Masterhesis: Development of an improved fitting routine. Classification and Reweighting of the $B \to K^* \mu\mu$ decay

Supervisors: Prof. Dr. Nicola Serra, Dr. Marcin Chzsaszcz

Author: Oliver Dahme

Here comes the abstract

Contents

1. Introduction

The work presented in this thesis consits of three independant parts:
First a new fitting routine has been implemnted, as a more transperent alternative to the broadly used routine Minuit [2].
Second a test of classifiers has been performed. Classifiers are a type of algorithm used to seperate signal from background. The test was performed to find the best classifier, for the analysis of the $B_0 \to K^* \mu^+ \mu^-$ decay.
Third data from the Run II of the LHCb detector was used to reweight the Monte Carlo simulation of the $B_0 \to K^* \mu^+ \mu^-$ decay. The reasons for choosing this decay are explained in section 4.1

1.1. New fitting routine

In particle physics the decay rate distributions depend on kinematic variables, the angles between the final state particles and their energies. To get the angular coefficients of the decay rate the experimentally obtained decay rates are fitted. So far this has been done with the broadly used algorithm MINUIT [2]. But like explained in section 2 MINUIT can make random fatal errors while fitting, without any obvious explanation. The second problem with MINUIT is the intransperency: One can not comprehend how the fit is performed or if it really has found the global minimum.
The new algorithm implemented as part of this thesis, is called RooMCMarkovChain. It is based on a Monte Carlo Markov Chain (see section 2.1), where one link in the chain respresents a point on the log-likelihood function of the fit. The chain is designed to follow the global minimum and to jump out of local minima. This provides two things: A fitting parameter set, for example the angular coefficents of a decay rate, at the global minimum. And a scan of the log-likelihood function in the vicinity of the global minimum, which can be used for the error estimation and can be send along side papers containing fits to make it transperent for the reader how the fit performed.

1.2. Test of classifiers

Raw data from a particle detector contains all kind possible decays. The first challenge of an anlysis is to distinguish the one decay one wants to analyse from all the others. The common prcedure to reach that goal in our days the usage of Machine Learning algorithms called Classifiers. The need two things befor performing the analysis: A training dataset where signal and background is labeld accordingly. And a list of parameters that could be different in signal and background. While training the algorithm will adapt thresholds on these parameters. With these thresholds the algorithm is able to distinguish signal from background in unlabeled data.
The test of classifiers presented in section 5 has the goal to find the best classifier to distinguish $B_0 \to K^* \mu^+ \mu^-$ decays in the raw data of the LHCb detector.

1.3. Reweighting

In data analysis on has to take into account the efficiencies of all the detector components.Those efficiencies can be obtained by Monte Carlo simulations. The problem is that the simulation has to be the same as data to get the correct answers. Since resimulating everything with different parameters until one hits the data distribution would take too much resources. One can simply reweight the simulation to match the data. As an initial weight the sWeights see equation 36 are used. The parameters used to reweight the MC samples are applied in the following order: 'nTracks', 'B_0 p_T' and the quality of the $K\pi\mu\mu$ vertex. The new weights derived by the difference in data and simulation of these parameters are used to weight all the MC samples. For this analysis the MC and Data of the $K^* \to J/\Psi K^{*0}$ are used, because it is a very clean channel.

2. RooMCMarkovChain

RooMCMarkovChain is a new class for the ROOT Data Analysis Framework [1]. You can find the install-instructions at [1]. It works best on a unix based operation system.

RooMCMarkovChain uses a metropolis algorithm as a minimizer for the negative log likelihood function. The metropolis algorithm is based on a Monte Carlo Markov Chain and can therfore easily be scaled to multidimensional parameter space and moreover, such kind of algorithms can easily be parallelized.

2.1. Monte Carlo Markov Chain (MCMC)

A Markov Chain is a random process which undergoes several states. From each state there is a probability ditribution to change into another state or to stay. Most important i the asumption that every next step just depends on the current state. The figure 1 illustrates the behavior of such a Markov Chain.

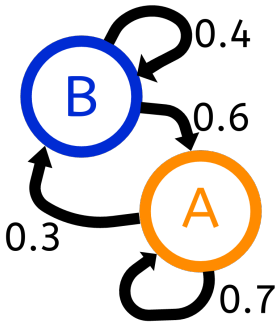


Figure 1: Illustration of states and the probability distribution to change or the stay in a state of a Monte Carlo Markov Chain. In state A there is a 0.3 probability to change into state B and a 0.7 probability to stay in state A.

2.2. Metropolis Hastings

Metropolis Hastings is a so called Monte Carlo updater. It updates the probabilitydistribution when a change of states is proposed. That is very usefull in the cases where the states are not discrete like in figure 1 but continous. Suppose that a specified distribution has unnormalized density h. The Metopolis Hastings update does the following:

1. The current state x is proposed to move to another state y with a conditional probability given x denoted as $q(x, \cdot)$.
2. The Hastings ratio is calculated:
$$r(x, y) = \frac{h(y) \cdot q(y, x)}{h(x) \cdot q(x, y)} \tag{1}$$
3. The proposed move to y is accepted with the probability $a(x, y)$:
$$a(x, y) = min(1, r(x, y)) \tag{2}$$

This principle is used in the RooMCMarkovChain class to minimize the negative log likelihood function. In detail a robust adaptive Metrpolis hastings process is implemented. It is defined as:

1. Compute next proposed state Y_n as a random change from the current state X_{n-1}:
$$Y_n \equiv X_{n-1} + S_{n-1} U_n$$
, where U_n is an independent random vector.
2. With the probability α_n the proposal is accepted and $X_n = Y_n$. Otherwise the proposal is rejected and $X_n = X_{n-1}$
$$\alpha_n \equiv min\{1, r(X_{n-1}, Y_n)\}$$
, where r is the hastings ratio see equation 1.
3. compute the lower-diagonal matrix S_n with positive diagonal elements satisfying the equation:
$$S_n S_n^T = S_{n-1} \left(I + \eta_n (\alpha_n - \alpha_*) \frac{U_n Y_n^T}{||U_n||^2} \right) S_{n-1}^T \tag{3}$$
,where $I \in R^{dxd}$ is the identity matrix.

2.3. Features

The features of the RooMCMarkovChain class will be shown by fitting the following probability density function (pdf):

$$g(x) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} + f \cdot \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \tag{4}$$

It is so called double gaus pdf, which is just the sum of two gaussian pdfs with a fractional parameter f. The result of the fit is compared with the result of the Minuit algorithm. Therfore 1000 points were simulated fowllowing the distribution 4:

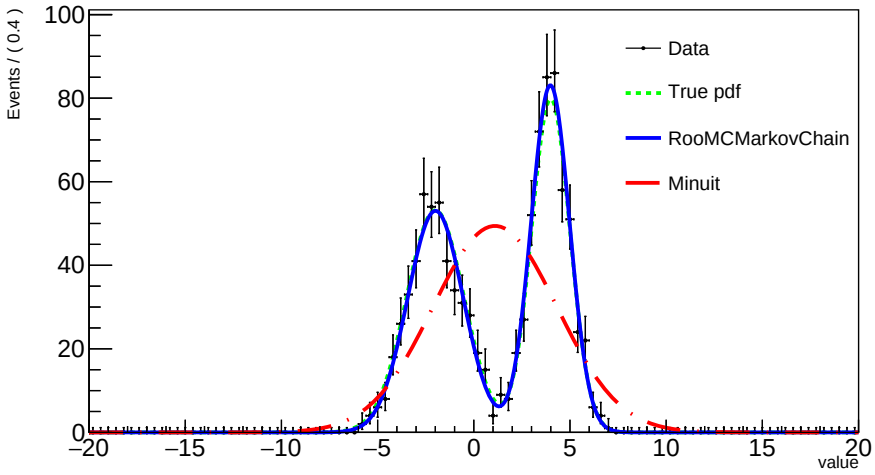


Figure 2: Fit of double gaus pdf (see equation 4) with RooMCMCarkovchain and with Minuit. The true pdf has the follwing values: $\mu_1 = 4$, $\sigma_1 = 1$, $\mu_2 = -2$, $\sigma_2 = 1.5$ and the fraction $f = 0.5$. The Minuit fit fails, because of a random error. The fits gets right if the starting parameters are changed.

As one can see in figure 2 the Minuit routine fails to fit the distribution. The reasons are unknwon, but the fit gets right if the starting parameters are changed. Intensive testing prooved RooMCMCarkovChain less sensitive to starting parameters. But in high parameters space test (up to 20) revealed that Minuit and RooMCMarkovChain are equal in performace. One should do another performace analysis with the number of parameters usually used in particle physics, with up to 50 parameters.
The terminal output, for the fit in figure 2, gives the estimated parameters with error interval and correlation coefficents of each parameter pair.

```
RooFit v3.60 -- Developed by Wouter Verkerke and David Kirkby
                Copyright (C) 2000-2013 NIKHEF, University of California & Stanford University
                All rights reserved, please read http://roofit.sourceforge.net/license.txt

Starting Monte Carlo Markov Chain Fit with 12000 points and cutoff after 7000 points
1% 2% 3% 4% 5% 6% 7% 8% 9% 10% 11% 12% 13% 14% 15% 16% 17% 18% 19% 20% 21% 22% 23% 24% 25% 26%
% 33% 34% 35% 36% 37% 38% 39% 40% 41% 42% 43% 44% 45% 46% 47% 48% 49% 50% 51% 52% 53% 54% 55% 5
 62% 63% 64% 65% 66% 67% 68% 69% 70% 71% 72% 73% 74% 75% 76% 77% 78% 79% 80% 81% 82% 83% 84% 85
91% 92% 93% 94% 95% 96% 97% 98% 99% 100%
NO.     NAME    VALUE           ERROR
1       frac    5.14084e-01     1.47307e-02
2       mean1   3.98243e+00     5.07205e-02
3       mean2    -1.99256e+00   7.51154e-02
4       sigma1  9.98988e-01     3.87089e-02
5       sigma2  1.44724e+00     5.86681e-02

CORRELATION COEFFICIENTS
NO.     1       2       3       4       5
1       1.000   -0.072  -0.043  0.095   -0.060
2       -0.072  1.000   0.131   -0.151  0.136
3       -0.043  0.131   1.000   -0.135  0.194
4       0.095   -0.151  -0.135  1.000   -0.171
5       -0.060  0.136   0.194   -0.171  1.000
```

Figure 3: Terminal output of the RooMCMarkovChain fit.

This gives a good initial overview of results after the fit has finished. The error calculation can be set to assume gaussian or non-gaussian errors.

In addition several other properties of the parameters can be obtainded:

1. The 1 dimensional profile of the negative log liklihood function for a given parameter.

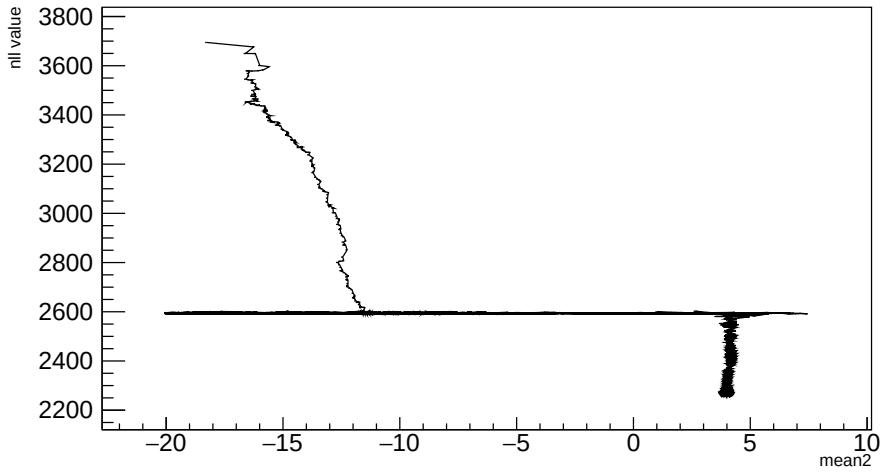


Figure 4: Profile of the negative log lilihood function for μ_2 in equation 4. There is a local minimum at the nll value of 2600.

   From this plot on can see how the algorithm reached the minimum of the negative log likelihood function and if there are local minima.

2. The walk distribution of a given parameter.

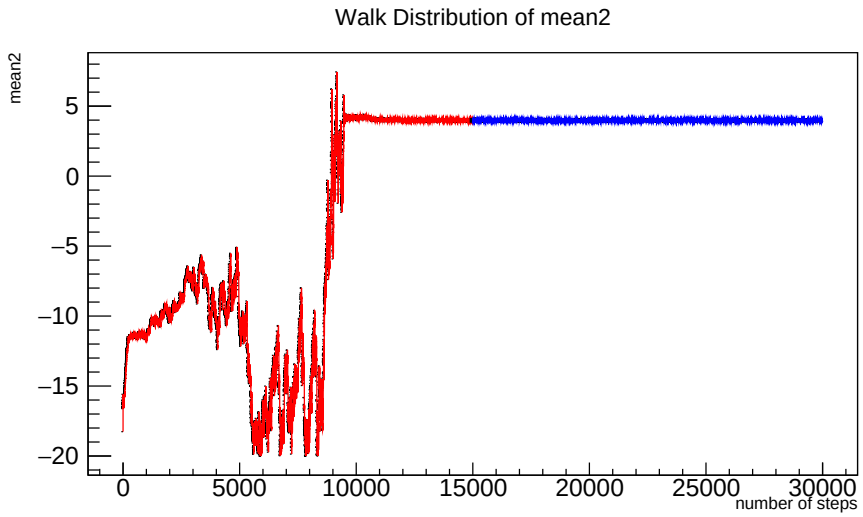


Figure 5: Walk distribution of μ_2 in equation 4. The red points are not considered for the error calculation.

This plot is very important for handling the RooMCMarkovChain class. Since the error calculation is based on the variance of the walk distribution, cutting of points in the begginning greatly reduces this variance. The user has to choose how many points are cut off. Plotting the walk distribution of all parameters helps choosing the right amount.

3. The walk distribution of a given parameter as a histogramm.

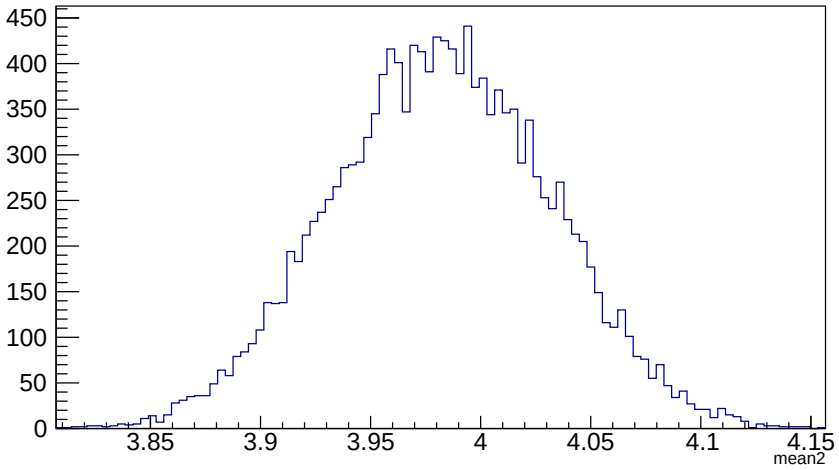


Figure 6: Walk distribution of μ_2 in equation 4 as a histogramm.

This plot can be used to check, which error strategy should be used. If the walk distribution histogramm is gaussian, gaussian errors can be assumed.

4. The scatterplot between two given parameters.

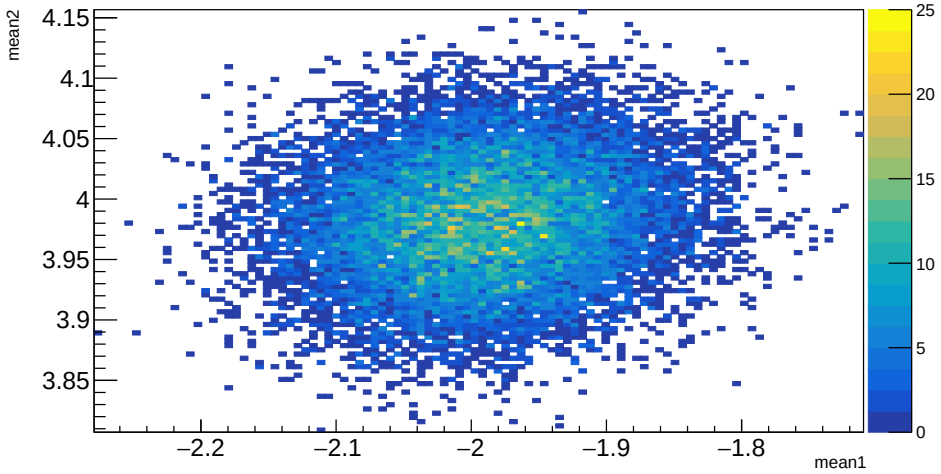


Figure 7: Scatterplot between μ_1 and μ_2 in equation 4.

This plot shows the correlation between two parameters graphically.

3. The LHCb Experiment

In this section the experimental setup of the detector is presented, which recorded the data used in this thesis. The Large Hadron Collider beauty experiment (LHCb) is one of four large experiments based at the CERN laboratory near Geneva in Switzerland. It is part of the Large Hadron Collider (LHC), a proton-proton accelerator and collider located in a vast unterground tunnel with 26.7 km circumference beneath the Swiss-French countryside. The other three experiments are CMS and ATLAS which are dedicated to a wide range of physics and have therefor very large detectors. ALICE investigates quark-gluon plasma and therefor needs geavy ion collisions, instead of proton collisions.
The protons in the LHC have a kinetic energy of 7 TeV, which allows a collision energy, in the LHCb detector, of 13 TeV. In the year 2016 the LHCb had a recorded luminosity of 1906 pb^{-1}. For this thesis 2280 pb^{-1} of data, collected at LHCb during the years 2011 to 2016 are used. LHCb is dedicated to falvour physics. It therefor investigates rare decays and CP violation in beauty and charm hadrons.

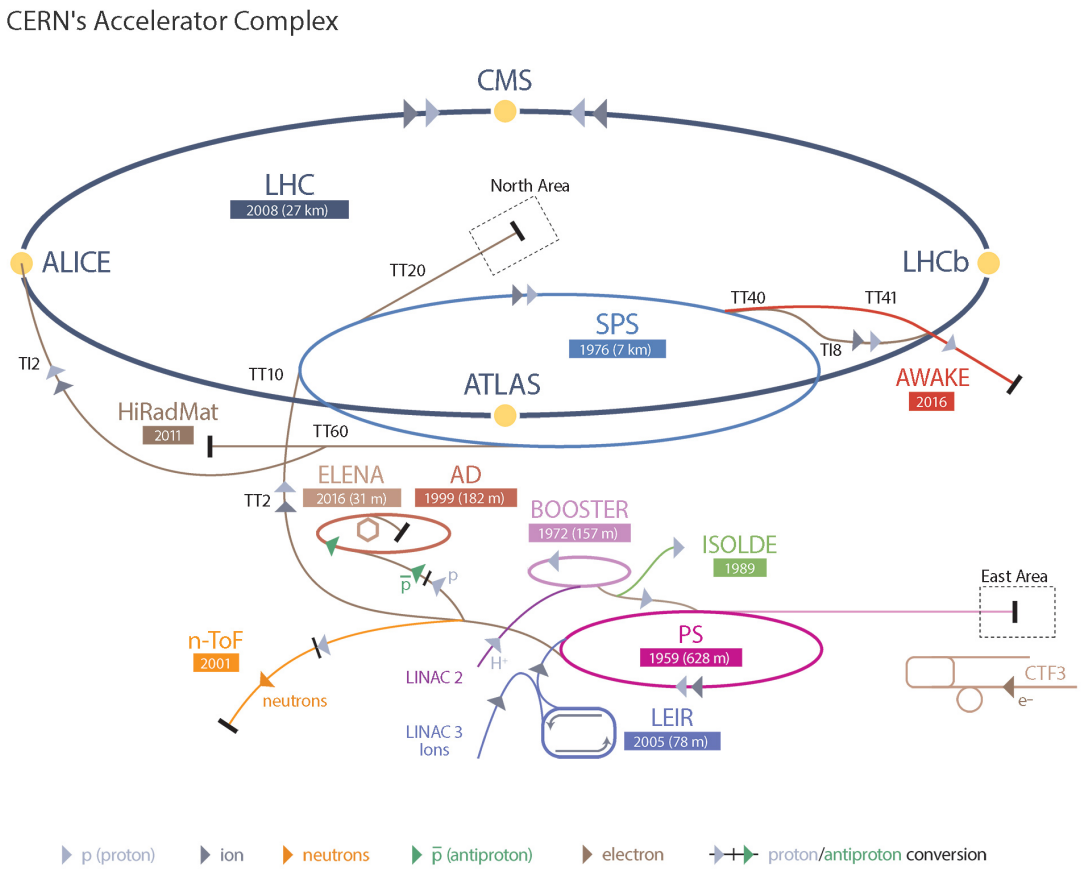


Figure 8: CERN's Accelerator Complex.[6] The protons get injected in the lineare accelerator LINAC2. Then they get pre-accelerated in 3 synchrotons (BOOSTER,PS,SPS) where the protons reach a kinetic energy of 450 GeV. That is the entering energy of the LHC which accelerates them futher up to 7 TeV, before they collide at the four detectors: CMS, ATLAS, LHCb and ALICE.

3.1. The LHCb Detector

The LHCb Detector has a fix target geometry, because beauty hadrons are manily produced at small angeles with respect to the beam pipe.

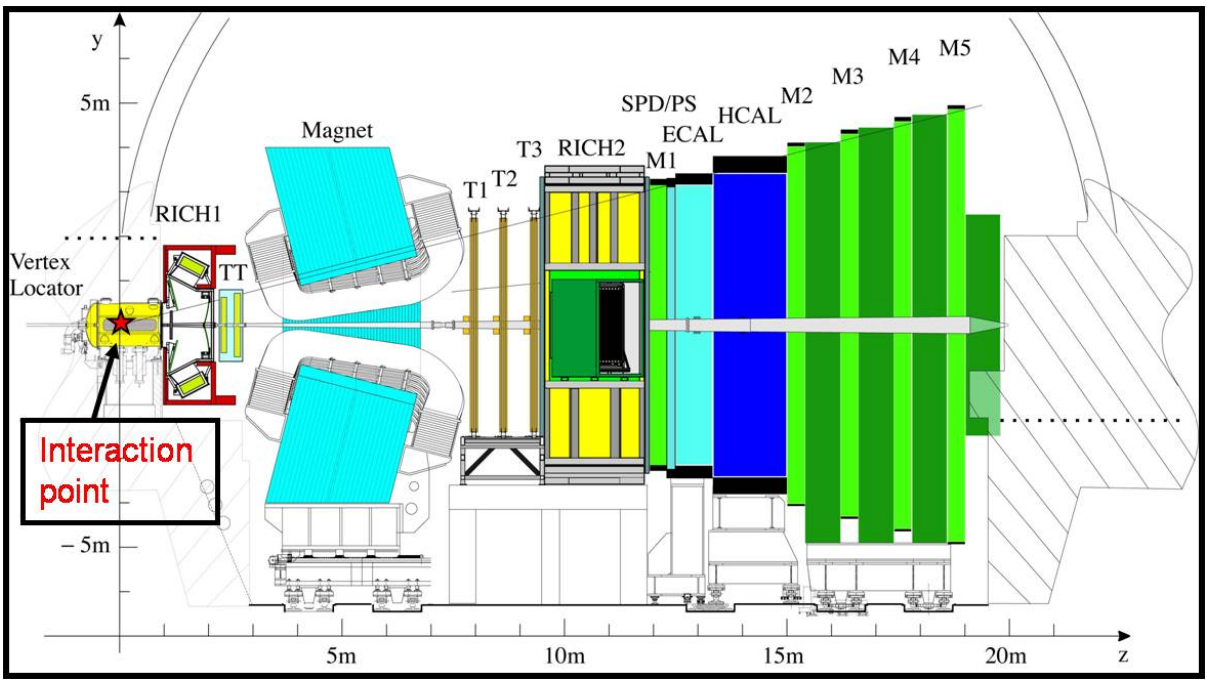


Figure 9: Basic layout of the LHCb detector [7]. The interaction point is inside the vertrex detector and the beam pipe passes through the center. The different subdetecors are the two Ring Imaging Cherenkov Detecors (RICH1 and RICH2), the tracking stations (TT and T1 to T3), the scintillator pad detector (SPD), the preshower electromagnetic calorimeter (ECAL), the hadronic calorimeter (HCAL) and the muon stations (M1-M5).

VErtex LOcator (VELO) [8] : Velo picks out B mesons from the multitude of other particles produced. This is a complex task since B mesons have very short livetimes spent colse to the beam. The VELO's silicon detecor elements must be placed at a distance of just five milimetres to the interaction point. To prevent damage to the detector during beam injection and stabilization it is mechanically moved to a safe distance. Velo measures B mesons indirectly be detecting its decay particles, nevertheless it has a resolution of 10 microns. .

Ring Imaging Cherenkov (RICH) detectors [9] : The RICH detectors meeasure the emission of Cherenkov radiation, which happens when a charged particle passes through a medium faster than light does. It is a similar effect like the sonic boom a aircraft produces by breaking the sound barrier. The shape of the light cone depends on the particle's velocity, enabling the detector to determine its speed.

Magnet [10] : The big magnet of the LHCb experiment weights 27 tonnes ands is mounted inside a 1,450 tonne steel frame. This powerful magnet forces all charged particels to change there trajectory. By examining the curvature of the path one can calculate its momentum.

Trackers [11] : The LHCb's tracking system consists of a series of four large rectangular stations, each covering an area of 40 m^2. While flying through this area charged particles will leave a trace, therefor one can estimate the trajectory of a particle. The trajectory is used to link the signals left in other detecor elements to the corresponding particle. In LHCb two different tracker technologies are used: The silicon tracker placed close to the beam pipe, uses silicon microstripes. If a charged particle passes such a stripe it collides with the silicon atoms, liverating electrons and creating an electric current, which is then recorded. The outer tracker situated further from the beam pipe consists of gas-filled tubes. The gas ionizes when a charged particle hits a gas molecules, producing electrons. These reach an anode wire situated in the centre of each tube. The position of the track is found by timing how long it takes

electrons to reach it.
Calorimeters [12] : Calorimeters stop particles as they pass through, measuring the amount of energy lost. In LHCb there are two different types: The elctromagnetic calorimeter responsible for light particles like electrons and photons and the hadronic calorimeter responsible for heavier particles containing quarks. Both have a sandwich-like structure, with alternating layers of metal and plastic plates. If a particles hits a metal plate it produces a shower of secondary particles. These will excite polystyrenne molecules in the plastic plates, which then emit ultraviolet light. The energy lost by the particle in the metal plate is propoertional to the amount of UV light produced in the platic plates.
Muon System [13] : The muon system consistes of 5 rectangular stations, which cover an area of 435 m^2. Each station has chambers filled with three gases: carbon dioxide, argon and tetrafluoromethane. Passing muons react with the mixture and electrodes detect the result.

3.2. The LHCb trigger system

The rate of events at the LHCb interaction point is 40 MHz. But the rate to have a B meson contained in the detector is 15 kHz. But the offline computing power just allows 2 kHz to be recorded. The LHCb trigger system aims to 'fill' this 2 kHz with intresting B decays and important control decays like J/ψ decays. The trigger has two levels:
The **Level Zero (L0)** trigger reduces the beginning 40 MHz to 1 MHz. To get this high rate it can only rely on fast sub-detectors as the calorimeters and the muon system. The L0 trigger looks for events with high transverse momentum with respect to the patrticle beam axis (pT), because particles from a B decay have this attribute, since B Mesons are always produced almost parallel to the beam axis. In addition the L0 trigger performs a simplified vertex reconstruction with the signal of two silicon layers of the VELO to identifie events with multple proton-proton collisions. They are rejected because for this kind of events its much more difiicult to reconstruct B meson decays, since it is harder to distinguish primary and secondary vertex of the B decay.
The **High Level Trigger (HTL)** is an algorithm that runs on a farm of 1000 16-core computers. It has two stages: HLT1 which reduces the event rate to a few tens of kHz and HLT2 which reduces the rate to the 2 kHz which are recorded. HLT1 gets all the candidates of the L0 trigger and uses the full detector information on them to search for particles with a high impact parameter with respect the proton-proton collision. These particles are most likly decay products from B mesons, because of its relatively long life-time. They typically fly 1 cm away from the collision before decaying resulting in a high impact parameter for the decay products. HLT2 does a complete reconstruction of the events. It starts with the track of the VELO and connects them to the tracks in the other sub-detectors. Most important are displaced vertices, since they are strong indicator for B decays. The selection is devided into two parts. The inclusive selection searches for resonance decays like D^* or J/ψ. The exclusive selection is desigened to provide the highest possible efficiency to fully reconstruct B decays of interest. It therfor uses all information available such as mass and vertex quality and intermediate resonances.

4. The $B \to K^* \mu \mu$ decay

This section describes the motivation to analyse this decay, then the operators needed to calculate the different couplings are derived and in the last section the effetive Wilson coefficients are derived. This section is mostly theoretical and should give an overview over the decay.

4.1. Motivation to analyse $B \to K^* \mu \mu$ decays

In the Standart Model of particle physics the different leptons, Electron, Muon and Taoun only differ in there masses. Therfore if these leptons have a high energy (TeV) where the mass becomes negligible the leptons all behave the same. This phenomen is called lepton universality. For a long time it was one of the ground pillars of the Standart Model. But recent experimental results of the LHCb collaboration [3] suggest a violation of the lepton universality. They measured the branching fraction of four B_0 deacys with a b quark to s quark transistion and two leptons in the final state. To reduce systematic uncertainties the following double ratio of these branching fractions has been considered a well defined test of lepton universaility. Since leptons at high energies should behave the same, this double ratio is expected to be equal to one.

$$\begin{aligned} R_{K*0} &= \frac{\mathcal{B}(B^0 \to K^{*0}\mu^+\mu^-)}{\mathcal{B}(B^0 \to K^{*0}J/\psi(\to \mu^+\mu^-))} \Big/ \frac{\mathcal{B}(B^0 \to K^{*0}e^+e^-)}{\mathcal{B}(B^0 \to K^{*0}J/\psi(\to e^+e^-))}\,, \\ R_{K*0} &= \begin{cases} 0.66^{+0.11}_{-0.07}(\text{stat}) \pm 0.03(\text{syst}) \text{for} 0.045 < q^2 < 1.1 \text{GeV}^2/c^4, \\ 0.69^{+0.11}_{-0.07}(\text{stat}) \pm 0.05(\text{syst}) \text{for} 1.1 < q^2 < 6.0 \text{GeV}^2/c^4. \end{cases} \end{aligned} \tag{5}$$

The mesurement shows a 2.1-2.3 and 2.4-2.5 σ deviation from the Standart Modell in the two q^2 regions, respectively. To investigate further the same measurment is performed with new data from the LHCb detector. The goal of this thesis is to contribute to this new measuremnt.

4.2. Kinematics

In this section the decay itself and its kinematics are explained: The Decay is a flavor changing neutral current (FCNC) with four charged particles in the final state. The FCNC is a current, which changes the falvor of a fermion without changing its electric charge. The four particles in the final stage are:
The K^+ and π^- from the K^* decay and two leptons from the loop or box diagrams:

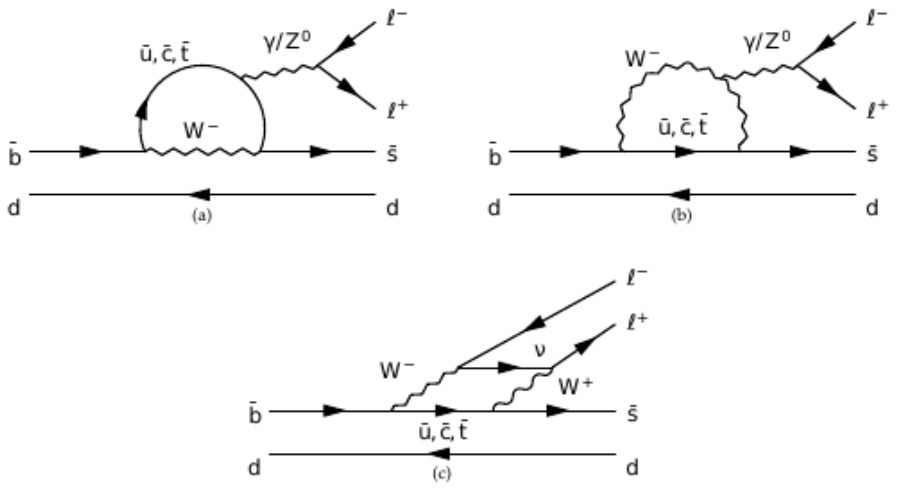


Figure 10: Feynman diagrams for decay $B_d \to \mu^+ \mu^-$ at lowest order

The kinematics of the decay are defined by the three angels θ_K, θ_L and ϕ, shown in figure 11 and the invariant di-muon mass square q^2.

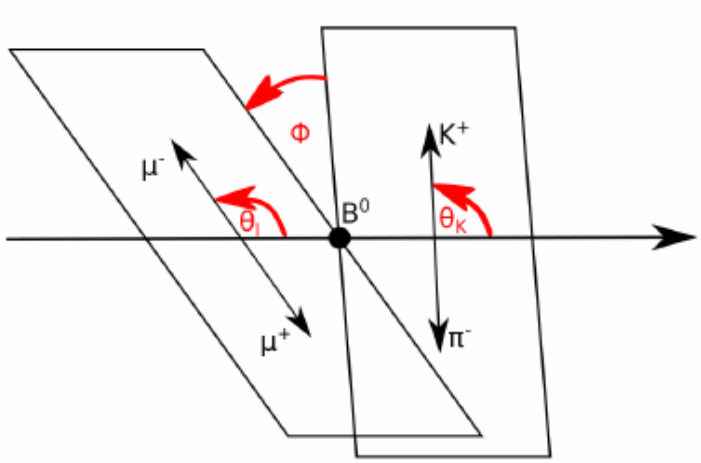


Figure 11: kinematic variables of the decay $B^0 \to K^{*0} \mu \mu$

The differential decay rate for the B^0 Meson can be expressed in terms of these variables:

$$\begin{aligned}
\frac{d^4\Gamma}{d\cos\theta_L d\cos\theta_K d\phi dq^2} &= \frac{9}{32\pi} I(q^2, \theta_L, \theta_K, \phi) \\
\text{with: } I(q^2, \theta_L, \theta_K, \phi) &= I_1^S \sin^2(\theta_K) + I_1^C \cos^2\theta_K + \left(I_2^S \sin^2\theta_K + I_2^C \cos^2\theta_K\right) \cos^2\theta_L \\
&+ I_3 \sin^2\theta_K \sin^2\theta_L \cos 2\phi + I_4 \sin 2\theta_K \sin 2\theta_L \cos\phi \\
&+ I_5 \sin 2\theta_K \sin\theta_L \cos\phi \\
&+ \left(I_6^S \sin^2\theta_K + I_6^C \cos^2\theta_K\right) \cos\theta_L + I_7 \sin 2\theta_K \sin\theta_L \sin\phi \\
&+ I_8 \sin 2\theta_K \sin 2\theta_L \sin\phi + I_9 \sin^2\theta_K \sin^2\theta_L \sin 2\phi
\end{aligned} \tag{6}$$

This decay rate is defined as the probability per unit time that the particle will decay. The I_i can be obtained by fitting this decay rate to the kinematic variables obtained by the detector. This fit is usually done by a standart fitting routine called MINUIT [2]. In section 2 an alternative is presented.

4.3. Operators for $B \to X_s l^+ l^-$ decays

In this section all the operators of the decay are derived from the effective Lagrangian. the operators in Quantum field theory are used to create or destroy particles, by applying them to a quantum field. This section should just give an overview if you need more detailed information consider reading [4].
The effective Lagrangian for $B \to X_s l^+ l^-$ decays has the form:

$$\begin{aligned}\mathcal{L}_{eff} &= \mathcal{L}_{QCD,QED}(u,d,s,c,b,e,\mu,\tau) \\ &+ \frac{4G_F}{\sqrt{(2)}}\left[V_{us}^* V_{ub}(C_1^c P_1^u + C_2^c P_2^u) + V_{cs}^* V_{cb}(C_1^c P_1^c + C_2^c P_2^c)\right] \\ &+ \frac{4G_F}{\sqrt{(2)}}\sum_{i=3}^{10}\left[(V_{us}^* V_{ub} + V_{cs}^* V_{cb})C_i^c + V_{ts}^* V_{tb} C_i^t\right] P_i.\end{aligned} \tag{7}$$

The first term in equation 7 contains the kinetic terms of the light SM particles as well as their QCD and QED interactions. The remaining two terms consist of $\Delta B = -\Delta S = 1$ local operators of dimension $(d \leq 6)$, which contain those light fields. The mass of the s quark can be negleted in comparisson with the b mass. One gets the following operators:

$$\begin{aligned}
\mathcal{O}_1^u &= (\bar{s}_L \gamma_\mu T^a u_L)(\bar{u}_L \gamma^\mu T^a b_L), & \mathcal{O}_6 &= (\bar{s}_L \gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3} T^a b_L)\sum_q (\bar{q}\gamma^{\mu_1}\gamma^{\mu_1}\gamma^{\mu_1} T^a q), \\
\mathcal{O}_2^u &= (\bar{s}_L \gamma_\mu u_L)(\bar{u}_L \gamma^\mu b_L), & & \\
\mathcal{O}_1^c &= (\bar{s}_L \gamma_\mu T^a c_L)(\bar{c}_L \gamma^\mu T^a b_L), g & \mathcal{O}_7 &= \frac{e}{g^2} m_b (\bar{s}_L \sigma^{\mu\nu} b_R) F_{\mu\nu}, g \\
\mathcal{O}_2^c &= (\bar{s}_L \gamma_\mu c_L)(\bar{c}_L \gamma^\mu b_L), & \mathcal{O}_8 &= \frac{1}{g} m_b (\bar{s}_L \sigma^{\mu\nu} T^a b_R) G_{\mu\nu}^a, \\
\mathcal{O}_3 &= (\bar{s}_L \gamma_\mu b_L)\sum_q (\bar{q}\gamma^\mu q), & & \\
\mathcal{O}_4 &= (\bar{s}_L \gamma_\mu T^a b_L)\sum_q (\bar{q}\gamma^\mu T^a q), & \mathcal{O}_9 &= \frac{e^2}{g^2}(\bar{s}_L \gamma_\mu b_L)\sum_l (\bar{l}\gamma^\mu l), \\
\mathcal{O}_5 &= (\bar{s}_L \gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3} b_L)\sum_q (\bar{q}\gamma^{\mu_1}\gamma^{\mu_1}\gamma^{\mu_1} q), & \mathcal{O}_{10} &= \frac{e^2}{g^2}(\bar{s}_L \gamma_\mu b_L)\sum_l (\bar{l}\gamma^\mu \gamma_5 l),
\end{aligned} \tag{8}$$

where $\sum_q$ and $\sum_l$ denote the sums over light quarks and all leptons, respectivly.

4.4. Wilson coefficients

In this section the Wilson coefficents are derived from the effective Hamiltonion of the deacy. The effective Hamiltonian for $b \to s\mu^+\mu^-$ transitions can be written as:

$$H_{eff} = -\frac{4G_F}{\sqrt{2}}\left(\lambda_t H_e f f^{(t)} + \lambda_u H_{eff}^{(} u)\right) \tag{9}$$

The λ_i can be expressed with CKM combinations $\lambda_i = V_{ib}V_{is}^*$.

$$H_{eff}^{(t)} = C_1\mathcal{O}_1^c + C_2\mathcal{O}_2^C + \sum_{i=3}^{6} C_i\mathcal{O}_i + \sum_{i=7,8,9,10,P,S} (C_i\mathcal{O}_i + C_i'\mathcal{O}_i') \tag{10}$$

$$H_{eff}^{(u)} = C_1(\mathcal{O}_1^C - \mathcal{O}_1^u) + C_2(\mathcal{O}_2^C - \mathcal{O}_2^u). \tag{11}$$

The contribution of $H_{eff}^{(u)}$ has a double Cabibbo supression and is therfore usually dropped. It is kept here since it is sensitive to complex phases of decay amplitudes. The operators $P_{i\leq 6}$ are the same as for general $B \to X_s l^+ l^-$ decays, see equation 8. The remaining ones are given by:

$$\begin{aligned}
\mathcal{O}_7 &= \frac{e}{g^2} m_b(\bar{s}\sigma_{\mu\nu}P_R b)F^{\mu\nu}, & \mathcal{O}_7' &= \frac{e}{g^2} m_b(\bar{s}\sigma_{\mu\nu}P_L b)F^{\mu\nu},\\
\mathcal{O}_8 &= \frac{1}{g} m_b(\bar{s}\sigma_{\mu\nu}T^a P_R b)G^{\mu\nu a}, & \mathcal{O}_8' &= \frac{1}{g} m_b(\bar{s}\sigma_{\mu\nu}T^a P_L b)G^{\mu\nu a},\\
\mathcal{O}_9 &= \frac{e^2}{g^2}(\bar{s}\sigma_\mu P_L b)(\bar{\mu}\gamma^\mu \mu), & \mathcal{O}_9' &= \frac{e^2}{g^2}(\bar{s}\gamma_\mu P_R b)(\bar{\mu}\gamma^\mu \mu),\\
\mathcal{O}_{10} &= \frac{e^2}{g^2}(\bar{s}\gamma_m u P_L b)(\bar{\mu}\gamma^\mu\gamma_5 \mu), & \mathcal{O}_{10}' &= \frac{e^2}{g^2}(\bar{s}\gamma_\mu P_R b)(\bar{\mu}\gamma^\mu\gamma_5 \mu),\\
\mathcal{O}_S &= \frac{e^2}{16\pi^2} m_b(\bar{s}P_R b)(\bar{\mu}\mu), & \mathcal{O}_S' &= \frac{e^2}{16\pi^2} m_b(\bar{s}P_L b)(\bar{\mu}\mu),\\
\mathcal{O}_P &= \frac{e^2}{16\pi^2} m_b(\bar{s}P_R b)(\bar{\mu}\gamma_5 \mu), & \mathcal{O}_P' &= \frac{e^2}{16\pi^2} m_b(\bar{s}P_L b)(\bar{\mu}\gamma_5 \mu),
\end{aligned} \tag{12}$$

where m_b denotes the running b mass in the $\overline{MS}$ scheme and g is the strong coupling constant and $P_{L,R} = (1 \pm \gamma_5)/2$. In the Standart Modell the primed Operators with opposite chirality to the unprimed operators vanish or are highly suppresd as are the $\mathcal{O}_S$ and $\mathcal{O}_P$. The contributions of $\mathcal{O}_{1,2,3,4,5,6}$ are neglected, since they are either heavily constrained or their impact turns out to be generically very small. For example in the left-right symmetric models or throughout gluino contributions in a general Minimal Supersymmetric Standard Model.

The C_i coefficients in the equations 10 and 11 are called Wilson coefficients. They encode short-distance physics and New Physics effects. For the calculation a matching scale $\mu = m_W$ is chosen, in a pertubative expansion in powers of $\alpha_s(m_W)$. Then the Wilson coefficents are evolved down to scales $\mu = m_b$ according to the solutions of the renomalization group equations. Contributions by New Physics enter through $C_i(m_W)$, while the low scales are determined by the Standart Modell. To allow a more organized expansion of the Wilson coefficients in pertubation theory the factors $16\pi^2/g^2 = 4\pi/\alpha_S$ are included into the definitions of the operators $\mathcal{O}_{i\geq 7}$. All the C_i expand as:

$$C_i = C_i^{(0)} + \frac{\alpha_s}{4\pi}C_i^{(1)} + \left(\frac{\alpha_s}{4\pi}\right)^2 C_i^{(2)} + O(\alpha_s^3) \tag{13}$$

where $C_i^{(0)}$ is the tree-level contribution, which is quale to zero for all operators except $\mathcal{O}_2$ and $C_i^{(n)}$ denotes the n-loop contributions. Before discussing the Wilson coefficents in details, lets look at the Operators again; the operators $\mathcal{O}_S'$ and $\mathcal{O}_P'$ are given in terms of conserved currents. They carry no scale-dependence. They do not mix with other operators and their Wilson coefficents are at the matching scale. $\mathcal{O}_9$ is also given by conserved curents. It mixes with $\mathcal{O}_{1,2,3,4,5,6}$ via a virtual photon decaying into $\mu^+\mu^-$. In addition their is a scale depedence from the factor $1/g^2$. This dependence is also present in C_{10} which otherwise would be scale independent.

C_7 and C_9 always appear in a particular combination with other Wilson coefficents in matrix elements.

Therfore effective coefficients are defined:

$$\begin{aligned} C_7^{eff} &= \frac{4\pi}{\alpha_s} C_7 - \frac{1}{3} C_3 - \frac{4}{9} C_4 - \frac{20}{3} C_5 - \frac{80}{9} C_6, \\ C_8^{eff} &= \frac{4\pi}{\alpha_s} C_8 + C_3 - \frac{1}{6} + 20 C_5 - \frac{10}{3} C_6, \\ C_9^{eff} &= \frac{4\pi}{\alpha_s} C_9 + \mathcal{Y}(q^2), \\ C_{10}^{eff} &= \frac{4\pi}{\alpha_s} C_{10}, \\ C_{7,8,9,10}^{\prime eff} &= \frac{4\pi}{\alpha_s} C_{7,8,9,10}^{\prime}, \end{aligned} \tag{14}$$

$$\begin{aligned} \text{where } \mathcal{Y}(q^2) = {} & h(q^2, m_c) \left(\frac{4}{3} C_1 + C_2 + 6 C_3 + 60 C_5 \right) \\ & - \frac{1}{2} h(q^2, m_b) \left(7 C_3 + \frac{4}{3} C_4 + 76 C_5 + \frac{64}{3} C_6 \right)_{env} \\ & - \frac{1}{2} h(q^2, 0) \left(C_3 + \frac{4}{3} C_4 + 16 C_5 + \frac{64}{3} C_6 \right) \\ & + \frac{4}{3} C_3 + \frac{64}{9} C_5 + \frac{64}{27} C_6. \end{aligned} \tag{15}$$

The function $h(q^2, m_q)$ comes from the fermion loop and for completness is presented in equation 16 below. If you need more details consider reading [5].

$$\begin{aligned} h(q^2, m_q) &= -\frac{4}{9} \left(\ln \frac{m_q^2}{\mu^2} - \frac{2}{3} - z \right) - \frac{4}{9} (2 + z) \sqrt{|z - 1|} \cdot \begin{cases} \arctan \frac{1}{\sqrt{z-1}} & z > 1 \\ \ln \frac{1 + \sqrt{1-z}}{\sqrt{z}} - \frac{i\pi}{2} & z \leq 1 \end{cases} \\ z &= \frac{4 m_q^2}{q^2} \end{aligned} \tag{16}$$

5. Classifiers test

In particle physics the first step of each analysis is to clean up the data. That means to subtract the background from the signal. Signal means for example a type of decay like the $B \to K^* \mu\mu$ decay. To find this decay in the raw data of a detector which contains all kind of decays one needs thresholds on experimental parameters which are different or in the best case unique to the decay. That can be all kind of parameters like vertex locations, momentas or angles between trajectories. For the $B \to K^* \mu\mu$ decay these parameters would be:

- Decay vertex location for reconstructed particles (ENDVERTEX)
- Primary vertex location (OWNPV)
- Impact parameter (IP_OWNPV)
- Flight distance (FD_OWNPV)
- The cosine of the angle between primary vertex and decay vertex and recorded momentum (DIRA_OWNPV)

The names in the bracets are the labels given to these parameters in the data structure of the ROOT Data Anlysis Framework [1]. Now one needs thresholds on these parameters, which define the signal-region. For example the flight distance has to be between 0.2 milimeters and 2 centimeters. But how do you obtain these thresholds? Thats were Classifiers are usefull. Classifiers are a type of algorithm

from the Machine Learning area. They can be trained to distinguish different types of data. In this case signal and background. Each event in the data gets a probability to be signal assigend to it. So after the classification one can just take all the events with a probability over lets say 80%. One has now eliminated 80% of the background and can continue cleaning up the signal but thats not a part of this thesis. The goal of this part was to find the best classifier to be used in the $B \to K^* \mu \mu$ decay analysis.

Seven different classifiers were tested. A test means that the classifier is given a training set consisting of Monte arlo simulated events containing only $B \to K^* \mu \mu$ decays and randomly choosen real detector data, containing all kinds of decays. In a training set the data is labeled. The Monte Carlo events are labeld with 1 for signal and the real data with 0 for background. The classifier will now train, that means the algorithm tries to label the data into signal and background by choosing different thresholds on the parameters mentioned above. After the training the thresholds get fixed. Now one has to test the classifer to check if everything worked fine. To do so a second set of data is prepared similar to the training set, just that the labels are now hidden from the classifier. Since the classifiers are not perfect, they do not assign 0 and 1 to each event but rather a probability to bi signal between 0 and 1. A good classifier will now label the signal events with a high probabilityand background with a low probabillity. A bad classifier will assign 0.5 probabilityto each event. With a 0.5 probability no information was gained by classification, because having a 50:50 chance is as good as guessing. This probability assignements can now be used as a threshold themselves.
Another property of classifiers is, that the data used in the training, can no longer be used in the analysis, because the classifier knows the training set too well and a bias is introduced if one reuses the training data in the actual analysis. That would mean that the part of the real detector data which is needed to train the classifiers would be lost for further analysis. To avoid such a waste of data a technique called k-folding is used.
K-folding means that the no training set is created. But the data Monte Carlo Mix with signal events and all kinds of other events is split into k equal parts. Now to classify one part all the other parts are used for training. After iterating over all parts, one has classified all the data without loosing any. The only disadvantage are the addidtional computer resources needed, since the classifier has now to be trained k times instead of one time.

The following list of classifers where tested and compared in terms of perfomance:

- Ada Boost [14]
- uGB [15] + knnAda (k-nearest neghbor AdaBoost)
- uBoost [16]
- uGB [15] + Fl (flatness loss)
- xgb [17]
- sk_bdtg [18]
- sk_bdt [19]

The test was performed with 30000 events from the 2016 LHCb $B \to K^* \mu \mu$ data and 10000 events from the Monte Carlo simulation.

To compare classifiers the so called ROC (receiver operating charactertic) curves are used. The ROC curve shows the true positive rate against the false positive rate. The true positive rate is the rate of signal events that have been classified correctly as signal events, while the false postive rate ist the rate of background events classified as signal events. On figure 12 a ROC curves of the classifiers above is displayed. The sample of 40000 events to test the classifiers was cut into ten equal parts, also called

folds. In figure 12 the ROC curve of one of these folds is displayed, the other nine can be found in the appendix A.
It turns out that all the classifiers classify the data mostly correctly with just some minor variances. The ROC curve is in that case not a good tool to compare the different classifiers.

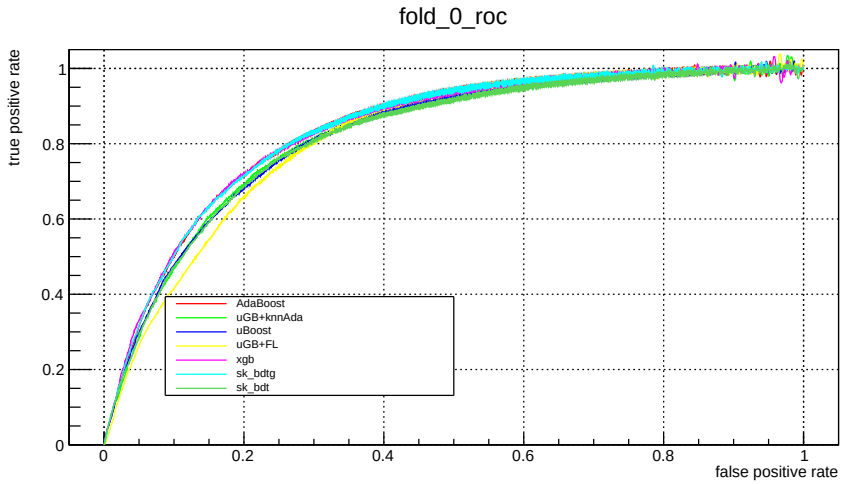


Figure 12: ROC curve of the first fold. One can see that all the classifiers are competetive in terms of classifing correctly

The next step is to check for correlations between the assigned probabilities and the kinematic variables. The kinematic variabels as explained in section 4.2 are used to obtain the angular coefficients of the decay rate. If there is a correlation between the assigned probabilities from the classifier and these variables, a peak is artificialy added into the distribution of these variables. Imagine a positive correlation between the B mass in the range 5000 to 6000 GeV and the probability to bi signal. After cutting at lets say 0.8 probability there will be a peak in the B mass distribution between 5000 and 6000 GeV. Normally such a peak suggest a new particle in this range. This is to avoid at all cost since it will compremise the whole analysis later on.
In the following the correlation between the probability to be signal assigned by the classifier and the B mass are shown. The correlation plots for the other kinematic variables can be found in appendix B.

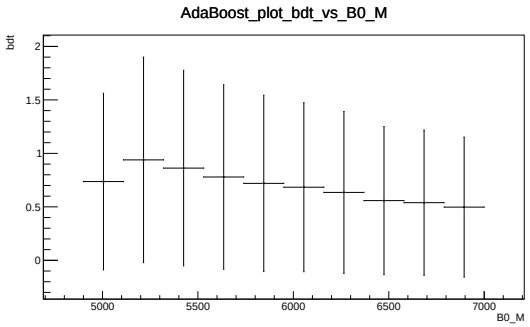


Figure 13: This plot showes the probabilities assign by the classifier as explained in the above paragraph, labeld with bdt on the Y achsis. On the X achsis the values for the B mass are shown. The X - errorbars indicate the length of each bin, while the Y errorbars represent the standart derivation on the probabilities. One can see in this plot the correlation between the classifiers probabilities and the B mass for the AdaBoost classifier. There is a obvius correlation from the second bin to the fourth bin.

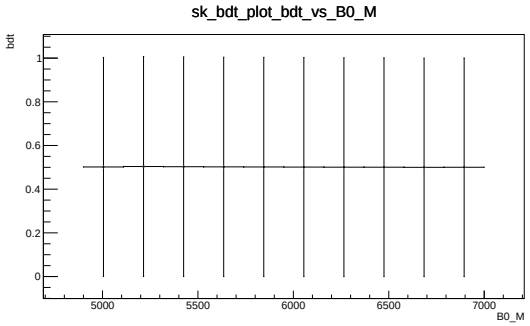


Figure 14: This plot shows the correlations between the probabilities to be signal assigned by the sk_bdt classifier and the B mass. There is no correlation.

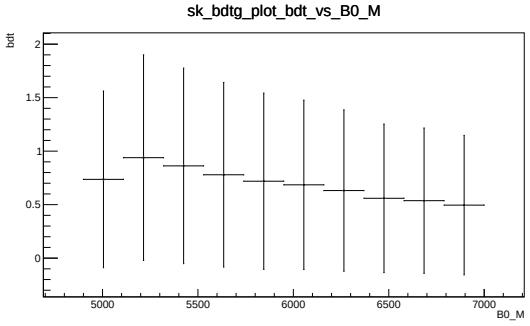


Figure 15: This plot shows the correlations between the probabilities to be signal assigned by the sk_bdtg classifier and the B mass. There is a obvius correlation from the second bin to the fourth bin.

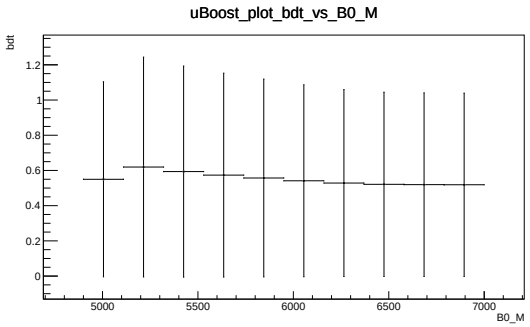


Figure 16: This plot shows the correlations between the probabilities to be signal assigned by the uBoost classifier and the B mass. There is just a very small correlation compared to other classifiers.

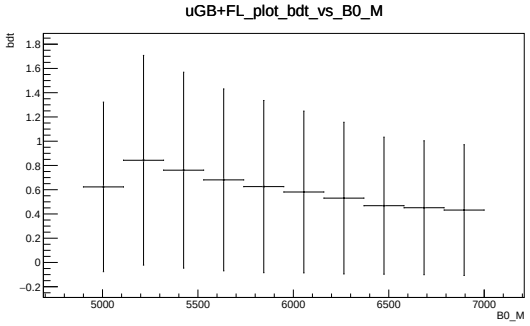


Figure 17: This plot shows the correlations between the probabilities to be signal assigned by the uGB+FL classifier and the B mass. There is a obvius correlation from the second bin to the fourth bin.

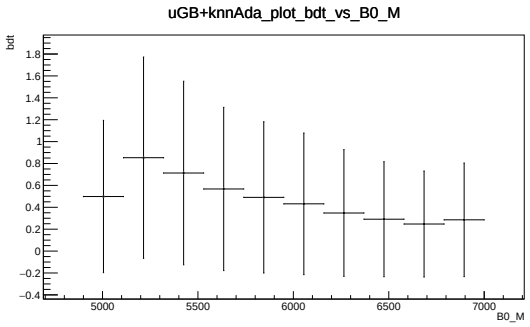


Figure 18: This plot shows the correlations between the probabilities to be signal assigned by the ugb+knnAda classifier and the B mass. There is a obvius correlation from the second bin to the fourth bin.

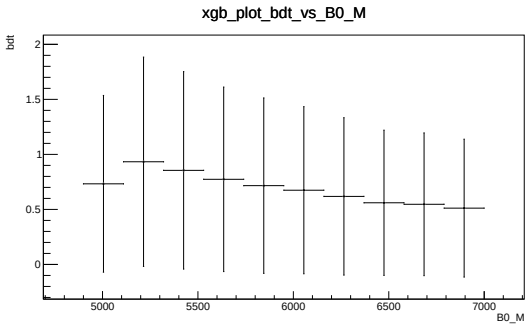


Figure 19: This plot shows the correlations between the probabilities to be signal assigned by the xgb classifier and the B mass. There is a obvius correlation from the second bin to the fourth bin.

The two classifers with the least correlation are the sk_bdt and the uBoost classifers. Therfore they could be used to do seperate between signal and background in the $B \to K^* \mu \mu$ data.

6. Reweighting

Reweighting is method to matchh the Monte Carlo data to the real detector data, to extract quantities not measurable by the detector like efficiencies. The match is done by applying weights to the Monte Carlo events.
As an initial weight the sWeights see equation 36 are used. The weights are calculated by comparing the the following parameters: 'nTracks', 'B_0 p_T' and the quality of the $K\pi\mu\mu$ vertex. The new weights derived by the difference in data and simulation of these parameters are used to weight all the MC samples. For this analysis the MC and Data of the $K^* \to J/\Psi K^{*0}$ are used, because it is a very clean channel.

6.1. Reweighting the $B \to K^*\ \mu\ \mu$ Monte Carlo data

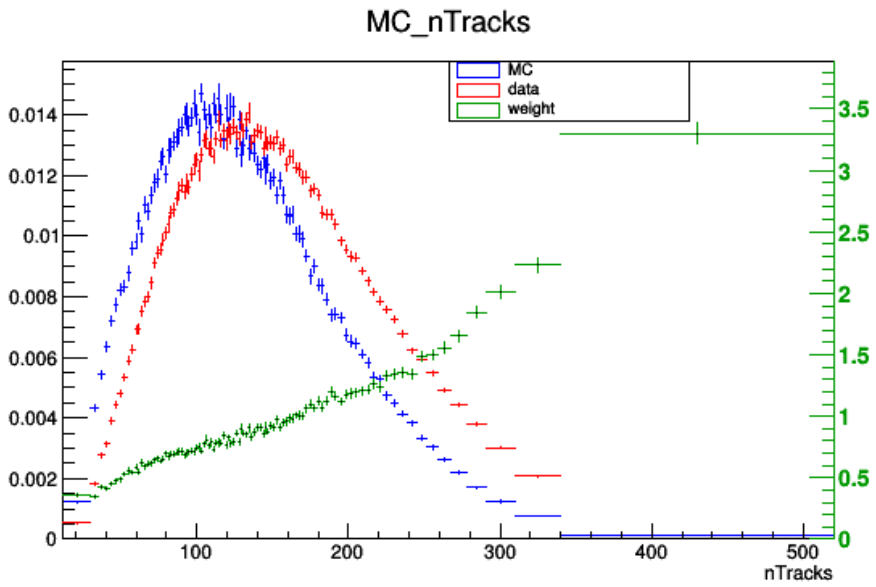


Figure 20: This plot shows the Monte Carlo Simulated data in blue, the detector data in red and in green the weight best to reweight the Monte Carlo data to match the detector data. Thx X achsis show the values for the number of tracks in an event (nTracks). On the Left achsis the fraction of events having so many nTracks is given, for data and Monte Carlo. On the right achsis the magnitude of the weights is shown.

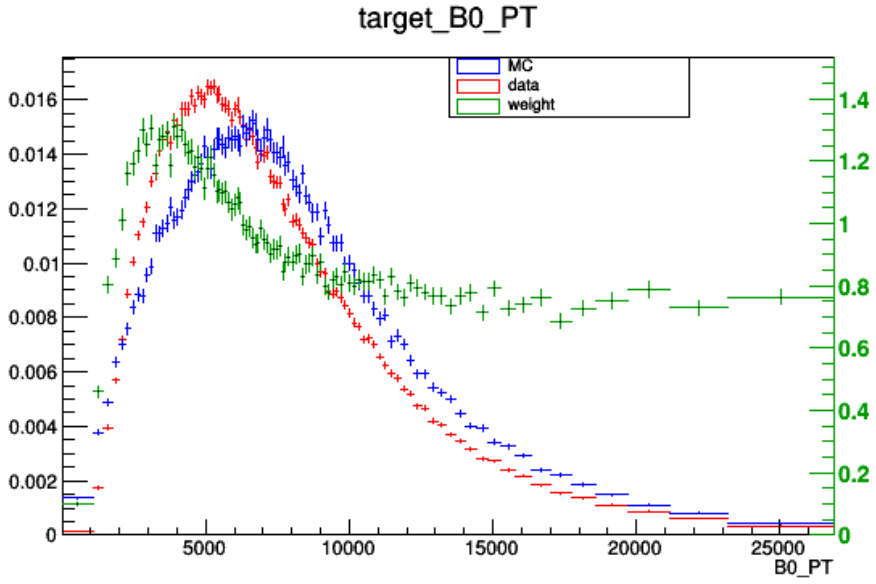


Figure 21: Same plot as in figure 20, for the transversal momentum of the $B-0$ meson (B0_PT)

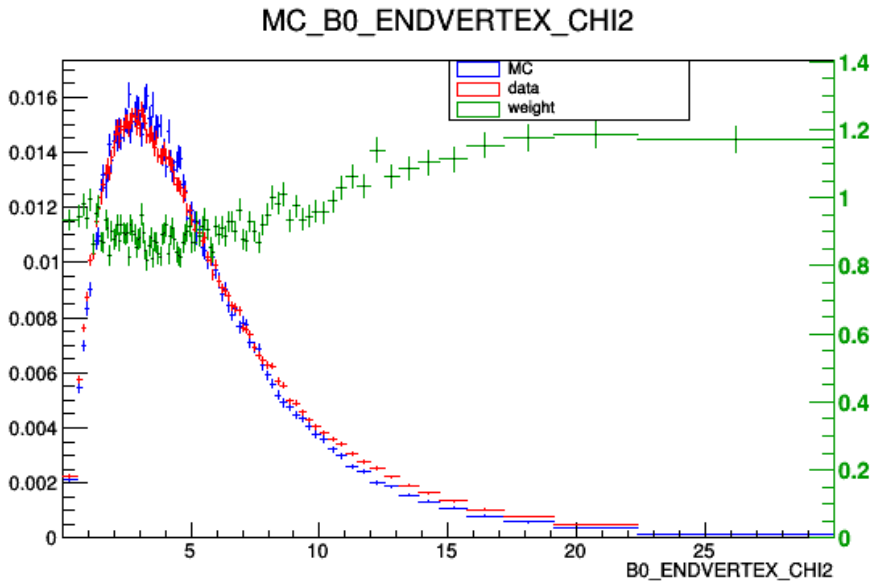


Figure 22: Same plot as in figure 20, for the quality of the $K\pi\mu\mu$ vertex (B0_ENDVERTEX_CHI2).

After applying those weights to the Monte Carlo data, the distributions of all the event parameters will change. To check if the simulation really matches the data. Some comparison plots have been made. One is presented here, while many others can be found in appendix C.

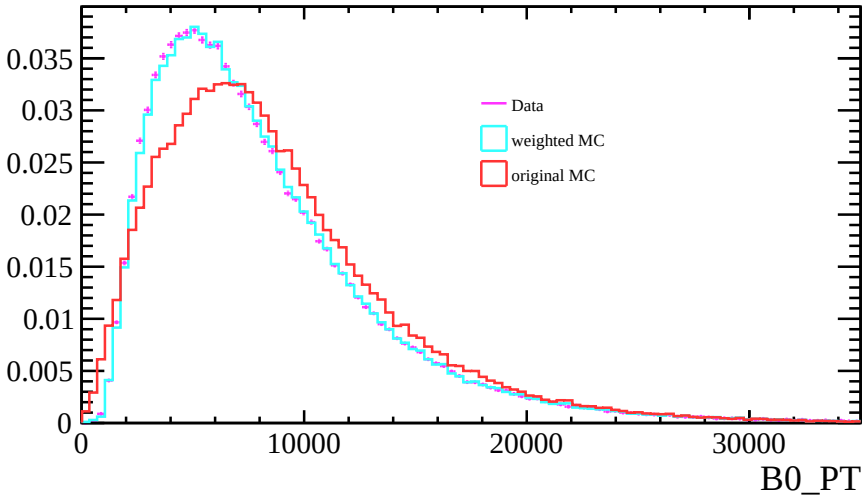


Figure 23: This plots shows the distribution of the B_0 transversal momentum. The violet points represent the distribution in the detector data. In red is the distribution befor reweighting the Monte Carlo data and in blue is the distribution after reweighting the Monte Carlo data. The X achsis shows the B_0 transversal momentum (B0_PT)

6.2. SPlot

In this section the SPlot Technique to sperate two or more merged distributions. It has been used to get the inital weights for the Reweighting in chapter 6.

6.2.1. Likelihood method

Consider an analysis of a data sample, which consists of several types of events. These types represent signal components and background components, for example from different experiments. The log-liklihood of such a data sample is expressed as:

$$L = \sum_{i=1}^{N} \ln \left[\sum_{j=1}^{N_S} N_j f_j(y_i) \right] - \sum_{i=j}^{N_S} N_j \tag{17}$$

- N = total number of events
- N_S = number of types
- N_i = expected average number of events for type i
- y = set of diciminating variables
- f_j = PDF of the ith type
- $f_j(y_i)$ = value of PDF for event y_i
- x = control variable, not a part of L by construction

The yields N_i and the free parameters of the PDF are obtained by maximizing the above log-likelihood (eq 17).

6.2.2. ${}_{in}Plot$ technique

Consider a varibale x which can be expressed as a function of the dicriminating variables y used in the fit. Furthermore a fit has been performed to determine the yields N_i for all types. From the knowledge of the PDF and the values of N_i a naive weight can be defined as:

$$P_n(y_i) = \frac{N_n f_n(y_i)}{\sum_{k=1}^{N_s} N_k f_k(y_i)} \tag{18}$$

which will leed to the x-distribution $\tilde{M}_n$ defined by:

$$N_n \tilde{M}_n(\bar{x}) = \sum_{i \subset \delta x} P_n(y_i) \tag{19}$$

where sum $\sum_{i \subset \delta x}$ contains alle events for which x_i lies in the interval centered on $\bar{x}$ and of total width δx. Therefor $N_n \tilde{M}_n(\bar{x}) \delta x$ is the x-distribution of the histogrammed events, using the weights of eq 18.

With this procedure one can on average reproduce the true distribution $\mathbf{M}_n(x)$. One can even replace the sum in eq 19 by an integral:

$$\left\langle \sum_{i \subset \delta x} \right\rangle \to \int dy \sum_{j=1}^{N_s} N_j f_j(y) \delta(x(y) - \bar{x}) \delta x \tag{20}$$

Furthermore through identifying the number of events N_i from the fit one gets:

$$\begin{aligned}
\langle N_n \rangle \tilde{M}_n(\bar{x}) &= \int dy \sum_{j=1}^{N_s} N_j f_j(y) \delta(x(y) - \bar{x}) P_n(y) && (21)\\
&= \int dy \sum_{j=1}^{N_s} N_j f_j(y) \delta(x(y) - \bar{x}) \cdot \frac{N_n f_n(y)}{\sum_{k=1}^{N_s} N_k f_k(y)} && (22)\\
&= N_n \int dy \delta(x(y) - \bar{x}) f_n(y) && (23)\\
&= N_n \mathbf{M}_n(\bar{x}) && (24)
\end{aligned}$$

One can see that the sum over enents of the daive weight P_n provides a direct estimate of the x-distribution for the nth type. But this procedure has a major drawback, since x is correlated to y, the PDFs of x enter implicity in the definition of the naive weight. Therfor the $\tilde{M}_n$ distributions are a bad estimate for the quality of the fit, since the these distributions are biased in a difficult way, when the PDFs $f_i(y)$ are not accurate.
Consider for example a data sample where one of the types has events on the tail of the x-distribution. Such events require the true ditribution to account for the tail. But since the events are averaged the weights on the tail are going to be very small missing those events in the extimated true distribution. Only the core of the x-distribution can be examined with ${}_{in}Plots$.

6.2.3. ${}_{S}Plot$ technique

In the previous section it was shown that if a variable x belongs to a set y of discriminating varibales, one can reconstruct the expected x distribution. Consider now two sets of variables x and y, where x does not belong to y and which are uncorrelated , hence the total PDFs $f_i(x, y)$ all factorize into products $\mathbf{M}_i(x) f_i(y)$. The equation 24 does not hold anymore because, when summing over the events the x-PDFs $\mathbf{M}_j(x)$ appear:

$$\begin{aligned}
\langle N_n \rangle \tilde{M}_n(\bar{x}) &= \int \int dy dx \sum_{j=1}^{N_s} N_j \mathbf{M}_j(x) f_j(y) \delta(x - \bar{x}) P_n && (25)\\
&= \int dy \sum_{j=1}^{N_s} N_j \mathbf{M}_j(\bar{x}) f_j(y) \frac{N_n f_n(y)}{\sum_{k=1}^{N_s} N_k f_k(y)} && (26)\\
&= N_n \sum_{j=1}^{N_s} \mathbf{M}_j(\bar{x}) \left(N_j \int dy \frac{f_n(y) f_j(y)}{\sum_{k=1}^{Ns} N_k f_k(y)} \right) && (27)\\
&\neq N_n \mathbf{M}_n(\bar{x}). && (28)
\end{aligned}$$

The correction term

$$N_j \int dy \frac{f_n(y) f_j(y)}{\sum_{k=1}^{N_s} N_k f_k(y)} \tag{29}$$

is not identical to the kroenecker delta δ_{jn}. In fact the $N_n \tilde{M}_n$ distribution obtained by the naive weight is a linear combination of the true distribution $\mathbf{M}_j$.
To go forward one has to realize that the correction term is related to the inverse of the covariance matrix, given by the second derivatives of $-L$, after the minimization.

$$\mathbf{V}_{nj}^{-1} = \frac{\partial^2(-L)}{\partial N_n \partial N_j} = \sum_{i=1}^{N} \frac{f_n(y_i) f_j(y_i)}{\left(\sum_{k=1}^{N_s} N_k f_k(y_i)\right)^2} \tag{30}$$

If one averages and is replacing the sum over events by intergals (eq 20) the varaince matrix reads:

$$\langle \mathbf{V}_{nj}^{-1} \rangle = \int\int dy dx \sum_{e=1}^{N_s} N_e \mathbf{M}_e(x) f_e(y) \frac{f_n(y) f_j(y)}{\left(\sum_{k=1}^{N_s} N_k f_k(y)\right)^2} \tag{31}$$

$$= \int dy \sum_{e=1}^{N_s} N_e f_e(y) \frac{f_n(y) f_j(y)}{\left(\sum_{k=1}^{N_s} N_k f_k(y)\right)^2} \cdot \int dx \mathbf{M}_l(x) \tag{32}$$

$$= \int dy \frac{f_n(y) f_j(y)}{\sum_{k=1}^{N_s} N_k f_k(y)} \tag{33}$$

Therefor equation 25 can be rewritten as:

$$\langle \tilde{M}_n(\bar{x}) \rangle = \sum_{j=1}^{N_s} \mathbf{M}_j(\bar{x}) N_j \langle \mathbf{V}_{nj}^{-1} \rangle. \tag{34}$$

To get the distribution of intrest one has to invert this matrix equation:

$$N_n \mathbf{M}_n(\bar{x}) = \sum_{j=1}^{N_s} \langle \mathbf{V}_n j \rangle \langle \tilde{M}_j(\bar{x}) \rangle \tag{35}$$

The true distribution of x can still be reconstructed using the naive weight (eq 18), through a linear combination of ${}_{in}Plots$. In other words: When x does not belong to the set y, the weights are not given by equation 18, they are given by a covariance-weighted quantity called sWeight defined by:

$${}_sP_n(y_i) = \frac{\sum_{j=1}^{N_s} \mathbf{V}_n j f_j(y_i)}{\sum_{k=1}^{N_s} N_k f_k(y_i)} \tag{36}$$

With the sWeights on can obtain the distribution of the x variable by histogramming the ${}_sPlot$:

$$N_{ns} \tilde{M}_n(\bar{x}) \delta x = \sum_{i \subset \delta x} {}_sP_n(y_i) \tag{37}$$

On average it reproduced the true distribution:

$$\langle N_{ns} \tilde{M}_n(x) \rangle = N_n \mathbf{M}_n(x) \tag{38}$$

In the case were x is significantly correlated with y, the ${}_sPlots$ from equation 37 connt be compared with the pure distributions of the various types. To solve that problem one can perform a Monte Carlo simulation of the procedure and obtain the expected distributions to which the ${}_sPlots$ should be compared with.
For more information on ${}_sPlots$ consider reading [20].

References

[1] ROOT Data Analysis Framework *https://root.cern.ch/*

[2] MINUIT Home page, *https://seal.web.cern.ch/seal/snapshot/work-packages/mathlibs/minuit/*

[3] JHEP08 (2017) 055

[4] C. Bobeth, M. Misiak and J. Urban, Nucl. Phys. B 574 (2000) 291 [arXiv:hep-ph/9910220].

[5] arXiv:0811.1214 [hep-ph]

[6] CERN Accelerator Complex, *http://www.stfc.ac.uk/research/particle-physics-and-particle-astrophysics/large-hadron-collider/cern-accelerator-complex/*

[7] Science and Technology Facilities Council article about LHCb , *https://www.ppd.stfc.ac.uk/Pages/LHCb.aspx*

[8] VELO description, *http://lhcb-public.web.cern.ch/lhcb-public/en/Detector/VELO2-en.html*

[9] RICH description, *http://lhcb-public.web.cern.ch/lhcb-public/en/Detector/RICH2-en.html*

[10] Magnet description, *http://lhcb-public.web.cern.ch/lhcb-public/en/Detector/Magnet2-en.html*

[11] Tracker description, *http://lhcb-public.web.cern.ch/lhcb-public/en/Detector/Trackers2-en.html*

[12] Calorimeters description, *http://lhcb-public.web.cern.ch/lhcb-public/en/Detector/Calorimeters2-en.html*

[13] Muon system description, *http://lhcb-public.web.cern.ch/lhcb-public/en/Detector/Muon2-en.html*

[14] A Short Introduction to Boosting, by Yoav Freund and Rovert E. Schapire *http://www.site.uottawa.ca/~stan/csi5387/boost-tut-ppr.pdf*

[15] J.H. Friedmann, " Greedy function approximation: A gradient boosting machine " , 2001

[16] J. Stevens and M. Williams, uBoost: A boosting method for producing uniform selection efficiencies from multivariate classifiers, JINST 8, P12013 (2013). [arXiv:1305.7248]

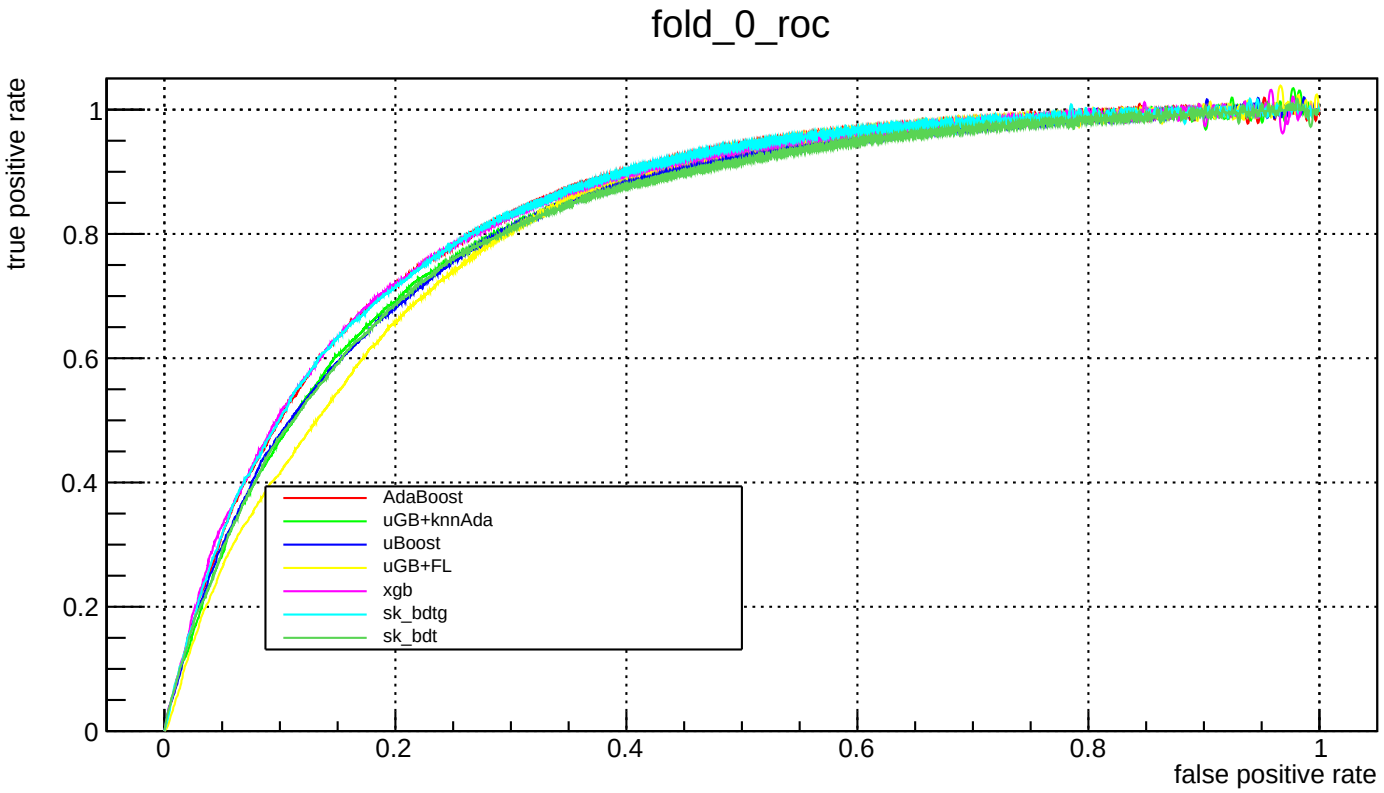
[17] arXiv:1603.02754 [cs.LG]

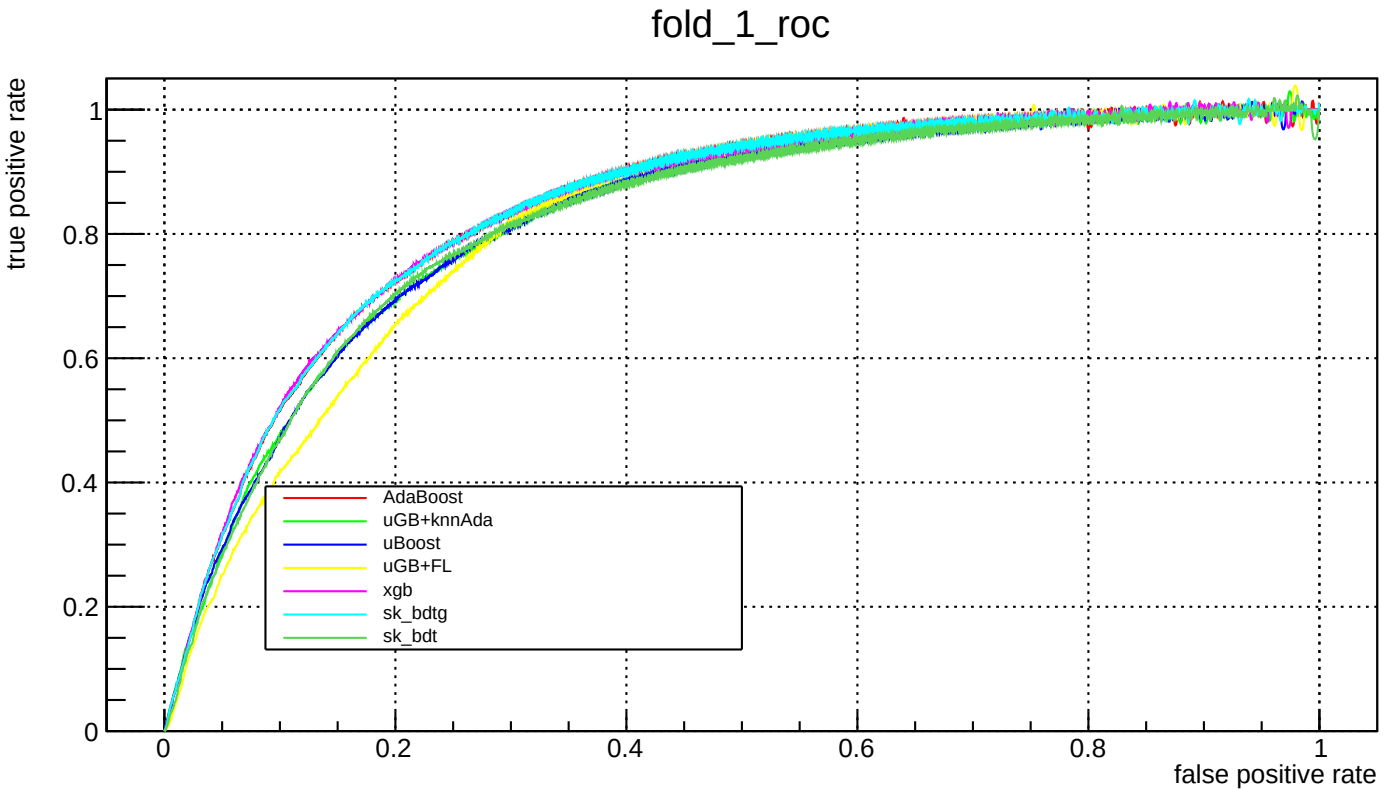
[18] Gradient Boosting classifier from the sk_learn python package *http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html*

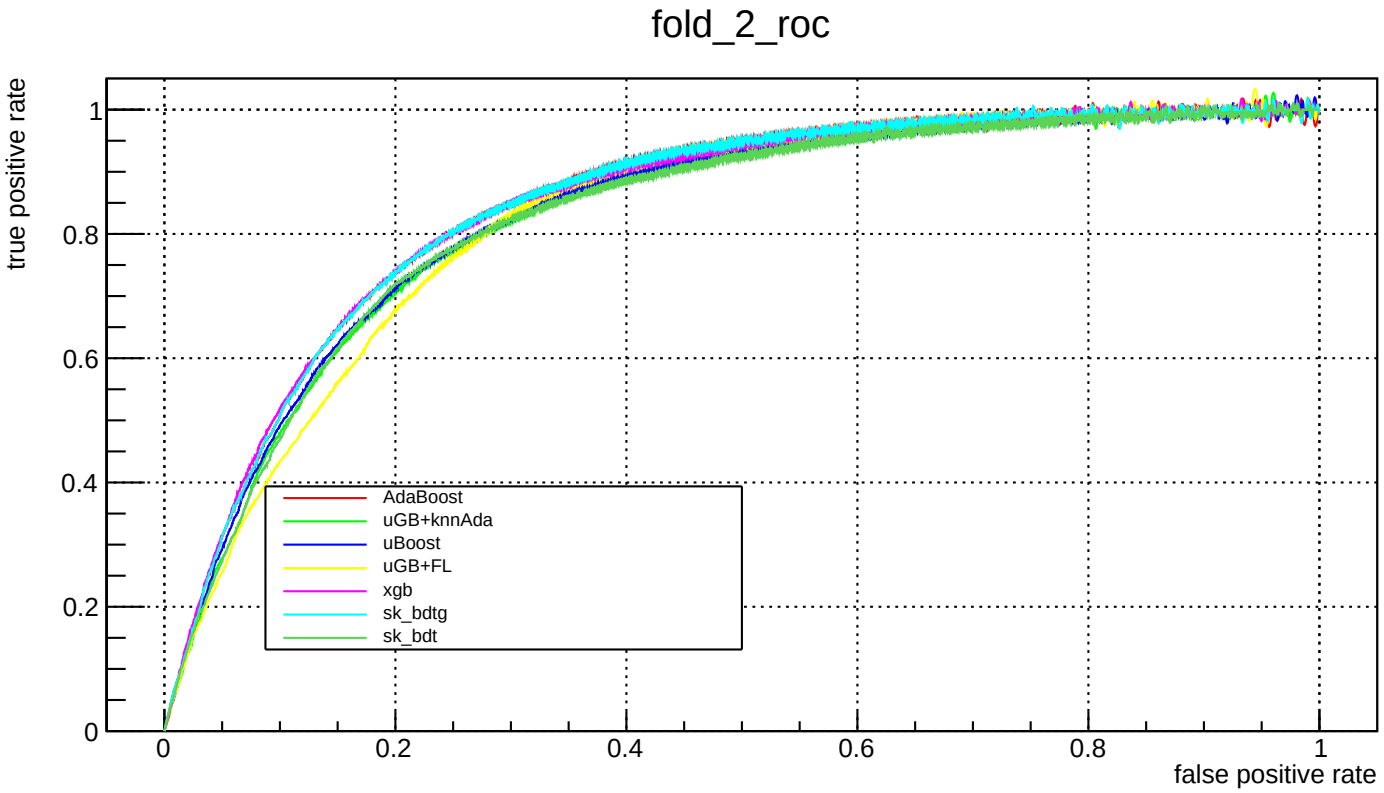
[19] A vaiation of the AdaBoost classifier called AdaBoost-SAMME *http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html*

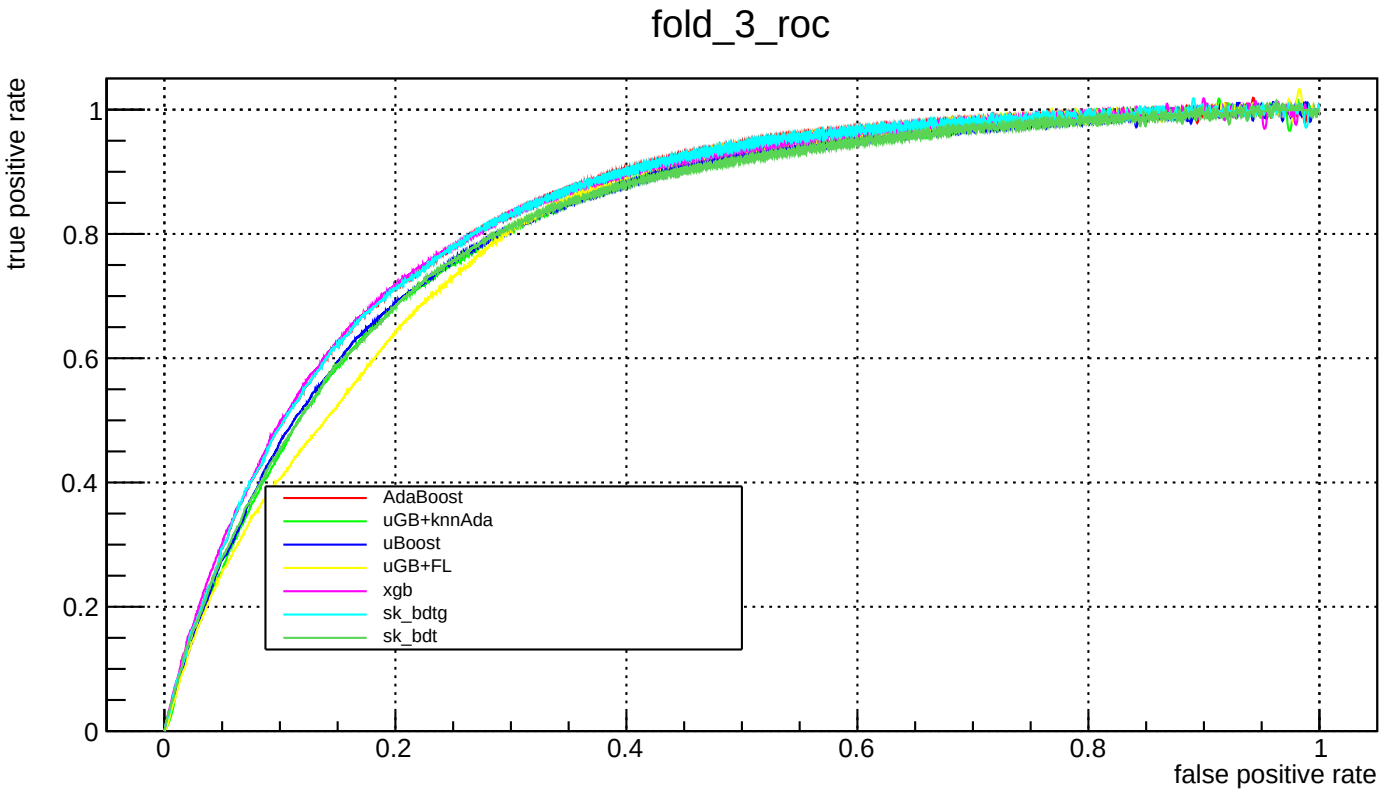
[20] sPlot: a statistical tool to unfold data distributions, arXiv:physics/0402083 [physics.data-an]

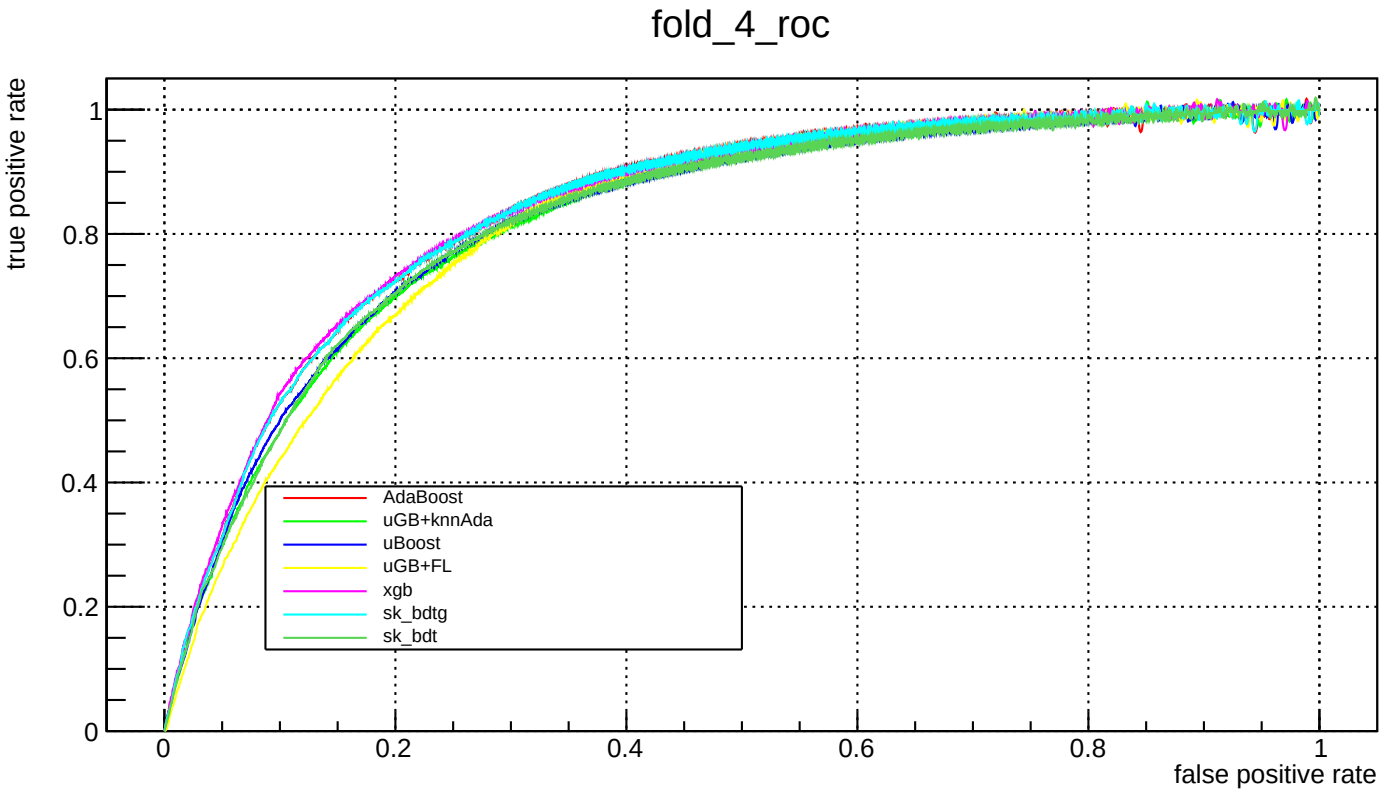
A. ROC curves

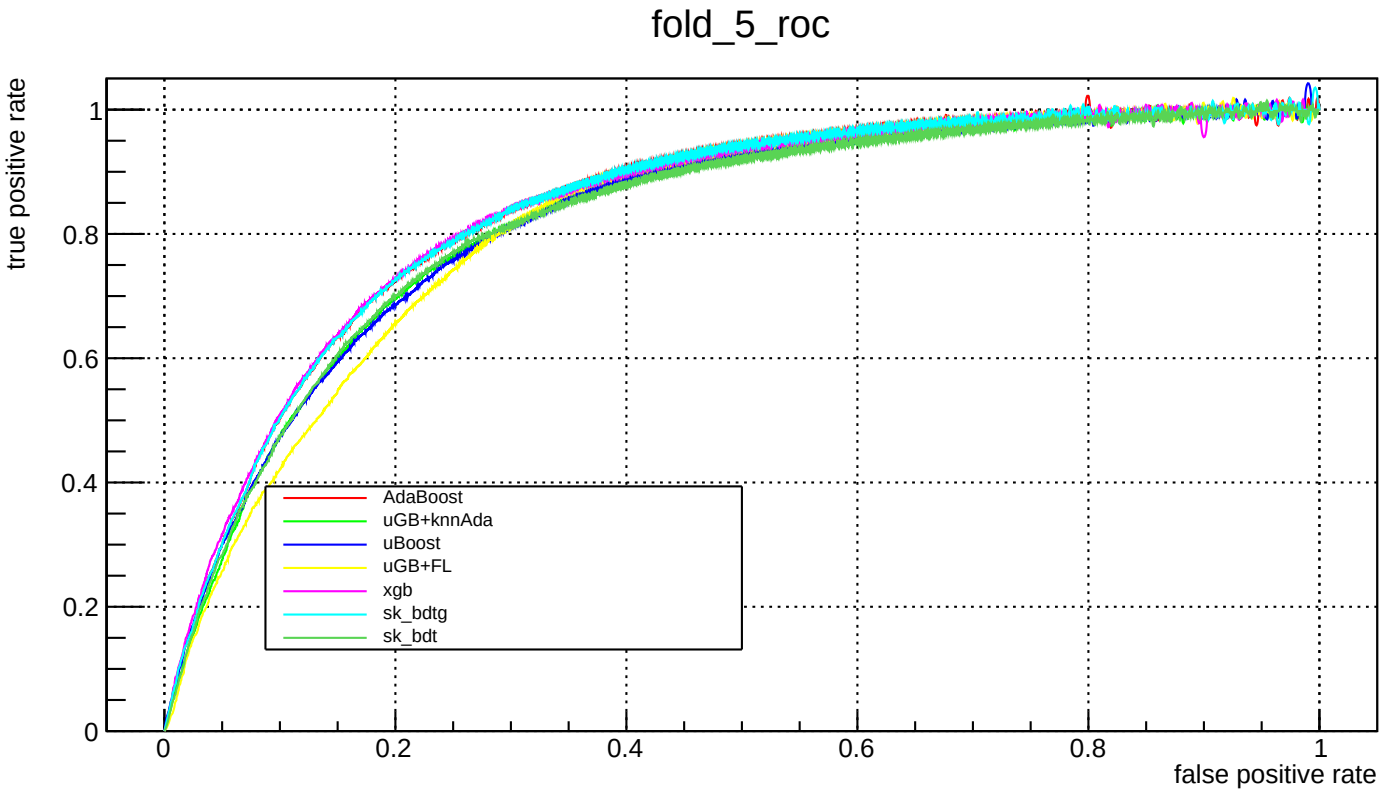

fold_0_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

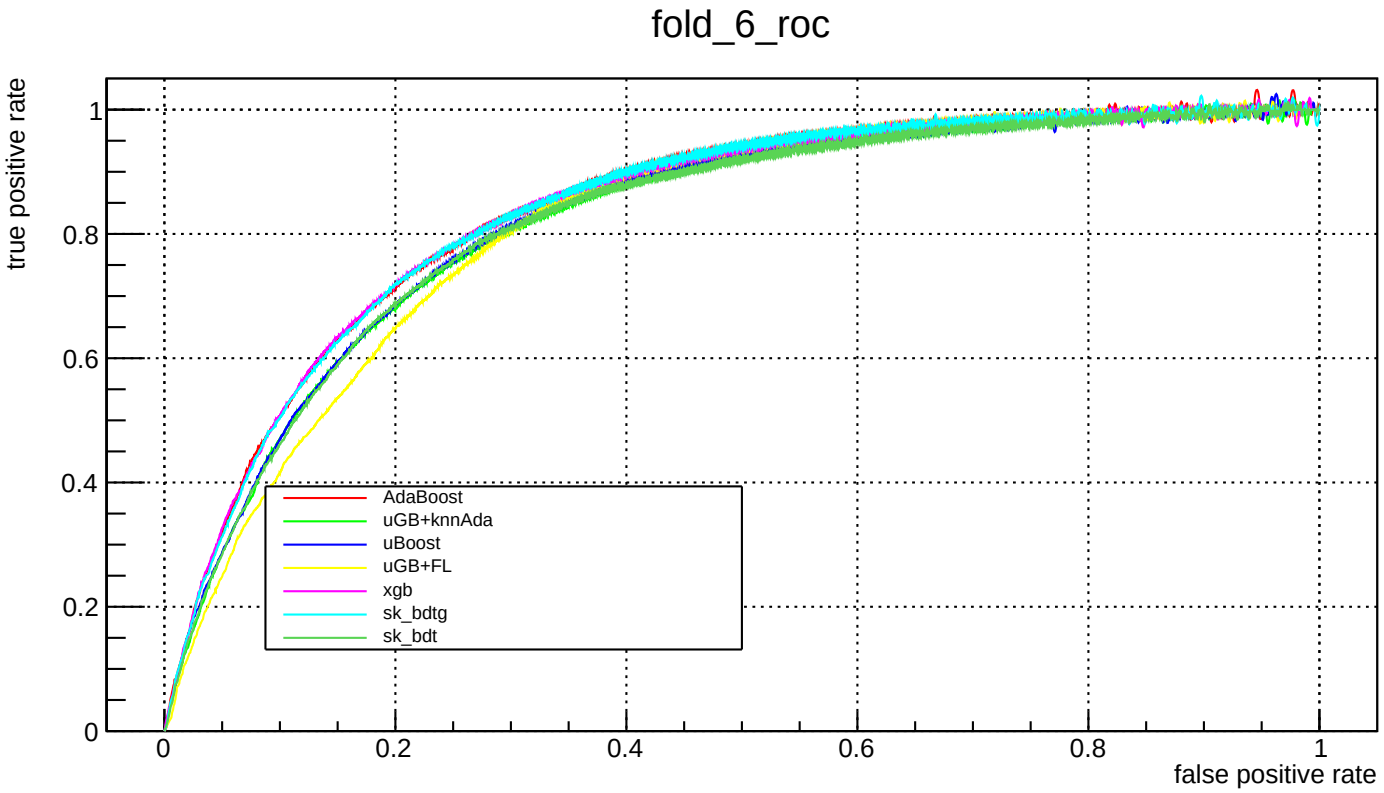

fold_1_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

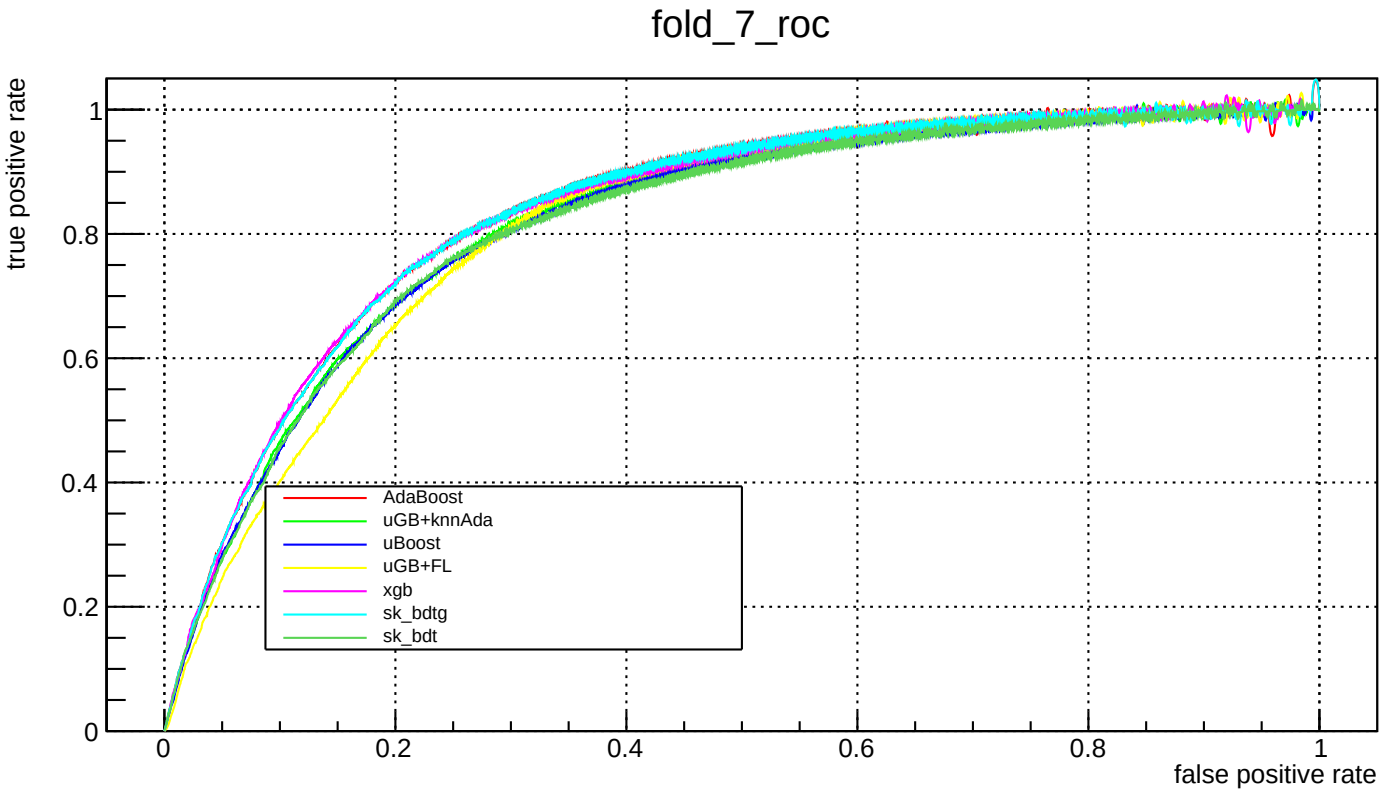

fold_2_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

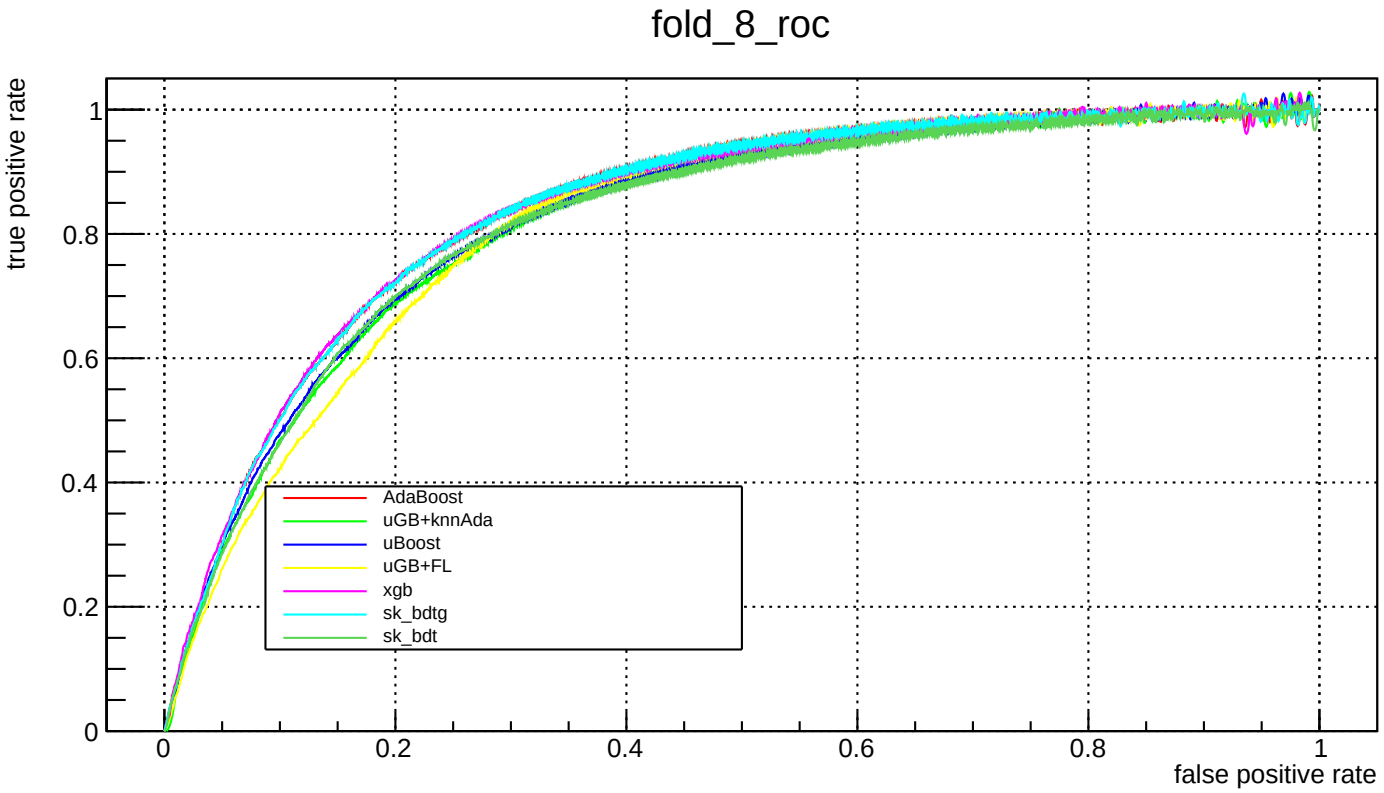

fold_3_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

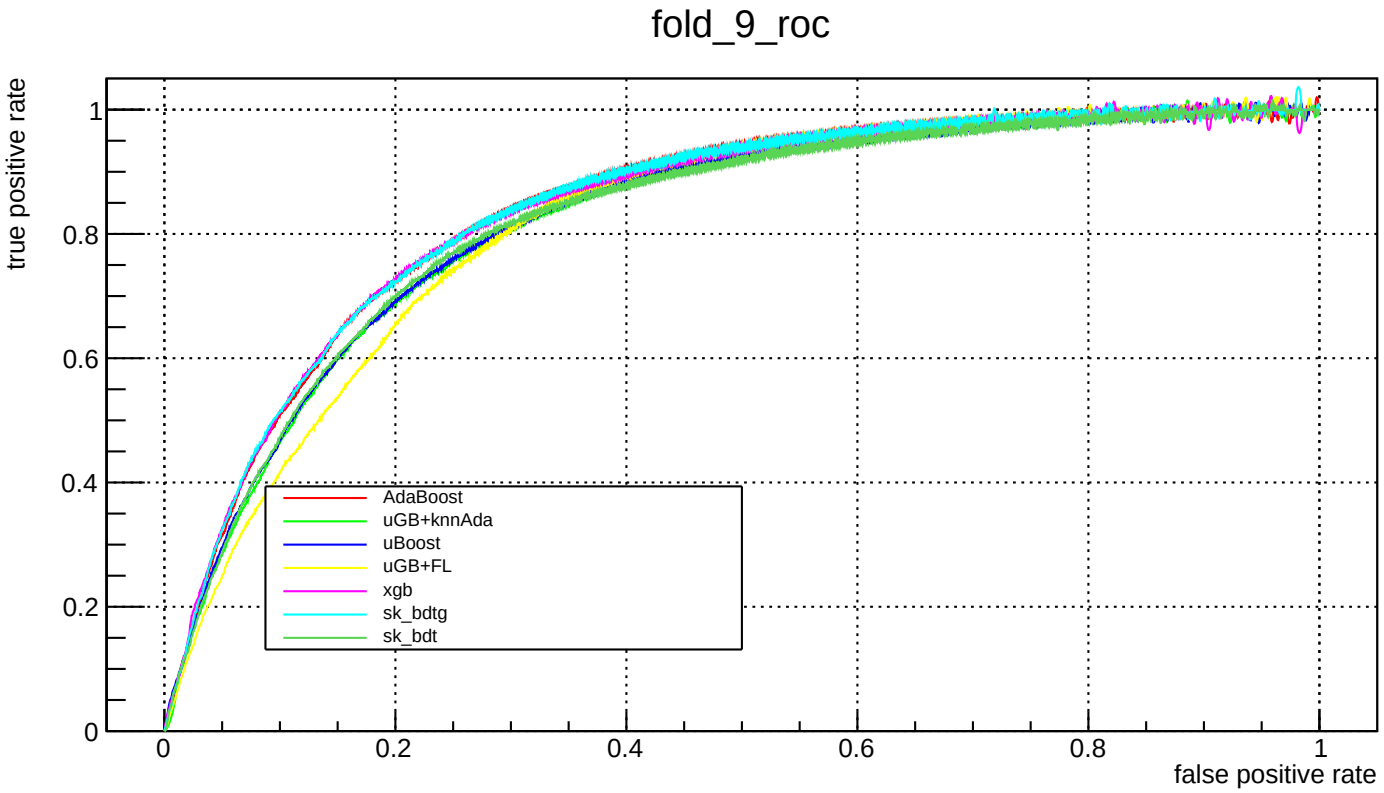

fold_4_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt


fold_5_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

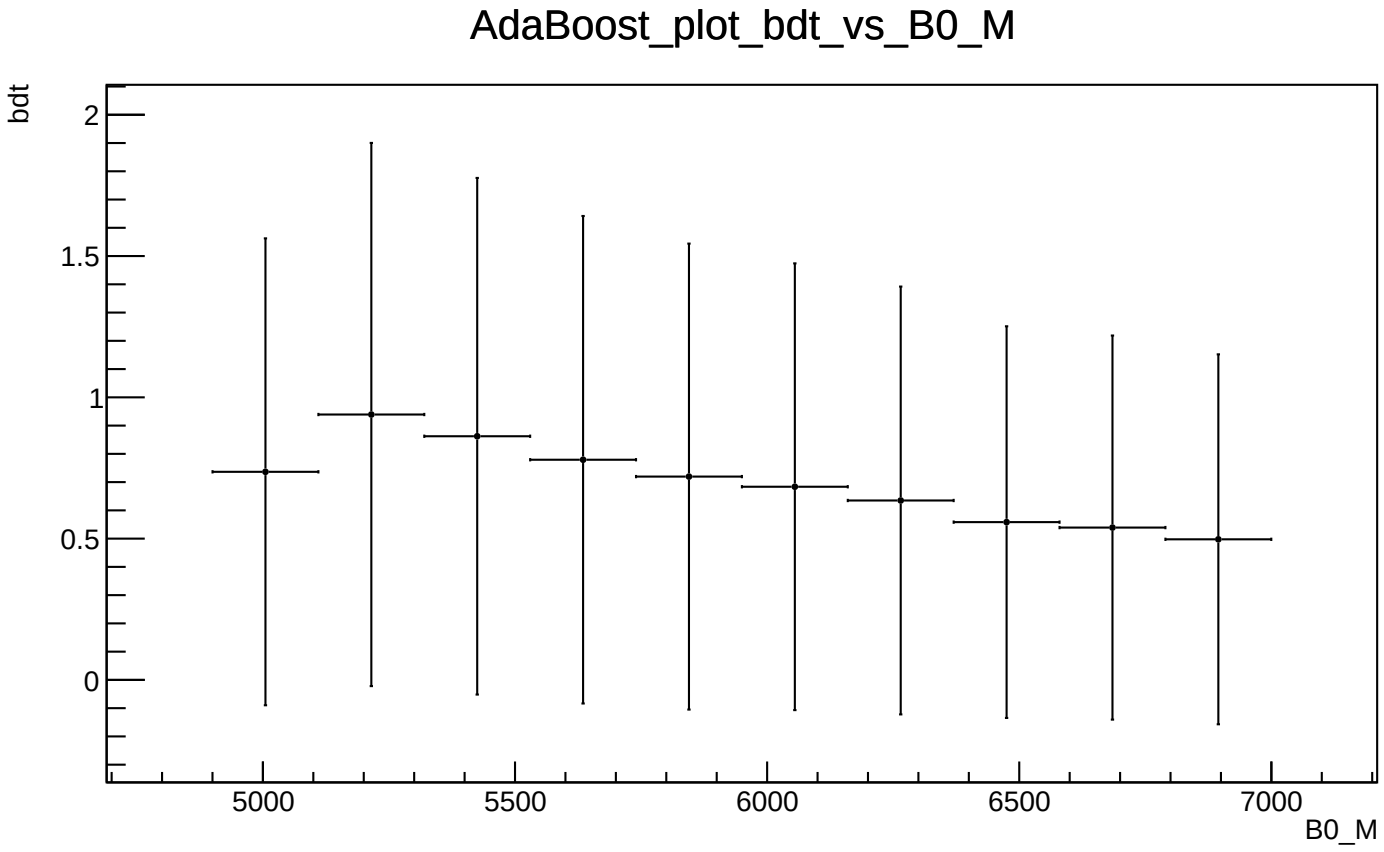

fold_6_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

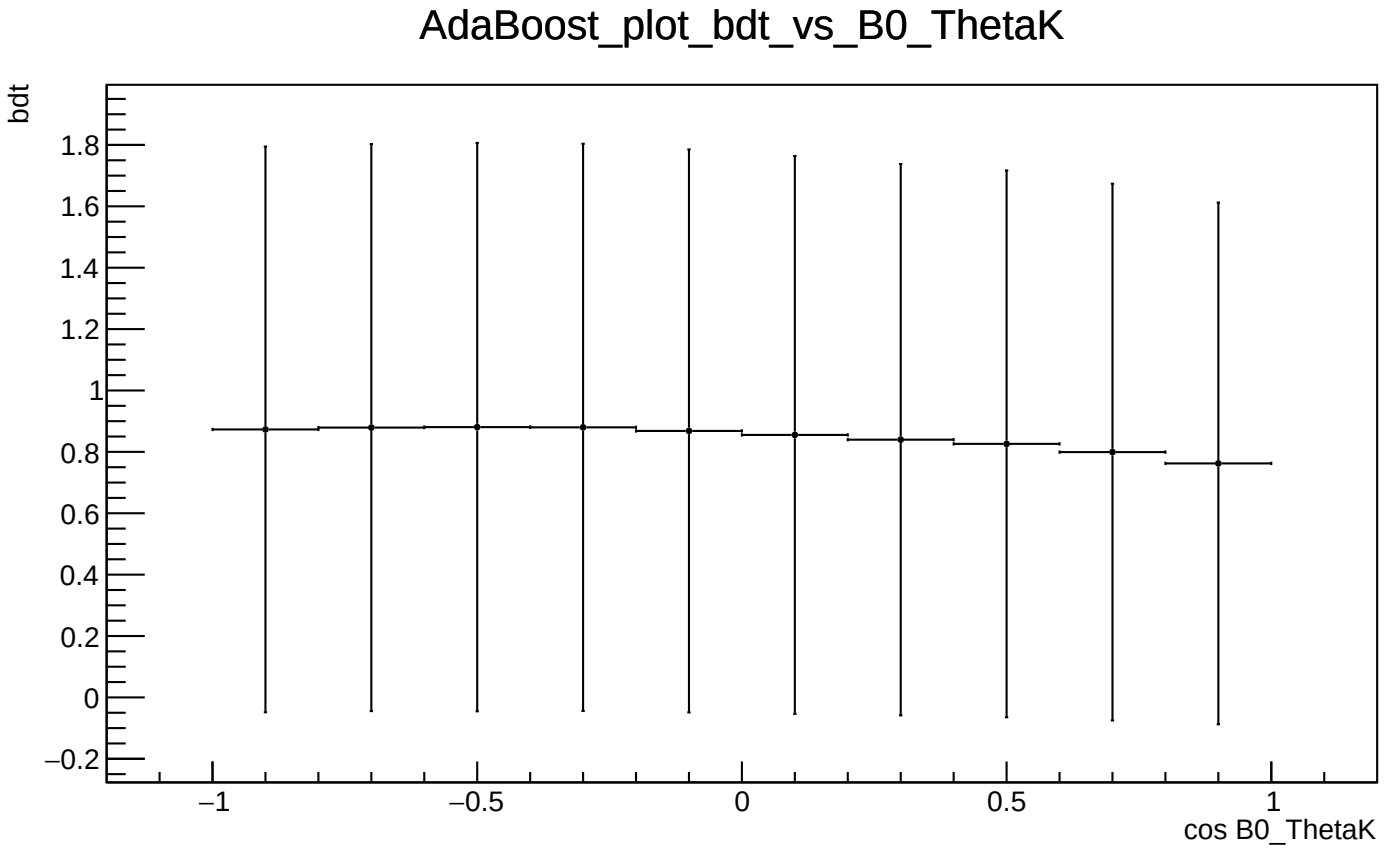

fold_7_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

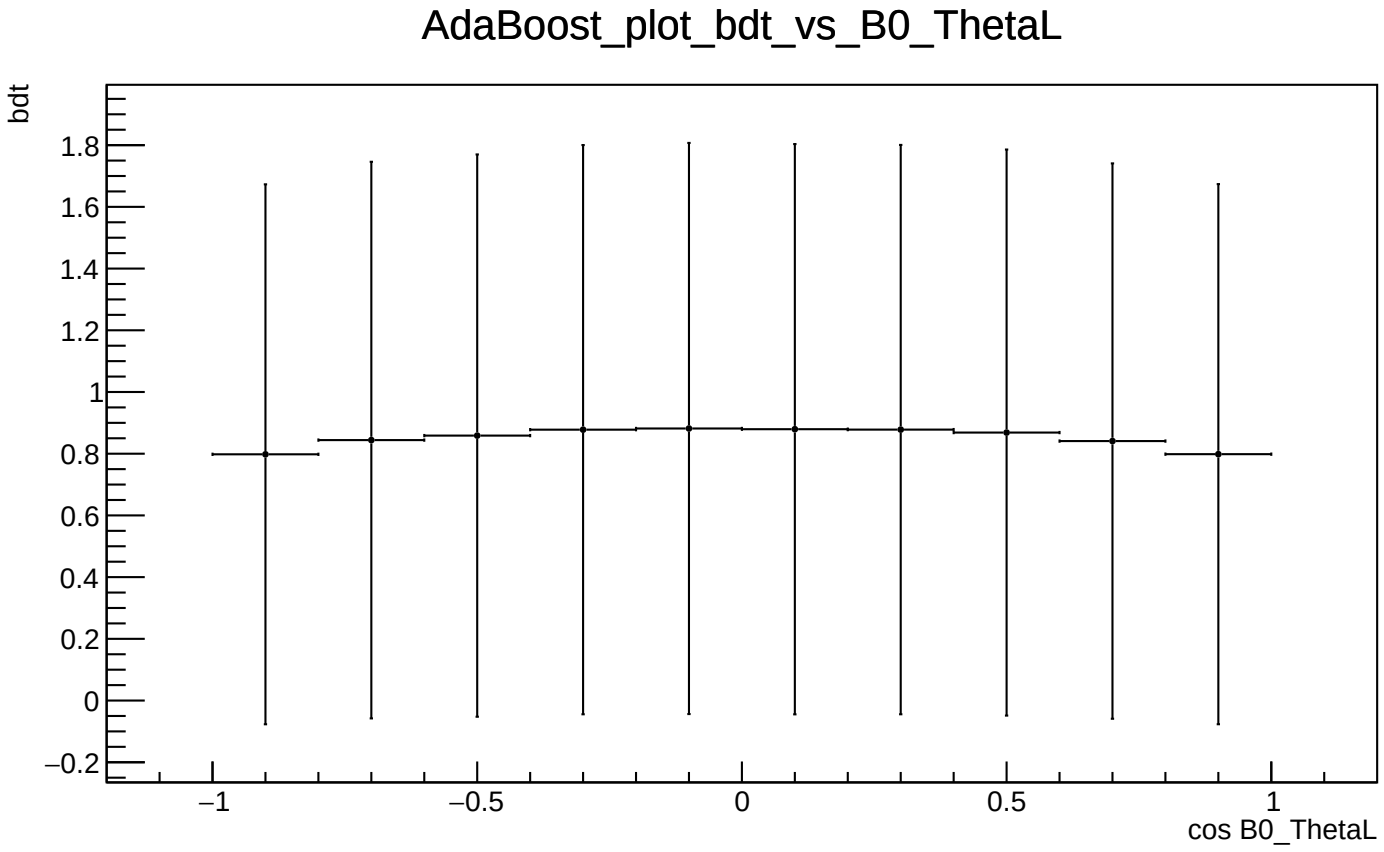

fold_8_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

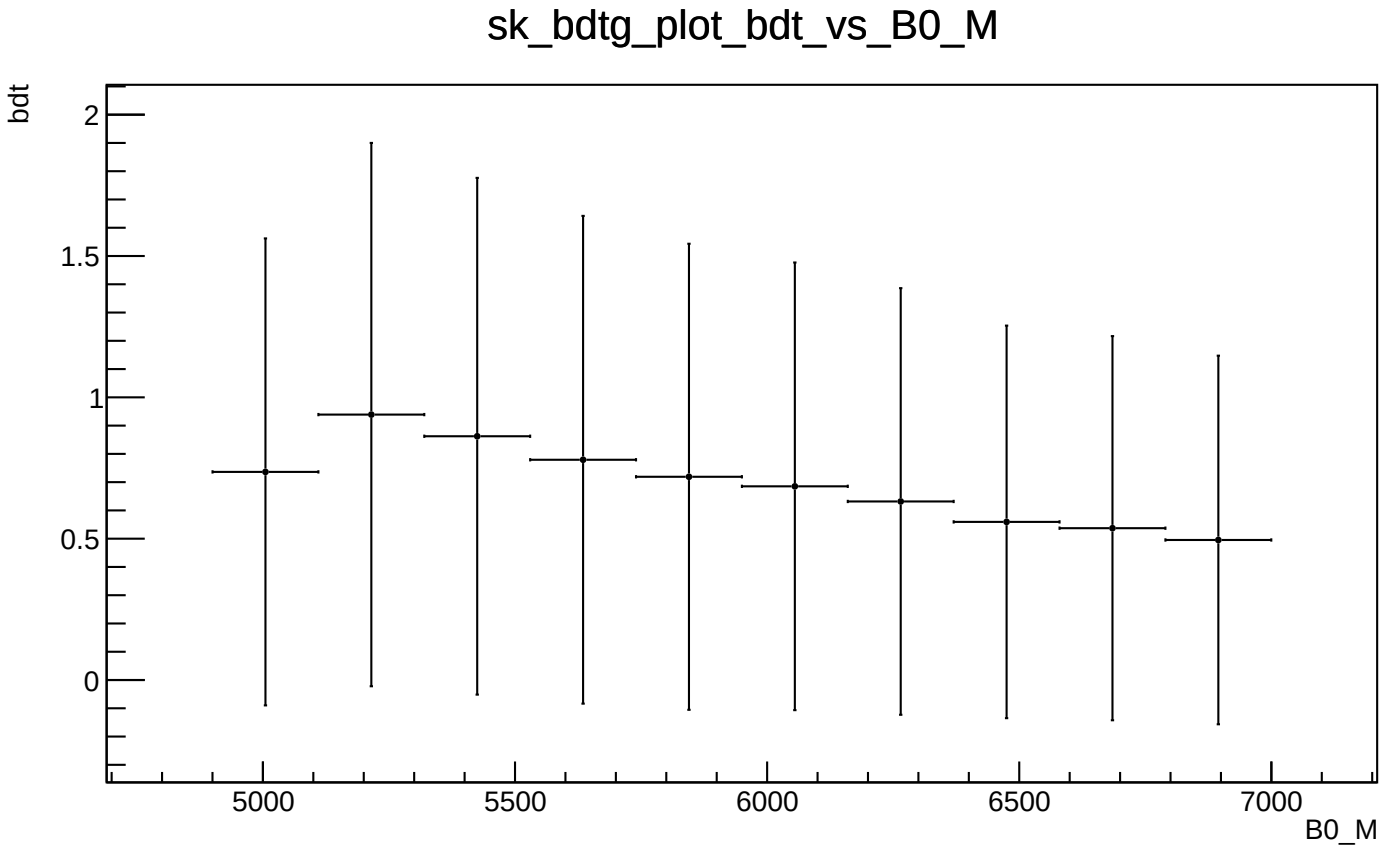

fold_9_roc
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
AdaBoost
uGB+knnAda
uBoost
uGB+FL
xgb
sk_bdtg
sk_bdt

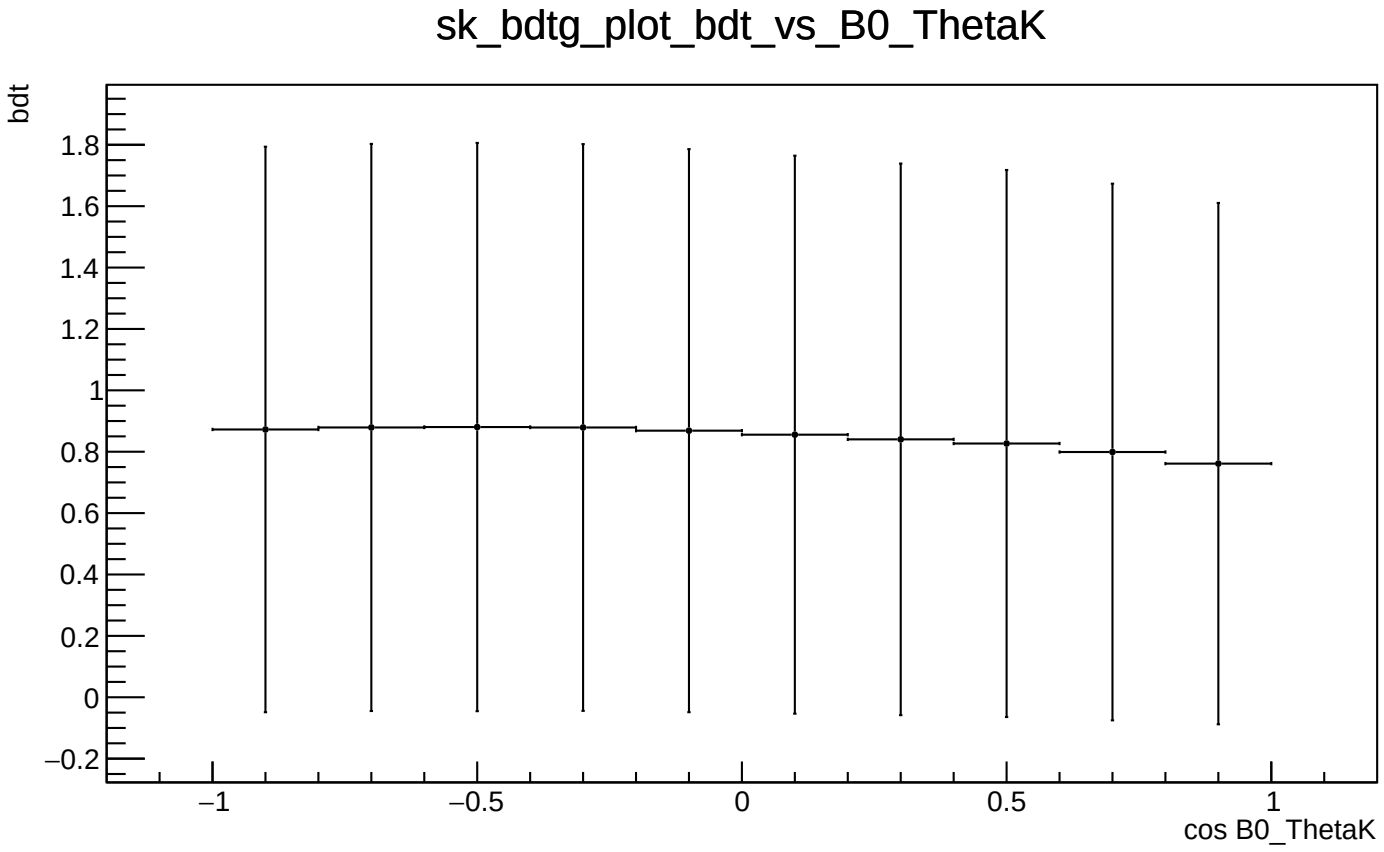
B. Correlation plots

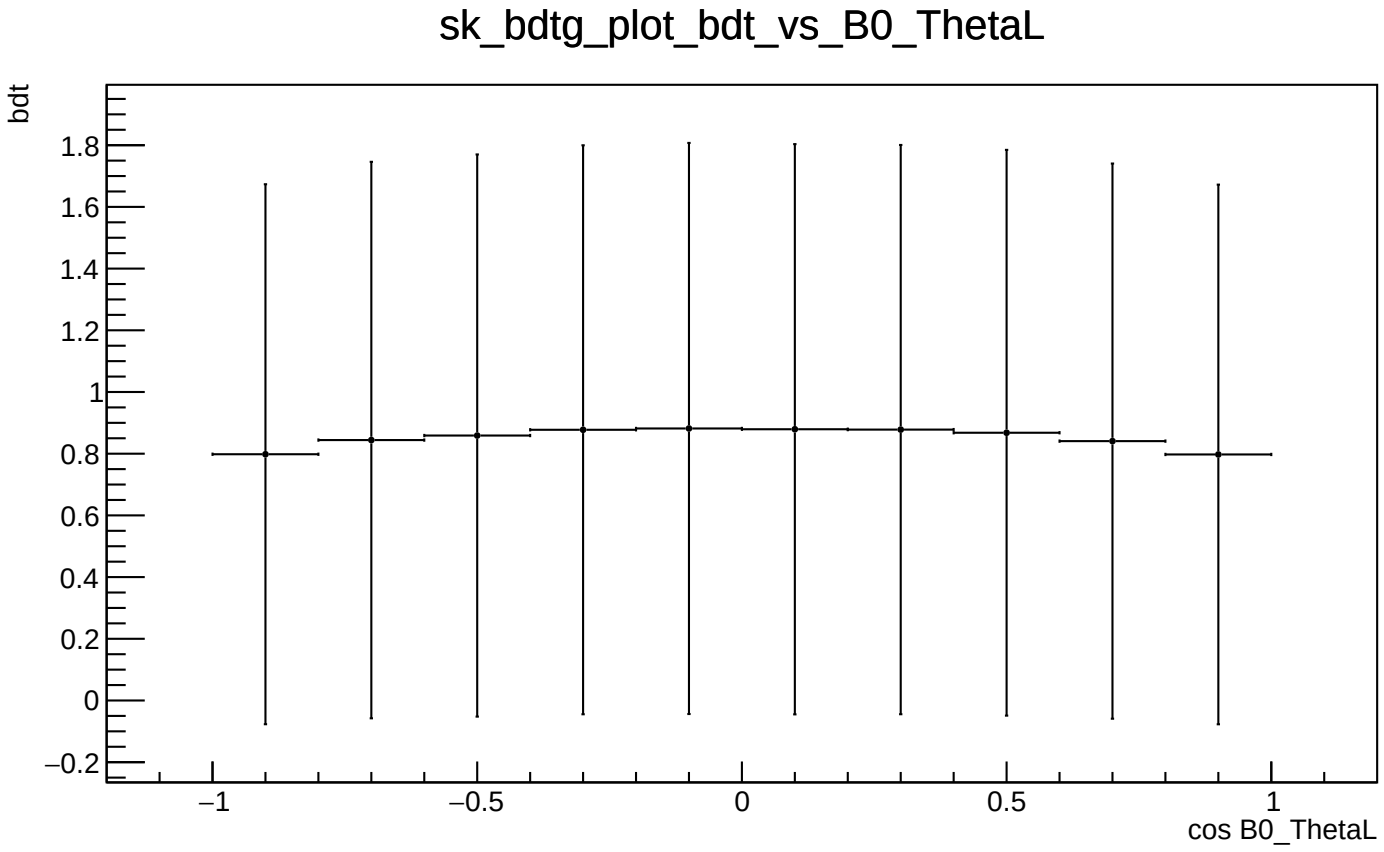

AdaBoost_plot_bdt_vs_B0_M
bdt
2
1.5
1
0.5
0
5000
5500
6000
6500
7000
B0_M

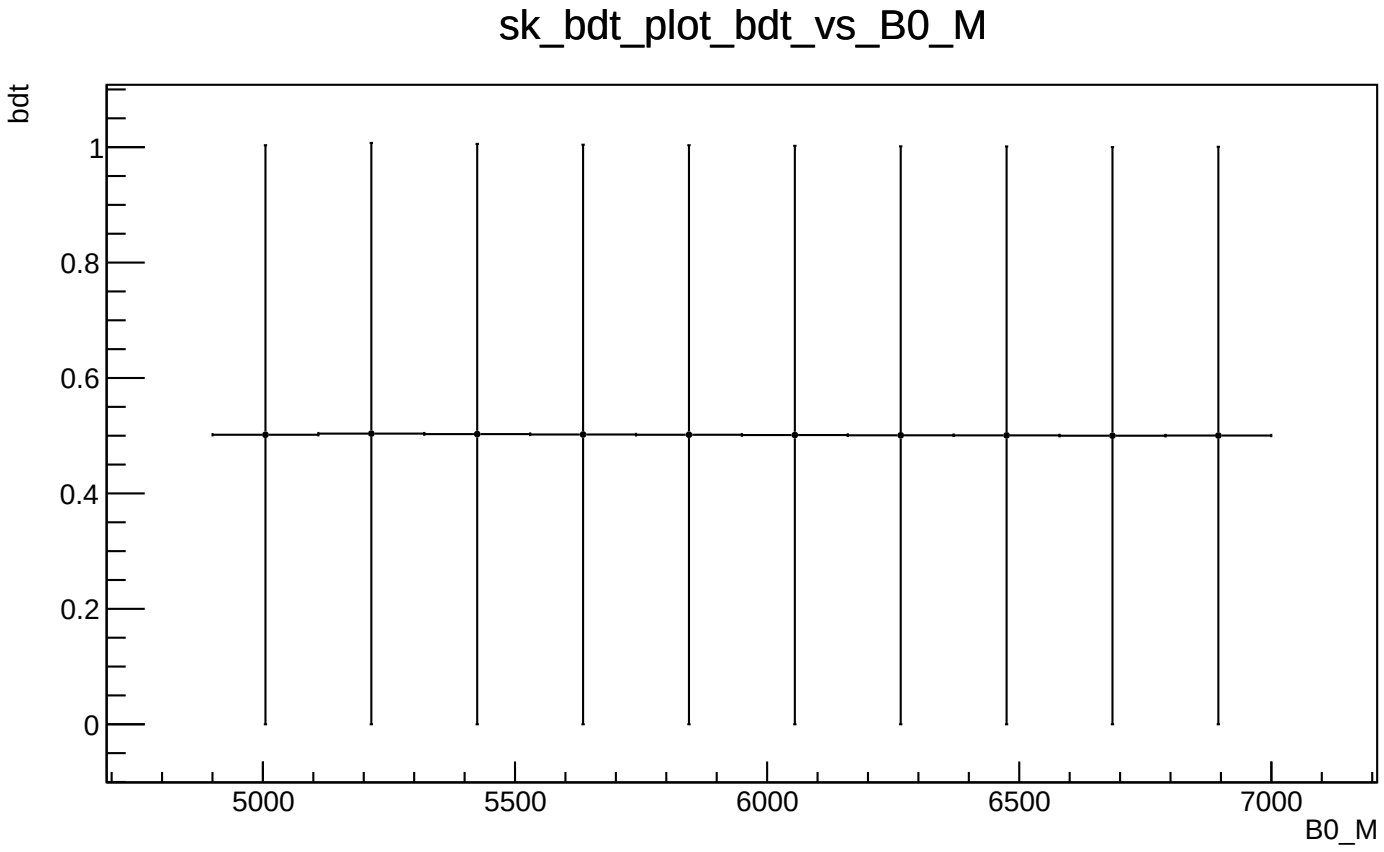

AdaBoost_plot_bdt_vs_B0_ThetaK
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaK

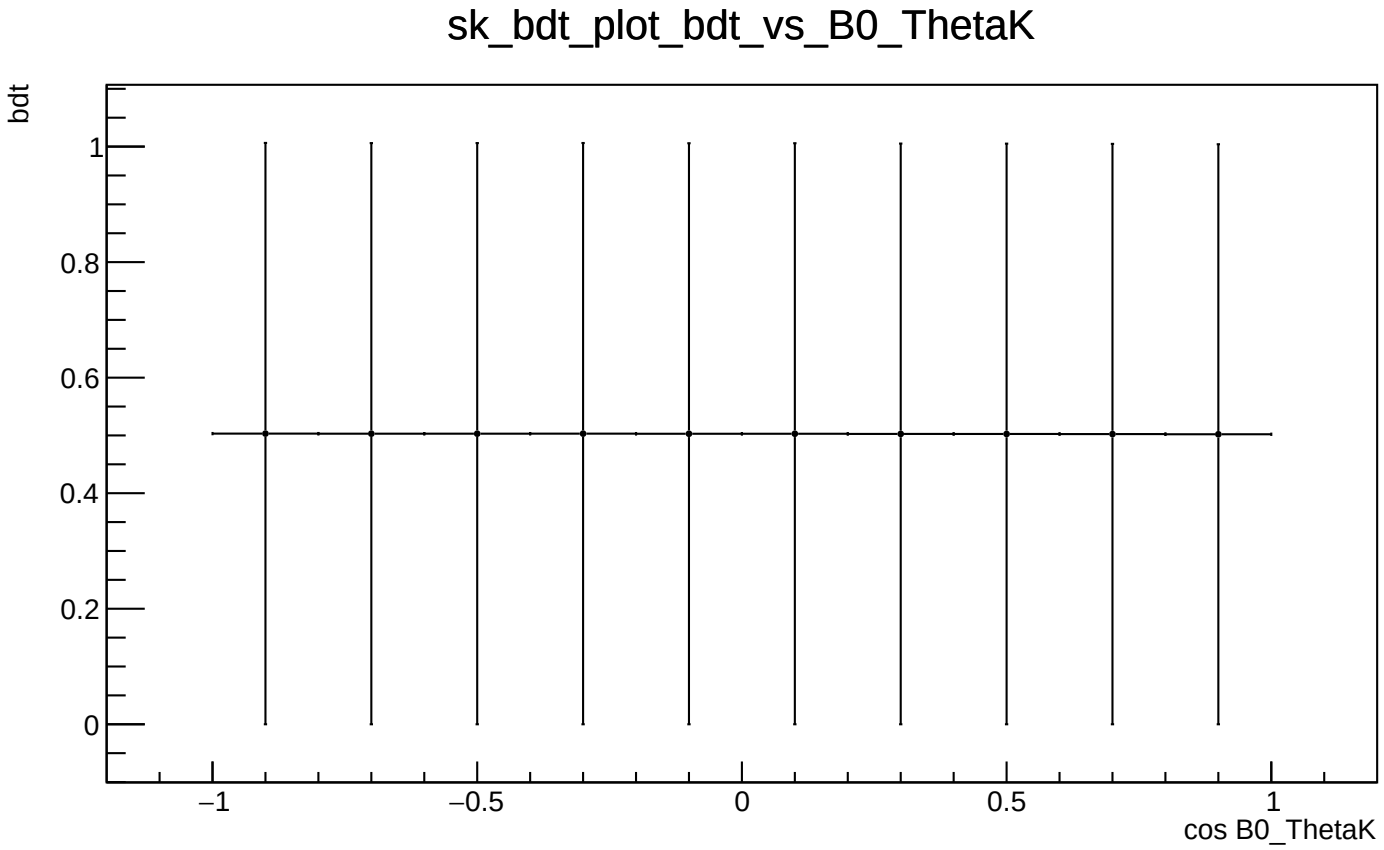

AdaBoost_plot_bdt_vs_B0_ThetaL
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaL

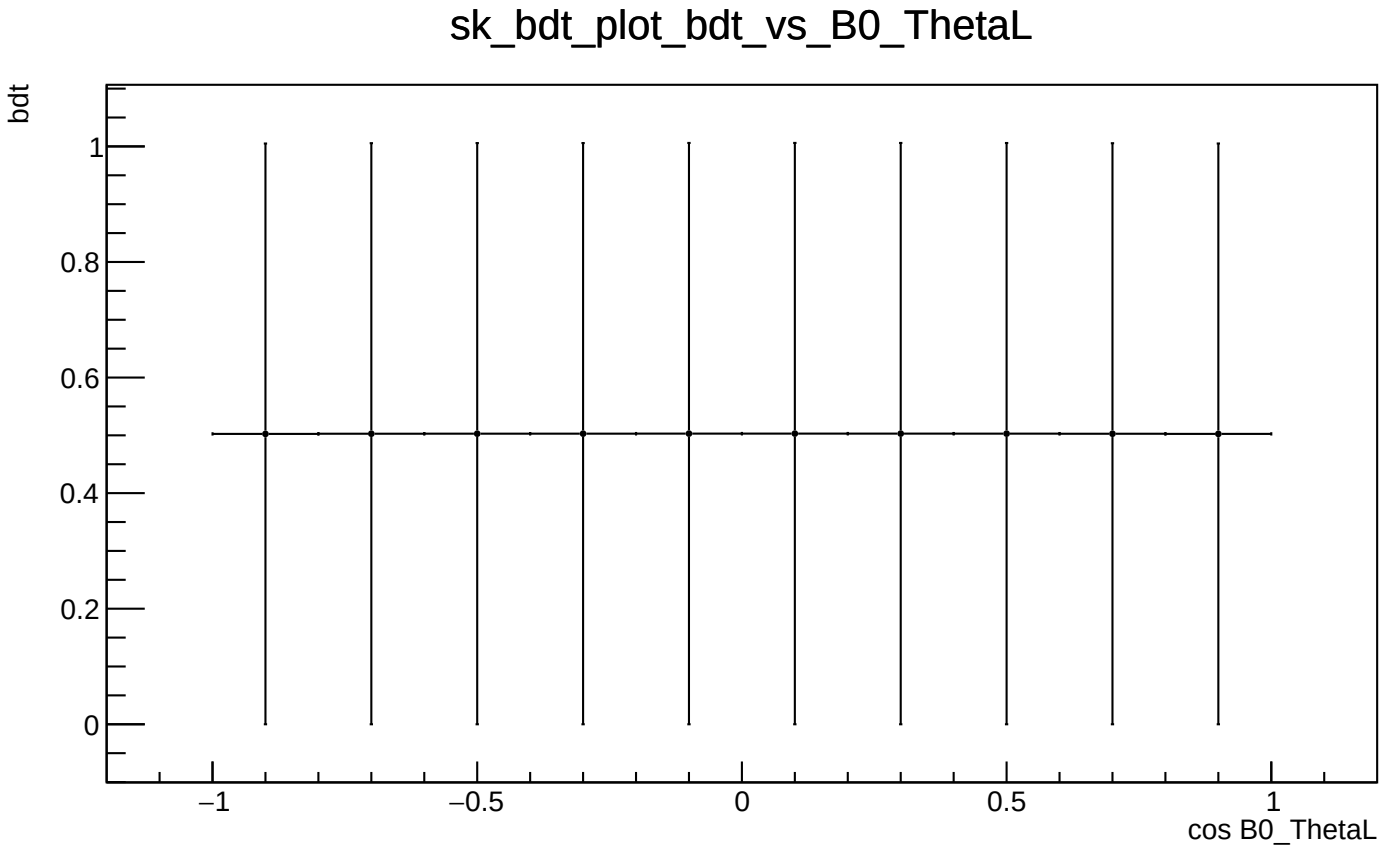

sk_bdtg_plot_bdt_vs_B0_M
bdt
2
1.5
1
0.5
0
5000
5500
6000
6500
7000
B0_M

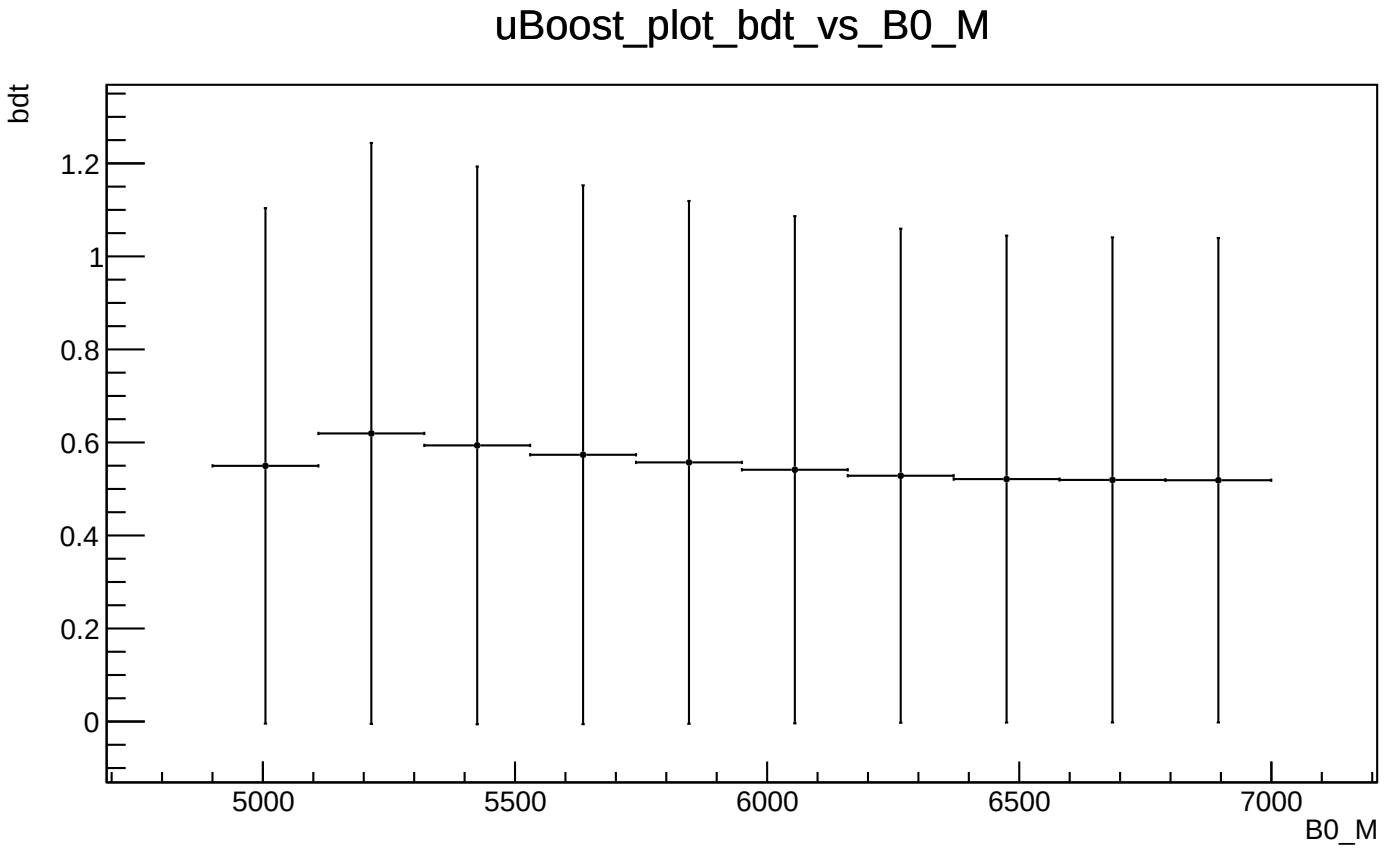

sk_bdtg_plot_bdt_vs_B0_ThetaK
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaK

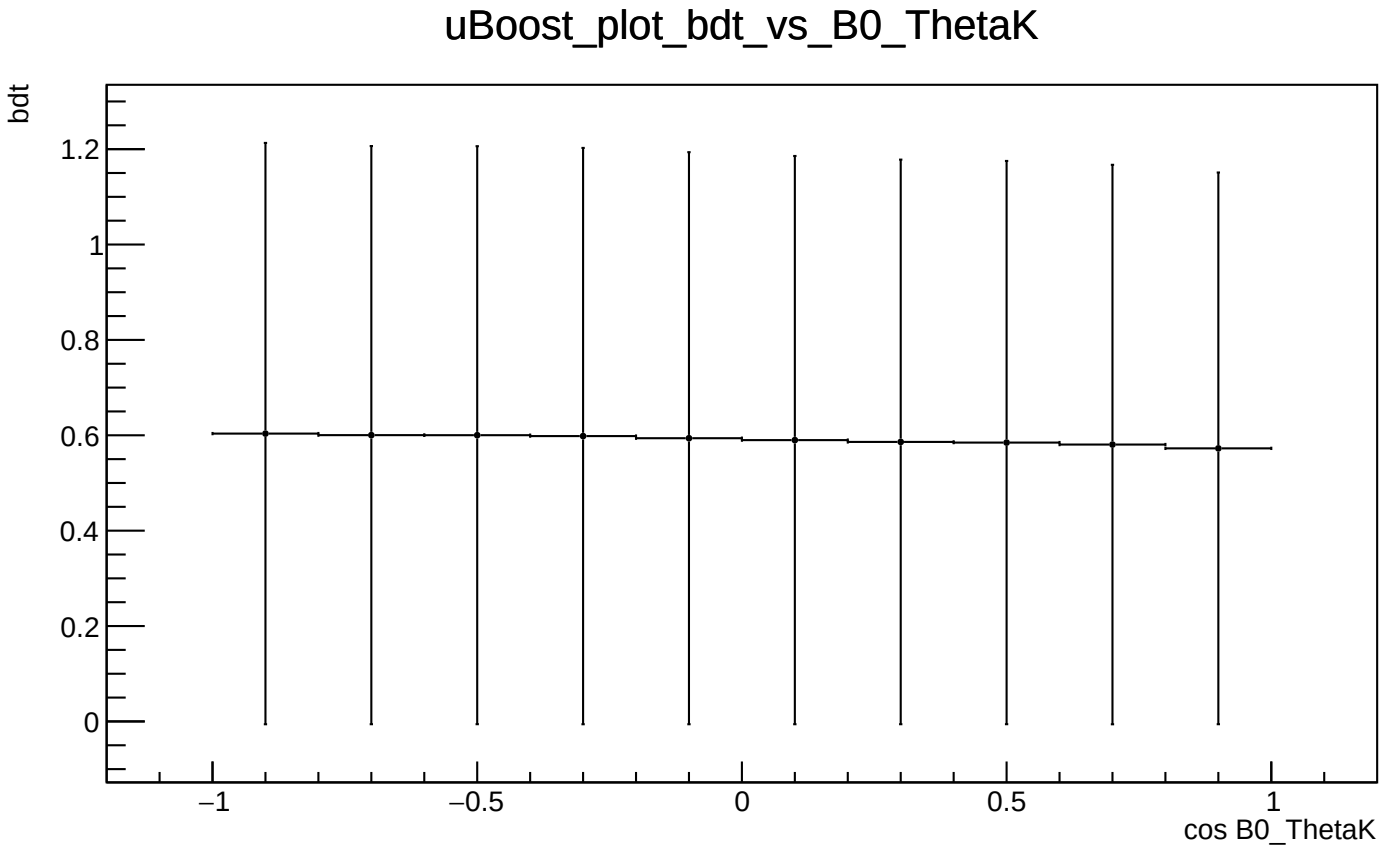

sk_bdtg_plot_bdt_vs_B0_ThetaL
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaL

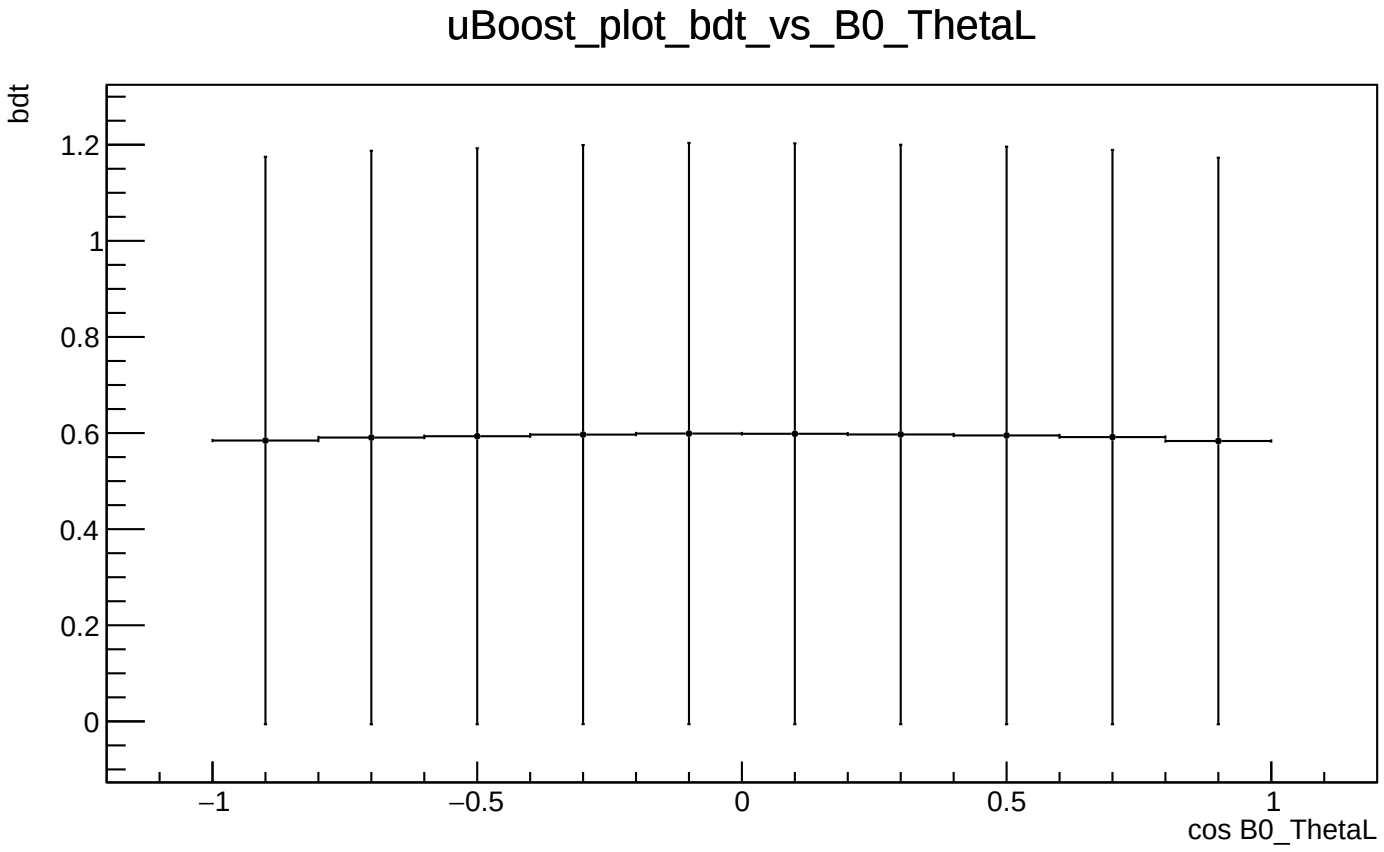

sk_bdt_plot_bdt_vs_B0_M
bdt
1
0.8
0.6
0.4
0.2
0
5000
5500
6000
6500
7000
B0_M

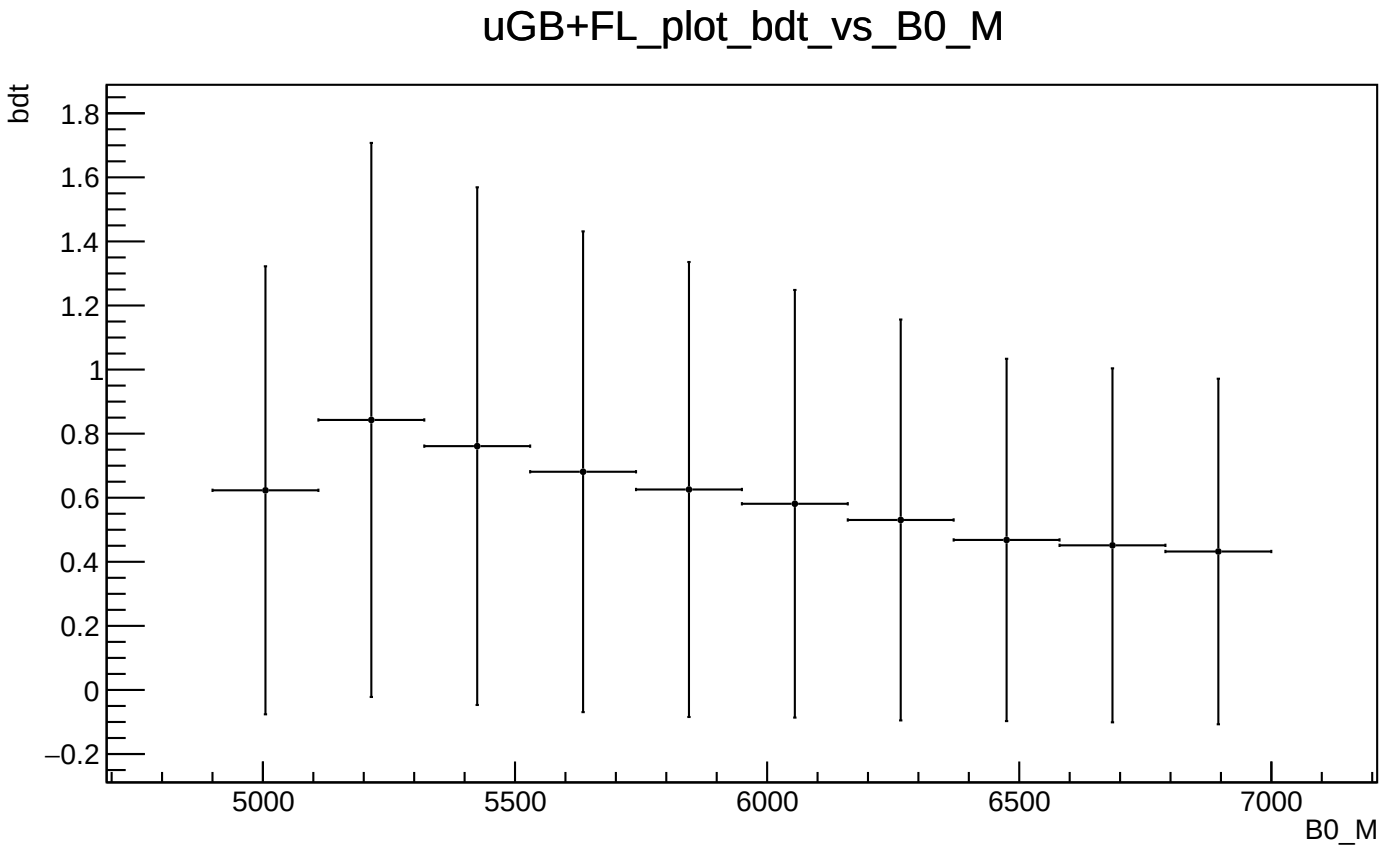

sk_bdt_plot_bdt_vs_B0_ThetaK
bdt
1
0.8
0.6
0.4
0.2
0
−1
−0.5
0
0.5
1
cos B0_ThetaK

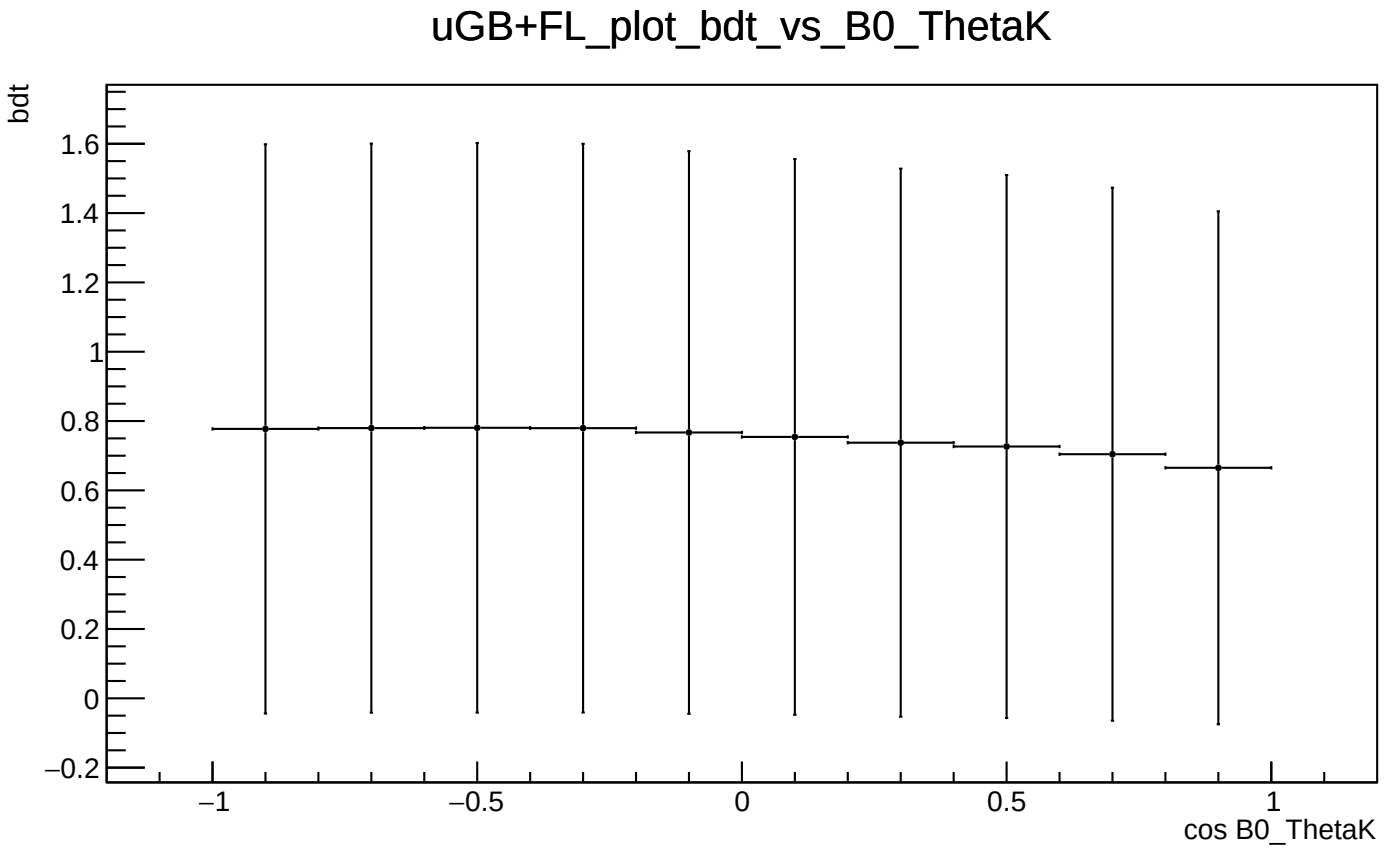

sk_bdt_plot_bdt_vs_B0_ThetaL
bdt
1
0.8
0.6
0.4
0.2
0
−1
−0.5
0
0.5
1
cos B0_ThetaL

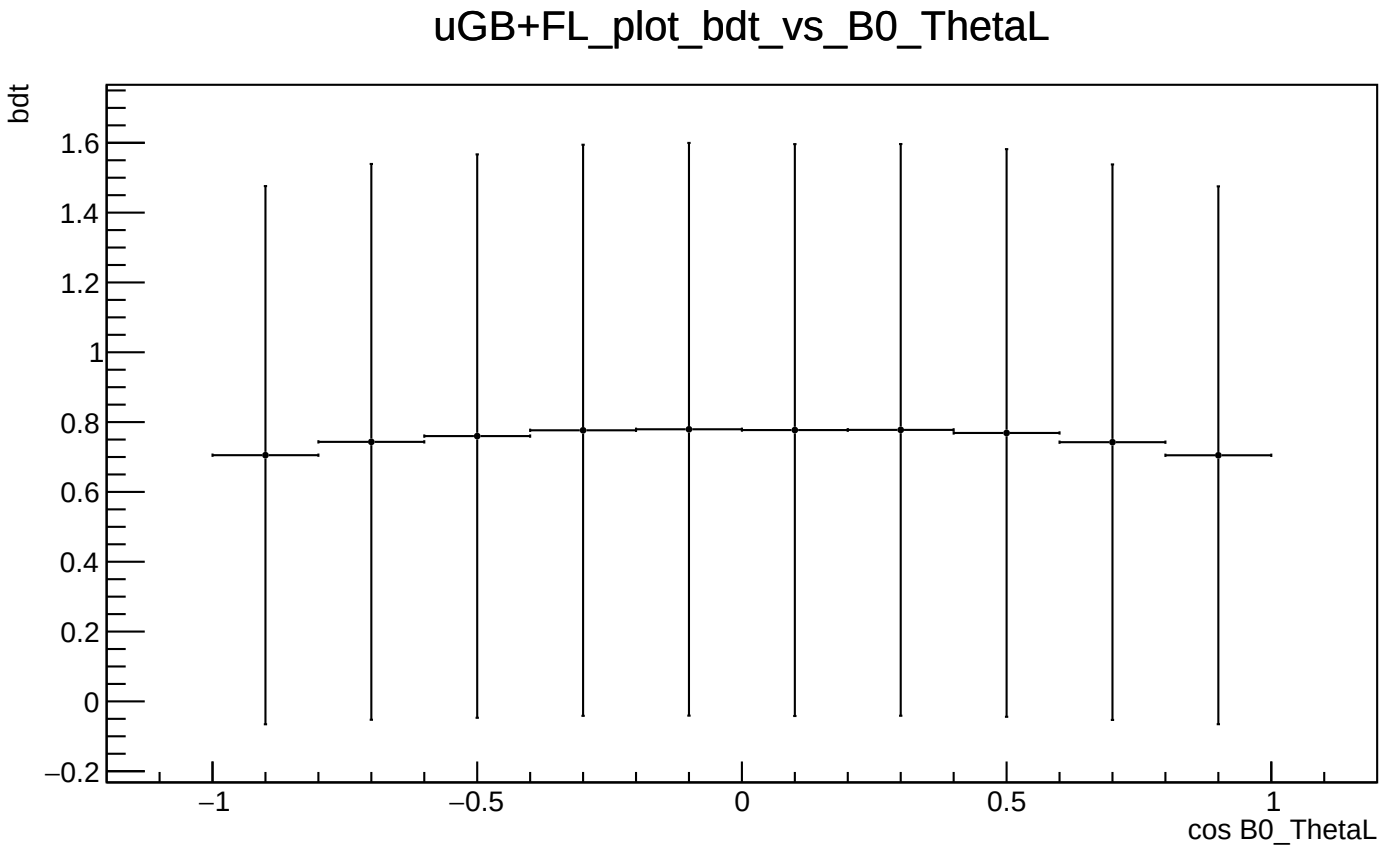

uBoost_plot_bdt_vs_B0_M
bdt
1.2
1
0.8
0.6
0.4
0.2
0
5000
5500
6000
6500
7000
B0_M

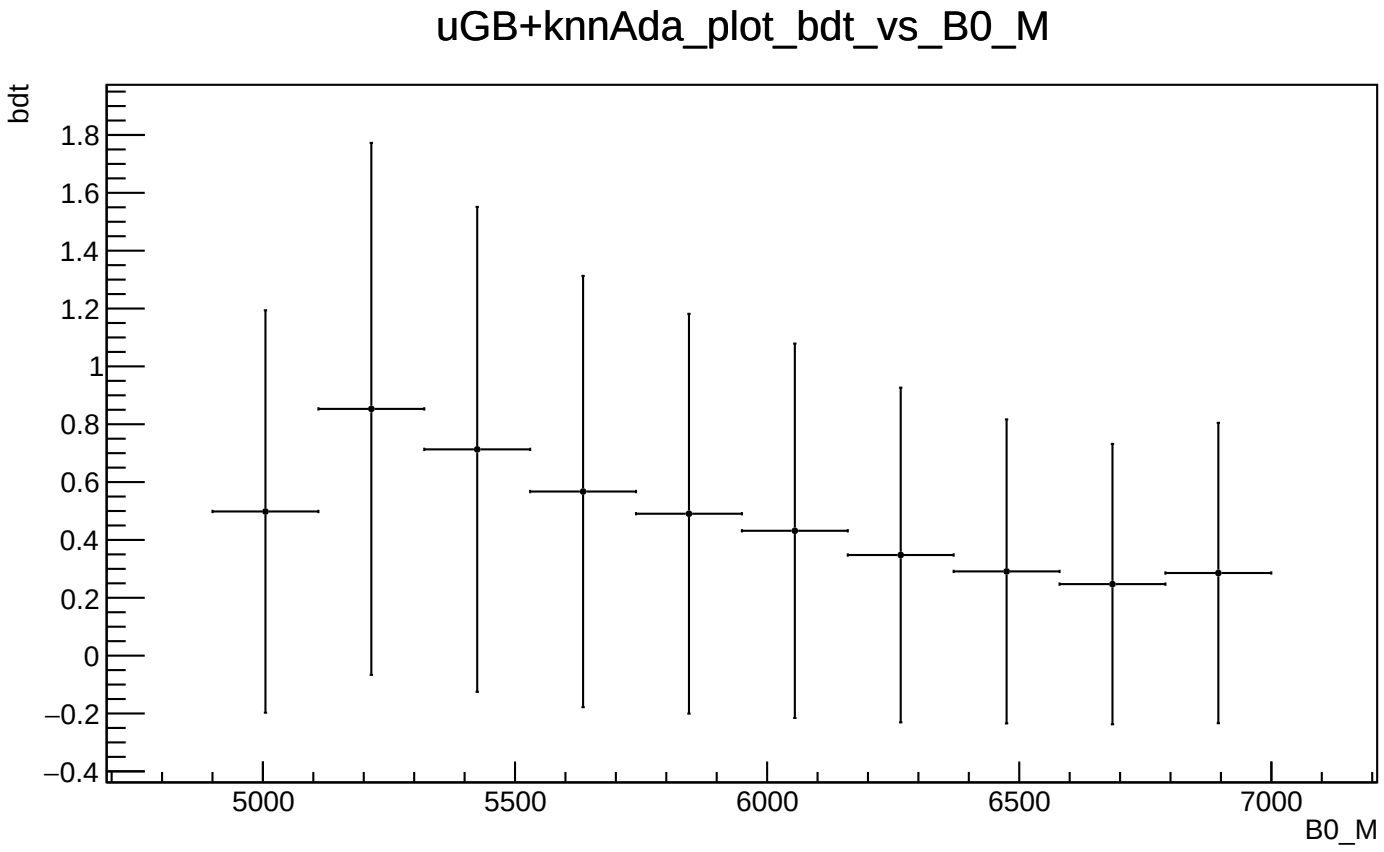

uBoost_plot_bdt_vs_B0_ThetaK
bdt
1.2
1
0.8
0.6
0.4
0.2
0
−1
−0.5
0
0.5
1
cos B0_ThetaK

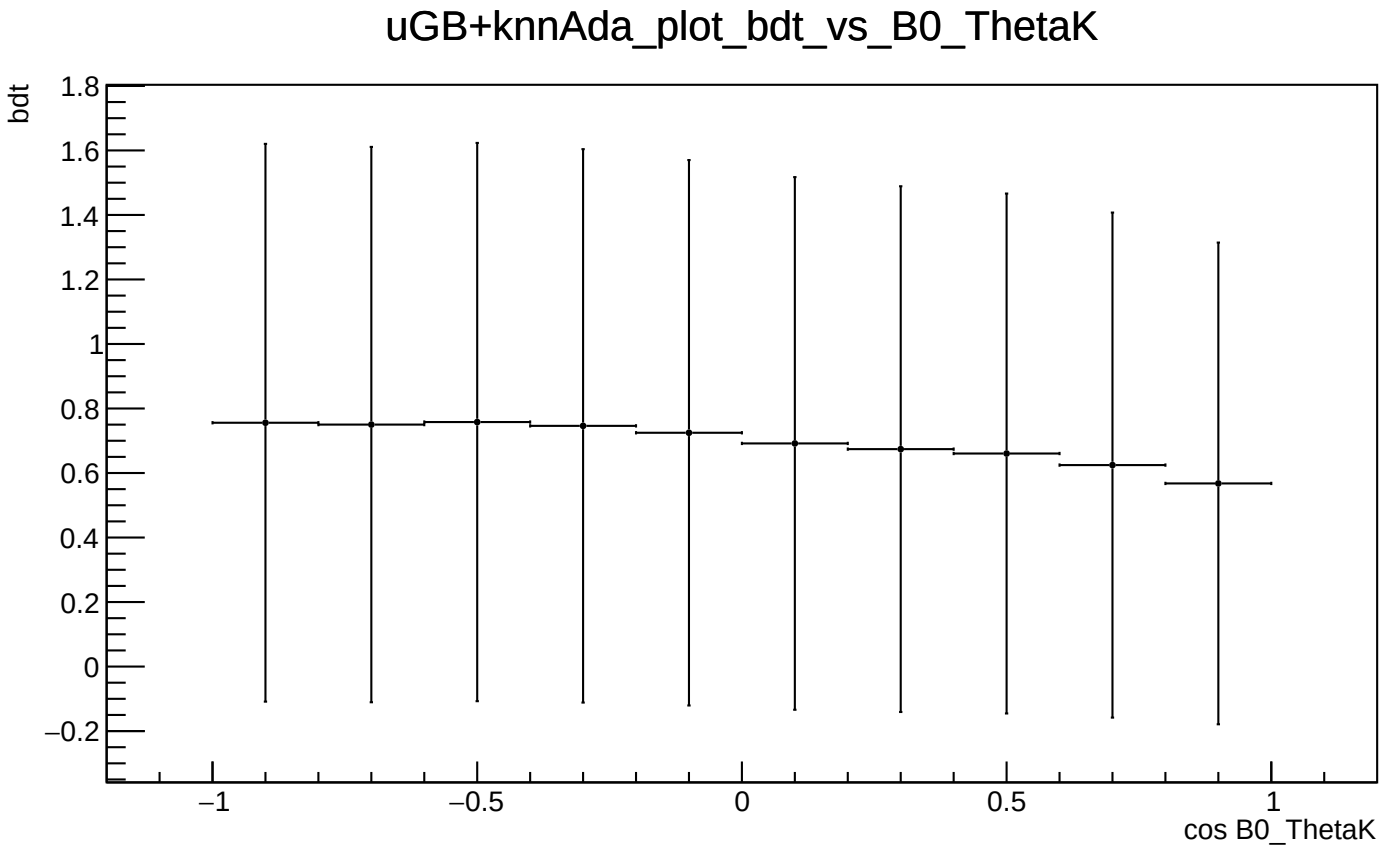

uBoost_plot_bdt_vs_B0_ThetaL
bdt
1.2
1
0.8
0.6
0.4
0.2
0
−1
−0.5
0
0.5
1
cos B0_ThetaL

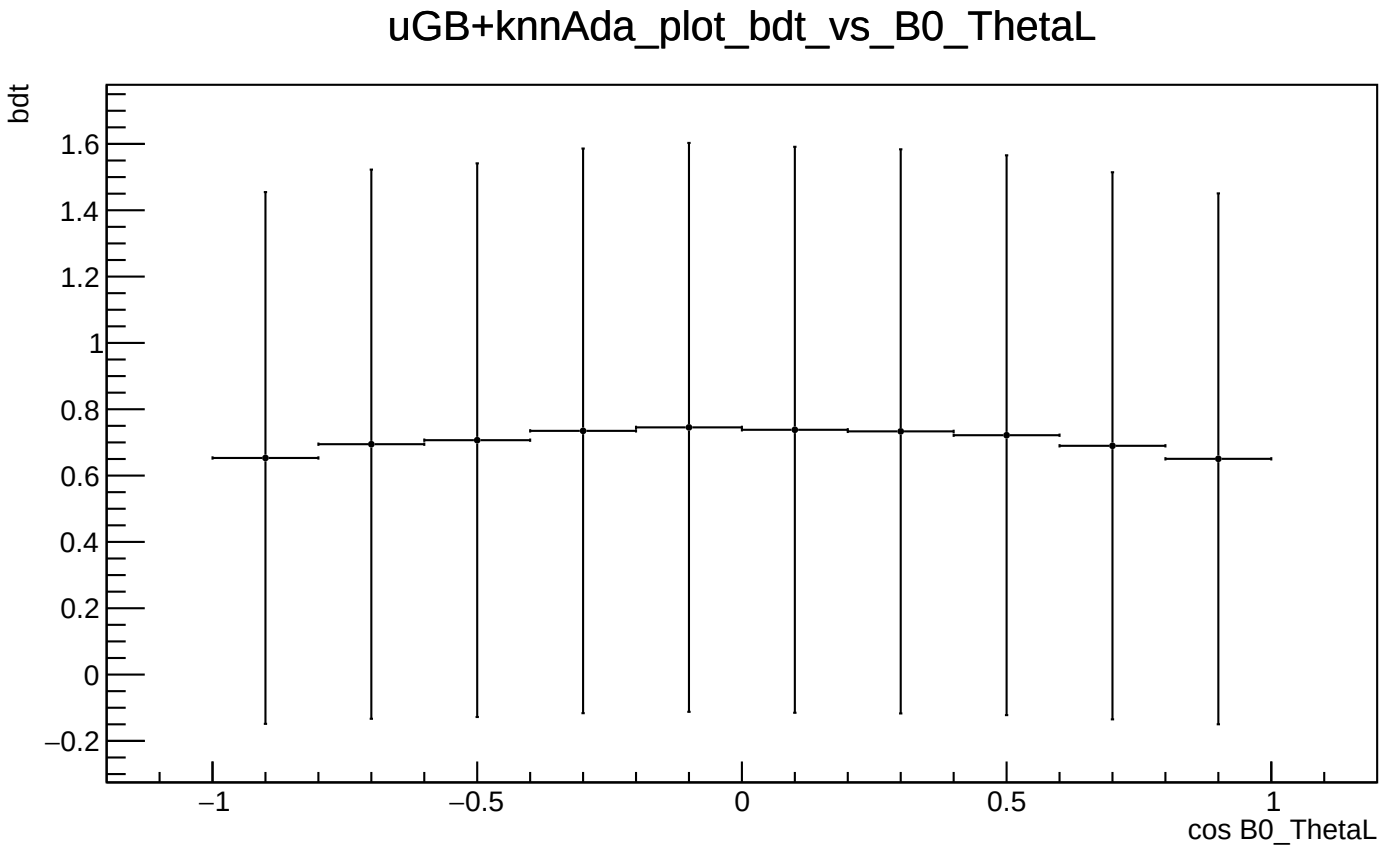

uGB+FL_plot_bdt_vs_B0_M
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
5000
5500
6000
6500
7000
B0_M

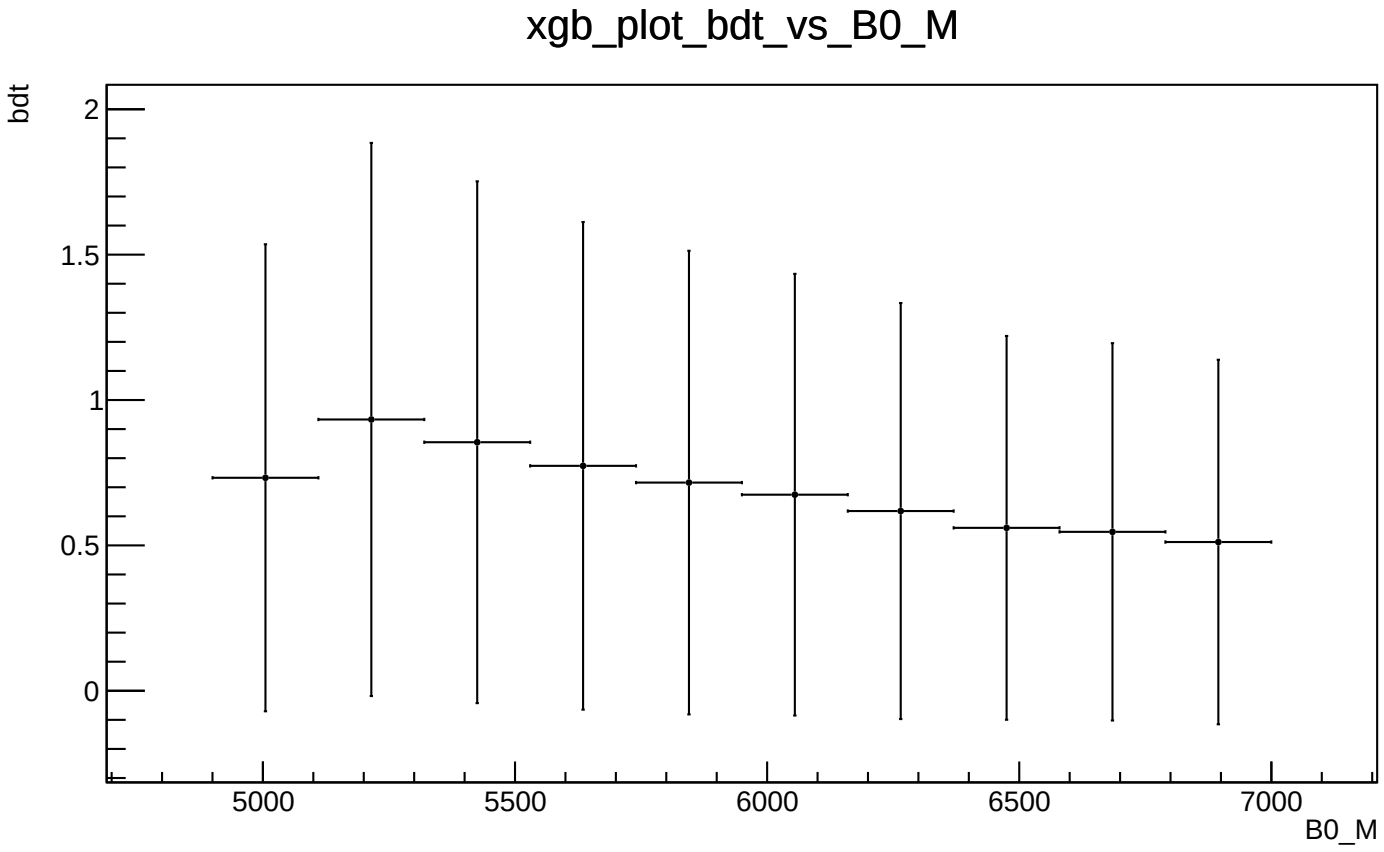

uGB+FL_plot_bdt_vs_B0_ThetaK
bdt
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaK

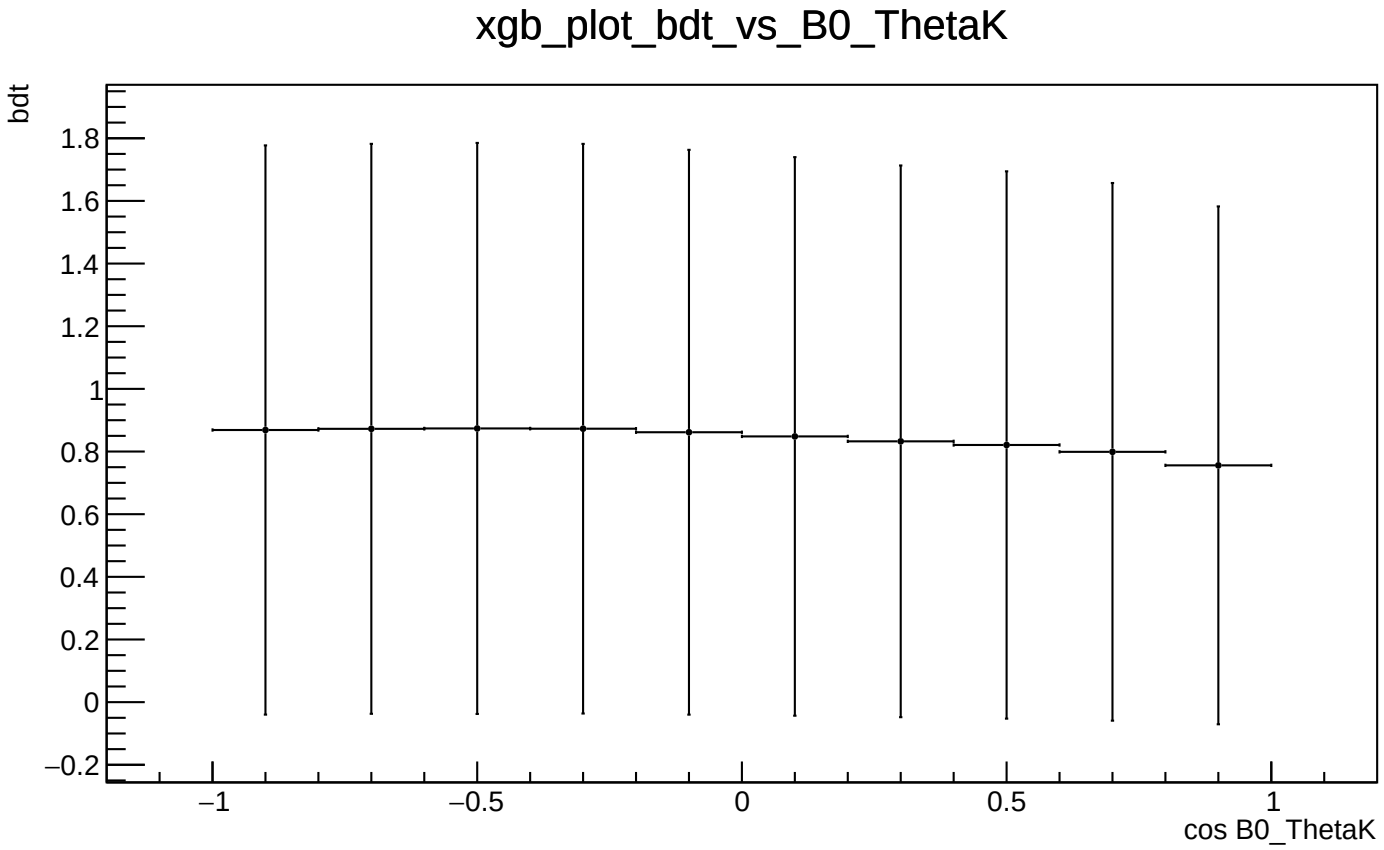

uGB+FL_plot_bdt_vs_B0_ThetaL
bdt
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaL

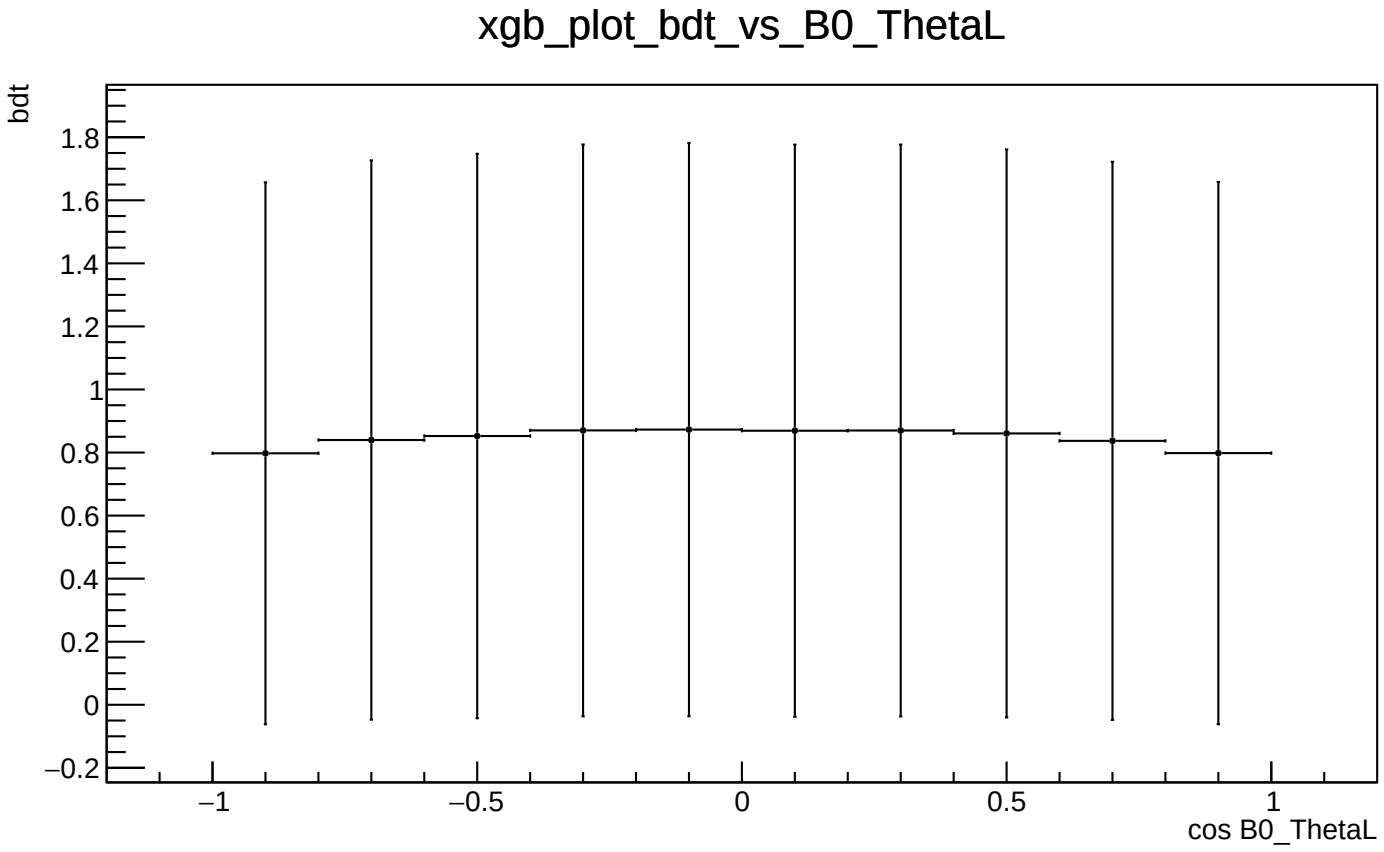

uGB+knnAda_plot_bdt_vs_B0_M
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
5000
5500
6000
6500
7000
B0_M


uGB+knnAda_plot_bdt_vs_B0_ThetaK
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaK

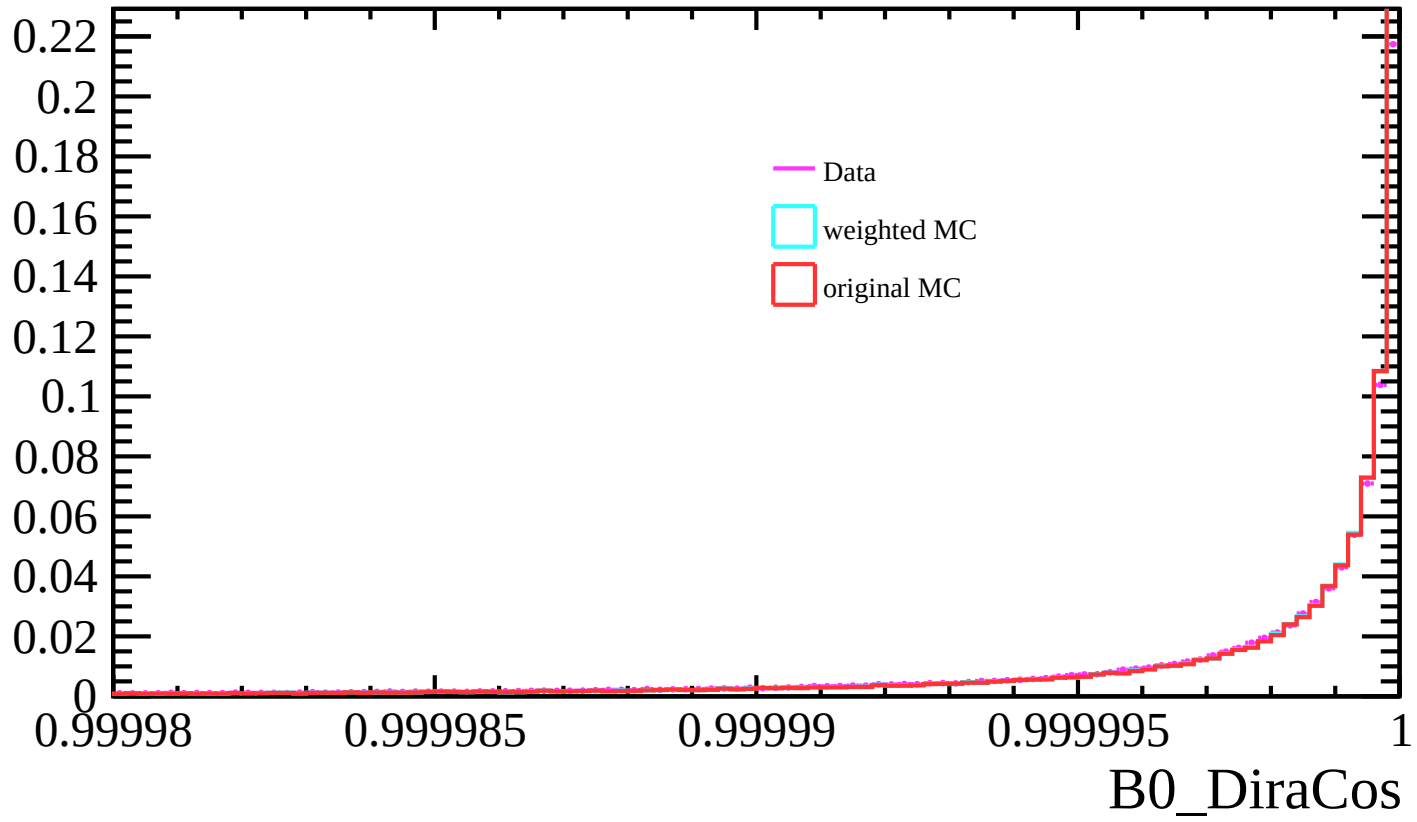

uGB+knnAda_plot_bdt_vs_B0_ThetaL
bdt
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaL

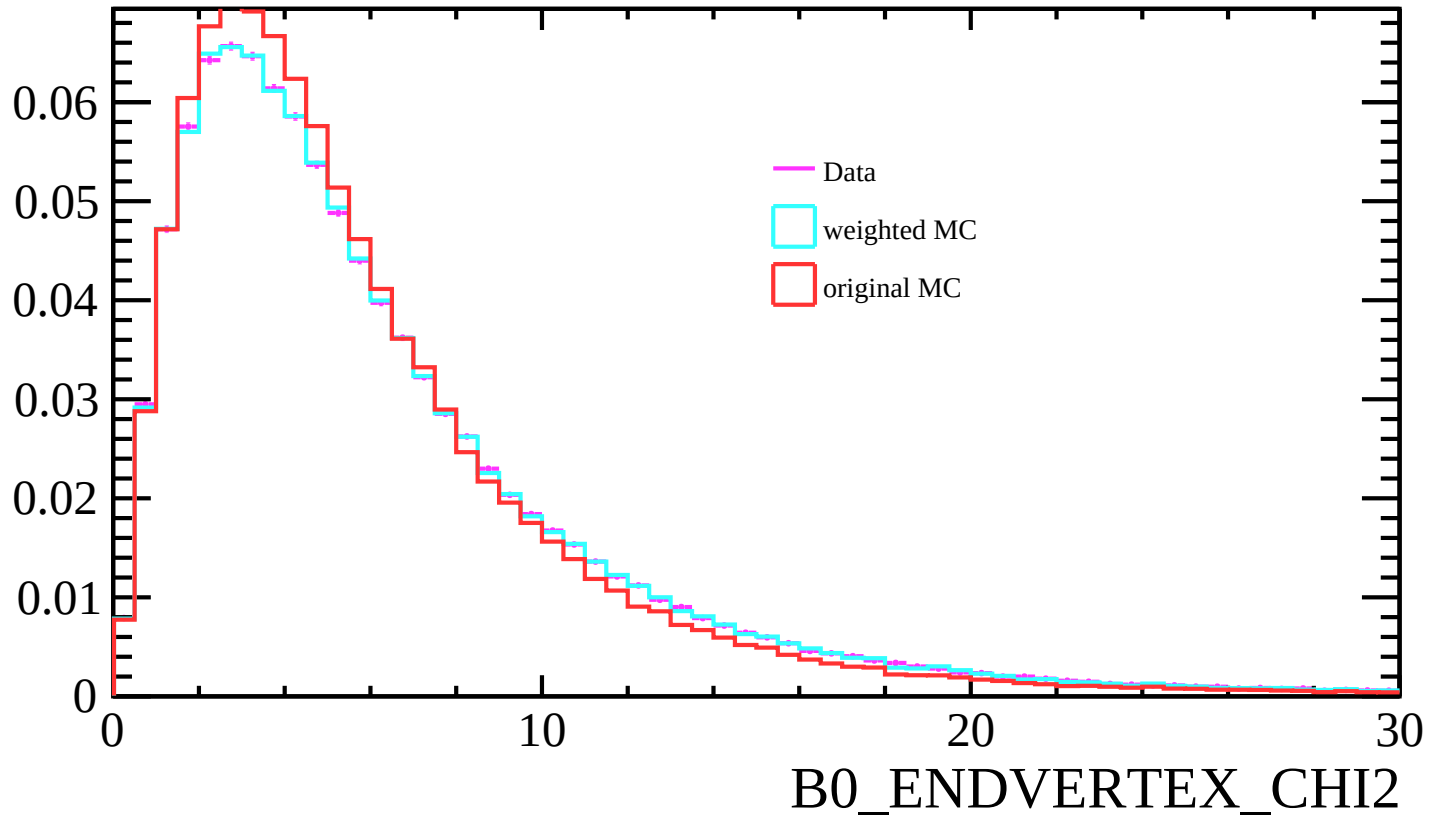

xgb_plot_bdt_vs_B0_M
bdt
2
1.5
1
0.5
0
5000
5500
6000
6500
7000
B0_M

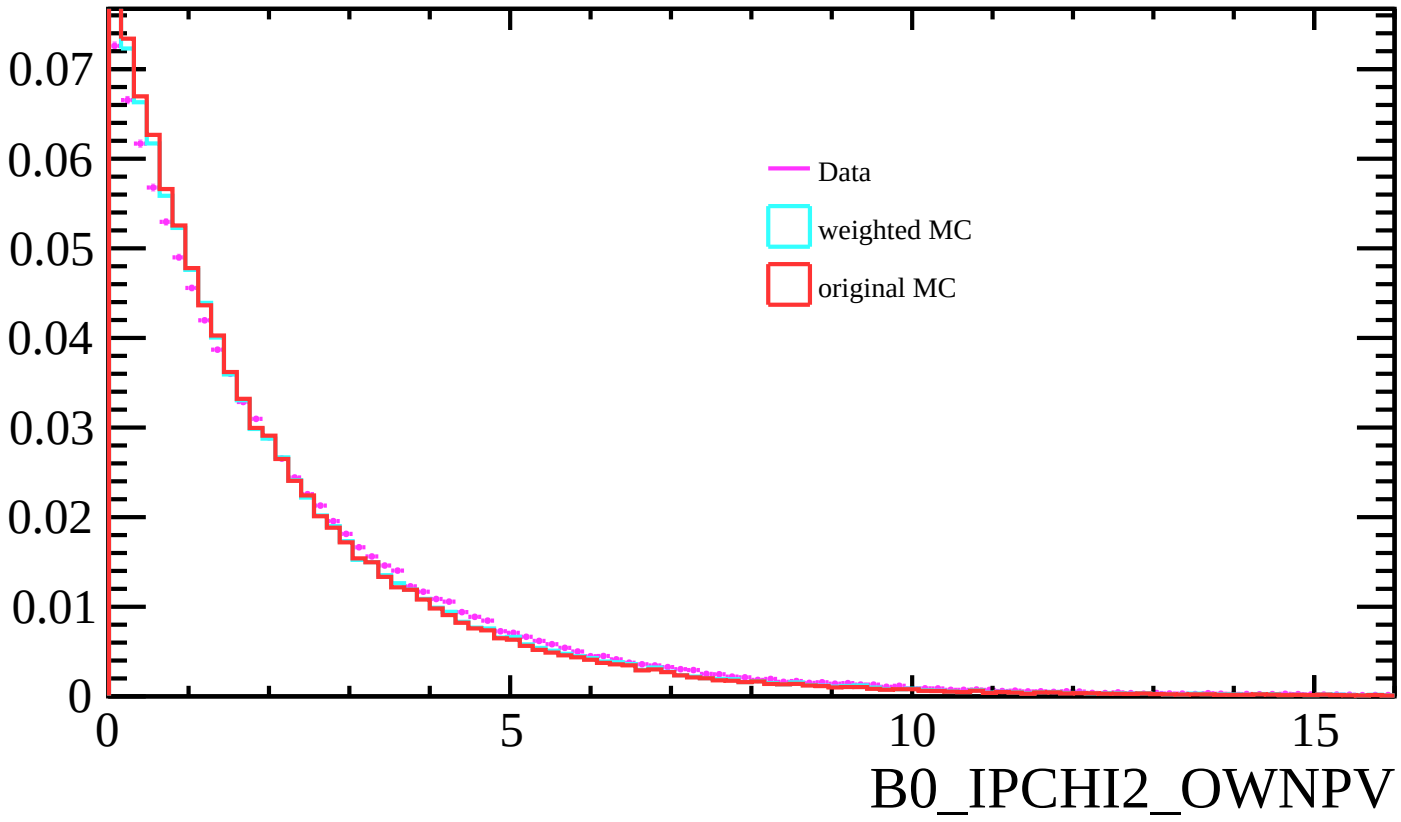

xgb_plot_bdt_vs_B0_ThetaK
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaK

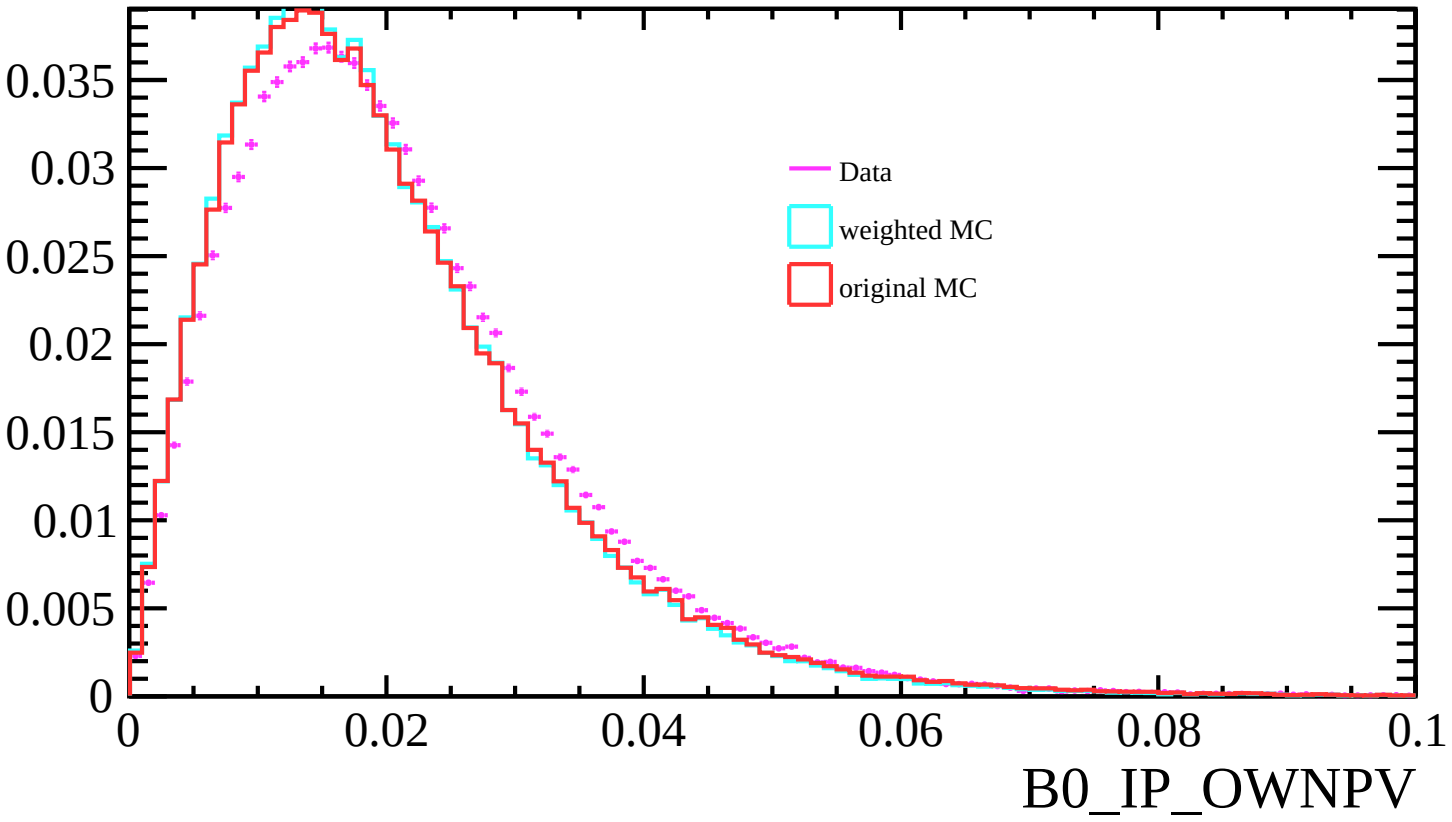

xgb_plot_bdt_vs_B0_ThetaL
bdt
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.5
0
0.5
1
cos B0_ThetaL

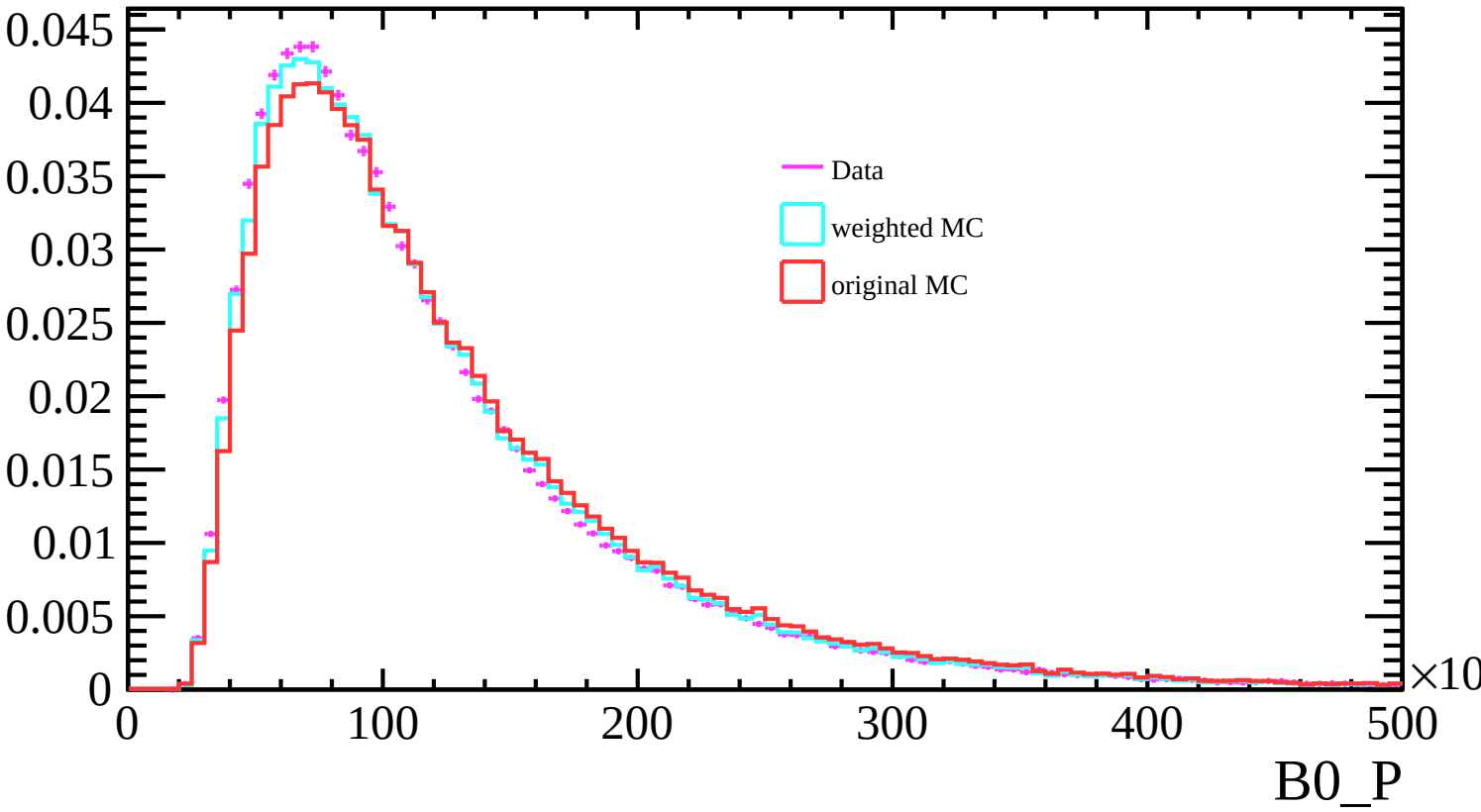
C. Reweighting plots

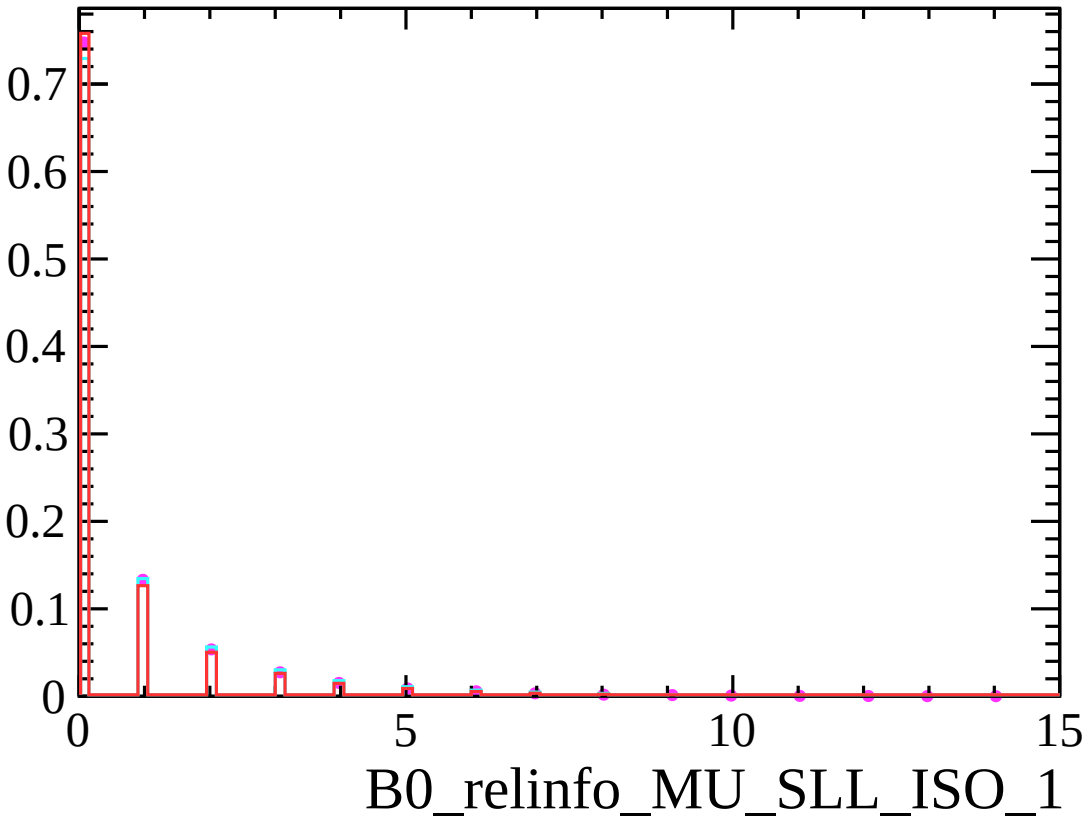

0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0.99998
0.999985
0.99999
0.999995
1
B0_DiraCos
Data
weighted MC
original MC

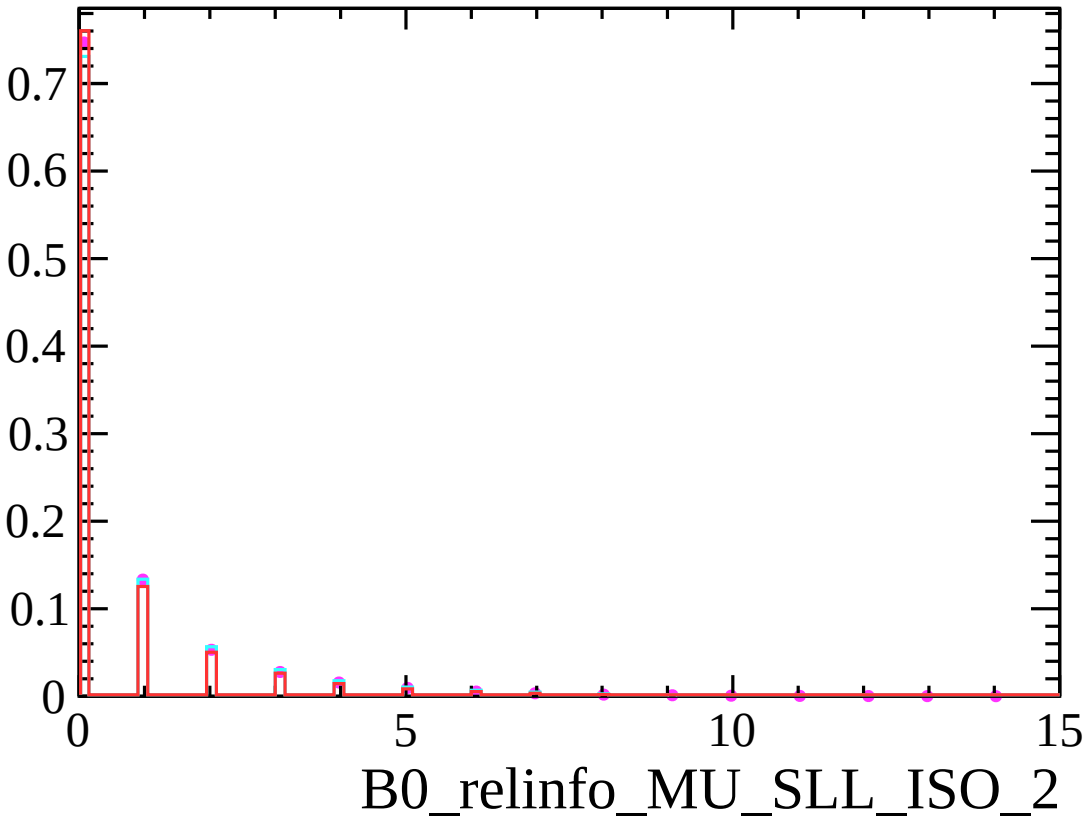

0.06
0.05
0.04
0.03
0.02
0.01
0
0
10
20
30
Data
weighted MC
original MC
B0_ENDVERTEX_CHI2

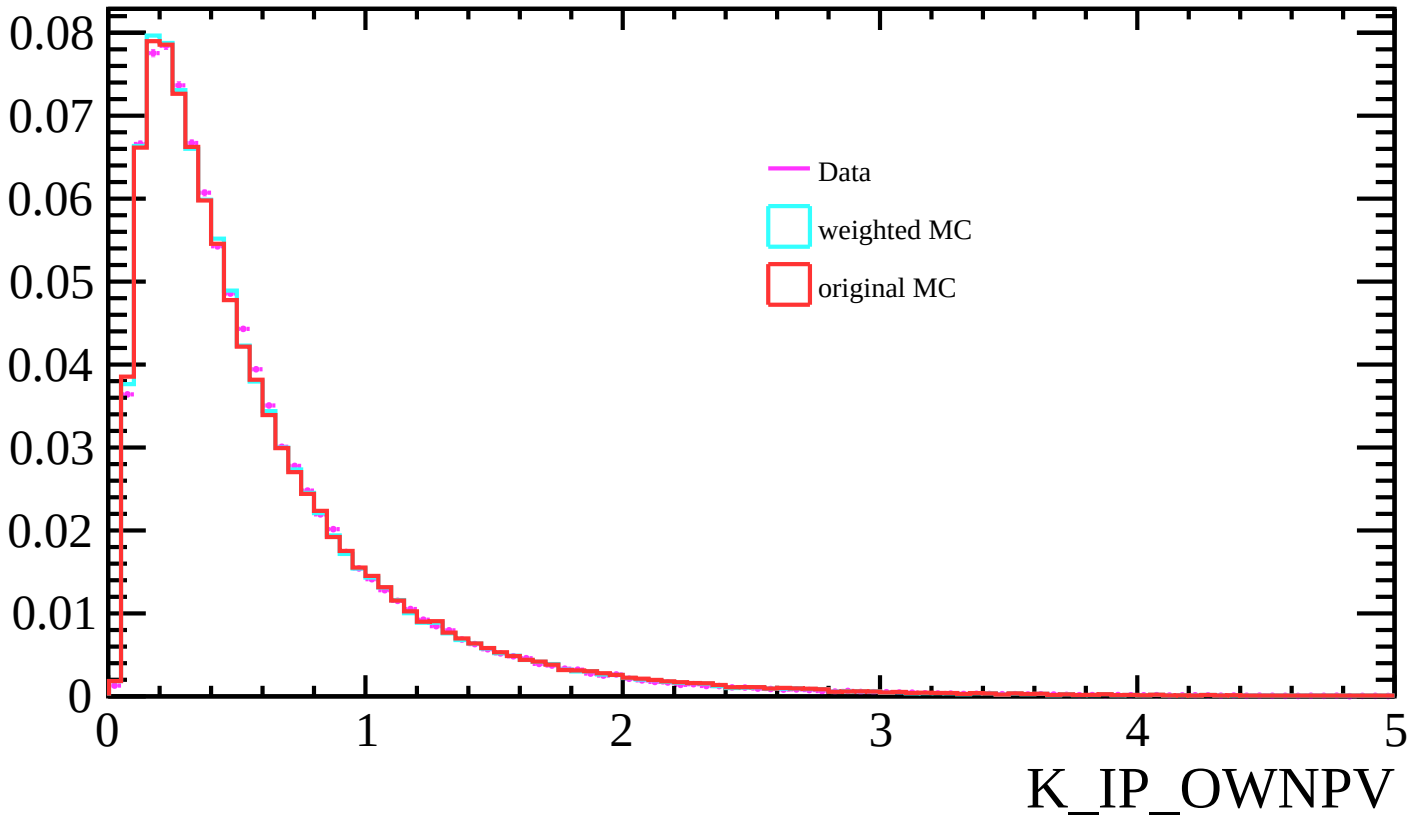

0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
5
10
15
B0_IPCHI2_OWNPV
Data
weighted MC
original MC

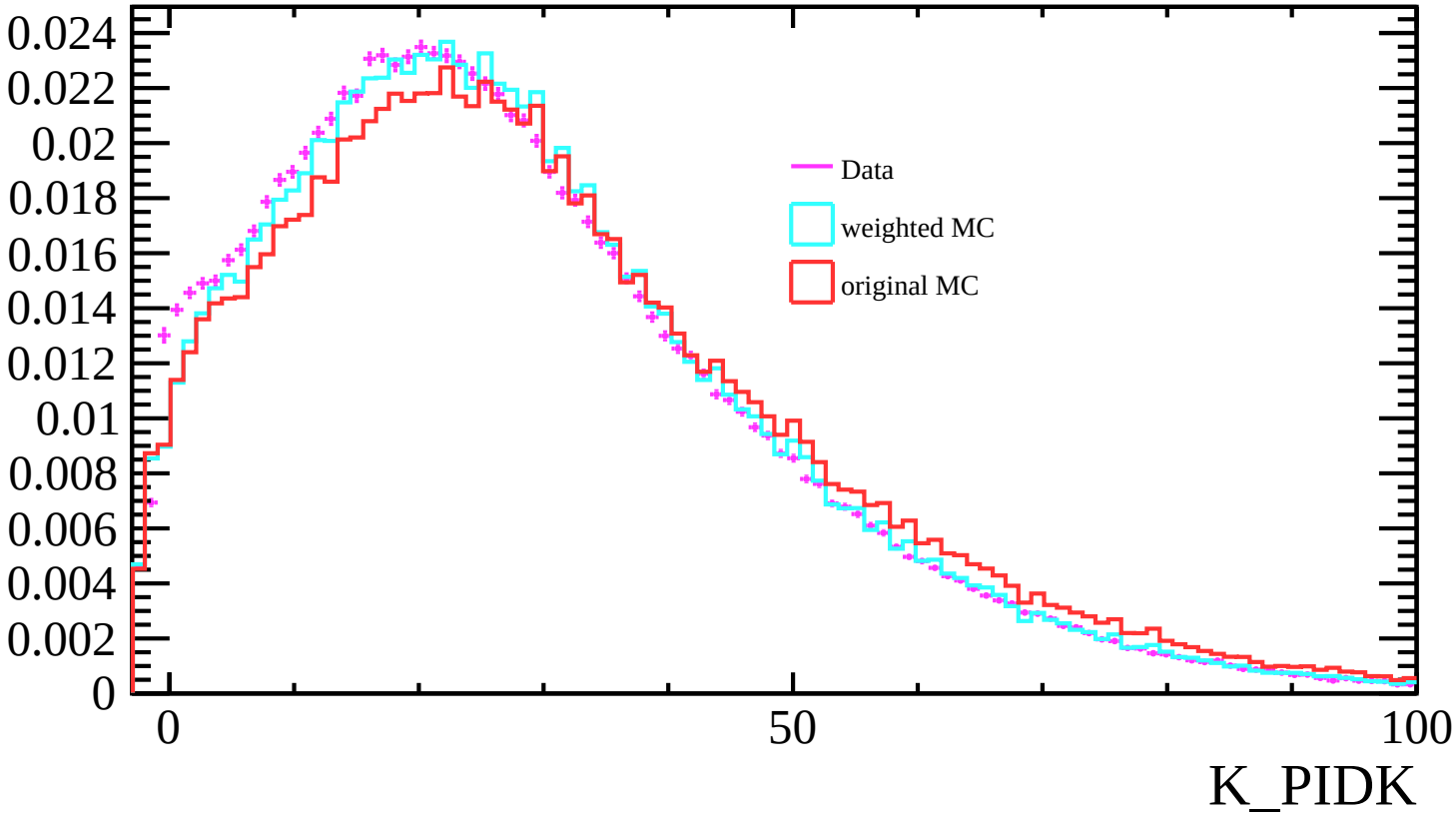

0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
0.02
0.04
0.06
0.08
0.1
B0_IP_OWNPV
Data
weighted MC
original MC

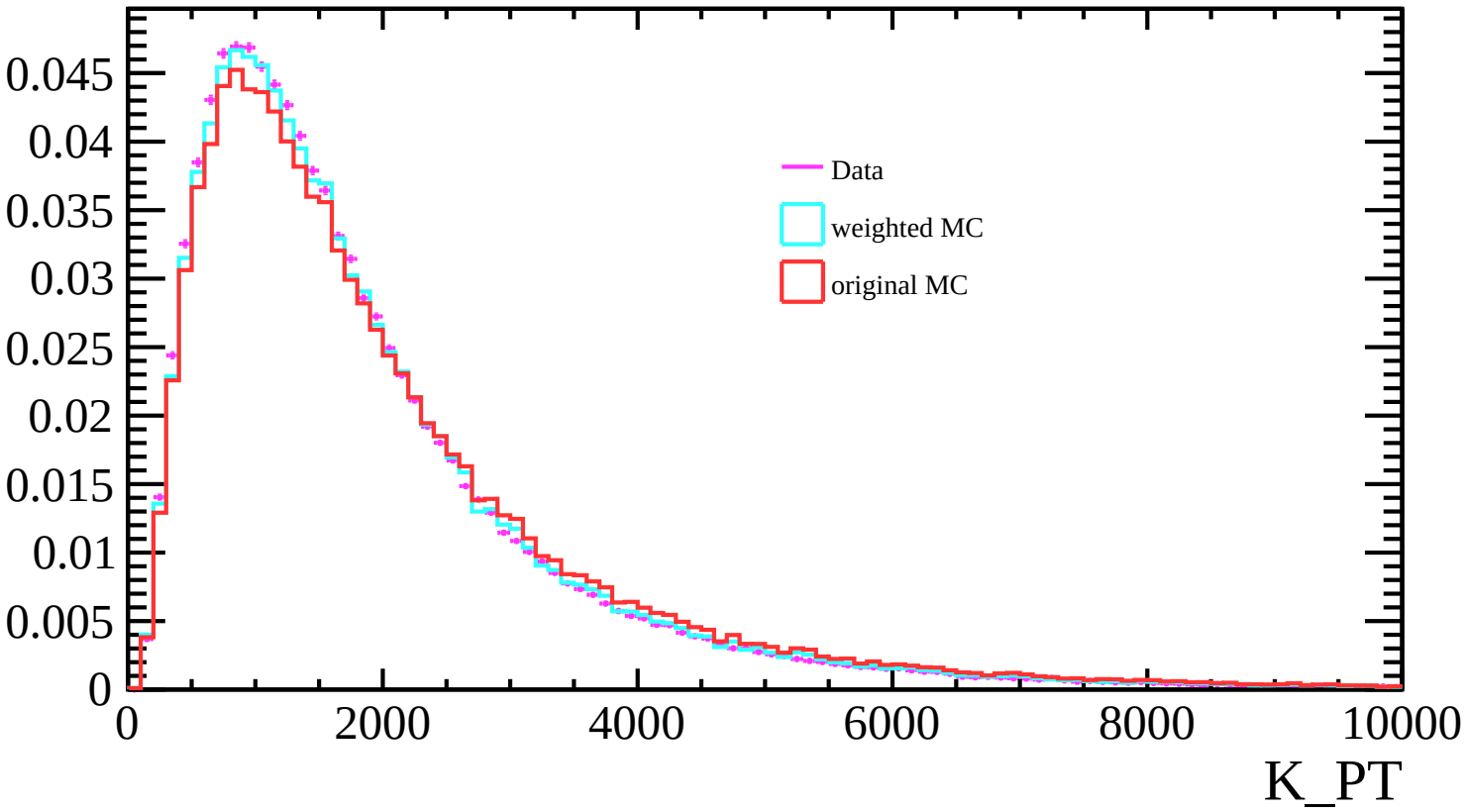

0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
100
200
300
400
500
×10
B0_P
Data
weighted MC
original MC

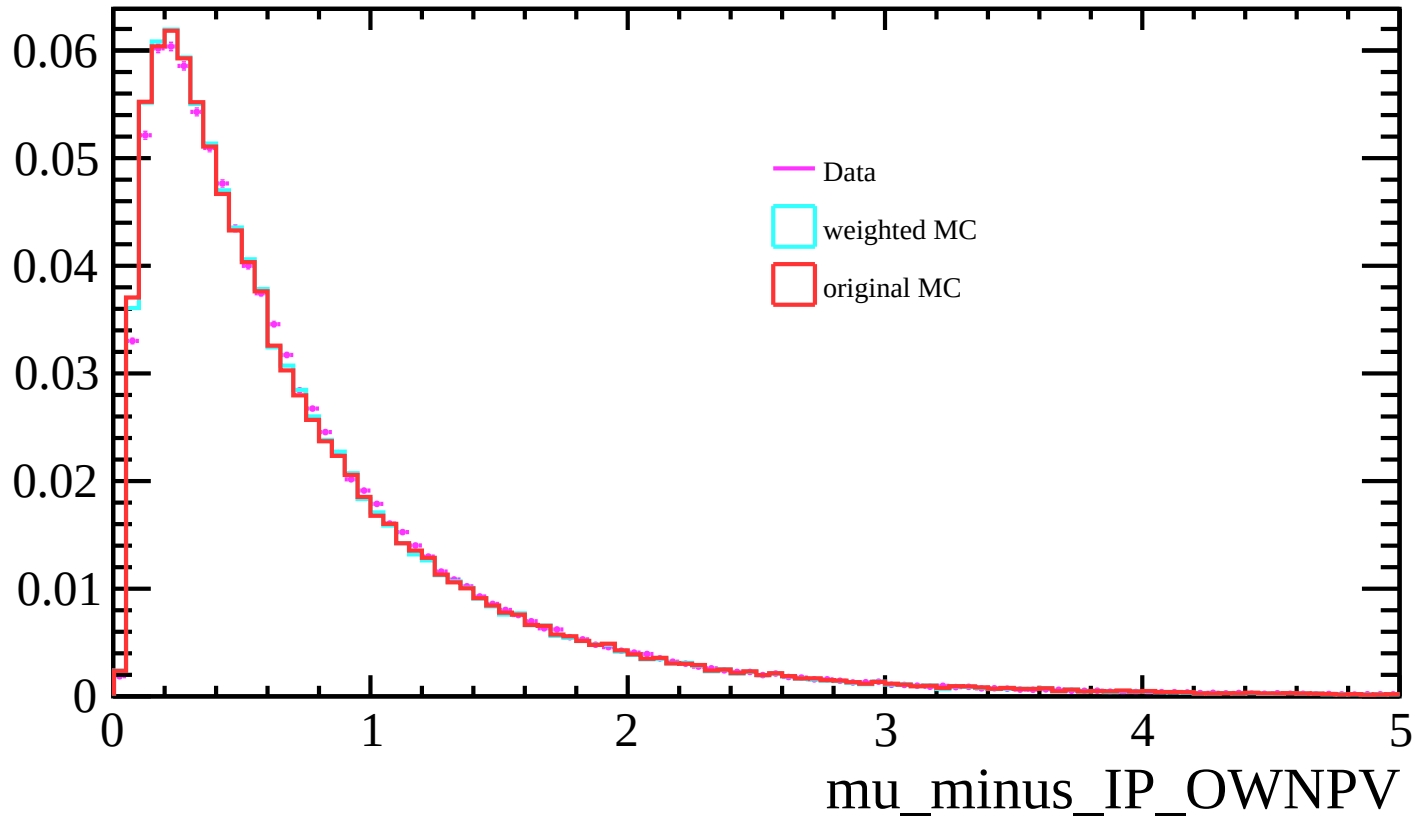

0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
B0_relinfo_MU_SLL_ISO_1

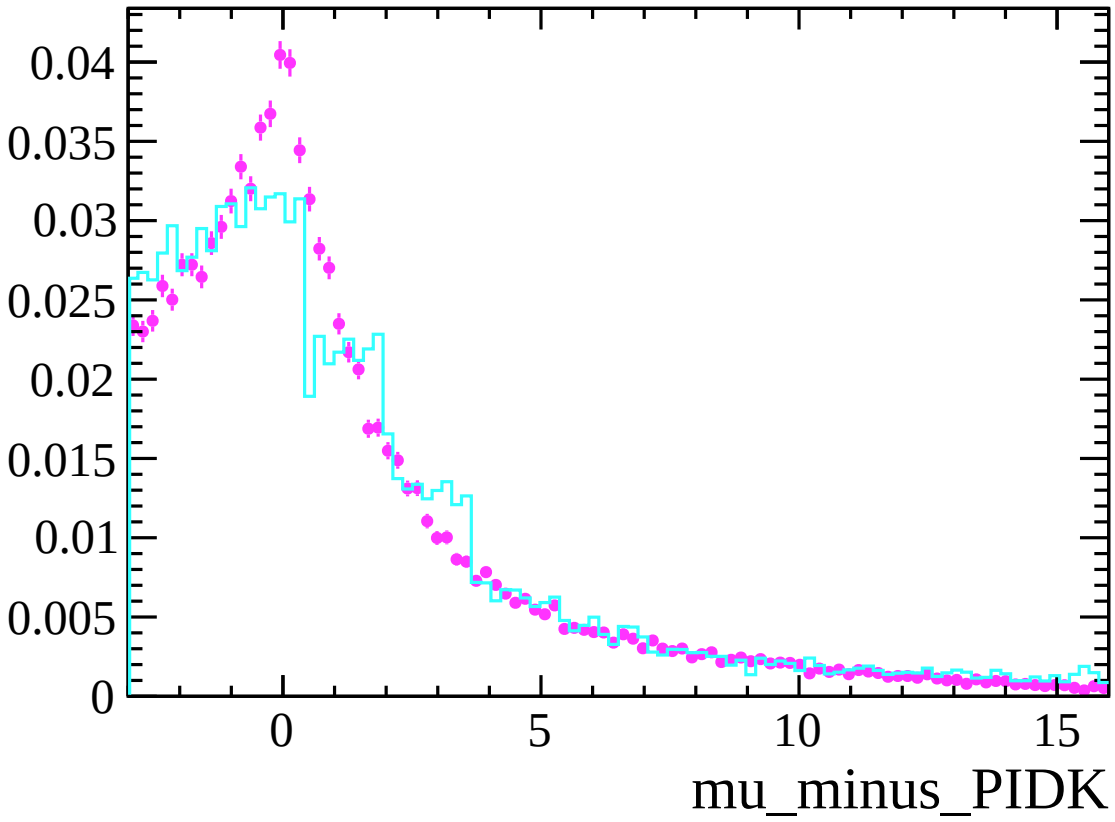

0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
B0_relinfo_MU_SLL_ISO_2

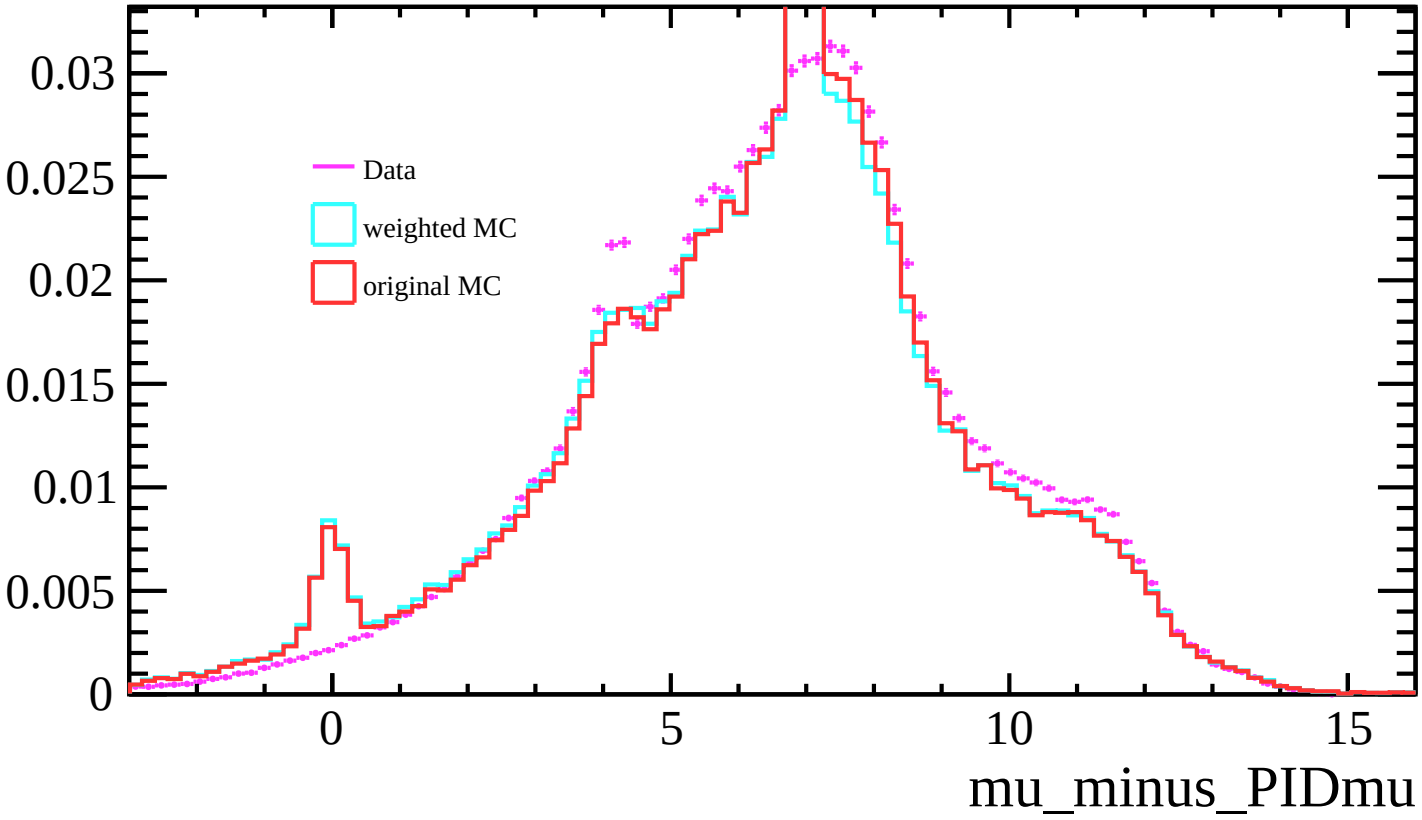

0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
1
2
3
4
5
K_IP_OWNPV
Data
weighted MC
original MC

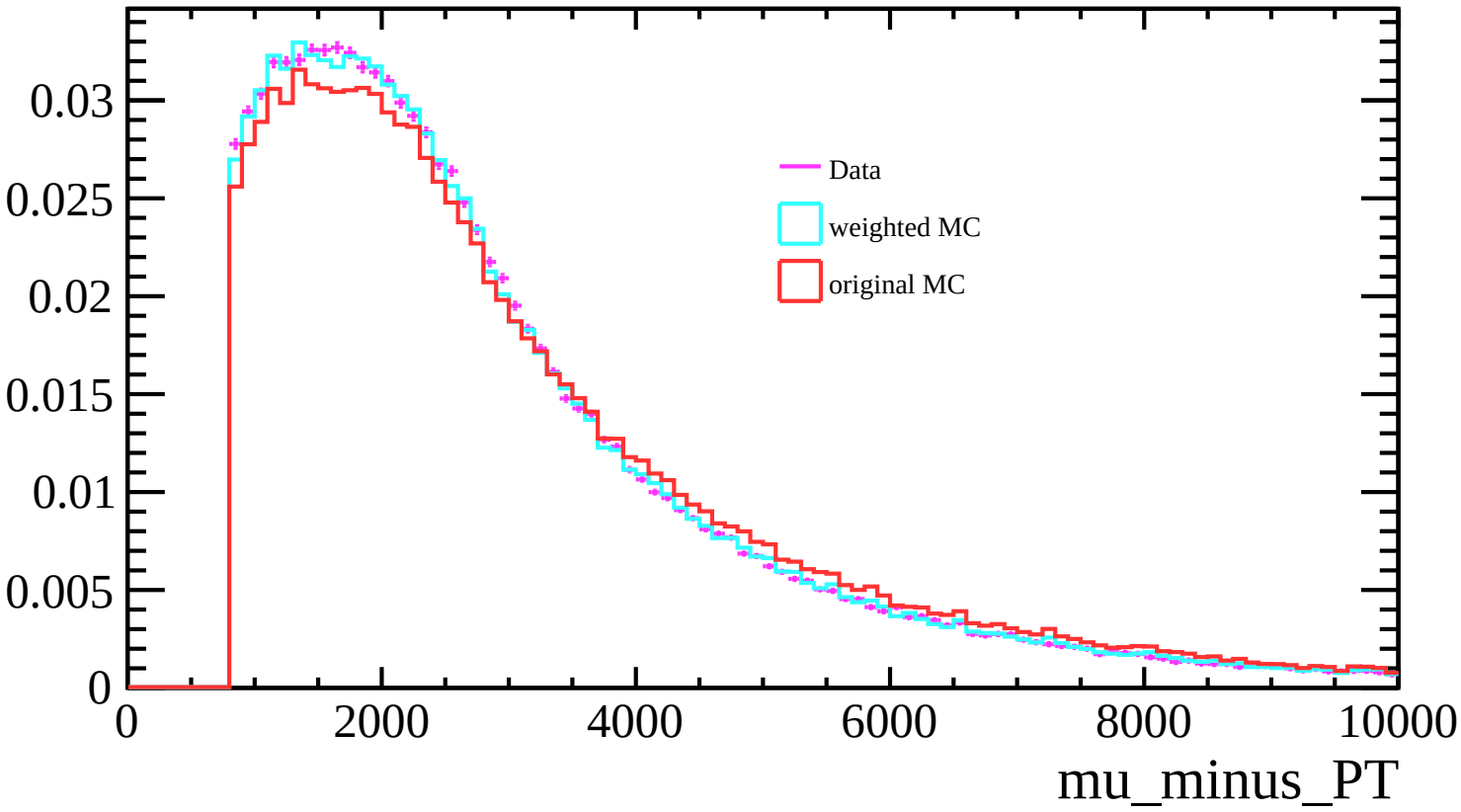

0.024
0.022
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
50
100
K_PIDK
Data
weighted MC
original MC

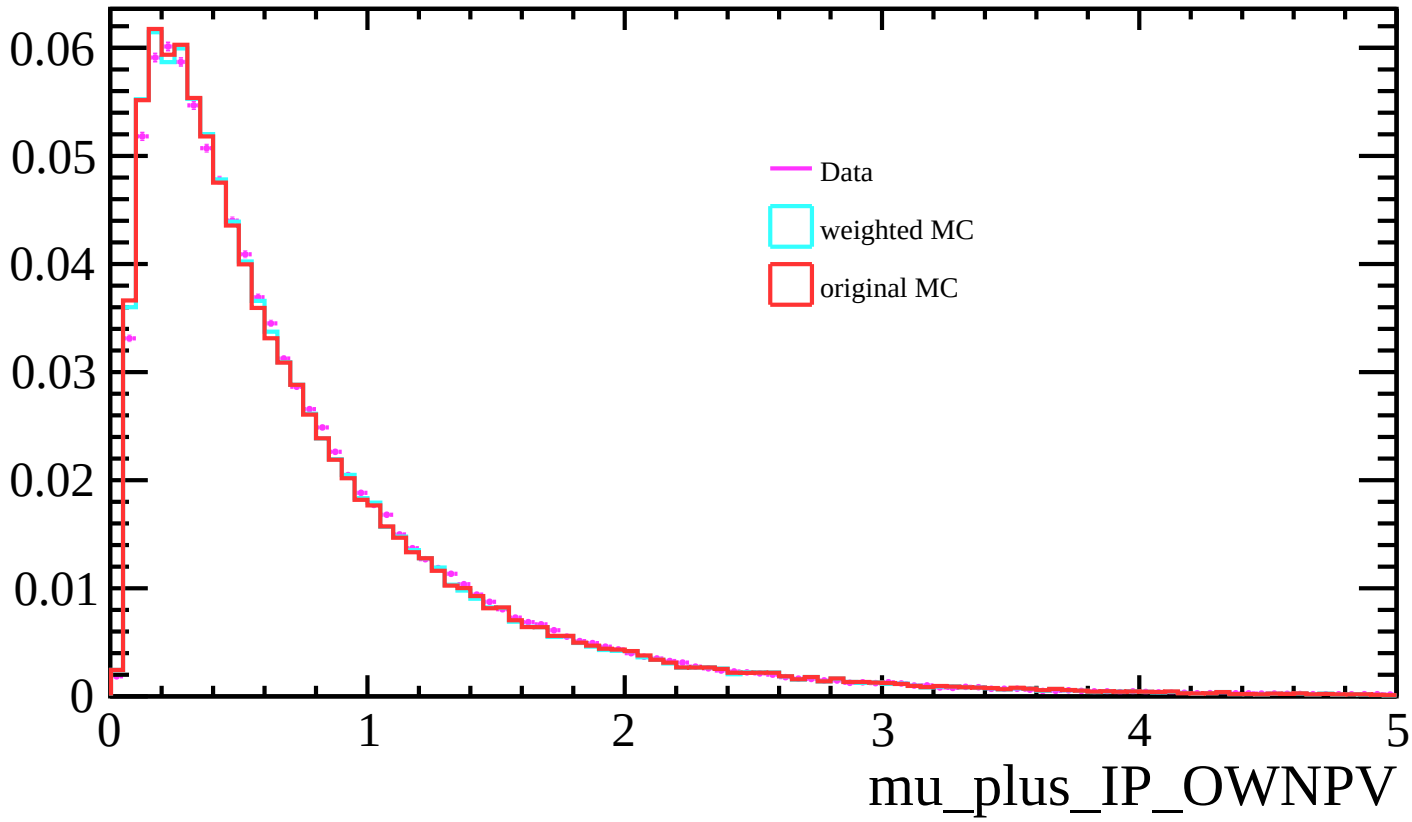

0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
2000
4000
6000
8000
10000
K_PT
Data
weighted MC
original MC

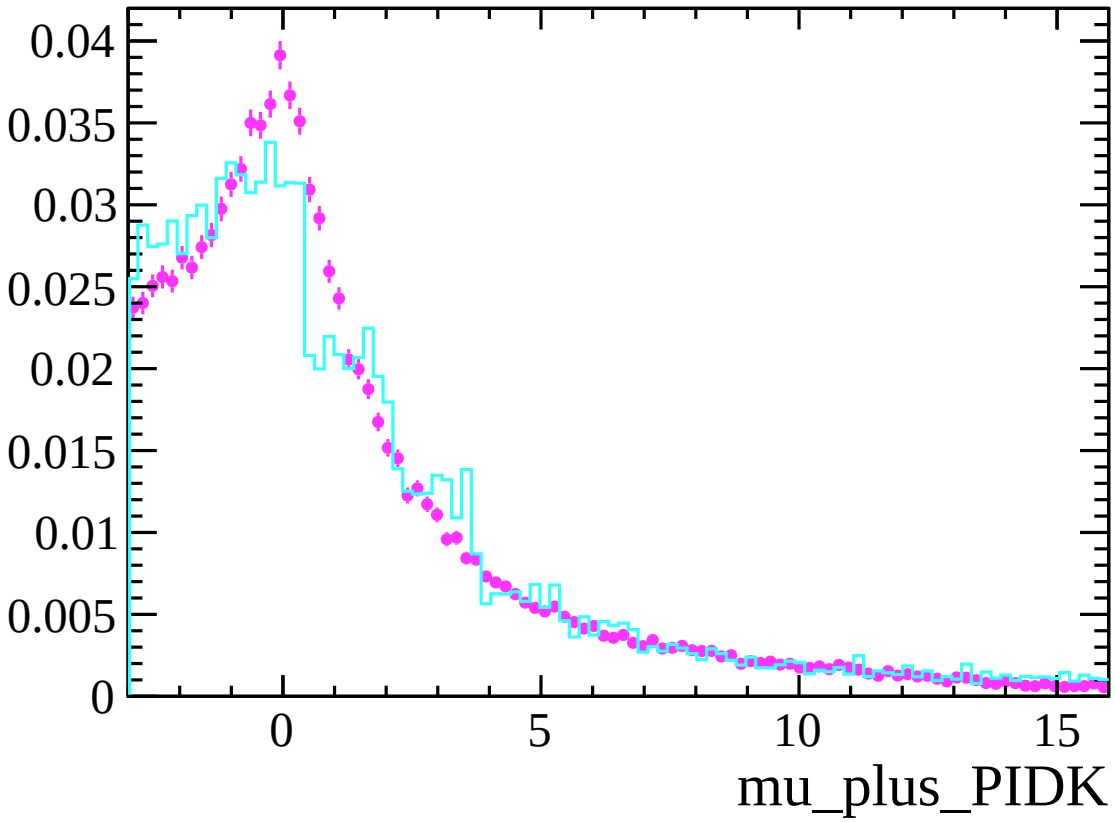

0.06
0.05
0.04
0.03
0.02
0.01
0
0
1
2
3
4
5
Data
weighted MC
original MC
mu_minus_IP_OWNPV

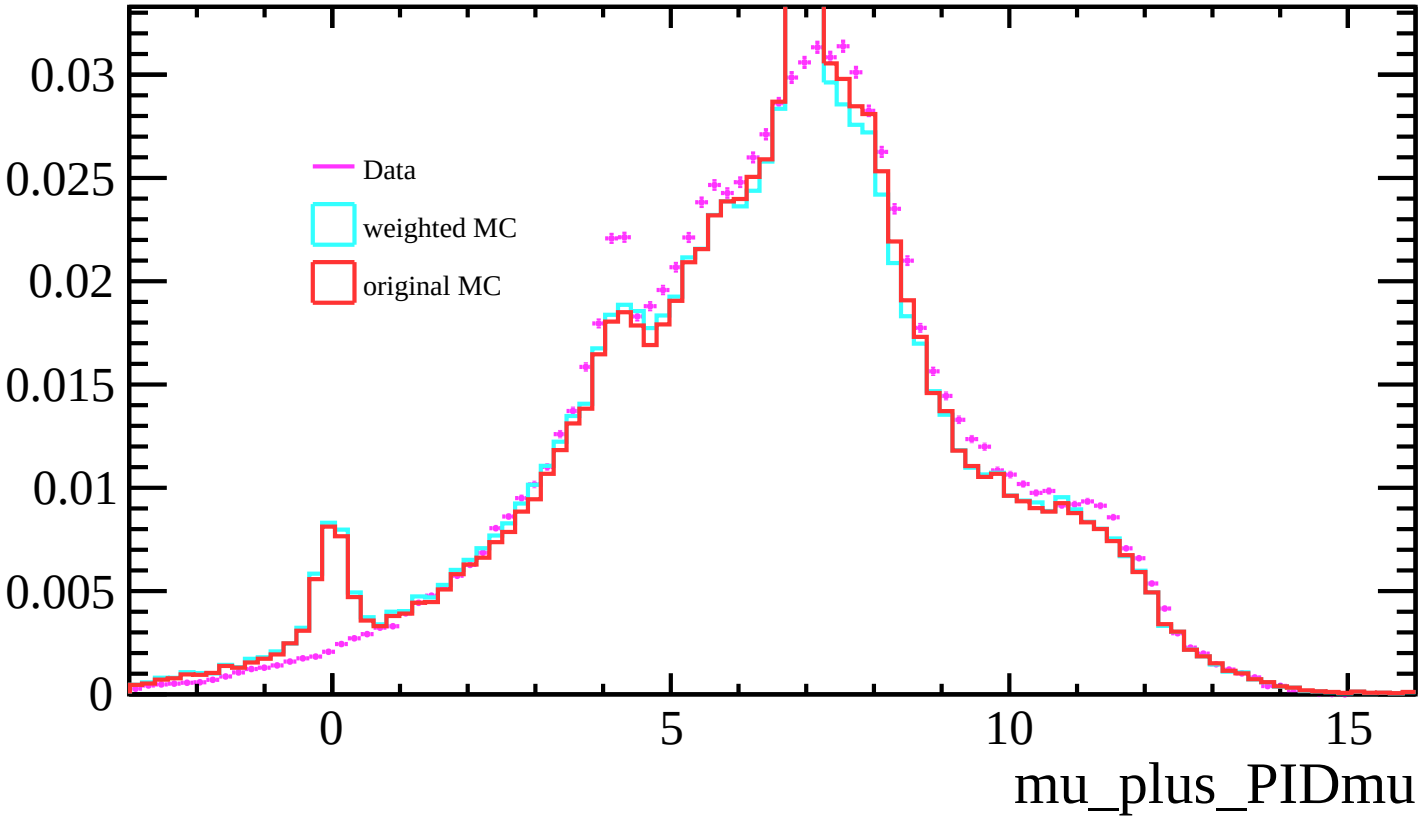

0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
5
10
15
mu_minus_PIDK

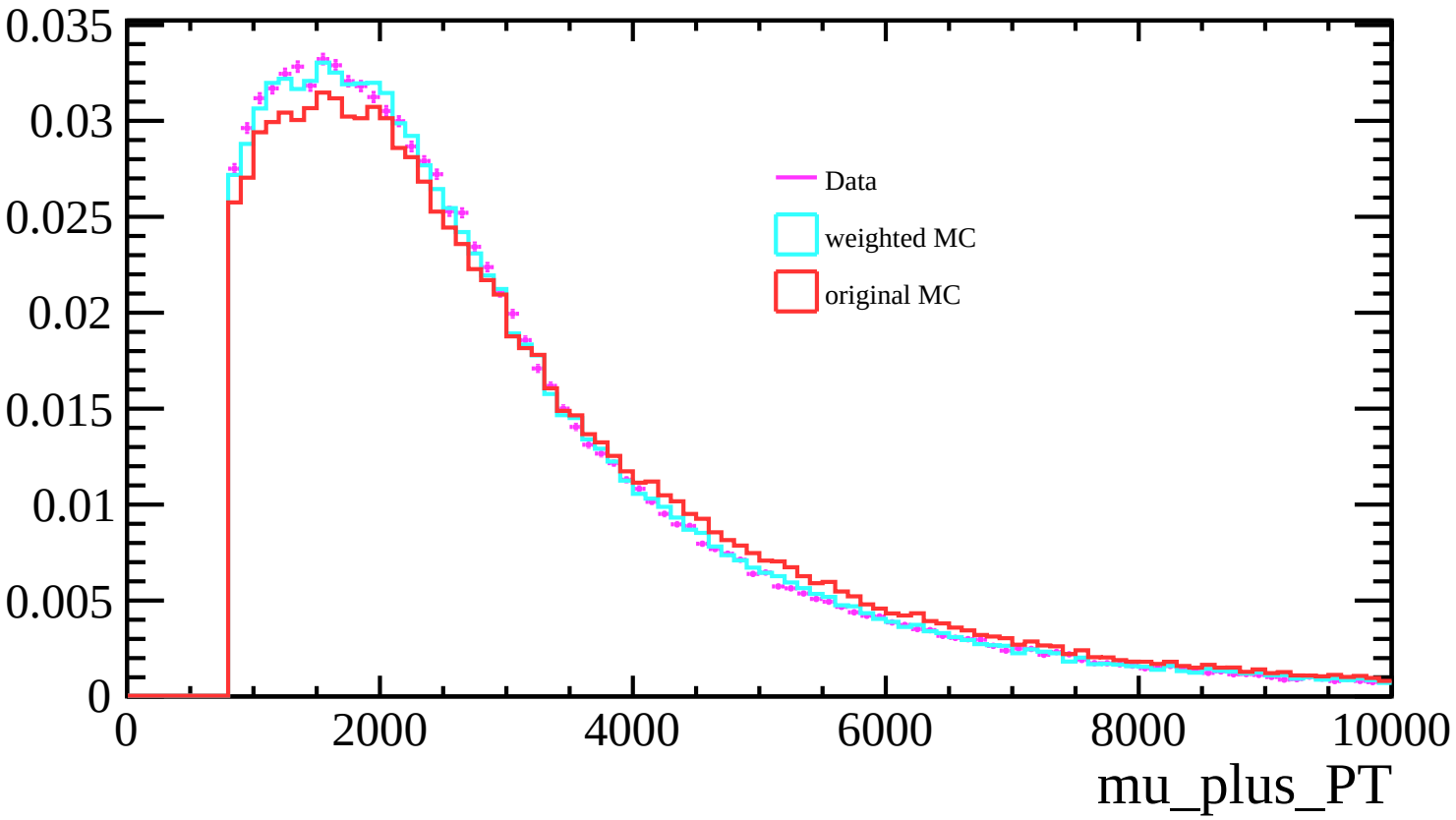

0.03
0.025
0.02
0.015
0.01
0.005
0
Data
weighted MC
original MC
0
5
10
15
mu_minus_PIDmu

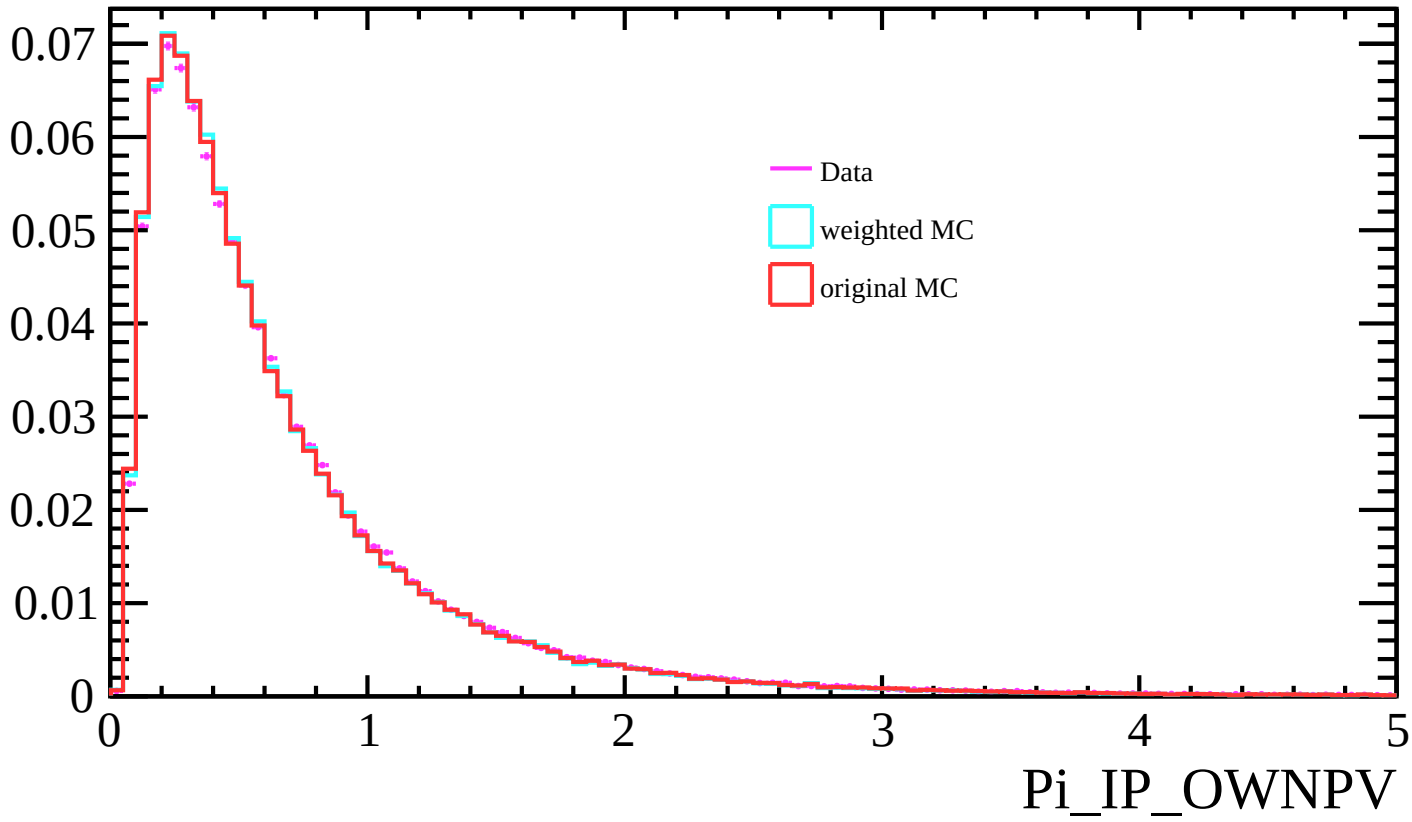

0.03
0.025
0.02
0.015
0.01
0.005
0
0
2000
4000
6000
8000
10000
mu_minus_PT
Data
weighted MC
original MC

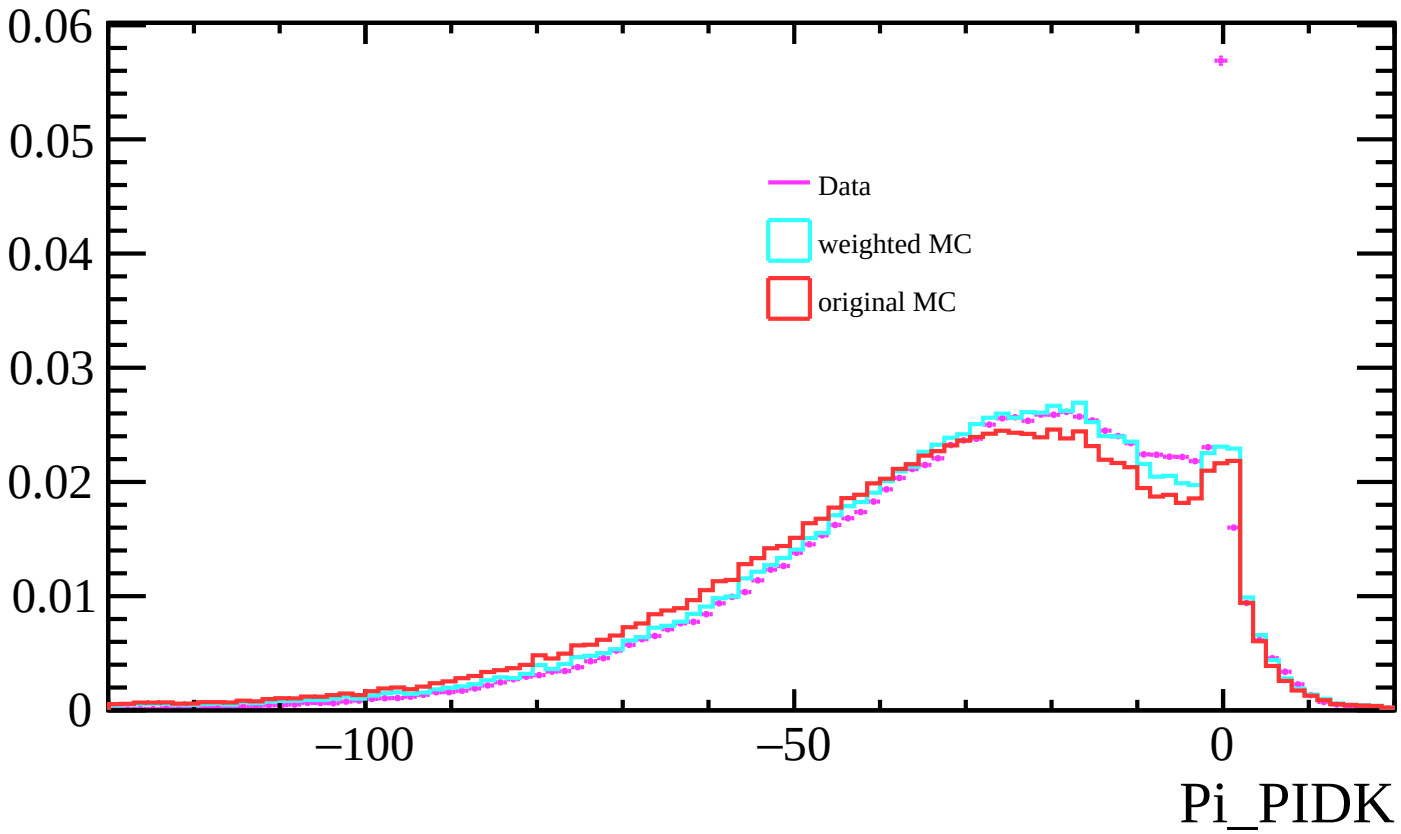

0.06
0.05
0.04
0.03
0.02
0.01
0
0
1
2
3
4
5
Data
weighted MC
original MC
mu_plus_IP_OWNPV

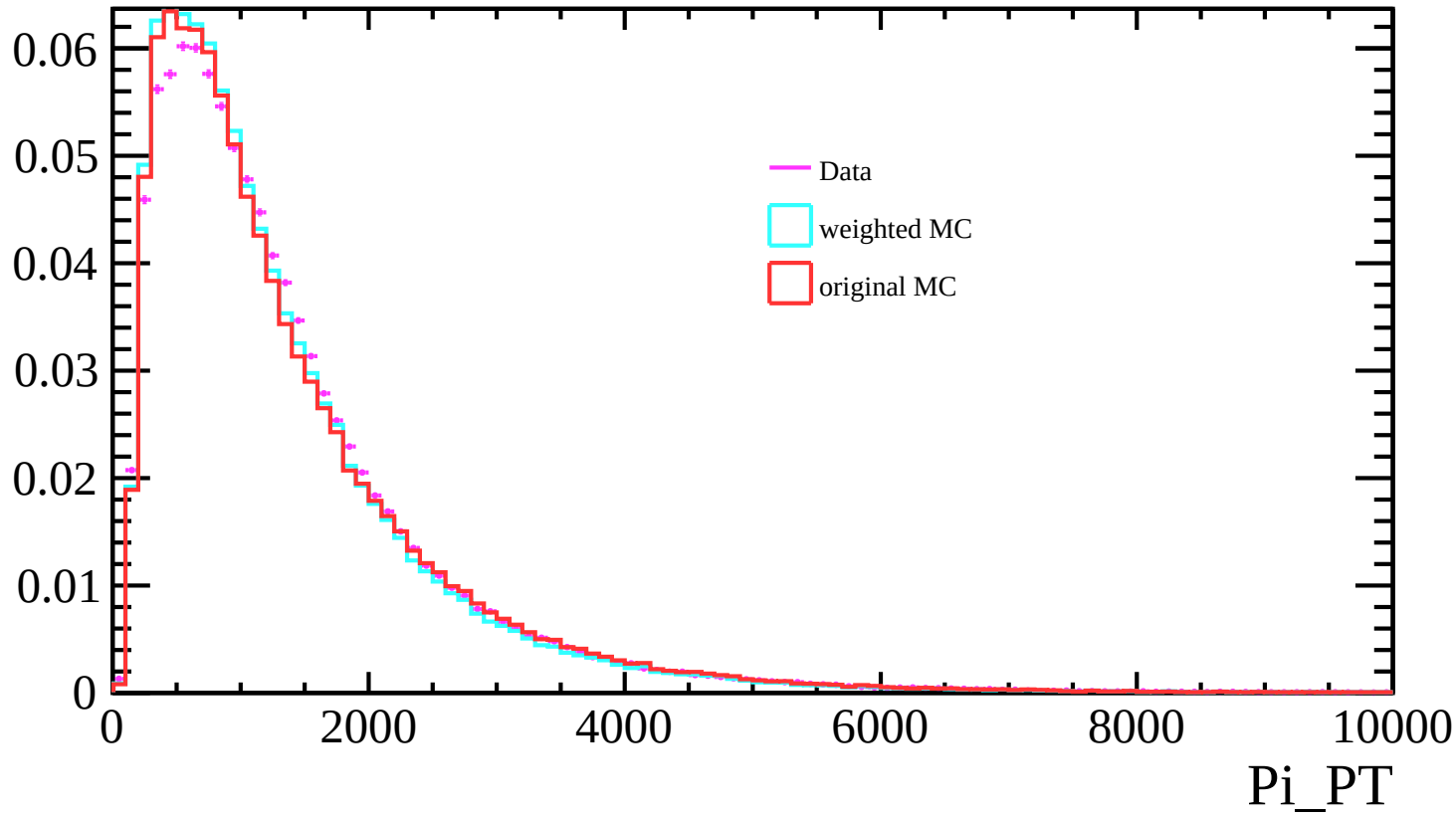

0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
5
10
15
mu_plus_PIDK


0.03
0.025
0.02
0.015
0.01
0.005
0
Data
weighted MC
original MC
0
5
10
15
mu_plus_PIDmu


0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
2000
4000
6000
8000
10000
mu_plus_PT
Data
weighted MC
original MC


0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
1
2
3
4
5
Pi_IP_OWNPV
Data
weighted MC
original MC


0.06
0.05
0.04
0.03
0.02
0.01
0
−100
−50
0
Pi_PIDK
Data
weighted MC
original MC


0.06
0.05
0.04
0.03
0.02
0.01
0
0
2000
4000
6000
8000
10000
Pi_PT
Data
weighted MC
original MC