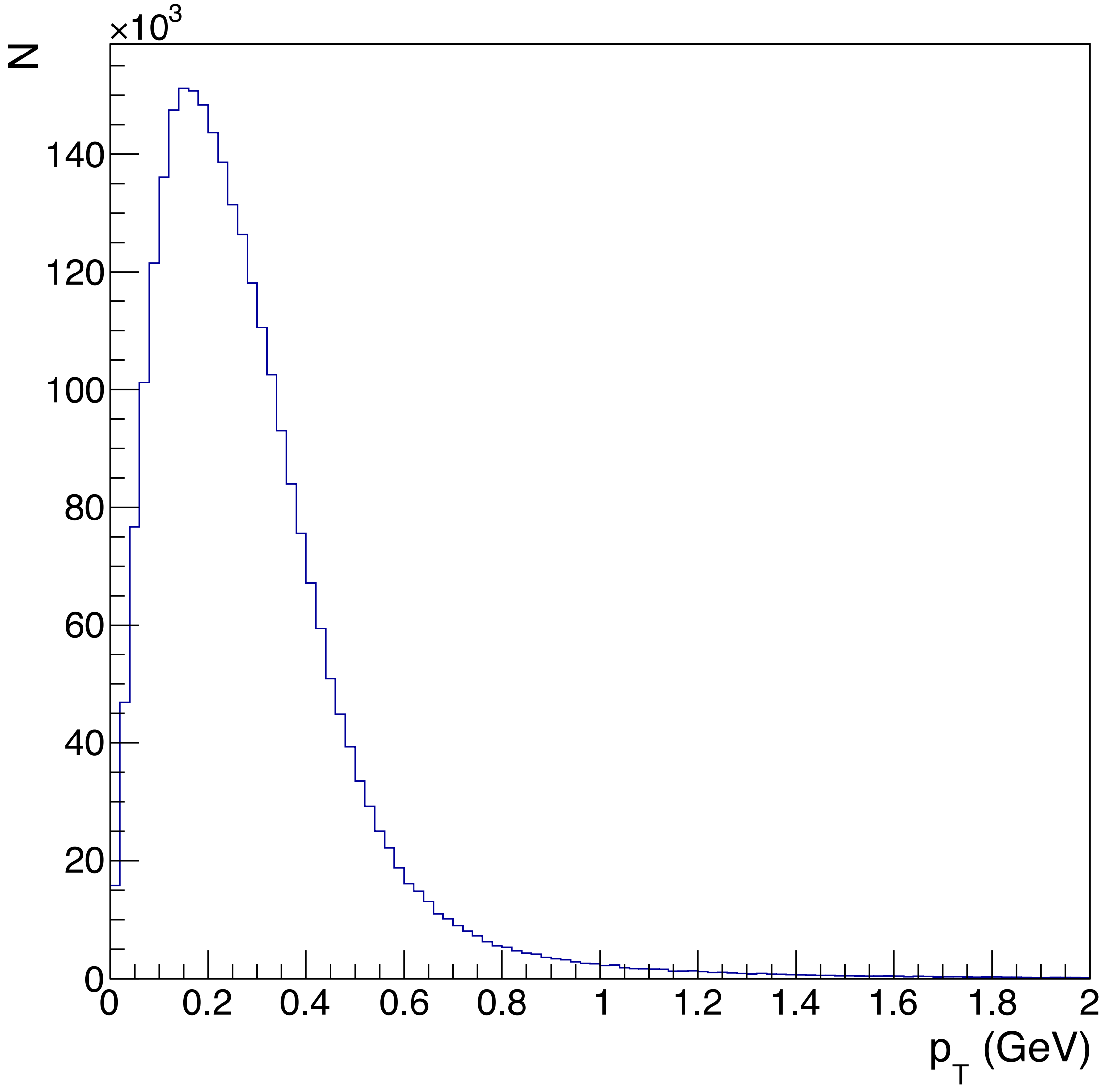
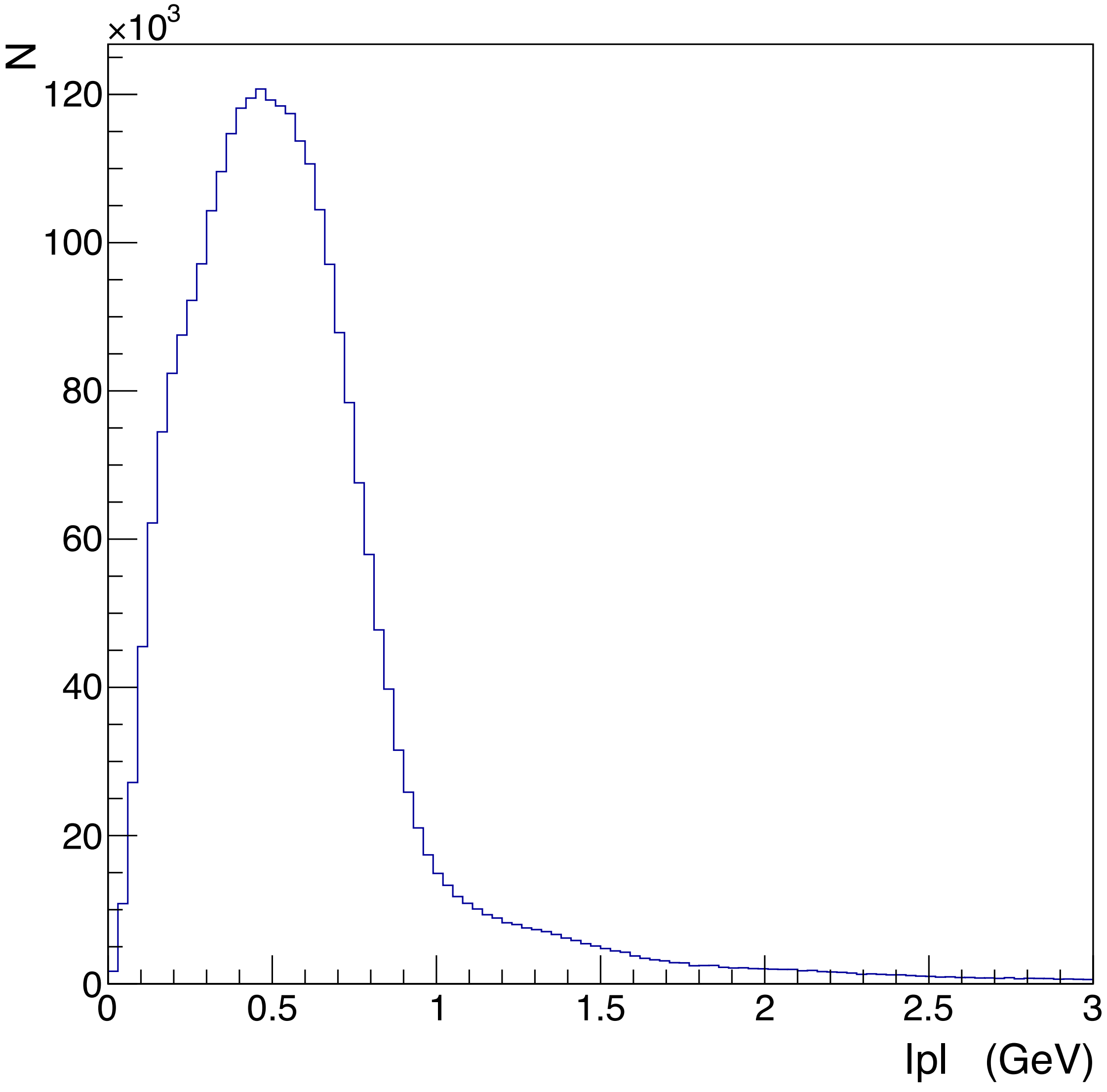
# TPC Simulation

정우

# IQMD momentum distribution





# IQMD particle ratios

| Particle Name | Fraction |
|---|---|
| Neutron | 0.52 |
| Proton | 0.33 |
| Deutron | 0.07 |
| Triton | 0.02 |

Mean multiplicity per event : **278**

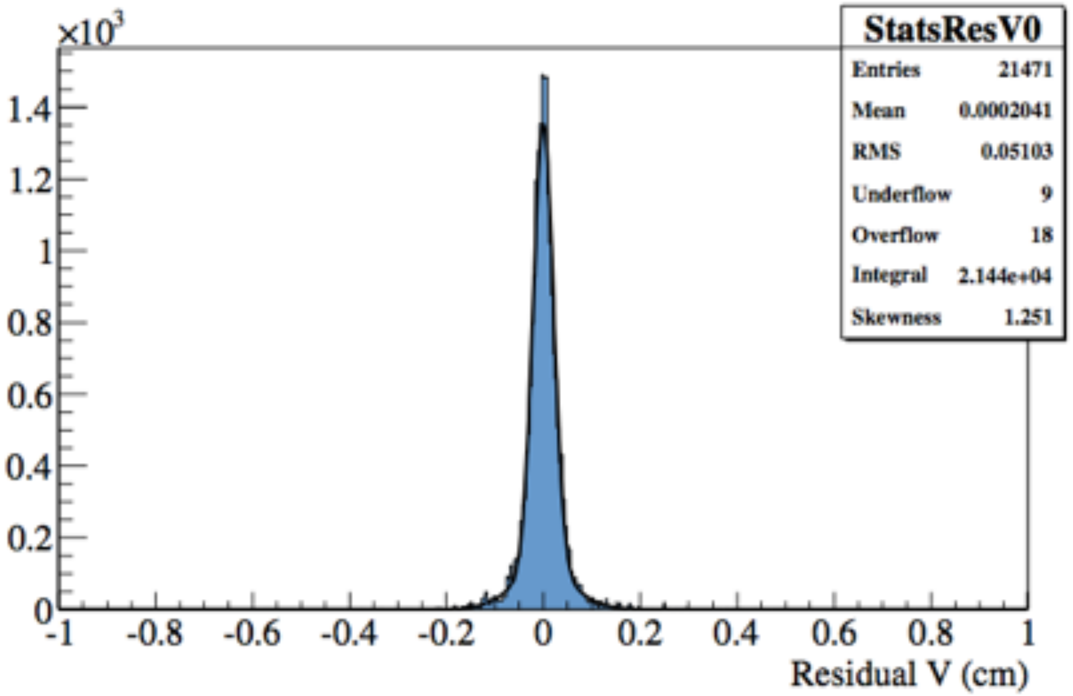
# PANDA TPC prototype

* Technical Design Report, PANDA, 2011

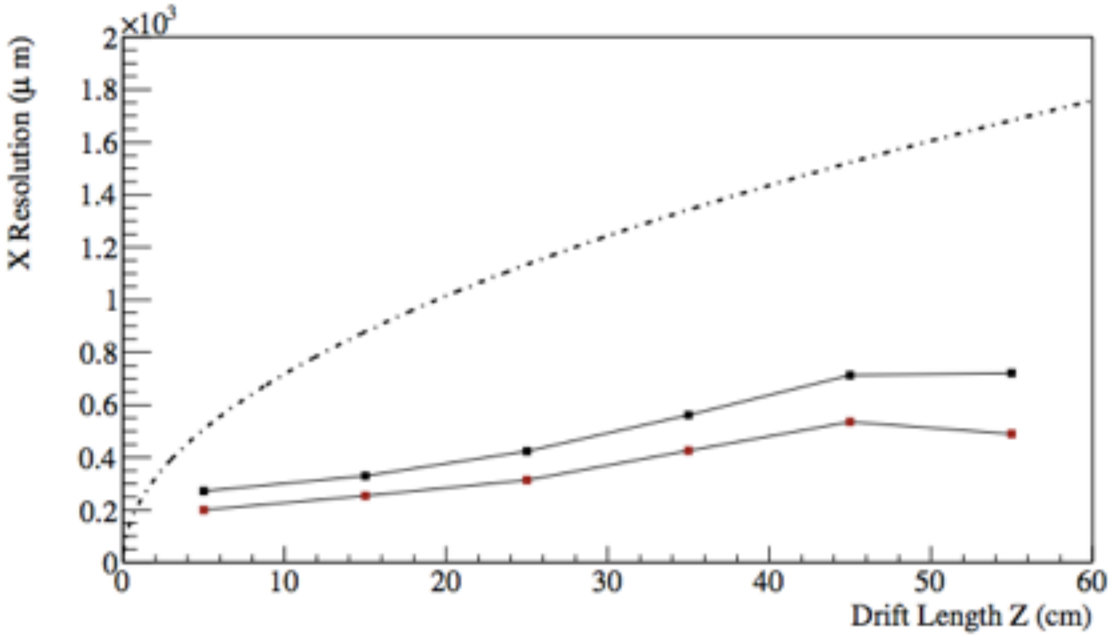
| | |
|---|---|
| TPC drift length | 727.8 mm |
| TPC inner diameter | 104 mm |
| TPC outer diameter | 308 mm |
| Hexagonal pad radius | 1.5 mm |
| Drift chamber field | 360 V/cm |
| Gas mixture | Ar/CO |

# PANDA TPC cosmic ray data

* Technical Design Report, PANDA, 2011



(a) Residual measured with cosmic tracks. The experimental distribution is fitted with two Gaussian functions. The weighted mean of the sigmas of the two Gaussian curves is 230 µm.



(b) Chamber resolution (along $x$) as a function of drift length. The dashed line is the transverse diffusion for single electron drift as calculated with GARFIELD. The dots show the results from residual distributions for tracks in 10 cm bins along the drift direction (cf. (a)). The red dots are the sigmas of the narrow Gaussian, while the black dots above correspond to the weighted mean of both Gaussian curves. Statistical error bars are included, but smaller than the data points.

# PANDA TPC momentum resolution

* Technical Design Report, PANDA, 2011

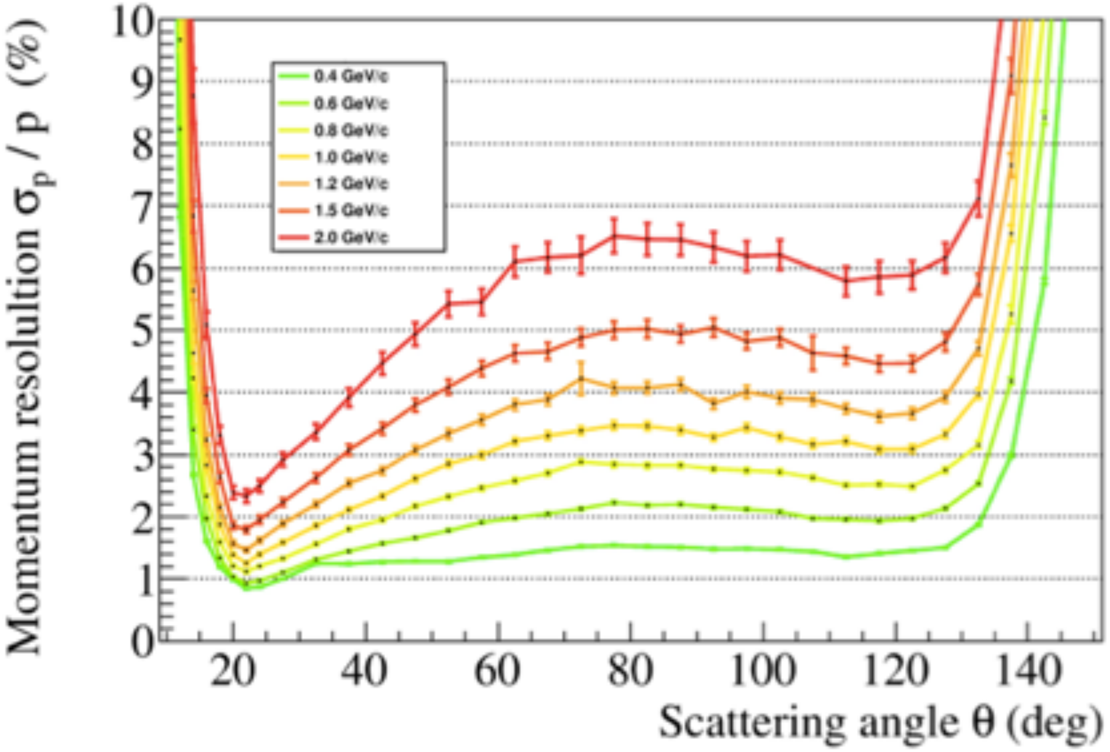


**Figure 4.59:** Momentum resolution $\sigma_p/p$ obtained from $\mu$-tracks reconstructed in the TPC alone. Each data point corresponds to a Gaussian fit to the momentum distribution as reconstructed from 5000 tracks. The error bars show the squared sum of the statistical uncertainties of sigma and mean of the final fit.

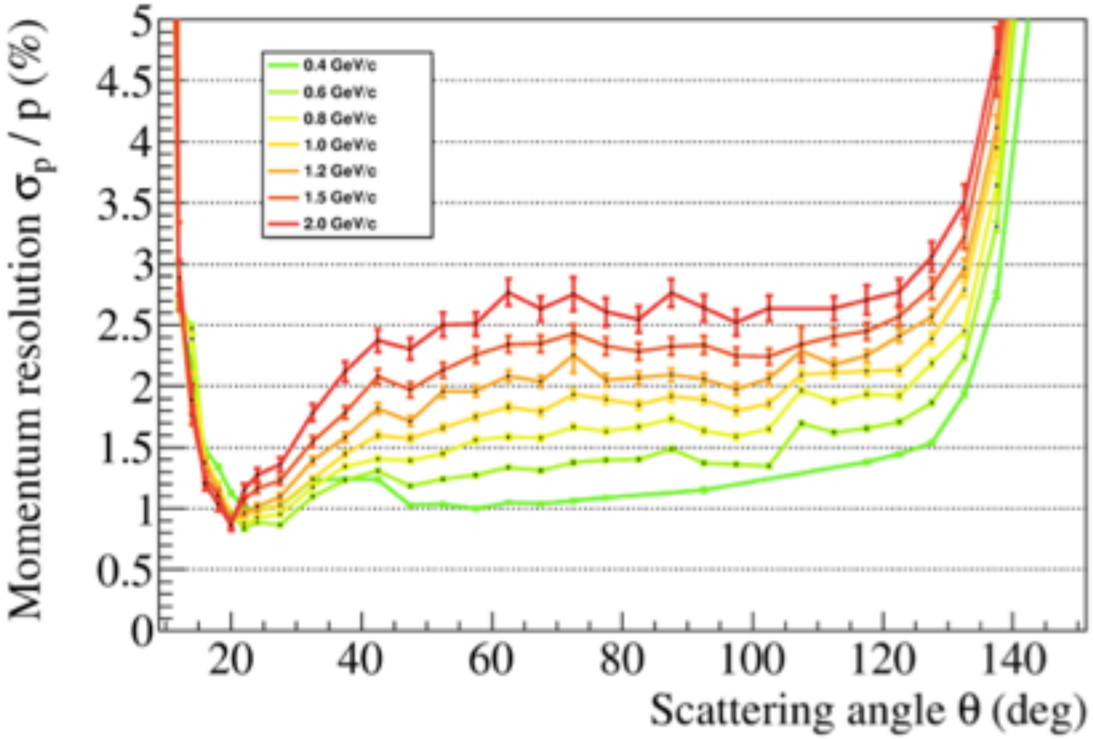
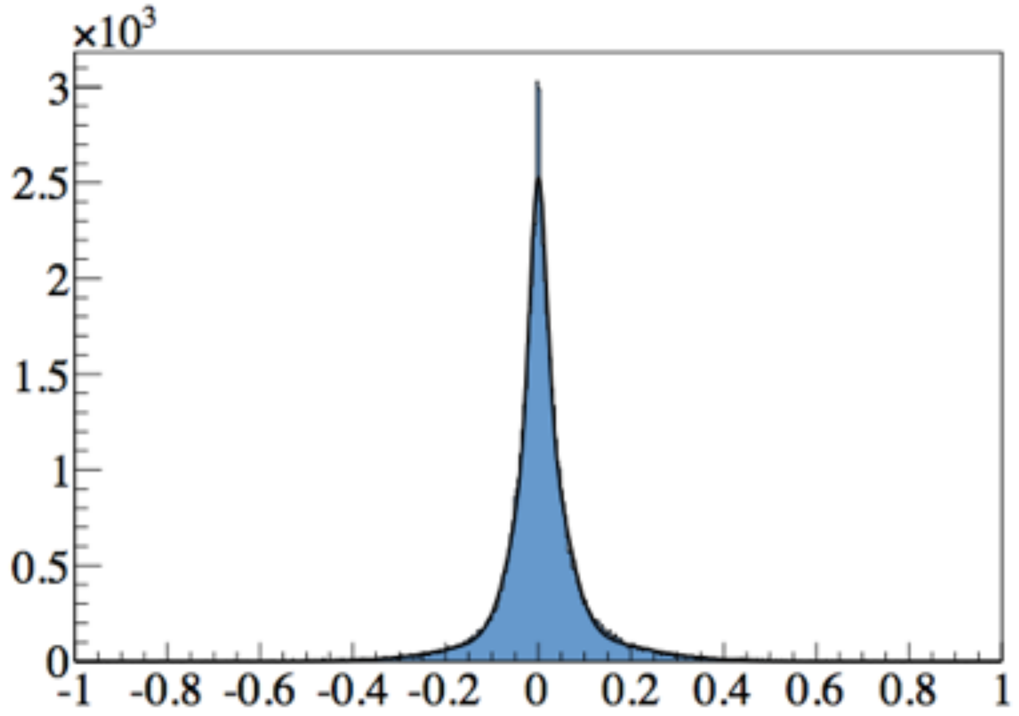


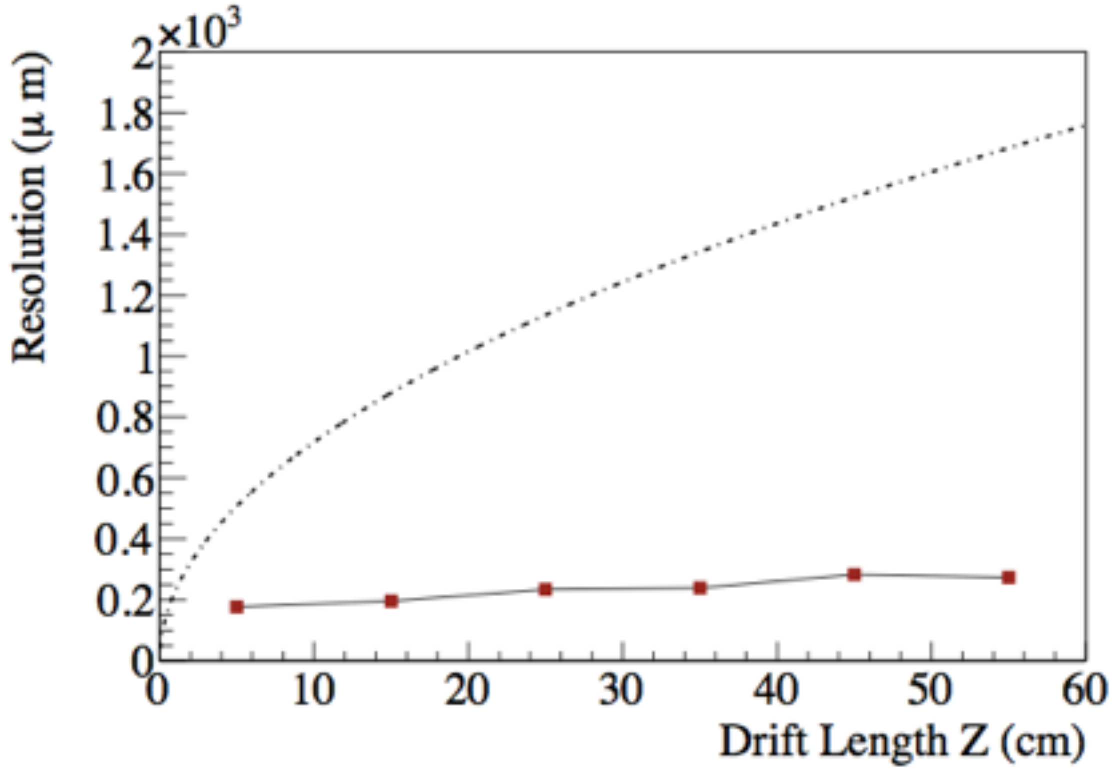
**Figure 4.60:** Momentum resolution $\sigma_p/p$ obtained from $\mu$-tracks reconstructed in the TPC and then merged with hits from the MVD and GEM detectors, using the same data sample as in Fig. 4.59. In each MVD (GEM) detector plane the closest hit inside a road width of 2 mm (5 mm) was added to the TPC track. No Monte Carlo information has been used for the matching.

# PANDA TPC beam data

* Technical Design Report, PANDA, 2011

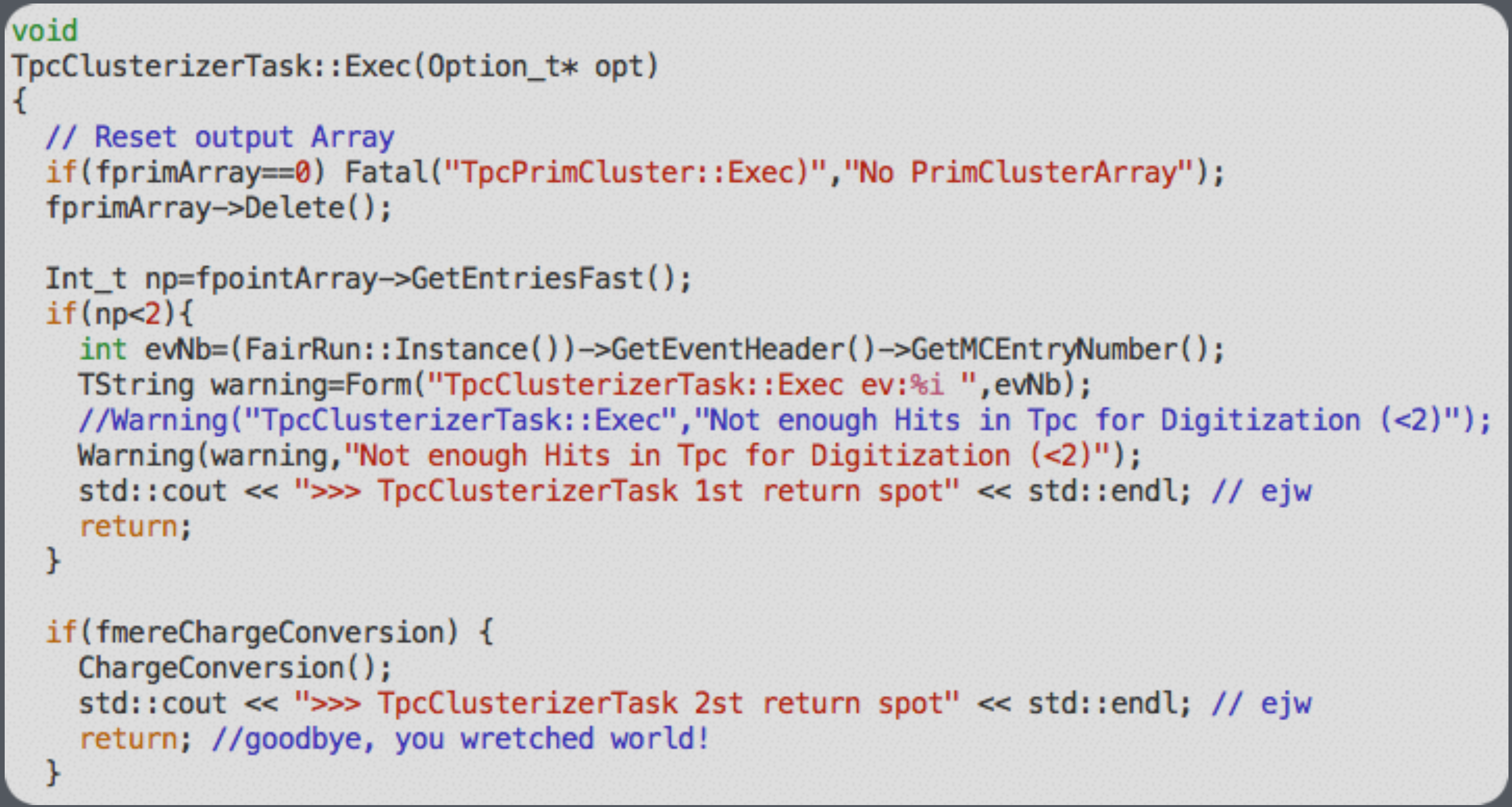


(a) Residuals in V and U direction for the first 10 cm of drift. The data are fitted with 3 Gaussian function. The line shows the sum of the resulting fitted curves.

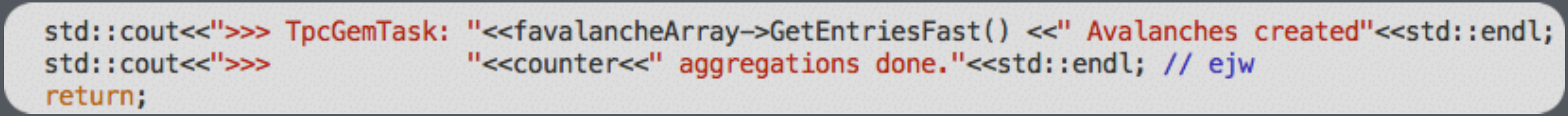


(b) Residual distributions for tracks reconstructed from $^{22}$Ne+Al at 1.7 AGeV collisions as a function of the drift length. The dashed line shows the transverse diffusion for single electron drift calculated with GARFIELD, the red dots represent the sigmas of the narrow Gaussian fits.

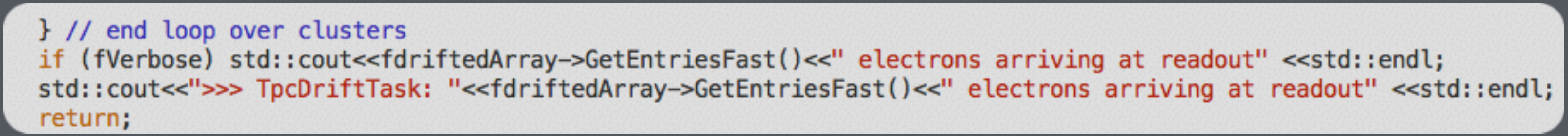
FOPI
ROOT



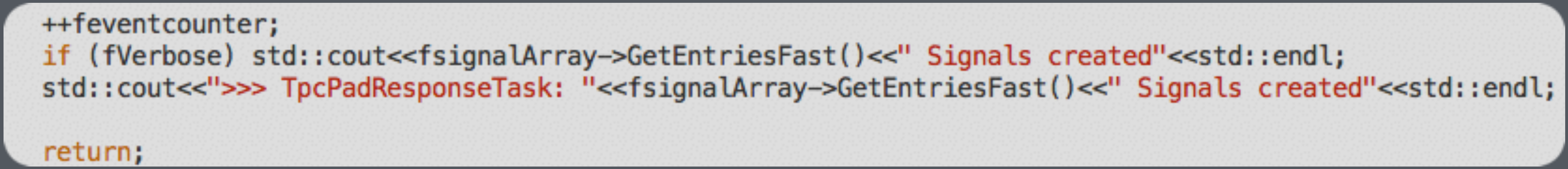
```
void
TpcClusterizerTask::Exec(Option_t* opt)
{
  // Reset output Array
  if(fprimArray==0) Fatal("TpcPrimCluster::Exec)","No PrimClusterArray");
  fprimArray->Delete();

  Int_t np=fpointArray->GetEntriesFast();
  if(np<2){
    int evNb=(FairRun::Instance())->GetEventHeader()->GetMCEntryNumber();
    TString warning=Form("TpcClusterizerTask::Exec ev:%i ",evNb);
    //Warning("TpcClusterizerTask::Exec","Not enough Hits in Tpc for Digitization (<2)");
    Warning(warning,"Not enough Hits in Tpc for Digitization (<2)");
    std::cout << ">>> TpcClusterizerTask 1st return spot" << std::endl; // ejw
    return;
  }

  if(fmereChargeConversion) {
    ChargeConversion();
    std::cout << ">>> TpcClusterizerTask 2st return spot" << std::endl; // ejw
    return; //goodbye, you wretched world!
  }
```



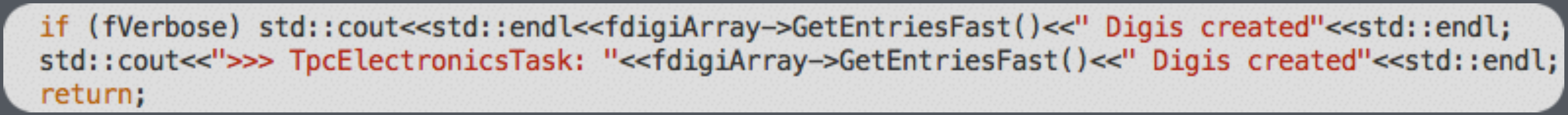
```
  std::cout<<">>> TpcGemTask: "<<favalancheArray->GetEntriesFast() <<" Avalanches created"<<std::endl;
  std::cout<<">>>             "<<counter<<" aggregations done."<<std::endl; // ejw
  return;
```



```
  } // end loop over clusters
  if (fVerbose) std::cout<<fdriftedArray->GetEntriesFast()<<" electrons arriving at readout" <<std::endl;
  std::cout<<">>> TpcDriftTask: "<<fdriftedArray->GetEntriesFast()<<" electrons arriving at readout" <<std::endl;
  return;
```



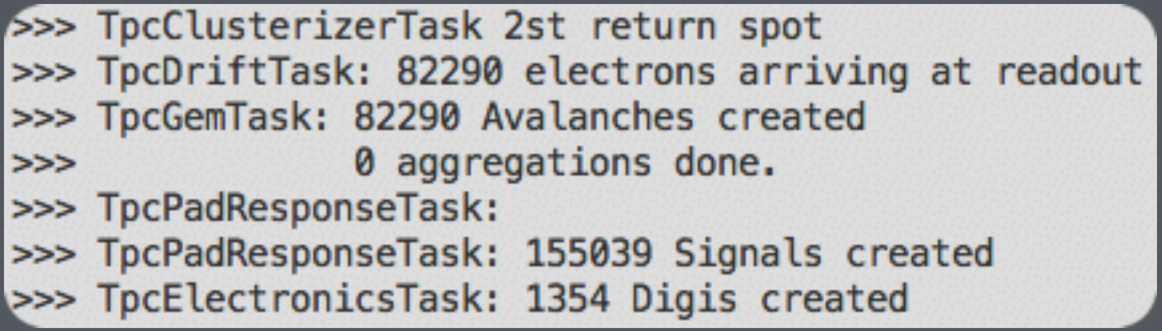
```
  ++feventcounter;
  if (fVerbose) std::cout<<fsignalArray->GetEntriesFast()<<" Signals created"<<std::endl;
  std::cout<<">>> TpcPadResponseTask: "<<fsignalArray->GetEntriesFast()<<" Signals created"<<std::endl;

  return;
```



```
 if (fVerbose) std::cout<<std::endl<<fdigiArray->GetEntriesFast()<<" Digis created"<<std::endl;
 std::cout<<">>> TpcElectronicsTask: "<<fdigiArray->GetEntriesFast()<<" Digis created"<<std::endl;
 return;
```

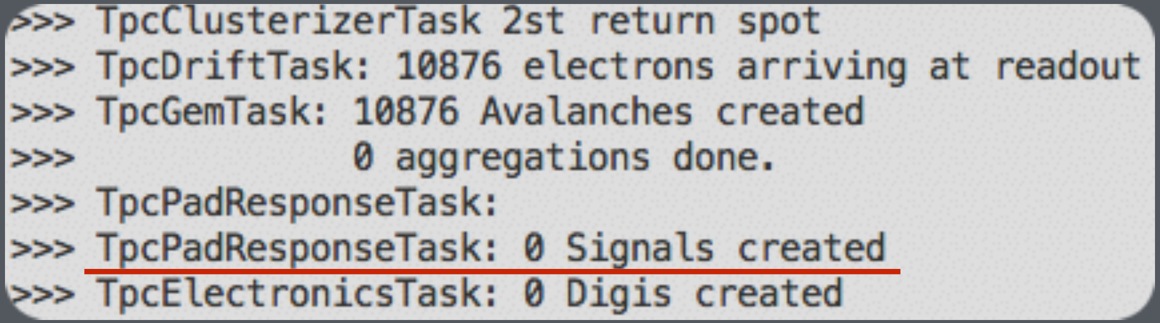
# FOPIROOT digi test >>>

## ROOT Geometry File



```
>>> TpcClusterizerTask 2st return spot
>>> TpcDriftTask: 82290 electrons arriving at readout
>>> TpcGemTask: 82290 Avalanches created
>>>             0 aggregations done.
>>> TpcPadResponseTask:
>>> TpcPadResponseTask: 155039 Signals created
>>> TpcElectronicsTask: 1354 Digis created
```

## Ascii Geometry File



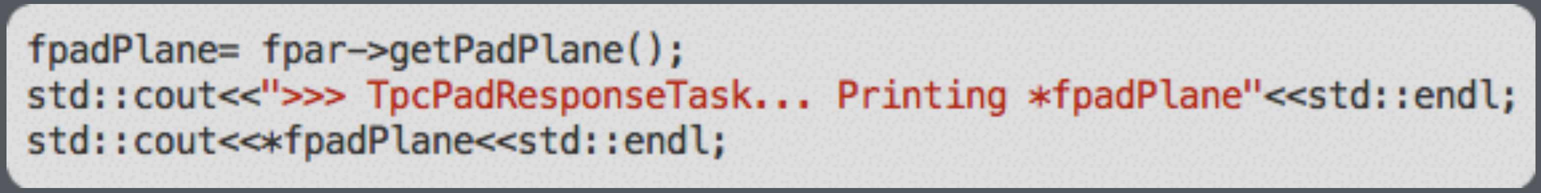
```
>>> TpcClusterizerTask 2st return spot
>>> TpcDriftTask: 10876 electrons arriving at readout
>>> TpcGemTask: 10876 Avalanches created
>>>             0 aggregations done.
>>> TpcPadResponseTask:
>>> TpcPadResponseTask: 0 Signals created
>>> TpcElectronicsTask: 0 Digis created
```

- TpcClusterizerTask ends in second return.
- Ascii geometry file MC data stops digitizing in **TpcPadResponseTask**.

# FOPIROOT TpcPadResponseTask

## TpcPadResponseTask::Init()

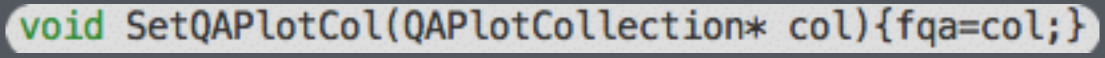
1



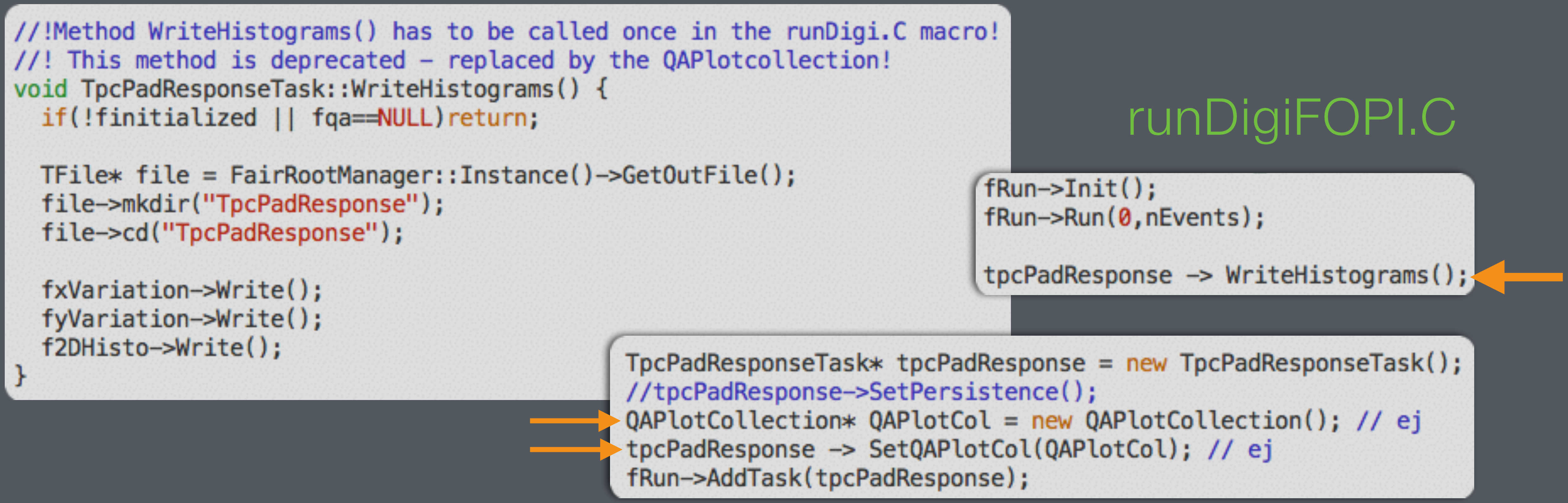
```
fpadPlane= fpar->getPadPlane();
std::cout<<">>> TpcPadResponseTask... Printing *fpadPlane"<<std::endl;
std::cout<<*fpadPlane<<std::endl;
```

## TpcPadResponseTask class

2



```
void SetQAPlotCol(QAPlotCollection* col){fqa=col;}
```



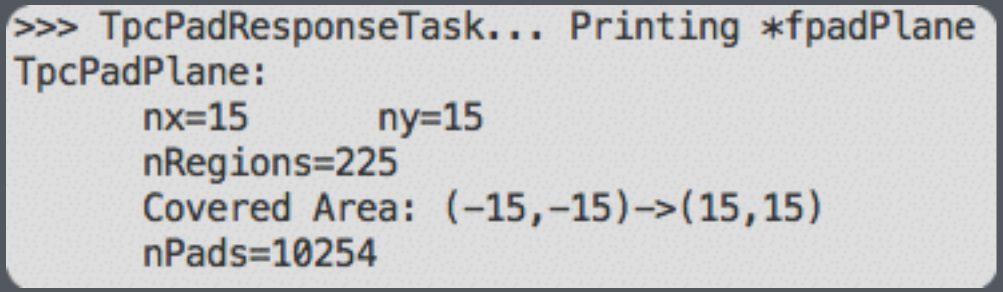
```
//!Method WriteHistograms() has to be called once in the runDigi.C macro!
//! This method is deprecated - replaced by the QAPlotcollection!
void TpcPadResponseTask::WriteHistograms() {
  if(!finitialized || fqa==NULL)return;

  TFile* file = FairRootManager::Instance()->GetOutFile();
  file->mkdir("TpcPadResponse");
  file->cd("TpcPadResponse");

  fxVariation->Write();
  fyVariation->Write();
  f2DHisto->Write();
}
```

runDigiFOPI.C

```
fRun->Init();
fRun->Run(0,nEvents);

tpcPadResponse -> WriteHistograms();
```

```
TpcPadResponseTask* tpcPadResponse = new TpcPadResponseTask();
//tpcPadResponse->SetPersistence();
QAPlotCollection* QAPlotCol = new QAPlotCollection(); // ej
tpcPadResponse -> SetQAPlotCol(QAPlotCol); // ej
fRun->AddTask(tpcPadResponse);
```

# FOPIROOT TpcPadResponseTask



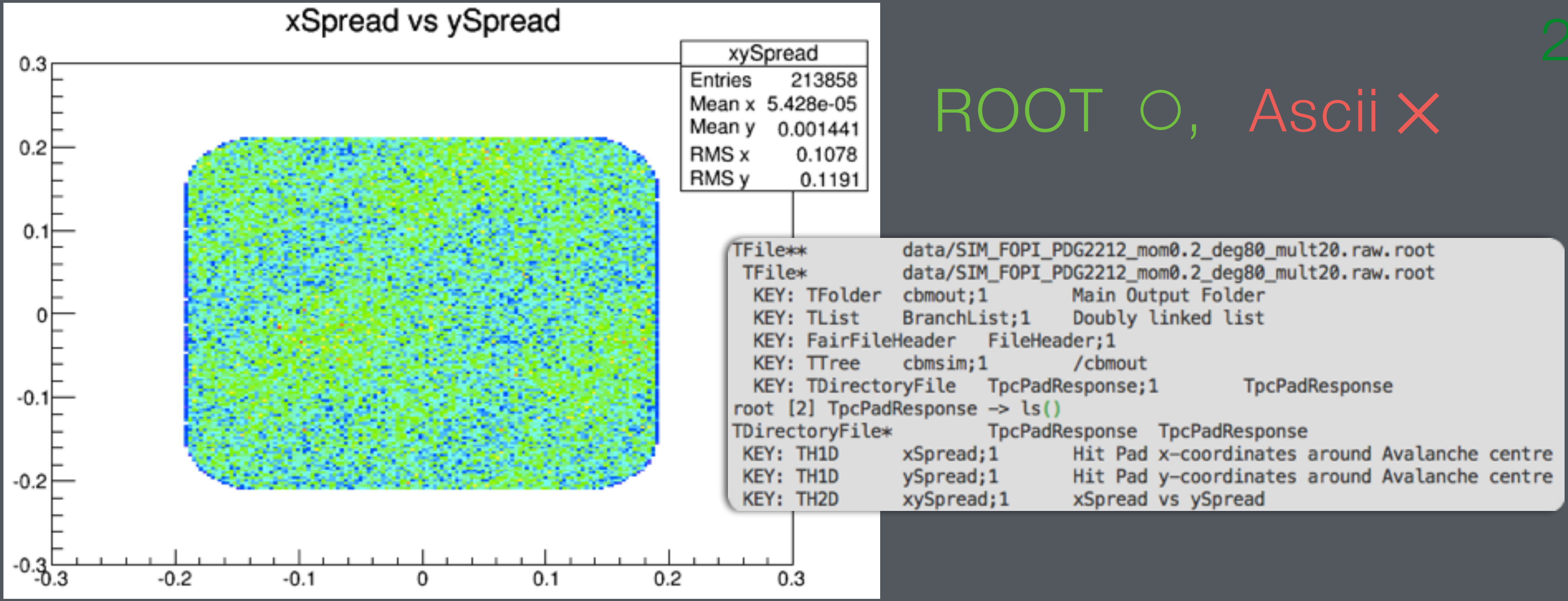
```
>>> TpcPadResponseTask... Printing *fpadPlane
TpcPadPlane:
        nx=15       ny=15
        nRegions=225
        Covered Area: (-15,-15)->(15,15)
        nPads=10254
```

ROOT ○, Ascii ○

1

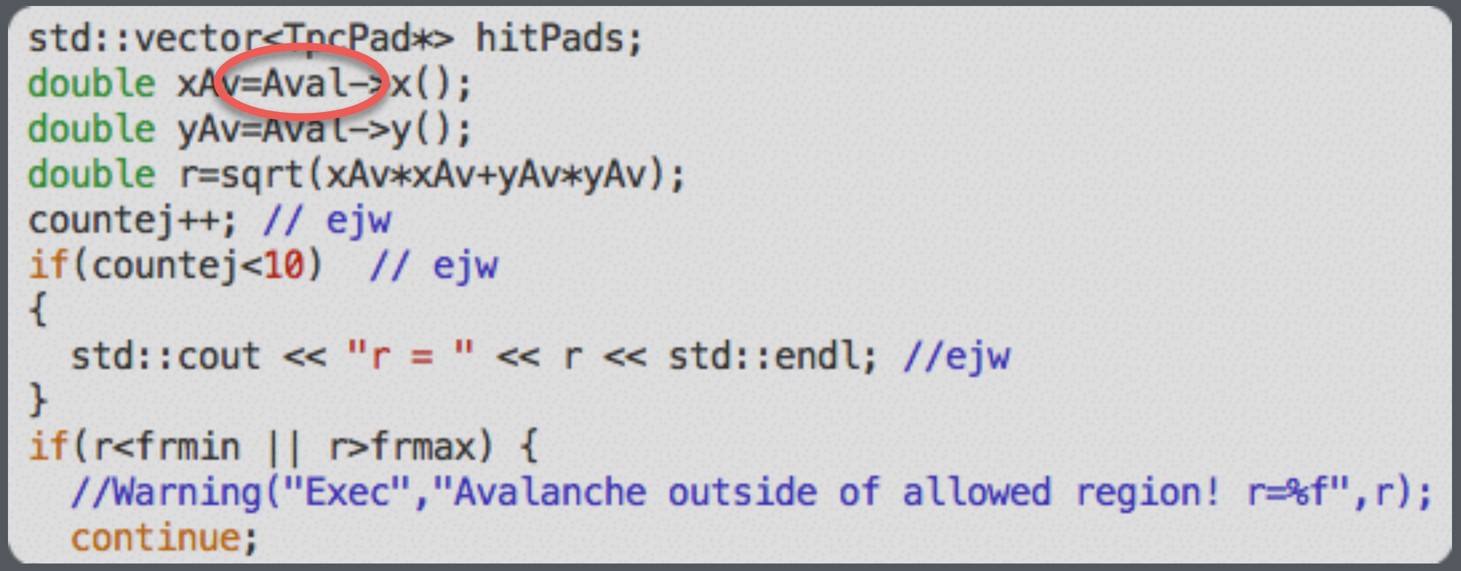
2

ROOT ○, Ascii ✕

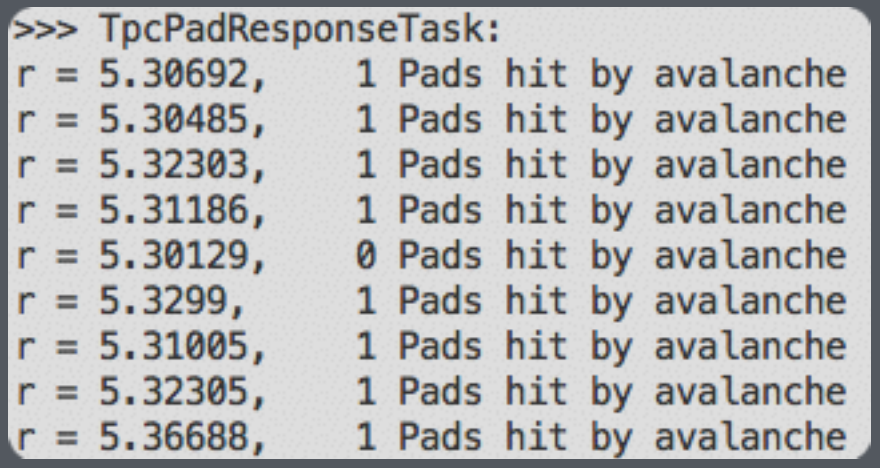


```
TFile**         data/SIM_FOPI_PDG2212_mom0.2_deg80_mult20.raw.root
 TFile*         data/SIM_FOPI_PDG2212_mom0.2_deg80_mult20.raw.root
  KEY: TFolder  cbmout;1        Main Output Folder
  KEY: TList    BranchList;1    Doubly linked list
  KEY: FairFileHeader   FileHeader;1
  KEY: TTree    cbmsim;1        /cbmout
  KEY: TDirectoryFile   TpcPadResponse;1        TpcPadResponse
root [2] TpcPadResponse -> ls()
TDirectoryFile*         TpcPadResponse  TpcPadResponse
 KEY: TH1D      xSpread;1       Hit Pad x-coordinates around Avalanche centre
 KEY: TH1D      ySpread;1       Hit Pad y-coordinates around Avalanche centre
 KEY: TH2D      xySpread;1      xSpread vs ySpread
```

# FOPIROOT TpcPadResponseTask

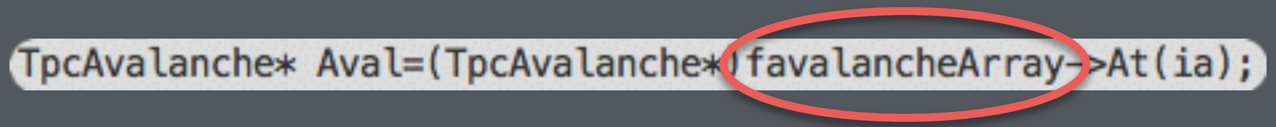


```
std::vector<TpcPad*> hitPads;
double xAv=Aval->x();
double yAv=Aval->y();
double r=sqrt(xAv*xAv+yAv*yAv);
countej++; // ejw
if(countej<10)  // ejw
{
  std::cout << "r = " << r << std::endl; //ejw
}
if(r<frmin || r>frmax) {
  //Warning("Exec","Avalanche outside of allowed region! r=%f",r);
  continue;
```

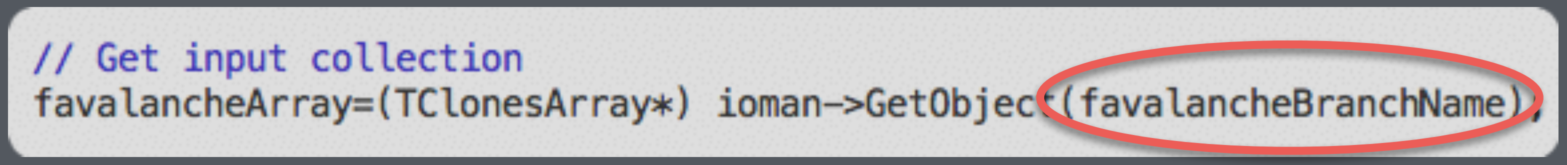


```
>>> TpcPadResponseTask:
r = 5.30692,     1 Pads hit by avalanche
r = 5.30485,     1 Pads hit by avalanche
r = 5.32303,     1 Pads hit by avalanche
r = 5.31186,     1 Pads hit by avalanche
r = 5.30129,     0 Pads hit by avalanche
r = 5.3299,      1 Pads hit by avalanche
r = 5.31005,     1 Pads hit by avalanche
r = 5.32305,     1 Pads hit by avalanche
r = 5.36688,     1 Pads hit by avalanche
```

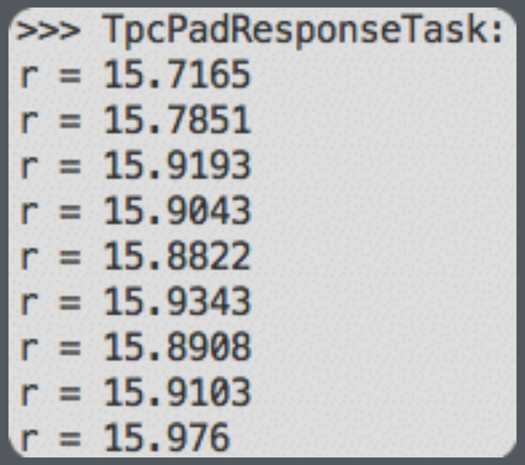
ROOT



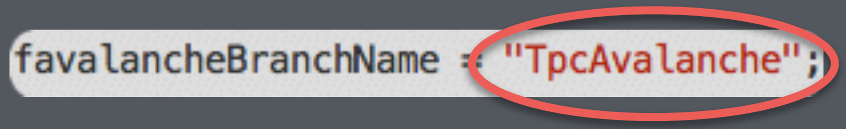
```
TpcAvalanche* Aval=(TpcAvalanche*)favalancheArray->At(ia);
```



```
// Get input collection
favalancheArray=(TClonesArray*) ioman->GetObject(favalancheBranchName);
```



```
>>> TpcPadResponseTask:
r = 15.7165
r = 15.7851
r = 15.9193
r = 15.9043
r = 15.8822
r = 15.9343
r = 15.8908
r = 15.9103
r = 15.976
```



```
favalancheBranchName = "TpcAvalanche";
```

Ascii

Avalanche are made from **TpcGemTask**!

# FOPIROOT

**Digitization Task List**

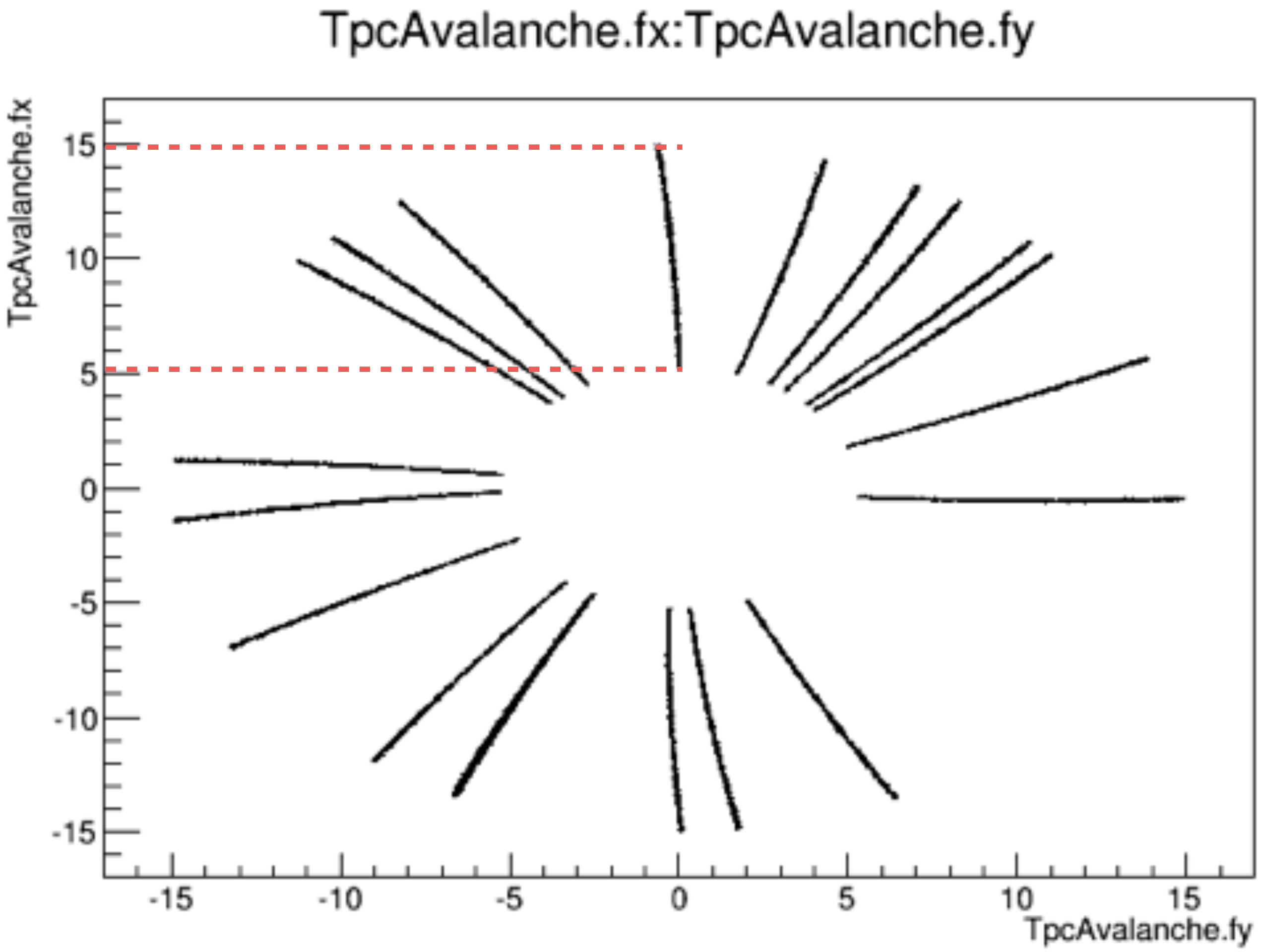
- ClusterizerTask
- DrifterTask
- GemTask
- PadResponseTask
- ElectronicsTask

SetPersistence() function saves data for each step!

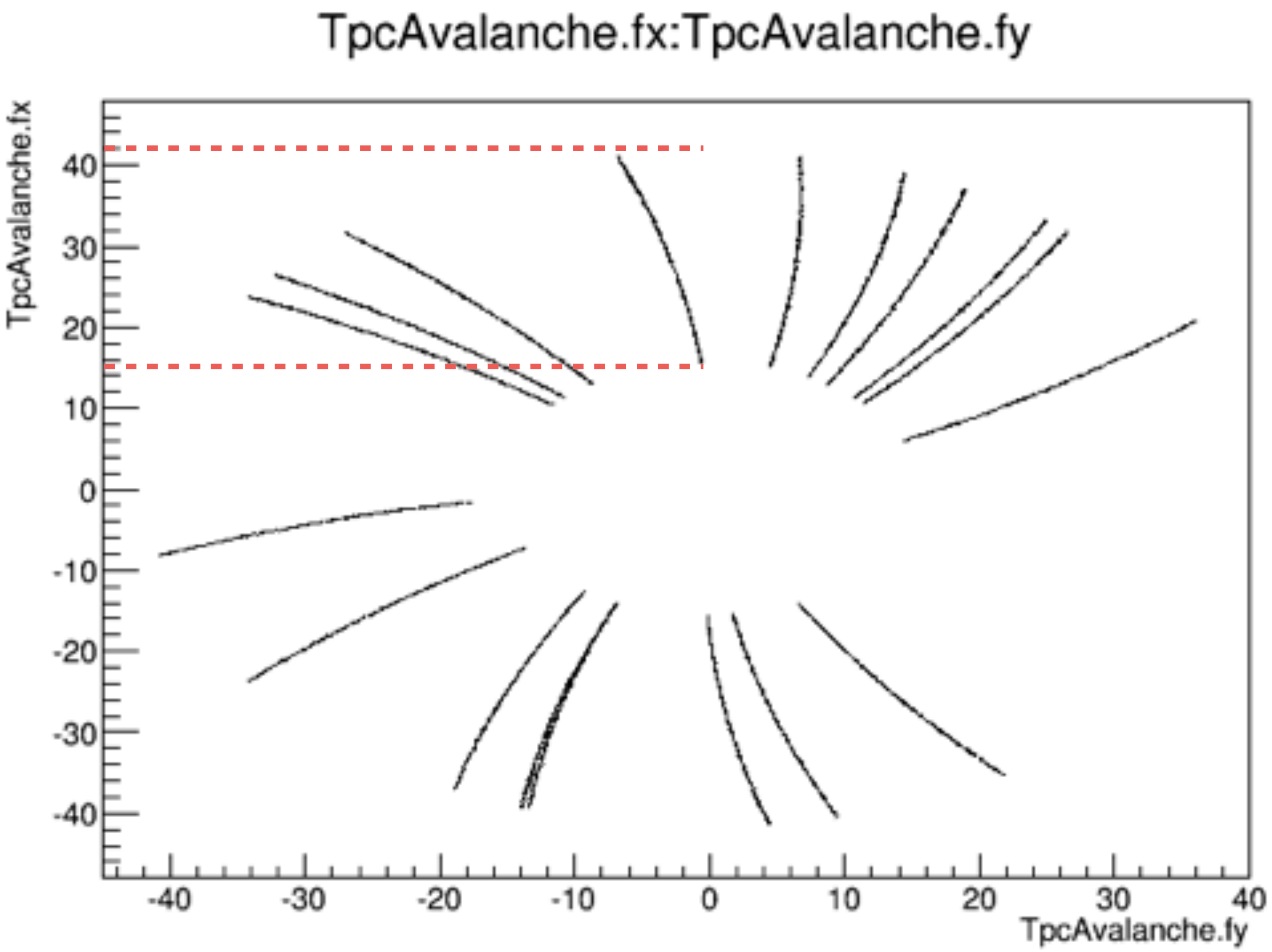
ex) tpcElec -> SetPersistence()

# FOPIROOT TpcGemTask

ROOT



Ascii



SOLVED.