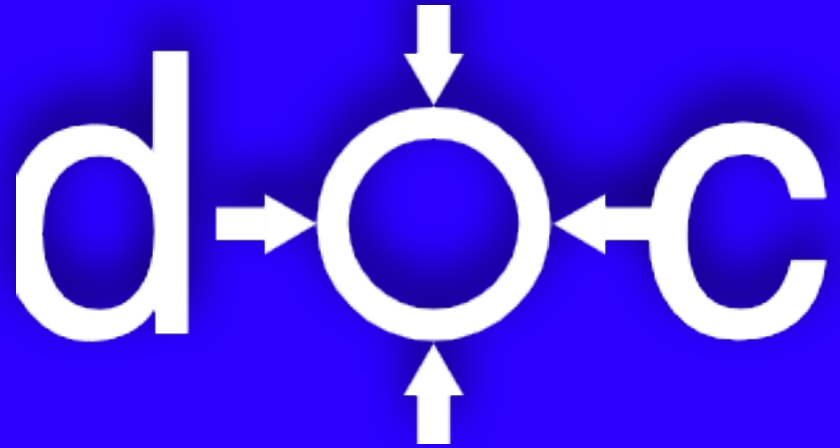
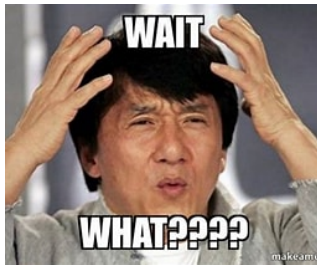


From Chaos to Clarity: CBM documentation



CBMROOT / Analysis documentation:

- Who of us not had/has problem with understanding the code of our colleagues?
- How about the beginners with less knowledge of the framework? Especially when there is no overlap with the seniors.



```
fHM->H1("fhMCTrueParticles")->Fill(4);

TLorentzVector lorVec1;
mctrack->Get4Momentum(lorVec1);
MCCascadeProton* fMCCascadeProton = new MCCascadeProton();
fMCCascadeProton->fLorentzVector = lorVec1;
fMCCascadeProton->fTrack = mctrack;
fMCCascadeProton->motherID = mctrack->GetMotherId();
fMCCascadeProton->ID = iTrack;

fMCCascadeProtonCands.push_back(fMCCascadeProton);
}

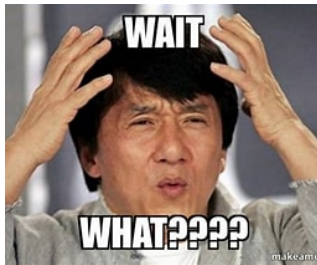
if (pdg == PDGPion && mctrack->GetCharge() < 0) {

fHM->H1("fhMCTrueParticles")->Fill(5);

TLorentzVector lorVec1;
mctrack->Get4Momentum(lorVec1);
MCCascadePion* fMCCascadePion = new MCCascadePion();
fMCCascadePion->fLorentzVector = lorVec1;
fMCCascadePion->fTrack = mctrack;
fMCCascadePion->motherID = mctrack->GetMotherId();
fMCCascadePion->ID = iTrack;
```

CBMROOT / Analysis documentation:

- Who of us not had/has problem with understanding the code of our colleagues?
- How about the beginners with less knowledge of the framework? Especially when there is no overlap with the seniors.



```
fHM->H1("fhMCTrueParticles")->Fill(4);

TLorentzVector lorVec1;
mctrack->Get4Momentum(lorVec1);
MCCascadeProton* fMCCascadeProton = new MCCascadeProton();
fMCCascadeProton->fLorentzVector = lorVec1;
fMCCascadeProton->fTrack = mctrack;
fMCCascadeProton->motherID = mctrack->GetMotherId();
fMCCascadeProton->ID = iTrack;

fMCCascadeProtonCands.push_back(fMCCascadeProton);
}

if (pdg == PDGPion && mctrack->GetCharge() < 0) {

fHM->H1("fhMCTrueParticles")->Fill(5);

TLorentzVector lorVec1;
mctrack->Get4Momentum(lorVec1);
MCCascadePion* fMCCascadePion = new MCCascadePion();
fMCCascadePion->fLorentzVector = lorVec1;
fMCCascadePion->fTrack = mctrack;
fMCCascadePion->motherID = mctrack->GetMotherId();
fMCCascadePion->ID = iTrack;
```

Many questions! one solution! — Documentation

Do we not have any?

Type to search

Introduction

Geometry

Simulation

Features

HowTo

Reconstruction

Structure

Features

mCbm Example

HowTo

Parameter

Creation

Container

Handling

DAQ Software

HowTo

Unpacking

Simulation

This chapter of the guide, takes the reader through the [simulation classes](#) of the rectangular pad design of the CBM-TRD.

This chapter summarizes the most important aspects of the simulation procedure as well as the necessary tools and options that are available there. However, it can only be seen as an introduction and a more detailed description of the implemented features can be found in the TRD [simulation note](#) and the PhD thesis of Etienne Bechtel ([PhD thesis](#)).

The TRD digitizer is supposed to simulate signal generation and collection within the MWPCs. This includes handling of TR photons, energy deposition in the gas volume and signal digitisation up to the readout ASIC.

Workflow structure

The general flow of information in a regular simulation can be described as:

Event generation → Transport → Detector response → Reconstruction

In this context the individual steps represent:

TRD group

CbmRoot

Main Page

Related Pages

Namespaces

Classes

Files

Search

CbmRoot

Interface to CBMROOT of littrack

CUDA implementation in littrack

ROOT macro and shell scripts

Main page of littrack documentation

Parallel implementation in littrack

QA and testing tools in littrack

Standard tracking implementation

Namespaces

Classes

Files

Interface to CBMROOT of littrack

main() function

CbmLitFindGlobalTracks is a main task to be called from macro in order to execute littrack tracking routines. The task automatically determines detector layout based on TGeo and configures track reconstruction in the TRD and MUCH detectors and hit-to-track merging in TOF.

Example:

```
CbmLitFindGlobalTracks* finder = new CbmLitFindGlobalTracks();

// Tracking method to be used
// "branch" - branching tracking
// "nn" - nearest neighbor tracking
// "nn_parallel" - parallel implementation of nearest neighbor tracking
finder->SetTrackingType("branch");

// Hit-to-track merger method to be used
// "nearest_hit" - assigns nearest hit to the track
finder->SetMergerType("nearest_hit");

run->AddTask(finder);
```

Field approximation

Polynomial field approximation is used in parallel tracking algorithm. Calculation of the polynomial coefficients for a specified field slice is implemented in the CbmLitFieldFitter class. Two options are implemented:

- Usage of Minuit for minimization of squared error between polynom approximation and field map;
- "Hand made" approximation based on least squares method.

Littrack

Background:

- Discussion with the physics coordinators: Anton and Chris: Necessary for detailed documentation of the CBMROOT/analysis framework.
- Idea to involve Juniors in developing a skeleton of the documentation, as we are the main users.
- We had internal discussion on November, 2024 and we overwhelmingly endorsed our efforts to engage and contribute to the documentation.
- Last CM in GSI, a task force is formed to make a skeleton of the documentation.

CBMROOT doc is ready:



Pavish



Abhishek



Adrian

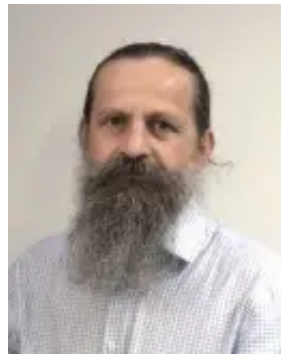
Thanks:



Karl-Heinz



Christian



Anton



Chris



Tetyana

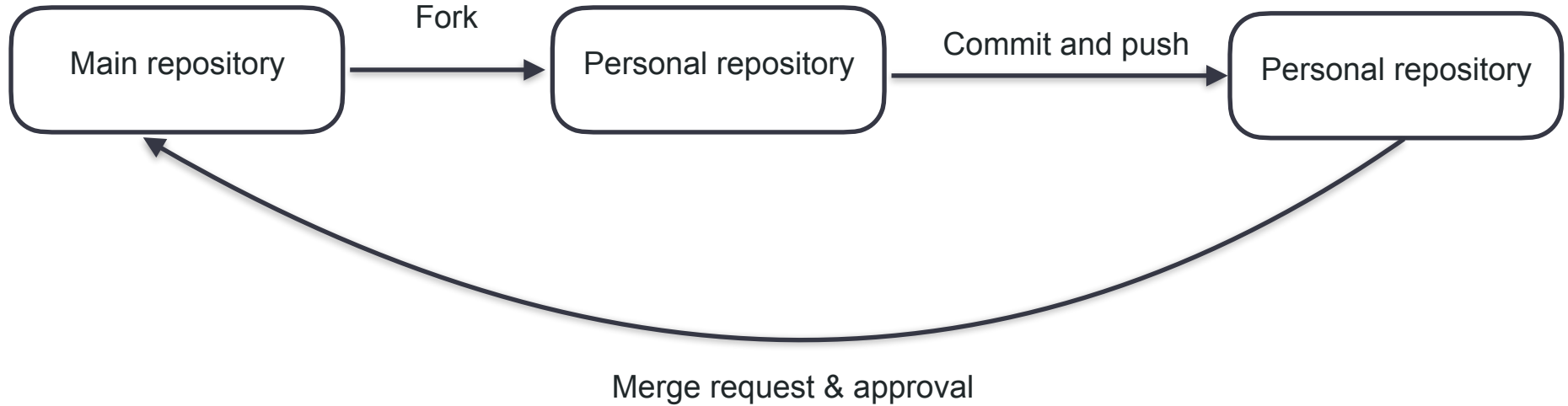
CBMROOT doc is ready:

At present available in: https://git.cbm.gsi.de/a_meye37_AT_uni-muenster.de/cbmroot-doc

At present deployed in: <https://cbmroot-doc-8ae785.gitpages.cbm.gsi.de/>



Workflow:

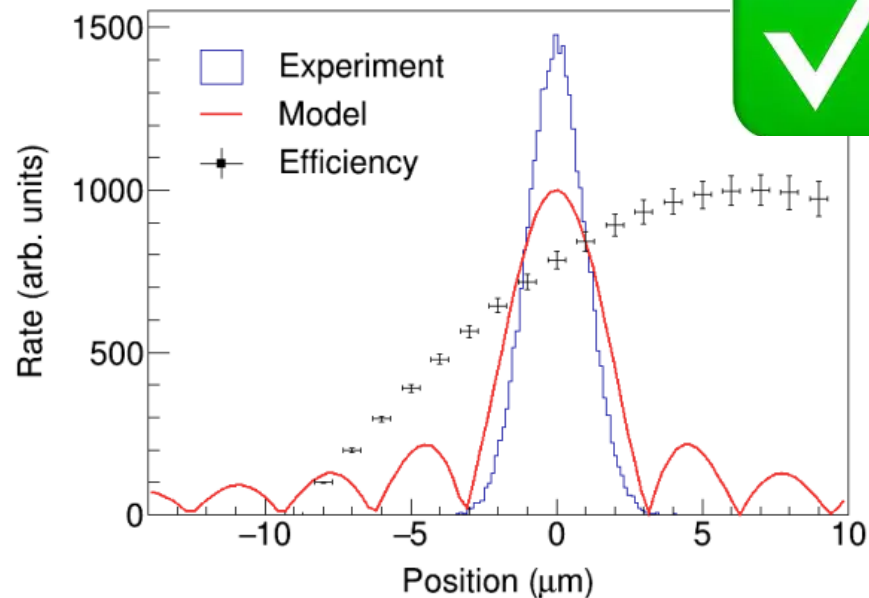
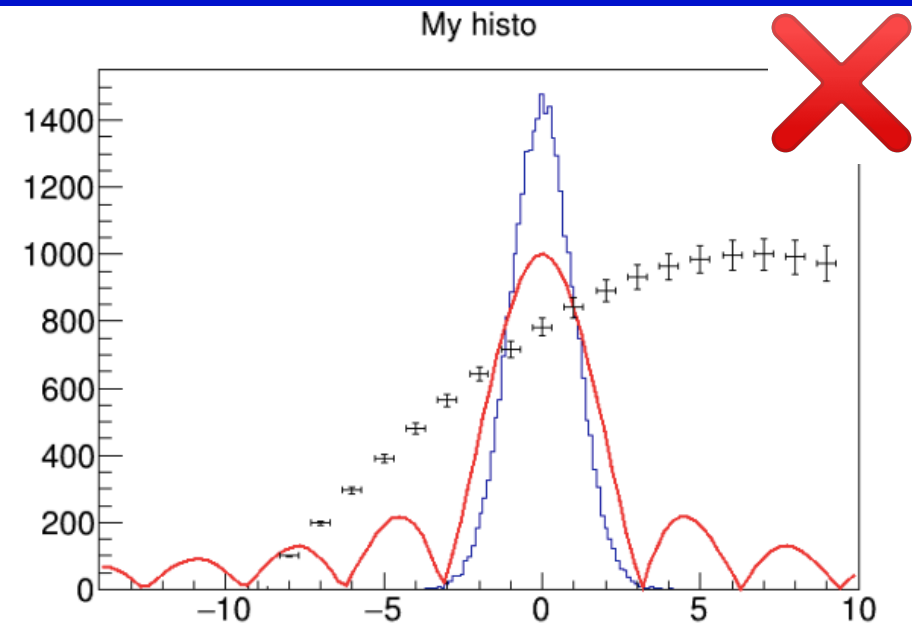


Features:

Features:

1. Getting started -> First steps towards analysis - Accounts - Installation- possibly tutorials.
2. Dedicated sections for simulation (Transport, Digi), Unpackers, Reconstruction, Various analysis frameworks.
3. Dedicated section for mCBM : Geometries, datasets and parameters, methods for Lambda reco.
4. External tools: Usage and documentation of Analysis tree, KFParticle, Femto, KinFit, ...
5. FAQs
6. Contributing to the documentation - Section explaining “How to” setup, edit and test the documentation.
7. *Summary page for easy click access - ALICE O2*

Best plotting practises:



Krzysztof Piasecki, Juniors seminar

https://indico.gsi.de/event/20920/contributions/84211/attachments/49315/71948/k.piasecki_good_plot_root.pdf

Features:

Features:

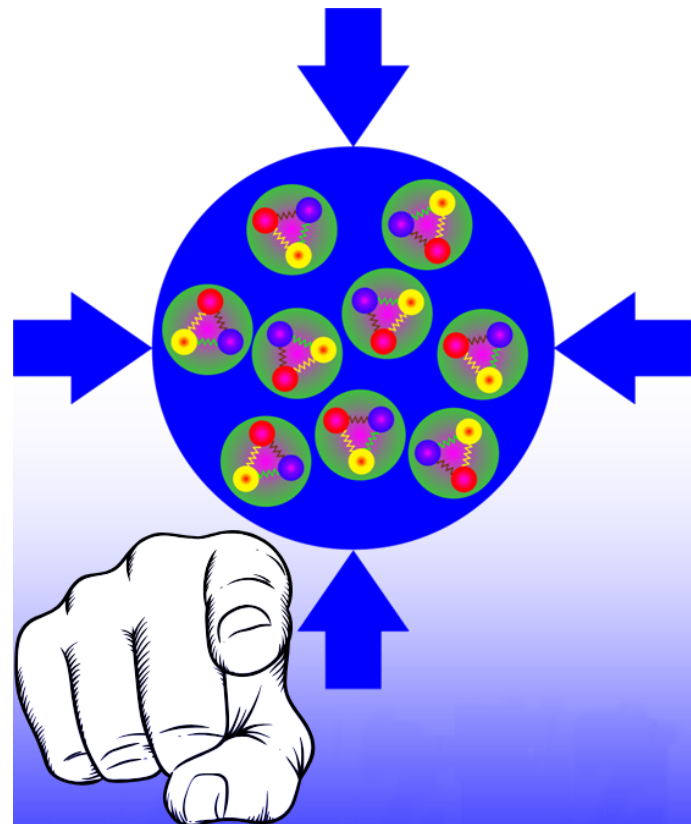
1. Getting started -> First steps towards analysis - Accounts - Installation- possibly tutorials.
2. Dedicated sections for simulation (Transport, Digi), Unpackers, Reconstruction, Various analysis frameworks.
3. Dedicated section for mCBM : Geometries, datasets and parameters, methods for Lambda reco.
4. External tools: Usage and documentation of Analysis tree, KFParticle, Femto, KinFit, ...
5. FAQs
6. Contributing to the documentation - Section explaining “How to” setup, edit and test the documentation.
7. *Summary page for easy click access - ALICE O2*
8. ***Best practices for plots***

CBMROOT doc is ready:

It is skeleton : Please contribute and build the repository..

From the words of Adrian the great:

“CBM needs you! This documentation lives from and depends on the contributions by all CBM members. Please consider sharing your knowledge by adding chapters, sections or sentences, no matter how small”



Thank you