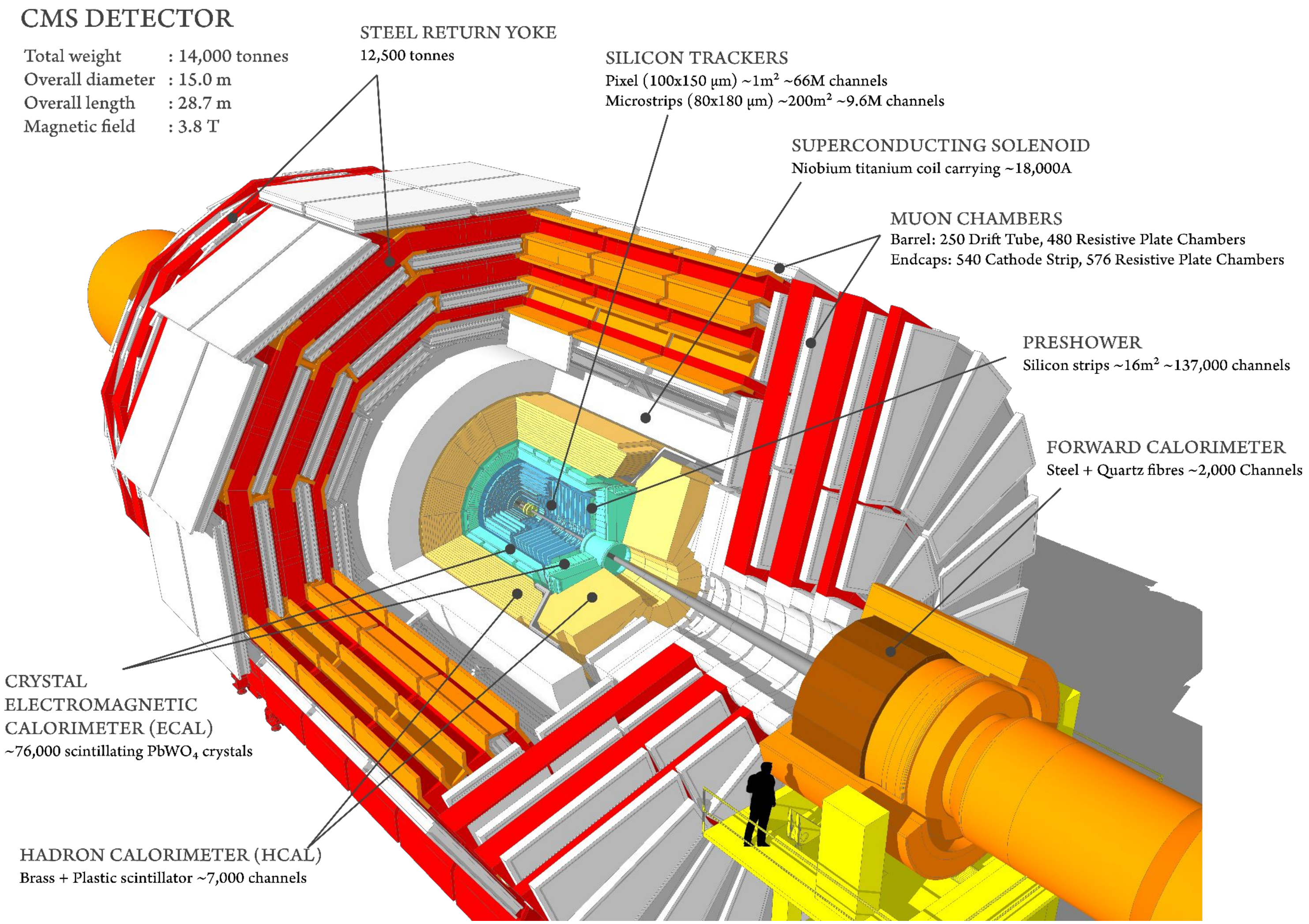
UNIVERSIDAD DE ANTIOQUIA

1803

Reproducción de la masa invariante de los Mesones D^{+}, D^{0} y D^{*+}

Santiago Galvis - Felipe Castello - Juan Esteban Ospina

Detector



UNIVERSIDAD DE ANTIOQUIA

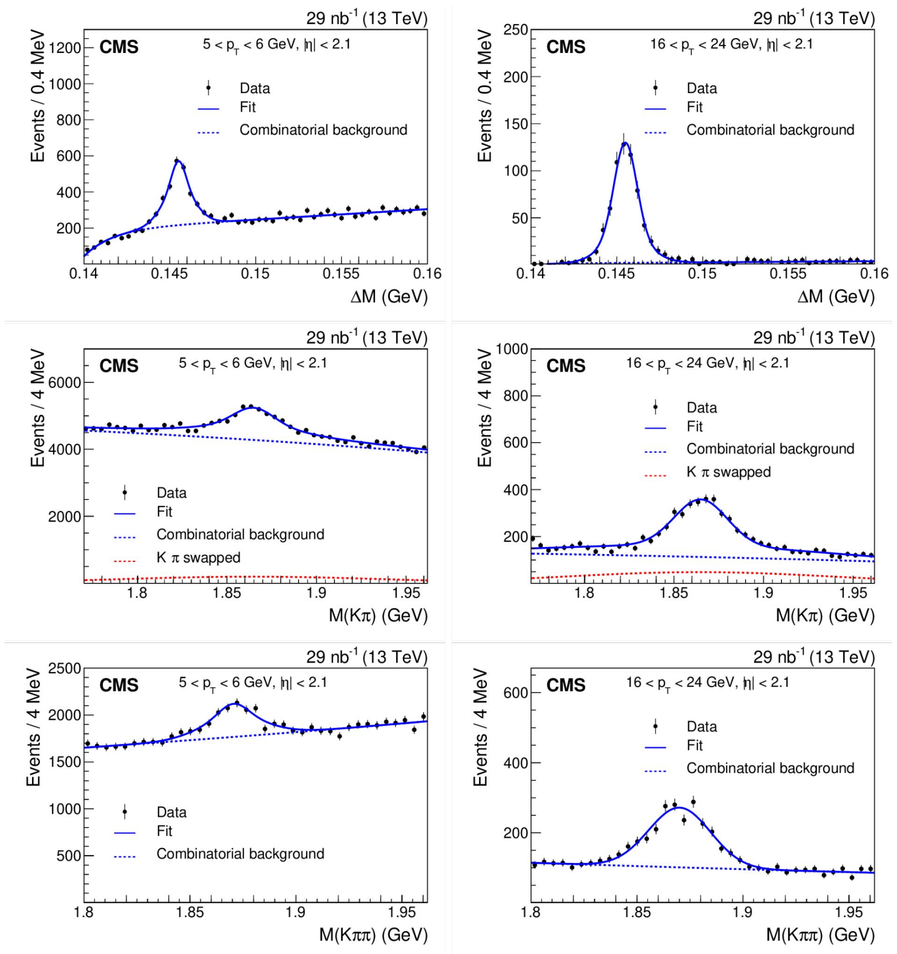
EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH (CERN)

CERN-EP-2021-085
2021/12/10

CMS-BPH-18-003

Measurement of prompt open-charm production cross sections in proton-proton collisions at $\sqrt{s} = 13$ TeV

The CMS Collaboration*



UNIVERSIDAD DE ANTIOQUIA
1803

Canales de decaimiento

Nombre de la partícula	Símbolo de la partícula	Símbolo de la antipartícula	Quarks que contiene	Masa en reposo(MeV/c^2)	I^G	J^{PC}	S	C	B'	Tiempo de vida medio (s)	Desintegraciones frecuentes (>5% de desintegracones)
Mesón D	D^+	D^-	$c\bar{d}$	1869.62 ± 0.20	1/2	0^-	0	+1	0	$(1.040 \pm 0.007) \times 10^{-12}$	
Mesón D extraño	D^+_s	D^-_s	$c\bar{s}$	1968.47 ± 0.33	0	0^-	+1	+1	0	$(5.00 \pm 0.07) \times 10^{-13}$	
Mesón D	D^0	$\overline{D}^0$	$c\bar{u}$	1864.84 ± 0.17	1/2	0^-	0	+1	0	$4.101 \pm 0.015 \times 10^{-13}$	
Mesón D	D^{*+} (2010)	D^{*-}(2010)	$c\bar{d}$	2010.27.62 ± 0.17	1/2	1^-	0	+1	0	$6.9 \pm 1.9 \times 10^{-21}$	$D^0 + \pi^+$ $D^+ + \pi^0$
Mesón D	D^{*0} (2007)	$\overline{D}^{*0}$(2007)	$c\bar{u}$	2006.97 ± 0.19	1/2	1^-	0	+1	0	$>3.1 \times 10^{-22}$	$D^0 + \pi^0$ $D^0 + \gamma$

- $\mathrm{pp} \to \mathrm{D}^{*+}X \to \mathrm{D}^0\pi_s^+ X \to \mathrm{K}^-\pi^+\pi_s^+ X,$
- $\mathrm{pp} \to \mathrm{D}^0 X \to \mathrm{K}^-\pi^+ X,$
- $\mathrm{pp} \to \mathrm{D}^+ X \to \mathrm{K}^-\pi^+\pi^+ X,$

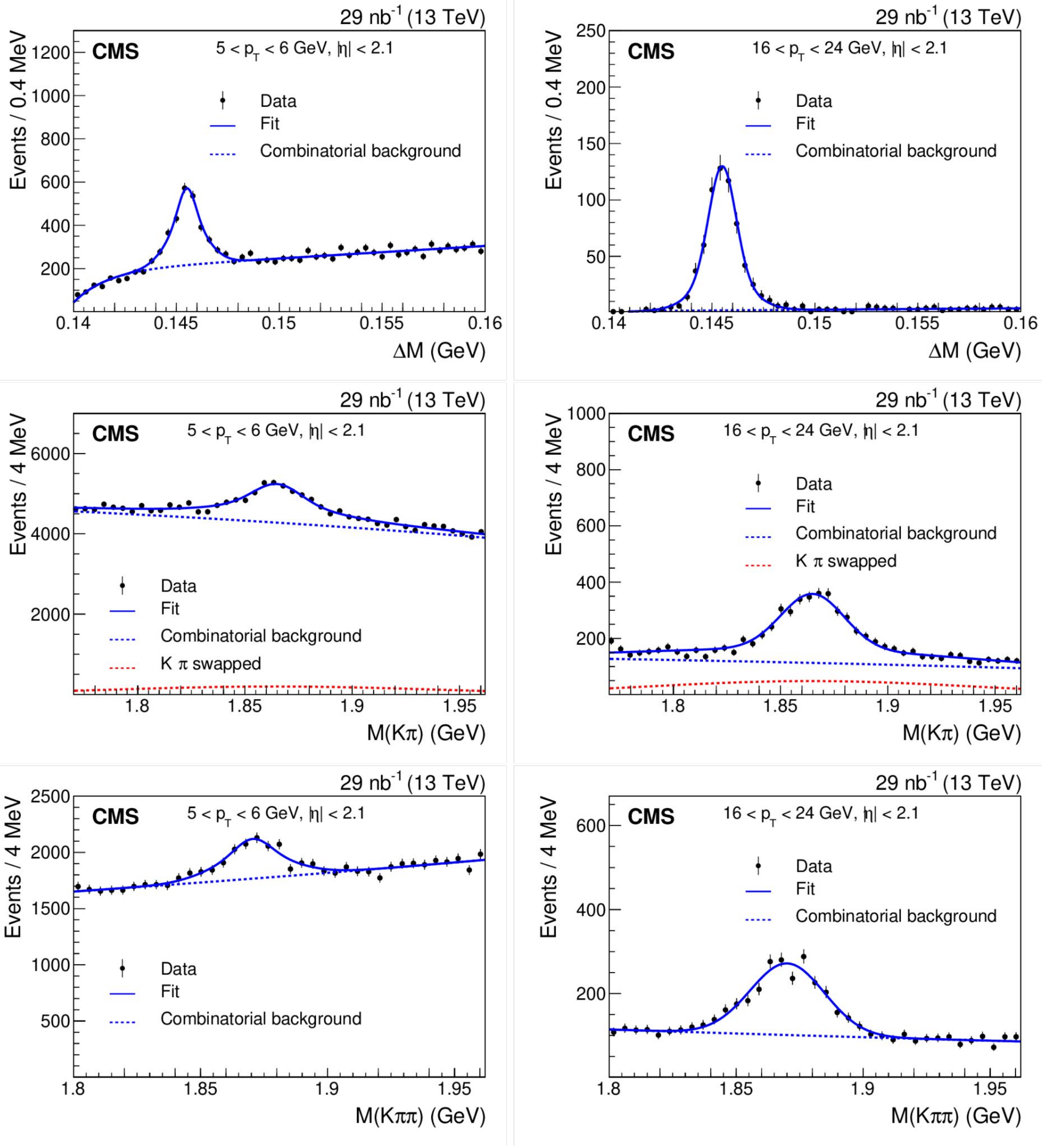
$$\Delta M = m(\mathrm{K}\pi\pi_S^+) - m(\mathrm{K}\pi)$$

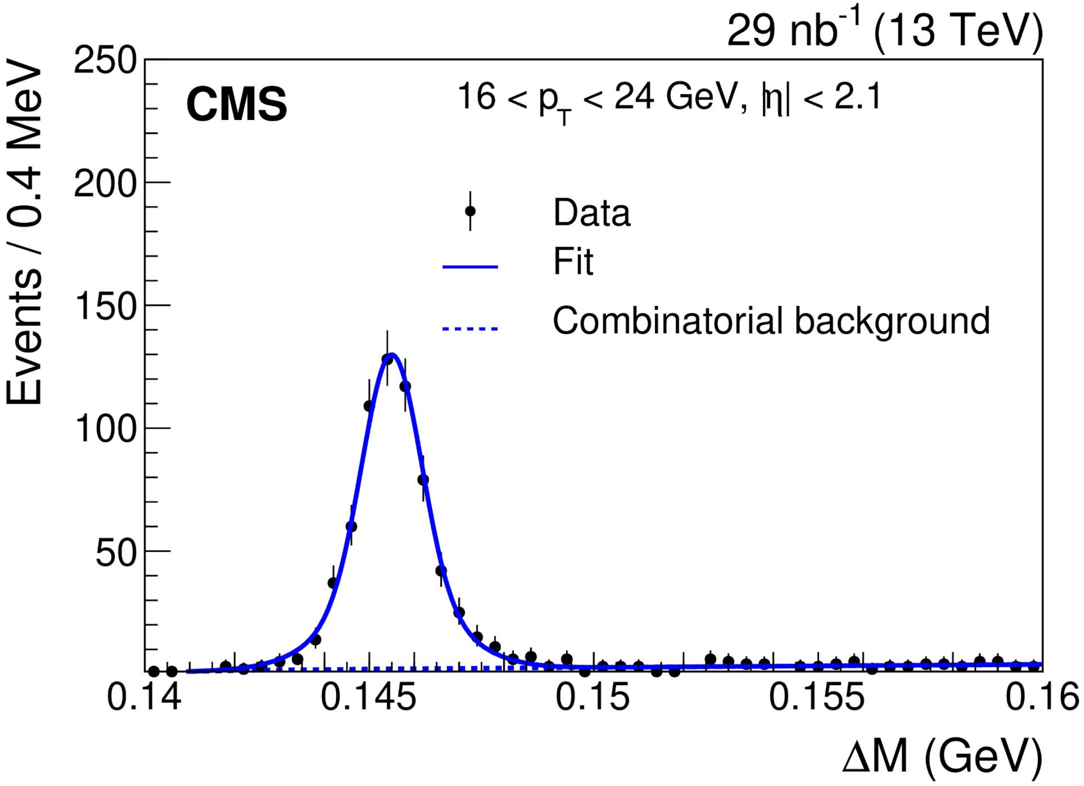
UNIVERSIDAD DE ANTIOQUIA
1 8 0 3

Repositorio:

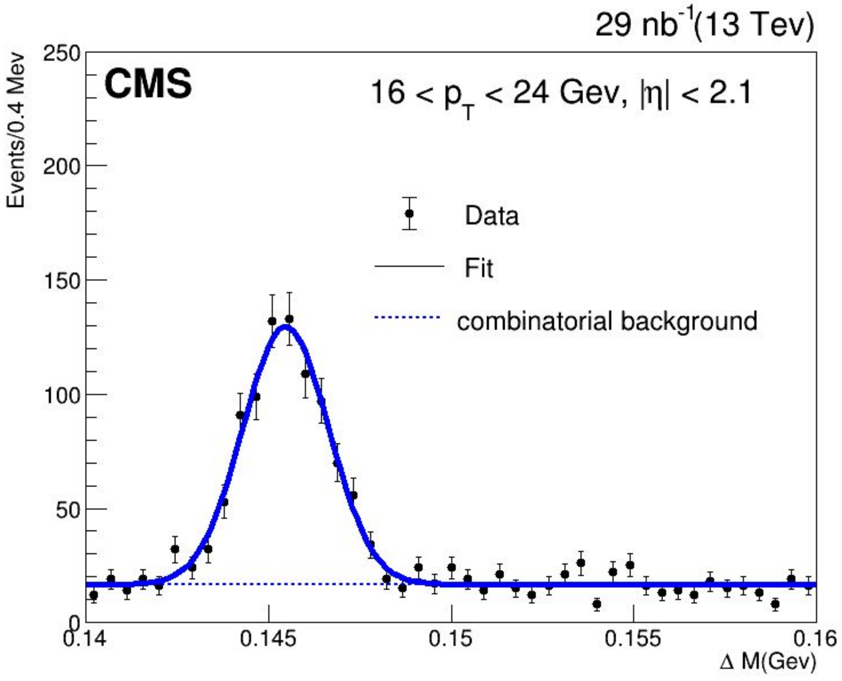
github.com/JEOspina25/Proyecto_BigData

UNIVERSIDAD DE ANTIOQUIA
1803


29 nb-1 (13 TeV)
CMS
5 < pT < 6 GeV, |η| < 2.1
Data
Fit
Combinatorial background
Events / 0.4 MeV
ΔM (GeV)
29 nb-1 (13 TeV)
CMS
16 < pT < 24 GeV, |η| < 2.1
Data
Fit
Combinatorial background
Events / 0.4 MeV
ΔM (GeV)
29 nb-1 (13 TeV)
CMS
5 < pT < 6 GeV, |η| < 2.1
Data
Fit
Combinatorial background
K π swapped
Events / 4 MeV
M(Kπ) (GeV)
29 nb-1 (13 TeV)
CMS
16 < pT < 24 GeV, |η| < 2.1
Data
Fit
Combinatorial background
K π swapped
Events / 4 MeV
M(Kπ) (GeV)
29 nb-1 (13 TeV)
CMS
5 < pT < 6 GeV, |η| < 2.1
Data
Fit
Combinatorial background
Events / 4 MeV
M(Kππ) (GeV)
29 nb-1 (13 TeV)
CMS
16 < pT < 24 GeV, |η| < 2.1
Data
Fit
Combinatorial background
Events / 4 MeV
M(Kππ) (GeV)



Original



Reconstruida

Reconstrucción

```
// Declare variables x,mean,sigma, c
RooRealVar x("x", "x", 0.14, 0.16);
RooRealVar mean("mean", "mean of gaussian", 0.1455, 0.14, 0.16);
RooRealVar sigma("sigma", "width of gaussian", 0.0012, 0, 0.2);
RooGaussian gauss("gauss", "gaussian PDF", x, mean, sigma);

RooRealVar c("c","c",0,-1,1);
RooPolynomial bkg1("bkg1","Background",x,RooArgSet(c),2);

//Pesos de Background y señal
RooRealVar Ns("Ns","Ns",0.,500);
RooRealVar Nb("Nb","Nb",0.,500);
//Modelo para la masa
RooAddPdf MassModel("MassModel","MassModel",RooArgList(gauss,bkg1),RooArgList(Ns,Nb));
```

UNIVERSIDAD DE ANTIOQUIA
1803

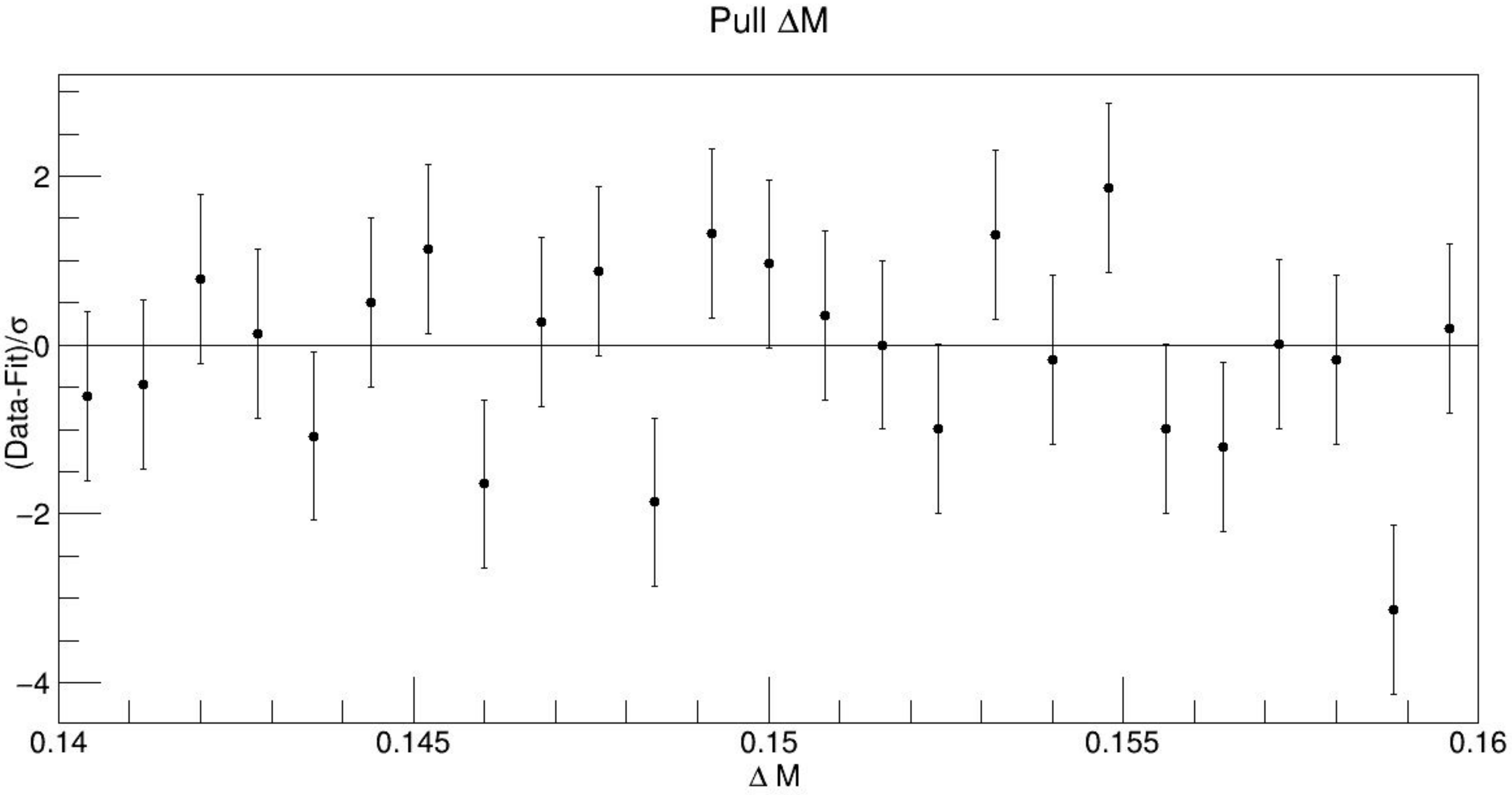
Reconstrucción

```
// Generate a dataset of  events in x from MassModel
RooDataSet *data = MassModel.generate(x, datos);

// Fit pdf to data
RooFitResult* fitres = MassModel.fitTo(*data,Extended(),Minos(kFALSE),Save(kTRUE), NumCPU(4));
RooPlot *xframe2 = x.frame();
```

```
//Pull
RooPlot* Mframe = x.frame(0.14,0.16,((0.16-0.14)/0.0008)+1);
data->plotOn(Mframe,DataError(RooAbsData::SumW2),MarkerSize(1.0),XErrorSize(0));
MassModel.plotOn(Mframe,DrawOption("F"),FillColor(0),LineWidth(2),Name("fittotal"));
RooHist* hpullm2 = Mframe->pullHist() ;
```

UNIVERSIDAD
DE ANTIOQUIA
1803

Pull al modelo de masa



UNIVERSIDAD DE ANTIOQUIA
1 8 0 3

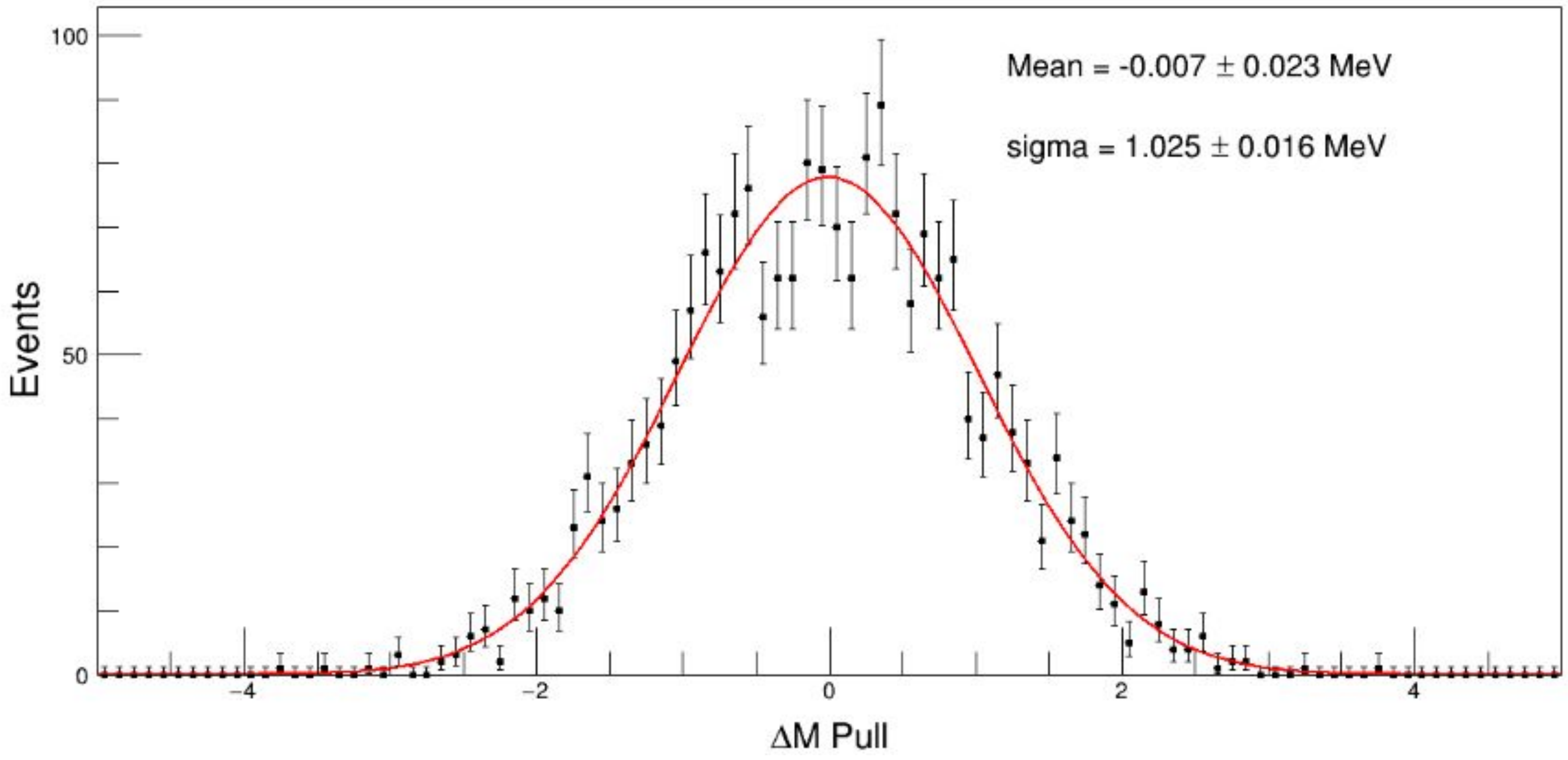
Montecarlo

```
for(Int_t n=0;n<ntotal;n++)
{
// cout<<"entro"<<endl;
mean.setVal(0.145);
mean.setError(0.000055);
width.setVal(0.0012);
c.setVal(0.0001);
Ns.setVal(1500);
Ns.setError(0.995);
Nb.setVal(500);
mui = 0.145;
nsi = 500;

//Generar montecarlo
RooDataSet *dataToy = MassModel.generate(RooArgSet(M), Extended(kTRUE));

//Fit a estos datos
RooFitResult* fitres = MassModel.fitTo(*dataToy,Extended(),Minos(kFALSE),Save(kTRUE), NumCPU(6));
```

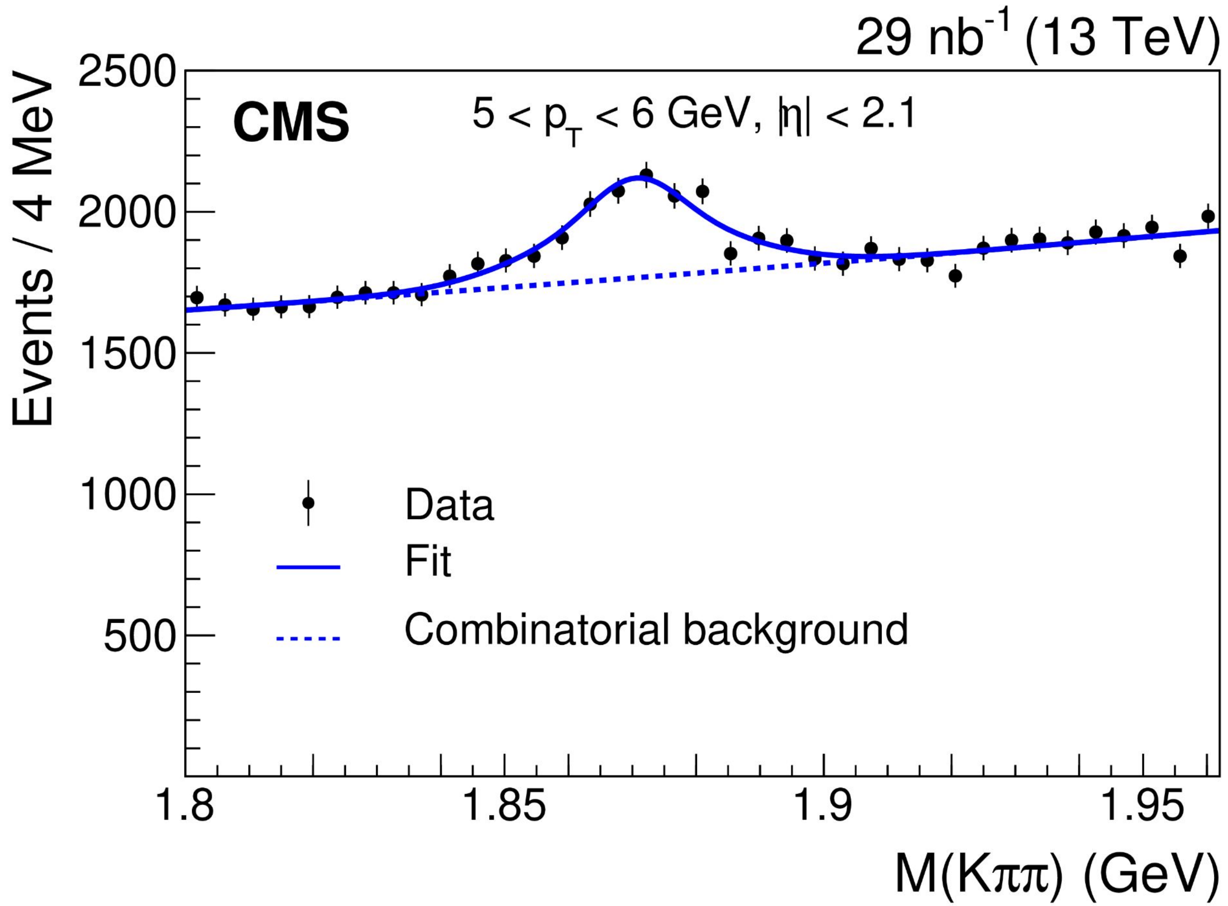
```
tree->Fill();
```

UNIVERSIDAD DE ANTIOQUIA

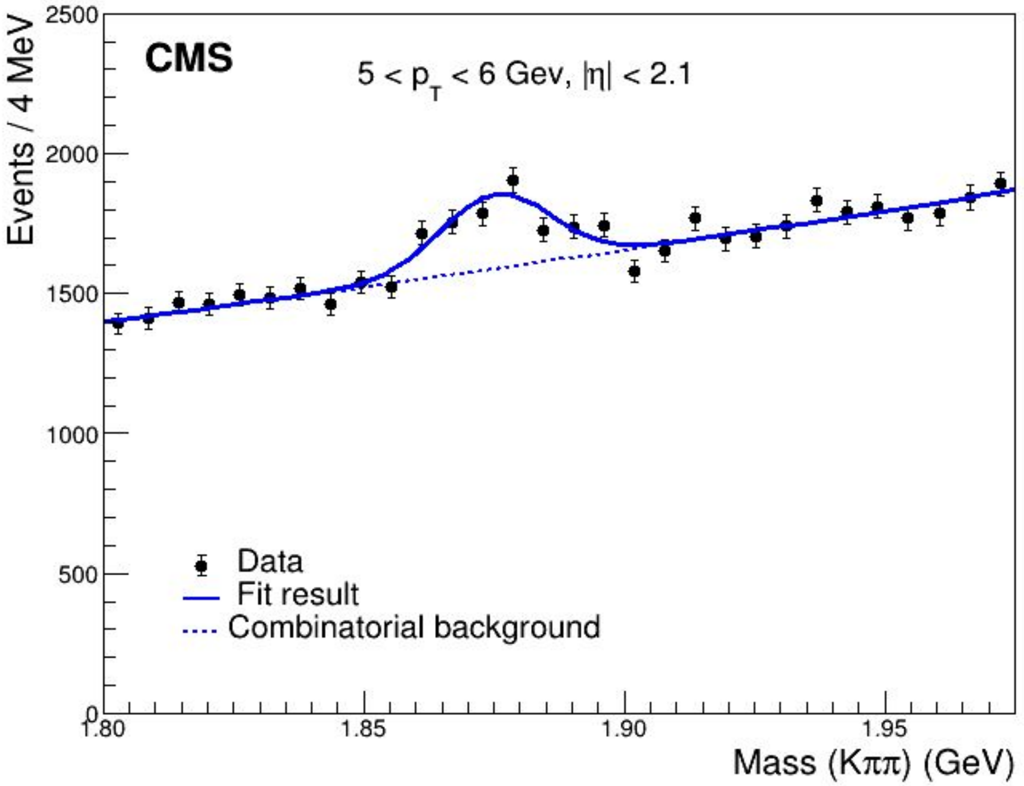
Montecarlo



UNIVERSIDAD DE ANTIOQUIA



Original



Reconstruida

Reconstrucción

```
// Variables a usar
RooRealVar M("M", "Mass (K#pi#pi) (GeV)", Mmin, Mmax);
// -Parámetros Señal-
RooRealVar mean("mean"," Mass mean",1.875,1.7,1.9,"GeV");
// Gausiana 1
RooRealVar width("width"," Mass width",0.010,0.001,0.012,"GeV"); // Sigma1
RooGaussian Sig("Sig"," Signal PDF",M,mean,width);
// Gausiana 2
RooRealVar width2("width2"," Mass width2 ",0.015,0.001,0.05,"GeV"); // Sigma2
RooGaussian Sig2("Sig2"," Signal PDF B",M,mean,width2);
// -Parámetros Background-
RooRealVar c0("c0","c0",0.0,10000.0);
RooRealVar c1("c1","c1",0.0,1.0);
RooPolynomial Bkg("Bkg","Exp. Background",M,RooArgList(c0,c1),3);
// Cantidad de datos por cada componente
RooRealVar Ns("Ns","Ns",0.,500);
RooRealVar Nb("Nb","Nb",1500.,20000);
RooRealVar fs("fs","fs",10,0.,1.);
// Suma de las dos gausianas
RooAddPdf Sumgaus("sumgau","sumgau",RooArgList(Sig,Sig2),RooArgList(fs));
// Modelo de masa (2 Gausianas + Exponencial)
RooAddPdf MassModel("MassModel","MassModel",RooArgList(Sumgaus,Bkg),RooArgList(Ns,Nb));
```

UNIVERSIDAD DE ANTIOQUIA

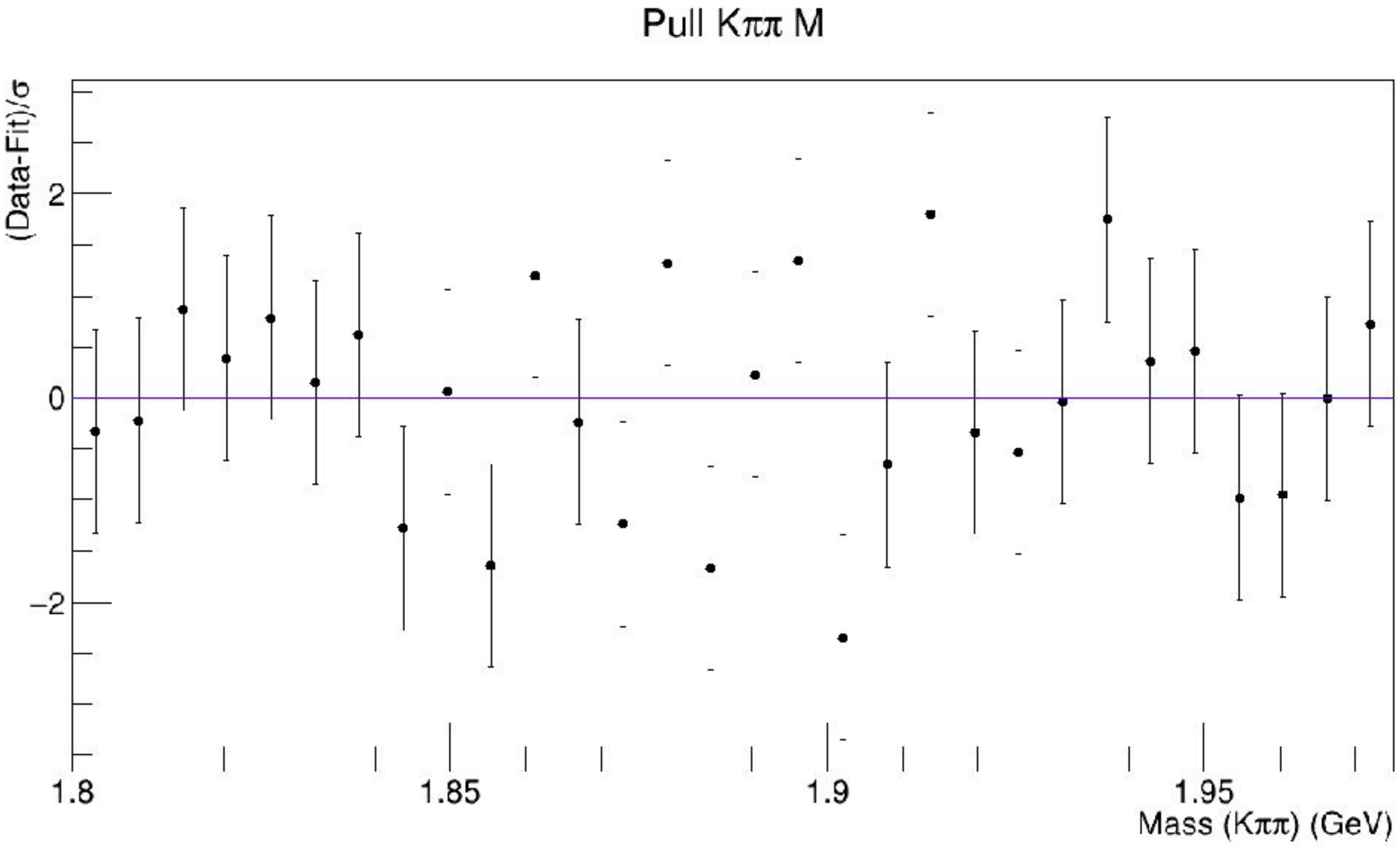
Reconstrucción

```
// Generación
RooDataSet* Data_M = MassModel.generate(M,datos);
// ---- Fitting ----
RooFitResult* ResultFit = MassModel.fitTo(*Data_M,Extended(),Minos(kFALSE),Save(kTRUE));
```

```
// Creando frame
RooPlot* Mframe = M.frame(infM,supM,nbin);
// Se dibujan los datos y el ajuste
Data_M->plotOn(Mframe,DataError(RooAbsData::SumW2),MarkerSize(1.0),XErrorSize(0));
MassModel.plotOn(Mframe,DrawOption("F"),FillColor(0),LineWidth(2),Name("fittotal"));

// Histograma Pull
RooHist* hpullm2 = Mframe->pullHist() ;
```

UNIVERSIDAD DE ANTIOQUIA
1803

Pull al modelo de masa



Implementación en una Clase de Root

UNIVERSIDAD DE ANTIOQUIA
1803

Muchas gracias