Unfolding with the TUnfold Software and its Application within the H1 collaboration

Alliance Workshop on Unfolding and Data Correction
May 2010

Physikalisches Institut Heidelberg

Michel Sauter

Introduction

Idea of Unfolding:

- Relate measured quantities with 'true' quantities. This Relation is often approximated via a matrix equation:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} \qquad (1)$$

A: ***Response matrix, determined from simulation. Incorporates detector response and signal/background models.***
y: ***Reconstructed vector (measured)***
b: ***Background vector***
x: ***True vector, to be determined***

The TUnfold software package:

- With the TUnfold software package one can solve equation (1) with **regularized unfolding**:
  - Determine an estimator for **x.**
  - Apply side conditions on smoothness and normalization of **x.**
- TUnfold are two C++ classes, written by Stefan Schmitt.
- A pre-version of the package is implemented in RooT-5.22. (not recommended)
- The latest version of TUnfold will be implemented in RooT-5.27 or can be found here: http://www.desy.de/~sschmitt/ (recommended)

Outline of this talk

- Implementation of TUnfold
- Application of TUnfold within the H1-collaboration

Implementation

		dimsnsion
A:	***Response matrix.***	***ny x nx***
y:	***Reconstructed vector (measured)***	***ny***
b:	***Background vector***	***ny***
x:	***True, unknown vector***	***nx***
V:	***Covariance matrix for y***	***ny x ny***
n_{obs} :	***number of observed events***	

TUnfold solves the inverse problem of

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} \qquad (1)$$

by minimizing the following X^2–function:

$$\chi^2(\hat{\mathbf{x}}, \tau, \lambda) := \chi^2_A(\hat{\mathbf{x}}) + \tau^2 \cdot \chi^2_L(\hat{\mathbf{x}}) + \lambda \cdot \chi^2_N(\hat{\mathbf{x}}) \qquad (2)$$

$$\chi^2_A(\hat{\mathbf{x}}) := 1/2\,(\mathbf{y} - \mathbf{b} - \mathbf{A}\hat{\mathbf{x}})^T\, V^{-1}\,(\mathbf{y} - \mathbf{b} - \mathbf{A}\hat{\mathbf{x}})$$

← standard X^2

$$\chi^2_L(\hat{\mathbf{x}}) := \hat{\mathbf{x}}^T \mathbf{L}^T \mathbf{L}\hat{\mathbf{x}}$$

← smoothness / regularization condition

$$\chi^2_N(\hat{\mathbf{x}}) := \left(n_{obs} - \sum_{j=1}^{n_y} (\mathbf{A}\hat{\mathbf{x}})_j \right)^2$$

← constraint on normalization (optional)

For given regularization condition (i.e. given τ and L) equation (2) is analytically solved for x. No iteration is performed.

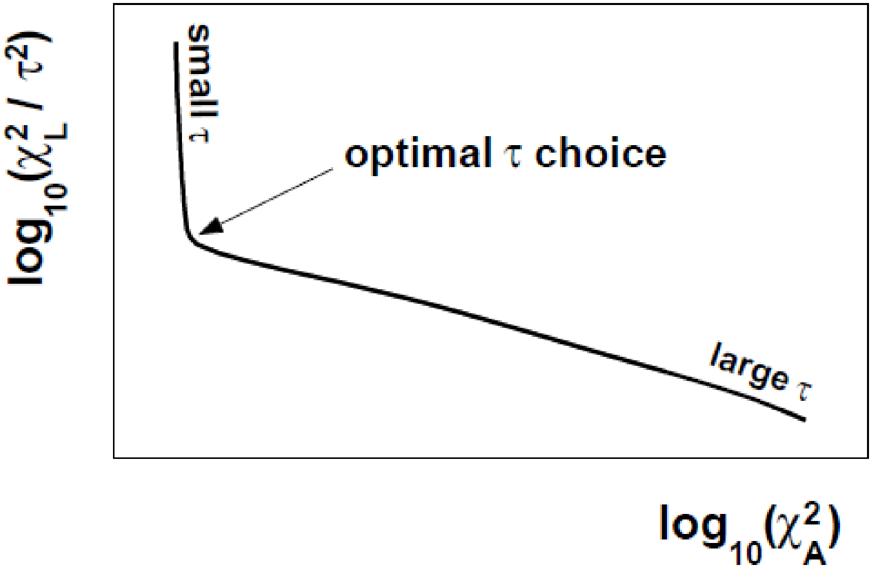
The algorithm is based on inverted sparse matrices, with the known limitations in numerical accuracy and computing cost for matrices with large dimensions.

Regularization Condition

- The regularization conditions are defined by the term $\tau^2 \cdot \chi^2_L(\hat{\mathbf{x}})$, with
  $\chi^2_L(\hat{\mathbf{x}}) := \hat{\mathbf{x}}^T \mathbf{L}^T \mathbf{L} \hat{\mathbf{x}}$

- The matrix L defines the regularization condition. The user has full flexibility to define L, however predefined functions to regularize on the **size**, the **derivative** or the **curvature** exist.

- The regularization parameter τ defines the 'strength of regularization'. It can either be defined by the user or by an automatic L–Curve scan implemented within the package.
  P. C. Hansen, “The L–Curve and its Use in the Numerical Treatment of Inverse Problems", pages 119{142, 2000.



Constraint on Normalization / on the Area

- The optional term $\lambda \cdot \chi^2_N(\hat{\mathbf{x}})$ fixes the number of estimated events to the number of observed events, whereas

$$\chi^2_N(\hat{\mathbf{x}}) := \left(n_{obs} - \sum_{j=1}^{n_y} (\mathbf{A}\hat{\mathbf{x}})_j \right)^2 , \qquad n_{obs} := \sum_{j=1}^{n_y} \mathbf{y_j}$$

- This constraint is important in the case of a reconstructed vector **y** having non–Gaussian errors.

- If **y** is derived from a histogram (Poisson errors), the unfolded result **x** will be biased, if no such constraint is applied.

- The reason for this behaviour: The underlying the X^2–function which is minimized within TUnfold. (A X^2–function is only unbiased in the case of Gaussian errors.)

- In the literature this aspect of estimators based on X^2–functions is often discussed for fits to histograms. The used term is "Preserving the area problem".

  See for instance:
  - Clarification on the use of chi–square and likelihood functions in fits to histograms", S. Baker, D. Cousins, Nucl. Inst. and Meth.221, (1984,) 437–442
  - G. Cowan, Statistical Data Analysis. p. , 100–103, Oxford University Press, 1998.

Error Treatment within TUnfold

Statistical errors:

- The errors of the input vector **y** are directly propagated to the result and are provided in the form of an error matrix.

Systematical errors:

- TUnfold also provides functions to treat systematical errors on
  - … the regularization parameter τ^2.
  - … the subtracted background.
  - … the response matrix **A.** The uncertainty of each element of **A** is propagated analytically to the result using standard error propagation.
- Several systematical error sources (in particular on the response matrix **A**) can be treated and the error matrix of each source can be accessed separately.

Further Aspects of TUnfold:

- Simple to use.
- Fast and flexible program.
- Offers many functions to interpret the result, for instance:
  - Correlations and average correlations
  - Folded unfolding result
  - X^2_A, X^2_L , NDF, etc.
- 4 tutorial demonstrating different aspects of unfolding.

Code excerpt of Tutorial 2:

```
//Define response histogram
TH2D *histMdetGenMC=new TH2D("MdetgenMC",";mass(det);mass(gen)",nDet,xminDet,xmaxDet,
                             nGen,xminGen,xmaxGen);
…

//TUnfold constructor
TUnfold unfold(histMdetGenMC,TUnfold::kHistMapOutputVert, TUnfold::kRegModeNone);

//this method scans the parameter tau and finds the kink in the L curve
// finally, the unfolding is done for the "best" choice of tau
unfold.ScanLcurve(nScan,tauMin,tauMax,&lCurve,&logTauX,&logTauY);

// extract unfolding results into histograms
TH1D *histMunfold=new TH1D("Unfolded",";mass(gen)",nGen,xminGen,xmaxGen);
unfold.GetOutput(histMunfold);
…
```

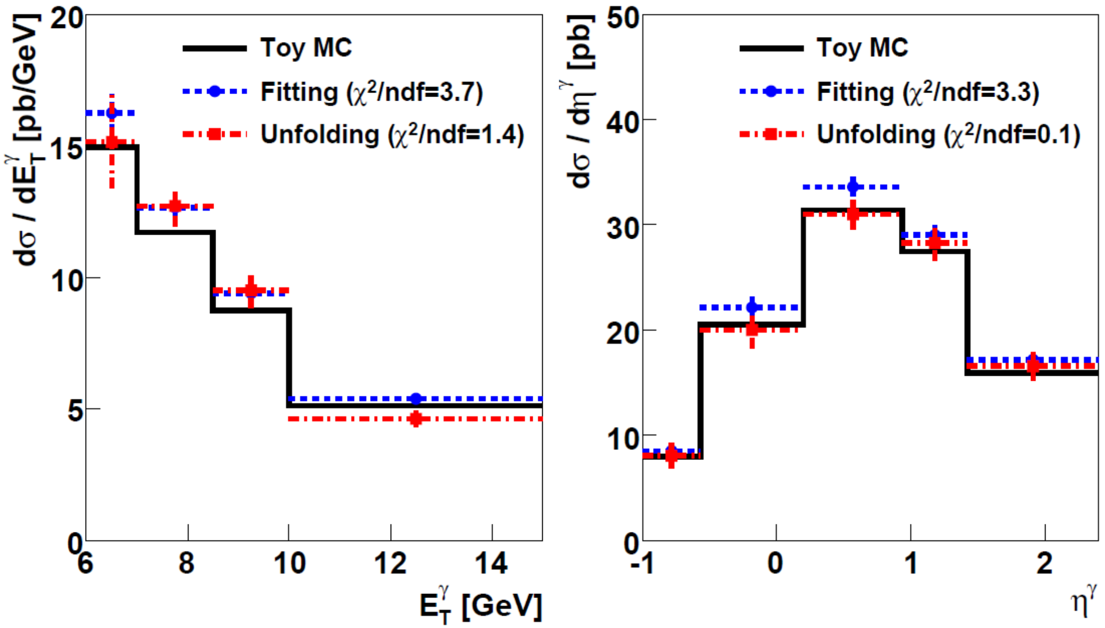
Application of TUnfold within the H1 Collaboration

- Prompt Photon Production in Photoproduction
  Krzysztof Nowak, Unfolding of x(γ), x(p), E_T(jet) in bins of η(γ) and E_T(γ), determination of background fractions, DESY–THESIS–2010–011 and H1 Collab., F.D. Aaron et al., Published in Eur.Phys.J. C66 (2010) 17.
- Multiple parton interactions in photoproduction
  Lluis Marti, unfloding of p_T jet spectra, DESY–THESIS–2009–007.
- Measurement of the D* meson cross section in DIS
  Andreas Jung, unfolding of p_T(D*) and η(D*), Gaussian errors due to fits, DESY–THESIS–2009–001.
- Exclusive diffractive J/ψ photoproduction at low $W_{\gamma p}$
  Florian Huber, unfolding of t and $W_{\gamma p}$ spectra, separation of pdis and elastic events, Gaussian errors.
- Inclusive jets at high Q^2
  Daniel Britzger, Roman Kogler, jet cross sections as a function of Q^2 and p_T^{jet} .
- F2–structure function
  David–Johannes Fischer, 2–dim unfolding of F2 in y and Q^2.
- Beauty photoproduction at threshold
  Michel Sauter, unfolding of the p_T(b) spectrum, determination of charm and uds background fractions, DESY–THESIS–2009–047.

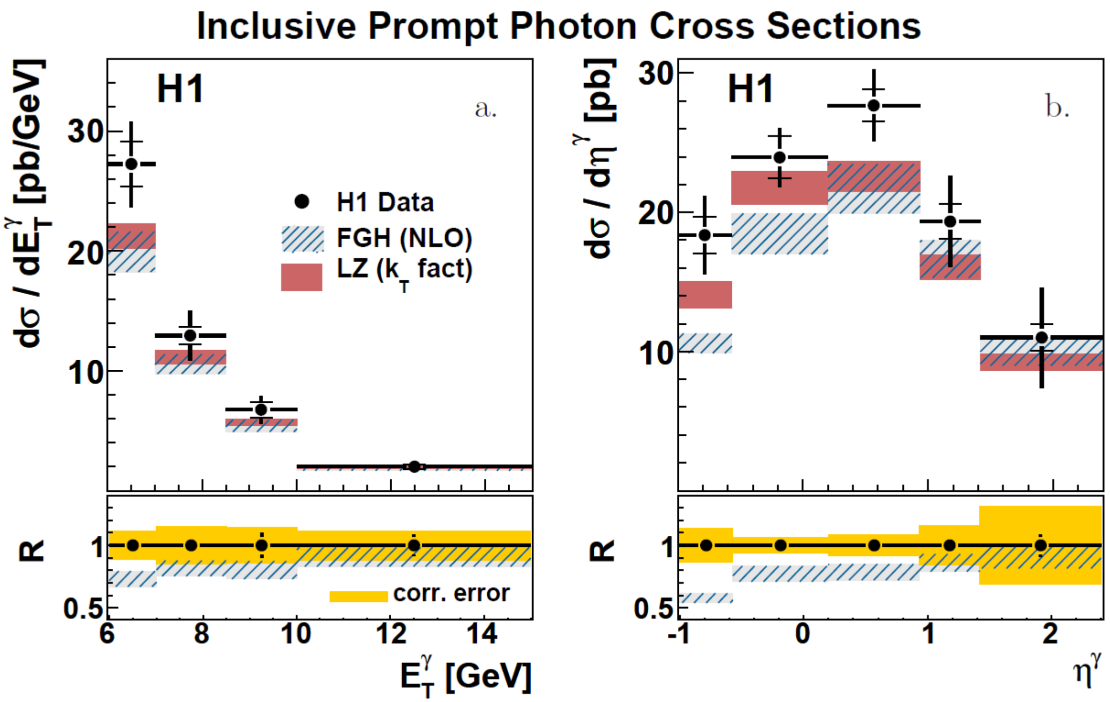
Bin–to–Bin Correction Compared to Unfolding

- Toy MC study

fitted = bin–to–bin correction.



- Data (unfolded)



Plots: taken from Krzysztof Nowak's thesis

Experience with TUnfold I

- How difficult was it get started?
  - “Once you have the migration matrix it is easy to implement and run.”
  - “TUnfold works great and is easy to use.”
  - “Very easy … actually too easy for the complexity of the problem.”
  - “TUnfold itself is working pretty fine and has a comprehensive functionality … on the other side, there is no dedicated documentation … “

- What is your overall experience with unfolding?
  - “Impression of unfolding in general: It clearly makes a difference and it seems to be the right way to treat migration effects. It seems to be full of pitfalls through – it is clearly not be "black box" that just works. Care is needed. Impression of TUnfold: Greeat and easy to use.”
  - “About unfolding: It's working fine, but one can cheat as you like. After month's of working on unfolding, we still do not consider our result being a truth 'unfolding' in the means of just an unfolding of the detector effects.”
  - “ … unfolding is tricky. Specially if you look at small changes … ”
  - “Unfolding is difficult to control and difficult to implement, one needs to know what he is doing to get reasonable results. Bin-to-bin correction works easier out-of-box. But apparently unfolding is safer if correctly implemented. ”

Experience with TUnfold II

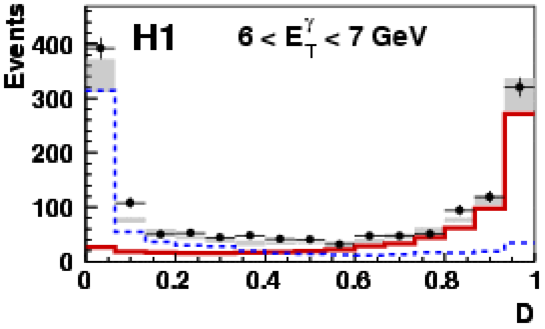
- Do you also use other unfolding programs? – If yes, what are the advantages / disadvantages of TUnfold / the other programs?
  - “The problem with one was that was very slow, with the other that was supporting only quadratic matrices. ”
  - “No, we considered very shortly the usage of RooUnfold. But actually, we only use TUnfold.”
  - “I started using RooUnfold for the Bayes unfolding because that translation is not easy to implement and perform changes. I hope RooUnfold provides me with the same quality of results while being easier to handle.”

- What literature concerning unfolding do you read?
  - G. Cowan, Statistical Data Analysis, Oxford University Press, 1998.
  - V. Blobel und E. Lohrmann, Statistische und numerische Methoden der Datenanalyse, Teubner Studienbücher, B.G. Teubner, Stuttgart, 1998.
  - V. Blobel website + Talks, http://www.desy.de/~blobel/unfold.html
  - Theses

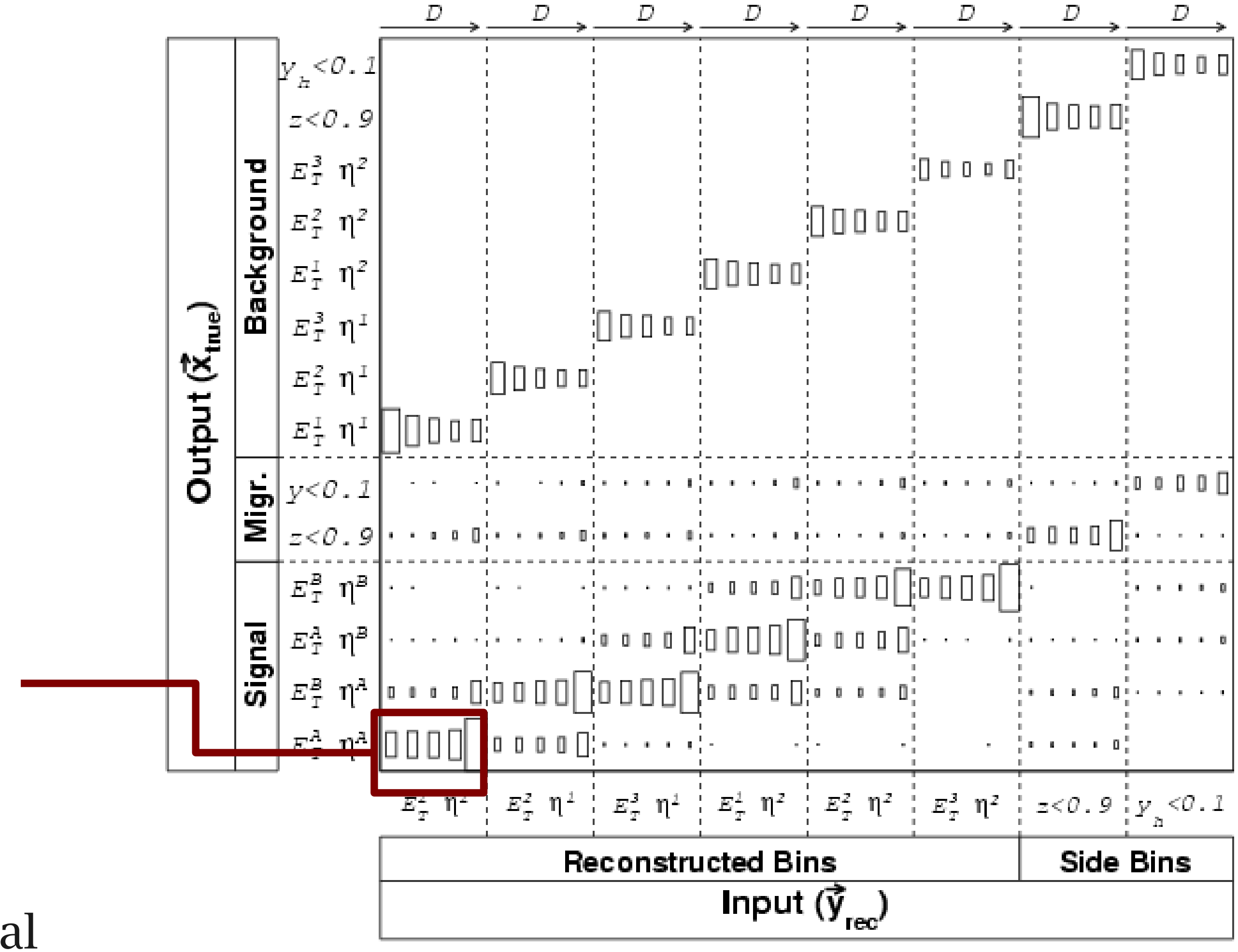
Example for a Sophisticated Response Matrix I

– “Once you have the migration matrix it is easy to implement and run.”

Discriminator



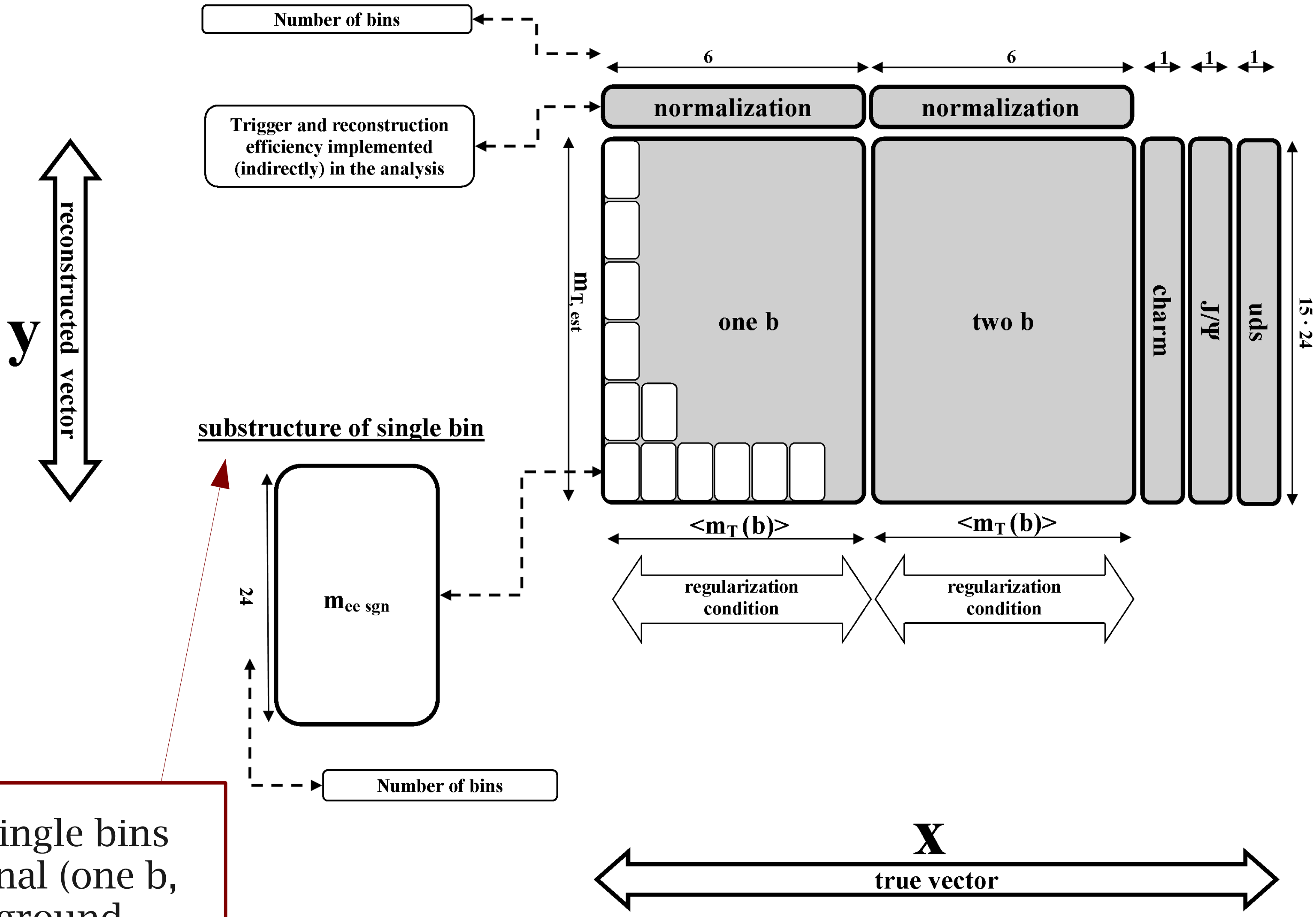
Background Signal



Prompt Photon Production in Photoproduction,
Krzysztof Nowak, DESY-THESIS-2010-011

Example for a Sophisticated Response Matrix II

– “Once you have the migration matrix it is easy to implement and run.”



Substructure of single bins discriminates signal (one b, two b) from background (charm, J/ψ, uds)

Beauty photoproduction at threshold
DESY–THESIS–2009–04

Summary

- TUnfold:
  - Simple, fast and flexible unfolding software.
  - Provides treatment of systematical errors.
  - Provides lots of unfolding related functions.
  - Implemented in RooT

    (but consult http://www.desy.de/~sschmitt/ for the latest version)

- Unfolding in general:
  - Even with good software not as simple, as one might naively expect at the beginning.

Thanks Stefan, well done!

Stefan Schmitt,
(author of TUnfold)