



## **Status of changes to recob::Track and related classes**

Giuseppe Cerati, Gianluca Petrillo, Erica Snider

LArSoft Coordination meeting

Jan. 31, 2017

# Introduction

- Update on top of the plans presented by Gianluca 2 weeks ago:
  - <https://indico.fnal.gov/getFile.py/access?contribId=1&resId=0&materialId=slides&confId=13672>
- Brief summary of what was presented:
  - two objects:
    - Trajectory (result of pattern reco): set of 3D points and directions/momenta
    - Track (result of fit): a Trajectory plus fitting information (covariance, PId, chi2)
  - implementation in 3 stages:
    - first stage with new classes but no interface change by this week
    - second stage with full definition of classes and breaking interface, basic tools
    - third stage: full support with fully defined user-friendly tools (façade)
- Many interesting comments during the discussion, in particular:
  - request for Track with inverse direction
    - cannot be provided by `recob::Track` interface, but fitter does it. 3rd stage will provide some functionalities to help with the task
  - bookkeeping of how hits and points were used inside the trajectory or track
    - this comment indeed triggered a change in the plans, see next slide
  - request for document about conventions, requirements and relations among products
    - not addressed yet

# TrackTrajectory object

- The only change with respect to the original plan so far is the introduction of a third object: `recob::TrackTrajectory`
- It inherits from `recob::Trajectory` and is stored in `recob::Track`
- It extends the simple `Trajectory` by adding a set of flags (bitmask)
- There is one bitmask for each hit/point pair
  - 1-1 correspondance (hit/point) becomes 1-1-1 (hit/point/flagset)
- Each flag can be in one of three states:
  1. the algorithm decided the flag condition is true (flag `_set_`)
  2. the algorithm decided the flag condition is false (flag `_unset_`)
  3. the algorithm declined to decide a state to the flag (flag `_undefined_`)

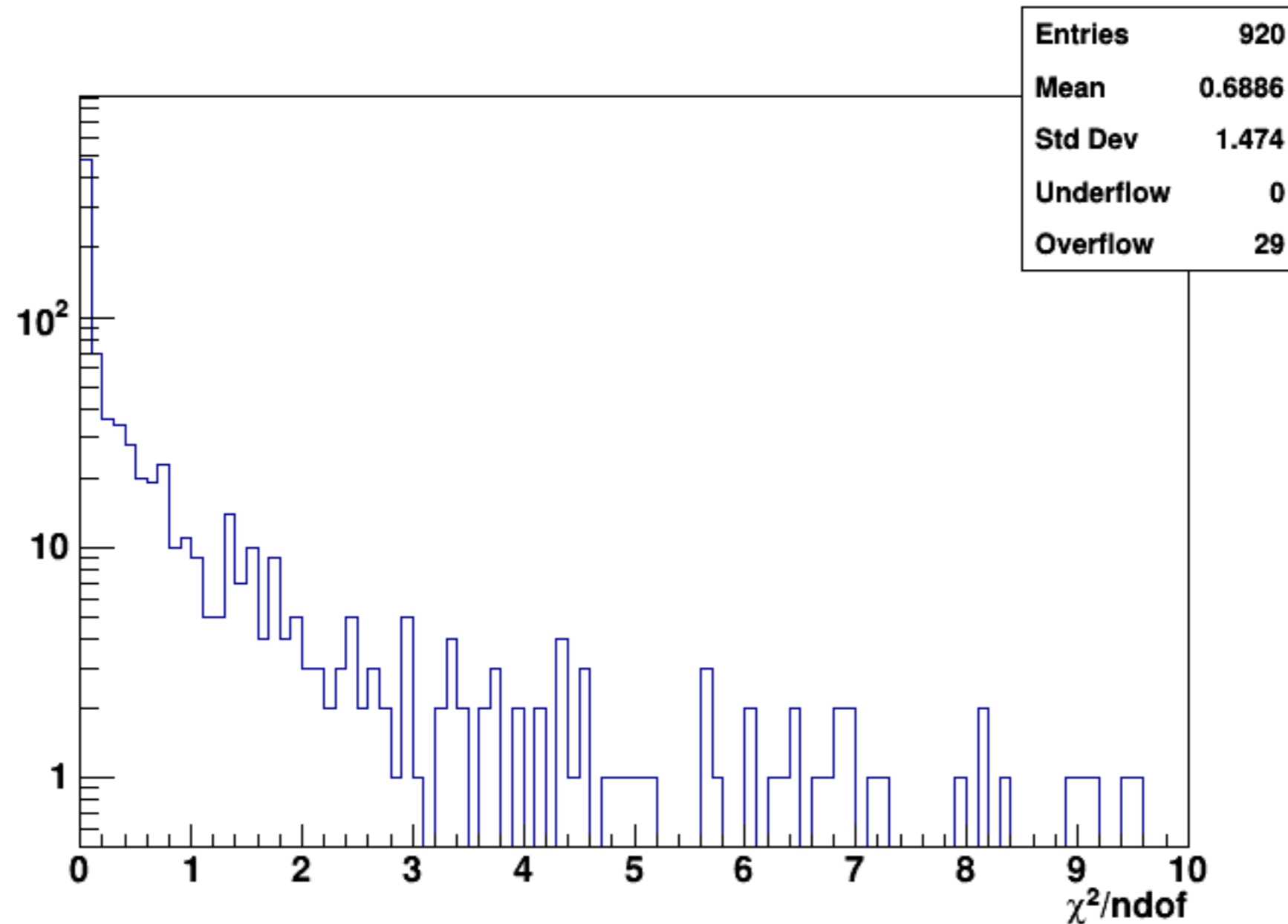
# The bitmask

- The bitmask is divided into 4 sections: **trajectory**, **track**, experiment, user
  - **HitIgnored**: Hit was not included for the computation of the trajectory
  - **NoPoint**: The trajectory point is not defined.
  - **Suspicious**: The point reconstruction is somehow questionable.
  - **Merged**: The hit might have contribution from particles other than this.
  - **DeltaRay**: The hit might have contribution from a delta ray.
  - **DetectorIssue**: The hit is associated to a problematic channel.
  - **Shared**: The hit is known to be associated also to another trajectory.
  - **ExcludedFromFit**: The point belongs to this track but it was not included in the fit.
  - **Rejected**: The hit is extraneous to this track.
  - **Reinterpreted**: The hit content has been re-elaborated by the fit.
- Experiment and user flags are to be defined by experiments and users

# Status

- Implementation of phase 1 nearly complete, with very little interface changes
- Only 3 packages affected so far (branch feature/cerati\_TrackTrajectory):
  - lardataobj: implementation of new classes
  - larreco: update Kalman fit to some new features (chi2), patch BezierTrack (deprecated)
  - uboonecode: need ALLOW\_DEPRECATED in CMakeLists.txt
- A few more packages were checked and gave no compilation errors:
  - lardata, lareventdisplay, larana, larpandora
- Will try to recompile the whole LArSoft when done
- The changes have been tested:
  - unit tests: helped the development and are successful
  - microboone reconstruction chain: smooth execution, sensible results
  - I/O rules tested reading and refitting tracks from previous LArSoft releases
- More changes are along the way for the Kalman fit:
  - residual object, SMatrix migration, usage of flags in TrackTrajectory
  - there is a chance these will be ready for this week, at least in a preliminary form
  - will be a separate branch anyway

# Testing the track $\chi^2$



- Input reco file from previous LArSoft release, single muon events
- Fitted tracks have chi2 properly filled
- Deeper study of fit inputs (eg hit errors) needed to understand plot details

# Conclusion

- Stage 1 of new recob::Track implementation should be ready for this week LArSoft release.
- Happy to discuss and include further modification, the discussion is open