IN THE NAME OF GOD

HLT Track Selection in CMS

Mohammad Hassan Hassanshahi

Sharif University of Technology(SUT), Tehran, Iran

E-mail:mhhassanshahi@yahoo.com

Supervisors: Simone Gennai, Mia Tosi

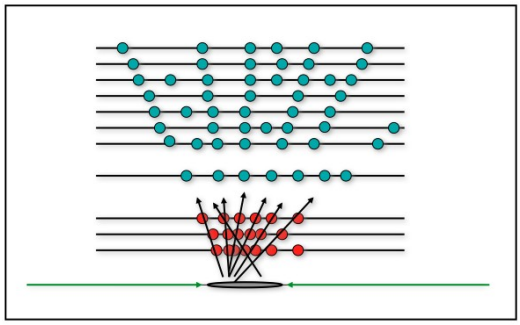
Introduction

CMS running on its peak has collisions of bunches every 25 ns. Each bunch is consist of about 10^{11} protons. At each collision there are around 25 events in 2012 LHC and it will be around 40 events in Run II. So the event rate is around 1 GHz (10^9 Hz). This huge amount of data cannot be stored for later analysis with current devices. Also interesting physics is rare and usually occurs in 10 Hz . So a reduction of several orders of magnitude in event rate is needed.

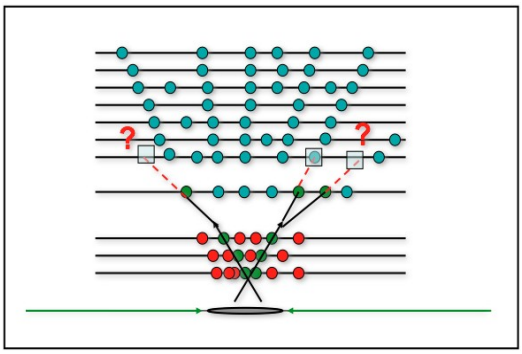
In CMS there are two kinds of triggers. One is the online trigger which happens while LHC is running and the other is the offline trigger which is done on the data stored at CERN data center. HLT(Higher Level Trigger) is an online trigger that is done after L1 trigger. L1 is a trigger that looks for simple signs of physics in a fast and automatic process . Online triggers are quite important due to their speed because when the next waves of particles come to the detector, they leave new paths in that and so they rewrite everything.

After the data, that is consist of many dots left by particles while passing through silicon layers, is gotten from the detector, a process called “track reconstructing” starts. The track reconstruction algorithms have a crucial role in HLT since they are used for reconstruction of physics objects and also the Primary Vertex(PV). The process of reconstructing tracks has 4 steps.

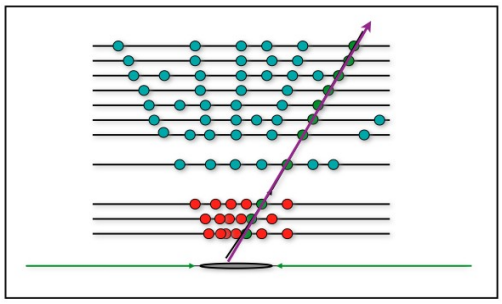
In “seeding”, the first step, seeds built in the inner part of the tracker are used to fit a rough path and estimate the momentum and the direction of the particles. In this step, the hits far from the Primary Vertex(PV) are discarded.



In the second step, with the help of rough estimates in the previous step, lines of trajectories are continued using “Kalman filter technique” to find more and more hits on their ways. Passing layer by layer, the parameters of the tracks are updated. If not enough hits are found in the direction of the line, which the track has to continue its path, the candidate is discarded.



Third step is the final track fitting. Best parameters are assigned to the tracks and the final lines are fitted using Kalman filter technique and smoother. In this step the hits, not compatible with the fits, are eliminated to better the parameters.



The fourth step is called “track filtering” that is discussed below.

Track Filtering in HLT

In order to find the real tracks in these huge numbers of tracks, track filtering in HLT is done in 4 steps that is called “iteration”. At each step a subset of reconstructed tracks is chosen. The more real tracks in our subset, the more efficient choice we have. The idea is to start with more energetic tracks (Tracks with higher P_T) and then remove the hits associated with those tracks and after that

repeat the procedure again. All are now happening in 4 steps(4 Iters) in CMS. Iter0, Iter1, Iter2 and Iter4. Iter0 mostly reconstructs high P_T tracks (tracks with P_T>0.9 GeV) by using seeds from pixel tracks. In Iter1 low P_T tracks with triplet seeding, in Iter2 tracks with pair pixel hits and in Iter4 tracks originated far from the beamspot are chosen.

Why track selection is important?

When we can fit the best tracks with given hits, why do we need track selection? One reason is that some hits are not real! There are some hits because of the noise in the devise or they are just duplicates so we must tackle with real and fake hits. Another reason is that the hits are just points and not a path. Based on your choice in the seeding part, you may fit a line to some hits that belong to a couple of particles. Although crossing of two trajectories rarely happens (since 2 line in 3 dimensional space rarely meet each other), the number of hits are very much specially in the nearest layers due to the path of low P_T particles and being near the center in a constant solid angle. So there are many fake trajectories that we must eliminate, because of online storage problems and because they do not have physics.

What I did!

The program used for CMS tracking selection is called “Analytical Track Selector”.Cuts on tracks are put once in each iteration. My final purpose in this project is to substitute the “Analytical Track Selector” with “Multi Track Selector”. In Multi Track Selector, as one can guess from the name, more than one selection is done in each iteration.

For testing the quality of the cuts, 1000 events monte carlo simulation run each time. In this simulation everything is taken into account, from the probability of creation and annihilation of particles, that is gotten from theoreticians, to the structure of the device specially the silicon layers. Then the procedure that is done online, now is done offline. It means, some hits are gotten from the simulation and then the 4 steps of reconstructing tracks are done. Now that we have the reconstructed tracks we can compare them with the simulated tracks to see whether we have chosen the right ones or not. In each cut that we put, there are 2 tools to measure the quality of the cuts. One is the “efficiency” which is the number of tracks that we've chosen it right over the whole number of right tracks. A reconstructed track is labeled as a “right” track if it is compatible with the simulated track in 75% of the hits. The other tool is the “fake rate” which is the number of fake tracks in our cut, over the number of reconstructed tracks in the cut.

My project has two parts. The first one is to substitute Analytical Iter0 with Multi Track Iter0 using new cuts and compare the efficiency and fake rate to see the changes. And the second one is to do the same for Iter1 using Multi Track for Iter0. The final changes, that were the best ones, are shown in the table for Iter0.

	Analytical Iter0	MultiTrackSelector Iter0				
		iter0Loose	iter0Tight (pre: iter0Loose)	iter0v1 (pre: Iter0Tight)	iter0v2	iter0v3
minNumberLayers	3	0	3	3	3	5
chi2n_par	0.7	1.6	0.7	0.7	0.7	0.3
nSigmaZ	3	4	4	3	4	4
dz_par2	0.4 4	0.45 4	0.4 4	0.4 4	1.1 3	1 4
dz_par1	0.35 4	0.65 4	0.35 4	0.35 4	1 3	1 4
maxNumberLostLayers	1	999	2	1	1	0
d0_par2	0.4 4	0.55 4	0.4 4	0.4 4	1.1 3	1 4
d0_par1	0.3 4	0.55 4	0.3 4	0.3 4	1.0 3	1 4
res_par	0.003 0.001	0.003 0.01	0.003 0.01	0.003 0.001	0.003 0.001	0.003 0.001
minHitsToBypassChecks	20	20	20	20	20	20

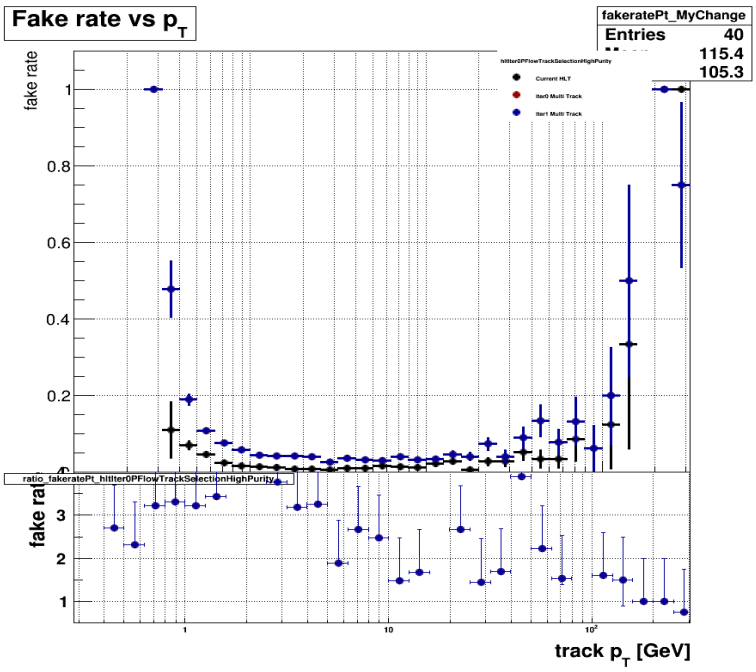
You can see the cuts for Analytical and Multi Track Selector in the table. The final cuts for Iter0 High Purity is the merge of Iter0v1 and Iter0v2 and Iter0v3. Since Iter0v3 is searching for further-originated tracks, we used tighter cuts in that. Iter0v1 has Iter0Tight and Iter0Tight has Iter0Loose as prefiltered cuts.

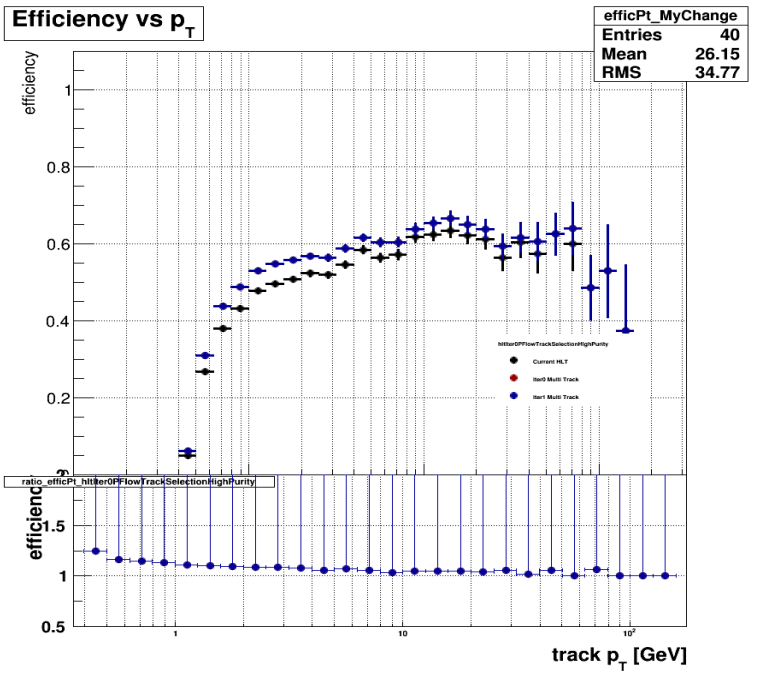
Next table is for Iter1 part.

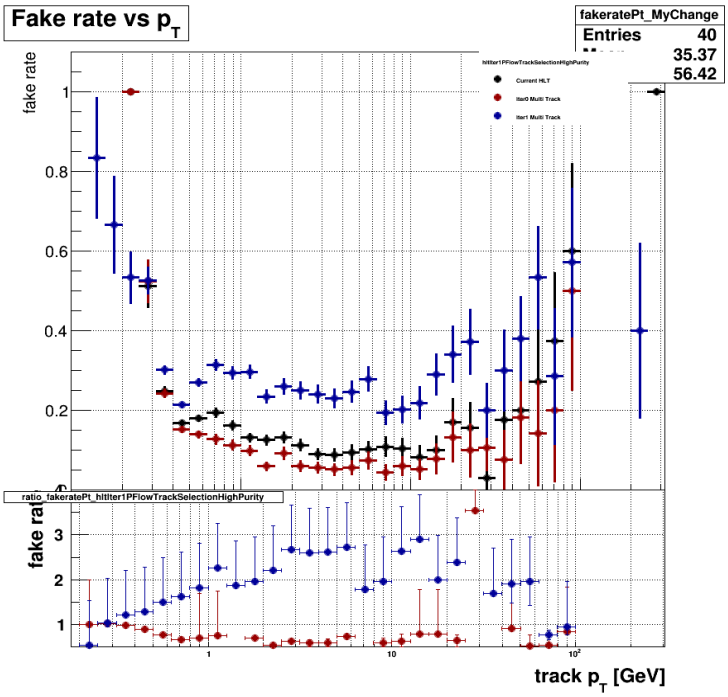
	Analytical Iter1		MultiTrackSelector Iter1		
	Loose	Tight	Iter1Loose	Iter1Tight (pre: Iter1Loose)	Iter1HighPurity (pre: Iter1Tight)
chi2n_par	0.7	0.4	3.5	1.3	1
res_par	0.003 0.001	0.003 0.001	0.003 0.002	0.003 0.002	0.003 0.001
minNumberLayers	3	5	3	3	3
maxNumberLostLayers	1	1	2	2	2
minNumber3DLayers	0	0	0	0	0
d0_par1	0.85 3	1 4	0.9 4	0.75 4	0.7 4
dz_par1	0.8 3	1 4	0.7 4	0.6 4	0.55 4
d0_par2	0.9 3	1 4	0.5 4	0.4 4	0.3 4
dz_par2	0.9 3	1 4	0.5 4	0.4 4	0.35 4

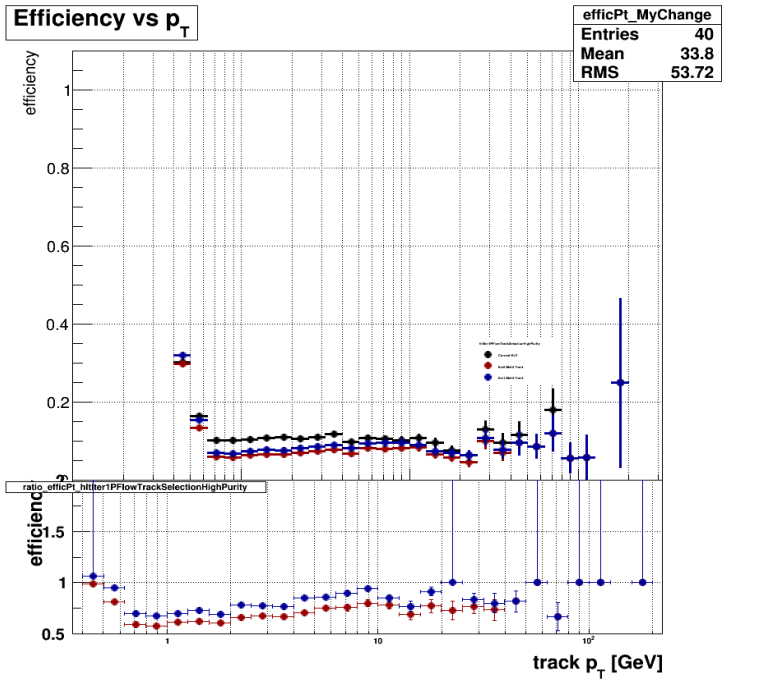
Here also you can see the prefiltered subsets. The Analytical Track Selector for Iter1 has two parts, one is “Loose” and the other is “Tight”. We substitute these two with a merge of 3 subsets: Iter1 High Purity, Analytical Iter1 Tight and Analytical Iter1 Loose. You can see the final parameters(cuts) in the table. The results of these changes are shown as follows:

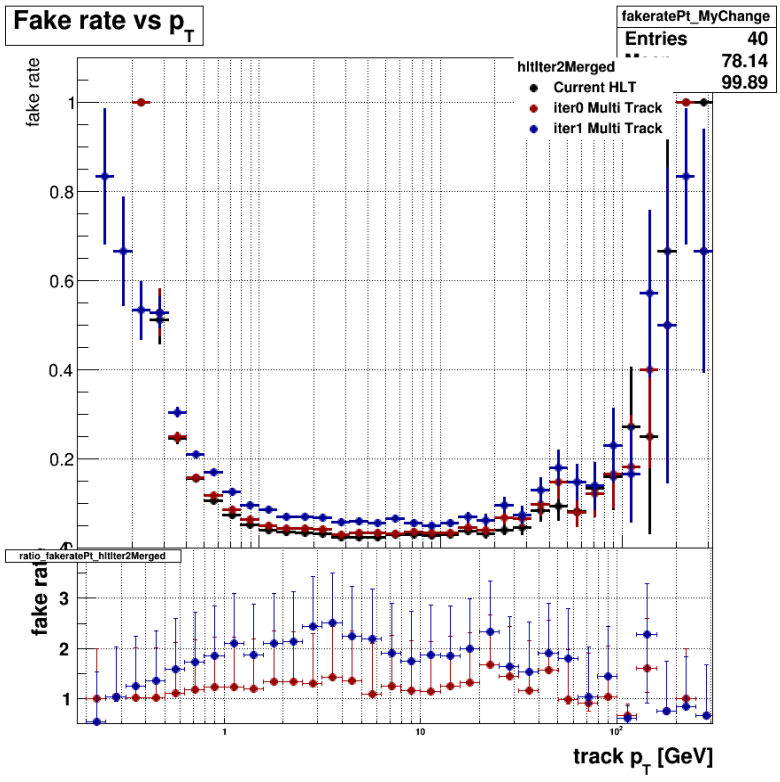
(In plots with 2 colors, Iter0 and Iter1 completely overlapped)

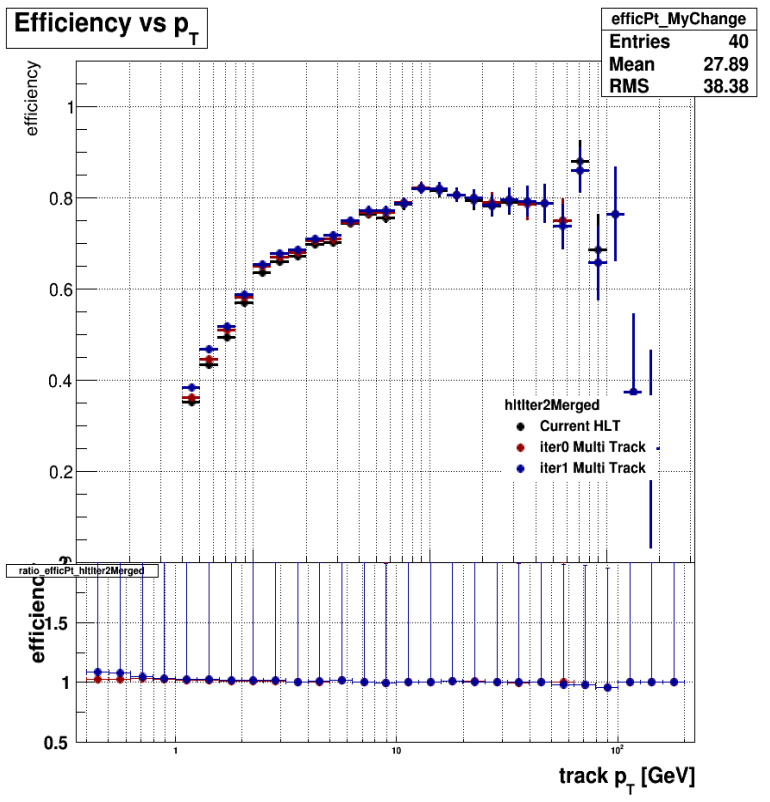

Fake rate vs p_T
fakeratePt_MyChange
Entries 40
115.4
105.3
fake rate
track p_T [GeV]

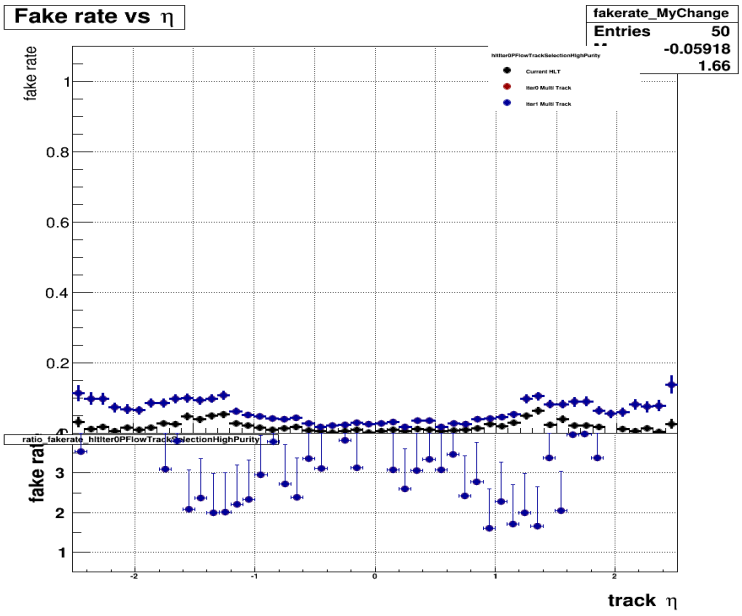

Efficiency vs p_T
efficPt_MyChange
Entries 40
Mean 26.15
RMS 34.77
efficiency
track p_T [GeV]

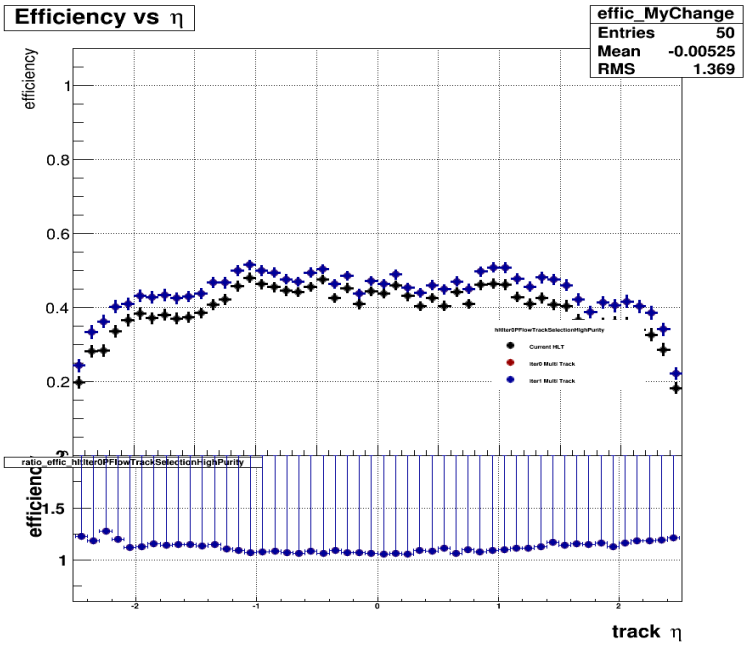

Fake rate vs p_T
fakeratePt_MyChange
Entries 40
35.37
56.42
fake rate
track p_T [GeV]

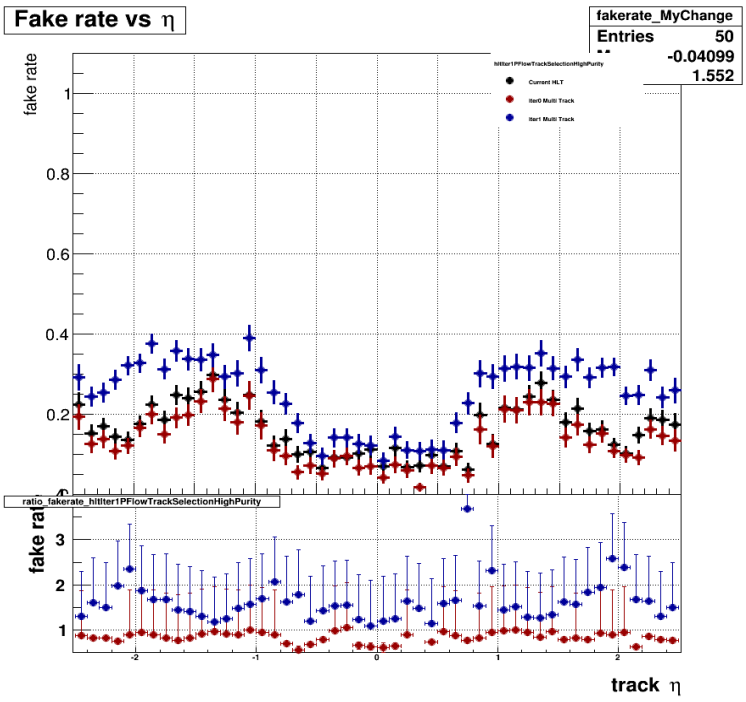

Efficiency vs p_T
efficPt_MyChange
Entries 40
Mean 33.8
RMS 53.72
efficiency
track p_T [GeV]

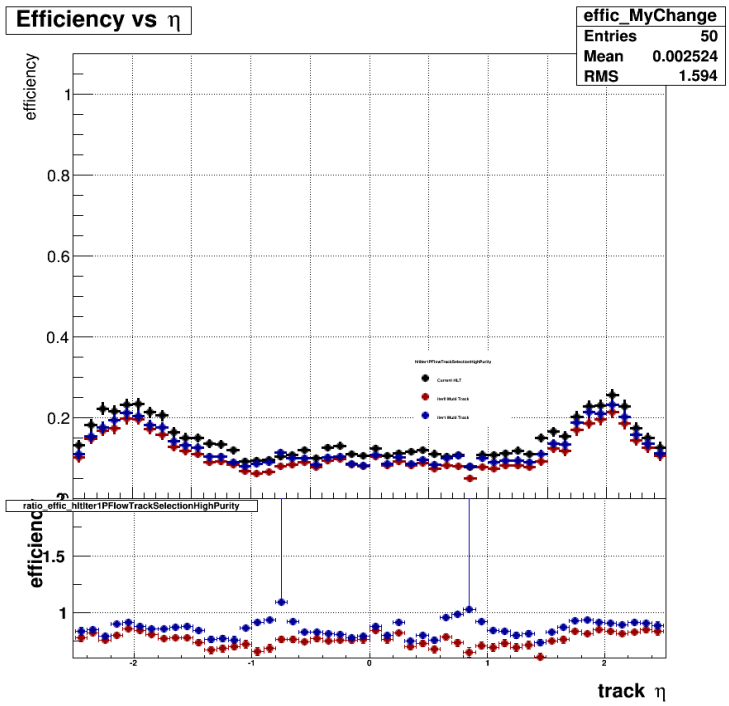

Fake rate vs p_T
fakeratePt_MyChange
Entries 40
78.14
99.89
hltIter2Merged
Current HLT
iter0 Multi Track
iter1 Multi Track
fake rate
track p_T [GeV]

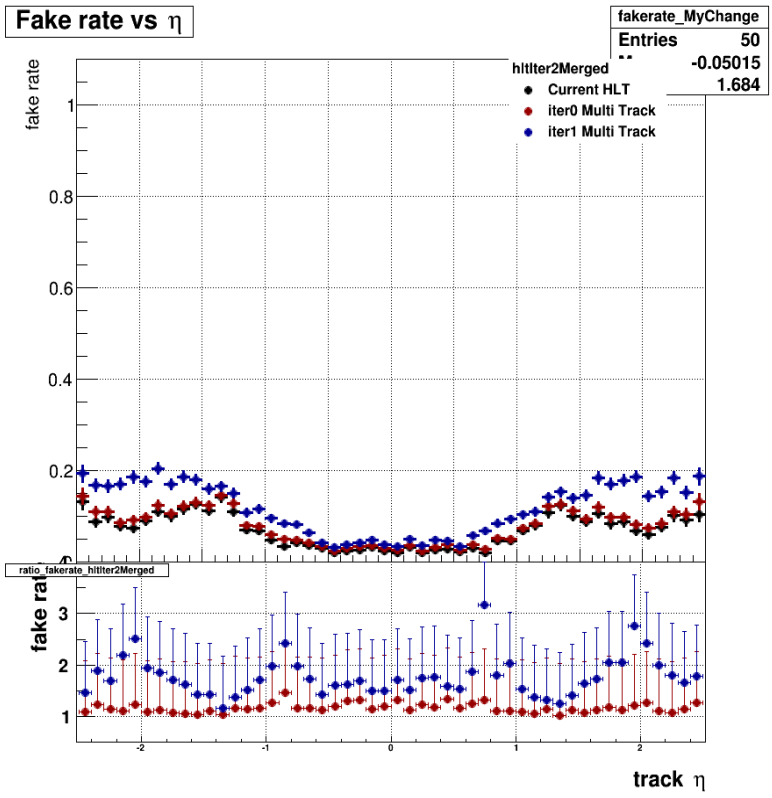

Efficiency vs p_T
efficPt_MyChange
Entries 40
Mean 27.89
RMS 38.38
hltIter2Merged
Current HLT
iter0 Multi Track
iter1 Multi Track
efficiency
track p_T [GeV]

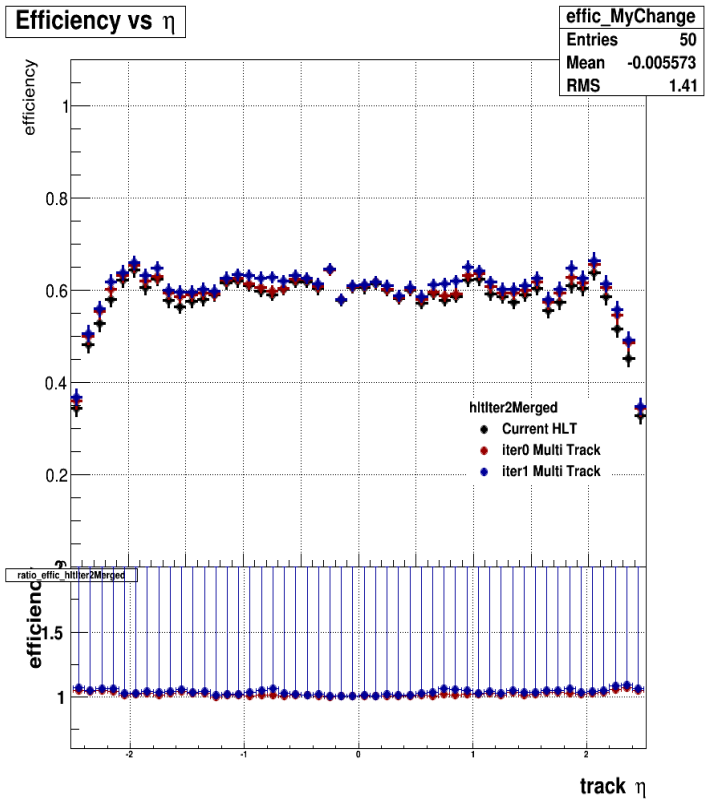

Fake rate vs η
fakerate_MyChange
Entries 50
-0.05918
1.66
fake rate
track η
ratio_fakerate_hItIter0PFlowTrackSelectionHighPurity

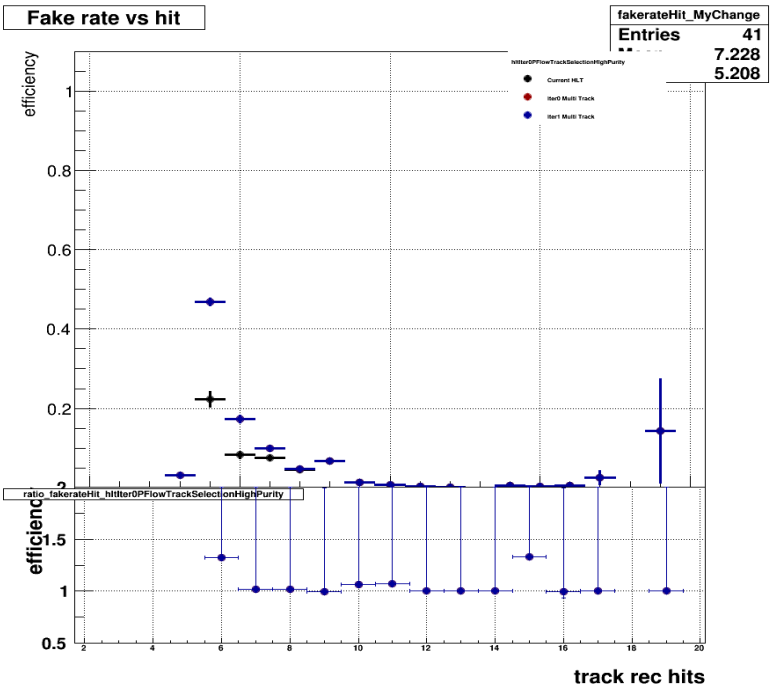

Efficiency vs η
effic_MyChange
Entries 50
Mean -0.00525
RMS 1.369
efficiency
track η
ratio_effic_hItIter0PFlowTrackSelectionHighPurity

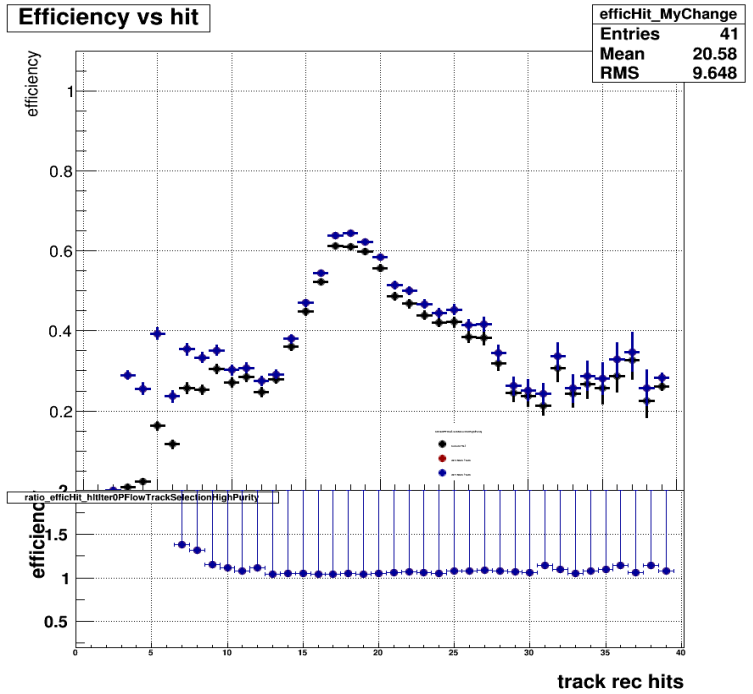

Fake rate vs η
fakerate_MyChange
Entries 50
-0.04099
1.552
fake rate
track η
ratio_fakerate_hItIter1PFlowTrackSelectionHighPurity

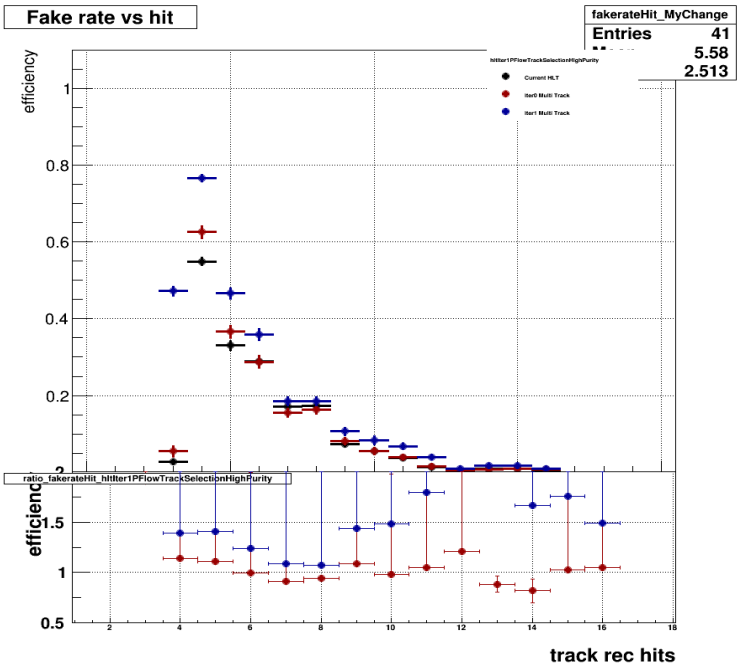

Efficiency vs η
effic_MyChange
Entries 50
Mean 0.002524
RMS 1.594
efficiency
track η
ratio_effic_hItIter1PFlowTrackSelectionHighPurity

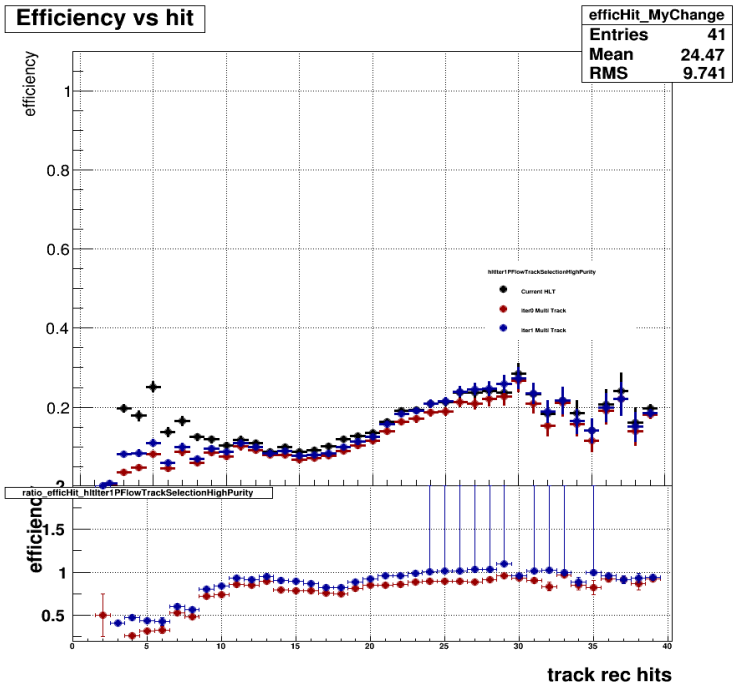

Fake rate vs η
fakerate_MyChange
Entries 50
-0.05015
1.684
hItIter2Merged
Current HLT
iter0 Multi Track
iter1 Multi Track
fake rate
track η
ratio_fakerate_hItIter2Merged

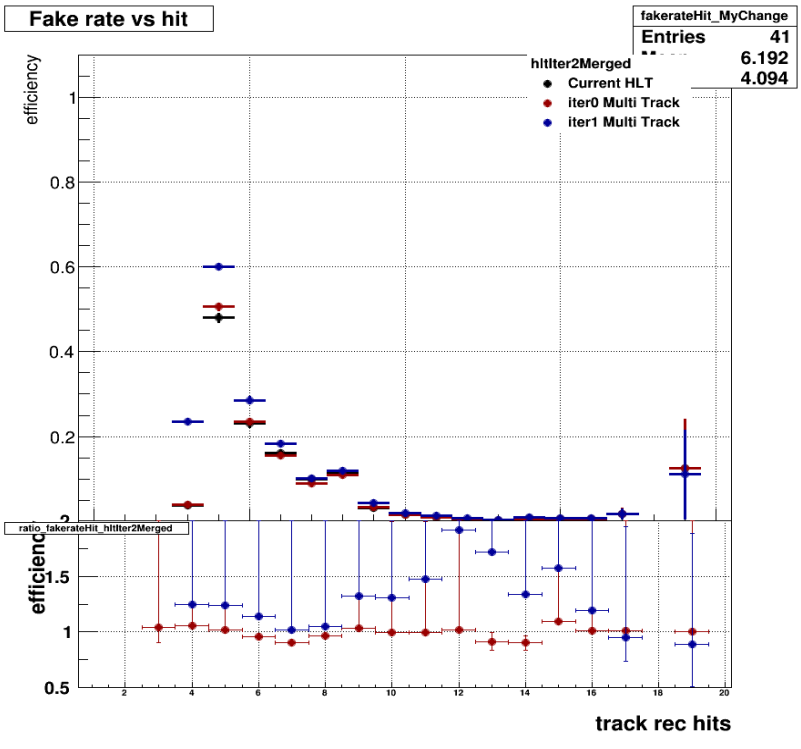

Efficiency vs η
effic_MyChange
Entries 50
Mean -0.005573
RMS 1.41
hItIter2Merged
Current HLT
iter0 Multi Track
iter1 Multi Track
efficiency
track η
ratio_effic_hItIter2Merged

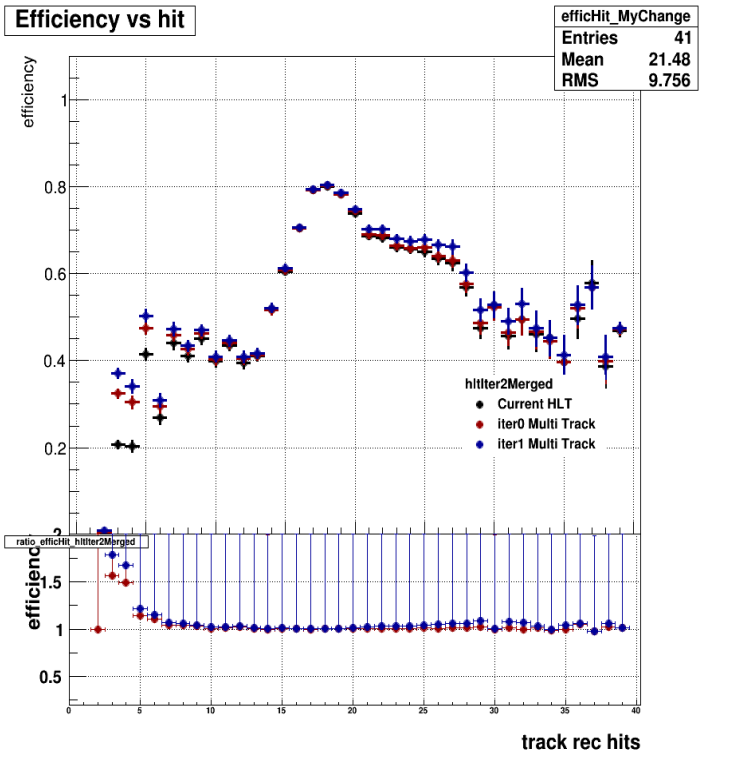

Fake rate vs hit
fakerateHit_MyChange
Entries 41
7.228
5.208
efficiency
track rec hits


Efficiency vs hit
efficHit_MyChange
Entries 41
Mean 20.58
RMS 9.648
efficiency
track rec hits


Fake rate vs hit
fakerateHit_MyChange
Entries 41
5.58
2.513
efficiency
track rec hits


Efficiency vs hit
efficHit_MyChange
Entries 41
Mean 24.47
RMS 9.741
efficiency
track rec hits


Fake rate vs hit
fakerateHit_MyChange
Entries 41
6.192
4.094
hItIter2Merged
Current HLT
iter0 Multi Track
iter1 Multi Track
efficiency
track rec hits


Efficiency vs hit
efficHit_MyChange
Entries 41
Mean 21.48
RMS 9.756
hItIter2Merged
Current HLT
iter0 Multi Track
iter1 Multi Track
efficiency
track rec hits