Update on tracking benchmarks

Shyam Kumar*, Annalisa Mastroserio, Domenico Elia
INFN Bari, Italy

Tracking Performances

Momentum and DCA resolutions Benchmark

- Wrote an script that runs single particle simulation and reconstruction on 10k events

mom_array=(0.5 1.0 2.0 5.0 10.0 15.0) We can further add/remove values

Simulation:

Latest geometry

Uniform in eta (range taken from thetaMin to thetaMax) Particle name

```
for ((i=0; i<${#mom_array[@]}; i++)); do
npsim --compactFile epic_craterlake_tracking_only.xml --outputFile sim${mom_array[i]}.edm4hep.root --numberOfEvents $nevents
--enableGun --gun.thetaMin 3*deg --gun.thetaMax 177*deg --gun.distribution eta --gun.particle pi- --gun.momentumMin $
{mom_array[i]}*GeV --gun.momentumMax ${mom_array[i]}*GeV --gun.multiplicity 1 --random.seed 100000
done
```

Momentum

Seed (keeping same seed)

Reconstruction:

Once there are some changes: simulation and reconstruction run with 10k events

```
for ((i=0; i<${#mom_array[@]}; i++)); do
eicrecon \
 -Pnthreads=1 \
 -Pjana:debug_plugin_loading=1 \
 -Pjana:nevents=$nevents \
 -Pacts:MaterialMap=calibrations/materials-map.cbor \
 -Ppodio:output_file="${filename}"_${mom_array[i]}.edm4eic.root \
 -Pdd4hep:xml_files=  epic_craterlake_tracking_only.xml \
 -Ppodio:output_collections="MCParticles,CentralCKFTrajectories,CentralCKFTrackParameters,CentralCKFSeededTrackParameters,CentralTrackVertices" \
 sim${mom_array[i]}.edm4hep.root
done
```

Output collection for the tree

Performances:

Tracking_Performances.C

It automatically reads the reconstruction output and stores the plots for momentum resolutions $(p_{rec}-p_{true})/p_{true}$

```
// MC and Reco information
TTreeReaderArray<Float_t> charge(myReader, "MCParticles.charge");
TTreeReaderArray<Double_t> vx_mc(myReader, "MCParticles.vertex.x");
TTreeReaderArray<Double_t> vy_mc(myReader, "MCParticles.vertex.y");
TTreeReaderArray<Double_t> vz_mc(myReader, "MCParticles.vertex.z");
TTreeReaderArray<Float_t> px_mc(myReader, "MCParticles.momentum.x");
TTreeReaderArray<Float_t> py_mc(myReader, "MCParticles.momentum.y");
TTreeReaderArray<Float_t> pz_mc(myReader, "MCParticles.momentum.z");
TTreeReaderArray<Int_t> status(myReader, "MCParticles.generatorStatus");
TTreeReaderArray<Int_t> pdg(myReader, "MCParticles.PDG");
TTreeReaderArray<Int_t> match_flag(myReader, Form("CentralCKF%sTrackParameters.type",name.Data()));
TTreeReaderArray<Float_t> d0xy(myReader, Form("CentralCKF%sTrackParameters.loc.a",name.Data()));
TTreeReaderArray<Float_t> d0z(myReader, Form("CentralCKF%sTrackParameters.loc.b",name.Data()));
TTreeReaderArray<Float_t> theta(myReader, Form("CentralCKF%sTrackParameters.theta",name.Data()));
TTreeReaderArray<Float_t> phi(myReader, Form("CentralCKF%sTrackParameters.phi",name.Data()));
TTreeReaderArray<Float_t> qoverp(myReader, Form("CentralCKF%sTrackParameters.qOverP",name.Data()));
```

MC Information

Reco Information (truth/real seed)

Track Parameters: $(l_0, l_1, \theta, \phi, q/p)$

Acts example

```
m_p = std::abs(1.0 / m_qop);
m_pt = m_p * std::sin(m_theta);
m_eta = std::atanh(std::cos(m_theta));
```

$(p_{rec}-p_{true})/p_{true}$

1d histogram for each eta range and momentum see backup

We want DCA as a function of p_T (Storing 3D histogram)

```
h_d0xy_3d->Fill(d0xy[j]*0.1, etamc, ptmc); // cm
h_d0z_3d->Fill(d0z[j]*0.1, etamc, ptmc); // cm
```

Momentum Resolutions:

doCompare_truth_real_widebins_mom.C

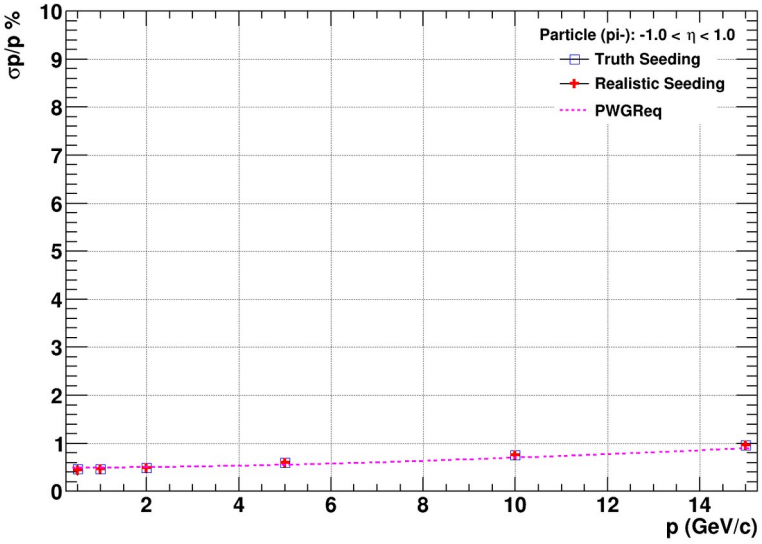
- Fitting 1D Gaussian distributions for each momentum and eta range stored in **debug_plots/** (can be checked)
- Fitting done in two steps to select the core region and avoiding long tails specially at low momentum

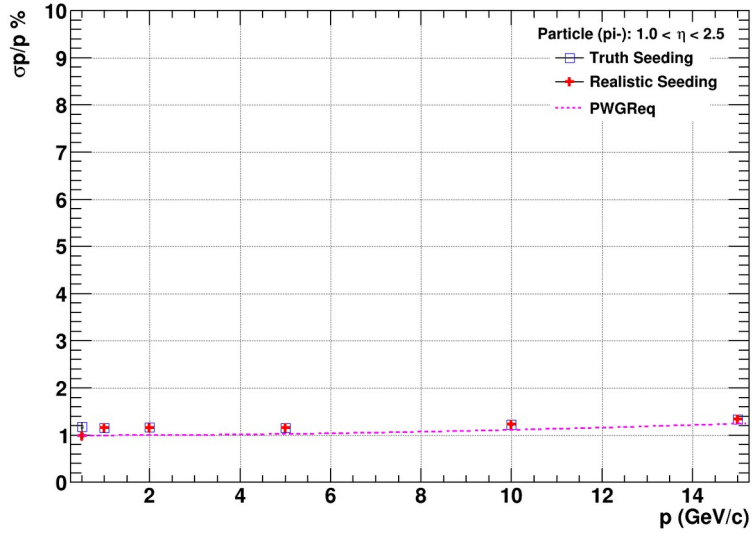
```
double mu_truth = hist_truth->GetMean();
double sigma_truth = hist_truth->GetStdDev();
func_truth->SetRange(mu_truth-2.0*sigma_truth,mu_truth+2.0*sigma_truth); // fit with in 2 sigma range
hist_truth->Fit(func_truth,"NR+");
mu_truth = func_truth->GetParameter(1);
sigma_truth = func_truth->GetParameter(2);
func_truth->SetRange(mu_truth-2.0*sigma_truth,mu_truth+2.0*sigma_truth);
```

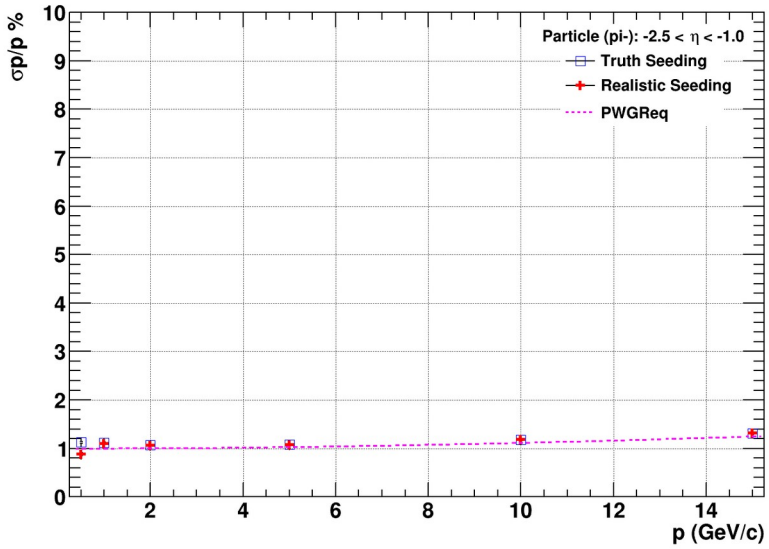
- Stored the png files together with the root files with multigraph. We can read the graph for the real seed which will be used in the TDR.
- The plots are produced automatically with the latest software (linked).
- If something looks strange we can check 1D Gaussian plots to understand more
- There is a further update in the code which also include software version and also storing the values of resolutions in text file.

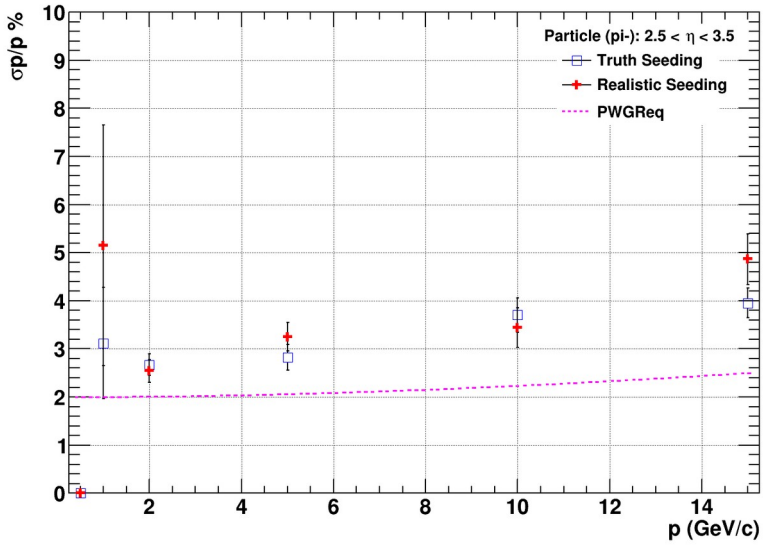
Benchmark plots

Momentum Resolutions









DCA_T Resolutions

Committed the code to the repository soon will be merged (again will produce debug plots/ final resolution plot)

```
h_d0xy_3d->Fill(d0xy[j]*0.1, etamc, ptmc); // cm
h_d0z_3d->Fill(d0z[j]*0.1, etamc, ptmc); // cm
```

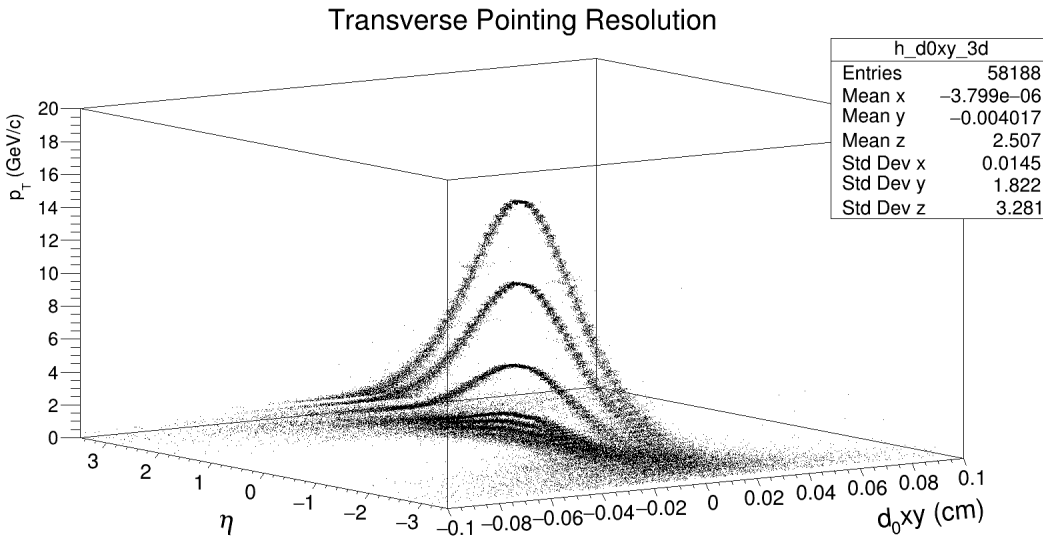
Tracking_Performances.C

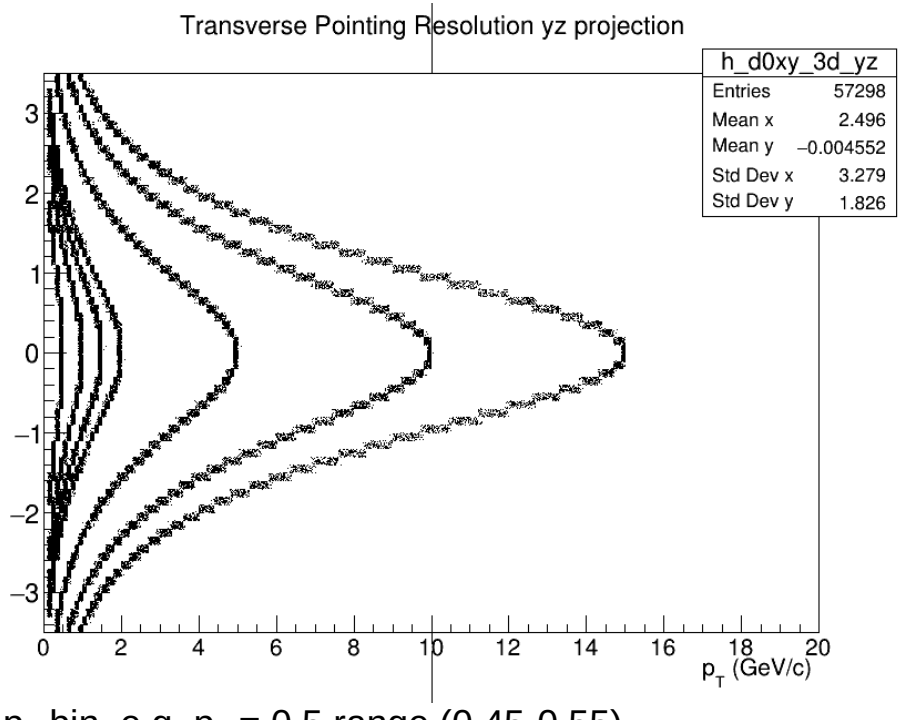
mom_array=(0.5 1.0 2.0 5.0 10.0 15.0)

Code committed after local test but not merged

Total expected entries = 60k (10k for each momentum)

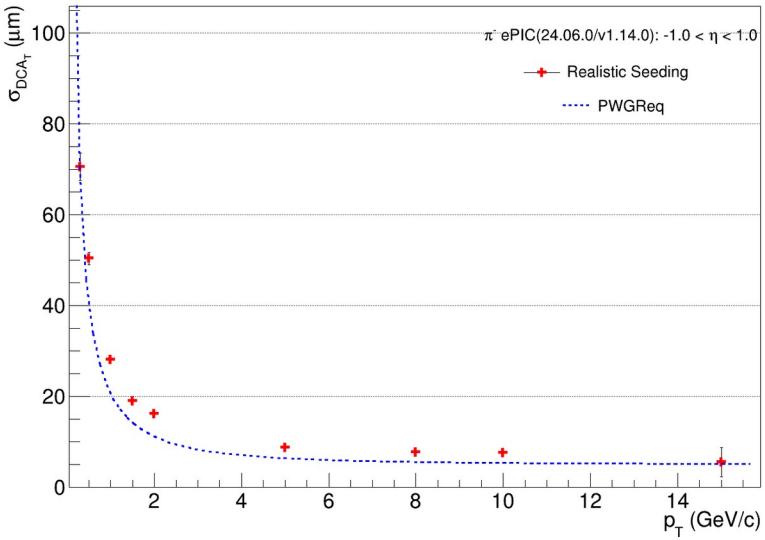
double pt[nptbins] ={0.5,1.0,2.0,5.0,10.0,15.0};

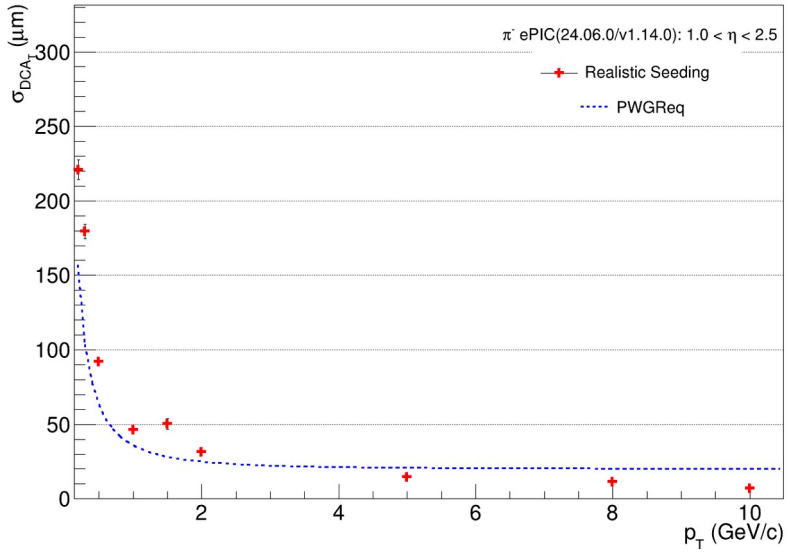


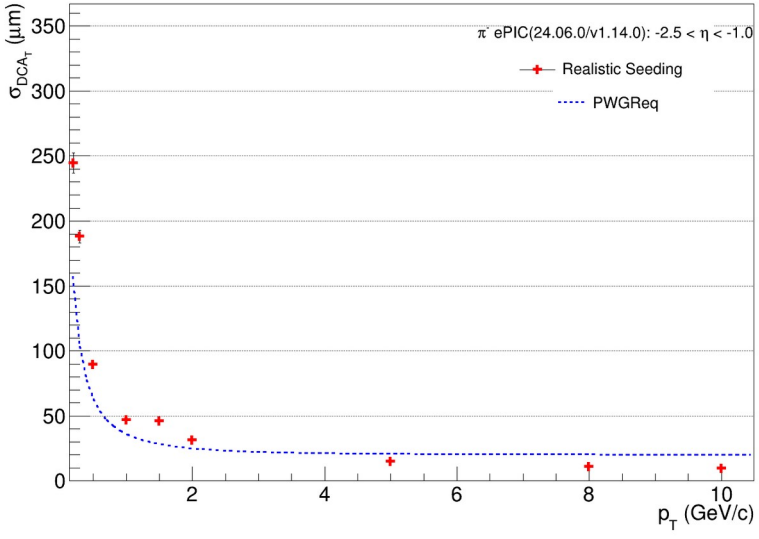


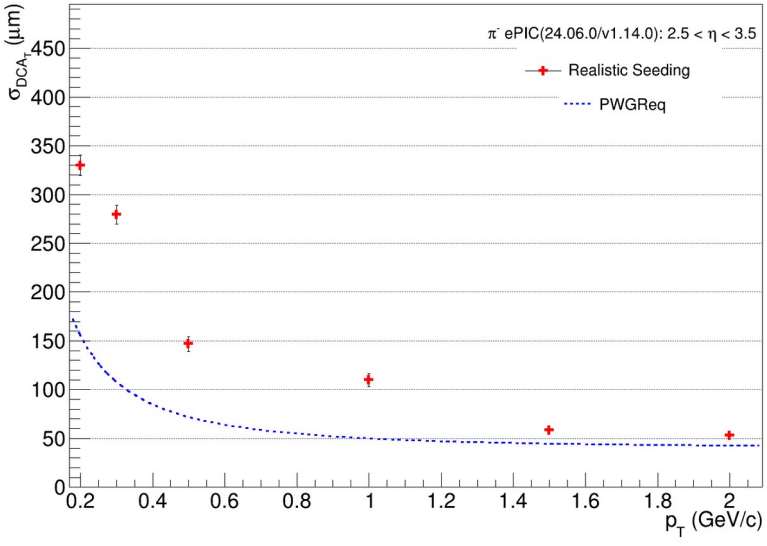
During projection: 10 % range is considered for each p_T bin, e.g. p_T = 0.5 range (0.45-0.55)

DCA_T Resolutions



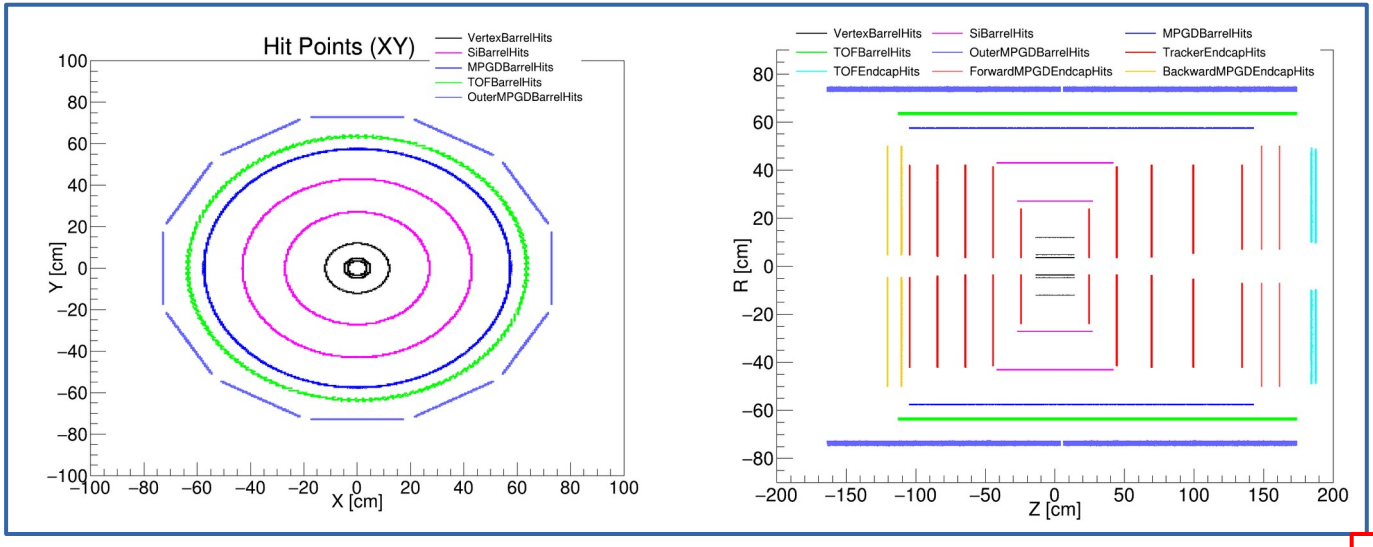






Hit Map

This is a further suggestion to include these plots

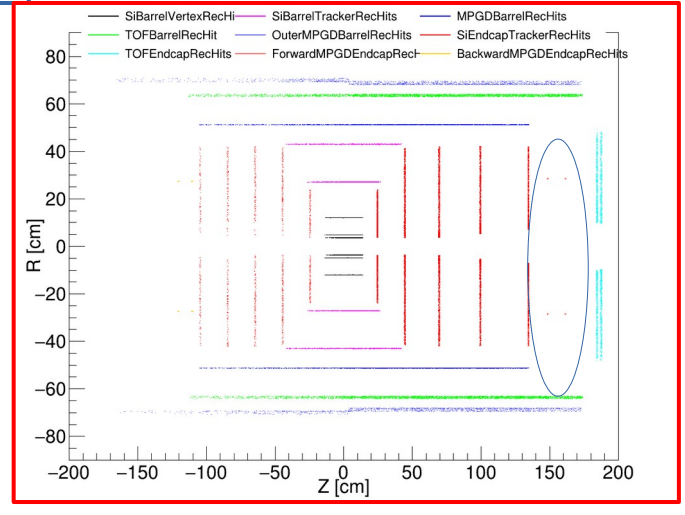


I spotted missing hits in MPGD disks: reported in tracking meeting **because of this plot**

The plot can be produced on the fly for one momentum while running npsim command

Code on EICRecon

October simulation campaign



Summary

- The tracking benchmarks showing the improved results after update of the material map
- If we can increase the statistics, we can directly use these plots for the performances
- Further will be useful to include detector hit maps plots
- Also working with Torri to create the performance plots for simulation campaigns (do we want to show results from May campaign?)

Thank You !!

Code to draw momentum resolution from root files

```
void draw_req_Mom(double etamin, double etamax, double xmin=0., double xmax=0.);
void draw_mom(TString particle = "pi-", double etamin=-1.0, double etamax=1.0){
   gStyle->SetPalette(1);
   gStyle->SetOptTitle(1);
   gStyle->SetTitleOffset(1.0,"XY");
   gStyle->SetTitleSize(.04,"XY");
   gStyle->SetLabelSize(.04,"XY");
   gStyle->SetHistLineWidth(2);
   gStyle->SetOptFit(1);
   gStyle->SetOptStat(1);

  TCanvas *c_mom = new TCanvas("cmom","cmom",1400,1000);
  c_mom->SetMargin(0.10, 0.05 ,0.1,0.05);
  c_mom->SetGrid();

  TFile *f = TFile::Open(Form("mom_resol_%1.1f_eta_%1.1f.root",etamin,etamax));
  TMultiGraph *mg = (TMultiGraph*)f->Get(Form("mom_resol_%1.1f_eta_
%1.1f",etamin,etamax));
  TGraphErrors *gr = (TGraphErrors*) mg->GetListOfGraphs()->At(1);
  c_mom->cd();
  gr->Draw("AP");
  gr->GetXaxis()->SetRangeUser(0.45,15.2);
  gr->GetYaxis()->SetRangeUser(0.,2.0*TMath::MaxElement(gr->GetN(),gr->GetY()));
  draw_req_Mom(etamin,etamax,0.,gr->GetXaxis()->GetXmax());

  TLegend *lmom = new TLegend(0.70,0.80,0.90,0.93);
  lmom->SetTextSize(0.03);
  lmom->SetBorderSize(0);
  lmom->SetHeader(Form("Particle (%s): %1.1f < #eta <
%1.1f",particle.Data(),etamin,etamax),"C");
  lmom->AddEntry(gr,"Real Seed");
  lmom->Draw("same");
}
```

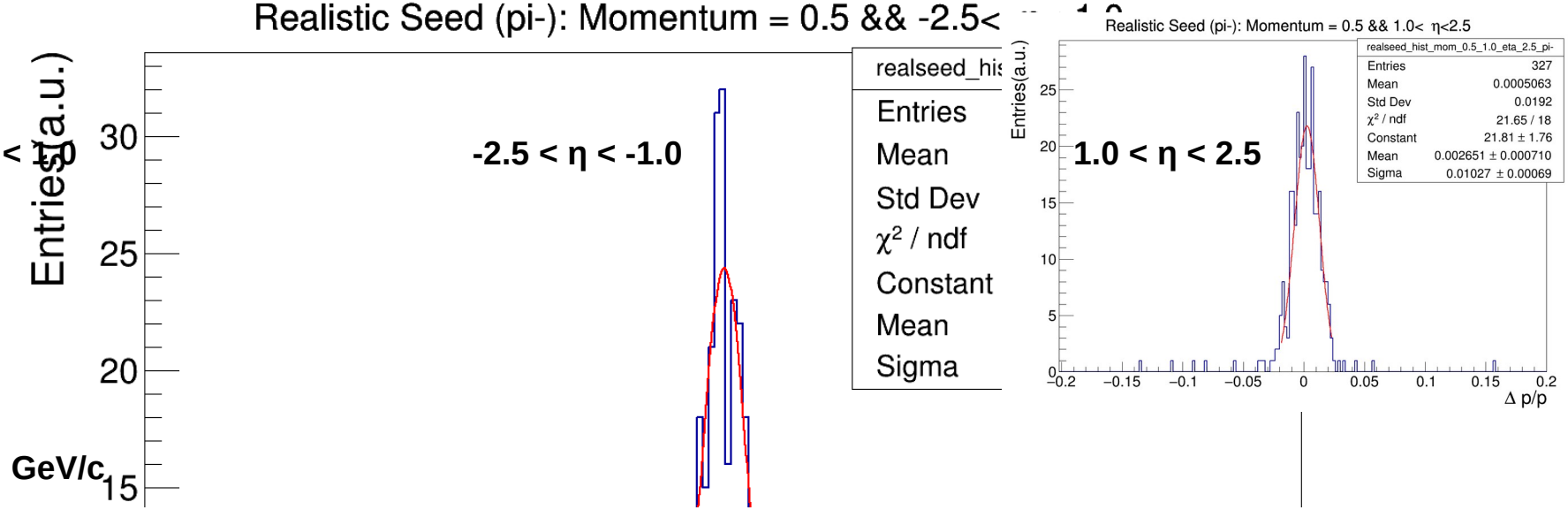
```
void draw_req_Mom(double etamin, double etamax, double xmin=0.,
double xmax=0.)
{

   TF1 *dd4hep_p;
   if (etamin >= -3.5 && etamax <= -2.5) dd4hep_p = new
TF1("dd4hep_p", "TMath::Sqrt((0.1*x)^2+2.0^2)",xmin,xmax);
   else if (etamin >= -2.5 && etamax <= -1.0) dd4hep_p = new
TF1("dd4hep_p", "TMath::Sqrt((0.05*x)^2+1.0^2)",xmin,xmax);
   else if (etamin >= -1.0 && etamax <= 1.0) dd4hep_p = new
TF1("dd4hep_p", "TMath::Sqrt((0.05*x)^2+0.5^2)",xmin,xmax);
   else if (etamin >= 1.0 && etamax <= 2.5) dd4hep_p = new
TF1("dd4hep_p", "TMath::Sqrt((0.05*x)^2+1.0^2)",xmin,xmax);
   else if (etamin >= 2.5 && etamax <= 3.5) dd4hep_p = new
TF1("dd4hep_p", "TMath::Sqrt((0.1*x)^2+2.0^2)",xmin,xmax);
   else return;
   dd4hep_p->SetLineStyle(7);
   dd4hep_p->SetLineColor(kMagenta);
   dd4hep_p->SetLineWidth(3.0);
   dd4hep_p->Draw("same");

  TLegend *l= new TLegend(0.70,0.75,0.90,0.80);
  l->SetTextSize(0.03);
  l->SetBorderSize(0);
  l->AddEntry(dd4hep_p,"PWGReq","l");
  l->Draw("same");
 }
```

Momentum Resolutions (Debug Plots)

10k Events

Debug plots for each η and momentum are produced

***p* = 0.5 GeV/c**



***p* = 10. GeV/c**