

Design of `recob::Track`

Gianluca Petrillo

University of Rochester/Fermilab

LArSoft architecture review meeting, October 21st, 2015

Full header in [Redmine](#) or [Doxygen](#). Currently includes:

- trajectory points (collection of `TVector3`)
- punctual direction (collection of `TVector3`)
- punctual covariance matrix (collection of `TMatrix3`)
- punctual dQ/dx (nested collection, one value per plane per point)
- punctual momentum modulus (collection of `double`)
(documented: at least at begin and end of track, at will more)
- an integer ID

Currently provides:

- computation of projected length on a plane
- rotation matrices to a “local” track coordinate system

It typically has associated `recob::Hit` and `recob::Cluster`, maybe `recob::SpacePoint` too.

List of points to work on:

- the sizes of all those lists are not necessarily matching
- a lot of information, not every tracker can fill it
- lack of goodness of track information
- conceptually mixing track (measured elements), trajectory (spatial path) and calorimetry-related (momentum)
- some advanced methods require access to external information

What to do

```
/* TODO: Fill this slide with a proposal so that  
everybody will say "Ooh how could we live without  
it?" */
```

Just a few points to remind myself my opinion:

- I prefer the “dumb data product” approach: one constructor, accessors
- (firm point) need for external information must be removed
- I would like to split the information in
 - generic information (something about reconstructed trajectory)
 - algorithm-specific information (additional data products)
 - trajectory representation (3D points and direction)

all connected via associations