

---

# PHYSICS-OBJECT STANDARDIZATION

JUNE 26 2018



# INTRODUCTION

- Goal: implement standard selections for physics-objects, prior to analysis cuts
  - Make the starting point for analyses easier to understand
  - Reduce possibilities for bugs in individual analysis codes
  - Improve computational efficiency, reduce file sizes (maybe)
- Need input from Bump-Hunters and Vertexers
- Need input from Tracking and Ecal
  
- To be discussed at software & analysis weekly meetings
- Slack channel #standardization

# WHAT OBJECTS TO STANDARDIZE?

- Useful tracks
- Valid reco particles
  - Electrons, positrons, photons
- Valid V0s
  - Unconstrained, target-constrained, beamspot-constrained?
- Multiple levels of selection? (e.g. LooseElectron and TightElectron?)

# WHERE TO DO THE SELECTING?

Options include:

- MergeTrackCollections driver, in evio-to-lcio reco sequence
  - Eliminate some tracks (e.g. using AmbiguityResolver) before building reco particles
  - Save computation time throughout the rest of reco, smaller output lcio
  - Discarded tracks unrecoverable
  - Rerunning reco on existing files may be impractical
- ReconParticleDriver, in evio-to-lcio reco sequence
  - Where track-cluster matching is currently performed, add more selections
  - Save some computation time throughout vertexing etc, smaller output lcio
  - Components (tracks, clusters, etc) of discarded particles still recoverable from lcio
  - Rerunning reco on existing files may be impractical

# WHERE TO DO THE SELECTING?

- Vertexing, on Lcio
  - Smaller output Lcio
  - Components of discarded vertices (reco particles etc) still recoverable from Lcio
  - Rerunning vertexing on existing files doesn't require rerunning all of reco
- Skimming drivers, to create skimmed Lcios that everyone would use.
  - Time-consuming to perform the skims
  - Extra space required to store skims, if we retain original Lcios
- dst-maker / tuple-maker
  - Smaller output dsts / tuples
  - Discarded stuff still recoverable from Lcio
  - Ensure same selections made by both makers (moot if we start making tuples from dsts)