```
#define AnalyzerMulti_cxx
#include "AnalyzerMulti.h"
#include <TH2.h>
#include <TGraph.h>
#include <TGraphAsymmErrors.h>
#include <TMultiGraph.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <TPaveText.h>
#include <TLegend.h>
#include <iostream>
#include <TRandom.h>
#include <TMath.h>

void AnalyzerMulti::InitFullHisto(){
  layerDepthElectron = new TProfile("layerDepthElectron","Calo Layer Depth",250,
0,250,0,24);
  layerDepthProton = new TProfile("layerDepthProton","Calo Layer Depth",250,0,25
0,0,24);

  energyAVGRangeElectron = new TProfile("energyAVGRangeElectron","Energy ratio b
etween LYSO and SCINT(AVG)",300,0,300,0,15);
  energySUMRangeElectron = new TProfile("energySUMRangeElectron","Energy ratio b
etween LYSO and SCINT(SUM)",300,0,300,0,15);

  energyAVGRangeProton = new TProfile("energyAVGRangeProton","Energy ratio betwe
en LYSO and SCINT(AVG)",300,0,300,0,15);
  energySUMRangeProton = new TProfile("energySUMRangeProton","Energy ratio betwe
en LYSO and SCINT(SUM)",300,0,300,0,15);

}

void AnalyzerMulti::CloseFullHisto(){
  Int_t nElectron = electronEnergy.size();
  Int_t nProton = protonEnergy.size();

  Double_t eleEn[nElectron],eleAcc[nElectron];
  Double_t protEn[nProton],protAcc[nProton];

  for(int i=0;i<nElectron;i++){
    eleEn[i]=electronEnergy[i];
    eleAcc[i]=electronAcceptance[i];
  }

  for(int i=0;i<nProton;i++){
    protEn[i]=protonEnergy[i];
    protAcc[i]=protonAcceptance[i];
  }

  TCanvas* c0f = new TCanvas("c0f","",200,10,700,500);

  TGraph* graphE = new TGraph(nElectron,eleEn,eleAcc);
  TGraph* graphP = new TGraph(nProton,protEn,protAcc);

  TMultiGraph* mgA = new TMultiGraph("mgA","Acceptance");

  graphE->SetName("graphE");
  graphE->SetTitle("electron");
  graphE->SetLineColor(kBlue);
  graphE->SetLineColor(kBlue);
  graphE->SetMarkerColor(kBlue);
  graphE->SetMarkerStyle(21);
  graphE->SetLineWidth(3);

  graphP->SetName("graphP");
  graphP->SetTitle("proton");
  graphP->SetLineColor(kRed);
  graphP->SetMarkerColor(kRed);
  graphP->SetMarkerStyle(21);
```

```
  graphP->SetLineWidth(3);

  mgA->Add(graphE);
  mgA->Add(graphP);

  mgA->Draw("APL");
  mgA->GetXaxis()->SetTitle("Energy [MeV]");
  mgA->GetXaxis()->CenterTitle(true);
  mgA->GetXaxis()->SetTitleSize(0.05);
  mgA->GetXaxis()->SetTitleOffset(0.90);
  mgA->GetXaxis()->SetLabelSize(0.05);
  mgA->GetYaxis()->SetTitle("cm^{2} - sr");
  mgA->GetYaxis()->CenterTitle(true);
  mgA->GetYaxis()->SetTitleSize(0.05);
  mgA->GetYaxis()->SetTitleOffset(0.90);
  mgA->GetYaxis()->SetLabelSize(0.05);

  c0f->SaveAs("Figure/Acceptance.eps");


  TCanvas* c1f = new TCanvas("c1f","",200,10,700,500);
  gStyle->SetOptStat(0);
  layerDepthElectron->SetMarkerColor(kBlue);
  layerDepthElectron->SetMarkerStyle(22);
  layerDepthElectron->SetMarkerSize(2);
  layerDepthElectron->GetXaxis()->SetTitle("Energy [MeV]");
  layerDepthElectron->GetXaxis()->CenterTitle(true);
  layerDepthElectron->GetXaxis()->SetTitleSize(0.05);
  layerDepthElectron->GetXaxis()->SetTitleOffset(0.90);
  layerDepthElectron->GetXaxis()->SetLabelSize(0.05);
  layerDepthElectron->GetYaxis()->SetTitle("<#> Layer");
  layerDepthElectron->GetYaxis()->CenterTitle(true);
  layerDepthElectron->GetYaxis()->SetTitleSize(0.05);
  layerDepthElectron->GetYaxis()->SetTitleOffset(0.90);
  layerDepthElectron->GetYaxis()->SetLabelSize(0.05);

  layerDepthProton->SetMarkerColor(kRed);
  layerDepthProton->SetMarkerStyle(23);
  layerDepthProton->SetMarkerSize(2);
  layerDepthProton->GetXaxis()->SetTitle("Energy [MeV]");
  layerDepthProton->GetXaxis()->CenterTitle(true);
  layerDepthProton->GetXaxis()->SetTitleSize(0.05);
  layerDepthProton->GetXaxis()->SetTitleOffset(0.90);
  layerDepthProton->GetXaxis()->SetLabelSize(0.05);
  layerDepthProton->GetYaxis()->SetTitle("<#> Layer");
  layerDepthProton->GetYaxis()->CenterTitle(true);
  layerDepthProton->GetYaxis()->SetTitleSize(0.05);
  layerDepthProton->GetYaxis()->SetTitleOffset(0.90);
  layerDepthProton->GetYaxis()->SetLabelSize(0.05);

  if(layerDepthElectron->GetBinContent(layerDepthElectron->GetMaximumBin())>laye
rDepthProton->GetBinContent(layerDepthProton->GetMaximumBin())){
    layerDepthElectron->Draw();
    layerDepthProton->Draw("same");
  }else{
    layerDepthProton->Draw();
    layerDepthElectron->Draw("same");
  }

  c1f->SaveAs("Figure/LayerDepth.eps");

  TCanvas* c2f = new TCanvas("c2f","",200,10,700,500);
  gStyle->SetOptStat(0);

  energyAVGRangeElectron->SetMarkerColor(kBlue);
  energyAVGRangeElectron->SetMarkerStyle(22);
  energyAVGRangeElectron->SetMarkerSize(1);
  energyAVGRangeElectron->SetTitle("");
  energyAVGRangeElectron->GetXaxis()->SetTitle("Energy [MeV]");
  energyAVGRangeElectron->GetYaxis()->SetTitle("E_{Rec LYSO}/<E_{Rec SCINT}>");
  energyAVGRangeElectron->SetLineColor(kBlue);

  energyAVGRangeElectron->Draw("BOX");

  c2f->SaveAs("Figure/TestE_AVG.eps");

  TCanvas* c3f = new TCanvas("c3f","",200,10,700,500);
  gStyle->SetOptStat(0);
  energySUMRangeElectron->SetMarkerColor(kBlue);
  energySUMRangeElectron->SetMarkerStyle(22);
  energyAVGRangeElectron->SetMarkerSize(1);
  energySUMRangeElectron->SetTitle("");
  energySUMRangeElectron->GetXaxis()->SetTitle("Energy [MeV]");
  energySUMRangeElectron->GetYaxis()->SetTitle("E_{Rec LYSO}/#Sigma E_{Rec SCINT
}");
  energySUMRangeElectron->SetLineColor(kBlue);

  energySUMRangeElectron->Draw("BOX");

  c3f->SaveAs("Figure/TestE_SUM.eps");

  TCanvas* c4f = new TCanvas("c4f","",200,10,700,500);
  gStyle->SetOptStat(0);
  energyAVGRangeProton->SetMarkerColor(kRed);
  energyAVGRangeProton->SetMarkerStyle(22);
  energyAVGRangeProton->SetMarkerSize(1);
  energyAVGRangeProton->SetTitle("");
  energyAVGRangeProton->GetXaxis()->SetTitle("Energy [MeV]");
  energyAVGRangeProton->GetYaxis()->SetTitle("E_{Rec LYSO}/<E_{Rec SCINT}>");
  energyAVGRangeProton->SetLineColor(kRed);

  energyAVGRangeProton->Draw("BOX");

  c4f->SaveAs("Figure/TestP_AVG.eps");

  TCanvas* c5f = new TCanvas("c5f","",200,10,700,500);
  gStyle->SetOptStat(0);
  energySUMRangeProton->SetMarkerColor(kRed);
  energySUMRangeProton->SetMarkerStyle(22);
  energySUMRangeProton->SetMarkerSize(1);
  energySUMRangeProton->SetTitle("");
  energySUMRangeProton->GetXaxis()->SetTitle("Energy [MeV]");
  energySUMRangeProton->GetYaxis()->SetTitle("E_{Rec LYSO}/#Sigma E_{Rec SCINT}"
);
  energySUMRangeProton->SetLineColor(kRed);

  energySUMRangeProton->Draw("BOX");

  c5f->SaveAs("Figure/TestP_SUM.eps");

}

void AnalyzerMulti::LoopElectron(float ERange)
{
  theERange=ERange;
  if (fTree == 0) return;
  Int_t binE = 200;
  Double_t eMax = 2.5;
  Int_t binA = 200;
  Double_t aRange = 50;
  Int_t binLY = 200;
  Double_t eLYMax = 2.5;
  if(theERange == 2.5){
    binE = 250;
    eMax = 2.6;
    binA = 200;
    aRange = 50;
    binLY = 200;
```

```
    eLYMax = 2.6;
  }else if(theERange == 3){
    binE =300;
    eMax =3.2;
    binA = 200;
    aRange = 50;
    binLY = 200;
    eLYMax = 3.2;
  }else if(theERange == 5){
    binE =450;
    eMax =6;
    binA = 200;
    aRange = 30;
    binLY = 200;
    eLYMax = 6;
  }else if(theERange == 10){
    binE =600;
    eMax =12;
    binA = 200;
    aRange = 30;
    binLY = 200;
    eLYMax = 12;
  }else if(theERange == 15){
    binE =600;
    eMax =17;
    binA = 200;
    aRange = 30;
    binLY = 300;
    eLYMax = 17;
  }else if(theERange == 25){
    binE =600;
    eMax =27.5;
    binA = 200;
    aRange = 15;
    binLY = 300;
    eLYMax = 27.5;
  }else if(theERange == 35){
    binE =600;
    eMax =37.5;
    binA = 200;
    aRange = 15;
    binLY = 400;
    eLYMax = 37.5;
  }else if(theERange == 45){
    binE =600;
    eMax =50;
    binA = 200;
    aRange = 15;
    binLY = 500;
    eLYMax = 50;
  }else if(theERange == 55){
    binE =600;
    eMax =60;
    binA = 200;
    aRange = 15;
    binLY = 600;
    eLYMax = 60;
  }else if(theERange == 75){
    binE =600;
    eMax =80;
    binA = 200;
    aRange = 10;
    binLY = 600;
    eLYMax = 80;
  }else if(theERange == 100){
    binE =600;
    eMax =120;
    binA = 200;
    aRange = 10;
    binLY = 600;
    eLYMax = 120;
  }else{
    binE =600;
    eMax =theERange*1.1;
    binA = 200;
    aRange = 50;
    binLY = 600;
    eLYMax = theERange*1.1;
  }

  TString histoFileName = Form("Figure/HistoElectron_%3.1f_MeV.root",theERange);

  TFile * histoFile = new TFile(histoFileName,"NEW");

  TString  paveText = Form("T_{e} = %3.1f MeV",theERange);
  TString  meanText;
  TString  rmsText;

  TH1F * thetaTotal       = new TH1F("thetaTotal","Theta Total",180,0,90);

  TH1F * siliconEdepHisto        = new TH1F("siliconEdepHisto","Energy deposited
 on silicon",binE,0,eMax);
  TH1F * scintS1EdepHisto        = new TH1F("scintS1EdepHisto","Energy deposited
 on scintillator S1",400,0,5);
  TH1F * scintS2EdepHisto        = new TH1F("scintS2EdepHisto","Energy deposited
 on scintillator S2",binE,0,eMax);
  TH1F * scintEdepHisto          = new TH1F("scintEdepHisto","Energy deposited o
n scintillator",binE,0,eMax);
  TH1F * caloEdepHisto           = new TH1F("caloEdepHisto","Energy deposited on
 calorimeter",binE,0,eMax);
  TH1F * totalEdepHisto          = new TH1F("totalEdepHisto","Total energy depos
ited on HEDP",binE,0,eMax);
  TH1F * totalEdepHistoNoCut     = new TH1F("totalEdepHistoNoCut","Total energy 
deposited on HEDP",binE,0,eMax);
  TH1F * totalEdepHistoNoS1      = new TH1F("totalEdepHistoNoS1","Total energy d
eposited on HEDP",binE,0,eMax);
  TH1F * totalEdepHistoNoS1Cut2      = new TH1F("totalEdepHistoNoS1Cut2","Total 
energy deposited on HEDP",binE,0,eMax);
  TH1F * totalEdepHistoNoS1NoCut = new TH1F("totalEdepHistoNoS1NoCut","Total ene
rgy deposited on HEDP",binE,0,eMax);
  TH1F * thetaVisible            = new TH1F("thetaVisible","Theta Visible",50,0,
90);
  TH1F * acceptance              = new TH1F("acceptance","Acceptance",50,0,90);
  TH1F * goodEventFraction              = new TH1F("goodEventFraction","GoodEven
tFraction",50,0,90);

  TH1F* thetares          = new TH1F("thetares","Angular Resolution",binA,-aRang
e,aRange);
  TH1F* thetaresws        = new TH1F("thetaresws","Angular Resolution",binA,-aRa
nge,aRange);

  TH2F* energyVSangle     = new TH2F("energyVSangle","Energy fraction vs Theta",
95,0,95,12,0,1.2);

  TH1F* bricksHitted      = new TH1F("bricksHitted","Number of hitted bricks",18
,0,18);
  TH1F* singlebrickDep    = new TH1F("singlebrickDep","Deposit on a single Brick
",binLY,0,eLYMax);
  TH1F* allbricksDep      = new TH1F("allbricksDep","Deposit on all Bricks",binL
Y,0,eLYMax);

  TProfile* layerDepElectron = new TProfile("layerDepElectron","Energy deposited
 in each detector layer",26,-0.5,25.5);

  layerDepElectron->GetXaxis()->SetBinLabel(1,"");
  layerDepElectron->GetXaxis()->SetBinLabel(2,"Tk1");
  layerDepElectron->GetXaxis()->SetBinLabel(3,"Tk2");
  layerDepElectron->GetXaxis()->SetBinLabel(4,"S1");
```

```
  layerDepElectron->GetXaxis()->SetBinLabel(5,"S2");
  layerDepElectron->GetXaxis()->SetBinLabel(6,"P1");
  layerDepElectron->GetXaxis()->SetBinLabel(7,"P2");
  layerDepElectron->GetXaxis()->SetBinLabel(8,"P3");
  layerDepElectron->GetXaxis()->SetBinLabel(9,"P4");
  layerDepElectron->GetXaxis()->SetBinLabel(10,"P5");
  layerDepElectron->GetXaxis()->SetBinLabel(11,"P6");
  layerDepElectron->GetXaxis()->SetBinLabel(12,"P7");
  layerDepElectron->GetXaxis()->SetBinLabel(13,"P8");
  layerDepElectron->GetXaxis()->SetBinLabel(14,"P9");
  layerDepElectron->GetXaxis()->SetBinLabel(15,"P10");
  layerDepElectron->GetXaxis()->SetBinLabel(16,"P11");
  layerDepElectron->GetXaxis()->SetBinLabel(17,"P12");
  layerDepElectron->GetXaxis()->SetBinLabel(18,"P13");
  layerDepElectron->GetXaxis()->SetBinLabel(19,"P14");
  layerDepElectron->GetXaxis()->SetBinLabel(20,"P15");
  layerDepElectron->GetXaxis()->SetBinLabel(21,"P16");
  layerDepElectron->GetXaxis()->SetBinLabel(22,"LYSO");


  Long64_t nentries = fTree->GetEntriesFast();
  TString fileName;
  Long64_t nbytes = 0, nb = 0;
  Double_t maxLayer = 0;
  for (Long64_t jentry=0; jentry<nentries;jentry++) {
    bool s1Hit=false;
    bool s2Hit=false;
    bool noVetoHit=false;
    bool hitOnTkLayer1=false;
    bool hitOnTkLayer2=false;
    bool goodHit=false;
    maxLayer=0;
    Long64_t ientry = LoadTree(jentry);
    if (ientry < 0) break;
    nb = fTree->GetEntry(jentry);   nbytes += nb;
    std::vector<RootTrack> myTracks = Event->GetTracks();
    std::vector<CaloRootHit> myCaloHit = Event->GetCaloHit();
    std::vector<CaloRootHit> myVetoHit = Event->GetVetoHit();
    std::vector<CaloRootHit> myScintHit = Event->GetScintHit();
    std::vector<TrackerRootHit> myTrackerHit = Event->GetTrackerHit();

    TVector3 electronDir = myTracks[0].GetDirection();
    float theta = electronDir.Theta()*180/TMath::Pi();
    if(theta>90)
      theta=180-theta;
    thetaTotal->Fill(theta);

    if(myScintHit.size()>0){
      for(size_t sh=0;sh<myScintHit.size();sh++){
        TString scintLayer = myScintHit[sh].GetVolume();
        if(scintLayer.Contains("S1"))
          s1Hit=true;
        if(scintLayer.Contains("S2"))
          s2Hit=true;
      }
      if(s1Hit&&s2Hit){
        if(myVetoHit.size()==0){
          noVetoHit=true;
        }
        if(myTrackerHit.size()>0&&noVetoHit){
          for(size_t th=0;th<myTrackerHit.size();th++){
            int detId = myTrackerHit[th].GetDetectorId();
            if(detId>200)
              hitOnTkLayer2=true;
            if(detId<200)
              hitOnTkLayer1=true;
          }
          if(hitOnTkLayer2&&hitOnTkLayer1)
            goodHit=true;
        }
      }
    }

    if(checkPos&&goodHit){
      for(size_t j=0;j<myTracks.size();j++){
        if(myTracks[j].GetTrackID()==1){
          Double_t Xpos =std::fabs(myTracks[j].GetPosition().X());
          Double_t Ypos =std::fabs(myTracks[j].GetPosition().Y());
          if(Xpos>Xlimit&&Ypos>Ylimit)
            goodHit=false;
        }
      }
    }
    if(checkTheta&&goodHit){
      Double_t myThetaWithSmearing = ComputeAngleWithSmearing(myTrackerHit,0.05)
*180/TMath::Pi();
      if(myThetaWithSmearing>Thetalimit)
        goodHit=false;
    }
    if(goodHit){
      Double_t myTheta = ComputeAngle(myTrackerHit)*180/TMath::Pi();
      Double_t myThetaWithSmearing = ComputeAngleWithSmearing(myTrackerHit,0.05)
*180/TMath::Pi();

      Double_t trackTheta=0;
      for(size_t j=0;j<myTracks.size();j++){
        if(myTracks[j].GetTrackID()==1)
          trackTheta=myTracks[j].GetDirection().Theta()*180/TMath::Pi();
      }
      if(trackTheta>90)
        trackTheta=180-trackTheta;

      thetares->Fill((myTheta-trackTheta));
      thetaresws->Fill((myThetaWithSmearing-trackTheta));

      thetaVisible->Fill(theta);
      float totalEdep   = 0;
      float totalEnoS1  = 0;
      float scintS1Edep = 0;
      float scintS2Edep = 0;
      float scintEdep   = 0;
      float caloEdep    = 0;
      float siliconEdep = 0;
      for(size_t i=0;i<myTrackerHit.size();i++){
        siliconEdep+=myTrackerHit[i].GetELoss();
        totalEdep+=myTrackerHit[i].GetELoss();
        totalEnoS1+=myTrackerHit[i].GetELoss();
        int detId = myTrackerHit[i].GetDetectorId();
        if(detId>200)
          layerDepElectron->Fill(1,myTrackerHit[i].GetELoss());
        if(detId<200)
          layerDepElectron->Fill(2,myTrackerHit[i].GetELoss());
      }
      siliconEdepHisto->Fill(siliconEdep);
      for(size_t i=0;i<myScintHit.size();i++){
        scintEdep+=myScintHit[i].GetTotalEdep();
        totalEdep+=myScintHit[i].GetTotalEdep();
        TString volume = myScintHit[i].GetVolume();
        if(volume.Contains("S1")){
          scintS1Edep+=myScintHit[i].GetTotalEdep();
          layerDepElectron->Fill(3,myScintHit[i].GetTotalEdep());
        }
        else if(volume.Contains("S2")){
          scintS2Edep+=myScintHit[i].GetTotalEdep();
          totalEnoS1+=myScintHit[i].GetTotalEdep();
          layerDepElectron->Fill(4,myScintHit[i].GetTotalEdep());
        }
      }
```

```
      scintS1EdepHisto->Fill(scintS1Edep);
      scintS2EdepHisto->Fill(scintS2Edep);
      scintEdepHisto->Fill(scintEdep);
      if(myCaloHit.size()){
        float E_Rec_Scint=0;
        float E_Rec_LYSO=0;
        int hittedBricks=0;
        float eDep = 0;
        float bricksEDep[9];
        for(int i =0;i<9;i++)
          bricksEDep[i] = 0;
        for(size_t i=0;i<myCaloHit.size();i++){
          eDep = myCaloHit[i].GetTotalEdep();;
          caloEdep+=eDep;
          totalEdep+=eDep;
          totalEnoS1+=eDep;
          TString layer = myCaloHit[i].GetVolume();
          int layernumb;
          if(numbLayerCrystal==-1&&layer.Contains("ActiveBlockCrystal")){
            layernumb = numbLayerScint+1;
            hittedBricks++;
            layer.Remove(0,18);
            bricksEDep[layer.Atoi()]+=eDep;
          }else if(layer.Contains("ActiveLayerCrystal")){
            layer.Remove(0,18);
            layernumb = numbLayerScint+numbLayerCrystal-layer.Atoi();
          }else if(layer.Contains("ActiveLayerScint")){
            layer.Remove(0,16);
            layernumb = numbLayerScint-layer.Atoi();
          }else{
            layer.Remove(0,11);
            layernumb = numbCaloLayer-layer.Atoi();
          }
          if(maxLayer<layernumb)
            maxLayer=layernumb;
          layerDepElectron->Fill(layernumb+4,eDep);
          if(layernumb== numbLayerScint+1)
            E_Rec_LYSO+=eDep;
          else
            E_Rec_Scint+=eDep;
        }
        float lysoDep=0;
        for(int i = 0;i<9;i++){
          if(bricksEDep[i]!=0){
            lysoDep+=bricksEDep[i];
            singlebrickDep->Fill(bricksEDep[i]);
          }
          if(lysoDep!=0&&i==8)
            allbricksDep->Fill(lysoDep);
        }
        bricksHitted->Fill(hittedBricks);
        caloEdepHisto->Fill(caloEdep);
        layerDepthElectron->Fill(theERange,maxLayer);
        energyAVGRangeElectron->Fill(theERange,E_Rec_LYSO/(E_Rec_Scint/numbLayer
Scint));
        energySUMRangeElectron->Fill(theERange,E_Rec_LYSO/E_Rec_Scint);
      }

      totalEdepHistoNoS1NoCut->Fill(totalEnoS1);
      totalEdepHistoNoCut->Fill(totalEdep);

      if(totalEnoS1>2.5)
        totalEdepHistoNoS1->Fill(totalEnoS1);
      if(totalEnoS1>1.5)
        totalEdepHistoNoS1Cut2->Fill(totalEnoS1);
      if(totalEdep>=2.5){
        totalEdepHisto->Fill(totalEdep);
        energyVSangle->Fill(myThetaWithSmearing,totalEdep/theERange);
      }
```

```
    }
  }

  float acc = 0;
  for(int j=0;j<50;j++){
    float bin_tot = thetaTotal->GetEntries();
    float bin_vis = thetaVisible->GetBinContent(j);
    float norm = bin_vis*45*45*TMath::Pi()/bin_tot;
    acc+=norm;
    goodEventFraction->SetBinContent(j,bin_vis/bin_tot);
    acceptance->SetBinContent(j,norm);
  }
  electronEnergy.push_back(ERange);
  electronAcceptance.push_back(acc);

  TCanvas* c0e = new TCanvas("c0e","",200,10,700,500);
  gStyle->SetOptStat(0);
  acceptance->SetTitle("");
  acceptance->GetXaxis()->SetTitle("Pitch Angle #theta [deg]");
  acceptance->GetXaxis()->CenterTitle(true);
  acceptance->GetXaxis()->SetTitleSize(0.05);
  acceptance->GetXaxis()->SetTitleOffset(0.90);
  acceptance->GetXaxis()->SetLabelSize(0.05);
  acceptance->GetYaxis()->SetTitle("cm^{2} - sr");
  acceptance->GetYaxis()->CenterTitle(true);
  acceptance->GetYaxis()->SetTitleSize(0.05);
  acceptance->GetYaxis()->SetTitleOffset(0.90);
  acceptance->GetYaxis()->SetLabelSize(0.05);
  acceptance->SetLineStyle(1);
  acceptance->SetLineColor(kBlue);
  acceptance->SetLineWidth(2);
  acceptance->Draw();
  fileName = Form("Figure/Acceptance Electron %3.1f MeV.eps",theERange);

  TPaveText* paveAcceptance = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC se
ts coords
  paveAcceptance->SetFillColor(0); // text is black on white
  paveAcceptance->SetTextSize(0.05);
  paveAcceptance->SetTextAlign(12);
  paveAcceptance->AddText(paveText);
  paveAcceptance->Draw("same");       //to draw your text object

  c0e->SaveAs(fileName);

  TCanvas* c0_1e = new TCanvas("c0_1e","",200,10,700,500);
  gStyle->SetOptStat(0);
  goodEventFraction->SetTitle("");
  goodEventFraction->GetXaxis()->SetTitle("Pitch Angle #theta [deg]");
  goodEventFraction->GetXaxis()->CenterTitle(true);
  goodEventFraction->GetXaxis()->SetTitleSize(0.05);
  goodEventFraction->GetXaxis()->SetTitleOffset(0.90);
  goodEventFraction->GetXaxis()->SetLabelSize(0.05);
  goodEventFraction->GetYaxis()->SetTitle("cm^{2} - sr");
  goodEventFraction->GetYaxis()->CenterTitle(true);
  goodEventFraction->GetYaxis()->SetTitleSize(0.05);
  goodEventFraction->GetYaxis()->SetTitleOffset(0.90);
  goodEventFraction->GetYaxis()->SetLabelSize(0.05);
  goodEventFraction->SetLineStyle(1);
  goodEventFraction->SetLineColor(kBlue);
  goodEventFraction->SetLineWidth(2);
  goodEventFraction->Draw();
  fileName = Form("Figure/GoodEventFraction Electron %3.1f MeV.eps",theERange);

  TPaveText* paveGoodEventFraction = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); //
 NDC sets coords
  paveGoodEventFraction->SetFillColor(0); // text is black on white
  paveGoodEventFraction->SetTextSize(0.05);
  paveGoodEventFraction->SetTextAlign(12);
  paveGoodEventFraction->AddText(paveText);
```

```
  paveGoodEventFraction->Draw("same");       //to draw your text object

  c0_1e->SaveAs(fileName);

  TCanvas* c1e = new TCanvas("c1e","",200,10,700,500);
  gStyle->SetOptStat(0);
  siliconEdepHisto->SetTitle("Tracker");
  siliconEdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  siliconEdepHisto->GetXaxis()->CenterTitle(true);
  siliconEdepHisto->GetXaxis()->SetTitleSize(0.05);
  siliconEdepHisto->GetXaxis()->SetTitleOffset(0.90);
  siliconEdepHisto->GetXaxis()->SetLabelSize(0.05);
  siliconEdepHisto->GetYaxis()->SetLabelSize(0.05);
  siliconEdepHisto->GetYaxis()->SetTitle("");
  siliconEdepHisto->SetLineStyle(1);
  siliconEdepHisto->SetLineColor(kBlue);
  siliconEdepHisto->SetLineWidth(2);
  siliconEdepHisto->Draw();
  fileName = Form("Figure/SiliconEdep Electron %3.1f MeV.eps",theERange);

  TPaveText* paveSilicon = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets
coords
  paveSilicon->SetFillColor(0); // text is black on white
  paveSilicon->SetTextSize(0.05);
  paveSilicon->SetTextAlign(12);
  paveSilicon->AddText(paveText);
  paveSilicon->Draw("same");       //to draw your text object

  c1e->SaveAs(fileName);

  TCanvas* c2e = new TCanvas("c2e","",200,10,700,500);
  gStyle->SetOptStat(0);
  scintEdepHisto->SetTitle("Scintillator");
  scintEdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  scintEdepHisto->GetXaxis()->CenterTitle(true);
  scintEdepHisto->GetXaxis()->SetTitleSize(0.05);
  scintEdepHisto->GetXaxis()->SetTitleOffset(0.90);
  scintEdepHisto->GetXaxis()->SetLabelSize(0.05);
  scintEdepHisto->GetYaxis()->SetLabelSize(0.05);
  scintEdepHisto->GetYaxis()->SetTitle("");
  scintEdepHisto->SetLineStyle(1);
  scintEdepHisto->SetLineColor(kBlue);
  scintEdepHisto->SetLineWidth(2);
  scintEdepHisto->Draw();
  fileName = Form("Figure/ScintEdep Electron %3.1f MeV.eps",theERange);

  TPaveText* paveScint = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets co
ords
  paveScint->SetFillColor(0); // text is black on white
  paveScint->SetTextSize(0.05);
  paveScint->SetTextAlign(12);
  paveScint->AddText(paveText);
  paveScint->Draw("same");       //to draw your text object

  c2e->SaveAs(fileName);

  TCanvas* c2e1 = new TCanvas("c2e1","",200,10,700,500);
  gStyle->SetOptStat(0);
  scintS1EdepHisto->SetTitle("Scintillator");
  scintS1EdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  scintS1EdepHisto->GetXaxis()->CenterTitle(true);
  scintS1EdepHisto->GetXaxis()->SetTitleSize(0.05);
  scintS1EdepHisto->GetXaxis()->SetTitleOffset(0.90);
  scintS1EdepHisto->GetXaxis()->SetLabelSize(0.05);
  scintS1EdepHisto->GetYaxis()->SetLabelSize(0.05);
  scintS1EdepHisto->GetYaxis()->SetTitle("");
  scintS1EdepHisto->SetLineStyle(1);
  scintS1EdepHisto->SetLineColor(kBlue);
  scintS1EdepHisto->SetLineWidth(2);
  scintS1EdepHisto->Draw();
  fileName = Form("Figure/ScintS1Edep Electron %3.1f MeV.eps",theERange);

  TPaveText* paveScintS1 = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets
coords
  paveScintS1->SetFillColor(0); // text is black on white
  paveScintS1->SetTextSize(0.05);
  paveScintS1->SetTextAlign(12);
  paveScintS1->AddText(paveText);
  paveScintS1->Draw("same");       //to draw your text object

  c2e1->SaveAs(fileName);

  TCanvas* c2e2 = new TCanvas("c2e2","",200,10,700,500);
  gStyle->SetOptStat(0);
  scintS2EdepHisto->SetTitle("Scintillator");
  scintS2EdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  scintS2EdepHisto->GetXaxis()->CenterTitle(true);
  scintS2EdepHisto->GetXaxis()->SetTitleSize(0.05);
  scintS2EdepHisto->GetXaxis()->SetTitleOffset(0.90);
  scintS2EdepHisto->GetXaxis()->SetLabelSize(0.05);
  scintS2EdepHisto->GetYaxis()->SetLabelSize(0.05);
  scintS2EdepHisto->GetYaxis()->SetTitle("");
  scintS2EdepHisto->SetLineStyle(1);
  scintS2EdepHisto->SetLineColor(kBlue);
  scintS2EdepHisto->SetLineWidth(2);
  scintS2EdepHisto->Draw();
  fileName = Form("Figure/ScintS2Edep Electron %3.1f MeV.eps",theERange);

  TPaveText* paveScintS2 = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets
coords
  paveScintS2->SetFillColor(0); // text is black on white
  paveScintS2->SetTextSize(0.05);
  paveScintS2->SetTextAlign(12);
  paveScintS2->AddText(paveText);
  paveScintS2->Draw("same");       //to draw your text object

  c2e2->SaveAs(fileName);

  TCanvas* c3e = new TCanvas("c3e","",200,10,700,500);
  gStyle->SetOptStat(0);
  caloEdepHisto->SetTitle("Calorimeter");
  caloEdepHisto->GetXaxis()->SetTitle("Deposiated Energy [MeV]");
  caloEdepHisto->GetXaxis()->CenterTitle(true);
  caloEdepHisto->GetXaxis()->SetTitleSize(0.05);
  caloEdepHisto->GetXaxis()->SetTitleOffset(0.90);
  caloEdepHisto->GetXaxis()->SetLabelSize(0.05);
  caloEdepHisto->GetYaxis()->SetLabelSize(0.05);
  caloEdepHisto->GetYaxis()->SetTitle("");
  caloEdepHisto->SetLineStyle(1);
  caloEdepHisto->SetLineColor(kBlue);
  caloEdepHisto->SetLineWidth(2);

  caloEdepHisto->Draw();

  fileName = Form("Figure/CaloEdep Electron %3.1f MeV.eps",theERange);

  TPaveText* paveCalo = new TPaveText(0.15,0.7,0.35,0.8, "NDC"); // NDC sets coo
rds
  paveCalo->SetFillColor(0); // text is black on white
  paveCalo->SetTextSize(0.05);
  paveCalo->SetTextAlign(12);
  paveCalo->AddText(paveText);
  paveCalo->Draw("same");       //to draw your text object

  c3e->SaveAs(fileName);

  TCanvas* c4e = new TCanvas("c4e","",200,10,700,500);
  gStyle->SetOptStat(0);
```

```
  totalEdepHisto->SetTitle("");
  totalEdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHisto->GetXaxis()->CenterTitle(true);
  totalEdepHisto->GetXaxis()->SetTitleSize(0.05);
  totalEdepHisto->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHisto->GetXaxis()->SetLabelSize(0.05);
  totalEdepHisto->GetYaxis()->SetLabelSize(0.05);
  totalEdepHisto->GetYaxis()->SetTitle("");
  totalEdepHisto->SetLineStyle(1);
  totalEdepHisto->SetLineColor(kBlue);
  totalEdepHisto->SetLineWidth(2);

  totalEdepHisto->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHisto->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHisto->GetRMS());
  fileName = Form("Figure/TotalEdep Electron %3.1f MeV.eps",theERange);

  TPaveText* paveTotal = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC sets c
oords
  paveTotal->SetFillColor(0); // text is black on white
  paveTotal->SetTextSize(0.05);
  paveTotal->SetTextAlign(12);
  paveTotal->AddText(paveText);
  paveTotal->AddText(meanText);
  paveTotal->AddText(rmsText);
  paveTotal->Draw("same");       //to draw your text object

  c4e->SaveAs(fileName);

  TCanvas* c4e1 = new TCanvas("c4e1","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHistoNoS1->SetTitle("");
  totalEdepHistoNoS1->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHistoNoS1->GetXaxis()->CenterTitle(true);
  totalEdepHistoNoS1->GetXaxis()->SetTitleSize(0.05);
  totalEdepHistoNoS1->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHistoNoS1->GetXaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1->GetYaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1->GetYaxis()->SetTitle("");
  totalEdepHistoNoS1->SetLineStyle(1);
  totalEdepHistoNoS1->SetLineColor(kBlue);
  totalEdepHistoNoS1->SetLineWidth(2);

  totalEdepHistoNoS1->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHistoNoS1->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHistoNoS1->GetRMS());
  fileName = Form("Figure/TotalEdepNoS1 Electron %3.1f MeV.eps",theERange);

  TPaveText* paveTotalnoS1 = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC se
ts coords
  paveTotalnoS1->SetFillColor(0); // text is black on white
  paveTotalnoS1->SetTextSize(0.05);
  paveTotalnoS1->SetTextAlign(12);
  paveTotalnoS1->AddText(paveText);
  paveTotalnoS1->AddText(meanText);
  paveTotalnoS1->AddText(rmsText);
  paveTotalnoS1->Draw("same");       //to draw your text object

  c4e1->SaveAs(fileName);

  TCanvas* c4e1c2 = new TCanvas("c4e1c2","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHistoNoS1Cut2->SetTitle("");
  totalEdepHistoNoS1Cut2->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHistoNoS1Cut2->GetXaxis()->CenterTitle(true);
  totalEdepHistoNoS1Cut2->GetXaxis()->SetTitleSize(0.05);
  totalEdepHistoNoS1Cut2->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHistoNoS1Cut2->GetXaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1Cut2->GetYaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1Cut2->GetYaxis()->SetTitle("");
  totalEdepHistoNoS1Cut2->SetLineStyle(1);
  totalEdepHistoNoS1Cut2->SetLineColor(kBlue);
  totalEdepHistoNoS1Cut2->SetLineWidth(2);

  totalEdepHistoNoS1Cut2->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHistoNoS1Cut2->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHistoNoS1Cut2->GetRMS());
  fileName = Form("Figure/TotalEdepNoS1Cut2 Electron %3.1f MeV.eps",theERange);

  TPaveText* paveTotalnoS1Cut2 = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // ND
C sets coords
  paveTotalnoS1Cut2->SetFillColor(0); // text is black on white
  paveTotalnoS1Cut2->SetTextSize(0.05);
  paveTotalnoS1Cut2->SetTextAlign(12);
  paveTotalnoS1Cut2->AddText(paveText);
  paveTotalnoS1Cut2->AddText(meanText);
  paveTotalnoS1Cut2->AddText(rmsText);
  paveTotalnoS1Cut2->Draw("same");       //to draw your text object

  c4e1c2->SaveAs(fileName);

  TCanvas* c4eNC = new TCanvas("c4eNC","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHistoNoCut->SetTitle("");
  totalEdepHistoNoCut->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHistoNoCut->GetXaxis()->CenterTitle(true);
  totalEdepHistoNoCut->GetXaxis()->SetTitleSize(0.05);
  totalEdepHistoNoCut->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHistoNoCut->GetXaxis()->SetLabelSize(0.05);
  totalEdepHistoNoCut->GetYaxis()->SetLabelSize(0.05);
  totalEdepHistoNoCut->GetYaxis()->SetTitle("");
  totalEdepHistoNoCut->SetLineStyle(1);
  totalEdepHistoNoCut->SetLineColor(kBlue);
  totalEdepHistoNoCut->SetLineWidth(2);

  totalEdepHistoNoCut->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHistoNoCut->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHistoNoCut->GetRMS());
  fileName = Form("Figure/TotalEdepNoCut Electron %3.1f MeV.eps",theERange);

  TPaveText* paveTotalNC = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC sets
 coords
  paveTotalNC->SetFillColor(0); // text is black on white
  paveTotalNC->SetTextSize(0.05);
  paveTotalNC->SetTextAlign(12);
  paveTotalNC->AddText(paveText);
  paveTotalNC->AddText(meanText);
  paveTotalNC->AddText(rmsText);
  paveTotalNC->Draw("same");       //to draw your text object

  c4eNC->SaveAs(fileName);

  TCanvas* c4e1NC = new TCanvas("c4e1NC","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHistoNoS1NoCut->SetTitle("");
  totalEdepHistoNoS1NoCut->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHistoNoS1NoCut->GetXaxis()->CenterTitle(true);
  totalEdepHistoNoS1NoCut->GetXaxis()->SetTitleSize(0.05);
  totalEdepHistoNoS1NoCut->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHistoNoS1NoCut->GetXaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1NoCut->GetYaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1NoCut->GetYaxis()->SetTitle("");
  totalEdepHistoNoS1NoCut->SetLineStyle(1);
  totalEdepHistoNoS1NoCut->SetLineColor(kBlue);
```

```
  totalEdepHistoNoS1NoCut->SetLineWidth(2);

  totalEdepHistoNoS1NoCut->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHistoNoS1NoCut->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHistoNoS1NoCut->GetRMS());
  fileName = Form("Figure/TotalEdepNoS1NoCut Electron %3.1f MeV.eps",theERange);

  TPaveText* paveTotalnoS1NC = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC
sets coords
  paveTotalnoS1NC->SetFillColor(0); // text is black on white
  paveTotalnoS1NC->SetTextSize(0.05);
  paveTotalnoS1NC->SetTextAlign(12);
  paveTotalnoS1NC->AddText(paveText);
  paveTotalnoS1NC->AddText(meanText);
  paveTotalnoS1NC->AddText(rmsText);
  paveTotalnoS1NC->Draw("same");       //to draw your text object

  c4e1NC->SaveAs(fileName);

  TCanvas* c5e = new TCanvas("c5e","",200,10,700,500);
  gStyle->SetOptStat(0);
  thetares->SetTitle("");
  thetares->GetXaxis()->SetTitle("#Delta #theta [deg]");
  thetares->GetXaxis()->CenterTitle(true);
  thetares->GetXaxis()->SetTitleSize(0.05);
  thetares->GetXaxis()->SetTitleOffset(0.90);
  thetares->GetXaxis()->SetLabelSize(0.05);
  thetares->GetYaxis()->SetLabelSize(0.05);
  thetares->GetYaxis()->SetTitle("");
  thetares->SetLineStyle(1);
  thetares->SetLineColor(kBlue);
  thetares->SetLineWidth(2);

  thetares->Draw();

  fileName = Form("Figure/Angle Resolution Electron %3.1f MeV.eps",theERange);

  TPaveText* paveTRes = new TPaveText(0.15,0.7,0.35,0.8, "NDC"); // NDC sets coo
rds
  paveTRes->SetFillColor(0); // text is black on white
  paveTRes->SetTextSize(0.05);
  paveTRes->SetTextAlign(12);
  paveTRes->AddText(paveText);
  paveTRes->Draw("same");       //to draw your text object

  c5e->SaveAs(fileName);

  TCanvas* c6e = new TCanvas("c6e","",200,10,700,500);
  gStyle->SetOptStat(0);
  thetaresws->SetTitle("");
  thetaresws->GetXaxis()->SetTitle("#Delta #theta [deg]");
  thetaresws->GetXaxis()->CenterTitle(true);
  thetaresws->GetXaxis()->SetTitleSize(0.05);
  thetaresws->GetXaxis()->SetTitleOffset(0.90);
  thetaresws->GetXaxis()->SetLabelSize(0.05);
  thetaresws->GetYaxis()->SetLabelSize(0.05);
  thetaresws->GetYaxis()->SetTitle("");
  thetaresws->SetLineStyle(1);
  thetaresws->SetLineColor(kBlue);
  thetaresws->SetLineWidth(2);

  thetaresws->Draw();

  meanText = Form("Mean = %3.1f",thetaresws->GetMean());
  rmsText = Form("RMS = %3.1f",thetaresws->GetRMS());
  fileName = Form("Figure/Angle Resolution WS Electron %3.1f MeV.eps",theERange)
;
```

```
  TPaveText* paveTResWS = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC sets
coords
  paveTResWS->SetFillColor(0); // text is black on white
  paveTResWS->SetTextSize(0.05);
  paveTResWS->SetTextAlign(12);
  paveTResWS->AddText(paveText);
  paveTResWS->AddText(meanText);
  paveTResWS->AddText(rmsText);
  paveTResWS->Draw("same");       //to draw your text object

  c6e->SaveAs(fileName);

  TCanvas* c7e = new TCanvas("c7e","",200,10,700,500);
  gStyle->SetOptStat(0);
  layerDepElectron->SetMarkerColor(kBlue);
  layerDepElectron->SetMarkerStyle(22);
  layerDepElectron->SetMarkerSize(2);
  layerDepElectron->GetXaxis()->SetTitle("# Layer");
  layerDepElectron->GetXaxis()->CenterTitle(true);
  layerDepElectron->GetXaxis()->SetTitleSize(0.05);
  layerDepElectron->GetXaxis()->SetTitleOffset(0.90);
  layerDepElectron->GetXaxis()->SetLabelSize(0.05);
  layerDepElectron->GetYaxis()->SetTitle("Energy [Mev]");
  layerDepElectron->GetYaxis()->CenterTitle(true);
  layerDepElectron->GetYaxis()->SetTitleSize(0.05);
  layerDepElectron->GetYaxis()->SetTitleOffset(0.90);
  layerDepElectron->GetYaxis()->SetLabelSize(0.05);
  layerDepElectron->Draw();

  fileName = Form("Figure/LayerDep Electron %3.1f MeV.eps",theERange);

  c7e->SaveAs(fileName);


  TCanvas* c8e = new TCanvas("c8e","",200,10,700,500);
  gStyle->SetOptStat(0);
  energyVSangle->SetTitle("");
  energyVSangle->GetXaxis()->SetTitle("#theta [deg]");
  energyVSangle->GetXaxis()->SetTitleSize(0.05);
  energyVSangle->GetXaxis()->SetTitleOffset(0.90);
  energyVSangle->GetXaxis()->SetLabelSize(0.05);
  energyVSangle->GetYaxis()->SetLabelSize(0.05);
  energyVSangle->GetYaxis()->SetTitle("E_{Rec}/E_{Gen}");
  energyVSangle->SetLineStyle(1);
  energyVSangle->SetLineColor(kBlue);
  energyVSangle->SetLineWidth(2);

  energyVSangle->Draw();

  fileName = Form("Figure/Energy Ratio VS Theta Electron %3.1f MeV.eps",theERang
e);

  TPaveText* paveTeVSa = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets co
ords
  paveTeVSa->SetFillColor(0); // text is black on white
  paveTeVSa->SetTextSize(0.05);
  paveTeVSa->SetTextAlign(12);
  paveTeVSa->AddText(paveText);
  paveTeVSa->Draw("same");       //to draw your text object

  c8e->SaveAs(fileName);

  TCanvas* c9e = new TCanvas("c9e","",200,10,700,500);
  gStyle->SetOptStat(0);
  bricksHitted->SetTitle("");
  bricksHitted->GetXaxis()->SetTitle("# LYSO Bricks");
  bricksHitted->GetXaxis()->SetTitleSize(0.05);
  bricksHitted->GetXaxis()->SetTitleOffset(0.90);
  bricksHitted->GetXaxis()->SetLabelSize(0.05);
```

```
  bricksHitted->SetLineStyle(1);
  bricksHitted->SetLineColor(kBlue);
  bricksHitted->SetLineWidth(2);

  bricksHitted->Draw();

  fileName = Form("Figure/Number LYSO Bricks Electron %3.1f MeV.eps",theERange);

  TPaveText* paveTHB = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets coor
ds
  paveTHB->SetFillColor(0); // text is black on white
  paveTHB->SetTextSize(0.05);
  paveTHB->SetTextAlign(12);
  paveTHB->AddText(paveText);
  paveTHB->Draw("same");       //to draw your text object

  c9e->SaveAs(fileName);

  TCanvas* c10e = new TCanvas("c10e","",200,10,700,500);
  gStyle->SetOptStat(0);
  singlebrickDep->SetTitle("");
  singlebrickDep->GetXaxis()->SetTitle("Energy Deposited [MeV]");
  singlebrickDep->GetXaxis()->SetTitleSize(0.05);
  singlebrickDep->GetXaxis()->SetTitleOffset(0.90);
  singlebrickDep->GetXaxis()->SetLabelSize(0.05);
  singlebrickDep->SetLineStyle(1);
  singlebrickDep->SetLineColor(kBlue);
  singlebrickDep->SetLineWidth(2);

  singlebrickDep->Draw();

  fileName = Form("Figure/Energy Deposited LYSO Brick Electron %3.1f MeV.eps",th
eERange);

  TPaveText* paveTEDB = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets coo
rds
  paveTEDB->SetFillColor(0); // text is black on white
  paveTEDB->SetTextSize(0.05);
  paveTEDB->SetTextAlign(12);
  paveTEDB->AddText(paveText);
  paveTEDB->Draw("same");       //to draw your text object

  c10e->SaveAs(fileName);

  TCanvas* c11e = new TCanvas("c11e","",200,10,700,500);
  gStyle->SetOptStat(0);
  allbricksDep->SetTitle("");
  allbricksDep->GetXaxis()->SetTitle("Energy Deposited [MeV]");
  allbricksDep->GetXaxis()->SetTitleSize(0.05);
  allbricksDep->GetXaxis()->SetTitleOffset(0.90);
  allbricksDep->GetXaxis()->SetLabelSize(0.05);
  allbricksDep->SetLineStyle(1);
  allbricksDep->SetLineColor(kBlue);
  allbricksDep->SetLineWidth(2);

  allbricksDep->Draw();

  fileName = Form("Figure/Energy Deposited All LYSO Brick Electron %3.1f MeV.eps
",theERange);

  TPaveText* paveTEDBA = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets co
ords
  paveTEDBA->SetFillColor(0); // text is black on white
  paveTEDBA->SetTextSize(0.05);
  paveTEDBA->SetTextAlign(12);
  paveTEDBA->AddText(paveText);
  paveTEDBA->Draw("same");       //to draw your text object

  c11e->SaveAs(fileName);
  histoFile->Write();
  histoFile->Close();

}


void AnalyzerMulti::LoopProton(float ERange)
{
  theERange=ERange;
  if (fTree == 0) return;
  Int_t binE = 400;
  Double_t eMax = 30;
  Int_t binA = 200;
  Double_t aRange = 50;
  Int_t binLY = 200;
  Double_t eLYMax = 30;
  if(theERange == 30){
    binE = 800;
    eMax = 32.5;
    binA = 200;
    aRange = 3;
    binLY = 800;
    eLYMax = 32.5;
  }else if(theERange == 35){
    binE =800;
    eMax =37.5;
    binA = 200;
    aRange = 3;
    binLY = 800;
    eLYMax = 37.5;
  }else if(theERange == 55){
    binE =1000;
    eMax =57.5;
    binA = 200;
    aRange = 2;
    binLY = 800;
    eLYMax = 57.5;
  }else if(theERange == 75){
    binE =1000;
    eMax =77.5;
    binA = 200;
    aRange = 2;
    binLY = 800;
    eLYMax = 77.5;
  }else if(theERange == 100){
    binE =800;
    eMax =105;
    binA = 200;
    aRange = 2;
    binLY = 800;
    eLYMax = 105;
  }else if(theERange == 125){
    binE =800;
    eMax =130;
    binA = 200;
    aRange = 1.5;
    binLY = 800;
    eLYMax = 130;
  }else if(theERange == 150){
    binE =800;
    eMax =155;
    binA = 200;
    aRange = 1.5;
    binLY = 800;
    eLYMax = 155;
  }else if(theERange == 200){
    binE =800;
    eMax =210;
```

```
    binA = 200;
    aRange = 1.5;
    binLY = 800;
    eLYMax = 210;
  }else{
    binE =800;
    eMax =theERange*1.1;
    binA = 200;
    aRange = 30;
    binLY = 800;
    eLYMax = theERange*1.1;
  }

  TString histoFileName = Form("Figure/HistoProton_%3.1f_MeV.root",theERange);

  TFile * histoFile = new TFile(histoFileName,"NEW");

  TString  paveText = Form("T_{p} = %3.1f MeV",theERange);
  TString  meanText;
  TString  rmsText;

  TH1F * thetaTotal       = new TH1F("thetaTotal","Theta Total",180,0,90);

  TH1F * siliconEdepHisto = new TH1F("siliconEdepHisto","Energy deposited on sil
icon",binE,0,eMax);
  TH1F * scintS1EdepHisto   = new TH1F("scintS1EdepHisto","Energy deposited on s
cintillator S1",400,0,25);
  TH1F * scintS2EdepHisto   = new TH1F("scintS2EdepHisto","Energy deposited on s
cintillator S2",binE,0,eMax);
  TH1F * scintEdepHisto   = new TH1F("scintEdepHisto","Energy deposited on scint
illator",binE,0,eMax);
  TH1F * caloEdepHisto    = new TH1F("caloEdepHisto","Energy deposited on calori
meter",binE,0,eMax);
  TH1F * totalEdepHisto   = new TH1F("totalEdepHisto","Total energy deposited on
 HEDP",binE,0,eMax);
  TH1F * totalEdepHistoNoCut     = new TH1F("totalEdepHistoNoCut","Total energy 
deposited on HEDP",binE,0,eMax);
  TH1F * totalEdepHistoNoS1 = new TH1F("totalEdepHistoNoS1","Total energy deposi
ted on HEDP",binE,0,eMax);
  TH1F * totalEdepHistoNoS1Cut2 = new TH1F("totalEdepHistoNoS1Cut2","Total energ
y deposited on HEDP",binE,0,eMax);
  TH1F * totalEdepHistoNoS1NoCut = new TH1F("totalEdepHistoNoS1NoCut","Total ene
rgy deposited on HEDP",binE,0,eMax);
  TH1F * thetaVisible     = new TH1F("thetaVisible","Theta Visible",50,0,90);
  TH1F * acceptance       = new TH1F("acceptance","Acceptance",50,0,90);
  TH1F * goodEventFraction       = new TH1F("goodEventFraction","GoodEventFracti
on",50,0,90);

  TH1F* thetares          = new TH1F("thetares","Angular Resolution",binA,-aRang
e,aRange);
  TH1F* thetaresws        = new TH1F("thetaresws","Angular Resolution",binA,-aRa
nge,aRange);

  TH2F* energyVSangle     = new TH2F("energyVSangle","Energy fraction vs Theta",
95,0,95,12,0,1.2);

  TH1F* bricksHitted      = new TH1F("bricksHitted","Number of hitted bricks",18
,0,18);
  TH1F* singlebrickDep   = new TH1F("singlebrickDep","Deposit on a single Brick"
,binLY,0,eLYMax);
  TH1F* allbricksDep      = new TH1F("allbricksDep","Deposit on all Bricks",binL
Y,0,eLYMax);

  TProfile* layerDepProton = new TProfile("layerDepProton","Energy deposited in 
each detector layer",26,-0.5,25.5);

  layerDepProton->GetXaxis()->SetBinLabel(1,"");
  layerDepProton->GetXaxis()->SetBinLabel(2,"Tk1");
  layerDepProton->GetXaxis()->SetBinLabel(3,"Tk2");
  layerDepProton->GetXaxis()->SetBinLabel(4,"S1");
  layerDepProton->GetXaxis()->SetBinLabel(5,"S2");
  layerDepProton->GetXaxis()->SetBinLabel(6,"P1");
  layerDepProton->GetXaxis()->SetBinLabel(7,"P2");
  layerDepProton->GetXaxis()->SetBinLabel(8,"P3");
  layerDepProton->GetXaxis()->SetBinLabel(9,"P4");
  layerDepProton->GetXaxis()->SetBinLabel(10,"P5");
  layerDepProton->GetXaxis()->SetBinLabel(11,"P6");
  layerDepProton->GetXaxis()->SetBinLabel(12,"P7");
  layerDepProton->GetXaxis()->SetBinLabel(13,"P8");
  layerDepProton->GetXaxis()->SetBinLabel(14,"P9");
  layerDepProton->GetXaxis()->SetBinLabel(15,"P10");
  layerDepProton->GetXaxis()->SetBinLabel(16,"P11");
  layerDepProton->GetXaxis()->SetBinLabel(17,"P12");
  layerDepProton->GetXaxis()->SetBinLabel(18,"P13");
  layerDepProton->GetXaxis()->SetBinLabel(19,"P14");
  layerDepProton->GetXaxis()->SetBinLabel(20,"P15");
  layerDepProton->GetXaxis()->SetBinLabel(21,"P16");
  layerDepProton->GetXaxis()->SetBinLabel(22,"LYSO");

  Long64_t nentries = fTree->GetEntriesFast();
  TString fileName;
  Long64_t nbytes = 0, nb = 0;
  Double_t maxLayer = 0;
  for (Long64_t jentry=0; jentry<nentries;jentry++) {
    bool s1Hit=false;
    bool s2Hit=false;
    bool noVetoHit=false;
    bool hitOnTkLayer1=false;
    bool hitOnTkLayer2=false;
    bool goodHit=false;
    maxLayer=0;
    Long64_t ientry = LoadTree(jentry);
    if (ientry < 0) break;
    nb = fTree->GetEntry(jentry);   nbytes += nb;
    std::vector<RootTrack> myTracks = Event->GetTracks();
    std::vector<CaloRootHit> myCaloHit = Event->GetCaloHit();
    std::vector<CaloRootHit> myVetoHit = Event->GetVetoHit();
    std::vector<CaloRootHit> myScintHit = Event->GetScintHit();
    std::vector<TrackerRootHit> myTrackerHit = Event->GetTrackerHit();

    TVector3 protonDir = myTracks[0].GetDirection();
    float theta = protonDir.Theta()*180/TMath::Pi();
    if(theta>90)
      theta=180-theta;
    thetaTotal->Fill(theta);

    if(myScintHit.size()>0){
      for(size_t sh=0;sh<myScintHit.size();sh++){
        TString scintLayer = myScintHit[sh].GetVolume();
        if(scintLayer.Contains("S1"))
          s1Hit=true;
        if(scintLayer.Contains("S2"))
          s2Hit=true;
      }
      if(s1Hit&&s2Hit){
        if(myVetoHit.size()==0){
          noVetoHit=true;
        }
        if(myTrackerHit.size()>0&&noVetoHit){
          for(size_t th=0;th<myTrackerHit.size();th++){
            int detId = myTrackerHit[th].GetDetectorId();
            if(detId>200)
              hitOnTkLayer2=true;
            if(detId<200)
              hitOnTkLayer1=true;
          }
          if(hitOnTkLayer2&&hitOnTkLayer1)
            goodHit=true;
```

```
        }
      }
    }
    if(checkPos&&goodHit){
      for(size_t j=0;j<myTracks.size();j++){
        if(myTracks[j].GetTrackID()==1){
          Double_t Xpos =std::fabs(myTracks[j].GetPosition().X());
          Double_t Ypos =std::fabs(myTracks[j].GetPosition().Y());
          if(Xpos>Xlimit&&Ypos>Ylimit)
            goodHit=false;
        }
      }
    }
    if(checkTheta&&goodHit){
      Double_t myThetaWithSmearing = ComputeAngleWithSmearing(myTrackerHit,0.05)
*180/TMath::Pi();
      if(myThetaWithSmearing>Thetalimit)
        goodHit=false;
    }
    if(goodHit){
      Double_t myTheta = ComputeAngle(myTrackerHit)*180/TMath::Pi();
      Double_t myThetaWithSmearing = ComputeAngleWithSmearing(myTrackerHit,0.05)
*180/TMath::Pi();

      Double_t trackTheta=0;
      for(size_t j=0;j<myTracks.size();j++){
        if(myTracks[j].GetTrackID()==1)
          trackTheta=myTracks[j].GetDirection().Theta()*180/TMath::Pi();
      }
      if(trackTheta>90)
        trackTheta=180-trackTheta;

      thetares->Fill((myTheta-trackTheta));
      thetaresws->Fill((myThetaWithSmearing-trackTheta));

      thetaVisible->Fill(theta);
      float totalEdep   = 0;
      float totalEnoS1  = 0;
      float scintS1Edep = 0;
      float scintS2Edep = 0;
      float scintEdep   = 0;
      float caloEdep    = 0;
      float siliconEdep = 0;
      for(size_t i=0;i<myTrackerHit.size();i++){
        siliconEdep+=myTrackerHit[i].GetELoss();
        totalEdep+=myTrackerHit[i].GetELoss();
        int detId = myTrackerHit[i].GetDetectorId();
        if(detId>200)
          layerDepProton->Fill(1,myTrackerHit[i].GetELoss());
        if(detId<200)
          layerDepProton->Fill(2,myTrackerHit[i].GetELoss());
      }
      siliconEdepHisto->Fill(siliconEdep);
      for(size_t i=0;i<myScintHit.size();i++){
        scintEdep+=myScintHit[i].GetTotalEdep();
        totalEdep+=myScintHit[i].GetTotalEdep();
        TString volume = myScintHit[i].GetVolume();
        if(volume.Contains("S1")){
          scintS1Edep+=myScintHit[i].GetTotalEdep();
          layerDepProton->Fill(3,myScintHit[i].GetTotalEdep());
        }
        else if(volume.Contains("S2")){
          scintS2Edep+=myScintHit[i].GetTotalEdep();
          totalEnoS1+=myScintHit[i].GetTotalEdep();
          layerDepProton->Fill(4,myScintHit[i].GetTotalEdep());
        }
      }
      scintS1EdepHisto->Fill(scintS1Edep);
      scintS2EdepHisto->Fill(scintS2Edep);
      scintEdepHisto->Fill(scintEdep);
      if(myCaloHit.size()){
        float E_Rec_Scint=0;
        float E_Rec_LYSO=0;
        int hittedBricks=0;
        float eDep = 0;
        float bricksEDep[9];
        for(int i =0;i<9;i++)
          bricksEDep[i] = 0;
        for(size_t i=0;i<myCaloHit.size();i++){
          eDep = myCaloHit[i].GetTotalEdep();
          caloEdep+=eDep;
          totalEdep+=eDep;
          totalEnoS1+=eDep;
          TString layer = myCaloHit[i].GetVolume();
          int layernumb;
          if(numbLayerCrystal==-1&&layer.Contains("ActiveBlockCrystal")){
            layernumb = numbLayerScint+1;
            hittedBricks++;
            layer.Remove(0,18);
            bricksEDep[layer.Atoi()]+=eDep;
          }else if(layer.Contains("ActiveLayerCrystal")){
            layer.Remove(0,18);
            layernumb = numbLayerScint+numbLayerCrystal-layer.Atoi();
          }else if(layer.Contains("ActiveLayerScint")){
            layer.Remove(0,16);
            layernumb = numbLayerScint-layer.Atoi();
          }else{
            layer.Remove(0,11);
            layernumb = numbCaloLayer-layer.Atoi();
          }
          if(maxLayer<layernumb)
            maxLayer=layernumb;
          layerDepProton->Fill(layernumb+4,eDep);
          if(layernumb== numbLayerScint+1)
            E_Rec_LYSO+=eDep;
          else
            E_Rec_Scint+=eDep;
        }
        float lysoDep=0;
        for(int i = 0;i<9;i++){
          if(bricksEDep[i]!=0){
            lysoDep+=bricksEDep[i];
            singlebrickDep->Fill(bricksEDep[i]);
          }
          if(lysoDep!=0&&i==8)
            allbricksDep->Fill(lysoDep);
        }
        bricksHitted->Fill(hittedBricks);
        caloEdepHisto->Fill(caloEdep);
        layerDepthProton->Fill(theERange,maxLayer);
        energyAVGRangeProton->Fill(theERange,E_Rec_LYSO/(E_Rec_Scint/numbLayerSc
int));
        energySUMRangeProton->Fill(theERange,E_Rec_LYSO/E_Rec_Scint);
      }


      totalEdepHistoNoS1NoCut->Fill(totalEnoS1);
      totalEdepHistoNoCut->Fill(totalEdep);

      if(totalEnoS1>2.5)
        totalEdepHistoNoS1->Fill(totalEnoS1);
      if(totalEnoS1>1.5)
        totalEdepHistoNoS1Cut2->Fill(totalEnoS1);

      if(totalEdep>=2.5){
        totalEdepHisto->Fill(totalEdep);
        energyVSangle->Fill(myThetaWithSmearing,totalEdep/theERange);
      }
```

```
    }
  }

  float acc = 0;
  for(int j=0;j<50;j++){
    float bin_tot = thetaTotal->GetEntries();
    float bin_vis = thetaVisible->GetBinContent(j);
    float norm = bin_vis*45*45*TMath::Pi()/bin_tot;
    acc+=norm;
    acceptance->SetBinContent(j,norm);
    goodEventFraction->SetBinContent(j,bin_vis/bin_tot);
  }
  protonEnergy.push_back(ERange);
  protonAcceptance.push_back(acc);

  TCanvas* c0p = new TCanvas("c0p","",200,10,700,500);
  gStyle->SetOptStat(0);
  acceptance->SetTitle("");
  acceptance->GetXaxis()->SetTitle("Pitch Angle #theta [deg]");
  acceptance->GetXaxis()->CenterTitle(true);
  acceptance->GetXaxis()->SetTitleSize(0.05);
  acceptance->GetXaxis()->SetTitleOffset(0.90);
  acceptance->GetXaxis()->SetLabelSize(0.05);
  acceptance->GetYaxis()->SetTitle("cm^{2} - sr");
  acceptance->GetYaxis()->CenterTitle(true);
  acceptance->GetYaxis()->SetTitleSize(0.05);
  acceptance->GetYaxis()->SetTitleOffset(0.90);
  acceptance->GetYaxis()->SetLabelSize(0.05);
  acceptance->SetLineStyle(1);
  acceptance->SetLineColor(kRed);
  acceptance->SetLineWidth(2);
  acceptance->Draw();
  fileName = Form("Figure/Acceptance Proton %3.1f MeV.eps",theERange);

  TPaveText* paveAcceptance = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC se
ts coords
  paveAcceptance->SetFillColor(0); // text is black on white
  paveAcceptance->SetTextSize(0.05);
  paveAcceptance->SetTextAlign(12);
  paveAcceptance->AddText(paveText);
  paveAcceptance->Draw("same");       //to draw your text object

  c0p->SaveAs(fileName);

  TCanvas* c0_p = new TCanvas("c0_p","",200,10,700,500);
  gStyle->SetOptStat(0);
  goodEventFraction->SetTitle("");
  goodEventFraction->GetXaxis()->SetTitle("Pitch Angle #theta [deg]");
  goodEventFraction->GetXaxis()->CenterTitle(true);
  goodEventFraction->GetXaxis()->SetTitleSize(0.05);
  goodEventFraction->GetXaxis()->SetTitleOffset(0.90);
  goodEventFraction->GetXaxis()->SetLabelSize(0.05);
  goodEventFraction->GetYaxis()->SetTitle("cm^{2} - sr");
  goodEventFraction->GetYaxis()->CenterTitle(true);
  goodEventFraction->GetYaxis()->SetTitleSize(0.05);
  goodEventFraction->GetYaxis()->SetTitleOffset(0.90);
  goodEventFraction->GetYaxis()->SetLabelSize(0.05);
  goodEventFraction->SetLineStyle(1);
  goodEventFraction->SetLineColor(kRed);
  goodEventFraction->SetLineWidth(2);
  goodEventFraction->Draw();
  fileName = Form("Figure/GoodEventFraction Proton %3.1f MeV.eps",theERange);

  TPaveText* paveGoodEventFraction = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); //
 NDC sets coords
  paveGoodEventFraction->SetFillColor(0); // text is black on white
  paveGoodEventFraction->SetTextSize(0.05);
  paveGoodEventFraction->SetTextAlign(12);
  paveGoodEventFraction->AddText(paveText);
  paveGoodEventFraction->Draw("same");       //to draw your text object

  c0_p->SaveAs(fileName);

  TCanvas* c1p = new TCanvas("c1p","",200,10,700,500);
  gStyle->SetOptStat(0);
  siliconEdepHisto->SetTitle("Tracker");
  siliconEdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  siliconEdepHisto->GetXaxis()->CenterTitle(true);
  siliconEdepHisto->GetXaxis()->SetTitleSize(0.05);
  siliconEdepHisto->GetXaxis()->SetTitleOffset(0.90);
  siliconEdepHisto->GetXaxis()->SetLabelSize(0.05);
  siliconEdepHisto->GetYaxis()->SetLabelSize(0.05);
  siliconEdepHisto->GetYaxis()->SetTitle("");
  siliconEdepHisto->SetLineStyle(1);
  siliconEdepHisto->SetLineColor(kRed);
  siliconEdepHisto->SetLineWidth(2);
  siliconEdepHisto->Draw();
  fileName = Form("Figure/SiliconEdep Proton %3.1f MeV.eps",theERange);


  TPaveText* paveSilicon = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets
coords
  paveSilicon->SetFillColor(0); // text is black on white
  paveSilicon->SetTextSize(0.05);
  paveSilicon->SetTextAlign(12);
  paveSilicon->AddText(paveText);
  paveSilicon->Draw("same");       //to draw your text object

  c1p->SaveAs(fileName);

  TCanvas* c2p = new TCanvas("c2p","",200,10,700,500);
  gStyle->SetOptStat(0);
  scintEdepHisto->SetTitle("Scintillator");
  scintEdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  scintEdepHisto->GetXaxis()->CenterTitle(true);
  scintEdepHisto->GetXaxis()->SetTitleSize(0.05);
  scintEdepHisto->GetXaxis()->SetTitleOffset(0.90);
  scintEdepHisto->GetXaxis()->SetLabelSize(0.05);
  scintEdepHisto->GetYaxis()->SetLabelSize(0.05);
  scintEdepHisto->GetYaxis()->SetTitle("");
  scintEdepHisto->SetLineStyle(1);
  scintEdepHisto->SetLineColor(kRed);
  scintEdepHisto->SetLineWidth(2);
  scintEdepHisto->Draw();
  fileName = Form("Figure/ScintEdep Proton %3.1f MeV.eps",theERange);

  TPaveText* paveScint = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets co
ords
  paveScint->SetFillColor(0); // text is black on white
  paveScint->SetTextSize(0.05);
  paveScint->SetTextAlign(12);
  paveScint->AddText(paveText);
  paveScint->Draw("same");       //to draw your text object

  c2p->SaveAs(fileName);

  TCanvas* c2p1 = new TCanvas("c2p1","",200,10,700,500);
  gStyle->SetOptStat(0);
  scintS1EdepHisto->SetTitle("Scintillator");
  scintS1EdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  scintS1EdepHisto->GetXaxis()->CenterTitle(true);
  scintS1EdepHisto->GetXaxis()->SetTitleSize(0.05);
  scintS1EdepHisto->GetXaxis()->SetTitleOffset(0.90);
  scintS1EdepHisto->GetXaxis()->SetLabelSize(0.05);
  scintS1EdepHisto->GetYaxis()->SetLabelSize(0.05);
  scintS1EdepHisto->GetYaxis()->SetTitle("");
  scintS1EdepHisto->SetLineStyle(1);
  scintS1EdepHisto->SetLineColor(kRed);
```

```
  scintS1EdepHisto->SetLineWidth(2);
  scintS1EdepHisto->Draw();
  fileName = Form("Figure/ScintS1Edep Proton %3.1f MeV.eps",theERange);

  TPaveText* paveScintS1 = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets
coords
  paveScintS1->SetFillColor(0); // text is black on white
  paveScintS1->SetTextSize(0.05);
  paveScintS1->SetTextAlign(12);
  paveScintS1->AddText(paveText);
  paveScintS1->Draw("same");       //to draw your text object

  c2p1->SaveAs(fileName);

  TCanvas* c2p2 = new TCanvas("c2p2","",200,10,700,500);
  gStyle->SetOptStat(0);
  scintS2EdepHisto->SetTitle("Scintillator");
  scintS2EdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  scintS2EdepHisto->GetXaxis()->CenterTitle(true);
  scintS2EdepHisto->GetXaxis()->SetTitleSize(0.05);
  scintS2EdepHisto->GetXaxis()->SetTitleOffset(0.90);
  scintS2EdepHisto->GetXaxis()->SetLabelSize(0.05);
  scintS2EdepHisto->GetYaxis()->SetLabelSize(0.05);
  scintS2EdepHisto->GetYaxis()->SetTitle("");
  scintS2EdepHisto->SetLineStyle(1);
  scintS2EdepHisto->SetLineColor(kRed);
  scintS2EdepHisto->SetLineWidth(2);
  scintS2EdepHisto->Draw();
  fileName = Form("Figure/ScintS2Edep Proton %3.1f MeV.eps",theERange);

  TPaveText* paveScintS2 = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets
coords
  paveScintS2->SetFillColor(0); // text is black on white
  paveScintS2->SetTextSize(0.05);
  paveScintS2->SetTextAlign(12);
  paveScintS2->AddText(paveText);
  paveScintS2->Draw("same");       //to draw your text object

  c2p2->SaveAs(fileName);


  TCanvas* c3p = new TCanvas("c3p","",200,10,700,500);
  gStyle->SetOptStat(0);
  caloEdepHisto->SetTitle("Calorimeter");
  caloEdepHisto->GetXaxis()->SetTitle("Deposiated Energy [MeV]");
  caloEdepHisto->GetXaxis()->CenterTitle(true);
  caloEdepHisto->GetXaxis()->SetTitleSize(0.05);
  caloEdepHisto->GetXaxis()->SetTitleOffset(0.90);
  caloEdepHisto->GetXaxis()->SetLabelSize(0.05);
  caloEdepHisto->GetYaxis()->SetLabelSize(0.05);
  caloEdepHisto->GetYaxis()->SetTitle("");
  caloEdepHisto->SetLineStyle(1);
  caloEdepHisto->SetLineColor(kRed);
  caloEdepHisto->SetLineWidth(2);

  caloEdepHisto->Draw();

  fileName = Form("Figure/CaloEdep Proton %3.1f MeV.eps",theERange);

  TPaveText* paveCalo = new TPaveText(0.15,0.7,0.35,0.8, "NDC"); // NDC sets coo
rds
  paveCalo->SetFillColor(0); // text is black on white
  paveCalo->SetTextSize(0.05);
  paveCalo->SetTextAlign(12);
  paveCalo->AddText(paveText);
  paveCalo->Draw("same");       //to draw your text object

  c3p->SaveAs(fileName);
  TCanvas* c4p = new TCanvas("c4p","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHisto->SetTitle("");
  totalEdepHisto->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHisto->GetXaxis()->CenterTitle(true);
  totalEdepHisto->GetXaxis()->SetTitleSize(0.05);
  totalEdepHisto->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHisto->GetXaxis()->SetLabelSize(0.05);
  totalEdepHisto->GetYaxis()->SetLabelSize(0.05);
  totalEdepHisto->GetYaxis()->SetTitle("");
  totalEdepHisto->SetLineStyle(1);
  totalEdepHisto->SetLineColor(kRed);
  totalEdepHisto->SetLineWidth(2);

  totalEdepHisto->Draw();

  meanText = Form("Mean = %3.1f ",totalEdepHisto->GetMean());
  rmsText  = Form("RMS = %3.1f ",totalEdepHisto->GetRMS());
  fileName = Form("Figure/TotalEdep Proton %3.1f MeV.eps",theERange);

  TPaveText* paveTotal = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC sets c
oords
  paveTotal->SetFillColor(0); // text is black on white
  paveTotal->SetTextSize(0.05);
  paveTotal->SetTextAlign(12);
  paveTotal->AddText(paveText);
  paveTotal->AddText(meanText);
  paveTotal->AddText(rmsText);
  paveTotal->Draw("same");       //to draw your text object

  c4p->SaveAs(fileName);

  TCanvas* c4p1 = new TCanvas("c4p1","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHistoNoS1->SetTitle("");
  totalEdepHistoNoS1->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHistoNoS1->GetXaxis()->CenterTitle(true);
  totalEdepHistoNoS1->GetXaxis()->SetTitleSize(0.05);
  totalEdepHistoNoS1->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHistoNoS1->GetXaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1->GetYaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1->GetYaxis()->SetTitle("");
  totalEdepHistoNoS1->SetLineStyle(1);
  totalEdepHistoNoS1->SetLineColor(kRed);
  totalEdepHistoNoS1->SetLineWidth(2);

  totalEdepHistoNoS1->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHistoNoS1->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHistoNoS1->GetRMS());
  fileName = Form("Figure/TotalEdepNoS1 Proton %3.1f MeV.eps",theERange);

  TPaveText* paveTotalnoS1 = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC se
ts coords
  paveTotalnoS1->SetFillColor(0); // text is black on white
  paveTotalnoS1->SetTextSize(0.05);
  paveTotalnoS1->SetTextAlign(12);
  paveTotalnoS1->AddText(paveText);
  paveTotalnoS1->AddText(meanText);
  paveTotalnoS1->AddText(rmsText);
  paveTotalnoS1->Draw("same");       //to draw your text object

  c4p1->SaveAs(fileName);

  TCanvas* c4p1c2 = new TCanvas("c4p1c2","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHistoNoS1Cut2->SetTitle("");
  totalEdepHistoNoS1Cut2->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHistoNoS1Cut2->GetXaxis()->CenterTitle(true);
```

```
  totalEdepHistoNoS1Cut2->GetXaxis()->SetTitleSize(0.05);
  totalEdepHistoNoS1Cut2->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHistoNoS1Cut2->GetXaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1Cut2->GetYaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1Cut2->GetYaxis()->SetTitle("");
  totalEdepHistoNoS1Cut2->SetLineStyle(1);
  totalEdepHistoNoS1Cut2->SetLineColor(kRed);
  totalEdepHistoNoS1Cut2->SetLineWidth(2);

  totalEdepHistoNoS1Cut2->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHistoNoS1Cut2->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHistoNoS1Cut2->GetRMS());
  fileName = Form("Figure/TotalEdepNoS1Cut2 Proton %3.1f MeV.eps",theERange);

  TPaveText* paveTotalnoS1Cut2 = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // ND
C sets coords
  paveTotalnoS1Cut2->SetFillColor(0); // text is black on white
  paveTotalnoS1Cut2->SetTextSize(0.05);
  paveTotalnoS1Cut2->SetTextAlign(12);
  paveTotalnoS1Cut2->AddText(paveText);
  paveTotalnoS1Cut2->AddText(meanText);
  paveTotalnoS1Cut2->AddText(rmsText);
  paveTotalnoS1Cut2->Draw("same");       //to draw your text object

  c4p1c2->SaveAs(fileName);

  TCanvas* c4pNC = new TCanvas("c4pNC","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHistoNoCut->SetTitle("");
  totalEdepHistoNoCut->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHistoNoCut->GetXaxis()->CenterTitle(true);
  totalEdepHistoNoCut->GetXaxis()->SetTitleSize(0.05);
  totalEdepHistoNoCut->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHistoNoCut->GetXaxis()->SetLabelSize(0.05);
  totalEdepHistoNoCut->GetYaxis()->SetLabelSize(0.05);
  totalEdepHistoNoCut->GetYaxis()->SetTitle("");
  totalEdepHistoNoCut->SetLineStyle(1);
  totalEdepHistoNoCut->SetLineColor(kRed);
  totalEdepHistoNoCut->SetLineWidth(2);

  totalEdepHistoNoCut->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHistoNoCut->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHistoNoCut->GetRMS());
  fileName = Form("Figure/TotalEdepNoCut Proton %3.1f MeV.eps",theERange);

  TPaveText* paveTotalNC = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC sets
 coords
  paveTotalNC->SetFillColor(0); // text is black on white
  paveTotalNC->SetTextSize(0.05);
  paveTotalNC->SetTextAlign(12);
  paveTotalNC->AddText(paveText);
  paveTotalNC->AddText(meanText);
  paveTotalNC->AddText(rmsText);
  paveTotalNC->Draw("same");       //to draw your text object

  c4pNC->SaveAs(fileName);

  TCanvas* c4p1NC = new TCanvas("c4p1NC","",200,10,700,500);
  gStyle->SetOptStat(0);
  totalEdepHistoNoS1NoCut->SetTitle("");
  totalEdepHistoNoS1NoCut->GetXaxis()->SetTitle("Deposited Energy [MeV]");
  totalEdepHistoNoS1NoCut->GetXaxis()->CenterTitle(true);
  totalEdepHistoNoS1NoCut->GetXaxis()->SetTitleSize(0.05);
  totalEdepHistoNoS1NoCut->GetXaxis()->SetTitleOffset(0.90);
  totalEdepHistoNoS1NoCut->GetXaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1NoCut->GetYaxis()->SetLabelSize(0.05);
  totalEdepHistoNoS1NoCut->GetYaxis()->SetTitle("");
  totalEdepHistoNoS1NoCut->SetLineStyle(1);
  totalEdepHistoNoS1NoCut->SetLineColor(kRed);
  totalEdepHistoNoS1NoCut->SetLineWidth(2);

  totalEdepHistoNoS1NoCut->Draw();

  meanText = Form("Mean = %3.1f",totalEdepHistoNoS1NoCut->GetMean());
  rmsText = Form("RMS = %3.1f",totalEdepHistoNoS1NoCut->GetRMS());
  fileName = Form("Figure/TotalEdepNoS1NoCut Proton %3.1f MeV.eps",theERange);

  TPaveText* paveTotalnoS1NC = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC
sets coords
  paveTotalnoS1NC->SetFillColor(0); // text is black on white
  paveTotalnoS1NC->SetTextSize(0.05);
  paveTotalnoS1NC->SetTextAlign(12);
  paveTotalnoS1NC->AddText(paveText);
  paveTotalnoS1NC->AddText(meanText);
  paveTotalnoS1NC->AddText(rmsText);
  paveTotalnoS1NC->Draw("same");       //to draw your text object

  c4p1NC->SaveAs(fileName);


  TCanvas* c5p = new TCanvas("c5p","",200,10,700,500);
  gStyle->SetOptStat(0);
  thetares->SetTitle("");
  thetares->GetXaxis()->SetTitle("#Delta #theta [deg]");
  thetares->GetXaxis()->CenterTitle(true);
  thetares->GetXaxis()->SetTitleSize(0.05);
  thetares->GetXaxis()->SetTitleOffset(0.90);
  thetares->GetXaxis()->SetLabelSize(0.05);
  thetares->GetYaxis()->SetLabelSize(0.05);
  thetares->GetYaxis()->SetTitle("");
  thetares->SetLineStyle(1);
  thetares->SetLineColor(kRed);
  thetares->SetLineWidth(2);

  thetares->Draw();

  fileName = Form("Figure/Angle Resolution Proton %3.1f MeV.eps",theERange);

  TPaveText* paveTRes = new TPaveText(0.15,0.7,0.35,0.8, "NDC"); // NDC sets coo
rds
  paveTRes->SetFillColor(0); // text is black on white
  paveTRes->SetTextSize(0.05);
  paveTRes->SetTextAlign(12);
  paveTRes->AddText(paveText);
  paveTRes->Draw("same");       //to draw your text object

  c5p->SaveAs(fileName);

  TCanvas* c6p = new TCanvas("c6p","",200,10,700,500);
  gStyle->SetOptStat(0);
  thetaresws->SetTitle("");
  thetaresws->GetXaxis()->SetTitle("#Delta #theta [deg]");
  thetaresws->GetXaxis()->CenterTitle(true);
  thetaresws->GetXaxis()->SetTitleSize(0.05);
  thetaresws->GetXaxis()->SetTitleOffset(0.90);
  thetaresws->GetXaxis()->SetLabelSize(0.05);
  thetaresws->GetYaxis()->SetLabelSize(0.05);
  thetaresws->GetYaxis()->SetTitle("");
  thetaresws->SetLineStyle(1);
  thetaresws->SetLineColor(kRed);
  thetaresws->SetLineWidth(2);

  thetaresws->Draw();

  meanText = Form("Mean = %3.1f ",thetaresws->GetMean());
  rmsText  = Form("RMS = %3.1f ",thetaresws->GetRMS());
```

```
  fileName = Form("Figure/Angle Resolution WS Proton %3.1f MeV.eps",theERange);

  TPaveText* paveTResWS = new TPaveText(0.15,0.55,0.35,0.8, "NDC"); // NDC sets
coords
  paveTResWS->SetFillColor(0); // text is black on white
  paveTResWS->SetTextSize(0.05);
  paveTResWS->SetTextAlign(12);
  paveTResWS->AddText(paveText);
  paveTResWS->AddText(meanText);
  paveTResWS->AddText(rmsText);
  paveTResWS->Draw("same");       //to draw your text object

  c6p->SaveAs(fileName);


  TCanvas* c7p = new TCanvas("c7p","",200,10,700,500);
  gStyle->SetOptStat(0);
  layerDepProton->SetMarkerColor(kRed);
  layerDepProton->SetMarkerStyle(22);
  layerDepProton->SetMarkerSize(2);
  layerDepProton->GetXaxis()->SetTitle("# Layer");
  layerDepProton->GetXaxis()->CenterTitle(true);
  layerDepProton->GetXaxis()->SetTitleSize(0.05);
  layerDepProton->GetXaxis()->SetTitleOffset(0.90);
  layerDepProton->GetXaxis()->SetLabelSize(0.05);
  layerDepProton->GetYaxis()->SetTitle("Energy [Mev]");
  layerDepProton->GetYaxis()->CenterTitle(true);
  layerDepProton->GetYaxis()->SetTitleSize(0.05);
  layerDepProton->GetYaxis()->SetTitleOffset(0.90);
  layerDepProton->GetYaxis()->SetLabelSize(0.05);
  layerDepProton->Draw();

  fileName = Form("Figure/LayerDep Proton %3.1f MeV.eps",theERange);

  c7p->SaveAs(fileName);


  TCanvas* c8p = new TCanvas("c8p","",200,10,700,500);
  gStyle->SetOptStat(0);
  energyVSangle->SetTitle("");
  energyVSangle->GetXaxis()->SetTitle("#theta [deg]");
  energyVSangle->GetXaxis()->SetTitleSize(0.05);
  energyVSangle->GetXaxis()->SetTitleOffset(0.90);
  energyVSangle->GetXaxis()->SetLabelSize(0.05);
  energyVSangle->GetYaxis()->SetLabelSize(0.05);
  energyVSangle->GetYaxis()->SetTitle("E_{Rec}/E_{Gen}");
  energyVSangle->SetLineStyle(1);
  energyVSangle->SetLineColor(kRed);
  energyVSangle->SetLineWidth(2);

  energyVSangle->Draw();

  fileName = Form("Figure/Energy Ratio VS Theta Proton %3.1f MeV.eps",theERange)
;

  TPaveText* paveTeVSa = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets co
ords
  paveTeVSa->SetFillColor(0); // text is black on white
  paveTeVSa->SetTextSize(0.05);
  paveTeVSa->SetTextAlign(12);
  paveTeVSa->AddText(paveText);
  paveTeVSa->Draw("same");       //to draw your text object

  c8p->SaveAs(fileName);


  TCanvas* c9p = new TCanvas("c9p","",200,10,700,500);
  gStyle->SetOptStat(0);
  bricksHitted->SetTitle("");
  bricksHitted->GetXaxis()->SetTitle("# LYSO Bricks");
  bricksHitted->GetXaxis()->SetTitleSize(0.05);
  bricksHitted->GetXaxis()->SetTitleOffset(0.90);
  bricksHitted->GetXaxis()->SetLabelSize(0.05);
  bricksHitted->SetLineStyle(1);
  bricksHitted->SetLineColor(kRed);
  bricksHitted->SetLineWidth(2);

  bricksHitted->Draw();

  fileName = Form("Figure/Number LYSO Bricks Proton %3.1f MeV.eps",theERange);

  TPaveText* paveTHB = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets coor
ds
  paveTHB->SetFillColor(0); // text is black on white
  paveTHB->SetTextSize(0.05);
  paveTHB->SetTextAlign(12);
  paveTHB->AddText(paveText);
  paveTHB->Draw("same");       //to draw your text object

  c9p->SaveAs(fileName);


  TCanvas* c10p = new TCanvas("c10p","",200,10,700,500);
  gStyle->SetOptStat(0);
  singlebrickDep->SetTitle("");
  singlebrickDep->GetXaxis()->SetTitle("Energy Deposited [MeV]");
  singlebrickDep->GetXaxis()->SetTitleSize(0.05);
  singlebrickDep->GetXaxis()->SetTitleOffset(0.90);
  singlebrickDep->GetXaxis()->SetLabelSize(0.05);
  singlebrickDep->SetLineStyle(1);
  singlebrickDep->SetLineColor(kRed);
  singlebrickDep->SetLineWidth(2);

  singlebrickDep->Draw();

  fileName = Form("Figure/Energy Deposited LYSO Brick Proton %3.1f MeV.eps",theE
Range);

  TPaveText* paveTEDB = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets coo
rds
  paveTEDB->SetFillColor(0); // text is black on white
  paveTEDB->SetTextSize(0.05);
  paveTEDB->SetTextAlign(12);
  paveTEDB->AddText(paveText);
  paveTEDB->Draw("same");       //to draw your text object

  c10p->SaveAs(fileName);

  TCanvas* c11p = new TCanvas("c11p","",200,10,700,500);
  gStyle->SetOptStat(0);
  allbricksDep->SetTitle("");
  allbricksDep->GetXaxis()->SetTitle("Energy Deposited [MeV]");
  allbricksDep->GetXaxis()->SetTitleSize(0.05);
  allbricksDep->GetXaxis()->SetTitleOffset(0.90);
  allbricksDep->GetXaxis()->SetLabelSize(0.05);
  allbricksDep->SetLineStyle(1);
  allbricksDep->SetLineColor(kRed);
  allbricksDep->SetLineWidth(2);

  allbricksDep->Draw();

  fileName = Form("Figure/Energy Deposited All LYSO Brick Electron %3.1f MeV.eps
",theERange);

  TPaveText* paveTEDBA = new TPaveText(0.65,0.7,0.85,0.8, "NDC"); // NDC sets co
ords
  paveTEDBA->SetFillColor(0); // text is black on white
  paveTEDBA->SetTextSize(0.05);
```

```
  paveTEDBA->SetTextAlign(12);
  paveTEDBA->AddText(paveText);
  paveTEDBA->Draw("same");       //to draw your text object

  c11p->SaveAs(fileName);

  histoFile->Write();
  histoFile->Close();

}

Double_t AnalyzerMulti::ComputeAngle(std::vector<TrackerRootHit>& myTrackerHit){

  std::vector<TrackerRootHit> layer2Hit;
  std::vector<TrackerRootHit> layer1Hit;

  for(size_t th=0;th<myTrackerHit.size();th++){
    int detId = myTrackerHit[th].GetDetectorId();
    if(detId>200)
      layer2Hit.push_back(myTrackerHit[th]);
    if(detId<200)
      layer1Hit.push_back(myTrackerHit[th]);
  }

  TVector3 posL2;
  TVector3 posL1;

  if(layer2Hit.size()==1){
    posL2 = (layer2Hit[0].GetEntryPoint()+layer2Hit[0].GetExitPoint())*0.5;
  }else{
    for(size_t j= 0;j<layer2Hit.size();j++){
      if(layer2Hit[j].GetTrackId()==1)
        posL2 = (layer2Hit[j].GetEntryPoint()+layer2Hit[j].GetExitPoint())*0.5;
    }
  }
  if(layer1Hit.size()==1){
    posL1 = (layer1Hit[0].GetEntryPoint()+layer1Hit[0].GetExitPoint())*0.5;
  }else{
    for(size_t j= 0;j<layer2Hit.size();j++){
      if(layer1Hit[j].GetTrackId()==1)
        posL1 = (layer1Hit[j].GetEntryPoint()+layer1Hit[j].GetExitPoint())*0.5;
    }
  }

  TVector3 segment = posL2-posL1;

  return segment.Theta();
}

Double_t AnalyzerMulti::ComputeAngleWithSmearing(std::vector<TrackerRootHit>& my
TrackerHit,Double_t delta){

  std::vector<TrackerRootHit> layer2Hit;
  std::vector<TrackerRootHit> layer1Hit;

  for(size_t th=0;th<myTrackerHit.size();th++){
    int detId = myTrackerHit[th].GetDetectorId();
    if(detId>200)
      layer2Hit.push_back(myTrackerHit[th]);
    if(detId<200)
      layer1Hit.push_back(myTrackerHit[th]);
  }

  TVector3 posL2;
  TVector3 posL1;

  if(layer2Hit.size()==1){
    posL2 = (layer2Hit[0].GetEntryPoint()+layer2Hit[0].GetExitPoint())*0.5;
  }else{
    for(size_t j= 0;j<layer2Hit.size();j++){
      if(layer2Hit[j].GetTrackId()==1)
        posL2 = (layer2Hit[j].GetEntryPoint()+layer2Hit[j].GetExitPoint())*0.5;
    }
  }
  if(layer1Hit.size()==1){
    posL1 = (layer1Hit[0].GetEntryPoint()+layer1Hit[0].GetExitPoint())*0.5;
  }else{
    for(size_t j= 0;j<layer2Hit.size();j++){
      if(layer1Hit[j].GetTrackId()==1)
        posL1 = (layer1Hit[j].GetEntryPoint()+layer1Hit[j].GetExitPoint())*0.5;
    }
  }

  TRandom myRand;

  Double_t xL1 = posL1.X()+myRand.Uniform(-delta,delta);
  Double_t yL1 = posL1.Y()+myRand.Uniform(-delta,delta);

  posL1.SetX(xL1);
  posL1.SetY(yL1);

  Double_t xL2 = posL2.X()+myRand.Uniform(-delta,delta);
  Double_t yL2 = posL2.Y()+myRand.Uniform(-delta,delta);

  posL2.SetX(xL2);
  posL2.SetY(yL2);

  TVector3 segment = posL2-posL1;

  return segment.Theta();
}
void AnalyzerMulti::SetAcceptanceWindows(Double_t X,Double_t Y){
  checkPos=true;
  Xlimit=X/2.;
  Ylimit=Y/2.;
}

void AnalyzerMulti::SetThetaAcceptance(Double_t aTheta){
  checkTheta=true;
  Thetalimit=aTheta;
}

void AnalyzerMulti::SetCalorimeterConfiguration(Int_t nCalo,Int_t nScint,Int_t n
Crystal){
  numbCaloLayer=nCalo;
  numbLayerScint=nScint;
  numbLayerCrystal=nCrystal;
}
```