. Need raw or deconvolved data to do some end analyses.
- We want to read out 100s of continuous waveforms for a supernova burst trigger: 150 TB per FD module. And we have four FD modules.
- We need to process pieces of this data at a time to reduce memory footprint.
- Prefer to process in parallel, but input data dominate memory budget – loading more data in memory at a time to use multithreaded workflows doesn't gain much.
- Highly segmented detector makes this processing embarrassingly parallel
- Can always be done, even with the waterfall-style *art* framework, but custom code has been/continues to be written to do it.
  - delayed reading
  - eager writing
  - dropping data from memory when it is no longer useful.

Oscillation Fits



Extrapolating from Near to Far



- Use the ND ν_μ sample to predict the FD ν_μ sample.
- Use the ND ν_μ sample to predict the FD ν_e signal.

Alex Himmel at Phystat-Nu 2019
https://indico.cern.ch/event/735431/contributions/3137791/attachments/1783219/2902091/2019-01-23-lbl-stats.pdf