

# Common Validation Code Base Discussion

**Thomas Hauth**

Tracking Meeting – 9.1.2015

# Introduction

- Shortly before Christmas last year, I started to investigate the possibility to have a common code base for tracking validation purposes

This common code will be useful in the following areas:

- Production of the validation plots for the nightly builds website
  - Documenting quality improvements to tracking code by developers
  - Quick & local feedback loop for developers modifying the tracking code
- My intention was to look for (already existing ?) code which offered a superset of all features of the current validation scripts:
    - Track finding efficiencies and fake rate etc.
    - Fitting quality criteria: residual and pull distributions of fitted tracks etc.

# Current Status

- Until recently, the validation scripts in the category “tracking” used totally different code bases to create their plots, although the plots are very similar
  - Oliver unified the cdcLegendre, cdcLocal and cosmics validation to use the same code base last week, running in the nightly validation now
- Giulia and Oliver were so kind to provide me with their most recent versions of their tracking validation code bases
- Both use the MCTrackMatcher module to create MC <> RECO association to determine efficiency and fake rate
- Both output their results to a root file in the form of histograms and profile plots

Giulia's validation script is a C++ module and located in  
***tracking/modules/trackingPerformanceEvaluation***

Oliver's validation scripts is a set of python-based classes located  
***tracking/scripts/tracking/validation***  
and a set of python scripts implementing concrete validation runs located in  
***tracking/validation/***

You can find the output of Oliver's validation runs here:

<https://belle2.cc.kek.jp/validation/#tracking> (but not all created plots are on the val. page, yet)

# Comparison of implemented plot set

Following is a list of the most important finding & fitting quantities provided by the two implementations:

## ***TrackingPerformanceEvaluation Module***

- Finding Efficiency over Pt, Theta, Phi
- Purity over Pt, Theta, Phi
- Multiplicity of tracks, fitted tracks per MC particle
- Error, Residuals and Pulls of  $d_0$ , Phi, Omega,  $z_0$ ,  $\cot\Theta$
- Number of used hits per fit and projections on the xy, rz planes

## **tracking/scripts/tracking/validation scripts**

- Finding Efficiency over  $d_0$ ,  $\tan(\lambda)$ , multiplicity, pt
- Hit efficiency over  $d_0$ ,  $\tan(\lambda)$ , multiplicity, pt
- Residuals, Pulls, P-Values, Errors over  $\tan(\lambda)$  & Omega
- Scalar numbers for Finding Efficiency, Fake Rate, Clone Rate and Hit Efficiency

# Discussion points

- What do developers expect from a common tracking validation code base ?
  - Would you be willing to use it to implement validation for your specific tracking modules?
  - Would you like to use it during your development process to track bugs or evaluate improvements?
- Are there any suggestions to the quantities ( in addition to the ones listed on the previous page ) which we should include ?
- How can we proceed in combining the nice work which has already been done by Giulia and Oliver ?
- On my todo-list, once we agreed on the next step:
  - Modify the rest of the tracking/validation scripts to use the common validation code base
  - Create a twiki page to document the feature and usage of the common validation code base