

# $p_T$ spectra as a function of Multiplicity and Transverse Spherocity in pp collisions using a Bayesian Unfolding

J. David Romo

14/09/21

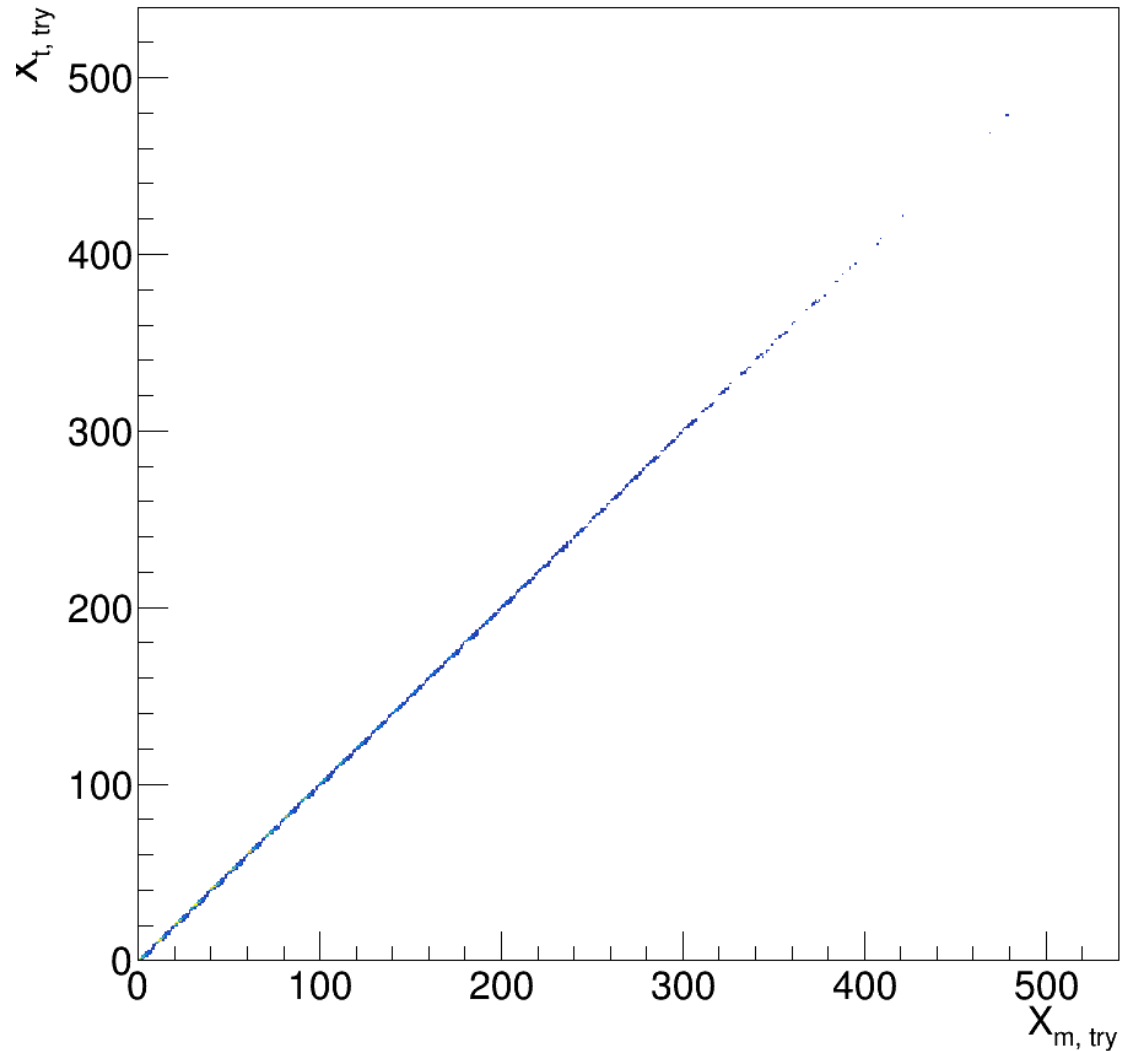
- Introducing a new analysis task for the unfolding procedure SpheroUnfolding:

```
C AliAnalysisTaskSpheroUnfolding.h
29 class AliAnalysisTaskSpheroUnfolding : public AliAnalysisTaskSE
30 {
31     public:
32         AliAnalysisTaskSpheroUnfolding();
33         AliAnalysisTaskSpheroUnfolding(const char *name);
34         virtual ~AliAnalysisTaskSpheroUnfolding();
35
36         virtual void UserCreateOutputObjects();
37         virtual void UserExec(Option_t* option);
38         virtual void Terminate(Option_t* option);
39
40         //void GetLeadingObjectFromArray(const std::vector<Float_t> &pt, const std::vector<Float_t> &phi, Int_t
41         void GetDetectorResponse(const std::vector<Float_t> &phiGen, const std::vector<Float_t> &ptGen, Int_t mu
42         void GetMultiplicityDistributionsTrue(const std::vector<Float_t> &phiGen, const std::vector<Float_t> &pt
43
44         void GetMultiplicityDistributions(const std::vector<Float_t> &phiRec, const std::vector<Float_t> &ptRec,
45
46         void GetMultiplicityDistributionsData(const std::vector<Float_t> &phiRec, const std::vector<Float_t> &pt
47
48         void SetPtMin(Double_t val) {fPtMin = val;} // Set pT cut for associated particles
49         void SetLeadingPtMin(Double_t PtLmin) {fLeadPtCutMin = PtLmin;} // use different ptcuts
50         void SetLeadingPtMax(Double_t PtLmax) {fLeadPtCutMax = PtLmax;} // use different ptcuts
51         void SetUseMC(Bool_t mc = kFALSE) {fUseMC = mc;} // use to analyse MC data
52         void SetMCclosureTest(Bool_t mcc = kFALSE) {fIsMCclosure = mcc;}
53         void SetIsHybridAnalysis(Bool_t isHy = kFALSE) {fIsHybAna = isHy;}
54         bool HasRecVertex();
55         //Systematic =====
56         void SetTPCclustersVar1(Bool_t TPCclustersVar1 = kFALSE) {fTPCclustersVar1 = TPCclustersVar1;}
57         void SetTPCclustersVar2(Bool_t TPCclustersVar2 = kFALSE) {fTPCclustersVar2 = TPCclustersVar2;}
```

- Based on Sushanta and Luz's AliAnalysisTaskChargedVsRT and Gyula's AliAnalysisTaskGenUeSpherocity, specially in the latter's use of pseudotracks.

# To be done: Detector resolution

Response Matrix only for Multiplicity and p<sub>T</sub>



- Because the use of pseudotracks, one must add by hand the detector effect on the p<sub>T</sub> resolution, which at this stage is missing.

# To be done: How to apply the $S_0$ percentiles?

- With xook, the percentiles histograms were obtained running two times the full statistics.
- This same approach probably would not be very convenient, given how resource consuming process the LEGO trails are.

# To be done: How to apply the $S_0$ percentiles?

■ With xook, the percentiles histograms were obtained running two times the full statistics.

■ This same approach probably would not be very convenient, given how resource consuming process the LEGO trails are.

■ Still Reading about the implementation of the Sphericity Percentile helper and Sphericity Utils

```
351 // Extract the event sphericity using the official cuts
352 Double_t S0m = -1.0;
353 S0m = fSpheroUtils->GetEventShape( event, hphiso, hetaso );
354 // sphericity percentile
355 Double_t SotPerc = -1;
356 SotPerc = GetSpheroPercentile(S0m, fnRefGlobal);
357 fbinSom = -1;
358 fbinSom = GetSpheroPercentileBin(SotPerc);
359 // Correlation between sphericity and multiplicity
360 hS0Global08->Fill(fnRefGlobal, S0m);
361 if (fbinSom >= 0 && fbinSom < fns0B)
362     hS0Global[fbinSom]->Fill(fnRefGlobal, S0m);
363
364
```

```
162 if (!fTrackFilter){
163     fTrackFilter = new AliAnalysisFilter("trackFilter2015");
164     SetTrackCuts(fTrackFilter);
165 }
166 // Helper to obtain the sphericity percentile
167 TFile * finPercent = 0;
168 TString nameSoHelper = AliDataFile::GetFileName("PWGMM/spheroLHC15fpass2.root");
169 finPercent = TFile::Open(nameSoHelper);
170 fns0B = fSoBining->GetNbinsX();
171 fnMultbins = 100;
172 for( Int_t i_mult = 0; i_mult < fnMultbins; ++i_mult ){
173     hS0Mperc[i_mult] = 0;
174     hS0Mperc[i_mult] = (TH1D *)finPercent->Get(Form("hS0Mperc%d", i_mult));
175     hS0MAux[i_mult] = 0;
176     hS0MAux[i_mult] = (TH2D *)finPercent->Get(Form("hS0MAuxMult%d", i_mult));
177 }
178
```