MC Truth Energy in LarSoft

Andrzej Szelc

The Problem

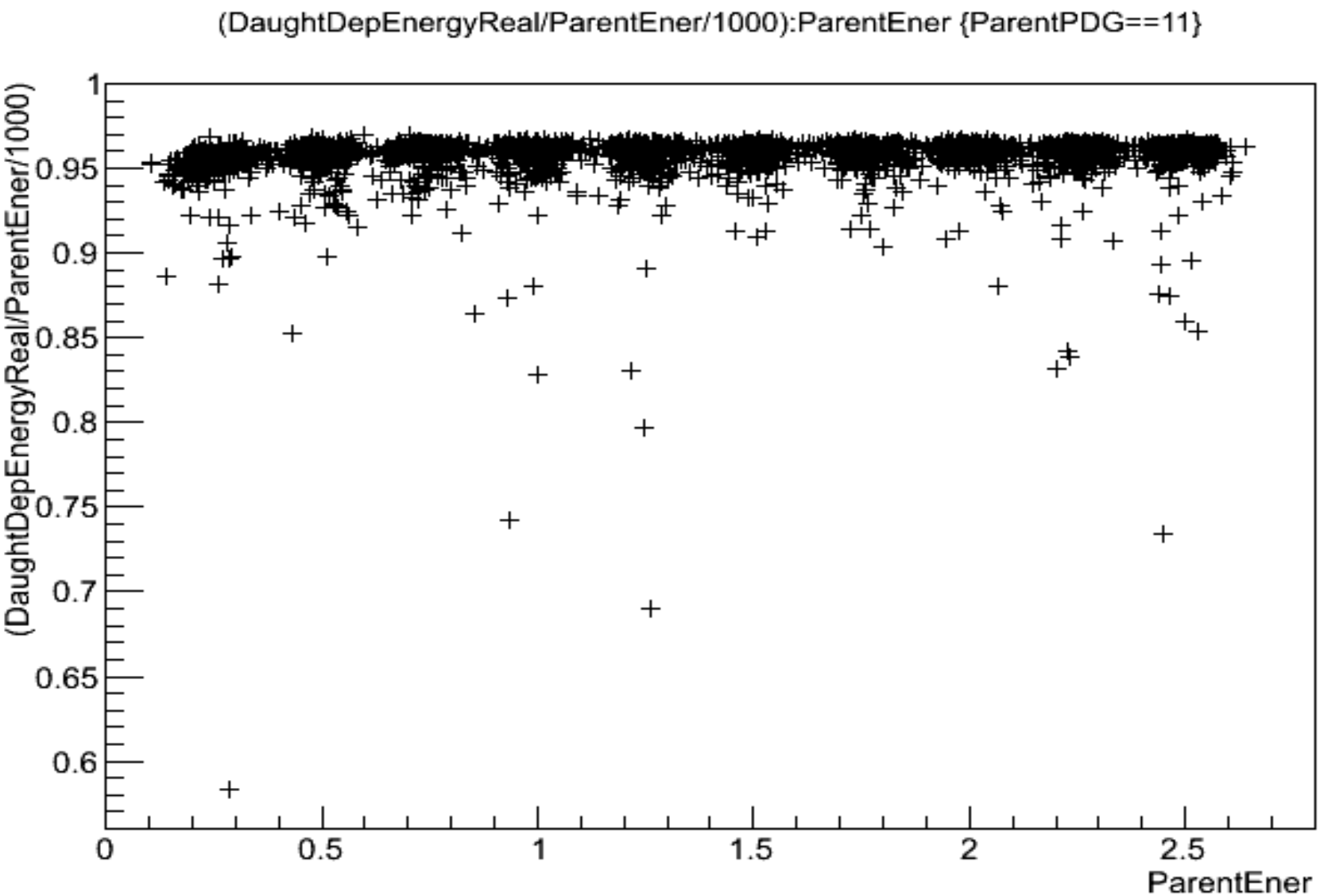
- After the VoxelList removal the energy was stored in sim::IDE objects. IDE stands for ID and Energy.
- These objects stored the number of electrons arriving at a given channel, the track ID from which they originated and the x,y,z coordinates where the energy deposit occurred.
- The number of electrons was already corrected for e-lifetime and Birks. In principle it would be possible to recalculate the energy from this, but would need to calculate dx at each point in track?

The Solution

- Add an energy member to the IDE class.
- The IDE calculation and assignment is somewhat complicated – accounts for track dispersion using clusters of electrons – means that a given (wire,tdc) pair may have charge originating from different geant4 steps → different IDEs, etc...
- The energy calculation mimics the IDE calculation – division into clusters then sums the components into the correct IDE objects.

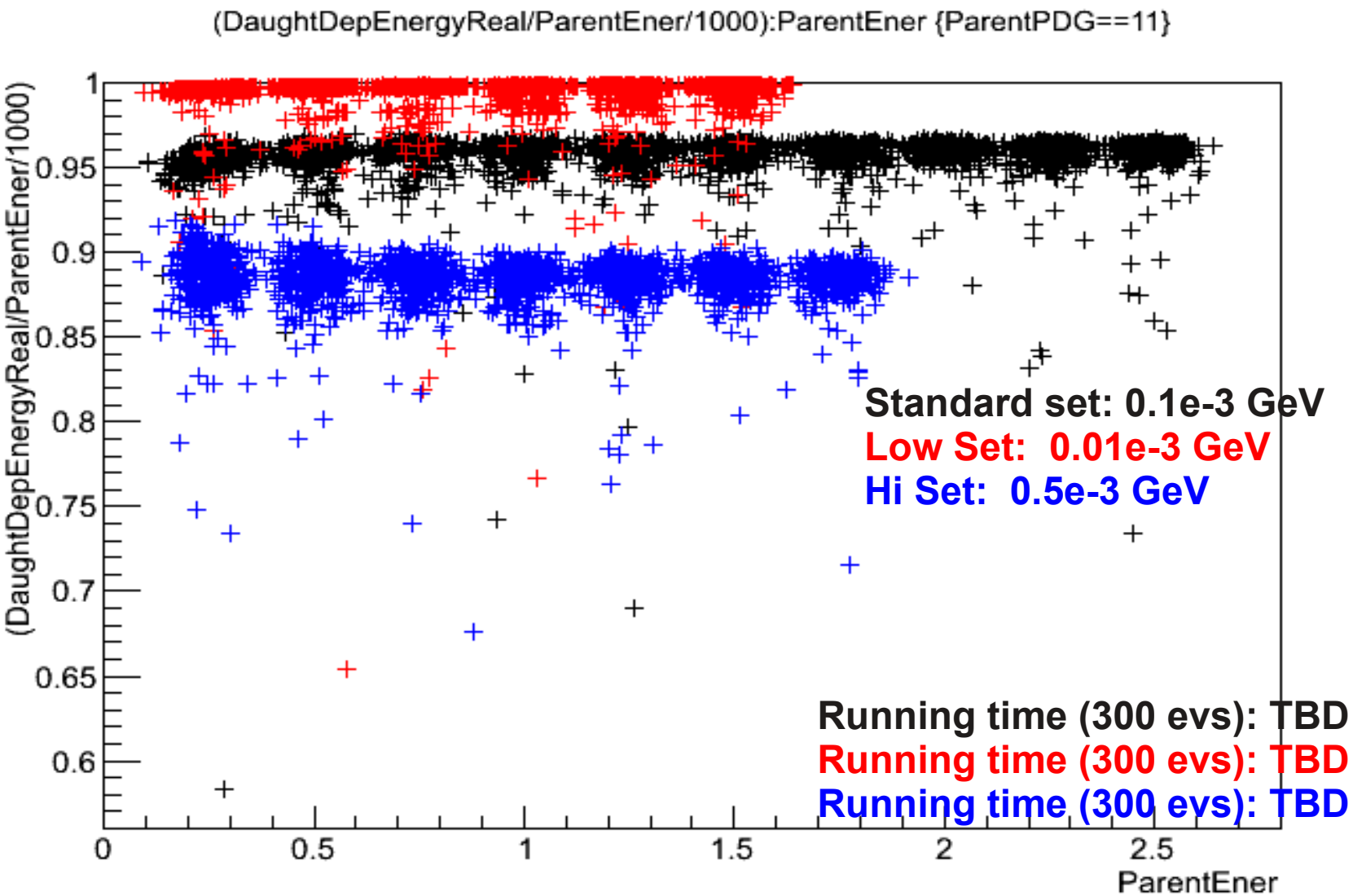
First Test

Showers generated in uBooNE geometry (should be contained)



Cross – check

(check the ParticleKineticEnergyCut setting)

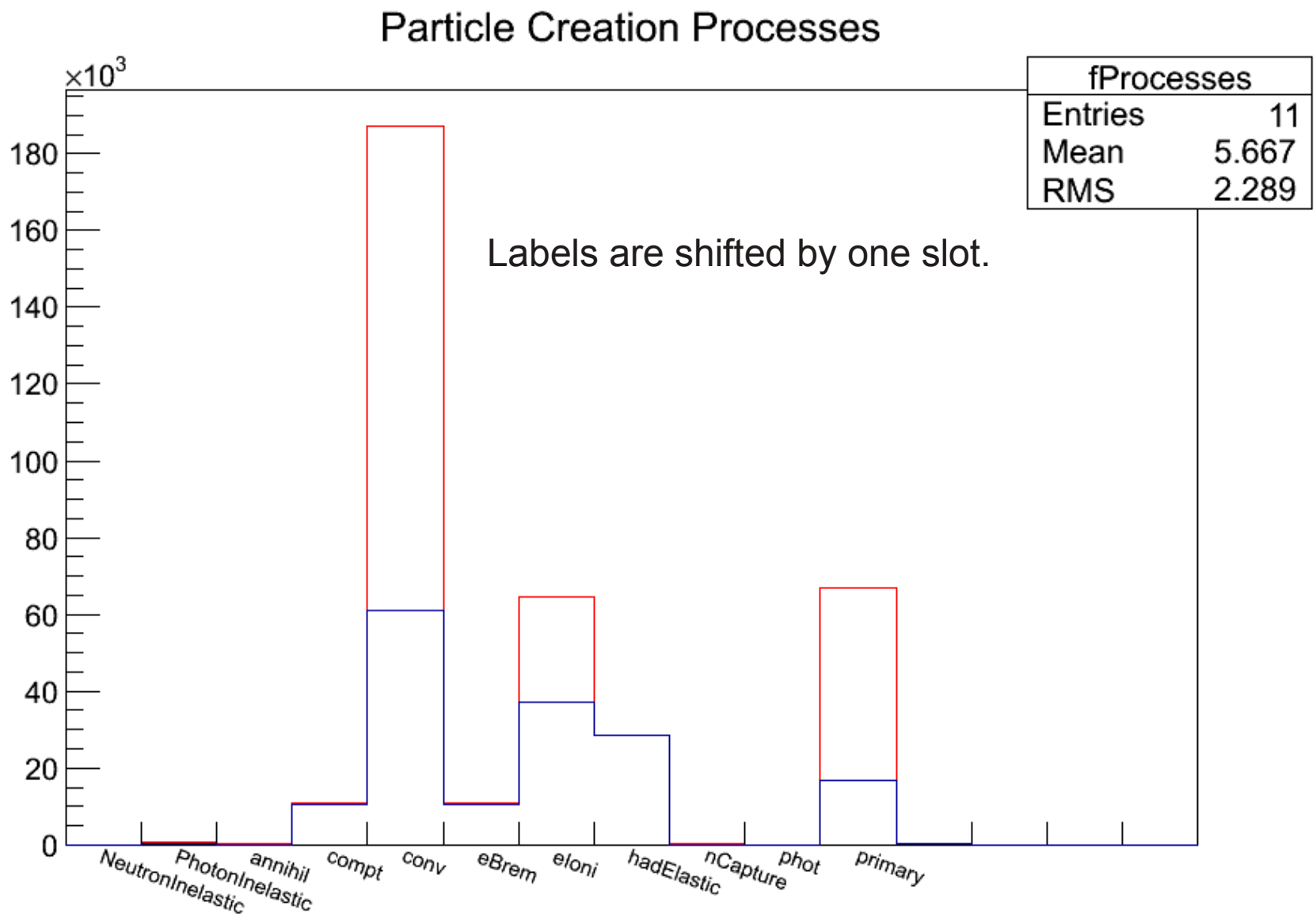


LArG4 50 evts Running Time:

Shower Energy:	1GeV	2GeV	3GeV
Cut @ 0.5e-3GeV	3866s	7958s	12421s
Cut @ 0.1e-3GeV	4595s	9179s	13878s
Cut @ 0.01e-3GeV	5146s	10136s	14111s

Not very scientific method, but it would seem the lower cut introduces a time hit of the order of ~10% at most.

What gets cut out.



How To Use

(e.g. in LArG4Ana.cxx)

```
double tot_energy=0.;
sim::ParticleList plist = sim::SimListUtils::GetParticleList(evt,fG4ModuleLabel);  // get list of particles
art::ServiceHandle<geo::Geometry> geom;
   std::vector<const sim::SimChannel*> sccol;
   evt.getView(fG4ModuleLabel, sccol);
// loop over all sim::SimChannels in the event
    for(size_t sc = 0; sc < sccol.size(); ++sc){
       //check whether the channel is collection - to make sure you sum only one wire plane.
       geom->ChannelToWire(sccol[sc]->Channel(),cstat,tpc,plane,wire);
       if(geom->Plane(plane).SignalType()!= geo::kCollection)
           continue;
// get map of IDE objects
       const std::map<unsigned short, std::vector<sim::IDE> >& tdcidemap = sccol[sc]->TDCIDEMap();
       std::map<unsigned short, std::vector<sim::IDE> >::const_iterator mapitr;
       for(mapitr = tdcidemap.begin(); mapitr != tdcidemap.end(); mapitr++){  // loop over the IDEMAP
         const std::vector<sim::IDE> idevec = (*mapitr).second;
          for(size_t iv = 0; iv < idevec.size(); ++iv){    // loop over the vector of IDE objects.
             if(plist.find( idevec[iv].trackID ) != plist.end()  //found the right track ID?
                      && idevec[iv].trackID != sim::NoParticleId)  {
                    // here you can put an if whether  idevec[iv].trackID==r particle TrackID  - if you want to calculate for said particle
                     tot_energy+=idevec[iv].energy;
             }  // end if plist
          } // end for loop on idevec
     } // end for loop on tdcidemap
   }// end for loop on sccol
```

Conclusion

- It is now easier to get true energy in LarSoft.
- This is new code, so stay vigilant when using it.
- Especially be careful with the ParticleKineticEnergy cutoff.