Unfolding

Comparison of methods

Lydia Brenner

Disclaimer

I am in ATLAS, not CMS

➔ I will do my best to not use ATLAS jargon

➔ I will do my best to explain the way of thinking

What I show is not true in every possible analysis or every possible situation.

I will try to keep the introduction brief.

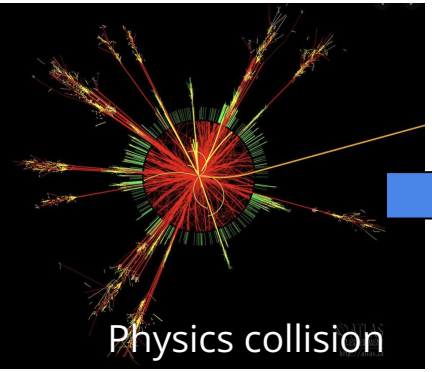
The unfolding problem

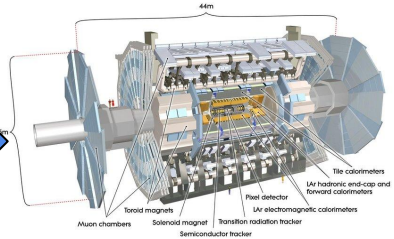
- Estimating the particle-level spectrum
  - of some physical quantity of interest
  - on the basis of observations
  - smeared by an imperfect measurement device
- Estimating the true distribution when you measure a smeared spectrum

The unfolding problem: A picture and a workshop

https://indico.cern.ch/event/1357972

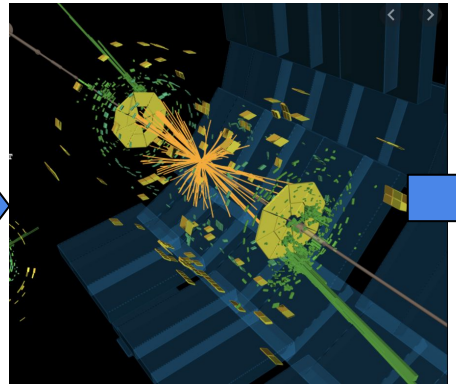
Standard measurement roadmap





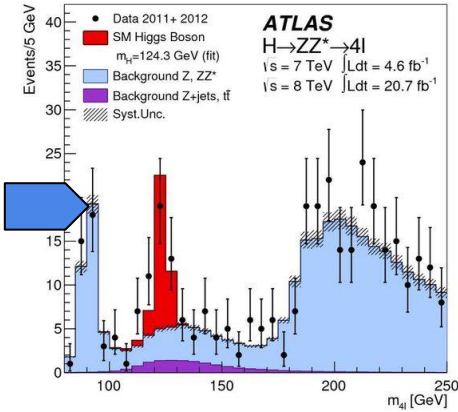
Detector interaction

Detector reconstruction

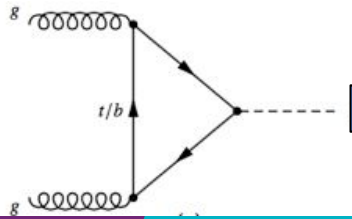


Event selection

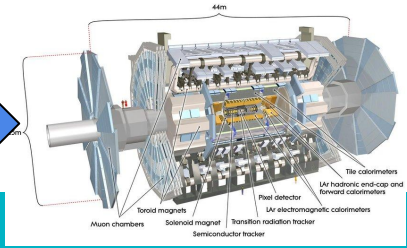
Analysis



Physics model

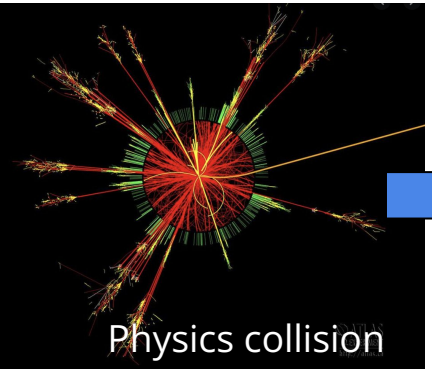


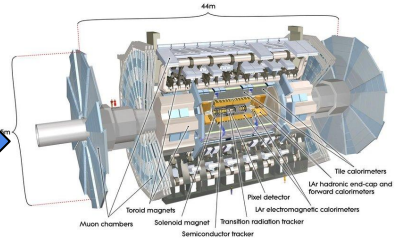
Detector simulation



Nikhef

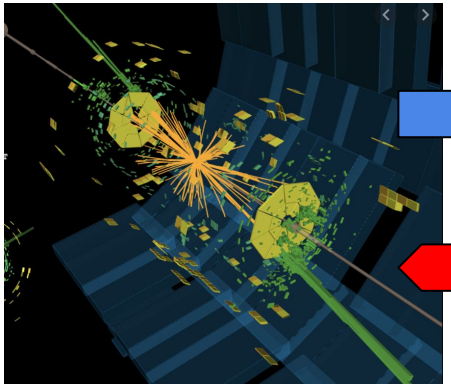
Alternative measurement roadmap: Unfolding





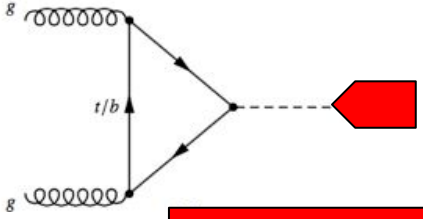
Detector interaction

Detector reconstruction

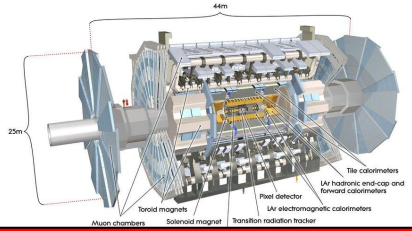


Event selection

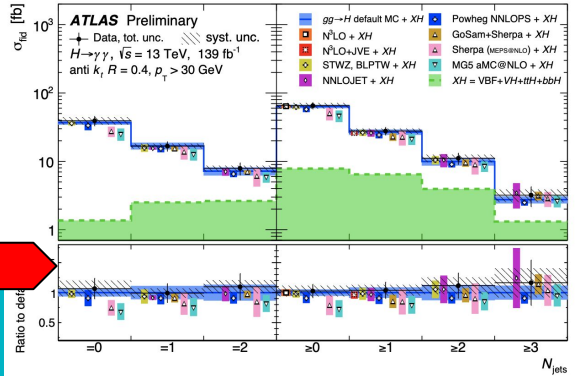
Physics model



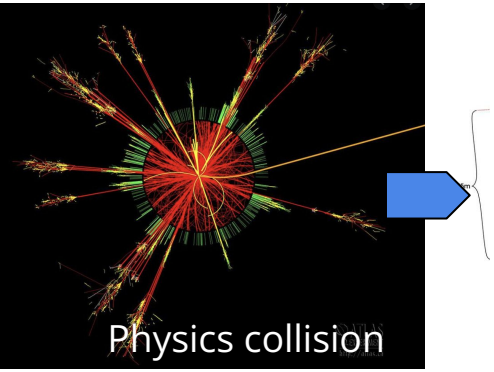
Detector simulation



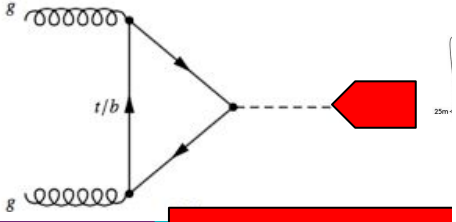
Analysis

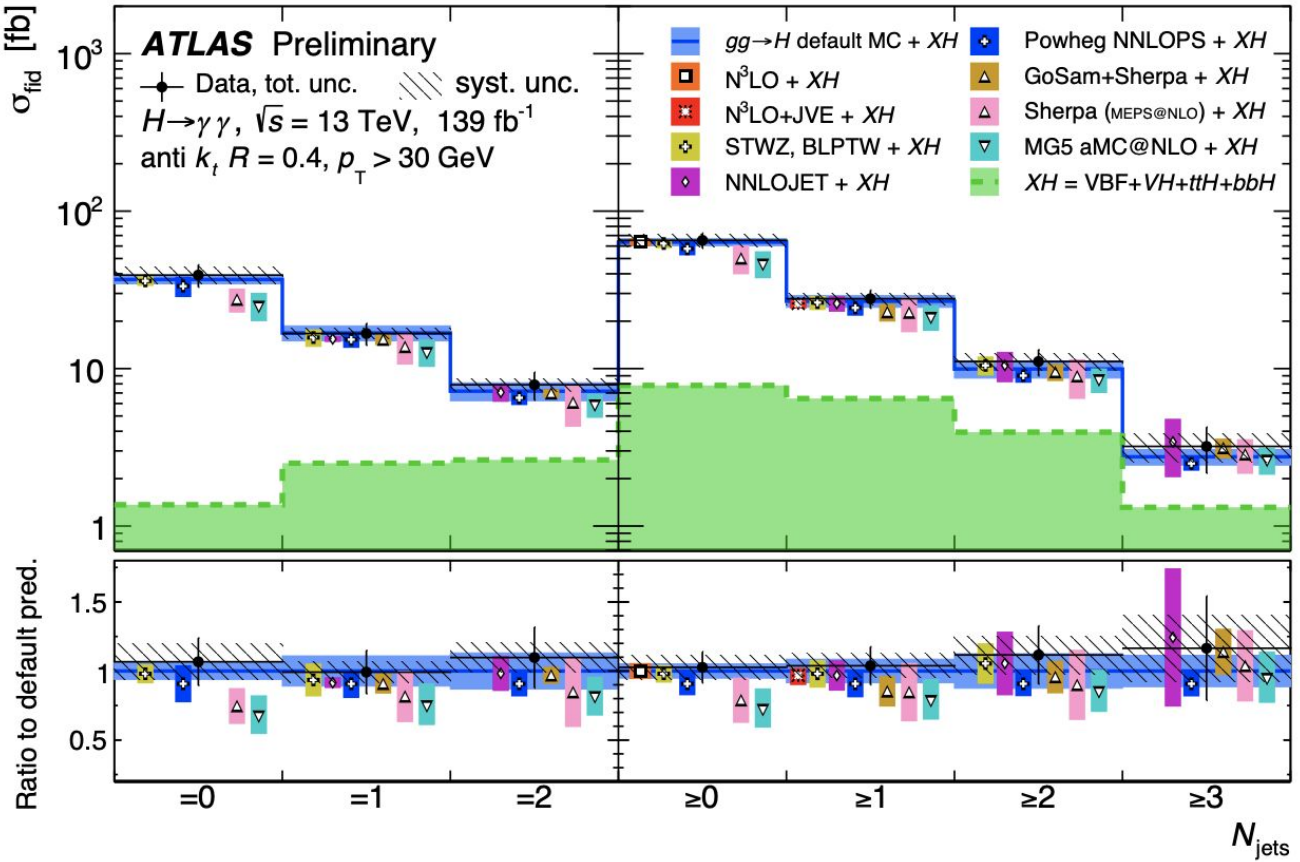


Alternative measurement roadmap: Unfolding

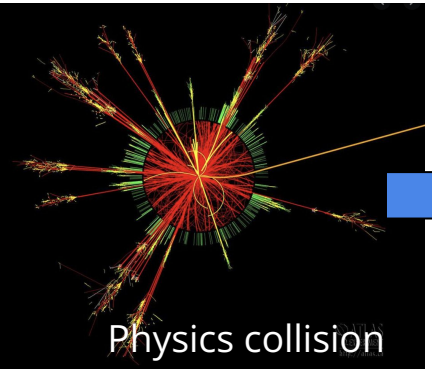


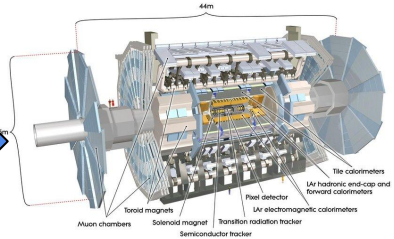
Physics model





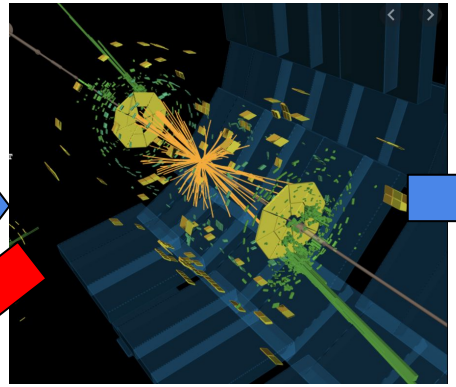
Roadmap comparison





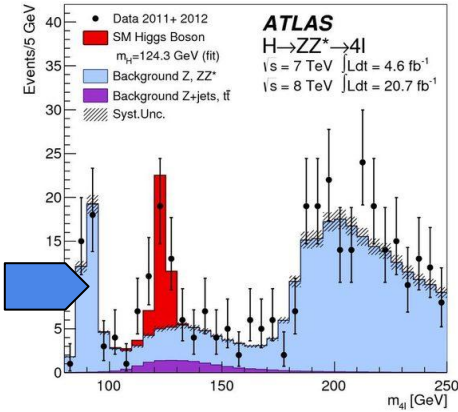
Detector interaction

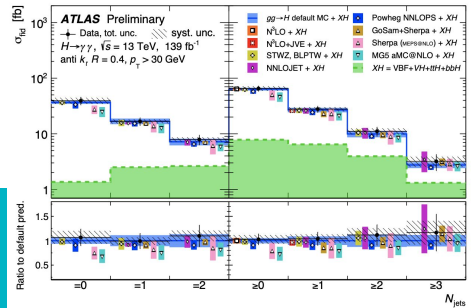
Detector reconstruction



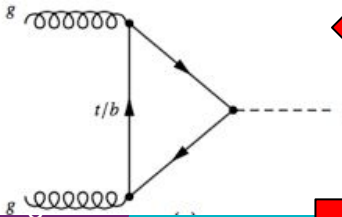
Event selection

Analysis

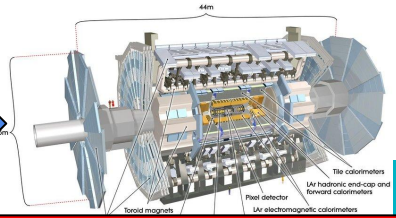




Physics model



Detector simulation



Roadmap comparison

Want to check how well the prediction matches the measurements?

➔ Standard roadmap!

- Hypothesis testing

So why/when do you want to use unfolding?

➔ Multiple predictions differ within the uncertainties

➔ Ready for future reinterpretations and combinations with other experiments

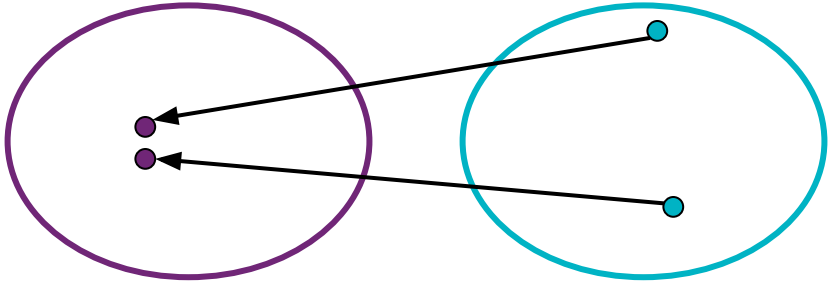
➔ Learn about physics model

- Not only likelihood, but physics quantities

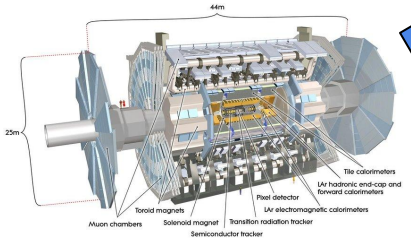
The unfolding problem

Heads up: Unfolding is an ill-posed inverse problem

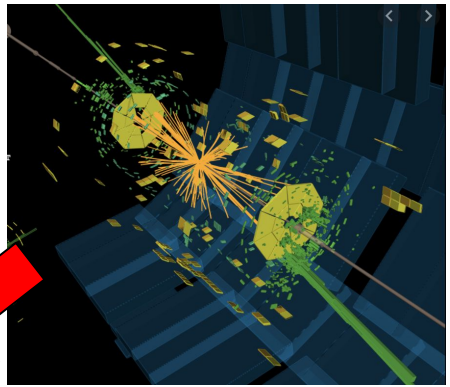
Smearing ‘removes’ information



Detector simulation



Detector reconstruction



Nikhef

Formulation of the unfolding problem

Consider a random variable y, goal is to determine $f(y)$

If parameterization $f(y;\theta)$ known, find Maximum Likelihood estimators $\hat{\theta}$.

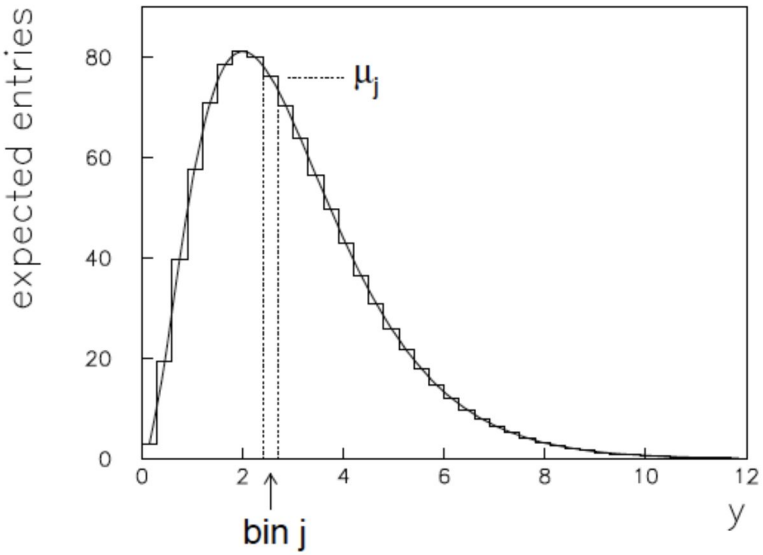
If no parameterization available, often construct histogram:

construct estimators for the μ_j (or p_j)

Where μ_j corresponds to the 'true' histogram

$$\mu_j = \mu_{tot}\, p_j$$

with $p_j = \int_{bin\,j} f(y)dy$



The response matrix

As an effect of performing a measurement we get migrations of entries between bins, aka $f(y)$ is 'smeared out'

$$f_{meas}(x) = \int R(x|y) f_{true}(y) dy$$

In case of discrete dataset, aka histogram, data are $n = (n_1, ..., n_N)$

$$\nu_i = E[n_i] = \sum_{j=1}^{M} R_{ij} \mu_j \, , i = 1, \ldots, N$$

Where

R_{ij} = P(observed in bin i | true in bin j) , is called the response matrix

Note μ and ν are constant, while n is subject to statistical fluctuations

Background and efficiency

Sometimes an observed event is due to a background process

$$\nu_i = \sum_{j=i}^{M} R_{ij}\mu_j + \beta_i$$

Where β_i is the expected number of background events in observed histogram

Sometimes a signal event goes undetected:

$\sum_{i=i}^{M} R_{ij} = \sum_{i=1}^{M}$ P(observed in bin i | true value in bin j) =

= P (observed anywhere | true value in bin j) = ϵ_j

This is called the efficiency

Notation summary

The ‘true’ histogram $\mu = (\mu_1, \ldots, \mu_M)$

Expectation values for observed histogram $\nu = (\nu_1, \ldots, \nu_N)$

Observed histogram $n = (n_1, \ldots, n_N)$

Expected background $\beta = (\beta_i, \ldots, \beta_N)$

Response matrix $R_{ij} =$ P(observed in bin *i* | true in bin *j*)

Related by $E[n] = \nu = R\,\mu + \beta$

Maximum likelihood (ML) solution

Let us take the most obvious solution;
Assume that the problem $\nu = R\,\mu + \beta$ can be inverted

$$\mu = R^{-1}(\nu - \beta)$$

- But remember; ν are the expectation values of the observed histogram, not the observed histogram (which is denoted by n).

Assume as well that the data can be considered as independent poisson distributions for each bin

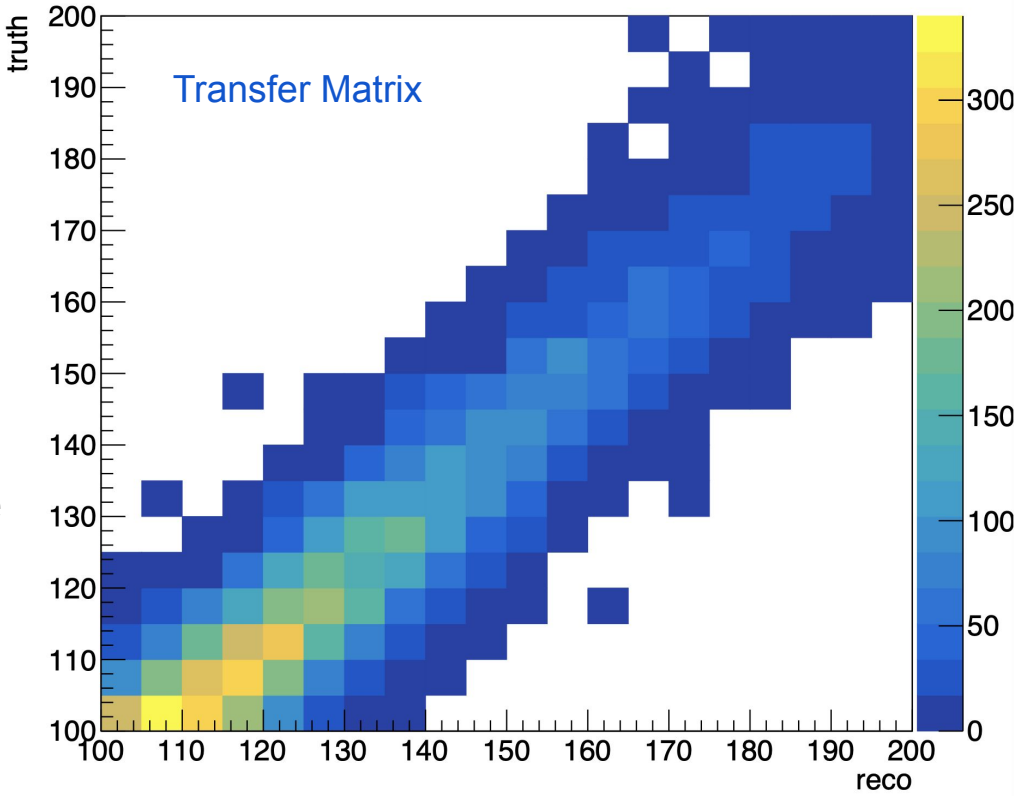
- Now the maximum likelihood estimator is $\hat{\nu} = n$ so then

$$\hat{\mu} = R^{-1}(n - \beta)$$

Example in practice

Instead of the matrix inversion shown in the previous slide, let us consider Single Value Decomposition (SVD) unfolding

- Can be understood as imposing smoothness by cutting off non-diagonal elements in response matrix
- Corresponds to matrix inversion when taking many components

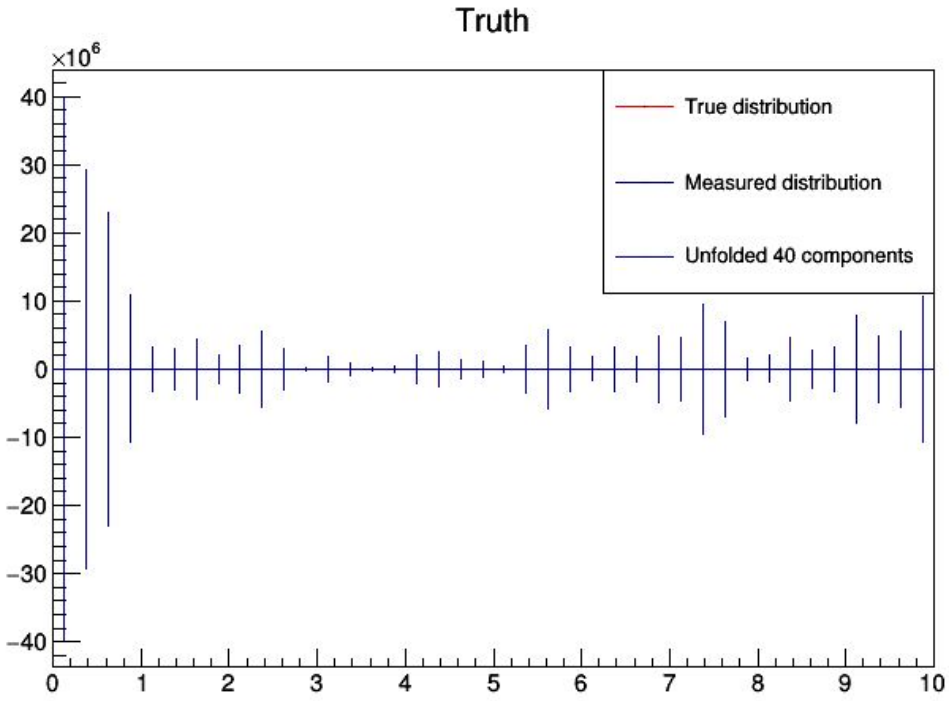


Nikhef

Example in practice

The problem: We don't have ν, only n

- R^{-1} interprets fluctuations in n as the residual of original fine-structure and puts this back into $\hat{\mu}$
- Causes breakdown
  - In this example 40 components corresponds to the matrix inversion solution



Nikhef

Bias, variance and coverage

Conclusion: In unfolding one must accept some bias in exchange for a (hopefully large) reduction in variance. So what do we need to consider?

- Bias: the difference between the estimator's expectation and the parameter's true value.
- Variance: the diagonal elements of the covariance matrix of the estimators for μ
- Coverage: the probability that the true value of μ_i falls between plus and minus one standard deviation about the estimator of μ_i

Note: the bias, variance and coverage are always defined with respect to the chosen true and expected data histograms.

Variance

Use a *toy* dataset with K events, the variance for bin *i* is then calculated as

$$\sigma^2_{\hat{\mu}_i} = \frac{1}{K-1} \sum_{\mathrm{k}}^{K} \left((\hat{\mu}_i)_k - \frac{1}{K} \sum_{\mathrm{k}} (\hat{\mu}_i)_k \right)$$

Bias

Use a *toy* dataset with K events, the bias for bin *i* is then calculated as

$$b_i = \frac{1}{K}\sum_{\mathrm{k}}^{K}(\hat{\mu}_i)_k - \mu_i$$

Coverage

Under the assumption that the $\hat{\vec{\mu}}$ are Gaussian distributed coverage probability can be calculated in closed form as

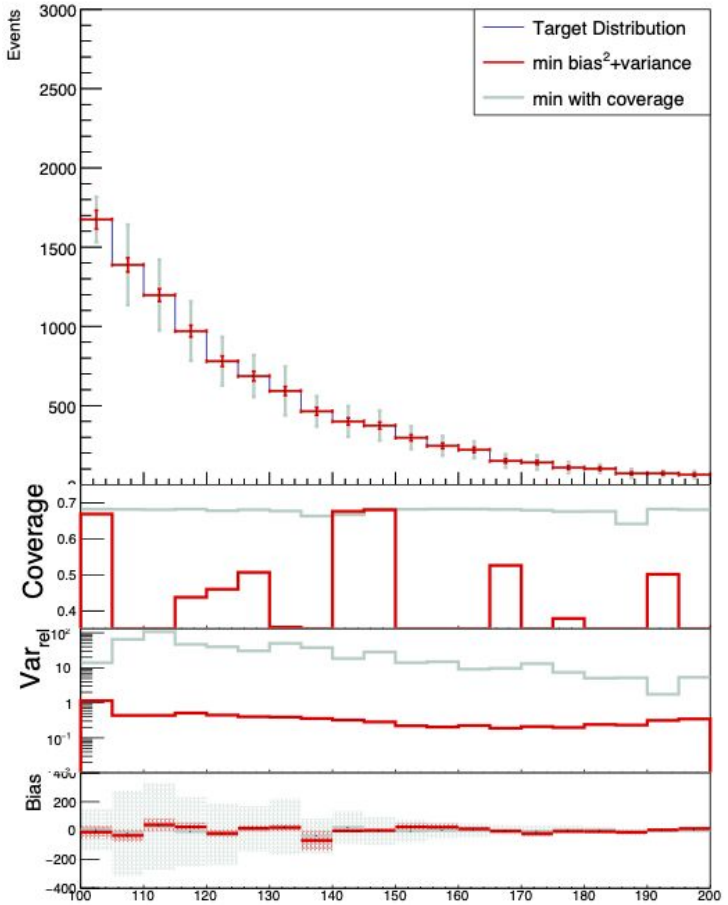
$$P_{\text{cov}} = \Phi\left(\frac{b_i}{\sigma_{\hat{\mu}_i}} + 1\right) - \Phi\left(\frac{b_i}{\sigma_{\hat{\mu}_i}} - 1\right)$$

Where Φ is the Standard Gaussian cumulative distribution function.

Choosing regularisation strength

So how do you take all these things into account in a smart way?

- Choose a regularisation strength that unconditionally minimises the bin-averaged MSE (the mean squared error, aka sum of bias squared and variance, averaged over the bins)
- Require the that the bin-averaged estimate of the coverage of the unfolded data reaches the target coverage of 68.3% within 1%.



Tutorial

- If you have a CERN account, you can just click on the button on the top right will clone a repository with the notebooks to your SWAN for you to work through online
- If you don't have a CERN account or prefer to work locally anyway, clone the repo github.com/vincecr0ft/RooUnfoldTutorials

The tutorial has 4 notebooks:

1. Methods: A basic walk-through on unfolding, showing how to use different methods. Available in C++ and python.
2. Regularisation: A slightly more extensive look at regularized methods
3. Response: An example showing how to use the (relatively new) RooUnfoldSpec for improved error propagation
4. BiasVarianceCoverage: Showing how to use Bias, Variance and Coverage with RooUnfold

Getting a bit more technical : Likelihood formalism

The data are described with a given probability model that determines the likelihood function $L(\vec{\mu}) = P(\vec{n}|\vec{\mu})$. Often the n_i are modelled as independent and Poisson distributed, so that the likelihood is

$$L(\vec{\mu}) = \prod_{i=1}^{N} \frac{\nu_i^{n_i}}{n_i!} e^{-\nu_i}$$

Where you can remember

$$\nu_i = \sum_{j=i}^{M} R_{ij}\mu_j + \beta_i$$

Getting a bit more technical : Regularisation function

To suppress the large variance of the maximum-likelihood estimator one chooses a $\hat{\vec{\mu}}$ that does not correspond to the maximum of the log-likelihood ln L_{max}, but rather one considers a region of $\vec{\mu}$-space where ln L($\vec{\mu}$) is within some threshold below ln L_{max}, and out of these the distribution is chosen that is smoothest by some measure.

This can be achieved by maximising not ln L($\vec{\mu}$) but rather a linear combination of it and a regularisation function S($\vec{\mu}$), which represents the smoothness of the histogram $\vec{\mu}$

So the estimators $\hat{\vec{\mu}}$ are determined by the maximum of $\varphi(\vec{\mu}) = \ln L(\vec{\mu}) + \tau S(\vec{\mu})$
Where τ determines the balance between the two terms.

Getting a bit more technical : Tikhonov regularisation

Two of the unfolding algorithms (TUnfold and SVD) included in the study presented today use Tikhonov regularisation, which for discrete bins can be expressed as

$$S(\vec{\mu}) = - \sum_{i=1}^{M-2} (-\mu_i + 2\mu_{i+1} - \mu_{i+2})^2$$

Other example method included in the comparison

Some algorithms regularise not through the maximisation of function but through and iterative method. Richardson-Lucy as implemented by D'Agostini updates a trial solution in succession. The number of updates plays the role of the regularisation parameter, such that zero iterations gives the maximally smooth trial solution and for a large number the estimators tend towards those of maximum likelihood.

➔ For essentially all unfolding methods the analyst must choose, explicitly or otherwise, some parameter that regulates the degree of smoothness imposed on the solution and thus determines the trade-off between bias and variance in the estimators $\hat{\vec{\mu}}$.

Study on smeared exponential distribution

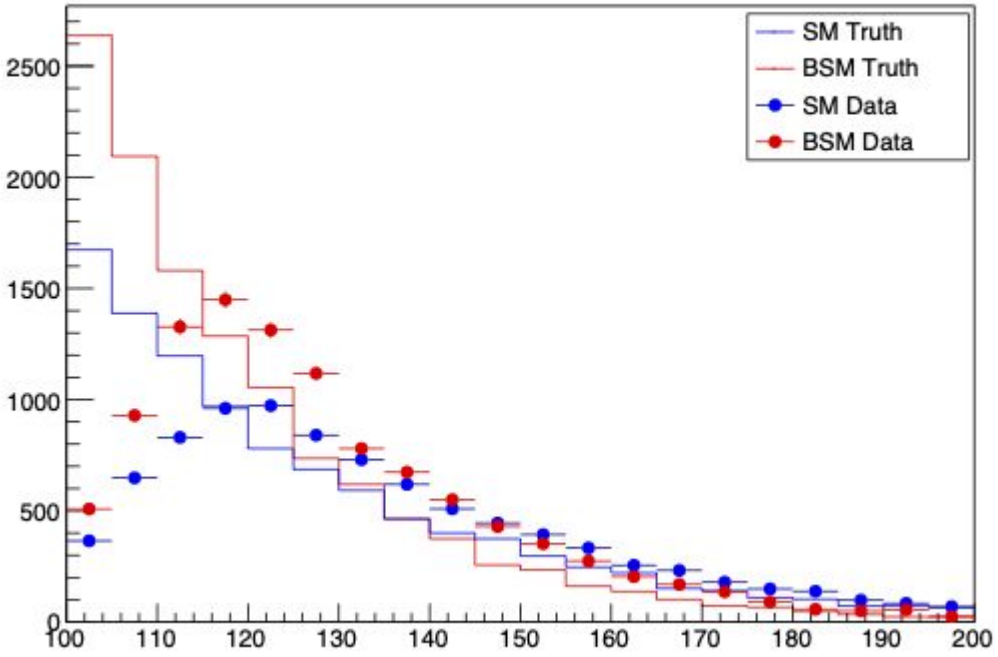
We define as benchmark model an exponential decay distribution, smeared with a resolution function that is loosely inspired on a calorimeter response:

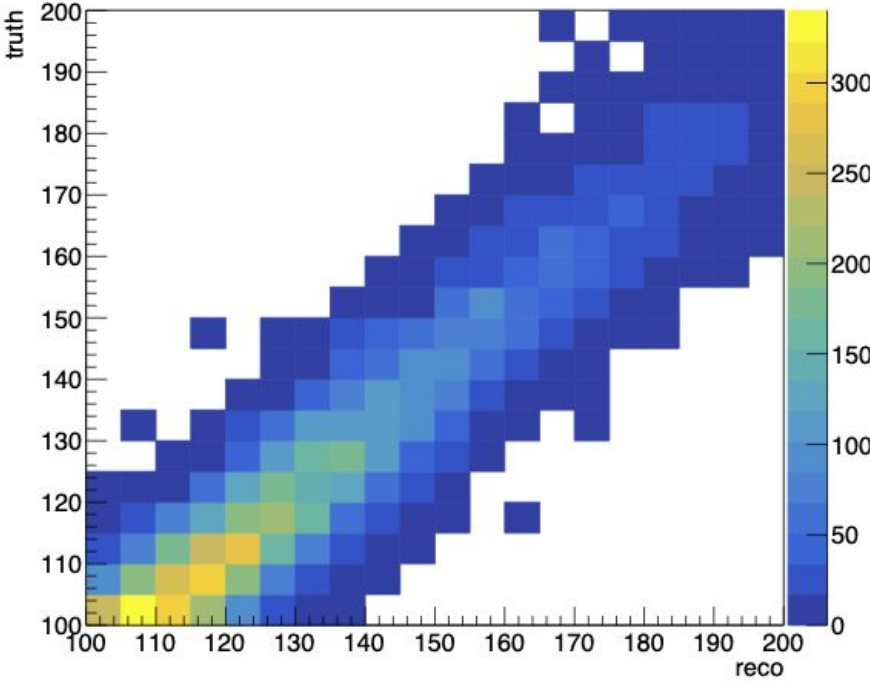
$$\begin{aligned} f(x|\alpha) &= f_{\text{physics}}(x_{\text{true}}|\alpha) * f_{\text{detector}}(x_{\text{true}}, x) \\ &= (\alpha \cdot \exp(-\alpha \cdot x_{\text{true}})) * \text{Gauss}\left(x - x_{\text{true}}, 7.5, 0.5 \cdot \sqrt{x_{\text{true}}} + 2.5\right) \end{aligned}$$

where the $*$ symbol represents the convolution operator.

We define two models variants, labeled SM (‘Standard Model’) and BSM (‘Beyond the Standard Model’), that correspond to an exponential distribution with a slope α of 0.035 and 0.05 respectively

Study distributions





Nikhef

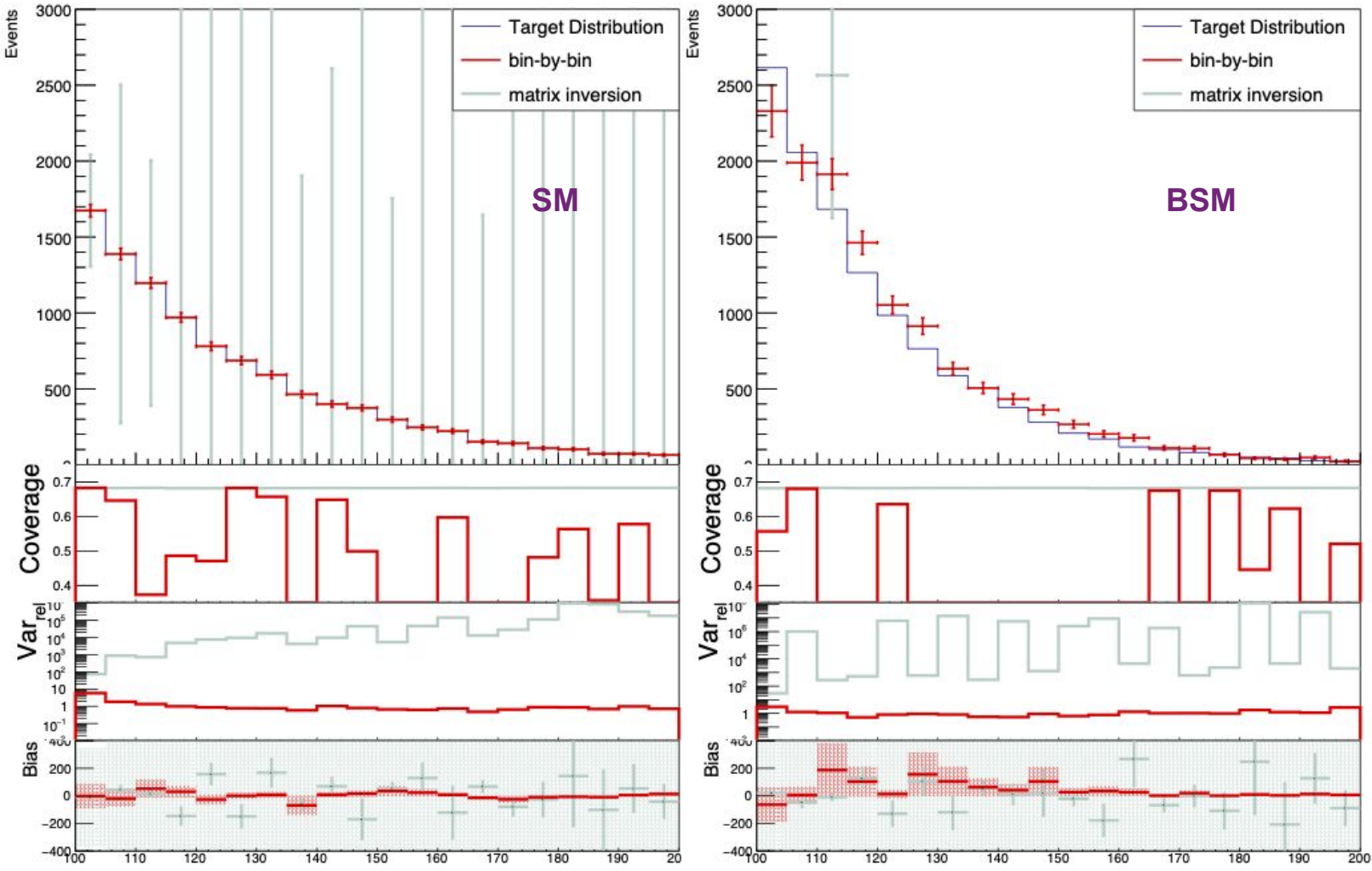
Other considerations on regularisation strength

Automated method for choosing the regularisation strength is included in latest release of RooUnfold

Besides bias, variance and coverage using a MSE method;

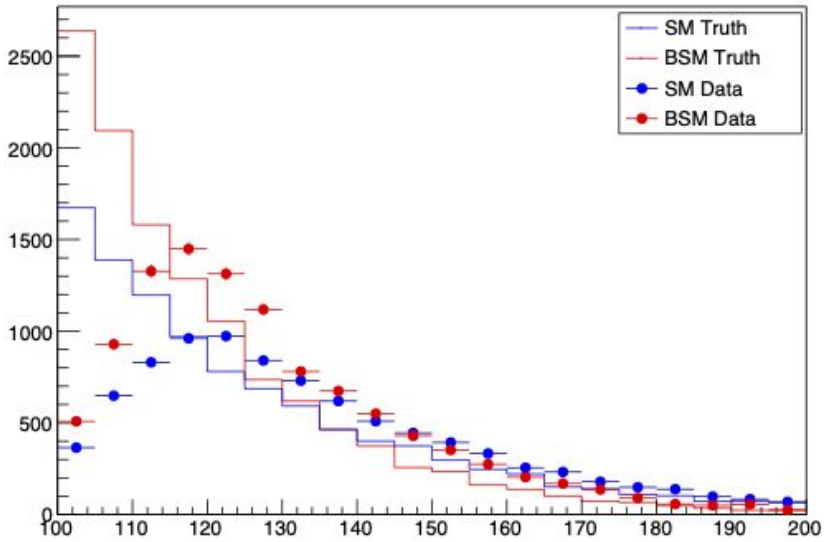
- Consider choosing the regularisation strength using a dataset that includes all uncertainties.
  - Look at the transfer matrix, with all uncertainties included, and reconsiders the binning if large off-diagonal elements are present.
- Test for the choice of regularisation strength on alternate samples with respect to the expected data histogram that was used to create the response matrix
  - One way of creating the alternate sample is by fluctuating the expected data histogram within its statistical uncertainty.

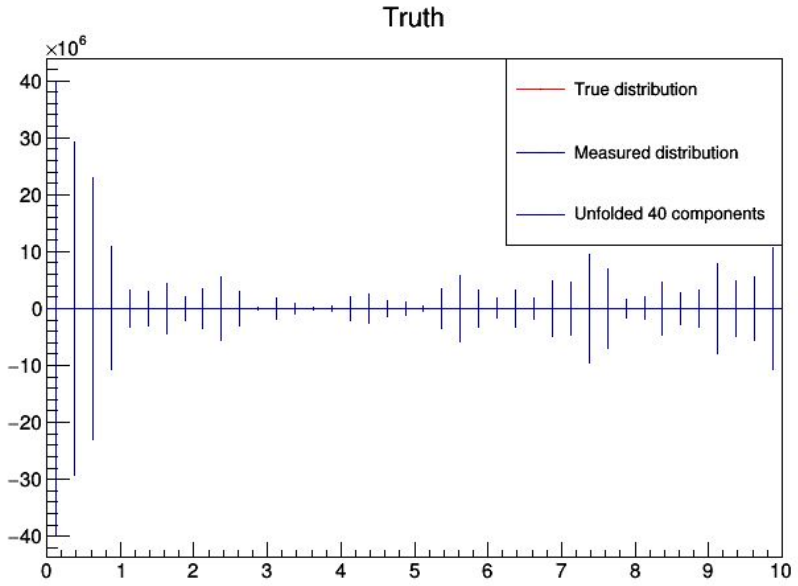
Methods that are not tunable



Single Value Decomposition (SVD)

Remember

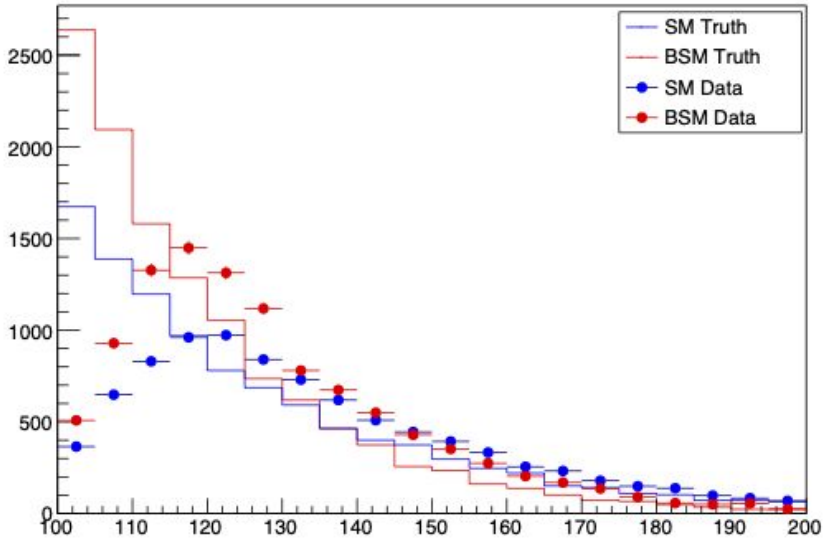


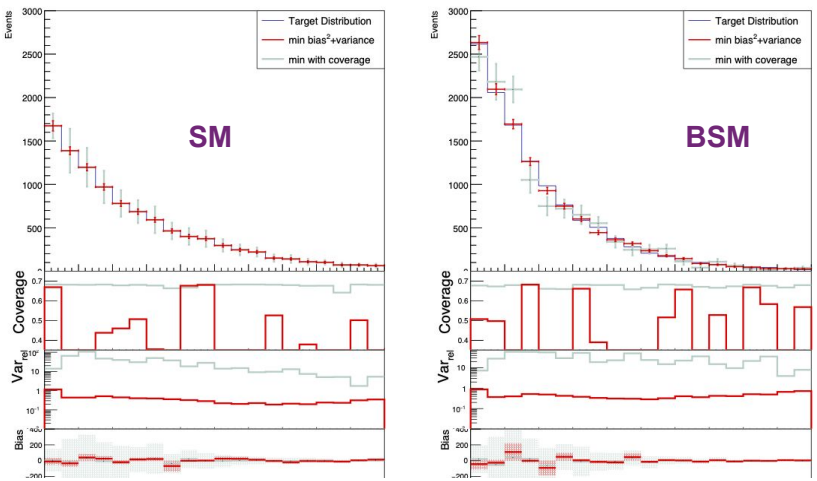


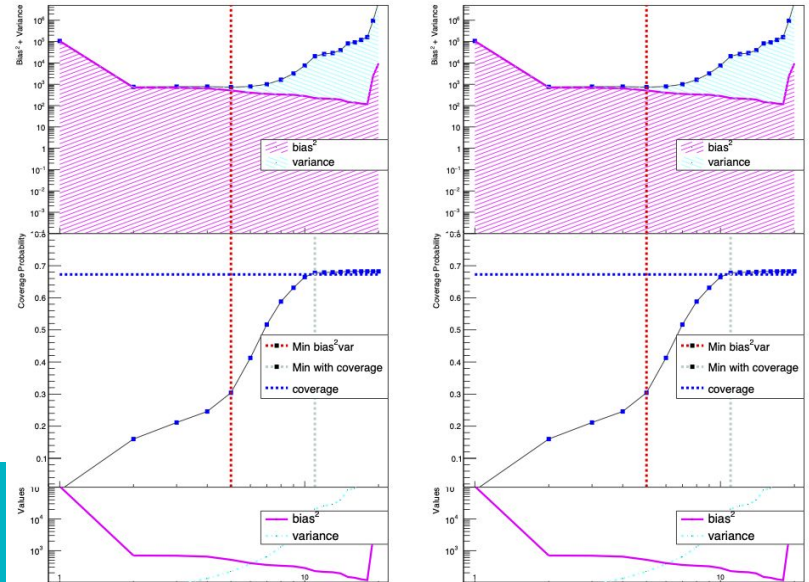
Nikhef

Single Value Decomposition (SVD)

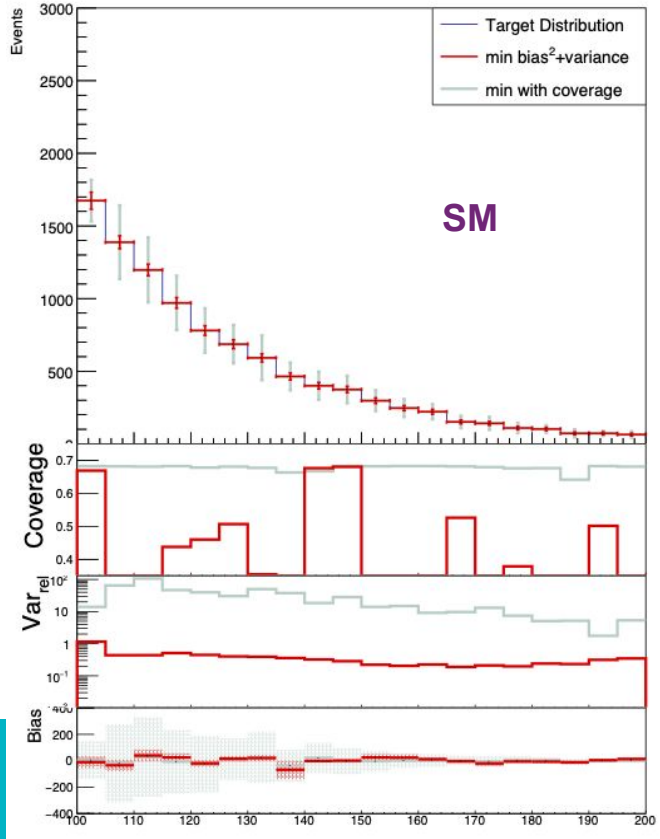
Looking at Coverage, Variance and Bias

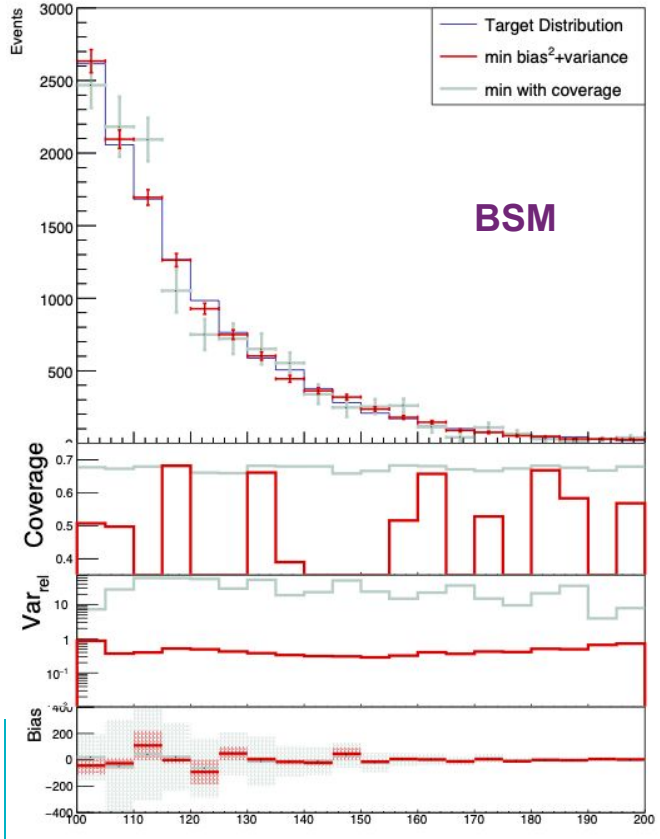




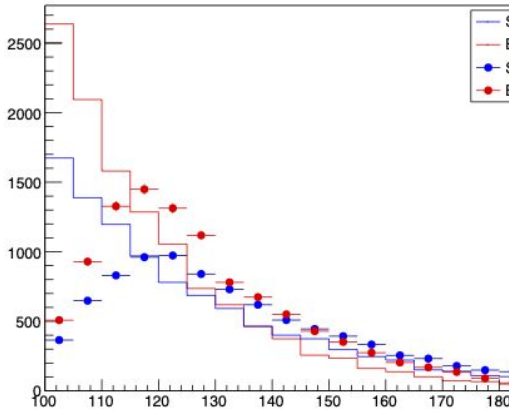


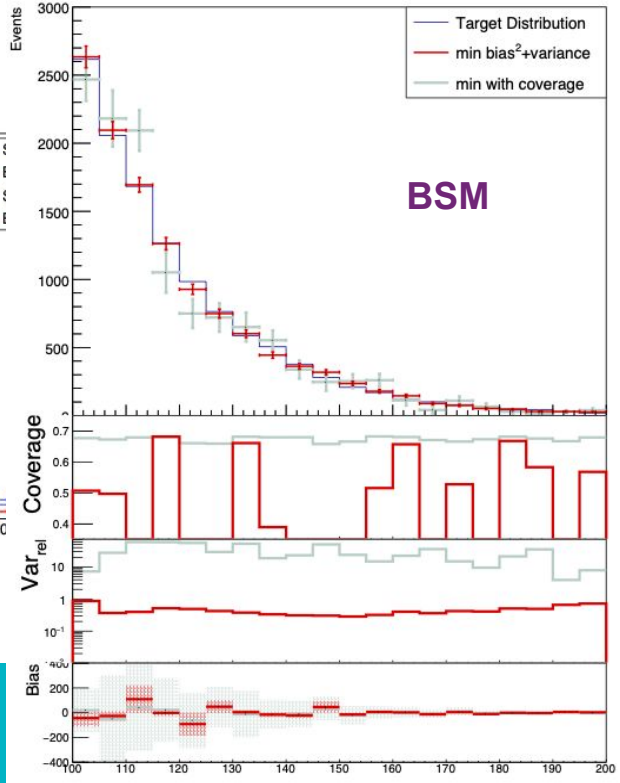
Single Value Decomposition (SVD)

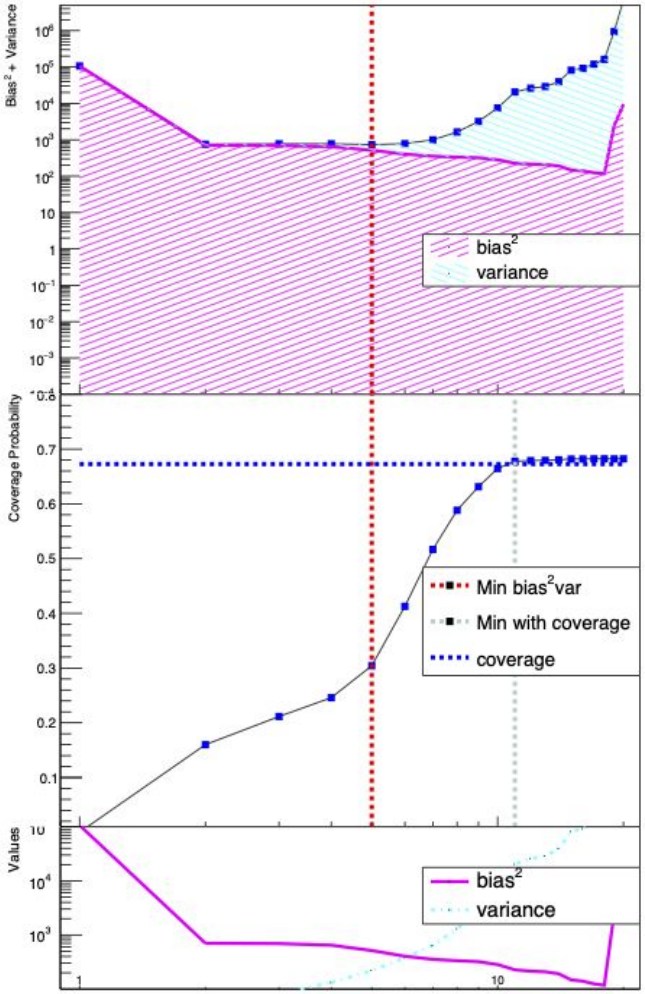




Single Value Decomposition (SVD)

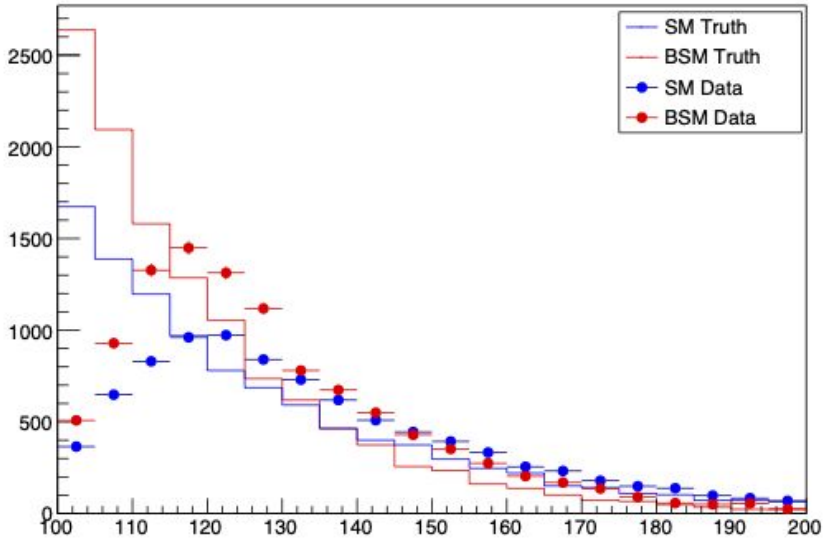


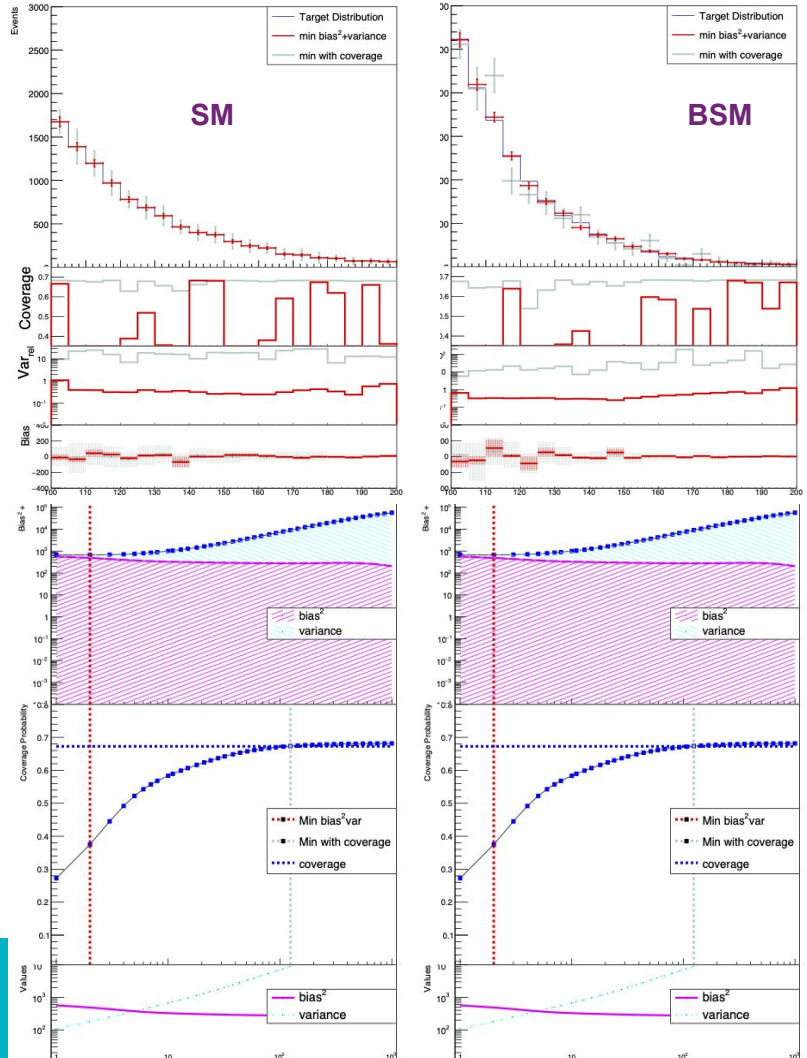




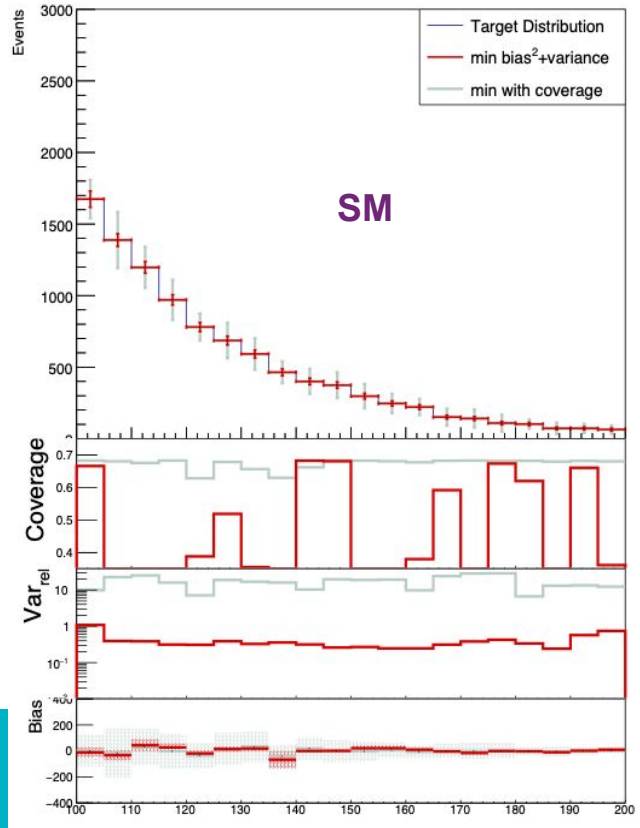
Richardson-Lucy (D'Agostini)

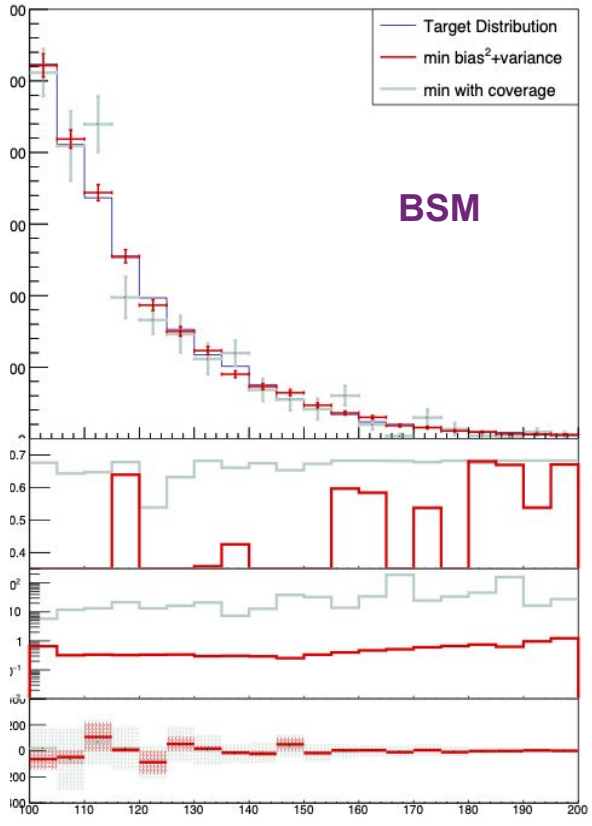
Looking at Coverage, Variance and Bias



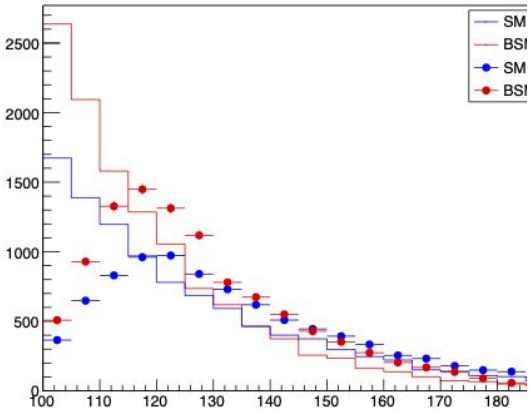


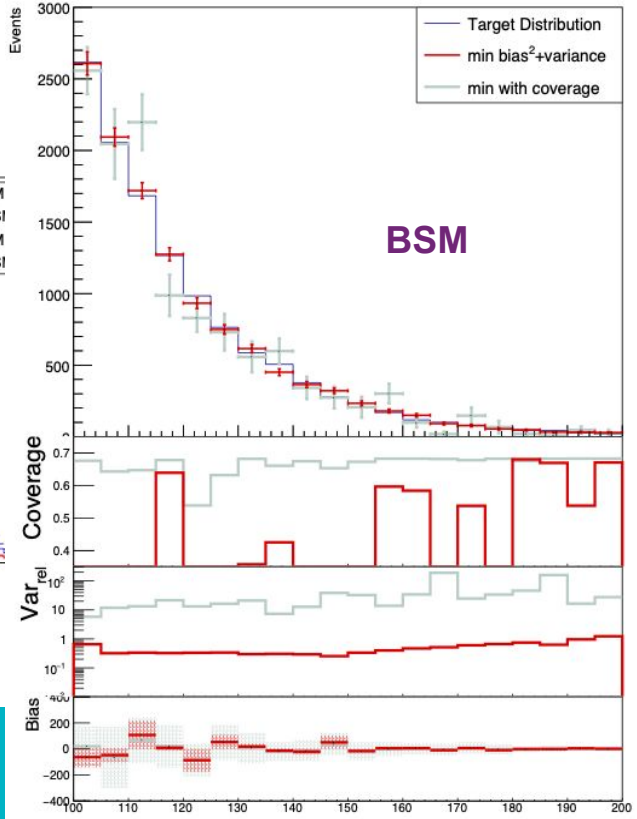
Richardson-Lucy (D’Agostini)

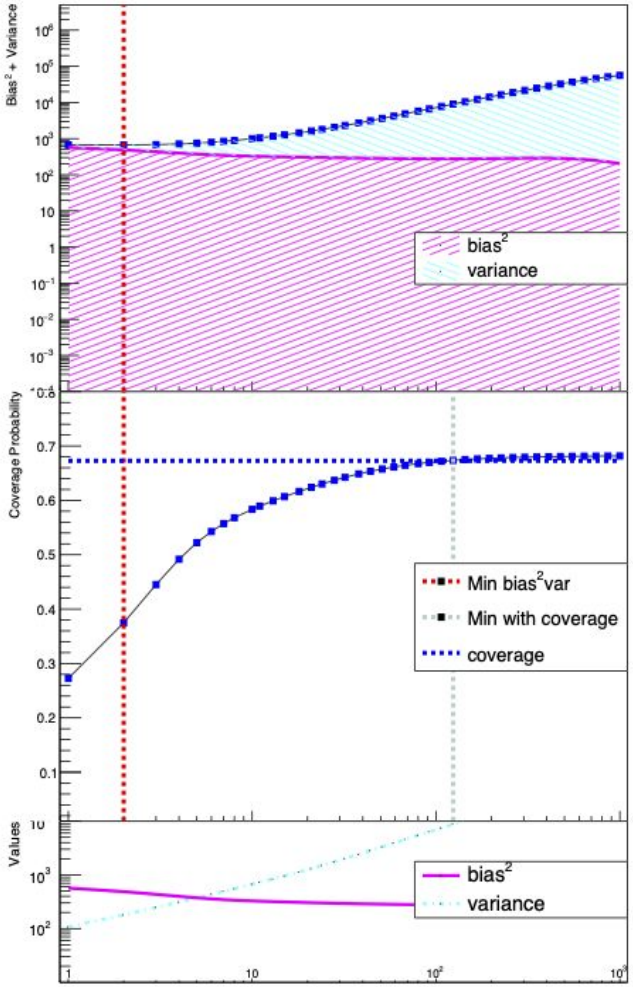




Richardson-Lucy (D'Agostini)







A different study: Bimodal distribution and biases

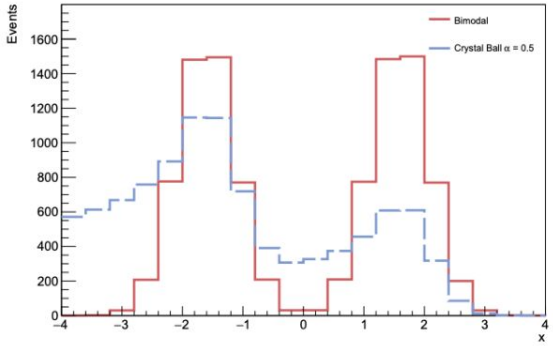
Look deeper into situations where the response matrix and the data are not sampled from the same model.

The bimodal model for this study is the sum of two Crystal Ball functions smeared by a a Gaussian resolution model

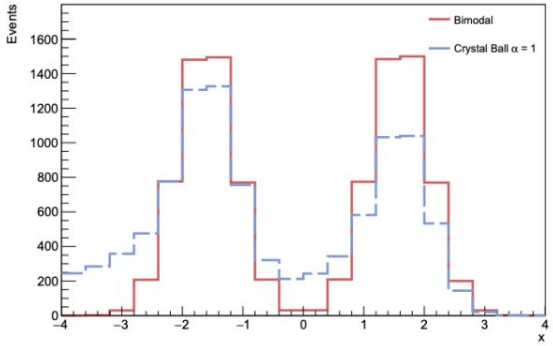
$$
\begin{aligned}
f(x|\alpha) &= f_{\text{physics}}(x_{\text{true}}|\alpha) * f_{\text{detector}}(x_{\text{true}}, x) \\
&= (0.5 \cdot f_{CB}(x_{\text{true}}|\mu = 2.4, \sigma = 0.48, \alpha, n = 1) + \\
&\quad 0.5 \cdot f_{CB}(x_{\text{true}}|\mu = 5.6, \sigma = 0.48, \alpha, n = 1)) \\
&\quad * Gauss(x - x_{\text{true}}, 0, 0.4)
\end{aligned}
$$

A different study: Bimodal distribution and biases

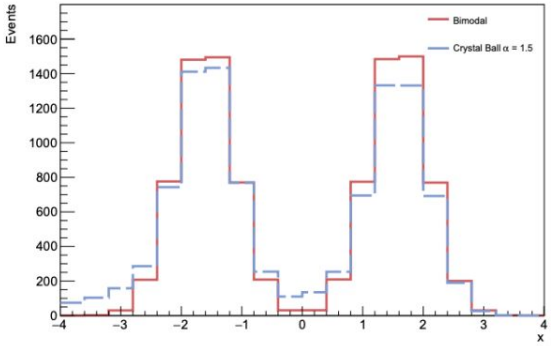
$\alpha = 0.5$



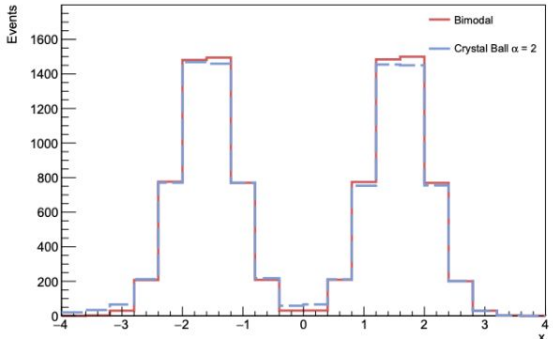
$\alpha = 1$



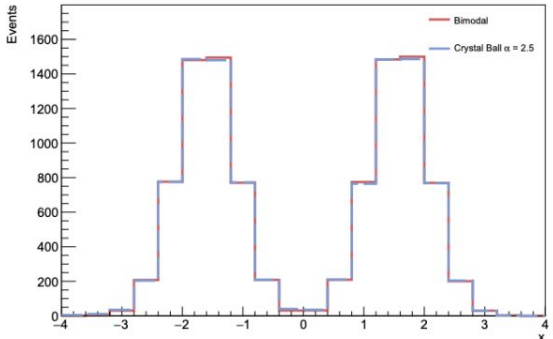
$\alpha = 1.5$



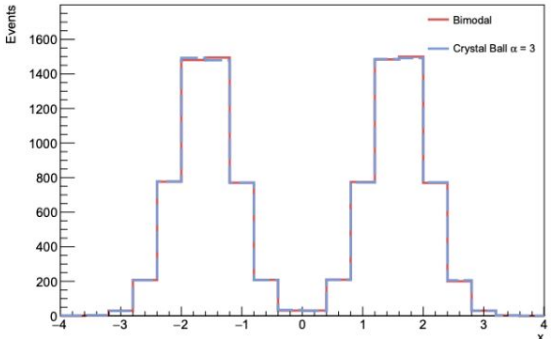
$\alpha = 2.0$



$\alpha = 2.5$

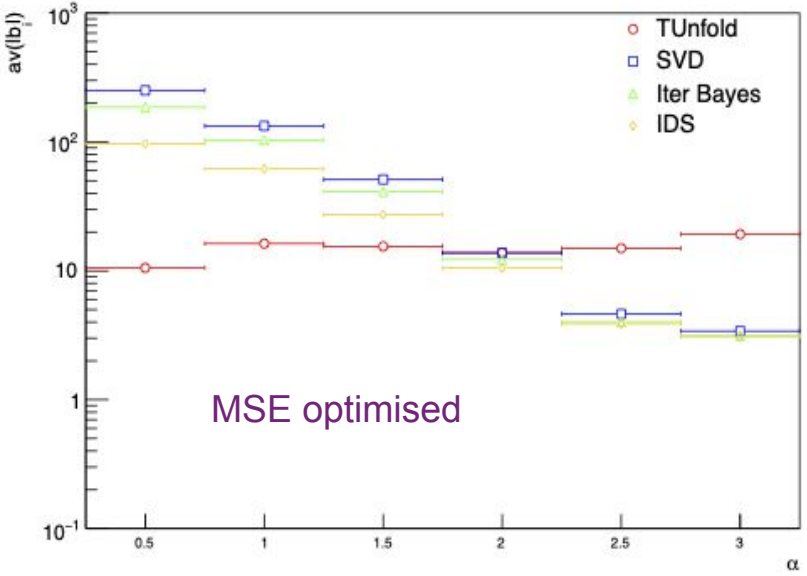


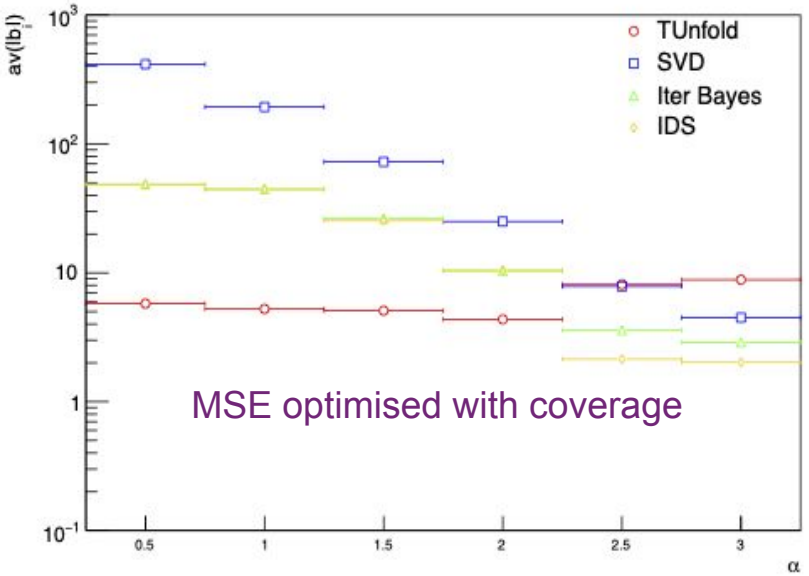
$\alpha = 3.0$



A different study: Bimodal distribution and biases

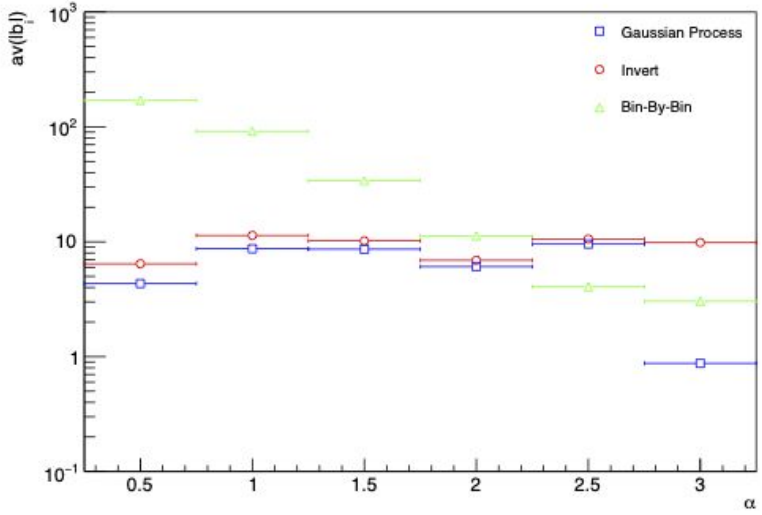
Bin-averaged unfolding bias for data from the distorted distribution unfolded with a response matrix for an undistorted distribution.





A different study: Bimodal distribution and biases

Bin-averaged unfolding bias for data from the distorted distribution unfolded with a response matrix for an undistorted distribution.



Nikhef

Conclusions and summary

- For the methods with a tuneable regularisation strength we observed that an optimisation of that strength solely on the smallest MSE does not automatically result in good coverage.
- Optimising your regularisation strength with a distorted distributed is advised.
- The unfolding bias always depends on the assumed true distribution,
  - This dependence is strongest for methods that regularise using the assumed true distribution.
- Unfolding remains to have no perfect solution

Outlook

- We are always open to more methods to be added to the HEP unfolding package RooUnfold
  - If you have suggestions, please let us know
  - Wide-bin unfolding method scheduled to go in soon
  - Additional uncertainty handling methods scheduled to go in
  - In-Likelihood unfolding method scheduled to go in soon
  - ML unfolding methods planned to go in

Where do I find more information and code?

- RooUnfold; https://gitlab.cern.ch/RooUnfold/RooUnfold
- User guide; https://gitlab.cern.ch/RooUnfold/documentation/-/blob/master/RooUnfold_user_guide.pdf
- Paper; Comparison of unfolding methods using RooFitUnfold., International Journal of Modern Physics A, Vol. 35, No. 24, 2050145 (2020) https://arxiv.org/abs/1910.14654

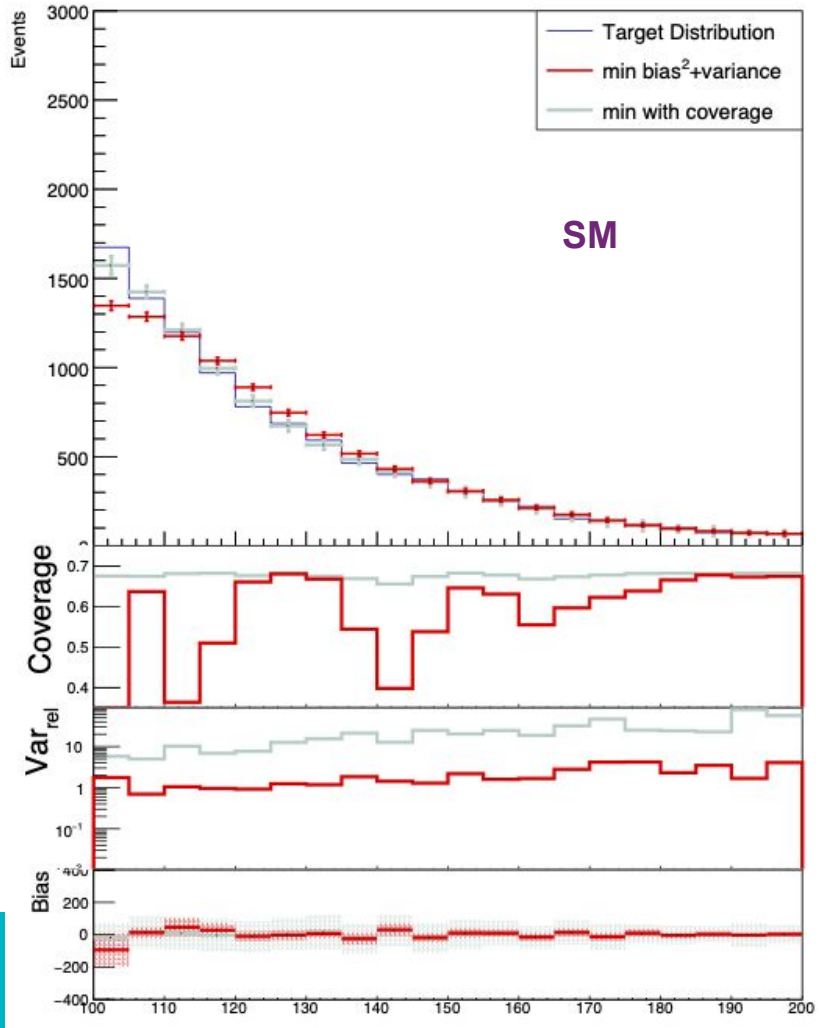
Issues or questions?
Email roounfold-support@cern.ch

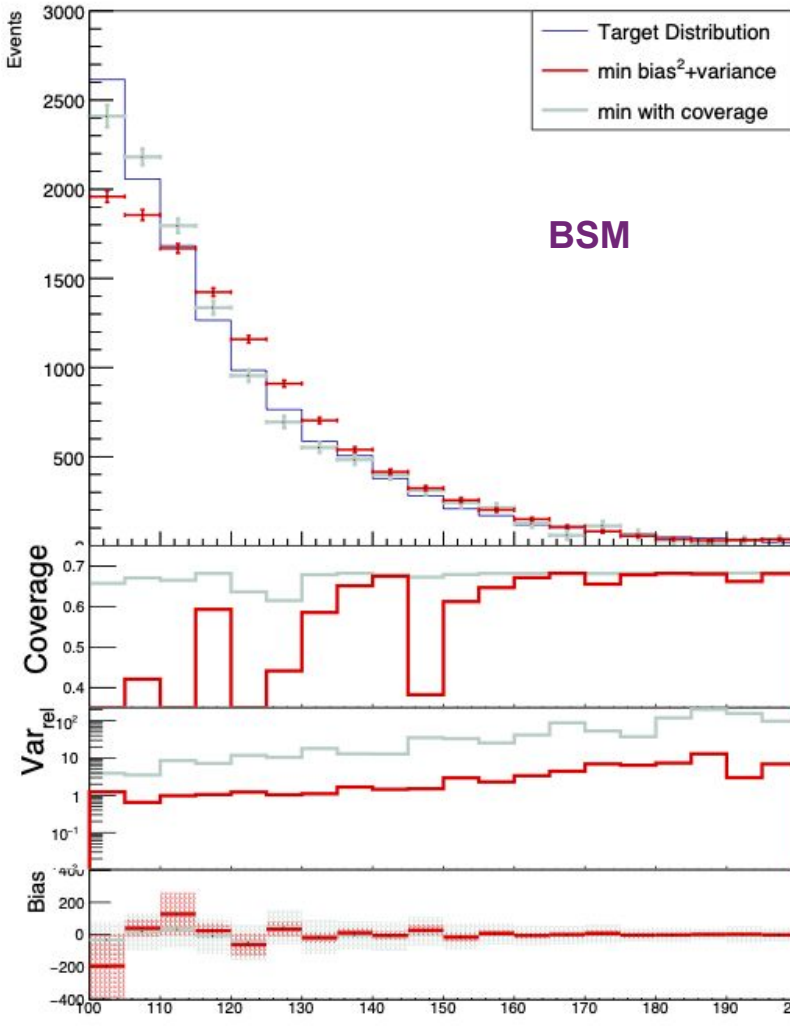
Thanks!

The whole RooUnfold team
Tim Adye, Carsten Burgardm Vincent Croft
Study collaborators
Pim Verschuuren, Glen Cowan, Wouter Verkerke

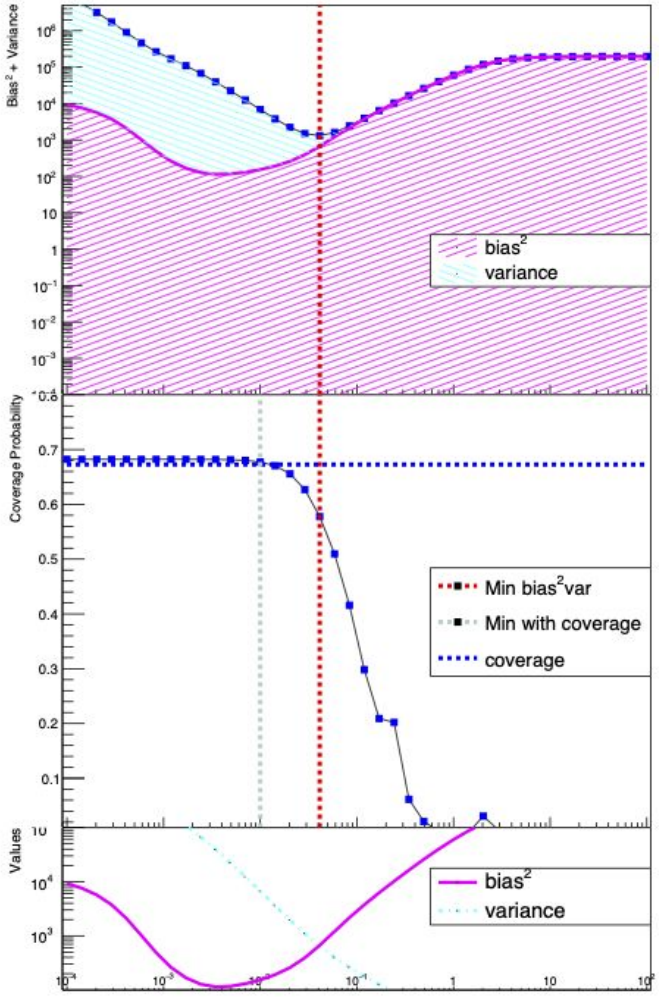
Back up

TUnfold

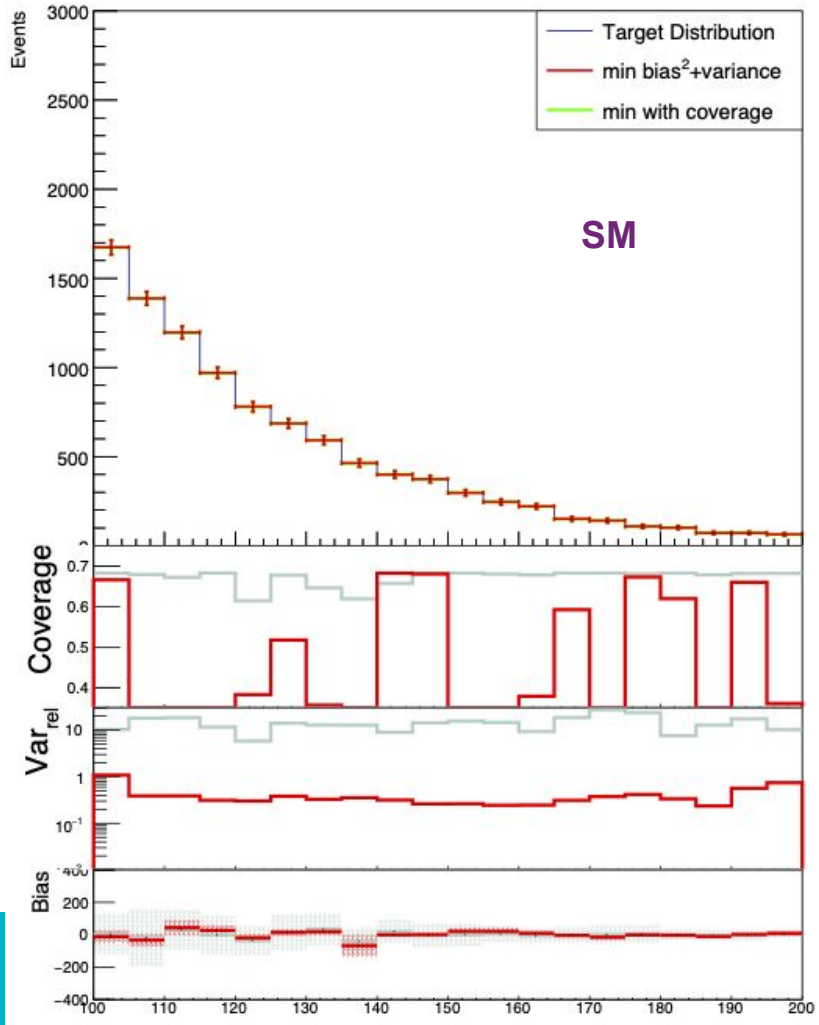

Events
Target Distribution
min bias²+variance
min with coverage
SM
Coverage
Var_rel
Bias

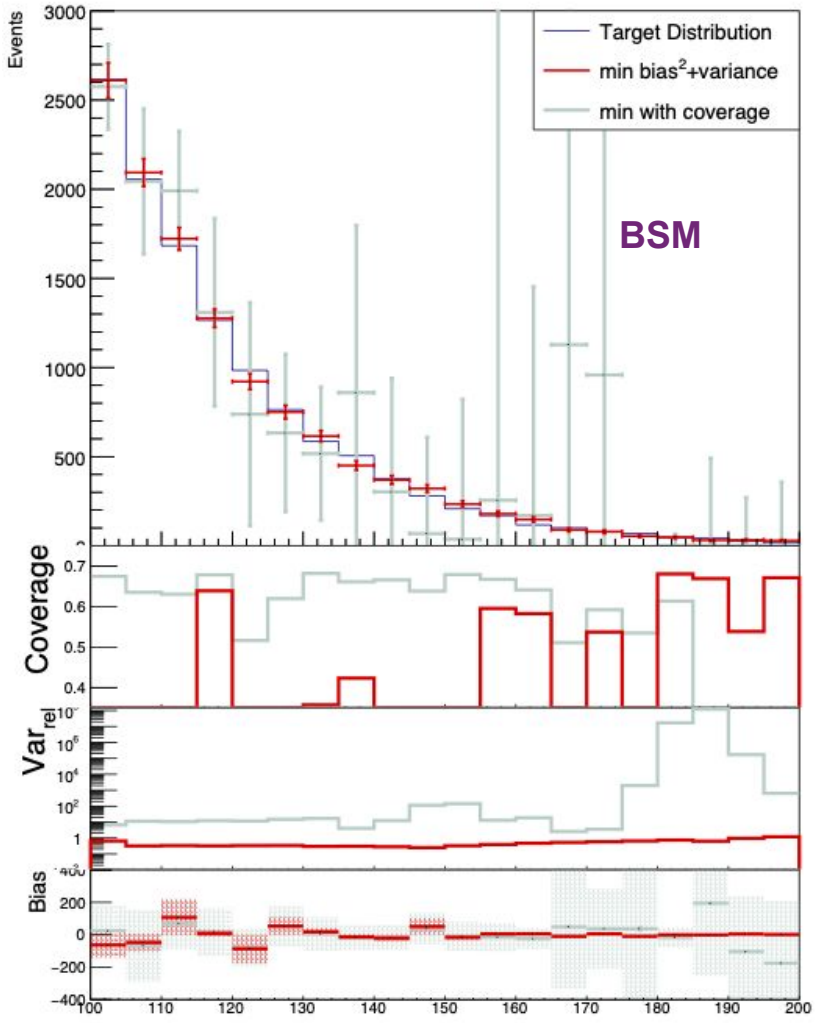

Events
Target Distribution
min bias²+variance
min with coverage
BSM
Coverage
Var_rel
Bias

TUnfold

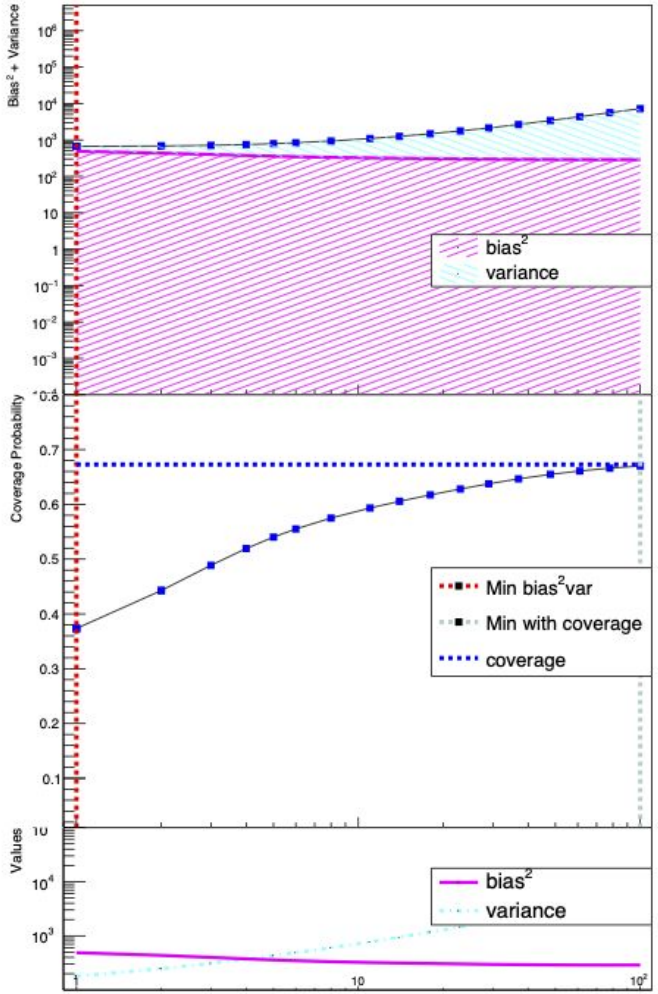

Bias² + Variance
bias²
variance
Coverage Probability
Min bias²var
Min with coverage
coverage
Values
bias²
variance

IDS


Events
Target Distribution
min bias²+variance
min with coverage
SM
Coverage
Var_rel
Bias


Events
Target Distribution
min bias²+variance
min with coverage
BSM
Coverage
Var_rel
Bias

IDS


Bias² + Variance
bias²
variance
Coverage Probability
Min bias²var
Min with coverage
coverage
Values
bias²
variance

Introduction RooUnfold update

- Many frameworks / implementations for unfolding exist
  - Most use RooUnfold as a backend internally
  - Main focus of many of these frameworks: uncertainty handling
- Updates to RooUnfold itself
  - Integration with RooFit
    - Easier uncertainty handling
  - Workspace handling
    - Easy for combination
- Make RooUnfold “future proof”
  - Ready for possible unbinned unfolding methods in the future
  - Improved user friendliness
- End product
  - Saved in a way to allow changing of method at later time

Unfolding and fitting

- Unfold taking systematic variations into account
  - Also done by other advanced frameworks
- Can unfold after fitting signal and background contributions
  - Need to bring fit results into a format suitable as unfolding input
  - Non-trivial to propagate uncertainties
- Why not do both at the same time?
  - Ideal solution: RooFit implementation of unfolding!

Introduction to RooFitUnfold

- Idea: Updated implementation of RooUnfold directly in RooFit
- Includes: Improved handling of uncertainties
  - Uses error propagation from any NPs to the unfolded distribution
  - Allows for inclusion of uncertainties coming from migration matrix
- Handels different input formats
  - Histograms (as already RooUnfold did)
  - pdfs -> Means unbinned distributions can now be unfolded
    - Binned methods allow setting of internal binning
    - unbinned methods can technically be included in the future
- Lives in workspaces

Methods included

All RooUnfold methods included

- Iterative Bayes
- IDS
- SVD
- TUnfold
- Gaussian Processes unfolding (NEW)
- Poisson unfolding, a simple likelihood unfolding (NEW)
- Unregularised
  - Bin-by-bin
  - Matrix inversion
- Can easily include more methods

Documentation: https://gitlab.cern.ch/roofitunfold-tutorial-2019/RooUnfold/blob/master/README.md

https://arxiv.org/pdf/1105.1160.pdf

Implementation

- RooUnfold uses TH1 objects as basis
  - Very user-friendly, but internally not ideal with RooFit
- Templated to use RooAbsReal as a base object
  - Can easily be plugged on top of an existing workspace
- Created RooUnfoldFunc
  - a RooAbsReal wrapper around RooUnfold

Implementation: Inputs

- Truth distristributions
  - Histograms (TH1) or pdf (RooFit/Workspace)
- Reco distributions
  - Histograms (TH1) or pdf (RooFit/Workspace)
- Response matrix
  - 2D Histogram (TH2) or pdf (RooFit/Workspace)
- Data: background subtracted if needed
  - binned (TH1 or RooDataHist) or unbinned (TTree or RooDataSet)

Implementation: Features

- RooUnfoldFunc can be imported into workspace
  - Can use any existing workspace as an input
  - Can update reco level workspace after unfolding
  - Can unfold and fit (on reco-level) simultaneously
  - Easy persistence
  - Extremely useful for combinations
- RooUnfoldSpec can be used to construct RooUnfoldFunc
  - Helper class similar to HistFactory
- Unfolding result is only cached
  - Can switch to a different unfolding method a-posteriori

Workspace write out

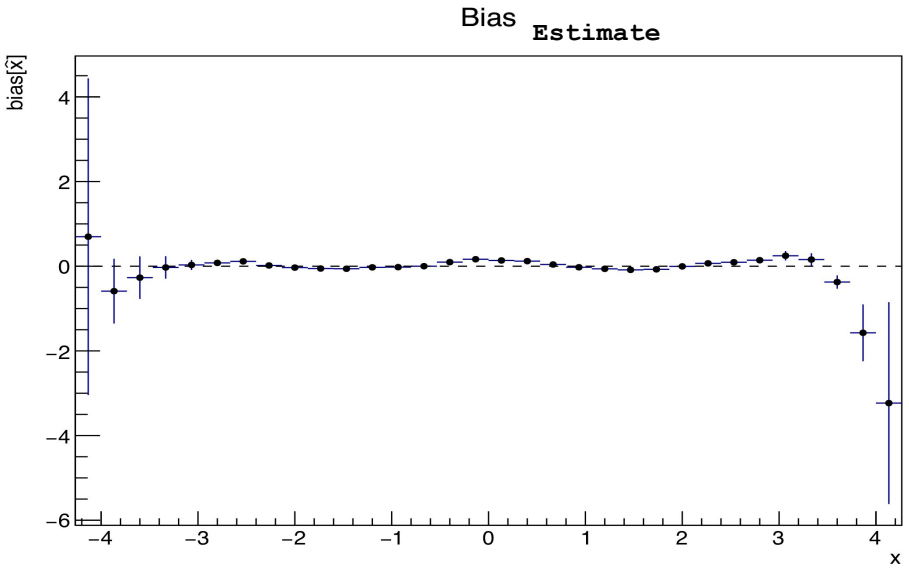
- Directly written out into a workspace
  - At any level of the analysis
  - Saves all information to be able to do a change of unfolding method on the fly
  - Includes error propagation
  - Writes out for ALL unfolding methods
    - So also for regularised methods

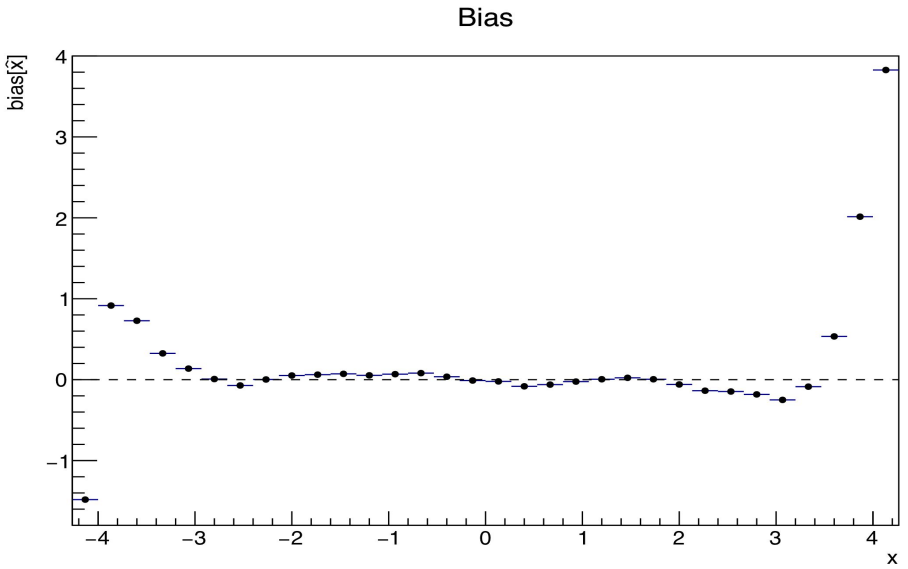
Error propagation

- Default RooUnfold can propagate simple uncertainties
  - Statistical uncertainties on Data
  - Bin-by-bin correlations
  - No handling of systematic uncertainties!
- RooFit functions (pdfs) can depend on arbitrarily many parameters
  - automatic error propagation from input parameters to all outputs by RooFit
  - ony requirement: the output needs to be a RooFit object
  - Nuisance parameter treatment comes “for free” with RooUnfold integration in RooFit
- No explicit handling of systematic uncertainties needed in RooUnfold
  - RooUnfold+RooFit handles uncertainties neatly :)
  - Some toy sampling methods required for bias calculation, but error bands on plots come directly from RooFit

Bias

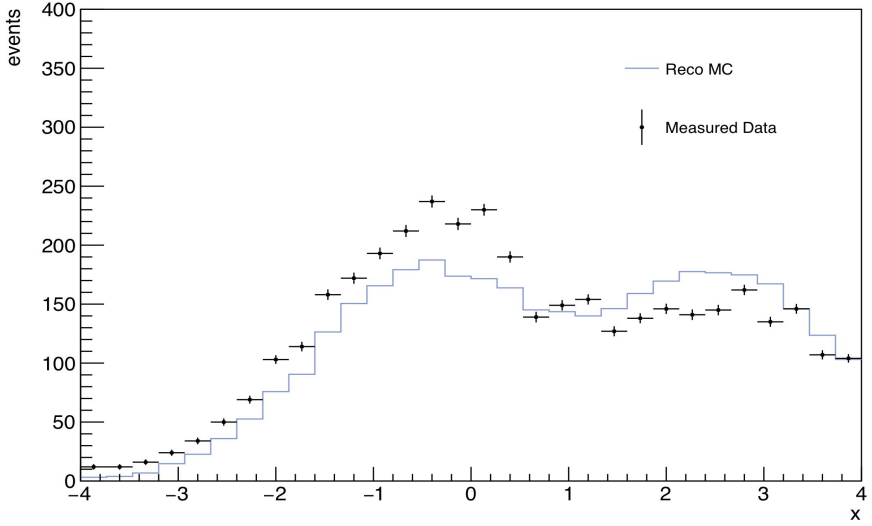
Two bias calculations included
Bias estimate without toys and a full bias calculation

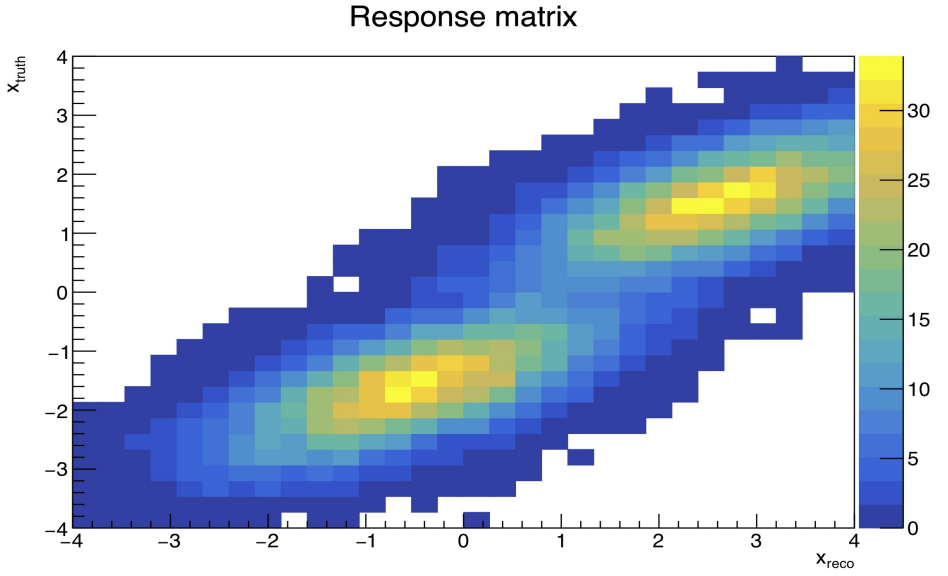




Example: Bimodal distribution

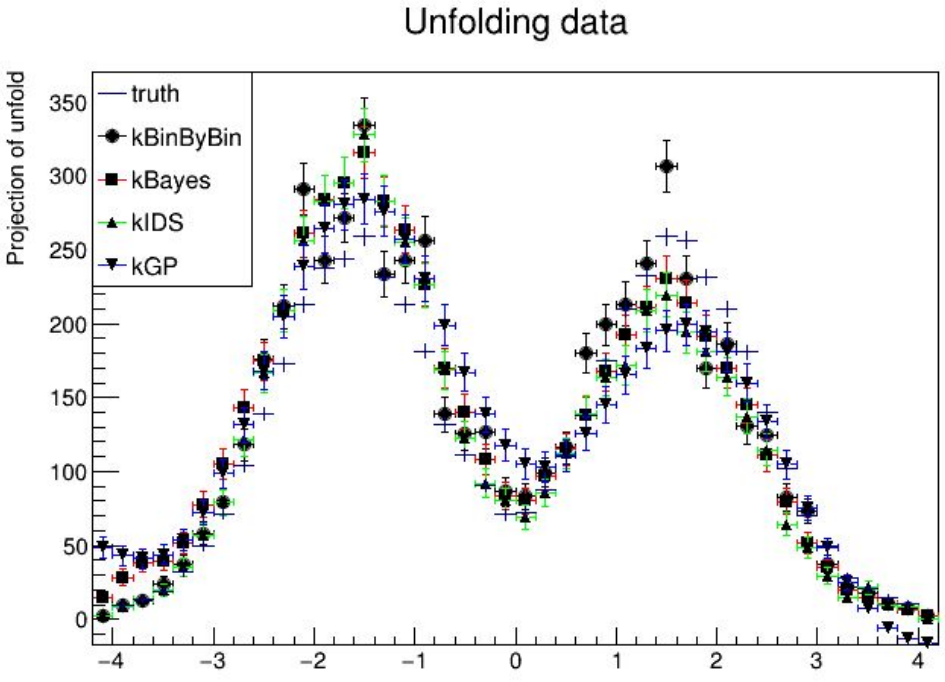
Reconstruction level plots





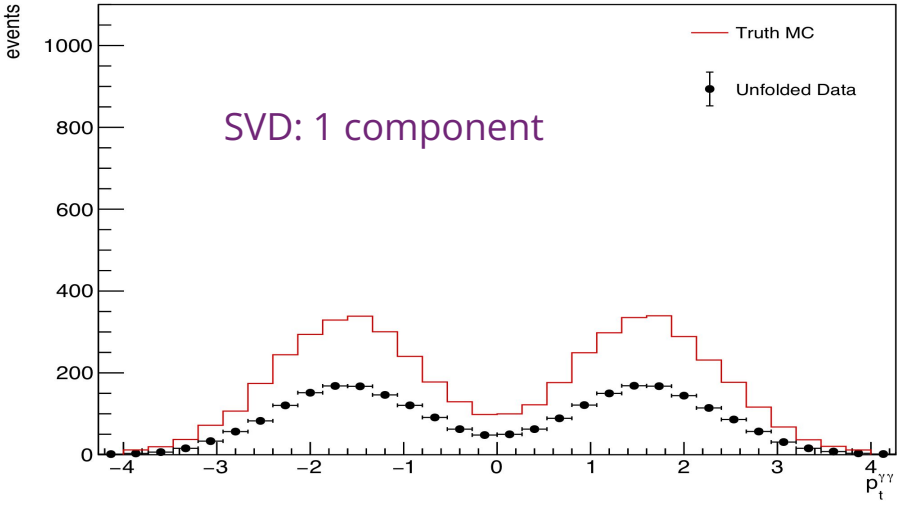
Example: Bimodal distribution

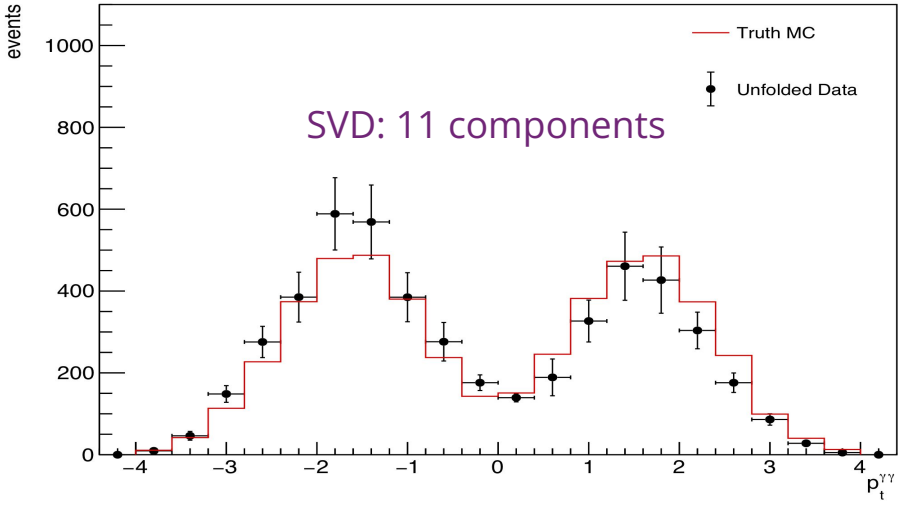
Compare different unfolding methods



Example: Bimodal distribution

Compare different regularisation strengths

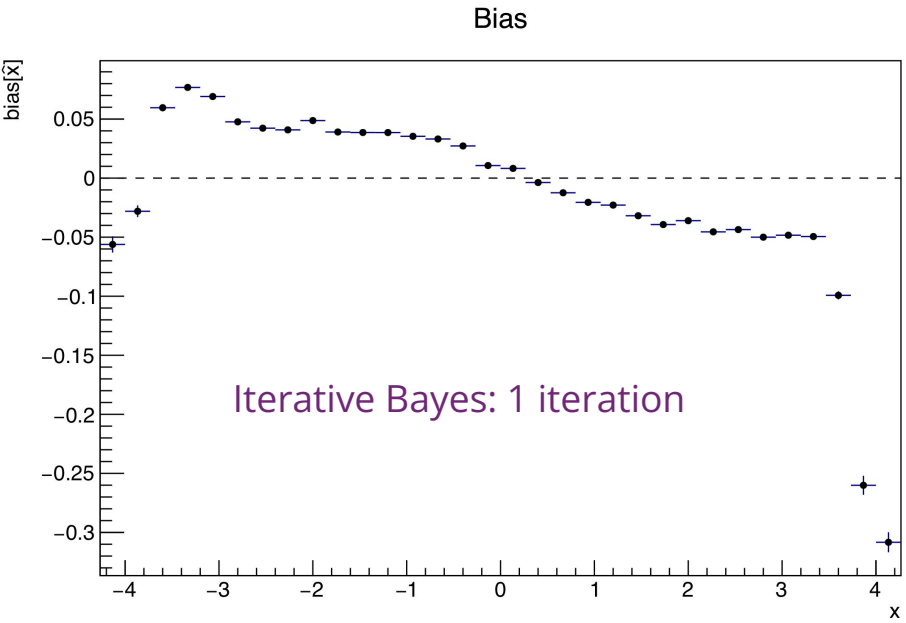


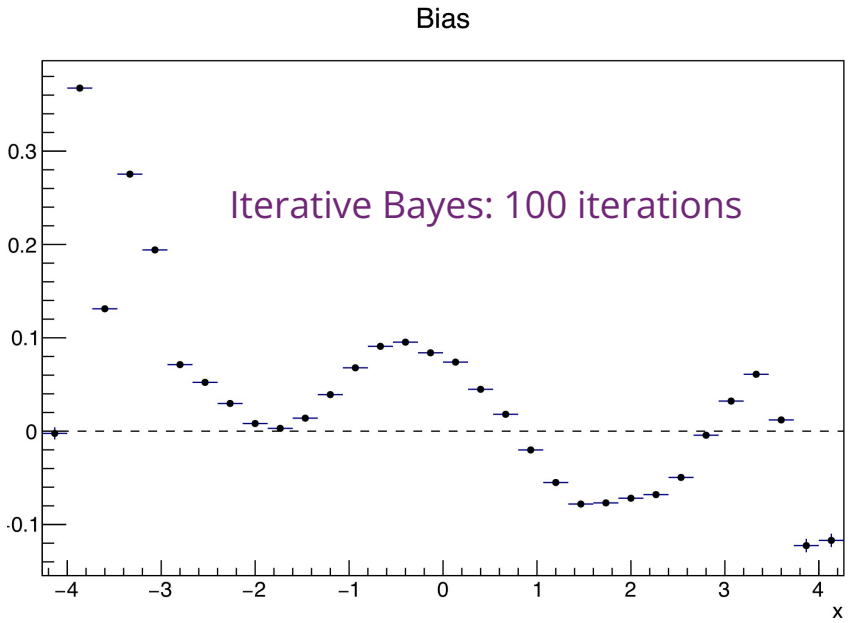


Nikhef

Example: Bimodal distribution

Compare different regularisation strengths: Don't forget the bias!





Bias calculation

(Asimov) data-driven

First the uncertainties are taken truth level from the unfolded Asimov dataset. Toys are thrown in each bin around the Asimov truth values based on the full uncertainty. These toys are called level 1 toys. For each level 1 toy further toys are thrown, called level 2 toys. Each of the level 2 toys is folded and then unfolded with the chosen unfolding method. The bias for each level 2 toy is calculated as

$$\text{biasl2} = (\sigma\text{refold} - \sigma\text{truth})/\sigma\text{truth},$$

where the truth refers to the value of the level 1 toy at truth level from which the level 2 toy is thrown and refold refers to the value of the level 2 toy after folding and unfolding. The bias of each bin in the distribution that is being unfolding is the average over all biasl2.