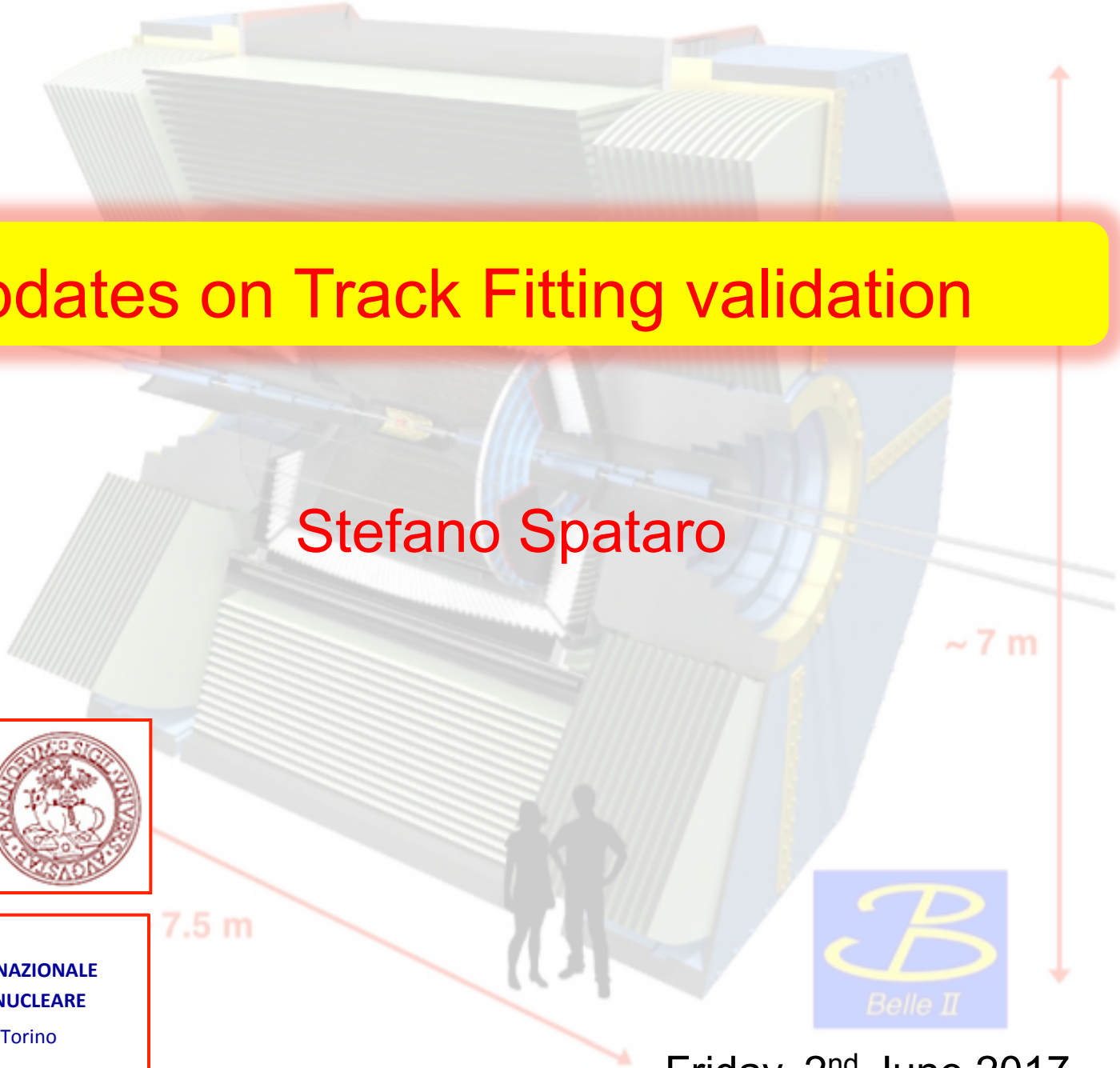
Updates on Track Fitting validation

Stefano Spataro

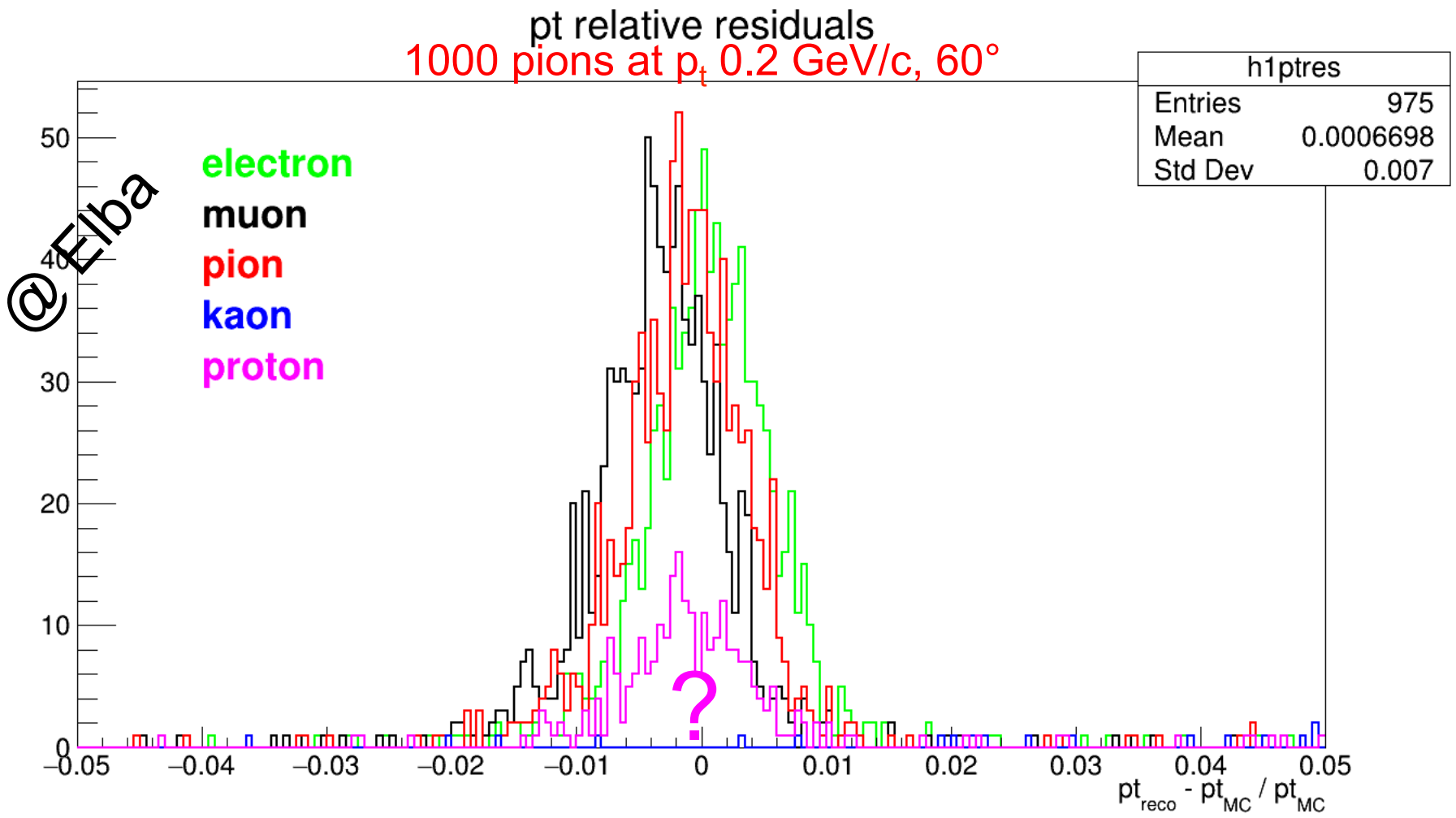
UNIVERSITÀ DEGLI STUDI DI TORINO
ALMA UNIVERSITAS TAURINENSIS

ISTITUTO NAZIONALE DI FISICA NUCLEARE
Sezione di Torino



Friday, 2nd June 2017

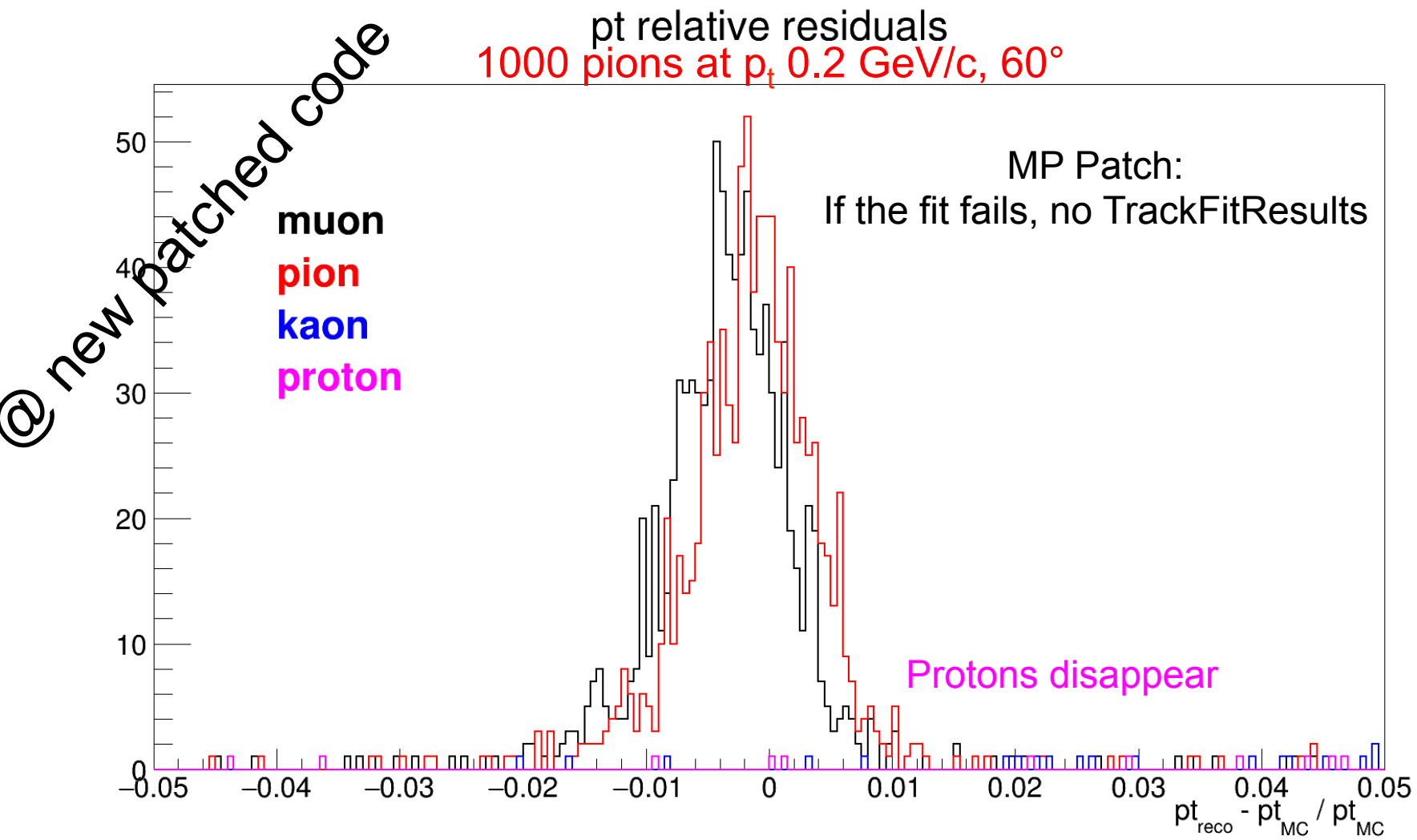
Open Problem #1



Q: Why proton residuals are peaked at the pion position?

A: If the fit fails, TrackFitResults returns the default pdg (pion) parameters and not the chosen one

Open Problem #1 - Understood

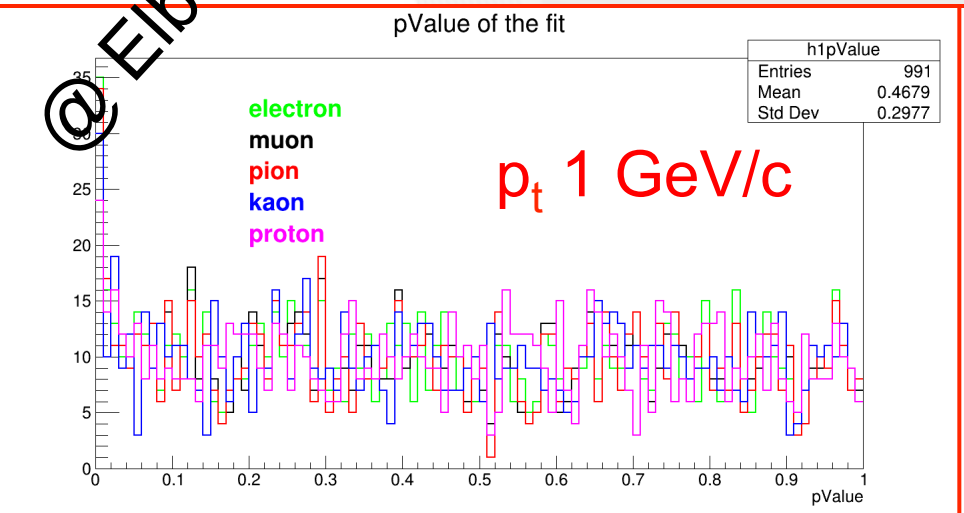


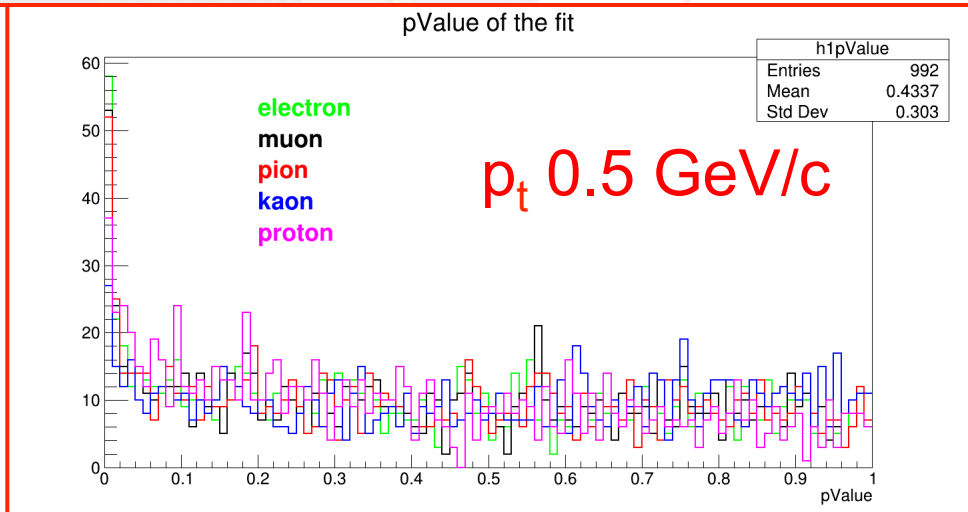
Just to confirm with plots what was said during the workshop

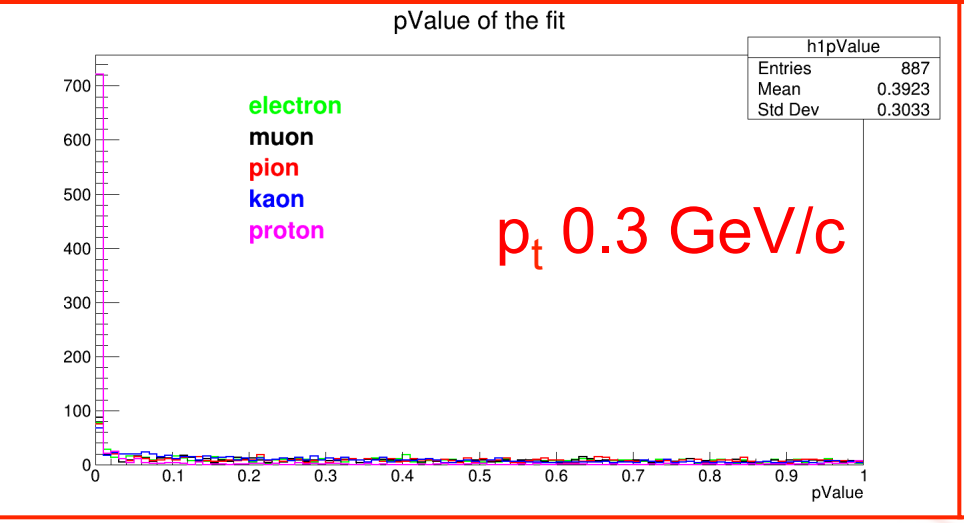
Open Problem #2

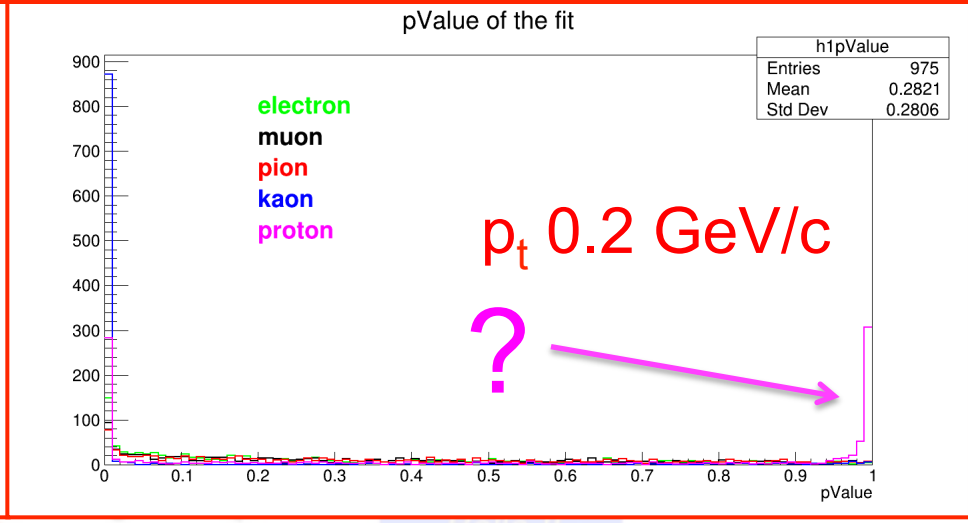
1000 pions at 60° - pValue

@Elba





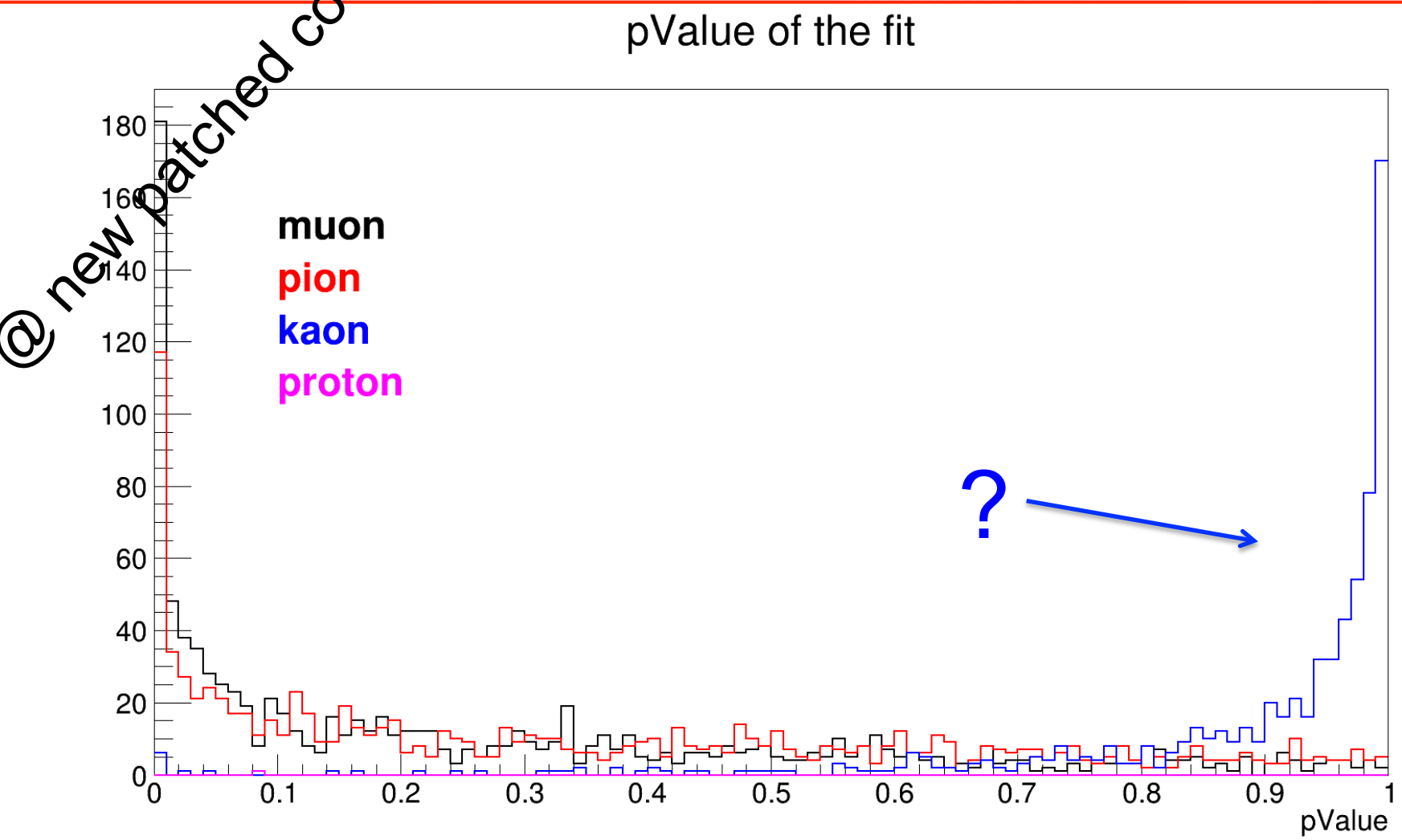




Open Problem #2

1000 pions at p_t 0.1 GeV/c, 60°

@ new patched code



What is pValue?

Genfit stores inside FitStatus object:

- χ^2
- Number of degrees of freedom (NDF)

pValue is calculated from χ^2 and NDF

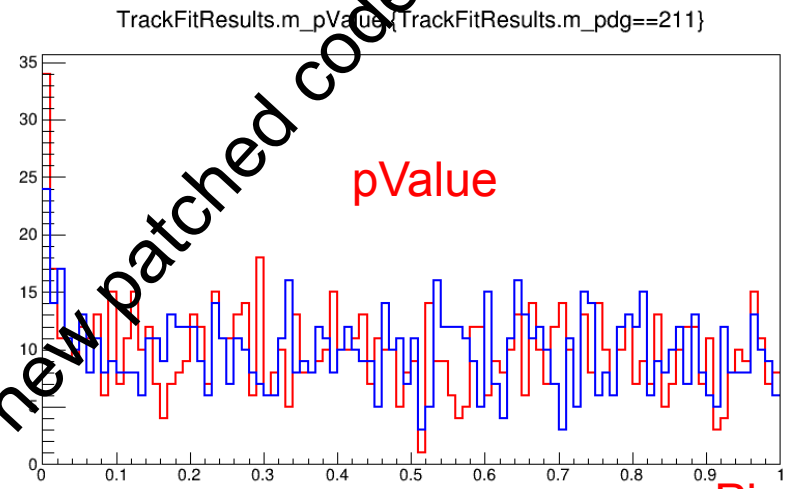
```
double getChi2() const {return chi2_;}
 //! Get the degrees of freedom of the fit.
 double getNdf() const {return ndf_;}
 /**
  * @brief Get the p value of the fit.
  *
  * Virtual, because the fitter may use a different probability distribution.
  */
 virtual double getPVal() const {return std::max(0.,ROOT::Math::chisquared_cdf_c(chi2_, ndf_));}
```

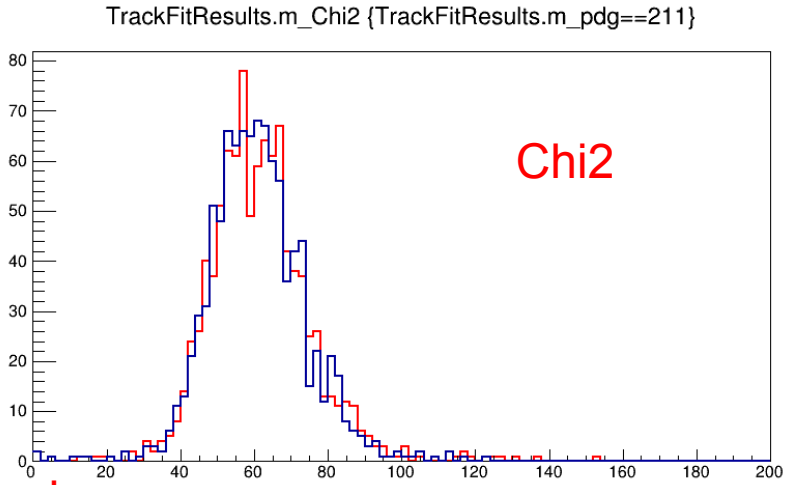
In TrackFitResults we store ONLY pValue (TrackBuilder.cc)

I have added few new data members to TrackFitResults as a crosscheck

1000 pions at p_t 1 GeV/c, 60°

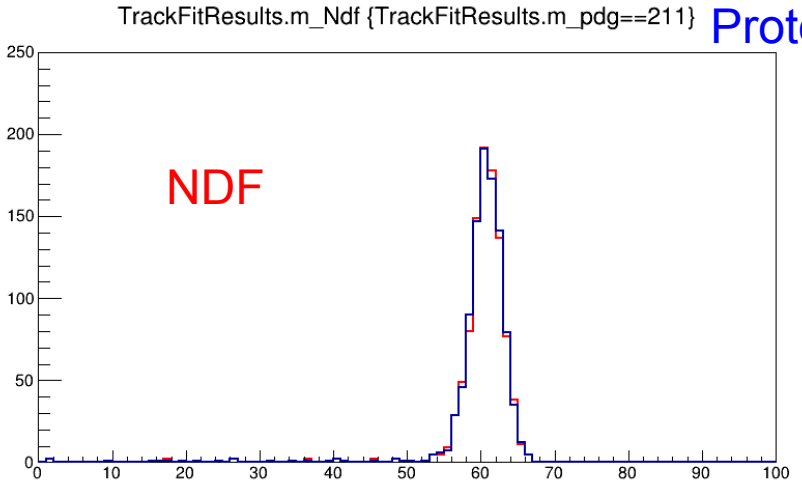
@ new patched code

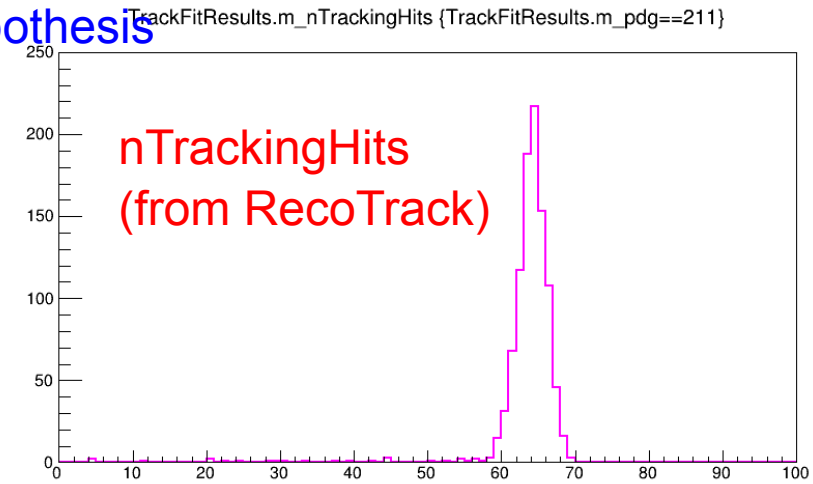




Pion hypothesis

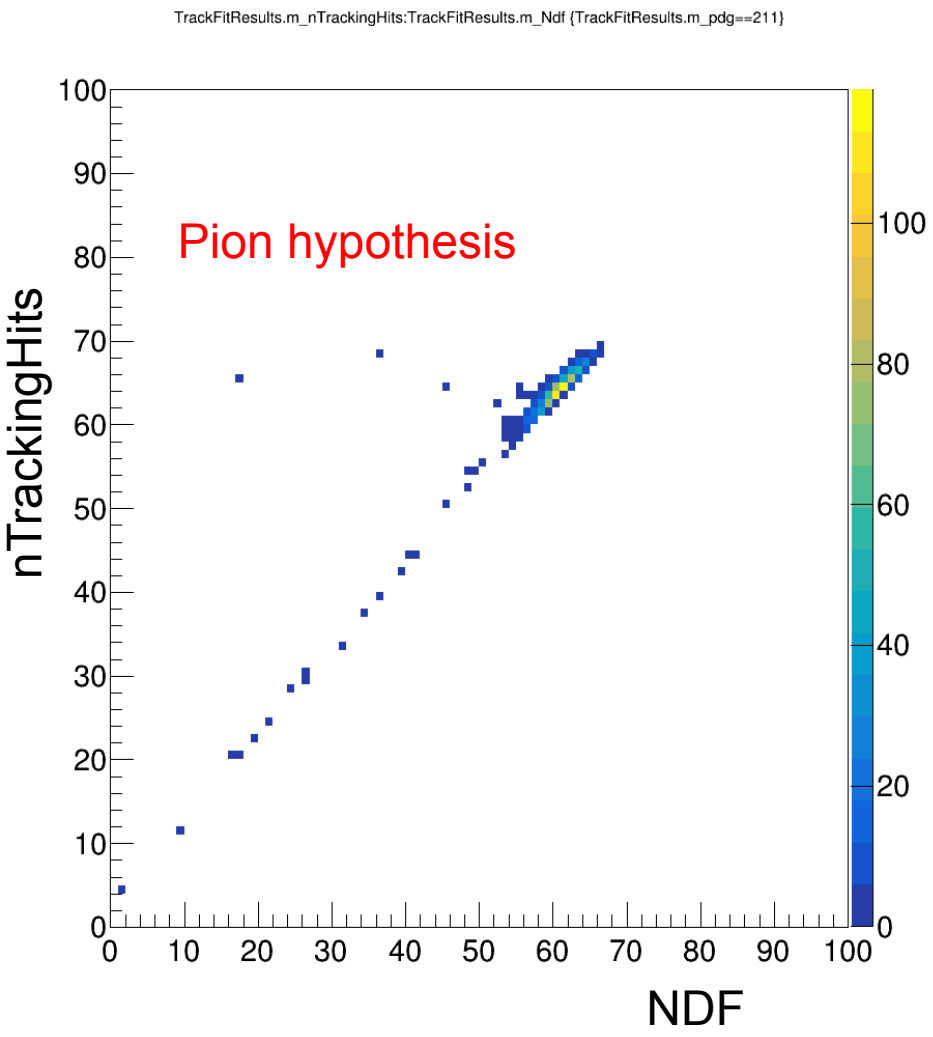
Proton hypothesis

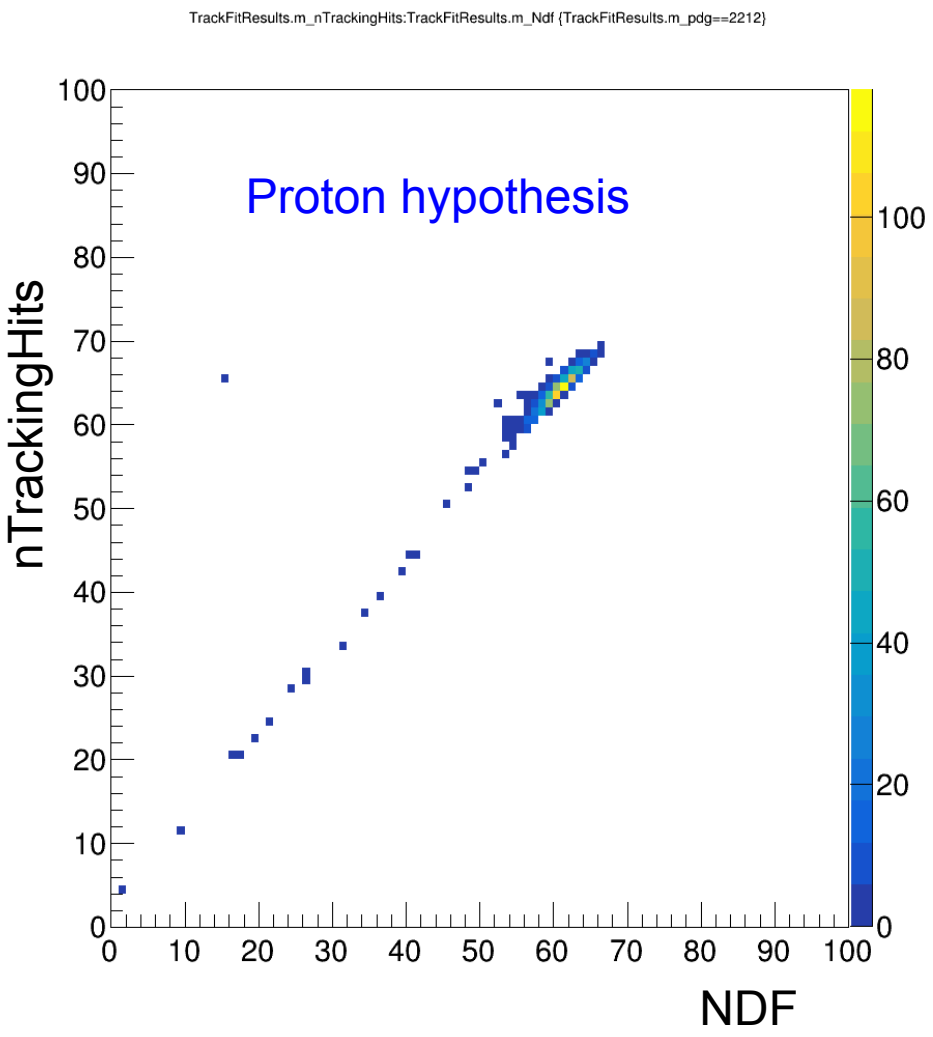




OK, Everything Understood

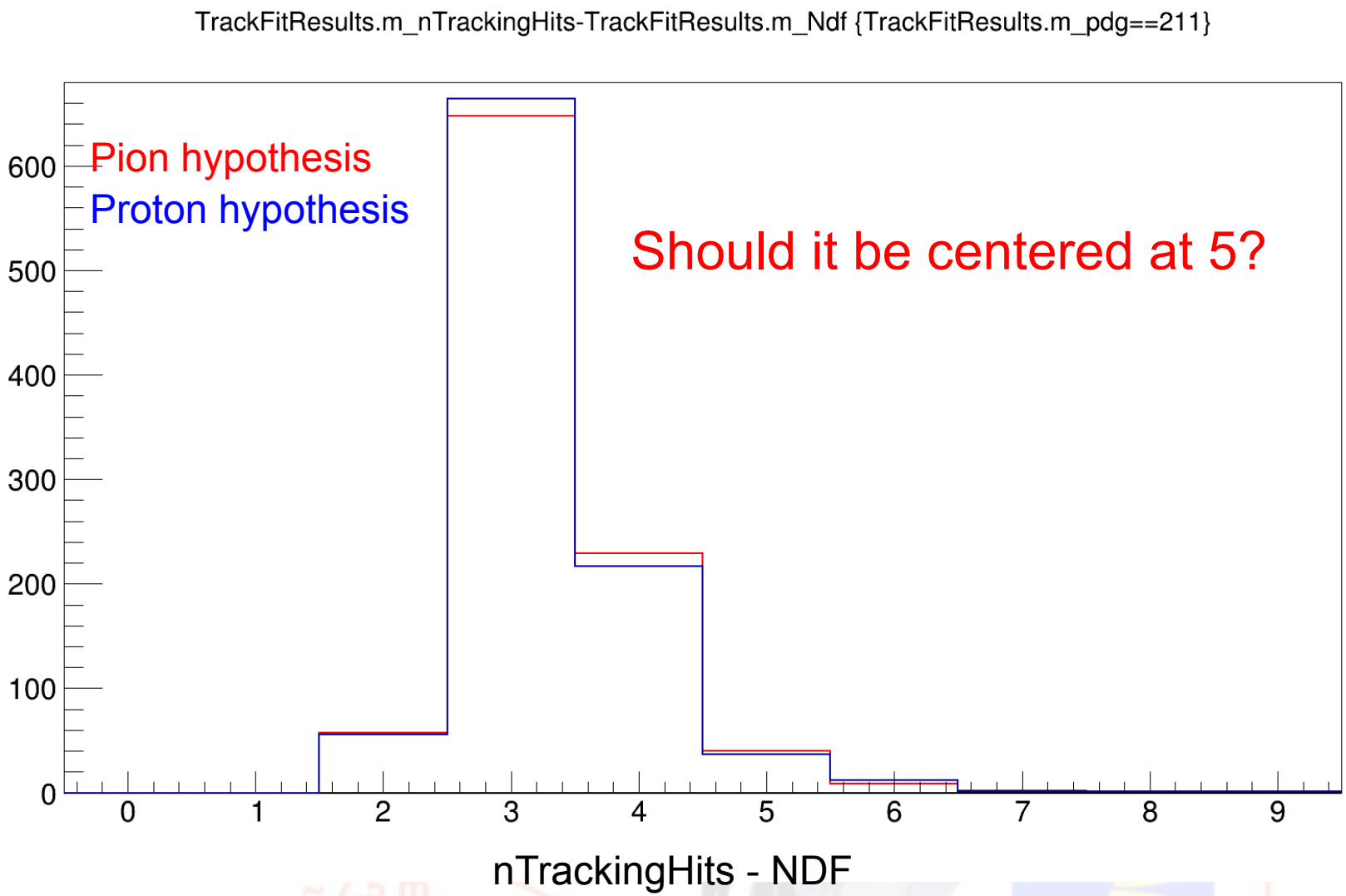
1000 pions at p_t 1 GeV/c, 60°





OK, Everything Understood

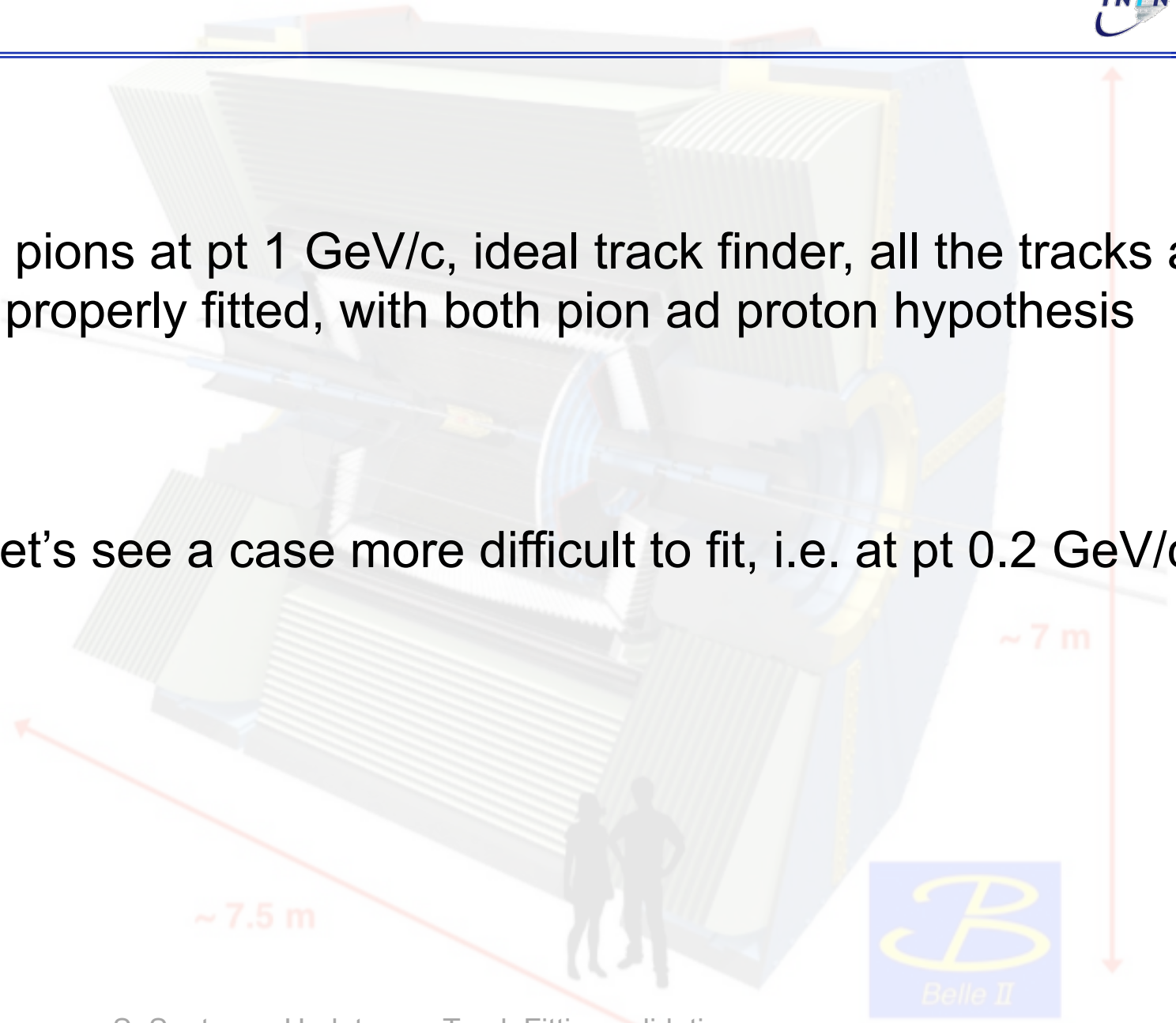
1000 pions at p_t 1 GeV/c, 60°



Almost Understood

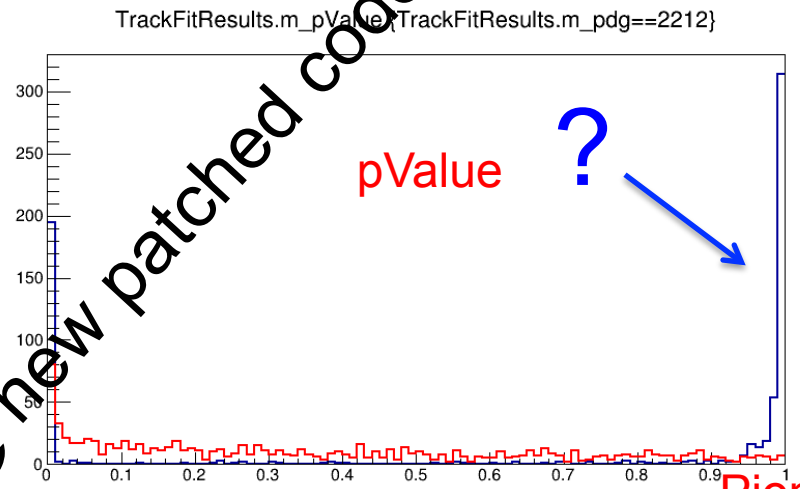
With pions at pt 1 GeV/c, ideal track finder, all the tracks are properly fitted, with both pion ad proton hypothesis

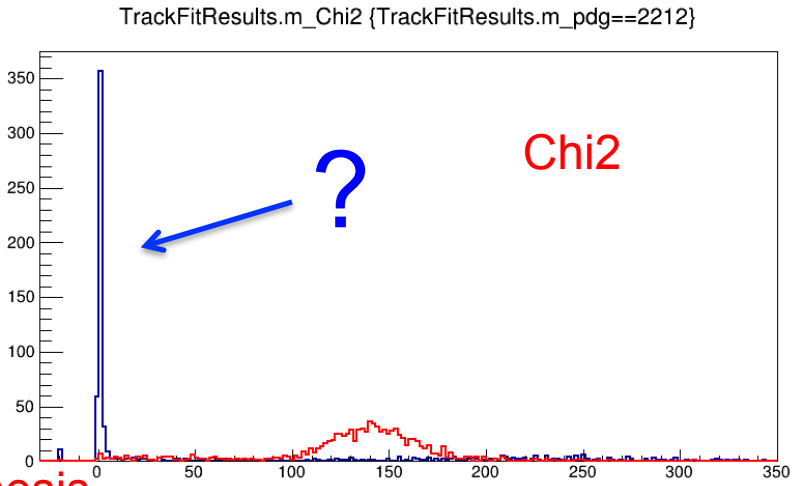
Let's see a case more difficult to fit, i.e. at pt 0.2 GeV/c



1000 pions at p_t 0.2 GeV/c, 60°

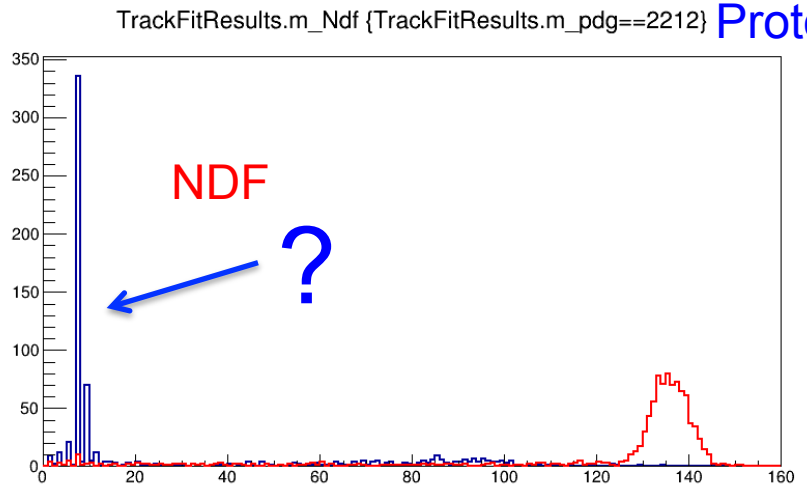
@ new patched code

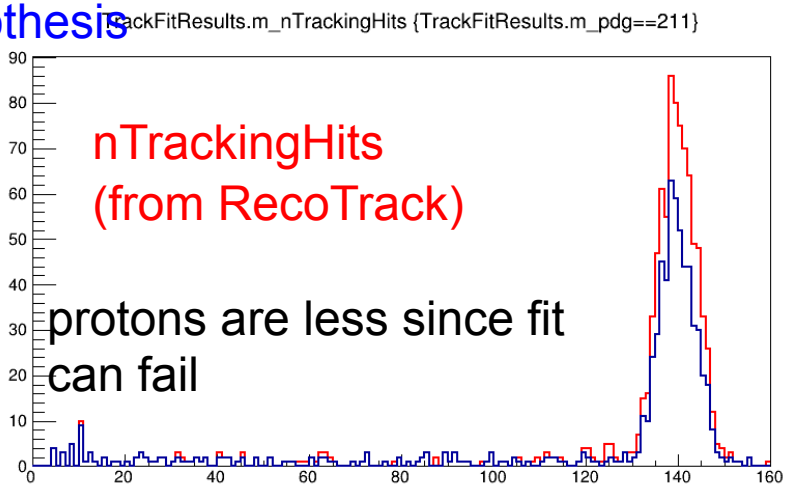




Pion hypothesis

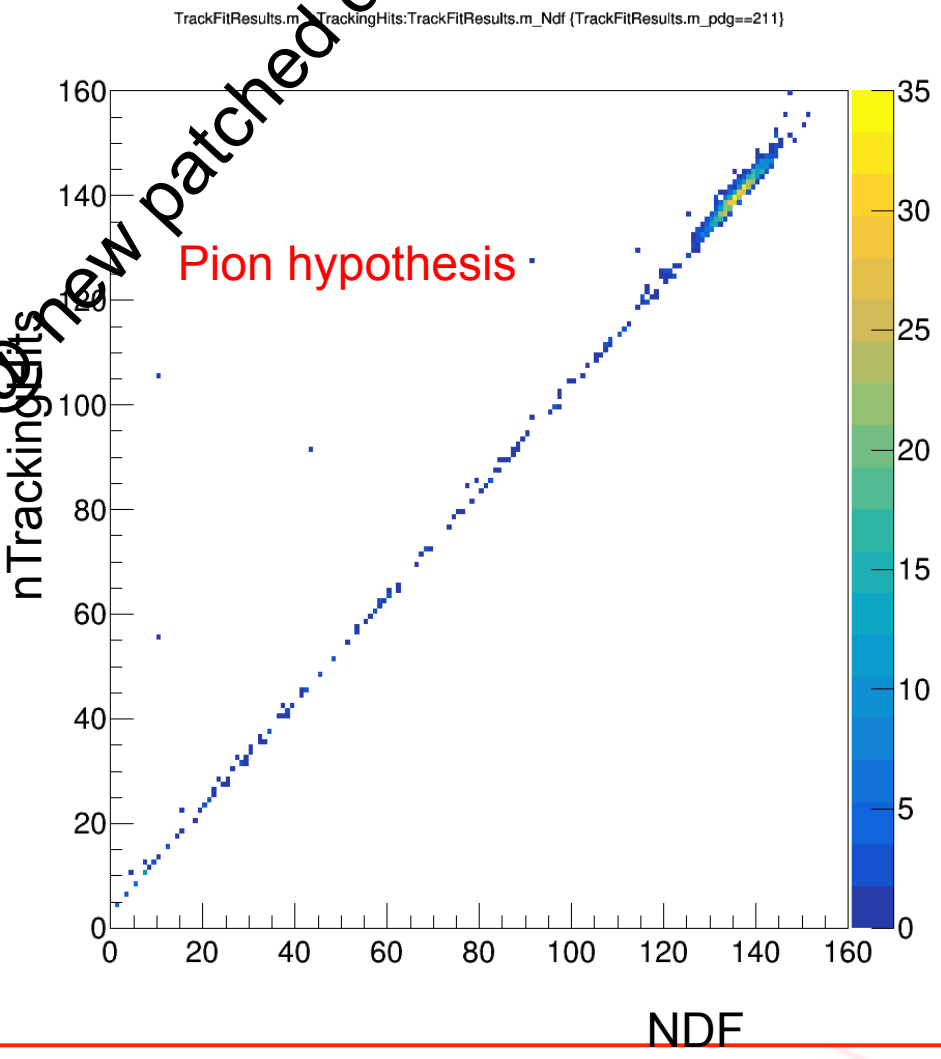
Proton hypothesis

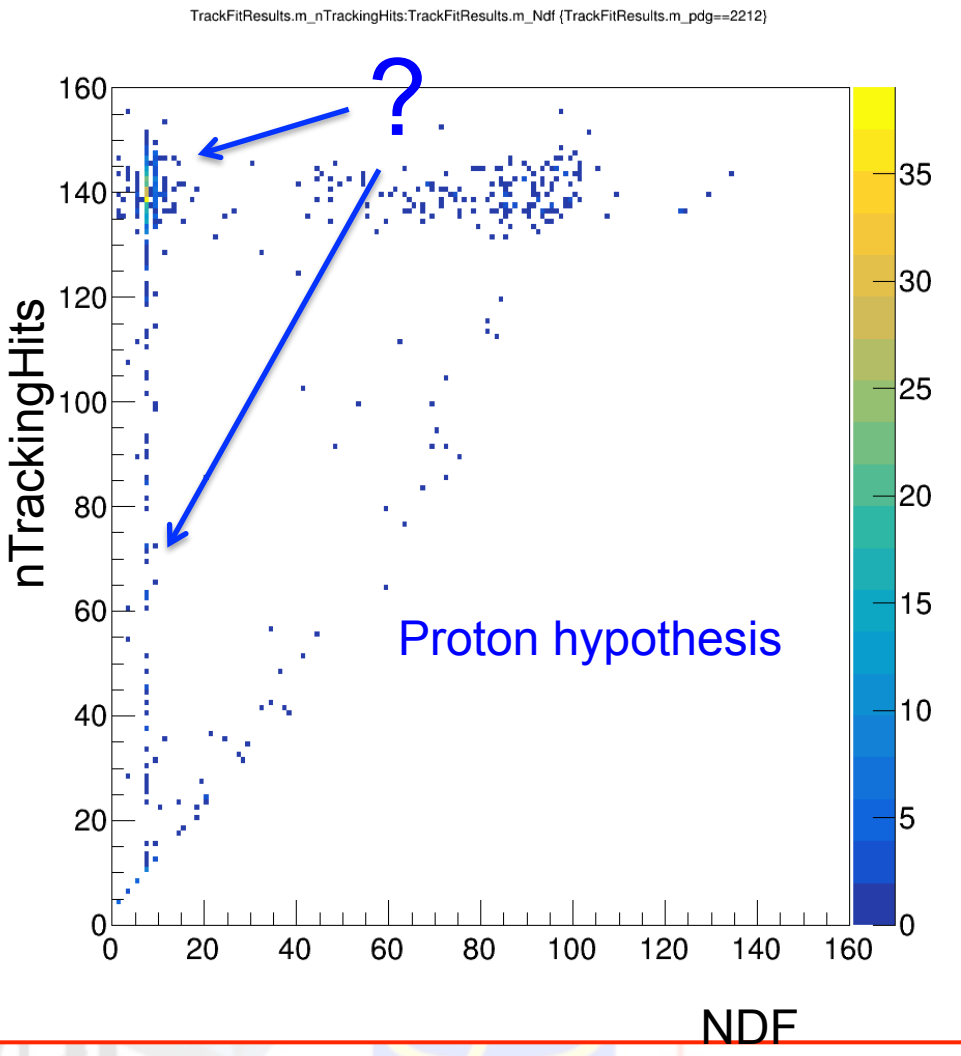




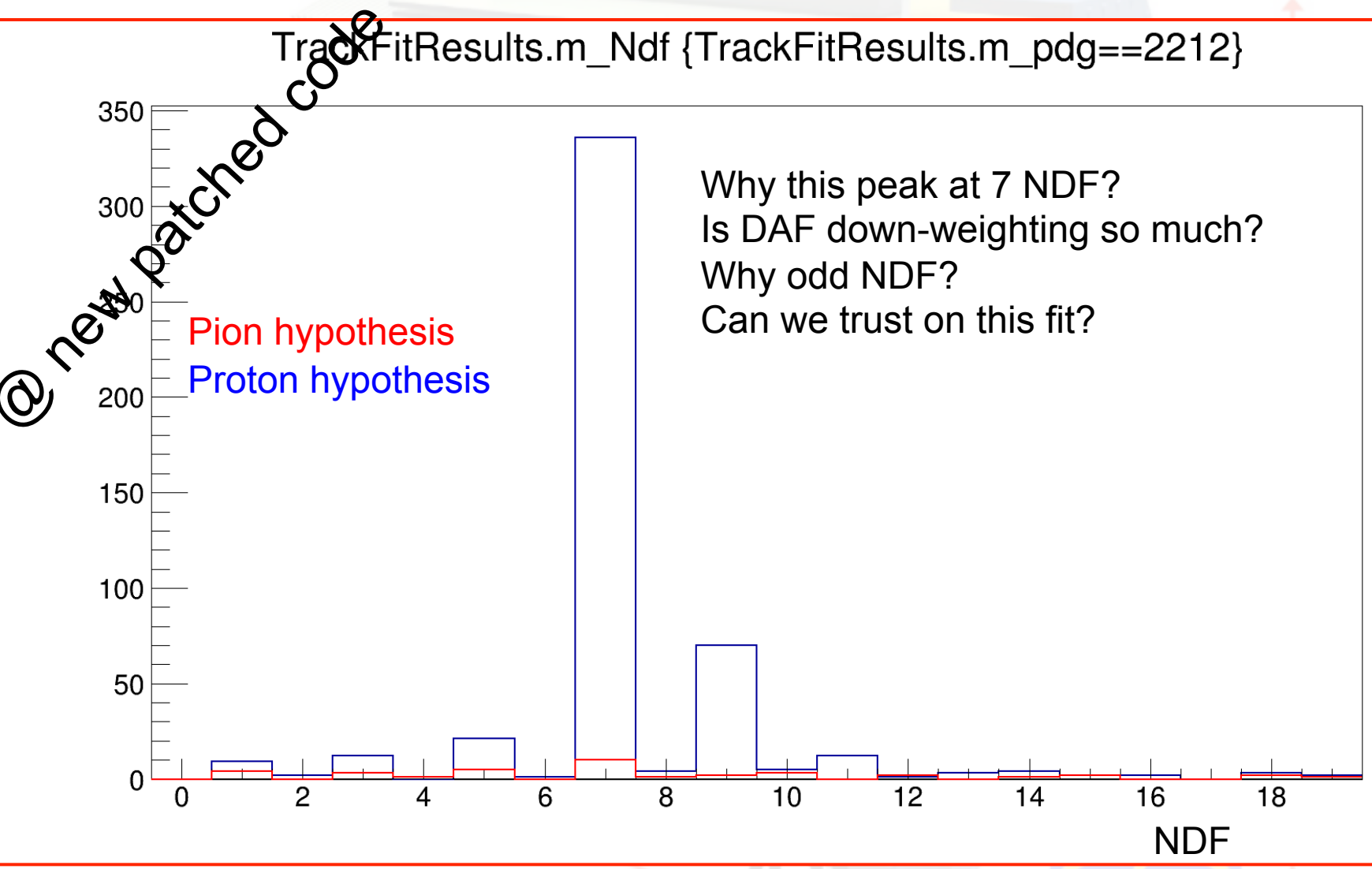
1000 pions at p_t 0.2 GeV/c, 60°

new patched code



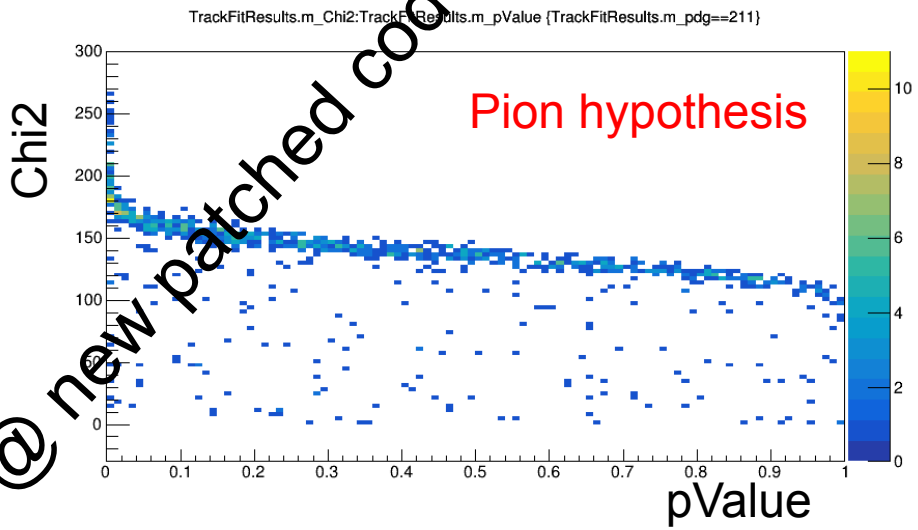


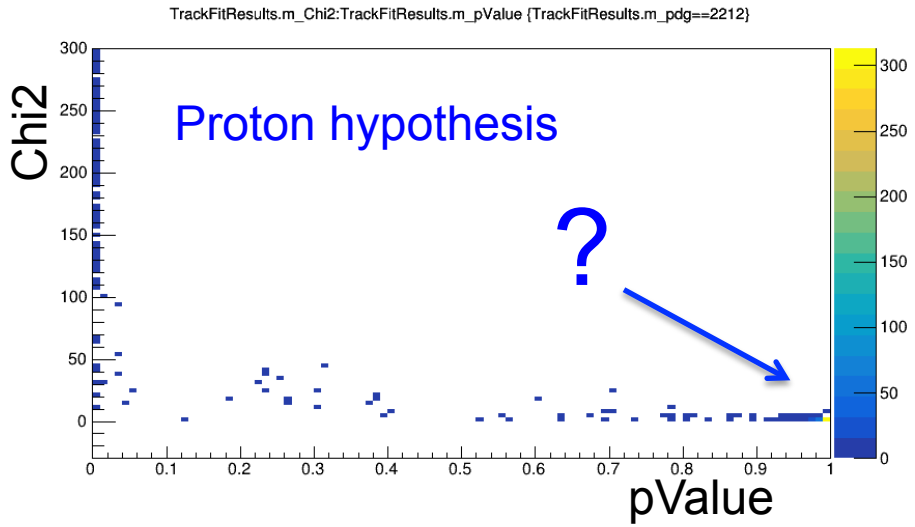
1000 pions at p_t 0.2 GeV/c, 60°

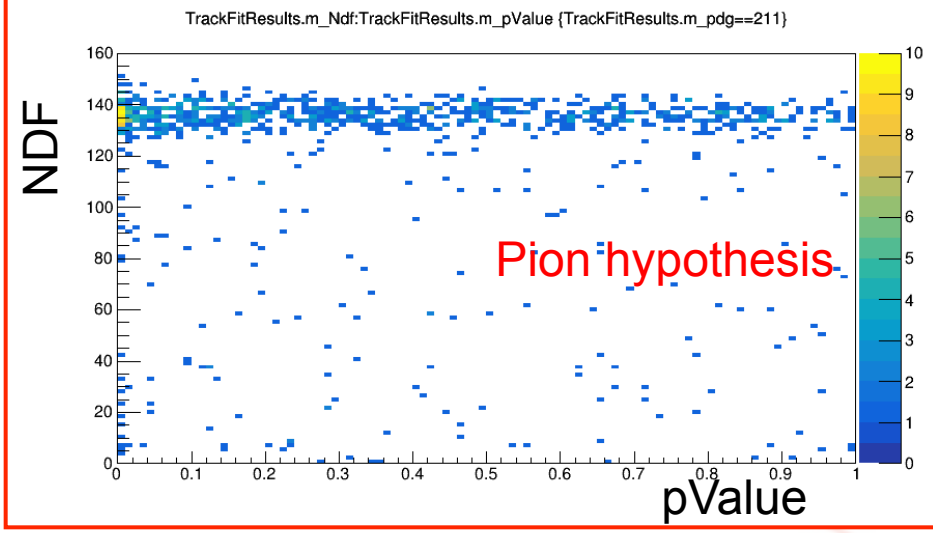


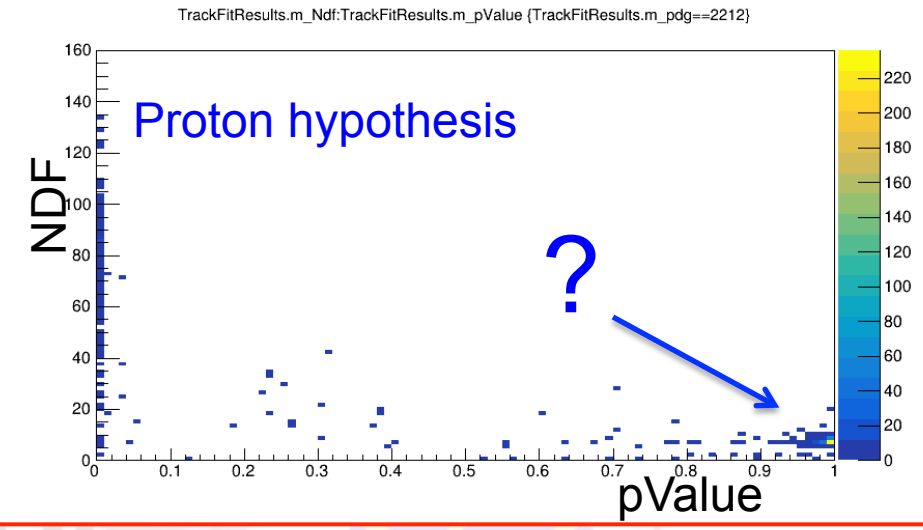
1000 pions at p_t 0.2 GeV/c, 60°

@ new patched code

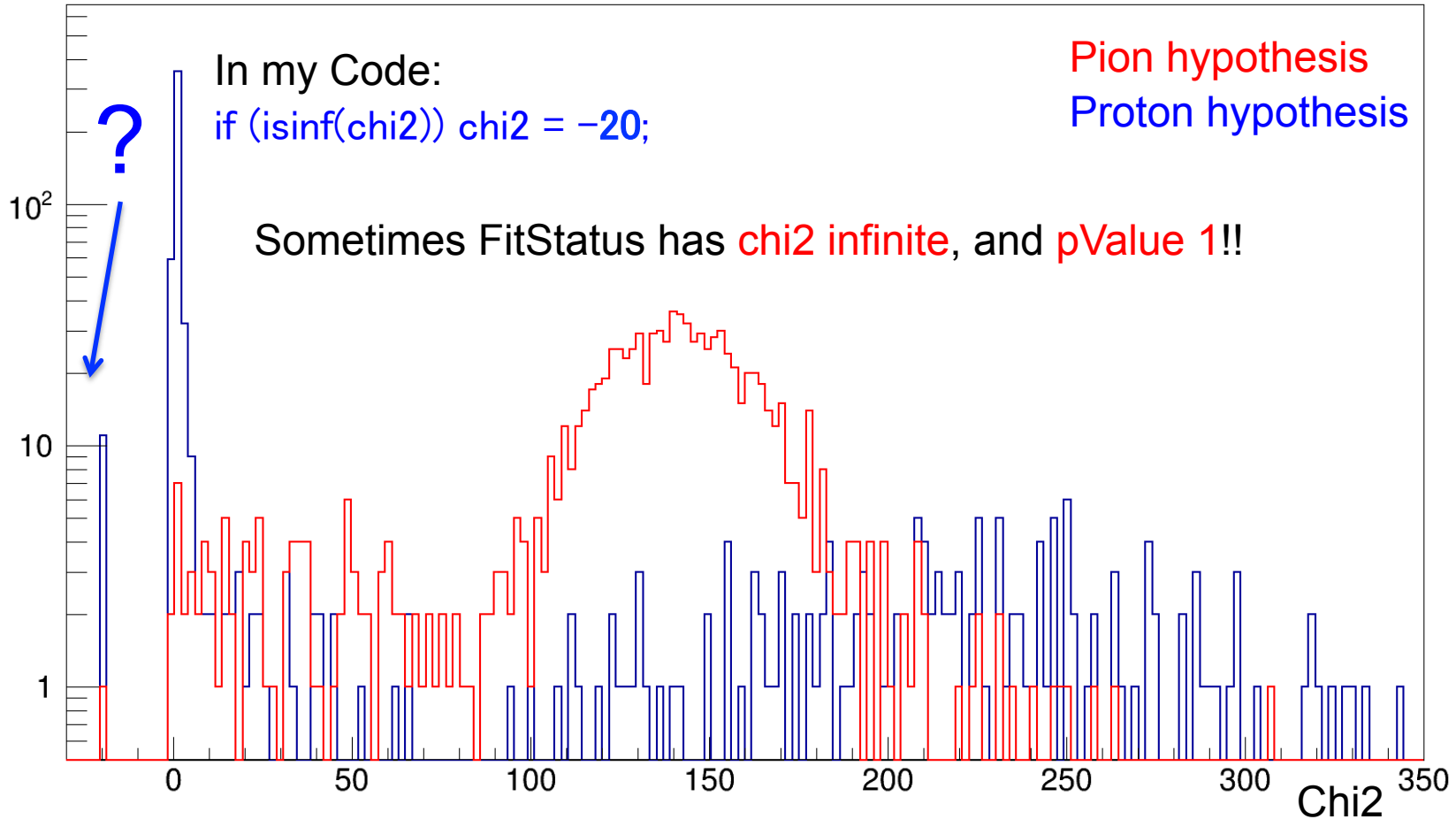








1000 pions at p_t 0.2 GeV/c, 60°



Trying to summarize

Sometimes FitStatus returns a “infinite” chi2, which is wrongly associated to pValue 1. This could be easily caught by TrackBuilder.

Very bad fits (i.e. low pt with wrong particle hypothesis) sometimes return a very low number of degrees-of-freedom, I believe not physical, which are associated to pValue = 1. Why NDF 7? Why odd NDF? Maybe one could check nTrackingHits/NDF to estimate the average weight and cut?

It is not clear to me why nTrackingHits-NDF for good fits is peaked at 3.

Maybe storing only pValue in TrackFitResults is not safe enough to catch fit artifacts.

What's next? (short term)

Check real pattern recognition, where tracks can have spurious hits and low pt track candidates could be shorter, without too many curls.

Take a look into eventdisplay, to understand how the bad fits look like.

Fast check of fitting with deuterons.

Any hints?