Tracking efficiency module

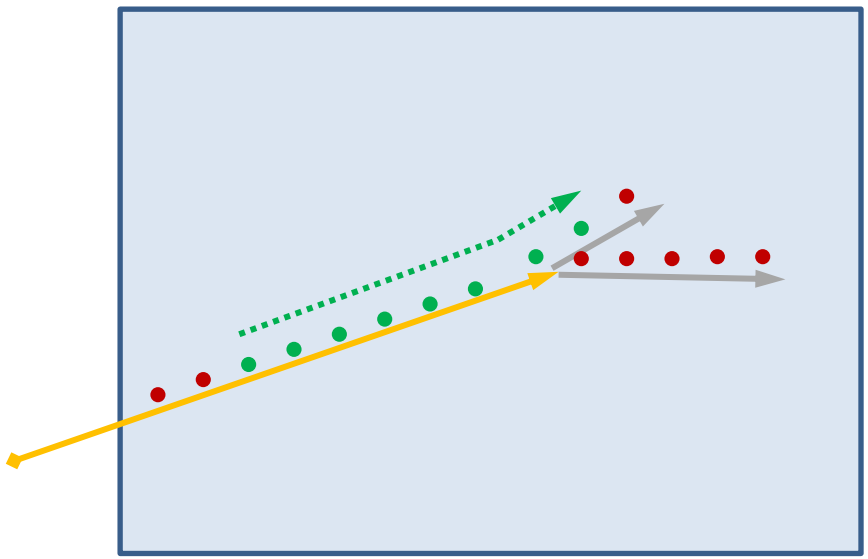
- a continuation of „write analyzer" exersise from the One day tutorial
- and a warm up before all the tasks we are going to try today/tomorrow

- **Inputs:** e.g. one of MCC9 sample

  /eos/experiment/neutplatform/protodune/npmcproddisk/mcc9/v06_33_00_01/mergeana

- **What is illustrated:**
  - one of many possible reconstruction efficiency metrics, applied to tracking
  - accessing several reco objects, navigating associations, using MC truth
  - your task will be to add selection of MC&reco objects used to compute the metric
  - in the end: access and display result in ROOT or SWAN

Efficiency definition implemented in the module



MC particle

reco hits

reco track

1) energy collected in the reco track is >50% from one of MC particles
2) reco track covers >50% of the energy deposited by that MC particle

$$\text{efficiency} = \frac{\text{\# reco tracks matched to MC particles}}{\text{\# MC particles}}$$

maybe not all particles are interesting?

Efficiency definition implemented in the module

$$\text{efficiency} = \frac{\text{\# reco tracks matched to MC particles}}{\text{\# MC particles}}$$

Module calculates the efficiency = f(# hits associated to MC particle).

Interesting MC particles:

- here: „reconstructabe" are selected ⇔ produced at least *N* hits in at least 2 planes
  - *N* is configurable in the job configuration (.fhc file)
  - we look at the *clustering and tracking algorithms* efficiency, not *overall* efficiency
- option 1: try modifying the selection in order to look at cosmic muons only, or primary beam particles, or daughters of that primaries…

Reconstructed tracks aslo can be filtered:

- option 2: try modifying reco tracks selection in order to skip tracks tagget as electrons

Run the module

```
lar -c
/eos/experiment/neutplatform/protodune/groupdisk/software/hackathonJune2017/job/tracking_eff_test.fcl
/eos/experiment/neutplatform/protodune/npmcproddisk/mcc9/v06_33_00_01/mergeana/ ProtoDUNE_beam_3GeV_cosmics_3ms/ProtoDUNE_beam_3GeV_cosmics_3ms_1*_20170614T223740_merged0.root
-T trk_eff_result.root
```

job configuration file, you can copy it to your folder and modify

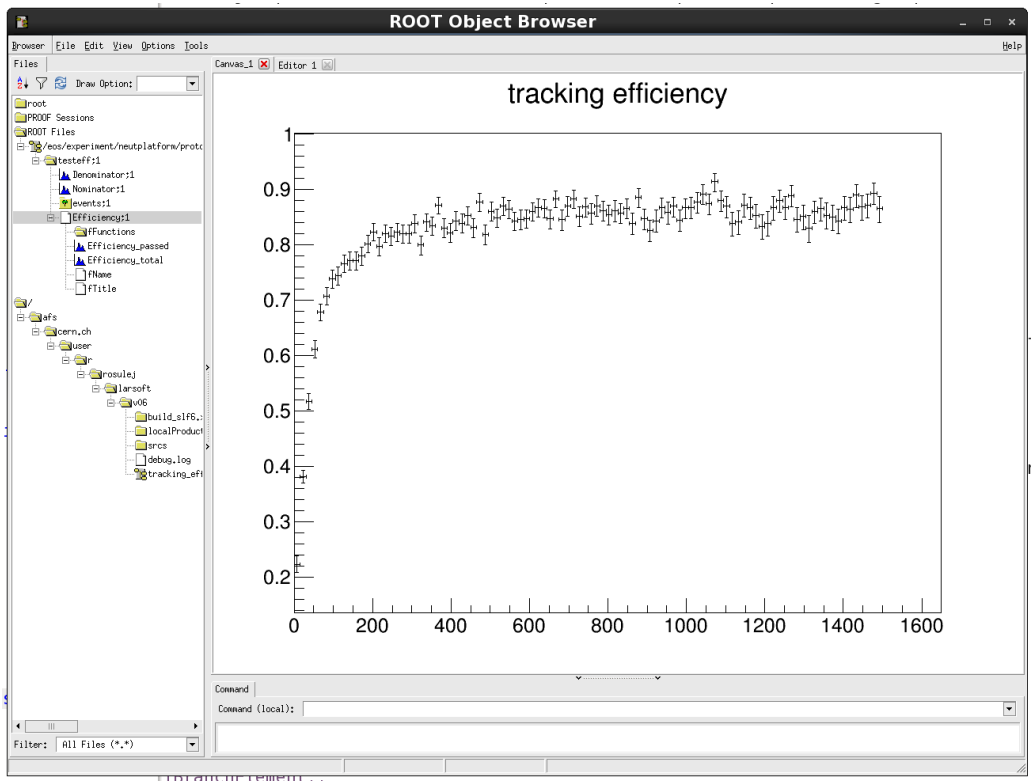
MCC9 samples stored in EOS, select only a few files with *, or you'll wait long…

specify name for the histogram ROOT file, created with TFileService (also in .fcl)

we are running analyzer module only, no output is specified with „-o" option

Look at the result

```
root trk_eff_result.root
root [1] TBrowser t;
root [2] TEfficiency* eff = (TEfficiency*)_file0->Get("testeff/Efficiency");
root [3] eff->Draw("AP");
```



Look at the result in SWAN

Login to https://cernbox.cern.ch (if not already done) – this will create CERNBox area for you (1TB on EOS, path is: /eos/user/*x*/*x*loginname/).

Visit SWAN page: https://swan.web.cern.ch

Open https://swan.cern.ch and launch the SWAN session.

or

Open in SWAN the notebook which is reading your efficiency results:
https://cernbox.cern.ch/index.php/s/F2T0ZoIsIQp2P6s

→ configure „Software stack" to „Development Bleeding Edge"

→ direct link to the example notebook will copy it to your CERNBox, you can edit your copy to open your ROOT file produced with efficiency testing module

→ nice way to code/plot with ROOT, but also mix with all the Python goods
→ tomorrow we are going to use more of such notebooks
→ we often share code examples in this way

Modify selections in the module

Edit the module code (srcs/dunetpc/dune/Protodune/singlephase/RecoEff_module.cc), there are some hints embedded in the analyse() function.

```
void proto::RecoEff::analyze(art::Event const & e)
{
…
 std::map<int, std::vector< recob::Hit > > mapTrackIDtoHits_filtered;
  for (auto const & p : mapTrackIDtoHits)
  {
    // *** here more conditions may be applied to select interesting MC particles, eg:
    // - check with BackTracker if this is a beam/cosmic particle:
    //auto origin = bt->TrackIDToMCTruth(p.first)->Origin();
    //if (origin == simb::kCosmicRay) { std::cout << "cosmic" << std::endl; }
    //else if (origin == simb::kSingleParticle) { std::cout << "beam" << std::endl; }
    //else { std::cout << "other" << std::endl; }
    // - check if this the process name is "primary" to select primary particles:
    //std::cout << bt->TrackIDToParticle(p.first)->Process() << std::endl;
    // - check if the mother of this MC particle is primary to select secondaries of the
    //primary interaction

    std::unordered_map<geo::View_t, size_t> hit_count;
    for (auto const & h : p.second) { hit_count[h.View()]++; }

    size_t nviews = 0;
    for (auto const & n : hit_count) {  if (n.second > fMinHitsPerPlane) { nviews++; } }
    if (nviews >= 2) { mapTrackIDtoHits_filtered.emplace(p); }
  }
  fReconstructable = mapTrackIDtoHits_filtered.size();
…
}
```

Recompile and run again the module

```
cd $MRB_BUILDDIR
make install -j4
…
cd ..
lar -c …
```