# Basic tests

M. De Mattia

August 11, 2009

**Abstract**

## 1 Basic tests

This section shows the results of applying MuScleFit after introducing biases in the data. In the first set of tests we keep the resolution fixed and fit only the $p_T$ scale.

The ansatz functions used for the resolution are

$$\sigma p_T/p_T = c1 + a1 p_T + b1 etaByPoints(\eta)\,,$$

$$\sigma \cot\theta = c2 + a2/p_T + b2|\eta| + d2\eta^2\,,$$

$$\sigma\phi = c3 + a3/p_T + b3|\eta| + d3\eta^2$$

and the values of the parameters are reported in table 1. Their derivation is described elsewhere [].

Table 1: Resolution starting parameters

| $\sigma p_T/p_T$ | |
|---|---|
| constant | -0.003 |
| linear $p_T$ | 0.000205 |
| etaByPoints | 1.0 |
| $\sigma\cot\theta$ | |
| constant | 0.00043 |
| $1/p_T$ | 0.0041 |
| linear $\lvert\eta\rvert$ | 0.0000028 |
| quadratic $\lvert\eta\rvert$ | 0.000077 |
| $\sigma\phi$ | |
| constant | 0.00011 |
| $1/p_T$ | 0.0018 |
| linear $\lvert\eta\rvert$ | -0.00000094 |
| quadratic $\lvert\eta\rvert$ | 0.000022 |

We consider a sample of 191570 events of the Summer08 redigi $Z \to \mu\mu$. This sample was filtered requiring at least 2 reconstructed global muons.

## 1.1 No-Bias

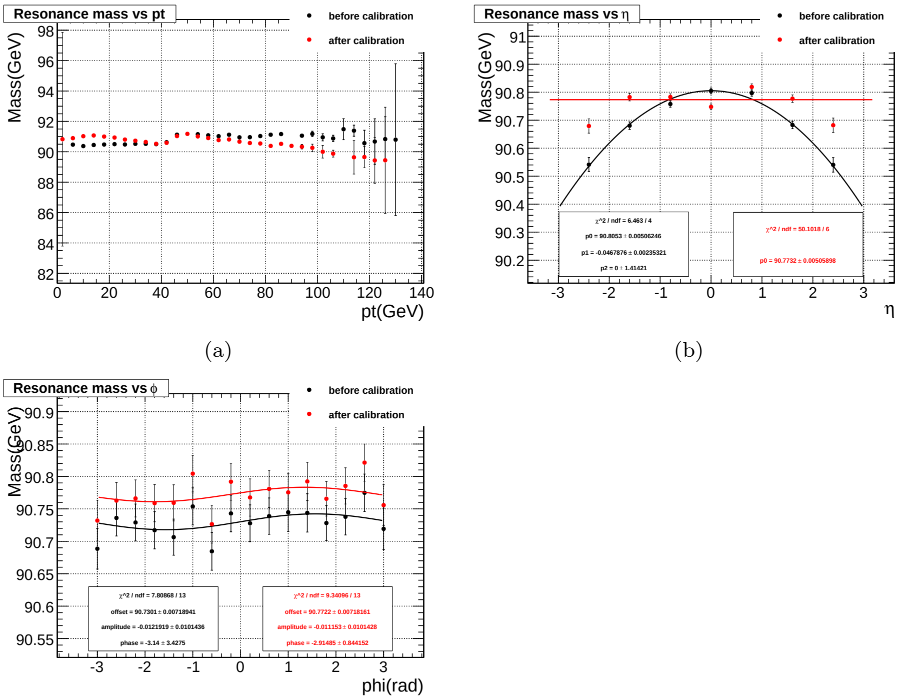
The first case is a consistency check: we do not apply any bias and fit the dataset. Of course the data themselves will have some biases which the algorithm will fit. Function used for the fit is

$$f(p_T, \eta) = a + bp_T + c|\eta| + d\eta^2 \tag{1}$$

and the fitted parameters are

$$a = 1.01379 \pm 0.000152375$$
$$b = -0.000380939 \pm 3.04641e-06$$
$$c = 0.00405687 \pm 5.38853e-05$$
$$d = -0.00153971 \pm 1.18328e-05$$

And the results are shown in figure 1.




(a) (b) (c)

Figure 1: Reconstructed mass vs muon $p_T$ (a), $\eta$ (b) and $\phi$ (c).

## 1.2 Linear bias in $p_T$ and quadratic fit

We add a bias of the following form

$$bias(p_T) = (a + bp_T)p_T\,, \tag{2}$$

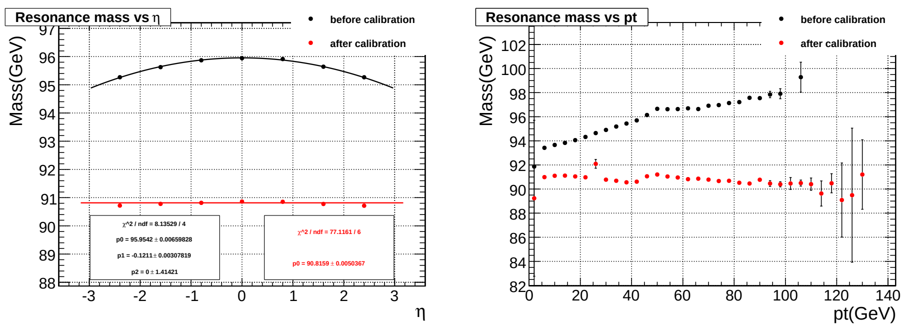
with $a = 1.015$ and $b = 0.001$. We fit with the a function with more parameters, to verify that the fit is correctly finding the bias, not because it is forced to do so. We fit with a function of the form:

$$f(p_T, \eta) = (a + bp_T + cp_T^2 + d|\eta| + e\eta^2)p_T \tag{3}$$

We find the following parameters

$$a = 1.00243 \pm 0.000117582$$
$$b = -0.00137233 \pm 1.40024e - 06$$
$$c = 2.58871e - 06 \pm 4.63024e - 08$$
$$d = -0.000793324 \pm 8.39588e - 05$$
$$e = 0.000146225 \pm 4.54723e - 05$$

And the correction is shown in figure 2




(a) (b)

Figure 2: Reconstructed mass vs muon $p_T$ (a) and $\eta$ (b).

### 1.3 Sinusoidal bias in $\phi$

We add a bias of the following form

$$bias(p_T, \phi) = (a + b\sin(\phi))p_T\ , \tag{4}$$

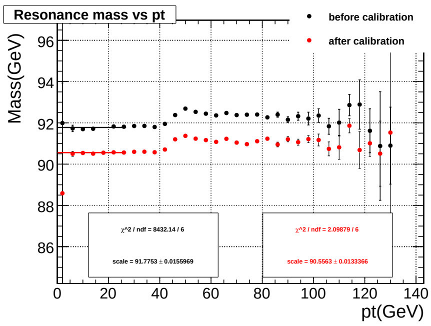
with $a = 1.015$ and $b = 0.025$. We fit with the same function and we find

$$a = 0.986406 \pm 7.92264e - 05$$
$$b = -0.0252447 \pm 0.000439811$$

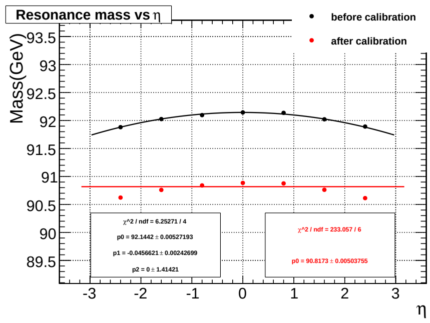
And the correction is shown in figure 3

## 2 Resolution fits

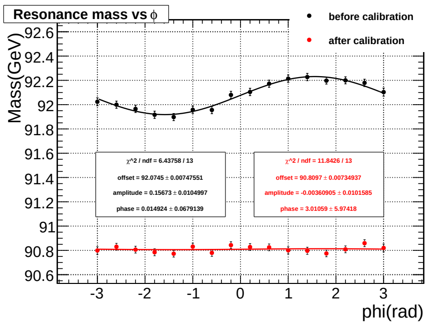
This section shows fits on the resolution parameters.




(a)




(b)




(c)

Figure 3: Reconstructed mass vs muon $p_T$ (a), $\eta$ (b) and $\phi+$ (c).

## 2.1 No smearing

In the first test we apply no smearing and fit the parameters of $\sigma p_T/p_T$. The ansatz function used is:

$$\sigma p_T/p_T(p_T,\eta) = a + b * p_T + c * etaByPoints(\eta)\,, \tag{5}$$

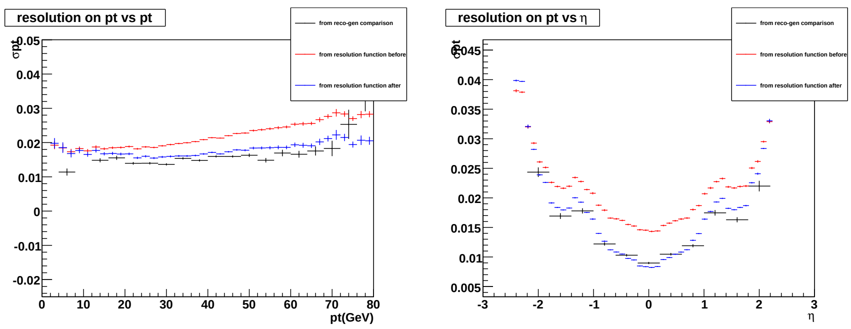
## 2.2 Linear smearing

Now we apply a linear smearing of the form:

$$\begin{aligned}
p_T &= p_T * (1.0 + y[0]parSmear[0]p_T)\,,\\
\phi &= \phi * (1.0 + y[1]parSmear[1])\,,\\
\cot\theta &= \cot\theta * (1.0 + y[2]parSmear[2])\,,\\
\eta &= -\log(\tan(\theta/2))\,,
\end{aligned}$$

where the last step takes into account the correct hemisphere and the $y$ are random variables with a gaussian pdf. r We fit with the function used in the previous case (5).

The results of the fit are shown in figure 4 together with to the resolutions computed comparing the reconstructed $p_T$ with the MonteCarlo value.




(a) (b)

Figure 4: Reconstructed mass vs muon $p_T$ (a), $\eta$ (b) and $\phi$ (c).

# 3 Resolution and Scale fit

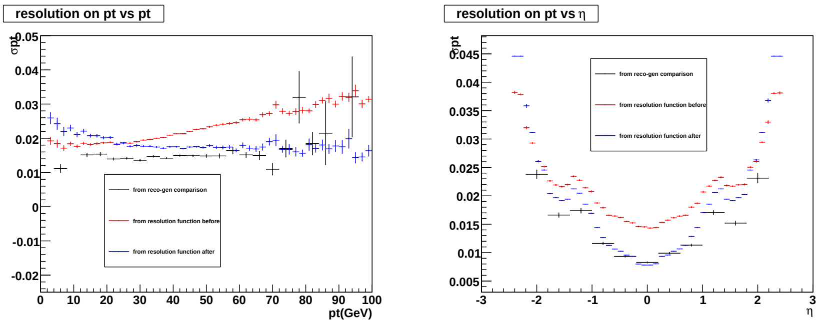
In this section we out together the resolution and the scale fit in successive iterations. We consider four cases:

- no smear and no bias
- smearing and no bias
- no smear and bias
- smearing and bias

We make three iterations starting with a resolution fit, followed by a scale fit and concluding with another resolution fit.

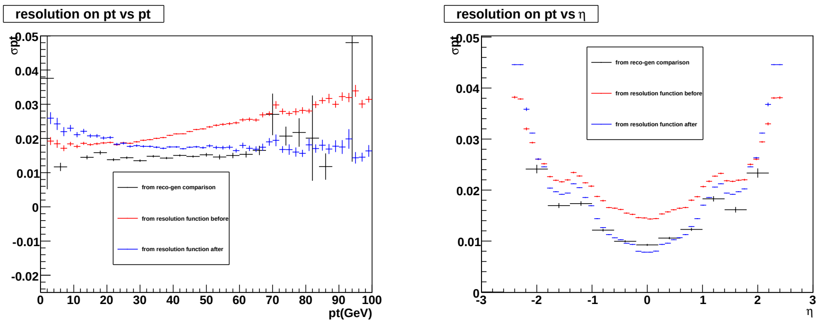
## 3.1 No smear and no bias CORRECT FIGURES: DONE WITH ONLY 10000 EVENTS

Figure 5 shows the resolution derived from MC comparison for uncorrected muons and those from the function before correction and after all corrections. Figure 6 has the same functions compared to the resolution derived from MC for corrected muons.




(a) (b)

Figure 5: Resolution given by $\sigma p_T / p_T$ vs $p_T$ (a), $\eta$ (b). They compare the last iteration with the uncorrected muons.




(a) (b)

Figure 6: Resolution given by $\sigma p_T / p_T$ vs $p_T$ (a), $\eta$ (b). They compare the last iteration with the uncorrected muons.

## 3.2 Bias and smearing

We apply both a bias (linear in $p_T$) and a smearing (again linear in $p_T$) and then fit scale and resolution with the same functions as before. Figure 3.2 shows the comparison between the reconstructed mass before corrections (but containing bias and smearing) and after all corrections (in this order: scale, resolution and again scale). In the same figure also the probability distributions before and after correction are shown. Each one is normalized to the corresponding mass distribution.

First the mass scale is fitted starting from the red distribution. Then the resolution is fitted, changing the probability ditribution from blue to green. Finally an additional scale fit improves the scale parameters on the base of the new resolution estimate. This produces the final black mass distribution.

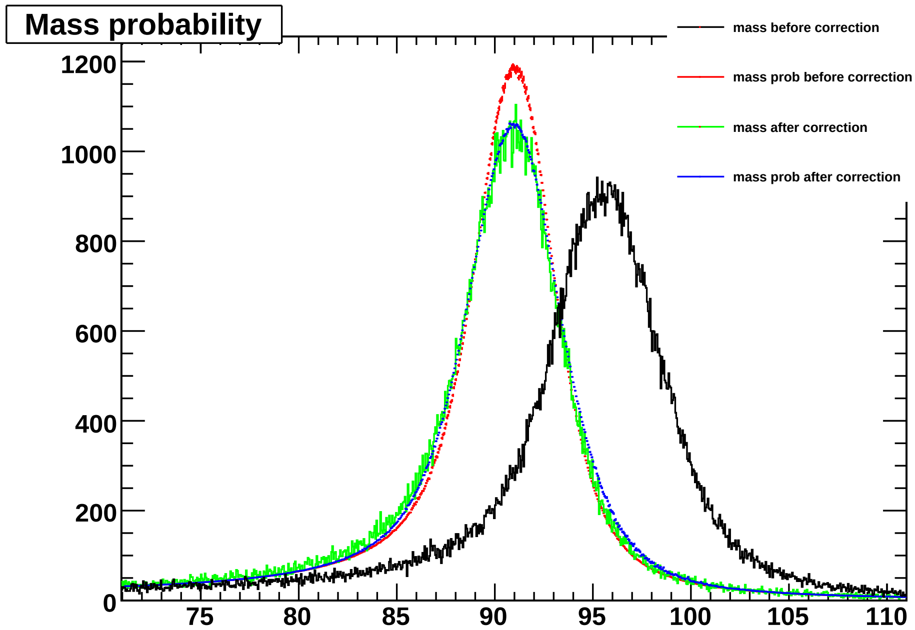



Figure 7: Biased and smeared muons from a $Z \rightarrow \mu\mu$ sample before and after the correction.

# 4 Fits with more than one resonance

In this section we show fits done with more than one resonance. All the fits described here are performed taking the tracker component of the muon track. We noticed that on the samples at our disposal the global muon track suffered from a significant loss in resolution below 10 GeV and we decided to use only the tracker component that does not have this problem.

The first example uses the three $\Upsilon$ resonances together. We applied the same linear in $p_T$ bias as in the previous section and the result is shown in figure 4.

We now test a biased and smeared sample of $\Upsilon 1S$, $\Upsilon 2S$, $\Upsilon 3S$ and $J/\Psi$. Figures 4 show the reconstructed mass and probability distributions of the muons after bias and smearing. The result of the application of a scale, resolution and then a new scale fit is shown in figures 4.

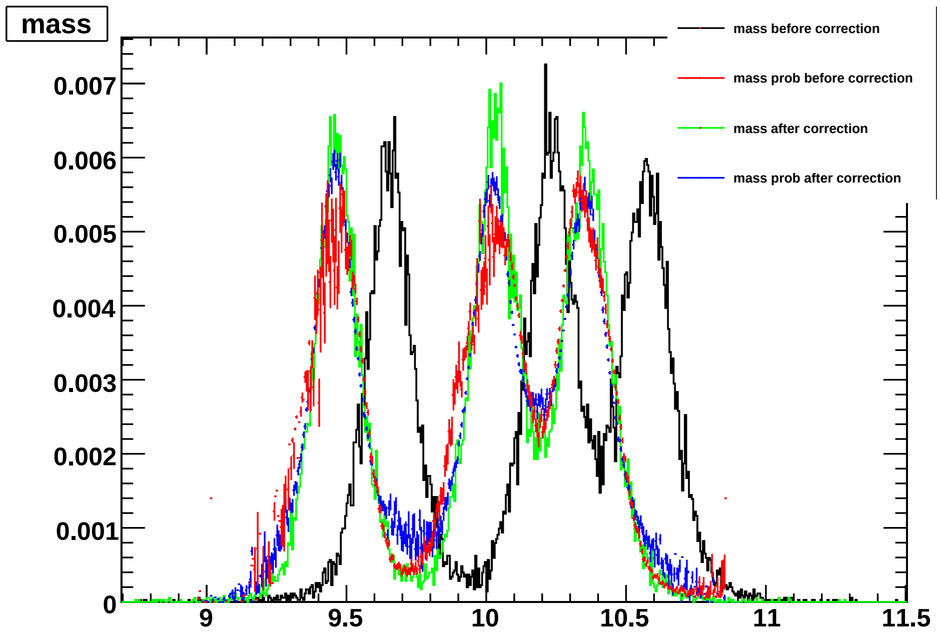



Figure 8: Biased muons from $\Upsilon$ resonances fitted together, shown before and after corrections.

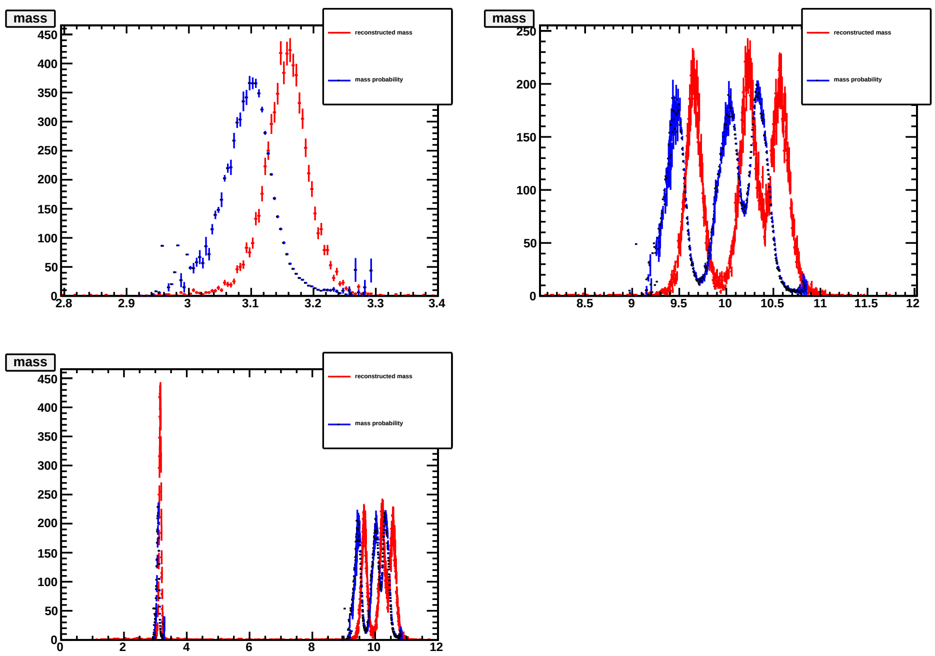



Figure 9: All the low $p_T$ resonances (but the $\Psi 2S$) before corrections.

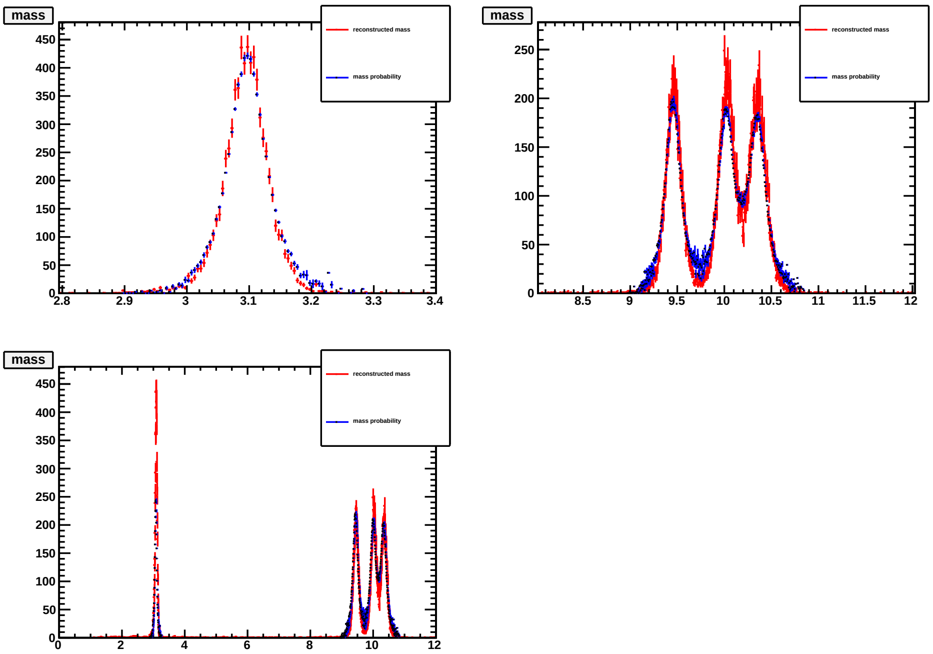



Figure 10: All the low $p_T$ resonances (but the $\Psi 2S$) after corrections.

## 5 Background fit

We test the background fit using a sample of XXX Upsilon 1S and YYY InclusivePPmuX. We fit an exponential function using a window of $\pm 5$ GeV around the Upsilon mass (10 times bigger than the one in the scale fit). The result is shown in figure 5.

## 6 BackgroundHandler

We introduced a new backgroundHandler class to handle the background. The mass spectrum is divided in three regions:

- Region 0 includes the $Z$
- Region 1 includes $\Upsilon 1S$, $\Upsilon 2S$ and $\Upsilon 3S$
- Region 2 includes $J/\Psi$ and $\Psi 2S$

Each region is centered on the mean value of the mass of corresponding resonances and its width of is configurable. When the background is fitted the regions are used, parameters for the functions are determined and the fraction of background events. When the background is not fitted the background fractions are rescaled to the resonance windows so that they provide an estimate for the scale and resolution fits.

We tested the new class applying a bias on the scale (same as in the previous sections) and performing a background fit followed by a scale fit. We did this

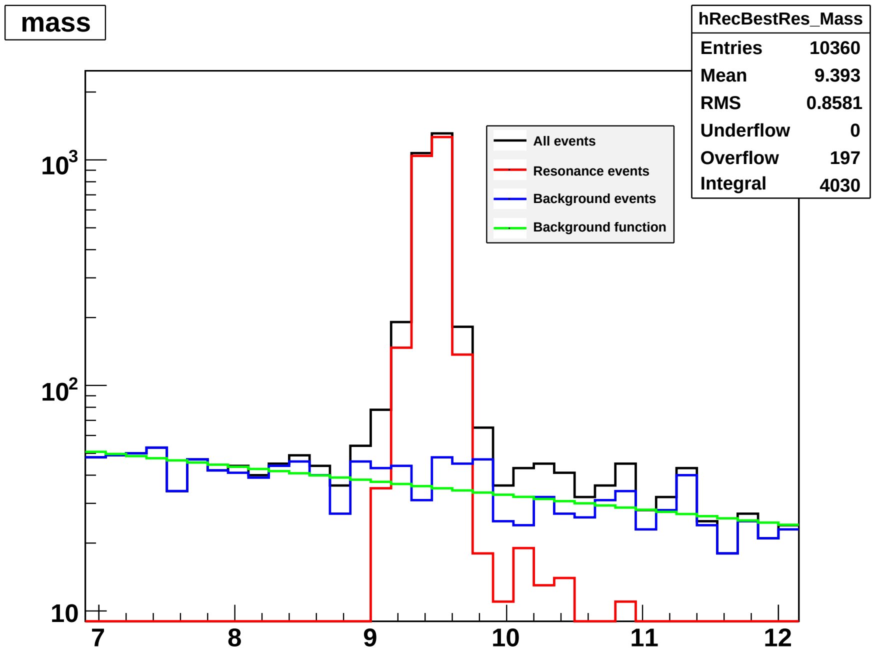



Figure 11: Background function fitted (green) vs background and signal distribution. The y axis is in logarithmic scale.

on a sample of 53317 InclusivePPmuX events (those include also prompt $J/\Psi$, but not $\Psi 2S$) and 3131 $\Upsilon 1S$. The $\Upsilon 1S$ correspond to approximately $0.3pb^{-1}$, while the InclusivePPmuX to about $0.09pb^{-1}$. The results of the background fits are shown in figure 12. After the background fit we perform a scale fit to recover the bias we introduced. The results are shown in figure 14.

The background fit gave an estimate of the fraction good enough to allow the scale fit to correct the bias.

# 7 $1pb^{-1}$

We fit a bias (same as above) with all the $\Upsilon$s and the $Z$. Considering the cross sections and the efficiency to have 2 reconstructed global muons, the expected number of $\Upsilon$s and $Z$ are shown in table **??** We perform a background and a

| Particle | Expected events in $1pb^{-1}$ |
|---|---|
| $Z$ | 1184 |
| $\Upsilon(1S)$ | 10873 |
| $\Upsilon(2S)$ | 5147 |
| $\Upsilon(3S)$ | 1477 |

scale fit in this order and we find background fractions consistent with 0 (within the error). The result of the scale fit is shown in figures **??**. The normalizations

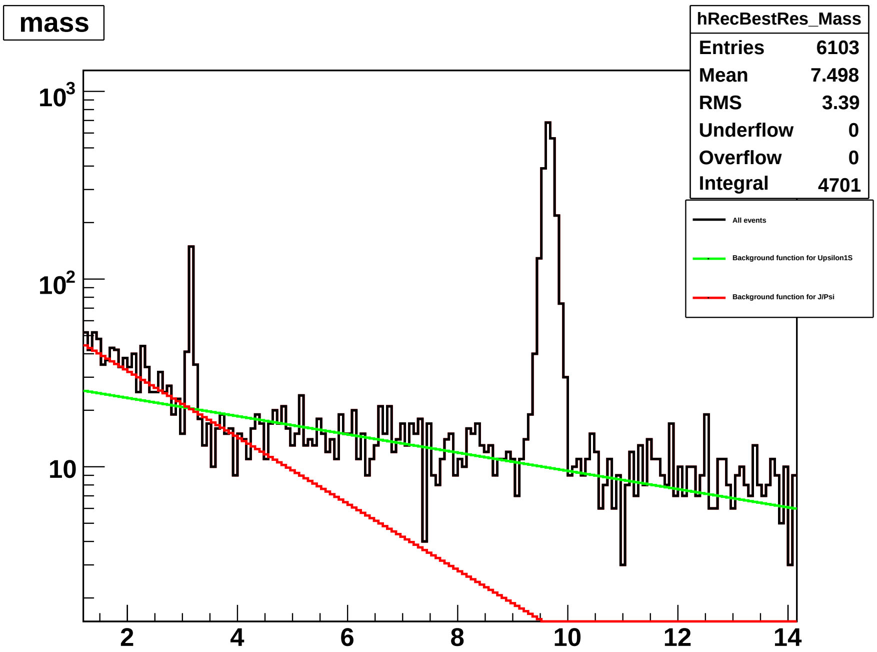



Figure 12: Background fit for $\Upsilon 1S$ and $J/\Psi$

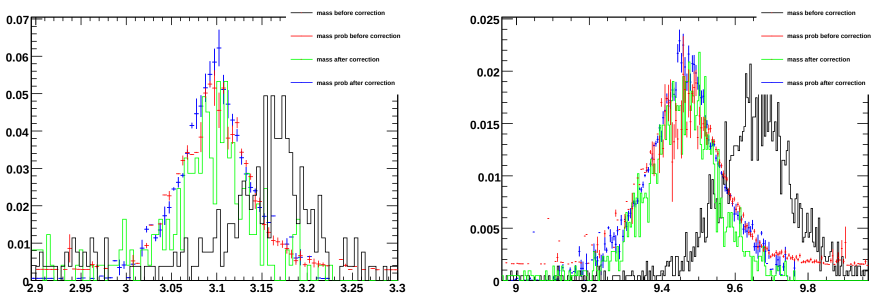



Figure 13: Mass vs probability comparison before and after the scale fit.

of the probability functions are different than those of the mass because the probability functions are all normalized to one independently. They do not take into account relative weights based on the cross sections. All the weights are at this point set to 1. 14. The scale fit is able to recover the bias, both on the $\Upsilon$s

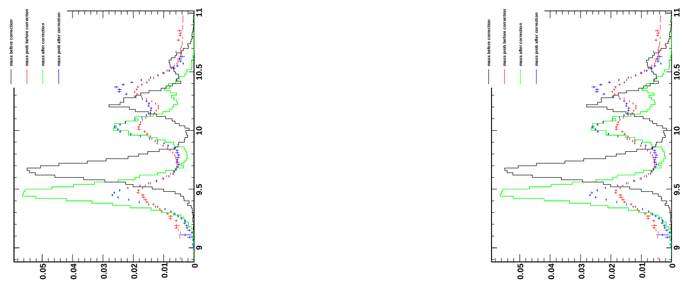


Figure 14: Mass vs probability comparison before and after the scale fit for (a) $/Upsilon$s and (b) $Z$.

and the $Z$ showing that it is capable of addressing to different $p_t$ ranges.

# 8 Magnetic Field effects

We test the effect of discrepancies between the magnetic field map and the real magnetic field. We use the two different magnetic fields

# 9 Appendix B

This section describes the functionalities of the algorithm and how to select them in a configuration file.

First of all, the algorithm needs to know what kind of muons it should use. StandAlone muons and Tracker muons have very different resolutions and are handled differently by the algorithm.

```
process.looper = cms.Looper(
    "MuScleFit",
    process.MuonServiceProxy,

# Choose the kind of muons you want to run on
# ------------------------------------------

    # // global muons //
    MuonLabel = cms.InputTag("muons"),
    MuonType = cms.int32(1),

    # // standalone muons //
    # MuonLabel = cms.InputTag("standAloneMuons:UpdatedAtVtx"),
    # muonType = cms.int32(2),

    # // tracker tracks //
    # MuonLabel = cms.InputTag("generalTracks"), # ctfWithMaterialTracks
    # muonType = cms.int32(3),
```

The likelihood settings are used to select the number of loops. One must then specify which kind of fits must be done in each iteration.

```
# Likelihood settings
# ------------------
    maxLoopNumber = cms.untracked.int32(2),
# Select which fits to do in which loop (0 = do not, 1 = do)
    doResolFit =      cms.vint32(0, 0),
    doScaleFit =      cms.vint32(1, 0),
    doBackgroundFit = cms.vint32(0, 0),
```

Select any bias to be applied to the muons at startup. The bias functions use the same set as the scale functions and they are implemented and documented in the "Functions.h" file in MuonAnalysis/MomentumScaleCalibration/interface. In the case of no bias, the function $type = 0$ must be used.

```
# BiasType = 1 means linear bias on the muons Pt
# the two parameters are the constant and the Pt
# coefficient in this order.
# ---------------------------------------------
    BiasType = cms.int32(1),
    parBias = cms.vdouble(1.015, 0.001),
```

The same approach is used to apply smearings to the muons.

```
# SmearType=0 means no smearing applied to muon momenta
# -----------------------------------------------------
    SmearType = cms.int32(0),
    parSmear = cms.vdouble(),
```

The following block of code shows how to select the resolution function and set its parameters. The additional parameters in `parResolFix` are used to fix some parameters so that the algorithm will not modify them. In the example the parameters for $\sigma \cot\theta/p_T$ and $\sigma\phi/p_T$ are fixed. The `parResolOrder` are used to run multiple iterations in the same event loop successively releasing parameters. In the example the $\sigma p_T/p_T$ are released in the order specified, starting from the first one (0) and moving to the other two in two successive iterations. Each time increasing the number of free parameters.

```
# ------------------------ #
# Resolution fit parameters #
# ------------------------ #
# The eleven parResol parameters of resolfittype=8 are respectively:
# constant of sigmaPt, Pt dep. of sigmaPt,
# scale of the eta dep. made by points with values derived from MuonGun.
# constant of sigmaCotgTheta, 1/Pt dep. of sigmaCotgTheta, Eta dep. of
# sigmaCotgTheta, Eta^2 dep of sigmaCotgTheta;
# constant of sigmaPhi, 1/Pt dep. of sigmaPhi, Eta dep. of sigmaPhi,
# Eta^2 dep. of sigmaPhi.
# ---------------------------------------------------------------
    ResolFitType = cms.int32(8),
    parResol = cms.vdouble(-0.003, 0.000205, 1.0,
                           0.00043, 0.0041, 0.0000028, 0.000077,
                           0.00011, 0.0018, -0.00000094, 0.000022),
    parResolFix = cms.vint32(0, 0, 0,
                             1, 1, 1, 1,
                             1, 1, 1, 1),
    parResolOrder = cms.vint32(0, 1, 2,
                               0, 0, 0, 0,
                               0, 0, 0, 0),
```

For the scale function the same considerations as for the resolution function apply. In this example a simple linear function is fitted.

```
# -------------------- #
# Scale fit parameters #
# -------------------- #
# Fit a linear Pt scale correction with parameters:
# Pt scale and Pt slope.
# ------------------------------------------------
ScaleFitType = cms.int32(1),
parScaleOrder = cms.vint32(0,0),
parScaleFix = cms.vint32(0,0),
parScale = cms.vdouble(1.0, 0.0),
```

Same structure also for the background fit. Note that if no background fit is to be done, the values of the background fraction (which will be fixed in the algorithm) must be 0.

```
# ------------------------ #
# Background fit parameters #
# ------------------------ #
# The two parameters of BgrFitType=2 are respectively:
# bgr fraction, (negative of) bgr exp. slope, bgr constant
# -------------------------------------------------------
    BgrFitType = cms.int32(2),
    parBgr = cms.vdouble(0., 0.),
    parBgrFix = cms.vint32(1, 1),
    parBgrOrder = cms.vint32(0, 0),
```

This last section shows how to select the resonances to fit, the fit strategy (2 is more accurate but more time consuming) and the speedup options. When speedup is selected the gen-sim data are not used and the comparison histograms will be empty.

```
# ---------------- #
# Select resonance #
# ---------------- #
# The resonances are to be specified in this order:
# Z0, Y(3S), Y(2S), Y(1S), Psi(2S), J/Psi
# -------------------------------------------------
    resfind = cms.vint32(1, 0, 0, 0, 0, 0),
    FitStrategy = cms.int32(2),

    speedup = cms.bool(False),
# Set this to false if you do not want to use simTracks.
# (Note that this is skipped anyway if speedup == True).
    compareToSimTracks = cms.bool(True),
# readPdfFromDB = cms.bool(False)
)
```