Michel Analysis Update

Aleena Rafique, Zelimir Djurcic

ProtoDUNE sim/reco meeting

02/19/2020

Argonne NATIONAL LABORATORY

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

Michel energy spectrum

- **DocDB 17579**
  - Total Michel energy = cone-only hits energy + additional hits from parent muon
- Investigating the high energy tail

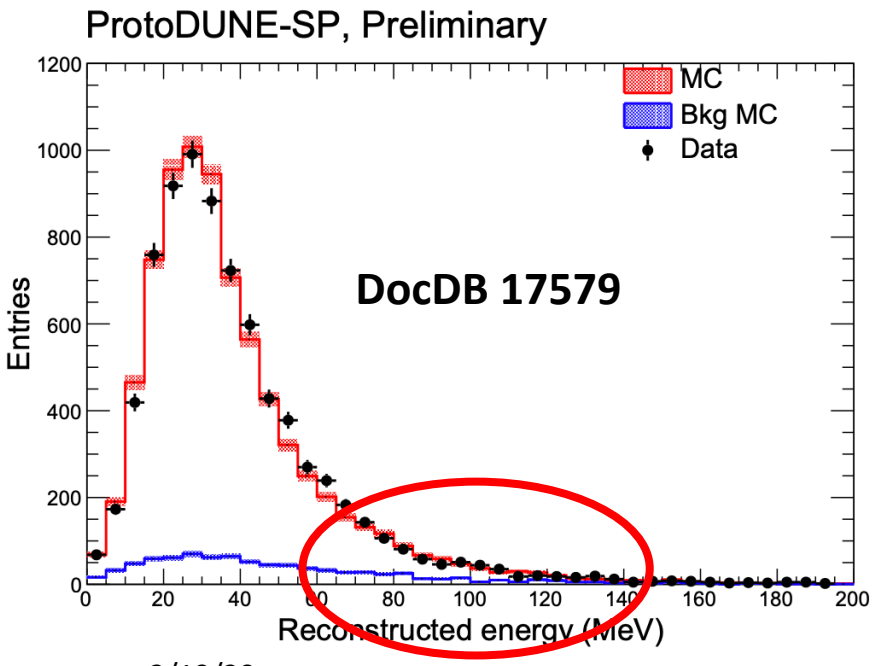
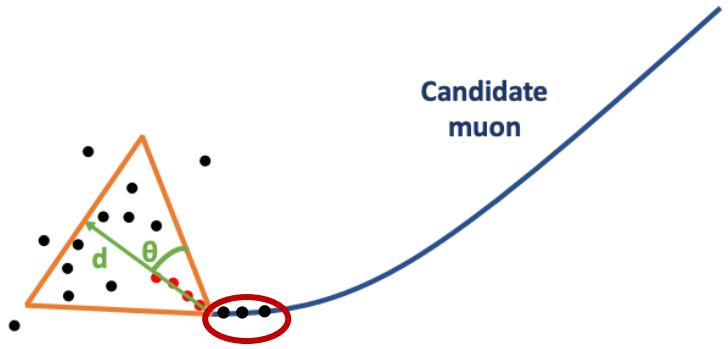
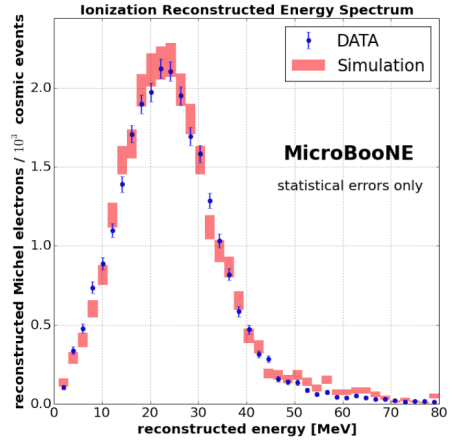
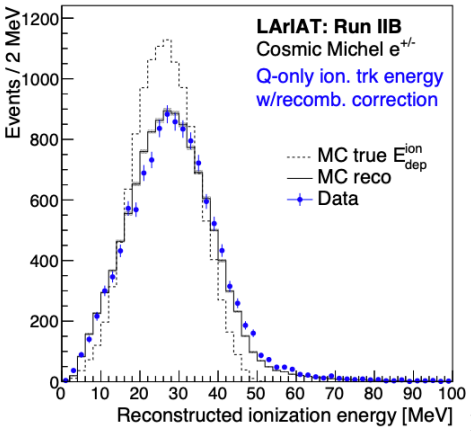


Figure 16: Michel energy spectrum from ProtoDUNE data and simulation.



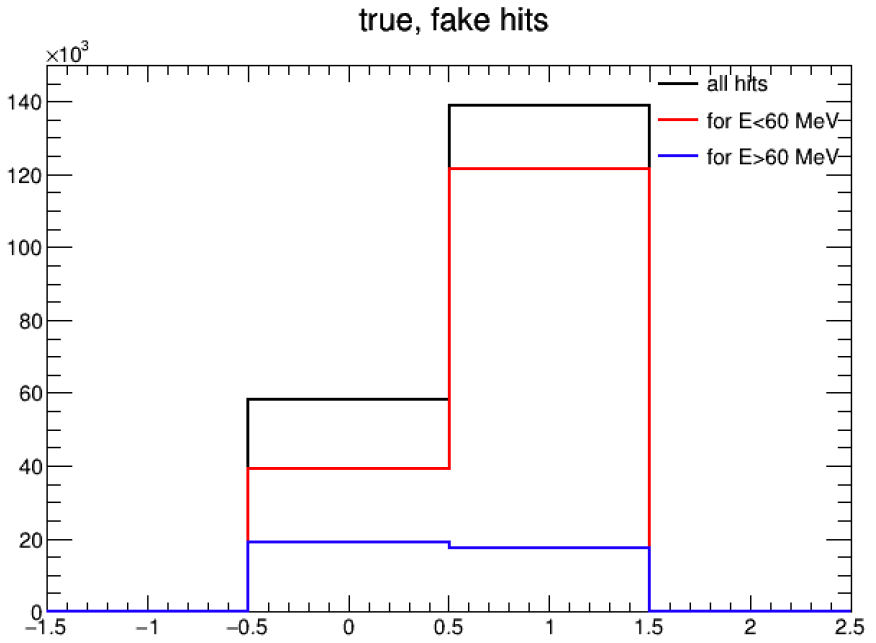




Tail is present in other experimental measurements as well, maybe not as prominent as in ProtoDUNE

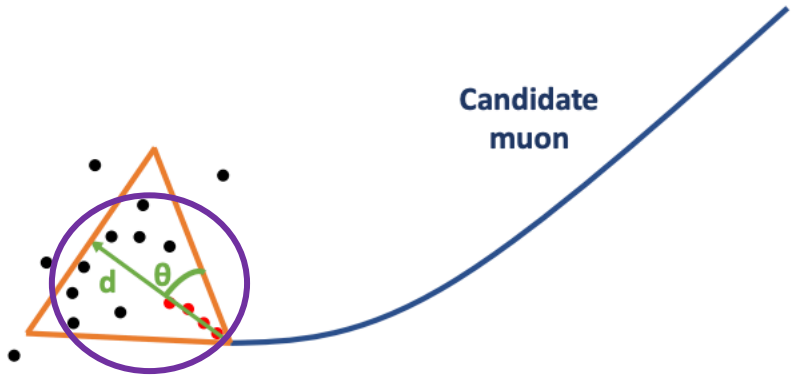
Sample hit purity

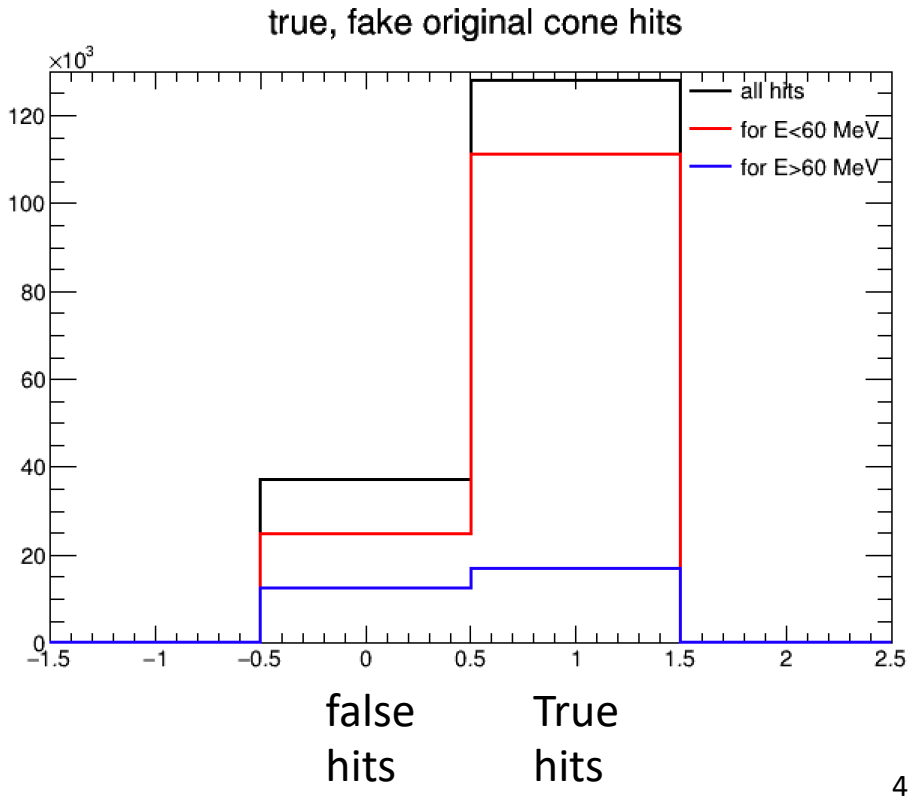
- **Event purity** is defined to be the number of selected muons in the sample that decay into true Michels (89%)
- Now looking at **hit purity** (selected reco hits that belong to the true Michels) (70%)
- Hit purity for events with
  - E<60 MeV: 77%
  - E>=60 MeV: 45%



Sample hit purity including cone-only hits

- Sample hit purity using cone-only hits (without adding any hits from parent muons) (77%)
- Hit purity for events with
  - E<60 MeV: 82%
  - E>=60 MeV: 59%

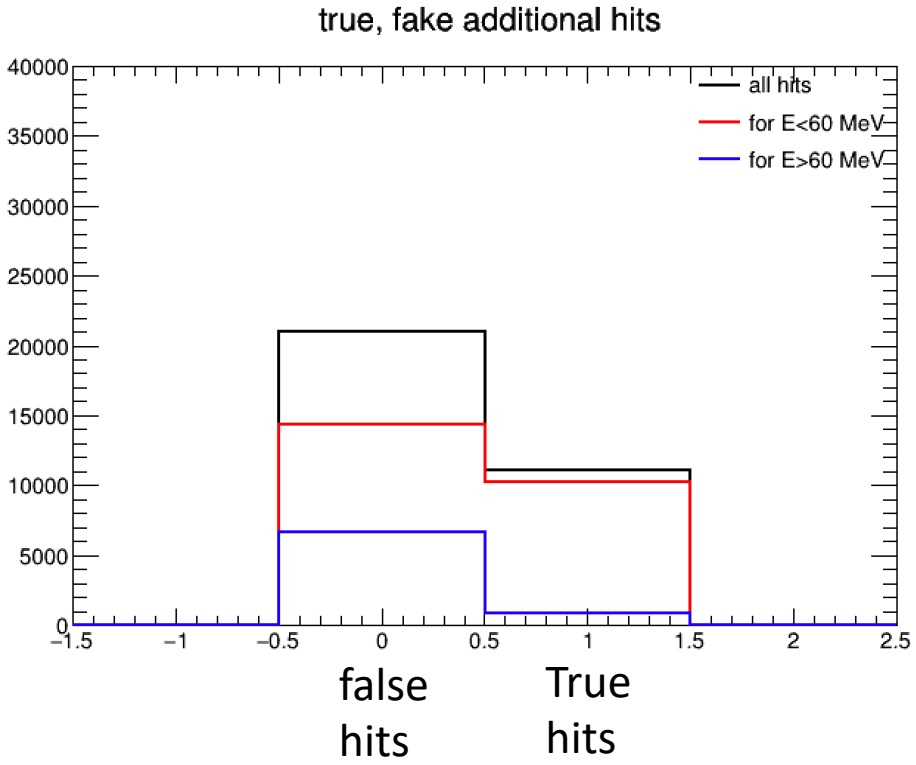




Sample hit purity using additional hits from parent muon

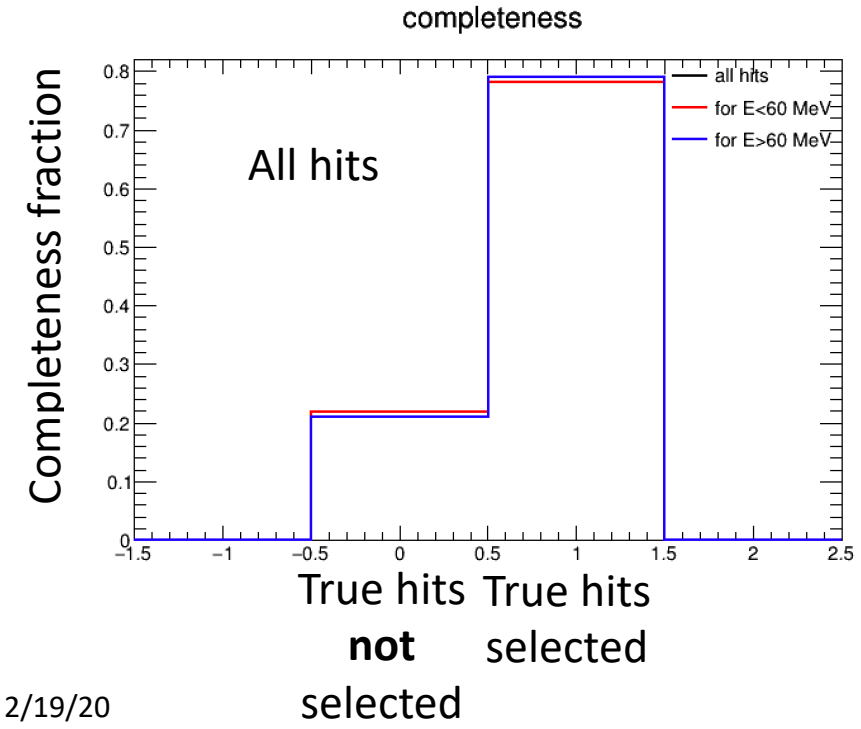
- Sample hit purity using additional hits from parent muons)(35%)
- Hit purity for events with
  - E<60 MeV: 40%
  - E>=60 MeV: 15%

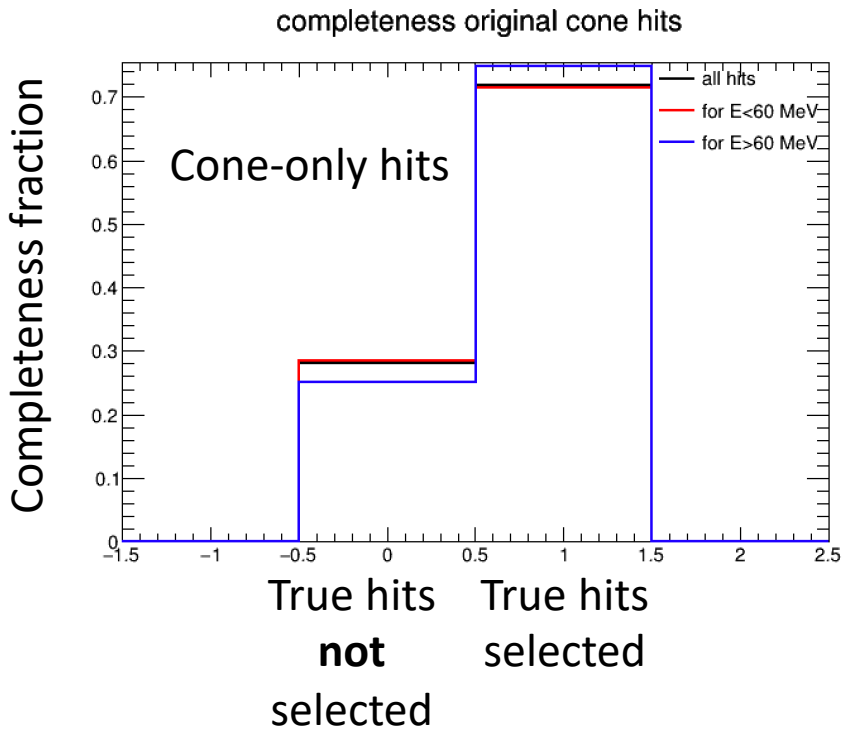
Most of the additional muon hits are **not true** Michel hits



Completeness

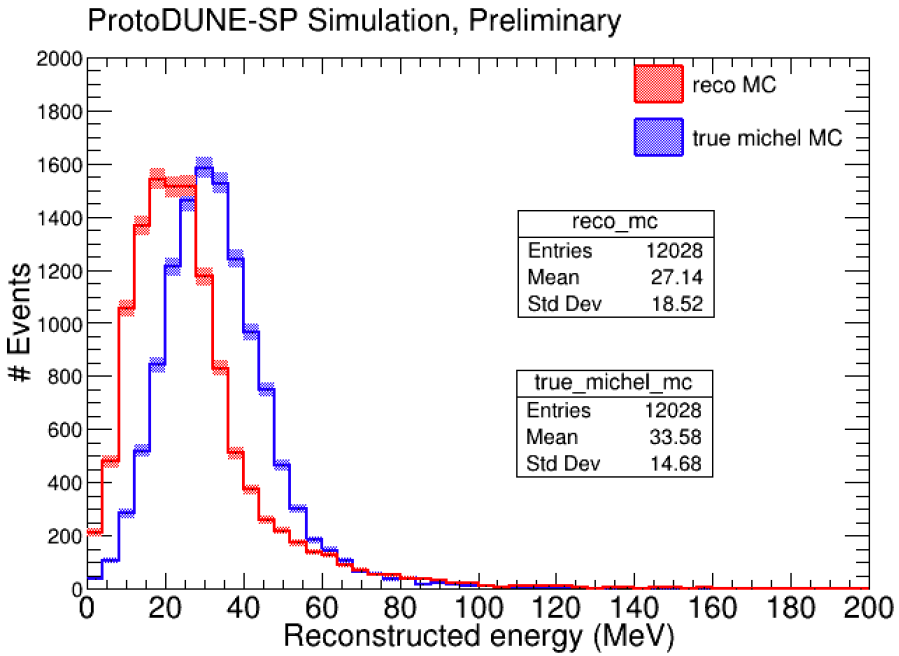
- Completeness: True Michel hits selected/True Michel hits total
- Completeness for “all hits” ~80% and “cone-only hits” ~75%



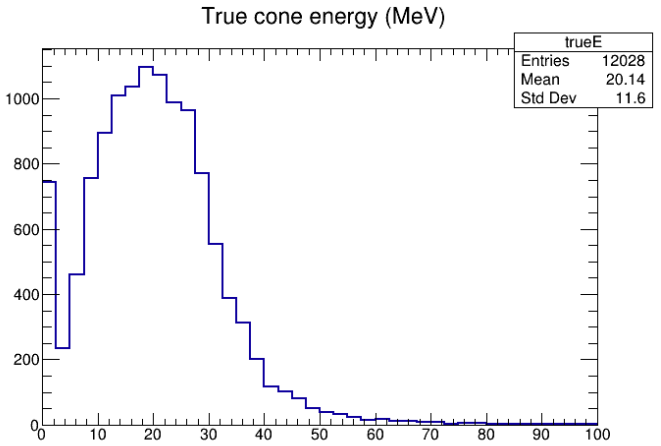


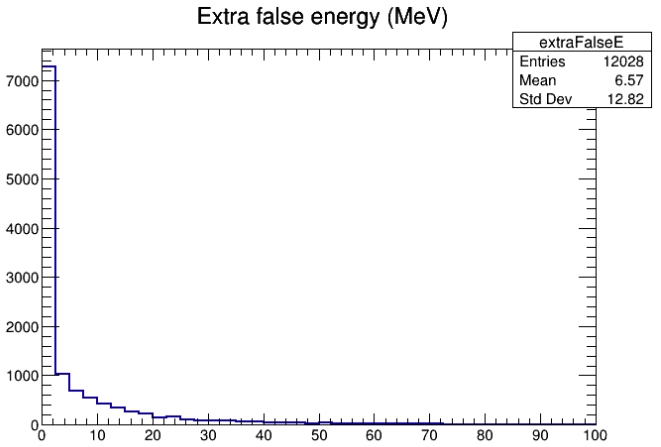
Michel reconstructed cone-only energy

Michel energy using cone-only hits



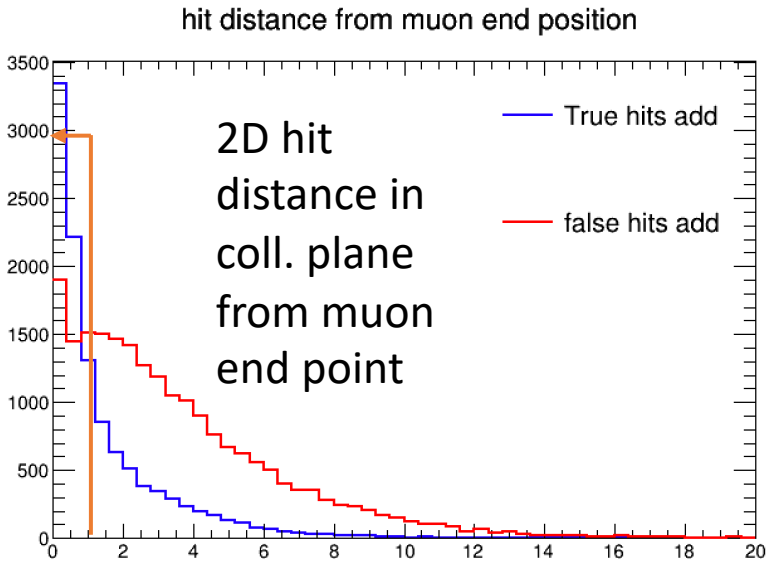
Red distribution consists of "True Michel hits + false hits "

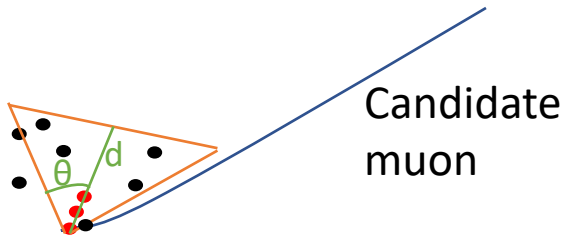


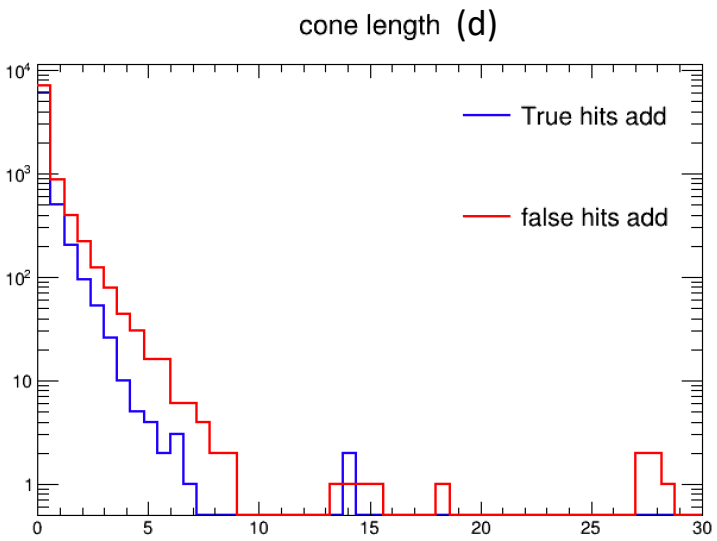


No tail: It mostly comes from additional parent muon hits

Additional muon hits properties

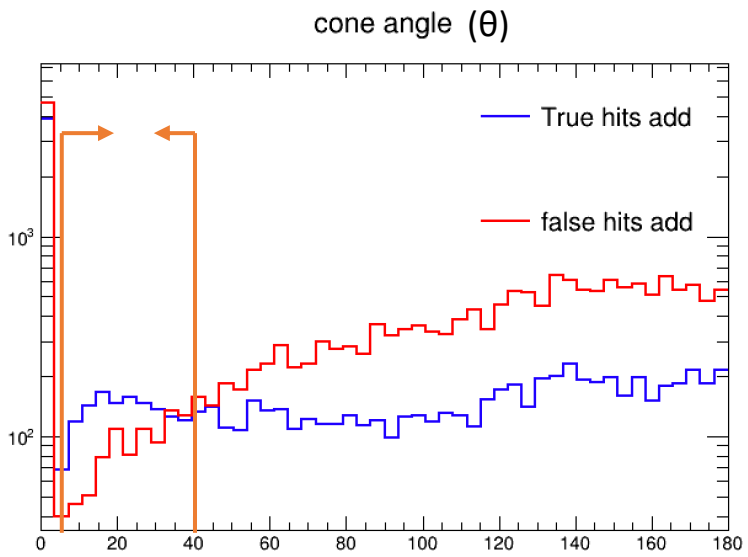






Applied an additional cut:

(Hit distance <=1 || 5<cone angle<40)

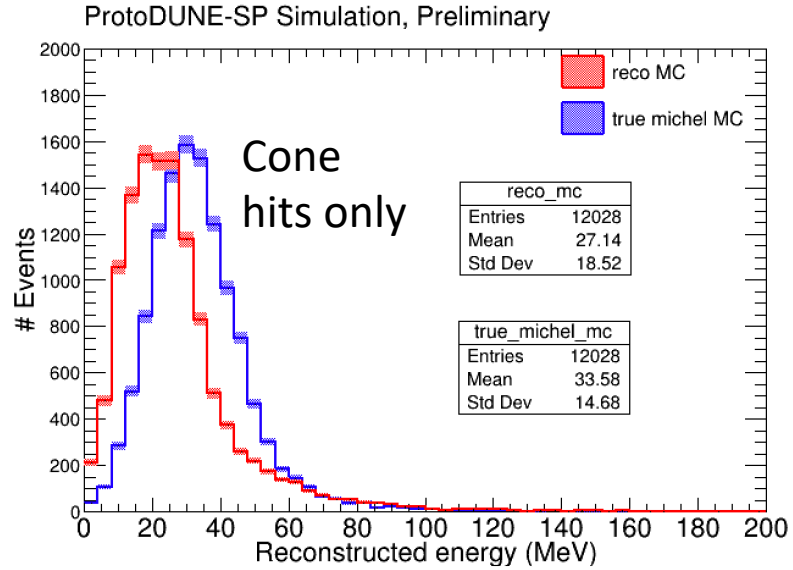


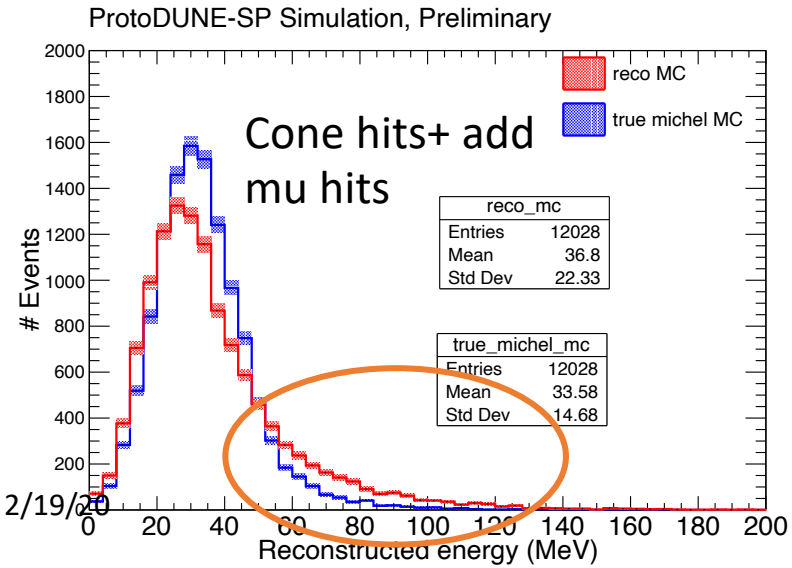
New Michel energy spectrum

From slide 7

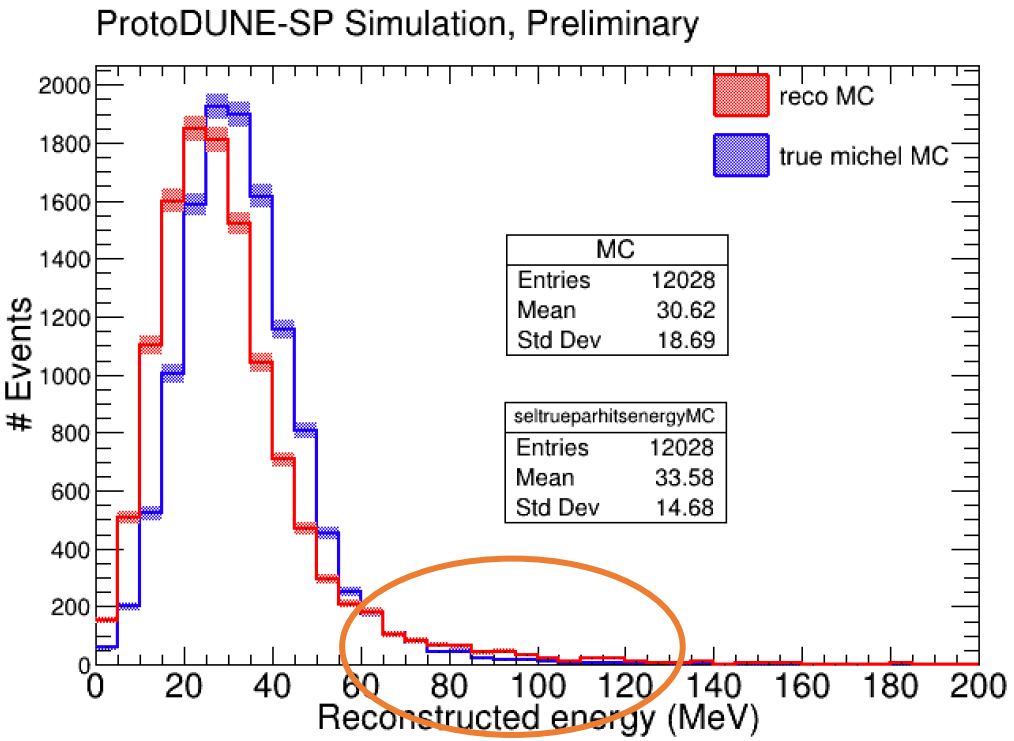
DocDB 17579

Before applying new cuts





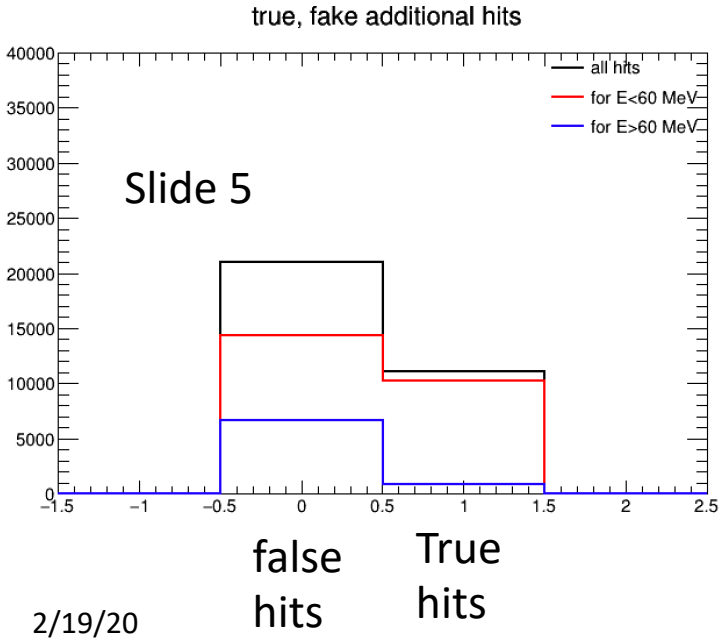
After applying new cuts



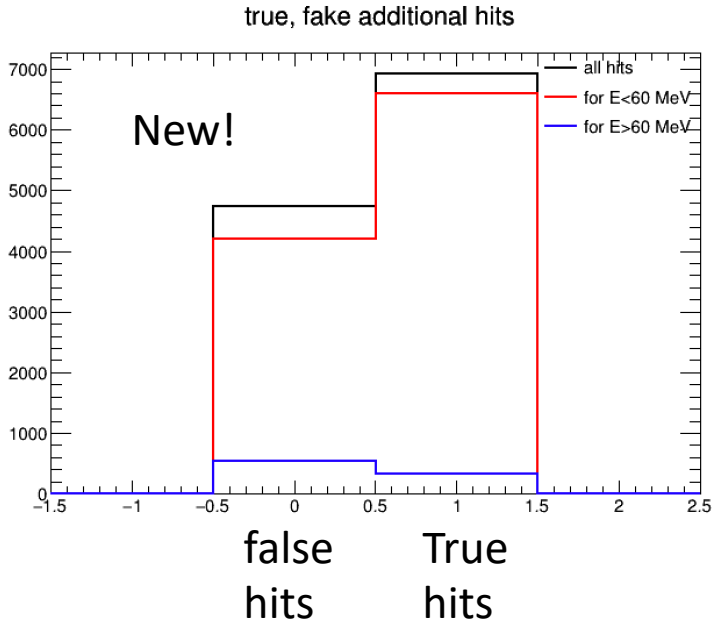
Better agreement without having any long tail

Updated additional hits purity

- Hit purity total (35%)
- Hit purity for events with
  - E<60 MeV: 40%
  - E>=60 MeV: 15%

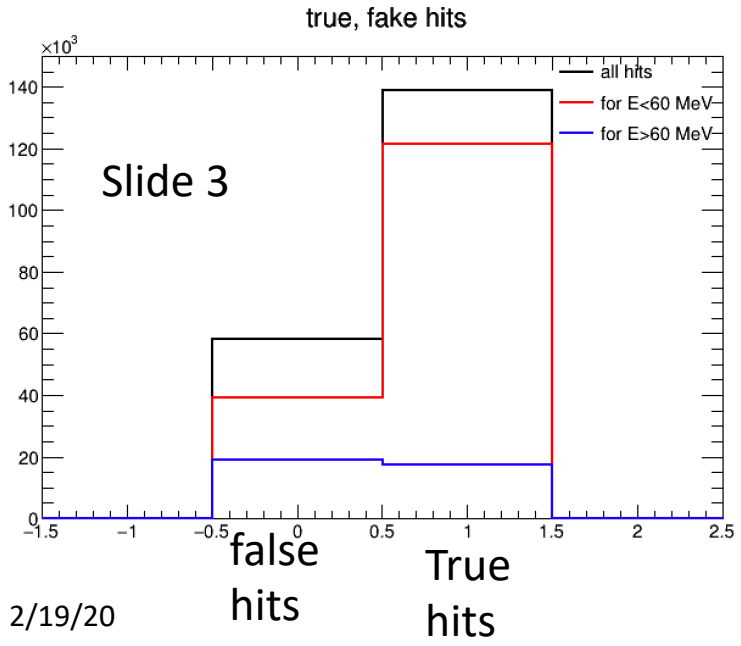


- Hit purity total (59%)
- Hit purity for events with
  - E<60 MeV: 61%
  - E>=60 MeV: 40%

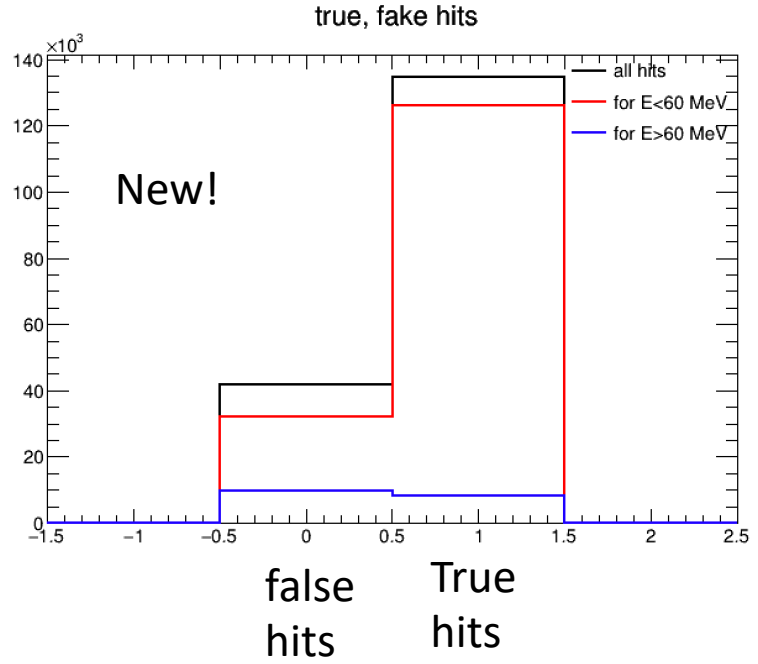


Updated total hit purity

- Hit purity (70%)
- Hit purity for events with
  - E<60 MeV: 77%
  - E>=60 MeV: 45%

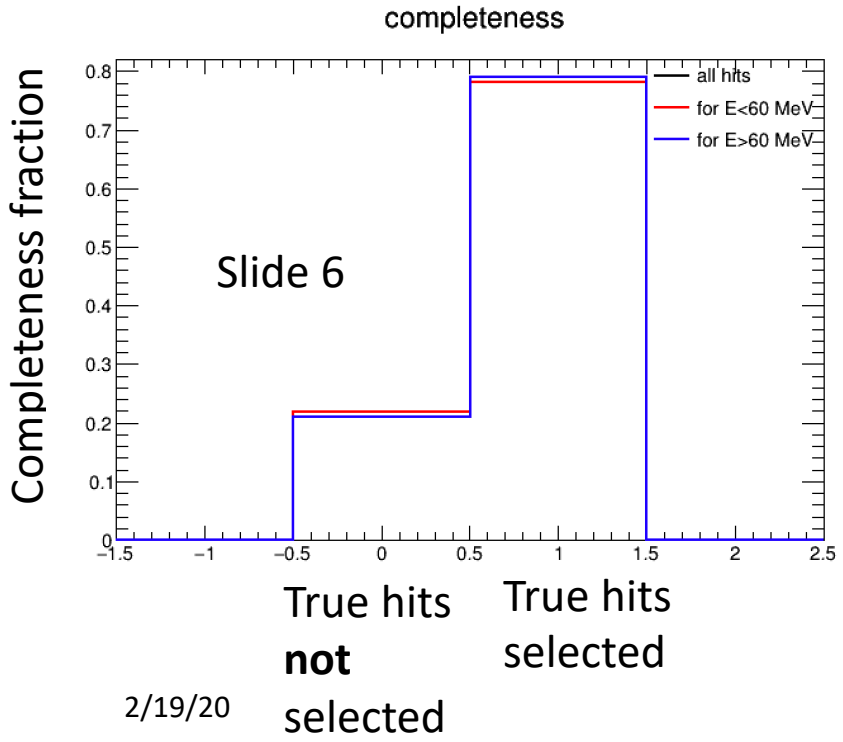


- Hit purity (76%)
- Hit purity for events with
  - E<60 MeV: 80%
  - E>=60 MeV: 45%

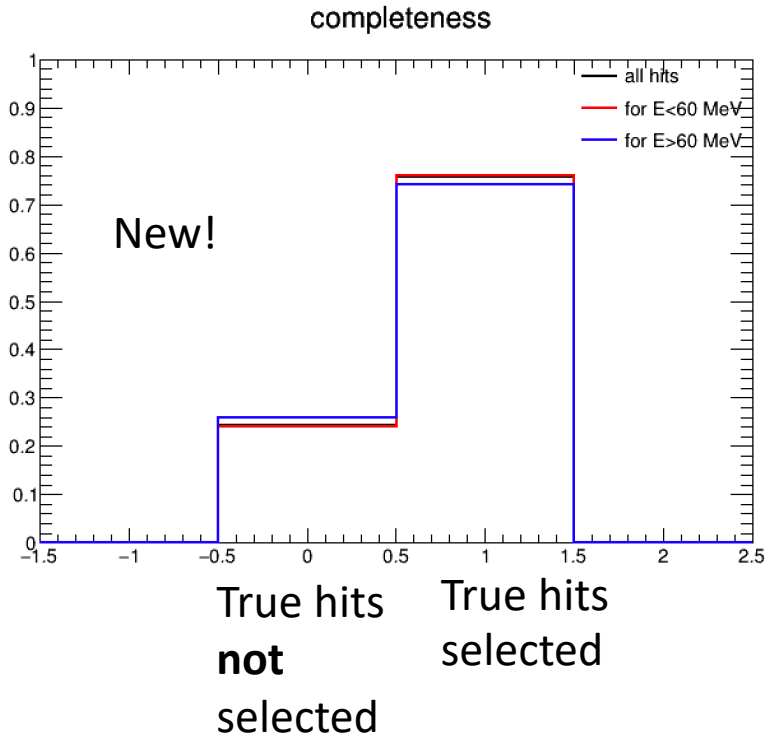


New total hit completeness

- Hit completeness (80%)



- Hit completeness (76%)



Michel energy data and MC comparison

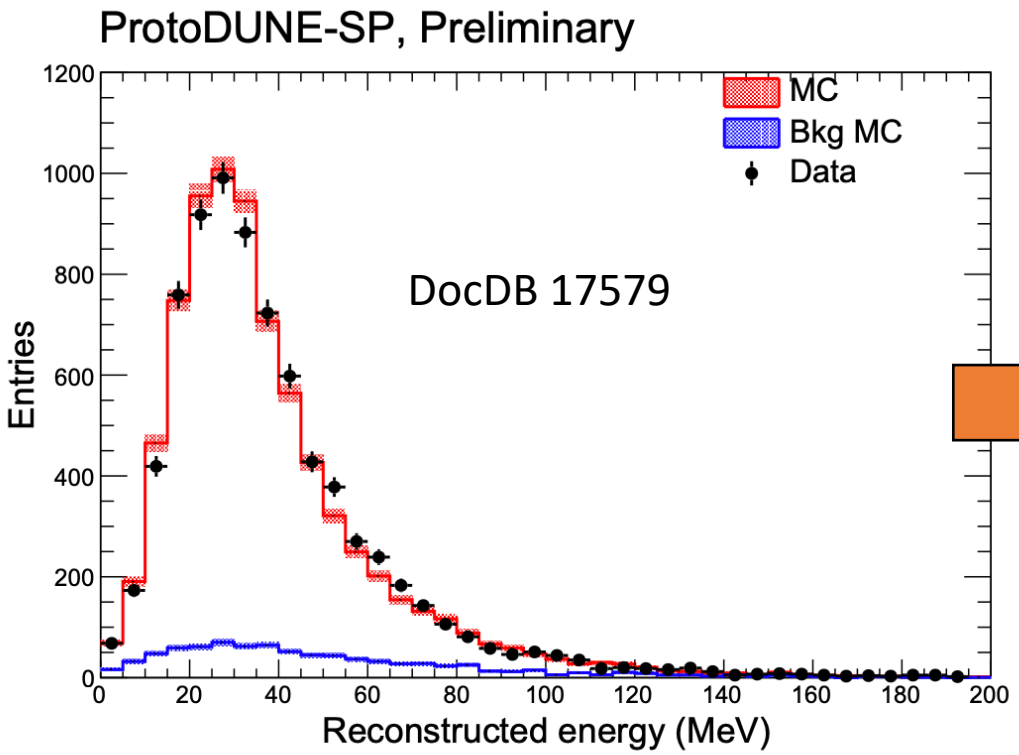
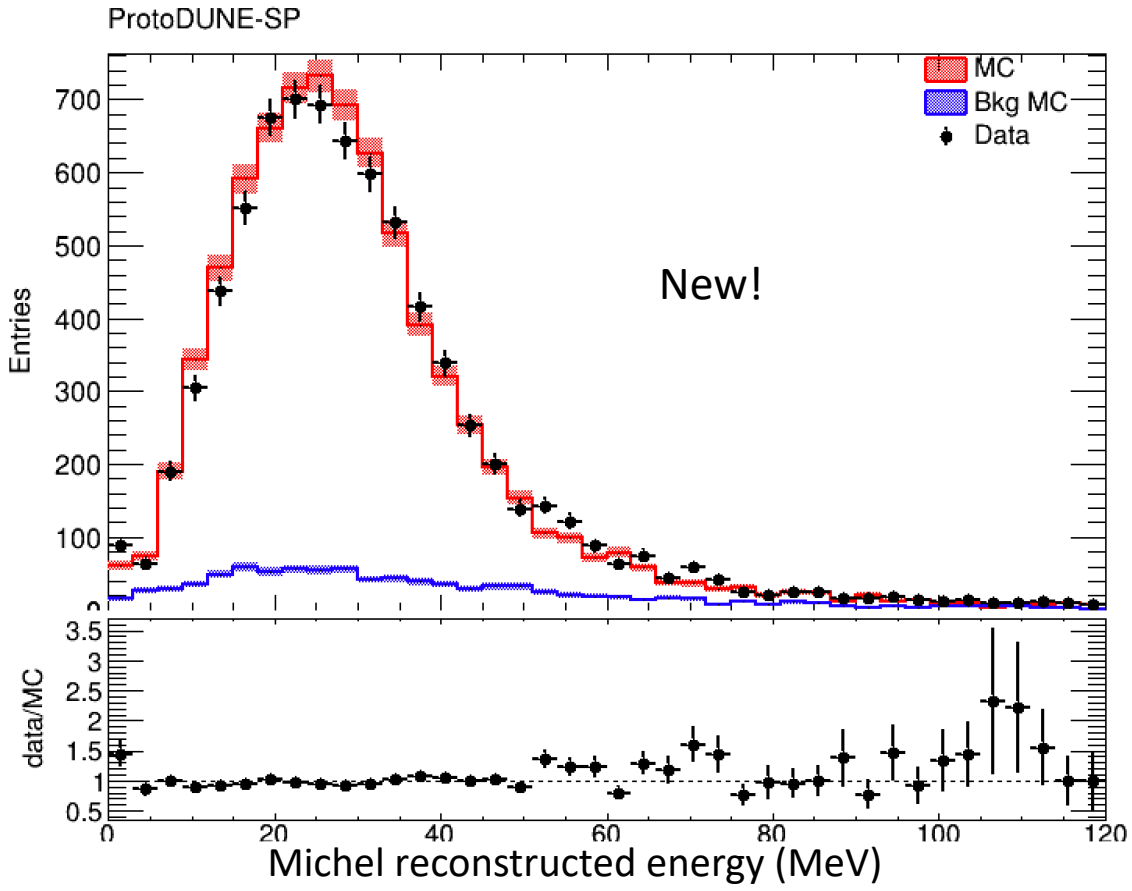
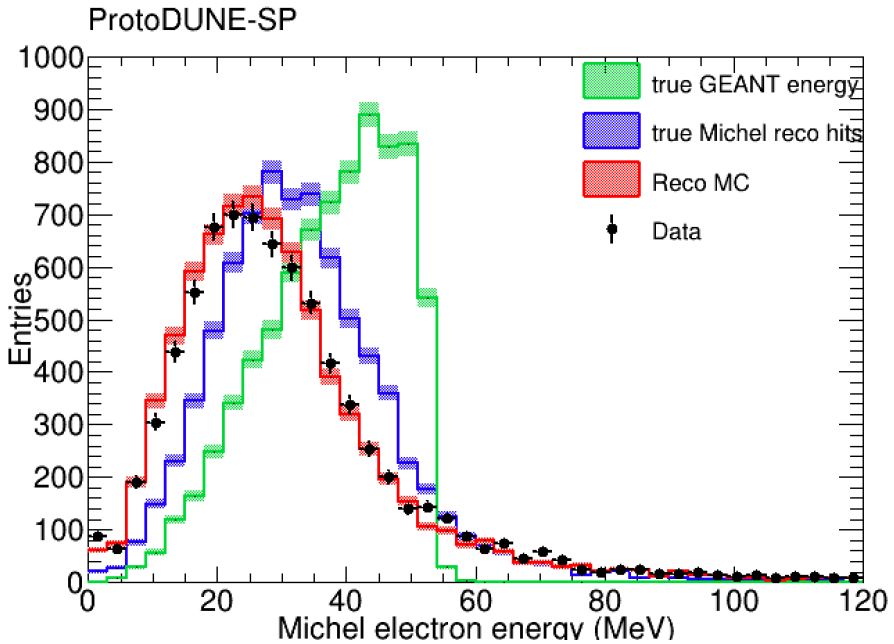


Figure 16: Michel energy spectrum from ProtoDUNE data and simulation.



Michel Energy calculations

- True energy:
  1. True GEANT energy:
     - Using “simb::MCParticle.E()”
     - Without any detector dependence, no hit information involved. It’s a total energy of Michel directly from GEANT.
  2. True Michel reco hits:
     - Reconstructed energy of a **true** Michel electron
     - Obtained by looping over all reconstructed hits of a **true** Michel of an event. Then applying reconstructed energy calibrations to convert hit charge into energy.
- Reconstructed energy:
  - Obtained by constructing a cone and assuming all hits in the cone belong to Michel electrons also by adding a few additional parent muon hits. Then applying reconstructed energy calibrations to convert hit charge into energy.
- Adding detector simulation and calorimetric effects changes the true GEANT energy to true Michel reco hits energy



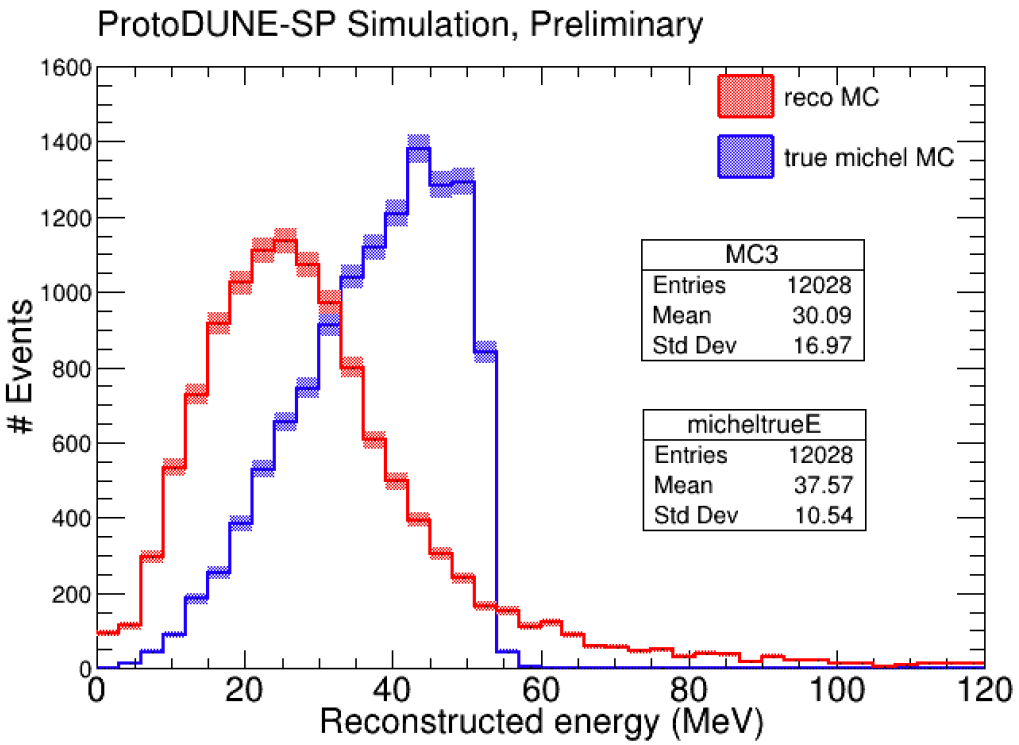
After adding all detector simulation, calorimetric and reconstruction effects to both data and MC, we observe a very good agreement. This indicates that the simulation is modeling the physics of muon decay and Michel electron propagation in liquid argon correctly.

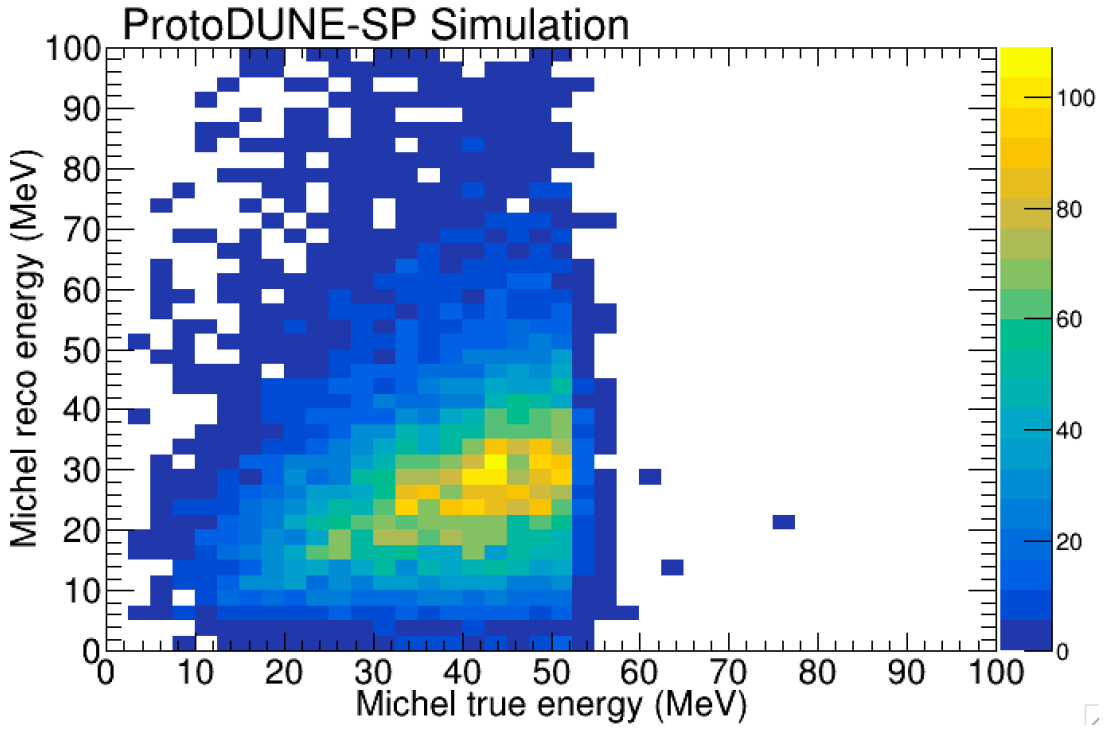
Scope of the Michel paper

- Suggested content
  1. Reconstruction of Michel electrons using MC in ProtoDUNE
  2. Compare simulation with ProtoDUNE data
  3. Energy resolution studies
  4. Perform systematic studies to understand sources of finite resolution
     - by changing various parameters slightly (~10%) such as calibration const, normalization factor, xyz corrections, recombination const etc, and a few selection cut parameters.
  - Other suggestions?

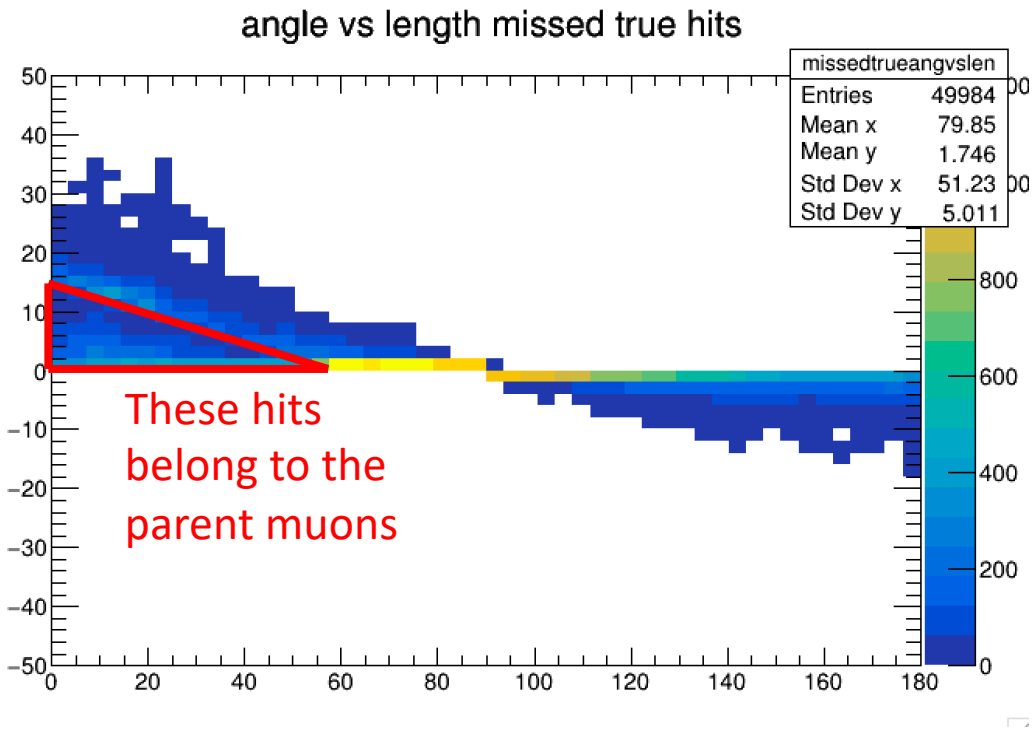
backup slides

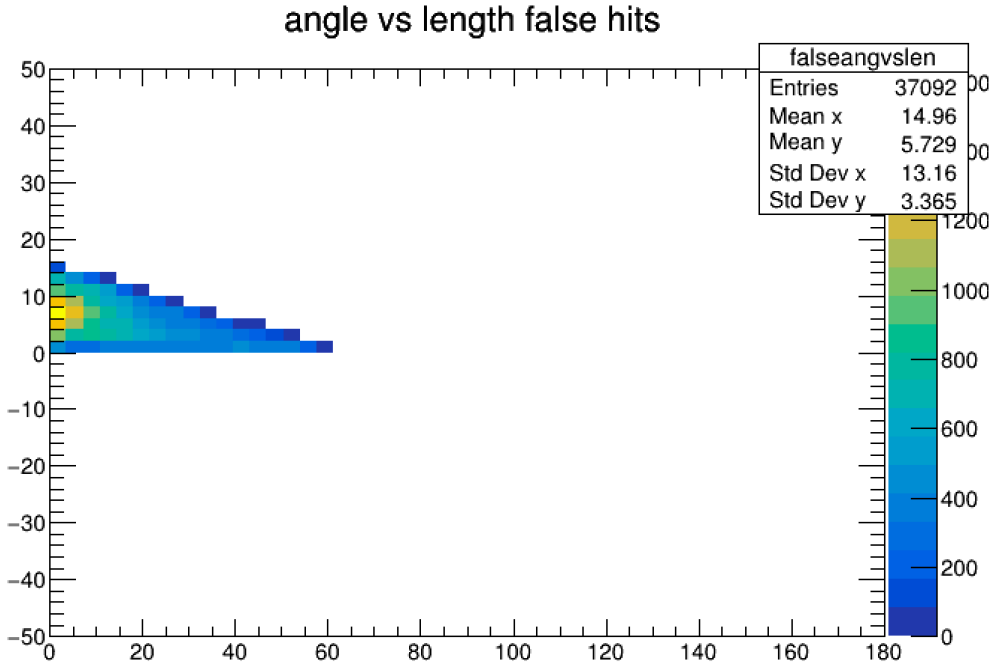
Michel true energy directly form GEANT





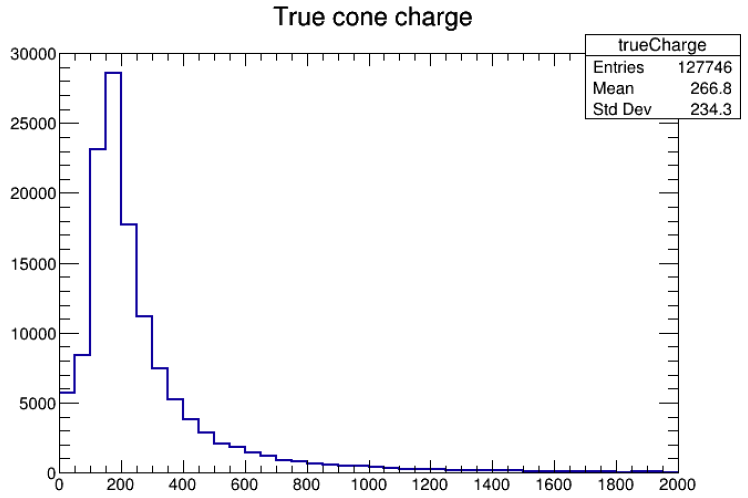
Cone angle vs cone length

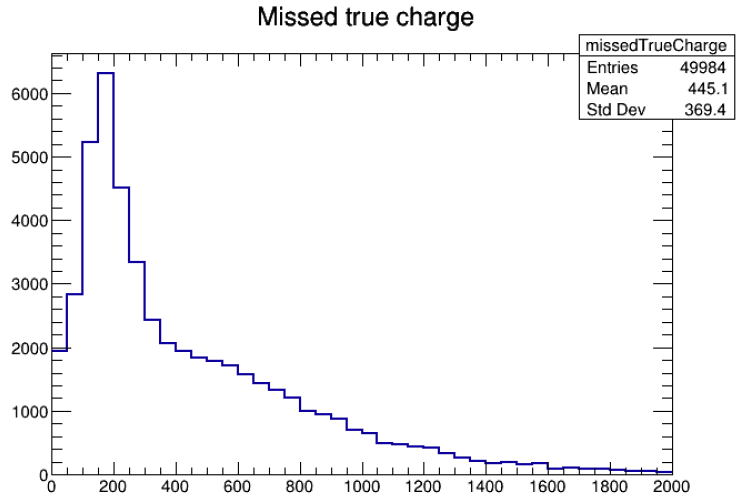


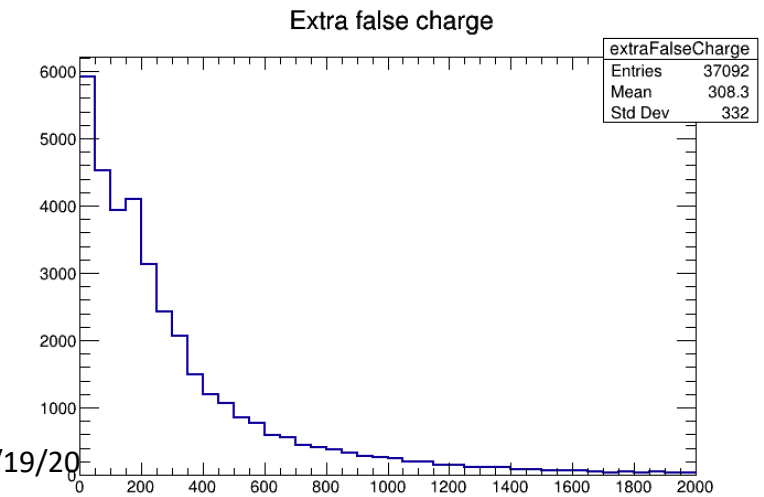


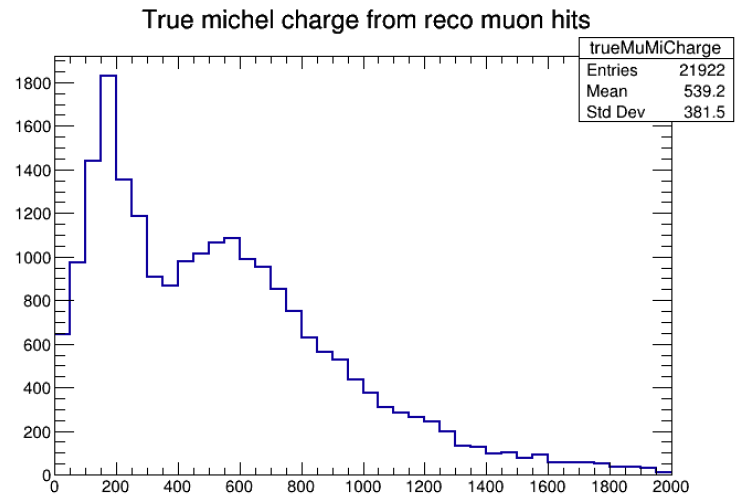
Look at muon last few hits and find a better solution to find more true hits

Hit charge

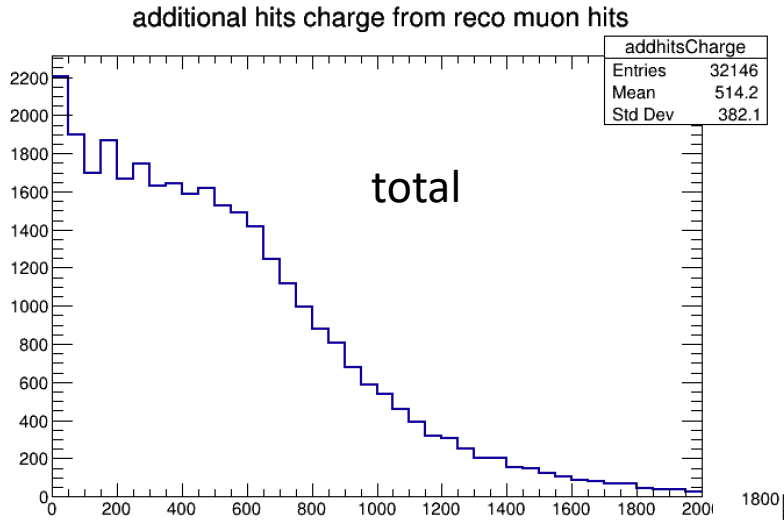

True cone charge
trueCharge
Entries 127746
Mean 266.8
Std Dev 234.3

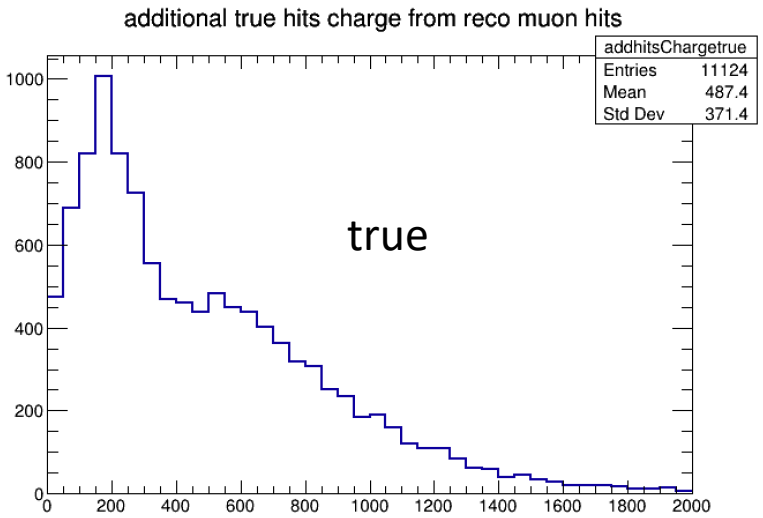

Missed true charge
missedTrueCharge
Entries 49984
Mean 445.1
Std Dev 369.4

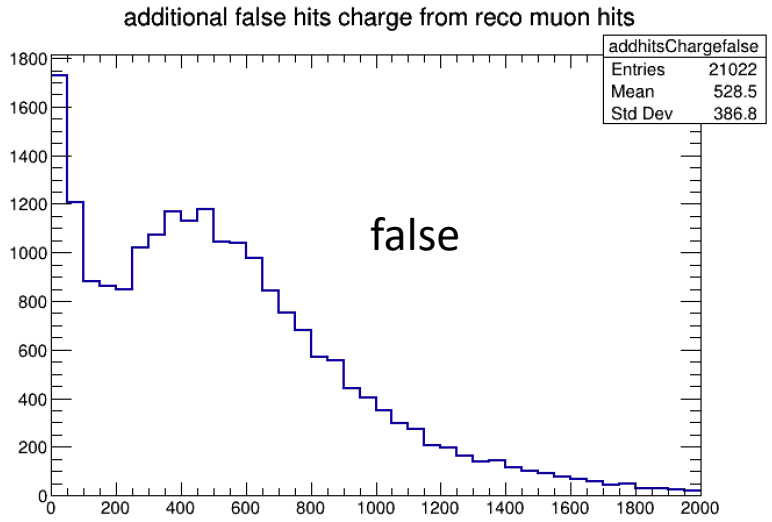

Extra false charge
extraFalseCharge
Entries 37092
Mean 308.3
Std Dev 332


True michel charge from reco muon hits
trueMuMiCharge
Entries 21922
Mean 539.2
Std Dev 381.5

Hit charge (cont.)

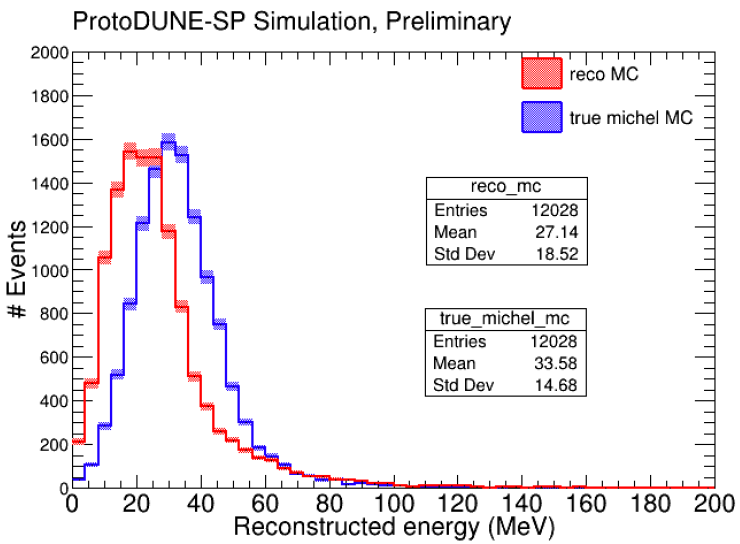




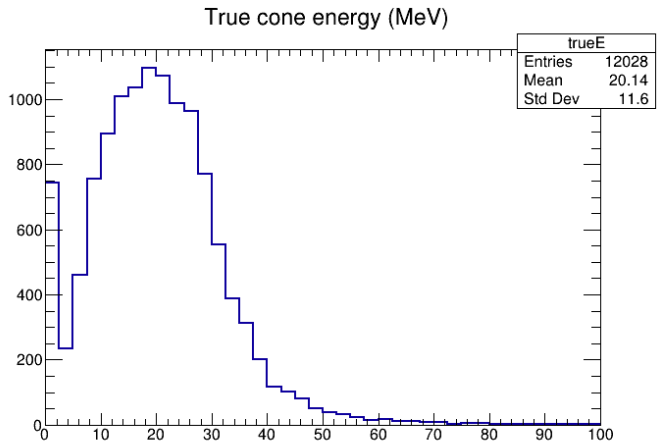


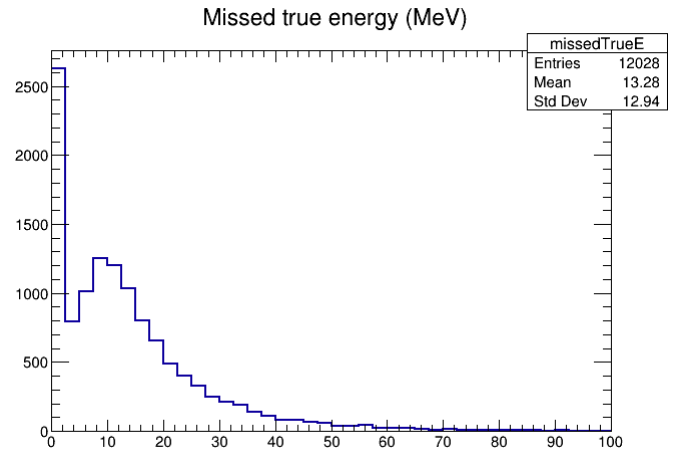
Michel reconstructed cone-only energy

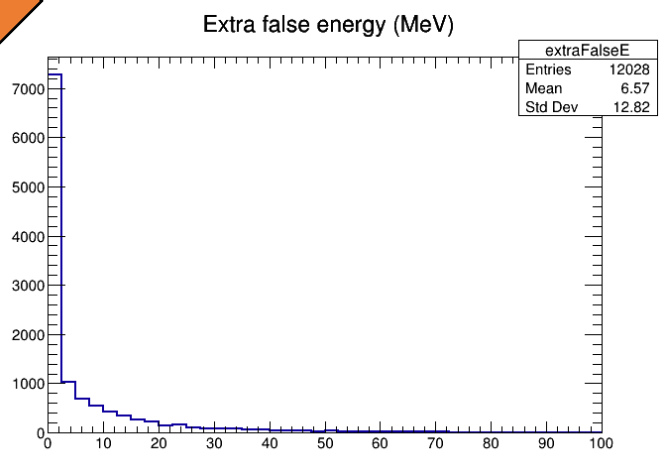
Without adding any hits from muons

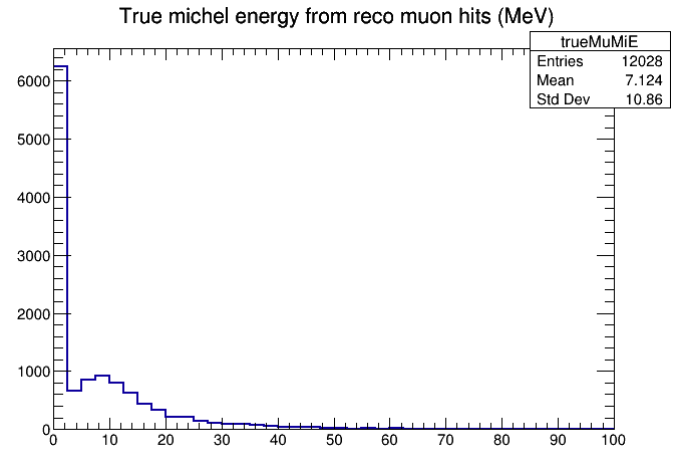


Red distribution consists of “True Michel hits + false hits ”









How to analyze data in DUNE/ProtoDUNE

- A reconstructed file contains all the information, but it is hard to know what variables are stored until you write an analyzer module and run the module over an input reco root file.
- First make a directory in …/src/dunetpc/dune (remember to add the directory in the parent CMakeLists.txt)
- This directory should contain at least three files: analyzer module itself with extension .cc; a fhicl file that is used to run the module with extension .fcl; and a CMakeLists.txt that you can copy from other similar directories
- Example directory is here: /dune/app/users/arafique/v08_30_02/srcs/dunetpc/dune/DiffTPCPDStime/
- You need to add/modify “DiffTPCPDStime_module.cc” after copying it to your local area on dune gpvm.
- A typical module file contains a bunch of includes and classes. You can add more to it.
- To run a module, do:

lar –c <fhich_file.fcl> -s <input_reco_file.root>

Extracting hit-related information

```
// * hits
 art::Handle< std::vector<recob::Hit> > hitListHandle;
 std::vector<art::Ptr<recob::Hit> > hitlist;
 if (evt.getByLabel(fHitsModuleLabel,hitListHandle))
   art::fill_ptr_vector(hitlist, hitListHandle);

const size_t NHits     = hitlist.size(); // number of hits

   for (size_t i = 0; i < NHits; ++i)
   {//loop over hits
     fData->hit_channel[i] = hitlist[i]->Channel();
     fData->hit_tpc[i]   = hitlist[i]->WireID().TPC;
     fData->hit_plane[i]   = hitlist[i]->WireID().Plane;
     fData->hit_wire[i]    = hitlist[i]->WireID().Wire;
     fData->hit_peakT[i]   = hitlist[i]->PeakTime();
     fData->hit_charge[i]  = hitlist[i]->Integral();
     fData->hit_ph[i]  = hitlist[i]->PeakAmplitude();
     fData->hit_startT[i] = hitlist[i]->PeakTimeMinusRMS();
     fData->hit_endT[i] = hitlist[i]->PeakTimePlusRMS();
     fData->hit_rms[i] = hitlist[i]->RMS();
     fData->hit_goodnessOfFit[i] = hitlist[i]->GoodnessOfFit();
     fData->hit_multiplicity[i] = hitlist[i]->Multiplicity();
 }
```

- You can also see what hit-related variables are stored in “recob::Hit” by looking at https://nusoft.fnal.gov/larsoft/doxsvn/html/classrecob_1_1Hit.html

How to analyze data in ProtoDUNE

- Look at ProtoDUNE data: https://wiki.dunescience.org/wiki/Look_at_ProtoDUNE_SP_data
- ProtoDUNE datasets: https://wiki.dunescience.org/wiki/ProtoDUNE-SP_datasets
- List of variables that can be looked at: https://cdcvs.fnal.gov/redmine/projects/uboonecode/wiki/AnalysisTree_variables#Flash-information