中国科学院高能物理研究所

本周汇报

Done:	绘制 shape 图
Ongoing:	学习 Signal region definition
To Do:	完成 Signal region definition

沈羽 2022.8.22

QCD.root

获得 QCD.root

```
   ------------------------------------------------------------------
  | Welcome to ROOT 6.20/06                        https://root.cern |
  | (c) 1995-2020, The ROOT Team; conception: R. Brun, F. Rademakers |
  | Built for linuxx8664gcc on Jun 10 2020, 06:10:57                 |
  | From tags/v6-20-06@v6-20-06                                      |
  | Try '.help', '.demo', '.license', '.credits', '.quit'/'.q'       |
   ------------------------------------------------------------------

root [0]
Attaching file QCD.root as _file0...
(TFile *) 0x12a1320
root [1] .ls
TFile**         QCD.root
 TFile*         QCD.root
  KEY: TTree    Staus_Nominal;1 SmallTree for fast analysis
  KEY: TH1F     MET_low;1       MET_low
  KEY: TH1F     MT2_low;1       MT2_low
  KEY: TH1F     tau1Pt_low;1    tau1Pt_low
  KEY: TH1F     tau2Pt_low;1    tau2Pt_low
  KEY: TH1F     tau1Mt_low;1    tau1Mt_low
  KEY: TH1F     tau2Mt_low;1    tau2Mt_low
  KEY: TH1F     dPhitt_low;1    dPhitt_low
  KEY: TH1F     dRtt_low;1      dRtt_low
  KEY: TH1F     Mtt_12_low;1    Mtt_12_low
```

```
root -l -b -q 'draw.C("MET_low")' &
root -l -b -q 'draw.C("MT2_low")' &
root -l -b -q 'draw.C("tau1Pt_low")' &
root -l -b -q 'draw.C("tau2Pt_low")' &
root -l -b -q 'draw.C("tau1Mt_low")' &
root -l -b -q 'draw.C("tau2Mt_low")' &
root -l -b -q 'draw.C("dPhitt_low")' &
root -l -b -q 'draw.C("dRtt_low")' &
root -l -b -q 'draw.C("Mtt_12_low")' &
```

Draw.C

Run Draw.C

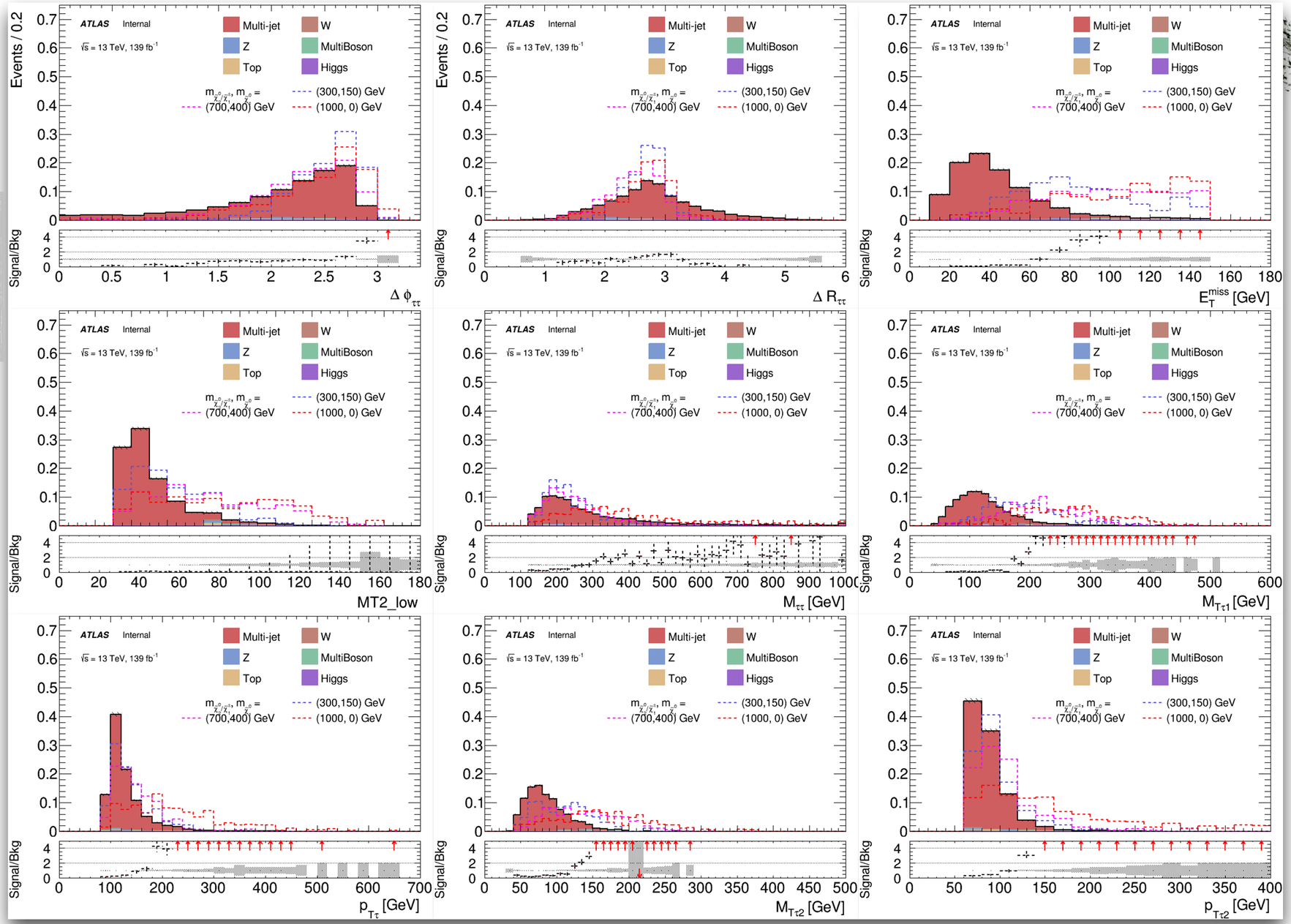
```
#include "Plotter/AtlasUtils.C"
#include "Plotter/AtlasStyle.C"
#include "Plotter/Plotter.h"
#include <iostream>
#include <math.h>
#include <vector>
#include <fstream>
#include <string>
enum{
  QCDColor = 46,
  MultiBosonColor=30,
  WColor=45,
  ZColor=38,
  ttColor=42,
  HiggsColor=kViolet-8,
  sig1Color=9,
  sig2Color=6,
  sig3Color=2,
  dataColor=kBlack,
  arrowColor=2
};
void draw(const std::string name) {
    std::string folder = "../root/";
    SetAtlasStyle();
    Plotter* mPlotter = new Plotter(Plotter::USE_COM_PAD,Plotter::NO_LOG_Y);
    TH1F* histHiggs = (TH1F*)Plotter::getHist(folder + "Higgs.root",  name,Plotter::Block,HiggsColor);
    TH1F* histTop = (TH1F*)Plotter::getHist(folder + "Top.root", name,Plotter::Block,ttColor);
    TH1F* histMultiBoson = (TH1F*)Plotter::getHist(folder + "MultiBoson.root",name,Plotter::Block,MultiBosonColor);
    TH1F* histZ = (TH1F*)Plotter::getHist(folder + "Z.root", name,Plotter::Block,ZColor);
    TH1F* histW = (TH1F*)Plotter::getHist(folder + "W.root",name,Plotter::Block,WColor);
    TH1F* histQCD = (TH1F*)Plotter::getHist(folder + "QCD.root", name,Plotter::Block,QCDColor);
    TH1F* histsig = (TH1F*)Plotter::getHist(folder + "300_150.root",name,Plotter::Dash,sig1Color);
    TH1F* histsig2 = (TH1F*)Plotter::getHist(folder + "700_400.root", name,Plotter::Dash,sig2Color);
    TH1F* histsig3 = (TH1F*)Plotter::getHist(folder + "1000_0.root", name,Plotter::Dash,sig3Color);
    histsig->Scale(1/histsig->Integral());
    histsig2->Scale(1/histsig2->Integral());
    histsig3->Scale(1/histsig3->Integral());
    TH1F*histBkg = (TH1F*)histHiggs->Clone();
    histBkg->Add(histTop);
    histBkg->Add(histMultiBoson);
    histBkg->Add(histZ);
    histBkg->Add(histW);
    histBkg->Add(histQCD);
    histBkg->Add(histQCD);
    histHiggs->Scale(1/histBkg->Integral());
    histTop->Scale(1/histBkg->Integral());
    histMultiBoson->Scale(1/histBkg->Integral());
    histZ->Scale(1/histBkg->Integral());
    histW->Scale(1/histBkg->Integral());
    histQCD->Scale(1/histBkg->Integral());
    std::cout<<"Scale"<<endl;
    mPlotter->addBkgHist(histHiggs);
    mPlotter->addBkgHist(histTop);
    mPlotter->addBkgHist(histMultiBoson);
    mPlotter->addBkgHist(histZ);
    mPlotter->addBkgHist(histW);
    mPlotter->addBkgHist(histQCD);
    mPlotter->addSigHist(histsig);
    mPlotter->addSigHist(histsig2);
    mPlotter->addSigHist(histsig3);
    mPlotter->setCompareHist(histsig,mPlotter->getBkgHist());
    mPlotter->setCompareHist(histsig2,mPlotter->getBkgHist());
    mPlotter->setCompareHist(histsig3,mPlotter->getBkgHist());
    mPlotter->addUpperText(0.17,0.76+0.05,"#sqrt{s} = 13 TeV, 139 fb^{-1}" , 0.04);
    mPlotter->addATLASText(0.17, 0.86+0.05, 0.04);
    mPlotter->addUpperText(0.17+0.10,0.86+0.05,"Internal", 0.04);
    TLegend *legendMC = mPlotter->addUpperLegend(0.5,0.68,0.92,0.95);
    legendMC->SetNColumns(2);
    legendMC->AddEntry(histQCD,"Multi-jet","f");
    legendMC->AddEntry(histW, "W", "f");
    legendMC->AddEntry(histZ, "Z","f");
    legendMC->AddEntry(histMultiBoson, "MultiBoson","f");
    legendMC->AddEntry(histTop, "Top","f");
    legendMC->AddEntry(histHiggs, "Higgs","f");
    TLegend *legendsig = mPlotter->addUpperLegend(0.4,0.45,0.92,0.65);
```

```
int rebinNum = 1;
if( name.find("_N_1") != string::npos || name.find("Mtt") != string::npos ){
    rebinNum = 2;
} else if( name.find("MET") != string::npos ){
    rebinNum = 3;
}else{
    rebinNum = 5;
}

std::string xaxis = name;
std::string yTitile = "Proportion";
if(name.find("tau1pt") != string::npos || name.find("tau1Pt") != string::npos){
    xaxis = "p_{T#tau} [GeV]";
} else if (name.find("tau2pt") != string::npos || name.find("tau2Pt") != string::npos){
    xaxis = "p_{T#tau2} [GeV]";
} else if (name.find("mu1Pt") != string::npos){
    xaxis = "p_{T#mu} [GeV]";
} else if (name.find("Mtt_high_N_1") != string::npos){
    mPlotter->setZnProperty(Plotter::FLIP_DIC,0.3);
    xaxis = "M_{#tau#tau} [GeV]";
} else if (name.find("Mtt") != string::npos){
    xaxis = "M_{#tau#tau} [GeV]";
} else if (name.find("Evt_MET") != string::npos){
    xaxis = "E_{T}^{miss} [GeV]";
} else if (name.find("Rtt") != string::npos){
    xaxis = "#Delta R_{#tau#tau}";
    mPlotter->autosetUpperYTitile(" ");
    mPlotter->setZnProperty(Plotter::FLIP_DIC,0.3);
} else if (name.find("topTagger") != string::npos){
    mPlotter->setZnProperty(Plotter::FLIP_DIC,0.3);
} else if (name.find("bTag_N_1") != string::npos){
    mPlotter->setZnProperty(Plotter::FLIP_DIC,0.3);
} else if (name.find("Njet") != string::npos){
    mPlotter->setZnProperty(Plotter::FLIP_DIC,0.3);
} else if (name.find("Ht_Jet") != string::npos){
    mPlotter->setZnProperty(Plotter::FLIP_DIC,0.3);
} else if (name.find("Phitt") != string::npos){
    xaxis = "#Delta #phi_{#tau#tau}";
    mPlotter->autosetUpperYTitile(" ");
    mPlotter->setZnProperty(Plotter::FLIP_DIC,0.3);
} else if (name.find("dRtm") != string::npos){
    xaxis = "#Delta R_{#tau#mu}";
} else if (name.find("Mll") != string::npos){
    xaxis = "M_{#tau#mu} [GeV]";
} else if (name.find("MT12") != string::npos){
    xaxis = "m_{T#tau}+m_{T#mu} [GeV]";
} else if (name.find("mT2") != string::npos){
    xaxis = "m_{T2} [GeV]";
} else if (name.find("met_sig_tj") != string::npos){
    xaxis = "E_{T}^{miss} Sig_{taujet} [#sqrt{GeV}]";
} else if (name.find("dPhitm") != string::npos){
    xaxis = "#Delta #phi_{#tau#mu}";
} else if (name.find("meff") != string::npos){
    xaxis = "M_{eff} [GeV]";
} else if (name.find("muMt") != string::npos || name.find("mu1Mt") != string::npos){
    xaxis = "M_{T#mu} [GeV]";
} else if (name.find("tau1Mt") != string::npos){
    xaxis = "M_{T#tau1} [GeV]";
} else if (name.find("tau2Mt") != string::npos){
    xaxis = "M_{T#tau2} [GeV]";
} else if (name.find("lepMt") != std::string::npos){
    xaxis = "M_{Tl} [GeV]";
} else if (name.find("lepPt") != std::string::npos){
    xaxis = "p_{Tl} [GeV]";
} else if (name.find("MET_low") != std::string::npos){
    xaxis = "E_{T}^{miss} [GeV]";
    mPlotter->setXRange(0,180);
} else if (name.find("MET_high") != std::string::npos){
    xaxis = "E_{T}^{miss} [GeV]";
    mPlotter->setXRange(100,500);
} else if (name.find("MTsum") != std::string::npos){
    xaxis = "m_{Tsum} [GeV]";
}
mPlotter->setLowerRange(0,5);
mPlotter->setUpperRange(0,0.75);
```

shape图

dPhitt	dRtt	MET
MT2	Mtt_12	tau1Mt
tau1Pt	tau2Mt	tau2Pt



下周工作安排

2022.8.22-2022.8.28

1. 阅读文献
2. 学习完成 Signal region definition
3. 复习备考

Question?

谢 谢 ！