

Producing SNB samples using refactored dunetpc

James Shen

Overview

- Determine the location of a supernova using neutrino pointing information.
- Based on previous research done by AJ Roeth for the TDR
- Relevant interactions:
 - Electron Scattering (ES): High pointing resolution, few events (200 per burst @ 10kpc)
 - Charge Current: Lower resolution but a lot more events (2600 per burst @ 10kpc).
- Generators:
 - **ES: NueScatter (by Dan Pershey): /dunetpc/dune/EventGenerator/NuE, current focus**
 - CC: Marley

Current Workflow

Using feature branch feature/jierans_SNBPointing

- Generator: /dune/app/users/jierans/SN_Pointing/fcl/prodnuescatter*.fcl
 - No radiologicals just yet.
- Geant4: supernova_g4_dune10kt_1x2x6.fcl
- Detsim: standard_detsim_dune10kt_1x2x6.fcl
- Analysis: pointrestree.fcl (custom module)

Generates 100 events per job. Each job takes about 8hrs, about 3GB of memory

Modifications Made:

- Detsim: disabling unnecessary Optical detector modules
- Reco: disabling unused reco modules for low energy events
 - Only using trajcluster and pmtracktc; no pandora

Issues Encountered: Geant4 stage

G4GDML reports a bunch of validation errors when reading the refactored geometry files. These errors appears benign and didn't seem to prevent code from running.

GDML File: `dunetpc/v09_25_00/gdml/dune10kt_v4_refactored_1x2x6.gdml`

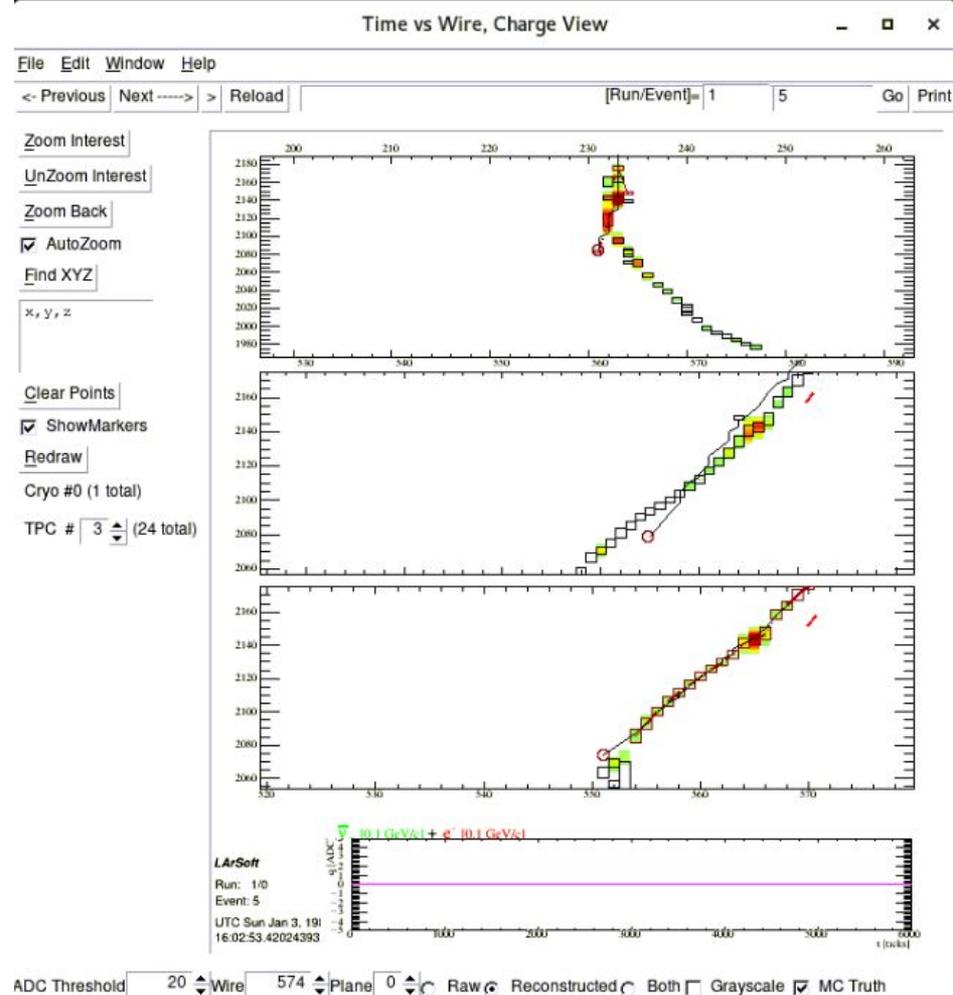
Hans-Joachim Wenzel is currently looking into these errors.

Detsim & reco stage

- Removed unnecessary opdet modules by modifying OpDetMultisim.fcl, keeping only the most relevant setting: DEF35QENonRefl2PE
- Raw Digits are significantly larger (as pointed out by Dom last week): about 12GB/100Events in the 1x2x6 workspace. This is also affecting runtime
 - Tom Junk suggests that most of the runtime penalty is caused by ROOT writing info into a tree (a worse than $O(N)$) process.

Event display

- Fcl file corresponding to the most current detsim/reco modules:
dunetpc/fcl/evd/evd_dunefd_refactor.red.fcl
- Pedestal & time tick is changed to be the same as the ProtoDune pedestal.
 - Is the same change necessary for detsim/reco stage?

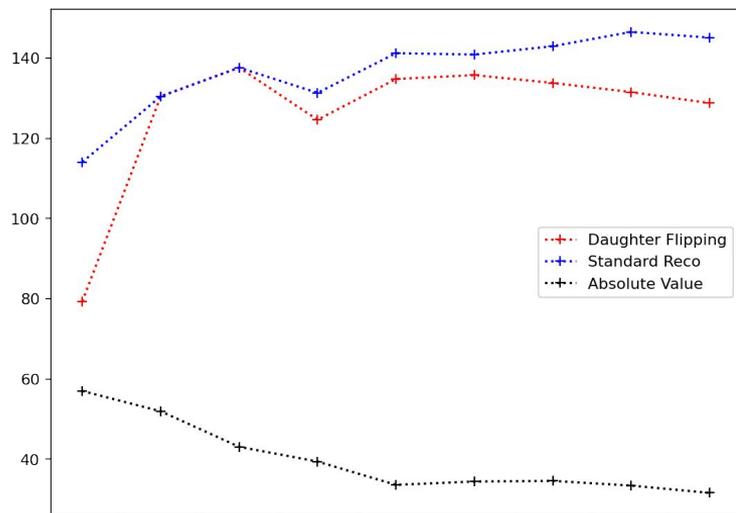
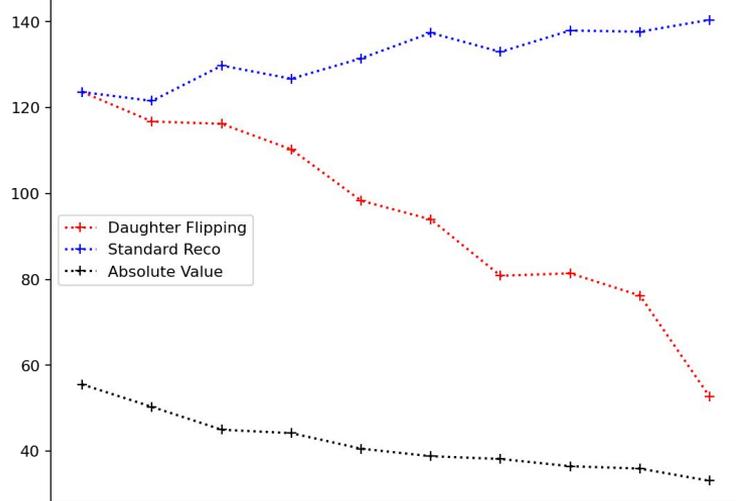


Analysis

- Function of PointResTree:
 - Determine the primary track of the electron
 - Resolve directional ambiguity of the track (daughter flipping)
 - Energy reconstruction of the electron
- Information provided by PointResTree can then be used to create a PDF. Further reco samples can then be used to determine the SN direction.

Daughter Flipping

- Time information of hits in TPC are used to determine their x coordinates -> the timing order of the hits are ambiguous.
- Resolve this by minimizing the spread between the daughter tracks and the primary(longest track) track
- **After Refactor, this is currently broken.**
 - **Number of Daughter tracks appear to have decreased**



Energy Reconstruction

- Determining the charge-energy correlation to allow reconstructing electron energy based on charge.
- Simply adding charges of hits tagged with the primary track does not work due to missing bremsstrahlung hits.
- Use Optical Detector information to do drift correction for charges.
- Adding charges of all hits in the event improves linearity of the correlation, especially in the collection.
- Removing noise/radiological background using a 3D distance cut. Might not be the best way to do it...

Charge vs. Energy, U plane

