Plan for finalising TrackCalib2

Martin Tat

Heidelberg University

2nd December 2025

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Introduction

Current situation:

- Analysis productions produce tuples...
- ... but there are two sets of analysis frameworks
  1. Rowina's code
  2. TrackCalib2
- In principle both should do the same job $\Longrightarrow$
  - Use TrackCalib2 by default

Keeping TrackCalib2 up-to-date

Rowina, Maurice and I have gone through TrackCalib2 and Rowina's code to make sure everything is consistent

1. Selection criteria
2. Matching criteria
3. Fit range
4. Latest AP with data/MC tuples
5. Binning schemes
   - Use Rowina's current 1D finer binning schemes in p and η
   - In 2D, start from the 1D x 1D binning and merge bins until each bin has a sufficient number of VeloMuon x Downstream candidates

- Updating 1–4 in TrackCalib2 is work in progress
- Fancy 2D binning needs more thinking and code refactoring

Developments in progess

TrackCalib2 needs more flexibility like Rowina's code:

1. Tuple preparation is currently by far the biggest bottlekneck
   - Aim: Perform offline selection of new tuples on-the-fly
   - Current status: Tuples take hours to produce and each data block takes over 10 GB of space
   - Bottlekneck: Uproot processes hundreds of files, but only one at the time, without any parallelisation
   - Solution: Change to RDataFrame, which performs selections lazily and multithreaded, and only save necessary variables
   - I am currently working on this

Developments in progess

TrackCalib2 needs more flexibility like Rowina's code:

2. Fitting can take time when tweaking or doing studies
   - Aim: Fit different 1D and 2D binning schemes, and allow for single-bin fits, potentially in parallel
   - Current status: Same 1D and 2D binning is used, and every single 1D and 2D bin is fitted each time
   - Solution: Need to rethink how to save fit results
   - This is the next task

What is the end goal?

Why am I doing this?

- Have a tool that can produce tracking efficiencies out-of-the-box
- Quickly produce correction tables when new data/MC is available
- Allows for more efficient debugging of current issues:
  1. MuonUT vs Combined discrepancies
  2. Blocks 5/6 and 7/8 discrepancies