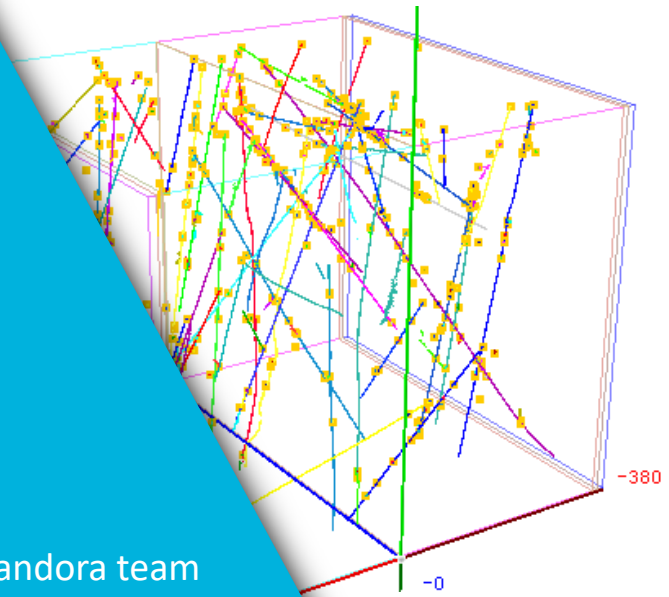
Introduction to LArTPC event reconstruction

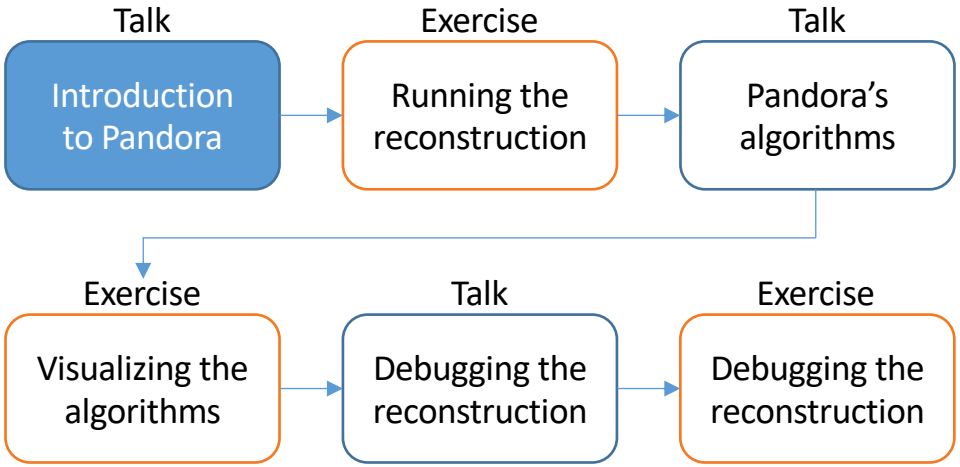


Maria Brigida Brunetti, Andy Chappell and Steve Dennis for the Pandora team

25/10/2022

8th UK LArTPC Software and Analysis Workshop

Reconstruction session

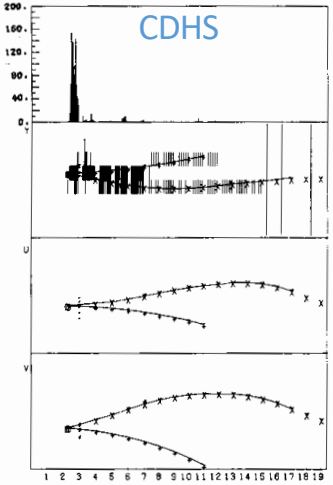


Credit: These slides are based on previous LArSoft workshop slides by Andrew Smith

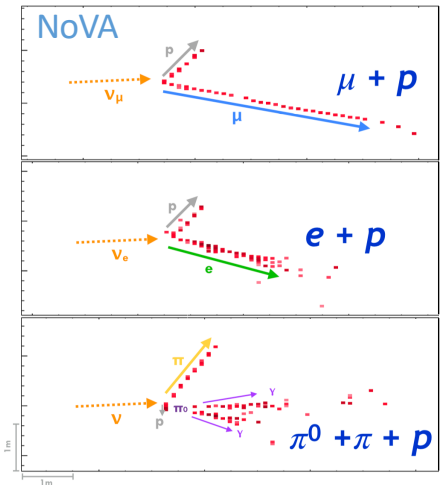
Key references:
Pandora ProtoDUNE paper
Pandora MicroBooNE paper

Neutrino detectors

New Frontiers in High-Energy Physics, pp227-261

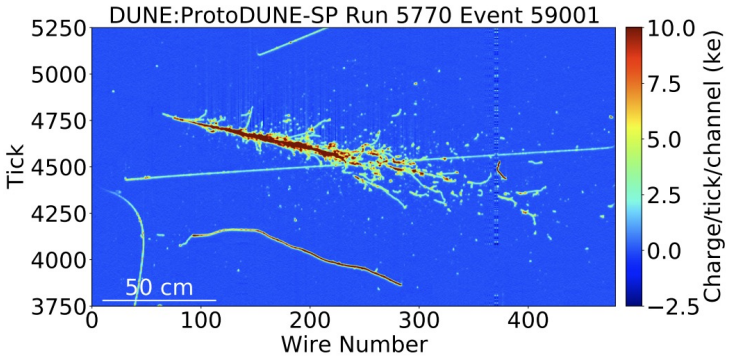


DOI: 10.5281/zenodo.1286758



B. Abi et al 2020 JINST 15 P12004

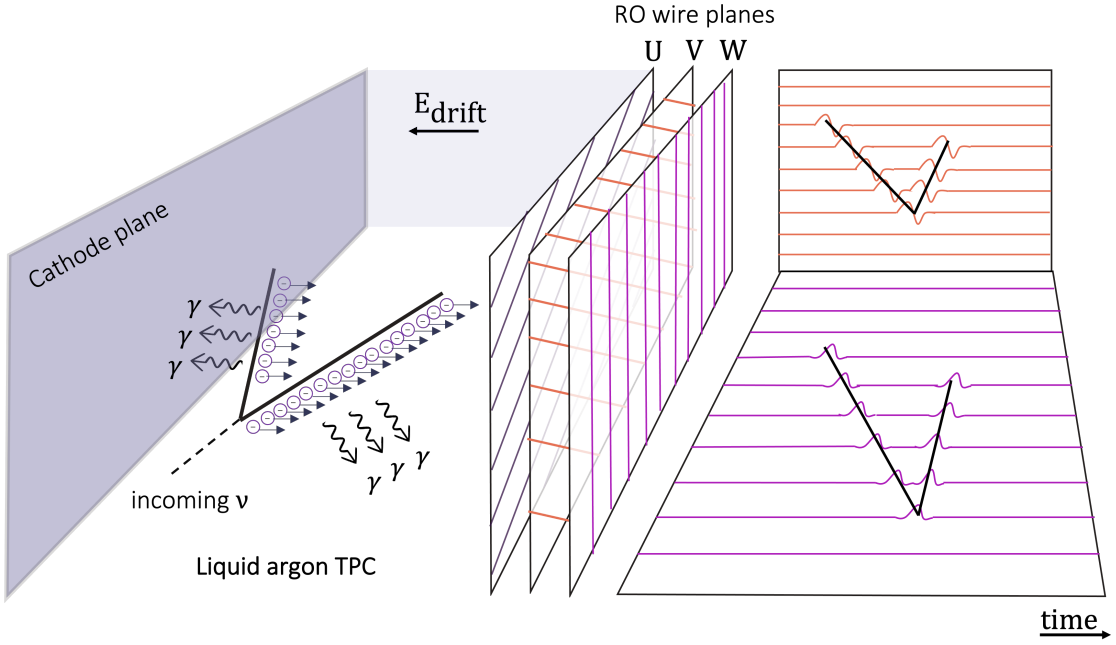
ProtoDUNE-SP



- Evolving detector technologies, with a general trend towards imaging neutrino interactions
  - Emphasis on identifying and characterizing individual visible particles
- Physics sensitivity now depends critically on both hardware and software
  - Need a sophisticated event reconstruction to harness information in the images
- Aim to reconstruct hierarchy of particles of identified types, with measured four-momenta
  - “Particle flow” reconstruction

LArTPC detectors

- Charged particles deposit ionization trails in liquid argon
- Ionization electrons drift in an applied electric field
- Electrons are detected by a series of readout planes (wire planes in this example)
- LArTPC detectors:
  - Past: ICARUS, ArgoNeuT, ProtoDUNE-SP/DP
  - Current: MicroBooNE, LArIAT, ICARUS@SBN
  - Coming soon: SBND, ProtoDUNE-HD

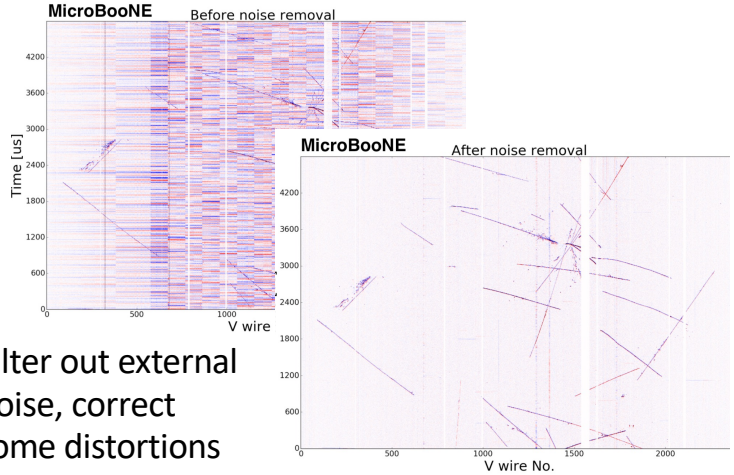


LArTPC animation link

arxiv:1612.05824

Event reconstruction – low-level

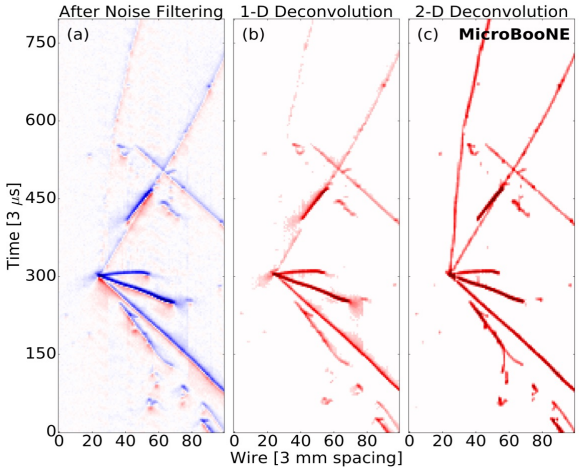
Noise filtering



Filter out external noise, correct some distortions

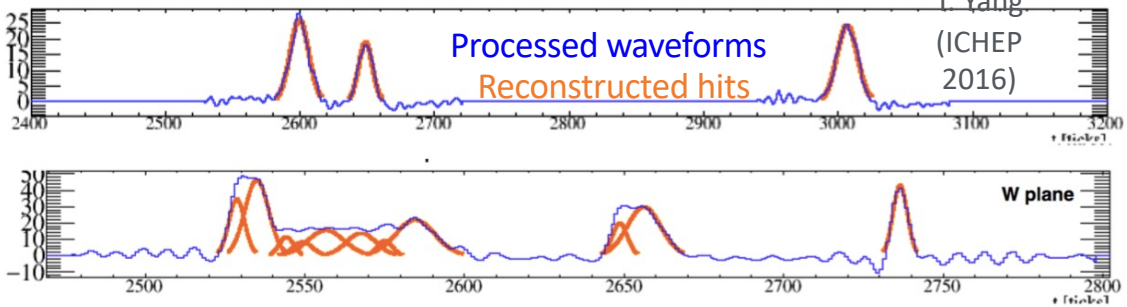
Signal processing

Convert digitized TPC waveform to number of ionization electrons passing through a wire plane at a given time (via deconvolution)



Hit finding

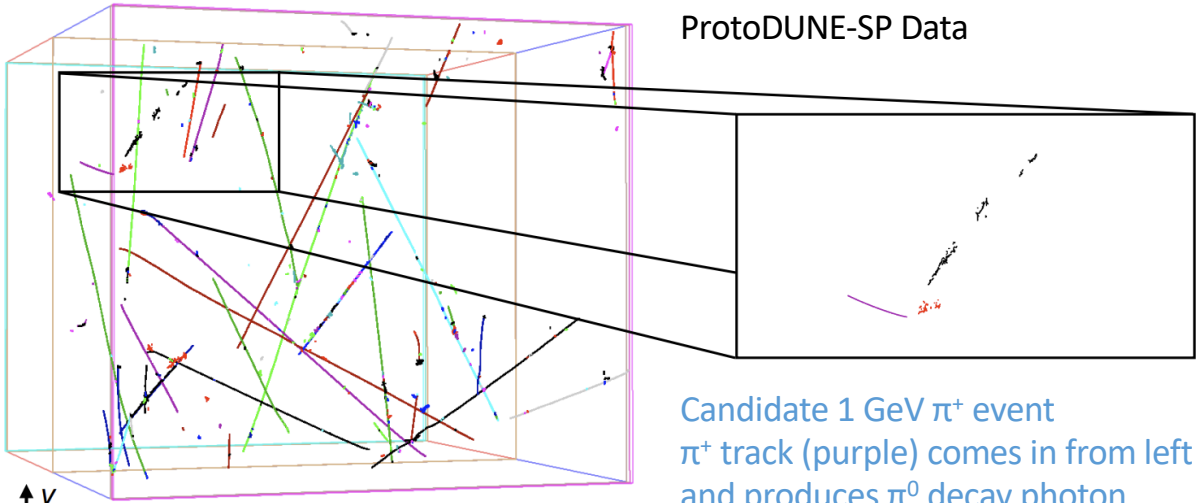
Fit clean waveform with N Gaussians, where N is number of peaks in a pulse. Each Gaussian represents a hit.



T. Yang (ICHEP 2016)

Event reconstruction – pattern recognition

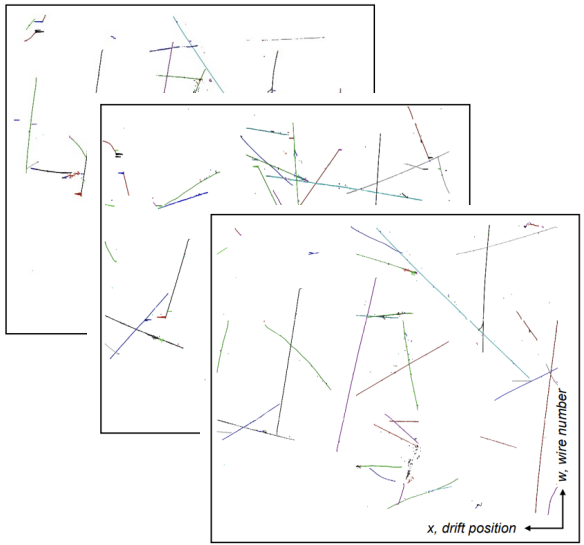
- Main aims of the pattern recognition step are to:
  - Produce 3D reconstructed particles, based on inputs of 3 x 2D images
  - Reconstruct the hierarchy of particles resulting from an interaction.



ProtoDUNE-SP Data

L. Whitehead

Candidate 1 GeV π^+ event
π^+ track (purple) comes in from left and produces π^0 decay photon showers (red and black)



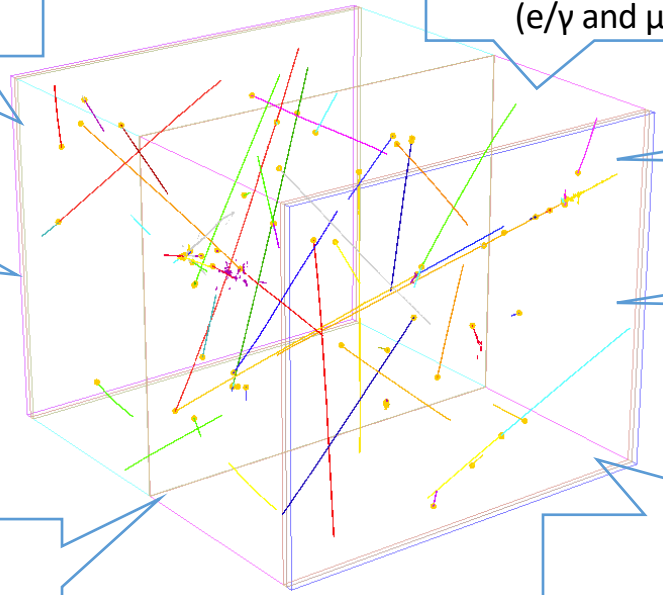
Output from hit finding

Event reconstruction – high-level characterisation

Calorimetry
dE/dx from ionisation

Particle ID
PID from dE/dx
Start/End for separation
(e/γ and μ/K/p)

Neutrino Interactions
Flavour ID
Background rejection
Energy estimation



π^0 Mass Reconstruction

Michel Electron Studies

ProtoDUNE-SP Simulation

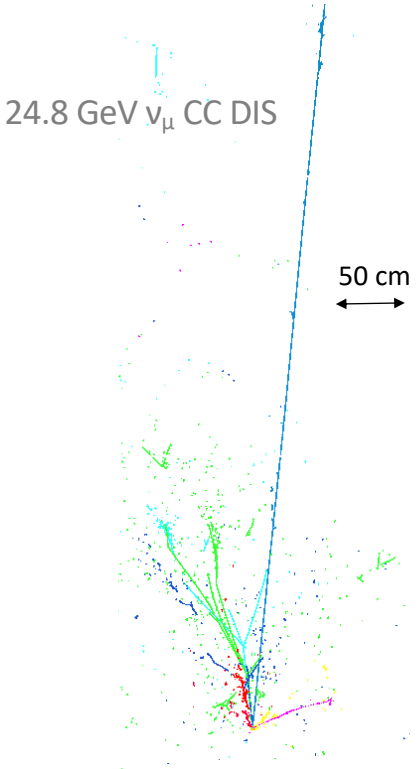
Energy Estimation
Muons
EM showers
Hadronic activity

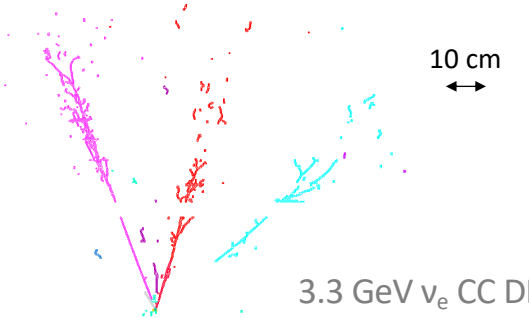
Track Fitting
Trajectory determination
Momentum from multiple Coulomb scattering for exiting tracks

Slide: S. Green

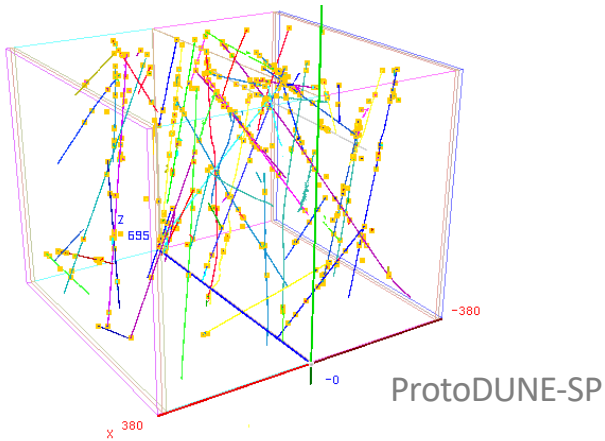
Event reconstruction – focus on pattern recognition

It is a significant challenge to develop automated, algorithmic LArTPC pattern recognition





- Complex, diverse topologies
- Long exposures due to long drift times (up to a few milliseconds)
- Significant cosmic-ray muon background in surface-based detectors



Pandora multi-algorithm approach

- Single clustering approach is unlikely to work for such complex topologies:
  - Mix of track-like and shower-like clusters
- Pandora uses a multi-algorithm approach:
  - Build up events gradually
  - Each step is incremental - aim not to make mistakes (undoing mistakes is hard…)
  - Deploy more sophisticated algorithms as picture of event develops
  - Build physics and detector knowledge into algorithms

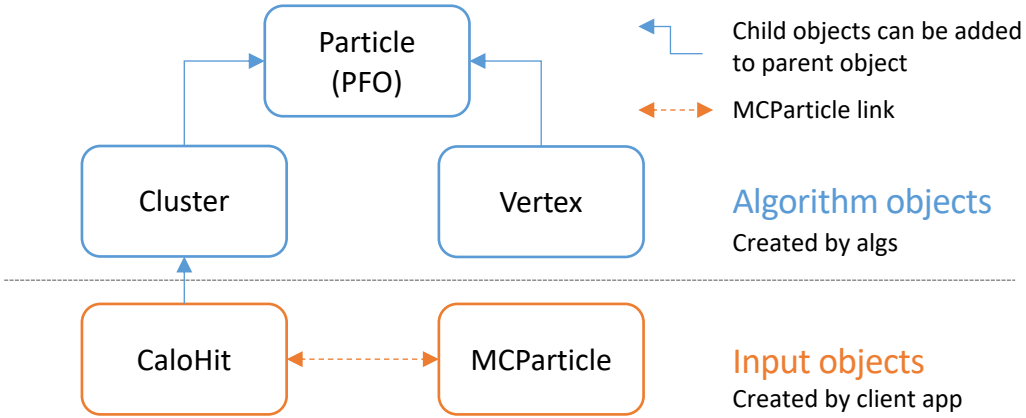
Pandora multi-algorithm approach

- Algorithms contain high-level logic and concentrate on the important bits
  - Physics and pattern recognition ideas
- Pandora software development kit (SDK) supports algorithms
  - Functions to access objects
  - Make new objects
  - Modify existing objects, etc.

Algorithm 1 Cluster creation pseudocode. The logic determining when to create new Clusters and when to extend existing Clusters will vary between algorithms.

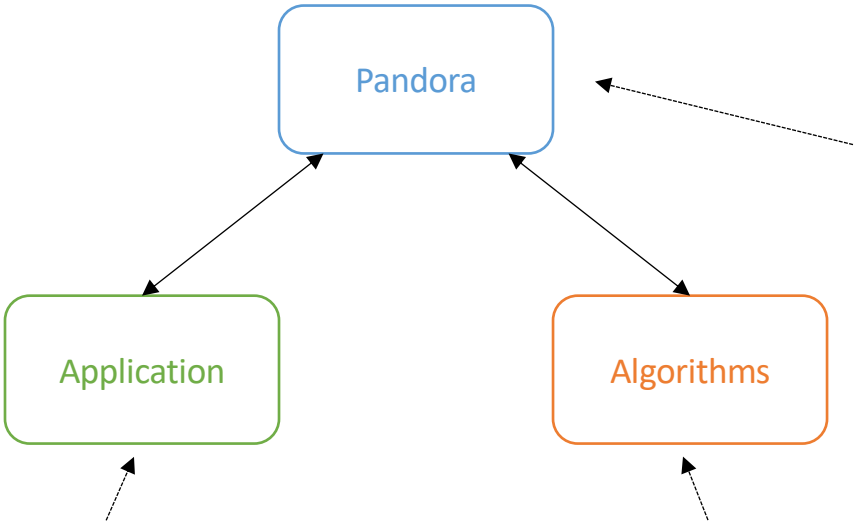
```
1:  procedure CLUSTER CREATION
2:      Create temporary Cluster list
3:      Get current CaloHit list
4:      for all CaloHits do
5:          if CaloHit available then
6:              for all newly-created Clusters do
7:                  Find best host Cluster
8:              if Suitable host Cluster found then
9:                  Add CaloHit to host Cluster
10:             else
11:                 Add CaloHit to a new Cluster
12:     Save new Clusters in a named list
```

Example algorithm structure



Event Data Model

Pandora in LArSoft



pandora (framework, visualization)

Re-usable libraries to support multi-alg approach

Hosted on PandoraPFA GitHub
pre-built as external package by LArSoft

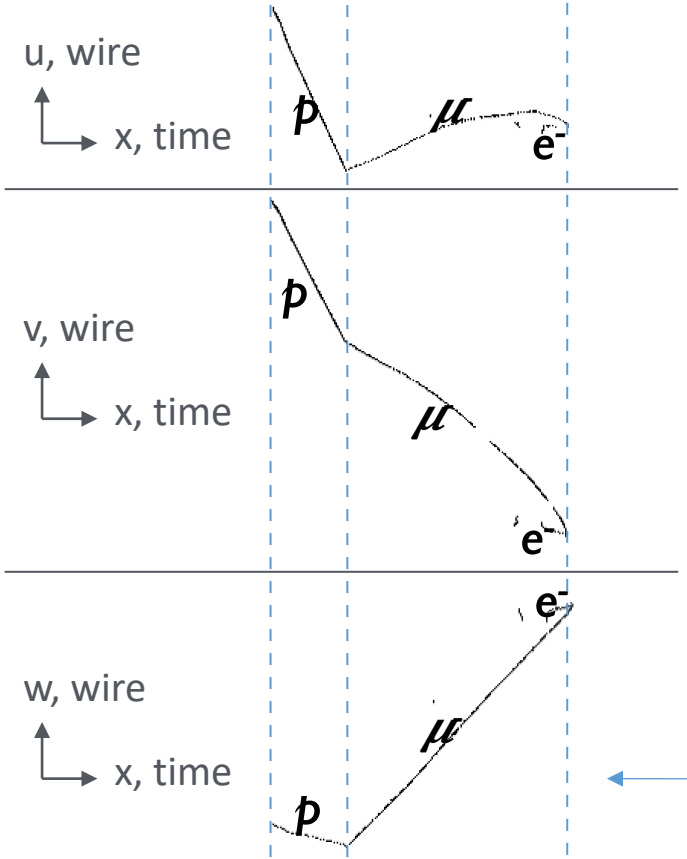
Producer module, provides translation LArSoft⟷Pandora

Hosted via LArSoft GitHub, built with mrb

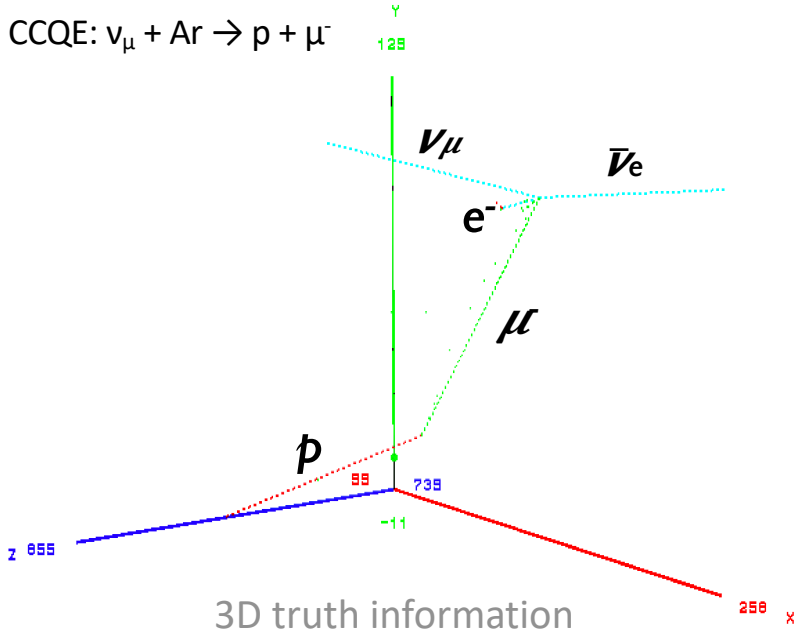
100+ algorithms and tools that control the patrec

Hosted via LArSoft and PandoraPFA GitHub, can build with mrb

Inputs to Pandora



CCQE: $\nu_\mu + Ar \rightarrow p + \mu^-$



3D truth information

Three 2D representations with common x coordinate, derived from drift time

Pandora application: larpandora

- The controlling application should be simple: create Pandora instance(s), register algorithms, provide a Pandora Settings XML file and handle event input/output:
  - In reality, often tends to get complicated (vindicating decision to separate algorithms from steps needed to access and control the inputs and outputs…)

```
/**
 *  @brief  ILArPandora class
 */
class ILArPandora : public art::EDProducer
```

Implement as an art::EDProducer

Relevant callbacks

```
void beginJob();
void produce(art::Event &evt);
```

Algorithm Pseudocode description of a client application for LAr TPC event reconstruction in a single drift volume

```
 1: procedure MAIN
 2:     Create a Pandora instance
 3:     Register Algorithms and Plugins
 4:     Ask Pandora to parse XML settings file
 5:     for all Events do
 6:         Create CaloHit instances
 7:         Create MCParticle instances
 8:         Specify MCParticle-CaloHit relationships
 9:         Ask Pandora to process the event
10:         Get output PFOs and write to file
11:         Reset Pandora before next event
```

We will now try running the larpandora module in LArSoft