Unfolding Tutorial

Introduction:

High energy physics applies a variety of techniques to obtain estimators for a probability function even if no parametric form is available and if the data are subject to additional random fluctuations due to limited resolution. These procedures are usually called *unfolding*.
In this tutorial we want to compare the performance of two methods frequently used in high-energy physics: Bin-by-Bin corrections and Bayesian unfolding. By unfolding various Toy MonteCarlo distributions, we want to study the advantages and limitations of each method.

The Code Framework:

Due to the limited time available for the tutorial, we have provided a code structure which allows you to:
- Create Toy MonteCarlo (exponential, Breit-Wigner, double-Breit-Wigner, ..)
  - Training distributions
  - Distributions simulating the measurement
- Simulate a detector response with a limited resolution and acceptance
- Perform Bayesian Unfolding
- Perform a BinbyBin Unfolding
- Compare the original and the unfolded Distributions

The whole structure is of course flexible enough to allow for creative modifications from your side.

Log in with the account "*skin07*" and the password provided to you.
Go to the folder *StatisticSchool/UnfoldingTutorial*. The following subfolders are available:

1) ToyMC:

   ToyMC.C fills Truth and smeared histograms, representing training samples (*hTruth, hReco*) the migration matrix (*hMatrix*) and the data (*hDataT, hData*).
   In a real situation, hDataT is of course not available but here it provides a useful consistency check. The following functions are already provided: Exponential, Breit-Wigner, Double Breit-Wigner. The function *reco(x)* is used to provide a detector simulation, which is comparable for all applications, in particular for the simulated MonteCarlo and for the

simulated Data.
The Script is loaded in root with the command *.L ToyMC.C* . This makes all functions available for the user. Output is stored in histogram files.
The output distributions can be checked conveniently with the script *Compare.C*. which has to be adapted to read the correct input histogram file. The histogram file has to be copied into the Input directory of the Unfolding Packages and linked properly.

2) BayesUnfolding:

The Program *Bayes* provides the user interface to the unfolding program *bayes_unfolding*. The Programs rely on libraries of the software *Octave* for the matrix calculations. The software is compiled by the command *make*.
The program *Bayes* is started with an integer argument which corresponds to the number of iterations.
The user interface reads *hMatrix, hTruth and hData* from the histogramfile *Input.root* in the folder *Input*, unfolds the data distribution and writes input distribution and the unfolded distribution *hUnfold* into the file *histos.root* in the Output folder.
In this folder we can use another version of Compare.C in order to conveniently compare the distributions.

3) BinbyBinUnfolding

The Program *BinbyBin* provides the user interface. From the file *Inpot.root* in the folder *Input*, it reads the truth and the reconstructed training distributions (*hTruth, hReco*), calculates the Bin-byBin corrections and corrects the data distribution *hData* . Input distributions and the corrected distribution *hUnfold* are written into the output files *histos.root* in the folder *Output*, where they can be tested conveniently using the script *Compare.C*

Programme:

As a first step copy the folder UnfoldingTutorial to a private working folder with a unique name.

Tasks:

1) Familiarize yourself with the toyMC:

   - design various truth distributions. Vary the “data” distribution with respect to the MC distribution. What follows for the reconstructed distributions ?
   - Play with the detector simulation: How can you increase the resolution, shift the reco distribution ?
   - What is the minimal number of MC events to populate all bins properly?
   - We are using the same detector simulation in “data” and “MC” . Which additional complication do we expect in a real detector and how can we account for it in the unfolding ?

2) Bayesian Unfolding:

   - Create the histograms for your favorite function, using the same distribution for data and MC and a sufficiently large MC statistic (typically > 1Mio events) . Store it in the Input directory of the BayesUnfolding package and link it to Input.root.
   - Unfold the data distribution using one iteration and compare unfolded data and truth data. Increase the number of iterations. What do you observe ? How would you determine the optimal number of iterations? What happens if you increase the number of iterations to a very large level?
   - In order to investigate this further create a similar ToyMC with less statistics (typically <10000 events). Increase again the number of iterations. Where would you now place the optimal number of iterations? What happens if you increase the number of iterations further? What trade-off are we dealing with ? Which unfolding technique do you approach with a large number of iterations and what are the problems related to this technique?
   - Create a new ToyMC where you change the “MC” distribution wrt the true “data” distribution, e.g. choose much large width for the “MC” distribution and shift its mean, and store it in the Bayesian Input folder.
   - Unfold various times, increasing the number of iterations. How does the unfolded distribution change and why ?

3) BinbyBin Correction

- Create the histograms of your favorite function, using the same distribution for data and MC. Store it in the Input directory of the BinbyBinUnfolding package and link it to Input.root.
- Does a BinbyBin correction constitute an “Unfolding” ? Why?
- What is the assumption that goes implicitly into this method? If this assumption is not fulfilled, under which conditions can we use the method nevertheless ?
- Unfold the distribution using the BinbyBin programme.
- Now create a new ToyMC where the true “MC” distribution differs from the true “data” distribution, e.g. By shifting the mean of the distribution or by adding resolution effects. What happens if you unfold the MC now?

4) Additional exercises

If you have finished the previous exercises and are eager to dig deeper into the matter there is many additional problems to investigate, e.g:

- Start with a completely flat true distribution in “MC”.
- Investigate the impact of the bin size, in particular bins size wrt resolution
- Instead of comparing graphs by eye – plot the residuals. This way its easier to quantify the bias and to spot oscillations introduced by low statistics
- Use “MC” and “data” with different statistics, e.g. high-statistics MC and low-statistics data
- Change the efficiency as a function of the true value of the measured quantity
- New physics: add a small peak in “data” but not in “MC”. Do you recover the additional peak in the data?
- Introduce migration from outside the histogrammed area towards the histogrammed area