TruthTrackerAlg

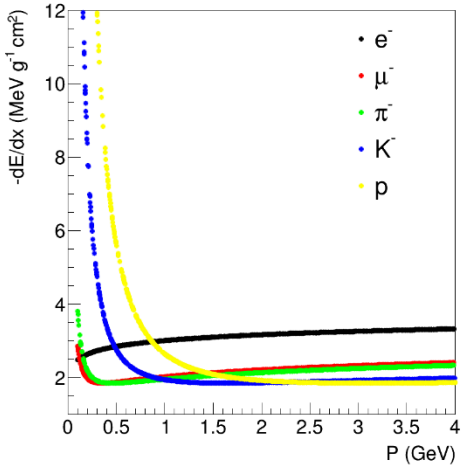
- An algorithm “TruthTrackerAlg” is created for track reconstruction based on MC information
- Input:
  - MCParticleCollection
  - TrackerHitCollection
- Output:
  - TrackCollection

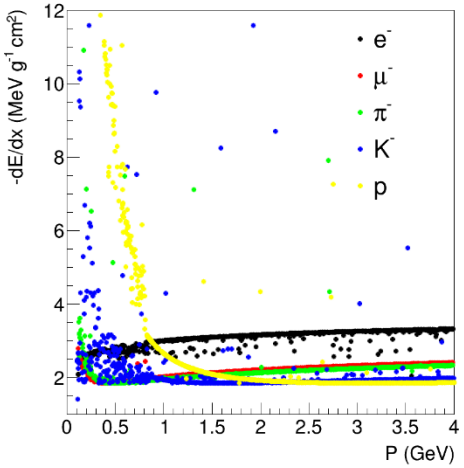
- Created a pull request: https://github.com/cepc/CEPCSW/pull/104
- Note:
  - This method only support single particle simulation currently.

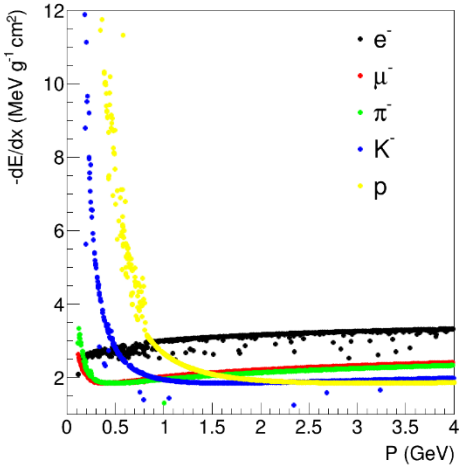
Use MC particle to create a track

- Using TrackerHit in this event to get dedx for this track (truncate mean method)
- The option of scale and smear of the track dedx is provided

TruthTrackerAlg





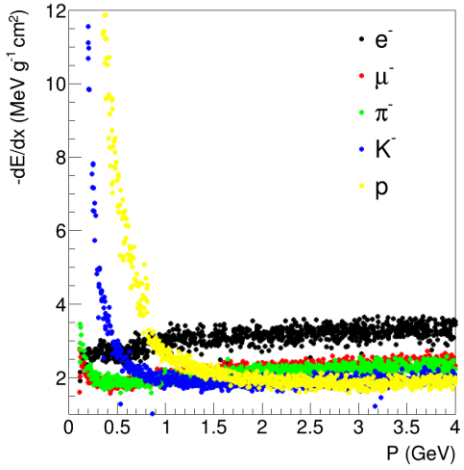


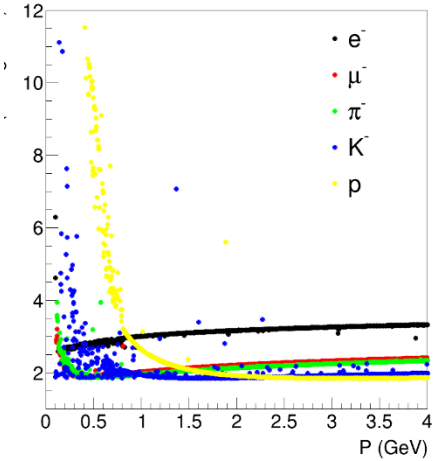
CRD

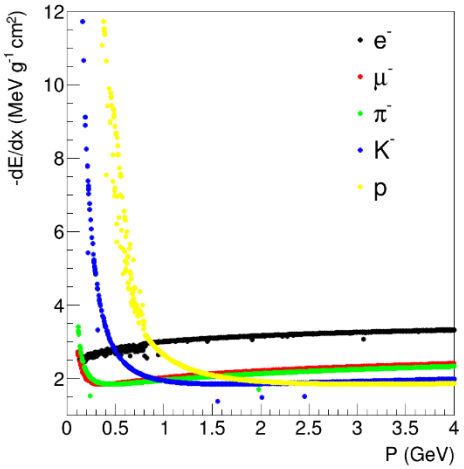
Theory

Track dedx

Track dedx (only save simTrackerHit from primary particle)

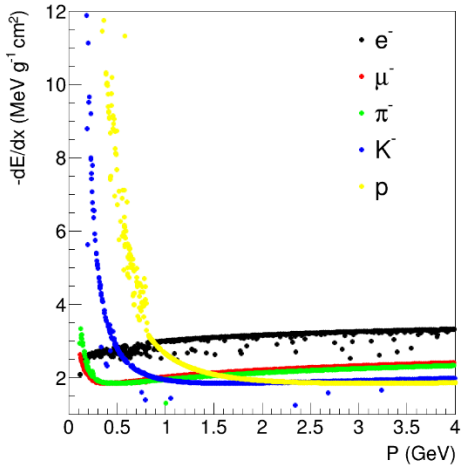




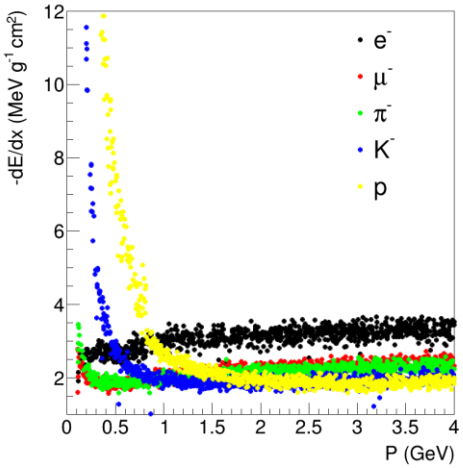


DCH

TruthTrackerAlg



Smear by 5%



CRD

Track dedx (only save simTrackerHit from primary particle)

Garfield simulation

>> sorry for my late reply. I would say it depends. There are some classes (like TrackHeed) that are not easily parallelizable (not sure if that's a word), others should be reasonably thread-safe and there also some that could probably be made thread-safe quite easily but we've not done it yet.
>> One thing that should work is
> - simulate a track, retrieve the coordinates of all primary electron/ion pairs along the track and store them in a vector,
> - write a parallelized for loop iterating over the primary electrons and create a separate AvalancheMicroscopic object for each of them.

The primary (or secondary) ionization electron is saved in SimTrackerHitCollection

- Using gaudi algorithm parallel technique to deal with SimTrackerHitCollection
- Each algorithm will get a portion of primary ionization electron from SimTrackerHitCollection and doing drift and avalanche. Finally write the induced current to edm (TPCHitColletcion)