

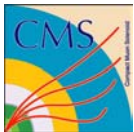


Tree Based Validation Tool for Track Reconstruction



- **Current validation tool and their limitations**
- **Improvements to Track Validation tools**
 - Implementation based on a ROOT tree
 - Comparison with previous track validator
 - Flexibility and time saving
- **Further developments**

R. Kelley, B. Mangano, and V. Sharma
University of California, San Diego



Current Validation Tool



The current default validation tool (MultiTrackValidator) has been developed for releases validation, and it has been around for a while.

Nevertheless it has shown limitations for actual trouble-shooting of bugs/errors in the tracking code and interactive analysis of the quality of reconstructed tracks.

This is mainly due to the fact that all the validation histograms are booked and filled by the same CMSSW job.

Drawback:

Every time one wants to redefine the binning of a histogram or apply a different set of cuts to the track analyzed, the job has to be **rerun** (sometimes **recompiled**). This is obviously time consuming:

- The Sim-Reco track association process is re-executed even if the matching is completely unrelated with respect to the histogram binning and it should be **done only once**.
- There is a large overhead due to CMSSW initialization.



Tree Based Implementation



1) A CMSSW analyzer reads reco tracks and tracking particles from the Event and performs the association based on the standard associators (byHits, byChi2) or a custom defined associator.

2) For each reco track, and corresponding simtrack, a set of variables are saved as leaves in a ROOT Tree:

- reco track: p_T , ϕ , η , P_T error, $\chi^2/d.o.f.$, #hits, ...
- (associated) tracking particle p_T , ϕ , η , ...
- association failing/success boolean

TObject	r_phi	r_etaError	s_pt_byHits	s_mass_byChi2	s_d0_byChi2	s_lambda_byCustom
charge	r_eta	r_phiError	s_qoverp_byHits	s_charge_byChi2	s_dz_byChi2	s_phi_byCustom
numValidHits	r_d0	r_d0Error	s_lambda_byHits	s_pdgId_byChi2	hasTPAssoc_byCustom	s_eta_byCustom
numPixelHits	r_dz	r_dzError	s_phi_byHits	s_pt_byChi2	s_mass_byCustom	s_d0_byCustom
nChi2	r_ptError	hasTPAssoc_byHits	s_eta_byHits	s_qoverp_byChi2	s_charge_byCustom	s_dz_byCustom
r_pt	r_qoverpError	s_mass_byHits	s_d0_byHits	s_lambda_byChi2	s_pdgId_byCustom	
r_qoverp	r_lambdaError	s_charge_byHits	s_dz_byHits	s_phi_byChi2	s_pt_byCustom	
r_lambda	r_thetaError	s_pdgId_byHits	hasTPAssoc_byChi2	s_eta_byChi2	s_qoverp_byCustom	

3) A set of macros are written to read the tree and produce the final histograms on the fly.



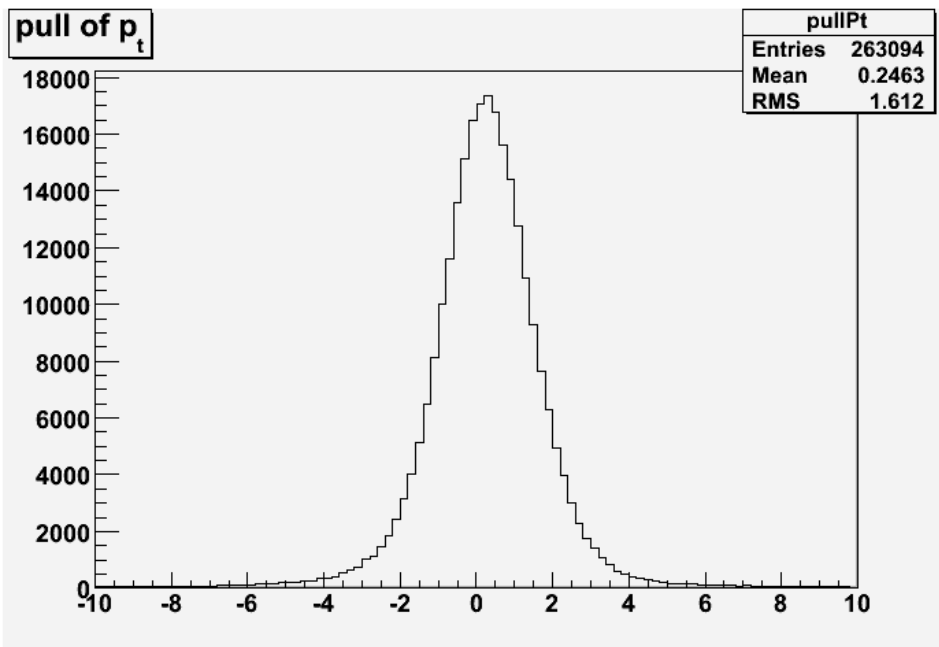
Benefits



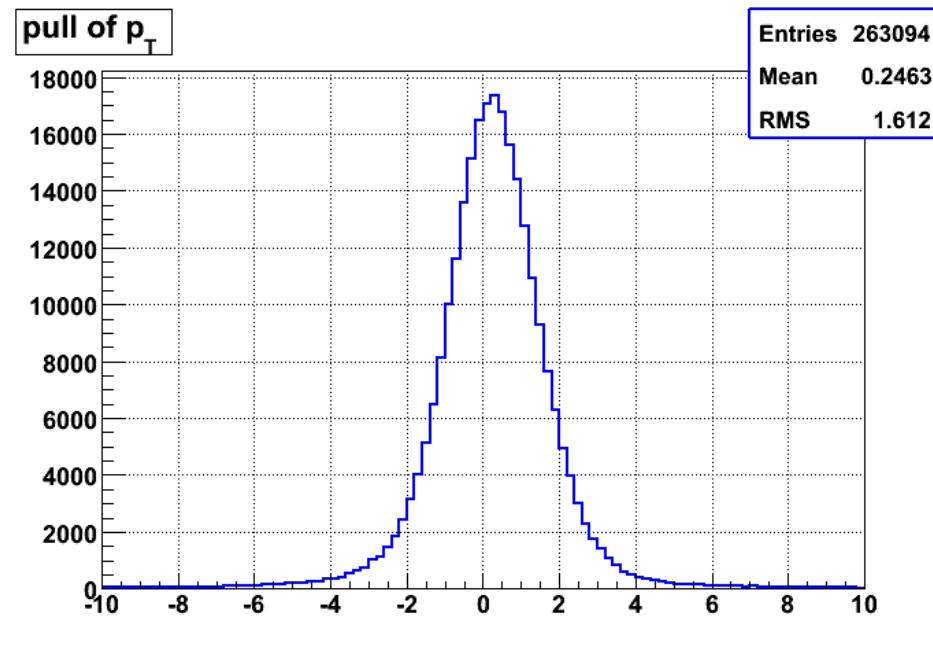
- 1) **The time consuming part of the analyzer (reco-sim track association) is performed only once.**
- 2) **Since all the histograms, including efficiency and fake rate plots, are produced at a later stage (after the job is run), the CMSSW analyzer process can be safely split in several parallel jobs**
 - **Only a fraction of the events for a sample are analyzed by each job**
 - **Then the ROOT trees from the output of the various jobs are merged together**
 - **Finally we can make any plot using the ROOT macros on the merged file.**

Pull Distributions of p_T

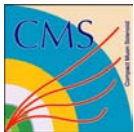
- CMSSW 1_6_0_pre6
- Ran over ~5k events from TTbar release validation sample
(/RelValTTbar/CMSSW_1_6_0-RelVal-1189470628/GEN-SIM-DIGI-RECO)



MultiTrackValidator



TreeBasedTrackValidator

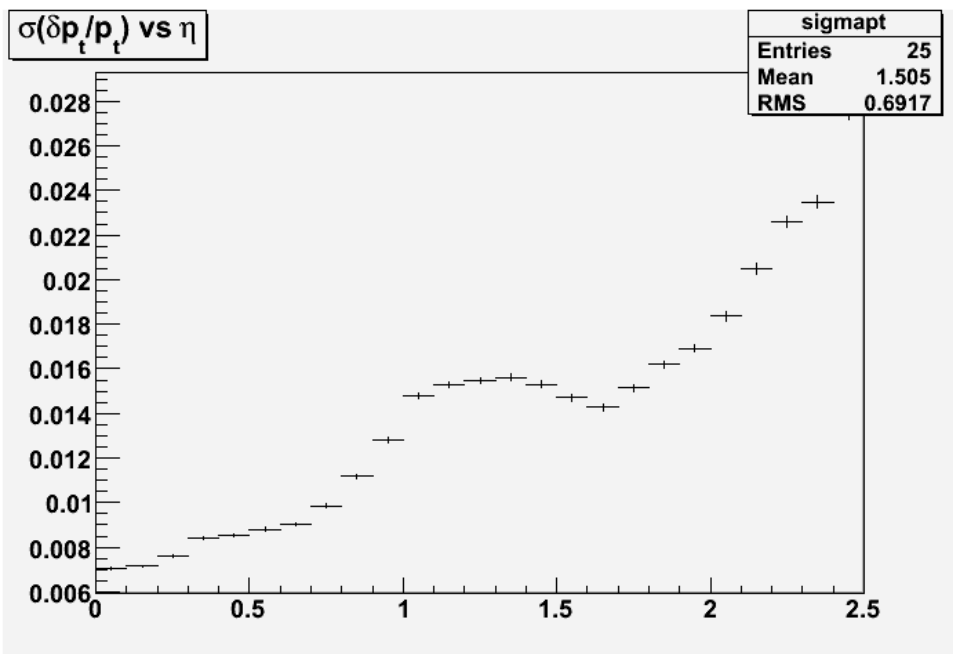


Validation of Validation Plots

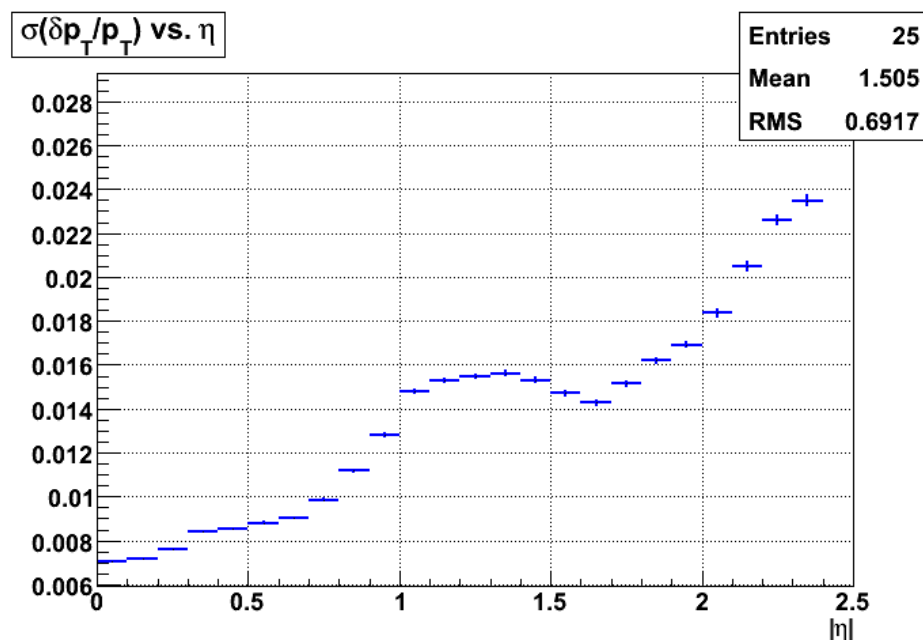


Resolution of p_T vs. η

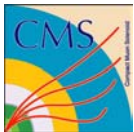
- CMSSW 1_6_0_pre6
- Ran over ~5k events from TTbar release validation sample
(/RelValTTbar/CMSSW_1_6_0-RelVal-1189470628/GEN-SIM-DIGI-RECO)



MultiTrackValidator



TreeBasedTrackValidator

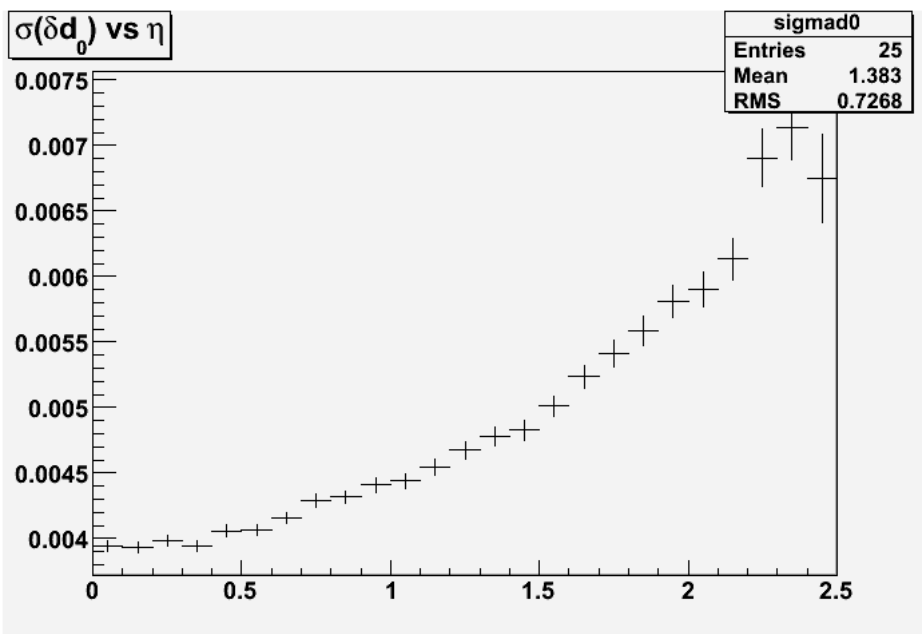


Validation of Validation Plots

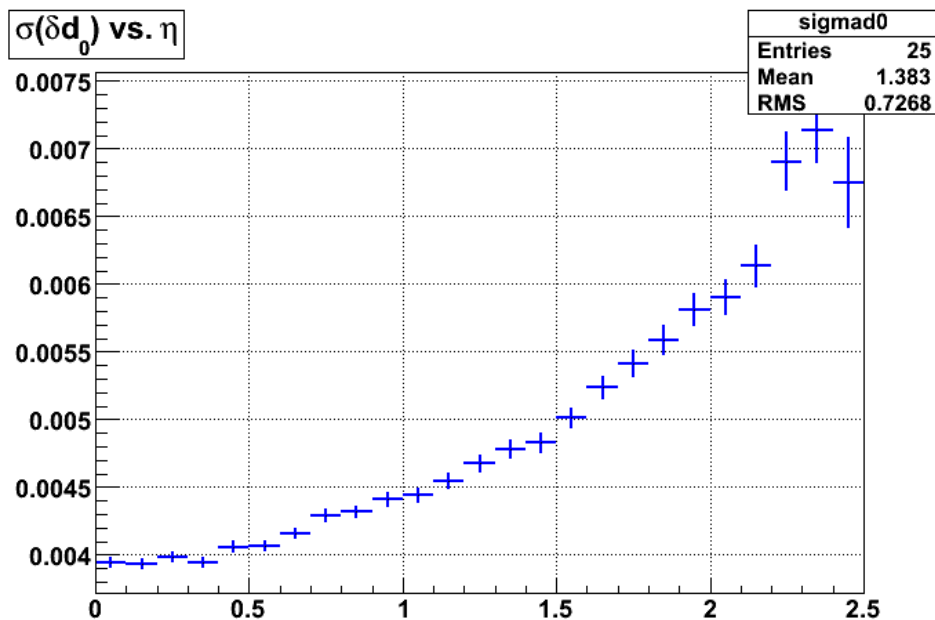


Resolution of d_0 vs. η

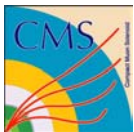
- CMSSW 1_6_0_pre6
- Ran over ~5k events from TTbar release validation sample
(/RelValTTbar/CMSSW_1_6_0-RelVal-1189470628/GEN-SIM-DIGI-RECO)



MultiTrackValidator



TreeBasedTrackValidator



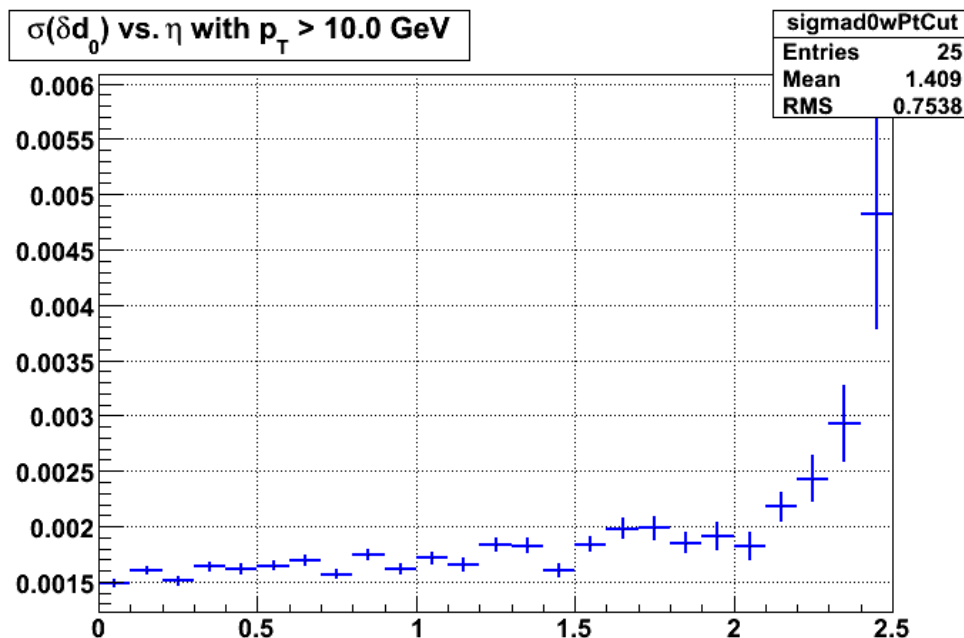
Want to Make a Cut?



Resolution of d_0 vs. η

- Let's remake the previous plot with a $p_T > 10.0$ GeV cut.

One would still be waiting for the output of the job...

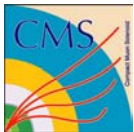


MultiTrackValidator

- ~ 2 minutes to load the magnetic field and geometry information
- ~ 1 hour to re-run the job (on 5k events)

TreeBasedTrackValidator

- ~ 10 seconds to edit macro and plot



Want to Make to Change the Binning?



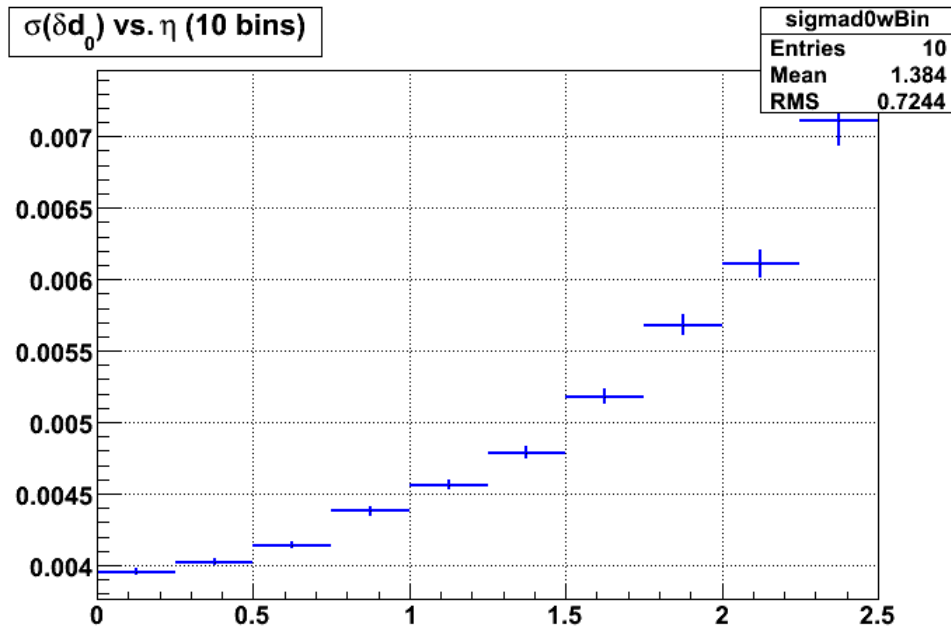
Resolution of d_0 vs. η

- Let's say now you want to change the binning to 10 instead of 25 and without the cut.

One would still be waiting for the output of the job...

MultiTrackValidator

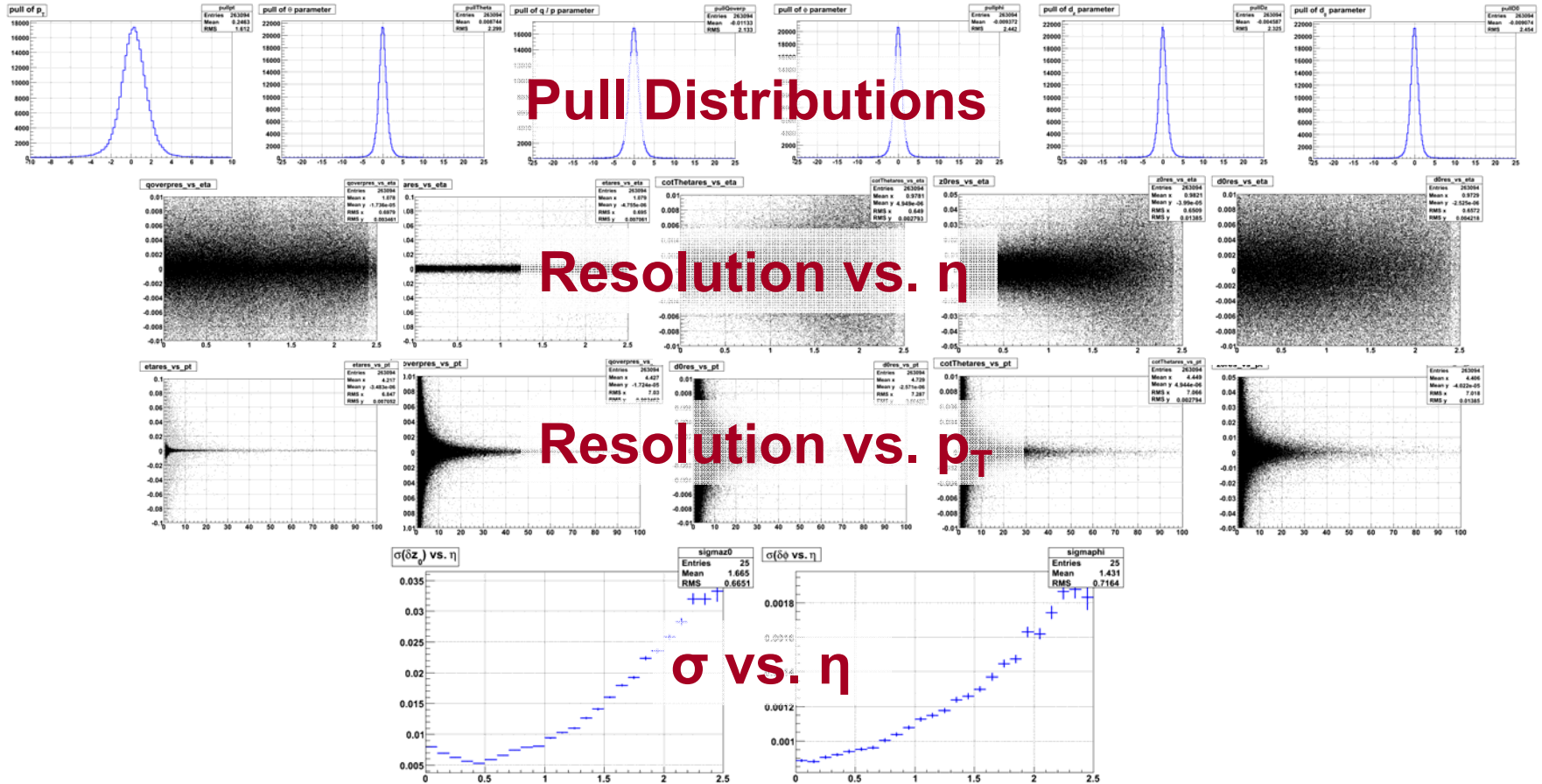
~ another hour to re-run the job



TreeBasedTrackValidator

~ 10 seconds to edit macro and plot

Other plots completed:



Remaining Plots: σ vs. p_T , #hits vs. η/p_T , σ of pulls vs. η , fakes, efficiency, etc...



Conclusions



- Existing validation code has been re-organized in order to exploit the ROOT tree features.
- The new implementation provides more flexibility, is faster and speeds up isolating problems.

Plans:

- Most of the validation plots are already available in the ROOT tree based validator. The remaining plots and additional ones will be added.
- Commit the code in Validation/RecoTrack package and perform tracking validation for the tracking group.