HistFactory: A tool for creating statistical models for use with RooFit and RooStats

Kyle Cranmer, George Lewis, Lorenzo Moneta, Akira Shibata, Wouter Verkerke

August 2, 2012

Contents

1 Introduction

The HistFactory is a tool to build parametrized probability density functions (pdfs) in the RooFit/RooStats framework based based on simple ROOT histograms organized in an XML file. The pdf has a restricted form, but it is sufficiently flexible to describe many analyses based on template histograms. The tool takes a modular approach to build complex pdfs from more primative conceptual building blocks. The resulting PDF is stored in a RooWorkspace which can be saved to and read from a ROOT file. This document describes the defaults and interface in HistFactory 5.32. Note, HistFactory 5.34 provides a C++ and python interface fully interoperable with the XML interface and classes for analytically fitting bin-by-bin statistical uncertainties on the templates. These developments will be included in a future version of this document.

1.1 Preliminaries

Let us begin by considering the simple case of a single channel with one signal and one background contribution and no systematics based on the discriminating variable is x. While we will not continue with this notation, let us start with the familiar convention where the number of signal events is denoted as S and the number of background events as B. Similarly, denote the signal and background "shapes" as $f_{\rm S}(x)$ and $f_{\rm B}(x)$ and note the these are probability density functions normalized so that $\int dx f(x) = 1$. It is common to introduce a "signal strength" parameter μ such that $\mu = 0$ corresponds to the background-only hypothesis and $\mu = 1$ corresponds to the nominal signal+background hypothesis. This continuous parameter μ is our parameter of interest.

Now we ask what the probability model is for obtaining n events in the data where the discriminating variable for event e has a value x_e; thus the full dataset will be denoted $\{x_1 \dots x_n\}$. First one must include the Poisson probability of obtaining n events when $\mu S+B$ are expected. Secondly, one must take into account the probability density of obtaining x_e based on the relative mixture $f_{\rm S}(x)$ and $f_{\rm B}(x)$ for a given value of μ. Putting those two ingredients together one obtains what statisticians call a "marked Poisson model":

$$\mathcal{P}(\{x_1 \dots x_n\}|\mu) = {\rm Pois}(n|\mu S + B) \left[\prod_{e=1}^{n} \frac{\mu S f_{\rm S}(x_e) + B f_{\rm B}(x_e)}{\mu S + B} \right] \, . \tag{1}$$

If one imagines the data as being fixed, then this equation depends on μ and is called the likelihood function $L(\mu)$. Simply taking the logarithm of the equation above and remembering that ${\rm Pois}(n|\nu) = \nu^n e^{-\nu}/n!$ gives us a familiar formula referred to by physicists as an "extended maximum likelihood fit" :

$$\begin{aligned} -\ln L(\mu) &= -n \ln(\mu S + B) + (\mu S + B) + \ln n! - \sum_{e=1}^{n} \ln \left[\frac{\mu S f_{\rm S}(x_e) + B f_{\rm B}(x_e)}{\mu S + B} \right] \\ &= (\mu S + B) + \ln n! - \sum_{e=1}^{n} \ln \left[\mu S f_{\rm S}(x_e) + B f_{\rm B}(x_e) \right] \, . \end{aligned} \tag{2}$$

Since HistFactory is based on histograms, it is natural to think of the binned equivalent of the probability model above. Denoted the signal and background histograms as $\nu_b^{\rm sig}$ and

$\nu_b^{\rm bkg}$, where b is the bin index and the histograms contents correspond to the number of events expected in the data. We can relate the bin ν_b and the shape $f(x)$ via

$$f_S(x_e) = \frac{\nu_{b_e}^{\rm sig}}{S\Delta_{b_e}} \qquad \text{and} \qquad f_B(x_e) = \frac{\nu_{b_e}^{\rm bkg}}{B\Delta_{b_e}} \;, \tag{3}$$

where b_e is the index of the bin containing x_e and Δ_{b_e} is the width of that same bins. Note, because the $f(x)$ are normalized to unity we have $S = \sum_b \nu_{\rm b}^{\rm sig}$ and $B = \sum_b \nu_{\rm b}^{\rm bkg}$.

Formally one can either write the probability model in terms of a product over Poisson distributions for each bin of the histogram, or one can also continue to use the unbinned expression above recognizing that the shapes $f(x)$ look like histograms (ie. they are discontinuous at the bin boundaries and constant between them). Technically, the `HistFactory` makes a model that looks more like the unbinned expression with a single RooAbsPdf that is "extended" with a discontinuous shape in x. Nevertheless, it can be more convenient to express the model in terms of the individual bins. Then we have

$$\mathcal{P}(n_b|\mu) = \text{Pois}(n_{\rm tot}|\mu S + B)\left[\prod_{b\in \rm bins} \frac{\mu\nu_b^{\rm sig} + \nu_b^{\rm bkg}}{\mu S + B}\right] = \mathcal{N}_{\rm comb} \prod_{b\in \rm bins} \text{Pois}(n_b|\mu\nu_b^{\rm sig} + \nu_b^{\rm bkg}) \;, \tag{4}$$

where n_b is the data histogram and $\mathcal{N}_{\rm comb}$ is a combinatorial factor that can be neglected since it is constant. Similarly, denote the data histogram is n_b.

1.2 Generalizations and Use-Cases

Based on the discussion above, we want to generalize the model in the following ways:

- Ability to include multiple signal and background samples
- Ability to include unconstrained scaling of the normalization of any sample (as was done with μ)
- Ability to parametrize variation in the normalization of any sample due to some systematic effect
- Ability to parameterize variations in the shape of any sample due to some systematic effect
- Ability to include bin-by-bin statistical uncertainty on the normalization of any sample
- Ability to incorporate an arbitrary contribution where each bin's content is parametrized individually
- Ability to combine multiple channels (regions of the data defined by disjoint event selections) and correlate the parameters across the various channels
- Ability to use the combination infrastructure to incorporate control samples for data-driven background estimation techniques
- Ability to reparametrize the model

	Constrained	Unconstrained
Normalization Variation	`OverallSys` (η_{cs})	`NormFactor` (ϕ_p)
Coherent Shape Variation	`HistoSys` σ_{csb}	–
Bin-by-bin variation	`ShapeSys` & `StatError` γ_{cb}	`ShapeFactor` γ_{csb}

Table 1: Conceptual building blocks for constructing more complicated PDFs: parameters.

2 The Likelihood Template

2.1 Index Convention

In what follows we use the term *channel* as a region of the data defined by the corresponding event selection, as opposed to a particular scattering process. The *channels* are required to have disjoint event selection requirements. We use the term *sample* for a set of scattering processes that can be added together incoherently; thus scattering processes that interfere quantum mechanically must be considered in the same sample.

We will use the following mnemonic index conventions:

- $e \in$ events
- $b \in$ bins
- $c \in$ channels
- $s \in$ samples
- $p \in$ parameters

We define the following subsets of parameters $\mathbb{N} = \{\phi_p\}$ the unconstrained normalization factors (ie. `NormFactor`), $\mathbb{S} = \{\alpha_p\}$ the parameters associated to systematic that have external constraints (ie. `OverallSys` and `HistoSys`), $\mathbf{\Gamma} = \{\gamma_{csb}\}$ (the bin-by-bin uncertainties with constraints (statistical errors, `ShapeSys` but *not* those associated to an unconstrained `ShapeFactor`). We also use greek symbols for parameters of the model and roman symbols for observable quantities with a frequentist notion of probability.

2.2 The Template

The parametrized probability density function constructed by the `HistFactory` is of a concrete form, but sufficiently flexible to describe many analyses based on template histograms. In general, the `HistFactory` produces probability density functions of the form

$$\mathcal{P}(n_c, x_e, a_p \mid \phi_p, \alpha_p, \gamma_b) = \prod_{c\in\text{channels}} \left[\text{Pois}(n_c|\nu_c) \prod_{e=1}^{n_c} f_c(x_e|\boldsymbol{\alpha}) \right] \cdot G(L_0|\lambda, \Delta_L) \cdot \prod_{p\in\mathbb{S}+\mathbf{\Gamma}} f_p(a_p|\alpha_p) \quad (5)$$

where $f_p(a_p|\alpha_p)$ is a constraint term describing an auxiliary measurement a_p that constrains the nuisance parameter α_p (see Section 4.2). Denote the bin containing x_e as b_e. We have the following expression for the expected (mean) number of events in a given bin

$$\nu_{cb}(\phi_p, \alpha_p, \gamma_b) = \lambda_{cs}\, \gamma_{cb}\, \phi_{cs}(\boldsymbol{\alpha})\, \eta_{cs}(\boldsymbol{\alpha})\, \sigma_{csb}(\boldsymbol{\alpha})\,, \quad (6)$$

where the meaning of the various terms is described below and the specific interpolation algorithms are described in Section 4.1. The mean number of events in each bin implies the following probability density

$$f_c(x_e|\phi_p, \alpha_p, \gamma_b) = \frac{\nu_{cb_e}}{\nu_c} \qquad \text{with} \qquad \nu_c = \sum_{b \in \text{bins of channel } c} \nu_{cb} \tag{7}$$

It is perhaps more convenient to think of the likelihood as a product over bins

$$\mathcal{P}(n_{cb}, a_p \mid \phi_p, \alpha_p, \gamma_b) = \prod_{c \in \text{channels}} \prod_{b \in \text{bins}} \text{Pois}(n_{cb}|\nu_{cb}) \cdot G(L_0|\lambda, \Delta_L) \cdot \prod_{p \in \mathbb{S}+\mathbf{\Gamma}} f_p(a_p|\alpha_p)$$

- λ_{cs} - luminosity parameter for a given channel and sample. Within a given channel this parameter is a common luminosity parameter for all the samples that include luminosity uncertainty (i.e.. `NormalizeByTheory="True"`). For all the samples with `NormalizeByTheory="False"` it is fixed to the nominal luminosity $\lambda_{cs} = L_0$.

- γ_{cb_e} - Bin-by-bin scale factor used for statistical uncertainties, bin-by-bin shape systematics (`ShapeSys`), and data-driven shape extrapolations (`ShapeFactor`). For statistical errors, the γ_{csb_e} is shared for all the samples in the channel (ie. subscript s can be omitted). For samples that do not have any bin-by-bin scale factors $\gamma_{csb_e} = 1$.

- ϕ_{cs} - Product of unconstrained normalization factors for a given sample within a given channel. These typically include the parameter of interest, eg. the signal cross-section or branching ratio.

$$\phi_{cs} = \prod_{p \in \mathbb{N}_c} \phi_p \tag{8}$$

- $\eta_{cs}(\boldsymbol{\alpha})$ - The parametrized normalization uncertainties (ie. `OverallSys`) for a given sample within a given channel (a factor around 1).

- σ_{csb_e} - The parametrized histogram (ie. the nominal histogram and the `HistoSys`) for a given sample within a given channel.

2.2.1 Incorporating Monte Carlo statistical uncertainty on the histogram templates

The histogram based approach described above are based Monte Carlo simulations of full detector simulation. These simulations are very computationally intensive and often the histograms are sparsely populated. In this case the histograms are not good descriptions of the underlying distribution, but are estimates of that distribution with some statistical uncertainty. Barlow and Beeston outlined a treatment of this situation in which each bin of each sample is given a nuisance parameter for the true rate, which is then fit using both the data measurement and the Monte Carlo estimate [?]. This approach would lead to several hundred nuisance parameters in the current analysis. Instead, the `HistFactory` employs a lighter weight version in which there is only one nuisance parameter per bin associated with the total Monte Carlo estimate and the total statistical uncertainty in that bin. If we focus on an individual bin with index b the contribution to the full statistical model is the factor

$$\text{Pois}(n_b|\nu_b(\boldsymbol{\alpha}) + \gamma_b \nu_b^{\text{MC}}(\boldsymbol{\alpha}))\, \text{Pois}(m_b|\gamma_b \tau_b)\ , \tag{9}$$

where n_b is the number of events observed in the bin, $\nu_b(\boldsymbol{\alpha})$ is the number of events expected in the bin where Monte Carlo statistical uncertainties need not be included (either because the estimate is data driven or because the Monte Carlo sample is sufficiently large), $\nu_b^{\rm MC}(\boldsymbol{\alpha})$ is the number of events estimated using Monte Carlo techniques where the statistical uncertainty needs to be taken into account. Both expectations include the dependence on the parameters $\boldsymbol{\alpha}$. The factor γ_b is the nuisance parameter reflecting that the true rate may differ from the Monte Carlo estimate $\nu_b^{\rm MC}(\boldsymbol{\alpha})$ by some amount. If the total statistical uncertainty is δ_b, then the relative statistical uncertainty is given by $\nu_b^{\rm MC}/\delta_b$. This corresponds to a total Monte Carlo sample in that bin of size $m_b = (\delta_b/\nu_b^{\rm MC})^2$. Treating the Monte Carlo estimate as an auxiliary measurement, we arrive at a Poisson constraint term $\mathrm{Pois}(m_b|\gamma_b\tau_b)$, where m_b would fluctuate about $\gamma_b\tau_b$ if we generated a new Monte Carlo sample. Since we have scaled γ to be a factor about 1, then we also have $\tau_b = (\nu_b^{\rm MC}/\delta_b)^2$; however, τ_b is treated as a fixed constant and does not fluctuate when generating ensembles of pseudo-experiments.

It is worth noting that the conditional maximum likelihood estimate $\hat{\hat{\gamma}}_b(\boldsymbol{\alpha})$ can be solved analytically with a simple quadratic expression.

$$\hat{\hat{\gamma}}_b(\boldsymbol{\alpha}) = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \ , \tag{10}$$

with

$$A = \nu_b^{\rm MC}(\boldsymbol{\alpha})^2 + \tau_b \nu_b^{\rm MC}(\boldsymbol{\alpha}) \tag{11}$$

$$B = \nu_b(\boldsymbol{\alpha})\tau + \nu_b(\boldsymbol{\alpha})\nu_b^{\rm MC}(\boldsymbol{\alpha}) - n_b\nu_b^{\rm MC}(\boldsymbol{\alpha}) - m_b\nu_b^{\rm MC}(\boldsymbol{\alpha}) \tag{12}$$

$$C = -m_b\nu_b(\boldsymbol{\alpha}) \ . \tag{13}$$

In a Bayesian technique with a flat prior on γ_b, the posterior distribution is a gamma distribution. Similarly, the distribution of $\hat{\gamma}_b$ will take on a skew distribution with an envelope similar to the gamma distribution, but with features reflecting the discrete values of m_b. Because the maximum likelihood estimate of γ_b will also depend on n_b and $\hat{\boldsymbol{\alpha}}$, the features from the discrete values of m_b will be smeared. This effect will be more noticeable for large statistical uncertainties where τ_b is small and the distribution of $\hat{\gamma}_b$ will have several small peaks. For smaller statistical uncertainties where τ_b is large the distribution of $\hat{\gamma}_b$ will be approximately Gaussian.

3 Using HistFactory

3.1 The HistFactory XML

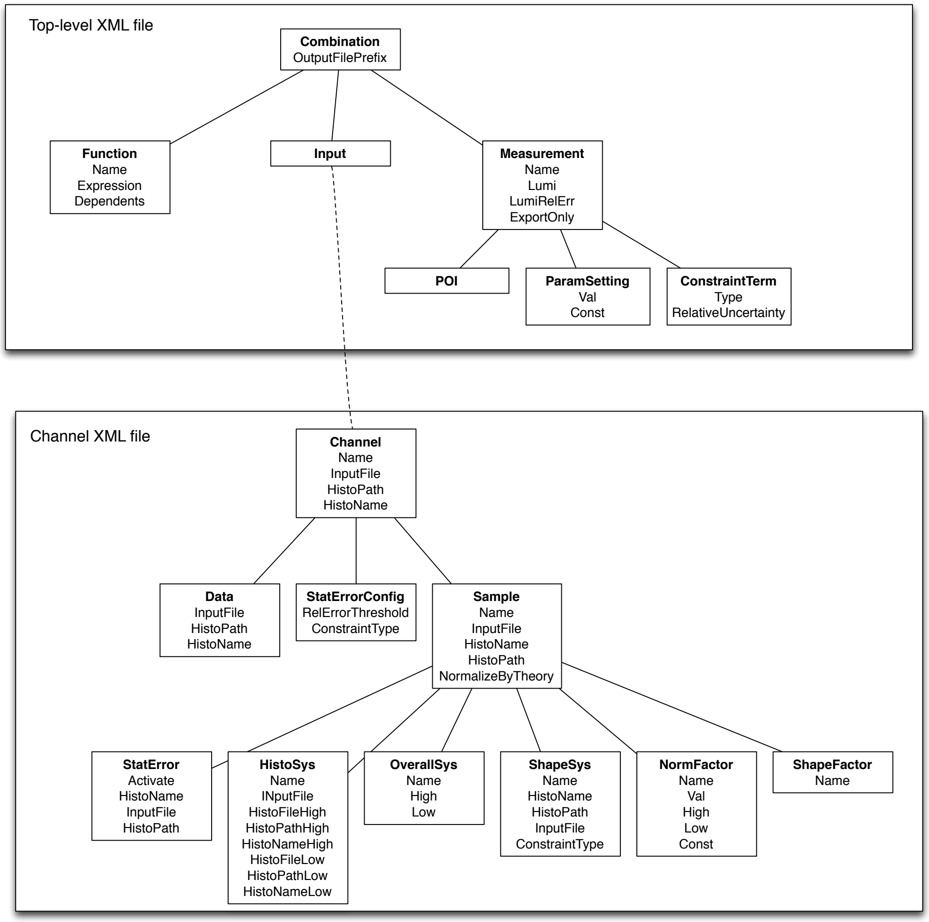


Figure 1: The graphical representation of the XML schema. Boxes are elements with name of element in bold and attributes listed below. Elements without any attributes take their input as "PCDATA", for example `<Input>someFile.xml</Input>`

Note, when using the `HistFactory` the production modes l and backgrounds j correspond to a single XML `Sample` element. The `HistoName` attribute inside each sample element specifies the histogram with the σ^0_{ism}. The index $s =' S$' is set by the `Name` attribute of the `Sample` element (eg. `<Sample Name="S">`). Between the open `<Sample>` and close `</Sample>` one can add

- An `OverallSys` element where the `Name="p"` attribute identifies which α_p is the source of the systematic and implies that the Gaussian constraint $f_p(a_p|\alpha_p)$ is present. The `High` attribute corresponds to η^+_{ps}, eg when the source of the systematic is at $+1\sigma$ and $\alpha_p = 1$. Similarly, the `Low` attribute corresponds to η^-_{ps}, eg when the source of the systematic is at -1σ and $\alpha_p = -1$. The nominal value is $\eta^0_{ps} = 1$ for the overall systematics. The distinction between the sign of the source α and the effect η allows one to have anti-correlated systematics. The `HistFactory` is able to deal with asymmetric uncertainties as well, by using a one of various interpolations.

- A `NormFactor` element is used to introduce an overall constant factor into the expected number of events. In the example below, the term $\mu = \sigma/\sigma_{SM}$ corresponds to the line `<NormFactor Name="SigXsecOverSM">`. In this case, the histograms were normalized to unity, so additional `NormFactor` elements were used to give the overall cross-sections σ_s.
- A `HistoSys` element is used to introduce shape systematics and the `HistoNameHigh` and `HistoNameLow` attributes have the variational histograms σ^+_{psb} and σ^-_{psb} corresponding to $\alpha_p = +1$ and $\alpha_p = -1$, respectively.

3.2 Normalization Conventions

The nominal and variational histograms should all have the same normalization convention. There are a few conventions possible:

Option 1:

- `Lumi="XXX"` in the measurement XML's element, where XXX is in fb^{-1}
- Histograms bins have units of fb
- Some samples have `NormFactor` that are all relative to prediction (eg. 1 is the nominal prediction)

Option 2:

- `Lumi="1."` in the measurement XML's element
- Histograms are normalized to unity
- Each sample has a `NormFactor` that is the expected number of events in data

Option 3:

- `Lumi="1"` in the measurement XML's element
- Histograms bins have units of number of events expected in data
- Some samples have `NormFactor` that are all relative to prediction (eg. 1 is the nominal prediction)

It's up to you. In the end, the expected number is the product Lumi×NormFactor(s)×BinContent corresponding to $\lambda_{cs}\phi_{cs}\sigma_{csb}$.

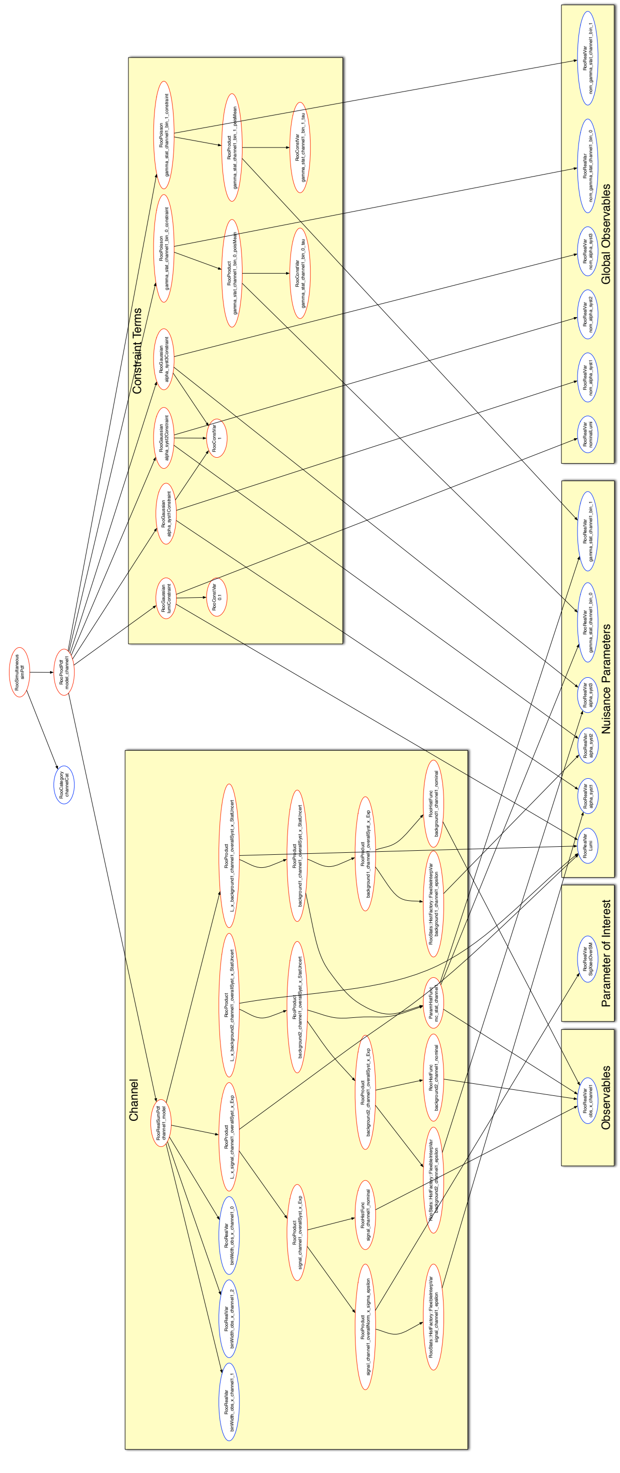


Figure 2: A graphical representation of the resulting `RooFit` model created from the standard example. The nodes have been organized according to their role in the model.

3.3 Usage of the HistFactory

ROOT installation

Download, install, and setup ROOT v5.28 or greater. It is recommended to use one of the patch releases of v5.28 as the ”standard form” described below was not available before the patch releases.

```
cd $ROOTSYS
source bin/thisroot.sh
```

This will setup your `MANPATH` environment variable so that you can use the command line help.

prepareHistFactory

```
man prepareHistFactory
prepareHistFactory
```

The command line executable `prepareHistFactory [dir_name]` is a simple script that prepares a working area (and creates the directory dir_name if specified). Within the directory dir_name, it creates a results/, data/, and config/ directory relative to the given path. It also copies the HistFactorySchema.dtd and example XML files into the config/ directory. Additionally, it copies a root file into the data/ directory for use with the examples. Once this is done, one is ready to run the example hist2workspace input.xml or edit the XML files for a new project.

hist2workspace

```
man hist2workspace
hist2workspace config/example.xml
```

The command line executable `hist2workspace [option] [input xml]` is a utility to create RooFit/RooStats workspace from histograms

OPTIONS:

- `-standard_form` default model (from v5.28.00a and beyond), which creates an extended PDF that interpolates between RooHistFuncs. This is much faster for models with many bins and uses significantly less memory.
- `-number_counting_form` this was the original model in 5.28 (without patches). It uses a Poisson for each bin of the histogram. This can become slow and memory intensive when there are many bins.

3.4 Usage with RooStats tools

Once one runs `hist2workspace` on an XML file there will be output root and eps files in the results directory. The files are named

```
   results/[Prefix]_[Channel]_[Measurement]_model.root
```

where Prefix is specified in the `<Combination>` element in the top-level XML file, for example:

```
<Combination OutputFilePrefix="./results/example" Mode="comb" >
```

Measurement is specified in each of the `<Measurement>` elements in the top-level XML file

```
<Measurement Name="AllSYS" ...>
```

and Channel is ”combined” for the combined model, but a model file is exported for each individual channel as well using the name taken from the `<Channel>` element of the corresponding channel’s XML file

```
<Channel Name="channelEle" ...>
```

These root files have inside a RooWorkspace which contains a RooDataSet and a ModelConfig that can be used with standard RooStats tools (see for example `$ROOTSYS/tutorials/RooStats/Standard*Demo.C`

```
$ hist2workspace config/example.xml
$ root.exe results/example_combined_GaussExample_model.root
root [0]
Attaching file results/example_combined_GaussExample_model.root as _file0...
root [1] .ls
TFile** results/example_combined_GaussExample_model.root
 TFile* results/example_combined_GaussExample_model.root
  KEY: RooWorkspace combined;1 combined
  KEY: TProcessID ProcessID0;1 1222429a-5b98-11e0-9717-0701a8c0beef

root [2] combined->Print()

RooWorkspace(combined) combined contents

variables
---------
...

p.d.f.s
-------
....

functions
--------
....

datasets
--------
RooDataSet::asimovData(channelCat,obs_channel1)
RooDataSet::obsData(channelCat,obs_channel1)

named sets
----------
...

generic objects
---------------
RooStats::ModelConfig::ModelConfig



root [3] using namespace RooStats
root [4] ModelConfig* mc = (ModelConfig*) combined->obj("ModelConfig")
root [5] mc->Print()

=== Using the following for ModelConfig ===
Observables:             RooArgSet:: = (obs_channel1,weightVar,channelCat)
Parameters of Interest:  RooArgSet:: = (SigXsecOverSM)
Nuisance Parameters:     RooArgSet:: = (alpha_syst2,alpha_syst3)
Global Observables:      RooArgSet:: = (nominalLumi,nom_alpha_syst1,nom_alpha_syst2,nom_alpha_syst3)
PDF:                     RooSimultaneous::simPdf[ indexCat=channelCat channel1=model_channel1 ] = 260.156
```

4 Interpolation & Constraints

The treatment of systematic uncertainties is subtle, particularly when one wishes to take into account the correlated effect of multiple sources of systematic uncertainty across many signal and background samples. The most important conceptual issue is that we separate the source of the uncertainty (for instance the uncertainty in the calorimeter's response to jets) from its effect on an individual signal or background sample (eg. the change in the acceptance and shape of a W+jets background). In particular, the same source of uncertainty has a different effect on the various signal and background samples [1]. The effect of these "$\pm 1\sigma$" variations about the nominal predictions $\eta_s^0 = 1$ and σ_{sb}^0 is quantified by dedicated studies that provide $\eta_{sp}^{\pm}$ and $\sigma_{spb}^{\pm}$. The result of these studies can be arranged in tables like those below. The main purpose of the `HistFactory` XML schema is to represent these tables.

Syst	Sample 1	...	Sample N
Nominal Value	$\eta^0_{s=1} = 1$	...	$\eta^0_{s=N} = 1$
p=`OverallSys` 1	$\eta^+_{p=1,s=1}$, $\eta^-_{p=1,s=1}$	...	$\eta^+_{p=1,s=N}$, $\eta^-_{p=1,s=N}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
p=`OverallSys` M	$\eta^+_{p=M,s=1}$, $\eta^-_{p=M,s=1}$	...	$\eta^+_{p=M,s=N}$, $\eta^-_{p=M,s=N}$
Net Effect	$\eta_{s=1}(\boldsymbol{\alpha})$	...	$\eta_{s=N}(\boldsymbol{\alpha})$

Table 2: Tabular representation of sources of uncertainties that produce a correlated effect in the normalization individual samples (eg. OverallSys). The η^+_{ps} represent histogram when $\alpha_s = 1$ and are inserted into the `High` attribute of the `OverallSys` XML element. Similarly, the η^-_{ps} represent histogram when $\alpha_s = -1$ and are inserted into the `Low` attribute of the `OverallSys` XML element. Note, this does not imply that $\eta^+ > \eta^-$, the ± superscript correspond to the variation in the source of the systematic, not the resulting effect.

Syst	Sample 1	...	Sample N
Nominal Value	$\sigma^0_{s=1,b}$	...	$\sigma^0_{s=N,b}$
p=`HistoSys` 1	$\sigma^+_{p=1,s=1,b}$, $\sigma^-_{p=1,s=1,b}$	...	$\sigma^+_{p=1,s=N,b}$, $\sigma^-_{p=1,s=N,b}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
p=`HistoSys` M	$\sigma^+_{p=M,s=1,b}$, $\sigma^-_{p=M,s=1,b}$	...	$\sigma^+_{p=M,s=N,b}$, $\sigma^-_{p=M,s=N,b}$
Net Effect	$\sigma_{s=1,b}(\boldsymbol{\alpha})$	...	$\sigma_{s=N,b}(\boldsymbol{\alpha})$

Table 3: Tabular representation of sources of uncertainties that produce a correlated effect in the normalization and shape individual samples (eg. `HistoSys`). The σ^+_{psb} represent histogram when $\alpha_s = 1$ and are inserted into the `HighHist` attribute of the `HistoSys` XML element. Similarly, the σ^-_{psb} represent histogram when $\alpha_s = -1$ and are inserted into the `LowHist` attribute of the `HistoSys` XML element.

Once one has tabulated the effects of the individual sources of systematic uncertainty as above, one must address two related issues to form a likelihood parametrized with continuous nuisance parameters. First, one must provide an interpolation algorithm to interpolate to define $\eta_s(\boldsymbol{\alpha})$ and $\sigma_{sb}(\boldsymbol{\alpha})$. Secondly, one must incorporate constraint terms on the α_p to reflect that the uncertain parameter has been estimated with some uncertainty by an auxiliary measurement. A strength of the histogram template based approach (compared to parametrized analytic functions) is that the effect of individual systematics

[1] Here we suppress the channel index c on η_{cs} and σ_{cab}

are tracked explicitly; however, the ambiguities associated to the interpolation and constraints are a weakness.

4.1 Interpolation Options

For each sample, one can interpolate and extrapolate from the nominal prediction $\eta_s^0 = 1$ and the variations $\eta_{ps}^{\pm}$ to produce a parametrized $\eta_s(\boldsymbol{\alpha})$. Similarly, one can interpolate and extrapolate from the nominal shape σ_{sb}^0 and the variations $\sigma_{psb}^{\pm}$ to produce a parametrized $\sigma_{sb}(\boldsymbol{\alpha})$. We choose to parametrize α_p such that $\alpha_p = 0$ is the nominal value of this parameter, $\alpha_p = \pm 1$ are the "$\pm 1\sigma$ variations". Needless to say, there is a significant amount of ambiguity in these interpolation and extrapolation procedures and they must be handled with care. In the future the `HistFactory` may support other types of shape interpolation, but as of ROOT 5.32 the shape interpolation is a 'vertical' style interpolation that is treated independently per-bin. Four interpolation strategies are described below and can be compared in Fig 3.

Piecewise Linear (InterpCode=0)

The piecewise-linear interpolation strategy is defined as

$$\eta_s(\boldsymbol{\alpha}) = 1 + \sum_{p \in \text{Syst}} I_{\text{lin.}}(\alpha_p; 1, \eta_{sp}^{+},\ \eta_{sp}^{-}) \tag{14}$$

and for shape interpolation it is

$$\sigma_{sb}(\boldsymbol{\alpha}) = \sigma_{sb}^0 + \sum_{p \in \text{Syst}} I_{\text{lin.}}(\alpha_p; \sigma_{sb}^0, \sigma_{psb}^{+},\ \sigma_{psb}^{-}) \tag{15}$$

with

$$I_{\text{lin.}}(\alpha; I^0, I^+, I^-) = \begin{cases} \alpha(I^+ - I^0) & \alpha \geq 0 \\ \alpha(I^0 - I^-) & \alpha < 0 \end{cases} \tag{16}$$

Pros: This approach is the most straightforward of the interpolation strategies.

Cons: It has two negative features. First, there is a kink (discontinuous first derivative) at $\alpha = 0$ (see Fig 3(b-d)), which can cause some difficulties for numerical minimization packages such as `Minuit`. Second, the interpolation factor can extrapolate to negative values. For instance, if $\eta^- = 0.5$ then we have $\eta(\alpha) < 0$ when $\alpha < -2$ (see Fig 3(c)).

Note that one could have considered the simultaneous variation of α_p and $\alpha_{p'}$ in a multiplicative way (see for example, Fig 4). The multiplicative accumulation is not an option currently.

Note that this is the default convention for $\sigma_{sb}(\boldsymbol{\alpha})$ (ie. `HistoSys`).

Piecewise Exponential (InterpCode=1)

The piecewise exponential interpolation strategy is defined as

$$\eta_s(\boldsymbol{\alpha}) = \prod_{p \in \text{Syst}} I_{\text{exp.}}(\alpha_p; 1, \eta_{sp}^{+},\ \eta_{sp}^{-}) \tag{17}$$

and for shape interpolation it is

$$\sigma_{sb}(\boldsymbol{\alpha}) = \sigma_{sb}^0 \prod_{p \in \text{Syst}} I_{\text{exp.}}(\alpha_p; \sigma_{sb}^0, \sigma_{psb}^{+},\ \sigma_{psb}^{-}) \tag{18}$$

with

$$I_{\mathrm{exp.}}(\alpha; I^0, I^+, I^-) = \begin{cases} (I^+/I_0)^{\alpha} & \alpha \geq 0 \\ (I^-/I_0)^{-\alpha} & \alpha < 0 \end{cases} \tag{19}$$

Pros: This approach ensures that $\eta(\alpha) \geq 0$ (see Fig 3(c)) and for small response to the uncertainties it has the same linear behavior near $\alpha \sim 0$ as the piecewise linear interpolation (see Fig 3(a)).

Cons: It has two negative features. First, there is a kink (discontinuous first derivative) at $\alpha = 0$, which can cause some difficulties for numerical minimization packages such as `Minuit`. Second, for large uncertainties it develops a different linear behavior compared to the piecewise linear interpolation. In particular, even if the systematic has a symmetric response (ie. $\eta^+ - 1 = 1 - \eta^-$) the interpolated response will develop a kink for large response to the uncertainties (see Fig 3(c)).

Note that the one could have considered the simultaneous variation of α_p and $\alpha_{p'}$ in an additive way, but this is not an option currently.

Note, that when paired with a Gaussian constraint on α this is equivalent to linear interpolation and a log-normal constraint in $\ln(\alpha)$. This is the default strategy for normalization uncertainties $\eta_s(\boldsymbol{\alpha})$ (ie. `OverallSys`) and is the standard convention for normalization uncertainties in the LHC Higgs Combination Group. In the future, the default may change to the Polynomial Interpolation and Exponential Extrapolation described below.

Quadratic Interpolation and Linear Extrapolation (InterpCode=2)

The quadratic interpolation and linear extrapolation strategy is defined as

$$\eta_s(\boldsymbol{\alpha}) = 1 + \sum_{p \in \mathrm{Syst}} I_{\mathrm{quad.|lin.}}(\alpha_p; 1, \eta^+_{sp},\ \eta^-_{sp}) \tag{20}$$

and for shape interpolation it is

$$\sigma_{sb}(\boldsymbol{\alpha}) = \sigma^0_{sb} + \sum_{p \in \mathrm{Syst}} I_{\mathrm{quad.|lin.}}(\alpha_p; \sigma^0_{sb}, \sigma^+_{psb},\ \sigma^-_{psb}) \tag{21}$$

with

$$I_{\mathrm{quad.|lin.}}(\alpha; I^0, I^+, I^-) = \begin{cases} (b+2a)(\alpha - 1) & \alpha > 1 \\ a\alpha^2 + b\alpha & |\alpha| \leq 1 \\ (b-2a)(\alpha + 1) & \alpha < -1 \end{cases} \tag{22}$$

and

$$a = \tfrac{1}{2}(I^+ + I^-) - I^0 \qquad \text{and} \qquad b = \tfrac{1}{2}(I^+ - I^-)\ . \tag{23}$$

Pros: This approach avoids the kink (discontinuous first derivative) at $\alpha = 0$ (see middle panel of Fig 3), which can cause some difficulties for numerical minimization packages such as `Minuit`.

Cons: It has a few negative features. First, in the case that both the response to both positive and negative variations have the same sign of effect relative to the nominal (ie. $(\eta^+ - 1)(\eta^- - 1) > 0$), the quadratic interpolation can lead to an an intermediate value with the opposite effect. For example, Fig 3(b) shows a case where $\eta(\alpha = -0.3) < 1$ while $\eta^{\pm} > 0$. Second, when the positive and negative variations have opposite signs, the extrapolation can reverse the trend. For example, Fig 3(d) shows an example for $\eta^- = 0.95$ and $\eta^+ = 1.5$ where for $\alpha \lesssim 1.5$ we have the reversal $\eta(\alpha) > 1$. Third, the interpolation factor can extrapolate to negative values. For instance, if $\eta^- = 0.5$ then we have $\eta(\alpha) < 0$ when $\alpha < -2$ (see Fig 3(c)).

Note that one could have considered the simultaneous variation of α_p and $\alpha_{p'}$ in a multiplicative way (see for example, Fig 4). The multiplicative accumulation is not an option currently.

Polynomial Interpolation and Exponential Extrapolation (InterpCode=4)

The strategy of this interpolation option is to use the piecewise exponential extrapolation as above with a polynomial interpolation that matches $\eta(\alpha = \pm\alpha_0)$, $d\eta/d\alpha|_{\alpha=\pm\alpha_0}$, and $d^2\eta/d\alpha^2|_{\alpha=\pm\alpha_0}$ and the boundary $\pm\alpha_0$ is defined by the user (with default $\alpha_0 = 1$).

$$\eta_s(\boldsymbol{\alpha}) = \prod_{p\in\text{Syst}} I_{\text{poly}|\text{exp.}}(\alpha_p; 1, \eta^+_{sp},\ \eta^-_{sp}, \alpha_0) \tag{24}$$

with

$$I_{\text{poly}|\text{exp.}}(\alpha; I^0, I^+, I^-, \alpha_0) = \begin{cases} (I^+/I_0)^\alpha & \alpha \geq \alpha_0 \\ 1 + \sum_{i=1}^{6} a_i \alpha^i & |\alpha| < \alpha_0 \\ (I^-/I_0)^{-\alpha} & \alpha \leq -\alpha_0 \end{cases} \tag{25}$$

and the a_i are fixed by the boundary conditions described above.

Pros: This approach avoids the kink (discontinuous first and second derivatives) at $\alpha = 0$ (see Fig 3(b-d)), which can cause some difficulties for numerical minimization packages such as `Minuit`. This approach ensures that $\eta(\alpha) \geq 0$ (see Fig 3(c)).

Note: This option is not available in ROOT 5.32.00, but is available for normalization uncertainties (OverallSys) in the subsequent patch releases. In future releases, this may become the default.

4.1.1 Defaults in ROOT 5.32

The default strategy for normalization uncertainties $\eta_s(\boldsymbol{\alpha})$ (ie. `OverallSys`) is the piecewise exponential option and it is the standard convention for normalization uncertainties in the LHC Higgs Combination Group.

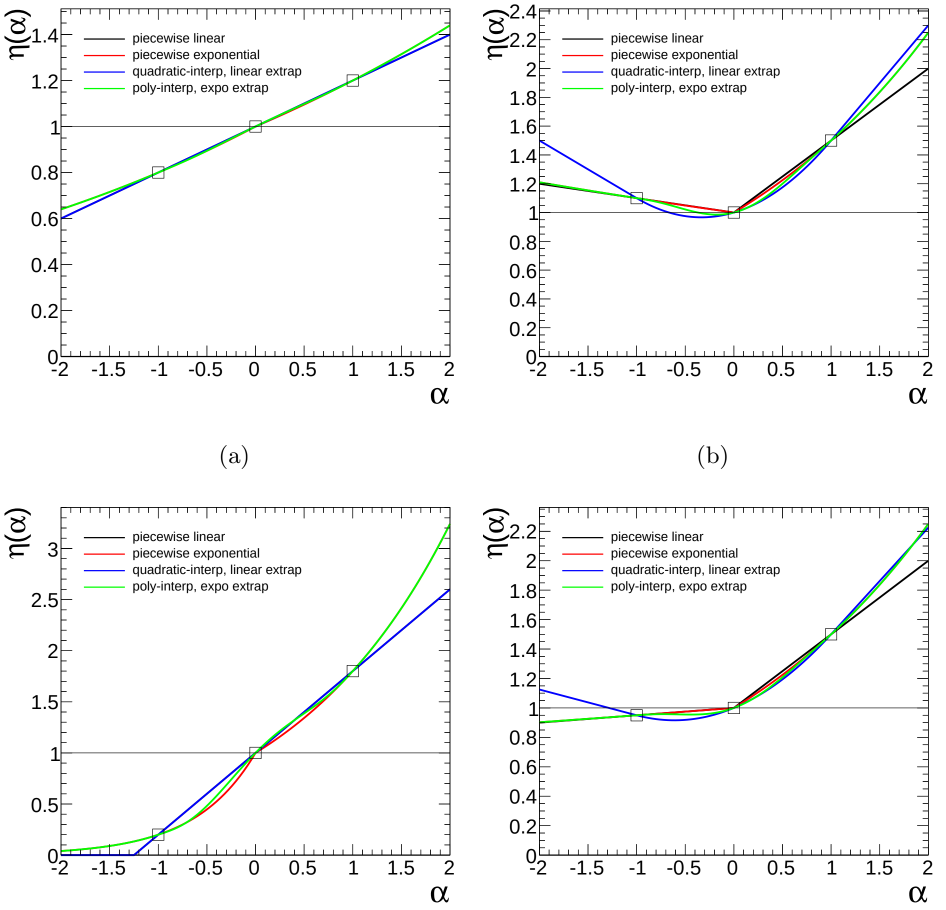
The default convention for $\sigma_{sb}(\boldsymbol{\alpha})$ (ie. `HistoSys`) is the piecewise linear option.

The code for $\eta_s(\boldsymbol{\alpha})$ can be found here:

`http://root.cern.ch/root/html532/src/RooStats__HistFactory__FlexibleInterpVar.cxx.html`

The code for $\sigma_{sb}(\boldsymbol{\alpha})$ can be found here:

`http://root.cern.ch/root/html532/src/PiecewiseInterpolation.cxx.html`



(a)

(b)

(c)

(d)

Figure 3: Comparison of the three interpolation options for different $\eta^{\pm}$. (a) $\eta^{-} = 0.8$, $\eta^{+} = 1.2$, (b) $\eta^{-} = 1.1$, $\eta^{+} = 1.5$, (c) $\eta^{-} = 0.2$, $\eta^{+} = 1.8$, and (d) $\eta^{-} = 0.95$, $\eta^{+} = 1.5$

4.2 Constraint Terms (+ global observables and nuisance parameter priors)

4.2.1 Consistent Bayesian and Frequentist modeling

The variational estimates $\eta^{\pm}$ and $\sigma^{\pm}$ correspond to so called "$\pm 1\sigma$ variations" in the source of the uncertainty. Here we are focusing on the source of the uncertainty, not its affect on rates and shapes. For instance, we might say that the jet energy scale has a 10% uncertainty. [2] This is common jargon, but what does it mean? The most common interpretation of this statement is that the uncertain parameter α_p (eg. the jet energy scale) has a Gaussian distribution. However, this way of thinking is manifestly bayesian. If the parameter was estimated from an auxiliary measurement, then it is the PDF for that measurement that we wish to include into our probability model. In the frequentist way of thinking, the jet energy scale has an unknown true value and upon repeating the experiment many times the auxiliary measurements estimating the jet energy scale would fluctuate randomly about this true value. To aid in this subtle distinction, we use greek letters for the parameters (eg. α_p) and roman letters for the auxiliary measurements a_p.

[2] Without loss of generality, we choose to parametrize α_p such that $\alpha_p = 0$ is the nominal value of this parameter, $\alpha_p = \pm 1$ are the "$\pm 1\sigma$ variations".

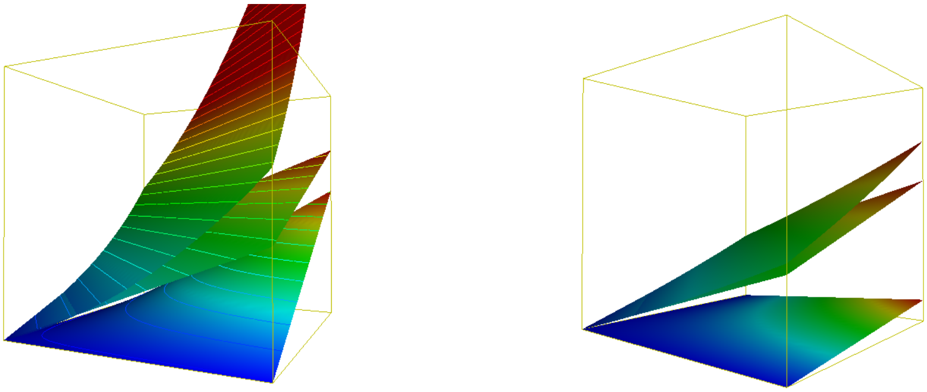


Figure 4: The upper-most curve corresponds to $\eta = (\eta_1^+)^{\alpha_1}(\eta_2^+)^{\alpha_2}$ (as in the exponential interpolation option). The middle surface corresponds to $\eta = 1 + \eta_1^+\alpha_1 + \eta_2^+\alpha_2$ (as in the linear interpolation option). The lowest surface corresponds to $\eta = 1+\eta_1^+\alpha_1\cdot\eta_2^+\alpha_2$ (currently not an opiton). The left frame has limits correspond to $\alpha_{1,2} \in [0,3]$ and $\eta(\alpha_1,\alpha_2) \in [0,5]$ and $\eta_1^+ = \eta_2^+ = 1.1$ (eg. a 10% relative uncertainty). The right frame has limits correspond to $\alpha_{1,2} \in [0,3]$ and $\eta(\alpha_1,\alpha_2) \in [0,5]$ and $\eta_1^+ = \eta_2^+ = 1.5$ (eg. a 50% relative uncertainty).

Furthermore, we interpret the "$\pm 1\sigma$" variation in the frequentist sense, which leads to the constraint term $G(a_p|\alpha_p,1)$. Then, we can pair the resulting likelihood with some prior on α_p to form a bayesian posterior if we wish.

It is worth mentioning here that the constraint terms are idealized versions of the auxiliary measurements. In reality, the measurements that were used to estimate the uncertainty in a quantity such as the jet energy scale are actually quite complex. Ideally, one would include the full likelihood function for those auxiliary measurements into the probability model, but that is often impractical. To the extent that the likelihood resulting from the auxiliary measurement is in the Gaussian regime, then this idealization is not a bad approximation.

It is often advocated that a "log-normal" or "gamma" distribution for α_p is more appropriate. Here we must take some care to build a probability model that can maintain a consistent interpretation in bayesian a frequentist settings. This will be discussed in the subsections below. Table 4 summarizes a few consistent treatments of the frequentist pdf, the likelihood function, a prior, and the resulting posterior.

Finally, it is worth mentioning that the uncertainty on some parameters is not the result of an auxiliary measurement – so the constraint term idealization, it is not just a convenience, but a real conceptual leap. This is particularly true for theoretical uncertainties from higher-order corrections or renormalizaiton and factorization scale dependence. In these cases a formal frequentist analysis would not include a constraint term for these parameters, and the result would simply depend on their assumed values. As this is not the norm, we can think of reading Table 4 from right-to-left with a subjective Bayesian prior $\pi(\alpha)$ being interpreted as coming from a fictional auxiliary measurement.

4.2.2 Options for Constraint Terms

Gaussian Constraint

The Gaussian constraint for α_p corresponds to the familiar situation. It is a good approximation of the auxiliary measurement when the likelihood function for α_p from that

PDF	Likelihood $\propto$	Prior π_0	Posterior π
$G(a_p \vert \alpha_p, \sigma_p)$	$G(\alpha_p \vert a_p, \sigma_p)$	$\pi_0(\alpha_p) \propto$ const	$G(\alpha_p \vert a_p, \sigma_p)$
$\text{Pois}(n_p \vert \tau_p \beta_p)$	$P_\Gamma(\beta_p \vert A = \tau_p; B = 1 + n_p)$	$\pi_0(\beta_p) \propto$ const	$P_\Gamma(\beta_p \vert A = \tau_p; B = 1 + n_p)$
$P_{\text{LN}}(n_p \vert \beta_p, \sigma_p)$	$\beta_p \cdot P_{\text{LN}}(\beta_p \vert n_p, \sigma_p)$	$\pi_0(\beta_p) \propto$ const	$P_{\text{LN}}(\beta_p \vert n_p, \sigma_p)$
$P_{\text{LN}}(n_p \vert \beta_p, \sigma_p)$	$\beta_p \cdot P_{\text{LN}}(\beta_p \vert n_p, \sigma_p)$	$\pi_0(\beta_p) \propto 1/\beta_p$	$P_{\text{LN}}(\beta_p \vert n_p, \sigma_p)$

Table 4: Table relating consistent treatments of PDF, likelihood, prior, and posterior for nuisance parameter constraint terms.

auxiliary measurement has a Gaussian shape. More formally, it is valid when the maximum likelihood estimate of α_p (eg. the best fit value of α_p) has a Gaussian distribution. Here we can identify the maximum likelihood estimate of α_p with the global observable a_p, remembering that it is a number that is extracted from the data and thus its distribution has a frequentist interpretation. In the RooFit workspace produced by `HistFactory`, this variable has a name like `nom_alpha_<name>` and it is included in the `ModelConfig`'s list of `GlobalObservables`. We chose to scale α_p (and thus a_p so that the distribution has unit variance: $G(a_p|\alpha_p, 1)$. Note that if we assume the true value $\alpha_p \neq 0$ and we sample a_p via (toy) Monte Carlo techniques, the distribution of a_p will not have a mean of 0.

$$G(a_p|\alpha_p, \sigma_p) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left[-\frac{(a_p - \alpha_p)^2}{2\sigma_p^2}\right] \tag{26}$$

with $\sigma_p = 1$ by default.

Note that the PDF of a_p and the likelihood for α_p are positive for all values. Thus if α_p represents a shifted and rescaled version of a more physical parameter that is bounded, then the Gaussian distribution is attributing some positive probability to the unphysical regions. For instance, energy scales, reconstruction efficiencies, and background normalizations must be ≥ 0. Consider a jet energy scale that is estimated with 25% uncertainty, then $\alpha < -4$ would correspond to an unphysical negative jet energy scale. One can also consider normalization uncertainties where α and $\eta(\alpha)$ are more directly related – in particular $\eta(\alpha)$ is a linear function. Consider a background that is estimated with 50% uncertainty, then for $\alpha < -2$ will correspond to a negative background estimate, and we will have $\eta(\alpha < 2) < 0$.

Technically, RooFit's PDF classes (`RooGaussian` in this case) make sure that the PDF is normalized to unity within the range of the observable (in this case a_p). So the technical implementation will actually correspond to a truncated and renormalized Gaussian (the default range for a_p is $[-5, 5]$).

Poisson ("Gamma") constraint

When the auxiliary measurement is actually based on counting events in a control region (eg. a Poisson process), a more accurate to describe the auxiliary measurement with a Poisson distribution. It has been shown that the truncated Gaussian constraint can lead to undercoverage (overly optimistic) results, which makes this issue practically relevant. Table 4 shows that a Poisson PDF together with a uniform prior leads to a gamma posterior, thus this type of constraint is often called a "gamma" constraint. This is a bit unfortunate since the gamma distribution is manifestly Bayesian and with a different choice of prior, one might not arrive at a gamma posterior. When dealing with the Poisson constraint, it is no longer convenient to work with our conventional scaling for α_p which can be negative. Instead, it is more natural to think of the number of events measured in the auxiliary measurement n_p and the mean of the Poisson parameter. This information is not usually available, instead one usually has some notion of the relative uncertainty in the parameter σ_p^{rel} (eg. a the jet energy scale is known to 10%). In order to give some uniformity to the different uncertainties of this type and think of relative uncertainty, the nominal rate is factored out into a constant τ_p and the mean of the Poisson is given by $\tau_p\beta_p$.

$$\mathrm{Pois}(n_p|\tau_p\beta_p) = \frac{(\tau_p\beta_p)^{n_p}\ e^{-\tau_p\beta_p}}{n_p!} \tag{27}$$

Here we can use the fact that $\mathrm{Var}[n_p] = \sqrt{\tau_p\beta_p}$ and reverse engineer the nominal auxiliary measurement

$$n_p^0 = \tau_p = (1/\sigma_p^{\mathrm{rel}})^2\ . \tag{28}$$

where the superscript 0 is to remind us that n_p will fluctuate in repeated experiments but n_p^0 is the value of our measured estimate of the parameter.

Thus the nominal situation corresponds to $\beta_p = 1$ and the "$\pm 1\sigma$ variations" (which is now ambiguous) conventionally correspond to $\beta_p = 1 \pm \sigma_p^{\mathrm{rel}} = 1 \pm \tau_p^{-1/2}$. It is more convenient to modify the constraint term while keeping the interpolation $\eta(\alpha)$ fixed, thus we introduce the linear relationship that satisfies $\alpha(\beta = 1) = 0$ and $\alpha(\beta = 1 \pm \tau_p^{-1/2}) = \pm 1$

$$\alpha_p(\beta_p) = \sqrt{\tau_p}\,(\beta_p - 1) \tag{29}$$

One important thing to keep in mind is that there is only one constraint term per nuisance parameter, so there must be only one σ_p^{rel} per nuisance parameter. This σ_p^{rel} is related to the fundamental uncertainty in the source and we cannot infer this from the various response terms $\eta_{ps}^{\pm}$ or $\sigma_{pub}^{\pm}$. In the XML this is not a property of a channel, but of a measurement and it is encoded in a term like

```
<ConstraintTerm Type="Gamma" RelativeUncertainty="0.1">JES</ConstraintTerm>
```

Another technical difficulty is that the Poisson distribution is discrete. So if one were to say the relative uncertainty was 30%, then we would find $n_p^0 = 11.11...$, which is not an integer. Rounding n_p to the nearest integer while maintaining $\tau_p = (1/\sigma_p^{\mathrm{rel}})^2$ will bias the maximum likelihood estimate of β_p away from 1. As of ROOT 5.32 the `ConstraintTerm Type="Gamma"` used the `RooGamma` (which generalizes more continuously) with

$$P_\Gamma(\beta_p|A = \tau_p, B = n_p - 1) = A(A\beta_p)^B e^{-A\beta_p}/\Gamma(B) \tag{30}$$

The implementation works fine for likelihood fits, bayesian calculations, and frequentist techniques based on asymptotic approximations, but it does not offer a consistent treatment of the pdf for the global observable n_p that is needed for techniques based on Monte Carlo techniques. In future versions of ROOT, the constraint will probably be replaced with `RooPoisson` with an option `setNoRounding(true)`.

Log-normal constraint

From Eadie et al., "The log-normal distribution represents a random variable whose logarithm follows a normal distribution. It provides a model for the error of a process involving many small multiplicative errors (from the Central Limit Theorem). It is also appropriate when the value of an observed variable is a random proportion of the previous observation."

As in the case of the "Gamma" constraints we need to reparametrize to a nuisance parameter β_p that is positive and centered around 1. Again we use α for the response of the systematics and relate the two via

$$\alpha_p(\beta_p) = \sqrt{\tau_p}\,(\beta_p - 1) \tag{31}$$

And the equivalent global observable is

$$n_p^0 = \tau_p = (1/\sigma_p^{\mathrm{rel}})^2 \ . \tag{32}$$

```
<ConstraintTerm Type="LogNormal" RelativeUncertainty="0.1">JES</ConstraintTerm>
```

$$P_{\mathrm{LN}}(n_p|\beta_p,\kappa_p) = \frac{1}{\sqrt{2\pi}\ln\kappa}\frac{1}{n_p}\exp\left[-\frac{\ln(n_p/\beta_p)^2}{2(\ln\kappa_p)^2}\right] \tag{33}$$

(blue curve in Fig. 5(a)).

$$L(\beta_p) = \frac{1}{\sqrt{2\pi}\ln\kappa}\frac{1}{n_p}\exp\left[-\frac{\ln(n_p/\beta_p)^2}{2(\ln\kappa_p)^2}\right] \tag{34}$$

(red curve in Fig. 5(b)).

$$\pi(\beta_p) \propto \pi_0(\beta_p)\,\frac{1}{\sqrt{2\pi}\ln\kappa}\frac{1}{n_p}\exp\left[-\frac{\ln(n_p/\beta_p)^2}{2(\ln\kappa_p)^2}\right] \tag{35}$$

When paired with an "ur-prior" $\pi_0(\beta_p) \propto 1/\beta_p$ (green curve in Fig. 5(b)), this results in a posterior distribution that is also of a log-normal form for β_p (blue curve in Fig. 5(b)).

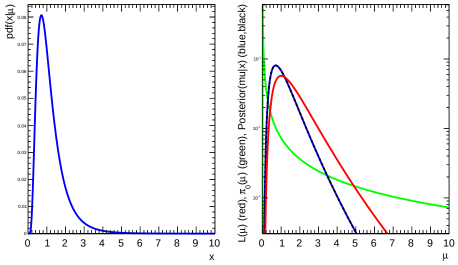


Figure 5: The lognormal constraint term: (left) the pdf for the global observable a_p and (right) the likelihood function, the posterior based on a flat prior on β_p, and the posterior based on a $1/\beta_p$ prior.

```
void LogNormalExample(double relUncert = 0.8){
  double kappa = 1+relUncert;
  RooWorkspace w("w");

  // make the PDF (likelihood term) which has a log-normal distribution for x
  w.factory(Form("Lognormal::pdf(x[1,0,10],mu[1,0,10],kappa[%f])",kappa));

  // make the ur-prior that gives a log-normal posterior
  w.factory("EXPR::urprior('1/mu',mu)");

  // Multiply prior*likelihood to get posterior numerically
  w.factory("PROD::posterior_numerical(pdf,urprior)");

  // check against analytic derivation of posterior
  w.factory(Form("Lognormal::posterior_analytical(mu,x,kappa)"));

  TCanvas* c1 = new TCanvas();
  c1->Divide(2);

  // The PDF for the measurement
  c1->cd(1);
  RooPlot* xframe = w.var("x")->frame();
  w.pdf("pdf")->plotOn(xframe);
  xframe->SetYTitle("pdf(x|#mu)");
  xframe->GetYaxis()->SetLabelSize(.02);
  xframe->GetYaxis()->SetTitleOffset(1.);
  xframe->Draw();

  // The likelihood and posterior
  c1->cd(2)->SetLogy();
  RooPlot* muframe = w.var("mu")->frame();
  w.pdf("urprior")->plotOn(muframe,LineColor(kGreen));
  w.pdf("posterior_numerical")->plotOn(muframe,LineColor(kBlue));
  w.pdf("posterior_analytical")->plotOn(muframe,LineColor(kBlack),LineStyle(kDashed)↩
      );
  w.pdf("pdf")->plotOn(muframe,LineColor(kRed));
  muframe->SetXTitle("#mu");
  muframe->SetYTitle("L(#mu) (red), #pi_{0}(#mu) (green), Posterior(mu|x) (blue,↩
      black)");
  muframe->GetYaxis()->SetLabelSize(.02);
  muframe->GetYaxis()->SetTitleOffset(1.);

  muframe->Draw();

}
```

5 Examples

5.1 A Simple Example

Here we consider a single channel example with one signal and two backgrounds. All three samples histograms are based on theoretical predictions (aka. Monte Carlo), thus the luminosity uncertainty should propagate to these channels – this is accomplished by `NormalizeByTheory="True"`. In this example, no shape uncertainties are included and statistical uncertainty on the histograms is not taken into account. The parameter of interest in this example is the signal cross section relative to the predicted value used to create the signal histogram – it is called `SigXsecOverSM`. Three systematic effects are considered that modify the normalization on the channels – here just named "syst1", "syst2", and "syst3". In this example syst3 affects the normalization of both backgrounds (though in opposite directions).

Syst	Signal (s=1)	Background1 (s=2)	Background2 (s=2)
syst1 ($p = 1$)	$\eta^+_{11} = 1.05,\ \ \eta^-_{11} = 0.95$	–	–
syst2 ($p = 2$)	–	$\eta^+_{22} = 1.07,\ \ \eta^-_{22} = 0.93$	–
syst3 ($p = 3$)	–	$\eta^+_{32} = 1.03,\ \ \eta^-_{32} = 0.95$	$\eta^+_{33} = 0.97,\ \ \eta^-_{33} = 1.02$

Table 5: Tabular representation of sources of uncertainties that produce a correlated effect in the normalization individual samples (eg. OverallSys). The η^+_{ps} represent histogram when $\alpha_s = 1$ and are inserted into the `High` attribute of the `OverallSys` XML element. Similarly, the η^-_{ps} represent histogram when $\alpha_s = -1$ and are inserted into the `Low` attribute of the `OverallSys` XML element. Note, this does not imply that $\eta^+ > \eta^-$, the ± superscript correspond to the variation in the source of the systematic, not the resulting effect.

```
<!DOCTYPE Channel  SYSTEM 'HistFactorySchema.dtd'>

  <Channel Name="channel1" InputFile="./data/example.root" >
    <Data HistoName="data"  />

    <Sample Name="signal" HistoName="signal" NormalizeByTheory="True" >
      <OverallSys Name="syst1" Low="0.95 High="1.05" "/>
      <NormFactor Name="SigXsecOverSM" Val="1" Low="0." High="3."  />
    </Sample>

    <Sample Name="background1" HistoName="background1" NormalizeByTheory="True" >
      <OverallSys Name="syst2" Low="0.93" High="1.07"/>
      <OverallSys Name="syst3" Low="0.95" High="1.03"/>
    </Sample>

    <Sample Name="background2" N HistoName="background2" NormalizeByTheory="True" >
      <OverallSys Name="syst3" Low="1.02" High="0.97"/>
    </Sample>
  </Channel>
```

```
<!DOCTYPE Combination  SYSTEM 'HistFactorySchema.dtd'>

<Combination OutputFilePrefix="./results/example" >

  <Input>./config/example_channel.xml</Input>

  <Measurement Name="GaussExample" Lumi="5." LumiRelErr="0.1" >
    <POI>SigXsecOverSM</POI>
  </Measurement>

</Combination>
```

5.2 ABCD

```
<!DOCTYPE Combination  SYSTEM ’HistFactorySchema.dtd’>

<Combination OutputFilePrefix="./results/ABCD"  >

  <Input>./config/A.xml</Input>
  <Input>./config/B.xml</Input>
  <Input>./config/C.xml</Input>
  <Input>./config/D.xml</Input>

  <Measurement Name="ABCD" Lumi="1." LumiRelErr="0.1" ExportOnly="True">
    <POI>mu</POI>
    <ParamSetting Const="True">Lumi b_acceptance c_acceptance d_acceptance mu_K_A mu_K_B mu_K_C mu_K_D</↩
         ParamSetting>
  </Measurement>


</Combination>
```

```
<!DOCTYPE Channel  SYSTEM ’HistFactorySchema.dtd’>

  <Channel Name="A" InputFile="./data/ABCD.root"  >
    <Data HistoName="A_data" HistoPath="" />

    <!-- This is the signal (eg. mu)-->
    <Sample Name="A_signal" HistoPath="" HistoName="unit_histogram">
      <!-- now mu is number of events-->
      <NormFactor Name="mu" Val="1" Low="0" High="200" />
      <OverallSys Name="syst1" High="1.01" Low="0.99" />
    </Sample>

    <!-- This bkg is estimated from MC (eg. mu_A^K) -->
    <Sample Name="A_backgroundMC" HistoPath="" NormalizeByTheory="True" HistoName="unit_histogram"  >
      <NormFactor Name="mu_K_A" Val="100" Low="0" High="200" />
    </Sample>

    <!-- Background 2 is completely Data-Driven -->
    <Sample Name="A_backgroundDD" HistoPath="" NormalizeByTheory="False" HistoName="unit_histogram" >
      <NormFactor Name="mu_D_U" Val="100" Low="24500" High="26000" />
      <NormFactor Name="etaB" Val="1" Low="0." High="0.02" Const="False" />
      <NormFactor Name="etaC" Val="1" Low="0." High="0.3" Const="False" />
      <!-- NormFactor and ShapeFactor same for a 1-bin histogram. But we can name NormFactor-->
    </Sample>

  </Channel>
```

```
<!DOCTYPE Channel  SYSTEM ’HistFactorySchema.dtd’>

  <Channel Name="B" InputFile="./data/ABCD.root"  >
    <Data HistoName="B_data" HistoPath="" />

    <!-- This is the signal contamination in B (eg. b*mu)-->
    <Sample Name="B_signal" HistoPath="" HistoName="unit_histogram">
      <NormFactor Name="mu" Val="1" Low=".2" High="1.5" />
      <NormFactor Name="b_acceptance" Val="0.1" Low="0." High="1.5" Const="True"/>
    </Sample>

    <!-- This bkg is estimated from MC (eg. mu_B^K) -->
    <Sample Name="B_backgroundMC" HistoPath="" NormalizeByTheory="True" HistoName="unit_histogram"  >
      <NormFactor Name="mu_K_B" Val="100" Low="0" High="200" />
    </Sample>

    <!-- Background 2 is completely Data-Driven -->
    <Sample Name="B_backgroundDD" HistoPath="" NormalizeByTheory="False" HistoName="unit_histogram" >
      <!-- Note, need some reasonable guess for the range of tauB -->
      <NormFactor Name="etaB" Val="10" Low="5" High="15" Const="False" />
      <NormFactor Name="mu_D_U" Val="100" Low="0" High="200" />
    </Sample>

  </Channel>
```

```
<!DOCTYPE Channel  SYSTEM ’HistFactorySchema.dtd’>

  <Channel Name="C" InputFile="./data/ABCD.root"  >
    <Data HistoName="C_data" HistoPath="" />

    <!-- This is the signal contamination in C (eg. c*mu)-->
    <Sample Name="C_signal" HistoPath="" HistoName="unit_histogram">
      <NormFactor Name="mu" Val="1" Low=".2" High="1.5" />
      <NormFactor Name="c_acceptance" Val="0.1" Low="0." High="1.5" Const="True"/>
    </Sample>

    <!-- This bkg is estimated from MC (eg. mu_C^K) -->
    <Sample Name="C_backgroundMC" HistoPath="" NormalizeByTheory="True" HistoName="unit_histogram"  >
      <NormFactor Name="mu_K_C" Val="100" Low="0" High="200" />
    </Sample>
    <!-- Background 2 is completely Data-Driven -->
    <Sample Name="C_backgroundDD" HistoPath="" NormalizeByTheory="False" HistoName="unit_histogram" >
      <!-- Note, need some reasonable guess for the range of tauC -->
      <NormFactor Name="etaC" Val="100" Low="50" High="150" Const="False" />
      <NormFactor Name="mu_D_U" Val="100" Low="20000" High="30000" />
    </Sample>

  </Channel>
```

```
<!DOCTYPE Channel  SYSTEM 'HistFactorySchema.dtd'>

  <Channel Name="D" InputFile="./data/ABCD.root"  >
    <Data HistoName="D_data" HistoPath="" />

    <!-- This is the signal contamination in D (eg. d*mu)-->
    <Sample Name="D_signal" HistoPath="" HistoName="unit_histogram">
      <NormFactor Name="mu" Val="1" Low=".2" High="1.5" />
      <NormFactor Name="d_acceptance" Val="0.1" Low="0." High="1.5" Const="True"/>
    </Sample>

    <!-- This bkg is estimated from MC (eg. mu_D^K) -->
    <Sample Name="D_backgroundMC" HistoPath="" NormalizeByTheory="True" HistoName="unit_histogram"  >
      <NormFactor Name="mu_K_D" Val="100" Low="0" High="200" />
    </Sample>

    <!-- Background 2 is completely Data-Driven -->
    <Sample Name="D_backgroundDD" HistoPath="" NormalizeByTheory="False" HistoName="unit_histogram" >
      <!--
       <NormFactor Name="tauB" Val="1" Low=".2" High="1.5" Const="False" />
       <NormFactor Name="tauC" Val="1" Low=".2" High="1.5" Const="False" />
       -->
      <NormFactor Name="mu_D_U" Val="100" Low="0" High="200" />
    </Sample>

  </Channel>
```

6 The HistFactory XML Schema in DTD Format

```
<!-- The top level combination spec -->
<!-- OutputFilePrefix: Prefix to the output root file to be created (inspection ↩
    histograms) -->
<!-- Mode: Type of the analysis -->
<!ELEMENT Combination (Function*,Input+,Measurement*)>
<!ATTLIST Combination
        OutputFilePrefix          CDATA            #REQUIRED
        Mode                      CDATA            #IMPLIED
>

<!-- Input files detailing the channels. One channel per file -->
<!ELEMENT Function  EMPTY>
<!ATTLIST Function
      Name                  CDATA          #REQUIRED
      Expression            CDATA          #REQUIRED
      Dependents            CDATA          #REQUIRED
>

<!-- Input files detailing the channels. One channel per file -->
<!ELEMENT Input (#PCDATA) >

<!-- Configuration for each measurement -->
<!-- Name: to be used as the heading in the table -->
<!-- Lumi: the luminosity of the measurement -->
<!-- LumiRelErr: the relative error known for the lumi -->
<!-- BinLow: the lowest bin number used for the measurement (inclusive) -->
<!-- BinHigh: the highest bin number used for the measurement (exclusive) -->
<!-- Mode: type of the measurement (a closed list of ...) -->
<!-- ExportOnly: if "True" skip fit, only export model -->
<!ELEMENT Measurement (POI,ParamSetting*,ConstraintTerm*) >
<!ATTLIST Measurement
        Name              CDATA            #REQUIRED
        Lumi              CDATA            #REQUIRED
        LumiRelErr        CDATA            #REQUIRED
        BinLow            CDATA            #IMPLIED
        BinHigh           CDATA            #IMPLIED
        Mode              CDATA            #IMPLIED
        ExportOnly        CDATA            #IMPLIED
>

<!-- Specify what you are measuring. Corresponds to the name specified in the ↩
    construction
of the model in the channel setup. Typically the NormFactor for xsec measurements --↩
    >
<!ELEMENT POI (#PCDATA) >

<!-- Specify what parameters are fixed, or have particular value -->
<!-- Val: set the value of the parameter -->
<!-- Const: set this parameter constant -->
<!ELEMENT ParamSetting (#PCDATA)>
<!ATTLIST ParamSetting
        Val               CDATA           #IMPLIED
        Const             CDATA           #IMPLIED
>

<!-- Specify an alternative shape to use for given constraint terms (Gaussian is ↩
    used if this is not specified) -->
<!-- Type: can be Gamma or Uniform -->
<!-- RelativeUncertainty: relative uncertainty on the shape -->
<!ELEMENT ConstraintTerm (#PCDATA)>
<!ATTLIST ConstraintTerm
        Type                  CDATA       #REQUIRED
        RelativeUncertainty   CDATA       #IMPLIED
>

<!-- Top element for channels. InputFile, HistoName and HistoPath
can be set at this level in which case they will become defaul to
all subsequent elements. Otherwise they can be set in individual
```

```
subelements -->
<!ELEMENT Channel (Data*,StatErrorConfig*,Sample+)>
<!-- InputFile: input file where the input histogram can be found (use abs path) -->
<!-- HistoPath: the path (within the root file) where the histogram can be found -->
<!-- HistoName: the name of the histogram to be used for this (and following in not overridden) item -->
<!ATTLIST Channel
        Name                CDATA            #REQUIRED
        InputFile           CDATA            #IMPLIED
        HistoPath           CDATA            #IMPLIED
        HistoName           CDATA            #IMPLIED
>

<!-- Data to be fit. If you don't provide it, Asimov data will be created -->
<!-- InputFile: any item set here will override the configuration for the subelements.
For this element there is no sublemenents so the setting will only have local effects -->
<!ELEMENT Data EMPTY>
<!ATTLIST Data
        InputFile           CDATA            #IMPLIED
        HistoPath           CDATA            #IMPLIED
        HistoName           CDATA            #IMPLIED
>

<!ELEMENT StatErrorConfig EMPTY>
<!ATTLIST StatErrorConfig
        RelErrorThreshold       CDATA            #IMPLIED
        ConstraintType          CDATA            #IMPLIED
>


<!-- Sample elements are made up of systematic variations -->
<!ELEMENT Sample (StatError | HistoSys | OverallSys | ShapeSys | NormFactor | ShapeFactor)*>
<!ATTLIST Sample
        Name                CDATA            #REQUIRED
        InputFile           CDATA            #IMPLIED
        HistoName           CDATA            #IMPLIED
        HistoPath           CDATA            #IMPLIED
        NormalizeByTheory       CDATA       #IMPLIED
>

<!-- Systematics for which the variation is provided by histograms -->
<!ELEMENT StatError EMPTY>
<!ATTLIST StatError
      Activate            CDATA            #REQUIRED
          HistoName           CDATA            #IMPLIED
          InputFile           CDATA            #IMPLIED
          HistoPath           CDATA            #IMPLIED
>

<!ELEMENT HistoSys EMPTY>
<!ATTLIST HistoSys
        Name                CDATA            #REQUIRED
        InputFile           CDATA            #IMPLIED
        HistoFileHigh       CDATA            #IMPLIED
        HistoPathHigh       CDATA            #IMPLIED
        HistoNameHigh       CDATA            #IMPLIED
        HistoFileLow        CDATA            #IMPLIED
        HistoPathLow        CDATA            #IMPLIED
        HistoNameLow        CDATA            #IMPLIED
        InputFileLow        CDATA            #IMPLIED
        InputFileHigh       CDATA            #IMPLIED
>

<!-- Systematics for which the variation is provided by simple overall scaling -->
<!ELEMENT OverallSys EMPTY>
<!ATTLIST OverallSys
        Name                CDATA            #REQUIRED
        High                CDATA            #REQUIRED
        Low                 CDATA             #REQUIRED
>

<!-- Systematics for which the variation is provided by simple overall scaling -->
<!ELEMENT ShapeSys EMPTY>
<!ATTLIST ShapeSys
          Name                CDATA             #REQUIRED
          HistoName           CDATA             #REQUIRED
          HistoPath           CDATA             #IMPLIED
          InputFile           CDATA             #IMPLIED
          ConstraintType      CDATA             #IMPLIED
>

<!-- Scaling factor, which may be the parameter of interest for cross section ↩
    measurements-->
<!ELEMENT NormFactor EMPTY>
<!ATTLIST NormFactor
        Name                CDATA             #REQUIRED
        Val                 CDATA             #REQUIRED
        High                CDATA             #REQUIRED
        Low                 CDATA             #REQUIRED
        Const               CDATA             #IMPLIED
>


<!-- Systematics for which the variation is provided by simple overall scaling -->
<!ELEMENT ShapeFactor EMPTY>
<!ATTLIST ShapeFactor
          Name                CDATA             #REQUIRED
>
```

One can convert this Gaussian constraints into a Poisson/Gamma systematic by adding lines like

```
  <ConstraintTerm Type="Gamma" RelativeUncertainty="0.1">JES</ConstraintTerm>
```

to the `Measurement` element.

7 C++ and Python

```

#include "RooStats/HistFactory/Measurement.h"



void MakeSimpleModel() {

  // Create a simple 1-channel model
  // using c++ and ROOT

  // Create the measurement object
  // This is the top node of the structure
  // We do some minor configuration as well
  RooStats::HistFactory::Measurement meas("my_measurement", "my measurement");

  // Set the prefix that will appear before
  // all output for this measurement
  // We Set ExportOnly to false, meaning
  // we will fit the measurement and make
  // plots in addition to saving the workspace
  meas.SetOutputPrefix("results/my_measurement");
  meas.SetExportOnly( False );

  // Set the name of the parameter of interest
  // Note that this parameter hasn't yet been
  // created, we are anticipating it
  meas.SetPOI( "SigXsecOverSM" );

  // Set the luminosity
  // There are a few conventions for this.
  // Here, we assume that all histograms have
  // already been scaled by luminosity
  // We also set a 10% uncertainty
  meas.SetLumi( 1.0 );
  meas.SetLumiRelErr( 0.10 );

  // Okay, now that we've configured the measurement,
  // we'll start building the tree.
  // We begin by creating the first channel
  RooStats::HistFactory::Channel chan( "channel" );

  // First, we set the 'data' for this channel
  // The data is a histogram represeting the
  // measured distribution.  It can have 1 or many bins.
  // In this example, we assume that the data histogram
  // is already made and saved in a ROOT file.
  // So, to 'set the data', we give this channel the
  // path to that ROOT file and the name of the data
  // histogram in that root file
  // The arguments are: SetData( HistogramName, HistogramFile )
  chan.SetData( "data", "data/example.root" );


  // Now that we have a channel and have attached
  // data to it, we will start creating our Samples
  // These describe the various processes that we
  // use to model the data.
  // Here, they just consist of a signal process
  // and a single background process.
  RooStats::HistFactory::Sample signal( "signal", "signal", "data/example.root" );

  // Having created this sample, we configure it
  // First, we add the cross-section scaling
  // parameter that we call SigXsecOverSM
  // Then, we add a systematic with a 5% uncertainty
  // Finally, we add it to our channel
  signal.AddNormFactor( "SigXsecOverSM", 1, 0, 3 );
  signal.AddOverallSys( "syst1",  0.95, 1.05 );
  chan.AddSample( signal );


  // We do a similar thing for our background
  RooStats::HistFactory::Sample background1( "background1", "background1", "data/↩
      example.root" );
  background1.ActivateStatError( "background1_statUncert", InputFile );
  background1.AddOverallSys( "syst2", 0.95, 1.05  );
  chan.AddSample( background1 );

  // And we create a second background for good measure
  Sample background2( "background2", "background2", "data/example.root" );
  background2.ActivateStatError();
  background2.AddOverallSys( "syst3", 0.95, 1.05  );
  chan.AddSample( background2 );


  // Now that we have fully configured our channel,
  // we add it to the main measurement
  meas.AddChannel( chan );

  // At this point, we have only given our channel
  // and measurement the input histograms as strings
  // We must now have the measurement open the files,
  // collect the histograms, copy and store them.
  // This step involves I/O
  meas.CollectHistograms();
```

```
  // Print to the screen a text representation of the model
  // just for minor debugging
  meas.PrintTree();

  // Finally, run the measurement.
  // This is the same thing that happens when
  // one runs ’hist2workspace’ on an xml files
  MakeModelAndMeasurementFast( meas );


}
```

```
#!/usr/bin/env python

#
# A pyROOT script demonstrating
# an example of writing a HistFactory
# model using python
#
# This example was written to match
# the example.xml analysis in
# $ROOTSYS/tutorials/histfactory/
#
# Written by George Lewis
#


def main():

    try:
        import ROOT
    except:
        print "It seems that pyROOT isn’t properly configured"
        return

    """
    Create a HistFactory measurement from python
    """

    InputFile = "./data/example.root"

    # Create the measurement
    meas = ROOT.RooStats.HistFactory.Measurement("meas", "meas")

    meas.SetOutputFilePrefix( "./results/example_UsingPy" )
    meas.SetPOI( "SigXsecOverSM" )
    meas.AddConstantParam("Lumi")
    meas.AddConstantParam("alpha_syst1")

    meas.SetLumi( 1.0 )
    meas.SetLumiRelErr( 0.10 )
    meas.SetExportOnly( False )

    # Create a channel

    chan = ROOT.RooStats.HistFactory.Channel( "channel1" )
    chan.SetData( "data", InputFile )
    chan.SetStatErrorConfig( 0.05, "Poisson" )

    # Now, create some samples

    # Create the signal sample
    signal = ROOT.RooStats.HistFactory.Sample( "signal", "signal", InputFile )
    signal.AddOverallSys( "syst1",  0.95, 1.05 )
    signal.AddNormFactor( "SigXsecOverSM", 1, 0, 3 )
    chan.AddSample( signal )


    # Background 1
    background1 = ROOT.RooStats.HistFactory.Sample( "background1", "background1", InputFile )
    background1.ActivateStatError( "background1_statUncert", InputFile )
    background1.AddOverallSys( "syst2", 0.95, 1.05  )
    chan.AddSample( background1 )


    # Background 1
    background2 = ROOT.RooStats.HistFactory.Sample( "background2", "background2", InputFile )
    background2.ActivateStatError()
    background2.AddOverallSys( "syst3", 0.95, 1.05  )
    chan.AddSample( background2 )


    # Done with this channel
    # Add it to the measurement:

    meas.AddChannel( chan )

    # Collect the histograms from their files,
    # print some output,
    meas.CollectHistograms()
    meas.PrintTree();

    # One can print XML code to an
    # output directory:
    # meas.PrintXML( "xmlFromCCode", meas.GetOutputFilePrefix() );

    meas.PrintXML( "xmlFromPy", meas.GetOutputFilePrefix() );

    # Now, do the measurement
    ROOT.RooStats.HistFactory.MakeModelAndMeasurementFast( meas );

    pass


if __name__ == "__main__":
    main()
```

8 Manual entries

```
man prepareHistFactory
PREPAREHISTFACTORY(1)                                                   PREPAREHISTFACTORY(1)

NAME
       prepareHistFactory - create a working directory for the HistFactory tools

SYNOPSIS
       prepareHistFactory [dir_name]

DESCRIPTION
       prepareHistFactory  is  a simple script that prepares a working area (and creates the directory
       dir_name if specified).  Within the directory dir_name, it creates a results/, data/, and  con-
       fig/  directory relative to the given path.  It also copies the HistFactorySchema.dtd and exam-
       ple XML files into the config/ directory.  Additionally, it copies a root file into  the  data/
       directory  for  use  with  the  examples.   Once  this is done, one is ready to run the example
       hist2workspace input.xml or edit the XML files for a new project.

ORIGINAL AUTHORS
              Dominique Tardif
               and Kyle Cranmer

COPYRIGHT
       This library is free software; you can redistribute it and/or modify it under the terms of  the
       GNU  Lesser General Public License as published by the Free Software Foundation; either version
       2.1 of the License, or (at your option) any later version.

       This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; with-
       out  even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
       GNU Lesser General Public License for more details.

       You should have received a copy of the GNU  Lesser  General  Public  License  along  with  this
       library;  if  not,  write  to  the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor,
       Boston, MA  02110-1301  USA

                                               DEC. 2010                        PREPAREHISTFACTORY(1)
```

```
man hist2workspace
HISTTOWORKSPACE(1)                                                         HISTTOWORKSPACE(1)

NAME
       hist2workspace - utility to create RooFit/RooStats workspace from histograms

SYNOPSIS
       hist2workspace [option] input.xml

DESCRIPTION
       hist2workspace is a utility to create RooFit/RooStats workspace from histograms

OPTIONS
       -standard_form  default model, which creates an extended PDF that interpolates between RooHist-
       Funcs.  This is much faster for models with many bins and uses significantly less memory.
       -number_counting_form this was the original model in 5.28 (without patches). It uses a  Poisson
       for  each  bin of the histogram.  This can become slow and memory intensive when there are many
       bins.

Prepare working area
       The ROOT release ships with a script prepareHistFactory in the $ROOTSYS/bin directory that pre-
       pares a working area.  It creates a results/, data/, and config/ directory.  It also copies the
       HistFactorySchema.dtd and example XML files  into  the  config/  directory.   Additionally,  it
       copies a root file into the data/ directory for use with the examples.

HistFactorySchema.dtd
       This  file is located in $ROOTSYS/etc/ specifies the XML schema.  It is typically placed in the
       config/ directory of a working area together with the top-level XML  file  and  the  individual
       channel XML files.  The user should not modify this file.
       The HistFactorySchema.dtd is commented to specify exactly the meaning of the various options.

Top-Level XML File
       (see  for  example $ROOTSYS/tutorials/histfactory/example.xml) This file is edited by the user.
       It specifies
        - A top level 'Combination' that is composed of:
           - several 'Channels', which are described in separate XML files.
           - several 'Measurements' (corresponding to a full fit of the model) each of which specifies
             - a name for this measurement to be used in tables and files
             - what is the luminosity associated to the measurement in picobarns
             - which bins of the histogram should be used
             - what is the relative uncertainty on the luminosity
             - what is (are) the parameter(s) of interest that will be measured
             - which parameters should be fixed/floating (eg. nuisance parameters)
             -  which type of constriants are desired      - Gaussian by default      - Gamma, LogNor
             mal, and Uniform are also supported
             - if the tool should export the model only and skip the default fit

Channel XML Files
       (see for example $ROOTSYS/tutorials/histfactory/example_channel.xml) This file is edited by the
       user.  It specifies for each channel
        - observed data
          - if absent the tool will use the expectation, which is useful for expected sensitivity
        - several 'Samples' (eg. signal, bkg1, bkg2, ...), each of which has:
          - a name
          - if the sample is normalized by theory (eg N = L*sigma) or not (eg. data driven)
          - a nominal expectation histogram
          - a named 'Normalization Factor' (which can be fixed or allowed to float in a fit)
          - several 'Overall Systematics' in normalization with:
            - a name
            - +/- 1 sigma variations (eg. 1.05 and 0.95 for a 5% uncertainty)
          - several 'Histogram Systematics' in shape with:
            - a name (which can be shared with the OverallSyst if correlated)
            - +/- 1 sigma variational histograms

ORIGINAL AUTHORS
              Kyle Cranmer , Akira Shibata , and Dominique Tardif

COPYRIGHT
       This  library is free software; you can redistribute it and/or modify it under the terms of the
       GNU Lesser General Public License as published by the Free Software Foundation; either  version
       2.1 of the License, or (at your option) any later version.

       This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; with-
       out even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See  the
       GNU Lesser General Public License for more details.

       You  should  have  received  a  copy  of  the GNU Lesser General Public License along with this
       library; if not, write to the Free Software Foundation, Inc.,  51  Franklin  St,  Fifth  Floor,
       Boston, MA  02110-1301  USA

                                            DEC. 2010                              HISTTOWORKSPACE(1)
```