Variable Scanning for Physics Efficiency Improvement in Upstream Tracking and Trackmatching Validation Algorithms in the Allen Framework

Student: Mauricio Rodríguez Alas

Supervisors: Johan Sebastian Bonilla

CERN Summer Student Program

This document summarizes the work done of the Upstream Tracking (UTStandalone Sequence) and the Trackmatching Validation (trackmatching_validation sequence) in the Allen framework. “UTStandalone” and “Track Matching” validation algorithms to improve the physics performance running on CPU using the Allen framework. Variable scanning and implementation of modified sequences and algorithms were implemented, to scan through multiple variables and find the optimal efficiency points while reducing reconstruction ghost rates i.e. fake tracks. Most of the optimization was focused on the Long tracks and Long high momentum (>5 GeV), and the reconstruction without the Velo tracker.

1. Introduction

The Allen framework is a full software HLT1 reconstruction sequence on GPU.
For this summer project, two sequences were selected to scan its variables and find the optimal efficiency point vs ghost rates, namely the Upstream Tracker Validation Sequence and the Track Matching Validation Sequence, technically written as *UTStandaloneValidationSequence.py* and *TrackMathingValidationSequence.py* respectively within the Allen framework. Multiple variables were added to the algorithm definitions using the remote terminal via lxplus. After the variables were added, multiple values were tested for each of the algorithms, recording the reconstruction efficiency, ghost rates and physics performance on CPU to find the optimal performance point and share these results with future Allen users looking to optimize reconstruction efficiency.
The plots and value analysis were done using Python and Jupyter Notebooks.

2. Upstream Tracking Validation Algorithm

For the upstream tracking algorithm the first variable that was added was the chi2cut, which was scanned from a value of 0.1 to 1.9. The optimal point is somewhere between 0.3 to 0.5 which maximizes efficiency and minimizes ghost rates while maintaining the maximum number of track candidates.

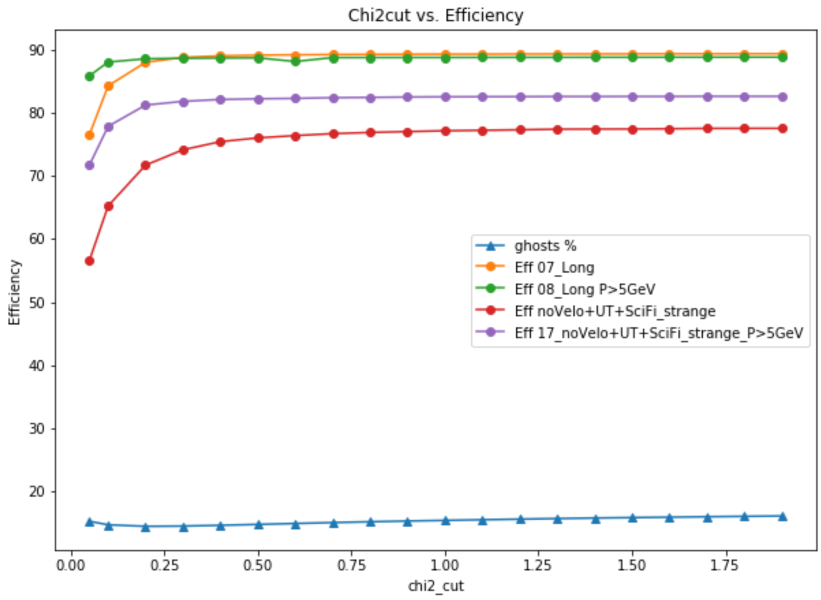

Figure 1. Chi2cut vs Efficiency plot in Upstream Tracking Validation algorithm. Source: Author.

The second variable scanned in the UT validation was the xMatch_cut variable, which was scanned from a value of 10 to 80, having an optimal point between a value of 20 to 30, depending on the track that we want to study. Values above 40 increase ghost rates above 10%, and rapidly increase toward higher values while not giving a significant improvement of efficiency rates.

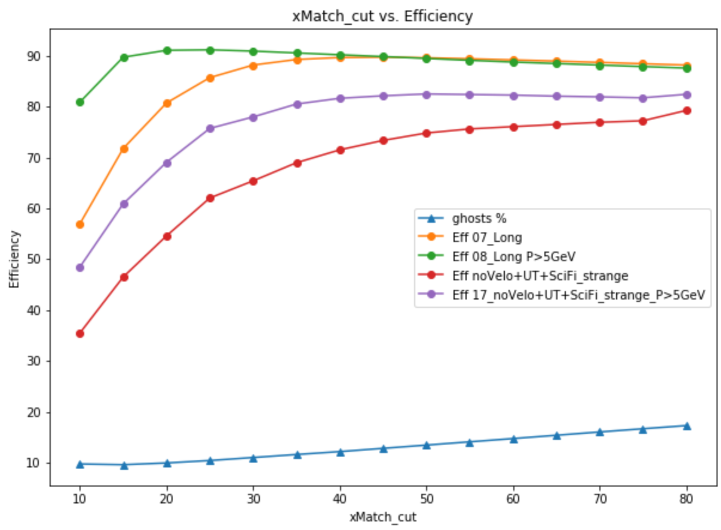

Figure 2. xMatch cut vs. Efficiency plot in Upstream Tracking Validation algorithm. Source: Author.

3. Trackmatching Validation Algorithm

For the tracmathing validation sequence the first variable that was scanned was the distribution in X or distX, scan was done from 5 to 100, showing an already optimized point between 25 and 30, efficiency being higher in the long and high momentum tracks. Ghost rates decrease rapidly for the first scan points at the cost of track samples in the final results.

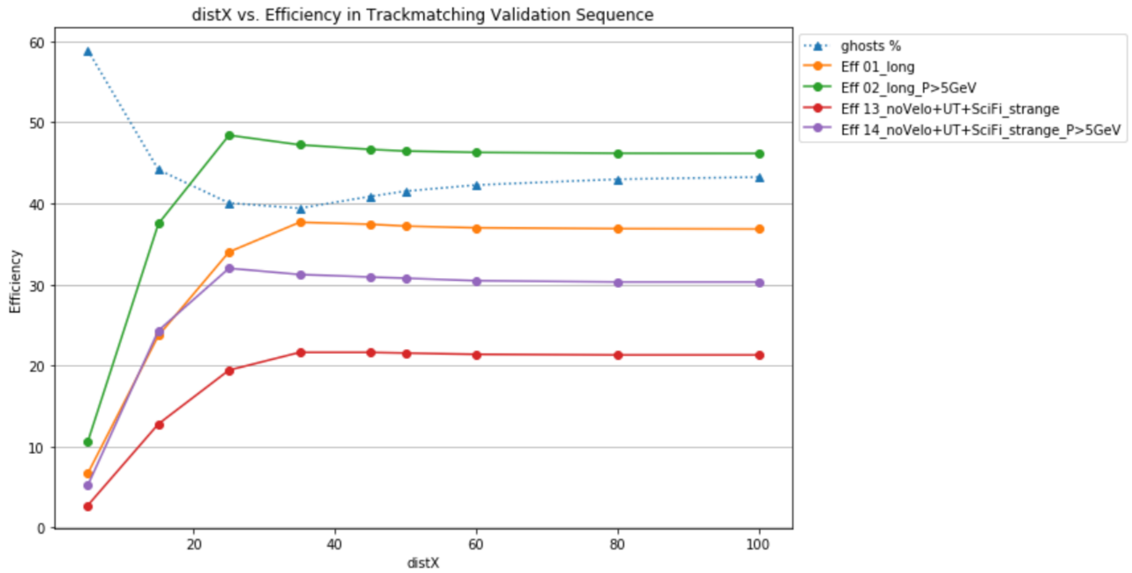

Figure 3. distX vs Efficiency plot in Track Matching algorithm. Source: Author.

Secondly, the distribution in y was scanned, as expected this did not have too much of an effect in the track matching algorithm, which is barely affected in the Y direction due to the magnetic field. Scan was done from 50 to 250 mm, being the optimal point close to 150 mm.

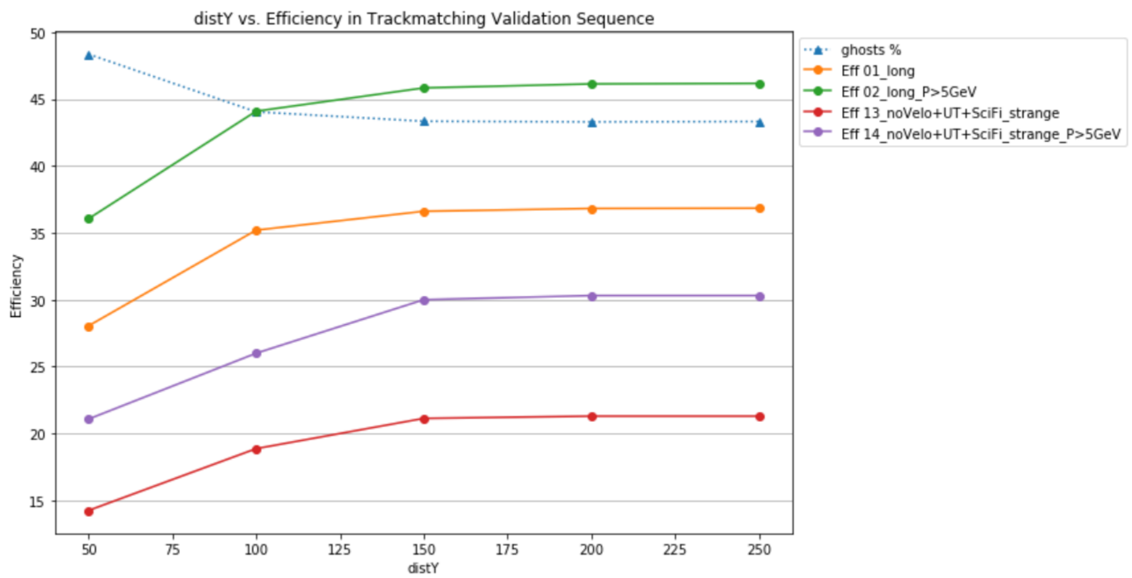

Figure 4. distY vs Efficiency plot in Track Matching algorithm. Source: Author.

Third variable that was scanned was the dx tolerance or dxTol, which was scanned from 1 to 10, having a negative impact on the efficiencies but a significant increase in ghost rates in the higher values. An optimal point from 1 to 2 was found.

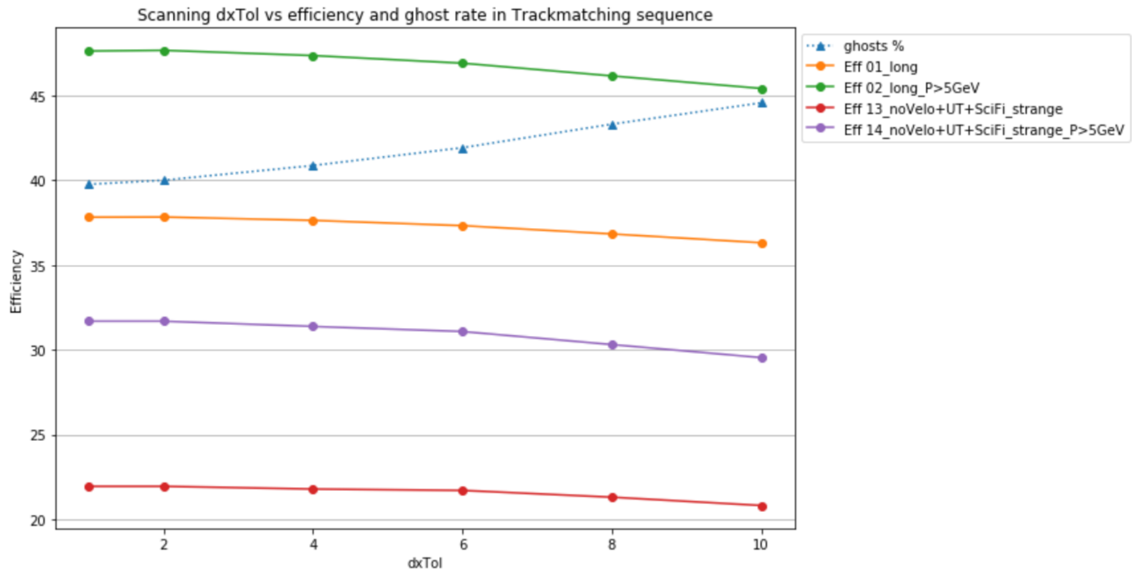

Figure 5. dxTolerance vs Efficiency plot in Track Matching algorithm. Source: Author.

The same scan was done for the dy Tolerance, which had a bigger impact on the efficiencies and ghost rates. An increase of close to 5% efficiency was achieved for the Long high momentum tracks, finding an optimal point between 15 to 25, at the cost of the number of tracks reconstructed.

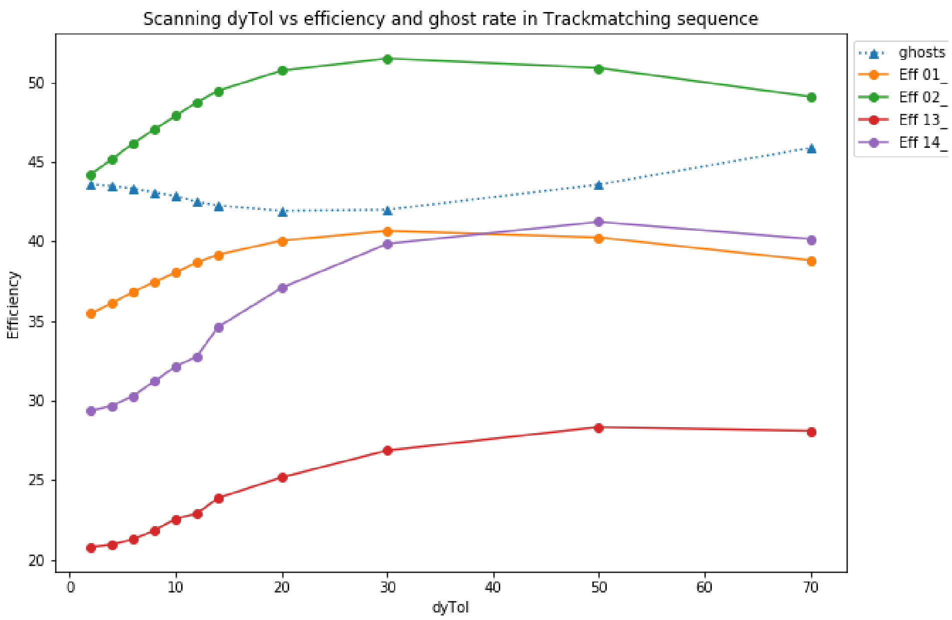

Figure 6. dyTolerance vs Efficiency plot in Track Matching algorithm. Source: Author.

4. Efficiency Improvement Results

Variable/Track	Long	Long P>5 GeV	noVelo+UT+SciFi strange	noVelo+UT+SciFi strange P>5 GeV
chi2cut	90	91	75	80
xMatchCut	90	90	70	80

Table 1. Upstream Tracking Validation Algorithm best efficiencies (%). Source: Author.

Variable/Track	Long	Long P>5 GeV	noVelo+UT+SciFi strange	noVelo+UT+SciFi strange P>5 GeV
distX	35	50	20	30
distY	35	45	20	30
dxTolerance	35	50	25	35
dyTolerance	40	55	40	27

Table 2. Trackmatching Validation Algorithm best efficiencies (%). Source: Author.

5. Conclussion

A best case improvement of 10% in the reconstruction efficiency was achieved for the Long tracks using the Trackmatching Sequence with the dy Tolerance parameter. Further efficiency improvements would require a multivariable automated analysis of these parameters and adding more variables to the sequences.

A lot of work still needs to be done in the algorithms itself before full optimization, as shown specially by the track matching sequences, which show significant improvement of the efficiencies at the cost of a lot of the reconstructed tracks, and show efficiencies well below acceptable levels.

Further work includes parallelization of algorithms and optimization of these and other sequences to run on GPUs. Before this can be achieved the baseline algorithms should be improved further, making use of the physics performance and comparing ghost rates to lost tracks and candidates.