```
//STANDARD ROOT INCLUDES
#include <TROOT.h>
#include <TH1.h>
#include <TH2.h>
#include <TH3.h>
#include <TProfile.h>
#include <TStyle.h>
#include <TCanvas.h>
#include <TTree.h>
#include <TFile.h>
#include <TChain.h>
#include <TChainElement.h>
#include <TDirectory.h>
#include <TSystemFile.h>
#include <TSystemDirectory.h>
#include <TMath.h>

//OUR OWN CLASSES TO READ THE TREE
#include "UATree/UADataFormat/interface/MyEvtId.h"
#include "UATree/UADataFormat/interface/MyHLTrig.h"
#include "UATree/UADataFormat/interface/MyCaloTower.h"
#include "UATree/UADataFormat/interface/MyTracks.h"
#include "UATree/UADataFormat/interface/MyVertex.h"
#include "UATree/UADataFormat/interface/MySiPixelCluster.h"
#include "UATree/UADataFormat/interface/MySiStripCluster.h"
#include "UATree/UADataFormat/interface/MyKshorts.h"
#include "UATree/UADataFormat/interface/MyLambdas.h"
#include "UATree/UADataFormat/interface/MyPart.h"

// TOTEM data formats
#include "RPRootDumpReconstructedProton.h"
#include "RPRootDumpReconstructedProtonPair.h"
#include "RPRootDumpTrackInfo.h"
#include "RPRootDumpDigiInfo.h"
#include "RPRootDumpPatternInfo.h"
#include "TriggerData.h"

#include "analysis_tools_Mirko2015.h"
//#include "rp_aperture_config.h"

//STANDARD C++ INCLUDES
#include <iostream>
#include <string>
#include <sstream>
#include <vector>
#include <map>
#include <cmath>
#include <algorithm>
#include <fstream>
#include <cstdlib>


using namespace std;

void anaRP(vector<string> const& fileNames, string const& outputFileName, string const& outputFil
eTOTEM, const Int_t nevt_max);

int main(int argc, char** argv) {

  if (argc < 5){
     cout<<"anaRP fileListName=filenames.txt outputFileName=output.root outputFileTOTEM=totemlist
.txt nevt_max=-100"<<endl;
     exit(-1);
  }

   string fileListName = argv[1];
   string outputFileName = argv[2];
   string outputFileTOTEM = argv[3];
   int nevt_max = -1;
   stringstream maxEvents_ss;
   maxEvents_ss << argv[4];
   maxEvents_ss >> nevt_max;
   ifstream infile( fileListName.c_str() );

   vector<string> fileNames;
   string file;
   while( infile >> file ) {
      cout << "Adding " << file << endl;
      fileNames.push_back( file );
   }
   infile.close();

   anaRP(fileNames,outputFileName,outputFileTOTEM,nevt_max);
   return 0;
}

/*
void anaRP(string const& fileListName="filenames.txt", string const& outputFileName = "output.roo
t", const Int_t nevt_max = -100){

   string file;

   ifstream infile( fileListName.c_str() );

   vector<string> fileNames;
   while( infile >> file ) {
      cout << "Adding " << file << endl;
      fileNames.push_back( file );
   }
   infile.close();

   anaRP(fileNames,outputFileName,nevt_max);
}
*/

void anaRP(vector<string> const& fileNames, string const& outputFileName = "output.root", string
const& outputFileTOTEM = "totemlist.txt", const Int_t nevt_max = -100){

  bool isMC  = false;
  string treeName = (!isMC) ? "cms_totem" : "evt";

  double wei = 1.;

  //==============================
  const Int_t nevt_max_corr = (nevt_max >= 0) ? nevt_max : 99999999;
  cout<<"nevt_max_corr = "<<nevt_max_corr <<endl;

//  ofstream fout(outputFileTOTEM.c_str());


  // Declaration of histograms
  map<string,TH1F*> histosTH1F;

  vector<string> selections;
  selections.push_back("TOTEM0");
  selections.push_back("2valid");
  selections.push_back("anyTB/BT/TT/BB");
  selections.push_back("exclusiveTB/BT/TT/BB");
  selections.push_back("fiducialXY");
  selections.push_back("notElastic");
  selections.push_back("#xi<0.1");
  //...important
  //selections.push_back("|#xi_{1,2}|<0.02");
  //selections.push_back("NvtxCMS=1");
  //selections.push_back("NtrkCMS=2");
  //selections.push_back("CT py bal.");
  //selections.push_back("CT px bal.");
  //selections.push_back("RP xVtx");
  //selections.push_back("CT xVtx");

  int nBinsEventSelection = selections.size();
  histosTH1F["EventSelection"] = new TH1F("EventSelection"," ",nBinsEventSelection,0,nBinsEventSe
lection);
```

```
  for(size_t k = 0; k < selections.size(); ++k)
    histosTH1F["EventSelection"]->GetXaxis()->SetBinLabel( (k + 1), selections[k].c_str() );

  histosTH1F["hnconf"] = new TH1F("hnconf", "Number of configurations (TB or BT or TT or BB)" , 5
, 0., 5.);

  histosTH1F["rp_x_020"] = new TH1F("rp_x_020", "x RP" , 200, -10., 10.);
  histosTH1F["rp_x_021"] = new TH1F("rp_x_021", "x RP" , 200, -10., 10.);
  histosTH1F["rp_x_024"] = new TH1F("rp_x_024", "x RP" , 200, -10., 10.);
  histosTH1F["rp_x_025"] = new TH1F("rp_x_025", "x RP" , 200, -10., 10.);

  histosTH1F["rp_x_120"] = new TH1F("rp_x_120", "x RP" , 200, -10., 10.);
  histosTH1F["rp_x_121"] = new TH1F("rp_x_121", "x RP" , 200, -10., 10.);
  histosTH1F["rp_x_124"] = new TH1F("rp_x_124", "x RP" , 200, -10., 10.);
  histosTH1F["rp_x_125"] = new TH1F("rp_x_125", "x RP" , 200, -10., 10.);

  histosTH1F["rp_y_020"] = new TH1F("rp_y_020", "y RP" , 500, -50., 50.);
  histosTH1F["rp_y_021"] = new TH1F("rp_y_021", "y RP" , 500, -50., 50.);
  histosTH1F["rp_y_024"] = new TH1F("rp_y_024", "y RP" , 500, -50., 50.);
  histosTH1F["rp_y_025"] = new TH1F("rp_y_025", "y RP" , 500, -50., 50.);

  histosTH1F["rp_y_120"] = new TH1F("rp_y_120", "y RP" , 500, -50., 50.);
  histosTH1F["rp_y_121"] = new TH1F("rp_y_121", "y RP" , 500, -50., 50.);
  histosTH1F["rp_y_124"] = new TH1F("rp_y_124", "y RP" , 500, -50., 50.);
  histosTH1F["rp_y_125"] = new TH1F("rp_y_125", "y RP" , 500, -50., 50.);

  //--- from TT/BB, above from TB/BT
  histosTH1F["rp2_x_020"] = new TH1F("rp2_x_020", "x RP" , 200, -10., 10.);
  histosTH1F["rp2_x_021"] = new TH1F("rp2_x_021", "x RP" , 200, -10., 10.);
  histosTH1F["rp2_x_024"] = new TH1F("rp2_x_024", "x RP" , 200, -10., 10.);
  histosTH1F["rp2_x_025"] = new TH1F("rp2_x_025", "x RP" , 200, -10., 10.);

  histosTH1F["rp2_x_120"] = new TH1F("rp2_x_120", "x RP" , 200, -10., 10.);
  histosTH1F["rp2_x_121"] = new TH1F("rp2_x_121", "x RP" , 200, -10., 10.);
  histosTH1F["rp2_x_124"] = new TH1F("rp2_x_124", "x RP" , 200, -10., 10.);
  histosTH1F["rp2_x_125"] = new TH1F("rp2_x_125", "x RP" , 200, -10., 10.);

  histosTH1F["rp2_y_020"] = new TH1F("rp2_y_020", "y RP" , 500, -50., 50.);
  histosTH1F["rp2_y_021"] = new TH1F("rp2_y_021", "y RP" , 500, -50., 50.);
  histosTH1F["rp2_y_024"] = new TH1F("rp2_y_024", "y RP" , 500, -50., 50.);
  histosTH1F["rp2_y_025"] = new TH1F("rp2_y_025", "y RP" , 500, -50., 50.);

  histosTH1F["rp2_y_120"] = new TH1F("rp2_y_120", "y RP" , 500, -50., 50.);
  histosTH1F["rp2_y_121"] = new TH1F("rp2_y_121", "y RP" , 500, -50., 50.);
  histosTH1F["rp2_y_124"] = new TH1F("rp2_y_124", "y RP" , 500, -50., 50.);
  histosTH1F["rp2_y_125"] = new TH1F("rp2_y_125", "y RP" , 500, -50., 50.);

  histosTH1F["thyEla"] = new TH1F("thyEla", "thyL+thyR" , 4000 , -0.0004 , 0.0004);
  histosTH1F["thxEla"] = new TH1F("thxEla", "thxL+thxR" , 4000 , -0.0004 , 0.0004);

  histosTH1F["thyEla_diag"] = new TH1F("thyEla_diag", "thyL+thyR diagonals" , 4000 , -0.0004 , 0.
0004);
  histosTH1F["thxEla_diag"] = new TH1F("thxEla_diag", "thxL+thxR diagonals" , 4000 , -0.0004 , 0.
0004);

  histosTH1F["thyEla_ttbb"] = new TH1F("thyEla_ttbb", "thyL+thyR TT/BB" , 4000 , -0.0004 , 0.0004
);
  histosTH1F["thxEla_ttbb"] = new TH1F("thxEla_ttbb", "thxL+thxR TT/BB" , 4000 , -0.0004 , 0.0004
);

  //histosTH1F["proton_right_xi"] = new TH1F("proton_right_xi", "#xi" , 200 , -1. , 1.);
  //...Luiz
  histosTH1F["proton_right_xi"] = new TH1F("proton_right_xi", "#xi" , 200 , -0.5 , 0.5);
  histosTH1F["proton_right_logXi"] = new TH1F("proton_right_logXi","log(#xi)",200,-5.,0.);

  //histosTH1F["proton_left_xi"] = new TH1F("proton_left_xi", "#xi" , 200 , -1. , 1.);
  //...Luiz
  histosTH1F["proton_left_xi"] = new TH1F("proton_left_xi", "#xi" , 200 , -0.5 , 0.5);
  histosTH1F["proton_left_logXi"] = new TH1F("proton_left_logXi","log(#xi)",200,-5.,0.);

  histosTH1F["proton_right_t"] = new TH1F("proton_right_t", "-t" , 1000 , 0. , 5.);
  histosTH1F["proton_left_t"] = new TH1F("proton_left_t", "-t" , 1000 , 0. , 5.);
  histosTH1F["proton_right_t_diag"] = new TH1F("proton_right_t_diag", "-t diagonal" , 1000 , 0. ,
 5.);
  histosTH1F["proton_left_t_diag"] = new TH1F("proton_left_t_diag", "-t diagonal" , 1000 , 0. , 5
.);

  histosTH1F["proton_right_t_ttbb"] = new TH1F("proton_right_t_ttbb", "-t" , 1000 , 0. , 5.);
  histosTH1F["proton_left_t_ttbb"] = new TH1F("proton_left_t_ttbb", "-t TT/BB" , 1000 , 0. , 5.);

  histosTH1F["eHF"] = new TH1F("eHF", "energy HF tower (GeV)" , 500 , 0. , 100.);
  histosTH1F["nHF"] = new TH1F("nHF", "n HF tower (eHF>5 GeV)" , 200 , 0. , 200.);

  histosTH1F["totem_py"] = new TH1F("totem_py", "p_{Y} TOTEM" , 500 , -5. , 5.);
  histosTH1F["totem_px"] = new TH1F("totem_px", "p_{X} TOTEM" , 500 , -5. , 5.);

  //...Luiz
  histosTH1F["totem_pyy"] = new TH1F("totem_pyy", "p_{Y} TOTEM" , 1000 , -1. , 1.);
  histosTH1F["totem_pxx"] = new TH1F("totem_pxx", "p_{X} TOTEM" , 1000 , -1. , 1.);

  histosTH1F["proton_dx0"]  = new TH1F("proton_dx0","xVtx_{56}-xVtx_{45}",300,-0.0003,0.0003);

  histosTH1F["hLS"] = new TH1F("hLS", "LS" , 800 , 0. , 800.);

  histosTH1F["htopo"] = new TH1F("htopo","1=TB 2=BT 3=TT 4=BB topology",5,0,5);

  //histosTH1F["hthyEla2_diag"] = new TH1F("hthyEla2_diag", "thyL+thyR dig" , 2000 , -0.0004 , 0.
0004);
  //histosTH1F["hthxEla2_diag"] = new TH1F("hthxEla2_diag", "thxL+thxR dig" , 2000 , -0.0004 , 0.
0004);
  //histosTH1F["hthyEla2_ttbb"] = new TH1F("hthyEla2_ttbb", "thyL+thyR TTBB" , 2000 , -0.0004 , 0
.0004);
  //histosTH1F["hthxEla2_ttbb"] = new TH1F("hthxEla2_ttbb", "thxL+thxR TTBB" , 2000 , -0.0004 , 0
.0004);
  //...Luiz
  histosTH1F["hthyEla2_diag"] = new TH1F("hthyEla2_diag", "thyL+thyR dig" , 4000 , -0.0004 , 0.00
04);
  histosTH1F["hthxEla2_diag"] = new TH1F("hthxEla2_diag", "thxL+thxR dig" , 4000 , -0.0004 , 0.00
04);
  histosTH1F["hthyEla2_ttbb"] = new TH1F("hthyEla2_ttbb", "thyL+thyR TTBB" , 4000 , -0.0004 , 0.0
004);
  histosTH1F["hthxEla2_ttbb"] = new TH1F("hthxEla2_ttbb", "thxL+thxR TTBB" , 4000 , -0.0004 , 0.0
004);

  //...2D
  map<string,TH2F*> histosTH2F;

  histosTH2F["rp_yx_020"] = new TH2F("rp_yx_020", "y vs x RP" , 200, -10., 10., 500, -50., 50.);
  histosTH2F["rp_yx_021"] = new TH2F("rp_yx_021", "y vs x RP" , 200, -10., 10., 500, -50., 50.);
  histosTH2F["rp_yx_024"] = new TH2F("rp_yx_024", "y vs x RP" , 200, -10., 10., 500, -50., 50.);
  histosTH2F["rp_yx_025"] = new TH2F("rp_yx_025", "y vs x RP" , 200, -10., 10., 500, -50., 50.);

  histosTH2F["rp_yx_120"] = new TH2F("rp_yx_120", "y vs x RP" , 200, -10., 10., 500, -50., 50.);
  histosTH2F["rp_yx_121"] = new TH2F("rp_yx_121", "y vs x RP" , 200, -10., 10., 500, -50., 50.);
  histosTH2F["rp_yx_124"] = new TH2F("rp_yx_124", "y vs x RP" , 200, -10., 10., 500, -50., 50.);
  histosTH2F["rp_yx_125"] = new TH2F("rp_yx_125", "y vs x RP" , 200, -10., 10., 500, -50., 50.);

  //--- from TT/BB, above from TB/BT
  histosTH2F["rp2_yx_020"] = new TH2F("rp2_yx_020", "y vs x RP" , 200, -10., 10., 500, -50., 50.)
;
  histosTH2F["rp2_yx_021"] = new TH2F("rp2_yx_021", "y vs x RP" , 200, -10., 10., 500, -50., 50.)
;
  histosTH2F["rp2_yx_024"] = new TH2F("rp2_yx_024", "y vs x RP" , 200, -10., 10., 500, -50., 50.)
;
  histosTH2F["rp2_yx_025"] = new TH2F("rp2_yx_025", "y vs x RP" , 200, -10., 10., 500, -50., 50.)
;

  histosTH2F["rp2_yx_120"] = new TH2F("rp2_yx_120", "y vs x RP" , 200, -10., 10., 500, -50., 50.)
;
  histosTH2F["rp2_yx_121"] = new TH2F("rp2_yx_121", "y vs x RP" , 200, -10., 10., 500, -50., 50.)
;
  histosTH2F["rp2_yx_124"] = new TH2F("rp2_yx_124", "y vs x RP" , 200, -10., 10., 500, -50., 50.)
;
```

```
  histosTH2F["rp2_yx_125"] = new TH2F("rp2_yx_125", "y vs x RP" , 200, -10., 10., 500, -50., 50.)
;

  histosTH2F["proton_x0_RvsL"]  = new TH2F("proton_x0_RvsL","xVtx_{56} vs xVtx_{45}",3000,-0.005,
0.001,3000,-0.005,0.001);

  //...Luiz
  histosTH2F["phi_proton_right_t"] = new TH2F("phi_proton_right_t","#varphi vs |-t|" , 1000 , 0.
, 5., 64, -3.2, 3.2);
  histosTH2F["phi_proton_left_t"]  = new TH2F("phi_proton_left_t", "#varphi vs |-t|" , 1000 , 0.
, 5., 64, -3.2, 3.2);
  //...Luiz
  histosTH2F["phi_proton_right_t_diag"] = new TH2F("phi_proton_right_t_diag","#varphi vs |-t|" ,
1000 , 0. , 5., 64, -3.2, 3.2);
  histosTH2F["phi_proton_left_t_diag"]  = new TH2F("phi_proton_left_t_diag", "#varphi vs |-t|" ,
1000 , 0. , 5., 64, -3.2, 3.2);
  //...Luiz
  histosTH2F["phi_proton_right_t_ttbb"] = new TH2F("phi_proton_right_t_ttbb","#varphi vs |-t|" ,
1000 , 0. , 5., 64, -3.2, 3.2);
  histosTH2F["phi_proton_left_t_ttbb"]  = new TH2F("phi_proton_left_t_ttbb", "#varphi vs |-t|" ,
1000 , 0. , 5., 64, -3.2, 3.2);
  //...Luiz
  histosTH2F["phi_proton_right_t_tt"] = new TH2F("phi_proton_right_t_tt","#varphi vs |-t|" , 1000
 , 0. , 5., 64, -3.2, 3.2);
  histosTH2F["phi_proton_left_t_tt"]  = new TH2F("phi_proton_left_t_tt", "#varphi vs |-t|" , 1000
 , 0. , 5., 64, -3.2, 3.2);
  //...Luiz
  histosTH2F["phi_proton_right_t_bb"] = new TH2F("phi_proton_right_t_bb","#varphi vs |-t|" , 1000
 , 0. , 5., 64, -3.2, 3.2);
  histosTH2F["phi_proton_left_t_bb"]  = new TH2F("phi_proton_left_t_bb", "#varphi vs |-t|" , 1000
 , 0. , 5., 64, -3.2, 3.2);
  //
  //...Luiz
  // delta phi between protons
  histosTH1F["dphi_proton"]  = new TH1F("dphi_proton", "#Delta#varphi" , 64, -3.2, 3.2);
  histosTH1F["dphi_proton_diag"]  = new TH1F("dphi_proton_diag", "#Delta#varphi DIAG" , 64, -3.2,
 3.2);
  histosTH1F["dphi_proton_ttbb"]  = new TH1F("dphi_proton_ttbb", "#Delta#varphi TTBB" , 64, -3.2,
 3.2);
  //
  histosTH2F["dphi_proton_mrec"]  = new TH2F("dphi_proton_mrec", "#Delta#varphi_{pp} vs M_{4#pi}"
 , 400 , 0., 8.0, 64, -3.2, 3.2);
  histosTH2F["dphi_proton_mrec_diag"]  = new TH2F("dphi_proton_mrec_diag", "#Delta#varphi_{pp} vs
 M_{4#pi} DIAG" , 400 , 0., 8.0, 64, -3.2, 3.2);
  histosTH2F["dphi_proton_mrec_ttbb"]  = new TH2F("dphi_proton_mrec_ttbb", "#Delta#varphi_{pp} vs
 M_{4#pi} TTBB" , 400 , 0., 8.0, 64, -3.2, 3.2);

//---------------------------------------------------

  int nbins_eta = 80;
  int nbins_pt = 100;
  int nbins_phi = 64;

  histosTH1F["hlooper"] = new TH1F("hlooper","isLooper",5,0,5);

  histosTH1F["hpt"] = new TH1F("hpt","p_{T}",nbins_pt,0,5);
  histosTH1F["heta"] = new TH1F("heta","#eta",nbins_eta,-4,4);
  histosTH1F["hphi"] = new TH1F("hphi","#varphi",nbins_phi,-3.2,3.2);

  //...Luiz
  histosTH1F["hphiL"] = new TH1F("hphiL","#varphi_{L}",60,-TMath::Pi(),TMath::Pi());
  histosTH1F["hphiR"] = new TH1F("hphiR","#varphi_{R}",60,-TMath::Pi(),TMath::Pi());
  histosTH1F["hdphi"] = new TH1F("hdphi","#Delta#varphi_{LR}",320,0,TMath::Pi());
  histosTH1F["hdphi_diag"] = new TH1F("hdphi_diag","#Delta#varphi_{LR} TB/BT",320,0,TMath::Pi());
  histosTH1F["hdphi_ttbb"] = new TH1F("hdphi_ttbb","#Delta#varphi_{LR} TT/BB",320,0,TMath::Pi());
  //

  //histosTH1F["hptP"] = new TH1F("hptP","p_{T} #pi+",nbins_pt,0,3);
  //...Luiz
  histosTH1F["hptP"] = new TH1F("hptP","p_{T} #pi+",2.0*nbins_pt,0,4);
  histosTH1F["hetaP"] = new TH1F("hetaP","#eta #pi+",nbins_eta,-4,4);
  histosTH1F["hphiP"] = new TH1F("hphiP","#varphi #pi+",nbins_phi,-3.2,3.2);
  //histosTH1F["hptM"] = new TH1F("hptM","p_{T} #pi-",nbins_pt,0,3);
  //...Luiz
  histosTH1F["hptM"] = new TH1F("hptM","p_{T} #pi-",2.0*nbins_pt,0,4);
  histosTH1F["hetaM"] = new TH1F("hetaM","#eta #pi-",nbins_eta,-4,4);
  histosTH1F["hphiM"] = new TH1F("hphiM","#varphi #pi-",nbins_phi,-3.2,3.2);

  //histosTH1F["hptRes"] = new TH1F("hptRes","p_{T} 4#pi",nbins_pt,0,3);
  //...Luiz
  histosTH1F["hptRes"] = new TH1F("hptRes","p_{T} 4#pi",2.0*nbins_pt,0,4);
  histosTH1F["hetaRes"] = new TH1F("hetaRes","#eta 4#pi",nbins_eta*1.5,-6,6);
  histosTH1F["hphiRes"] = new TH1F("hphiRes","#varphi 4#pi",nbins_phi,-3.2,3.2);

  //  histosTH1F["htopo"] = new TH1F("htopo","1=TB 2=BT 3=TT 4=BB topology",5,0,5);

  //histosTH1F["hthyEla_diag"] = new TH1F("hthyEla_diag", "thyL+thyR dig" , 2000 , -0.0004 , 0.00
04);
  //histosTH1F["hthxEla_diag"] = new TH1F("hthxEla_diag", "thxL+thxR dig" , 2000 , -0.0004 , 0.00
04);
  //histosTH1F["hthyEla_ttbb"] = new TH1F("hthyEla_ttbb", "thyL+thyR TTBB" , 2000 , -0.0004 , 0.0
004);
  //histosTH1F["hthxEla_ttbb"] = new TH1F("hthxEla_ttbb", "thxL+thxR TTBB" , 2000 , -0.0004 , 0.0
004);
  //...Luiz
  histosTH1F["hthyEla_diag"] = new TH1F("hthyEla_diag", "thyL+thyR dig" , 4000 , -0.0004 , 0.0004
);
  histosTH1F["hthxEla_diag"] = new TH1F("hthxEla_diag", "thxL+thxR dig" , 4000 , -0.0004 , 0.0004
);
  histosTH1F["hthyEla_ttbb"] = new TH1F("hthyEla_ttbb", "thyL+thyR TTBB" , 4000 , -0.0004 , 0.000
4);
  histosTH1F["hthxEla_ttbb"] = new TH1F("hthxEla_ttbb", "thxL+thxR TTBB" , 4000 , -0.0004 , 0.000
4);

  histosTH1F["hntrk0"] = new TH1F("hntrk0","Ntrk",150,0,150);
  histosTH1F["hntrk"] = new TH1F("hntrk","Ntrk for nPixelHits>0",150,0,150);
  histosTH1F["hntrkvtx"] = new TH1F("hntrkvtx","Ntrkvtx",150,0,150);
  histosTH1F["hntrkvtx0"] = new TH1F("hntrkvtx0","Ntrkvtx0",150,0,150);
  histosTH1F["hntrkvtx2"] = new TH1F("hntrkvtx2","Ntrkvtx2",150,0,150);
  histosTH1F["hntrkvtx3"] = new TH1F("hntrkvtx3","Ntrkvtx3",150,0,150);
  histosTH1F["hntrkvtx4"] = new TH1F("hntrkvtx4","Ntrkvtx4",150,0,150);
  histosTH1F["hntrkntrkvtx2"] = new TH1F("hntrkntrkvtx2","Ntrk for Ntrkvtx=2",150,0,150);
  histosTH1F["hntrk2ntrkvtx"] = new TH1F("hntrk2ntrkvtx","Ntrkvtx for Ntrk=2",150,0,150);

  histosTH2F["hntrkntrkvtx"] = new TH2F("hntrkntrkvtx","Ntrk vs Ntrkvtx",150,0,150,150,0,150);

  histosTH1F["hvtx"] = new TH1F("hvtx","vtx.isFake()",2,0,2);
  //...Luiz
  histosTH1F["hvtx2"] = new TH1F("hvtx2","vtx.isFake() 4 tracks",2,0,2);
  histosTH1F["hvtx3"] = new TH1F("hvtx3","vtx.isFake() 4 tracks both |#eta|<2.5 and OS",2,0,2);

  histosTH1F["hnvtx"] = new TH1F("hnvtx","Nvtx",10,0,10);


  //...Kshorts
  histosTH1F["hnks"] = new TH1F("hnks","N Kshorts",10,0,10);
  histosTH2F["hntrknks"] = new TH2F("hntrknks","# of Vees vs # of Tracks",150,0,150,10,0,10);
  histosTH2F["hnvtxnks"] = new TH2F("hnvtxnks","# of Vees vs # of Vertices",150,0,150,10,0,10);
  histosTH2F["hntrknvtx"] = new TH2F("hntrknvtx","# of Vertices vs # of Tracks",150,0,150,150,0,1
50);
  histosTH1F["hksvertexx"] = new TH1F("hksvertexx","K0s X vertex",120,-30.,30.);
  histosTH1F["hksvertexy"] = new TH1F("hksvertexy","K0s Y vertex",120,-30.,30.);
  histosTH1F["hksvertexz"] = new TH1F("hksvertexz","K0s Z vertex",120,-30.,30.);
  histosTH1F["hksradius"] = new TH1F("hksradius","K0s vertex radius",60,0.,30.);
  histosTH1F["hkslifetime"] = new TH1F("hkslifetime","K0s lifetime",20,0.,10.);
  //...2D
  histosTH2F["h2dimksxy"] = new TH2F("h2dimksxy","K0s X vs Y vtx",300,-30.,30.,300,-30.,30.);
  histosTH2F["h2dimksxz"] = new TH2F("h2dimksxz","K0s X vs Z vtx",300,-30.,30.,300,-30.,30.);
  histosTH2F["h2dimksyz"] = new TH2F("h2dimksyz","K0s Y vs Z vtx",300,-30.,30.,300,-30.,30.);
  //
  histosTH1F["hkspt"] = new TH1F("hkspt","K0s pt",100,0.,5.);
  histosTH1F["hkseta"] = new TH1F("hkseta","K0s #eta",80,-4.,4.);
  histosTH1F["hksphi"] = new TH1F("hksphi","K0s #varphi",64,-3.2,3.2);
  histosTH1F["hksmass"] = new TH1F("hksmass","K0s mass",250,0.,5.);
  //
  histosTH1F["hksmassv1"] = new TH1F("hksmassv1","K0s mass 1 vertex",250,0.,5.);
  histosTH1F["hksmassv2"] = new TH1F("hksmassv2","K0sK0s mass 2 vertices",250,0.,5.);
  histosTH1F["hksmassv3"] = new TH1F("hksmassv3","K0s mass 3 vertices",250,0.,5.);


  //...Lambdas
  histosTH1F["hnlam"] = new TH1F("hnlam","N #Lambda’s",10,0,10);
  histosTH1F["hlamvertexx"] = new TH1F("hlamvertexx","#Lambda X vertex",120,-30.,30.);
  histosTH1F["hlamvertexy"] = new TH1F("hlamvertexy","#Lambda Y vertex",120,-30.,30.);
  histosTH1F["hlamvertexz"] = new TH1F("hlamvertexz","#Lambda Z vertex",120,-30.,30.);
  histosTH1F["hlamradius"] = new TH1F("hlamradius","#Lambda vertex radius",60,0.,30.);
  //...2D
  histosTH2F["h2dimlamxy"] = new TH2F("h2dimlamxy","#Lambda X vs Y vtx",300,-30.,30.,300,-30.,30.
);
  histosTH2F["h2dimlamxz"] = new TH2F("h2dimlamxz","#Lambda X vs Z vtx",300,-30.,30.,300,-30.,30.
);
  histosTH2F["h2dimlamyz"] = new TH2F("h2dimlamyz","#Lambda Y vs Z vtx",300,-30.,30.,300,-30.,30.
);
  //
  histosTH1F["hlampt"] = new TH1F("hlampt","#Lambda pt",100,0.,5.);
  histosTH1F["hlameta"] = new TH1F("hlameta","#Lambda #eta",80,-4.,4.);
  histosTH1F["hlamphi"] = new TH1F("hlamphi","#Lambda #varphi",64,-3.2,3.2);
  histosTH1F["hlammass"] = new TH1F("hlammass","#Lambda mass",250,0.,5.);


  histosTH1F["hvtxx"] = new TH1F("hvtxx","X vtx",1000,-1.,1.);
  histosTH1F["hvtxy"] = new TH1F("hvtxy","Y vtx",1000,-1.,1.);
  histosTH1F["hvtxx4"] = new TH1F("hvtxx4","X vtx",1000,-1.,1.);
  histosTH1F["hvtxy4"] = new TH1F("hvtxy4","Y vtx",1000,-1.,1.);
  ////histosTH1F["hvtxx"] = new TH1F("hvtxx","X vtx",10000,-5000.,5000.);
  ////histosTH1F["hvtxy"] = new TH1F("hvtxy","Y vtx",10000,-5000.,5000.);
  histosTH1F["hvtxz"] = new TH1F("hvtxz","Z vtx",300,-30.,30.);
  histosTH1F["hvtxz4"] = new TH1F("hvtxz4","Z vtx",300,-30.,30.);

  //...Luiz
  histosTH2F["hvtx2dimxy"] = new TH2F("hvtx2dimxy","X vs Y vtx",1000,-1.,1.,1000,-1.,1.);
  //histosTH2F["hvtx2dimxy"] = new TH2F("hvtx2dimxy","X vs Y vtx",1000,-10.,10.,1000,-10.,10.);
  histosTH2F["hvtx2dimxz"] = new TH2F("hvtx2dimxz","X vs Z vtx",1000,-1.,1.,300,-3.,3.);
  histosTH2F["hvtx2dimyz"] = new TH2F("hvtx2dimyz","Y vs Z vtx",1000,-1.,1.,300,-3.,3.);
  //...3D
  ////map<string,TH3F*> histosTH3F;
  //...3D
  //histosTH3F["hvtx3dimxyz"] = new TH3F("hvtx3dimxyz","XYZ vtx",1000,-1.,1.,1000,-1.,1.,300,-30.
,30.);
  ////histosTH3F["hvtx3dimxyz"] = new TH3F("hvtx3dimxyz","XYZ vtx",100,-1.,1.,100,-1.,1.,300,-15.
,15.);
  //ntrk==4
  histosTH2F["hvtx2dimxy4"] = new TH2F("hvtx2dimxy4","X vs Y vtx",1000,-5.,5.,1000,-5.,5.);
  histosTH2F["hvtx2dimxz4"] = new TH2F("hvtx2dimxz4","X vs Z vtx",1000,-5.,5.,300,-30.,30.);
  histosTH2F["hvtx2dimyz4"] = new TH2F("hvtx2dimyz4","Y vs Z vtx",1000,-5.,5.,300,-30.,30.);
  ////histosTH2F["hvtx2dimxy"] = new TH2F("hvtx2dimxy","X vs Y vtx",10000,-5000.,5000.,10000,-500
0.,5000.);
  //...3D
  ////histosTH3F["hvtx3dimxyz4"] = new TH3F("hvtx3dimxyz4","XYZ vtx",100,-1.,1.,100,-1.,1.,300,-1
5.,15.);

  //...secondaryVertex
  ////histosTH1F["vertex_multiplicity"] = new TH1F("vertex_multiplicity","n vertices",30,0,30);
  //
  histosTH1F["sec_vtx_xpos"] = new TH1F("sec_vtx_xpos","X secondary vtx",150,-10.,10.);
  histosTH1F["sec_vtx_ypos"] = new TH1F("sec_vtx_ypos","Y secondary vtx",150,-10.,10.);
  histosTH1F["sec_vtx_zpos"] = new TH1F("sec_vtx_zpos","Z secondary vtx",150,-30.,30.);
  //
  ///// histosTH1F["sec_vtx_ndof"] = new TH1F("","Ndof secondary vtx",100,0.,15.);
  /////histosTH1F["sec_vtx_chi2"] = new TH1F("","chi2 secondary vtx",100,0.,10.);
  /////histosTH1F["sec_vtx_chi2n"] = new TH1F("","chi2n secondary vtx",100,0.,10.);
  /////histosTH1F["sec_vtx_ntracks"] = new TH1F("","Ntracks secondary vtx",30,0,30);
  /////histosTH1F["sec_vtx_sumpt"] = new TH1F("","SumPt secondary vtx",100,0.,100.);

  //...Kshort
```

```
histosTH1F["hxk"] = new TH1F("hxk","X vtx kshorts",1000,-10.,10.);
histosTH1F["hyk"] = new TH1F("hyk","Y vtx kshorts",1000,-10.,10.);
histosTH1F["hzk"] = new TH1F("hzk","Z vtx kshorts",300,-30.,30.);
histosTH2F["h2dimxyk"] = new TH2F("h2dimxyk","X vs Y vtx kshorts",1000,-10.,10.,1000,-10.,10.);
histosTH2F["h2dimxzk"] = new TH2F("h2dimxzk","X vs Z vtx kshorts",1000,-10.,10.,300,-30.,30.);
histosTH2F["h2dimyzk"] = new TH2F("h2dimyzk","Y vs Z vtx kshorts",1000,-10.,10.,300,-30.,30.);


//histosTH1F["hvtxchi2"] = new TH1F("hvtxchi2","chi2 vtx",1100,-100.,1000.);
//histosTH1F["hvtxchi2fin"] = new TH1F("hvtxchi2fin","chi2 vtx",1100,-100.,1000.);
//...Luiz
histosTH1F["hvtxchi2"] = new TH1F("hvtxchi2","#chi^{2} vtx",1100,-100.,1000.);
histosTH1F["hvtxndof"] = new TH1F("hvtxndof","ndof vtx",1020,-2.,100.);
histosTH1F["hvtxchi2fin"] = new TH1F("hvtxchi2fin","#chi^{2} vtx fin",1100,-100.,1000.);

//histosTH1F["heHF"] = new TH1F("heHF","HF tower energy",550,-10,100);
//...Luiz
histosTH1F["heHF"] = new TH1F("heHF","HF tower energy (GeV)",550,-10,100);
histosTH1F["hnHF"] = new TH1F("hnHF","n HF towers (E>5 GeV)",200,0,200);

histosTH1F["hxiL"] = new TH1F("hxiL","#xiL ",100,-0.1,0.1);
histosTH1F["hxiR"] = new TH1F("hxiR","#xiR ",100,-0.1,0.1);
//...Luiz
histosTH1F["hrapy"] = new TH1F("hrapy","rapidity",2000,-10,10);
//
histosTH1F["hxiL2"] = new TH1F("hxiL2","#xiL ",100,-0.1,0.1);
histosTH1F["hxiR2"] = new TH1F("hxiR2","#xiR ",100,-0.1,0.1);
//...Luiz
histosTH1F["hrapy2"] = new TH1F("hrapy2","rapidity 2",2000,-10,10);
//

int massbins=250;

histosTH1F["hm"] = new TH1F("hm","M_{4#pi} ",massbins,0,5.);
//...Luiz
//  histosTH1F["hmxicut"] = new TH1F("hmxicit","M_{4#pi} ",massbins,0,5.);
histosTH1F["hmxicut"] = new TH1F("hmxicut","M_{4#pi} ",massbins,0,5.);

histosTH1F["hm2rec"] = new TH1F("hm2rec","M_{4#pi} ",massbins,0,5.);
histosTH1F["hm2recbis"] = new TH1F("hm2recbis","M_{4#pi}",2*massbins,0,5.);

//...Luiz
histosTH1F["hm2recPPPP"] = new TH1F("hm2recPPPP","M_{4#pi} ",massbins,0,5.);
//histosTH1F["hm2recPP"] = new TH1F("hm2recPP","M_{#pi#pi} ",massbins,0,5.);
histosTH1F["hm2recKKKK"] = new TH1F("hm2recKKKK","M_{4K} ",massbins,0,5.);
//histosTH1F["hm2recMM"] = new TH1F("hm2recMM","M_{2#mu} ",massbins,0,5.);
//histosTH1F["hm2recEE"] = new TH1F("hm2recEE","M_{2e} ",massbins,0,5.);
//histosTH1F["hm2recpp"] = new TH1F("hm2recpp","M_{2p} ",massbins,0,5.);

//...OS-SS
histosTH1F["hm2recOS"] = new TH1F("hm2recOS","M_{4#pi} OS",massbins,0,5.);
histosTH1F["hm2recOS2"] = new TH1F("hm2recOS2","M_{4#pi} OS",2.0*massbins,0,10.);
//
histosTH1F["hm2recSS"] = new TH1F("hm2recSS","M_{4#pi} SS",massbins,0,5.);
//
histosTH1F["hm2recOS_diag"] = new TH1F("hm2recOS_diag","M_{4#pi} TB/BT OS",massbins,0,5.);
histosTH1F["hm2recSS_diag"] = new TH1F("hm2recSS_diag","M_{4#pi} TB/BT SS",massbins,0,5.);
histosTH1F["hm2recOS_ttbb"] = new TH1F("hm2recOS_ttbb","M_{4#pi} TT/BB OS",massbins,0,5.);
histosTH1F["hm2recSS_ttbb"] = new TH1F("hm2recSS_ttbb","M_{4#pi} TT/BB SS",massbins,0,5.);

//...2OS-2SS
histosTH1F["hm2rec2OS"] = new TH1F("hm2rec2OS","M_{4#pi} OS",massbins,0,5.);
histosTH1F["hm2rec2OSvee"] = new TH1F("hm2rec2OSvee","M_{4#pi} OS",massbins,0,5.);
//
histosTH1F["hm2rec2OSvee11"] = new TH1F("hm2rec2OSvee11","M_{K#pi} OS",massbins,0,5.);


histosTH1F["hm2rec2OSvee11a"] = new TH1F("hm2rec2OSvee11a","M_{K#pi} OS",massbins,0,5.);
histosTH1F["hm2rec2OSvee11b"] = new TH1F("hm2rec2OSvee11b","M_{K#pi} OS",massbins,0,5.);
histosTH1F["hm2rec2OSvee11c"] = new TH1F("hm2rec2OSvee11c","M_{K#pi} OS",massbins,0,5.);
histosTH1F["hm2rec2OSvee11d"] = new TH1F("hm2rec2OSvee11d","M_{K#pi} OS",massbins,0,5.);
histosTH1F["hm2rec2OSvee11e"] = new TH1F("hm2rec2OSvee11e","M_{K#pi} OS",massbins,0,5.);
```

```
  histosTH1F["hm2rec2OSvee11f"] = new TH1F("hm2rec2OSvee11f","M_{K#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee11g"] = new TH1F("hm2rec2OSvee11g","M_{K#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee11h"] = new TH1F("hm2rec2OSvee11h","M_{K#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee11i"] = new TH1F("hm2rec2OSvee11i","M_{K#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee11j"] = new TH1F("hm2rec2OSvee11j","M_{K#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee11k"] = new TH1F("hm2rec2OSvee11k","M_{K#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee11m"] = new TH1F("hm2rec2OSvee11m","M_{K#pi} OS",massbins,0,5.);


  histosTH1F["hm2rec2OSvee02"] = new TH1F("hm2rec2OSvee02","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee01"] = new TH1F("hm2rec2OSvee01","M_{2#pi} OS",massbins,0,5.);
  //
  histosTH1F["hm2rec2OSvee9"] = new TH1F("hm2rec2OSvee9","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee90"] = new TH1F("hm2rec2OSvee90","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee91"] = new TH1F("hm2rec2OSvee91","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvee92"] = new TH1F("hm2rec2OSvee92","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvtx0"] = new TH1F("hm2rec2OSvtx0","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvtx01"] = new TH1F("hm2rec2OSvtx01","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvtx02"] = new TH1F("hm2rec2OSvtx02","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvtx11"] = new TH1F("hm2rec2OSvtx11","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvtx1"] = new TH1F("hm2rec2OSvtx1","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSvtx2"] = new TH1F("hm2rec2OSvtx2","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSveeno"] = new TH1F("hm2rec2OSveeno","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSveeno11"] = new TH1F("hm2rec2OSveeno11","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSveeno02"] = new TH1F("hm2rec2OSveeno02","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec2OSveeno01"] = new TH1F("hm2rec2OSveeno01","M_{4#pi} OS",massbins,0,5.);
  //
  histosTH1F["hm2rec2OS2"] = new TH1F("hm2rec2OS2","M_{4#pi} OS",2.0*massbins,0,10.);
  // 12 34 13 24  ...for now
  histosTH1F["hm2rec2OS_pipi"] = new TH1F("hm2rec2OS_pipi","M_{#pi#pi} OS",massbins,0,5.);
  //
  //...primary
  histosTH1F["hm2rec2OS_pi1pi2"] = new TH1F("hm2rec2OS_pi1pi2","M_{#pi_{1}#pi_{2}} OS",massbins,0
,5.);
  histosTH1F["hm2rec2OS_pi3pi4"] = new TH1F("hm2rec2OS_pi3pi4","M_{#pi_{3}#pi_{4}} OS",massbins,0
,5.);
  histosTH1F["hm2rec2OS_pi1pi3"] = new TH1F("hm2rec2OS_pi1pi3","M_{#pi_{1}#pi_{3}} OS",massbins,0
,5.);
  histosTH1F["hm2rec2OS_pi2pi4"] = new TH1F("hm2rec2OS_pi2pi4","M_{#pi_{2}#pi_{4}} OS",massbins,0
,5.);
  //...v2
  histosTH1F["hm2rec2OS_pi1pi2v2"] = new TH1F("hm2rec2OS_pi1pi2v2","M_{#pi_{1}#pi_{2}} OS",massbi
ns,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4v2"] = new TH1F("hm2rec2OS_pi3pi4v2","M_{#pi_{3}#pi_{4}} OS",massbi
ns,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3v2"] = new TH1F("hm2rec2OS_pi1pi3v2","M_{#pi_{1}#pi_{3}} OS",massbi
ns,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4v2"] = new TH1F("hm2rec2OS_pi2pi4v2","M_{#pi_{2}#pi_{4}} OS",massbi
ns,0,5.);
  //...2dim
  histosTH2F["hm2dim2OS_pi1pi2_pi3pi4"] = new TH2F("hm2dim2OS_pi1pi2_pi3pi4","M_{#pi_{1}#pi_{2}}
vs M_{#pi_{3}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  histosTH2F["hm2dim2OS_pi1pi3_pi2pi4"] = new TH2F("hm2dim2OS_pi1pi3_pi2pi4","M_{#pi_{1}#pi_{3}}
vs M_{#pi_{2}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  //...v2
  histosTH2F["hm2dim2OS_pi1pi2_pi3pi4v2"] = new TH2F("hm2dim2OS_pi1pi2_pi3pi4v2","M_{#pi_{1}#pi_{
2}} vs M_{#pi_{3}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  histosTH2F["hm2dim2OS_pi1pi3_pi2pi4v2"] = new TH2F("hm2dim2OS_pi1pi3_pi2pi4v2","M_{#pi_{1}#pi_{
3}} vs M_{#pi_{2}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  //
  //...testing vee9  ........cut 9
  histosTH1F["hm2rec2OS_pi1pi2vee9"] = new TH1F("hm2rec2OS_pi1pi2vee9","M_{#pi_{1}#pi_{2}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vee9"] = new TH1F("hm2rec2OS_pi3pi4vee9","M_{#pi_{3}#pi_{4}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vee9"] = new TH1F("hm2rec2OS_pi1pi3vee9","M_{#pi_{1}#pi_{3}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vee9"] = new TH1F("hm2rec2OS_pi2pi4vee9","M_{#pi_{2}#pi_{4}} OS",ma
ssbins,0,5.);
  //
  histosTH1F["hm2rec2OS_pi1pi2vee90"] = new TH1F("hm2rec2OS_pi1pi2vee90","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
```

```
  histosTH1F["hm2rec2OS_pi3pi4vee90"] = new TH1F("hm2rec2OS_pi3pi4vee90","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vee90"] = new TH1F("hm2rec2OS_pi1pi3vee90","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vee90"] = new TH1F("hm2rec2OS_pi2pi4vee90","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
  //
  histosTH1F["hm2rec2OS_pi1pi2vee91"] = new TH1F("hm2rec2OS_pi1pi2vee91","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vee91"] = new TH1F("hm2rec2OS_pi3pi4vee91","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vee91"] = new TH1F("hm2rec2OS_pi1pi3vee91","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vee91"] = new TH1F("hm2rec2OS_pi2pi4vee91","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
  //
  histosTH1F["hm2rec2OS_pi1pi2vee92"] = new TH1F("hm2rec2OS_pi1pi2vee92","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vee92"] = new TH1F("hm2rec2OS_pi3pi4vee92","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vee92"] = new TH1F("hm2rec2OS_pi1pi3vee92","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vee92"] = new TH1F("hm2rec2OS_pi2pi4vee92","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
  //
  histosTH1F["hm2rec2OS_pi1pi2vtx0"] = new TH1F("hm2rec2OS_pi1pi2vtx0","M_{#pi_{1}#pi_{2}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vtx0"] = new TH1F("hm2rec2OS_pi3pi4vtx0","M_{#pi_{3}#pi_{4}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vtx0"] = new TH1F("hm2rec2OS_pi1pi3vtx0","M_{#pi_{1}#pi_{3}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vtx0"] = new TH1F("hm2rec2OS_pi2pi4vtx0","M_{#pi_{2}#pi_{4}} OS",ma
ssbins,0,5.);
   //
  histosTH1F["hm2rec2OS_pi1pi2vtx01"] = new TH1F("hm2rec2OS_pi1pi2vtx01","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vtx01"] = new TH1F("hm2rec2OS_pi3pi4vtx01","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vtx01"] = new TH1F("hm2rec2OS_pi1pi3vtx01","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vtx01"] = new TH1F("hm2rec2OS_pi2pi4vtx01","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
   //
  histosTH1F["hm2rec2OS_pi1pi2vtx02"] = new TH1F("hm2rec2OS_pi1pi2vtx02","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vtx02"] = new TH1F("hm2rec2OS_pi3pi4vtx02","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vtx02"] = new TH1F("hm2rec2OS_pi1pi3vtx02","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vtx02"] = new TH1F("hm2rec2OS_pi2pi4vtx02","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
   //
  histosTH1F["hm2rec2OS_pi1pi2vtx11"] = new TH1F("hm2rec2OS_pi1pi2vtx11","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vtx11"] = new TH1F("hm2rec2OS_pi3pi4vtx11","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vtx11"] = new TH1F("hm2rec2OS_pi1pi3vtx11","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vtx11"] = new TH1F("hm2rec2OS_pi2pi4vtx11","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
  //
  histosTH1F["hm2rec2OS_pi1pi2vtx1"] = new TH1F("hm2rec2OS_pi1pi2vtx1","M_{#pi_{1}#pi_{2}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vtx1"] = new TH1F("hm2rec2OS_pi3pi4vtx1","M_{#pi_{3}#pi_{4}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vtx1"] = new TH1F("hm2rec2OS_pi1pi3vtx1","M_{#pi_{1}#pi_{3}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vtx1"] = new TH1F("hm2rec2OS_pi2pi4vtx1","M_{#pi_{2}#pi_{4}} OS",ma
ssbins,0,5.);
  //
  histosTH1F["hm2rec2OS_pi1pi2vtx2"] = new TH1F("hm2rec2OS_pi1pi2vtx2","M_{#pi_{1}#pi_{2}} OS",ma
ssbins,0,5.);
```

```
  histosTH1F["hm2rec2OS_pi3pi4vtx2"] = new TH1F("hm2rec2OS_pi3pi4vtx2","M_{#pi_{3}#pi_{4}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vtx2"] = new TH1F("hm2rec2OS_pi1pi3vtx2","M_{#pi_{1}#pi_{3}} OS",ma
ssbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vtx2"] = new TH1F("hm2rec2OS_pi2pi4vtx2","M_{#pi_{2}#pi_{4}} OS",ma
ssbins,0,5.);
  //

  //...secondary vee11
  /*
  histosTH1F["hm2rec2OS_pi1pi2vee11"] = new TH1F("hm2rec2OS_pi1pi2vee11","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vee11"] = new TH1F("hm2rec2OS_pi3pi4vee11","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vee11"] = new TH1F("hm2rec2OS_pi1pi3vee11","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vee11"] = new TH1F("hm2rec2OS_pi2pi4vee11","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
  */

   /*
     ...first combining, then select the Q_pairs=0

     pi1pi2 pi3k4
     pi1pi3 pi2k4
     pi2pi3 pi1k4

     pi1pi2 k3pi4
     pi1pi4 k3pi2
     pi2pi4 k3pi1

     pi1k2 pi3pi4
     pi3k2 pi1pi4
     pi4k2 pi1pi3

     k1pi2 pi3pi4
     k1pi3 pi2pi4
     k1pi4 pi2pi3

   */

   //...A
   histosTH1F["hm2rec2OS_pi1pi2vee11"] = new TH1F("hm2rec2OS_pi1pi2vee11","M_{#pi_{1}#pi_{2}} OS"
,massbins,0,5.);
   histosTH1F["hm2rec2OS_pi3k4vee11"] = new TH1F("hm2rec2OS_pi3k4vee11","M_{#pi_{3}K_{4}} OS",mas
sbins,0,5.);
   //
   histosTH1F["hm2rec2OS_pi1pi3vee11"] = new TH1F("hm2rec2OS_pi1pi3vee11","M_{#pi_{1}#pi_{3}} OS"
,massbins,0,5.);
   histosTH1F["hm2rec2OS_pi2k4vee11"] = new TH1F("hm2rec2OS_pi2k4vee11","M_{#pi_{2}K_{4}} OS",mas
sbins,0,5.);
   //
   histosTH1F["hm2rec2OS_pi2pi3vee11"] = new TH1F("hm2rec2OS_pi2pi3vee11","M_{#pi_{2}#pi_{3}} OS"
,massbins,0,5.);
   histosTH1F["hm2rec2OS_pi1k4vee11"] = new TH1F("hm2rec2OS_pi1k4vee11","M_{#pi_{1}K_{4}} OS",mas
sbins,0,5.);

   //...B
   //histosTH1F["hm2rec2OS_pi1pi2vee11"] = new TH1F("hm2rec2OS_pi1pi2vee11","M_{#pi_{1}#pi_{2}} O
S",massbins,0,5.);
   histosTH1F["hm2rec2OS_k3pi4vee11"] = new TH1F("hm2rec2OS_k3pi4vee11","M_{K_{3}#pi_{4}} OS",mas
sbins,0,5.);
   //
   histosTH1F["hm2rec2OS_pi1pi4vee11"] = new TH1F("hm2rec2OS_pi1pi4vee11","M_{#pi_{1}#pi_{4}} OS"
,massbins,0,5.);
   histosTH1F["hm2rec2OS_k3pi2vee11"] = new TH1F("hm2rec2OS_k3pi2vee11","M_{K_{3}#pi_{2}} OS",mas
sbins,0,5.);
   //
   histosTH1F["hm2rec2OS_pi2pi4vee11"] = new TH1F("hm2rec2OS_pi2pi4vee11","M_{#pi_{2}#pi_{4}} OS"
,massbins,0,5.);
   histosTH1F["hm2rec2OS_k3pi1vee11"] = new TH1F("hm2rec2OS_k3pi1vee11","M_{K_{3}#pi_{1}} OS",mas
sbins,0,5.);
```

```
   //...C
   histosTH1F["hm2rec2OS_pi1k2vee11"] = new TH1F("hm2rec2OS_pi1k2vee11","M_{#pi_{1}K_{2}} OS",mas
sbins,0,5.);
   histosTH1F["hm2rec2OS_pi3pi4vee11"] = new TH1F("hm2rec2OS_pi3pi4vee11","M_{#pi_{3}#pi_{4}} OS"
,massbins,0,5.);
   //
   histosTH1F["hm2rec2OS_pi3k2vee11"] = new TH1F("hm2rec2OS_pi3k2vee11","M_{#pi_{3}K_{2}} OS",mas
sbins,0,5.);
   //histosTH1F["hm2rec2OS_pi1pi4vee11"] = new TH1F("hm2rec2OS_pi1pi4vee11","M_{#pi_{1}#pi_{4}} O
S",massbins,0,5.);
   //
   histosTH1F["hm2rec2OS_pi4k2vee11"] = new TH1F("hm2rec2OS_pi4k2vee11","M_{#pi_{4}K_{2}} OS",mas
sbins,0,5.);
   //histosTH1F["hm2rec2OS_pi1pi3vee11"] = new TH1F("hm2rec2OS_pi1pi3vee11","M_{#pi_{1}#pi_{3}} O
S",massbins,0,5.);

   //...D
   histosTH1F["hm2rec2OS_k1pi2vee11"] = new TH1F("hm2rec2OS_k1pi2vee11","M_{K_{1}#pi_{2}} OS",mas
sbins,0,5.);
   //histosTH1F["hm2rec2OS_pi3pi4vee11"] = new TH1F("hm2rec2OS_pi3pi4vee11","M_{#pi_{3}#pi_{4}} O
S",massbins,0,5.);
   //
   histosTH1F["hm2rec2OS_k1pi3vee11"] = new TH1F("hm2rec2OS_k1pi3vee11","M_{K_{1}#pi_{3}} OS",mas
sbins,0,5.);
   //histosTH1F["hm2rec2OS_pi2pi4vee11"] = new TH1F("hm2rec2OS_pi2pi4vee11","M_{#pi_{2}#pi_{4}} O
S",massbins,0,5.);
   //
   histosTH1F["hm2rec2OS_k1pi4vee11"] = new TH1F("hm2rec2OS_k1pi4vee11","M_{K_{1}#pi_{4}} OS",mas
sbins,0,5.);
   //histosTH1F["hm2rec2OS_pi2pi3vee11"] = new TH1F("hm2rec2OS_pi2pi3vee11","M_{#pi_{2}#pi_{3}} O
S",massbins,0,5.);


  //...vee02
  histosTH1F["hm2rec2OS_pi1pi2vee02"] = new TH1F("hm2rec2OS_pi1pi2vee02","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vee02"] = new TH1F("hm2rec2OS_pi3pi4vee02","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vee02"] = new TH1F("hm2rec2OS_pi1pi3vee02","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vee02"] = new TH1F("hm2rec2OS_pi2pi4vee02","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
   //...vee01
  histosTH1F["hm2rec2OS_pi1pi2vee01"] = new TH1F("hm2rec2OS_pi1pi2vee01","M_{#pi_{1}#pi_{2}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4vee01"] = new TH1F("hm2rec2OS_pi3pi4vee01","M_{#pi_{3}#pi_{4}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3vee01"] = new TH1F("hm2rec2OS_pi1pi3vee01","M_{#pi_{1}#pi_{3}} OS",
massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4vee01"] = new TH1F("hm2rec2OS_pi2pi4vee01","M_{#pi_{2}#pi_{4}} OS",
massbins,0,5.);
 //...2dim vee11
  histosTH2F["hm2dim2OS_pi1pi2_pi3pi4vee11"] = new TH2F("hm2dim2OS_pi1pi2_pi3pi4vee11","M_{#pi_{1
}#pi_{2}} vs M_{#pi_{3}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  histosTH2F["hm2dim2OS_pi1pi3_pi2pi4vee11"] = new TH2F("hm2dim2OS_pi1pi3_pi2pi4vee11","M_{#pi_{1
}#pi_{3}} vs M_{#pi_{2}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  //...vee02
  histosTH2F["hm2dim2OS_pi1pi2_pi3pi4vee02"] = new TH2F("hm2dim2OS_pi1pi2_pi3pi4vee02","M_{#pi_{1
}#pi_{2}} vs M_{#pi_{3}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  histosTH2F["hm2dim2OS_pi1pi3_pi2pi4vee02"] = new TH2F("hm2dim2OS_pi1pi3_pi2pi4vee02","M_{#pi_{1
}#pi_{3}} vs M_{#pi_{2}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  //...vee01
  histosTH2F["hm2dim2OS_pi1pi2_pi3pi4vee01"] = new TH2F("hm2dim2OS_pi1pi2_pi3pi4vee01","M_{#pi_{1
}#pi_{2}} vs M_{#pi_{3}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  histosTH2F["hm2dim2OS_pi1pi3_pi2pi4vee01"] = new TH2F("hm2dim2OS_pi1pi3_pi2pi4vee01","M_{#pi_{1
}#pi_{3}} vs M_{#pi_{2}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  //
  //...secondary vee11 with no PID
  histosTH1F["hm2rec2OS_pi1pi2veeno11"] = new TH1F("hm2rec2OS_pi1pi2veeno11","M_{#pi_{1}#pi_{2}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4veeno11"] = new TH1F("hm2rec2OS_pi3pi4veeno11","M_{#pi_{3}#pi_{4}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3veeno11"] = new TH1F("hm2rec2OS_pi1pi3veeno11","M_{#pi_{1}#pi_{3}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4veeno11"] = new TH1F("hm2rec2OS_pi2pi4veeno11","M_{#pi_{2}#pi_{4}}
OS",massbins,0,5.);
  //...veeno02
  histosTH1F["hm2rec2OS_pi1pi2veeno02"] = new TH1F("hm2rec2OS_pi1pi2veeno02","M_{#pi_{1}#pi_{2}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4veeno02"] = new TH1F("hm2rec2OS_pi3pi4veeno02","M_{#pi_{3}#pi_{4}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3veeno02"] = new TH1F("hm2rec2OS_pi1pi3veeno02","M_{#pi_{1}#pi_{3}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4veeno02"] = new TH1F("hm2rec2OS_pi2pi4veeno02","M_{#pi_{2}#pi_{4}}
OS",massbins,0,5.);
  //...veeno01
  histosTH1F["hm2rec2OS_pi1pi2veeno01"] = new TH1F("hm2rec2OS_pi1pi2veeno01","M_{#pi_{1}#pi_{2}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi3pi4veeno01"] = new TH1F("hm2rec2OS_pi3pi4veeno01","M_{#pi_{3}#pi_{4}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi1pi3veeno01"] = new TH1F("hm2rec2OS_pi1pi3veeno01","M_{#pi_{1}#pi_{3}}
OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_pi2pi4veeno01"] = new TH1F("hm2rec2OS_pi2pi4veeno01","M_{#pi_{2}#pi_{4}}
OS",massbins,0,5.);
  //...2dim veeno11
  histosTH2F["hm2dim2OS_pi1pi2_pi3pi4veeno11"] = new TH2F("hm2dim2OS_pi1pi2_pi3pi4veeno11","M_{#p
i_{1}#pi_{2}} vs M_{#pi_{3}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  histosTH2F["hm2dim2OS_pi1pi3_pi2pi4veeno11"] = new TH2F("hm2dim2OS_pi1pi3_pi2pi4veeno11","M_{#p
i_{1}#pi_{3}} vs M_{#pi_{2}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  //...veeno02
  histosTH2F["hm2dim2OS_pi1pi2_pi3pi4veeno02"] = new TH2F("hm2dim2OS_pi1pi2_pi3pi4veeno02","M_{#p
i_{1}#pi_{2}} vs M_{#pi_{3}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  histosTH2F["hm2dim2OS_pi1pi3_pi2pi4veeno02"] = new TH2F("hm2dim2OS_pi1pi3_pi2pi4veeno02","M_{#p
i_{1}#pi_{3}} vs M_{#pi_{2}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  //...veeno01
  histosTH2F["hm2dim2OS_pi1pi2_pi3pi4veeno01"] = new TH2F("hm2dim2OS_pi1pi2_pi3pi4veeno01","M_{#p
i_{1}#pi_{2}} vs M_{#pi_{3}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  histosTH2F["hm2dim2OS_pi1pi3_pi2pi4veeno01"] = new TH2F("hm2dim2OS_pi1pi3_pi2pi4veeno01","M_{#p
i_{1}#pi_{3}} vs M_{#pi_{2}#pi_{4}} OS",massbins,0,5.,massbins,0,5.);
  //
  //..........
  //
  //...Kaons
  histosTH1F["hm2rec2OS_k1k2"] = new TH1F("hm2rec2OS_k1k2","M_{k_{1}k_{2}} OS",2*massbins,0,5.);
  histosTH1F["hm2rec2OS_k3k4"] = new TH1F("hm2rec2OS_k3k4","M_{k_{3}k_{4}} OS",2*massbins,0,5.);
  histosTH1F["hm2rec2OS_k1k3"] = new TH1F("hm2rec2OS_k1k3","M_{k_{1}k_{3}} OS",2*massbins,0,5.);
  histosTH1F["hm2rec2OS_k2k4"] = new TH1F("hm2rec2OS_k2k4","M_{k_{2}k_{4}} OS",2*massbins,0,5.);
  //...v2
  histosTH1F["hm2rec2OS_k1k2v2"] = new TH1F("hm2rec2OS_k1k2v2","M_{k_{1}k_{2}} OS",2*massbins,0,5
.);
  histosTH1F["hm2rec2OS_k3k4v2"] = new TH1F("hm2rec2OS_k3k4v2","M_{k_{3}k_{4}} OS",2*massbins,0,5
.);
  histosTH1F["hm2rec2OS_k1k3v2"] = new TH1F("hm2rec2OS_k1k3v2","M_{k_{1}k_{3}} OS",2*massbins,0,5
.);
  histosTH1F["hm2rec2OS_k2k4v2"] = new TH1F("hm2rec2OS_k2k4v2","M_{k_{2}k_{4}} OS",2*massbins,0,5
.);
  //...2dim
  histosTH2F["hm2dim2OS_k1k2_k3k4"] = new TH2F("hm2dim2OS_k1k2_k3k4","M_{k_{1}k_{2}} vs M_{k_{3}k
_{4}} OS",2*massbins,0,5.,2*massbins,0,5.);
  histosTH2F["hm2dim2OS_k1k3_k2k4"] = new TH2F("hm2dim2OS_k1k3_k2k4","M_{k_{1}k_{3}} vs M_{k_{2}k
_{4}} OS",2*massbins,0,5.,2*massbins,0,5.);
  //...v2
  histosTH2F["hm2dim2OS_k1k2_k3k4v2"] = new TH2F("hm2dim2OS_k1k2_k3k4v2","M_{k_{1}k_{2}} vs M_{k_
{3}k_{4}} OS",2*massbins,0,5.,2*massbins,0,5.);
  histosTH2F["hm2dim2OS_k1k3_k2k4v2"] = new TH2F("hm2dim2OS_k1k3_k2k4v2","M_{k_{1}k_{3}} vs M_{k_
{2}k_{4}} OS",2*massbins,0,5.,2*massbins,0,5.);
  //...Kaons
  //
  histosTH1F["hm2rec2SS"] = new TH1F("hm2rec2SS","M_{4#pi} SS",massbins,0,5.);
  //...2OSdiag
  histosTH1F["hm2rec2OS_diag"] = new TH1F("hm2rec2OS_diag","M_{4#pi} TB/BT OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_diag2"] = new TH1F("hm2rec2OS_diag2","M_{4#pi} TB/BT OS",1.60*massbins,0.
0,8.0);
```

```
  histosTH1F["hm2rec2OS_diag3"] = new TH1F("hm2rec2OS_diag3","M_{4#pi} TB/BT OS",0.50*massbins,0.
0,2.5);
  histosTH1F["hm2rec2OS_diag4"] = new TH1F("hm2rec2OS_diag4","M_{4#pi} TB/BT OS",0.24*massbins,2.
5,4.0);
  histosTH1F["hm2rec2OS_diag5"] = new TH1F("hm2rec2OS_diag5","M_{4#pi} TB/BT OS",0.32*massbins,4.
0,8.0);

 //0-2.5 (125bins), 2.5-4(60bins), 4-8(80bins)
 double xmin1 = 0.;
 double xmax1 = 2.5;
 int nbins1 = 125;

 double xmin2 = 2.5;
 double xmax2 = 4.;
 int nbins2 = 60;

 double xmin3 = 4.;
 double xmax3 = 8.;
 int nbins3 = 80;

 double bwidth1 = (xmax1 - xmin1)/nbins1;
 double bwidth2 = (xmax2 - xmin2)/nbins2;
 double bwidth3 = (xmax3 - xmin3)/nbins3;

 int nbinstot = nbins1 + nbins2 + nbins3;
 //...Luiz
 double edges[nbinstot+1] ;

 //nbinstot++;

 int nbins=0;

 for( int i=0; i<nbins1; i++){ edges[nbins] = xmin1 + bwidth1 * i; nbins++;}
 for( int i=0; i<nbins2; i++){ edges[nbins] = xmin2 + bwidth2 * i; nbins++;}
 //...Luiz
 for( int i=0; i<=nbins3; i++){ edges[nbins] = xmin3 + bwidth3 * i; nbins++;}

 histosTH1F["hm2rec2OS_ttbb2varbin"] = new TH1F("hm2rec2OS_ttbb2varbin","TTBB variable bins",nbin
stot,edges);
 histosTH1F["hm2rec2OS_diag2varbin"] = new TH1F("hm2rec2OS_diag2varbin","DIAG variable bins",nbin
stot,edges);

  //...Pions
  histosTH1F["hm2rec2OS_diag_pi1pi2"] = new TH1F("hm2rec2OS_diag_pi1pi2","M_{#pi_{1}#pi_{2}} OS",
2.0*massbins,0,10.);
  histosTH1F["hm2rec2OS_diag_pi3pi4"] = new TH1F("hm2rec2OS_diag_pi3pi4","M_{#pi_{3}#pi_{4}} OS",
2.0*massbins,0,10.);
  histosTH1F["hm2rec2OS_diag_pi1pi3"] = new TH1F("hm2rec2OS_diag_pi1pi3","M_{#pi_{1}#pi_{3}} OS",
2.0*massbins,0,10.);
  histosTH1F["hm2rec2OS_diag_pi2pi4"] = new TH1F("hm2rec2OS_diag_pi2pi4","M_{#pi_{2}#pi_{4}} OS",
2.0*massbins,0,10.);
  //...Kaons
  histosTH1F["hm2rec2OS_diag_k1k2"] = new TH1F("hm2rec2OS_diag_k1k2","M_{k_{1}k_{2}} OS",2.0*mass
bins,0,10.);
  histosTH1F["hm2rec2OS_diag_k3k4"] = new TH1F("hm2rec2OS_diag_k3k4","M_{k_{3}k_{4}} OS",2.0*mass
bins,0,10.);
  histosTH1F["hm2rec2OS_diag_k1k3"] = new TH1F("hm2rec2OS_diag_k1k3","M_{k_{1}k_{3}} OS",2.0*mass
bins,0,10.);
  histosTH1F["hm2rec2OS_diag_k2k4"] = new TH1F("hm2rec2OS_diag_k2k4","M_{k_{2}k_{4}} OS",2.0*mass
bins,0,10.);

  //...2SSdiag
  histosTH1F["hm2rec2SS_diag"] = new TH1F("hm2rec2SS_diag","M_{4#pi} TB/BT SS",massbins,0,5.);
  //
  //...2OSttbb
  histosTH1F["hm2rec2OS_ttbb"] = new TH1F("hm2rec2OS_ttbb","M_{4#pi} TT/BB OS",massbins,0,5.);
  histosTH1F["hm2rec2OS_ttbb2"] = new TH1F("hm2rec2OS_ttbb2","M_{4#pi} TT/BB OS",1.60*massbins,0.
0,8.0);
  histosTH1F["hm2rec2OS_ttbb3"] = new TH1F("hm2rec2OS_ttbb3","M_{4#pi} TT/BB OS",0.50*massbins,0.
0,2.5);
  histosTH1F["hm2rec2OS_ttbb4"] = new TH1F("hm2rec2OS_ttbb4","M_{4#pi} TT/BB OS",0.24*massbins,2.
5,4.0);
```

```
  histosTH1F["hm2rec2OS_ttbb5"] = new TH1F("hm2rec2OS_ttbb5","M_{4#pi} TT/BB OS",0.32*massbins,4.
0,8.0);

  //...Pions
  histosTH1F["hm2rec2OS_ttbb_pi1pi2"] = new TH1F("hm2rec2OS_ttbb_pi1pi2","M_{#pi_{1}#pi_{2}} OS",
2.0*massbins,0,10.);
  histosTH1F["hm2rec2OS_ttbb_pi3pi4"] = new TH1F("hm2rec2OS_ttbb_pi3pi4","M_{#pi_{3}#pi_{4}} OS",
2.0*massbins,0,10.);
  histosTH1F["hm2rec2OS_ttbb_pi1pi3"] = new TH1F("hm2rec2OS_ttbb_pi1pi3","M_{#pi_{1}#pi_{3}} OS",
2.0*massbins,0,10.);
  histosTH1F["hm2rec2OS_ttbb_pi2pi4"] = new TH1F("hm2rec2OS_ttbb_pi2pi4","M_{#pi_{2}#pi_{4}} OS",
2.0*massbins,0,10.);
  //...Kaons
  histosTH1F["hm2rec2OS_ttbb_k1k2"] = new TH1F("hm2rec2OS_ttbb_k1k2","M_{k_{1}k_{2}} OS",2.0*mass
bins,0,10.);
  histosTH1F["hm2rec2OS_ttbb_k3k4"] = new TH1F("hm2rec2OS_ttbb_k3k4","M_{k_{3}k_{4}} OS",2.0*mass
bins,0,10.);
  histosTH1F["hm2rec2OS_ttbb_k1k3"] = new TH1F("hm2rec2OS_ttbb_k1k3","M_{k_{1}k_{3}} OS",2.0*mass
bins,0,10.);
  histosTH1F["hm2rec2OS_ttbb_k2k4"] = new TH1F("hm2rec2OS_ttbb_k2k4","M_{k_{2}k_{4}} OS",2.0*mass
bins,0,10.);
  //...2SSttbb
  histosTH1F["hm2rec2SS_ttbb"] = new TH1F("hm2rec2SS_ttbb","M_{4#pi} TT/BB SS",massbins,0,5.);
  //

  //histosTH1F["hm2rec2OS_diag_trkP"] = new TH1F("hm2rec2OS_diag_trkP","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec2OS_diag_trkM"] = new TH1F("hm2rec2OS_diag_trkM","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}<0",massbins,0,5.);
  //histosTH1F["hm2rec2OS_ttbb_trkP"] = new TH1F("hm2rec2OS_ttbb_trkP","M_{4#pi} TT/BB OS, py_{#p
i1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec2OS_ttbb_trkM"] = new TH1F("hm2rec2OS_ttbb_trkM","M_{4#pi} TT/BB OS, py_{#p
i1}py_{#pi2}<0",massbins,0,5.);
  //...Luiz
  histosTH1F["hm2rec2OS_diag_trkP"] = new TH1F("hm2rec2OS_diag_trkP","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec2OS_diag_trkM"] = new TH1F("hm2rec2OS_diag_trkM","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}<0",massbins,0,5.);
  histosTH1F["hm2rec2OS_ttbb_trkP"] = new TH1F("hm2rec2OS_ttbb_trkP","M_{4#pi} TT/BB OS, py_{#pi_
{1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec2OS_ttbb_trkM"] = new TH1F("hm2rec2OS_ttbb_trkM","M_{4#pi} TT/BB OS, py_{#pi_
{1}}py_{#pi_{2}}<0",massbins,0,5.);

  histosTH1F["hm2rec2OS_diag_pypxP"] = new TH1F("hm2rec2OS_diag_pypxP","M_{4#pi} TB/BT OS, |py/px
|_{4#pi} > 1",massbins,0,5.);
  histosTH1F["hm2rec2OS_diag_pypxM"] = new TH1F("hm2rec2OS_diag_pypxM","M_{4#pi} TB/BT OS, |py/px
|_{4#pi} < 1",massbins,0,5.);
  histosTH1F["hm2rec2OS_ttbb_pypxP"] = new TH1F("hm2rec2OS_ttbb_pypxP","M_{4#pi} TT/BB OS, |py/px
|_{4#pi} > 1",massbins,0,5.);
  histosTH1F["hm2rec2OS_ttbb_pypxM"] = new TH1F("hm2rec2OS_ttbb_pypxM","M_{4#pi} TT/BB OS, |py/px
|_{4#pi} < 1",massbins,0,5.);

  histosTH1F["hm2rec3OS"] = new TH1F("hm2rec3OS","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec3SS"] = new TH1F("hm2rec3SS","M_{4#pi} SS",massbins,0,5.);
  histosTH1F["hm2rec3OS_diag"] = new TH1F("hm2rec3OS_diag","M_{4#pi} TB/BT OS",massbins,0,5.);
  histosTH1F["hm2rec3SS_diag"] = new TH1F("hm2rec3SS_diag","M_{4#pi} TB/BT SS",massbins,0,5.);
  histosTH1F["hm2rec3OS_ttbb"] = new TH1F("hm2rec3OS_ttbb","M_{4#pi} TT/BB OS",massbins,0,5.);
  histosTH1F["hm2rec3SS_ttbb"] = new TH1F("hm2rec3SS_ttbb","M_{4#pi} TT/BB SS",massbins,0,5.);

  //histosTH1F["hm2rec3OS_diag_trkP"] = new TH1F("hm2rec3OS_diag_trkP","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec3OS_diag_trkM"] = new TH1F("hm2rec3OS_diag_trkM","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}<0",massbins,0,5.);
  //histosTH1F["hm2rec3OS_ttbb_trkP"] = new TH1F("hm2rec3OS_ttbb_trkP","M_{4#pi} TT/BB OS, py_{#p
i1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec3OS_ttbb_trkM"] = new TH1F("hm2rec3OS_ttbb_trkM","M_{4#pi} TT/BB OS, py_{#p
i1}py_{#pi2}<0",massbins,0,5.);
  //...Luiz
  histosTH1F["hm2rec3OS_diag_trkP"] = new TH1F("hm2rec3OS_diag_trkP","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec3OS_diag_trkM"] = new TH1F("hm2rec3OS_diag_trkM","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}<0",massbins,0,5.);
```

```
  histosTH1F["hm2rec3OS_ttbb_trkP"] = new TH1F("hm2rec3OS_ttbb_trkP","M_{4#pi} TT/BB OS, py_{#pi_
{1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec3OS_ttbb_trkM"] = new TH1F("hm2rec3OS_ttbb_trkM","M_{4#pi} TT/BB OS, py_{#pi_
{1}}py_{#pi_{2}}<0",massbins,0,5.);

  histosTH1F["hm2rec3OS_diag_pypxP"] = new TH1F("hm2rec3OS_diag_pypxP","M_{4#pi} TB/BT OS, |py/px
|_{4#pi} > 1",massbins,0,5.);
  histosTH1F["hm2rec3OS_diag_pypxM"] = new TH1F("hm2rec3OS_diag_pypxM","M_{4#pi} TB/BT OS, |py/px
|_{4#pi} < 1",massbins,0,5.);
  histosTH1F["hm2rec3OS_ttbb_pypxP"] = new TH1F("hm2rec3OS_ttbb_pypxP","M_{4#pi} TT/BB OS, |py/px
|_{4#pi} > 1",massbins,0,5.);
  histosTH1F["hm2rec3OS_ttbb_pypxM"] = new TH1F("hm2rec3OS_ttbb_pypxM","M_{4#pi} TT/BB OS, |py/px
|_{4#pi} < 1",massbins,0,5.);

  histosTH1F["hm2rec4OS"] = new TH1F("hm2rec4OS","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec4SS"] = new TH1F("hm2rec4SS","M_{4#pi} SS",massbins,0,5.);
  histosTH1F["hm2rec4OS_diag"] = new TH1F("hm2rec4OS_diag","M_{4#pi} TB/BT OS",massbins,0,5.);
  histosTH1F["hm2rec4SS_diag"] = new TH1F("hm2rec4SS_diag","M_{4#pi} TB/BT SS",massbins,0,5.);
  histosTH1F["hm2rec4OS_ttbb"] = new TH1F("hm2rec4OS_ttbb","M_{4#pi} TT/BB OS",massbins,0,5.);
  histosTH1F["hm2rec4SS_ttbb"] = new TH1F("hm2rec4SS_ttbb","M_{4#pi} TT/BB SS",massbins,0,5.);

  //histosTH1F["hm2rec4OS_diag_trkP"] = new TH1F("hm2rec4OS_diag_trkP","M_{4#pi}TB/BT OS, py_{#pi
1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec4OS_diag_trkM"] = new TH1F("hm2rec4OS_diag_trkM","M_{4#pi}TB/BT OS, py_{#pi
1}py_{#pi2}<0",massbins,0,5.);
  //histosTH1F["hm2rec4OS_ttbb_trkP"] = new TH1F("hm2rec4OS_ttbb_trkP","M_{4#pi}TT/BB OS, py_{#pi
1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec4OS_ttbb_trkM"] = new TH1F("hm2rec4OS_ttbb_trkM","M_{4#pi}TT/BB OS, py_{#pi
1}py_{#pi2}<0",massbins,0,5.);
  //...Luiz
  histosTH1F["hm2rec4OS_diag_trkP"] = new TH1F("hm2rec4OS_diag_trkP","M_{4#pi}TB/BT OS, py_{#pi_{
1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec4OS_diag_trkM"] = new TH1F("hm2rec4OS_diag_trkM","M_{4#pi}TB/BT OS, py_{#pi_{
1}}py_{#pi_{2}}<0",massbins,0,5.);
  histosTH1F["hm2rec4OS_ttbb_trkP"] = new TH1F("hm2rec4OS_ttbb_trkP","M_{4#pi}TT/BB OS, py_{#pi_{
1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec4OS_ttbb_trkM"] = new TH1F("hm2rec4OS_ttbb_trkM","M_{4#pi}TT/BB OS, py_{#pi_{
1}}py_{#pi_{2}}<0",massbins,0,5.);

  histosTH1F["hm2rec4OS_diag_pypxP"] = new TH1F("hm2rec4OS_diag_pypxP","M_{4#pi} TB/BT OS, |py/px
|_{4#pi} > 1",massbins,0,5.);
  histosTH1F["hm2rec4OS_diag_pypxM"] = new TH1F("hm2rec4OS_diag_pypxM","M_{4#pi} TB/BT OS, |py/px
|_{4#pi} < 1",massbins,0,5.);
  histosTH1F["hm2rec4OS_ttbb_pypxP"] = new TH1F("hm2rec4OS_ttbb_pypxP","M_{4#pi} TT/BB OS, |py/px
|_{4#pi} > 1",massbins,0,5.);
  histosTH1F["hm2rec4OS_ttbb_pypxM"] = new TH1F("hm2rec4OS_ttbb_pypxM","M_{4#pi} TT/BB OS, |py/px
|_{4#pi} < 1",massbins,0,5.);

  histosTH1F["hm2rec5OS"] = new TH1F("hm2rec5OS","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec5SS"] = new TH1F("hm2rec5SS","M_{4#pi} SS",massbins,0,5.);
  histosTH1F["hm2rec5OS_diag"] = new TH1F("hm2rec5OS_diag","M_{4#pi} TB/BT OS",massbins,0,5.);
  histosTH1F["hm2rec5SS_diag"] = new TH1F("hm2rec5SS_diag","M_{4#pi} TB/BT SS",massbins,0,5.);
  histosTH1F["hm2rec5OS_ttbb"] = new TH1F("hm2rec5OS_ttbb","M_{4#pi} TT/BB OS",massbins,0,5.);
  histosTH1F["hm2rec5SS_ttbb"] = new TH1F("hm2rec5SS_ttbb","M_{4#pi} TT/BB SS",massbins,0,5.);

  //histosTH1F["hm2rec5OS_diag_trkP"] = new TH1F("hm2rec5OS_diag_trkP","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec5OS_diag_trkM"] = new TH1F("hm2rec5OS_diag_trkM","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}<0",massbins,0,5.);
  //histosTH1F["hm2rec5OS_ttbb_trkP"] = new TH1F("hm2rec5OS_ttbb_trkP","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec5OS_ttbb_trkM"] = new TH1F("hm2rec5OS_ttbb_trkM","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}<0",massbins,0,5.);
  //...Luiz
  histosTH1F["hm2rec5OS_diag_trkP"] = new TH1F("hm2rec5OS_diag_trkP","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec5OS_diag_trkM"] = new TH1F("hm2rec5OS_diag_trkM","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}<0",massbins,0,5.);
  histosTH1F["hm2rec5OS_ttbb_trkP"] = new TH1F("hm2rec5OS_ttbb_trkP","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec5OS_ttbb_trkM"] = new TH1F("hm2rec5OS_ttbb_trkM","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}<0",massbins,0,5.);
```

```
  histosTH1F["hm2rec5OS_diag_pypxP"] = new TH1F("hm2rec5OS_diag_pypxP","M_{4#pi} TB/BT OS, |py/px
|_{4#pi} > 1",massbins,0,5.);
  histosTH1F["hm2rec5OS_diag_pypxM"] = new TH1F("hm2rec5OS_diag_pypxM","M_{4#pi} TB/BT OS, |py/px
|_{4#pi} < 1",massbins,0,5.);
  histosTH1F["hm2rec5OS_ttbb_pypxP"] = new TH1F("hm2rec5OS_ttbb_pypxP","M_{4#pi} TT/BB OS, |py/px
|_{4#pi} > 1",massbins,0,5.);
  histosTH1F["hm2rec5OS_ttbb_pypxM"] = new TH1F("hm2rec5OS_ttbb_pypxM","M_{4#pi} TT/BB OS, |py/px
|_{4#pi} < 1",massbins,0,5.);

  histosTH1F["hm2rec6OS"] = new TH1F("hm2rec6OS","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec6SS"] = new TH1F("hm2rec6SS","M_{4#pi} SS",massbins,0,5.);
  histosTH1F["hm2rec6OS_diag"] = new TH1F("hm2rec6OS_diag","M_{4#pi} TB/BT OS",massbins,0,5.);
  histosTH1F["hm2rec6SS_diag"] = new TH1F("hm2rec6SS_diag","M_{4#pi} TB/BT SS",massbins,0,5.);
  histosTH1F["hm2rec6OS_ttbb"] = new TH1F("hm2rec6OS_ttbb","M_{4#pi} TT/BB OS",massbins,0,5.);
  histosTH1F["hm2rec6SS_ttbb"] = new TH1F("hm2rec6SS_ttbb","M_{4#pi} TT/BB SS",massbins,0,5.);

  //histosTH1F["hm2rec6OS_diag_trkP"] = new TH1F("hm2rec6OS_diag_trkP","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec6OS_diag_trkM"] = new TH1F("hm2rec6OS_diag_trkM","M_{4#pi} TB/BT OS, py_{#p
i1}py_{#pi2}<0",massbins,0,5.);
  //histosTH1F["hm2rec6OS_ttbb_trkP"] = new TH1F("hm2rec6OS_ttbb_trkP","M_{4#pi} TT/BB OS, py_{#p
i1}py_{#pi2}>0",massbins,0,5.);
  //histosTH1F["hm2rec6OS_ttbb_trkM"] = new TH1F("hm2rec6OS_ttbb_trkM","M_{4#pi} TT/BB OS, py_{#p
i1}py_{#pi2}<0",massbins,0,5.);
  //...Luiz
  histosTH1F["hm2rec6OS_diag_trkP"] = new TH1F("hm2rec6OS_diag_trkP","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec6OS_diag_trkM"] = new TH1F("hm2rec6OS_diag_trkM","M_{4#pi} TB/BT OS, py_{#pi_
{1}}py_{#pi_{2}}<0",massbins,0,5.);
  histosTH1F["hm2rec6OS_ttbb_trkP"] = new TH1F("hm2rec6OS_ttbb_trkP","M_{4#pi} TT/BB OS, py_{#pi_
{1}}py_{#pi_{2}}>0",massbins,0,5.);
  histosTH1F["hm2rec6OS_ttbb_trkM"] = new TH1F("hm2rec6OS_ttbb_trkM","M_{4#pi} TT/BB OS, py_{#pi_
{1}}py_{#pi_{2}}<0",massbins,0,5.);

  histosTH1F["hm2recHFvetoOS"] = new TH1F("hm2recHFvetoOS","M_{4#pi} HFv OS",massbins,0,5.);
  histosTH1F["hm2recHFvetoSS"] = new TH1F("hm2recHFvetoSS","M_{4#pi} HFv SS",massbins,0,5.);

  histosTH1F["hm2rec45OS"] = new TH1F("hm2rec45OS","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec45SS"] = new TH1F("hm2rec45SS","M_{4#pi} SS",massbins,0,5.);
  histosTH1F["hm2rec4515OS"] = new TH1F("hm2rec4515OS","M_{4#pi} OS",massbins,0,5.);
  histosTH1F["hm2rec4515SS"] = new TH1F("hm2rec4515SS","M_{4#pi} SS",massbins,0,5.);

  histosTH1F["hm2rec9919"] = new TH1F("hm2rec9919","M_{4#pi} 9919",massbins,0,5.);
  histosTH1F["hm2rec9922"] = new TH1F("hm2rec9922","M_{4#pi} 9919,9922",massbins,0,5.);
  histosTH1F["hm2rec9971"] = new TH1F("hm2rec9971","M_{4#pi} 9971",massbins,0,5.);
  histosTH1F["hm2rec9978"] = new TH1F("hm2rec9978","M_{4#pi} 9978",massbins,0,5.);

  histosTH1F["hnclusters"] = new TH1F("hnclusters","nPixelClusters",500,0,500.);
  histosTH1F["hnclusters2"] = new TH1F("hnclusters2","nStripClusters",500,0,500.);
  histosTH1F["hnclustersOSdiag"] = new TH1F("hnclustersOSdiag","nPixelClusters",500,0,500.);
  histosTH1F["hnclusters2OSdiag"] = new TH1F("hnclusters2OSdiag","nStripClusters",500,0,500.);

  histosTH1F["halgo"] = new TH1F("halgo","Algo",15,0,15.);
  histosTH1F["hnhits"] = new TH1F("hnhits","nhits pix+strip",40,0,40.);
  histosTH1F["hchi2"] = new TH1F("hchi2","normalized #chi^{2}",1050,-50,1000.);
  //histosTH1F["hdz"] = new TH1F("hdz","dz",1000,-200,200.);
  //histosTH1F["hd0"] = new TH1F("hd0","d0",2000,-200,200.);
  //...Luiz
  histosTH1F["hdz"] = new TH1F("hdz","dz",2000,-100,100.);
  histosTH1F["hd0"] = new TH1F("hd0","d0",2000,-100,100.);

  histosTH1F["halgov"] = new TH1F("halgov","Algo",15,0,15.);
  histosTH1F["hnhitsv"] = new TH1F("hnhitsv","nhits pixel",40,0,40.);
  histosTH1F["hchi2v"] = new TH1F("hchi2v","normalized #chi^{2} vtx-fitted",550,-50,500.);
  //histosTH1F["hdzv"] = new TH1F("hdzv","dz vtx-fitted",500,-100,100.);
  //...Luiz
  histosTH1F["hdzv"] = new TH1F("hdzv","dz vtx-fitted",1000,-100,100.);
  histosTH1F["hd0v"] = new TH1F("hd0v","d0 vtx-fitted",2000,-20,20.);

  histosTH1F["hchi2fin"] = new TH1F("hchi2fin","normalized #chi^{2} vtx-fitted",550,-50,500.);
  //histosTH1F["hdzfin"] = new TH1F("hdzfin","dz vtx-fitted",500,-100,100.);
  //...Luiz
  histosTH1F["hdzfin"] = new TH1F("hdzfin","dz vtx-fitted",1000,-100,100.);
  histosTH1F["hd0fin"] = new TH1F("hd0fin","d0 vtx-fitted",2000,-20,20.);

  //histosTH1F["hdeltaR"] = new TH1F("hdeltaR","#Delta R trk-trk",200,0,10.);
  //histosTH1F["hdeltaR2"] = new TH1F("hdeltaR2","#Delta R trk-trk",200,0,10.);
  //...Luiz
  histosTH1F["hdeltaR"] = new TH1F("hdeltaR","#DeltaR trk-trk",200,0,10.);
  histosTH1F["hdeltaR2"] = new TH1F("hdeltaR2","#DeltaR trk-trk",200,0,10.);

  //------------------
  histosTH2F["h2dimdpyAll"] = new TH2F("h2dimdpyAll", "p_{y}^{TOTEM} vs p_{y}^{CMS}",200,-2.,2.,2
00,-2., 2.);
  histosTH2F["h2dimdpy"] = new TH2F("h2dimdpy","p_{y}^{TOTEM} vs p_{y}^{CMS}",200,-2.,2.,200,-2.,
2.);
  histosTH2F["h2dimdpy_diag"] = new TH2F("h2dimdpy_diag","p_{y}^{TOTEM} vs p_{y}^{CMS} diag",100,
-2.,2.,100,-2.,2.);
  histosTH2F["h2dimdpy_ttbb"] = new TH2F("h2dimdpy_ttbb","p_{y}^{TOTEM} vs p_{y}^{CMS} TT/BB",100
,-2.,2.,100,-2.,2.);

  //histosTH1F["hdpyAll"] = new TH1F("hdpyAll"  ,"#Delta p_{Y} CMS-TOTEM",500,-0.5,0.5);
  //histosTH1F["hdpy"] = new TH1F("hdpy"     ,"#Delta p_{Y} CMS-TOTEM",500,-0.5,0.5);
  //histosTH1F["hdpy_diag"] = new TH1F("hdpy_diag","#Delta p_{Y} CMS-TOTEM TB/BT",500,-0.5,0.5);
  //histosTH1F["hdpy_ttbb"] = new TH1F("hdpy_ttbb","#Delta p_{Y} CMS-TOTEM TT/BB",500,-0.5,0.5);
  //...Luiz
  histosTH1F["hdpyAll"] = new TH1F("hdpyAll"  ,"#Deltap_{Y} CMS-TOTEM",500,-0.5,0.5);
  histosTH1F["hdpy"] = new TH1F("hdpy"     ,"#Deltap_{Y} CMS-TOTEM",500,-0.5,0.5);
  histosTH1F["hdpy_diag"] = new TH1F("hdpy_diag","#Deltap_{Y} CMS-TOTEM TB/BT",500,-0.5,0.5);
  histosTH1F["hdpy_ttbb"] = new TH1F("hdpy_ttbb","#Deltap_{Y} CMS-TOTEM TT/BB",500,-0.5,0.5);

  histosTH2F["h2dimdpxAll"] = new TH2F("h2dimdpxAll", "p_{x}^{TOTEM} vs p_{x}^{CMS}",200,-2.,2.,2
00,-2., 2.);
  histosTH2F["h2dimdpx"] = new TH2F("h2dimdpx","p_{x}^{TOTEM} vs p_{x}^{CMS}",200,-2.,2.,200,-2.,
2.);
  histosTH2F["h2dimdpx_diag"] = new TH2F("h2dimdpx_diag","p_{x}^{TOTEM} vs p_{x}^{CMS} diag",100,
-2.,2.,100,-2.,2.);
  histosTH2F["h2dimdpx_ttbb"] = new TH2F("h2dimdpx_ttbb","p_{x}^{TOTEM} vs p_{x}^{CMS} TT/BB",100
,-2.,2.,100,-2.,2.);

  //histosTH1F["hdpxAll"] = new TH1F("hdpxAll", "#Delta p_{X} CMS-TOTEM",500,-0.5,0.5);
  //histosTH1F["hdpx"] = new TH1F("hdpx", "#Delta p_{X} CMS-TOTEM",500,-0.5,0.5);
  //histosTH1F["hdpx_diag"] = new TH1F("hdpx_diag", "#Delta p_{X} CMS-TOTEM TB/BT",500,-0.5,0.5);
  //histosTH1F["hdpx_ttbb"] = new TH1F("hdpx_ttbb", "#Delta p_{X} CMS-TOTEM TT/BB",500,-0.5,0.5);
  //...Luiz
  histosTH1F["hdpxAll"] = new TH1F("hdpxAll", "#Deltap_{X} CMS-TOTEM",500,-0.5,0.5);
  histosTH1F["hdpx"] = new TH1F("hdpx", "#Deltap_{X} CMS-TOTEM",500,-0.5,0.5);
  histosTH1F["hdpx_diag"] = new TH1F("hdpx_diag", "#Deltap_{X} CMS-TOTEM TB/BT",500,-0.5,0.5);
  histosTH1F["hdpx_ttbb"] = new TH1F("hdpx_ttbb", "#Deltap_{X} CMS-TOTEM TT/BB",500,-0.5,0.5);

  //-------------------
  histosTH2F["h2dimxVtxRL"] = new TH2F("h2dimxVtxRL","xVtxL vs xVtxR (m)",1000,-0.004,0.001,1000,
-0.004,0.001);
  histosTH2F["h2dimxVtxcmsR"] = new TH2F("h2dimxVtxcmsR","xVtxCMS vs xVtxR (cm)",300,-0.3,0.3,400
,-0.3,0.5);
  histosTH2F["h2dimxVtxcmsL"] = new TH2F("h2dimxVtxcmsL","xVtxCMS vs xVtxL (cm)",300,-0.3,0.3,400
,-0.3,0.5);
  histosTH2F["h2dimxVtxcmsRL"] = new TH2F("h2dimxVtxcmsRL","xVtxCMS vs xVtxRL (cm)",300,-0.3,0.3,
400,-0.3,0.5);

  histosTH2F["h2dimxVtxcmsR2"] = new TH2F("h2dimxVtxcmsR2","xVtxCMS vs xVtxR (cm) (|xVtxL-xVtxR|<
3e-5)",300,-0.3,0.3,400,-0.3,0.5);
  histosTH2F["h2dimxVtxcmsL2"] = new TH2F("h2dimxVtxcmsL2","xVtxCMS vs xVtxL (cm) (|xVtxL-xVtxR|<
3e-5)",300,-0.3,0.3,400,-0.3,0.5);
  histosTH2F["h2dimxVtxcmsRL2"] = new TH2F("h2dimxVtxcmsRL2","xVtxCMS vs xVtxRL (cm)",300,-0.3,0.
3,400,-0.3,0.5);

  histosTH2F["h2dimxVtx_zVtx_CT"] = new TH2F("h2dimxVtx_zVtx_CT","xVtxCMS-xVtxTOTEM vs zVtx (cm)"
,300,-20.,20.,400,-0.3,0.5);
  histosTH2F["h2dimxVtx_zVtx_C"] = new TH2F("h2dimxVtx_zVtx_C","xVtxCMS vs zVtx (cm)",300,-20.,20
.,400,-0.3,0.5);
  histosTH2F["h2dimxVtx_zVtx_T"] = new TH2F("h2dimxVtx_zVtx_T","xVtxTOTEM vs zVtx (cm)",300,-20.,
20.,400,-0.3,0.5);
```

```
  histosTH1F["hxVtxRL"] = new TH1F("hxVtxRL","xVtxR-xVtxL (m)",300,-0.0003,0.0003);
  //histosTH1F["hxVtxcmsR"] = new TH1F("hxVtxcmsR","xVtxCMS-xVtxR (cm)",300,-0.5,0.5);
  //histosTH1F["hxVtxcmsL"] = new TH1F("hxVtxcmsL","xVtxCMS-xVtxL (cm)",300,-0.5,0.5);
  //histosTH1F["hxVtxcmsRL"] = new TH1F("hxVtxcmsRL","xVtxCMS-xVtxTOTEM (cm)",300,-0.5,0.5);
  //...Luiz
  histosTH1F["hxVtxcmsR"] = new TH1F("hxVtxcmsR","xVtxCMS-xVtxR (cm)",500,-0.5,0.5);
  histosTH1F["hxVtxcmsL"] = new TH1F("hxVtxcmsL","xVtxCMS-xVtxL (cm)",500,-0.5,0.5);
  histosTH1F["hxVtxcmsRL"] = new TH1F("hxVtxcmsRL","xVtxCMS-xVtxTOTEM (cm)",500,-0.5,0.5);

  histosTH1F["hxVtxRL_diag"] = new TH1F("hxVtxRL_diag","xVtxR-xVtxL (m)",300,-0.0003,0.0003);
  //histosTH1F["hxVtxcmsR_diag"] = new TH1F("hxVtxcmsR_diag","xVtxCMS-xVtxR (cm)",300,-0.5,0.5);
  //histosTH1F["hxVtxcmsL_diag"] = new TH1F("hxVtxcmsL_diag","xVtxCMS-xVtxL (cm)",300,-0.5,0.5);
  //histosTH1F["hxVtxcmsRL_diag"] = new TH1F("hxVtxcmsRL_diag","xVtxCMS-xVtxTOTEM (cm)",300,-0.5,
0.5);
  //...Luiz
  histosTH1F["hxVtxcmsR_diag"] = new TH1F("hxVtxcmsR_diag","xVtxCMS-xVtxR (cm)",500,-0.5,0.5);
  histosTH1F["hxVtxcmsL_diag"] = new TH1F("hxVtxcmsL_diag","xVtxCMS-xVtxL (cm)",500,-0.5,0.5);
  histosTH1F["hxVtxcmsRL_diag"] = new TH1F("hxVtxcmsRL_diag","xVtxCMS-xVtxTOTEM (cm)",500,-0.5,0.
5);

  histosTH1F["hxVtxRL_ttbb"] = new TH1F("hxVtxRL_ttbb","xVtxR-xVtxL (m)",300,-0.0003,0.0003);
  //histosTH1F["hxVtxcmsR_ttbb"] = new TH1F("hxVtxcmsR_ttbb","xVtxCMS-xVtxR (cm)",300,-0.5,0.5);
  //histosTH1F["hxVtxcmsL_ttbb"] = new TH1F("hxVtxcmsL_ttbb","xVtxCMS-xVtxL (cm)",300,-0.5,0.5);
  //histosTH1F["hxVtxcmsRL_ttbb"] = new TH1F("hxVtxcmsRL_ttbb","xVtxCMS-xVtxTOTEM (cm)",300,-0.5,
0.5);
  //...Luiz
  histosTH1F["hxVtxcmsR_ttbb"] = new TH1F("hxVtxcmsR_ttbb","xVtxCMS-xVtxR (cm)",500,-0.5,0.5);
  histosTH1F["hxVtxcmsL_ttbb"] = new TH1F("hxVtxcmsL_ttbb","xVtxCMS-xVtxL (cm)",500,-0.5,0.5);
  histosTH1F["hxVtxcmsRL_ttbb"] = new TH1F("hxVtxcmsRL_ttbb","xVtxCMS-xVtxTOTEM (cm)",500,-0.5,0.
5);

  //  histosTH2F["hdedx"] = new TH2F("hdedx","dE/dx vs p", 300, 0.,5.,500, 0.,100.);
  //histosTH2F["hdedx"] = new TH2F("hdedx","dE/dx vs p", 300, 0.,5.,1000, 0.,200.);
  //...Luiz
  //histosTH2F["hdedx"] = new TH2F("hdedx","dE/dx vs p", 500, 0.,5.,1000, 0.,200.);
  histosTH2F["hdedx"] = new TH2F("hdedx","dE/dx vs p", 1000, 0.,20.,1000, 0.,200.);
  histosTH2F["hdedxvee11"] = new TH2F("hdedxvee11","dE/dx vs p type:11", 1000, 0.,20.,1000, 0.,20
0.);
  histosTH2F["hdedxvee02"] = new TH2F("hdedxvee02","dE/dx vs p type:02", 1000, 0.,20.,1000, 0.,20
0.);
  histosTH2F["hdedxvee01"] = new TH2F("hdedxvee01","dE/dx vs p type:01", 1000, 0.,20.,1000, 0.,20
0.);
  //histosTH2F["hdedxvee02"] = new TH2F("hdedxvee02","dE/dx vs p type:02", 500, 0.,5.,1000, 0.,20
0.);
  //histosTH2F["hdedxvee01"] = new TH2F("hdedxvee01","dE/dx vs p type:02", 500, 0.,5.,1000, 0.,20
0.);
  //...Luiz
  histosTH2F["hlndedx"]  = new TH2F("hlndedx","ln dE/dx vs p", 500, 0.,5.,1000, 0.,5.);
  histosTH2F["hl10dedx"] = new TH2F("hl10dedx","log10 dE/dx vs p", 500, 0.,5.,1000, 0.,5.);

  //-----------------------------------------------------

  for(map<string,TH1F*>::const_iterator it = histosTH1F.begin(); it != histosTH1F.end(); ++it)
      it->second->Sumw2();
  for(map<string,TH2F*>::const_iterator it = histosTH2F.begin(); it != histosTH2F.end(); ++it)
      it->second->Sumw2();
  //for(map<string,TH3F*>::const_iterator it = histosTH3F.begin(); it != histosTH3F.end(); ++it)
  //  it->second->Sumw2();
  //===================

  //vector<TString>* vfiles = new vector<TString>(1,"merged_reduced_8372_198903_LP_Jets1_1_test_v
1.root");
  vector<TString>* vfiles = new vector<TString>;
  for(size_t idx_file = 0; idx_file < fileNames.size(); ++idx_file) vfiles->push_back( fileNames[
idx_file] );

  // Declaration of tree and its branches variables
  TTree* tree = NULL;
  MyEvtId*           evtId        = NULL;
  vector<MyCaloTower>* calo_coll = NULL;
  vector<MyTracks>*   track_coll = NULL;
```

```
vector<MyVertex>*  vertex_coll = NULL;
vector<MyKshorts>* kshort_coll = NULL;
vector<MyLambdas>* lambda_coll = NULL;
vector<MySiPixelCluster>* sipixelcluster_coll = NULL;
MySiStripCluster*         sistripcluster_coll = NULL;


RPRootDumpReconstructedProton* rec_proton_left  = NULL;
RPRootDumpReconstructedProton* rec_proton_right = NULL;
map<unsigned int, RPRootDumpTrackInfo*> rp_track_info;
//  TriggerData   *trigData       = NULL;

//===================

std::map< int, TMatrix> AlltransportMatrixPlus;
std::map< int, TMatrix> AlltransportMatrixMinus;

//XRPV.B6R5.B1      B1 is Right , CMS minus
TMatrix M220M(6,6);

M220M(0,0)=-1.871247999249703e+00  ;
M220M(0,1)=1.733151135160244e-02  ;
M220M(0,2)=0.000000000000000e+00  ;
M220M(0,3)=0.000000000000000e+00  ;
M220M(0,4)=-3.821064474332431e-02 ;
M220M(0,5)=-3.821064474332431e-02  ;
M220M(1,0)=5.528023408827136e-02 ;
M220M(1,1)=-5.349147148886547e-01  ;
M220M(1,2)=0.000000000000000e+00  ;
M220M(1,3)=0.000000000000000e+00  ;
M220M(1,4)=2.332546482011731e-03  ;
M220M(1,5)=2.332546482011731e-03  ;
M220M(2,0)=0.000000000000000e+00  ;
M220M(2,1)=0.000000000000000e+00 ;
M220M(2,2)=-2.321378009782771e-08  ;
M220M(2,3)=2.629525462245173e+02  ;
M220M(2,4)=0.000000000000000e+00  ;
M220M(2,5)=0.000000000000000e+00  ;
M220M(3,0)=0.000000000000000e+00  ;
M220M(3,1)=0.000000000000000e+00 ;
M220M(3,2)=-3.802967965874805e-03  ;
M220M(3,3)=4.731545364353734e+00  ;
M220M(3,4)=0.000000000000000e+00  ;
M220M(3,5)=0.000000000000000e+00  ;
M220M(5,0)=2.252479551546639e-03  ;
M220M(5,1)=2.039900958275588e-02  ;
M220M(5,2)=0.000000000000000e+00  ;
M220M(5,3)=0.000000000000000e+00  ;
M220M(5,4)=1.000000000000000e+00  ;
M220M(5,5)=9.584144208086515e-05  ;
M220M(5,0)=0.000000000000000e+00  ;
M220M(5,1)=0.000000000000000e+00  ;
M220M(5,2)=0.000000000000000e+00  ;
M220M(5,3)=0.000000000000000e+00  ;
M220M(5,4)=0.000000000000000e+00  ;
M220M(5,5)=1.000000000000000e+00  ;

//XRPV.B6L5.B2      B2 is Left, CMS plus
TMatrix M220P(6,6);
M220P(0,0)= -1.897523818078534e+00  ;
M220P(0,1)=1.062411421653394e-01;//1.062411421653394e-01  ;
M220P(0,2)=0.000000000000000e+00   ;
M220P(0,3)=0.000000000000000e+00   ;
M220P(0,4)=5.198622934357949e-02   ; //to cross check
M220P(0,5)=5.198622934357949e-02   ;
M220P(1,0)=5.401504221523073e-02   ;
M220P(1,1)=-5.300268751290215e-01  ;
M220P(1,2)=0.000000000000000e+00    ;
M220P(1,3)=0.000000000000000e+00    ;
M220P(1,4)= -2.668640114664157e-03  ;
M220P(1,5)=-2.668640114664157e-03   ;
M220P(2,0)=0.000000000000000e+00    ;
```

```
M220P(2,1)=0.000000000000000e+00   ;
M220P(2,2)=-3.186327537105585e-09    ;
M220P(2,3)=2.618610731959413e+02   ;
M220P(2,4)=0.000000000000000e+00  ;
M220P(2,5)=0.000000000000000e+00   ;
M220P(3,0)=0.000000000000000e+00   ;
M220P(3,1)=0.000000000000000e+00   ;
M220P(3,2)=-3.818818897730703e-03   ;
M220P(3,3)=4.676450995853369e+00    ;
M220P(3,4)=0.000000000000000e+00    ;
M220P(3,5)=0.000000000000000e+00   ;
M220P(5,0)=-2.255769806850952e-03   ;
M220P(5,1)=-2.727057931490794e-02   ;
M220P(5,2)=0.000000000000000e+00    ;
M220P(5,3)=0.000000000000000e+00    ;
M220P(5,4)=1.000000000000000e+00   ;
M220P(5,5)=1.107429910138087e-04   ;
M220P(5,0)=0.000000000000000e+00   ;
M220P(5,1)=0.000000000000000e+00   ;
M220P(5,2)=0.000000000000000e+00   ;
M220P(5,3)=0.000000000000000e+00  ;
M220P(5,4)=0.000000000000000e+00  ;
M220P(5,5)=1.000000000000000e+00    ;


TMatrix M215P(6,6);

M215P(0,0)=-2.187692624858721e+00    ;
M215P(0,1)=2.953545515358119e+00    ;
M215P(0,2)=0.000000000000000e+00    ;
M215P(0,3)=0.000000000000000e+00    ;
M215P(0,4)=6.603743360296731e-02    ;
M215P(0,5)=6.603743360296731e-02    ;
M215P(1,0)=5.401504221523073e-02   ;
M215P(1,1)=-5.300268751290215e-01   ;
M215P(1,2)= 0.000000000000000e+00   ;
M215P(1,3)=0.000000000000000e+00    ;
M215P(1,4)=-2.668640114664157e-03   ;
M215P(1,5)=-2.668640114664157e-03   ;
M215P(2,0)=0.000000000000000e+00   ;
M215P(2,1)=0.000000000000000e+00   ;
M215P(2,2)=2.051469193227947e-02    ;
M215P(2,3)=2.367391784462199e+02   ;
M215P(2,4)=0.000000000000000e+00    ;
M215P(2,5)=0.000000000000000e+00    ;
M215P(3,0)=0.000000000000000e+00    ;
M215P(3,1)=0.000000000000000e+00   ;
M215P(3,2)=-3.818818897730703e-03    ;
M215P(3,3)=4.676450995853369e+00    ;
M215P(3,4)=0.000000000000000e+00    ;
M215P(3,5)=0.000000000000000e+00   ;
M215P(4,0)=-2.271149533403129e-03   ;
M215P(4,1)=-2.711966453134992e-02  ;
M215P(4,2)=0.000000000000000e+00    ;
M215P(4,3)=0.000000000000000e+00    ;
M215P(4,4)=1.000000000000000e+00   ;
M215P(4,5)= 1.113908988335683e-04   ;
M215P(5,0)= 0.000000000000000e+00   ;
M215P(5,1)= 0.000000000000000e+00    ;
M215P(5,2)=0.000000000000000e+00   ;
M215P(5,3)= 0.000000000000000e+00   ;
M215P(5,4)= 0.000000000000000e+00   ;
M215P(5,5)= 1.000000000000000e+00  ;

TMatrix M215M(6,6);

M215M(0,0)= -2.168213416771863e+00  ;
M215M(0,1)= 2.890893359733129e+00   ;
M215M(0,2)= 0.000000000000000e+00   ;
M215M(0,3)= 0.000000000000000e+00   ;
M215M(0,4)= -5.074108445998152e-02  ;
M215M(0,5)= -5.074108445998152e-02   ;
```

```
  M215M(1,0)= 5.528023408827136e-02  ;
  M215M(1,1)= -5.349147148886547e-01   ;
  M215M(1,2)= 0.000000000000000e+00   ;
  M215M(1,3)= 0.000000000000000e+00   ;
  M215M(1,4)= 2.332546482011731e-03   ;
  M215M(1,5)= 2.332546482011731e-03   ;
  M215M(2,0)= 0.000000000000000e+00   ;
  M215M(2,1)= 0.000000000000000e+00   ;
  M215M(2,2)= 2.042952069889703e-02   ;
  M215M(2,3)= 2.375346845272119e+02   ;
  M215M(2,4)= 0.000000000000000e+00   ;
  M215M(2,5)= 0.000000000000000e+00   ;
  M215M(3,0)= 0.000000000000000e+00   ;
  M215M(3,1)= 0.000000000000000e+00  ;
  M215M(3,2)= -3.802967965874805e-03  ;
  M215M(3,3)= 4.731545364353734e+00   ;
  M215M(3,4)= 0.000000000000000e+00   ;
  M215M(3,5)= 0.000000000000000e+00   ;
  M215M(4,0)= 2.252479550701315e-03   ;
  M215M(4,1)= 2.039900959093559e-02  ;
  M215M(4,2)= 0.000000000000000e+00   ;
  M215M(4,3)= 0.000000000000000e+00   ;
  M215M(4,4)= 1.000000000000000e+00   ;
  M215M(4,5)= 9.572950680001605e-05   ;
  M215M(5,0)= 0.000000000000000e+00   ;
  M215M(5,1)= 0.000000000000000e+00   ;
  M215M(5,2)= 0.000000000000000e+00   ;
  M215M(5,3)= 0.000000000000000e+00   ;
  M215M(5,4)= 0.000000000000000e+00   ;
  M215M(5,5)= 1.000000000000000e+00   ;

  //...Luiz
TMatrix M213P(6,6);

M213P(0,0)=-2.275629113585150e+00    ;
M213P(0,1)=3.816429268068490e+00    ;
M213P(0,2)=0.000000000000000e+00    ;
M213P(0,3)=0.000000000000000e+00    ;
M213P(0,4)=7.029569133459326e-02    ;
M213P(0,5)=7.029569133459326e-02    ;
M213P(1,0)=5.401504221523073e-02   ;
M213P(1,1)=-5.300268751290215e-01   ;
M213P(1,2)= 0.000000000000000e+00   ;
M213P(1,3)=0.000000000000000e+00    ;
M213P(1,4)=-2.668640114664157e-03   ;
M213P(1,5)=-2.668640114664157e-03   ;
M213P(2,0)=0.000000000000000e+00   ;
M213P(2,1)=0.000000000000000e+00   ;
M213P(2,2)=2.673172909778739e-02    ;
M213P(2,3)=2.291259162249677e+02   ;
M213P(2,4)=0.000000000000000e+00    ;
M213P(2,5)=0.000000000000000e+00    ;
M213P(3,0)=0.000000000000000e+00    ;
M213P(3,1)=0.000000000000000e+00   ;
M213P(3,2)=-3.818818897730703e-03    ;
M213P(3,3)=4.676450995853369e+00    ;
M213P(3,4)=0.000000000000000e+00    ;
M213P(3,5)=0.000000000000000e+00   ;
M213P(4,0)=-2.275810403624082e-03   ;
M213P(4,1)=-2.707392937356277e-02  ;
M213P(4,2)=0.000000000000000e+00    ;
M213P(4,3)=0.000000000000000e+00    ;
M213P(4,4)=1.000000000000000e+00   ;
M213P(4,5)= 1.115872491557145e-04   ;
M213P(5,0)= 0.000000000000000e+00   ;
M213P(5,1)= 0.000000000000000e+00    ;
M213P(5,2)=0.000000000000000e+00   ;
M213P(5,3)= 0.000000000000000e+00   ;
M213P(5,4)= 0.000000000000000e+00   ;
M213P(5,5)= 1.000000000000000e+00  ;

TMatrix M213M(6,6);
```

```
M213M(0,0)= -2.143392591666300e+00  ;
M213M(0,1)= 3.761734515572186e+00   ;
M213M(0,2)= 0.000000000000000e+00   ;
M213M(0,3)= 0.000000000000000e+00   ;
M213M(0,4)= -5.453847013733222e-02  ;
M213M(0,5)= -5.453847013733222e-02   ;
M213M(1,0)= 5.528023408827136e-02  ;
M213M(1,1)= -5.349147148886547e-01   ;
M213M(1,2)= 0.000000000000000e+00   ;
M213M(1,3)= 0.000000000000000e+00   ;
M213M(1,4)= 2.332546482011731e-03   ;
M213M(1,5)= 2.332546482011731e-03   ;
M213M(2,0)= 0.000000000000000e+00   ;
M213M(2,1)= 0.000000000000000e+00   ;
M213M(2,2)= 2.662075254734354e-02   ;
M213M(2,3)= 2.298317286740411e+02   ;
M213M(2,4)= 0.000000000000000e+00   ;
M213M(2,5)= 0.000000000000000e+00   ;
M213M(3,0)= 0.000000000000000e+00   ;
M213M(3,1)= 0.000000000000000e+00  ;
M213M(3,2)= -3.802967965874805e-03  ;
M213M(3,3)= 4.731545364353734e+00   ;
M213M(3,4)= 0.000000000000000e+00   ;
M213M(3,5)= 0.000000000000000e+00   ;
M213M(4,0)= 2.252479550701315e-03   ;
M213M(4,1)= 2.039900959093559e-02  ;
M213M(4,2)= 0.000000000000000e+00   ;
M213M(4,3)= 0.000000000000000e+00   ;
M213M(4,4)= 1.000000000000000e+00   ;
M213M(4,5)= 9.569558449226801e-05   ;
M213M(5,0)= 0.000000000000000e+00   ;
M213M(5,1)= 0.000000000000000e+00   ;
M213M(5,2)= 0.000000000000000e+00   ;
M213M(5,3)= 0.000000000000000e+00   ;
M213M(5,4)= 0.000000000000000e+00   ;
M213M(5,5)= 1.000000000000000e+00   ;


  AlltransportMatrixPlus.insert(std::make_pair(220,M220P));
  AlltransportMatrixMinus.insert(std::make_pair(220,M220M));

  AlltransportMatrixPlus.insert(std::make_pair(215,M215P));
  AlltransportMatrixMinus.insert(std::make_pair(215,M215M));
  //...Luiz
  AlltransportMatrixPlus.insert(std::make_pair(213,M213P));
  AlltransportMatrixMinus.insert(std::make_pair(213,M213M));

  //===================

  int i_tot = 0 , nevt_tot = 0;
  //starting Loop over files, stops at end of list of files or when reached nevt_max
  for(vector<TString>::iterator itfiles = vfiles->begin() ; itfiles != vfiles->end() && i_tot < n
evt_max_corr ; ++itfiles){

  cout << "Opening file " << *itfiles << endl;

    TFile* file = TFile::Open(*itfiles,"READ");
    if (!file || file->IsZombie()){
      cout<<"corrupted file - skipping "<<endl;
      continue;
    }

    // Access TTree from current file
    tree = (TTree*) file->Get( treeName.c_str() );
    int nev = int(tree->GetEntriesFast());
    nevt_tot += nev;
    //RC
    cout<< nev <<" entries in "<< *itfiles << endl;

    // Add branches to TTree ---------------------------------------------------------------------
--
```

```
    tree->SetBranchAddress("cmsEvtUA",&evtId);
    tree->SetBranchAddress("cmsCaloTowersUA",&calo_coll);
    // tracks
    //    tree->SetBranchAddress("cmsTracksUA",&track_coll);//generalTracks
    tree->SetBranchAddress("cmsTracksPIDUA", &track_coll); // refittedTracks

    tree->SetBranchAddress("cmsVerticesUA",&vertex_coll);
    //...Kshorts
    tree->SetBranchAddress("Kshort",&kshort_coll);
    tree->SetBranchAddress("Lambda",&lambda_coll);
    tree->SetBranchAddress("SiPixelClusters", &sipixelcluster_coll);
    tree->SetBranchAddress("SiStripClusters", &sistripcluster_coll);

    //    tree->SetBranchAddress("trigger_data.",&trigData);
    tree->SetBranchAddress("rec_prot_left.",&rec_proton_left);
    tree->SetBranchAddress("rec_prot_right.",&rec_proton_right);
    std::vector<unsigned int> rp_list;
    rp_list.push_back(20); rp_list.push_back(21); rp_list.push_back(24); rp_list.push_back(25);
    rp_list.push_back(120); rp_list.push_back(121); rp_list.push_back(124); rp_list.push_back(125
);
    char br_name[200];
    for (unsigned int a = 0; a < 2; ++a) {
      int s = 2;
      for (unsigned int r = 0; r < 6; r++) {
        unsigned int id = 100 * a + 10 * s + r;
        if( std::find(rp_list.begin(), rp_list.end(), id) == rp_list.end() ) continue;

        sprintf(br_name, "track_rp_%u.", id);
        //RC
        //      std::cout << br_name << std::endl;
        tree->SetBranchAddress(br_name, &rp_track_info[id]);
      }
    }

    //starting loop over events, stops when reached end of file or nevt_max
    for(int i_evt = 0; i_evt < nev && i_tot < nevt_max_corr; ++i_evt , ++i_tot){

      if( ((i_tot+1) % 5000) == 0) cout <<int(double(i_tot+1)/1000)<<"k done"<<endl;
      tree->GetEntry(i_evt);

      //-----------------------------------------------------------------------------------------
--------
      // TOTEM RP protons

      histosTH1F["EventSelection"]->Fill( "TOTEM0", wei );

      bool proton_right_valid = rec_proton_right->valid;
      bool proton_left_valid = rec_proton_left->valid;

      if(!(proton_right_valid && proton_left_valid) ) continue;

      histosTH1F["EventSelection"]->Fill( "2valid", wei );


      //--------------------------------------------
      // fiducial cut

      RPRootDumpTrackInfo* rp_020 = rp_track_info[20];
      RPRootDumpTrackInfo* rp_021 = rp_track_info[21];
      RPRootDumpTrackInfo* rp_024 = rp_track_info[24];
      RPRootDumpTrackInfo* rp_025 = rp_track_info[25];

      RPRootDumpTrackInfo* rp_120 = rp_track_info[120];
      RPRootDumpTrackInfo* rp_121 = rp_track_info[121];
      RPRootDumpTrackInfo* rp_124 = rp_track_info[124];
      RPRootDumpTrackInfo* rp_125 = rp_track_info[125];

      bool rp_valid_020 = rp_020->valid;
      bool rp_valid_021 = rp_021->valid;
      bool rp_valid_024 = rp_024->valid;
      bool rp_valid_025 = rp_025->valid;
      bool rp_valid_120 = rp_120->valid;
      bool rp_valid_121 = rp_121->valid;
      bool rp_valid_124 = rp_124->valid;
      bool rp_valid_125 = rp_125->valid;

      //------------------------------------------------
      // -z                    IP                    +z
      //          sec45                   sec56
      //top:  024       020           120       124
      //ver:     023 022                 122 123
      //bot:  025       021           121       125
      //
      //------------------------------------------------

      bool diag_top45_bot56 = rp_valid_020 && rp_valid_024 && rp_valid_121 && rp_valid_125;
      bool diag_bot45_top56 = rp_valid_021 && rp_valid_025 && rp_valid_120 && rp_valid_124;

      bool top45_top56      = rp_valid_020 && rp_valid_024 && rp_valid_120 && rp_valid_124;
      bool bot45_bot56      = rp_valid_021 && rp_valid_025 && rp_valid_121 && rp_valid_125;

      int nconf=0;
      if(diag_top45_bot56) nconf++;
      if(diag_bot45_top56) nconf++;
      if(top45_top56) nconf++;
      if(bot45_bot56) nconf++;

      //      if(diag_top45_bot56 || diag_bot45_top56 || top45_top56 || bot45_bot56);
      //      else continue;

      if(nconf==0) continue;

      histosTH1F["EventSelection"]->Fill( "anyTB/BT/TT/BB", wei );
      histosTH1F["hnconf"]->Fill(nconf, wei );

      if(nconf != 1) continue;

      histosTH1F["EventSelection"]->Fill( "exclusiveTB/BT/TT/BB", wei );

      bool fiducialCutTB=true;
      if(diag_top45_bot56){

        double x_020 = rp_020->x;
        double y_020 = rp_020->y;
        histosTH1F["rp_x_020"]->Fill( x_020, wei );
        histosTH1F["rp_y_020"]->Fill( y_020, wei );
        histosTH2F["rp_yx_020"]->Fill( x_020, y_020, wei );

        double x_024 = rp_024->x;
        double y_024 = rp_024->y;
        histosTH1F["rp_x_024"]->Fill( x_024, wei );
        histosTH1F["rp_y_024"]->Fill( y_024, wei );
        histosTH2F["rp_yx_024"]->Fill( x_024, y_024, wei );

        double x_121 = rp_121->x;
        double y_121 = rp_121->y;
        histosTH1F["rp_x_121"]->Fill( x_121, wei );
        histosTH1F["rp_y_121"]->Fill( y_121, wei );
        histosTH2F["rp_yx_121"]->Fill( x_121, y_121, wei );

        double x_125 = rp_125->x;
        double y_125 = rp_125->y;
        histosTH1F["rp_x_125"]->Fill( x_125, wei );
        histosTH1F["rp_y_125"]->Fill( y_125, wei );
        histosTH2F["rp_yx_125"]->Fill( x_125, y_125, wei );

        if( x_020<-1.5 ) fiducialCutTB=false;
        if( x_024<-1.5 ) fiducialCutTB=false;
        if( x_121<-1.5 ) fiducialCutTB=false;
        if( x_125<-1.5 ) fiducialCutTB=false;

        if( y_020< 6.0 || y_020 > 26.0) fiducialCutTB=false;
        if( y_024< 6.7 || y_024 > 28.7) fiducialCutTB=false;
```

```
    if( y_121< -25.8 || y_121 > -6.4) fiducialCutTB=false;
    if( y_125< -28.6 || y_125 > -7.1) fiducialCutTB=false;

  }

  bool fiducialCutBT=true;
  if(diag_bot45_top56){

    double x_021 = rp_021->x;
    double y_021 = rp_021->y;

    histosTH1F["rp_x_021"]->Fill( x_021, wei );
    histosTH1F["rp_y_021"]->Fill( y_021, wei );
    histosTH2F["rp_yx_021"]->Fill( x_021, y_021, wei );

    double x_025 = rp_025->x;
    double y_025 = rp_025->y;
    histosTH1F["rp_x_025"]->Fill( x_025, wei );
    histosTH1F["rp_y_025"]->Fill( y_025, wei );
    histosTH2F["rp_yx_025"]->Fill( x_025, y_025, wei );

    double x_120 = rp_120->x;
    double y_120 = rp_120->y;
    histosTH1F["rp_x_120"]->Fill( x_120, wei );
    histosTH1F["rp_y_120"]->Fill( y_120, wei );
    histosTH2F["rp_yx_120"]->Fill( x_120, y_120, wei );

    double x_124 = rp_124->x;
    double y_124 = rp_124->y;
    histosTH1F["rp_x_124"]->Fill( x_124, wei );
    histosTH1F["rp_y_124"]->Fill( y_124, wei );
    histosTH2F["rp_yx_124"]->Fill( x_124, y_124, wei );

    if(x_021<-1.5) fiducialCutBT=false;
    if(x_025<-1.5) fiducialCutBT=false;
    if(x_120<-1.5) fiducialCutBT=false;
    if(x_124<-1.5) fiducialCutBT=false;

    if( y_021< -26.3 || y_021 > -6.4) fiducialCutBT=false;
    if( y_025< -29.0 || y_025 > -7.0) fiducialCutBT=false;
    if( y_120< 7.7 || y_120 > 24.3) fiducialCutBT=false;
    if( y_124< 8.5 || y_124 > 26.8) fiducialCutBT=false;

  }

  bool fiducialCutTT=true;
  if(top45_top56){

    double x_020 = rp_020->x;
    double y_020 = rp_020->y;
    histosTH1F["rp2_x_020"]->Fill( x_020, wei );
    histosTH1F["rp2_y_020"]->Fill( y_020, wei );
    histosTH2F["rp2_yx_020"]->Fill( x_020, y_020, wei );

    double x_024 = rp_024->x;
    double y_024 = rp_024->y;
    histosTH1F["rp2_x_024"]->Fill( x_024, wei );
    histosTH1F["rp2_y_024"]->Fill( y_024, wei );
    histosTH2F["rp2_yx_024"]->Fill( x_024, y_024, wei );

    double x_120 = rp_120->x;
    double y_120 = rp_120->y;
    histosTH1F["rp2_x_120"]->Fill( x_120, wei );
    histosTH1F["rp2_y_120"]->Fill( y_120, wei );
    histosTH2F["rp2_yx_120"]->Fill( x_120, y_120, wei );

    double x_124 = rp_124->x;
    double y_124 = rp_124->y;
    histosTH1F["rp2_x_124"]->Fill( x_124, wei );
    histosTH1F["rp2_y_124"]->Fill( y_124, wei );
    histosTH2F["rp2_yx_124"]->Fill( x_124, y_124, wei );
```

```
        if(x_020<-1.5 ) fiducialCutTT=false;
        if(x_024<-1.5 ) fiducialCutTT=false;
        if(x_120<-1.5) fiducialCutTT=false;
        if(x_124<-1.5) fiducialCutTT=false;

        if( y_020< 6.0 || y_020 > 26.0) fiducialCutTT=false;
        if( y_024< 6.7 || y_024 > 28.7) fiducialCutTT=false;
        if( y_120< 7.7 || y_120 > 24.3) fiducialCutTT=false;
        if( y_124< 8.5 || y_124 > 26.8) fiducialCutTT=false;
      }

      bool fiducialCutBB=true;
      if(bot45_bot56){

        double x_021 = rp_021->x;
        double y_021 = rp_021->y;

        histosTH1F["rp2_x_021"]->Fill( x_021, wei );
        histosTH1F["rp2_y_021"]->Fill( y_021, wei );
        histosTH2F["rp2_yx_021"]->Fill( x_021, y_021, wei );

        double x_025 = rp_025->x;
        double y_025 = rp_025->y;
        histosTH1F["rp2_x_025"]->Fill( x_025, wei );
        histosTH1F["rp2_y_025"]->Fill( y_025, wei );
        histosTH2F["rp2_yx_025"]->Fill( x_025, y_025, wei );

        double x_121 = rp_121->x;
        double y_121 = rp_121->y;
        histosTH1F["rp2_x_121"]->Fill( x_121, wei );
        histosTH1F["rp2_y_121"]->Fill( y_121, wei );
        histosTH2F["rp2_yx_121"]->Fill( x_121, y_121, wei );

        double x_125 = rp_125->x;
        double y_125 = rp_125->y;
        histosTH1F["rp2_x_125"]->Fill( x_125, wei );
        histosTH1F["rp2_y_125"]->Fill( y_125, wei );
        histosTH2F["rp2_yx_125"]->Fill( x_125, y_125, wei );

        if(x_021<-1.5) fiducialCutBB=false;
        if(x_025<-1.5) fiducialCutBB=false;
        if(x_121<-1.5) fiducialCutBB=false;
        if(x_125<-1.5) fiducialCutBB=false;

        if( y_021< -26.3 || y_021 > -6.4) fiducialCutBB=false;
        if( y_025< -29.0 || y_025 > -7.0) fiducialCutBB=false;
        if( y_121< -25.8 || y_121 > -6.4) fiducialCutBB=false;
        if( y_125< -28.6 || y_125 > -7.1) fiducialCutBB=false;
      }

      int nfidu=0;
      if(diag_top45_bot56 && fiducialCutTB) nfidu++;
      if(diag_bot45_top56 && fiducialCutBT) nfidu++;
      if(     top45_top56 && fiducialCutTT) nfidu++;
      if(     bot45_bot56 && fiducialCutBB) nfidu++;

      if(nfidu==0) continue;

      histosTH1F["EventSelection"]->Fill( "fiducialXY", wei );


      //---------------------------------------------
      // here xVtxL and xVtxR, and thxL and thyR
      // elastic approximation

      double ThxR, ThyR, ThxL, ThyL, xVtxL, xVtxR;

      //bool diag_top45_bot56 = rp_valid_020 && rp_valid_024 && rp_valid_121 && rp_valid_125;
      if(diag_top45_bot56) LikeElastic_ThetaLeftThetaRight220FAR(20, 24,121, 125, rp_track_info,
rp_list,
                                                  AlltransportMatrixPlus, AlltransportMatrixMinus
,
                                                  ThxR, ThyR, ThxL, ThyL, xVtxL, xVtxR) ;

      //bool diag_bot45_top56 = rp_valid_021 && rp_valid_025 && rp_valid_120 && rp_valid_124;
      if(diag_bot45_top56) LikeElastic_ThetaLeftThetaRight220FAR(21, 25,120, 124, rp_track_info, 
rp_list,
                                                  AlltransportMatrixPlus, AlltransportMatrixMinus
,
                                                  ThxR, ThyR, ThxL, ThyL, xVtxL, xVtxR) ;

      //bool top45_top56      = rp_valid_020 && rp_valid_024 && rp_valid_120 && rp_valid_124;
      if(top45_top56) LikeElastic_ThetaLeftThetaRight220FAR(20, 24,120, 124, rp_track_info, rp_li
st,
                                                  AlltransportMatrixPlus, AlltransportMatrixMinus
,
                                                  ThxR, ThyR, ThxL, ThyL, xVtxL, xVtxR) ;

      //bool bot45_bot56      = rp_valid_021 && rp_valid_025 && rp_valid_121 && rp_valid_125;
      if(bot45_bot56) LikeElastic_ThetaLeftThetaRight220FAR(21, 25,121, 125, rp_track_info, rp_li
st,
                                                  AlltransportMatrixPlus, AlltransportMatrixMinus
,
                                                  ThxR, ThyR, ThxL, ThyL, xVtxL, xVtxR) ;


      //notElastic
      // this is average theta_x and thetha_y, both measure the same thing
      // one is negative one is positive, so minus is needed
      //      double thX=0.5*(thx_proton_left-thx_proton_right);
      //      double thY=0.5*(thy_proton_left-thy_proton_right);
      // not needed here

      // diagonal in thxL vs thxR plane, and
      // diagonal in thyL vs thyR plane

      //      histosTH1F["thyEla"]->Fill(thy_proton_left+thy_proton_right, wei);
      //      histosTH1F["thxEla"]->Fill(thx_proton_left+thx_proton_right, wei);
      histosTH1F["thyEla"]->Fill(ThyL+ThyR, wei);
      histosTH1F["thxEla"]->Fill(ThxL+ThxR, wei);

      if(diag_top45_bot56 || diag_bot45_top56){
        //      histosTH1F["thyEla_diag"]->Fill(thy_proton_left+thy_proton_right, wei);
        //      histosTH1F["thxEla_diag"]->Fill(thx_proton_left+thx_proton_right, wei);
        histosTH1F["thyEla_diag"]->Fill(ThyL+ThyR, wei);
        histosTH1F["thxEla_diag"]->Fill(ThxL+ThxR, wei);
      }else{
        //      histosTH1F["thyEla_ttbb"]->Fill(thy_proton_left+thy_proton_right, wei);
        //      histosTH1F["thxEla_ttbb"]->Fill(thx_proton_left+thx_proton_right, wei);
        histosTH1F["thyEla_ttbb"]->Fill(ThyL+ThyR, wei);
        histosTH1F["thxEla_ttbb"]->Fill(ThxL+ThxR, wei);
      }

      bool isElastic = false;
      //      if(TMath::Abs(thy_proton_left+thy_proton_right)< 8e-6 &&
      //         TMath::Abs(thx_proton_left+thx_proton_right)<30e-6) isElastic=true;

      if(TMath::Abs(ThyL+ThyR)< 8e-6 &&
         TMath::Abs(ThxL+ThxR)<30e-6) isElastic=true;

      if(isElastic) continue;

      histosTH1F["EventSelection"]->Fill( "notElastic", wei );

      //---------------------------------------------
      //xi selection

      double xi_proton_right = rec_proton_right->xi;
      double t_proton_right = rec_proton_right->t;

      double xi_proton_left = rec_proton_left->xi;
      double t_proton_left = rec_proton_left->t;

      //...Luiz
      double phi_proton_right = rec_proton_right->phi;
      double phi_proton_left = rec_proton_left->phi;
      double dphi_proton = phi_proton_right-phi_proton_left;
      //

      //-------------------------------------
      //from now on xi - positive
      xi_proton_right = -xi_proton_right;
      xi_proton_left = -xi_proton_left;

      histosTH1F["proton_right_xi"]->Fill( xi_proton_right, wei );
      histosTH1F["proton_left_xi"]->Fill( xi_proton_left, wei );

      // Mx_max=130 GeV
      //       bool proton_right_good = xi_proton_right < 0.01;
      //       bool proton_left_good = xi_proton_left < 0.01;

      // Mx_max=1300 GeV
      bool proton_right_good = xi_proton_right < 0.1;
      bool proton_left_good = xi_proton_left < 0.1;

      // Mx_max=2600 GeV, could do but didn't do
      //       bool proton_right_good = xi_proton_right < 0.2;
      //       bool proton_left_good = xi_proton_left < 0.2;

      if(proton_right_good && proton_left_good);
      else continue;

      histosTH1F["EventSelection"]->Fill( "#xi<0.1", wei );

      histosTH1F["proton_right_logXi"]->Fill( log10(xi_proton_right), wei );
      histosTH1F["proton_left_logXi"]->Fill( log10(xi_proton_left), wei );

      histosTH1F["proton_right_t"]->Fill( -t_proton_right, wei );
      histosTH1F["proton_left_t"]->Fill( -t_proton_left, wei );

      //...Luiz
      histosTH2F["phi_proton_right_t"]->Fill( -t_proton_right, phi_proton_right );
      histosTH2F["phi_proton_left_t"]->Fill( -t_proton_left, phi_proton_left );
      // delta phi between protons
      histosTH1F["dphi_proton"]->Fill( dphi_proton );

      if(diag_top45_bot56 || diag_bot45_top56){
        histosTH1F["proton_right_t_diag"]->Fill( -t_proton_right, wei );
        histosTH1F["proton_left_t_diag"]->Fill( -t_proton_left, wei );
        //...Luiz
        histosTH2F["phi_proton_right_t_diag"]->Fill( -t_proton_right, phi_proton_right );
        histosTH2F["phi_proton_left_t_diag"]->Fill( -t_proton_left, phi_proton_left );
        // delta phi between protons
        histosTH1F["dphi_proton_diag"]->Fill( dphi_proton );
        //
      }else{
        histosTH1F["proton_right_t_ttbb"]->Fill( -t_proton_right, wei );
        histosTH1F["proton_left_t_ttbb"]->Fill( -t_proton_left, wei );
        //...Luiz
        histosTH2F["phi_proton_right_t_ttbb"]->Fill( -t_proton_right, phi_proton_right );
        histosTH2F["phi_proton_left_t_ttbb"]->Fill( -t_proton_left, phi_proton_left );
        // delta phi between protons
        histosTH1F["dphi_proton_ttbb"]->Fill( dphi_proton );
        //
      }
      //...Luiz
      if(top45_top56){
        histosTH2F["phi_proton_right_t_tt"]->Fill( -t_proton_right, phi_proton_right );
        histosTH2F["phi_proton_left_t_tt"]->Fill( -t_proton_left, phi_proton_left );
      }
      //...Luiz
      if(bot45_bot56){
        histosTH2F["phi_proton_right_t_bb"]->Fill( -t_proton_right, phi_proton_right );
        histosTH2F["phi_proton_left_t_bb"]->Fill( -t_proton_left, phi_proton_left );
      }
      histosTH1F["proton_dx0"]->Fill(xVtxL-xVtxR);
      histosTH2F["proton_x0_RvsL"]->Fill(xVtxL, xVtxR);

      // HF veto
      /*
      int nHF = 0;
      for(vector<MyCaloTower>::iterator it_ct = calo_coll->begin() ; it_ct != calo_coll->end() ;
++it_ct){

        if(it_ct->hasHF){
          double eHF = it_ct->emEnergy + it_ct->hadEnergy;
          histosTH1F["eHF"]->Fill( eHF , wei );
          if(eHF>5.) nHF++;
        }
      }
      histosTH1F["nHF"]->Fill( nHF , wei );
      */


      //comment if not writing to txt file
      //      int HFveto = 0;
      //      if(nHF>0) HFveto = 1;

      //-------------------
      // After selection
      //-------------------
      //

      int run = evtId->Run;
      int evt = evtId->Evt;
      int LS = evtId->LumiSect;
      //      int runTOTEM = trigData->run_num;
      //      int evtTOTEM = trigData->event_num;

      // double mx_TOTEM=13000.*TMath::Sqrt(*xi_proton_left*xi_proton_right);
      // MX_max=13000.*xi_max;
      // xi<0.01 -> m< 130
      // xi<0.1  -> m<1300
      // not only to 0.1 or 0.01 but with the vertex cut it should be limited to ~10e-4
      // 10e-4 = 1e-3
      // xi<0.001 -> m<13
      //      double pyTOTEM= 6500.*(thy_proton_left+thy_proton_right);

      //...Luiz
      double TOTEMpy1= 6500.*(ThyL);
      double TOTEMpy2= 6500.*(ThyR);
      //
      double TOTEMpy= 6500.*(ThyL+ThyR);
      //For p_x it is more delicate and at the moment we do it for low-xi protons only (|xi| < 3.
 * 0.006). We first reconstruct th_x as for
      //elastic scattering (see attachment) and then do the sum as above:  3*0.006 = 0.018
      //...Luiz
      double TOTEMpx1=-6500*(ThxL);
      double TOTEMpx2=-6500*(ThxR);
      //
      double TOTEMpx=-6500*(ThxL+ThxR);

      //...Luiz
      double TOTEMpt1= TMath::Sqrt(pow(TOTEMpx1,2)+pow(TOTEMpy1,2));
      double TOTEMpt2= TMath::Sqrt(pow(TOTEMpx2,2)+pow(TOTEMpy2,2));

      double TOTEMphiL = TMath::ATan2(ThyL,ThxL);
      double TOTEMphiR = TMath::ATan2(ThyR,ThxR);

      double TOTEMdphi = TOTEMphiL-TOTEMphiR;

      if(TOTEMdphi<0) TOTEMdphi = TOTEMdphi + 2*TMath::Pi();           // from (-2pi,2pi) to (0,2
pi)
      if(TOTEMdphi>TMath::Pi()) TOTEMdphi = 2*TMath::Pi() - TOTEMdphi; // from (0,2pi) to (0,pi)

      //
      histosTH1F["totem_py"]->Fill(TOTEMpy, wei);
      histosTH1F["totem_px"]->Fill(TOTEMpx, wei);

      //...Luiz
      histosTH1F["totem_pxx"]->Fill(TOTEMpx, wei);
      histosTH1F["totem_pyy"]->Fill(TOTEMpy, wei);

      int tb=0;
      if(diag_top45_bot56) tb=1;
      if(diag_bot45_top56) tb=2;
      if(top45_top56) tb=3;
      if(bot45_bot56) tb=4;
      int Topol = tb;

      histosTH1F["hLS"]->Fill(LS, wei);
      //      histosTH1F["htopo"]->Fill(Topol, wei);

      bool diag=false;
      if(Topol==1 || Topol==2) diag = true;

      //     if(diag){
      //       histosTH1F["hthyEla2_diag"]->Fill(ThyL+ThyR, wei);
      //       histosTH1F["hthxEla2_diag"]->Fill(ThxL+ThxR, wei);
      //     }else{
      //       histosTH1F["hthyEla2_ttbb"]->Fill(ThyL+ThyR, wei);
      //       histosTH1F["hthxEla2_ttbb"]->Fill(ThxL+ThxR, wei);
      //     }

// old
//      fout<<run<<" "<<ls<<" "<<evt<<" "<<tb<<" "<<xi_proton_left<<" "<<xi_proton_right<<" "<<px
TOTEM<<" "<<pyTOTEM<<" "<<xVtxL<<" "<<xVtxR<<" "<<HFveto<<endl;
// latest
//      fout<<run<<" "<<ls<<" "<<evt<<" "<<tb<<" "<<xi_proton_left<<" "<<xi_proton_right<<" "<<Th
xL<<" "<<ThxR<<" "<<ThyL<<" "<<ThyR<<" "<<xVtxL<<" "<<xVtxR<<" "<<HFveto<<endl;


     //-----------------------------------
     ///////////////////////////////////////////////////////////////////////////////
       ///////////////////////////////////////////////////////////////////////////////
       ///////////////////////////////////////////////////////////////////////////////

      //...Luiz
      //double m_pi=0.13957; new PDG
       double m_pi=0.13957061;
       //double m_k =0.493667; new PDG
       double m_k =0.493677;
       //double m_mu = 0.1056583715; new PDG
       double m_mu = 0.1056583745;
       double m_e = 0.0005109989461;

       //...Luiz
       double m_p = 0.9382720813;

       //-------------------------------
       //accept only 9919, 9922
       //   if(run==259237 && ( (LS>=78 && LS <=100) || (LS>=432 && LS <=576) ) );
       //   else continue;
       //-------------------------------
       // remove 9920
       //   if(run==259237 && (LS>=103 && LS <=423)) continue;
       // remove 9940,9950
       //   if(run==259352 && (LS>=6 && LS <=153)) continue;
       //   if(run==259352 && (LS>=248 && LS <=283)) continue;
       // remove 9976
       //   if(run==259388 && (LS>=59 && LS <=360)) continue;
       // remove 9985
       //   if(run==259399 && (LS>=362 && LS <=387)) continue;
       // remove 9998
       //   if(run==259431 && (LS>=43 && LS <=354)) continue;
       //-------------------------------------------------

       double xiL = xi_proton_left;
```

```
//...Luiz
//       double xiR = xi_proton_left;
double xiR = xi_proton_right;
//   int Topol  = totemTopol[itotem];
//   double ThyL    = totemThyL[itotem];
//   double ThyR    = totemThyR[itotem];
//   double ThxL    = totemThxL[itotem];
//   double ThxR    = totemThxR[itotem];
//   double xVtxL   = totemxVtxL[itotem];
//   double xVtxR   = totemxVtxR[itotem];
//   double TOTEMpx =-6500.*(ThxL+ThxR);
//   double TOTEMpy = 6500.*(ThyL+ThyR);
//   bool HFveto = true; // no activity in HF
//   if(totemHFveto[itotem]>0) HFveto = false;

//Topol
//1 - TB, 2 - BT
//3 - TT, 4 - BB

//       bool diag=false;
//       if(Topol==1 || Topol==2) diag = true;

//   htopo->Fill(Topol);

//---------------------------------------------------
//   hthyEla->Fill(ThyL+ThyR);
//   hthxEla->Fill(ThxL+ThxR);
if(diag){
  histosTH1F["hthyEla_diag"]->Fill(ThyL+ThyR, wei);
  histosTH1F["hthxEla_diag"]->Fill(ThxL+ThxR, wei);
}else{
  histosTH1F["hthyEla_ttbb"]->Fill(ThyL+ThyR, wei);
  histosTH1F["hthxEla_ttbb"]->Fill(ThxL+ThxR, wei);
}

//---------------------------------------------------
// tighter elastic rejection

bool isElastic2 = false;

//      if(TMath::Abs(ThyL+ThyR)< 8e-6 &&
//        TMath::Abs(ThxL+ThxR)<30e-6) isElastic=true;

if(TMath::Abs(ThyL+ThyR)< 15e-6 &&
   TMath::Abs(ThxL+ThxR)<45e-6) isElastic2=true;

if(isElastic2) continue;

//---------------------------------------------------
if(diag){
  histosTH1F["hthyEla2_diag"]->Fill(ThyL+ThyR, wei);
  histosTH1F["hthxEla2_diag"]->Fill(ThxL+ThxR, wei);
}else{
  histosTH1F["hthyEla2_ttbb"]->Fill(ThyL+ThyR, wei);
  histosTH1F["hthxEla2_ttbb"]->Fill(ThxL+ThxR, wei);
}

//---------------------------------------------------
// after new anti-elastic cut
histosTH1F["htopo"]->Fill(Topol, wei);

//---------------------------------------------------

bool fiducialRegion = false;
bool fiducialRegionK = false;
double etaCut= 2.5;
bool fiducialRegionPt = false;
bool fiducialRegionPtK = false;
//double ptCut= 0.2;
//...Luiz
//double ptCut= 0.1;
double ptCut= 0.0;
```

```
      //tracks in 4track-events (npixelhits>0)
      //...Luiz
      TLorentzVector pi1(0.,0.,0.,0.);
      TLorentzVector pi2(0.,0.,0.,0.);
      TLorentzVector pi3(0.,0.,0.,0.);
      TLorentzVector pi4(0.,0.,0.,0.);
      //
      TLorentzVector k1(0.,0.,0.,0.);
      TLorentzVector k2(0.,0.,0.,0.);
      TLorentzVector k3(0.,0.,0.,0.);
      TLorentzVector k4(0.,0.,0.,0.);
      //
      TLorentzVector piA(0.,0.,0.,0.);
      TLorentzVector piB(0.,0.,0.,0.);
      TLorentzVector piC(0.,0.,0.,0.);
      TLorentzVector piD(0.,0.,0.,0.);
      //
      TLorentzVector kA(0.,0.,0.,0.);
      TLorentzVector kB(0.,0.,0.,0.);
      TLorentzVector kC(0.,0.,0.,0.);
      TLorentzVector kD(0.,0.,0.,0.);

      //...Luiz
      //TLorentzVector pipiRec(0.,0.,0.,0.);
      TLorentzVector pi1pi2Rec(0.,0.,0.,0.);
      TLorentzVector pi3pi4Rec(0.,0.,0.,0.);
      TLorentzVector pi1pi3Rec(0.,0.,0.,0.);
      TLorentzVector pi2pi4Rec(0.,0.,0.,0.);
      //
      TLorentzVector k1k2Rec(0.,0.,0.,0.);
      TLorentzVector k3k4Rec(0.,0.,0.,0.);
      TLorentzVector k1k3Rec(0.,0.,0.,0.);
      TLorentzVector k2k4Rec(0.,0.,0.,0.);
      //
      //...for completeness
      TLorentzVector pi1pi4Rec(0.,0.,0.,0.);
      TLorentzVector pi2pi3Rec(0.,0.,0.,0.);
      //...for completeness
      TLorentzVector k1k4Rec(0.,0.,0.,0.);
      TLorentzVector k2k3Rec(0.,0.,0.,0.);

    //...combining pions and kaons for the event selection type = 11 (one primary & one Vee)
          /*
    ...first combining, then select the Q_pairs=0

    pi1pi2 pi3k4
    pi1pi3 pi2k4
    pi2pi3 pi1k4

    pi1pi2 k3pi4
    pi1pi4 k3pi2
    pi2pi4 k3pi1

    pi1k2 pi3pi4
    pi3k2 pi1pi4
    pi4k2 pi1pi3

    k1pi2 pi3pi4
    k1pi3 pi2pi4
    k1pi4 pi2pi3

          */
      TLorentzVector pi3k4Rec(0.,0.,0.,0.);
      TLorentzVector pi2k4Rec(0.,0.,0.,0.);
      TLorentzVector pi1k4Rec(0.,0.,0.,0.);
      //TLorentzVector pi2pi3Rec(0.,0.,0.,0.); //...pay attention
      //
      TLorentzVector k3pi4Rec(0.,0.,0.,0.);
      TLorentzVector k3pi2Rec(0.,0.,0.,0.);
      TLorentzVector k3pi1Rec(0.,0.,0.,0.);
      //TLorentzVector pi1pi4Rec(0.,0.,0.,0.); //...pay attention
```

```
       //
       TLorentzVector pi1k2Rec(0.,0.,0.,0.);
       TLorentzVector pi3k2Rec(0.,0.,0.,0.);
       TLorentzVector pi4k2Rec(0.,0.,0.,0.);
       //
       TLorentzVector k1pi2Rec(0.,0.,0.,0.);
       TLorentzVector k1pi3Rec(0.,0.,0.,0.);
       TLorentzVector k1pi4Rec(0.,0.,0.,0.);
       //

       //TLorentzVector pipiRec(0.,0.,0.,0.);
       //...Luiz
       TLorentzVector pipipipiRec(0.,0.,0.,0.);

       int totcharge=0;

       //...Luiz
       TLorentzVector kkkkRec(0.,0.,0.,0.);
       //TLorentzVector mmRec(0.,0.,0.,0.);
       //TLorentzVector eeRec(0.,0.,0.,0.);
       //...Luiz./submit-condorRP.csh luianaRP4 t0RP389relui4 eos.t0.re.4499.txt

       //TLorentzVector ppRec(0.,0.,0.,0.);

       //int charray[2]={0,0};
       //double chi2array[2]={0.,0.};
       //double d0array[2]={0.,0.};
       //double dzarray[2]={0.,0.};
       //...Luiz
       //
       int charray[4]={0,0,0,0};
       double chi2array[4]={0.,0.,0.,0.};
       double d0array[4]={0.,0.,0.,0.};
       double dzarray[4]={0.,0.,0.,0.};
       int pidarray[4]={0,0,0,0};
       //...ordering
       int arraych[4]={0,0,0,0};
       double arraychi2[4]={0.,0.,0.,0.};
       double arrayd0[4]={0.,0.,0.,0.};
       double arraydz[4]={0.,0.,0.,0.};
       int arraypid[4]={0,0,0,0};

       int ntrk0=0;
       int ntrk=0;
       int ntrkvtx=0;

       std::cout << " *** <<charge>> ***   " << std::endl;

       //       for(TrackCollection::const_iterator itTrack = tracks->begin();itTrack != tracks->
end();++itTrack) {
       for(vector<MyTracks>::iterator itTrack = track_coll->begin() ; itTrack != track_coll->end(
) ; ++itTrack){

         int looper = itTrack->isLooper;
         double pt = itTrack->pt();
         double pz = itTrack->pz();
         double eta = itTrack->eta();
         double phi = itTrack->phi();
         double charge = itTrack->charge;
         int npixelhits = itTrack->nValidPixelHits;
         int nstriphits = itTrack->nValidStripHits;
         int algo = itTrack->trackAlgo;
         double chi2 = itTrack->chi2n;
         double d0 = itTrack->d0;
         double dz = itTrack->dz;

         histosTH1F["hpt"]->Fill(pt);
         histosTH1F["heta"]->Fill(eta);
         histosTH1F["hphi"]->Fill(phi);
         histosTH1F["halgo"]->Fill(algo);
         histosTH1F["hnhits"]->Fill(npixelhits+nstriphits);
```

```
  ntrk0++;

  if(npixelhits>0){
    //    if(npixelhits>0 && TMath::Abs(d0)<1. && TMath::Abs(dz)<20.){

    histosTH1F["hlooper"]->Fill(looper);
    histosTH1F["hchi2"]->Fill(chi2);
    histosTH1F["hd0"]->Fill(d0);
    histosTH1F["hdz"]->Fill(dz);

    histosTH2F["hdedx"]->Fill(itTrack->p,itTrack->harmonic2_dEdx);
    //...Luiz
    double lndEdx=TMath::Log(itTrack->harmonic2_dEdx);
    histosTH2F["hlndedx"]->Fill(itTrack->p,lndEdx);
    double l10dEdx=TMath::Log10(itTrack->harmonic2_dEdx);
    histosTH2F["hl10dedx"]->Fill(itTrack->p,l10dEdx);

    //...Luiz
    totcharge += itTrack->charge;
    double ene=TMath::Sqrt(pt*pt+pz*pz+m_pi*m_pi);
    TLorentzVector trk_lorentz(itTrack->px(),itTrack->py(),itTrack->pz(),ene);
    pipipipiRec += trk_lorentz;

    //...beware of the index here
    //...beware of the index here
    /*
    if(ntrk==0) pi1 = trk_lorentz;
    if(ntrk==1) pi2 = trk_lorentz;
    if(ntrk==2) pi3 = trk_lorentz;
    if(ntrk==3) pi4 = trk_lorentz;

    if(ntrk==0 || ntrk==1) pi1pi2Rec += trk_lorentz;
    if(ntrk==2 || ntrk==3) pi3pi4Rec += trk_lorentz;
    if(ntrk==0 || ntrk==2) pi1pi3Rec += trk_lorentz;
    if(ntrk==1 || ntrk==3) pi2pi4Rec += trk_lorentz;
    */

    //...first, tagging by track number
    if(ntrk==0) piA = trk_lorentz;
    if(ntrk==1) piB = trk_lorentz;
    if(ntrk==2) piC = trk_lorentz;
    if(ntrk==3) piD = trk_lorentz;

EPID pid2 = GetPIDSafe2(itTrack->p, itTrack->harmonic2_dEdx);

    //std::cout << "pid2 = " << pid2 << std::endl;

    std::cout << " charge = " << charge << std::endl;

    //...beware of the index here
    if(ntrk==0){
      charray[0]=charge;
      chi2array[0]=chi2;
      d0array[0]=d0;
      dzarray[0]=dz;
      pidarray[0]=pid2;
    }
    if(ntrk==1){
      charray[1]=charge;
      chi2array[1]=chi2;
      d0array[1]=d0;
      dzarray[1]=dz;
      pidarray[1]=pid2;
    }
    if(ntrk==2){
      charray[2]=charge;
      chi2array[2]=chi2;
      d0array[2]=d0;
      dzarray[2]=dz;
      pidarray[2]=pid2;
    }
    if(ntrk==3){
      charray[3]=charge;
      chi2array[3]=chi2;
      d0array[3]=d0;
      dzarray[3]=dz;
      pidarray[3]=pid2;
    }

//...ordering pions and kaons by momentum, index=1 is the highest Pt
//...we need to include kaons for the selection 11 : one primary and one Vee

vector<Double_t> piVec = { piA.Pt(), piB.Pt(), piC.Pt(), piD.Pt() };
//
sort(piVec.begin(), piVec.end());

//...ordering by Pt and connecting the charges & PID's to the particles...tricky!

if(piVec[3]!=0.0 && piVec[3]==piA.Pt()){ pi1 = piA ;
  arraych[0]=charray[0];
arraychi2[0]=chi2array[0];
  arrayd0[0]=d0array[0];
  arraydz[0]=dzarray[0];
 arraypid[0]=pidarray[0];}
if(piVec[3]!=0.0 && piVec[3]==piB.Pt()){ pi1 = piB ;
  arraych[0]=charray[1];
arraychi2[0]=chi2array[1];
  arrayd0[0]=d0array[1];
  arraydz[0]=dzarray[1];
 arraypid[0]=pidarray[1];}
if(piVec[3]!=0.0 && piVec[3]==piC.Pt()){ pi1 = piC ;
  arraych[0]=charray[2];
arraychi2[0]=chi2array[2];
  arrayd0[0]=d0array[2];
  arraydz[0]=dzarray[2];
 arraypid[0]=pidarray[2];}
if(piVec[3]!=0.0 && piVec[3]==piD.Pt()){ pi1 = piD ;
  arraych[0]=charray[3];
arraychi2[0]=chi2array[3];
  arrayd0[0]=d0array[3];
  arraydz[0]=dzarray[3];
 arraypid[0]=pidarray[3];}
//
if(piVec[2]!=0.0 && piVec[2]==piA.Pt()){ pi2 = piA ;
  arraych[1]=charray[0];
arraychi2[1]=chi2array[0];
  arrayd0[1]=d0array[0];
  arraydz[1]=dzarray[0];
 arraypid[1]=pidarray[0];}
if(piVec[2]!=0.0 && piVec[2]==piB.Pt()){ pi2 = piB ;
  arraych[1]=charray[1];
arraychi2[1]=chi2array[1];
  arrayd0[1]=d0array[1];
  arraydz[1]=dzarray[1];
 arraypid[1]=pidarray[1];}
if(piVec[2]!=0.0 && piVec[2]==piC.Pt()){ pi2 = piC ;
  arraych[1]=charray[2];
arraychi2[1]=chi2array[2];
  arrayd0[1]=d0array[2];
  arraydz[1]=dzarray[2];
 arraypid[1]=pidarray[2];}
if(piVec[2]!=0.0 && piVec[2]==piD.Pt()){ pi2 = piD ;
  arraych[1]=charray[3];
arraychi2[1]=chi2array[3];
  arrayd0[1]=d0array[3];
  arraydz[1]=dzarray[3];
 arraypid[1]=pidarray[3];}
//
if(piVec[1]!=0.0 && piVec[1]==piA.Pt()){ pi3 = piA ;
  arraych[2]=charray[0];
arraychi2[2]=chi2array[0];
  arrayd0[2]=d0array[0];
  arraydz[2]=dzarray[0];
 arraypid[2]=pidarray[0];}
```

```
if(piVec[1]!=0.0 && piVec[1]==piB.Pt()){ pi3 = piB ;
  arraych[2]=charray[1];
  arraychi2[2]=chi2array[1];
  arrayd0[2]=d0array[1];
  arraydz[2]=dzarray[1];
  arraypid[2]=pidarray[1];}
if(piVec[1]!=0.0 && piVec[1]==piC.Pt()){ pi3 = piC ;
  arraych[2]=charray[2];
arraychi2[2]=chi2array[2];
  arrayd0[2]=d0array[2];
  arraydz[2]=dzarray[2];
 arraypid[2]=pidarray[2];}
if(piVec[1]!=0.0 && piVec[1]==piD.Pt()){ pi3 = piD ;
  arraych[2]=charray[3];
arraychi2[2]=chi2array[3];
  arrayd0[2]=d0array[3];
  arraydz[2]=dzarray[3];
 arraypid[2]=pidarray[3];}
//
if(piVec[0]!=0.0 && piVec[0]==piA.Pt()){ pi4 = piA ;
  arraych[3]=charray[0];
arraychi2[3]=chi2array[0];
  arrayd0[3]=d0array[0];
  arraydz[3]=dzarray[0];
 arraypid[3]=pidarray[0];}
if(piVec[0]!=0.0 && piVec[0]==piB.Pt()){ pi4 = piB ;
  arraych[3]=charray[1];
arraychi2[3]=chi2array[1];
  arrayd0[3]=d0array[1];
  arraydz[3]=dzarray[1];
 arraypid[3]=pidarray[1];}
if(piVec[0]!=0.0 && piVec[0]==piC.Pt()){ pi4 = piC ;
  arraych[3]=charray[2];
arraychi2[3]=chi2array[2];
  arrayd0[3]=d0array[2];
  arraydz[3]=dzarray[2];
 arraypid[3]=pidarray[2];}
if(piVec[0]!=0.0 && piVec[0]==piD.Pt()){ pi4 = piD ;
  arraych[3]=charray[3];
arraychi2[3]=chi2array[3];
  arrayd0[3]=d0array[3];
  arraydz[3]=dzarray[3];
 arraypid[3]=pidarray[3];}


    //-----------------------
    double eneK=TMath::Sqrt(pt*pt+pz*pz+m_k*m_k);
    TLorentzVector trk_lorentzK(itTrack->px(),itTrack->py(),itTrack->pz(),eneK);
    kkkkRec += trk_lorentzK;

    /*
    //...Kaons
    if(ntrk==0) k1 = trk_lorentzK;
    if(ntrk==1) k2 = trk_lorentzK;
    if(ntrk==2) k3 = trk_lorentzK;
    if(ntrk==3) k4 = trk_lorentzK;

    if(ntrk==0 || ntrk==1) k1k2Rec += trk_lorentzK;
    if(ntrk==2 || ntrk==3) k3k4Rec += trk_lorentzK;
    if(ntrk==0 || ntrk==2) k1k3Rec += trk_lorentzK;
    if(ntrk==1 || ntrk==3) k2k4Rec += trk_lorentzK;
    */

    if(ntrk==0) kA = trk_lorentzK;
    if(ntrk==1) kB = trk_lorentzK;
    if(ntrk==2) kC = trk_lorentzK;
    if(ntrk==3) kD = trk_lorentzK;

vector<Double_t> kVec = { kA.Pt(), kB.Pt(), kC.Pt(), kD.Pt() };
//
sort(kVec.begin(), kVec.end());
```

```
        //
       if(kVec[3]!=0.0 && kVec[3]==kA.Pt()){ k1 = kA ; }
       if(kVec[3]!=0.0 && kVec[3]==kB.Pt()){ k1 = kB ; }
       if(kVec[3]!=0.0 && kVec[3]==kC.Pt()){ k1 = kC ; }
       if(kVec[3]!=0.0 && kVec[3]==kD.Pt()){ k1 = kD ; }
       //
       if(kVec[2]!=0.0 && kVec[2]==kA.Pt()){ k2 = kA ; }
       if(kVec[2]!=0.0 && kVec[2]==kB.Pt()){ k2 = kB ; }
       if(kVec[2]!=0.0 && kVec[2]==kC.Pt()){ k2 = kC ; }
       if(kVec[2]!=0.0 && kVec[2]==kD.Pt()){ k2 = kD ; }
       //
       if(kVec[1]!=0.0 && kVec[1]==kA.Pt()){ k3 = kA ; }
       if(kVec[1]!=0.0 && kVec[1]==kB.Pt()){ k3 = kB ; }
       if(kVec[1]!=0.0 && kVec[1]==kC.Pt()){ k3 = kC ; }
       if(kVec[1]!=0.0 && kVec[1]==kD.Pt()){ k3 = kD ; }
       //
       if(kVec[0]!=0.0 && kVec[0]==kA.Pt()){ k4 = kA ; }
       if(kVec[0]!=0.0 && kVec[0]==kB.Pt()){ k4 = kB ; }
       if(kVec[0]!=0.0 && kVec[0]==kC.Pt()){ k4 = kC ; }
       if(kVec[0]!=0.0 && kVec[0]==kD.Pt()){ k4 = kD ; }


           ntrk++;

           } //...end of npixelhits>0
           } //...end of MyTracks

           //-----------------------
           //double eneM=TMath::Sqrt(pt*pt+pz*pz+m_mu*m_mu);
           //TLorentzVector trk_lorentzM(itTrack->px(),itTrack->py(),itTrack->pz(),eneM);
           //mmRec += trk_lorentzM;
           //double eneE=TMath::Sqrt(pt*pt+pz*pz+m_e*m_e);
           //TLorentzVector trk_lorentzE(itTrack->px(),itTrack->py(),itTrack->pz(),eneE);
           //eeRec += trk_lorentzE;

           //...Luiz
           //double enep=TMath::Sqrt(pt*pt+pz*pz+m_p*m_p);
           //TLorentzVector trk_lorentzp(itTrack->px(),itTrack->py(),itTrack->pz(),enep);
           //ppRec += trk_lorentzp;

           //
           pi1pi2Rec = pi1 + pi2;
           pi3pi4Rec = pi3 + pi4;
           pi1pi3Rec = pi1 + pi3;
           pi2pi4Rec = pi2 + pi4;

           //
           k1k2Rec = k1 + k2;
           k3k4Rec = k3 + k4;
           k1k3Rec = k1 + k3;
           k2k4Rec = k2 + k4;

           //...combining pions and kaons for the event selection type = 11 (one primary & one Ve
e)
           //
           /*
     ...first combining, then select the Q_pairs=0

     pi1pi2 pi3k4
     pi1pi3 pi2k4
     pi2pi3 pi1k4

     pi1pi2 k3pi4
     pi1pi4 k3pi2
     pi2pi4 k3pi1

     pi1k2 pi3pi4
     pi3k2 pi1pi4
     pi4k2 pi1pi3

     k1pi2 pi3pi4
     k1pi3 pi2pi4
```

```
  k1pi4 pi2pi3

        */
        //...commented out means already defined
        //
        //pi1pi2Rec
        pi3k4Rec  = pi3 + k4;
        //pi1pi3Rec
        pi2k4Rec  = pi2 + k4;
        pi2pi3Rec = pi2 + pi3; //...pay attention
        pi1k4Rec  = pi1 + k4;
        //
        //
        //pi1pi2Rec
        k3pi4Rec  = k3  + pi4;
        pi1pi4Rec = pi1 + pi4; //...pay attention
        k3pi2Rec  = k3  + pi2;
        //pi2pi4Rec
        k3pi1Rec  = k3  + pi1;
        //
        //
        pi1k2Rec  = pi1 + k2;
        //pi3pi4Rec
        pi3k2Rec  = pi3 + k2;
        //pi1pi4Rec
        pi4k2Rec  = pi4 + k2;
        //pi1pi3Rec
        //
        //
        k1pi2Rec  = k1  + pi2;
        //pi3pi4Rec
        k1pi3Rec  = k1  + pi3;
        //pi2pi4Rec

        k1pi4Rec  = k1  + pi4;
        //pi2pi3Rec

        /////} //...end of npixelhits>0
 /////} //...end of MyTracks

        /*
    std::cout << "***track***   " << std::endl;
    std::cout << "pidarray[0] = " << pidarray[0] << std::endl;
    std::cout << "pidarray[1] = " << pidarray[1] << std::endl;
    std::cout << "pidarray[2] = " << pidarray[2] << std::endl;
    std::cout << "pidarray[3] = " << pidarray[3] << std::endl;
    std::cout << "pidarrayk[0] = " << pidarrayk[0] << std::endl;
    std::cout << "pidarrayk[1] = " << pidarrayk[1] << std::endl;
    std::cout << "pidarrayk[2] = " << pidarrayk[2] << std::endl;
    std::cout << "pidarrayk[3] = " << pidarrayk[3] << std::endl;
        */

        /*
    std::cout << " piA.Pt() = " << piA.Pt() << std::endl;
    std::cout << " piB.Pt() = " << piB.Pt() << std::endl;
    std::cout << " piC.Pt() = " << piC.Pt() << std::endl;
    std::cout << " piD.Pt() = " << piD.Pt() << std::endl;
    //
    std::cout << " pi1.Pt() = " << pi1.Pt() << std::endl;
    std::cout << " pi2.Pt() = " << pi2.Pt() << std::endl;
    std::cout << " pi3.Pt() = " << pi3.Pt() << std::endl;
    std::cout << " pi4.Pt() = " << pi4.Pt() << std::endl;

    std::cout << " kA.Pt() = " << kA.Pt() << std::endl;
    std::cout << " kB.Pt() = " << kB.Pt() << std::endl;
    std::cout << " kC.Pt() = " << kC.Pt() << std::endl;
    std::cout << " kD.Pt() = " << kD.Pt() << std::endl;
    //
    std::cout << " k1.Pt() = " << k1.Pt() << std::endl;
    std::cout << " k2.Pt() = " << k2.Pt() << std::endl;
    std::cout << " k3.Pt() = " << k3.Pt() << std::endl;
    std::cout << " k4.Pt() = " << k4.Pt() << std::endl;
```

```
          */

          /*
      std::cout << "***charge***   " << std::endl;
      std::cout << "charray[0] = " << charray[0] << std::endl;
      std::cout << "charray[1] = " << charray[1] << std::endl;
      std::cout << "charray[2] = " << charray[2] << std::endl;
      std::cout << "charray[3] = " << charray[3] << std::endl;
      std::cout << "arraych[0] = " << arraych[0] << std::endl;
      std::cout << "arraych[1] = " << arraych[1] << std::endl;
      std::cout << "arraych[2] = " << arraych[2] << std::endl;
      std::cout << "arraych[3] = " << arraych[3] << std::endl;
          */

      //...reseting to original definition with new order

        charray[0] =   arraych[0] ;// charge;
      chi2array[0] = arraychi2[0] ;// chi2;
        d0array[0] =   arrayd0[0] ;// d0;
        dzarray[0] =   arraydz[0] ;// dz;
       pidarray[0] =  arraypid[0] ;// pid2;
       //
        charray[1] =   arraych[1] ;// charge;
      chi2array[1] = arraychi2[1] ;// chi2;
        d0array[1] =   arrayd0[1] ;// d0;
        dzarray[1] =   arraydz[1] ;// dz;
       pidarray[1] =  arraypid[1] ;// pid2;
       //
        charray[2] =   arraych[2] ;// charge;
      chi2array[2] = arraychi2[2] ;// chi2;
        d0array[2] =   arrayd0[2] ;// d0;
        dzarray[2] =   arraydz[2] ;// dz;
       pidarray[2] =  arraypid[2] ;// pid2;
       //
        charray[3] =   arraych[3] ;// charge;
      chi2array[3] = arraychi2[3] ;// chi2;
        d0array[3] =   arrayd0[3] ;// d0;
        dzarray[3] =   arraydz[3] ;// dz;
       pidarray[3] =  arraypid[3] ;// pid2;
       //

       /*
      std::cout << "*** reseting ***   " << std::endl;
      std::cout << "charray[0] = " << charray[0] << std::endl;
      std::cout << "charray[1] = " << charray[1] << std::endl;
      std::cout << "charray[2] = " << charray[2] << std::endl;
      std::cout << "charray[3] = " << charray[3] << std::endl;
       */

          /*
    pi1pi2 pi3k4
    pi1pi3 pi2k4
    pi2pi3 pi1k4

    pi1pi2 k3pi4
    pi1pi4 k3pi2
    pi2pi4 k3pi1

    pi1k2 pi3pi4
    pi3k2 pi1pi4
    pi4k2 pi1pi3

    k1pi2 pi3pi4
    k1pi3 pi2pi4
    k1pi4 pi2pi3
          */
      /*
      std::cout << " ***mass***********************   " << std::endl;
      std::cout << " pi1pi2Rec.M() = " << pi1pi2Rec.M() << std::endl;
      std::cout << " pi3pi4Rec.M() = " << pi3pi4Rec.M() << std::endl;
      std::cout << " pi1pi3Rec.M() = " << pi1pi3Rec.M() << std::endl;
      std::cout << " pi2pi4Rec.M() = " << pi2pi4Rec.M() << std::endl;
```

```
       std::cout << " pi1pi4Rec.M() = " << pi1pi4Rec.M() << std::endl;
       std::cout << " pi2pi3Rec.M() = " << pi2pi3Rec.M() << std::endl;
       std::cout << " ...             " << std::endl;
       std::cout << " pi3k4Rec.M()  = " << pi3k4Rec.M() << std::endl;
       std::cout << " pi2k4Rec.M()  = " << pi2k4Rec.M() << std::endl;
       std::cout << " pi1k4Rec.M()  = " << pi1k4Rec.M() << std::endl;
       std::cout << " ...             " << std::endl;
       std::cout << " k3pi4Rec.M()  = " << k3pi4Rec.M() << std::endl;
       std::cout << " k3pi2Rec.M()  = " << k3pi2Rec.M() << std::endl;
       std::cout << " k3pi1Rec.M()  = " << k3pi1Rec.M() << std::endl;
       std::cout << " ...             " << std::endl;
       std::cout << " pi1k2Rec.M()  = " << pi1k2Rec.M() << std::endl;
       std::cout << " pi3k2Rec.M()  = " << pi3k2Rec.M() << std::endl;
       std::cout << " pi4k2Rec.M()  = " << pi4k2Rec.M() << std::endl;
       std::cout << " ...             " << std::endl;
       std::cout << " k1pi2Rec.M()  = " << k1pi2Rec.M() << std::endl;
       std::cout << " k1pi3Rec.M()  = " << k1pi3Rec.M() << std::endl;
       std::cout << " k1pi4Rec.M()  = " << k1pi4Rec.M() << std::endl;
       */

       histosTH1F["hntrk0"]->Fill(ntrk0);
       histosTH1F["hntrk"]->Fill(ntrk);


       if(ntrk==0){
         int nclusters=  sipixelcluster_coll->size();
         int nclusters2= sistripcluster_coll->nStripClusters;

         histosTH1F["hnclusters"]->Fill(nclusters);
         histosTH1F["hnclusters2"]->Fill(nclusters2);
       }

       int nvtx=0;
       //        for(VertexCollection::const_iterator itVtx = vertices->begin();itVtx != vertices-
>end();++itVtx) {
       for(vector<MyVertex>::iterator itVtx = vertex_coll->begin() ; itVtx != vertex_coll->end()
; ++itVtx){
         int vtxisfake = itVtx->fake;
         if(vtxisfake==0) nvtx++;
         else continue;

         ntrkvtx = itVtx->ntracks;
         //...Luiz
         //////itVtx->Print();

         //...nvtx counting is ok
       }

       histosTH1F["hnvtx"]->Fill(nvtx);
       if(nvtx==1) histosTH1F["hntrkvtx"]->Fill(ntrkvtx);
       //...Luiz
       if(nvtx==0) histosTH1F["hntrkvtx0"]->Fill(ntrkvtx);
       if(nvtx==2) histosTH1F["hntrkvtx2"]->Fill(ntrkvtx);
       if(nvtx==3) histosTH1F["hntrkvtx3"]->Fill(ntrkvtx);
       if(nvtx==4) histosTH1F["hntrkvtx4"]->Fill(ntrkvtx);


       //not yet vertex cut, checking vertex-finding efficiency
       int  isfake = vertex_coll->begin()->fake;
       double xvtx = vertex_coll->begin()->x;
       //double xvtx = vertex_coll->begin()->x; only primary vertex?
       double yvtx = vertex_coll->begin()->y;
       double zvtx = vertex_coll->begin()->z;

       //        double chi2vtx = vertices->begin()->normalizedChi2();
       // not sure if the same variable.
       // myvertex.chi2         = p->chi2();
       double chi2vtx = vertex_coll->begin()->chi2;
       double ndofvtx = vertex_coll->begin()->ndof;
       //        double ndofvtx = vertex_coll->begin()->ndof;

       //        ntrkvtx = vertex_coll->begin()->ntracks;
       //...........
       //...Kshort collection...Luiz
       //..isVee
       bool isKshort = false;
       int nks=0;
       for(vector<MyKshorts>::iterator it_ks = kshort_coll->begin() ; it_ks != kshort_coll->end()
 ; ++it_ks){

         nks++;
         isKshort = nks;
         double ksvertexx = it_ks->vertexx;
         double ksvertexy = it_ks->vertexy;
         double ksvertexz = it_ks->vertexz;
         double kspt = it_ks->pt;
         double kseta = it_ks->eta;
         double ksphi = it_ks->phi;
         double ksmass = it_ks->mass;
         double ksradius = TMath::Sqrt((ksvertexx-xvtx)*(ksvertexx-xvtx)+(ksvertexy-yvtx)*(ksvert
exy-yvtx));
         double energy = TMath::Sqrt(kspt*kspt+0.4976*0.4976);
         double gammalorentz = energy/0.4976;
         double kslifetime = ksradius/gammalorentz;
         histosTH1F["hkspt"]->Fill(kspt,wei);
         histosTH1F["hkseta"]->Fill(kseta,wei);
         histosTH1F["hksphi"]->Fill(ksphi,wei);
         histosTH1F["hksmass"]->Fill(ksmass,wei);
         //
         if(nks == 1)histosTH1F["hksmassv1"]->Fill(ksmass,wei);
         if(nks == 2)histosTH1F["hksmassv2"]->Fill(ksmass,wei);
         if(nks == 3)histosTH1F["hksmassv3"]->Fill(ksmass,wei);
         //
         histosTH1F["hksvertexx"]->Fill(ksvertexx,wei);
         histosTH1F["hksvertexy"]->Fill(ksvertexy,wei);
         histosTH1F["hksvertexz"]->Fill(ksvertexz,wei);
         histosTH1F["hksradius"]->Fill(ksradius,wei);
         histosTH1F["hkslifetime"]->Fill(kslifetime,wei);
         histosTH2F["h2dimksxy"]->Fill(ksvertexx,ksvertexy);
         histosTH2F["h2dimksxz"]->Fill(ksvertexx,ksvertexz);
         histosTH2F["h2dimksyz"]->Fill(ksvertexy,ksvertexz);
         //std::cout << " nks = " << nks << std::endl;
         //std::cout << " ksvertexx = " << ksvertexx << std::endl;
         //std::cout << " ksvertexy = " << ksvertexy << std::endl;
         //std::cout << " ksvertexz = " << ksvertexz << std::endl;
         //std::cout << " ksmass = " << ksmass << std::endl;
         //it_ks->Print();
       }
       //...end Kshort
       histosTH1F["hnks"]->Fill(nks);
       histosTH2F["hntrknks"]->Fill(ntrk,nks);
       histosTH2F["hnvtxnks"]->Fill(nvtx,nks);
       histosTH2F["hntrknvtx"]->Fill(ntrk,nvtx);
       //std::cout << " --------------------------- " << std::endl;
       //std::cout << " nks  = " << nks << std::endl;
       //std::cout << " ntrk = " << ntrk << std::endl;
       //std::cout << " nvtx = " << nvtx << std::endl;
       //std::cout << " isKshort = " << isKshort << std::endl;
       //std::cout << " --------------------------- " << std::endl;
       //................

      /*
       //...Kshort...secondaryVertex
       //int  isfake = kshorts_coll->begin()->fake;
       double xk = kshort_coll->begin()->vertexx;
       double yk = kshort_coll->begin()->vertexy;
       double zk = kshort_coll->begin()->vertexz;
       ////double chi2vtxk = kshorts_coll->begin()->chi2n;
       //...Kshort
       histosTH1F["hxk"]->Fill(xk,wei);
       histosTH1F["hyk"]->Fill(yk,wei);
       histosTH1F["hzk"]->Fill(zk,wei);
       //
       histosTH2F["h2dimxyk"]->Fill(xk,yk);
       histosTH2F["h2dimxzk"]->Fill(xk,zk);
       histosTH2F["h2dimyzk"]->Fill(yk,zk);
       */
       /*
       //...secondaryVertex
       //
       MyKshorts& secondaryVertex = kshort_coll->at(0);
       //  at 2.6844 cm
       histosTH1F["sec_vtx_xpos"]->Fill(secondaryVertex.vertexx,wei);
       histosTH1F["sec_vtx_ypos"]->Fill(secondaryVertex.vertexy,wei);
       histosTH1F["sec_vtx_zpos"]->Fill(secondaryVertex.vertexz,wei);
       */

       //
      /////histosTH1F["sec_vtx_ndof"]->Fill(secondaryVertex.ndof);
      /////histosTH1F["sec_vtx_chi2"]->Fill(secondaryVertex.chi2);
      /////histosTH1F["sec_vtx_chi2n"]->Fill(secondaryVertex.chi2n());
      /////histosTH1F["sec_vtx_ntracks"]->Fill(secondaryVertex.ntracks);
      /////histosTH1F["sec_vtx_sumpt"]->Fill(secondaryVertex.SumPtTracks);

       //...Lambda collection...Luiz
       bool isLambda = false;
       int nlam=0;
       for(vector<MyLambdas>::iterator it_lam = lambda_coll->begin() ; it_lam != lambda_coll->end
() ; ++it_lam){

         nlam++;
         isLambda = nlam;
         double lamvertexx = it_lam->vertexx;
         double lamvertexy = it_lam->vertexy;
         double lamvertexz = it_lam->vertexz;
         double lampt = it_lam->pt;
         double lameta = it_lam->eta;
         double lamphi = it_lam->phi;
         double lammass = it_lam->mass;
         double lamradius = TMath::Sqrt((lamvertexx-xvtx)*(lamvertexx-xvtx)+(lamvertexy-yvtx)*(la
mvertexy-yvtx));
         histosTH1F["hlampt"]->Fill(lampt,wei);
         histosTH1F["hlameta"]->Fill(lameta,wei);
         histosTH1F["hlamphi"]->Fill(lamphi,wei);
         histosTH1F["hlammass"]->Fill(lammass,wei);
         histosTH1F["hlamvertexx"]->Fill(lamvertexx,wei);
         histosTH1F["hlamvertexy"]->Fill(lamvertexy,wei);
         histosTH1F["hlamvertexz"]->Fill(lamvertexz,wei);
         histosTH1F["hlamradius"]->Fill(lamradius,wei);
         histosTH2F["h2dimlamxy"]->Fill(lamvertexx,lamvertexy);
         histosTH2F["h2dimlamxz"]->Fill(lamvertexx,lamvertexz);
         histosTH2F["h2dimlamyz"]->Fill(lamvertexy,lamvertexz);
         //std::cout << " ksvertexx = " << ksvertexx << std::endl;
         //std::cout << " ksvertexy = " << ksvertexy << std::endl;
         //std::cout << " ksvertexz = " << ksvertexz << std::endl;
         //std::cout << " ksmass = " << ksmass << std::endl;
         //it_ks->Print();
       }
       //...end Lambda
       histosTH1F["hnlam"]->Fill(nlam);


       //for vertex plots
       //...Luiz  ntrk==4
       //fiducialRegion   = (ntrk==2 && TMath::Abs(pi1.Eta())<etaCut && TMath::Abs(pi2.Eta())<eta
Cut);
       //fiducialRegionPt = (ntrk==2 && pi1.Pt()>ptCut && pi2.Pt()>ptCut);
       //...Luiz
       //...Pions
       fiducialRegion   = (ntrk==4 && TMath::Abs(pi1.Eta())<etaCut && TMath::Abs(pi2.Eta())<etaCu
t &&
                   TMath::Abs(pi3.Eta())<etaCut && TMath::Abs(pi4.Eta())<etaCut);
       fiducialRegionPt = (ntrk==4 && pi1.Pt()>ptCut && pi2.Pt()>ptCut &&
                           pi3.Pt()>ptCut && pi4.Pt()>ptCut);
       //...Kaons
       fiducialRegionK   = (ntrk==4 && TMath::Abs(k1.Eta())<etaCut && TMath::Abs(k2.Eta())<etaCut
 &&
                   TMath::Abs(k3.Eta())<etaCut && TMath::Abs(k4.Eta())<etaCut);
       fiducialRegionPtK = (ntrk==4 && k1.Pt()>ptCut && k2.Pt()>ptCut &&
                           k3.Pt()>ptCut && k4.Pt()>ptCut);
       ////fiducialRegion   = (ntrk==4);
       ////fiducialRegionPt = (ntrk==4);
       histosTH1F["hvtx"]->Fill( isfake );
       //...Luiz
       if(ntrk==4){
         histosTH1F["hvtx2"]->Fill( isfake );
         if(fiducialRegion && totcharge==0) histosTH1F["hvtx3"]->Fill( isfake );
       }

       //...very important...needed for theVees
       //......not this--> if(nvtx!=0 || nvtx!=1) continue;
       if(nvtx!=0){
         if(nvtx!=1){
           if(nvtx!=2) continue;
         }
       }
       //----------------------------
       //invariant mass
       //...Luiz
       double mrec=pipipipiRec.M();
       double mrecKKKK=kkkkRec.M();
       //double mrecMM=mmRec.M();
       //double mrecEE=eeRec.M();
       //...Luiz
       //double mrecpp=ppRec.M();

       // M(1,2) M(3,4) M(1,3) M(2,4)
       double mrecpi1pi2=pi1pi2Rec.M();
       double mrecpi3pi4=pi3pi4Rec.M();
       double mrecpi1pi3=pi1pi3Rec.M();
       double mrecpi2pi4=pi2pi4Rec.M();

       // M(1,2) M(3,4) M(1,3) M(2,4)
       double mreck1k2=k1k2Rec.M();
       double mreck3k4=k3k4Rec.M();
       double mreck1k3=k1k3Rec.M();
       double mreck2k4=k2k4Rec.M();

     //...combining pions and kaons for the event selection type = 11 (one primary & one Vee)
           /*
     ...first combining, then select the Q_pairs=0

     pi1pi2 pi3k4
     pi1pi3 pi2k4
     pi2pi3 pi1k4

     pi1pi2 k3pi4
     pi1pi4 k3pi2
     pi2pi4 k3pi1

     pi1k2 pi3pi4
     pi3k2 pi1pi4
     pi4k2 pi1pi3

     k1pi2 pi3pi4
     k1pi3 pi2pi4
     k1pi4 pi2pi3

           */
       double mrecpi3k4=pi3k4Rec.M();
       double mrecpi2k4=pi2k4Rec.M();
       double mrecpi1k4=pi1k4Rec.M();
       double mreck3pi4=k3pi4Rec.M();
       double mreck3pi2=k3pi2Rec.M();
       double mreck3pi1=k3pi1Rec.M();
       double mrecpi1k2=pi1k2Rec.M();
       double mrecpi3k2=pi3k2Rec.M();
       double mrecpi4k2=pi4k2Rec.M();
       double mreck1pi2=k1pi2Rec.M();
       double mreck1pi3=k1pi3Rec.M();
       double mreck1pi4=k1pi4Rec.M();
       //
       //...for completeness
       double mrecpi1pi4=pi1pi4Rec.M();
       double mrecpi2pi3=pi2pi3Rec.M();
       //...for completeness
       double mreck1k4=k1k4Rec.M();
       double mreck2k3=k2k3Rec.M();

       double mrecKpi = 0.0;

       //----------------------------
       // xi cut
       // Mmax=13000*xi_max
       // 0.1 -> 1300 GeV
       // 0.01 -> 130 GeV
       // 0.001 -> 13 GeV

        //...Luiz...rapidity = 1/2 ln ( xi_proton_2/xi_proton_1 )
        double rapy = 0.5*TMath::Log(xiR/xiL);
        //
       //----------------------------


        //...cut 9...........theVees

           if(ntrk==4){
           if(totcharge==0){
             if(isKshort){

               histosTH1F["hm2rec2OSvee9"]->Fill(mrec);
                 if(charray[0]+charray[1] == 0)
                    {
               histosTH1F["hm2rec2OS_pi1pi2vee9"]->Fill(mrecpi1pi2);
               histosTH1F["hm2rec2OS_pi3pi4vee9"]->Fill(mrecpi3pi4);
                    }else if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_pi1pi3vee9"]->Fill(mrecpi1pi3);
               histosTH1F["hm2rec2OS_pi2pi4vee9"]->Fill(mrecpi2pi4);
                   }

               if(nvtx==0){
               histosTH1F["hm2rec2OSvee90"]->Fill(mrec);
                 if(charray[0]+charray[1] == 0)
                    {
               histosTH1F["hm2rec2OS_pi1pi2vee90"]->Fill(mrecpi1pi2);
               histosTH1F["hm2rec2OS_pi3pi4vee90"]->Fill(mrecpi3pi4);
                    }else if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_pi1pi3vee90"]->Fill(mrecpi1pi3);
               histosTH1F["hm2rec2OS_pi2pi4vee90"]->Fill(mrecpi2pi4);
                   }
               } //end of nvtx=0

             if(nvtx==1 && nks==1){
               histosTH1F["hm2rec2OSvee91"]->Fill(mrec);
                 if(charray[0]+charray[1] == 0)
                    {
               histosTH1F["hm2rec2OS_pi1pi2vee91"]->Fill(mrecpi1pi2);
               histosTH1F["hm2rec2OS_pi3pi4vee91"]->Fill(mrecpi3pi4);
                    }else if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_pi1pi3vee91"]->Fill(mrecpi1pi3);
               histosTH1F["hm2rec2OS_pi2pi4vee91"]->Fill(mrecpi2pi4);
                   }
               } //end of nks=1

             if(nvtx==0 && nks==2){
               histosTH1F["hm2rec2OSvee92"]->Fill(mrec);
                 if(charray[0]+charray[1] == 0)
                    {
               histosTH1F["hm2rec2OS_pi1pi2vee92"]->Fill(mrecpi1pi2);
```

```
  histosTH1F["hm2rec2OS_pi3pi4vee92"]->Fill(mrecpi3pi4);
       }else if(charray[0]+charray[2] == 0){
  histosTH1F["hm2rec2OS_pi1pi3vee92"]->Fill(mrecpi1pi3);
  histosTH1F["hm2rec2OS_pi2pi4vee92"]->Fill(mrecpi2pi4);
      }
  } //end of nks=2

if(nvtx==0){
  histosTH1F["hm2rec2OSvtx0"]->Fill(mrec);
    if(charray[0]+charray[1] == 0)
       {
  histosTH1F["hm2rec2OS_pi1pi2vtx0"]->Fill(mrecpi1pi2);
  histosTH1F["hm2rec2OS_pi3pi4vtx0"]->Fill(mrecpi3pi4);
       }else if(charray[0]+charray[2] == 0){
  histosTH1F["hm2rec2OS_pi1pi3vtx0"]->Fill(mrecpi1pi3);
  histosTH1F["hm2rec2OS_pi2pi4vtx0"]->Fill(mrecpi2pi4);
      }
  } //end of nvtx=0

if(nvtx==0 && nks==1){
  histosTH1F["hm2rec2OSvtx01"]->Fill(mrec);
    if(charray[0]+charray[1] == 0)
       {
  histosTH1F["hm2rec2OS_pi1pi2vtx01"]->Fill(mrecpi1pi2);
  histosTH1F["hm2rec2OS_pi3pi4vtx01"]->Fill(mrecpi3pi4);
       }else if(charray[0]+charray[2] == 0){
  histosTH1F["hm2rec2OS_pi1pi3vtx01"]->Fill(mrecpi1pi3);
  histosTH1F["hm2rec2OS_pi2pi4vtx01"]->Fill(mrecpi2pi4);
      }
  } //end of nvtx=0 and nks=1

if(nvtx==0 && nks==2){
  histosTH1F["hm2rec2OSvtx02"]->Fill(mrec);
    if(charray[0]+charray[1] == 0)
       {
  histosTH1F["hm2rec2OS_pi1pi2vtx02"]->Fill(mrecpi1pi2);
  histosTH1F["hm2rec2OS_pi3pi4vtx02"]->Fill(mrecpi3pi4);
       }else if(charray[0]+charray[2] == 0){
  histosTH1F["hm2rec2OS_pi1pi3vtx02"]->Fill(mrecpi1pi3);
  histosTH1F["hm2rec2OS_pi2pi4vtx02"]->Fill(mrecpi2pi4);
      }
  } //end of nvtx=0 and nks=2

if(nvtx==1 && nks==1){
  histosTH1F["hm2rec2OSvtx11"]->Fill(mrec);
    if(charray[0]+charray[1] == 0)
       {
  histosTH1F["hm2rec2OS_pi1pi2vtx11"]->Fill(mrecpi1pi2);
  histosTH1F["hm2rec2OS_pi3pi4vtx11"]->Fill(mrecpi3pi4);
       }else if(charray[0]+charray[2] == 0){
  histosTH1F["hm2rec2OS_pi1pi3vtx11"]->Fill(mrecpi1pi3);
  histosTH1F["hm2rec2OS_pi2pi4vtx11"]->Fill(mrecpi2pi4);
      }
  } //end of nvtx=1 and nks=1

if(nvtx==1){
  histosTH1F["hm2rec2OSvtx1"]->Fill(mrec);
    if(charray[0]+charray[1] == 0)
       {
  histosTH1F["hm2rec2OS_pi1pi2vtx1"]->Fill(mrecpi1pi2);
  histosTH1F["hm2rec2OS_pi3pi4vtx1"]->Fill(mrecpi3pi4);
       }else if(charray[0]+charray[2] == 0){
  histosTH1F["hm2rec2OS_pi1pi3vtx1"]->Fill(mrecpi1pi3);
  histosTH1F["hm2rec2OS_pi2pi4vtx1"]->Fill(mrecpi2pi4);
      }
  } //end of nvtx=1

if(nvtx==2){
  histosTH1F["hm2rec2OSvtx2"]->Fill(mrec);
    if(charray[0]+charray[2] == 0)
       {
  histosTH1F["hm2rec2OS_pi1pi2vtx2"]->Fill(mrecpi1pi2);
```

```
               histosTH1F["hm2rec2OS_pi3pi4vtx2"]->Fill(mrecpi3pi4);
                    }else if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_pi1pi3vtx2"]->Fill(mrecpi1pi3);
               histosTH1F["hm2rec2OS_pi2pi4vtx2"]->Fill(mrecpi2pi4);
                   }
               } //end of nvtx=2

              } //...end of isKshort
             } //...end of totalcharge=0
            } //end of ntrk=4
        //...end of cut 9

           //...checking dEdx efficiency 11, 02, 01
           //...fiducial: each entry in the scatter plot dE/dx vs p is a track!
           // int ntrkvee=0;
             for(vector<MyTracks>::iterator itTrack = track_coll->begin() ; itTrack != track_coll
->end() ; ++itTrack){
               int npixelhitsvee = itTrack->nValidPixelHits;
               if(npixelhitsvee>0){
                 if(fiducialRegion && fiducialRegionPt && fiducialRegionK && fiducialRegionPtK &&
 totcharge==0 && isKshort && nvtx==1 && nks==1){
                   histosTH2F["hdedxvee11"]->Fill(itTrack->p,itTrack->harmonic2_dEdx);}
                 if(fiducialRegion && fiducialRegionPt && fiducialRegionK && fiducialRegionPtK &&
 totcharge==0 && isKshort && nvtx==0 && nks==2){
                   histosTH2F["hdedxvee02"]->Fill(itTrack->p,itTrack->harmonic2_dEdx);}
                 if(fiducialRegion && fiducialRegionPt && fiducialRegionK && fiducialRegionPtK &&
 totcharge==0 && isKshort && nvtx==0 && nks==1){
                   histosTH2F["hdedxvee01"]->Fill(itTrack->p,itTrack->harmonic2_dEdx);}
               }
             // ntrkvee++;
             // std::cout << " ntrkvee = " << ntrkvee << std::endl;
             } //...end of Mytracks-2



        //...cut 8.................theVees
        //...fiducial Vees
        if(fiducialRegion && fiducialRegionPt && fiducialRegionK && fiducialRegionPtK){
        //if(fiducialRegion && fiducialRegionPt){

          //AA...using PID...not yet
          if(totcharge==0){

          //...using PID Pions & Kaons for selection=11
          //       if(
          //   (pidarray[0]==2 && pidarray[1]==3 && pidarray[2]==3 && pidarray[3]==3)  ||
          //   (pidarray[0]==3 && pidarray[1]==2 && pidarray[2]==3 && pidarray[3]==3)  ||
          //   (pidarray[0]==3 && pidarray[1]==3 && pidarray[2]==2 && pidarray[3]==3)  ||
          //   (pidarray[0]==3 && pidarray[1]==3 && pidarray[2]==3 && pidarray[3]==2)   )
          // {
             //...Luiz
             //histosTH1F["hm2rec2OSvee"]->Fill(mrec);

             if(isKshort){

             //...one primary & one Vee  //  K+pi- pi+pi-  or  K-pi+ pi+pi-
             if(nvtx==1 && nks==1){

           /*
     ...first combining, then select the Q_pairs=0

     pi1pi2 pi3k4
     pi1pi3 pi2k4
     pi2pi3 pi1k4

     pi1pi2 k3pi4
     pi1pi4 k3pi2
     pi2pi4 k3pi1

     pi1k2 pi3pi4
     pi3k2 pi1pi4
     pi4k2 pi1pi3
     k1pi2 pi3pi4
     k1pi3 pi2pi4
     k1pi4 pi2pi3

           */
               //double mrecKpi = 0.0 ;
               //
               if(charray[2]+charray[3] == 0 && pidarray[2]==3 && pidarray[3]==2 ) {mrecKpi = mre
cpi3k4 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               if(charray[1]+charray[3] == 0 && pidarray[1]==3 && pidarray[3]==2 ) {mrecKpi = mre
cpi2k4 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               if(charray[0]+charray[3] == 0 && pidarray[0]==3 && pidarray[3]==2 ) {mrecKpi = mre
cpi1k4 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               //
               if(charray[2]+charray[3] == 0 && pidarray[2]==2 && pidarray[3]==3 ) {mrecKpi = mre
ck3pi4 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               if(charray[2]+charray[1] == 0 && pidarray[2]==2 && pidarray[1]==3 ) {mrecKpi = mre
ck3pi2 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               if(charray[2]+charray[0] == 0 && pidarray[2]==2 && pidarray[0]==3 ) {mrecKpi = mre
ck3pi1 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               //
               if(charray[0]+charray[1] == 0 && pidarray[0]==3 && pidarray[1]==2 ) {mrecKpi = mre
cpi1k2 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               if(charray[2]+charray[1] == 0 && pidarray[2]==3 && pidarray[1]==2 ) {mrecKpi = mre
cpi3k2 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               if(charray[3]+charray[1] == 0 && pidarray[3]==3 && pidarray[1]==2 ) {mrecKpi = mre
cpi4k2 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               //
               if(charray[0]+charray[1] == 0 && pidarray[0]==2 && pidarray[1]==3 ) {mrecKpi = mre
ck1pi2 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               if(charray[0]+charray[2] == 0 && pidarray[0]==2 && pidarray[2]==3 ) {mrecKpi = mre
ck1pi3 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}
               if(charray[0]+charray[3] == 0 && pidarray[0]==2 && pidarray[3]==3 ) {mrecKpi = mre
ck1pi4 ; histosTH1F["hm2rec2OSvee11"]->Fill(mrecKpi);}


               if(charray[2]+charray[3] == 0 && pidarray[2]==3 && pidarray[3]==2 ) histosTH1F["hm
2rec2OSvee11a"]->Fill( mrecpi3k4 );
               if(charray[1]+charray[3] == 0 && pidarray[1]==3 && pidarray[3]==2 ) histosTH1F["hm
2rec2OSvee11b"]->Fill( mrecpi2k4 );
               if(charray[0]+charray[3] == 0 && pidarray[0]==3 && pidarray[3]==2 ) histosTH1F["hm
2rec2OSvee11c"]->Fill( mrecpi1k4 );
               //
               if(charray[2]+charray[3] == 0 && pidarray[2]==2 && pidarray[3]==3 ) histosTH1F["hm
2rec2OSvee11d"]->Fill( mreck3pi4 );
               if(charray[2]+charray[1] == 0 && pidarray[2]==2 && pidarray[1]==3 ) histosTH1F["hm
2rec2OSvee11e"]->Fill( mreck3pi2 );
               if(charray[2]+charray[0] == 0 && pidarray[2]==2 && pidarray[0]==3 ) histosTH1F["hm
2rec2OSvee11f"]->Fill( mreck3pi1 );
               //
               if(charray[0]+charray[1] == 0 && pidarray[0]==3 && pidarray[1]==2 ) histosTH1F["hm
2rec2OSvee11g"]->Fill( mrecpi1k2 );
               if(charray[2]+charray[1] == 0 && pidarray[2]==3 && pidarray[1]==2 ) histosTH1F["hm
2rec2OSvee11h"]->Fill( mrecpi3k2 );
               if(charray[3]+charray[1] == 0 && pidarray[3]==3 && pidarray[1]==2 ) histosTH1F["hm
2rec2OSvee11i"]->Fill( mrecpi4k2 );
               //
               if(charray[0]+charray[1] == 0 && pidarray[0]==2 && pidarray[1]==3 ) histosTH1F["hm
2rec2OSvee11j"]->Fill( mreck1pi2 );
               if(charray[0]+charray[2] == 0 && pidarray[0]==2 && pidarray[2]==3 ) histosTH1F["hm
2rec2OSvee11k"]->Fill( mreck1pi3 );
               if(charray[0]+charray[3] == 0 && pidarray[0]==2 && pidarray[3]==3 ) histosTH1F["hm
2rec2OSvee11m"]->Fill( mreck1pi4 );


               /*
       std::cout << "*** charge cut 8 vee11 ***   " << std::endl;
       std::cout << "charray[0] = " << charray[0] << std::endl;
       std::cout << "charray[1] = " << charray[1] << std::endl;
       std::cout << "charray[2] = " << charray[2] << std::endl;
       std::cout << "charray[3] = " << charray[3] << std::endl;
```

```
               */

               /*
               std::cout << " ***charged pairs***   " << std::endl;
               if(charray[2]+charray[3] == 0) histosTH1F["hm2rec2OSvee11a"]->Fill( mrecpi3k4 );
               std::cout << "charray[2] = " << charray[2] << std::endl;
               std::cout << "charray[3] = " << charray[3] << std::endl;
               if(charray[1]+charray[3] == 0) histosTH1F["hm2rec2OSvee11b"]->Fill( mrecpi2k4 );
               std::cout << "charray[1] = " << charray[1] << std::endl;
               std::cout << "charray[3] = " << charray[3] << std::endl;
               if(charray[0]+charray[3] == 0) histosTH1F["hm2rec2OSvee11c"]->Fill( mrecpi1k4 );
               std::cout << "charray[0] = " << charray[0] << std::endl;
               std::cout << "charray[3] = " << charray[3] << std::endl;
               //
               if(charray[2]+charray[3] == 0) histosTH1F["hm2rec2OSvee11d"]->Fill( mreck3pi4 );
               std::cout << "charray[2] = " << charray[2] << std::endl;
               std::cout << "charray[3] = " << charray[3] << std::endl;
               if(charray[2]+charray[1] == 0) histosTH1F["hm2rec2OSvee11e"]->Fill( mreck3pi2 );
               std::cout << "charray[2] = " << charray[2] << std::endl;
               std::cout << "charray[1] = " << charray[1] << std::endl;
               if(charray[2]+charray[0] == 0) histosTH1F["hm2rec2OSvee11f"]->Fill( mreck3pi1 );
               std::cout << "charray[2] = " << charray[2] << std::endl;
               std::cout << "charray[0] = " << charray[0] << std::endl;
               //
               if(charray[0]+charray[1] == 0) histosTH1F["hm2rec2OSvee11g"]->Fill( mrecpi1k2 );
               std::cout << "charray[0] = " << charray[0] << std::endl;
               std::cout << "charray[1] = " << charray[1] << std::endl;
               if(charray[2]+charray[1] == 0) histosTH1F["hm2rec2OSvee11h"]->Fill( mrecpi3k2 );
               std::cout << "charray[2] = " << charray[2] << std::endl;
               std::cout << "charray[1] = " << charray[1] << std::endl;
               if(charray[3]+charray[1] == 0) histosTH1F["hm2rec2OSvee11i"]->Fill( mrecpi4k2 );
               std::cout << "charray[3] = " << charray[3] << std::endl;
               std::cout << "charray[1] = " << charray[1] << std::endl;
               //
               if(charray[0]+charray[1] == 0) histosTH1F["hm2rec2OSvee11j"]->Fill( mreck1pi2 );
               std::cout << "charray[0] = " << charray[0] << std::endl;
               std::cout << "charray[1] = " << charray[1] << std::endl;
               if(charray[0]+charray[2] == 0) histosTH1F["hm2rec2OSvee11k"]->Fill( mreck1pi3 );
               std::cout << "charray[0] = " << charray[0] << std::endl;
               std::cout << "charray[2] = " << charray[2] << std::endl;
               if(charray[0]+charray[3] == 0) histosTH1F["hm2rec2OSvee11m"]->Fill( mreck1pi4 );
               std::cout << "charray[0] = " << charray[0] << std::endl;
               std::cout << "charray[3] = " << charray[3] << std::endl;
               std::cout << " ***end***   " << std::endl;
               */

               //A
               if(charray[0]+charray[1] == 0 && pidarray[0]==3 && pidarray[1]==3 ) histosTH1F["hm
2rec2OS_pi1pi2vee11"]->Fill(mrecpi1pi2);
               if(charray[2]+charray[3] == 0 && pidarray[2]==3 && pidarray[3]==2 ) histosTH1F["hm
2rec2OS_pi3k4vee11"]->Fill(mrecpi3k4);
               ///histosTH2F["hm2dim2OS_pi1pi2_pi3k4vee11"]->Fill(mrecpi1pi2,mrecpi3k4);
               //
               if(charray[0]+charray[2] == 0 && pidarray[0]==3 && pidarray[2]==3 ) histosTH1F["hm
2rec2OS_pi1pi3vee11"]->Fill(mrecpi1pi3);
               if(charray[1]+charray[3] == 0 && pidarray[1]==3 && pidarray[3]==2 ) histosTH1F["hm
2rec2OS_pi2k4vee11"]->Fill(mrecpi2k4);
               ///histosTH2F["hm2dim2OS_pi1pi3_pi2k4vee11"]->Fill(mrecpi1pi3,mrecpi2k4);
               //
               if(charray[1]+charray[2] == 0 && pidarray[1]==3 && pidarray[2]==3 ) histosTH1F["hm
2rec2OS_pi2pi3vee11"]->Fill(mrecpi2pi3);
               if(charray[0]+charray[3] == 0 && pidarray[0]==3 && pidarray[3]==2 ) histosTH1F["hm
2rec2OS_pi1k4vee11"]->Fill(mrecpi1k4);
               ///histosTH2F["hm2dim2OS_pi2pi3_pi1k4vee11"]->Fill(mrecpi2pi3,mrecpi1k4);

               //B
               //if(charray[0]+charray[1] == 0) histosTH1F["hm2rec2OS_pi1pi2vee11"]->Fill(mrecpi1
pi2);
               if(charray[2]+charray[3] == 0 && pidarray[2]==2 && pidarray[3]==3 ) histosTH1F["hm
2rec2OS_k3pi4vee11"]->Fill(mreck3pi4);
               ///histosTH2F["hm2dim2OS_pi1pi2_k3pi4vee11"]->Fill(mrecpi1pi2,mreck3pi4);
               //
               if(charray[0]+charray[3] == 0 && pidarray[0]==3 && pidarray[3]==3 ) histosTH1F["hm
2rec2OS_pi1pi4vee11"]->Fill(mrecpi1pi4);
               if(charray[2]+charray[1] == 0 && pidarray[2]==2 && pidarray[1]==3 ) histosTH1F["hm
2rec2OS_k3pi2vee11"]->Fill(mreck3pi2);
               ///histosTH2F["hm2dim2OS_pi1pi4_k3pi2vee11"]->Fill(mrecpi1pi4,mreck3pi2);
               //
               if(charray[1]+charray[3] == 0 && pidarray[1]==3 && pidarray[3]==3 ) histosTH1F["hm
2rec2OS_pi2pi4vee11"]->Fill(mrecpi2pi4);
               if(charray[2]+charray[0] == 0 && pidarray[2]==2 && pidarray[0]==3 ) histosTH1F["hm
2rec2OS_k3pi1vee11"]->Fill(mreck3pi1);
               ///histosTH2F["hm2dim2OS_pi2pi4_k3pi1vee11"]->Fill(mrecpi2pi4,mreck3pi1);

               //C
               if(charray[0]+charray[1] == 0 && pidarray[0]==3 && pidarray[1]==2 ) histosTH1F["hm
2rec2OS_pi1k2vee11"]->Fill(mrecpi1k2);
               if(charray[2]+charray[4] == 0 && pidarray[2]==3 && pidarray[3]==3 ) histosTH1F["hm
2rec2OS_pi3pi4vee11"]->Fill(mrecpi3pi4);
               ///histosTH2F["hm2dim2OS_pi1k2_pi3pi4vee11"]->Fill(mrecpi1k2,mrecpi3pi4);
               //
               if(charray[2]+charray[1] == 0 && pidarray[2]==3 && pidarray[1]==2 ) histosTH1F["hm
2rec2OS_pi3k2vee11"]->Fill(mrecpi3k2);
               //if(charray[0]+charray[3] == 0) histosTH1F["hm2rec2OS_pi1pi4vee11"]->Fill(mrecpi1
pi4);
               ///histosTH2F["hm2dim2OS_pi3k2_pi1pi4vee11"]->Fill(mrecpi3k2,mrecpi1pi4);
               //
               if(charray[3]+charray[1] == 0 && pidarray[3]==3 && pidarray[1]==2 ) histosTH1F["hm
2rec2OS_pi4k2vee11"]->Fill(mrecpi4k2);
               //if(charray[0]+charray[2] == 0) histosTH1F["hm2rec2OS_pi1pi3vee11"]->Fill(mrecpi1
pi3);
               ///histosTH2F["hm2dim2OS_pi4k2_pi1pi3vee11"]->Fill(mrecpi4k2,mrecpi1pi3);

               //D
               if(charray[0]+charray[1] == 0 && pidarray[0]==2 && pidarray[1]==3 ) histosTH1F["hm
2rec2OS_k1pi2vee11"]->Fill(mreck1pi2);
               //if(charray[2]+charray[3] == 0) histosTH1F["hm2rec2OS_pi3pi4vee11"]->Fill(mrecpi3
pi4);
               ///histosTH2F["hm2dim2OS_k1pi2_pi3pi4vee11"]->Fill(mreck1pi2,mrecpi3pi4);
               //
               if(charray[0]+charray[2] == 0 && pidarray[0]==2 && pidarray[2]==3 ) histosTH1F["hm
2rec2OS_k1pi3vee11"]->Fill(mreck1pi3);
               //if(charray[1]+charray[3] == 0) histosTH1F["hm2rec2OS_pi2pi4vee11"]->Fill(mrecpi2
pi4);
               ///histosTH2F["hm2dim2OS_k1pi3_pi2pi4vee11"]->Fill(mreck1pi3,mrecpi2pi4);
               //
               if(charray[0]+charray[3] == 0 && pidarray[0]==2 && pidarray[3]==3 ) histosTH1F["hm
2rec2OS_k1pi4vee11"]->Fill(mreck1pi4);
               //if(charray[1]+charray[2] == 0) histosTH1F["hm2rec2OS_pi2pi3vee11"]->Fill(mrecpi2
pi3);
               ///histosTH2F["hm2dim2OS_k1pi3_pi2pi4vee11"]->Fill(mreck1pi3,mrecpi2pi4);

             } //end of nvtx=1 nks=1
             //} //...end of PID Pions & Kaons

          //...using PID Pions for selection=02 or 01
          //if(pidarray[0]==3 && pidarray[1]==3 && pidarray[2]==3 && pidarray[3]==3)
          //  {
             //...no primary & two Vees
             if(nvtx==0 && nks==2){
             histosTH1F["hm2rec2OSvee02"]->Fill(mrec);
                 if(charray[0]+charray[1] == 0)
                 {
               histosTH1F["hm2rec2OS_pi1pi2vee02"]->Fill(mrecpi1pi2);
               histosTH1F["hm2rec2OS_pi3pi4vee02"]->Fill(mrecpi3pi4);
               histosTH2F["hm2dim2OS_pi1pi2_pi3pi4vee02"]->Fill(mrecpi1pi2,mrecpi3pi4);
                 }else if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_pi1pi3vee02"]->Fill(mrecpi1pi3);
               histosTH1F["hm2rec2OS_pi2pi4vee02"]->Fill(mrecpi2pi4);
               histosTH2F["hm2dim2OS_pi1pi3_pi2pi4vee02"]->Fill(mrecpi1pi3,mrecpi2pi4);
                 }
               } //end of nvtx=0 nks=2

             //...no primary & 1 Vee
```

```
      if(nvtx==0 && nks==1){
      histosTH1F["hm2rec2OSvee01"]->Fill(mrec);
          if(charray[0]+charray[1] == 0)
          {
        histosTH1F["hm2rec2OS_pi1pi2vee01"]->Fill(mrecpi1pi2);
        histosTH1F["hm2rec2OS_pi3pi4vee01"]->Fill(mrecpi3pi4);
        histosTH2F["hm2dim2OS_pi1pi2_pi3pi4vee01"]->Fill(mrecpi1pi2,mrecpi3pi4);
          }else if(charray[0]+charray[2] == 0){
        histosTH1F["hm2rec2OS_pi1pi3vee01"]->Fill(mrecpi1pi3);
        histosTH1F["hm2rec2OS_pi2pi4vee01"]->Fill(mrecpi2pi4);
        histosTH2F["hm2dim2OS_pi1pi3_pi2pi4vee01"]->Fill(mrecpi1pi3,mrecpi2pi4);
          }
        } //end of nvtx=0 nks=1

      //} //...end of PID Pions
      } //...end of isKshort
    } //...end of totalcharge=0
    //AA...end of PID


    //BB...no PID Pions
    if(totcharge==0){

      if(isKshort){

      //...Luiz
      histosTH1F["hm2rec2OSveeno"]->Fill(mrecKpi);

      if(nvtx==1 && nks==1){
      histosTH1F["hm2rec2OSveeno11"]->Fill(mrecKpi);
          if(charray[0]+charray[1] == 0)
             {
        histosTH1F["hm2rec2OS_pi1pi2veeno11"]->Fill(mrecpi1pi2);
        histosTH1F["hm2rec2OS_pi3pi4veeno11"]->Fill(mrecpi3pi4);
        histosTH2F["hm2dim2OS_pi1pi2_pi3pi4veeno11"]->Fill(mrecpi1pi2,mrecpi3pi4);
             }else if(charray[0]+charray[2] == 0){
        histosTH1F["hm2rec2OS_pi1pi3veeno11"]->Fill(mrecpi1pi3);
        histosTH1F["hm2rec2OS_pi2pi4veeno11"]->Fill(mrecpi2pi4);
        histosTH2F["hm2dim2OS_pi1pi3_pi2pi4veeno11"]->Fill(mrecpi1pi3,mrecpi2pi4);
            }
        } //end of nvtx=1 nks=1

      if(nvtx==0 && nks==2){

std::cout << "*** charge cut 8 vee02 ***   " << std::endl;
std::cout << "charray[0] = " << charray[0] << std::endl;
std::cout << "charray[1] = " << charray[1] << std::endl;
std::cout << "charray[2] = " << charray[2] << std::endl;
std::cout << "charray[3] = " << charray[3] << std::endl;

      histosTH1F["hm2rec2OSveeno02"]->Fill(mrec);
          if(charray[0]+charray[1] == 0)
          {
        histosTH1F["hm2rec2OS_pi1pi2veeno02"]->Fill(mrecpi1pi2);
        histosTH1F["hm2rec2OS_pi3pi4veeno02"]->Fill(mrecpi3pi4);
        histosTH2F["hm2dim2OS_pi1pi2_pi3pi4veeno02"]->Fill(mrecpi1pi2,mrecpi3pi4);
          }else if(charray[0]+charray[2] == 0){
        histosTH1F["hm2rec2OS_pi1pi3veeno02"]->Fill(mrecpi1pi3);
        histosTH1F["hm2rec2OS_pi2pi4veeno02"]->Fill(mrecpi2pi4);
        histosTH2F["hm2dim2OS_pi1pi3_pi2pi4veeno02"]->Fill(mrecpi1pi3,mrecpi2pi4);
          }
        } //end of nvtx=0 nks=2

      if(nvtx==0 && nks==1){

std::cout << "*** charge cut 8 vee01 ***   " << std::endl;
std::cout << "charray[0] = " << charray[0] << std::endl;
std::cout << "charray[1] = " << charray[1] << std::endl;
std::cout << "charray[2] = " << charray[2] << std::endl;
std::cout << "charray[3] = " << charray[3] << std::endl;

      histosTH1F["hm2rec2OSveeno01"]->Fill(mrec);
```

```
          if(charray[0]+charray[1] == 0)
          {
        histosTH1F["hm2rec2OS_pi1pi2veeno01"]->Fill(mrecpi1pi2);
        histosTH1F["hm2rec2OS_pi3pi4veeno01"]->Fill(mrecpi3pi4);
        histosTH2F["hm2dim2OS_pi1pi2_pi3pi4veeno01"]->Fill(mrecpi1pi2,mrecpi3pi4);
          }else if(charray[0]+charray[2] == 0){
        histosTH1F["hm2rec2OS_pi1pi3veeno01"]->Fill(mrecpi1pi3);
        histosTH1F["hm2rec2OS_pi2pi4veeno01"]->Fill(mrecpi2pi4);
        histosTH2F["hm2dim2OS_pi1pi3_pi2pi4veeno01"]->Fill(mrecpi1pi3,mrecpi2pi4);
          }
        } //end of nvtx=0 nks=1

      } //...end of isKshort
    } //BB...end of totalcharge=0
  } //...end of fiducial Vees
//-----------end of cut 8


//...Luiz ...nvtx=1 or 2
//////if(nvtx!=1) continue;
//
// if(nvtx!=2) continue;

//if(nvtx!=0) continue;
//////if(nvtx!=1) continue;

if(nvtx!=1) {
 if(nvtx!=2) continue;
}
//if(nvtx!=2) continue;
//
//if(nvtx!=1) continue;

//...vertex
histosTH1F["hvtxx"]->Fill(xvtx);
histosTH1F["hvtxy"]->Fill(yvtx);
histosTH1F["hvtxz"]->Fill(zvtx);
//...Luiz
histosTH2F["hvtx2dimxy"]->Fill(xvtx,yvtx);
histosTH2F["hvtx2dimxz"]->Fill(xvtx,zvtx);
histosTH2F["hvtx2dimyz"]->Fill(yvtx,zvtx);
//...3D
////histosTH3F["hvtx3dimxyz"]->Fill(xvtx,yvtx,zvtx);

if(ntrk==4){
histosTH1F["hvtxx4"]->Fill(xvtx);
histosTH1F["hvtxy4"]->Fill(yvtx);
histosTH1F["hvtxz4"]->Fill(zvtx);
//...Luiz...2D
histosTH2F["hvtx2dimxy4"]->Fill(xvtx,yvtx);
histosTH2F["hvtx2dimxz4"]->Fill(xvtx,zvtx);
histosTH2F["hvtx2dimyz4"]->Fill(yvtx,zvtx);
//...3D
////histosTH3F["hvtx3dimxyz4"]->Fill(xvtx,yvtx,zvtx);
}

histosTH1F["hvtxchi2"]->Fill(chi2vtx);
/////histosTH1F["hvtxndof"]->Fill(ndofvtx);
histosTH2F["hntrkntrkvtx"]->Fill(ntrkvtx,ntrk);

/*

//----------------------------
//invariant mass
//...Luiz
double mrec=pipipipiRec.M();
double mrecKKKK=kkkkRec.M();
//double mrecMM=mmRec.M();
//double mrecEE=eeRec.M();
//...Luiz
//double mrecpp=ppRec.M();
```

```
// M(1,2) M(3,4) M(1,3) M(2,4)
double mrecpi1pi2=pi1pi2Rec.M();
double mrecpi3pi4=pi3pi4Rec.M();
double mrecpi1pi3=pi1pi3Rec.M();
double mrecpi2pi4=pi2pi4Rec.M();

// M(1,2) M(3,4) M(1,3) M(2,4)
double mreck1k2=k1k2Rec.M();
double mreck3k4=k3k4Rec.M();
double mreck1k3=k1k3Rec.M();
double mreck2k4=k2k4Rec.M();

//----------------------------
// xi cut
// Mmax=13000*xi_max
// 0.1 -> 1300 GeV
// 0.01 -> 130 GeV
// 0.001 -> 13 GeV

  //...Luiz...rapidity = 1/2 ln ( xi_proton_2/xi_proton_1 )
  double rapy = 0.5*TMath::Log(xiR/xiL);
  //

  */

if(fiducialRegion && fiducialRegionPt){
  histosTH1F["hxiL"]->Fill(xiL);
  histosTH1F["hxiR"]->Fill(xiR);
  histosTH1F["hm"]->Fill(mrec);
  //...Luiz
  histosTH1F["hrapy"]->Fill(rapy);
}

// last one, before Simone
if(TMath::Abs(xiL)<0.02 && TMath::Abs(xiR)<0.02);
//  if(TMath::Abs(xiL)<0.01 && TMath::Abs(xiR)<0.01);
else continue;

if(fiducialRegion && fiducialRegionPt){
  histosTH1F["hxiL2"]->Fill(xiL);
  histosTH1F["hxiR2"]->Fill(xiR);
  histosTH1F["hmxicut"]->Fill(mrec);
  //...Luiz
  histosTH1F["hrapy2"]->Fill(rapy);
}

//-------------------------------
// balance cut - py cut


// was in the first submission for 9919,9922
//  double CMSpx=pipiRec.Px();
//  double CMSpy=pipiRec.Py();
//
//...Luiz
double CMSpx=pipipipiRec.Px();
double CMSpy=pipipipiRec.Py();

histosTH2F["h2dimdpyAll"]->Fill(CMSpy,TOTEMpy);
histosTH1F["hdpyAll"]->Fill(CMSpy+TOTEMpy);

if(fiducialRegion && fiducialRegionPt){
  histosTH2F["h2dimdpy"]->Fill(CMSpy,TOTEMpy);
  histosTH1F["hdpy"]->Fill(CMSpy+TOTEMpy);

  if(diag){
    histosTH2F["h2dimdpy_diag"]->Fill(CMSpy,TOTEMpy);
    histosTH1F["hdpy_diag"]->Fill(CMSpy+TOTEMpy);
  }else{
    histosTH2F["h2dimdpy_ttbb"]->Fill(CMSpy,TOTEMpy);
    histosTH1F["hdpy_ttbb"]->Fill(CMSpy+TOTEMpy);
  }
```

```
}

// last one, before Simone
bool CTpycut = TMath::Abs(CMSpy+TOTEMpy)<0.06;
//  bool CTpycut = TMath::Abs(CMSpy+TOTEMpy)<0.03;
//  bool CTpycut = TMath::Abs(CMSpy+TOTEMpy)<0.015; // 1/4

// Robert's suggestion
//if(!CTpycut) continue;

// px for completeness
histosTH2F["h2dimdpxAll"]->Fill(CMSpx,TOTEMpx);
histosTH1F["hdpxAll"]->Fill(CMSpx+TOTEMpx);

if(fiducialRegion && fiducialRegionPt){
  histosTH2F["h2dimdpx"]->Fill(CMSpx,TOTEMpx);
  histosTH1F["hdpx"]->Fill(CMSpx+TOTEMpx);

  if(diag){
    histosTH2F["h2dimdpx_diag"]->Fill(CMSpx,TOTEMpx);
    histosTH1F["hdpx_diag"]->Fill(CMSpx+TOTEMpx);
  }else{
    histosTH2F["h2dimdpx_ttbb"]->Fill(CMSpx,TOTEMpx);
    histosTH1F["hdpx_ttbb"]->Fill(CMSpx+TOTEMpx);
  }

}

// last one, before Simone
bool CTpxcut = TMath::Abs(CMSpx+TOTEMpx)<0.15;
//  bool CTpxcut = TMath::Abs(CMSpx+TOTEMpx)<0.075;
//  bool CTpxcut = TMath::Abs(CMSpx+TOTEMpx)<0.0375; // 1/4

//------------------------------
// from now on, only 2 vertex tracks. |eta|<etaCut (=2.5)
//    if(!fiducialRegion) continue;

//------------------------------
//RP vertex

//xVtxL,xVtxR in meters, see Fig. 6 of Hubert's PAS
// so meed to multiply by 100 to get in cm

// 2012
//  double vertexResolution = 8.3e-6;
//  bool RPvertex = abs(xVtxL-xVtxR) < 3*vertexResolution;
// 2015 - Mirko
//last one, before Simone
bool RPvertex = abs(xVtxL-xVtxR) < 3e-5;
//  bool RPvertex = abs(xVtxL-xVtxR) < 1.5e-5;
//  bool RPvertex = abs(xVtxL-xVtxR) < 0.75e-5; //1/4

double xvtxT=(xVtxR+xVtxL)/2.;
//last one before Simone
bool CTvertex = -0.04<(xvtx-xvtxT*1e2) && (xvtx-xvtxT*1e2)<0.18;
//  bool CTvertex = 0.015<(xvtx-xvtxT*1e2) && (xvtx-xvtxT*1e2)<0.125;
//  bool CTvertex = 0.04<(xvtx-xvtxT*1e2) && (xvtx-xvtxT*1e2)<0.1; //1/4

// my
//  bool RPvertex = abs(xVtxL-xVtxR) < 5e-5;
//  bool RPvertex = abs(xVtxL-xVtxR) < 4e-5;

// reject if no RP vertex
//0.025mm = 25\mum
// < 3 * vertexResolution = 0.0000249               // <- mm
//       if(TMath::Abs(dx0)>0.025) continue;   // <- mum


if(fiducialRegion && fiducialRegionPt){

  histosTH2F["h2dimxVtxRL"]->Fill(xVtxL,xVtxR);
  histosTH2F["h2dimxVtxcmsR"]->Fill(xVtxR*1e2,xvtx);
  histosTH2F["h2dimxVtxcmsL"]->Fill(xVtxL*1e2,xvtx);

  histosTH2F["h2dimxVtxcmsRL"]->Fill(xvtxT*1e2,xvtx);

  if(RPvertex){
    histosTH2F["h2dimxVtxcmsR2"]->Fill(xVtxR*1e2,xvtx);
    histosTH2F["h2dimxVtxcmsL2"]->Fill(xVtxL*1e2,xvtx);
    histosTH2F["h2dimxVtxcmsRL2"]->Fill(xvtxT*1e2,xvtx);

    histosTH2F["h2dimxVtx_zVtx_CT"]->Fill(zvtx,xvtx-xvtxT*1e2);
    histosTH2F["h2dimxVtx_zVtx_C"]->Fill(zvtx,xvtx);
    histosTH2F["h2dimxVtx_zVtx_T"]->Fill(zvtx,xvtxT*1e2);

    histosTH1F["hxVtxcmsRL"]->Fill(xvtx-xvtxT*1.e2);
    if(diag)  histosTH1F["hxVtxcmsRL_diag"]->Fill(xvtx-xvtxT*1.e2);
    else      histosTH1F["hxVtxcmsRL_ttbb"]->Fill(xvtx-xvtxT*1.e2);
  }

   histosTH1F["hxVtxRL"]->Fill(xVtxR-xVtxL);
   histosTH1F["hxVtxcmsR"]->Fill(xvtx-xVtxR*1.e2);
   histosTH1F["hxVtxcmsL"]->Fill(xvtx-xVtxL*1.e2);

  if(diag){
     histosTH1F["hxVtxRL_diag"]->Fill(xVtxR-xVtxL);
     histosTH1F["hxVtxcmsR_diag"]->Fill(xvtx-xVtxR*1e2);
     histosTH1F["hxVtxcmsL_diag"]->Fill(xvtx-xVtxL*1e2);
  }else{
     histosTH1F["hxVtxRL_ttbb"]->Fill(xVtxR-xVtxL);
     histosTH1F["hxVtxcmsR_ttbb"]->Fill(xvtx-xVtxR*1e2);
     histosTH1F["hxVtxcmsL_ttbb"]->Fill(xvtx-xVtxL*1e2);
  }

}

//------------------------------
////fR: ntrk==2, nvtx==1, |eta|<etaCut
//
//...Luiz
//fR: ntrk==4, nvtx==1 or 2, |eta|<etaCut

//  totcharge=totcharge0;

if(fiducialRegion && fiducialRegionPt){

  // how many tracks with pixel if at vertex 2 tracks
  histosTH1F["hntrkntrkvtx2"]->Fill(ntrk);
  histosTH1F["hntrk2ntrkvtx"]->Fill(ntrkvtx);

  histosTH1F["hm2rec"]->Fill(mrec);
  histosTH1F["hm2recbis"]->Fill(mrec);

//...cut 1    nvtx==1 or 2

  if(totcharge==0){
    //...Luiz
    histosTH1F["hm2recOS"]->Fill(mrec);
    histosTH1F["hm2recOS2"]->Fill(mrec);
    if(diag) histosTH1F["hm2recOS_diag"]->Fill(mrec);
    else     histosTH1F["hm2recOS_ttbb"]->Fill(mrec);
  }else{
    histosTH1F["hm2recSS"]->Fill(mrec);
    if(diag) histosTH1F["hm2recSS_diag"]->Fill(mrec);
    else     histosTH1F["hm2recSS_ttbb"]->Fill(mrec);
  }

  //...cut 2
  //      if(RPvertex && CTpxcut && CTvertex){

  //Robert's suggestion...remove CTpxcut
  //if(CTpxcut){

  /////...nvtx=2
```

```
/////if(nvtx==2){

//if(ntrkvtx==2){

//...using PID Pions
if(pidarray[0]==pidPion && pidarray[1]==pidPion &&
           pidarray[2]==pidPion && pidarray[3]==pidPion)
  {

  if(totcharge==0){

    //...Luiz
    histosTH1F["hm2rec2OS"]->Fill(mrec);

    // dphi(pp) vs mrec(4pi)
    histosTH2F["dphi_proton_mrec"]->Fill( dphi_proton, mrec );

    // 12 34 13 24..using PID
    //if(charray[0]*charray[1] < 0 && pidarray[0]==pidPion && pidarray[1]==pidPion)
    //if(pidarray[0]==pidPion && pidarray[1]==pidPion &&
    //     pidarray[2]==pidPion && pidarray[3]==pidPion)
    //   {

      //...nvtx=1
    if(nvtx==1){

        histosTH1F["hm2rec2OS2"]->Fill(mrec);

        if(charray[0]+charray[1] == 0)
           {
      histosTH1F["hm2rec2OS_pi1pi2"]->Fill(mrecpi1pi2);
      histosTH1F["hm2rec2OS_pi3pi4"]->Fill(mrecpi3pi4);
      histosTH2F["hm2dim2OS_pi1pi2_pi3pi4"]->Fill(mrecpi1pi2,mrecpi3pi4);
           }else if(charray[0]+charray[2] == 0){
      histosTH1F["hm2rec2OS_pi1pi3"]->Fill(mrecpi1pi3);
      histosTH1F["hm2rec2OS_pi2pi4"]->Fill(mrecpi2pi4);
      histosTH2F["hm2dim2OS_pi1pi3_pi2pi4"]->Fill(mrecpi1pi3,mrecpi2pi4);
          }
      } //end of nvtx=1

      //...nvtx=2
    if(nvtx==2){

        if(charray[0]+charray[1] == 0)
        {
      histosTH1F["hm2rec2OS_pi1pi2v2"]->Fill(mrecpi1pi2);
      histosTH1F["hm2rec2OS_pi3pi4v2"]->Fill(mrecpi3pi4);
      histosTH2F["hm2dim2OS_pi1pi2_pi3pi4v2"]->Fill(mrecpi1pi2,mrecpi3pi4);
        }else if(charray[0]+charray[2] == 0){
      histosTH1F["hm2rec2OS_pi1pi3v2"]->Fill(mrecpi1pi3);
      histosTH1F["hm2rec2OS_pi2pi4v2"]->Fill(mrecpi2pi4);
      histosTH2F["hm2dim2OS_pi1pi3_pi2pi4v2"]->Fill(mrecpi1pi3,mrecpi2pi4);
        }
      } //end of nvtx=2

      //}//endPID

    //...Luiz
    if(diag) {
      histosTH1F["hm2rec2OS_diag"]->Fill(mrec);
      histosTH1F["hm2rec2OS_diag2"]->Fill(mrec);
      histosTH1F["hm2rec2OS_diag2varbin"]->Fill(mrec);
      histosTH1F["hm2rec2OS_diag3"]->Fill(mrec);
      histosTH1F["hm2rec2OS_diag4"]->Fill(mrec);
      histosTH1F["hm2rec2OS_diag5"]->Fill(mrec);
      // dphi_proton_mrec_diag
      histosTH2F["dphi_proton_mrec_diag"]->Fill( dphi_proton, mrec );

    // 12 34 13 24...using PID
      if(charray[0]+charray[1] == 0){
      histosTH1F["hm2rec2OS_diag_pi1pi2"]->Fill(mrecpi1pi2);
      histosTH1F["hm2rec2OS_diag_pi3pi4"]->Fill(mrecpi3pi4);
```

```
               }else{
               if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_diag_pi1pi3"]->Fill(mrecpi1pi3);
               histosTH1F["hm2rec2OS_diag_pi2pi4"]->Fill(mrecpi2pi4);
                }
               }
             }else{
               histosTH1F["hm2rec2OS_ttbb"]->Fill(mrec);
               histosTH1F["hm2rec2OS_ttbb2"]->Fill(mrec);
               histosTH1F["hm2rec2OS_ttbb2varbin"]->Fill(mrec);
               histosTH1F["hm2rec2OS_ttbb3"]->Fill(mrec);
               histosTH1F["hm2rec2OS_ttbb4"]->Fill(mrec);
               histosTH1F["hm2rec2OS_ttbb5"]->Fill(mrec);
               // dphi_proton_mrec_ttbb
               histosTH2F["dphi_proton_mrec_ttbb"]->Fill( dphi_proton, mrec );

               // 12 34 13 24...using PID
               if(charray[0]+charray[1] == 0){
               histosTH1F["hm2rec2OS_ttbb_pi1pi2"]->Fill(mrecpi1pi2);
               histosTH1F["hm2rec2OS_ttbb_pi3pi4"]->Fill(mrecpi3pi4);
               }else{
               if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_ttbb_pi1pi3"]->Fill(mrecpi1pi3);
               histosTH1F["hm2rec2OS_ttbb_pi2pi4"]->Fill(mrecpi2pi4);
                }
               }
             }//end of ttbb/diag
           }else{
             histosTH1F["hm2rec2SS"]->Fill(mrec);
             if(diag) histosTH1F["hm2rec2SS_diag"]->Fill(mrec);
             else     histosTH1F["hm2rec2SS_ttbb"]->Fill(mrec);
           } //...end of totalcharge=0

           //....???????
           if(totcharge==0 && diag){
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec2OS_diag_trkP"]->Fill(mr
ec);

             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec2OS_diag_trkM"]->Fill(mr
ec);
           }
           if(totcharge==0 && !diag){
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec2OS_ttbb_trkP"]->Fill(mr
ec);

             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec2OS_ttbb_trkM"]->Fill(mr
ec);
           }

           //...Luiz  ??????
           if(totcharge==0 && diag){
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) histosTH1F["hm2rec2O
S_diag_pypxP"]->Fill(mrec);
             else histosTH1F["hm2rec2OS_diag_pypxM"]->Fill(mrec);
             //...Luiz
             int pypx=0;
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) pypx=1;
             else pypx=0;
             //...?????
             if(mrec>=1.65 && mrec<=1.75) cout<<"scan2OSdiag: "<<run<<" "<<LS<<" "<<evt<<" "<<mre
c<<" "<<pypx<<endl;
           }
           //...Luiz
           if(totcharge==0 && !diag){
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) histosTH1F["hm2rec2O
S_ttbb_pypxP"]->Fill(mrec);
             else  histosTH1F["hm2rec2OS_ttbb_pypxM"]->Fill(mrec);
           }

           //-------------------------
           //...Luiz
           if(totcharge==0 && diag){
             histosTH1F["hm2recPPPP"]->Fill(mrec);
             //histosTH1F["hm2recKKKK"]->Fill(mrecKKKK);
```

```
      //histosTH1F["hm2recMM"]->Fill(mrecMM);
      //histosTH1F["hm2recEE"]->Fill(mrecEE);
      //...Luiz
      //histosTH1F["hm2recpp"]->Fill(mrecpp);
    }

    //...Luiz : dphi new definition
    if(totcharge==0){
      histosTH1F["hphiL"]->Fill(TOTEMphiL);
      histosTH1F["hphiR"]->Fill(TOTEMphiR);
      histosTH1F["hdphi"]->Fill(TOTEMdphi);
      if(diag) histosTH1F["hdphi_diag"]->Fill(TOTEMdphi);
      else     histosTH1F["hdphi_ttbb"]->Fill(TOTEMdphi);
    }

}//00...end of PID Pions
  //...Luiz
  //}//ntrkvtx==2,4
  //}...end of Robert's suggestion


   //...using PID Kaons
   if(pidarray[0]==pidKaon && pidarray[1]==pidKaon &&
      pidarray[2]==pidKaon && pidarray[3]==pidKaon)
  // {
  ////if(pidarray[0]==2 && pidarray[1]==2 &&
  ////    pidarray[2]==2 && pidarray[3]==2)
   {

    if(totcharge==0){

      //...Luiz
      /*
      histosTH1F["hm2rec2OS"]->Fill(mrec);
      histosTH1F["hm2rec2OS2"]->Fill(mrec);
      */
      // dphi(pp) vs mrec(4pi)
      ////histosTH2F["dphi_proton_mrec"]->Fill( dphi_proton, mrec );

      // 12 34 13 24..using PID
      //if(charray[0]*charray[1] < 0 && pidarray[0]==pidPion && pidarray[1]==pidPion)
      //if(pidarray[0]==pidPion && pidarray[1]==pidPion &&
      //     pidarray[2]==pidPion && pidarray[3]==pidPion)
      //  {

        //...nvtx=1
        if(nvtx==1){
          if(charray[0]+charray[1] == 0)
             {
        histosTH1F["hm2rec2OS_k1k2"]->Fill(mreck1k2);
        histosTH1F["hm2rec2OS_k3k4"]->Fill(mreck3k4);
        histosTH2F["hm2dim2OS_k1k2_k3k4"]->Fill(mreck1k2,mreck3k4);
         }else if(charray[0]+charray[2] == 0){
        histosTH1F["hm2rec2OS_k1k3"]->Fill(mreck1k3);
        histosTH1F["hm2rec2OS_k2k4"]->Fill(mreck2k4);
        histosTH2F["hm2dim2OS_k1k3_k2k4"]->Fill(mreck1k3,mreck2k4);
          }
        } //end of nvtx=1

        //...nvtx=2
        if(nvtx==2){
          if(charray[0]+charray[1] == 0)
             {
        histosTH1F["hm2rec2OS_k1k2v2"]->Fill(mreck1k2);
        histosTH1F["hm2rec2OS_k3k4v2"]->Fill(mreck3k4);
        histosTH2F["hm2dim2OS_k1k2_k3k4v2"]->Fill(mreck1k2,mreck3k4);
         }else if(charray[0]+charray[2] == 0){
        histosTH1F["hm2rec2OS_k1k3v2"]->Fill(mreck1k3);
        histosTH1F["hm2rec2OS_k2k4v2"]->Fill(mreck2k4);
        histosTH2F["hm2dim2OS_k1k3_k2k4v2"]->Fill(mreck1k3,mreck2k4);
          }
        } //end of nvtx=2
```

```
             //...Luiz
             if(diag) {
               // 12 34 13 24...using PID
               if(charray[0]+charray[1] == 0){
               histosTH1F["hm2rec2OS_diag_k1k2"]->Fill(mreck1k2);
               histosTH1F["hm2rec2OS_diag_k3k4"]->Fill(mreck3k4);
               }else{
               if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_diag_k1k3"]->Fill(mreck1k3);
               histosTH1F["hm2rec2OS_diag_k2k4"]->Fill(mreck2k4);
                }
               }
             }else{
               // 12 34 13 24...using PID
               if(charray[0]+charray[1] == 0){
               histosTH1F["hm2rec2OS_ttbb_k1k2"]->Fill(mreck1k2);
               histosTH1F["hm2rec2OS_ttbb_k3k4"]->Fill(mreck3k4);
               }else{
               if(charray[0]+charray[2] == 0){
               histosTH1F["hm2rec2OS_ttbb_k1k3"]->Fill(mreck1k3);
               histosTH1F["hm2rec2OS_ttbb_k2k4"]->Fill(mreck2k4);
                }
               }
             }//ttbb

           }//totalcharge=0

           //...Luiz
           if(totcharge==0 && diag){
             histosTH1F["hm2recKKKK"]->Fill(mrecKKKK);
           }
         }//...end PID Kaons
          //-----------end of cut 2

         //...cut 3
         //.... OS:totcharge==0 SS:totcharge!=0
         if(RPvertex && CTpxcut){
           if(totcharge==0){
             histosTH1F["hm2rec3OS"]->Fill(mrec);
             if(diag) histosTH1F["hm2rec3OS_diag"]->Fill(mrec);
             else     histosTH1F["hm2rec3OS_ttbb"]->Fill(mrec);
           }else{
             histosTH1F["hm2rec3SS"]->Fill(mrec);
             if(diag) histosTH1F["hm2rec3SS_diag"]->Fill(mrec);
             else     histosTH1F["hm2rec3SS_ttbb"]->Fill(mrec);
           }

           //...Luiz
           //             OS: pi+pi+ or pi-pi-    ?????
           if(totcharge==0 && diag){
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec3OS_diag_trkP"]->Fill(mr
ec);

             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec3OS_diag_trkM"]->Fill(mr
ec);
           }
           if(totcharge==0 && !diag){
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec3OS_ttbb_trkP"]->Fill(mr
ec);

             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec3OS_ttbb_trkM"]->Fill(mr
ec);
           }
           //...Luiz
           if(totcharge==0 && diag){
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) histosTH1F["hm2rec3O
S_diag_pypxP"]->Fill(mrec);
             else histosTH1F["hm2rec3OS_diag_pypxM"]->Fill(mrec);
           }
           if(totcharge==0 && !diag){
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) histosTH1F["hm2rec3O
S_ttbb_pypxP"]->Fill(mrec);
             else  histosTH1F["hm2rec3OS_ttbb_pypxM"]->Fill(mrec);
           }

         }

         //...cut 4
         //.... OS:totcharge==0 SS:totcharge!=0
         //    if(RPvertex && CTpxcut && nvtx==1 && CTvertex){
         if(RPvertex && CTpxcut && CTvertex && TMath::Abs(zvtx)<5.){ // core

           if(totcharge==0){
             histosTH1F["hm2rec4OS"]->Fill(mrec);
             if(diag) histosTH1F["hm2rec4OS_diag"]->Fill(mrec);
             else     histosTH1F["hm2rec4OS_ttbb"]->Fill(mrec);
           }else{
             histosTH1F["hm2rec4SS"]->Fill(mrec);
             if(diag) histosTH1F["hm2rec4SS_diag"]->Fill(mrec);
             else     histosTH1F["hm2rec4SS_ttbb"]->Fill(mrec);
           }
           //...Luiz
           //             OS: pi+pi+ or pi-pi-    ?????
           if(totcharge==0 && diag){
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec4OS_diag_trkP"]->Fill(mr
ec);
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec4OS_diag_trkM"]->Fill(mr
ec);
           }
           //  ?????
           if(totcharge==0 && !diag){
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec4OS_ttbb_trkP"]->Fill(mr
ec);
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec4OS_ttbb_trkM"]->Fill(mr
ec);
           }
           //...Luiz
           if(totcharge==0 && diag){
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) histosTH1F["hm2rec4O
S_diag_pypxP"]->Fill(mrec);
             else histosTH1F["hm2rec4OS_diag_pypxM"]->Fill(mrec);
           }
           //...Luiz
           if(totcharge==0 && !diag){
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) histosTH1F["hm2rec4O
S_ttbb_pypxP"]->Fill(mrec);
             else  histosTH1F["hm2rec4OS_ttbb_pypxM"]->Fill(mrec);
           }

         }

         //...cut 5   nvtx==1 or 2
         //.... OS:totcharge==0 SS:totcharge!=0
         //    if(RPvertex && CTpxcut && nvtx==1 && CTvertex && TMath::Abs(zvtx)>5.){//tails
         //    if(RPvertex && CTpxcut && nvtx==1 && CTvertex){
         // no dpx cut applied
         if(RPvertex && CTvertex){

           if(totcharge==0){
             histosTH1F["hm2rec5OS"]->Fill(mrec);
             if(diag) histosTH1F["hm2rec5OS_diag"]->Fill(mrec);
             else     histosTH1F["hm2rec5OS_ttbb"]->Fill(mrec);
           }else{
             histosTH1F["hm2rec5SS"]->Fill(mrec);
             if(diag) histosTH1F["hm2rec5SS_diag"]->Fill(mrec);
             else     histosTH1F["hm2rec5SS_ttbb"]->Fill(mrec);
           }
           //...Luiz
           //             OS: pi+pi+ or pi-pi-    ?????
           if(totcharge==0 && diag){
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec5OS_diag_trkP"]->Fill(mr
ec);
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec5OS_diag_trkM"]->Fill(mr
ec);
           }
           //   ?????
           if(totcharge==0 && !diag){
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec5OS_ttbb_trkP"]->Fill(mr
ec);
             if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec5OS_ttbb_trkM"]->Fill(mr
ec);
           }
           //...Luiz
           if(totcharge==0 && diag){
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) histosTH1F["hm2rec5O
S_diag_pypxP"]->Fill(mrec);
             else histosTH1F["hm2rec5OS_diag_pypxM"]->Fill(mrec);
           }
           //...Luiz
           if(totcharge==0 && !diag){
             if(TMath::Abs(pipipipiRec.Py()) > TMath::Abs(pipipipiRec.Px())) histosTH1F["hm2rec5O
S_ttbb_pypxP"]->Fill(mrec);
             else  histosTH1F["hm2rec5OS_ttbb_pypxM"]->Fill(mrec);
           }

         }

         //...cut 6    nvtx==1 or 2
         //.... OS:totcharge==0 SS:totcharge!=0
         //    if(RPvertex && CTpxcut && nvtx==1){
         if(RPvertex && CTpxcut && CTvertex){

           double etaCut2=1.5;
           //...Luiz
           if(TMath::Abs(pi1.Eta())<etaCut2 && TMath::Abs(pi2.Eta())<etaCut2 &&
              TMath::Abs(pi3.Eta())<etaCut2 && TMath::Abs(pi4.Eta())<etaCut2 ){
             if(totcharge==0){
               histosTH1F["hm2rec6OS"]->Fill(mrec);
               if(diag) histosTH1F["hm2rec6OS_diag"]->Fill(mrec);
               else     histosTH1F["hm2rec6OS_ttbb"]->Fill(mrec);
             }else{
               histosTH1F["hm2rec6SS"]->Fill(mrec);
               if(diag) histosTH1F["hm2rec6SS_diag"]->Fill(mrec);
               else     histosTH1F["hm2rec6SS_ttbb"]->Fill(mrec);
             }
             //...Luiz           ?????
           //             OS: pi+pi+ or pi-pi-    ?????
             if(totcharge==0 && diag){
               if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec6OS_diag_trkP"]->Fill(
mrec);
               if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec6OS_diag_trkM"]->Fill(
mrec);
             }
             //...Luiz           ?????
             if(totcharge==0 && !diag){
               if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()>0) histosTH1F["hm2rec6OS_ttbb_trkP"]->Fill(
mrec);
               if(pi1.Py()*pi2.Py()*pi3.Py()*pi4.Py()<0) histosTH1F["hm2rec6OS_ttbb_trkM"]->Fill(
mrec);
             }

           }
         }

         //...cut 7
         if(diag && RPvertex && CTpxcut){

           if(totcharge==0) histosTH1F["hm2recHFvetoOS"]->Fill(mrec);
           else histosTH1F["hm2recHFvetoSS"]->Fill(mrec);

           //-------------
           //...Luiz   ????
           if(pi1.Pt()>0.45 && pi2.Pt()>0.45 && pi3.Pt()>0.45 && pi4.Pt()>0.45){
             if(totcharge==0) histosTH1F["hm2rec45OS"]->Fill(mrec);
             else histosTH1F["hm2rec45SS"]->Fill(mrec);

             double etaCut2=1.5;
```

```
       //...Luiz    ??????
       if(TMath::Abs(pi1.Eta())<etaCut2 && TMath::Abs(pi2.Eta())<etaCut2 &&
          TMath::Abs(pi3.Eta())<etaCut2 && TMath::Abs(pi4.Eta())<etaCut2){
         if(totcharge==0) histosTH1F["hm2rec4515OS"]->Fill(mrec);
         else histosTH1F["hm2rec4515SS"]->Fill(mrec);
       }
     }

   }

   if(diag && totcharge==0 &&RPvertex && CTpxcut){
     //      if(run==259237 && LS>=78 && LS<=100) histosTH1F["hm2rec9919"]->Fill(mrec);
     if(run==259237 && LS>=78 && LS<=100) histosTH1F["hm2rec9922"]->Fill(mrec);
     if(run==259237 && LS>=432 && LS<=576) histosTH1F["hm2rec9922"]->Fill(mrec);
     if(run==259385 && LS>=253 && LS<=538) histosTH1F["hm2rec9971"]->Fill(mrec);
     if(run==259388 && LS>=369 && LS<=747) histosTH1F["hm2rec9978"]->Fill(mrec);
   }

 } //...end of fiducial cut=1


 //-----------------------------
 // track variables
//...Luiz
if(ntrk==4 && CTpycut && CTpxcut && RPvertex){
   if(totcharge==0 && diag){

     histosTH1F["hptRes"]->Fill(pipipipiRec.Pt());
     histosTH1F["hetaRes"]->Fill(pipipipiRec.Eta());
     histosTH1F["hphiRes"]->Fill(pipipipiRec.Phi());

     if(charray[0]>0){
       histosTH1F["hptP"]->Fill(pi1.Pt());
       histosTH1F["hetaP"]->Fill(pi1.Eta());
       histosTH1F["hphiP"]->Fill(pi1.Phi());
     }else{
       histosTH1F["hptM"]->Fill(pi1.Pt());
       histosTH1F["hetaM"]->Fill(pi1.Eta());
       histosTH1F["hphiM"]->Fill(pi1.Phi());
     }

     if(charray[1]>0){
       histosTH1F["hptP"]->Fill(pi2.Pt());
       histosTH1F["hetaP"]->Fill(pi2.Eta());
       histosTH1F["hphiP"]->Fill(pi2.Phi());
     }else{
       histosTH1F["hptM"]->Fill(pi2.Pt());
       histosTH1F["hetaM"]->Fill(pi2.Eta());
       histosTH1F["hphiM"]->Fill(pi2.Phi());
     }
     //...Luiz
     if(charray[2]>0){
       histosTH1F["hptP"]->Fill(pi3.Pt());
       histosTH1F["hetaP"]->Fill(pi3.Eta());
       histosTH1F["hphiP"]->Fill(pi3.Phi());
     }else{
       histosTH1F["hptM"]->Fill(pi3.Pt());
       histosTH1F["hetaM"]->Fill(pi3.Eta());
       histosTH1F["hphiM"]->Fill(pi3.Phi());
     }
     //...Luiz
     if(charray[3]>0){
       histosTH1F["hptP"]->Fill(pi4.Pt());
       histosTH1F["hetaP"]->Fill(pi4.Eta());
       histosTH1F["hphiP"]->Fill(pi4.Phi());
     }else{
       histosTH1F["hptM"]->Fill(pi4.Pt());
       histosTH1F["hetaM"]->Fill(pi4.Eta());
       histosTH1F["hphiM"]->Fill(pi4.Phi());
     }

     histosTH1F["hvtxchi2fin"]->Fill(chi2vtx);
```

```
           //...Luiz
           histosTH1F["hchi2fin"]->Fill(chi2array[0]);
           histosTH1F["hchi2fin"]->Fill(chi2array[1]);
           histosTH1F["hchi2fin"]->Fill(chi2array[2]);
           histosTH1F["hchi2fin"]->Fill(chi2array[3]);
           histosTH1F["hd0fin"]->Fill(d0array[0]);
           histosTH1F["hd0fin"]->Fill(d0array[1]);
           histosTH1F["hd0fin"]->Fill(d0array[2]);
           histosTH1F["hd0fin"]->Fill(d0array[3]);
           histosTH1F["hdzfin"]->Fill(dzarray[0]);
           histosTH1F["hdzfin"]->Fill(dzarray[1]);
           histosTH1F["hdzfin"]->Fill(dzarray[2]);
           histosTH1F["hdzfin"]->Fill(dzarray[3]);

           int nclustersOSdiag=  sipixelcluster_coll->size();
           int nclusters2OSdiag=  sistripcluster_coll->nStripClusters;
           histosTH1F["hnclustersOSdiag"]->Fill(nclustersOSdiag);
           histosTH1F["hnclusters2OSdiag"]->Fill(nclusters2OSdiag);

         }//...end of totalcharge=0 and diag
      }//...end of track variables

    } // End of loop over events in a file

    // Close current file
    file->Close();

  } // End of loop over files


  // Output file
  TFile* output = new TFile(outputFileName.c_str(),"RECREATE");
  output->cd();

  for(map<string,TH1F*>::iterator it_histo = histosTH1F.begin();
                                  it_histo != histosTH1F.end(); ++it_histo)
     (*it_histo).second->Write();
  for(map<string,TH2F*>::iterator it_histo = histosTH2F.begin();
                                  it_histo != histosTH2F.end(); ++it_histo)
     (*it_histo).second->Write();

  output->Close();

//  fout.close();

}
```