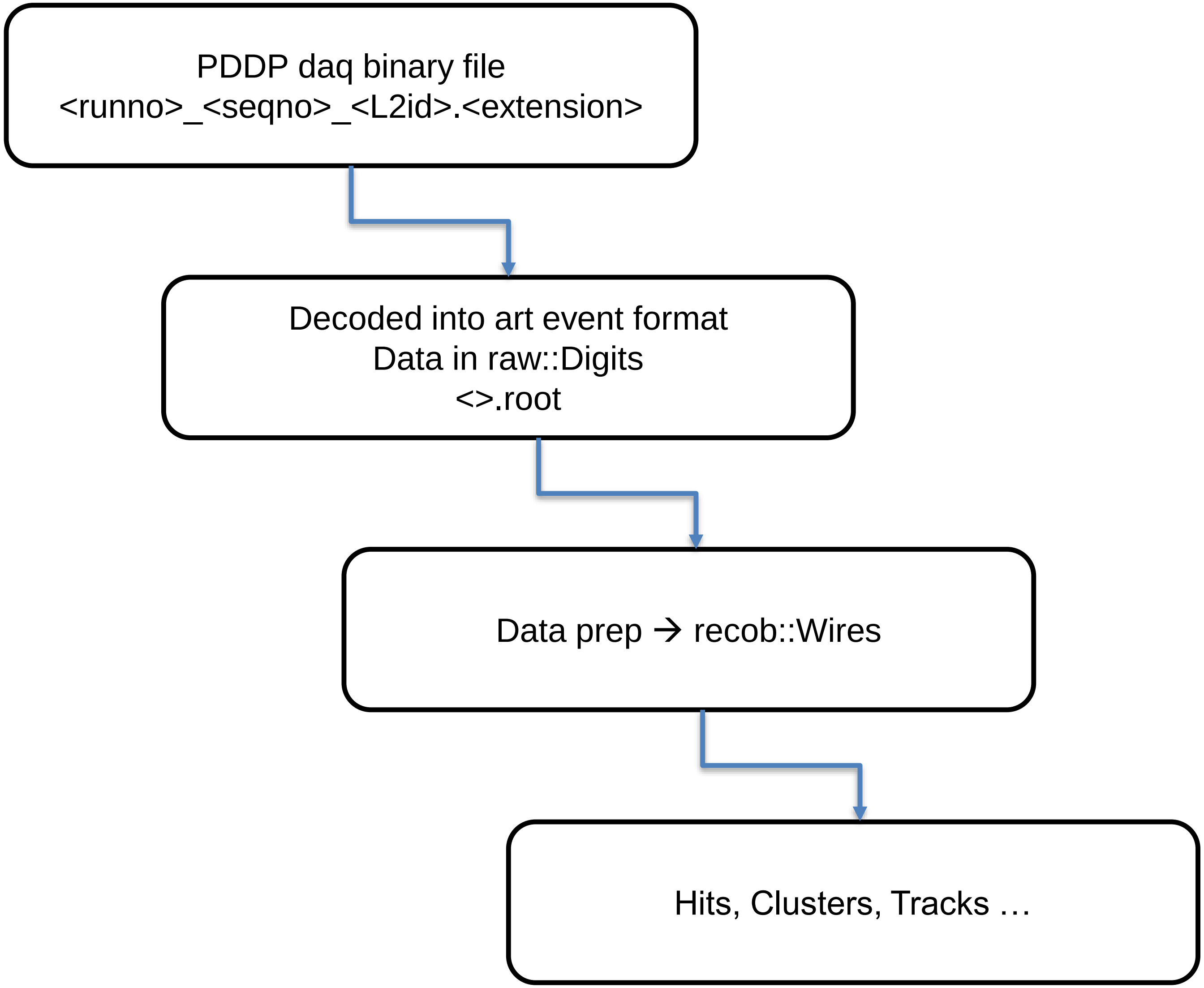
Accessing and reconstructing ProtoDUNE – DP data

Vyacheslav Galymov
IP2I Lyon

Steps



PDDP daq binary file
<runno>_<seqno>_<L2id>.<extension>

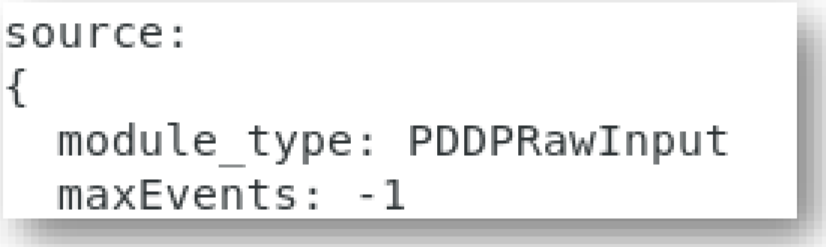
Decoded into art event format
Data in raw::Digits
<>.root

Data prep → recob::Wires

Hits, Clusters, Tracks …

Raw data decoder

- The decoder to read in binary data is activated via custom art source input PDDPRawInput
  - E.g.,



```
source:
{
  module_type: PDDPRawInput
  maxEvents: -1
```

- Converter fcl file: dunetpc/dune/Protodune/dualphase/fcl/**pddp_daq_converter.fcl**

- Other configuration parameters are related to where various larsoft products are placed:
  - **OutputLabelRawDigits** (default if not given **daq**)
  - **OutputLabelRDTime** (default if not given **daq**), should be **timingrawdecoder:daq**
  - **OutputLabelRDStatus** (default if not given **daq**)
  - **InvertBaseline** (default is 0) if > 0, the signals will be inverted. This is only needed for earlier commissioning runs (run numbers < 1257)
    - InvertBasline = 200 should be used for run nb < 1257

Order of raw::Digits

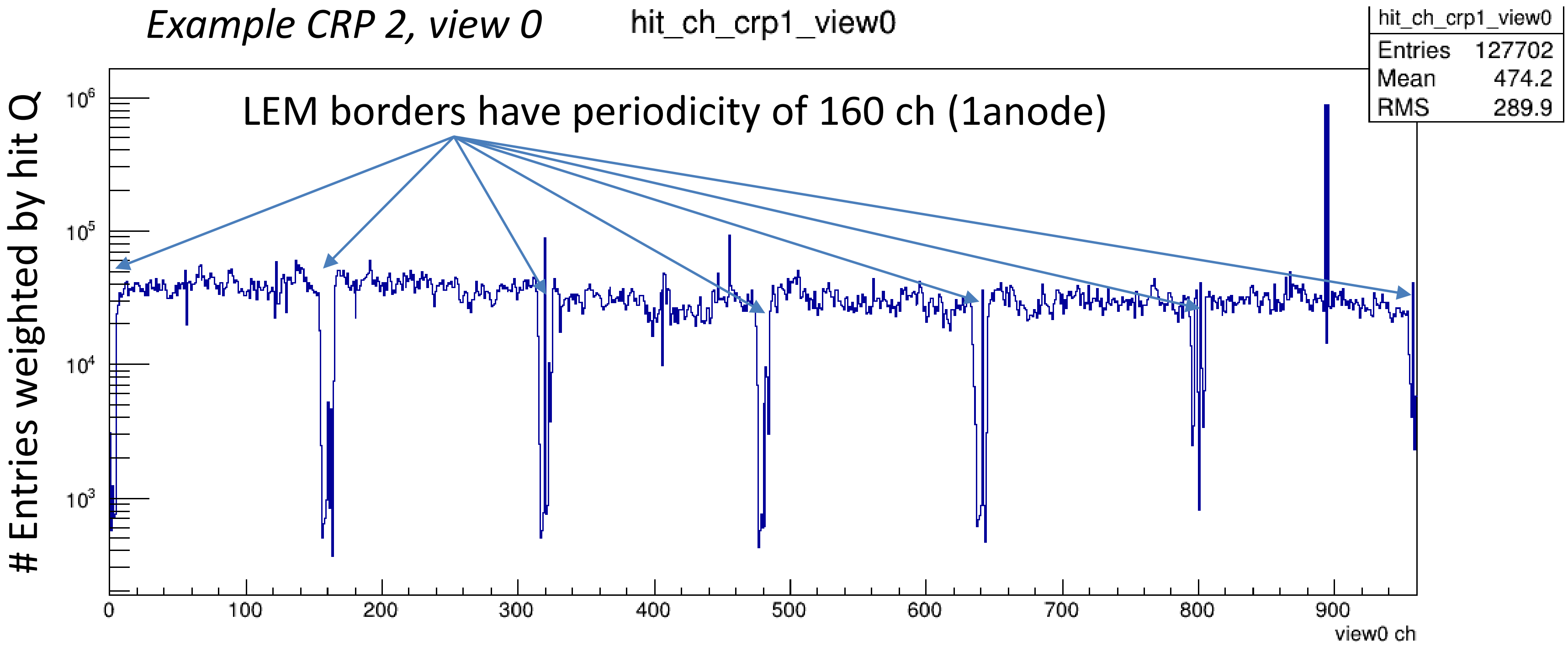
- After the channel data are decoded, the channel map PDDPChannelMap service puts them in the CRP No / View channel order
- So the channels of raw::Digits written to data store are ordered as follows:
  - CRP 0: view Z ch 0 ... 959, view X ch 0 ... 959
    CRP 1: view Z ch 0 ... 959, view X ch 0 ... 959
    CRP 2: view Z ch 0 ... 959, view X ch 0 ... 959
    CRP 3: view Z ch 0 ... 959, view X ch 0 ... 959

PDDP ChannelStatus service

- The list of bad / noisy channels is specified here:
  - /dunetpc/dune/Protodune/dualphase/fcl/channelstatus_pddp.fcl
  - To be declared in the standard way:
    - services.ChannelStatusService: @local::pddp_channel_status

- Currently the list of bad channels contains:
  - The 3200 channels that are not read in the 2 CRP configuration of ProtoDUNE-DP, but still written by the L1 event builder
  - The list of 280 channels which do not see any signal due to LEMs border. This is 5 channels around each LEM (so periodicity of 160 channels) → 280 channels in total
- To be added: a few of channels that did show any signals during tests with CRP charge injection system

Blind channels due to LEM border



- The border around each LEM masks 5 anode channels
- 5 x 3.125 ~ 16 mm, which is what is expected from LEM design
  - 10 mm FR4 border + 5 mm Cu rim

Drift field

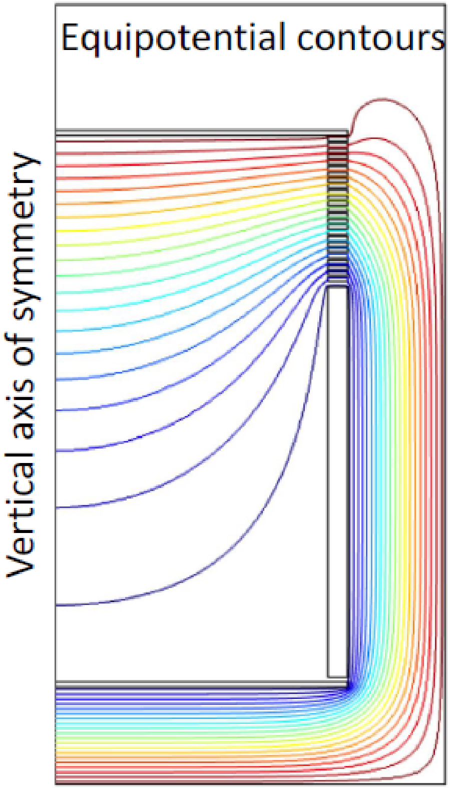
- The drift field in PDDP is non-uniform after about a meter from CRP
  - Problem with VHV feedthrough shorting on one of the field cage profiles via the first degrader link

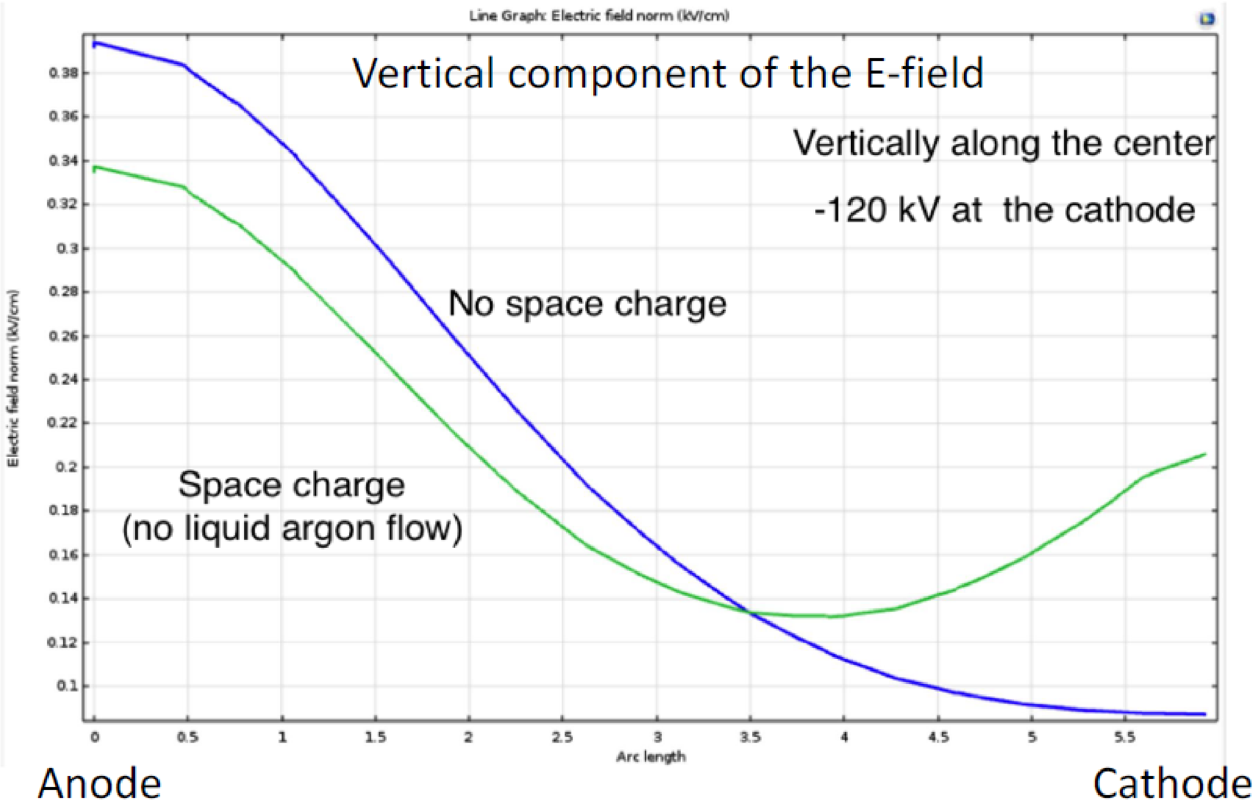
- With the reduced field due to VHV feedthrough issue, the drift field is not 0.5 kV/cm:
  - So far have been running at 50 kV (the drift velocity is ~0.8 mm/us or ½ from what it should have been)
  - Then according to simulations:
services.DetectorPropertiesService.Efield: [**0.166**, 4.0, 0.0]
    - The other two number should not be touched normally. This is an artifact of the drift field configuration that includes fields from U → V and V → Z in APA planes

Electric field simulation

Simulation of F. Resnati from the report on HV problem

- Given our present understanding, the electric field in the drift volume has been calculated with and without space-charge (only the contribution from primary ionization, no from the ion back flow)





Reco stage

- First version: dunetpc/dune/Protodune/dualphase/fcl/pddp_reco.fcl
- Tool-based data prep
- Hit finding with DPRawHitFinder developed by C. Alt
  - Fits raw waveform with asymmetric function
- Tried to get tracks:
  - trajcluster / pmtrack / pmtrajfit

Basic DataPrep list of tools for DPDP reconstruction

pddp_datapreptools: [

“digitReader”, ← reads in RawDigits (mandatory)

“pddp_RemoveBadChannels”, ← masks bad channels (optional)

“pddp_adcPedestalFit”, ← runs pedestal fit (mandatory)

“adcSampleFiller”, ← ADC – pedestal (mandatory)

“pddp_adcMultiThreshSignalFinder” ← builds ROI recob::Wire for hit reco using multi threshold ROI search algorithm

]

ROI search tool

```
tools.pddp_adcMultiThreshSignalFinder: {
  tool_type: AdcMultiThreshSignalFinder
  LogLevel: 1
  UseStandardDeviation: true
  Threshold1: 2
  SamplesAboveThreshold1: 10
  Threshold2: 2
  SamplesAboveThreshold2: 10
  ThresholdMax: 3
  ThresholdMin: 0
  BinsBefore: 10
  BinsAfter: 50
}
```

Absolute or in units of PedRMS

Min threshold 2 x PedRMS ~ 3 ADC

Min nb of consecutive samples above

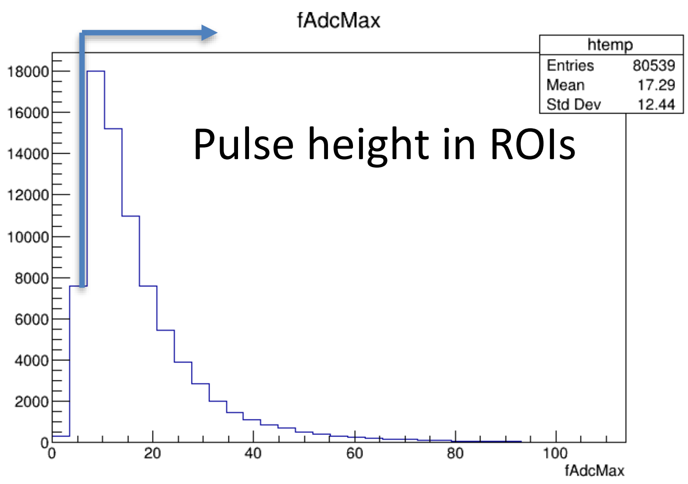
Optionally can set another threshold

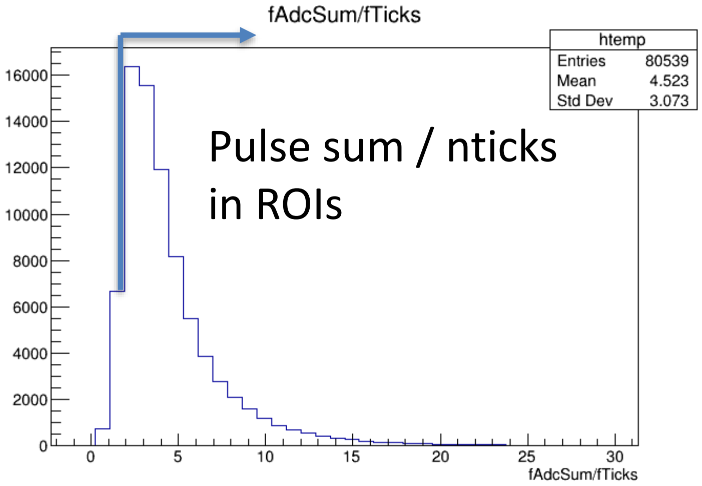
Require max value within ROI region to be above 3 x PedRMS

Decides where to stop ROI expansion

Number of ticks to pad on each side

Parameters





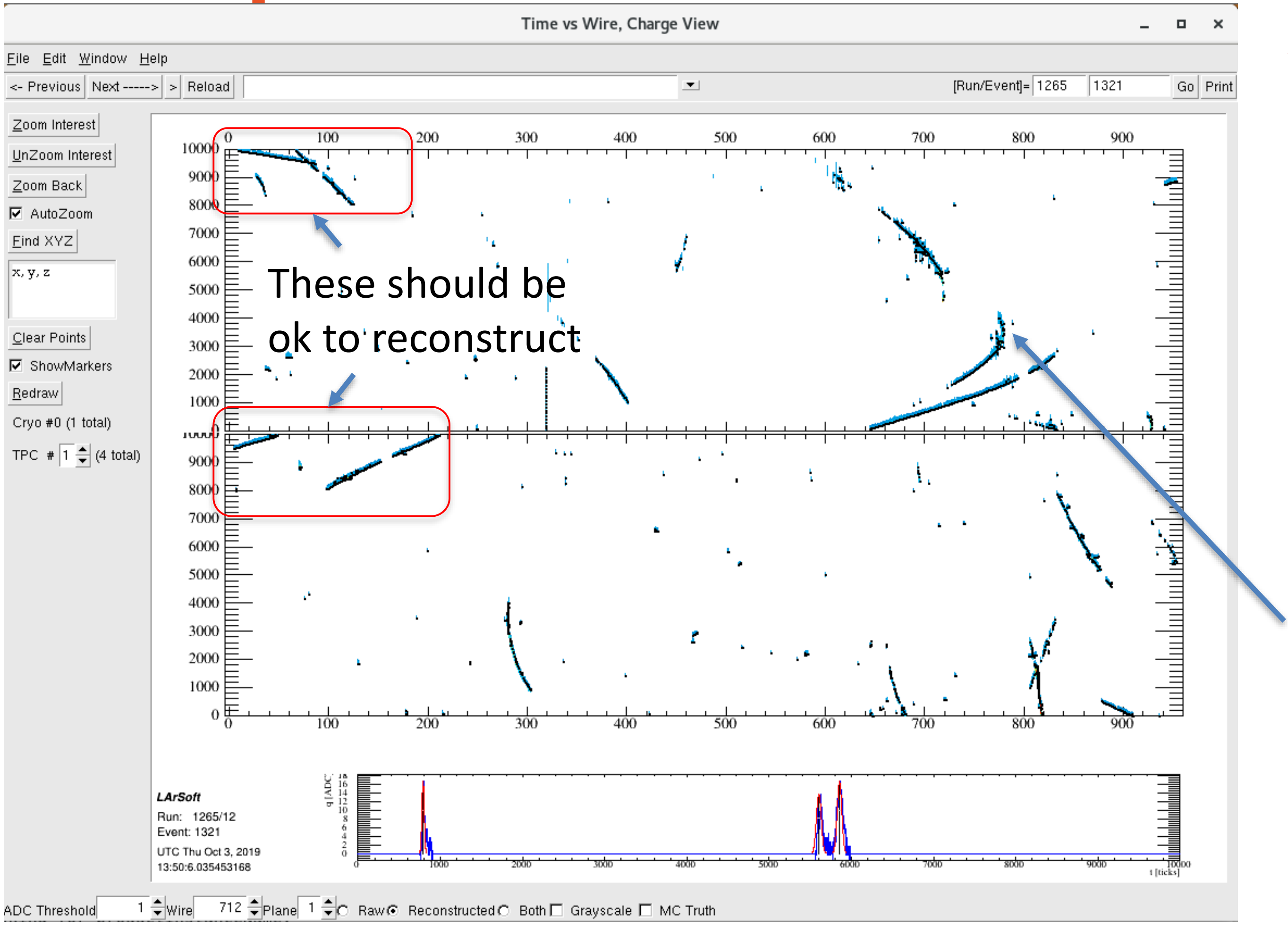
- Hits:
  - Min pulse height 2 ADC
  - Avg pulse sum in 2 ADC / tick
  - Min pulse width 5 ticks

Parameters

- trajcluster: parameters of standard_trajcluster / standard_trajclusteralg
  - Fraction of hits used is ~80% per event
  - Since we are dealing only with cosmics perhaps could still tune to improve handling of trajectory breaks by delta rays

- pmtrack/pmtrajfit: standard parameters dunefd_pmalgtrackmaker / dunefd_pmalgtrajfitter

Example event: ROI + clusters



[Run/Event]= 1265 1321

Cryo #0 (1 total)

TPC # 1 (4 total)

LArSoft
Run: 1265/12
Event: 1321
UTC Thu Oct 3, 2019
13:50:6.035453168

ADC Threshold 1 Wire 712 Plane 1 ○ Raw ● Reconstructed ○ Both □ Grayscale □ MC Truth

These should be ok to reconstruct

Drift field distortions

Run 1265

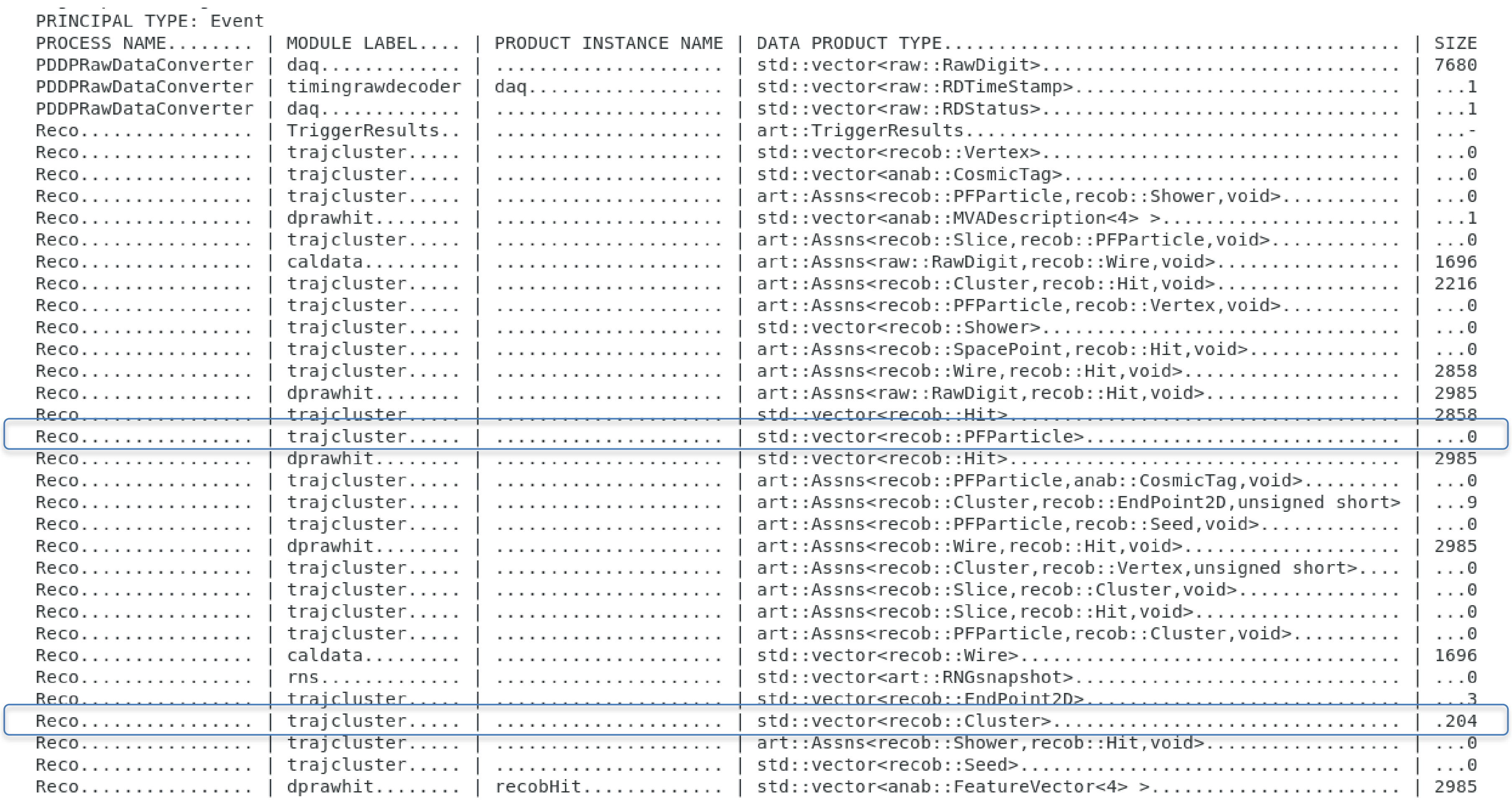
LEM dV = 3.1 kV

Most of the tracks are stubs: τ_e ~ 500 us for this run

➔ the drift volume transparency is ~40 cm and muon tracks fade away making them harder to reconstruct towards the end

However, cosmics that come sufficiently late after trigger should still give a good sample to work with (they are cut by the length of the readout window)

eventdump



```
PRINCIPAL TYPE: Event
PROCESS NAME........ | MODULE LABEL.... | PRODUCT INSTANCE NAME | DATA PRODUCT TYPE.......................................... | SIZE
PDDPRawDataConverter | daq............. | ..................... | std::vector<raw::RawDigit>................................. | 7680
PDDPRawDataConverter | timingrawdecoder | daq.................. | std::vector<raw::RDTimeStamp>.............................. | ...1
PDDPRawDataConverter | daq............. | ..................... | std::vector<raw::RDStatus>................................. | ...1
Reco................ | TriggerResults.. | ..................... | art::TriggerResults........................................ | ...-
Reco................ | trajcluster..... | ..................... | std::vector<recob::Vertex>................................. | ...0
Reco................ | trajcluster..... | ..................... | std::vector<anab::CosmicTag>............................... | ...0
Reco................ | trajcluster..... | ..................... | art::Assns<recob::PFParticle,recob::Shower,void>........... | ...0
Reco................ | dprawhit........ | ..................... | std::vector<anab::MVADescription<4> >...................... | ...1
Reco................ | trajcluster..... | ..................... | art::Assns<recob::Slice,recob::PFParticle,void>............ | ...0
Reco................ | caldata......... | ..................... | art::Assns<raw::RawDigit,recob::Wire,void>................. | 1696
Reco................ | trajcluster..... | ..................... | art::Assns<recob::Cluster,recob::Hit,void>................. | 2216
Reco................ | trajcluster..... | ..................... | art::Assns<recob::PFParticle,recob::Vertex,void>........... | ...0
Reco................ | trajcluster..... | ..................... | std::vector<recob::Shower>................................. | ...0
Reco................ | trajcluster..... | ..................... | art::Assns<recob::SpacePoint,recob::Hit,void>.............. | ...0
Reco................ | trajcluster..... | ..................... | art::Assns<recob::Wire,recob::Hit,void>.................... | 2858
Reco................ | dprawhit........ | ..................... | art::Assns<raw::RawDigit,recob::Hit,void>.................. | 2985
Reco................ | trajcluster..... | ..................... | std::vector<recob::Hit>.................................... | 2858
Reco................ | trajcluster..... | ..................... | std::vector<recob::PFParticle>............................. | ...0
Reco................ | dprawhit........ | ..................... | std::vector<recob::Hit>.................................... | 2985
Reco................ | trajcluster..... | ..................... | art::Assns<recob::PFParticle,anab::CosmicTag,void>......... | ...0
Reco................ | trajcluster..... | ..................... | art::Assns<recob::Cluster,recob::EndPoint2D,unsigned short> | ...9
Reco................ | trajcluster..... | ..................... | art::Assns<recob::PFParticle,recob::Seed,void>............. | ...0
Reco................ | dprawhit........ | ..................... | art::Assns<recob::Wire,recob::Hit,void>.................... | 2985
Reco................ | trajcluster..... | ..................... | art::Assns<recob::Cluster,recob::Vertex,unsigned short>.... | ...0
Reco................ | trajcluster..... | ..................... | art::Assns<recob::Slice,recob::Cluster,void>............... | ...0
Reco................ | trajcluster..... | ..................... | art::Assns<recob::Slice,recob::Hit,void>................... | ...0
Reco................ | trajcluster..... | ..................... | art::Assns<recob::PFParticle,recob::Cluster,void>.......... | ...0
Reco................ | caldata......... | ..................... | std::vector<recob::Wire>................................... | 1696
Reco................ | rns............. | ..................... | std::vector<art::RNGsnapshot>.............................. | ...0
Reco................ | trajcluster..... | ..................... | std::vector<recob::EndPoint2D>............................. | ...3
Reco................ | trajcluster..... | ..................... | std::vector<recob::Cluster>................................ | .204
Reco................ | trajcluster..... | ..................... | art::Assns<recob::Shower,recob::Hit,void>.................. | ...0
Reco................ | trajcluster..... | ..................... | std::vector<recob::Seed>................................... | ...0
Reco................ | dprawhit........ | recobHit............. | std::vector<anab::FeatureVector<4> >....................... | 2985
```

3D matching seems to be failing

The number of clusters also is the high side: many broken trajectories?

Test production

```
{
 "1265_12_a_pddprawdata_reco.root": {
  "application.family": "art",
  "application.version": "v08_32_01",
  "file_type": "detector",
  "art.run_type": "protodune-dp",
  "group": "dune",
  "art.process_name": "Reco",
  "data_tier": "hit-reconstructed",
  "data_stream": "cosmics",
  "file_format": "artroot",
  "start_time": "2019-10-15T13:22:10",
  "end_time": "2019-10-15T13:41:22",
  "runs": [ [ 1265, 12, "protodune-dp" ] ],
  "event_count": 30,
  "first_event": 1321,
  "last_event": 1437,
  "parents": [ "1265_12_a_pddprawdata.root" ],
  "art.first_event": [ 1265, 12, 1321 ],
  "art.last_event": [ 1265, 12, 1437 ],
  "art.file_format_era": "ART_2011a",
  "art.file_format_version": 13
 }
}
```

- Close to start test production within the dune-production group
  - Checking generated metadata is ok

Summary

- The decoder for ProtoDUNE-DP data and the channel map service to go from online to offline channel numbering have been available for some time
  - One possible future improvement is to restructure the decoder such that, as in the case of SP, it can be run in the data prep stage. This would allow to eventually ditch raw::Digits from event store (avoid duplicating bits)
  - One would still have a custom source input running on binary file that must generate some minimal info from decoded headers (run/sub-run #, event #, event timestamp …)
  - Then the decoder would be called to unpack the digits data as part of data prep tools
  - This would require some work and testing. But if the volume of data to be processed becomes substantially large it would be worth investigating
- The data prep stage runs smoothly on the unpacked data (D. Adams fixed the issue #23283 with errors in pedestal fits)
- Higher level reconstruction (after hits) requires tuning / fixing to get tracking working