The Espresso Performance Monitor (EPM) package

Vincenzo Battista[1]

[1] École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

A&S Week, Flavour Tagging Session, 24/01/2018

Outline

Introduction

The Espresso Performance Monitor (EPM) package

Developed and (previously) maintained by Jack Wimberley (who left HEP).

Useful to:

- perform flavour tagging calibrations on both neutral and charged modes;
- evaluate tagging performance (ε_{tag}, ε_{eff} etc...);
- combine taggers and add calibrated mistag (ω) in a tuple;
- produce publication-quality plots.

Motivations:

- standardise and simplify calibration process for analysts;
- new paradigm: a calibration for each analysis (no more global calibration from FT group!)

Theoretical framework described in:
J. Wimberley, *Calibrating flavor tagging algorithms with binomial regression*, LHCb-INT-2017-002 (link)
A. Agresti, *Categorical Data Analysis*, John Wiley & Sons, Apr. 2014

Talk at the Statistics WG meeting: link

It's being documented also in the upcoming FT performance paper.

Twiki pages (link) with wide documentation and extensive details.

Flavour Tagging calibration: the "traditional" methods

Charged decay modes:

- E.g. $B^+ \to J/\psi K^+$ and $B^+ \to D^0\pi^+$ for B^0, $B^*_{s2} \to B^+K^-$ for B^0_s.
- Count true mistag fraction $\langle\omega_i\rangle$ in bins of η, then do linear fit.

$\Rightarrow$ systematics due to binning choice.

Neutral decay modes:

- E.g. $B^0 \to J/\psi K^*$ for B^0, $B^0_s \to D_s\pi$ for B^0_s.
- Full time-dependent analysis with η as dimension, fit directly for $p_{0,1}$ and $\Delta p_{0,1}$.
- Alternative: time-dependent analysis in bins of η to obtain $\langle\omega_i\rangle$, then do linear fit.

$\Rightarrow$ systematics due to time-dependent analyses (resolution, acceptance, production asymmetries...) and eventually binning choice.

We have a continuous predictor (ω) and a binary output (right tag, wrong tag)
$\Rightarrow$ we better use a *binomial regression*

Binomial regression for charged modes

Assume a *charged* decay mode (no oscillations).

Calibrated mistag ω is a function of (uncalibrated) η, i.e. $\omega = \omega(\eta, \theta)$.

θ is the set of *calibration parameters* to fit (aka p_i and Δp_i).

Probability that decay flavour = production flavour (i.e. correct tag): $\pi = \bar{\omega} = 1 - \omega$.

In terms of dilution $D = 2\bar{\omega} - 1$, we have $\pi = \frac{1+D}{2}$.

We can maximise the following log-likelihood:

$$L(\theta) = \sum_k \begin{cases} \pi_k(\theta) & \text{right tag} \\ 1 - \pi_k(\theta) & \text{wrong tag} \end{cases}$$

$\Rightarrow$ no binning effects!
$\Rightarrow$ using maximum likelihood estimator instead of least-squares.

Binomial regression for neutral modes

Probability π in the likelihood needs to take *oscillations* into account:

- Probability that tagged flavour = production flavour (right tag): $\bar{\omega} = 1 - \omega$.
- Probability that production flavour = decay flavour (no oscillation):

$$p^{nomix} = \frac{1}{2}\left(1 + \cos \Delta m t \operatorname{sech} \frac{1}{2}\Delta\Gamma t\right) = \frac{1 + D^{nomix}}{2}$$

$\Rightarrow$ Probability that decay flavour = production flavour:

$$\pi = \bar{\omega} p^{nomix} + (1 - \bar{\omega})(1 - p^{nomix}) = \frac{1 + DD^{nomix}}{2}.$$

Dilution due to *decay-time resolution*:

$$p^{nomix} \to \int p^{nomix}(t')\mathcal{R}(t,t')dt = \frac{1 + D^{res}D^{nomix}}{2} \Rightarrow \pi = \frac{1 + DD^{res}D^{nomix}}{2}$$

Other effects (acceptance, production asymmetries . . .) are found to be negligible
$\Rightarrow$ see [LHCb-INT-2017-002] for an extensive treatment.

Calibration models

Simple linear model (so far the standard for all tagged analyses):

$$\omega(\eta) = \left(p_0 \pm \frac{\Delta p_0}{2}\right) + \left(p_1 \pm \frac{\Delta p_1}{2}\right)(\eta - \langle\eta\rangle)\,.$$

EPM provides *Generalised Linear Models* (GLMs):

$$\omega(\eta) = g\left[g^{-1}(\eta) + \sum_i \theta_i f_i(\eta)\right]$$

g is the link function:

- inverse of CDFs (map $[-\infty, +\infty]$ to $[0, 1]$);
- EPM provides `LOGIT`, `PROBIT` or `CAUCHIT` (named after the parent PDF);
- from v0.7, also modified links to have ω in $[0, 0.5]$ (`RLOGIT`, `RPROBIT` or `RCAUCHIT`).
- the default is no link, i.e. the identity $g = 1$ (`MISTAG`)

$f_i(\eta)$ are basis functions:

- simple polynomials (`POLY`);
- more refined spline functions (`NSPLINE`, `BSPLINE`)
- they are orthogonal to minimise correlations among calibration parameters θ_i's.

Install, compile and run

To build on lxplus:

```
git clone ssh://git@gitlab.cern.ch:7999/lhcb-ft/EspressoPerformanceMonitor.git
cd EspressoPerformanceMonitor
git checkout v0.7
git submodule update --init --recursive
source setenv.sh
cmake .
make
```

The main executable you need is `SimpleEvaluator`, which takes an option file as argument:

```
./<path-to>EspressoPerformanceMonitor/bin/SimpleEvaluator my_options.py
```

Example: $B^+ \to D^0\pi^+$ calibration

$B^+ \to D^0\pi^+$ calibration: introduction

Option file in `options/example_BuD0piComplete.py`

In this example, $B^+ \to D^0\pi^+$ is divided in two subsamples.
First half:

- calibrate OSComb and SSPionBDT using a GLM model;
- save calibrations (to XML files).

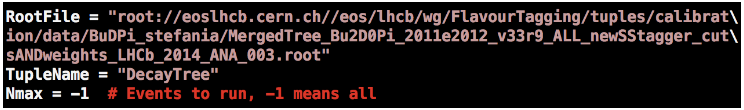
Second half:

- apply calibrations obtained before on OSComb and SSPionBDT, and evaluate performance;
- combine calibrated OSComb and SSPionBDT;
- calibrate combination and evaluate performance.

In total, the EPM has to be run twice (once for each subsample).

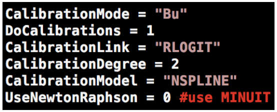
$B^+ \to D^0\pi^+$ calibration on first 1/2 of data

Input root file:



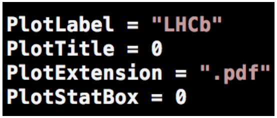
```
RootFile = "root://eoslhcb.cern.ch//eos/lhcb/wg/FlavourTagging/tuples/calibrat\
ion/data/BuDPi_stefania/MergedTree_Bu2D0Pi_2011e2012_v33r9_ALL_newSStagger_cut\
sANDweights_LHCb_2014_ANA_003.root"
TupleName = "DecayTree"
Nmax = -1  # Events to run, -1 means all
```

Calibration model:



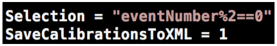
```
CalibrationMode = "Bu"
DoCalibrations = 1
CalibrationLink = "RLOGIT"
CalibrationDegree = 2
CalibrationModel = "NSPLINE"
UseNewtonRaphson = 0 #use MINUIT
```

Plotting settings:



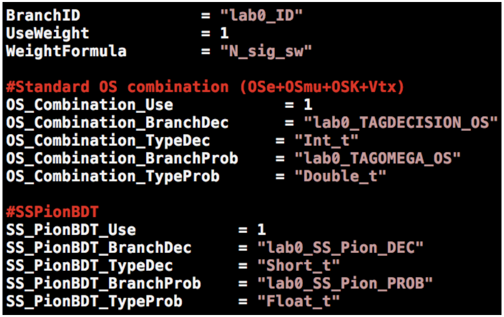
```
PlotLabel = "LHCb"
PlotTitle = 0
PlotExtension = ".pdf"
PlotStatBox = 0
```

Split data and save calibration:



```
Selection = "eventNumber%2==0"
SaveCalibrationsToXML = 1
```

Tagger branches, ID, weights, and combination:

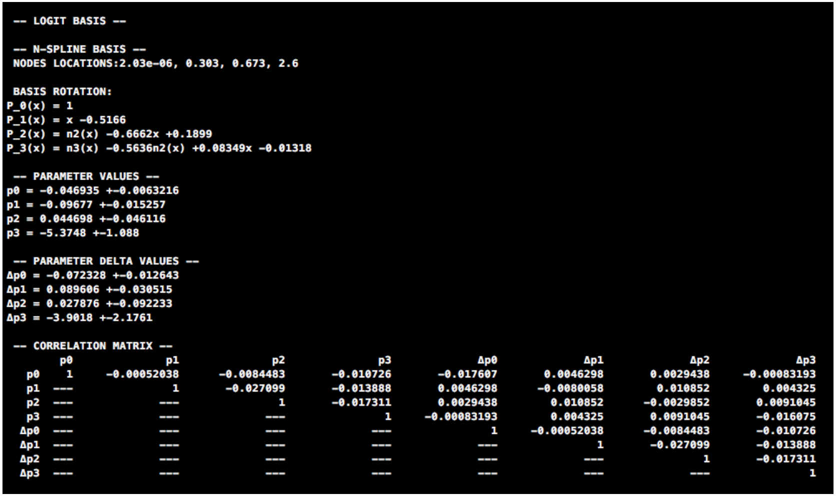


```
BranchID                = "lab0_ID"
UseWeight               = 1
WeightFormula           = "N_sig_sw"

#Standard OS combination (OSe+OSmu+OSK+Vtx)
OS_Combination_Use                = 1
OS_Combination_BranchDec          = "lab0_TAGDECISION_OS"
OS_Combination_TypeDec          = "Int_t"
OS_Combination_BranchProb       = "lab0_TAGOMEGA_OS"
OS_Combination_TypeProb         = "Double_t"

#SSPionBDT
SS_PionBDT_Use              = 1
SS_PionBDT_BranchDec        = "lab0_SS_Pion_DEC"
SS_PionBDT_TypeDec          = "Short_t"
SS_PionBDT_BranchProb       = "lab0_SS_Pion_PROB"
SS_PionBDT_TypeProb         = "Float_t"
```

$B^+ \to D^0\pi^+$ calibration output

EPM returns calibrated parameters and correlations.
Everything is saved in the output XML file.

Example: OSComb.

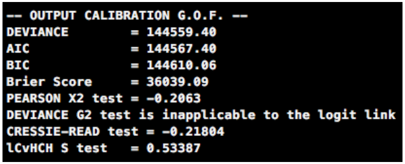


```
 -- LOGIT BASIS --

 -- N-SPLINE BASIS --
 NODES LOCATIONS:2.03e-06, 0.303, 0.673, 2.6

 BASIS ROTATION:
P_0(x) = 1
P_1(x) = x -0.5166
P_2(x) = n2(x) -0.6662x +0.1899
P_3(x) = n3(x) -0.5636n2(x) +0.08349x -0.01318

 -- PARAMETER VALUES --
p0 = -0.046935 +-0.0063216
p1 = -0.09677 +-0.015257
p2 = 0.044698 +-0.046116
p3 = -5.3748 +-1.088

 -- PARAMETER DELTA VALUES --
Δp0 = -0.072328 +-0.012643
Δp1 = 0.089606 +-0.030515
Δp2 = 0.027876 +-0.092233
Δp3 = -3.9018 +-2.1761

 -- CORRELATION MATRIX --
       p0           p1           p2           p3          Δp0          Δp1          Δp2          Δp3
  p0    1  -0.00052038   -0.0084483    -0.010726    -0.017607    0.0046298    0.0029438  -0.00083193
  p1  ---            1    -0.027099    -0.013888    0.0046298   -0.0080058     0.010852     0.004325
  p2  ---          ---            1    -0.017311    0.0029438     0.010852   -0.0029852    0.0091045
  p3  ---          ---          ---            1  -0.00083193     0.004325    0.0091045    -0.016075
 Δp0  ---          ---          ---          ---            1  -0.00052038   -0.0084483    -0.010726
 Δp1  ---          ---          ---          ---          ---            1    -0.027099    -0.013888
 Δp2  ---          ---          ---          ---          ---          ---            1    -0.017311
 Δp3  ---          ---          ---          ---          ---          ---          ---            1
```

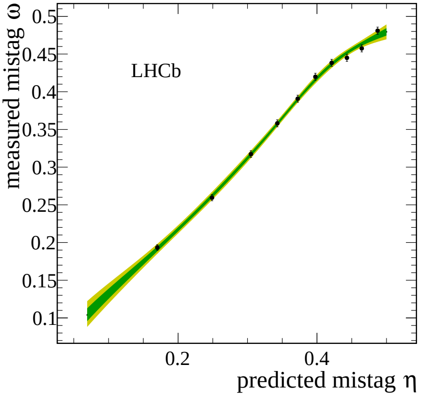
$B^+ \to D^0\pi^+$ calibration: GOF

EPM provides different GOF tests for each calibration.

Example: OSComb.



```
-- OUTPUT CALIBRATION G.O.F. --
DEVIANCE          = 144559.40
AIC               = 144567.40
BIC               = 144610.06
Brier Score       = 36039.09
PEARSON X2 test = -0.2063
DEVIANCE G2 test is inapplicable to the logit link
CRESSIE-READ test = -0.21804
lCvHCH S test     = 0.53387
```

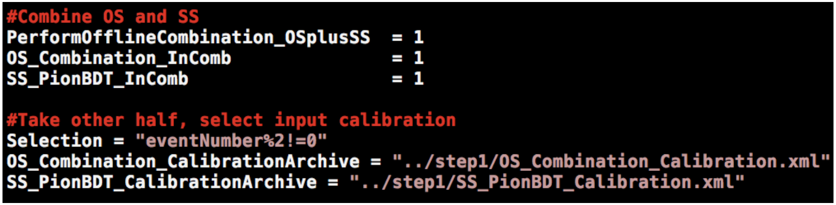


Pearson, G2, Cressie-Read and lCvHCH tests output is *normally distributed*
$\Rightarrow$ expect 0 for perfect fit, 1 for 1 σ discrepancy etc...

More details in Twiki and LHCb-INT-2017-002.

$B^+ \to D^0\pi^+$ calibration applied on second 1/2 of data

Select other half of data, retrieve calibrations from previous step, combine calibrated taggers:



```
#Combine OS and SS
PerformOfflineCombination_OSplusSS  = 1
OS_Combination_InComb               = 1
SS_PionBDT_InComb                   = 1

#Take other half, select input calibration
Selection = "eventNumber%2!=0"
OS_Combination_CalibrationArchive = "../step1/OS_Combination_Calibration.xml"
SS_PionBDT_CalibrationArchive = "../step1/SS_PionBDT_Calibration.xml"
```

Output: now input calibration is applied on second 1/2 of data, then the performance is evaluated. From `EspressoPerformanceTable.tex`:

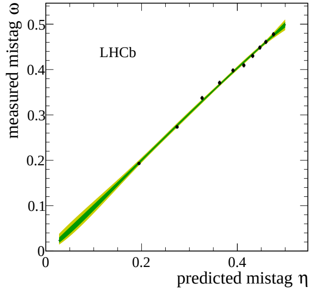
Tagger	$\varepsilon_{\rm tag}$ (%)	ω (%)	$\epsilon_{\rm eff}$ (%)
SS π BDT	20.688 ± 0.072	38.883 ± 0.023(stat) ± 0.188(cal)	1.023 ± 0.006(stat) ± 0.035(cal)
OS Comb	36.906 ± 0.086	34.671 ± 0.025(stat) ± 0.132(cal)	3.469 ± 0.014(stat) ± 0.060(cal)

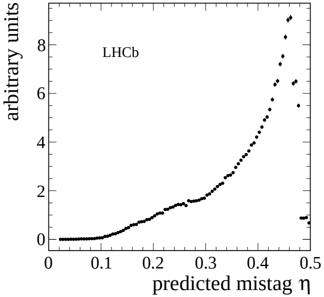
Finally, OSComb+SSPionBDT is calibrated, and the performance is computed. From `EspressoCalibratedPerformanceTable.tex`:

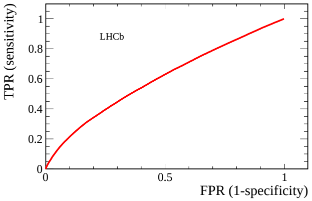
Tagger	$\varepsilon_{\rm tag}$ (%)	ω (%)	$\epsilon_{\rm eff}$ (%)
Comb	49.681 ± 0.089	35.220 ± 0.022(stat) ± 0.114(cal)	4.341 ± 0.015(stat) ± 0.067(cal)

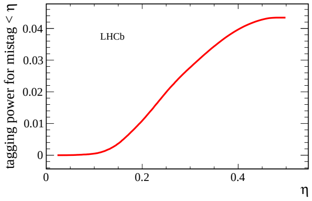
$B^+ \to D^0\pi^+$: examples of plots

Plots from OSComb+SSPionBDT combination.









Example: $B_s^0 \rightarrow D_s^- \pi^+$ calibration

$B^0 \rightarrow D_s^- \pi^+$ calibration: introduction

Option file in `options/example_BsDspi.py`

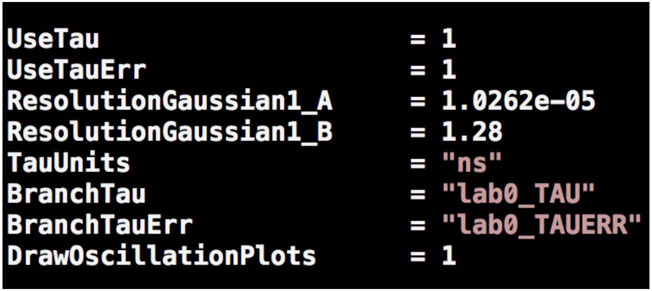
In this example, the SSKaonNNet is calibrated on $B^0 \rightarrow D_s^- \pi^+$ with a standard linear function.

A per-event time resolution model is used.
The gaussian width is obtained via a *linear calibration* of the decay-time error δ:

$$\sigma = A + \delta B$$

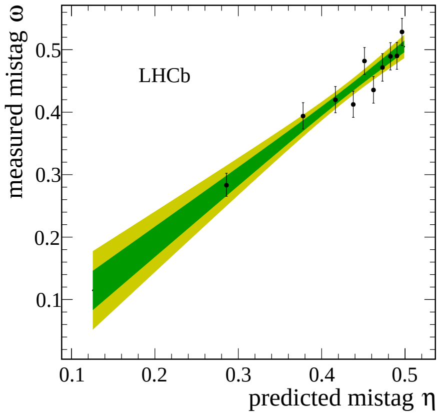
Moreover, the folded time-dependent asymmetry corrected for the dilution is plotted
$\Rightarrow$ should be $\cos(\Delta mt)$ within the uncertainties.

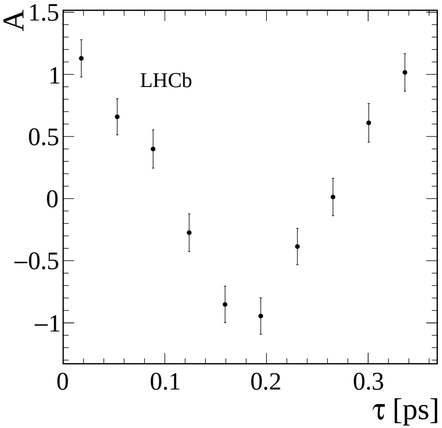
Relevant options:



```
UseTau                      = 1
UseTauErr                   = 1
ResolutionGaussian1_A       = 1.0262e-05
ResolutionGaussian1_B       = 1.28
TauUnits                    = "ns"
BranchTau                   = "lab0_TAU"
BranchTauErr                = "lab0_TAUERR"
DrawOscillationPlots        = 1
```

$B^0 \to D_s^- \pi^+$ calibration: some output plots





Example: use EPM models in your decay-time fit

Introduction

Best practice in time-dependent analyses:

- find calibration $\omega(\eta)$ on control channel;
- include η as dimension in the fit;
- float calibration coefficients $(p_i, \Delta p_i)$ in the fit, apply Gaussian constraints.

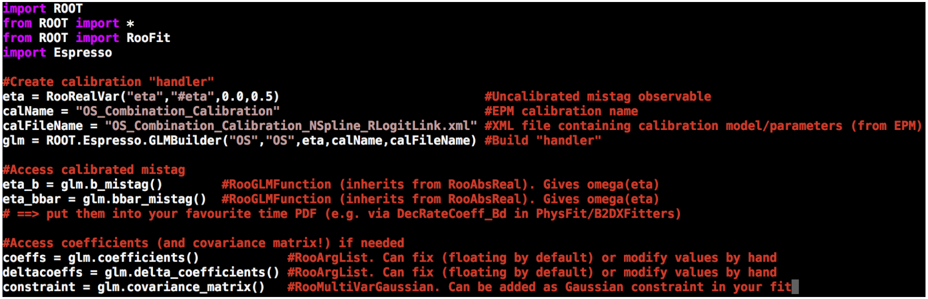
$\Rightarrow$ uncertainties properly propagated to CPV/oscillation measurement!

It's now possible to use any GLM model from EPM in any RooFit-based fitter within Urania.

```
lb-dev Urania/v7r0
cd UraniaDev_v7r0
git lb-use Urania
git lb-checkout Urania/master FT/Espresso
git lb-checkout Urania/master <your-analysis-code>
```

Small example

How your RooFit script would look like:



```
import ROOT
from ROOT import *
from ROOT import RooFit
import Espresso

#Create calibration "handler"
eta = RooRealVar("eta","#eta",0.0,0.5)                          #Uncalibrated mistag observable
calName = "OS_Combination_Calibration"                          #EPM calibration name
calFileName = "OS_Combination_Calibration_NSpline_RLogitLink.xml" #XML file containing calibration model/parameters (from EPM)
glm = ROOT.Espresso.GLMBuilder("OS","OS",eta,calName,calFileName) #Build "handler"

#Access calibrated mistag
eta_b = glm.b_mistag()          #RooGLMFunction (inherits from RooAbsReal). Gives omega(eta)
eta_bbar = glm.bbar_mistag()  #RooGLMFunction (inherits from RooAbsReal). Gives omega(eta)
# ==> put them into your favourite time PDF (e.g. via DecRateCoeff_Bd in PhysFit/B2DXFitters)

#Access coefficients (and covariance matrix!) if needed
coeffs = glm.coefficients()             #RooArgList. Can fix (floating by default) or modify values by hand
deltacoeffs = glm.delta_coefficients() #RooArgList. Can fix (floating by default) or modify values by hand
constraint = glm.covariance_matrix()   #RooMultiVarGaussian. Can be added as Gaussian constraint in your fit
```

`eta_b` (ω^B) and `eta_bbar` ($\omega^{\bar{B}}$) depend on `eta` (η) via the chosen GLM from EPM.

You can build your time PDF (`RooAbsPdf`) by including them
$\Rightarrow$ the calibration coefficients will appear as parameters in the fit.

Examples are in `PhysFit/B2DXFitters`.

Conclusions and summary

Conclusions and summary

The EPM became the standard tool for calibration and perfomance in the FT group.

We encourage analyst to use it! In case of questions/bugs/requests:

- FT WG mailing list: `lhcb-phys-flavour-tagging@cern.ch`
- EPM JIRA group (LHCBESPCAL)

Didn't have time to show all details.
Particularly relevant is the toy generator to check GOF and calibration parameters.
$\Rightarrow$ dedicated Twiki page here