Experiment-independent HBT analysis

Dariusz Miśkowiec

- intro: what is an HBT analysis
- why experiment independent
- analysis scheme
- experiment independence
- multidimensional histograms
- some results

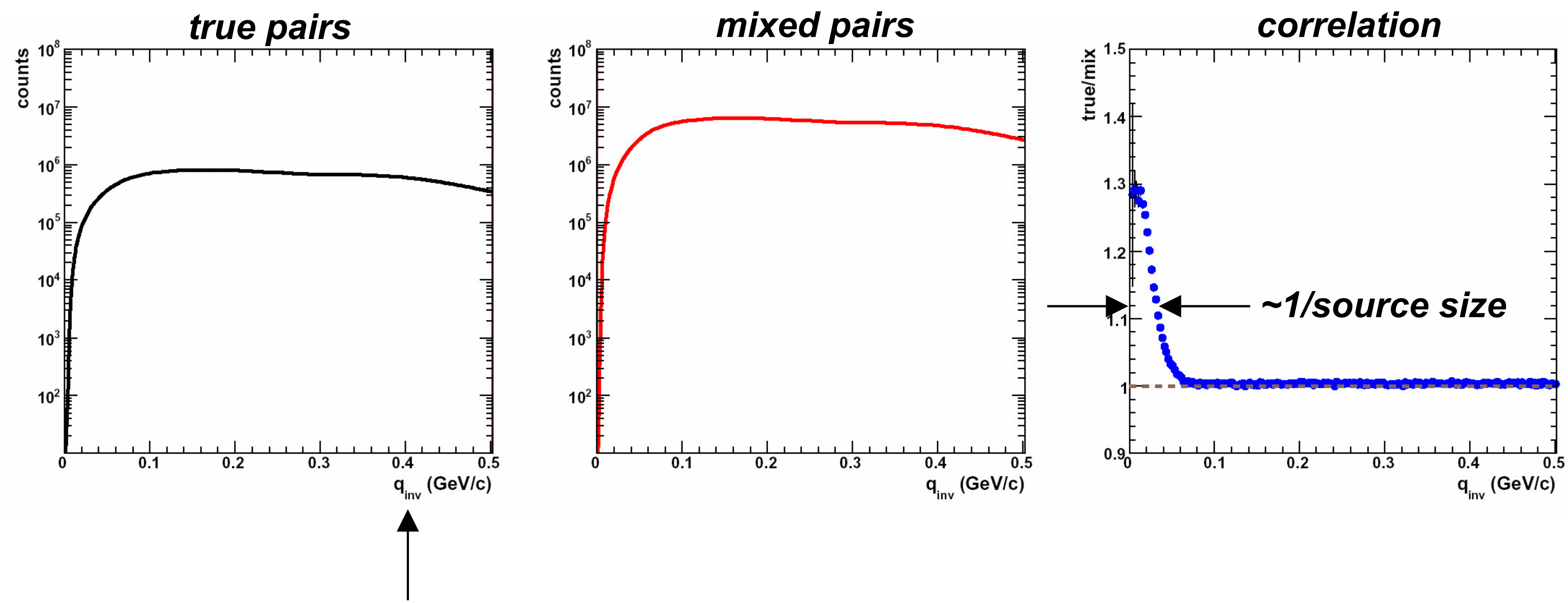
the simplest HBT analysis

D. Antonczyk, thesis intro

- loop over events
- make pi-pi- pairs
- fill a p2-p1 histogram

- mix events
- make pi-pi- pairs
- fill a p2-p1 histogram

- divide tru/mix



p2-p1 in pair c.m.

real analysis

- ***p2-p1 is a vector → correlation function is 3-dim***
- ***pair-y and pair-pt dependent***
- ***finite two-track resolution → suppress close pairs***
- ***correction for Coulomb and momentum resolution***

Why experiment-independent?

typical analysis involves:

- *particle momentum and id*
- *event, track, pair cuts*

} *10%*

- *pairing, event mixing*
- *kinematics*
- *histogramming*
- *projecting, fitting*
- *Coulomb correction*
- *momentum resolution correction*
- *analysis of HBT radii etc.*

} *90%*

→ *Advantage 1: save work*

Why experiment-independent?

Switching between experiments becomes easy, so...

- ***a new method proposed within one experiment can be immediately applied to others***
- ***a new analysis can be tested on old data, for which one knows what should come out***

→ Advantage 2: cross-semination and cross-checking

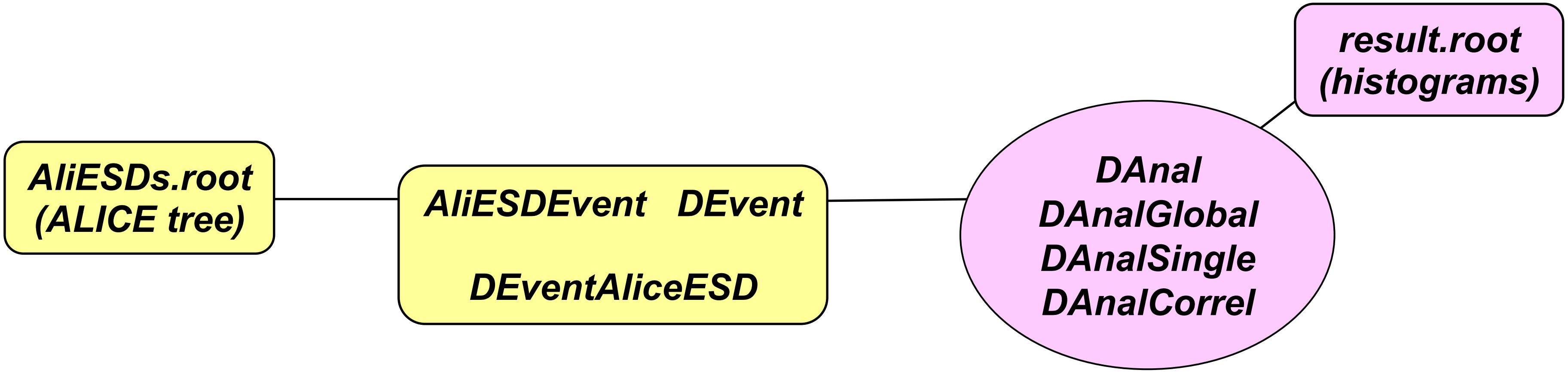
Why experiment-independent?

Comparing between experiments gets easier because

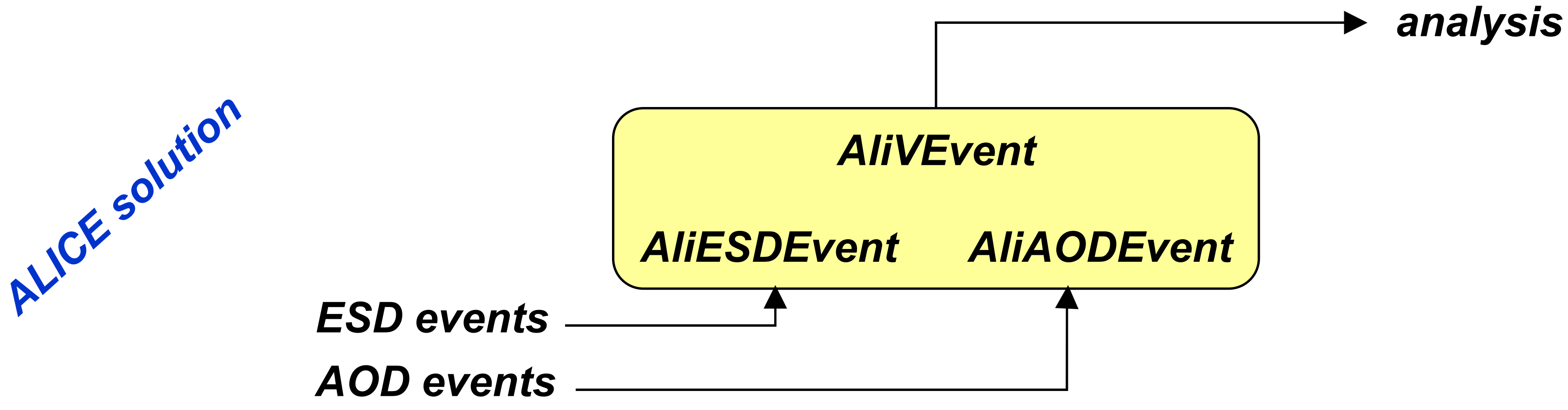
- ***same variables***
- ***same representations of results***
- ***same analysis bugs***

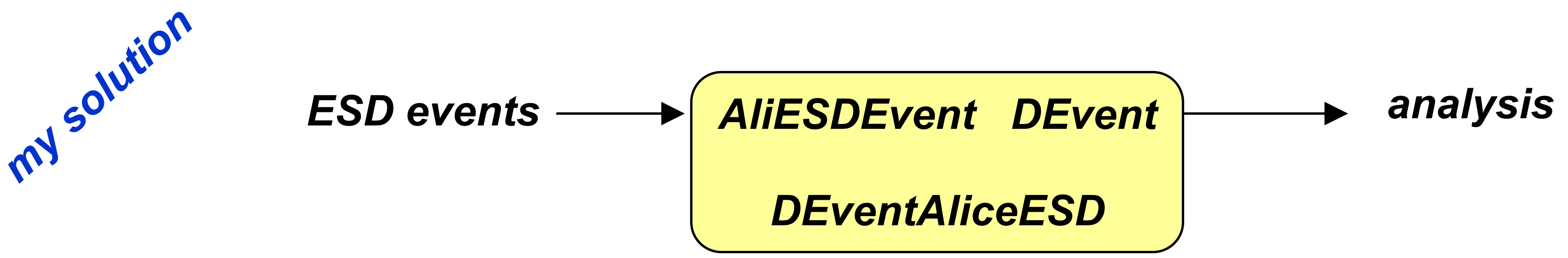
→ Advantage 3: easier comparison

analysis scheme

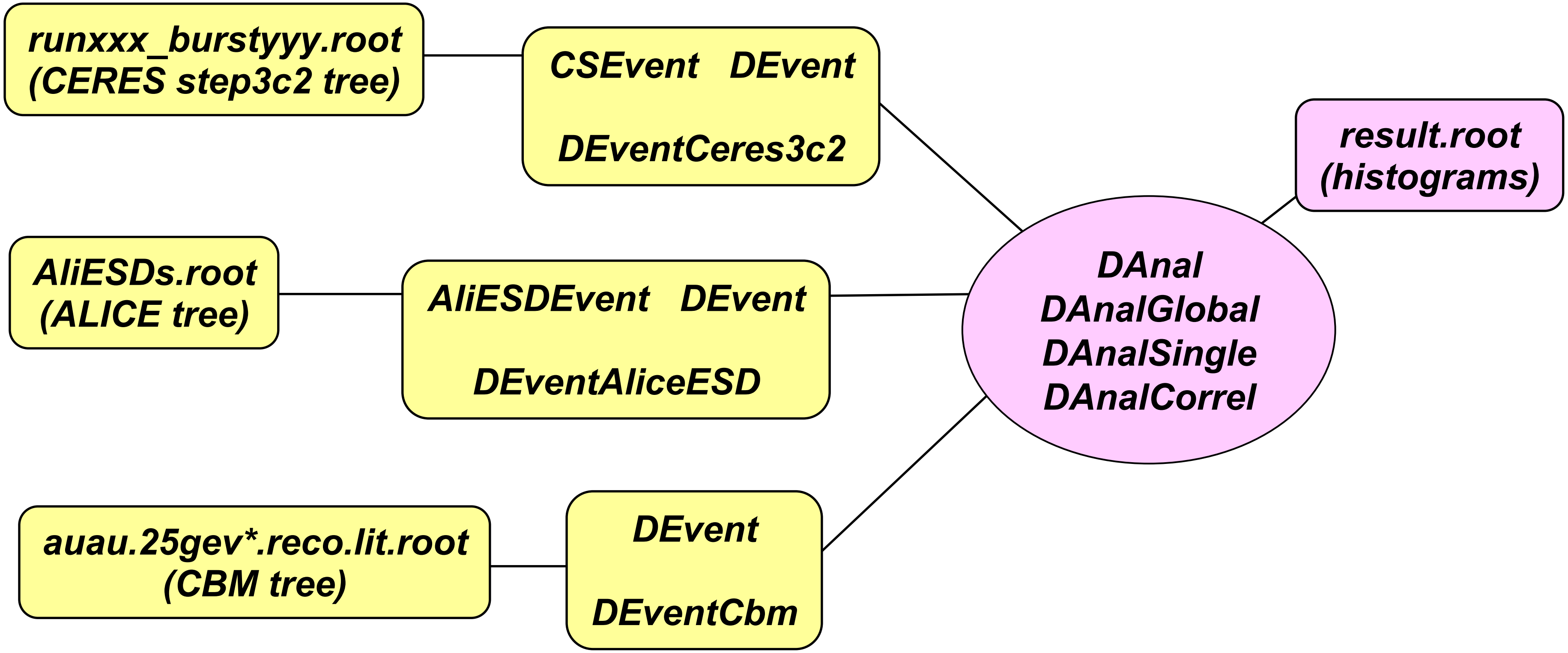


event format independence idea inspired by Alice way handling ESD/AOD

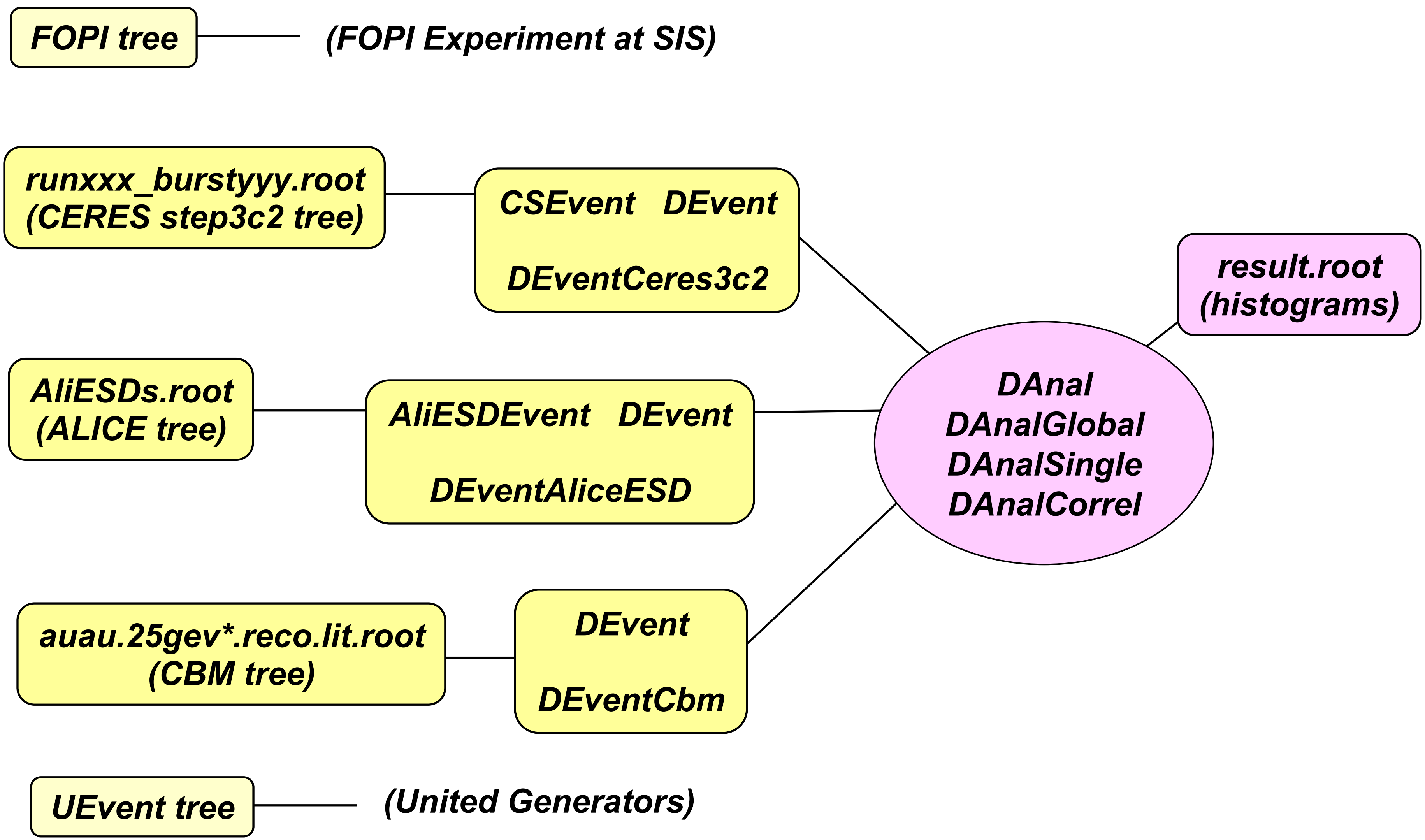




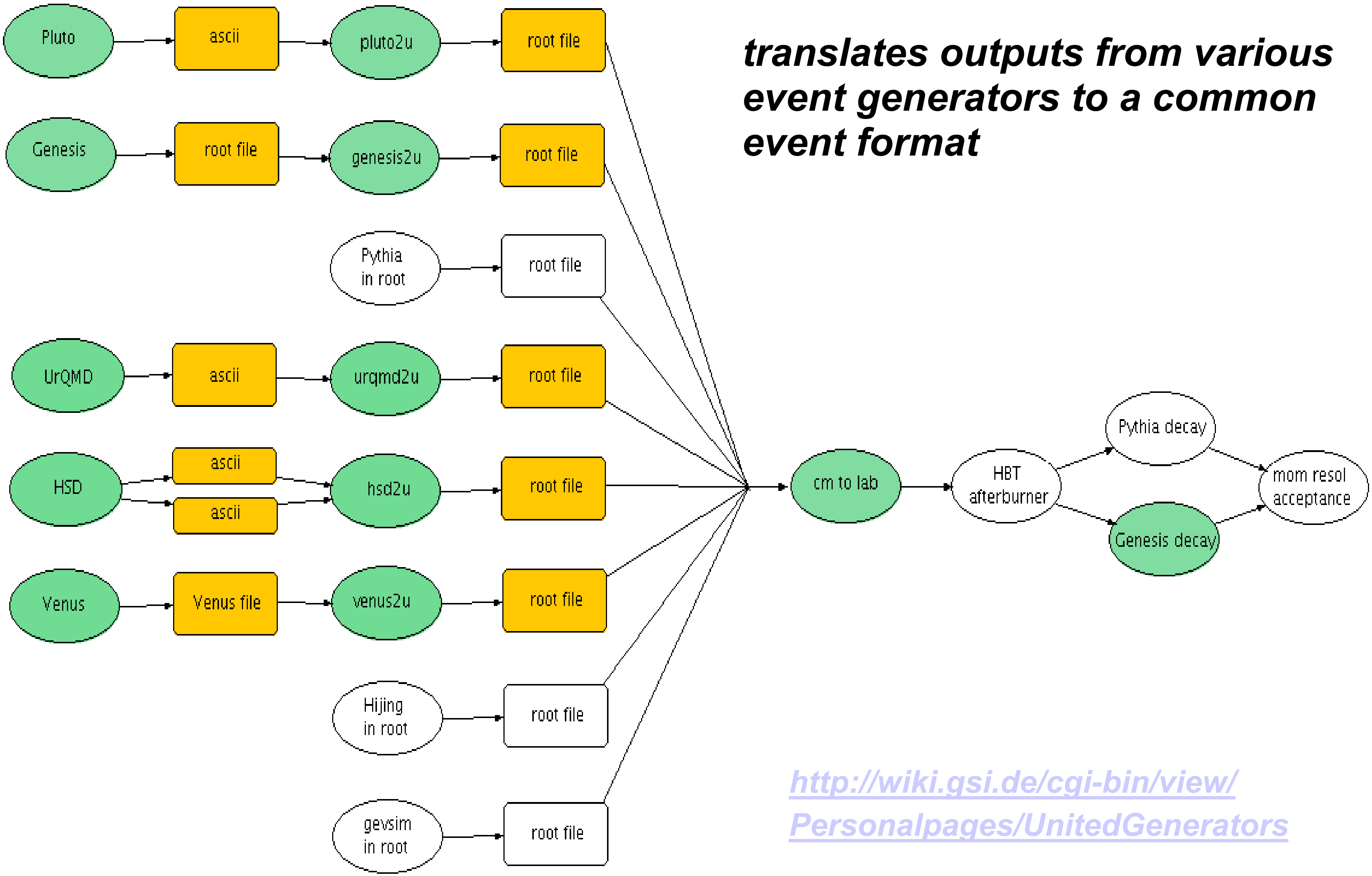
can at present handle ALICE, CERES, and CBM data



will be extended to some others



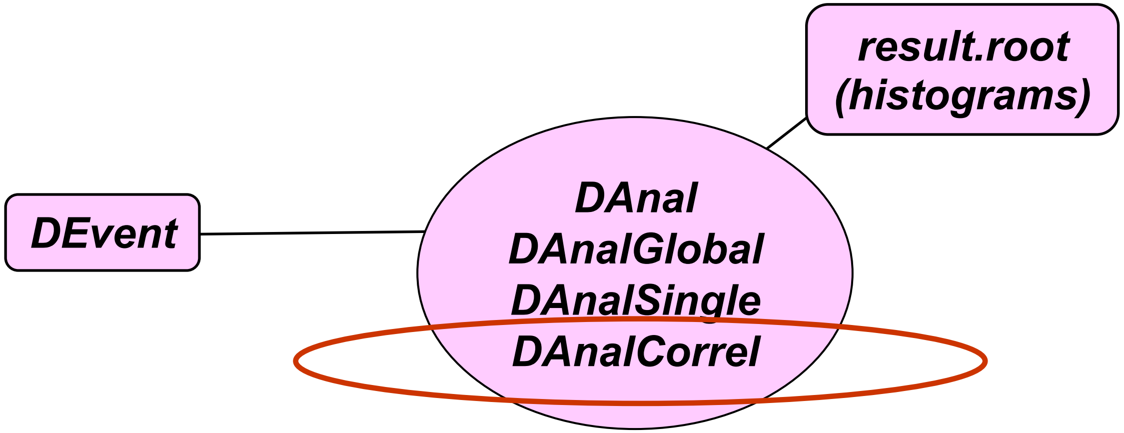
United Generators project at GSI, 2005-2008



translates outputs from various event generators to a common event format

http://wiki.gsi.de/cgi-bin/view/Personalpages/UnitedGenerators

analysis part



pair loop and histogramming in DAnalCorrel

```
// loop over pairs

for (int i=0; i<ev0->NParticles(); i++) {
  if (!ev0->ParticleGood(i,fPid0)) continue;
  for (int j=0; j<ev1->NParticles(); j++) {
    if (ev0==ev1 && phirot==0 && j==i) continue;
    if (ev0==ev1 && phirot==0 && j<i && fPid0==fPid1 ) continue;
    if (!ev1->ParticleGood(j,fPid1)) continue;
    fPa.Set0(fMass0,ev0->ParticleP(i),ev0->ParticleTheta(i),ev0->ParticlePhi(i));
    fPa.Set1(fMass1,ev1->ParticleP(j),ev1->ParticleTheta(j),ev1->ParticlePhi(j)+phirot);
    if (ev0==ev1 && phirot==0 && fPid0==fPid1 && ran.Rndm()>=0.5) fPa.Swap();
    fPa.CalcLAB();
    fPa.bbeta = fPa.beta; // CM will mean pair c.m.s.
    fPa.CalcCM();
    double phi = TVector2::Phi_mpi_pi(fPa->p.Phi()-rpphi);
    fPair->Fill(flag,                    // 0 for tru, 1 for mix, 2 for rot
                cent,                    // centrality
                fPa->p.Rapidity(),       // pair rapidity
                phi,                     // pair phi wrt reaction plane
                fPa->p.Pt()/2.0,         // half of pair pt
                fPa->QCMTheta(),         // polar angle of Q
                fPa->QCMPhi(),           // azimuthal angle of Q
                fPa->QCM(),              // |p2-p1| in c.m.s.
                1.0);                    // weigth
  }
}
```

Resulting correlation functions are stored in an 8-dimensional histogram

tru/mix/rot	3 bins
centrality	5 bins
pair y	4 bins
pair phi wrt. event plane	8 bins
pair pt	7 bins
q-polar angle	8 bins
q-azimuthal angle	16 bins
q-magnitude	64 bins

P. Danielewicz's dream: two-particle tensor!
Complete info, like pair ntuple -- but binned.

27,525,120 bins, 210 MB. Too big?

for comparison: traditional way of histogramming

```
for (fl=0; fl<2; fl++) {
for (pp=0; pp<5; pp++) {
for (th=0; th<3; th++) {
for (ph=0; ph<1; ph++) {
  sprintf(myst,"%s%d%d%d%s%s_%s",trmi[fl],pp,th,ph,pnam[i0],pnam[i1],"qinv");
  h1[fl][pp][th][ph][i0][i1][0] =  new TH1D(myst,myst,500,0,5);
  sprintf(myst,"%s%d%d%d%s%s_%s",trmi[fl],pp,th,ph,pnam[i0],pnam[i1],"minv");
  h1[fl][pp][th][ph][i0][i1][1] =  new TH1D(myst,myst,500,0,5);
  sprintf(myst,"%s%d%d%d%s%s_%s",trmi[fl],pp,th,ph,pnam[i0],pnam[i1],"dtheta:dphi");
  h2[fl][pp][th][ph][i0][i1][0] =  new TH2D(myst,myst,50,-250,250,50,-50,50);
  sprintf(myst,"%s%d%d%d%s%s_%s",trmi[fl],pp,th,ph,pnam[i0],pnam[i1],"qper:qpar");
  h2[fl][pp][th][ph][i0][i1][1] =  new TH2D(myst,myst,20,0.0,0.2,40,-0.2,0.2);
  sprintf(myst,"%s%d%d%d%s%s_%s",trmi[fl],pp,th,ph,pnam[i0],pnam[i1],"qout:qside:qlong");
  h3[fl][pp][th][ph][i0][i1][0] =  new TH3D(myst,myst,30,-0.15,0.15,30,-0.15,0.15,30,-0.15,0.15);
}
}
}
}
```

array of histograms, same size, less convenient handling

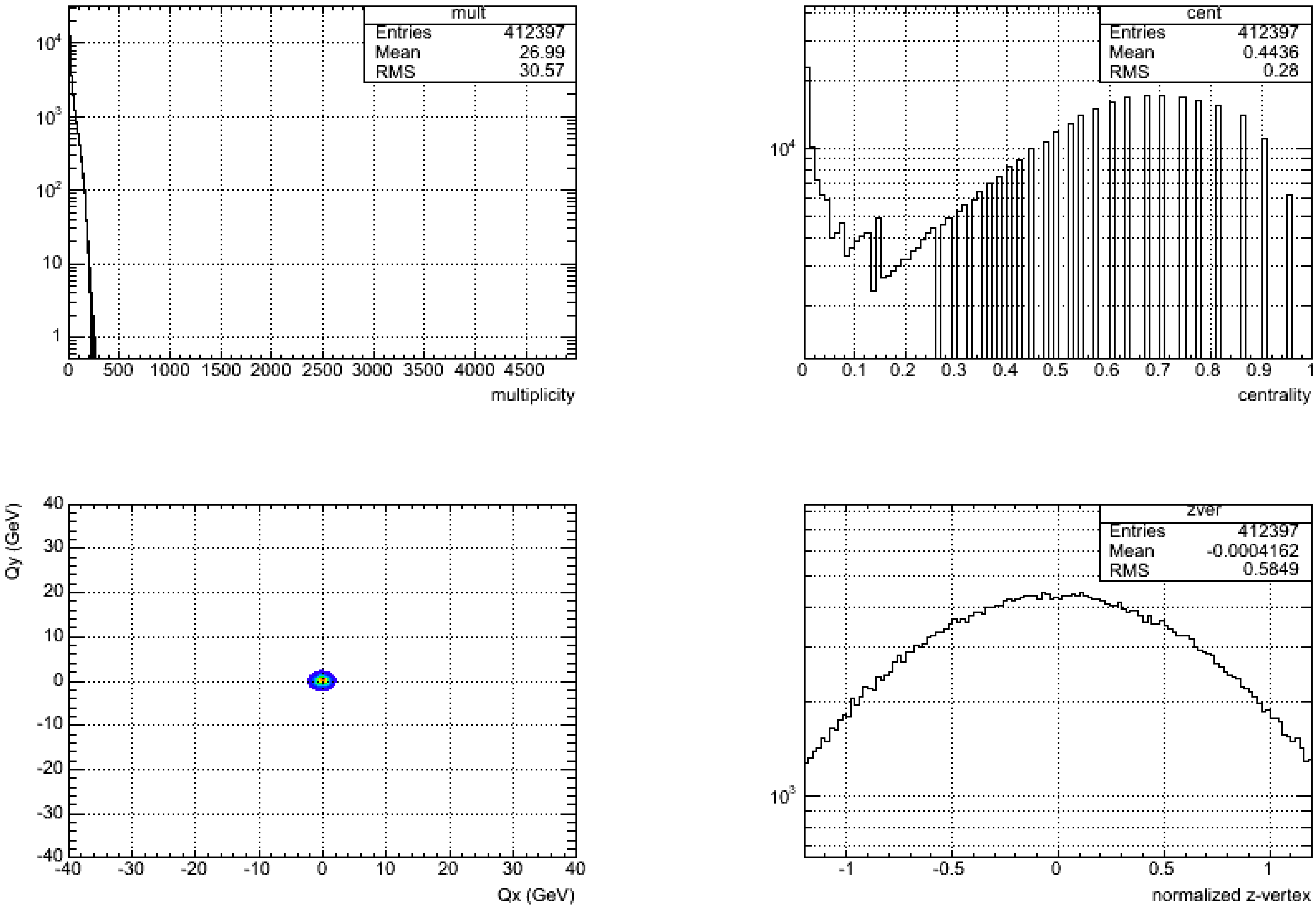
some results

CERES/SPS	*PbAu*	*158 AGeV*	*exp*	*24k*
ALICE/LHC	*pp*	*sqrt(s)=10 TeV*	*sim*	*400k*
CBM/SIS300	*AuAu*	*25 AGeV*	*sim*	*44k*

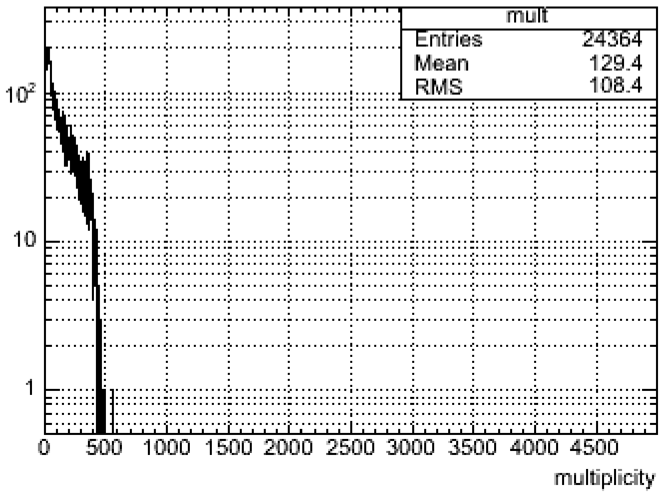
disclaimer:

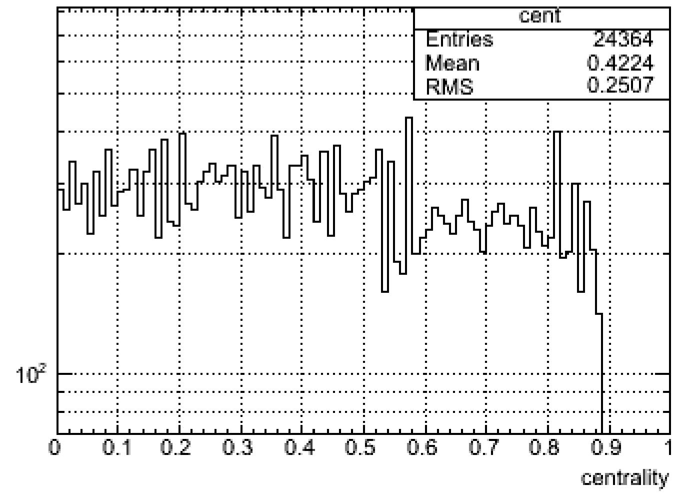
- ***no (serious) particle identification***
- ***no two-track cut***

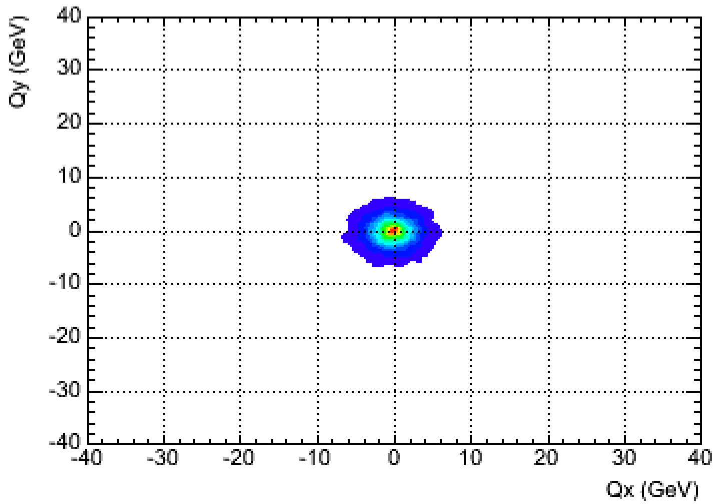
ALICE global variables

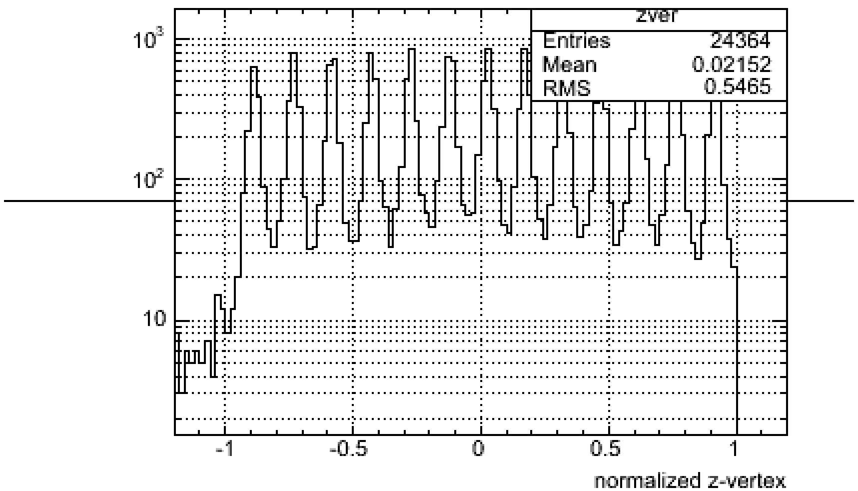


CERES global variables

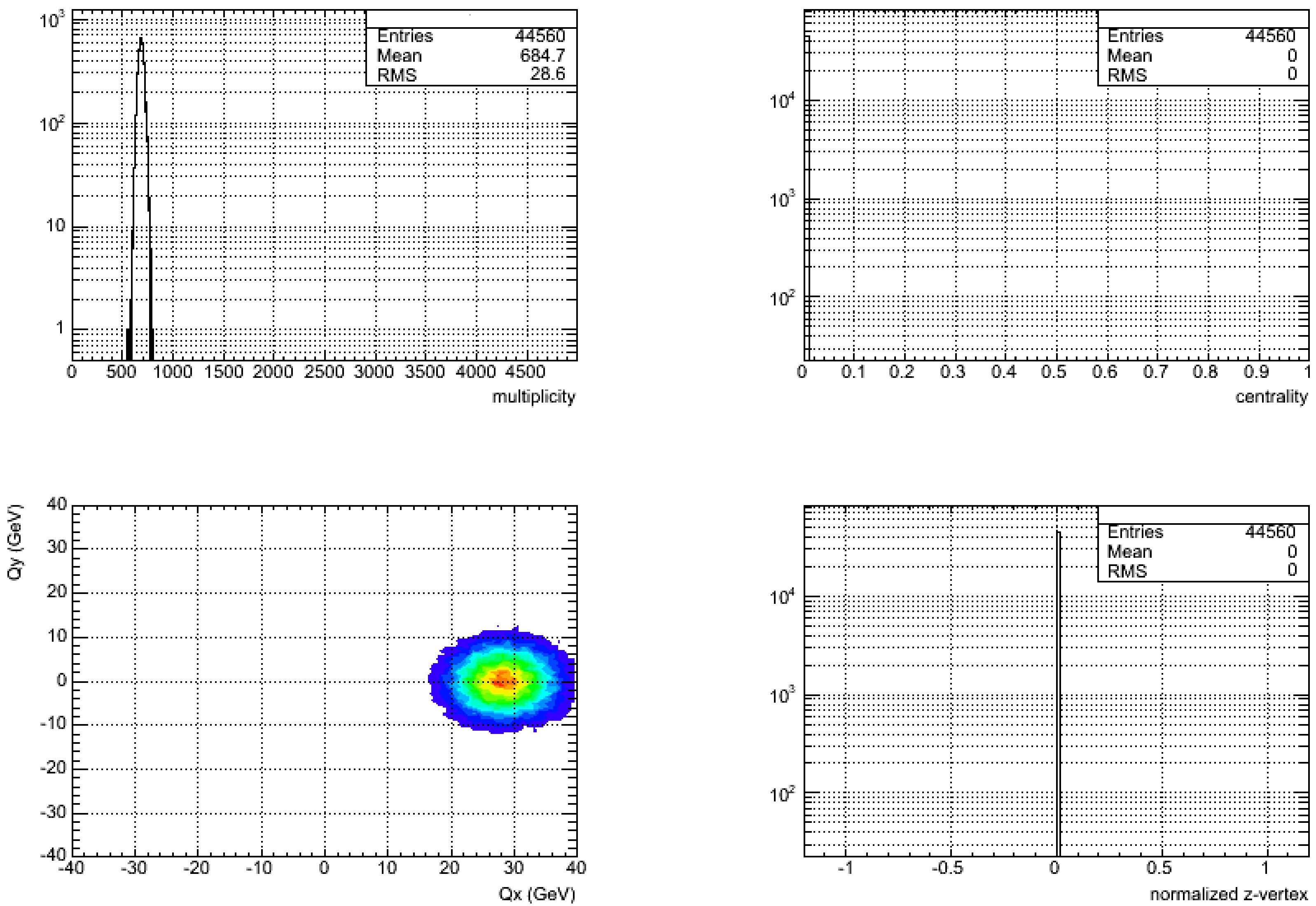




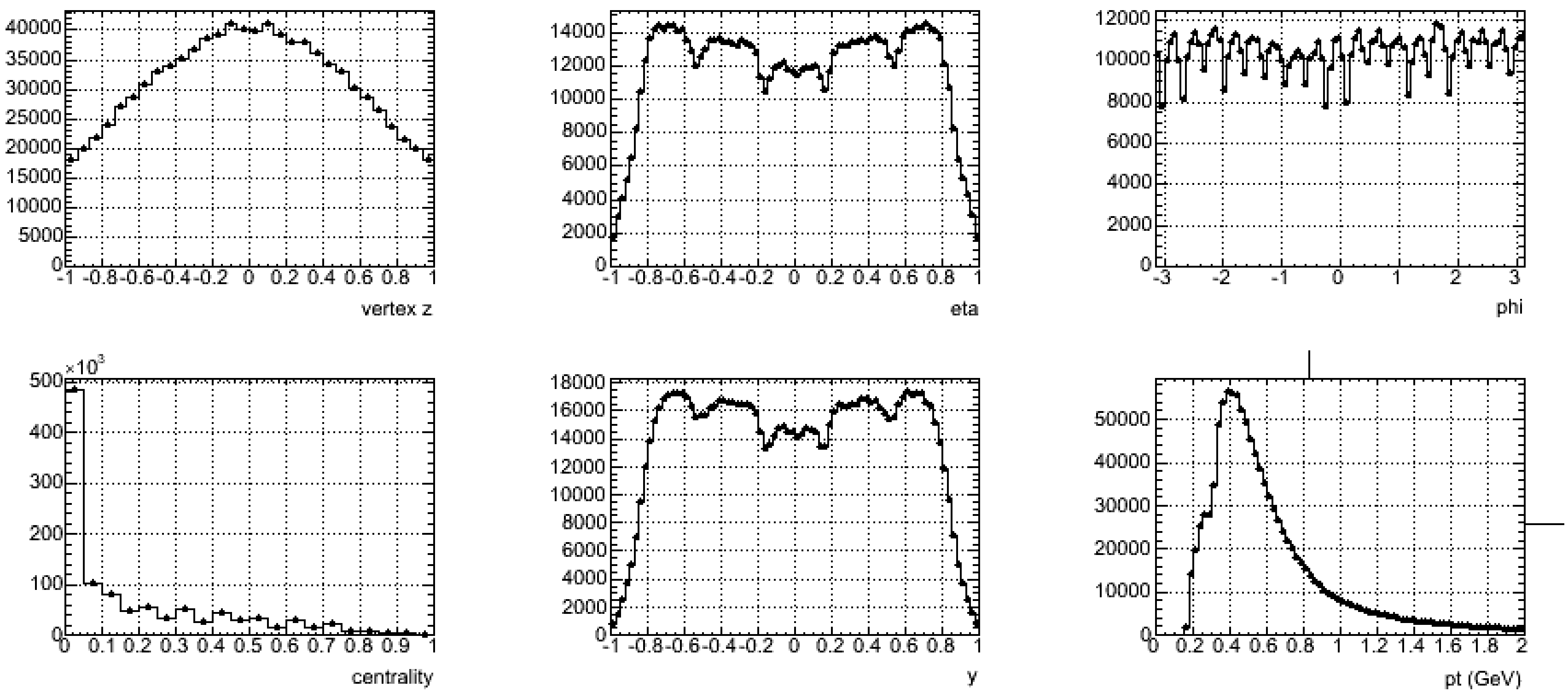




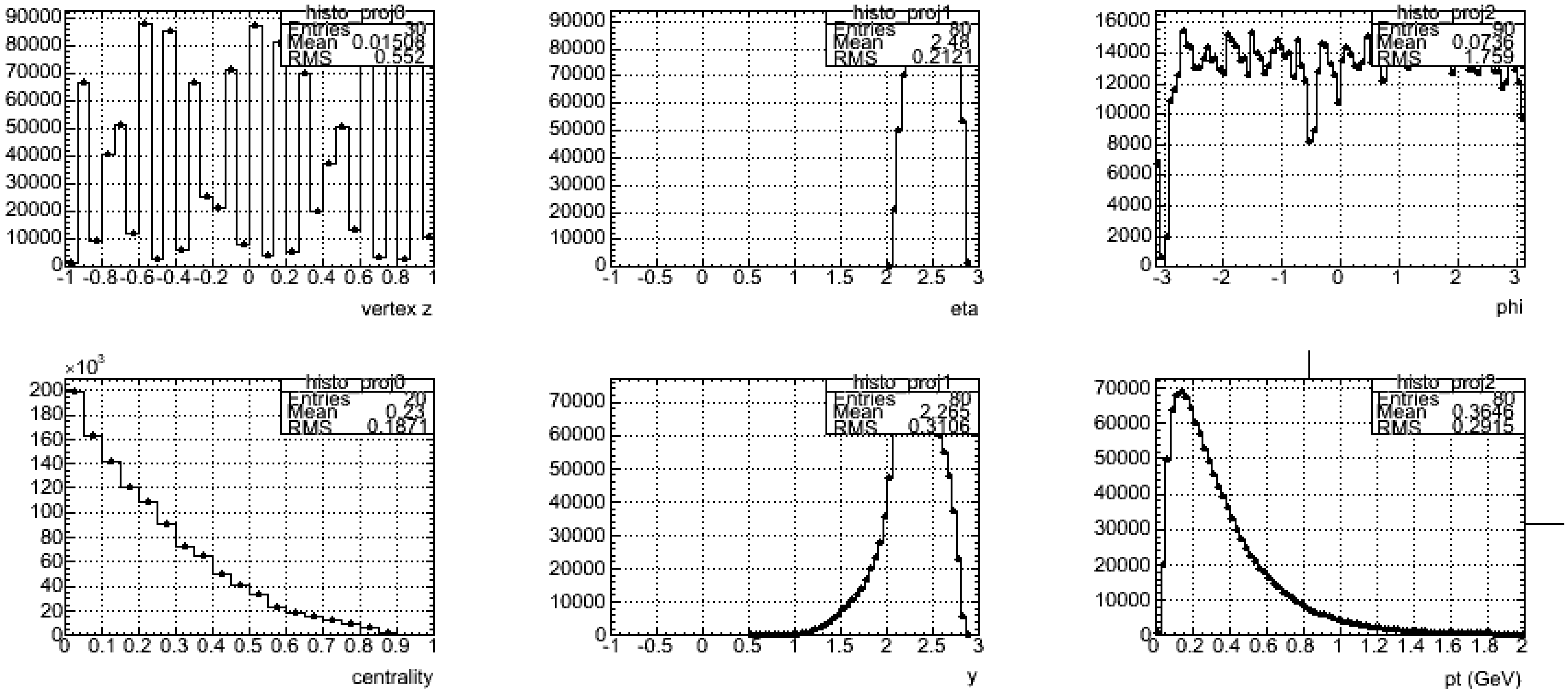
CBM global variables



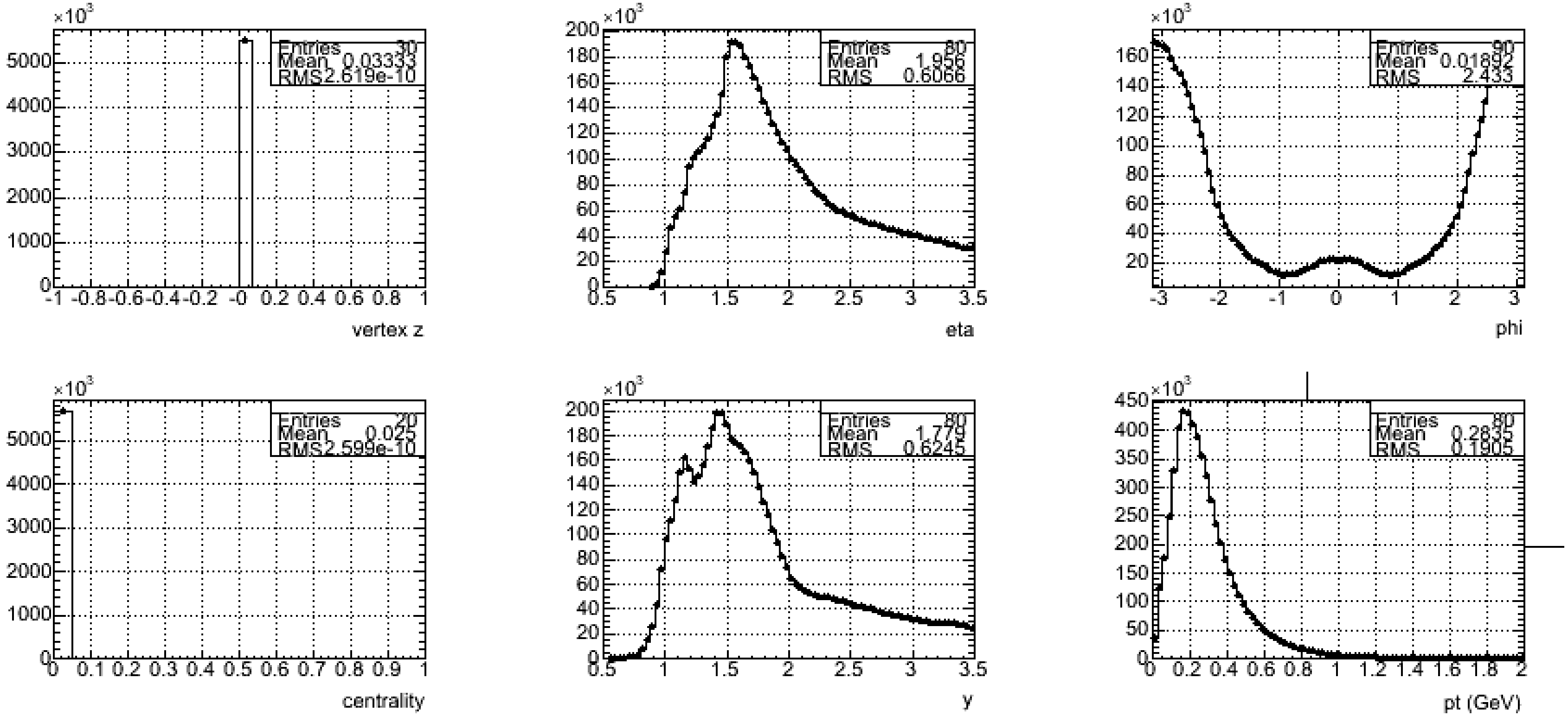
ALICE singles



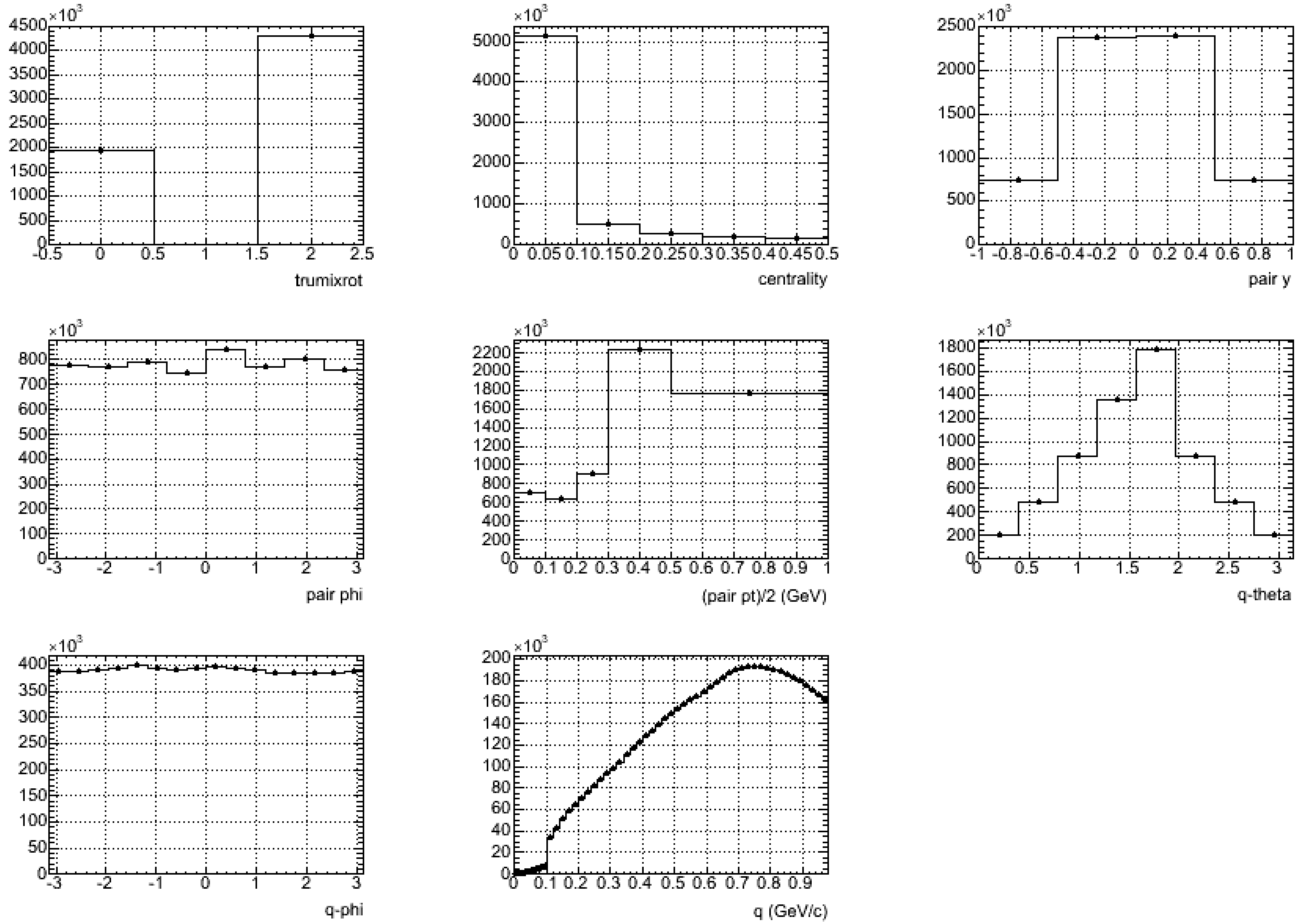
CERES singles



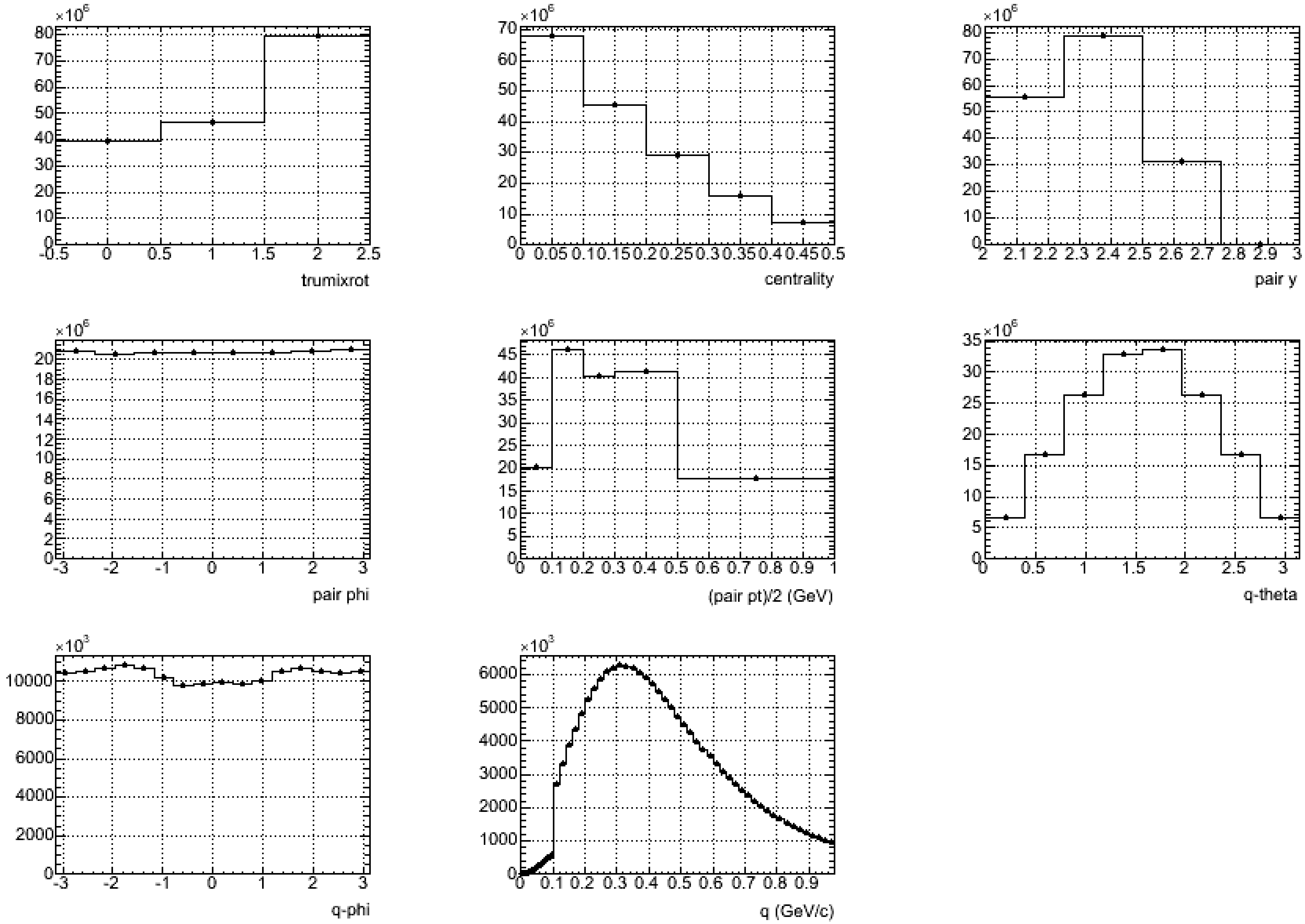
CBM singles


×10³
Entries 30
Mean 0.03333
RMS2.619e-10
vertex z
×10³
Entries 80
Mean 1.956
RMS 0.6066
eta
×10³
Entries 90
Mean 0.01892
RMS 2.433
phi
×10³
Entries 20
Mean 0.025
RMS2.599e-10
centrality
×10³
Entries 80
Mean 1.779
RMS 0.6245
y
×10³
Entries 80
Mean 0.2835
RMS 0.1905
pt (GeV)

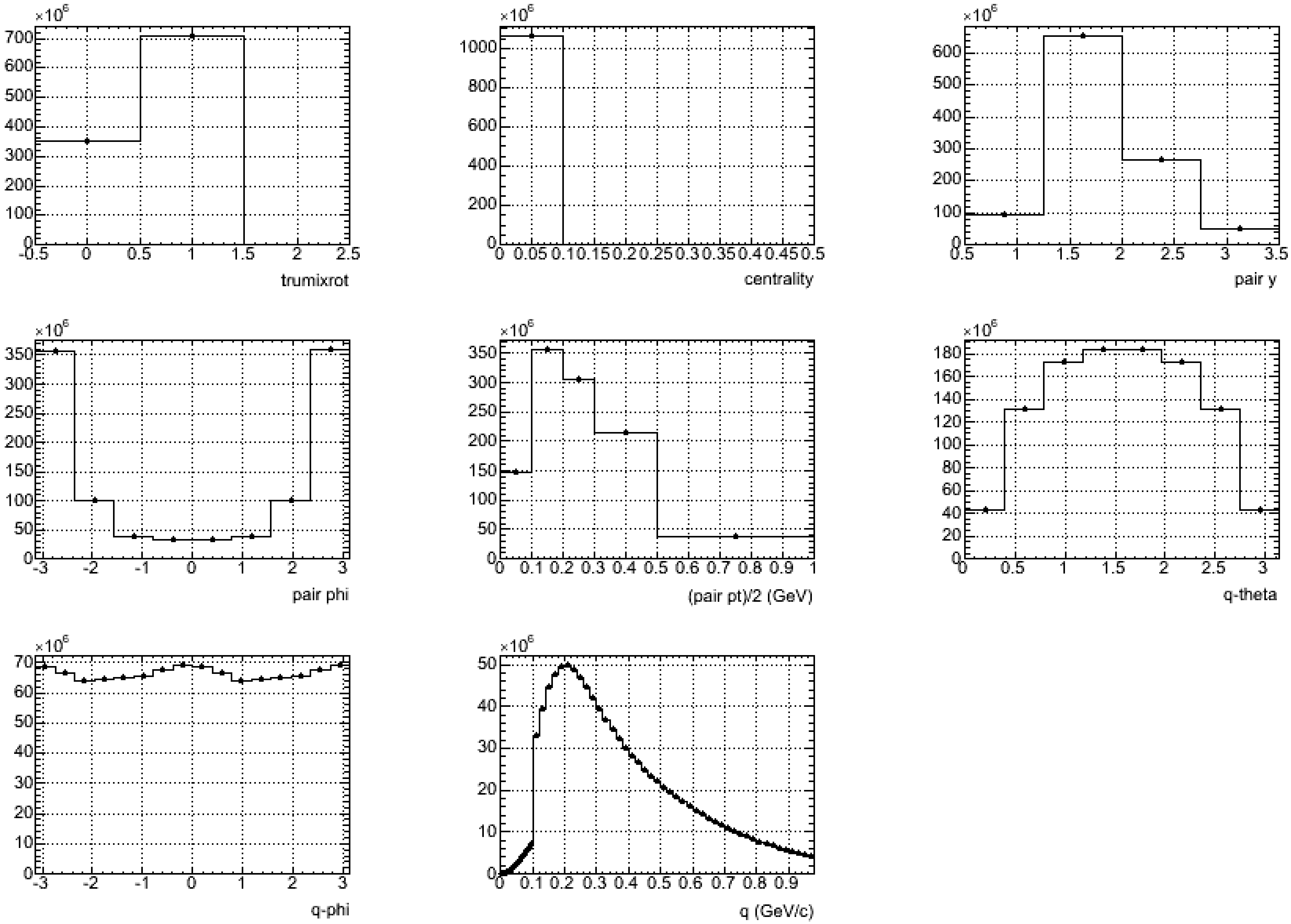
ALICE correlations (projections)



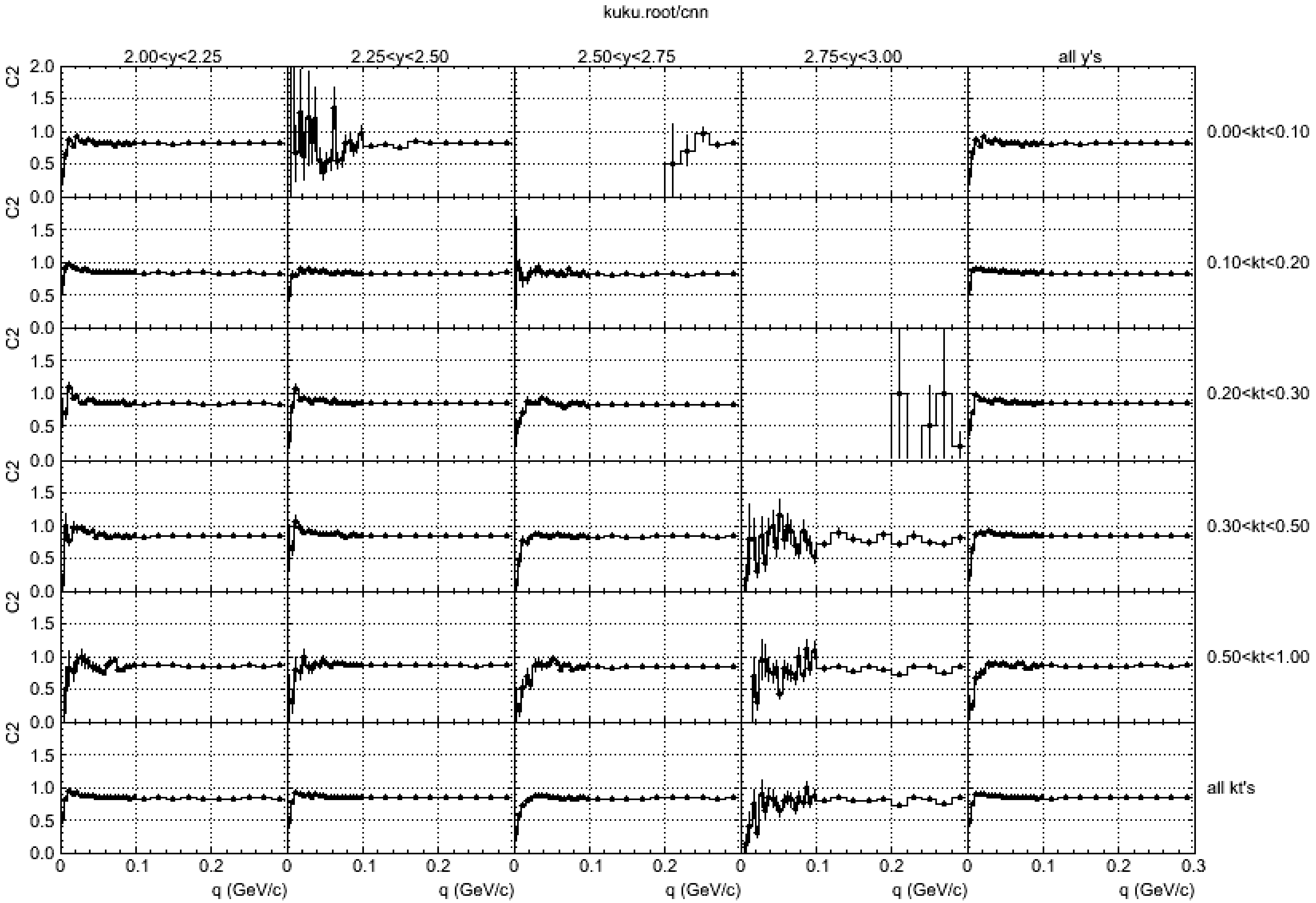
CERES correlations (projections)



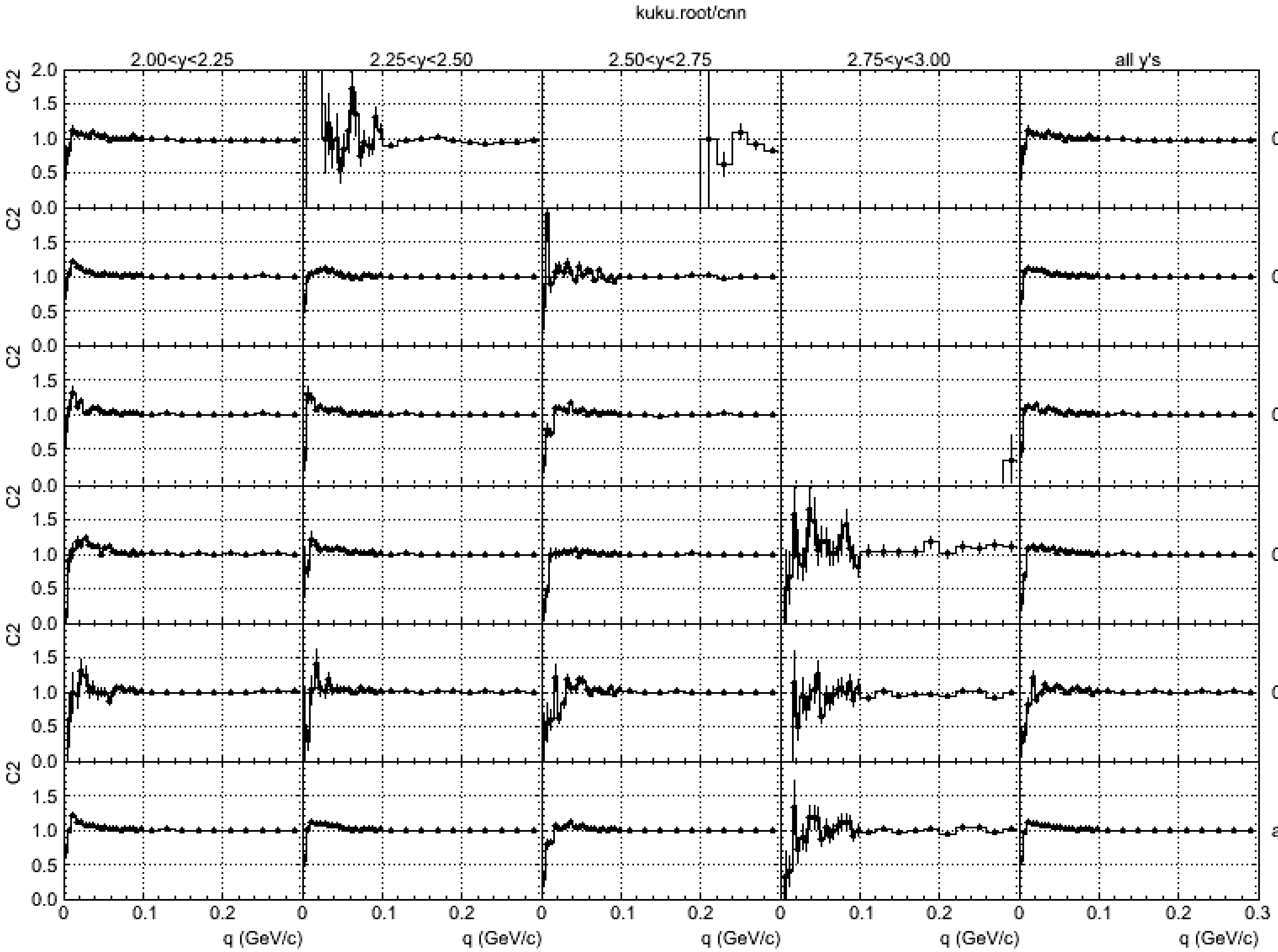
CBM correlations (projections)



CERES correlations (denominator by event mixing)

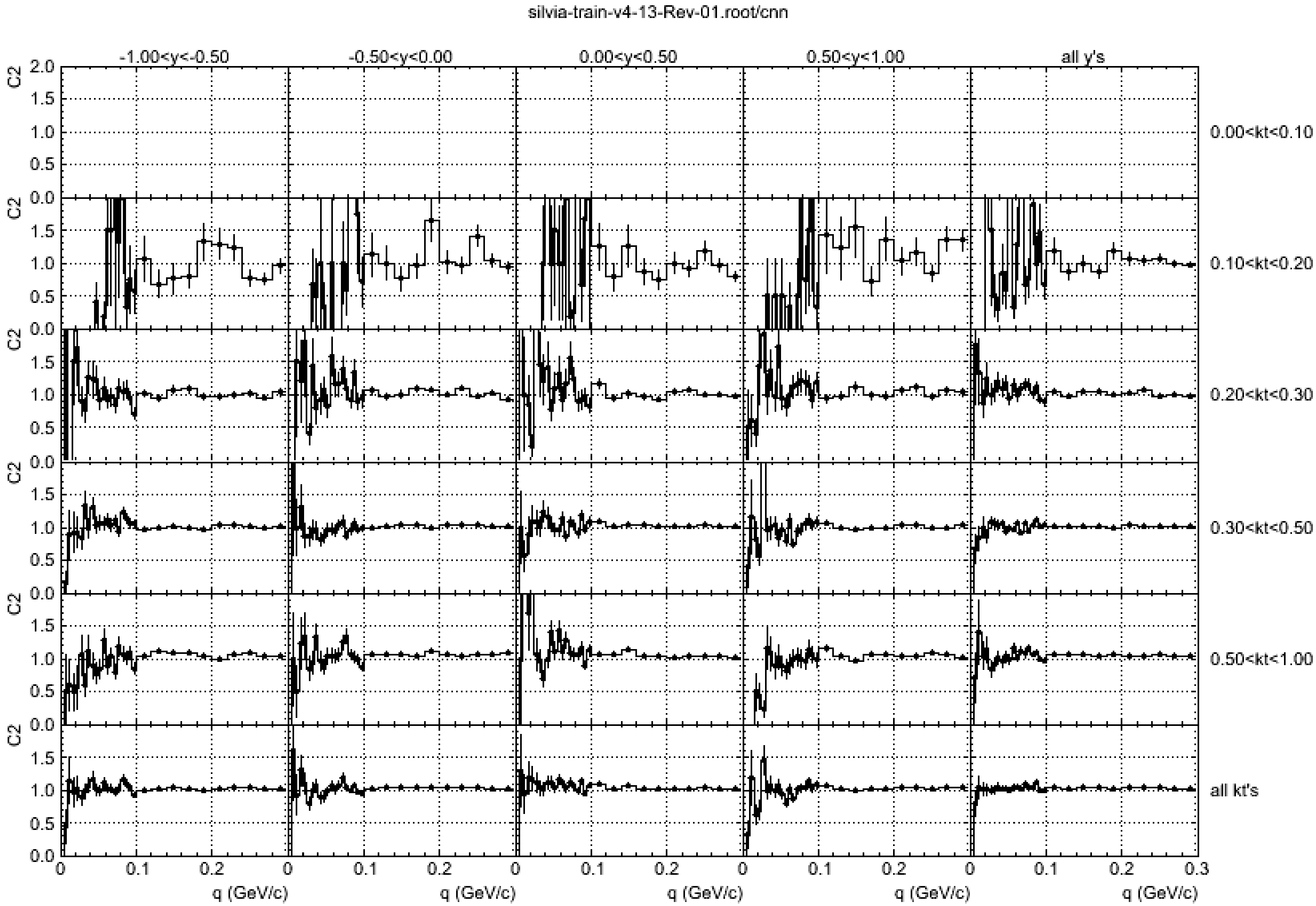


CERES correlations (denominator by rotating)

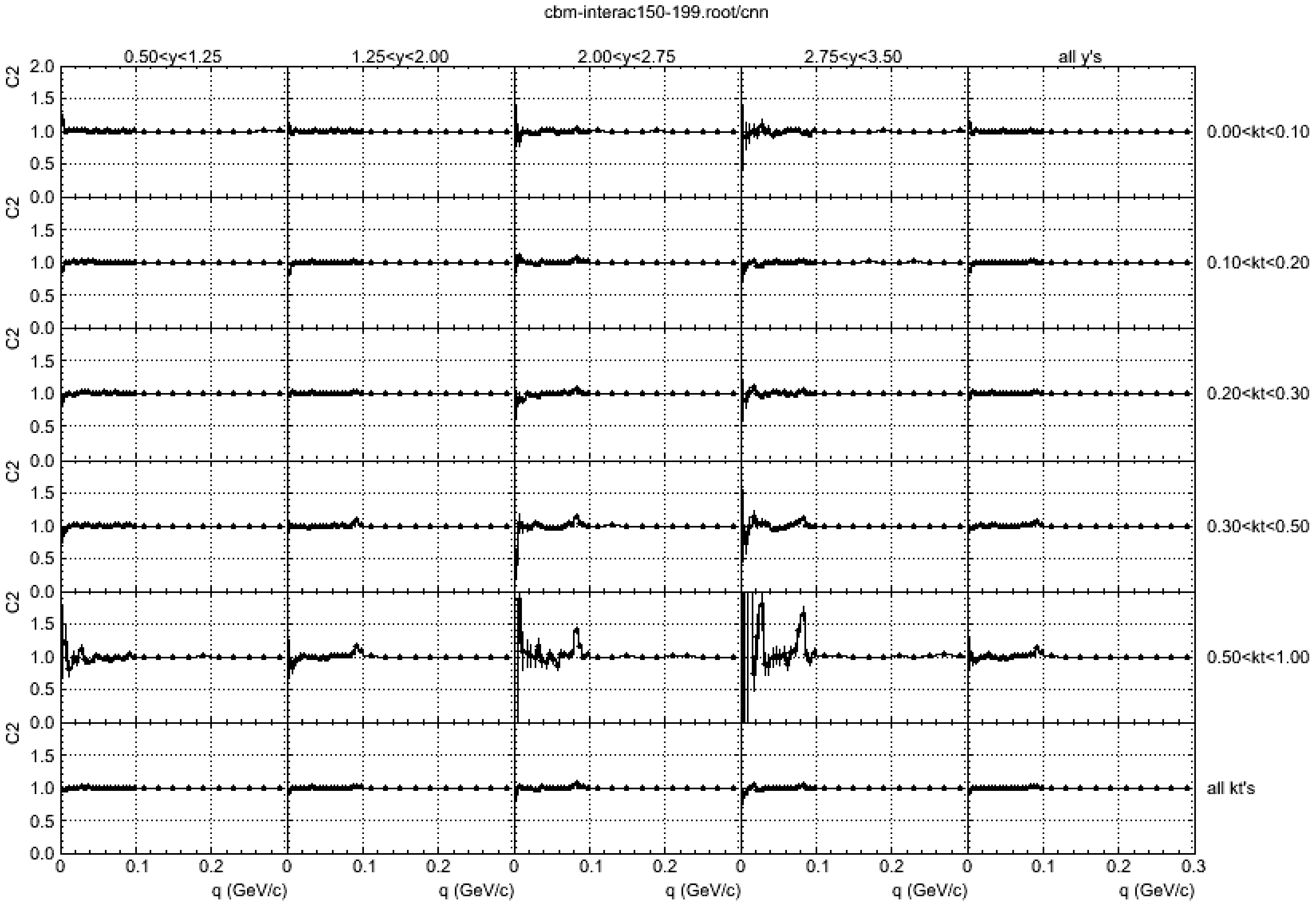


ALICE correlations (denominator by rotating)

(1.2 M event, actually)



CBM correlations (denominator by event mixing)



summary: main features of this analysis

- ***experiment independence → easy comparison***
- ***multidimensional histograms → efficient storage of results***
- ***simplicity → easy debugging, good for quick start***

backup

what I like most in this analysis: it is short

- ***analysis part:***	***1397 lines of code***
- ***ceres3c2 interface:***	***143 lines of code***
- ***alice ESD interface:***	***99 lines of code***
+ analysis task	***130 lines of code***
- ***cbm interface:***	***131 lines of code***

--

1900 lines of code

DEventAliceESD

my solution


```
class DEventAliceESD : public DEvent, public AliESDEvent {

 public:
               DEventAliceESD();
  virtual     ~DEventAliceESD();
  void         AttachTree(TTree *tr) {ReadFromTree(tr);}
  void         CopyFrom(const AliESDEvent& source); // shallow import of event
  Bool_t       Good() const;
  Double_t     Centrality() {return 0.9999*exp(-NParticles()/20.0);} // OK for pp
  void         RP(Double_t &qx, Double_t &qy) const {qx=0; qy=0;}
  Double_t     Zver() const {return AliESDEvent::GetPrimaryVertex()->GetZv()/5.0;}
  Int_t        NParticles() const {return AliESDEvent::GetNumberOfTracks();}

  Bool_t       ParticleGood(Int_t i, Int_t pidi) const;
  Int_t        ParticleSign(Int_t i) const {return 1; // to be fixed}
  Double_t     ParticleP(Int_t i)     const {return AliESDEvent::GetTrack(i)->GetConstrainedParam()->P();}
  Double_t     ParticlePhi(Int_t i)   const {return AliESDEvent::GetTrack(i)->GetConstrainedParam()->Phi();}
  Double_t     ParticleTheta(Int_t i) const {return AliESDEvent::GetTrack(i)->GetConstrainedParam()->Theta();}

  ClassDef(DEventAliceESD,0)
```

DAnalCorrel.h

```
class DAnalCorrel : public DAnal {

 public:
  DAnalCorrel(Char_t *nam="correl", Int_t pid0=0, Int_t pid1=0);  // constructor
  virtual ~DAnalCorrel();                                          // destructor
  // process one (tru) or two (mix) events
  void Process(Int_t tmr, DEvent *ev0, DEvent *ev1, Double_t phirot);

 protected:
  Int_t    fPid0;                           // particle species 0
  Int_t    fPid1;                           // particle species 1
  Double_t fMass0;                          // mass 0
  Double_t fMass1;                          // mass 1
  DPair    fPa;                             // pair buffer for calculations
  DHN*     fPair;                           // pair multi-histogram
  void     Fill(Int_t flag, Double_t cent, Double_t rpphi, DPair *pa);
                                            // fill pair histograms
  ClassDef(DAnalCorrel,1)
```

8-dimensional histogram

multidimensional histogram class DHN

424 lines of code

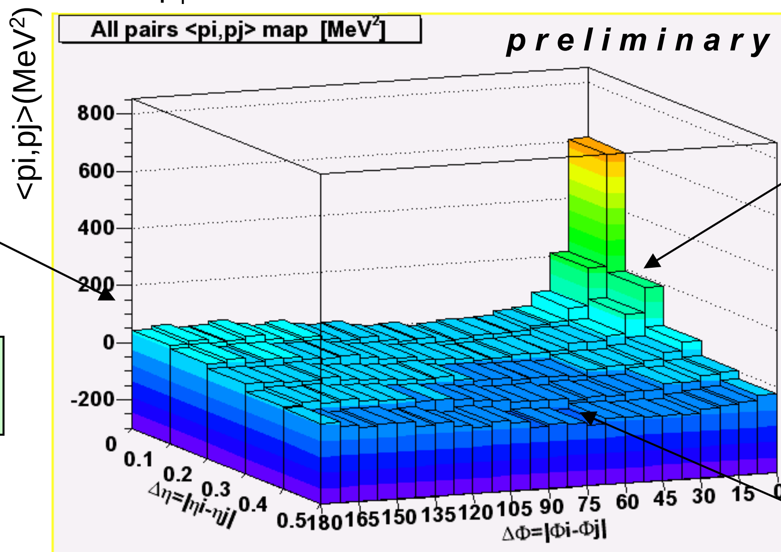
```
class DHN : public TH1D {

 public:

  DHN();                                          // default constructor
  DHN(Char_t *nam, Int_t ndim, TAxis **ax);  // constructor from scratch
  DHN(Char_t *filename, Char_t *name);       // constructor from file
  virtual ~DHN();                                 // destructor
  Int_t GetNdim()                                 {return fNdim;}
  TAxis *GetAxis(Int_t i)                         {return &fAxis[i];}
  void Fill(Double_t *xx, Double_t y=1);          // fill histo
  void Fill(Double_t x0=0, Double_t x1=0,
            Double_t x2=0, Double_t x3=0,
            Double_t x4=0, Double_t x5=0,
            Double_t x6=0, Double_t x7=0,
            Double_t x8=0, Double_t x9=0,
            Double_t x10=0) {                     // fill histo; fNdim-th arg is weight
    Double_t xx[fMaxNdim+1] = {x0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10};
    Fill(xx,xx[fNdim]);}
  Int_t Write();                                  // save histo and axis on file
  Int_t Write(const char *, Int_t, Int_t)         {return Write();}
  // project along (integrate over) one axis
  DHN  *ProjectAlong(char *nam, Int_t dim, Int_t first=-1, Int_t last=-1);
  // project on 1-dim histogram
  TH1D *ProjectOn(char *nam, Int_t dim, Int_t *first=0, Int_t *last=0);
  // project on 1-dim histogram
  TH1D *ProjectOn(char *nam, Int_t dim, Double_t *first, Double_t *last);
  // project on 2-dim histogram
  TH2D *ProjectOn(char *nam, Int_t dim0, Int_t dim1, Int_t *first=0, Int_t *last=0);
```

pt fluctuation intro, CERES data

Pb+Au at 158 GeV per nucleon

Georgios Tsiledakis

0.1 < p_T < 1.5 GeV/c



away-side correlations

elliptic flow, jets?

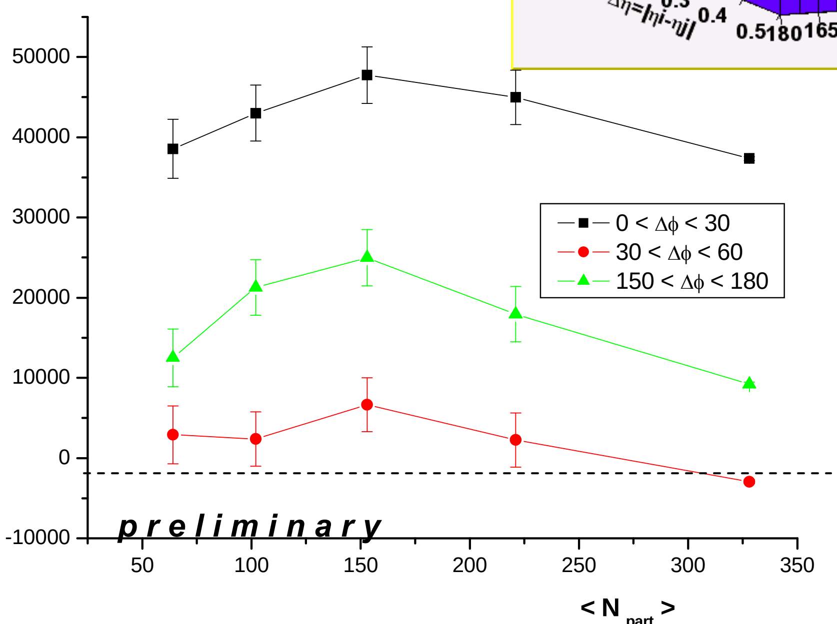
short range correlations

confined to Q_{INV}<70 MeV
narrower and weaker for unlike pairs → HBT and Coulomb?

rich structure → averaging over Δφ and Δη is not good

decline with Δη

reproduced with event mixing
trivial effect of pt(η) dependence?



short range and away-side correlation produce the observed centrality dependence

region around Δφ = 45° is not affected by the elliptic flow, HBT, Coulomb, jets → look here for the critical point

CBM pt-fluctuations