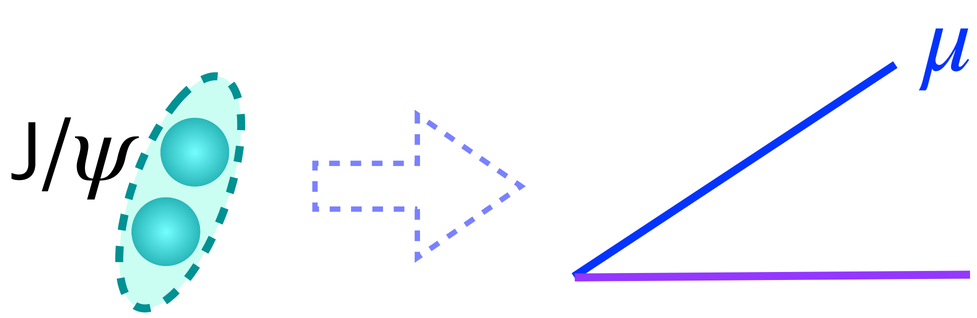
Tag-and-Probe efficiency study for muons in Alice

O2 tutorial

Batoul DIAB - 08/11/2022

The Tag-and-Probe method

- Tag-and-Probe (T&P) is a data-driven efficiency calculation technique
- Simulations are not ideal → need data calibration
- based on the decays of known resonances, e.g. J/ψ



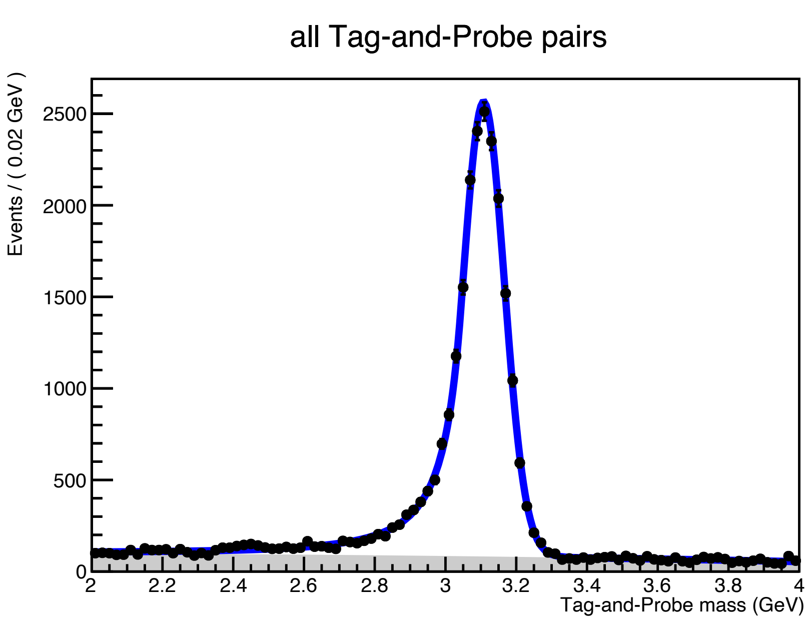
Well identified, Tight selection

Unbiased, very loose selection

Condition?

Passing probes

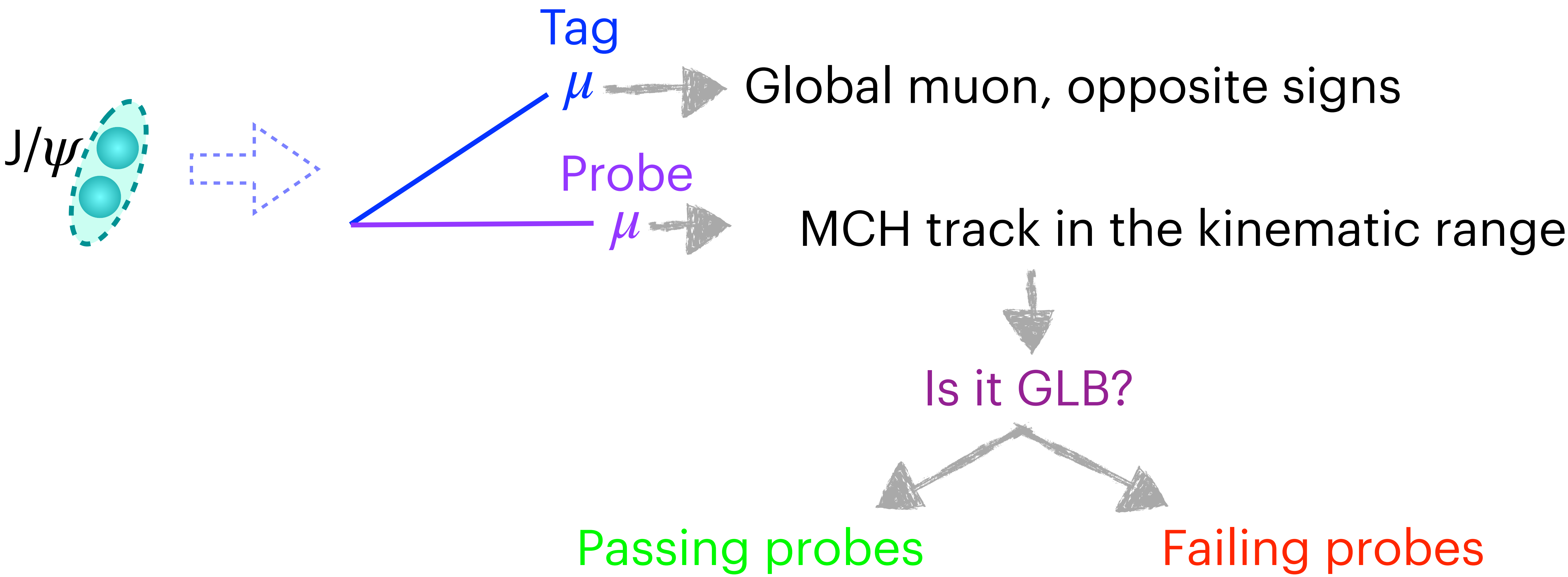
Failing probes



$$\varepsilon = \frac{\text{probes passing condition}}{\text{all probes}}$$

The starting point

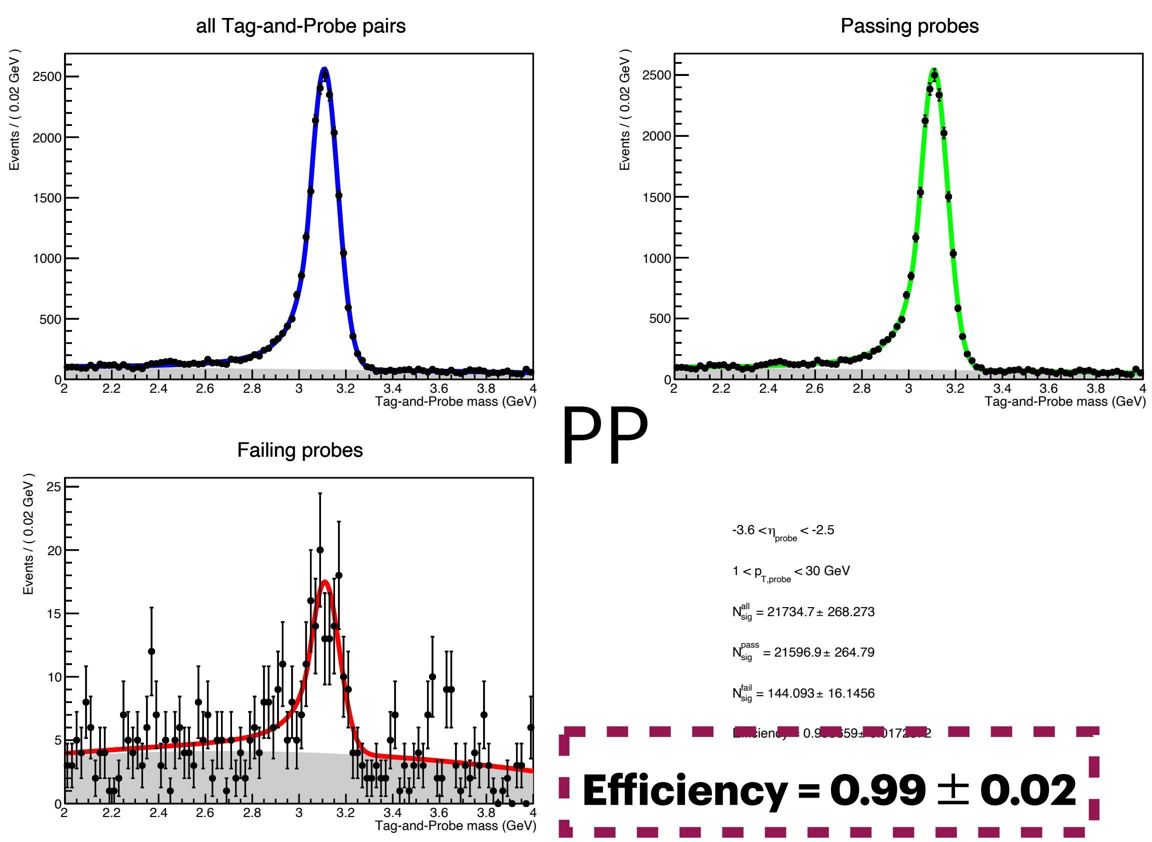
- For now we only have MC simulation
- Using the nonprompt J/ψ simulations in pp (LHC21i3g) and PbPb (LHC22b2a)

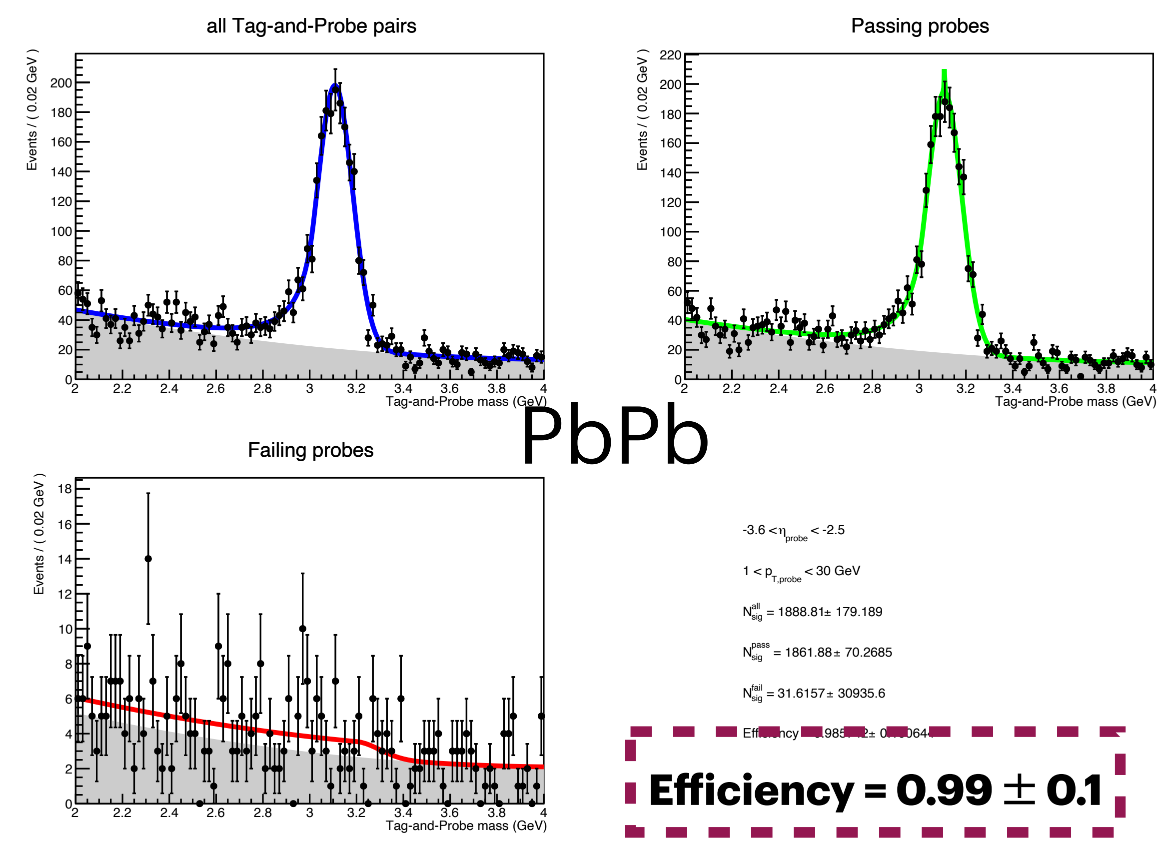


- All kinematic quantities (p_T, η, M…) are taken from the MCH tracks
- No additional criteria are applied on the quality of the tracks (e.g χ^2)
- In PbPb there is no centrality spectrum correction. Centrality is flat here which does not realistically represent data

Tag-and-probe fits

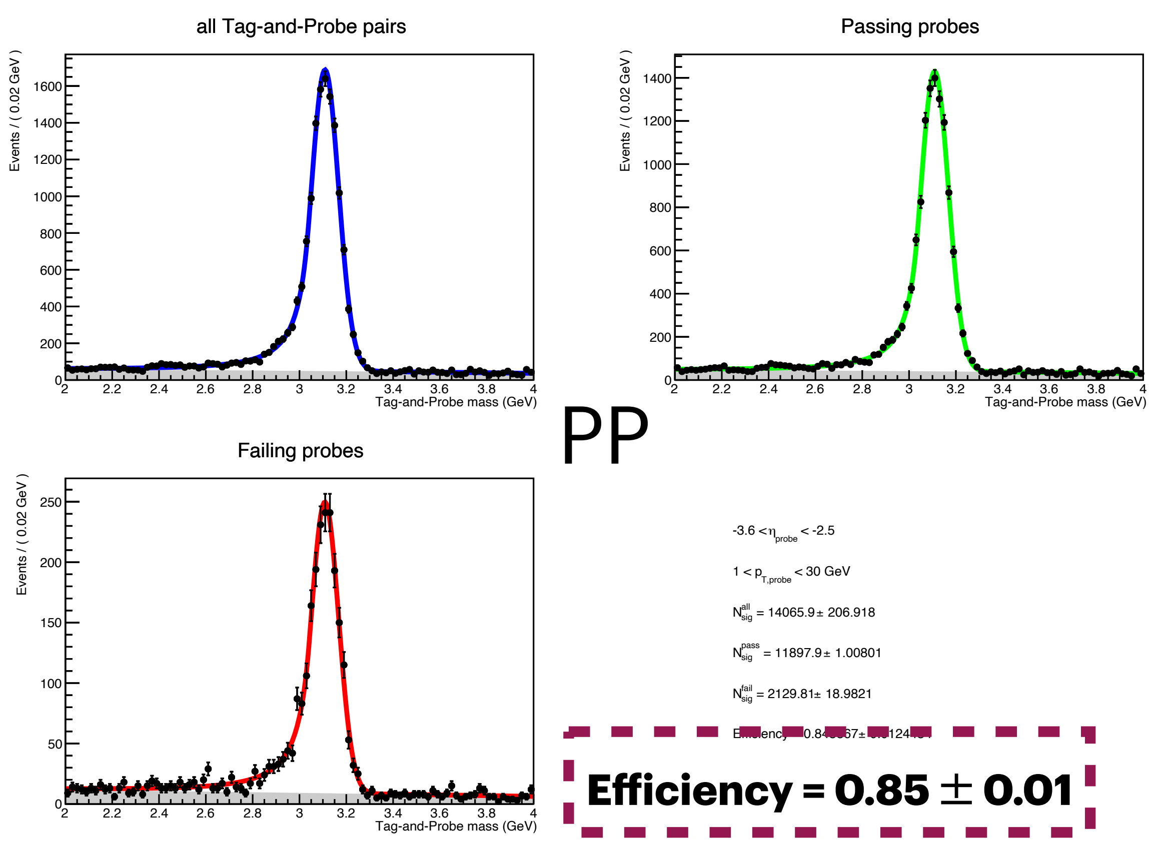
- Fits on the invariant mass of the T&P pairs, done with the rooFit package
- The fits are done for three categories: all probes, passing probes and failing probes but only all and passing fits are taken into account in the efficiency calculations
- For the signal: two Crystal Ball functions and for the background: 1st order Chebychev polynomial
- The fits definitely need more work but they give a good estimate for now
- An unrealistic efficiency of 0.99 for both pp and PbPb





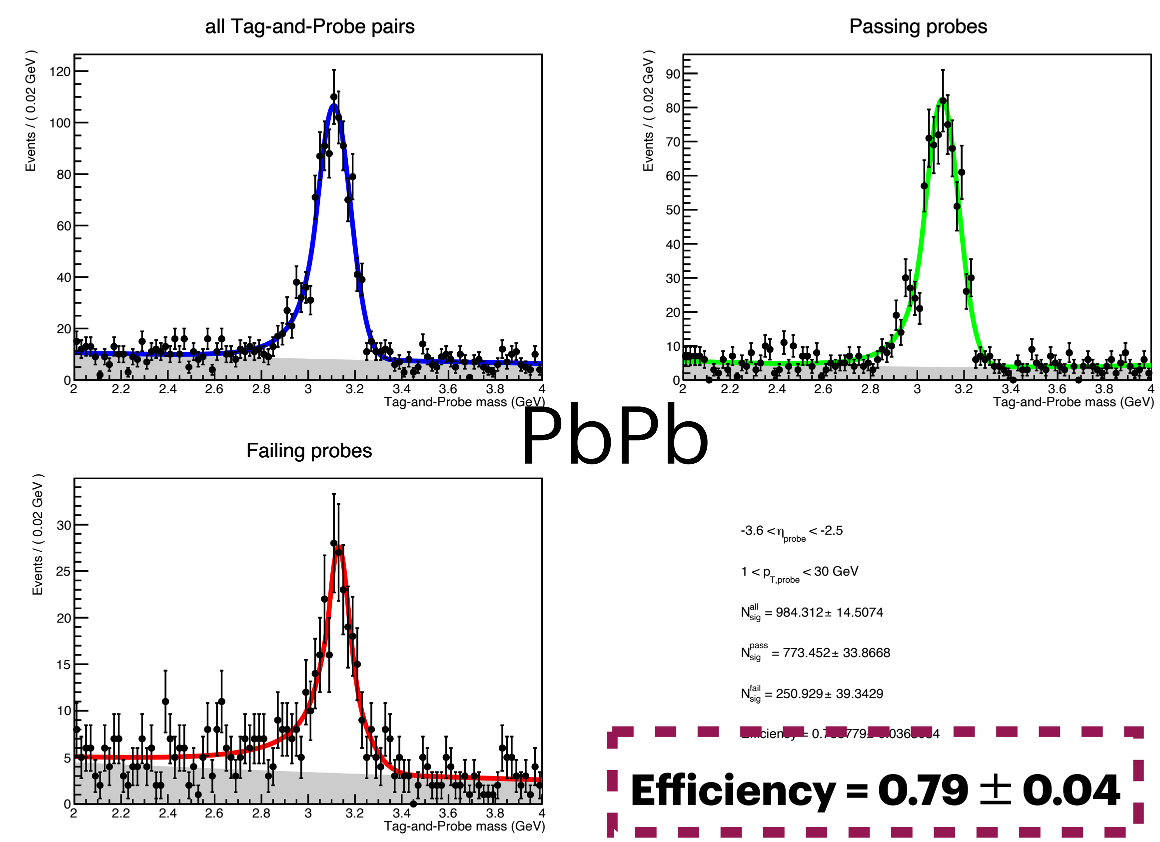
Tag-and-probe fits

- It is not enough for the MCH to have a match in the MFT. It needs to be the right match
- In MC this information is available (🛑 will not be available in data)
- In these fits the match is required to be the correct one for both the tag and the passing probes
- A decrease in the efficiency from 0.99 to 0.85 in pp and 0.79 in PbPb (would even be lower in PbPb when the centrality distribution is corrected)
- The main challenge is not the inefficiency itself but the mismatching



PP

Efficiency = 0.85 ± 0.01

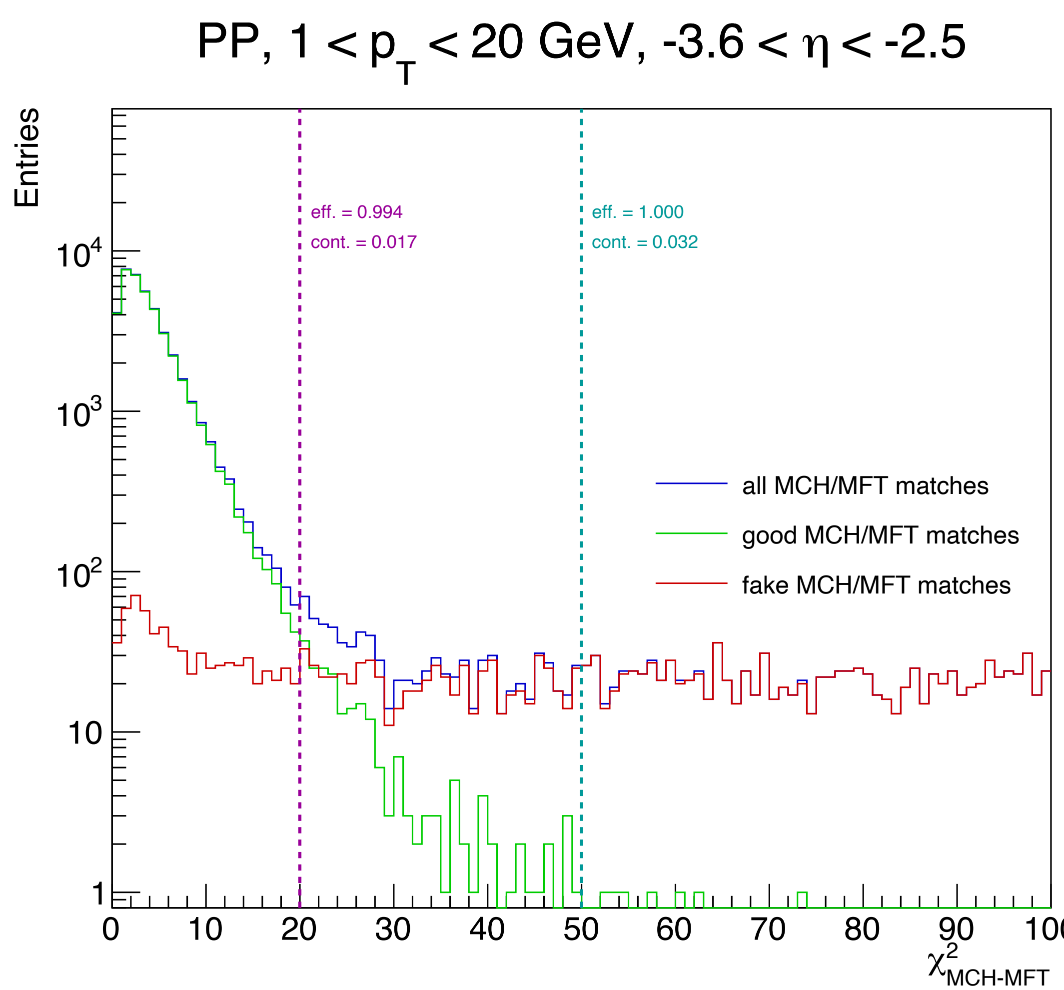


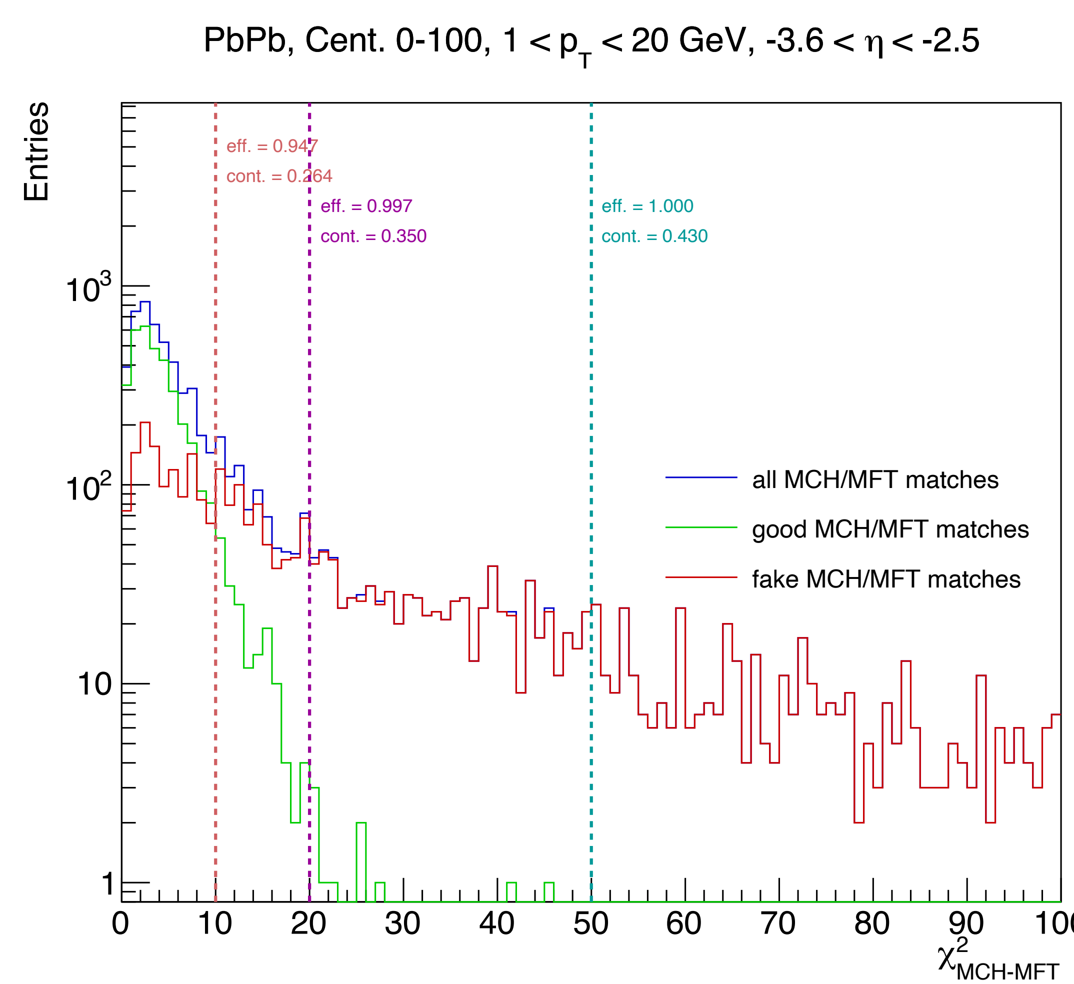
PbPb

Efficiency = 0.79 ± 0.04

Challenges for fake match rejection

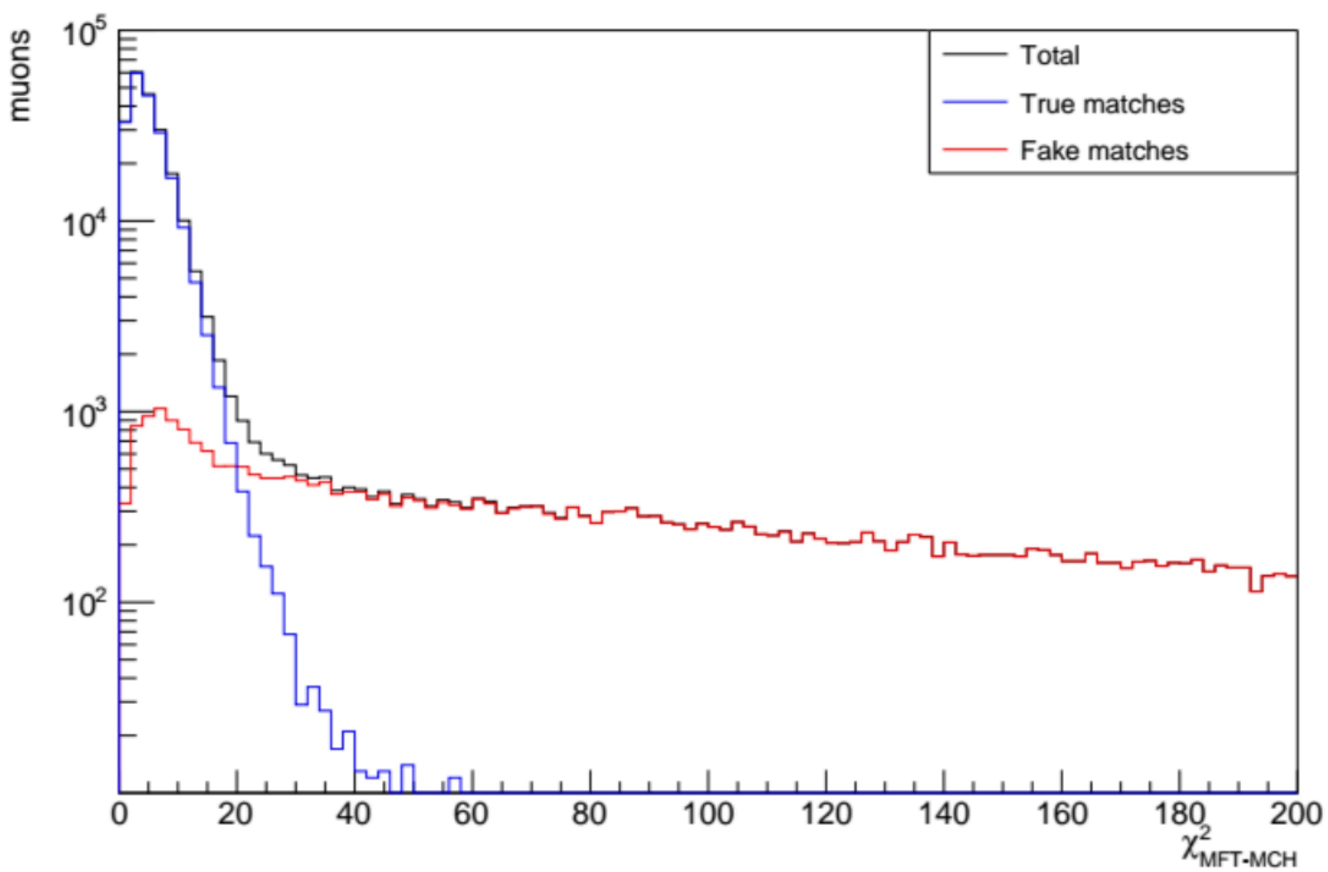
- Need a discriminator that exist in data as well as MC
- Let's start by looking at the χ^2 of the MCH/MFT match in MC for good and bad matches and look for a pattern
- In PP: big peak of good matches at low χ^2, almost flat distribution of bad matches
- Harder to separate in PbPb
- With a simple cut on the χ^2 we need to know the inefficiency and the contamination
- Not a good solution especially in PbPb
- Machine learning will be used to reject fake matches



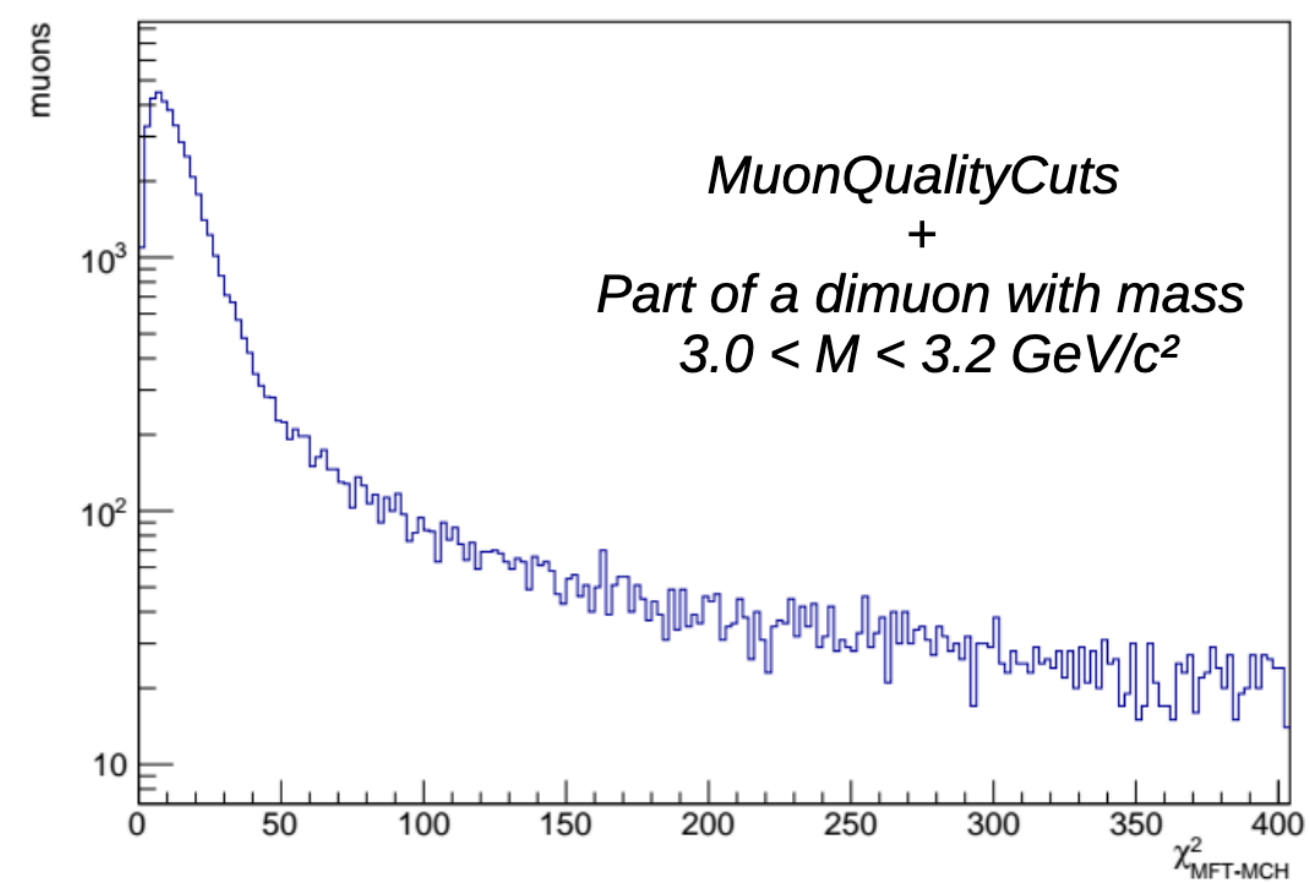


χ^2 matching for J/ψ analyses

- In the J/ψ mass window, the χ^2 distribution starts resembling the MC shape: peak + background
- Different widths of the distributions: might be closer with a better MC
- Two tasks: chose a cut on χ^2 and estimate the efficiency + contamination



Distribution in LHC21i3d2 (MC)

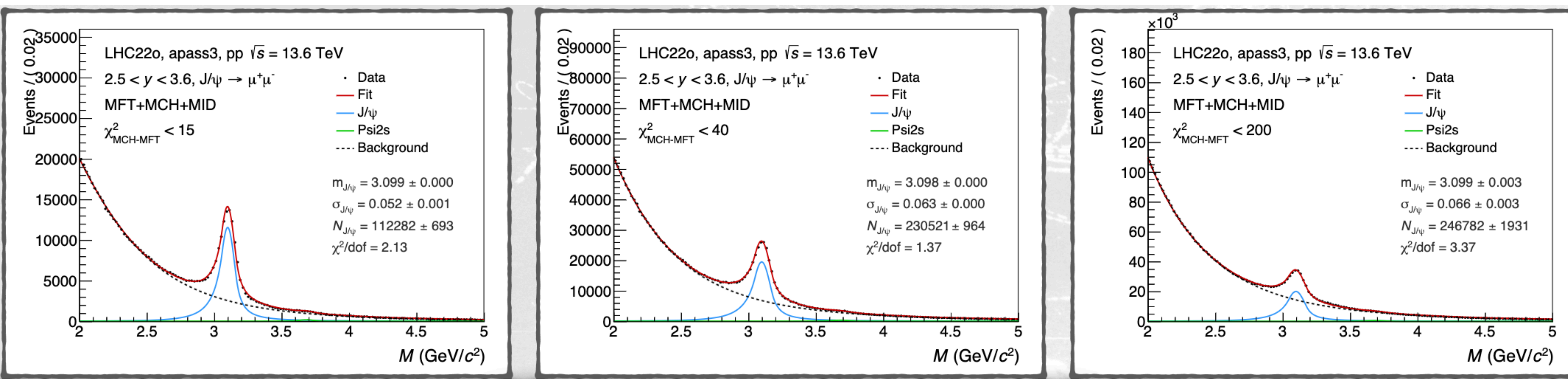


Distribution in LHC22o_pass2 (Data)

By M. Coquet

Using the peak to chose a χ^2 cut

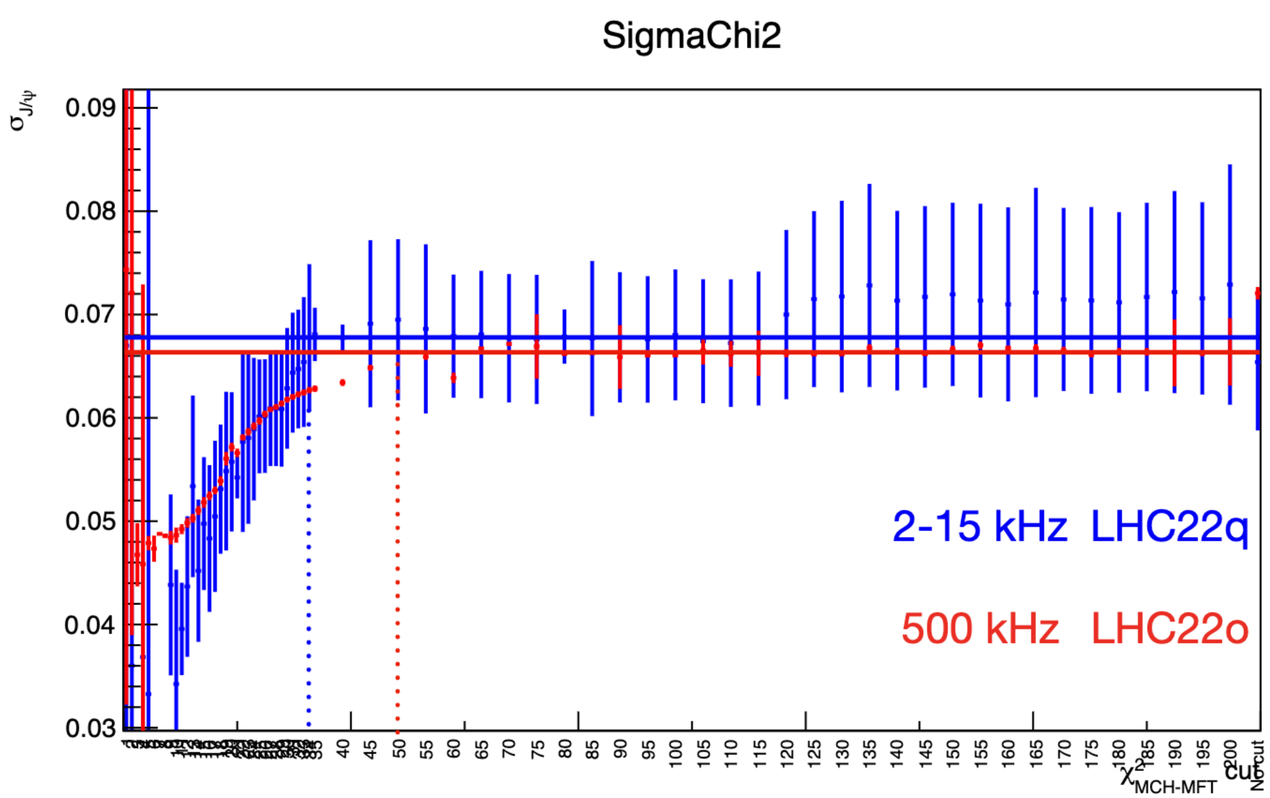
- We use fits on the global dimuon invariant mass using different χ^2 cuts and compare the results
- These fits are done in apass3 but we check for each apass
- An overall improvement was actually observed for apass3 when compared to apass2
- We also check different run conditions: the IR has an impact on the performance

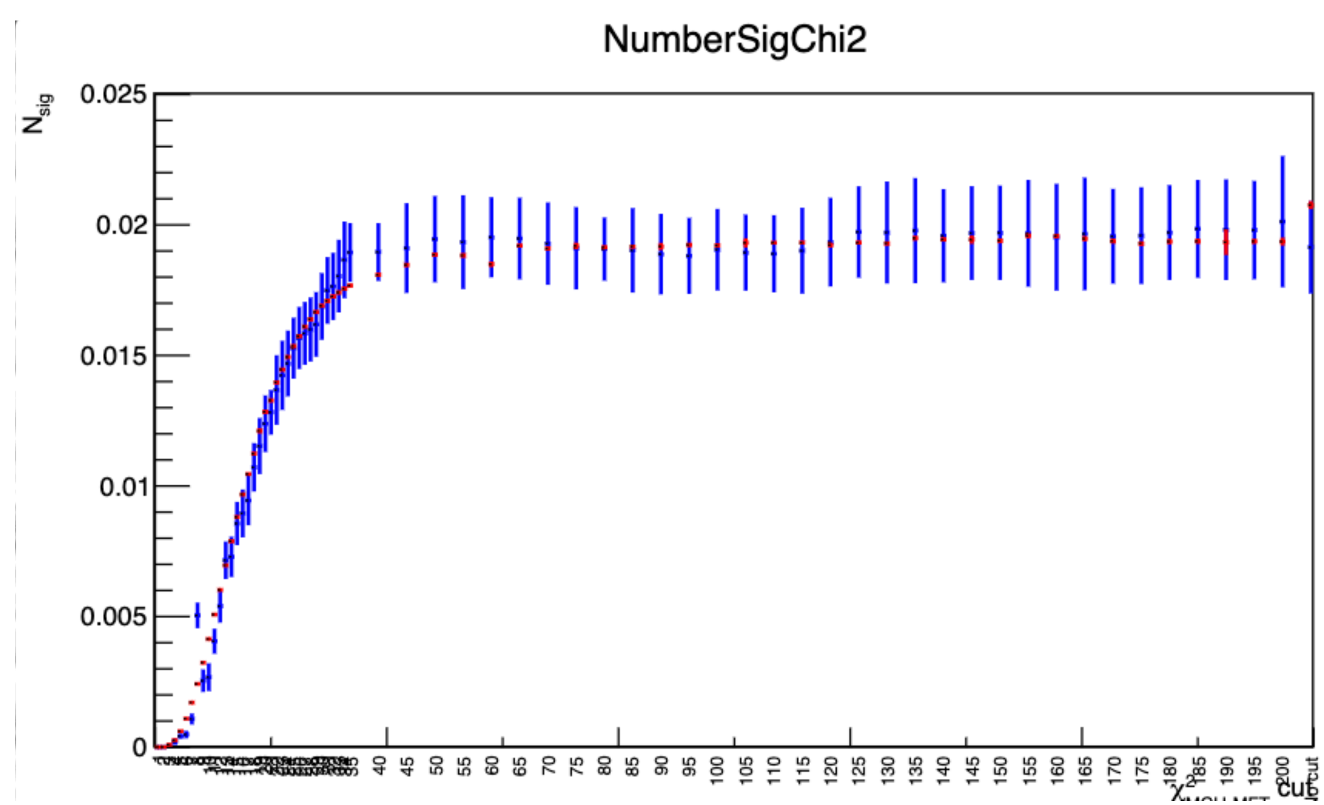


By N. Bizé

Using the peak to chose a χ^2 cut

- The resolution and the signal reach a plateau after a certain cut
- For both cases LHC22q (low IR) reaches the plateau faster than LHC22o (high IR)
  - 22q: $\chi^2 < 35$
  - 22o: $\chi^2 < 50$
- We still need to estimate the efficiency and the leftover contamination

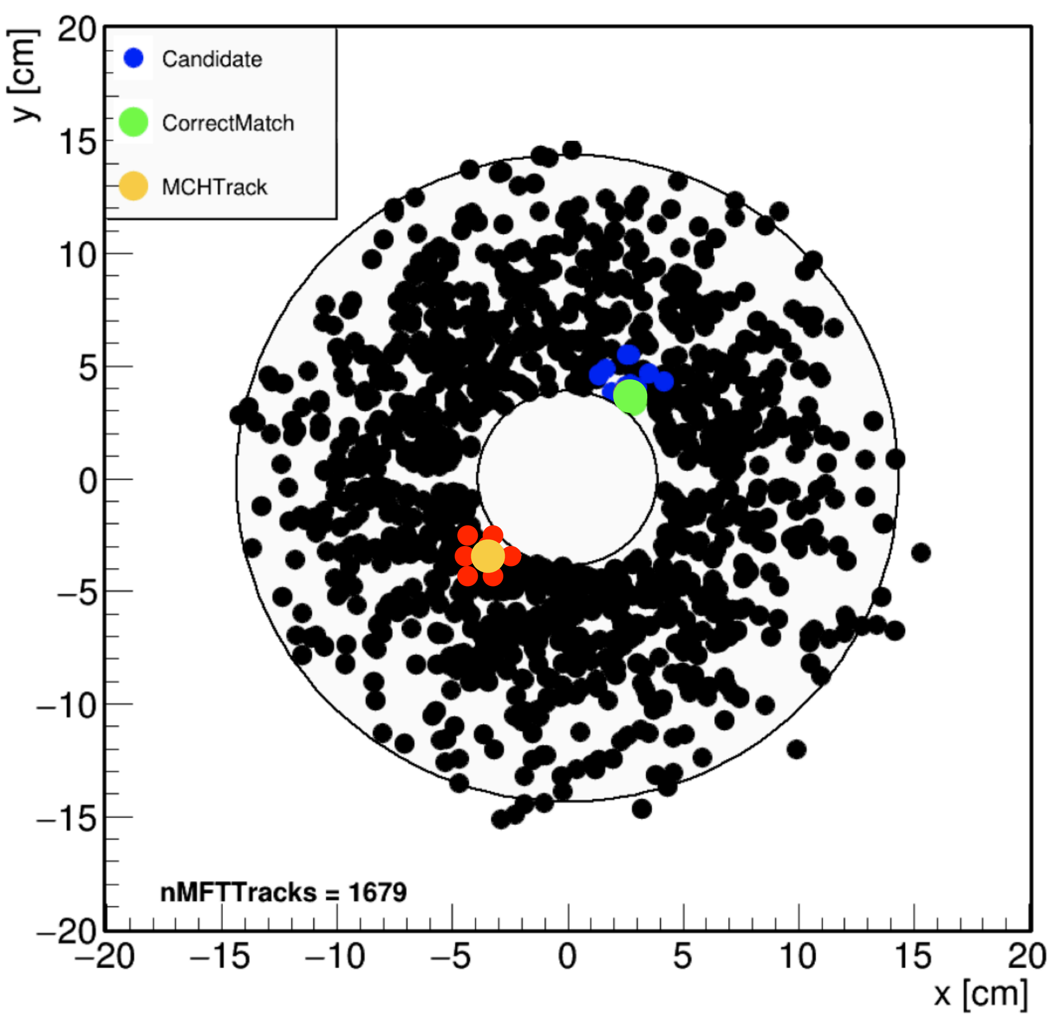


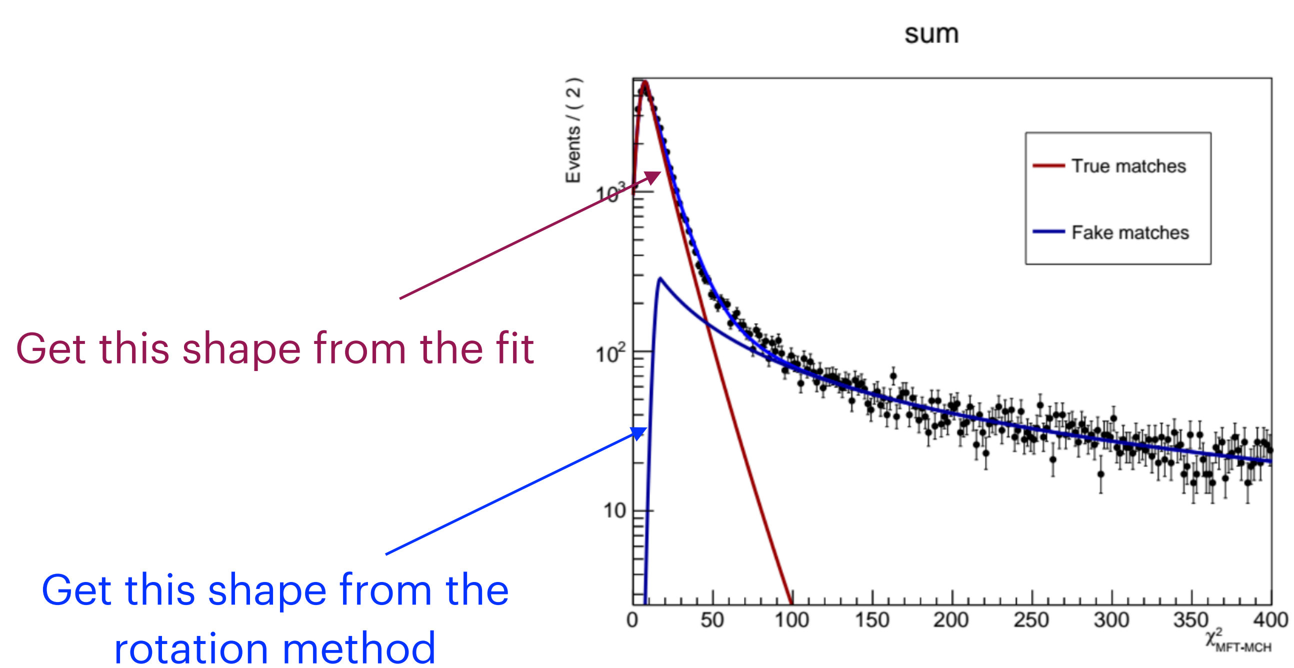


By N. Bizé

New idea to get the efficiency

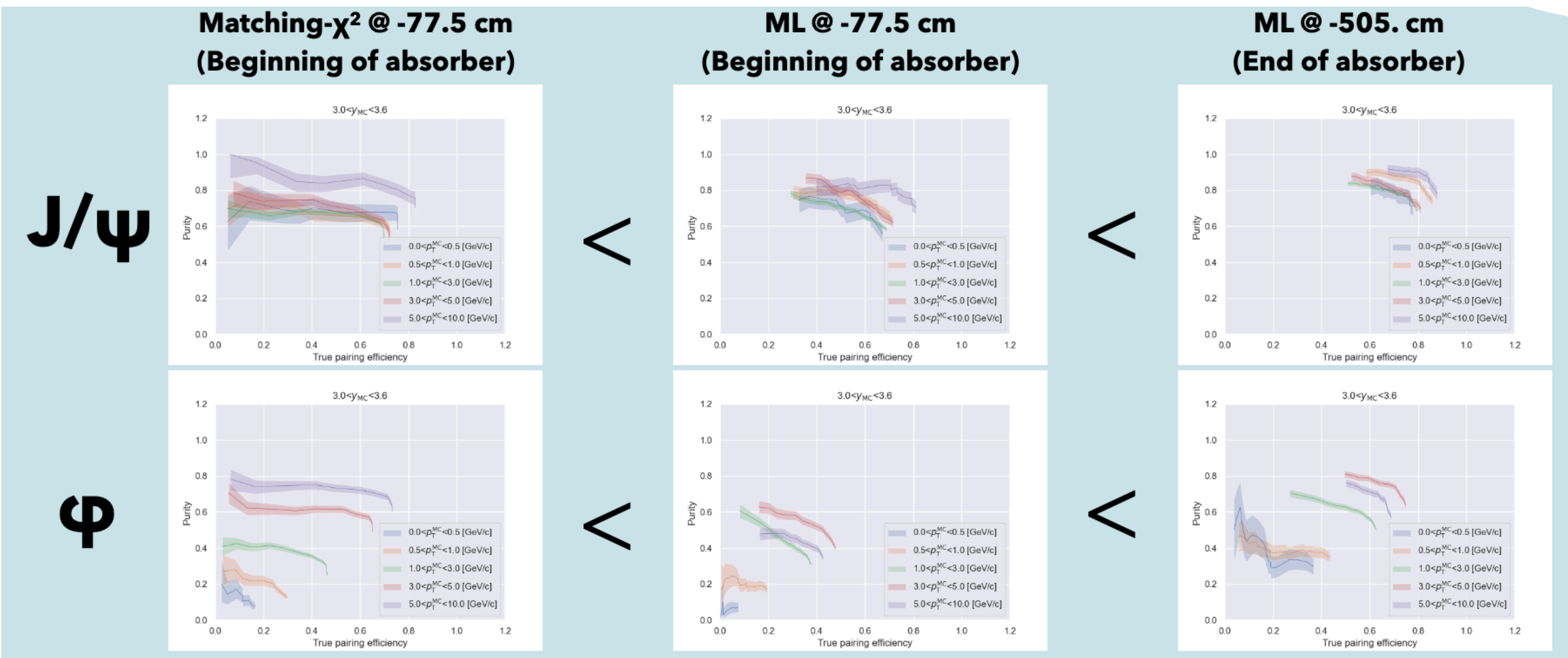
- A similar strategy can be used for the MFT-MCH mismatch shape
- Take the MCH and MFT candidates in our data, and rotate the MCH track (a ϕ rotation should be good) and apply the exact matching procedure as normal
- Since all the matches are going to be fake, this will give us the shape of the fake matches
- From this shape we build a model (fit or template) of the fake matches and then use the total distribution in data to fit and get the good matches as well
- This will gives the efficiency and contamination of our matching
- Can also be used in PbPb for the Machine Learning training data





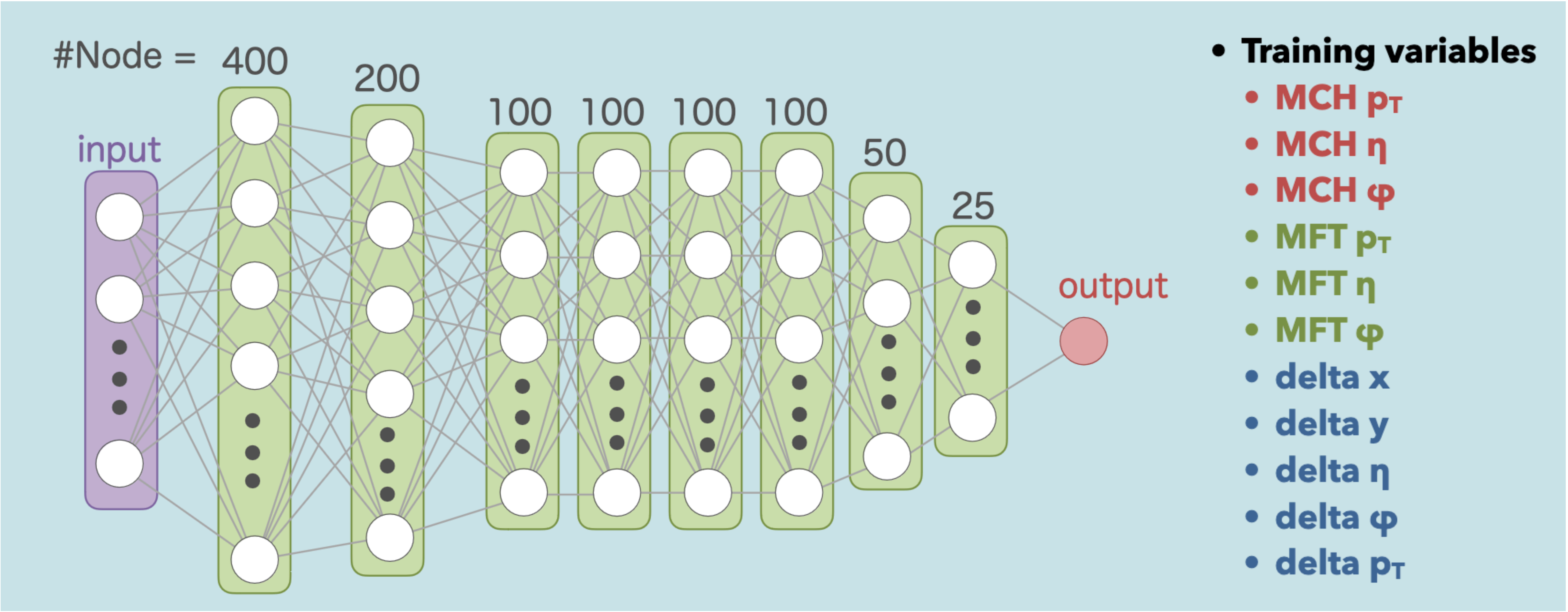
Using ML for MFT-MCH matching

- The χ^2 performance is not good enough for low mass particles like ϕ
- We explored other options like using machine learning to improve the matching
- Important for low mass and can improve the J/ψ matching
- Different options are explored using AO2D



By S. Yano and M. Oya

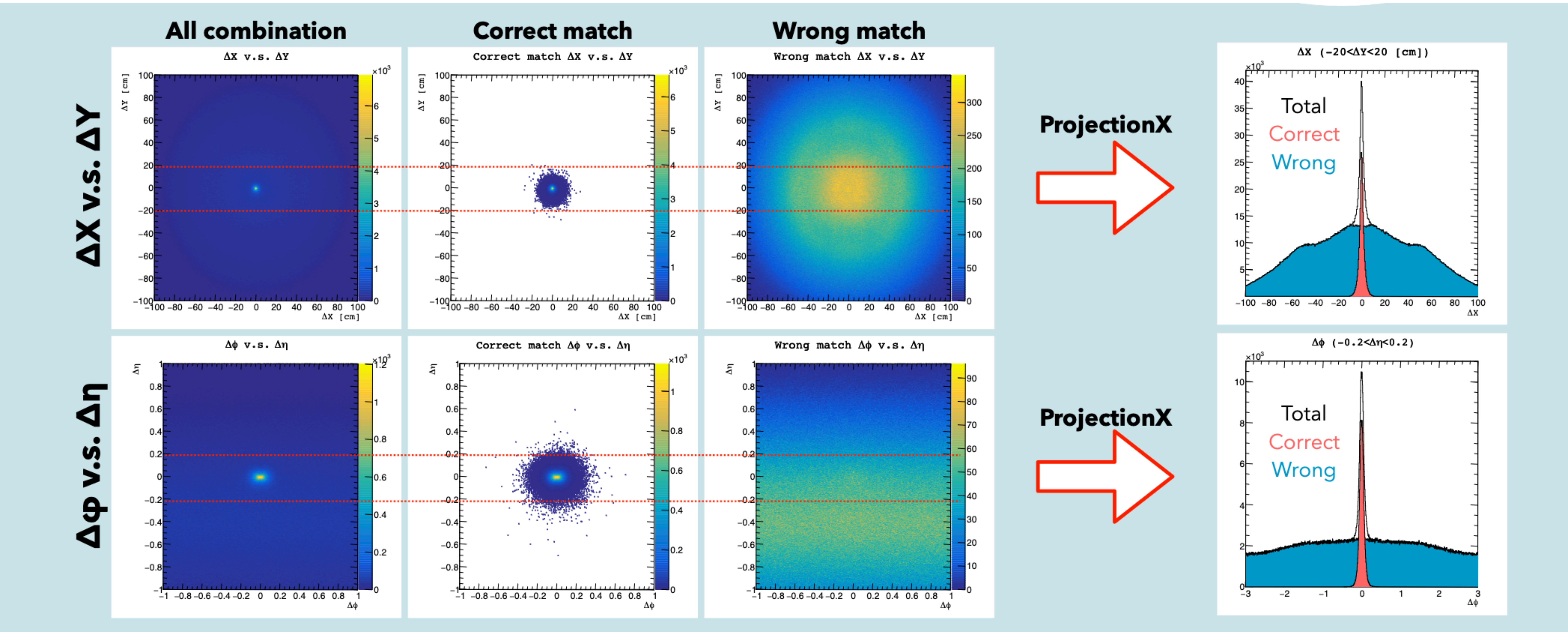
Deep Neural Network model



By S. Yano and M. Oya

Matching plane choice

- The preliminary results show a good performance in MC
- The Δx, Δy and $\Delta\phi$ distributions show a peak of correct matches with a background of fake matches
- A similar shape is also seen in data but more tests need to be done
- Similar to χ^2, we need to find a data driven method to estimate the leftover background: opposite sign matching



Workflow for ML matching

Purpose

MFT-MCH matching by machine learning based on the AO2D.root adaptable to the DQ framework

New workflow: o2-analysis-mftmchmatchingml

- Match mfttrack and fwdtrack (MuonStandaloneTrack) by machine learning
- Create new table “FwdTracksML”

New process function in table-maker: processMuonMLOnly

- Use FwdTracksML instead of FwdTracks to create reducedAOD

o2-analysis-mftmchmatchingml

1. Select mfttrack and fwdtrack (MuonStandaloneTrack) from AO2D
2. Download onnx file from CCDB and get matching score for the selected track pair
3. Fill a new table named "FwdTracksML" if the track pair is true
   - The structure of table is same as FwdTracks
   - All tracks have TrackType=0 (GlobalMuonTrack)
   - Momentum is recalculated

- $p_x = p_{MCH} \sin\theta_{MFT} \cos\varphi_{MFT}$
- $p_y = p_{MCH} \sin\theta_{MFT} \sin\varphi_{MFT}$
- $p_z = p_{MCH} \cos\varphi_{MFT}$

Configurables

```
Configurable<std::string> cfgCCDBURL{"ccdb-url", "http://ccdb-test.cern.ch:8080", "URL of the CCDB repository"};
Configurable<std::string> cfgModelDir{"ccdb-path", "Users/m/mooya/models", "base path to the ONNX models"};
Configurable<std::string> cfgModelName{"ccdb-file", "model_LHC22o.onnx", "name of ONNX model file"};
Configurable<float> cfgThrScore{"threshold-score", 0.5, "Threshold value for matching score"};
Configurable<int> cfgColWindow{"collision-window", 1, "Search window (collision ID) for MFT track"};
Configurable<float> cfgXYWindow{"XY-window", 3, "Search window (delta XY) for MFT track"};
```

New process function in table-maker

```
void processMuonMLOnly(MyEvents::iterator const& collision, aod::BCsWithTimestamps const& bcs,
                       soa::Filtered<aod::FwdTracksML> const& tracksMuon)
{
  fullSkimming<gkEventFillMap, 0u, gkMuonFillMap>(collision, bcs, nullptr, tracksMuon, nullptr, nullptr);
}
```

- fullSkimming use FwdTracksML instead of FwdTracks
- After reduced AOD is created, usage is the same as the case using FwdTracks

---

- The effectiveness of the ML tool has not been verified with real data yet.
- The DQ software team has pointed out that this strategy needs improvement because it completely replaces the existing fwdtrack, resulting in the loss of relevant information.

Summary

- The Tag-and-Probe method will be used for the first time for muons in ALICE
- The focus will be the matching between the muon chambers and muon forward tracker
- The work started but still very preliminary
- Many challenges will be faced but they don't only concern Tag-and-Probe: MCH-MFT matching
- pp options: χ^2 matching and ML matching for low mass
- PbPb options: only ML
- Fake match rate estimation: track rotation and opposite sign matching

Thank you