

Table of Contents

Hit-by-hit matching	1
Goal.....	1
Approach.....	1
Dissection of the muon MCTruth software.....	1
Definitions:.....	1
Methods for hit counting.....	1
Key steps:.....	2
Problems and open questions:.....	3
Results.....	3
Summary&Conclusions.....	3

Hit-by-hit matching

Goal

- find the best purity/efficiency/quality cut that gives the correct/stable MC correction for muon reconstruction.

Approach

- CMSSW_3_7_0 the working version: [RelVal_official](#)
- generate several samples with
 - ◆ simple Z/Jpsi/upsilon->mumu data samples:

```
/castor/cern.ch/user/m/mironov/cmssw370/digireco/root/
```
 - ◆ different combinations of the cuts for the Purity (which can be done only in the configuration level, as in the map is stored the quality of the amthc, which is done fct simHits) and other switches in the configuration of the the muon matcher
 - ◆ for globalMuons, standAloneMuon:UpdatedAtVertex, and generalTracks
 - ◆ using hi-tracking and pp-tracking for the silicon part
- look how pT resolution, efficiency vary
- calculate correction (1/Eff), correct the reconstructed spectra, and compare then with the input, initial spectra. (if the correction is correct, the corrected reco spectra and the sim spectra should coincide)

Dissection of the muon MCTruth [software](#)

Definitions:

- purity: $x_{\text{purity}} = x_{\text{shared}}/n_{x_{\text{selected_reco_hits}}}$ (x=STA, GLB, TRK) , where the **selected** are the **valid** of the recoTrack
- efficiency/matching_quality(written in the sim2reco map): $x_{\text{quality}} = x_{\text{shared}}/n_{x_{\text{selected_sim_hits}}}$, where **selected** are the **recounted sim hits**, subdetector, by subdetector
- Hit types :
 - ◆ valid = valid hit
 - ◆ missing = detector is good, but no rec hit found
 - ◆ inactive = detector is off, so there was no hope
 - ◆ bad = there were many bad strips within the ellipse (in Tracker)
 - ◆ = hit is compatible with the trajectory, but chi2 is too large (in Muon System)

Methods for hit counting

- sim hits (same as in the matching software) [Hide settings](#)

```
for(std::vector<PSimHit>::const_iterator simHit = simTrack.pSimHit_begin();
    simHit!= simTrack.pSimHit_end(); simHit++)
{
    const DetId detId = DetId(simHit->detUnitId());
    DetId::Detector detector = detId.det();
    int subdetId = static_cast<int>(detId.subdetId());

    if(detector == DetId::Tracker && (subdetId==PixelSubdetector::PixelBarrel || subdetId

    if (detector == DetId::Tracker &&
        (subdetId==SiStripDetId::TIB||subdetId==SiStripDetId::TOB||
         subdetId==SiStripDetId::TID||subdetId==SiStripDetId::TEC) ) simstriphits++;
```

HbyHm < Main < TWiki

```
if(detector == DetId::Muon && subdetId == MuonSubdetId::DT) simdthits++;
if(detector == DetId::Muon && subdetId == MuonSubdetId::CSC) simcschits++;
if(detector == DetId::Muon && subdetId == MuonSubdetId::RPC) simrpchits++;
}
```

• reco hits [Hide settings](#)

```
const reco::HitPattern& hp = trkRef.get()->hitPattern();

result.push_back(hp.numberOfHits());
result.push_back(hp.numberOfValidHits());

result.push_back(hp.numberOfValidTrackerHits());
result.push_back(hp.numberOfValidPixelHits());
result.push_back(hp.numberOfValidStripHits());

result.push_back(hp.numberOfValidMuonHits());
result.push_back(hp.numberOfValidMuonDTHits());
result.push_back(hp.numberOfValidMuonCSCHits());
result.push_back(hp.numberOfValidMuonRPCHits());

result.push_back(hp.numberOfLostHits());

result.push_back(hp.numberOfLostTrackerHits());
result.push_back(hp.numberOfLostPixelHits());
result.push_back(hp.numberOfLostStripHits());

result.push_back(hp.numberOfLostMuonHits());
result.push_back(hp.numberOfLostMuonDTHits());
result.push_back(hp.numberOfLostMuonCSCHits());
result.push_back(hp.numberOfLostMuonRPCHits());

result.push_back(hp.numberOfBadHits());
result.push_back(hp.numberOfBadMuonHits());
result.push_back(hp.numberOfBadMuonDTHits());
result.push_back(hp.numberOfBadMuonCSCHits());
result.push_back(hp.numberOfBadMuonRPCHits());
```

Key steps:

- getMatchedIds:
 - ◆ Input: the XHitAssociators, all hits of a recoTrack
 - ◆ Output: valid and invalid, matched (and all) hits from: tracker, muon_total, rpc, dt,csc.
 - ◆ How:
 - ◇ loop over all reco hits and:
 - 1. count/get hits in each subdetector X (tracker, muon pieces)
 - 2. for each hit, call the corresponding XHitAssociator: valid, valid_matched, invalid, invalid_matched
- getShared:
 - ◆ Input: map of matched Ids (indexed over the rechits of the reco::Track, no double-counting allowed), and TrackingParticleCollection
 - ◆ Output: number of hits that are shared/matched between recoTrack and simTrack
 - ◆ How:
 - ◇ 1. loop over the map, and for each recoHit, get the vector of associated simhits
 - a. for each simhit, get the trackId to which it belongs and the EncodedEventId
 - b. loop over all TrackingParticles, and get the simTrack that the hit belongs to (based on trackId and EncodedEventId)
 - ◇ 2. count all the matches/shares

- Matching decision
 - ◆ sim2reco (x_quality_cut = Efficiency_cut in the config file)
 - ◇ 1) X=STA, TRK: X_quality <= X_quality_cut -> XOk=false (1st criteria: the stub has to pass the quality/efficiency_cut)
 - ◇ 2) X_purity <= PurityCut_X (set in the configuration file) XOk = false;
 - ◇ 3) GLB=OK if TRK_OK && STA_OK (2st criteria, both stubs, have to pass the purity cut)
 - ◇ 4) add to the sim2Reco collection, with the (outputCollection[tpindex].push_back(IndexMatch(tindex,global_quality)));) **global_quality** the sorting criteria

Problems and open questions:

- SimHits (too many-pixel, RPC, or outside the detector coverage (DT))
 - ◆ Understood (Thanks to Phillip for the idea): the subDetector id-s, are just some enumerations <-> numbers ==> can have double counting
 - ◇ pixel [↗](#)
 - ◇ strips [↗](#): 3,4,5,6
 - ◇ DT,CSC,RPC [↗](#): 1,2,3

Hence, if you don't check also on the detector part (f(detector == DetId::Tracker && (subdetId==PixelSubdetector::PixelBarrel || subdetId==PixelSubdetector::PixelEndcap)) simpixelhits++), to the pixel count you'll add the dt's and rpc's hits (same subDetId, 1 and 2)! Duh!

This is done correctly in the matching code, hence all the efficiency, resolutions, corrections, etc plots, still stand. The corrected plots for my loop, are in the Summary slide.

- number of RecHits (validHits) for CSC : 3\% cases in which the total number of CSC valid hits on the track is >24 (maximum possible)
 - ◆ from back-2-back in phi muons, with pT ~=: csc.pdf

Results

- Definitions (mine):
 - ◆ purity_rate = reco_matched/reco_all
 - ◆ efficiency = sim_matched/sim_all
 - ◆ correction_factor = purity_rate/efficiency
- doReco.pdf: looking at reco2Sim collection
- doSimMatchedRelativeHitsContrib.pdf: relative hits contributions to the full track
- doSimRecoHits_glb.pdf: GLB- hits cuts 2D histos
- doSimRecoHits_sta.pdf: STA: hits, 2D histos
- doSimRecoHits_trk.pdf: TRK: hits, 2D histos
- doPerformance.pdf: efficiency, resolutions, corrections, cuts sistematics -- this stands

Summary&Conclusions

- matching decision based only on x_purity and fracHitMatch>0/0.5 and <0.75 are giving the right results
- in HI, the fake rate will be higher--> in case one wants to present it separately, should chose a working point (purity,fracHitMatch) for which this rate is small
- (purity_X=75%, and quality>0.5) suggested working point

mport SimMuon.MCTruth.MuonAssociatorByHits_cfi

HbyHm < Main < TWiki

```
process.glbMuonAssociatorByHits = SimMuon.MCTruth.MuonAssociatorByHits_cfi.muonAssociatorByHits.c
process.glbMuonAssociatorByHits.tracksTag = 'globalMuons'
process.glbMuonAssociatorByHits.UseTracker = True
process.glbMuonAssociatorByHits.UseMuon = True
process.glbMuonAssociatorByHits.PurityCut_track = 0.75
process.glbMuonAssociatorByHits.EfficiencyCut_track = 0.
process.glbMuonAssociatorByHits.PurityCut_muon = 0.75
process.glbMuonAssociatorByHits.EfficiencyCut_muon = 0.
process.glbMuonAssociatorByHits.includeZeroHitMuons = False
process.glbMuonAssociatorByHits.acceptOneStubMatchings = False
```

- Note: corrections wrt pT_{gen} : correct up to resolution effects (best seen in the sTA plots)

-- CameliaMironov - 12-Oct-2010

- [summaryHbHm.pdf](#): summary

This topic: [Main > HbyHm](#)

Topic revision: r10 - 2010-11-25 - CameliaMironov



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? [Send feedback](#)