Non-Prompt J/psi Analysis

PbPb @ 5.02 TeV

Himanshu Sharma

Mar 02, 2020

IFJ-ALICE Meetings

Activities -

- Results of Improver Task
- J/psi-cut Efficiencies & Stuff

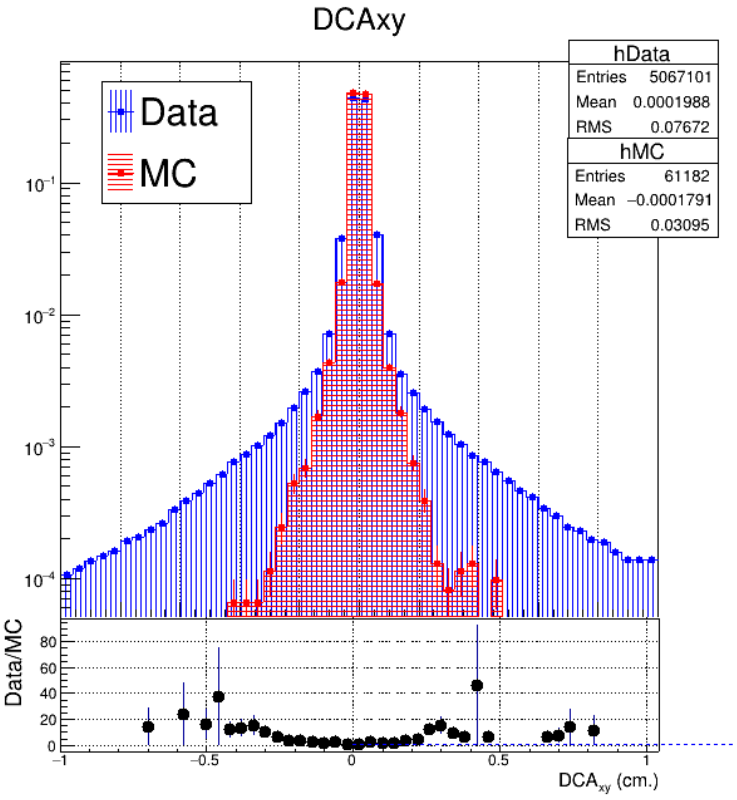
Improver Test details -

- Dataset : Full run - 297414 from LHC19f1b
- 0-10% Centrality (PbPb)
- Purpose : To match the DCAs in Data and MC

Improver Test details -

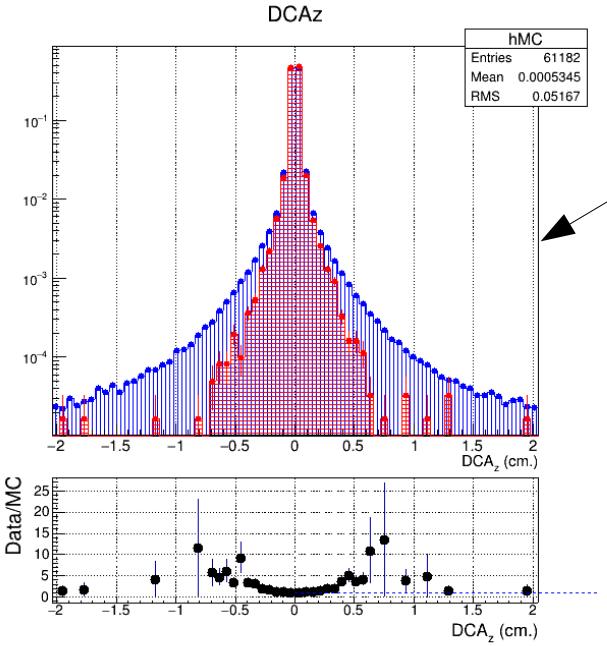
- Dataset : Full run - 297414 from LHC19f1b
- 0-10% Centrality (PbPb)
- Purpose : To match the DCAs in Data and MC

For Example : DCAxy for
(1.0 < pT < 1.5) GeV/c



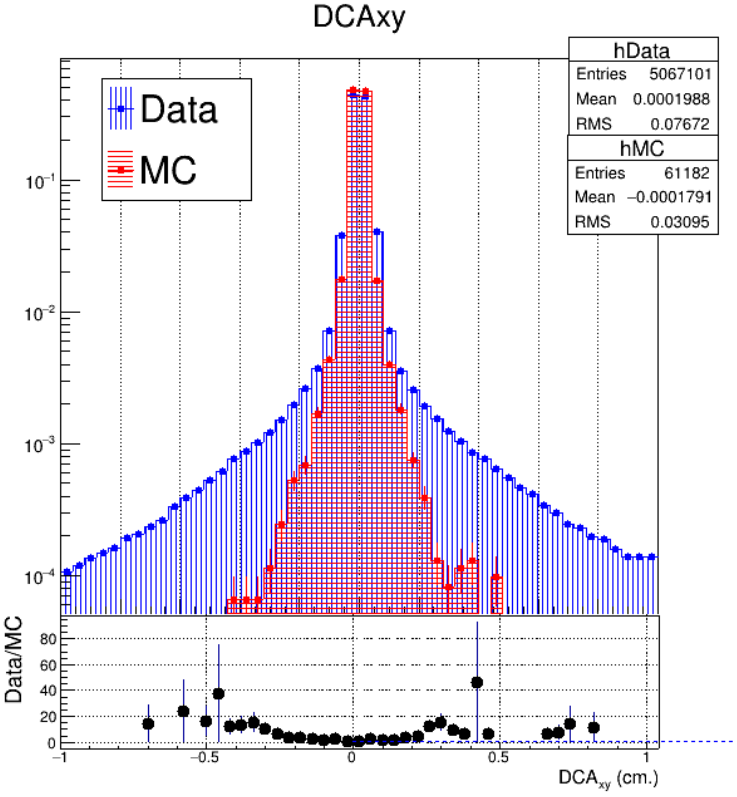
Improver Test details -

- Dataset : Full run - 297414 from LHC19f1b
- 0-10% Centrality (PbPb)
- Purpose : To match the DCAs in Data and MC



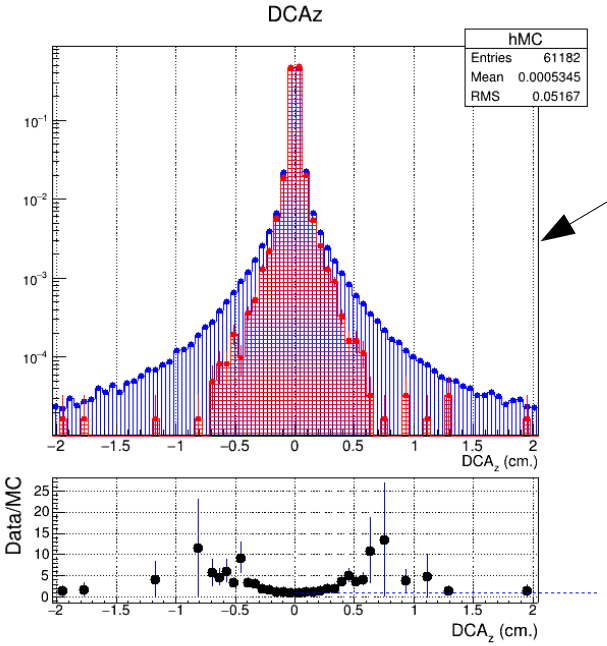
For Example : DCAz for
(1.0 < pT < 1.5) GeV/c

For Example : DCAxy for
(1.0 < pT < 1.5) GeV/c



Improver Test details -

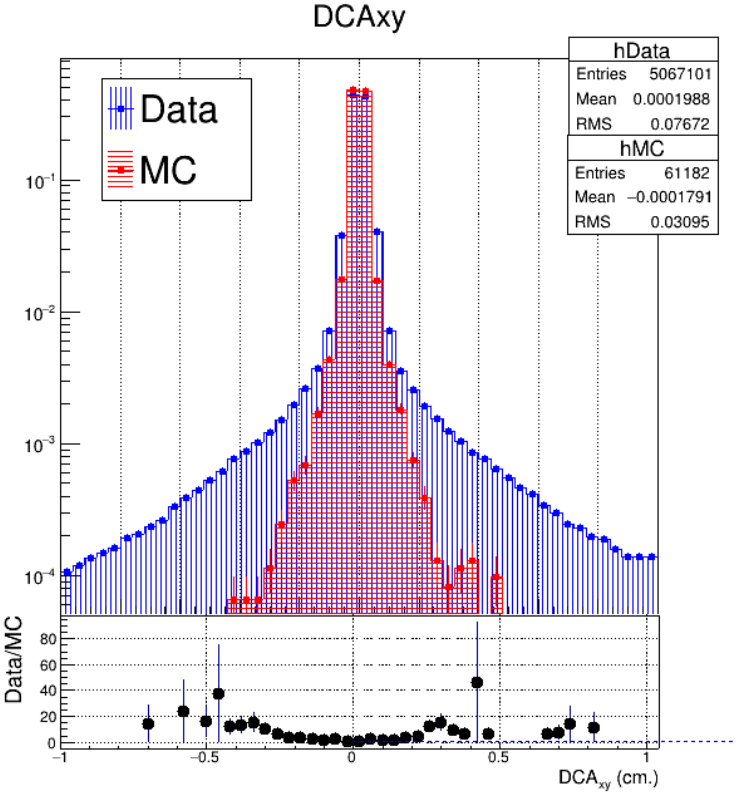
- Dataset : Full run - 297414 from LHC19f1b
- 0-10% Centrality (PbPb)
- Purpose : To match the DCAs in Data and MC



For Example : DCAz for
(1.0 < pT < 1.5) GeV/c

So what do we want ?

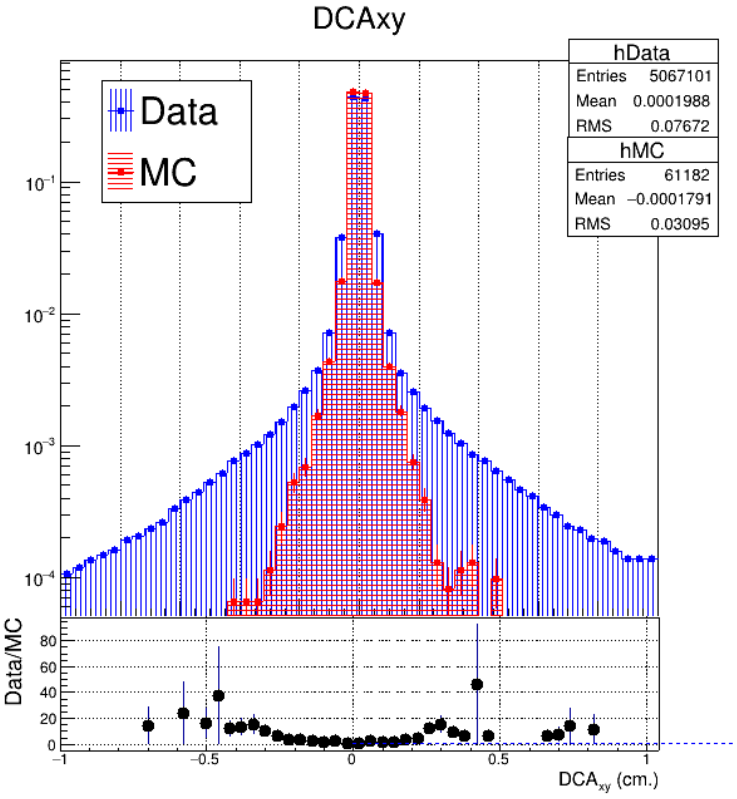
For Example : DCAxy for
(1.0 < pT < 1.5) GeV/c

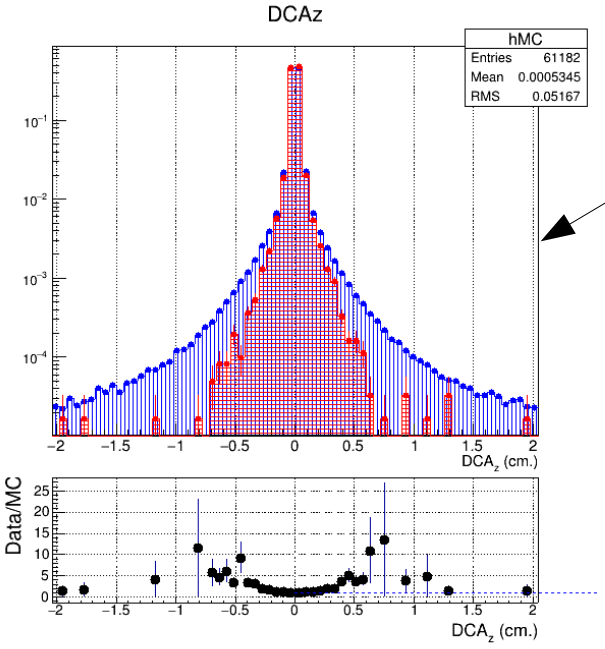


Improver Test details -

- Dataset : Full run - 297414 from LHC19f1b
- 0-10% Centrality (PbPb)
- Purpose : To match the DCAs in Data and MC

For Example : DCAxy for (1.0 < pT < 1.5) GeV/c





For Example : DCAz for (1.0 < pT < 1.5) GeV/c

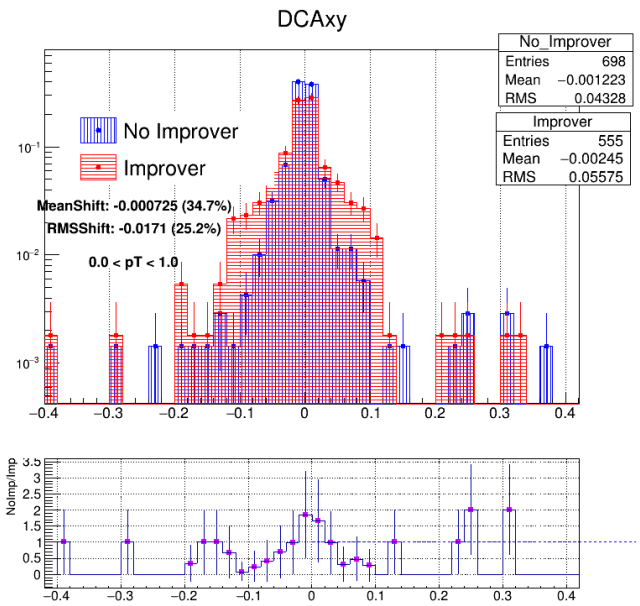
So what do we want ?

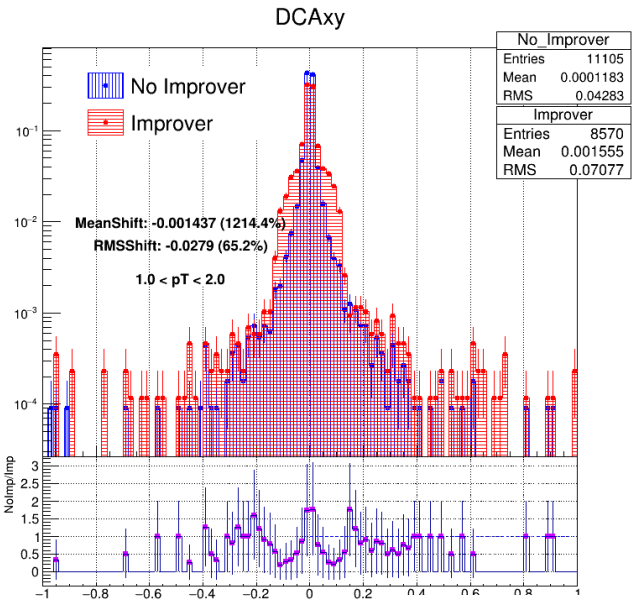
To match the data & MC distribution we tune the MC with Vertex Improver task :

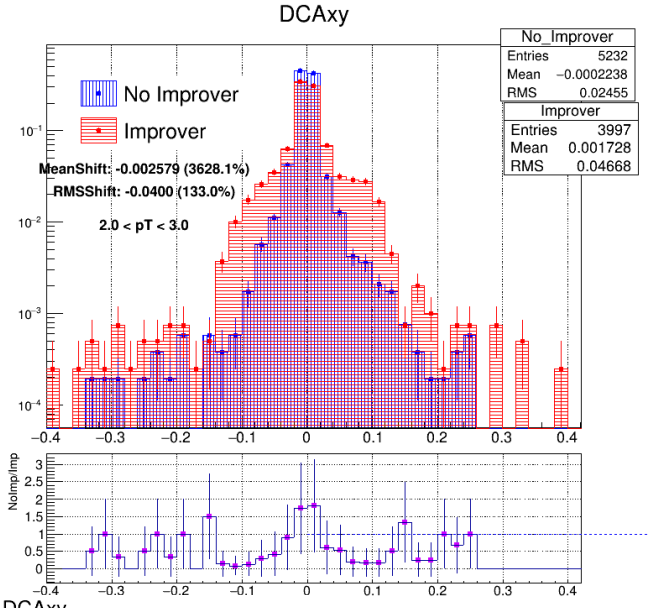
The MC-distributions shapes for DCAs should be broader to match with the data distributions.

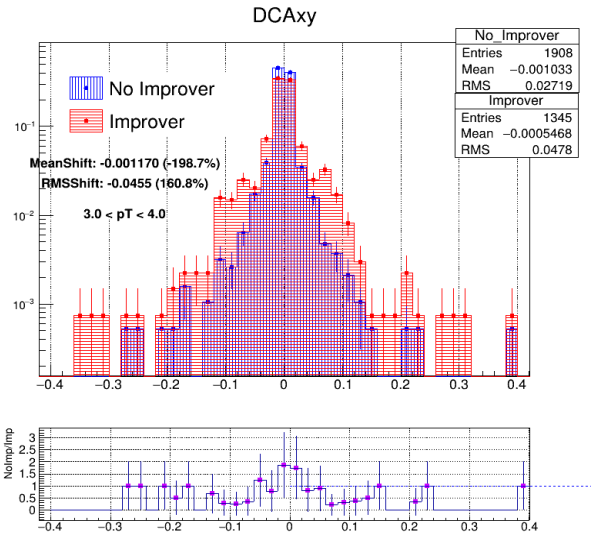
Results from Improver:

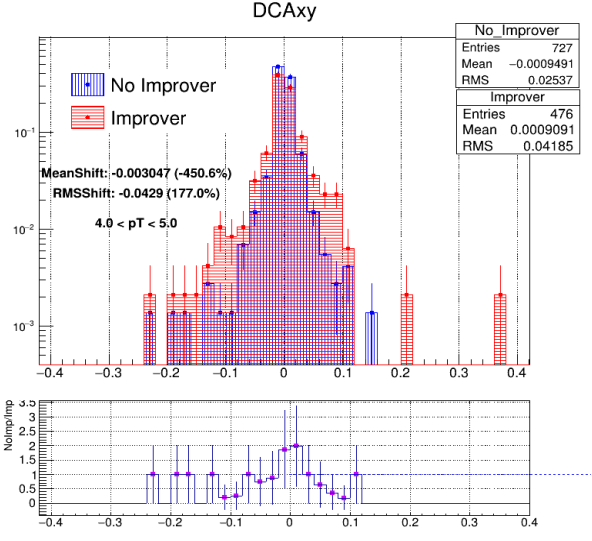
- The following results for are for all the tracks (Not only for Jpsi candidates), due to low-statistics.
- After having good stats, we will see :
    - Impact of Improver on Lxy
    - Comparision of Improved MC DCAs with the DCAs in Real-Data
- These results are for ~10K events.

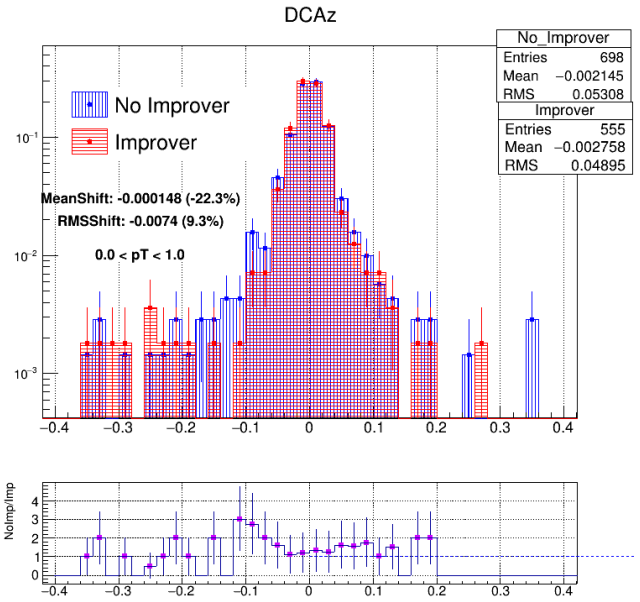

DCAxy
No_Improver
Entries 698
Mean −0.001223
RMS 0.04328
Improver
Entries 555
Mean −0.00245
RMS 0.05575
No Improver
Improver
MeanShift: -0.000725 (34.7%)
RMSShift: -0.0171 (25.2%)
0.0 < pT < 1.0
NoImp/Imp

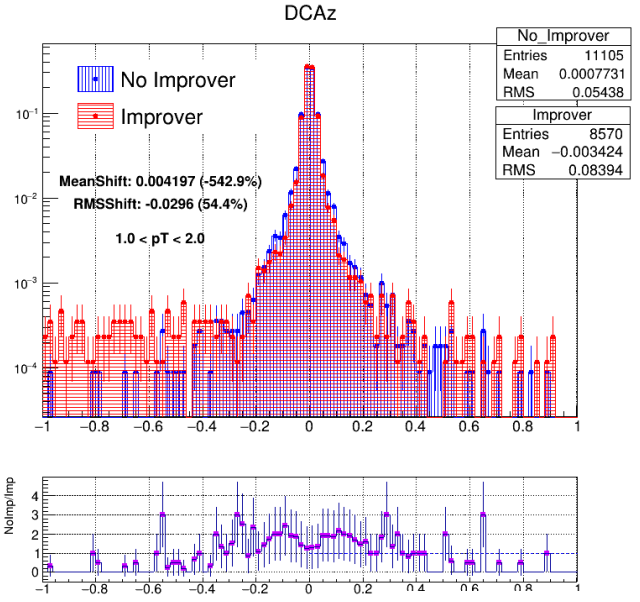

DCAxy
No_Improver
Entries 11105
Mean 0.0001183
RMS 0.04283
Improver
Entries 8570
Mean 0.001555
RMS 0.07077
No Improver
Improver
MeanShift: -0.001437 (1214.4%)
RMSShift: -0.0279 (65.2%)
1.0 < pT < 2.0
NoImp/Imp

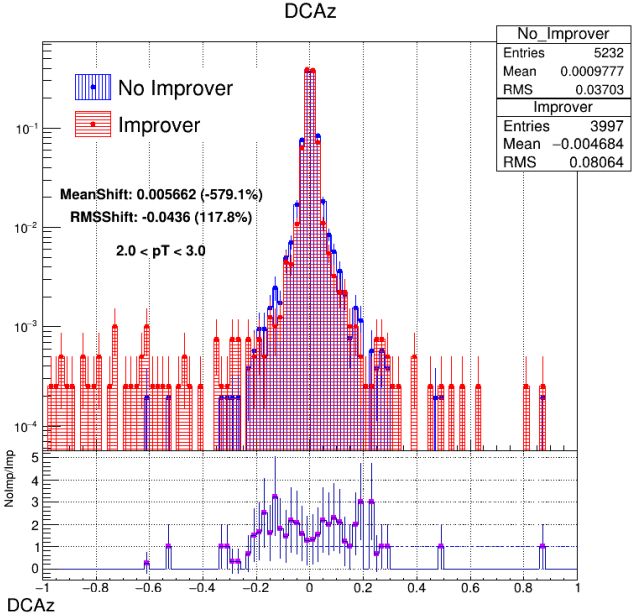

DCAxy
No_Improver
Entries 5232
Mean −0.0002238
RMS 0.02455
Improver
Entries 3997
Mean 0.001728
RMS 0.04668
No Improver
Improver
MeanShift: -0.002579 (3628.1%)
RMSShift: -0.0400 (133.0%)
2.0 < pT < 3.0
NoImp/Imp

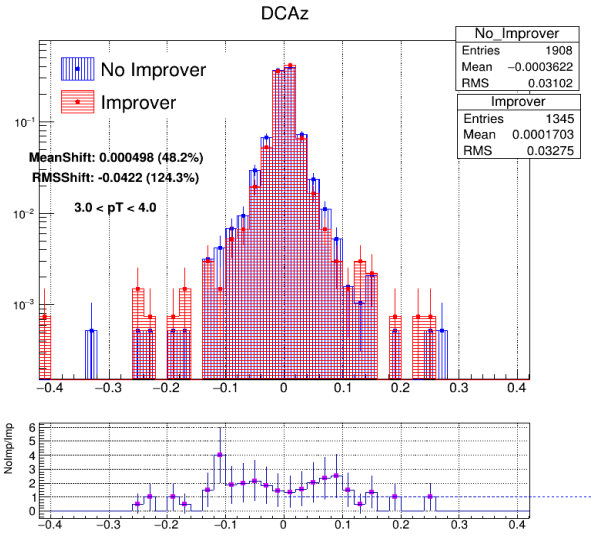

DCAxy
No_Improver
Entries 1908
Mean −0.001033
RMS 0.02719
Improver
Entries 1345
Mean −0.0005468
RMS 0.0478
No Improver
Improver
MeanShift: -0.001170 (-198.7%)
RMSShift: -0.0455 (160.8%)
3.0 < pT < 4.0
NoImp/Imp

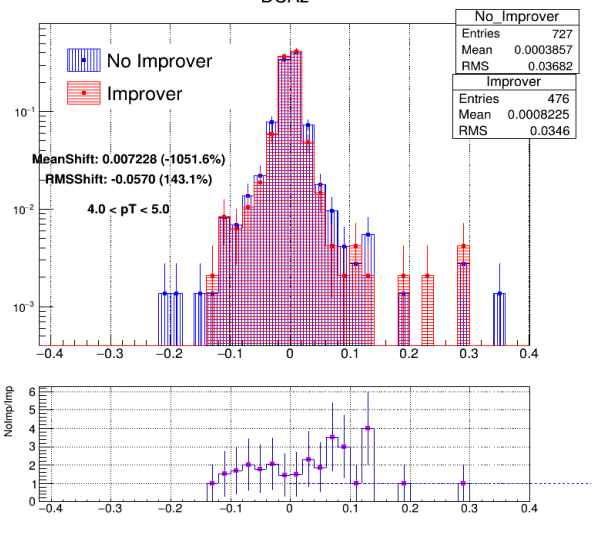

DCAxy
No_Improver
Entries 727
Mean −0.0009491
RMS 0.02537
Improver
Entries 476
Mean 0.0009091
RMS 0.04185
No Improver
Improver
MeanShift: -0.003047 (-450.6%)
RMSShift: -0.0429 (177.0%)
4.0 < pT < 5.0
NoImp/Imp

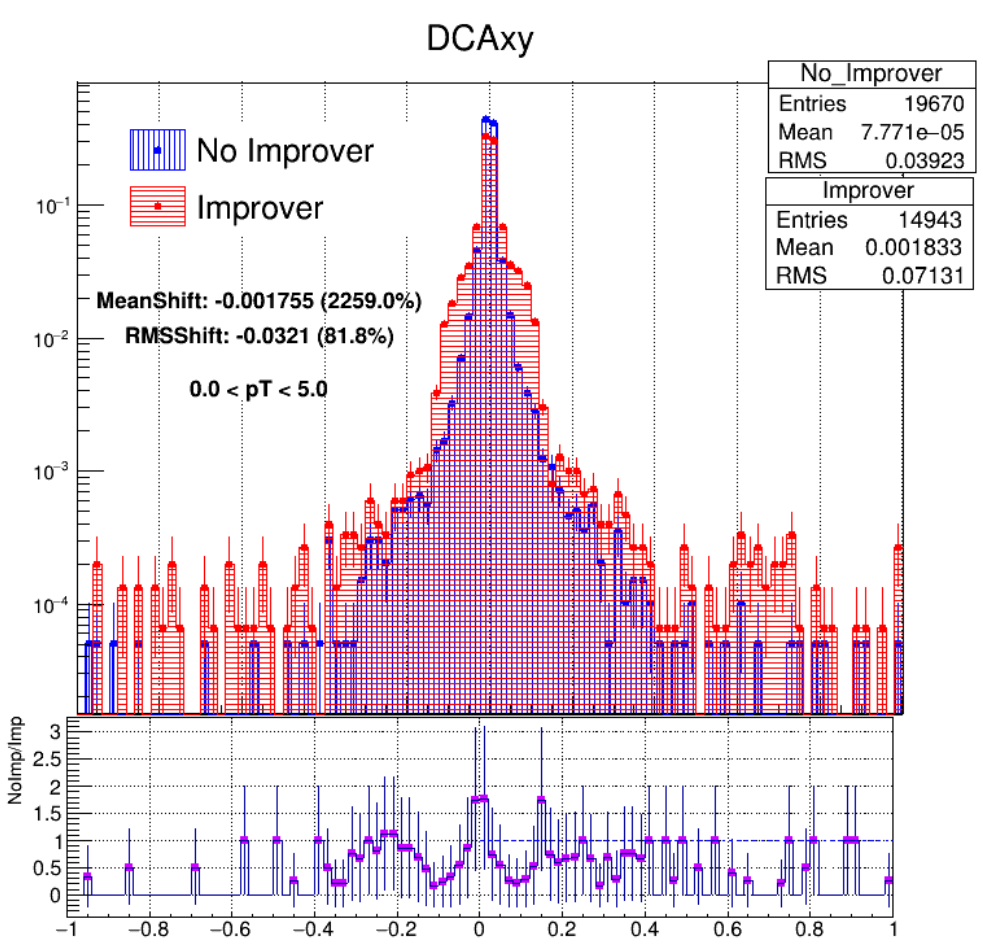

DCAz
No_Improver
Entries 698
Mean −0.002145
RMS 0.05308
Improver
Entries 555
Mean −0.002758
RMS 0.04895
No Improver
Improver
MeanShift: -0.000148 (-22.3%)
RMSShift: -0.0074 (9.3%)
0.0 < pT < 1.0
NoImp/Imp

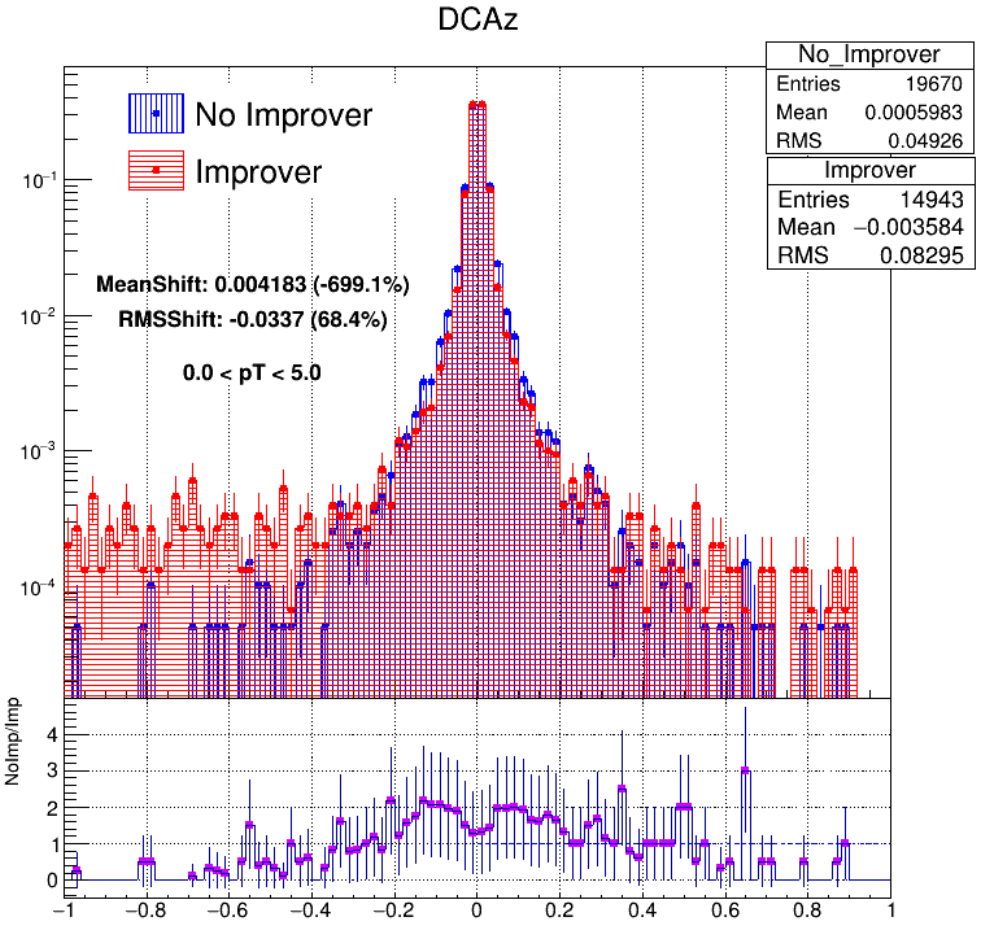

DCAz
No_Improver
Entries 11105
Mean 0.0007731
RMS 0.05438
Improver
Entries 8570
Mean −0.003424
RMS 0.08394
No Improver
Improver
MeanShift: 0.004197 (-542.9%)
RMSShift: -0.0296 (54.4%)
1.0 < pT < 2.0
NoImp/Imp


DCAz
No_Improver
Entries 5232
Mean 0.0009777
RMS 0.03703
Improver
Entries 3997
Mean −0.004684
RMS 0.08064
No Improver
Improver
MeanShift: 0.005662 (-579.1%)
RMSShift: -0.0436 (117.8%)
2.0 < pT < 3.0
NoImp/Imp

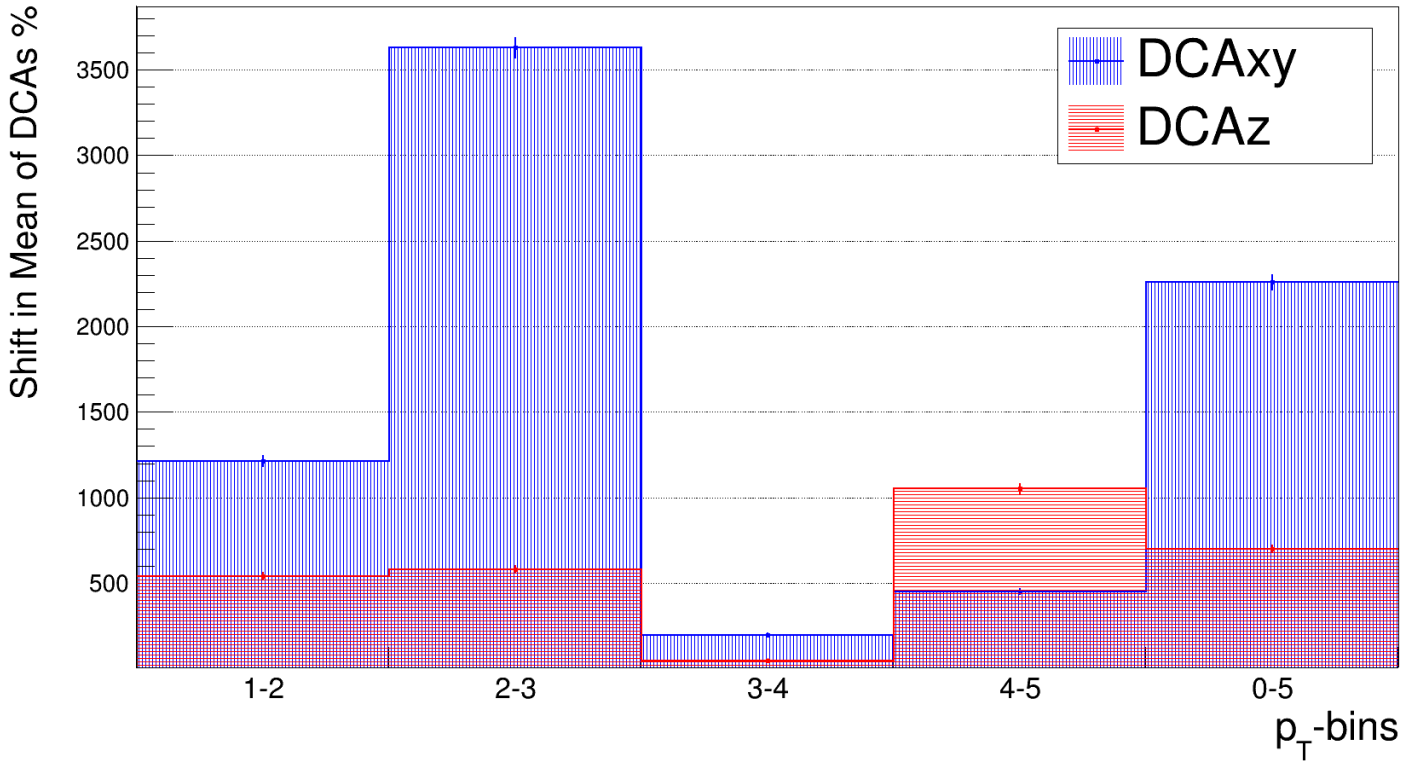

DCAz
No_Improver
Entries 1908
Mean −0.0003622
RMS 0.03102
Improver
Entries 1345
Mean 0.0001703
RMS 0.03275
No Improver
Improver
MeanShift: 0.000498 (48.2%)
RMSShift: -0.0422 (124.3%)
3.0 < pT < 4.0
NoImp/Imp


DCAz
No_Improver
Entries 727
Mean 0.0003857
RMS 0.03682
Improver
Entries 476
Mean 0.0008225
RMS 0.0346
No Improver
Improver
MeanShift: 0.007228 (-1051.6%)
RMSShift: -0.0570 (143.1%)
4.0 < pT < 5.0
NoImp/Imp


DCAxy
No_Improver
Entries 19670
Mean 7.771e−05
RMS 0.03923
Improver
Entries 14943
Mean 0.001833
RMS 0.07131
No Improver
Improver
MeanShift: -0.001755 (2259.0%)
RMSShift: -0.0321 (81.8%)
0.0 < pT < 5.0
NoImp/Imp


DCAz
No_Improver
Entries 19670
Mean 0.0005983
RMS 0.04926
Improver
Entries 14943
Mean −0.003584
RMS 0.08295
No Improver
Improver
MeanShift: 0.004183 (-699.1%)
RMSShift: -0.0337 (68.4%)
0.0 < pT < 5.0
NoImp/Imp

Comparision of the Mean positions shifts of DCA (xy & z)



Activities -

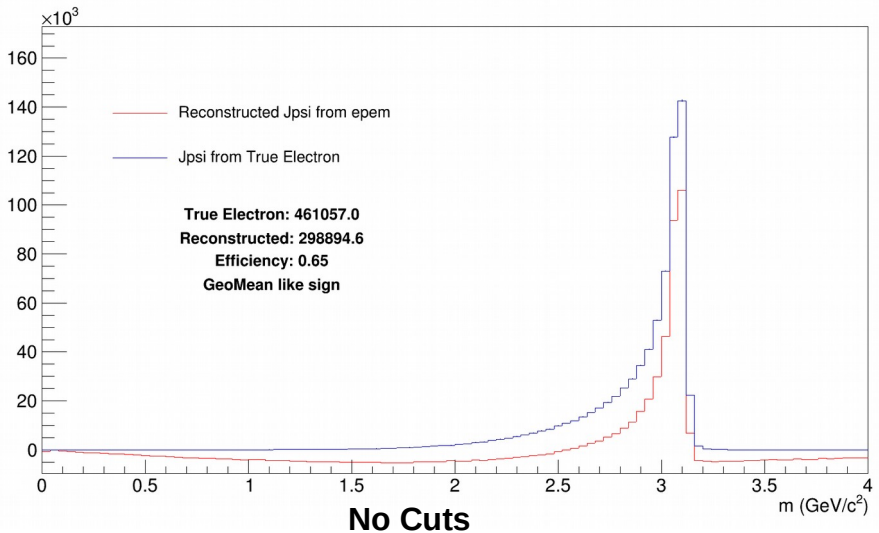
- Results of Improver Task
- J/psi-cut Efficiencies & Stuff

Efficiency of Jpsi-signal in different Selection Crieteria:

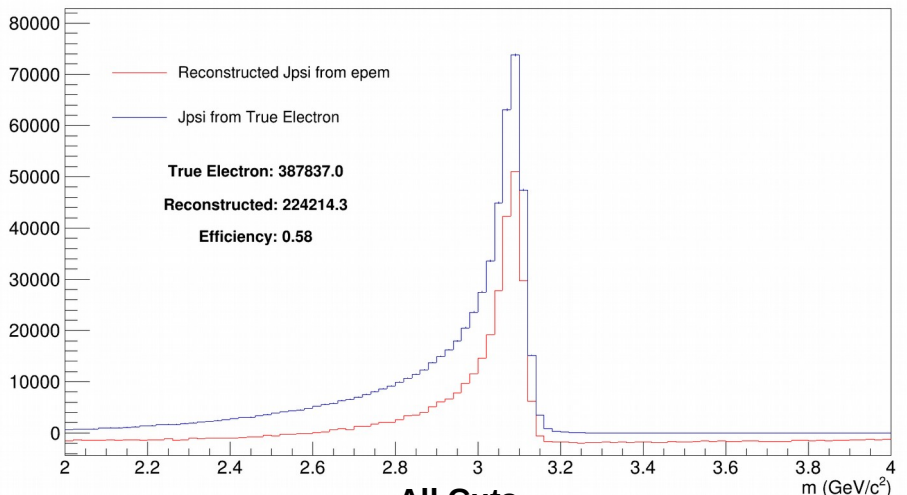
$$\frac{\text{Jpsi (recons.)}}{\text{Jpsi (MC truth)}} = \text{Efficiency}$$

DataSet :

- ~2M Events
- LHC19f1b (0-10%) Pb-Pb



No Cuts



All Cuts

Efficiency of Jpsi-signal in different Selection Crieteria:

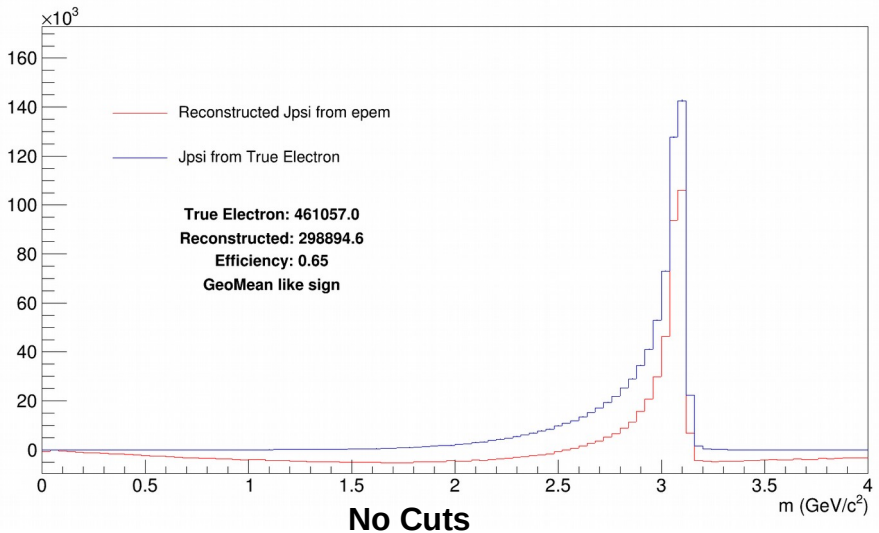
$$\frac{\text{Jpsi (recons.)}}{\text{Jpsi (MC truth)}} = \text{Efficiency}$$

DataSet :

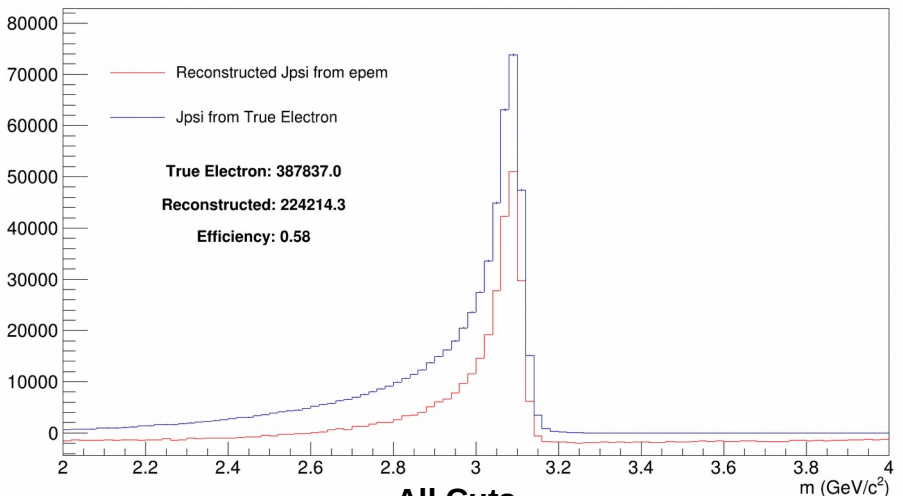
- ~2M Events
- LHC19f1b (0-10%) Pb-Pb

→ **Analysed Using Jpsi2ee task (Ionut's macro)**

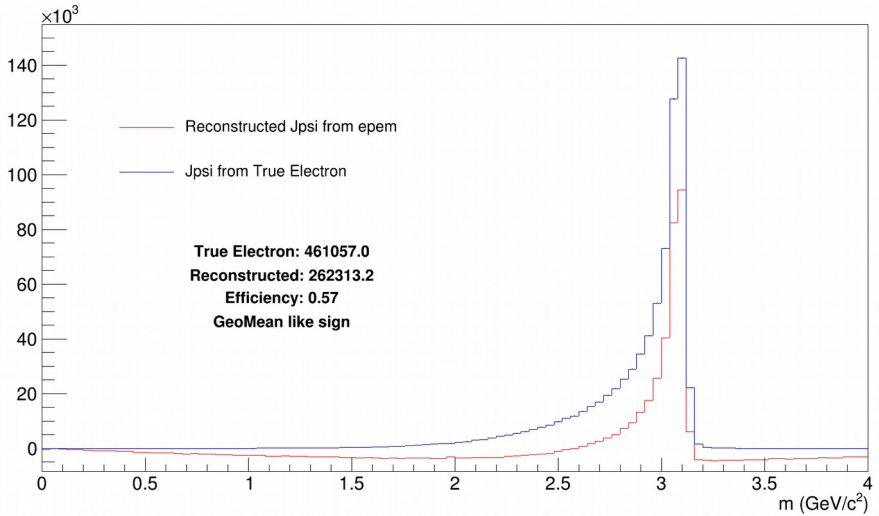
Cuts described in Backup slides



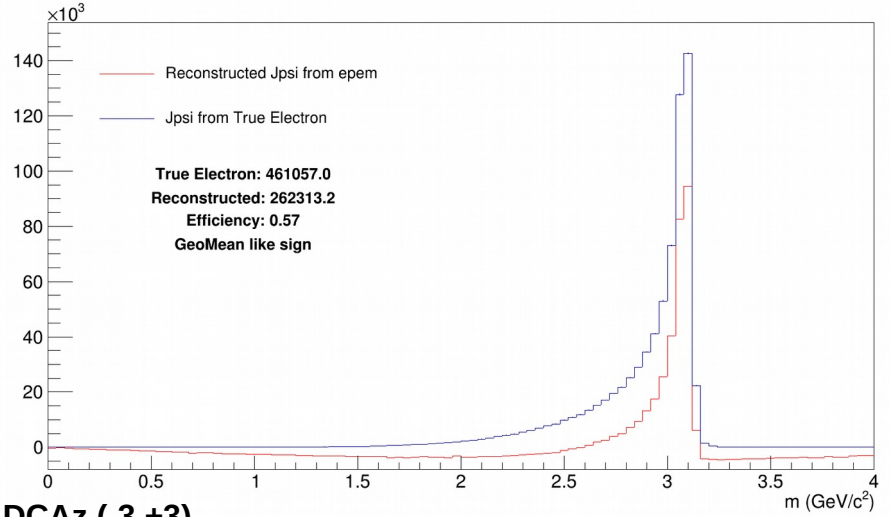
No Cuts



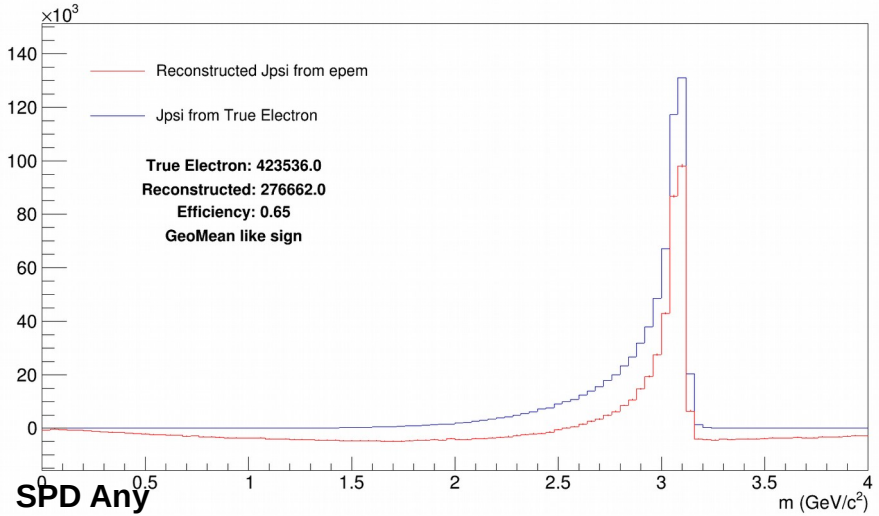
All Cuts



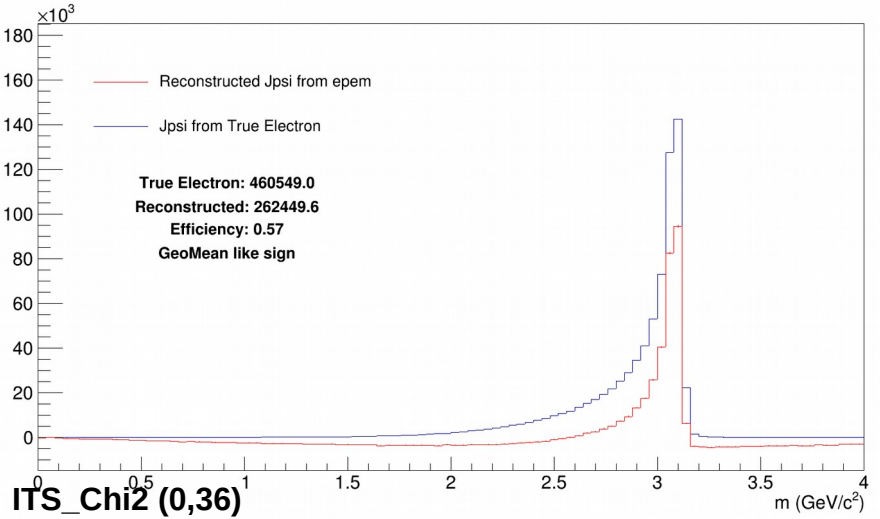
DCAxy (-1,+1)



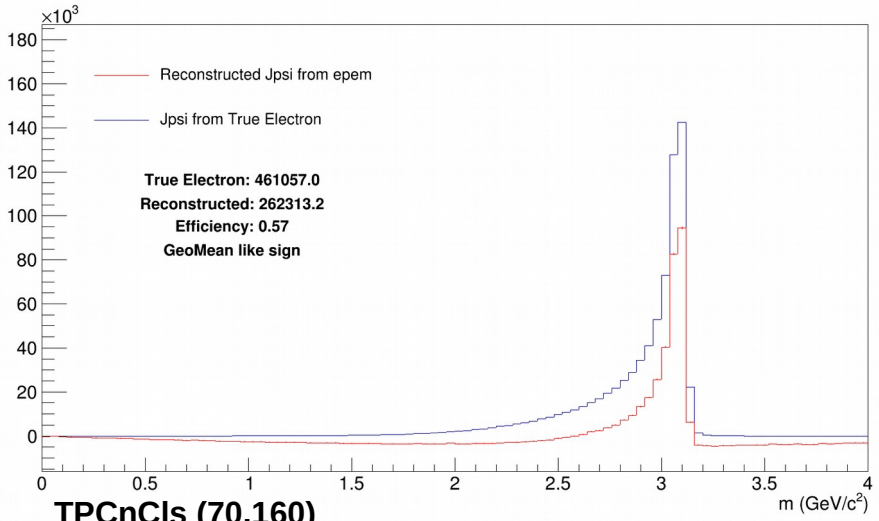
DCAz (-3,+3)



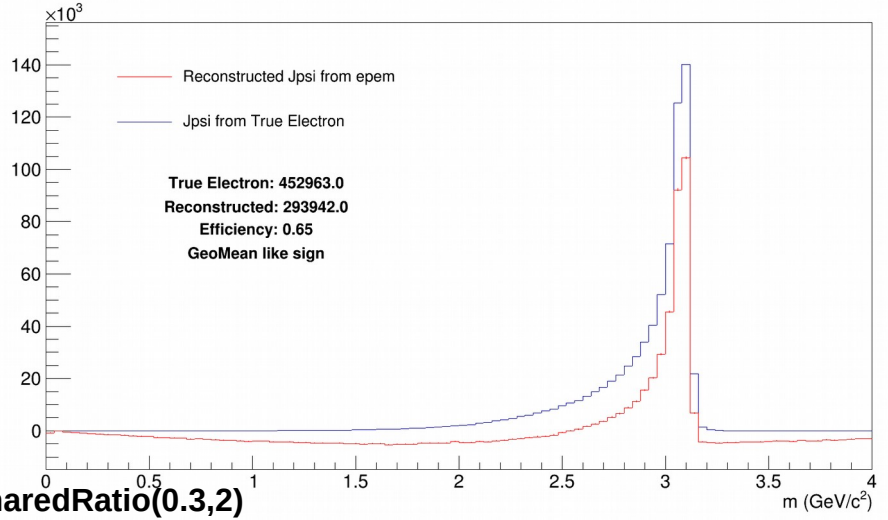
SPD Any



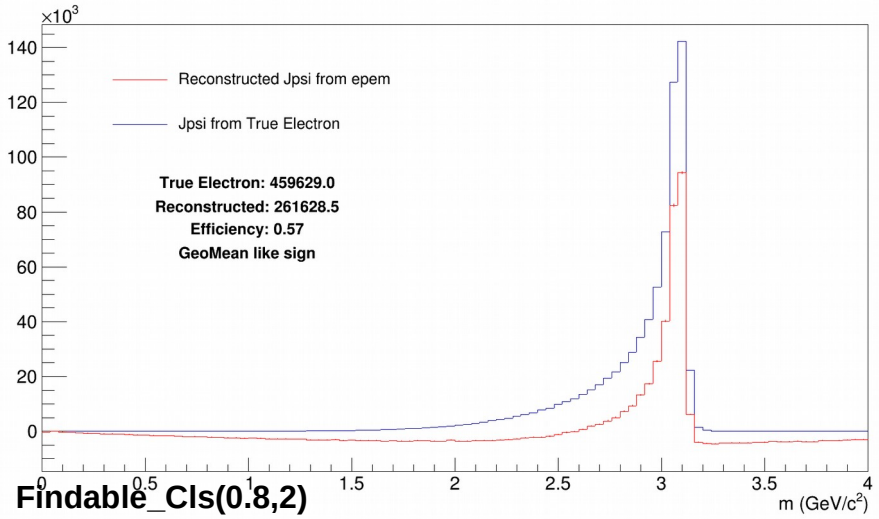
ITS_Chi2 (0,36)



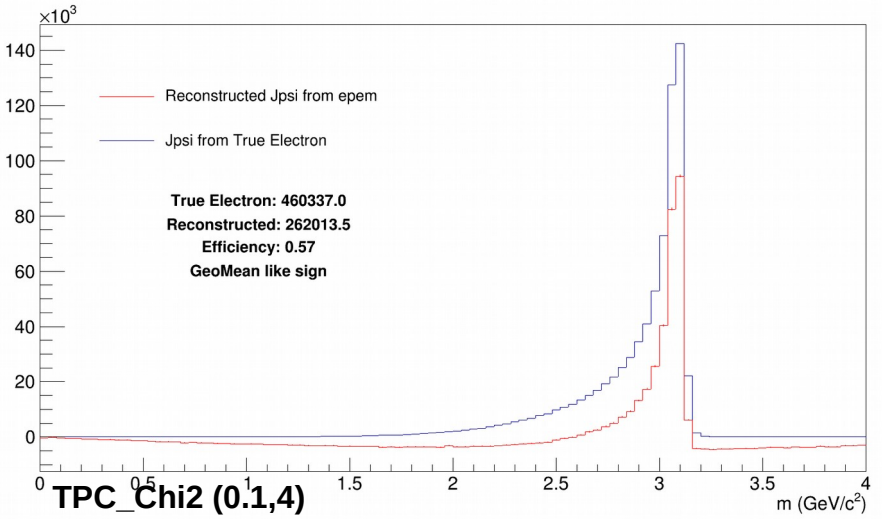
TPCnCls (70,160)



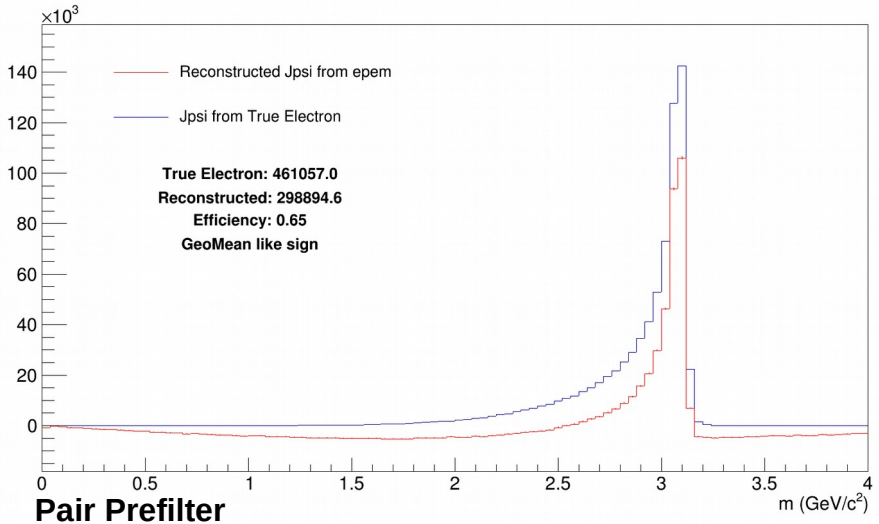
SharedRatio(0.3,2) Exclude



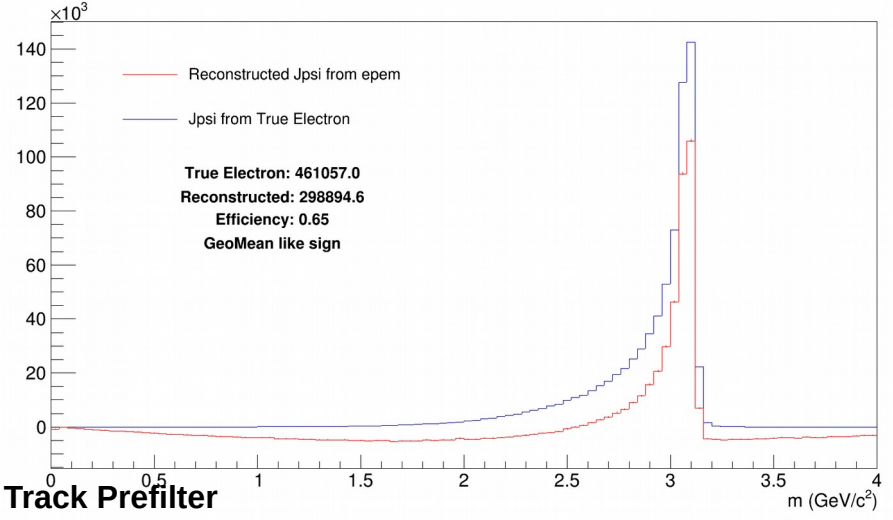
Findable_Cls(0.8,2)



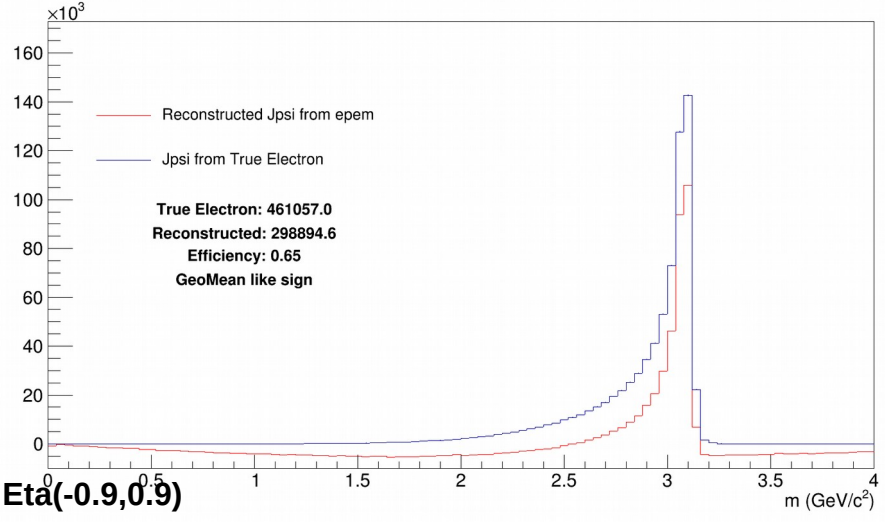
TPC_Chi2 (0.1,4)



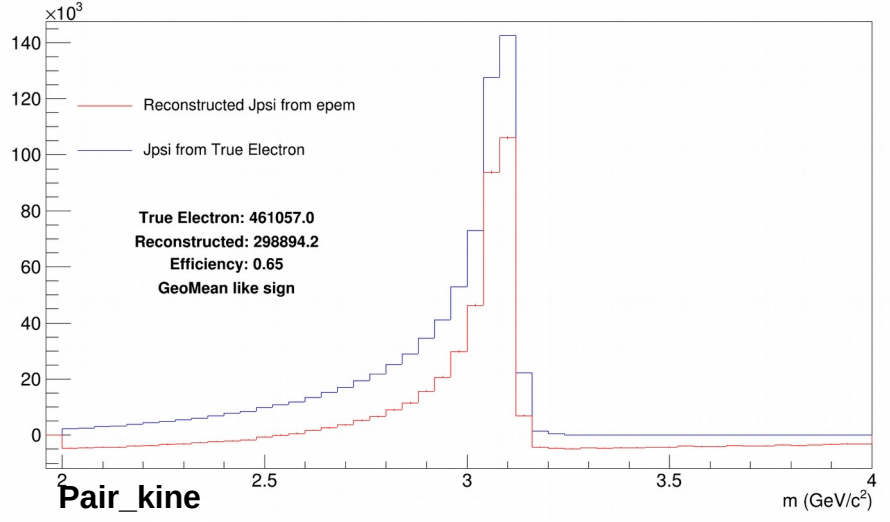
Pair Prefilter



Track Prefilter

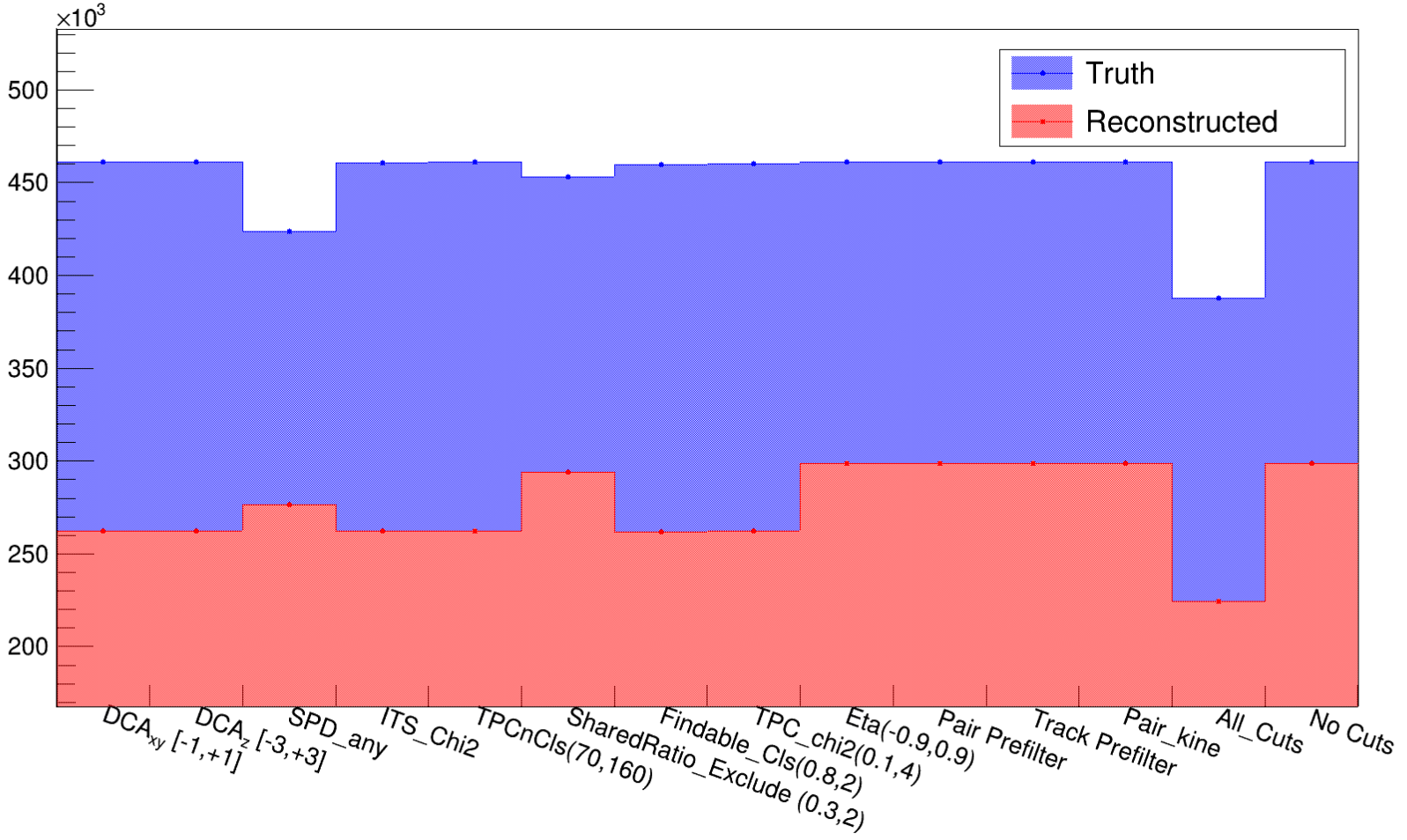


Eta(-0.9,0.9)

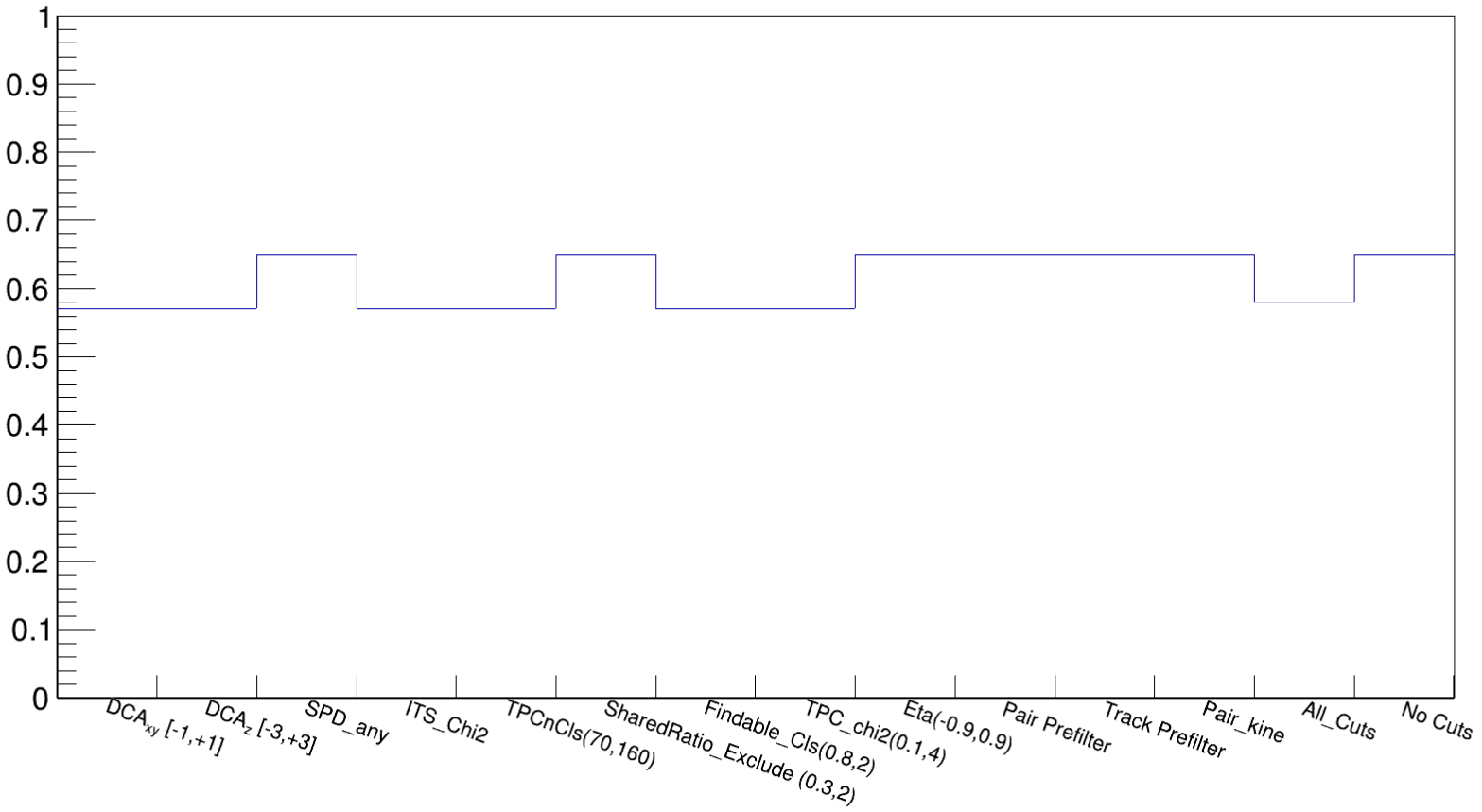


Pair_kine

Comparison of Truth & Reconstructed Jpsi (Cut-wise)



Efficiency (Cut-wise)



Regarding the Lxy distributions comparisons (signal and bkg), I will discuss offline with Jacek B.

What's Next ??

- Efficiency in different pT and Eta bins

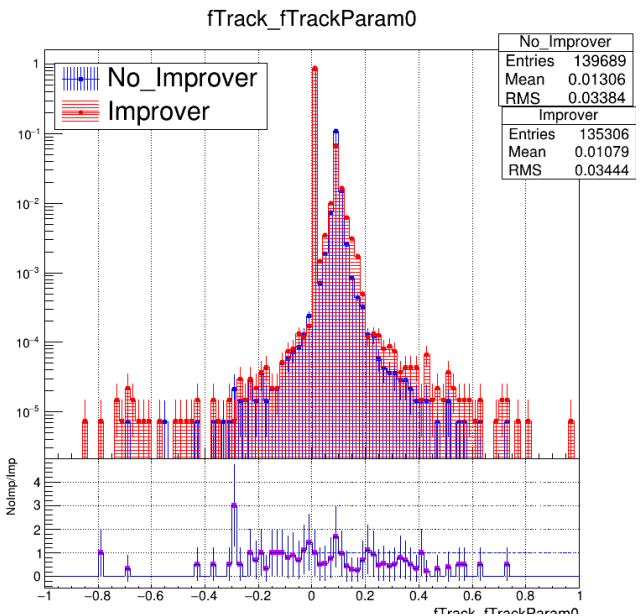
Regarding the Lxy distributions comparisons (signal and bkg), I will discuss offline with Jacek B.

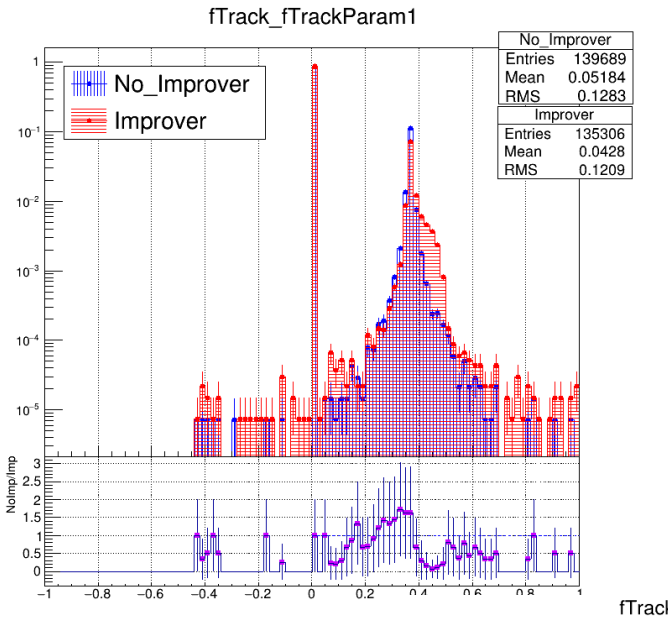
What's Next ??

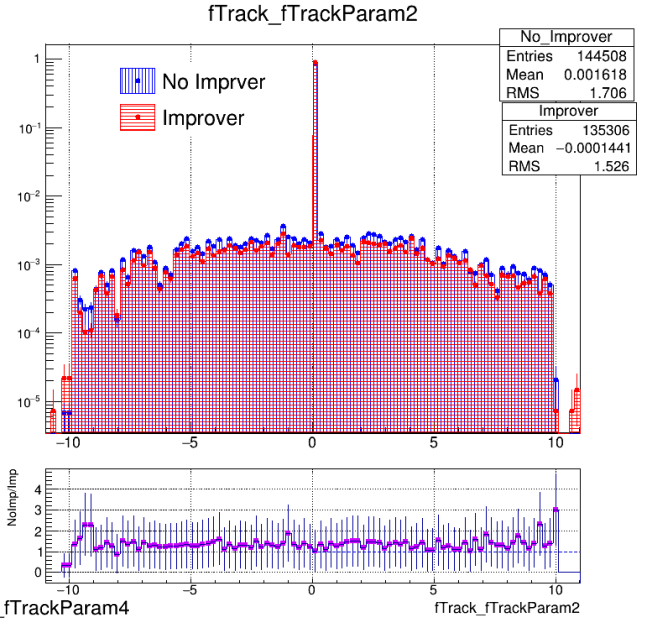
- Efficiency in different pT and Eta bins

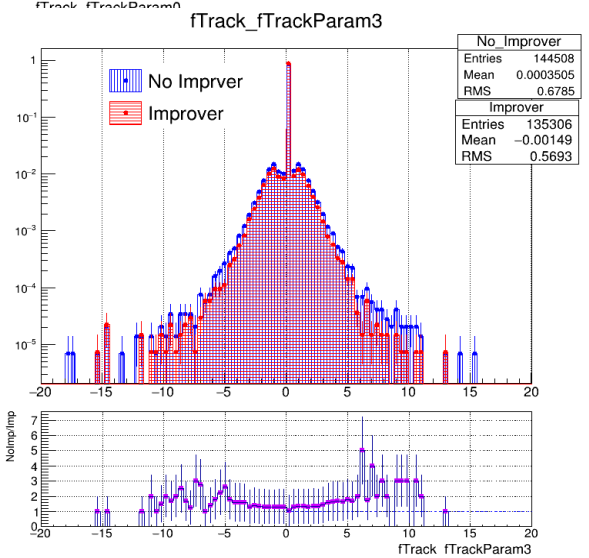
Thank you!

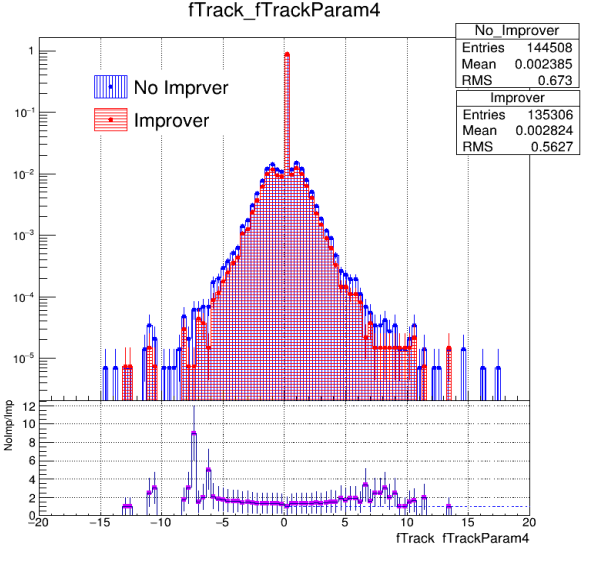
Back-Up


fTrack_fTrackParam0
No_Improver
Improver
No_Improver
Entries 139689
Mean 0.01306
RMS 0.03384
Improver
Entries 135306
Mean 0.01079
RMS 0.03444
NoImp/Imp


fTrack_fTrackParam1
No_Improver
Improver
No_Improver
Entries 139689
Mean 0.05184
RMS 0.1283
Improver
Entries 135306
Mean 0.0428
RMS 0.1209
NoImp/Imp

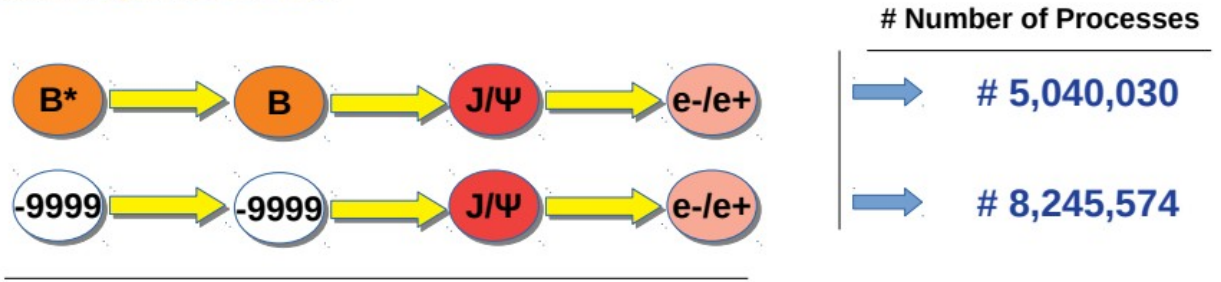

fTrack_fTrackParam2
No Imprver
Improver
No_Improver
Entries 144508
Mean 0.001618
RMS 1.706
Improver
Entries 135306
Mean −0.0001441
RMS 1.526
NoImp/Imp
fTrack_fTrackParam2


fTrack_fTrackParam3
No Imprver
Improver
No_Improver
Entries 144508
Mean 0.0003505
RMS 0.6785
Improver
Entries 135306
Mean −0.00149
RMS 0.5693
NoImp/Imp
fTrack fTrackParam3

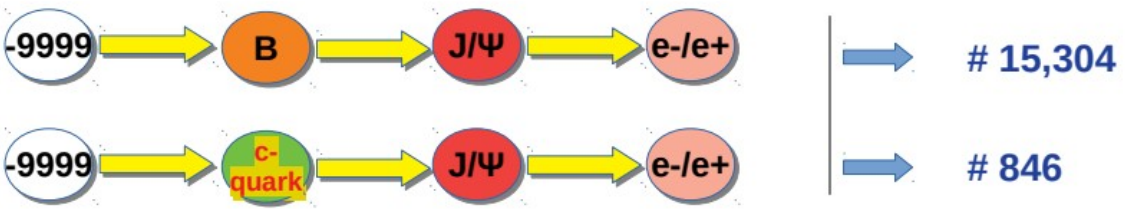

fTrack_fTrackParam4
No Imprver
Improver
No_Improver
Entries 144508
Mean 0.002385
RMS 0.673
Improver
Entries 135306
Mean 0.002824
RMS 0.5627
NoImp/Imp
fTrack fTrackParam4

A Glance of All Processes hierarchy in our MC

Dominating Processes :



Rare Processes :



```
Cuts Used
------------
    - StandardCut
        - BasicCut
            - standardPrimary
                - DCAxy
                - DCAz
            - rejectKinks
            - standardITStracking
                - ITS refit
                - SPD any
                - chi2 (0,36)

            - standardTPCtracking
                - TPCnCls(70.,160.0)
                - kTPCnclsSharedRatio, 0.3, 2 (Exclude this region)
                - kTPCcrossedRowsOverFindableClusters, 0.8, 2.
                - kTPCchi2, 0.1, 4.0
        - StandardKine
            - Pt (1,30)
            - Eta (-0.9,0.9)                -
    - Pair Prefilter ====> (kMass, 0.0, 0.05, kTRUE) Mass Exclusion Cut
    - Track Prefilter ====> (kPt, 0.9,1000.0) && SetTrackFilterBit(kPrefilterCut)
    - PairKine ===>
                  pairKine->AddCut(AliReducedVarManager::kPt, 0.0,100.0);
                  pairKine->AddCut(AliReducedVarManager::kRap, -0.9,0.9);
                  pairKine->AddCut(AliReducedVarManager::kMass, 2.0, 4.0);
```