Rivet package for Drell–Yan dimuon production in proton-lead collisions at $\sqrt{s_{NN}} = 8.16$ TeV

M. H. Karam Sichani[1]
Supervisors: S. Taheri Monfared, H. Jung

[1] Isfahan University of Technology, Isfahan, Iran

September 22, 2022

Abstract

We present the Rivet package and its validation for Drell–Yan dimuon production in proton-lead collisions at $\sqrt{s_{NN}} = 8.16$ TeV

1 Introduction

The differential cross-section for Drell–Yan dimuon production in proton-lead collisions at $\sqrt{s_{NN}} = 8.16$TeV have been measured with data acquired by the CMS collaboration at the LHC during the year 2016 [1]. We provide a plugin for the Rivet [2] package to allow the comparison of the prediction from any event generator supporting the HepMC [3, 4] format with this measurement. In this note we present the plugin and show its validation. The Rivet package identifier, that is based on the Inspire record identifier, is `CMS_2021_1849180`.

2 Measurement and Rivet package description

Rivet packages include an information file that describes the measurement and provide relevant information for comparison with a prediction.

3 Rivet plugin

The code producing the histograms to compare with the measurement is written in the C++ language using the libraries provided by the Rivet framework. We checked especially the distribution of ϕ^* as variable that can help us with prediction of small-Q_T region and resummation techniques. The variable ϕ^* is defined as:

$$\phi^* \equiv \tanh\left(\frac{\pi - \Delta\phi}{2}\right) \sin\theta^*_\eta.$$

where$\nabla\phi$ is the opening angle between the leptons, defined as the difference of their azimuthal angles in the plane transverse to the beam axis, and θ^{*}_{η} is related to the emission angle of the dilepton system with respect to the beam. The Z-boson candidates were slected by ZFinder projection from Rivet toolkit. It selects leptons in final state that Z-boson decays to.

```
 for(const Particle& zmumu : zmumus) {


Particles leptons = sortBy(ZmumuFinder.constituents(), cmpMomByPt);
  if (leptons[0].pt()<15 or leptons[1].pt()<10 or abs(deltaEta(leptons[0],leptons[1]))>2.4) {continue
const FourMomentum lminus = leptons[0].charge() < 0 ? leptons[0].momentum() : leptons[1].momentum();
const FourMomentum lplus = leptons[0].charge() < 0 ? leptons[1].momentum() : leptons[0].momentum();

const double phi_acop = M_PI - deltaPhi(lminus, lplus);
const double costhetastar = tanh((lminus.eta()-lplus.eta())/2.0);
const double sin2thetastar = (costhetastar <= 1) ? 1.0 - sqr(costhetastar) : 0;
const double phistar = tan(phi_acop/2.0) * sqrt(sin2thetastar);


}
```

4 Validation

In order to validate the correctness of the routine code and of the provided data, we select one of the Monte-Carlo generator used in the measurement-prediction comparison performed in the publication. We run POWHEG to produce LHE files then we use PYTHIA8 for parton showering as described in the publication. The parton density parametrization of CT14 is used.

In Fig.1 and Fig.2 we show the differential fiducial cross section as measured in publication and compare them with our prediction using Rivet plugin for mass distribution. Same as publication we have some cuts for our calculations: $|\eta| < 2.4$,$P_T > 15GeV$ and $15 < m_{\mu\mu} < 600GeV$.

In Fig.3 and Fig.4 we show the differential cross section as measured in publication and compare them with our prediction using Rivet plugin for mass distribution.

In Fig.5 we show the contribution of each incoming flavor.

In Fig.6 and Fig.7 we show the complete set of our results that obtained by using Rivet plugin, but these results were produced with different Monte Carlo generator (we used madgraph5_aMCatNLO and Cascade3).

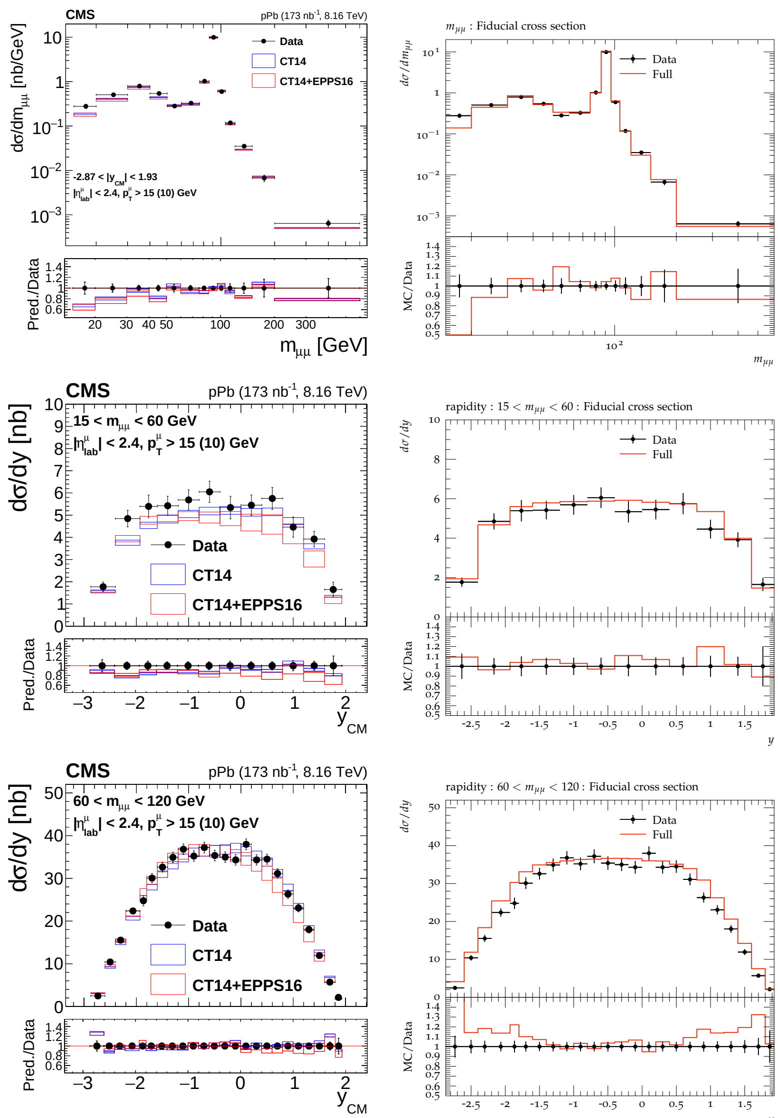

Figure 1: Differential fiducial cross section (without the acceptance correction) for the DY process measured in the muon channel, as a function of the dimuon invariant mass and rapidity in the centre-of-mass frame for $15 < m < 60$GeV and $60 < m < 120$GeV . The error bars on the data represent the quadratic sum of the statistical and systematic uncertainties. The results of [1] are on the left and ours on the right.

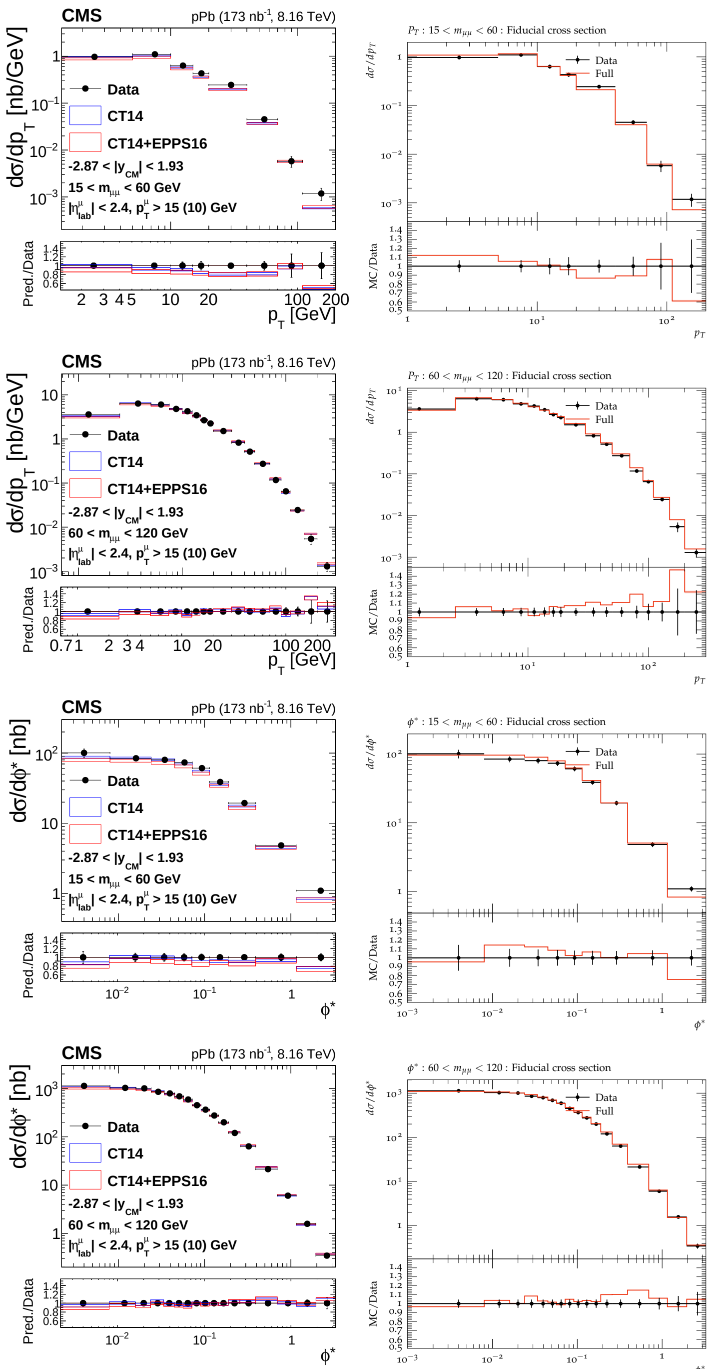

Figure 2: Differential fiducial cross sections (without the acceptance correction) for the DY process measured in the muon channel, as functions of $p_{\mathrm{T}}$and ϕ^*, for $15 < m < 60$GeV and $60 < m < 120$GeV. The first bin of the $p_{\mathrm{T}}$and ϕ^* measurements starts at 0. The error bars on the data represent the quadratic sum of the statistical and systematic uncertainties. The results of [1] are on the left and ours on the right.

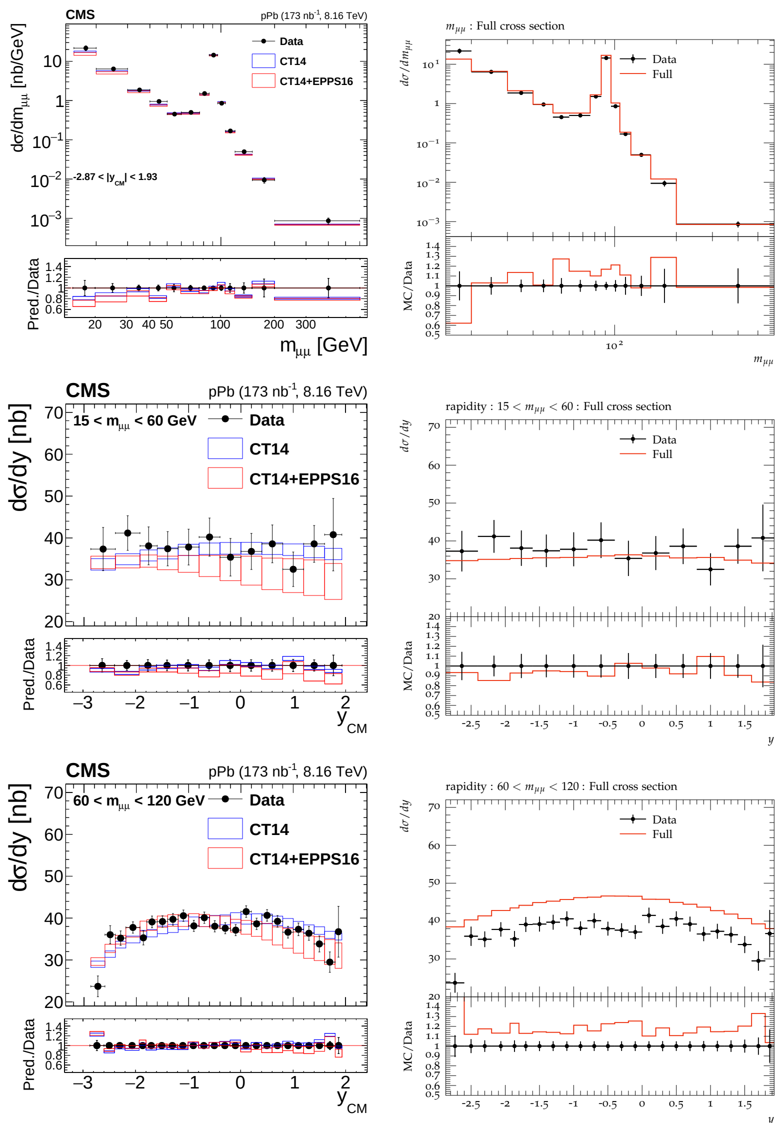

Figure 3: Differential cross section for the DY process measured in the muon channel, as a function of the dimuon invariant mass and rapidity in the centre-of-mass frame for $15 < m < 60\text{GeV}$ and $60 < m < 120\text{GeV}$. The error bars on the data represent the quadratic sum of the statistical and systematic uncertainties. The results of [1] are on the left and ours on the right.

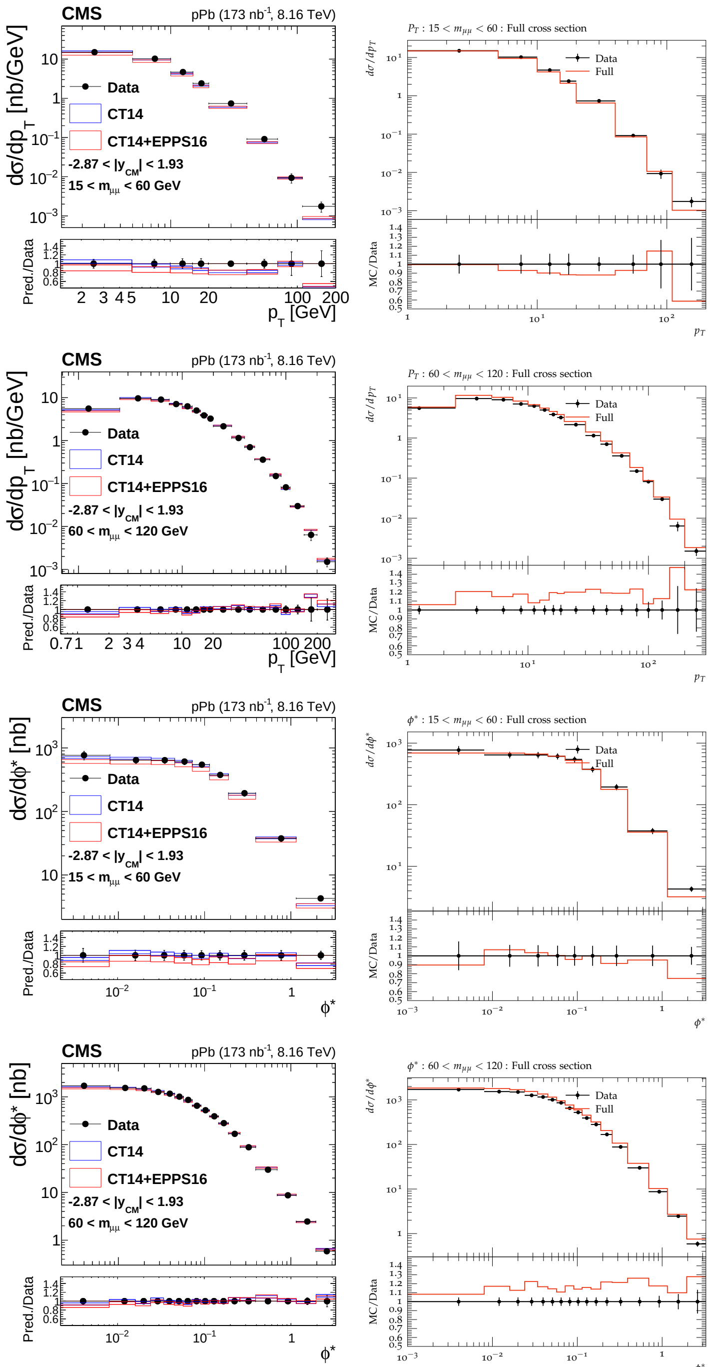

Figure 4: Differential cross sections for the DY process measured in the muon channel, as functions of $p_{\mathrm{T}}$and ϕ^*, for $15 < m < 60$GeVand $60 < m < 120$GeV. The first bin of the $p_{\mathrm{T}}$and ϕ^* measurements starts at 0. The error bars on the data represent the quadratic sum of the statistical and systematic uncertainties. The results of [1] are on the left and ours on the right.

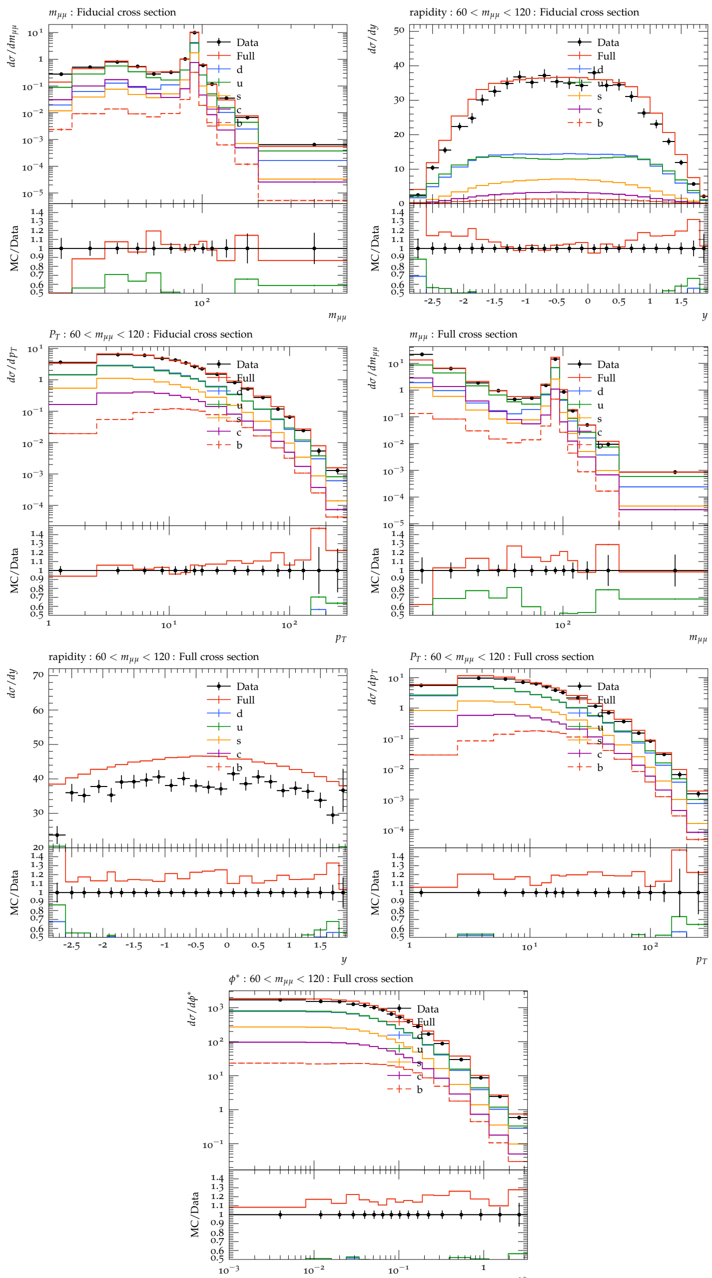

Figure 5: Participation of all possible flavors is shown for mass, Pt, rapidity, and ϕ^* distributions in high masses. We have the same cuts as before for the fiducial cross-section. We consider the mass of b-flavor negligible. .

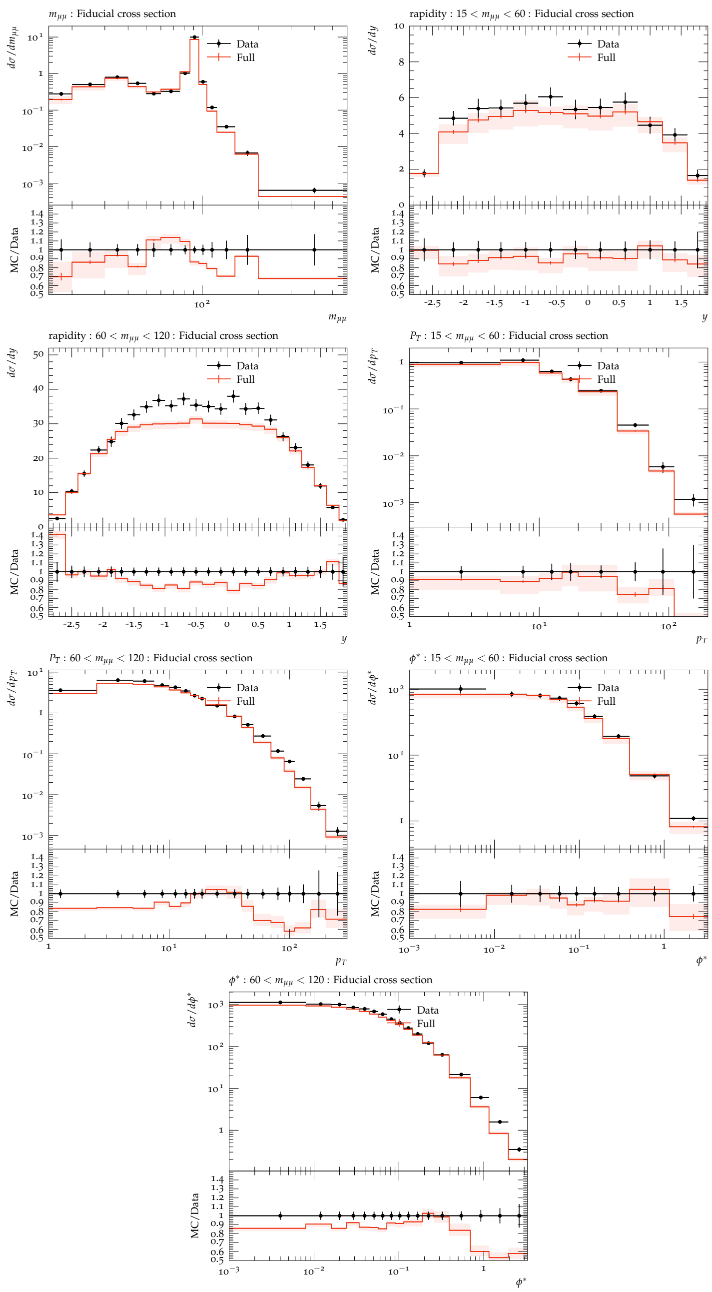

Figure 6: Differential fiducial cross section (without the acceptance correction) for the DY process measured in the muon channel, as a function of the dimuon invariant mass and rapidity in the centre-of-mass frame for $15 < m < 60$GeV and $60 < m < 120$GeV $p_{\mathrm{T}}$and ϕ^*, for $15 < m < 60$GeV and $60 < m < 120$GeV . The first bin of the $p_{\mathrm{T}}$and ϕ^* measurements starts at 0. The error bars on the data represent the quadratic sum of the statistical and systematic uncertainties. we used madgraph5_aMCatNLO and Cascade3.

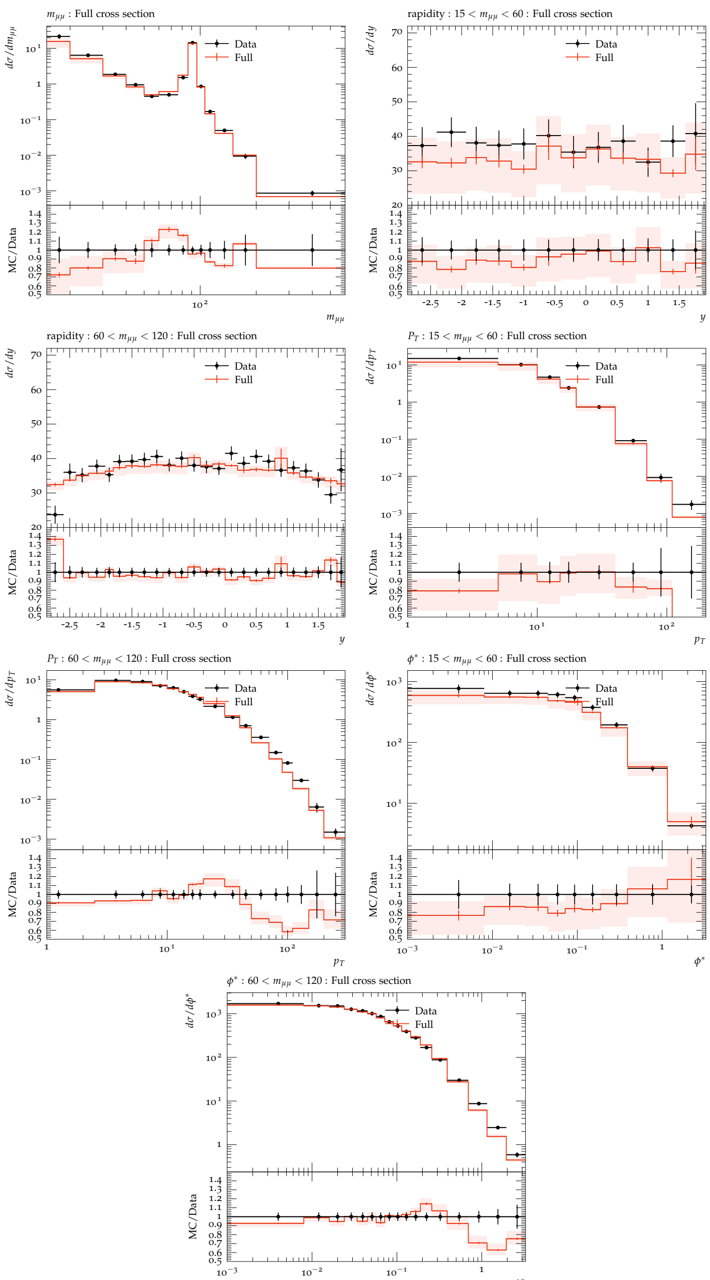

Figure 7: Differential cross section for the DY process measured in the muon channel, as a function of the dimuon invariant mass and rapidity in the centre-of-mass frame for $15 < m < 60$GeV and $60 < m < 120$GeV $p_{\mathrm{T}}$and ϕ^*, for $15 < m < 60$GeV and $60 < m < 120$GeV . The first bin of the $p_{\mathrm{T}}$and ϕ^* measurements starts at 0. The error bars on the data represent the quadratic sum of the statistical and systematic uncertainties. we used madgraph5_aMCatNLO and Cascade3.

Appendix: complete Rivet routine code

```
\newpage
// -*- C++ -*-
#include "Rivet/Analysis.hh"
#include "Rivet/Projections/FinalState.hh"
#include "Rivet/Projections/FastJets.hh"
#include "Rivet/Projections/DressedLeptons.hh"
#include "Rivet/Projections/MissingMomentum.hh"
#include "Rivet/Projections/PromptFinalState.hh"
#include "Rivet/Projections/ZFinder.hh"
#include "Rivet/Projections/Beam.hh"
#include "Rivet/Math/LorentzTrans.hh"
#include "Rivet/Math/Vector3.hh"
#include "Rivet/Math/Units.hh"

namespace Rivet {


  /// @brief Add a short analysis description here
  class CMS_2021_I1849180 : public Analysis {
  public:

    /// Constructor
    RIVET_DEFAULT_ANALYSIS_CTOR(CMS_2021_I1849180);
    float y_shift = 0;

    /// @name Analysis methods
    ///@{

    /// Book histograms and initialise projections before the run
    void init() {


      const FinalState fs;//(Cuts::abseta < 4.9);
      const FinalState fsfull;//(Cuts::abseta < 25);

      const ParticlePair& beam = beams();
      pcom = beam.first.momentum() + beam.second.momentum();
      if (beam.first.mom().E() == beam.second.mom().E()){
      y_shift = - 0.465;
      }


      ZFinder zmumuFinder(fs, Cuts::abseta<2.4 && Cuts::pT>10*GeV, PID::MUON, 15.0*GeV , 600.0*GeV,0.1, ZFinder::ClusterPhotons::NODECAY);

      declare(zmumuFinder, "ZmumuFinder");

     ZFinder TotzmumuFinder(fsfull, Cuts::pT>0.0*GeV,PID::MUON, 0.0*GeV , 1000.0*GeV,0.1, ZFinder::ClusterPhotons::NODECAY);

      declare(TotzmumuFinder, "TotzmumuFinder");

      declare(FinalState(), "FS");


      book(_h["111"], 1, 1, 1);
      book(_h["211"], 2, 1, 1);
      book(_h["311"], 3, 1, 1);
      book(_h["411"], 4, 1, 1);
      book(_h["511"], 5, 1, 1);
      book(_h["611"], 6, 1, 1);
      book(_h["711"], 7, 1, 1);
      book(_h["911"], 9, 1, 1); //inmass_mumu-noCUT
      book(_h["1011"], 10, 1, 1); //rap-low60-noCUT
      book(_h["1111"], 11, 1, 1); //rap-high60-noCUT
      book(_h["1211"], 12, 1, 1); //pt-low60-noCUT
      book(_h["1311"], 13, 1, 1); //pt-high60-noCUT
      book(_h["1411"], 14, 1, 1); //phist-low60-noCUT
      book(_h["1511"], 15, 1, 1); //phist-high60-noCUT



      book(_h["Zpt-lowmass"],"Zpt-lowmass", logspace(100, 1e-2, 1));
      book(_h["Zpt-highmass"],"Zpt-highmass", logspace(100, 1e-2, 1));

      book(_h["15_x1_60"],"x1_low", logspace(50, 1e-4, 1));
      book(_h["15_x2_60"],"x2_low", logspace(50, 1e-4, 1));
      book(_h_x_low60,"x_low", logspace(50, 1e-4, 1));
      book(_h["60_x1"],"x1_high", logspace(50, 1e-4, 1));
      book(_h["60_x2"],"x2_high", logspace(50, 1e-4, 1));
```

```
      book(_h_x_high60,"x_high", logspace(50, 1e-4, 1));

    }

    /// Perform the per-event analysis
    void analyze(const Event& event) {

         const ZFinder& ZmumuFinder = applyProjection<ZFinder>(event,"ZmumuFinder");
         const ZFinder& TotzmumuFinder = applyProjection<ZFinder>(event,"TotzmumuFinder");
         const Particles&  zmumus = ZmumuFinder.bosons();
         const Particles&  totzmumus = TotzmumuFinder.bosons();

      const Vector3 betacom = pcom.betaVec();
      const LorentzTransform comboost = LorentzTransform::mkFrameTransformFromBeta(betacom);

      if ( zmumus.size()==1){
      for(const Particle& zmumu : zmumus) {

Particles leptons = sortBy(ZmumuFinder.constituents(), cmpMomByPt);
  if (leptons[0].pt()<15 or leptons[1].pt()<10 or abs(deltaEta(leptons[0],leptons[1]))>2.4) {continue;}
const FourMomentum lminus = leptons[0].charge() < 0 ? leptons[0].momentum() : leptons[1].momentum();
const FourMomentum lplus = leptons[0].charge() < 0 ? leptons[1].momentum() : leptons[0].momentum();

const double phi_acop = M_PI - deltaPhi(lminus, lplus);
const double costhetastar = tanh((lminus.eta()-lplus.eta())/2.0);
const double sin2thetastar = (costhetastar <= 1) ? 1.0 - sqr(costhetastar) : 0;
const double phistar = tan(phi_acop/2.0) * sqrt(sin2thetastar);

      const FourMomentum Zcm = comboost.transform(zmumu.momentum());

      const double Zmass = zmumu.mass()/GeV;
      const double Zpt = zmumu.momentum().pT()/GeV;
      const double Zylab = zmumu.momentum().rapidity();
      const double Zy = Zcm.rapidity()+ y_shift;

             if (Zcm.rapidity() <1.93 && Zcm.rapidity() >-2.87 ){
     _h["111"]->fill(Zmass);  }
     if (Zmass <60 && Zmass  > 15){
             if (Zcm.rapidity() <1.93 && Zcm.rapidity() >-2.87 ){
     _h["411"]->fill(Zpt);
     _h["Zpt-lowmass"]->fill(Zpt);
     _h["611"]->fill(phistar);
      }
             _h["211"]->fill(Zy);

              }
```

```
        if (Zmass <120 && Zmass > 60){
              if (Zcm.rapidity() <1.93 && Zcm.rapidity() >-2.87 ){
       _h["511"]->fill(Zpt);
       _h["Zpt-highmass"]->fill(Zpt);
       _h["711"]->fill(phistar);
}
       _h["311"]->fill(Zy);

                }
       }
       }

       if ( totzmumus.size()==1){
        for(const Particle& totzmumu : totzmumus) {

Particles leptons = sortBy(TotzmumuFinder.constituents(), cmpMomByPt);
  //if(leptons[0].eta() >2.5) {cout << "leptons[0].pt() = " << leptons[0].eta()<< endl;}
const FourMomentum lminus = leptons[0].charge() < 0 ? leptons[0].momentum() : leptons[1].momentum();
const FourMomentum lplus = leptons[0].charge() < 0 ? leptons[1].momentum() : leptons[0].momentum();

const double phi_acop = M_PI - deltaPhi(lminus, lplus);
const double costhetastar = tanh((lminus.eta()-lplus.eta())/2.0);
const double sin2thetastar = (costhetastar <= 1) ? 1.0 - sqr(costhetastar) : 0;
const double totphistar = tan(phi_acop/2.0) * sqrt(sin2thetastar);



              const FourMomentum totZcm = comboost.transform(totzmumu.momentum());

       const double totZmass = totzmumu.mass()/GeV;
       const double totZpt = totzmumu.momentum().pT()/GeV;
       const double totZylab = totzmumu.momentum().rapidity();
       const double totZy = totZcm.rapidity()+ y_shift;



if (totZcm.rapidity() <1.93 && totZcm.rapidity() >-2.87 ){
        _h["911"]->fill(totZmass);}

             if (totZmass <60 && totZmass>15){
          if (totZcm.rapidity() <1.93 && totZcm.rapidity() >-2.87 ){
      _h["1211"]->fill(totZpt);
      _h["1411"]->fill(totphistar);}
      _h["1011"]->fill(totZy);
}
        if (totZmass <120 && totZmass > 60){
        if (totZcm.rapidity() <1.93 && totZcm.rapidity() >-2.87 ){
       _h["1311"]->fill(totZpt);
       _h["1511"]->fill(totphistar);}
       _h["1111"]->fill(totZy);
        }
        }
       }

             }


     /// Normalise histograms etc., after the run
     void finalize() {





     double norm=crossSection()*208/nanobarn/sumW();

       scale(_h["111"], norm);
       scale(_h["211"], norm);
       scale(_h["311"], norm);
       scale(_h["411"], norm);
       scale(_h["511"], norm);
       scale(_h["611"], norm);
       scale(_h["711"], norm);
       scale(_h["911"], norm);
       scale(_h["1011"], norm);
       scale(_h["1111"], norm);
       scale(_h["1211"], norm);
       scale(_h["1311"], norm);
       scale(_h["1411"], norm);
```

```
      scale(_h["1511"], norm);

      scale(_h["Zpt-highmass"], norm);
      scale(_h["Zpt-lowmass"], norm);



    }

    ///@}


    /// @name Histograms
    ///@{
    FourMomentum pcom;


    map<string, Histo1DPtr> _h;
    map<string, Profile1DPtr> _p;
    map<string, CounterPtr> _c;
    ///@}


  };


  RIVET_DECLARE_PLUGIN(CMS_2021_I1849180);

}
```

References

[1] CMS Collaboration, "Study of Drell-Yan dimuon production in proton-lead collisions at $\sqrt{s_{\mathrm{NN}}} = 8.16$ TeV", *JHEP* **05** (2021) 182, arXiv:2102.13648.

[2] C. Bierlich et al., "Robust Independent Validation of Experiment and Theory: Rivet version 3", *SciPost Phys.* **8** (2020) 026, arXiv:1912.05451.

[3] M. Dobbs and J. B. Hansen, "The HepMC C++ Monte Carlo event record for High Energy Physics", *Comput. Phys. Commun.* **134** (2001) 41.

[4] A. Buckley et al., "The HepMC3 Event Record Library for Monte Carlo Event Generators", arXiv:1912.08005.