Antiprotons and Protons Selection

A.Tiseni

Tof Group Bologna

Summary:

Definition of the cut for the selection.
Results 200: runs analyzed

Definition of the cut: Minimum Bias

For the selection of protons and antiprotons there are some “Minimum Bias” Conditions:

•One Particle

• science runtag:

```
HeaderR* header = &(pev->fHeader);
  if((header->RunType>>12)!=0xf) return false;
```

•Live Time (fraction of non busy time) > 0.65

```
  Level1R * trig=pev-> pLevel1(0);
  if( trig->LiveTime <=0.65) return false;
```

• One Tracker Track and fitID > 0

•Downgoing particle with good β: $0.6 < \beta < 1.2$ ($T > 0.2$ GeV)

•Pole and ssa exclusion

•Trigger minimum bias. TOF minimum bias, TRD minimum bias, ECAL minimum bias

Definition of the cut: Minimum Bias

1) Trigger Minimum Bias Selection: 3 layers out of 4 must exceed the High Threeshold.

```
minimum_bias = level1->TofFlag1>=0 &&   level1->TofFlag1<5
```

2) TOF Minimum Bias Selection: 3 layers out of 4 with certain properties of TOF Cluster as a signal both from side n and p.

```
for (int i=7; i<13; i++) if ((cluster->Status>>i)&1==1) good_c=false;
                         if ((cluster->Status>>2)&1==1) good_c=false;
                         if ((cluster->Status>>4)&1==1) good_c=false;

                                  if(good_c) goodlayer[layer]=true;

minimum_bias=((goodlayer[0] && goodlayer[1] && goodlayer[2]) ||
                         (goodlayer[0] && goodlayer[1] && goodlayer[3]) ||
                         (goodlayer[0] && goodlayer[2] && goodlayer[3]) ||
                        (goodlayer[1] && goodlayer[2] && goodlayer[3]));
```

Definition of the cut: Minimum Bias

3) Ecal Minimum Bias Selection: at least 5 plans along x axis and 6 plans along y axis with energy deposited greater than zero.

```
int nLAYERSMINX=5;
int nLAYERSMINY=6;
int nLAYERs=18;
int nVIEWs=2;
for (int ilayer=0;ilayer<nLAYERs;ilayer++){
view = 1-(int) (ilayer/2)%2;
if (LayerEneDep[ilayer]>0.) ngoodlayers[view]++;}
minimum_bias=(ngoodlayers[0]>= nLAYERSMINX && ngoodlayers[1]>=
nLAYERSMINY);
```

4) TRD Minimum Bias Selection: trd track and 2 TrdHsegment

```
TRD2_nTrdHTrack->Fill(pev->NTrdHTrack());
for(int i=0;i<pev->NTrdHTrack();i++) {
TrdHTrackR *trd_track = pev->pTrdHTrack(i);
if(!trd_track) return false;
TRD2_NTrdHSegment->Fill(trd_track->NTrdHSegment());
if(!(trd_track->NTrdHSegment()==2) return false) ;
```

Definition of the cut: Golden Selection

There are some Golden Condition together with Minimum Bias conditions:

•Golden Tracker: Two routines, see next slides; Which routine do we use?

•Golden Tof: see next slides

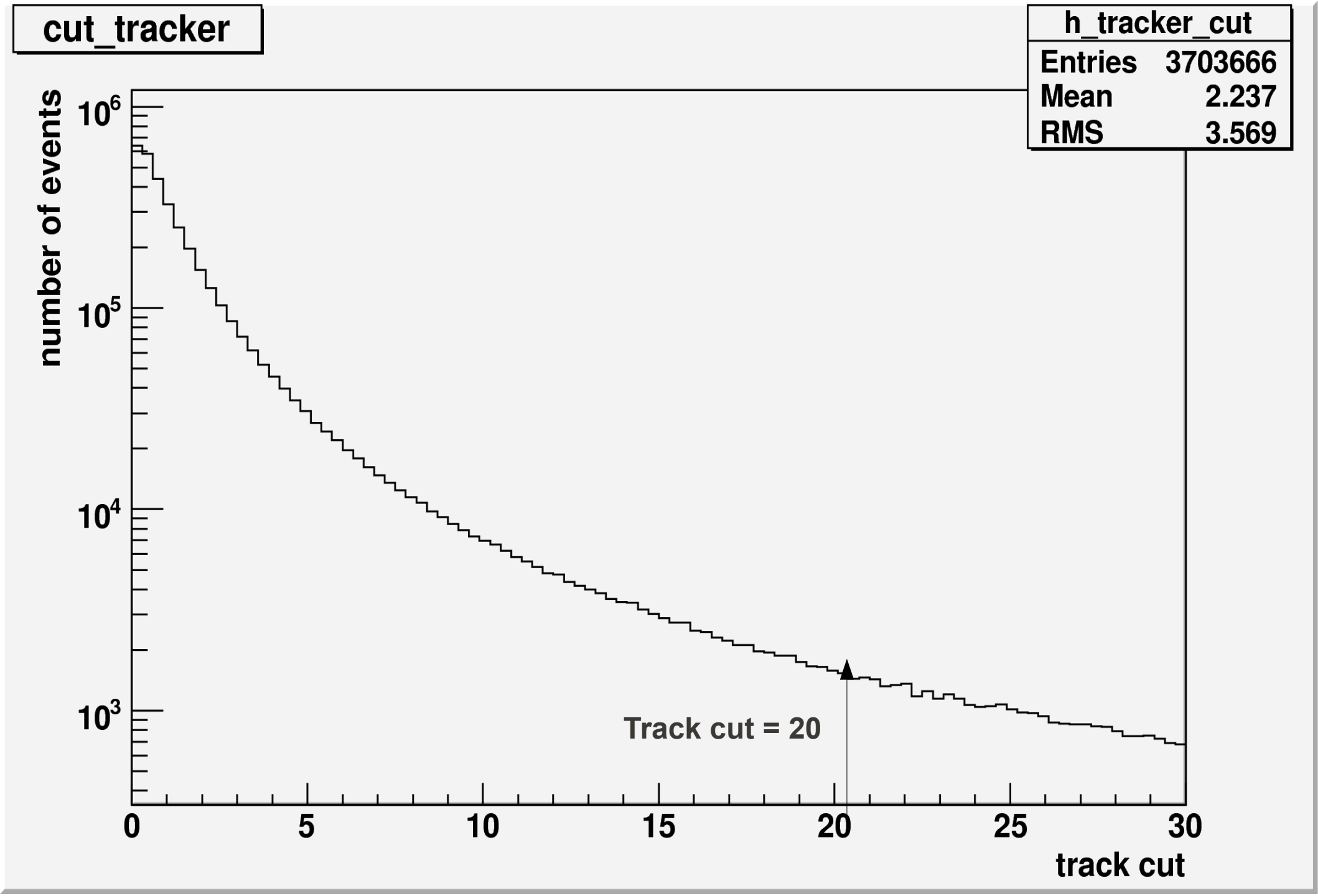
•Golden Ecal: see next slides

•Golden trd: I don't undestand weel,

Tracker routine number one

Tracker Golden: Using only the inner planes, a Tracker golden recquires a number of conditions as at least one hit in each inner plane, fitID greater than zero and normalized chisquare less than 20.

```
Double_t TRACK_QUALITY = 20;
TrTrackR* track = pev->pTrTrack(itrack);
int fitID = track->iTrTrackPar(1, fit, 1);
if (fitID < 0 || !track->ParExists(fitID)) return false;

Double_t rgt = track->GetRigidity(fitID);
Double_t csq = track->GetNormChisqY(fitID);
if (rgt == 0 || csq < 0) return false;

Int_t span = TrTrackSelection::GetSpanFlags(track) & 0xff;
if (!(span & TrTrackSelection::kAllPlane)) return false;

Double_t qtrk;
if(fit<7) qtrk = track->GetNormChisqY(fitID); else{
qtrk = TrTrackSelection::GetHalfRessq(track, 7, 1, 0);
if (qtrk > TRACK_QUALITY) return false;
```


cut_tracker
h_tracker_cut
Entries 3703666
Mean 2.237
RMS 3.569
number of events
10^6
10^5
10^4
10^3
Track cut = 20
0
5
10
15
20
25
30
track cut

Tracker routine number two

Different cut: request of one hit in each inner plane with a different code.

```
TrRecHitR *hit;
for(int i=2; i<9; i++){
hit=track->GetHitLJ(i);
if(hit){
if(!hit->OnlyY()) tr_lay[i-1][0]++;
if(!hit->OnlyX()) tr_lay[i-1][1]++;
}
}  int n_layers[2]={0,0};  for(int i=1; i<8; i++){
if(tr_lay[i][0]>0) n_layers[0]++;
if(tr_lay[i][1]>0) n_layers[1]++;
}
// One hit on every inner tracker plane in X,Y.
if(!(tr_lay[1][0]&&(tr_lay[2][0]||tr_lay[3][0]) && (tr_lay[4][0]||tr_lay[5][0]) && (tr_lay[6][0]||
tr_lay[7][0]))) return false;
if(!(tr_lay[1][1]&&(tr_lay[2][1]||tr_lay[3][1]) && (tr_lay[4][1]||tr_lay[5][1]) && (tr_lay[6][1]||
tr_lay[7][1]))) return false;

if(track->GetNormChisqY(id) >=5. ) return false; // Now is 10.
```

Tracker routine number two

Interpolation of the track onto Tracker layers 1 and 9 with alignment correction. If the interpolation is not good the routine returns false

```
pnt[0]=track->InterpolateLayerJ(1, id);
•TkSens ss(0);
float err=0.1;
bool sen[4][2];
AMSPoint a[4], b[4];

a[0].setp(pnt[0].x()+err, pnt[0].y(),pnt[0].z());//it sets the position
a[1].setp(pnt[0].x(), pnt[0].y()+err,pnt[0].z());
a[2].setp(pnt[0].x()-err, pnt[0].y(),pnt[0].z());
a[3].setp(pnt[0].x(), pnt[0].y()-err,pnt[0].z());

for(int i=0; i<4; i++){
sen[i][0]=false;
ss.SetGlobalCoo(a[i]);
if(ss.LadFound ()) sen[i][0]=true;
}
if(!(sen[0][0] || sen[1][0] || sen[2][0] || sen[3][0])) return false;
```

Tracker routine number two

Interpolation on the top of Ecal (EcalTop)

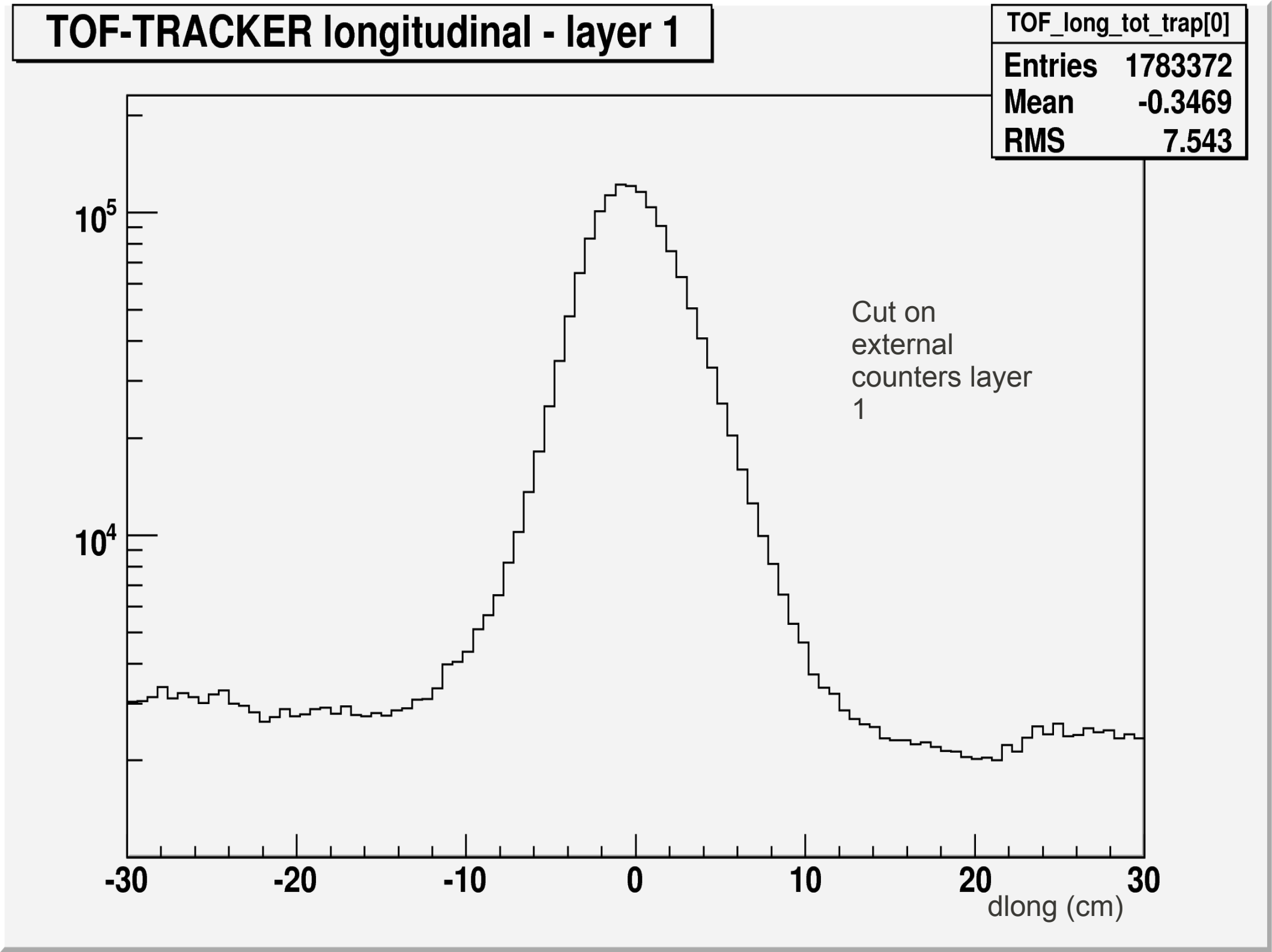
```
AMSPoint ecaltop;
AMSDir ecal;
float z=-139.3;  //EcalTop Now 142.8
track->Interpolate(z, ecaltop, ecal, id);
if(fabs(ecaltop.x())>=32.4 || fabs(ecaltop.y())>=32.4 ) return false;
```

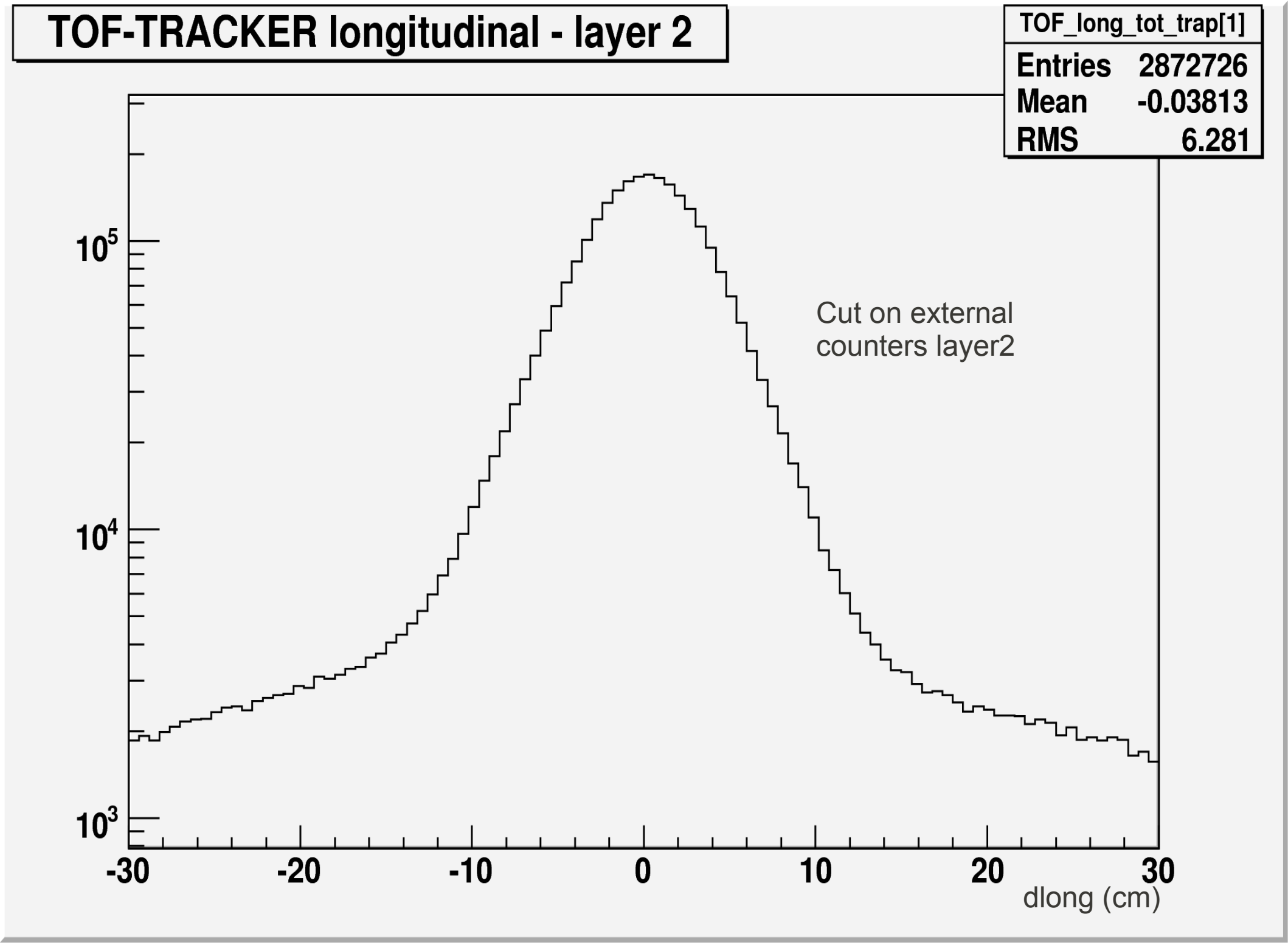
Efficiency of two routines very different (see Results)

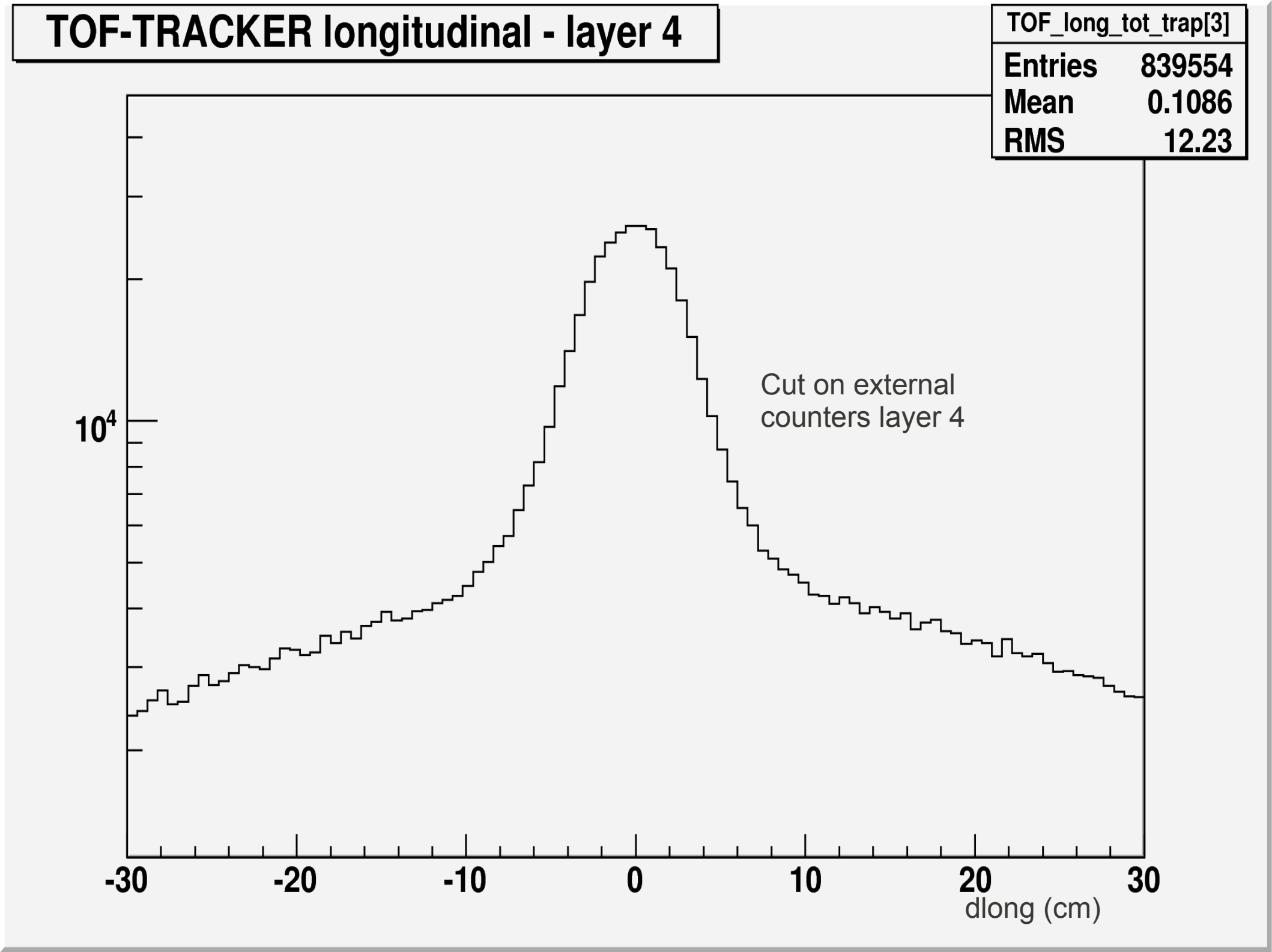
Definition of The cut: Golden Selection

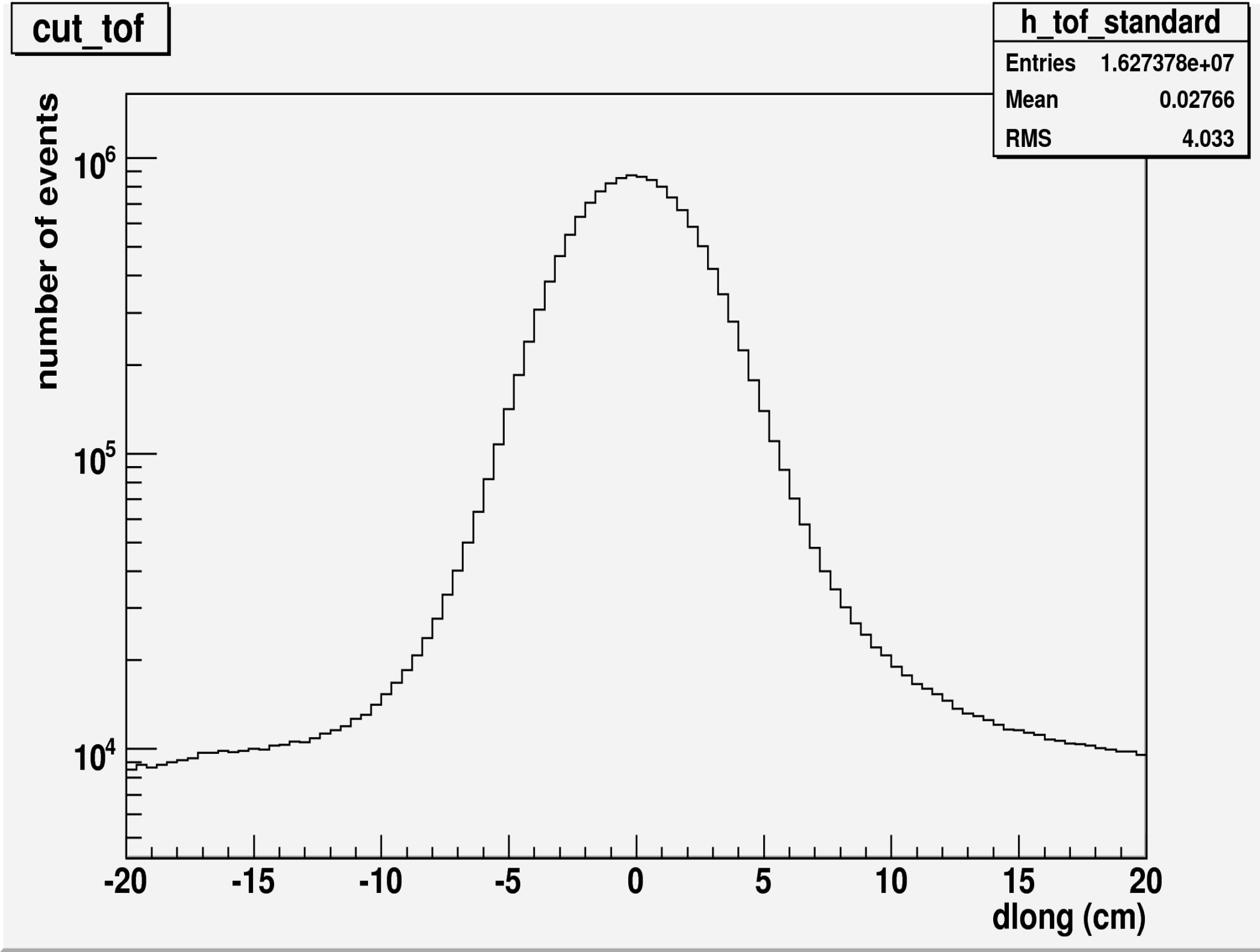
TOF Golden Selection: 4 layers out of 4 with a good match with the track

```
float LONGCUT[4][10]={
    9.,8.,8.,8.,8.,8.,8.,9.,0.,0.,
   12.,8.,8.,8.,8.,8.,8.,12.,0.,0.,
   12.,8.,8.,8.,8.,8.,8.,8.,8.,12.,
10.,8.,8.,8.,8.,8.,8.,10.,0.,0.}; // cm (cut on tof-track longitudinal coord.)
float TRANCUT[4][10]={
   13.,6.,6.,6.,6.,6.,6.,13.,0.,0.,
   14.,6.,6.,6.,6.,6.,6.,14.,0.,0.,
   10.,6.,6.,6.,6.,6.,6.,6.,6.,10.,
14.,6.,6.,6.,6.,6.,6.,14.,0.,0.}; // cm (cut on tof-track trasversal coord.)

int longit[4]={0,1,1,0}; int tranit[4]={1,0,0,1};

tlen=track->Interpolate(cluster->Coo[2],pnt,dir,fitID);
    dlong=cluster->Coo[longit[layer]]-pnt[longit[layer]];
    dtran=cluster->Coo[tranit[layer]]-pnt[tranit[layer]];

if(fabs(dlong)<LONGCUT[layer][bar] && fabs(dtran)<TRANCUT[layer][bar])
    goodlayer[layer]=true;
```


TOF-TRACKER longitudinal - layer 1
TOF_long_tot_trap[0]
Entries 1783372
Mean -0.3469
RMS 7.543
10^5
10^4
Cut on external counters layer 1
-30
-20
-10
0
10
20
30
dlong (cm)


TOF-TRACKER longitudinal - layer 2
TOF_long_tot_trap[1]
Entries 2872726
Mean -0.03813
RMS 6.281
Cut on external
counters layer2
10^5
10^4
10^3
-30
-20
-10
0
10
20
30
dlong (cm)

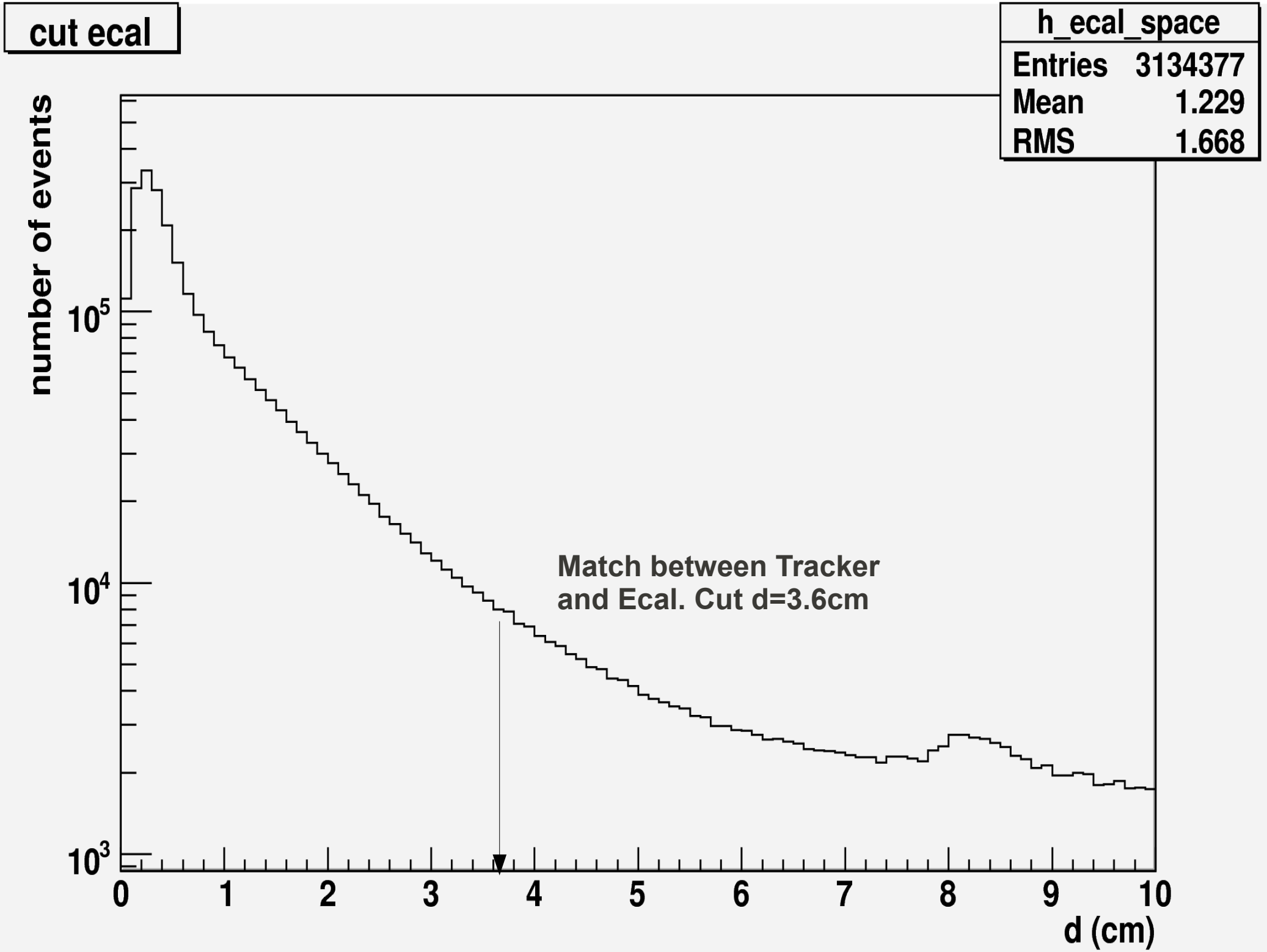

TOF-TRACKER longitudinal - layer 4
TOF_long_tot_trap[3]
Entries 839554
Mean 0.1086
RMS 12.23
Cut on external
counters layer 4
10^4
-30
-20
-10
0
10
20
30
dlong (cm)


cut_tof
h_tof_standard
Entries 1.627378e+07
Mean 0.02766
RMS 4.033
number of events
10^6
10^5
10^4
-20
-15
-10
-5
0
5
10
15
20
dlong (cm)

Definition of the cut: Golden Selection

Ecal Golden Selection: Match between tracker track and Ecal's shower. Cut on entry coordinate of the shower

```
shr=pev->pEcalShower(highE);

if(fabs(shr->Entry[0])>=30.6 || fabs(shr->Entry[1])>=30.6) return false;
if(fabs(shr->Exit[0])>=30.6 || fabs(shr->Exit[1])>=30.6) return false;

TrTrackR *tr_track=pev->pTrTrack(0);
int id=tr_track->iTrTrackPar(1,3,1);

AMSPoint tk_pnt;
AMSDir tk_dir;

tr_track->Interpolate(shr->CofG[2], tk_pnt, tk_dir, id);

float d= sqrt(pow(tk_pnt[0]-shr->CofG[0], 2)+ pow(tk_pnt[1]-shr->CofG[1], 2));

if(d>=3.6) return false;
```


cut ecal
h_ecal_space
Entries 3134377
Mean 1.229
RMS 1.668
number of events
10^5
10^4
10^3
Match between Tracker
and Ecal. Cut d=3.6cm
0
1
2
3
4
5
6
7
8
9
10
d (cm)

Definition of the cut: Golden TRD selection

```
if( pev->nTrdHTrack()!=1) return false;

if( pev->nTrdHSegment() !=2) return false;

TrdHTrackR *trd_track=pev->pTrdHTrack(0);
float TrdChi2=trd_track->Chi2;//
int nTrdHits=trd_track->Nhits;//Number of hits on track.

if (TrdChi2<0.0 || TrdChi2/nTrdHits>=3.0) return false;

int hitsontrack=0;
int pat[3]={0,0,0};
int lay[20];
for(int i=0; i<20;i++) lay[i]=0;

for(int is=0; is<trd_track->nTrdHSegment(); is++){
TrdHSegmentR *seg=trd_track->pTrdHSegment(is);
for(int ih=0; ih<seg->nTrdRawHit(); ih++){
TrdRawHitR *hit=seg->pTrdRawHit(ih);
if(hit->Amp>10){//amplitude (adc counts)
hitsontrack++;
lay[hit->Layer]++;
}
}
}
```

```
for(int i=0; i<4; i++){
  if(lay[i]>0) pat[0]++;
}
for(int i=4; i<16; i++){
  if(lay[i]>0) pat[1]++;
}
for(int i=16; i<20; i++){
  if(lay[i]>0) pat[2]++;
}

if(!(pat[0]>2 && pat[1]>8 && pat[2]>2)) return false;//?

//still not clear if needed
//if( (float)(hitsontrack)/ (float)(pev->nTrdRawHit()) <0.5) return false;

TrTrackR *tr_track=pev->pTrTrack(0);
int id=tr_track->iTrTrackPar(1,3,1);
float P=tr_track->GetRigidity(id);

// TRD point and direction at z=UToF;
float zpl= 63.65 ;  // UToF;
AMSPoint trd_pnt0(trd_track->Coo[0], trd_track->Coo[1], trd_track->Coo[2]);
AMSDir   trd_dir(trd_track->Dir[0], trd_track->Dir[1], trd_track->Dir[2]);

double X_TRD= (zpl-trd_pnt0[2])*trd_dir[0]/trd_dir[2]+trd_pnt0[0];
double Y_TRD= (zpl-trd_pnt0[2])*trd_dir[1]/trd_dir[2]+trd_pnt0[1];
AMSPoint trd_pnt(X_TRD,Y_TRD,zpl);
```

```
AMSPoint tk_pnt;
AMSDir tk_dir;

tr_track->Interpolate(zpl, tk_pnt, tk_dir, id);

double const Pi=4*atan(1);
float TrdTrkDx=trd_pnt[0]-tk_pnt[0];
float TrdTrkDy=trd_pnt[1]-tk_pnt[1];
float TrdTrktheta=tk_dir.gettheta()-Pi+trd_dir.gettheta();
float TrdTrkphi=tk_dir.getphi()+Pi-trd_dir.getphi();
if(TrdTrkphi>Pi) TrdTrkphi=TrdTrkphi-2*Pi;

TF1 *fTrdSigmaDx = new TF1("fTrdSigmaDx",FunTrdSigmaDy,0.0,1000.0,3);
fTrdSigmaDx->SetParameters(2.484,0.1183,0.3487);
TF1 *fTrdSigmaDy = new TF1("fTrdSigmaDy",FunTrdSigmaDy,0.0,1000.0,3);
fTrdSigmaDy->SetParameters(1.686,0.1505,0.2347);

TF1 *fTrd95Da = new TF1("fTrd95Da",FunTrdSigmaDy,0.0,1000.0,3);
fTrd95Da->SetParameters(0.7729,0.7324,0.2005);

double TrdCutDa = 10.0*fTrd95Da->Eval(fabs(P));
double TrdCutDx =  10.0*fTrdSigmaDx->Eval(fabs(P));
double TrdCutDy =  10.0*fTrdSigmaDy->Eval(fabs(P));


if (fabs(TrdTrkDx)>TrdCutDx || fabs(TrdTrkDy)>TrdCutDy) return false;
if( fabs(TrdTrktheta)>TrdCutDa || fabs(TrdTrkphi)>TrdCutDa) return false;
```

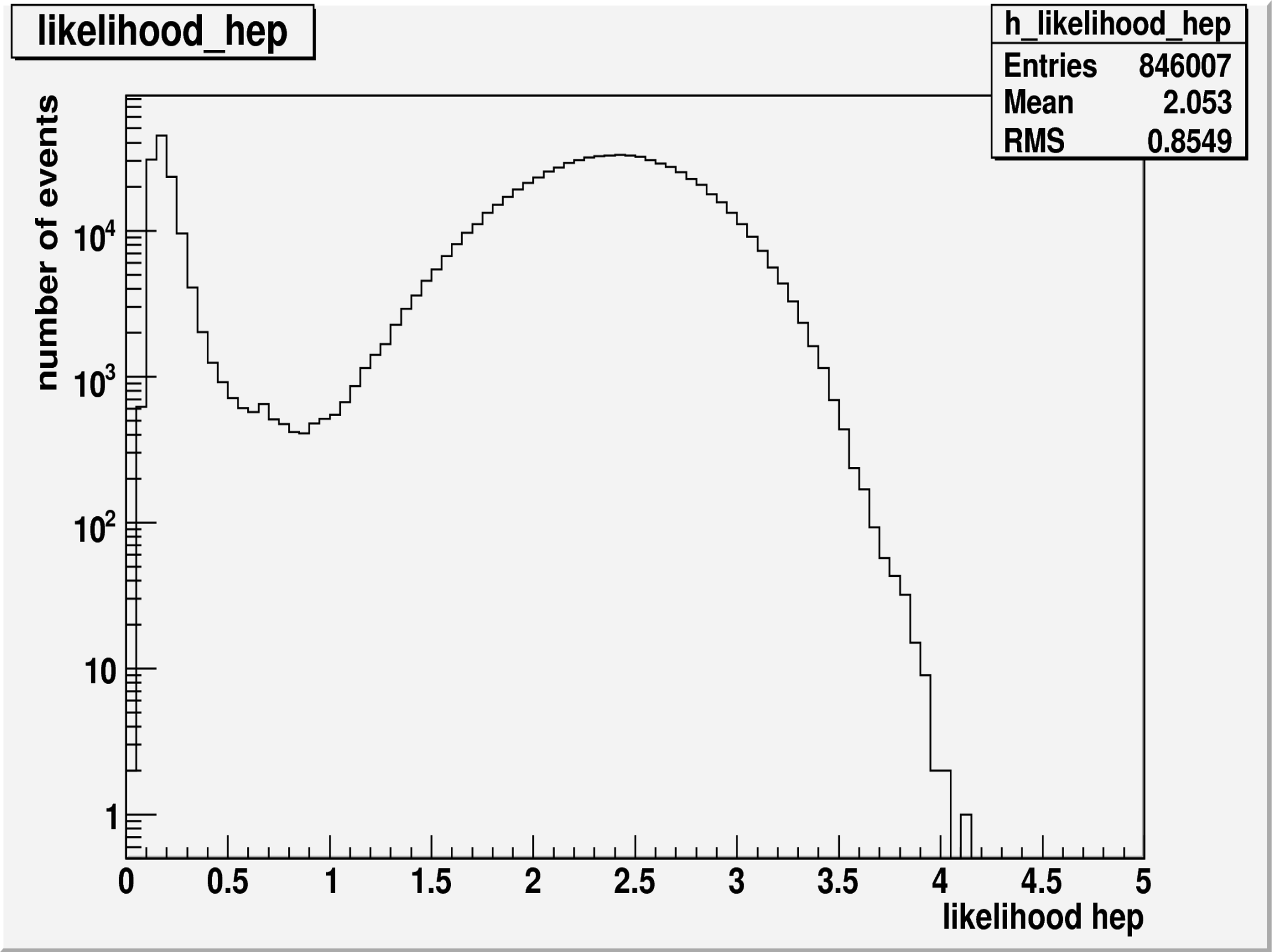
```
 for(int itrack=0;itrack<pev->nTrTrack();itrack++) {
      // loop sugli itrack anche se itrack =0 sempre perchè nel minimum bias scelgo eventi con una sola traccia!
      TrTrackR* track = pev->pTrTrack(itrack);
      int fitID = track->iTrTrackPar(1,fit,1);
      // li rifaccio calcolare. Questa cosa viene già fatto nella routine del tracker. Tuttavia avevo trovato delle incogruenze fra le routine di melanie e le routine di roma sul fitID. Per questo lo rifaccio qui.

      int a = TRACKERSelectionTrack(pev,itrack,fit);


if( pev->nParticle() != 1 ) continue; // questi continue non li capisco comunque eventi con una sola particella li seleziono nel minimum bias
      float Psigned = pev->pParticle(0)->Momentum;
      if(TMath::Abs(Psigned) > 100. || TMath::Abs(Psigned) < 3.0) continue;// qui ci andrebbe break perchè sapevo che la likelihood funzionava solo fra 3 e 100 GeV. Sulla versione 3 del trd ho trovato così però.
```

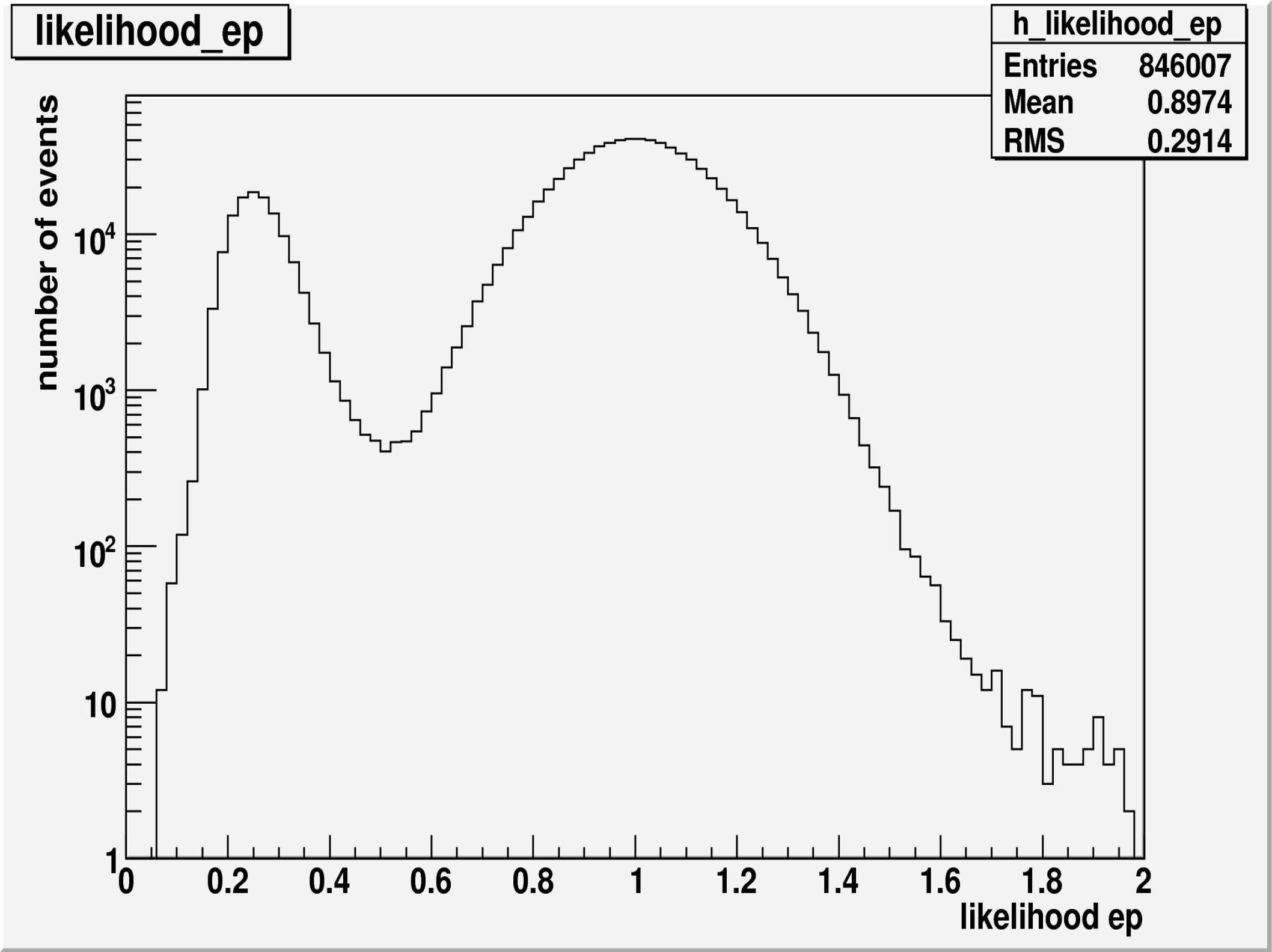
Definition of the Cut: Cut for antiprotons and protons

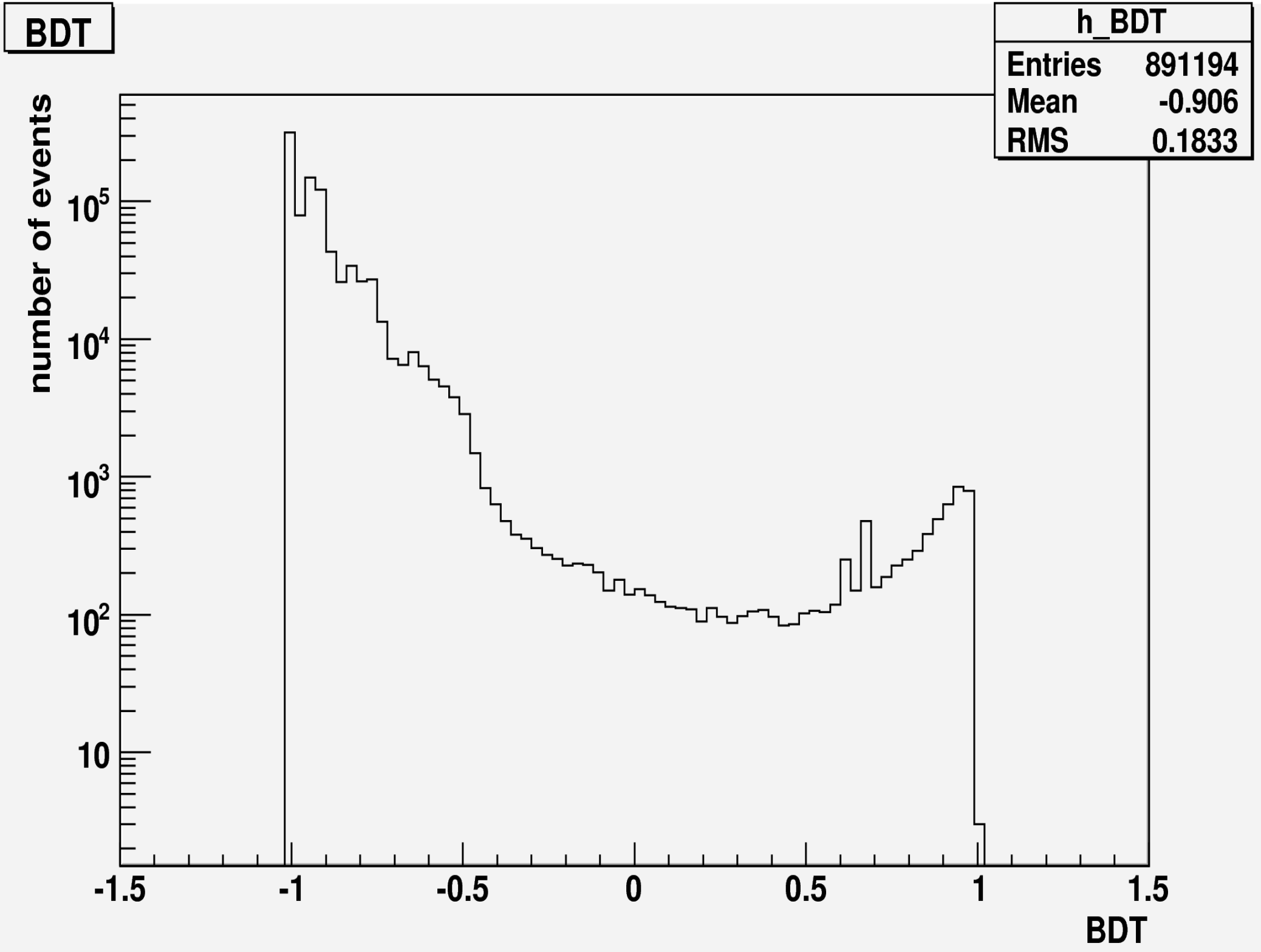
- Trd likelihood electrons/protons > 0.6
- Trd likelihood helium/protons > 1.0
- ECAL BDT < 0.142

Proton's Cut:

- Rigidity > 0, abs(charge -1) <0.3, Momentum > 0, Energy Ecal/Rigidity < 0.8
- Antiproton's Cut:
- Rigidity > 0, abs(charge -1) <0.3, Momentum > 0, Energy Ecal/Rigidity < 0.8


likelihood_hep
h_likelihood_hep
Entries 846007
Mean 2.053
RMS 0.8549
number of events
10^4
10^3
10^2
10
1
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
likelihood hep


likelihood_ep
h_likelihood_ep
Entries 846007
Mean 0.8974
RMS 0.2914
number of events
10^4
10^3
10^2
10
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
likelihood ep


BDT
h_BDT
Entries 891194
Mean -0.906
RMS 0.1833
number of events
10^5
10^4
10^3
10^2
10
-1.5
-1
-0.5
0
0.5
1
1.5
BDT

Results:

•Analyzed Events: 117221939

•Minimum Bias && ssa: 4260284 (3,63 % of Analyzed Events)
•Tracker Golden: 3592716
•TOF Golden: 3372278
•TRD Golden: 2156339
•ECAL Golden: 2465127

•Golden: 905939 (0.77 % of Analyzed Events)

•TRD antip and p selection: 722137 (80% of Golden Events)
•Ecal antip and p selection: 720631

•Protons:705692 (98% of TRD selection)
•Antiprotons: 219 (0.3 * 10^{-3})
•Protons with beta >1: 273978 (39 % of protons)
•Antiprotons with beta >1: 99

Results Melanie:

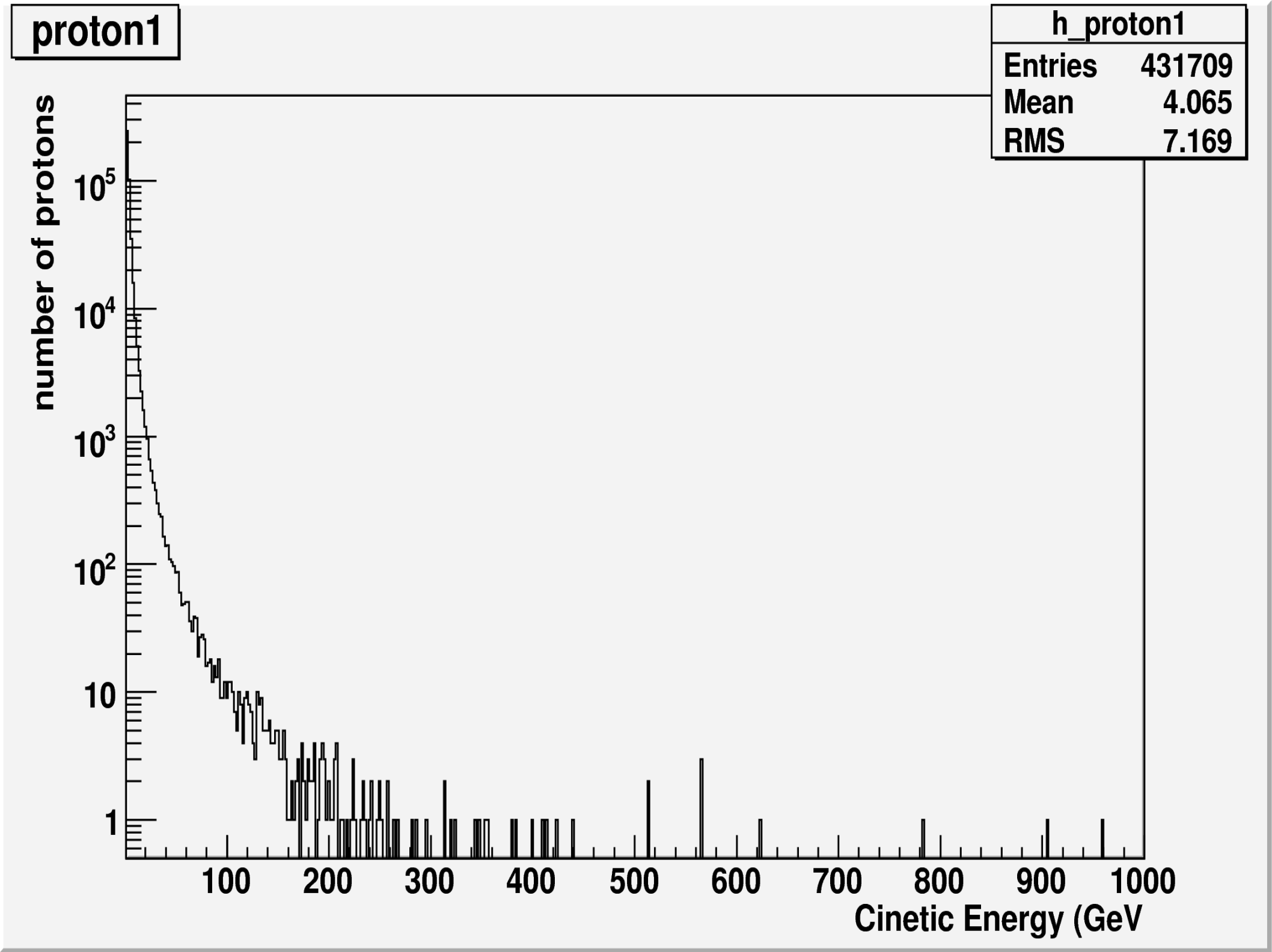
•Analyzed Events: 117221939

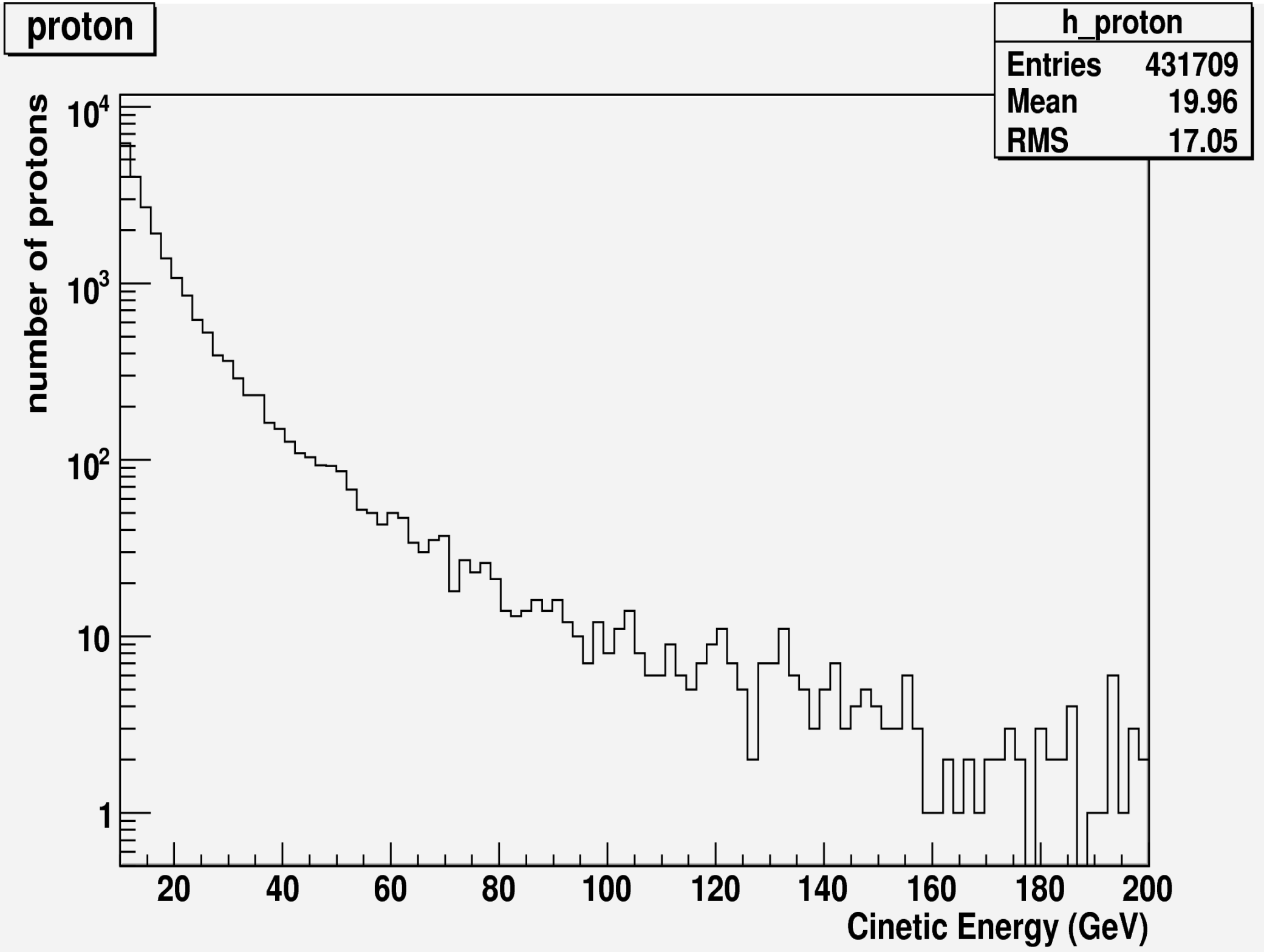
•Minimum Bias && ssa: 4260284 (3,63 % of Analyzed Events)
•Tracker Golden: 1125764
•TOF Golden: 3372278
•TRD Golden: 2156339
•ECAL Golden: 2465127

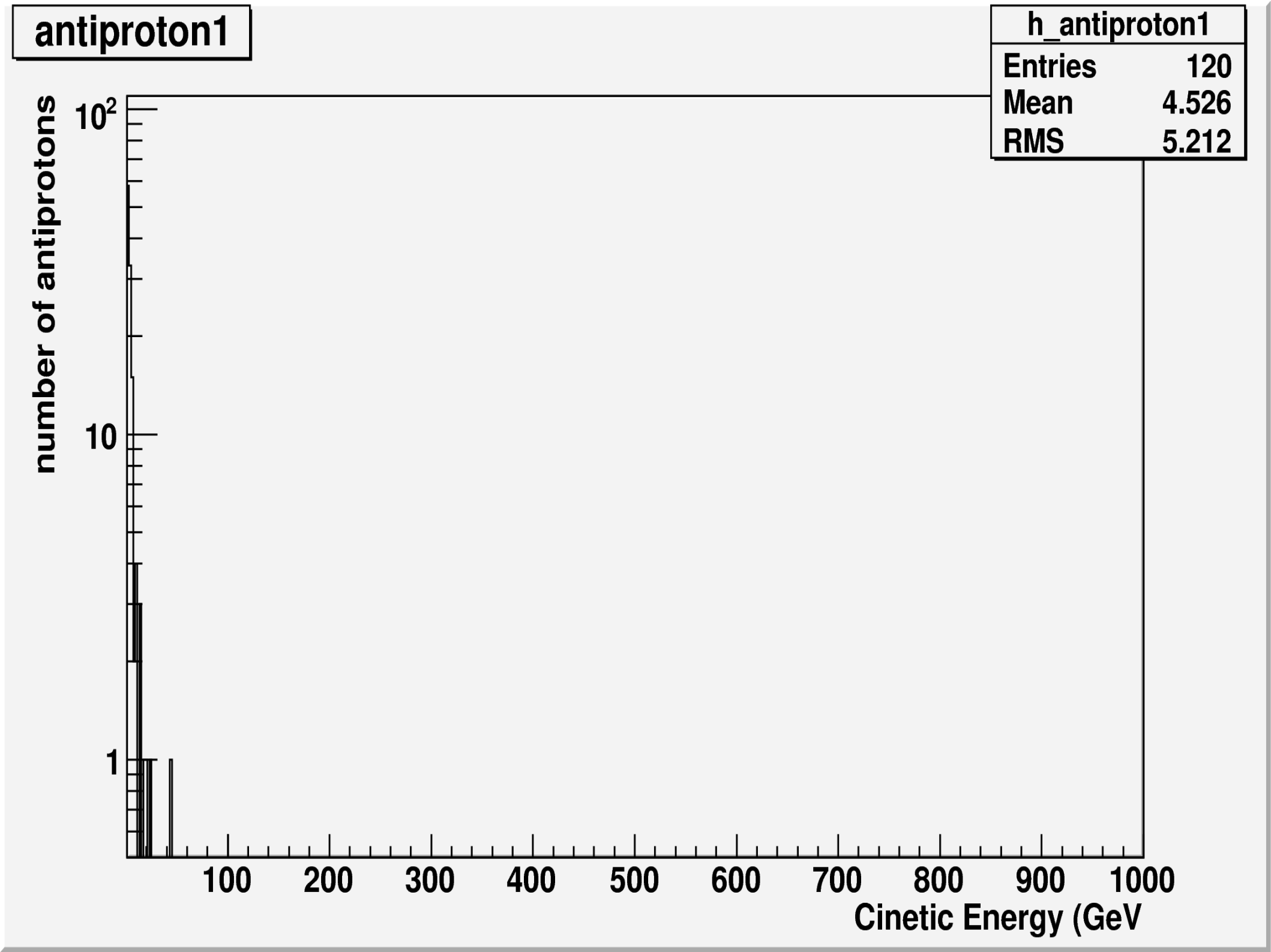
•Golden: 390408 (0.33 % of Analyzed Events)

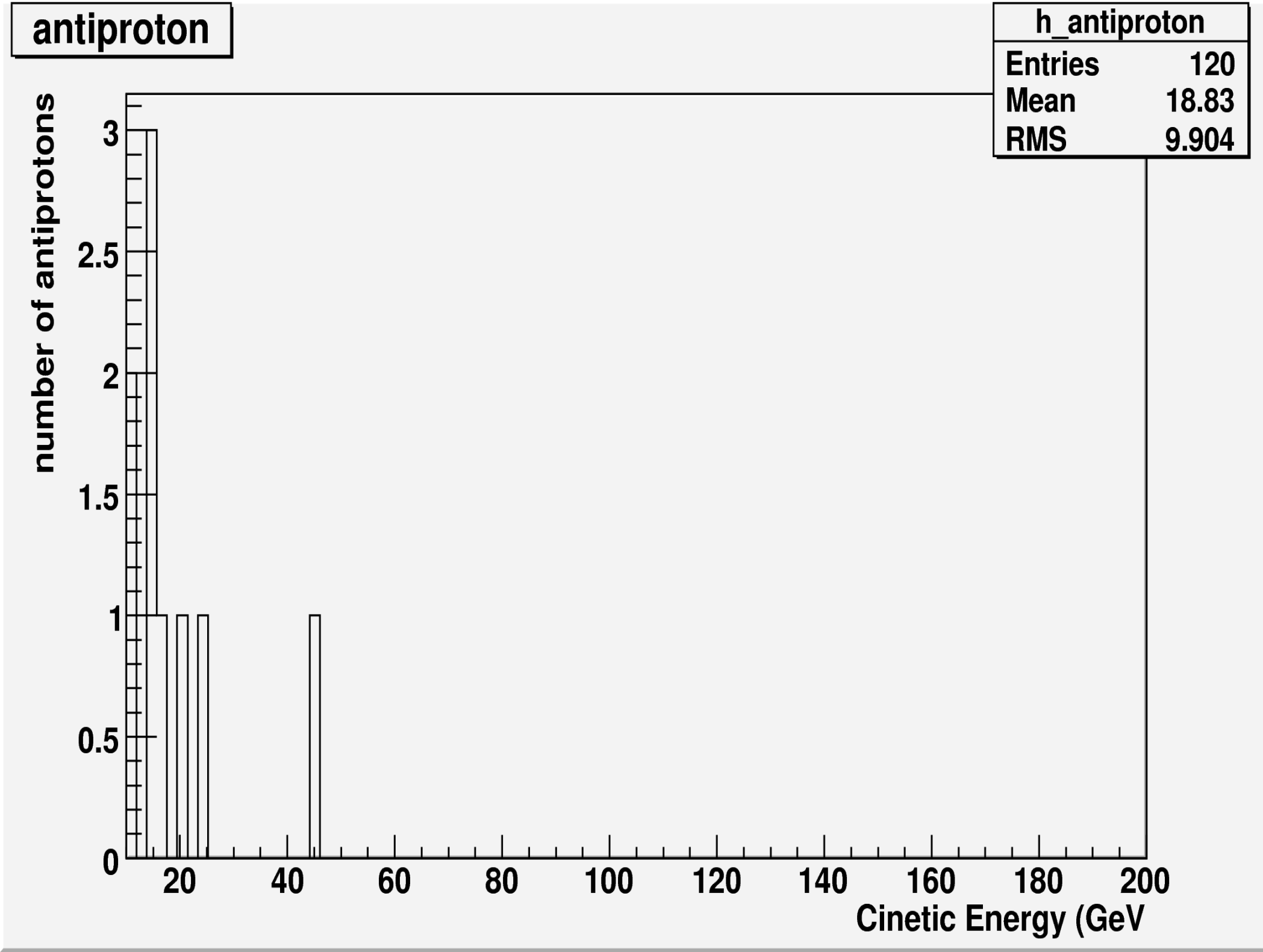
•TRD antip and p selection: 304350 (77% of Golden Events)
•Ecal antip and p selection: 303743

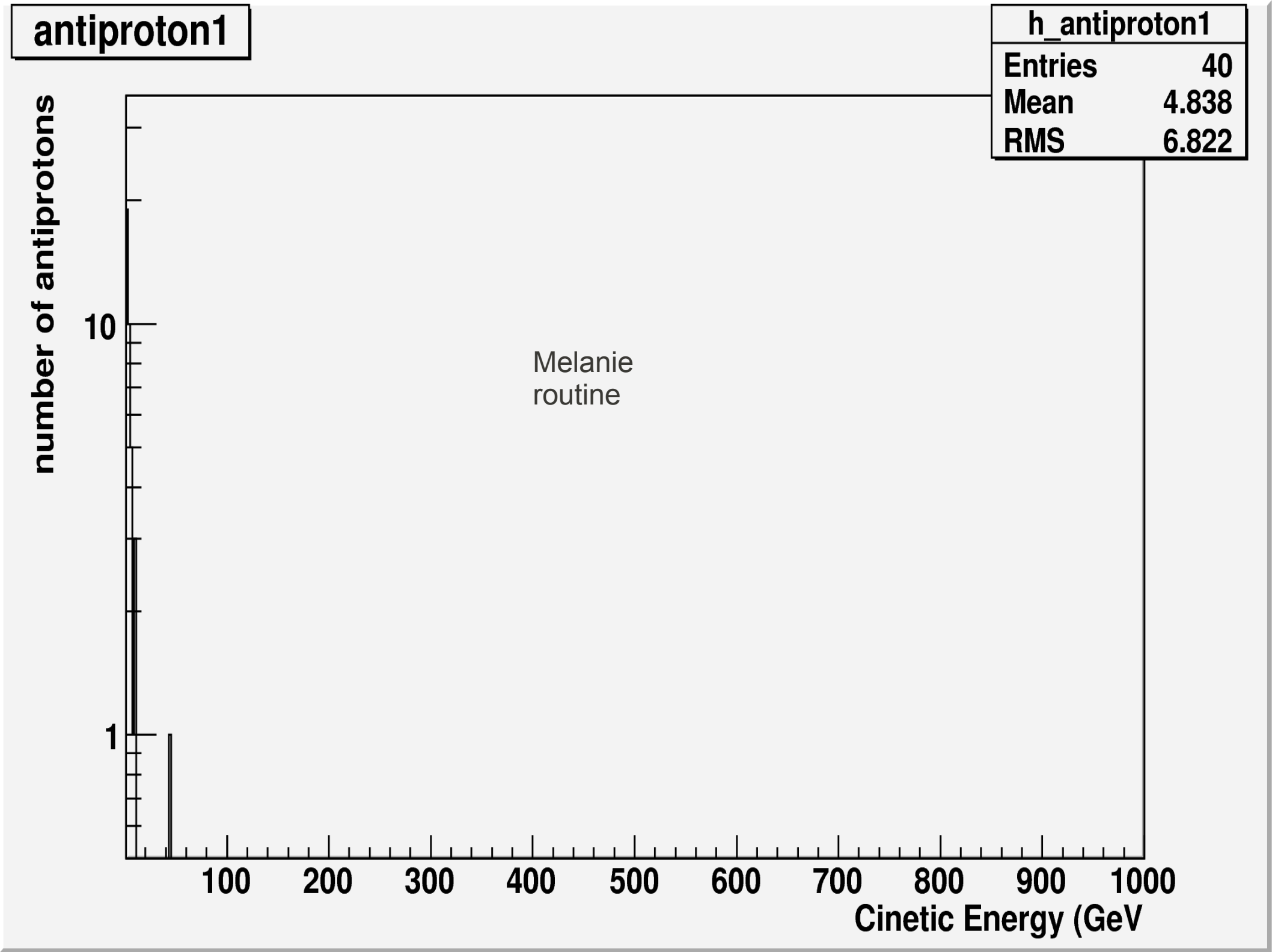
•Protons: 297541 (98% of TRD selection)
•Antiprotons: 66 (0.2 * 10^{-3} of protons)
•Protons with beta >1: 115050 (39 % of protons)
•Antiprotons with beta >1: 26

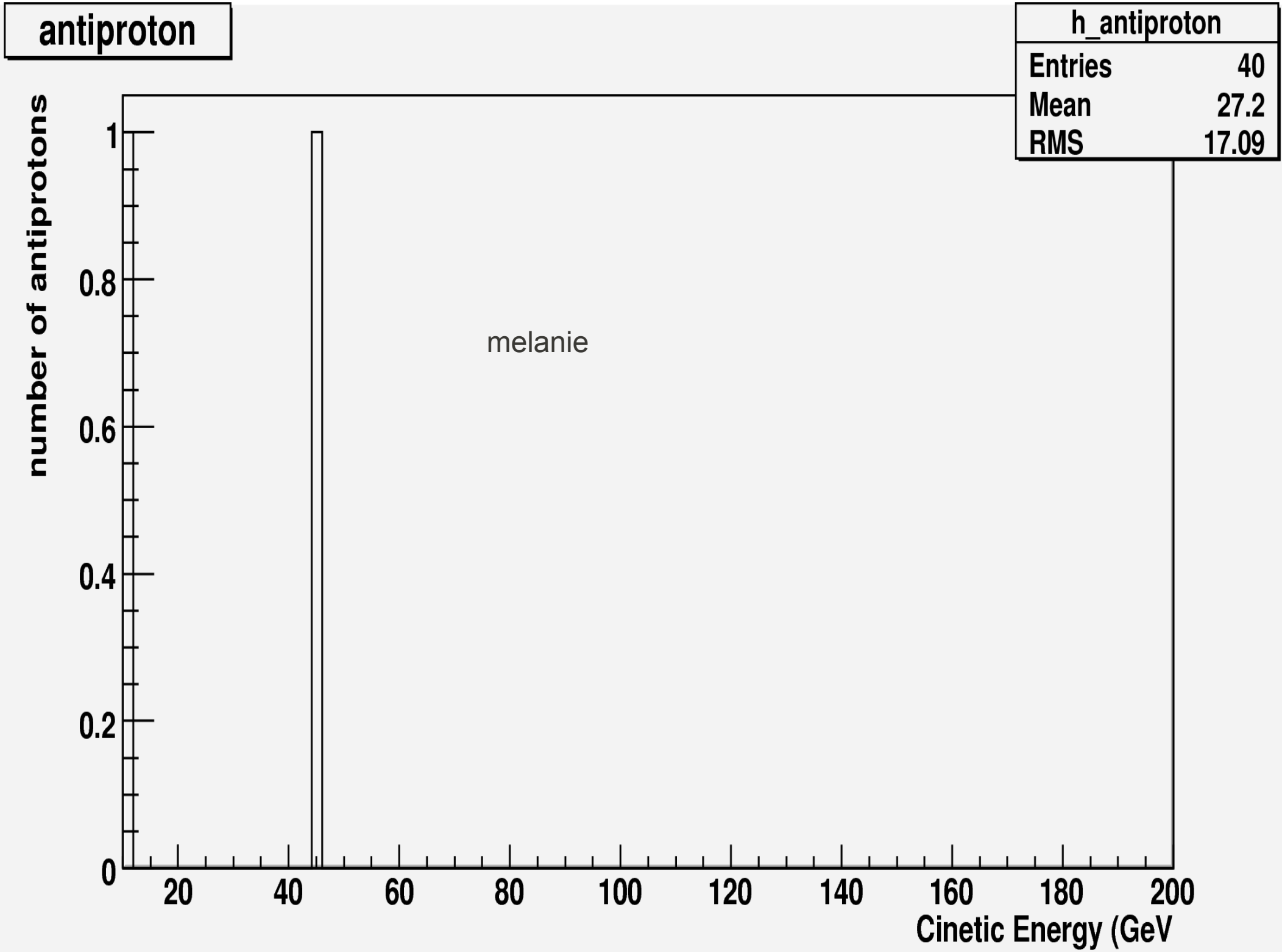

proton1
h_proton1
Entries 431709
Mean 4.065
RMS 7.169
number of protons
10^5
10^4
10^3
10^2
10
1
100
200
300
400
500
600
700
800
900
1000
Cinetic Energy (GeV

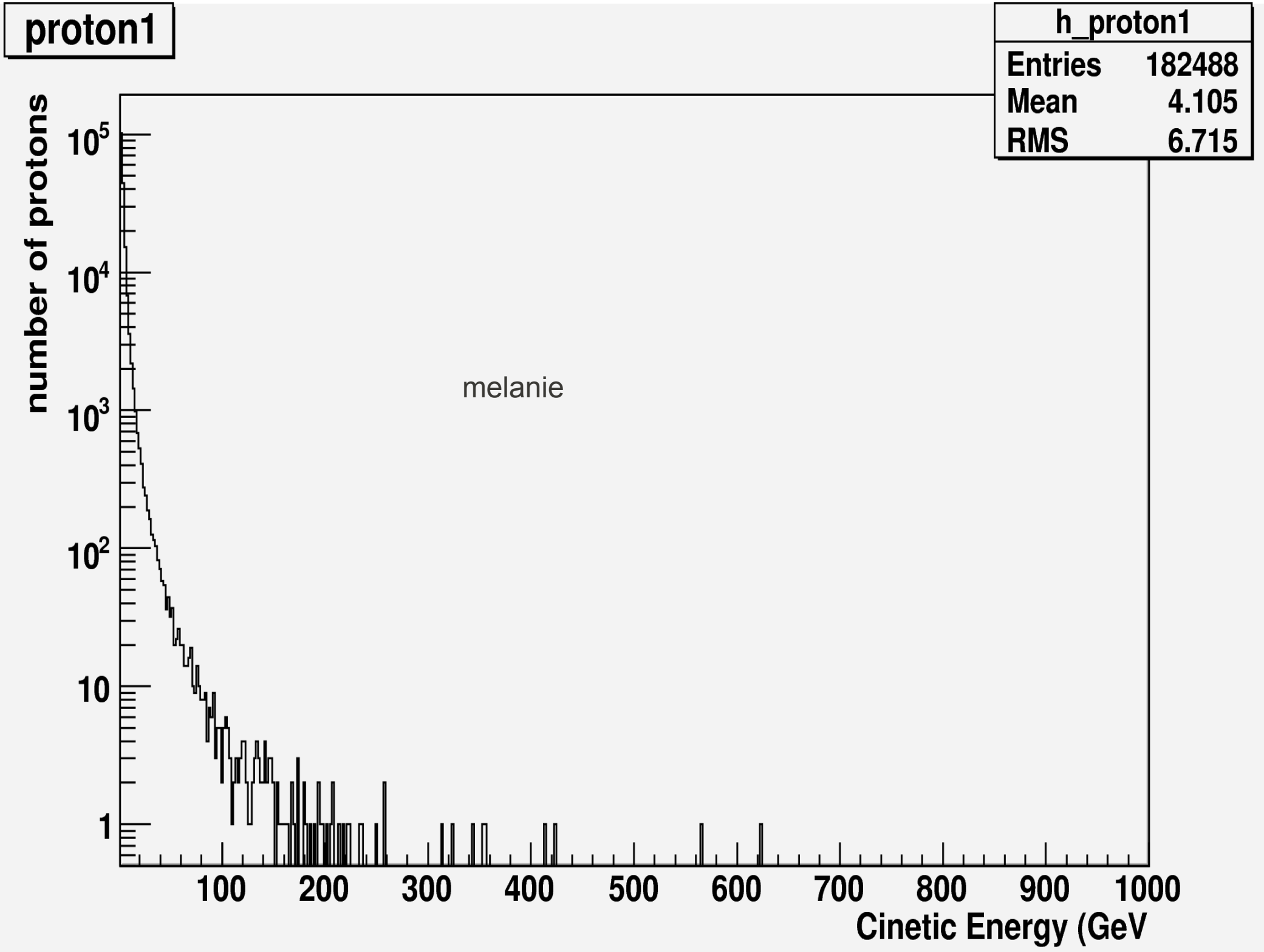

proton
h_proton
Entries 431709
Mean 19.96
RMS 17.05
number of protons
10^4
10^3
10^2
10
1
20
40
60
80
100
120
140
160
180
200
Cinetic Energy (GeV)

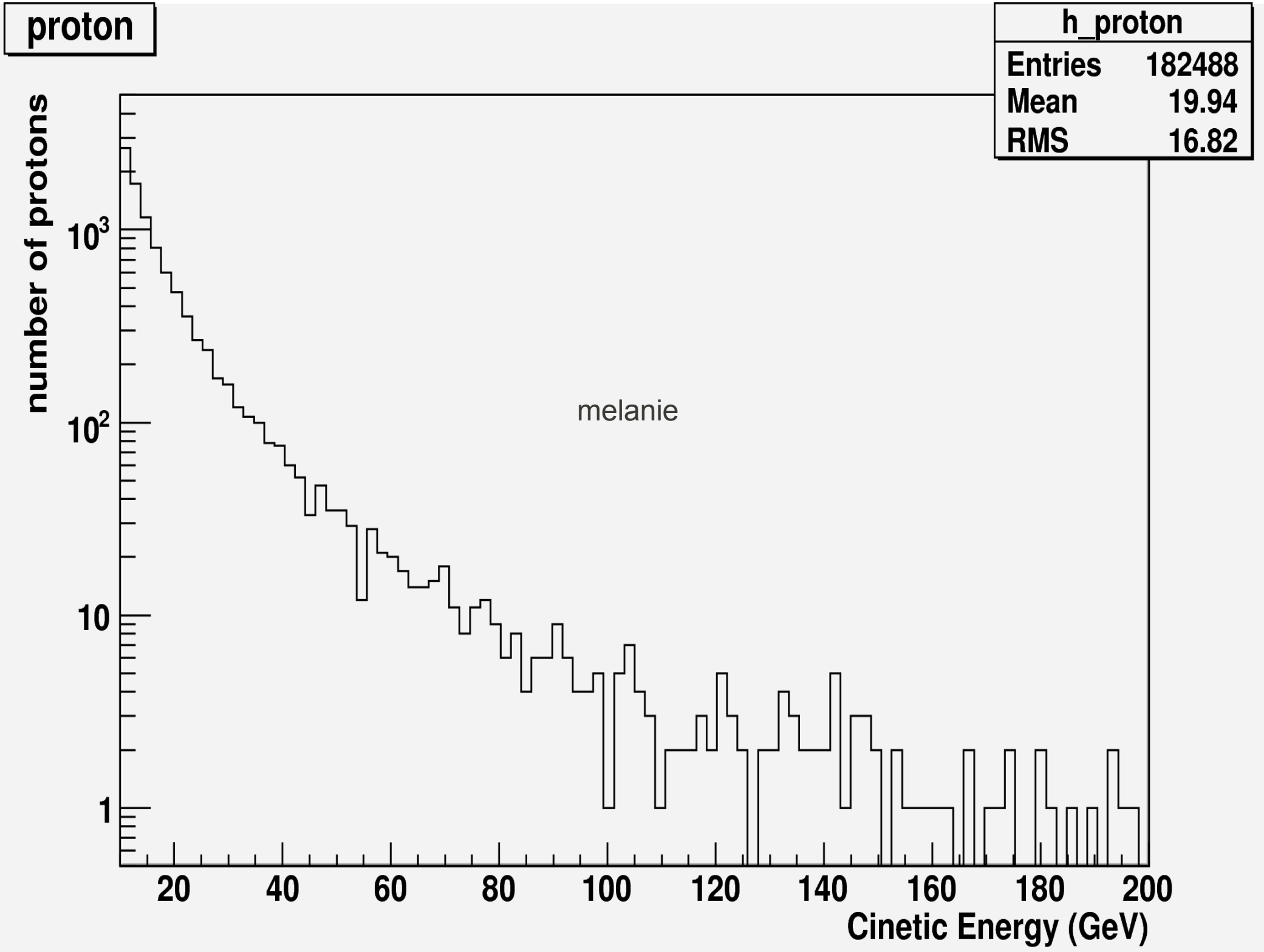

antiproton1
h_antiproton1
Entries 120
Mean 4.526
RMS 5.212
number of antiprotons
10^2
10
1
100
200
300
400
500
600
700
800
900
1000
Cinetic Energy (GeV


antiproton
h_antiproton
Entries 120
Mean 18.83
RMS 9.904
number of antiprotons
0
0.5
1
1.5
2
2.5
3
20
40
60
80
100
120
140
160
180
200
Cinetic Energy (GeV


antiproton1
h_antiproton1
Entries 40
Mean 4.838
RMS 6.822
number of antiprotons
10
1
Melanie
routine
100
200
300
400
500
600
700
800
900
1000
Cinetic Energy (GeV


antiproton
h_antiproton
Entries 40
Mean 27.2
RMS 17.09
number of antiprotons
1
0.8
0.6
0.4
0.2
0
melanie
20
40
60
80
100
120
140
160
180
200
Cinetic Energy (GeV


proton1
h_proton1
Entries 182488
Mean 4.105
RMS 6.715
number of protons
10^5
10^4
10^3
10^2
10
1
melanie
100
200
300
400
500
600
700
800
900
1000
Cinetic Energy (GeV


proton
h_proton
Entries 182488
Mean 19.94
RMS 16.82
number of protons
10^3
10^2
10
1
melanie
20
40
60
80
100
120
140
160
180
200
Cinetic Energy (GeV)

Next Steps:

- Does the program need other cuts or change the values of cuts?
- Define the efficiency and purity of the minimum bias, golden and antip and p cuts (Montecarlo?)
- How can we obtain the flux in standard unit?