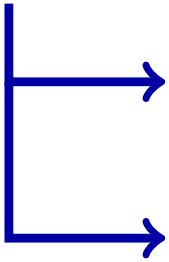
HistFitter

Jeanette Lorenz (LMU München/Excellence Cluster Universe)
Max Baak (CERN)
Geert-Jan Besjes (Radboud University Nijmegen/Nikhef)
David Côté (University of Texas)
Alex Koutsman (TRIUMF)
Dan Short (University of Oxford)

01.09.2014 / ACAT 2014

Introduction

HistFitter: software framework for statistical data analysis.



Built on top of HistFactory/RooFit (construction of parametric models) and RooStats (statistical tests of data)

Consists of a Python part for configuration and a C++ part for CPU-intensive calculations

HistFitter extends RooFit/HistFactory/RooStats in four key areas:

- Programmable framework: performing complete statistical analyses, using a user-defined configuration file
- Analysis strategy: Concepts of analysis control, validation and signal regions deeply woven into the design of HistFitter
- Bookkepping: HistFitter keeps track of numerous data models - including construction and statistical tests of all of them in an organized way
- Presentation and interpretation: Collection of tools to: determine the statistical significance of signal hypotheses, estimate the quality of likelihood fits, produce tables and plots expressing the results

HistFitter is used in numerous analyses (mostly searches for SUSY) of the ATLAS Collaboration.

Outline

Data analysis strategy

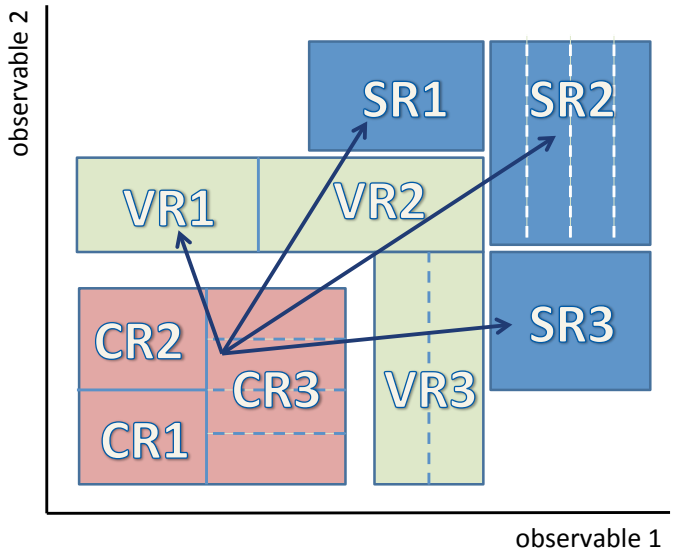
Particle physics experiments analyze large data samples in order to measure properties of fundamental particles and to discover new physical processes.

Data interpreted using external predictions for background and signal components.

→ **HistFitter configures and builds parametric models to describe the observed data, and provides tools to interpret the data.**

Construction and handling of models based on the concept of signal, control and validation regions:

- Signal regions: region in phase space in which a particular signal model predicts an significant excess over the background level
- Control regions: used to estimate background in the signal regions in a semi-data-driven way via extrapolation; dominant backgrounds can be controlled in the control regions through comparison to data
- Validation regions: validation of the extrapolation; regions typically placed between control and signal regions



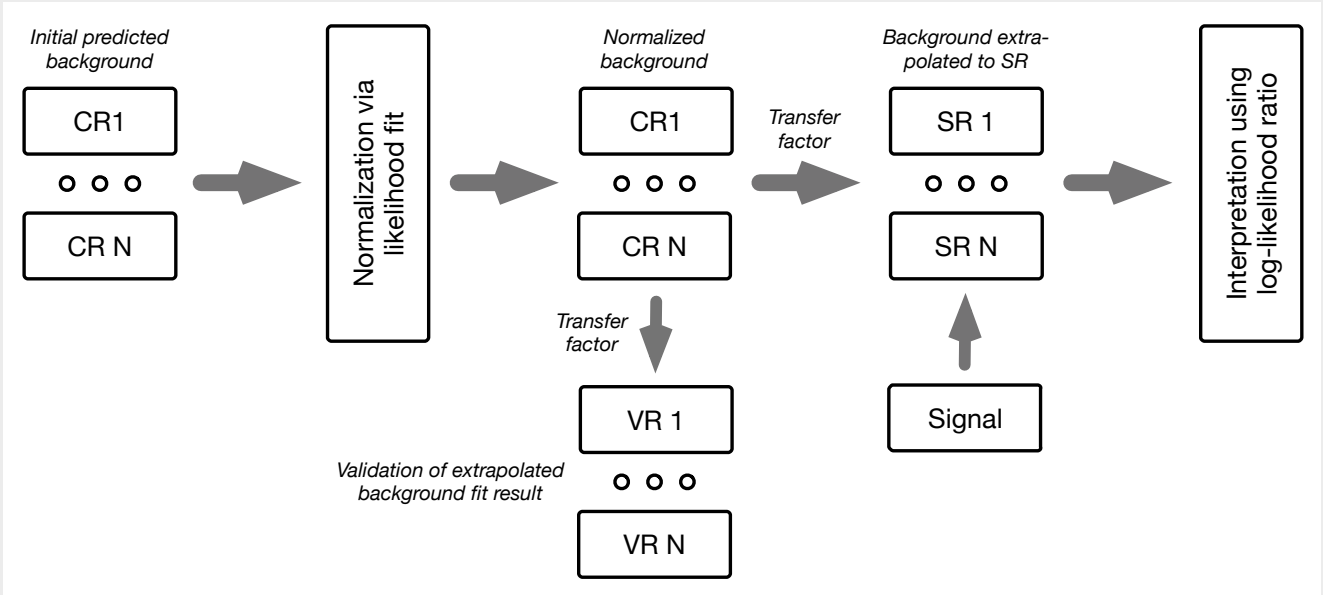
Concept deeply woven into design of HistFitter.

Typical analysis strategy with HistFitter

Model represented by a Probability Density Function (PDF):

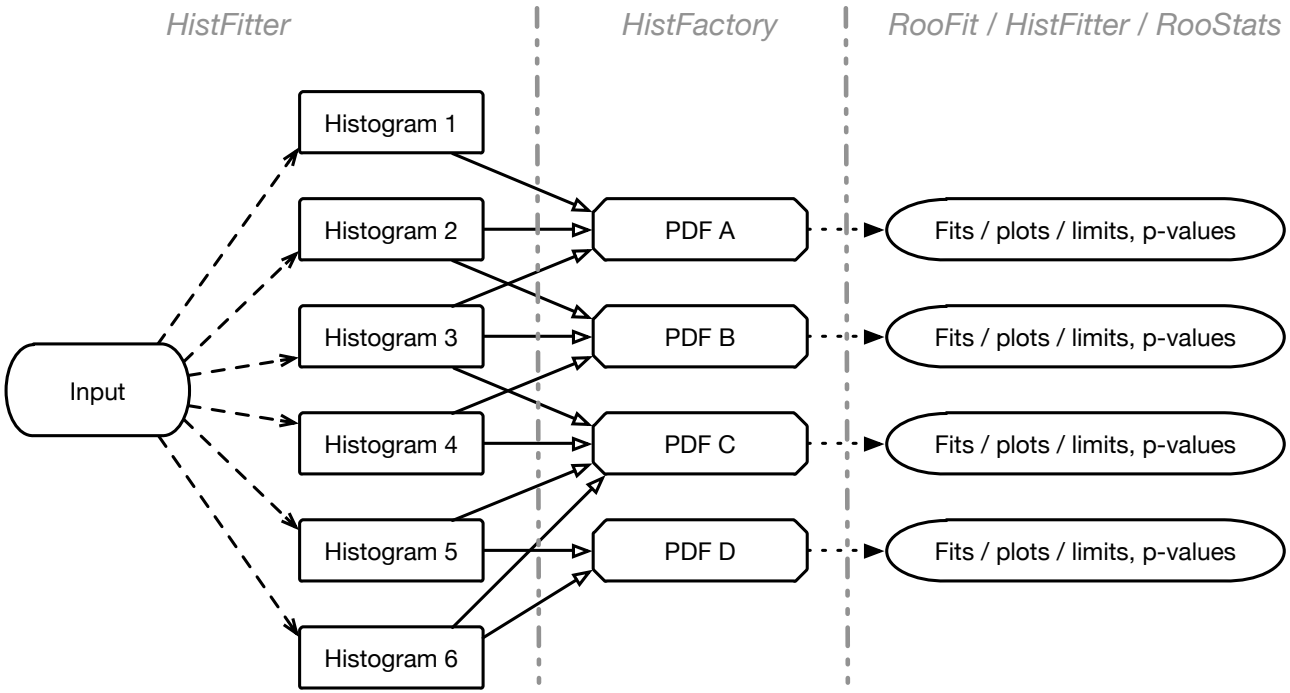
- Parameters are adjusted by a likelihood fit
- Control, signal and validation regions are statistically independent - each region can be modeled by a separate PDF that can thus be combined in a simultaneous fit to all regions
- PDF parameters can be shared in different regions, thus all information about background, signal, systematic uncertainty can be used consistently in all regions

- Background normalized to data in a fit to data in control regions
- Extrapolate to validation or signal regions using transfer factors (ratio of expected event count between each control and validation/signal region)



HistFitter software framework

Based on user-defined configuration and raw data as input, the processing sequence of HistFitter consists of three steps:



Histogram construction from input data and configuration

Construction of PDFs from binned histograms

Analysis of models

Configuration and bookkeeping

Substantial bookkeeping and configuration machinery required for presented analysis flow (in particular if working with multiple different signal hypotheses)

Realized through a user-defined Python configuration file which interacts with a configuration Manager within HistFitter.

Benefits of using a configuration file for an analysis:

- Simplifies collaboration within an analysis team
- Allows to rerun analysis quickly once histograms are built

Technical side: configuration manager

- Realized as two singleton objects in Python and in C++
  - The user interacts with the Python configuration manager
- The configuration Manager can be understood as “factories of factories”: it organizes and creates so-called `fitConfig` objects
  - The `fitConfig` objects contain the PDF of the studied model along with meta-data giving additional information about the construction, fitting, visualizing and interpretation of the model.
  - A `fitConfig` object is thus an own “factory” and represents one row on the last slide.

Benefit of the configuration Manager: histogram recycling - storing of unique auto-generated names for each histogram used in the construction of the PDF in a python dictionary. Histograms can thus be reused if appearing in different models.

Construction of PDFs

Construction of parametric models as PDFs via HistFactory from binned input histograms

General form of constructed likelihood:

$$L(\boldsymbol{n}, \boldsymbol{\theta}^0 | \mu_{\text{sig}}, \boldsymbol{b}, \boldsymbol{\theta}) = P_{\text{SR}} \times P_{\text{CR}} \times C_{\text{syst}}$$

$\rightarrow$ product of Poisson distributions of event counts in signal and control regions (P_{SR} and P_{CR}) and of additional constraint terms for systematic uncertainties (C_{syst})

Likelihood depends on number of observed events in all regions ($\boldsymbol{n}$), nuisance parameters parametrizing the impact of systematic uncertainties ($\boldsymbol{\theta}$) with their central values $\boldsymbol{\theta}^0$, signal strength μ_{sig} and predictions $\boldsymbol{b}$ for various background sources.

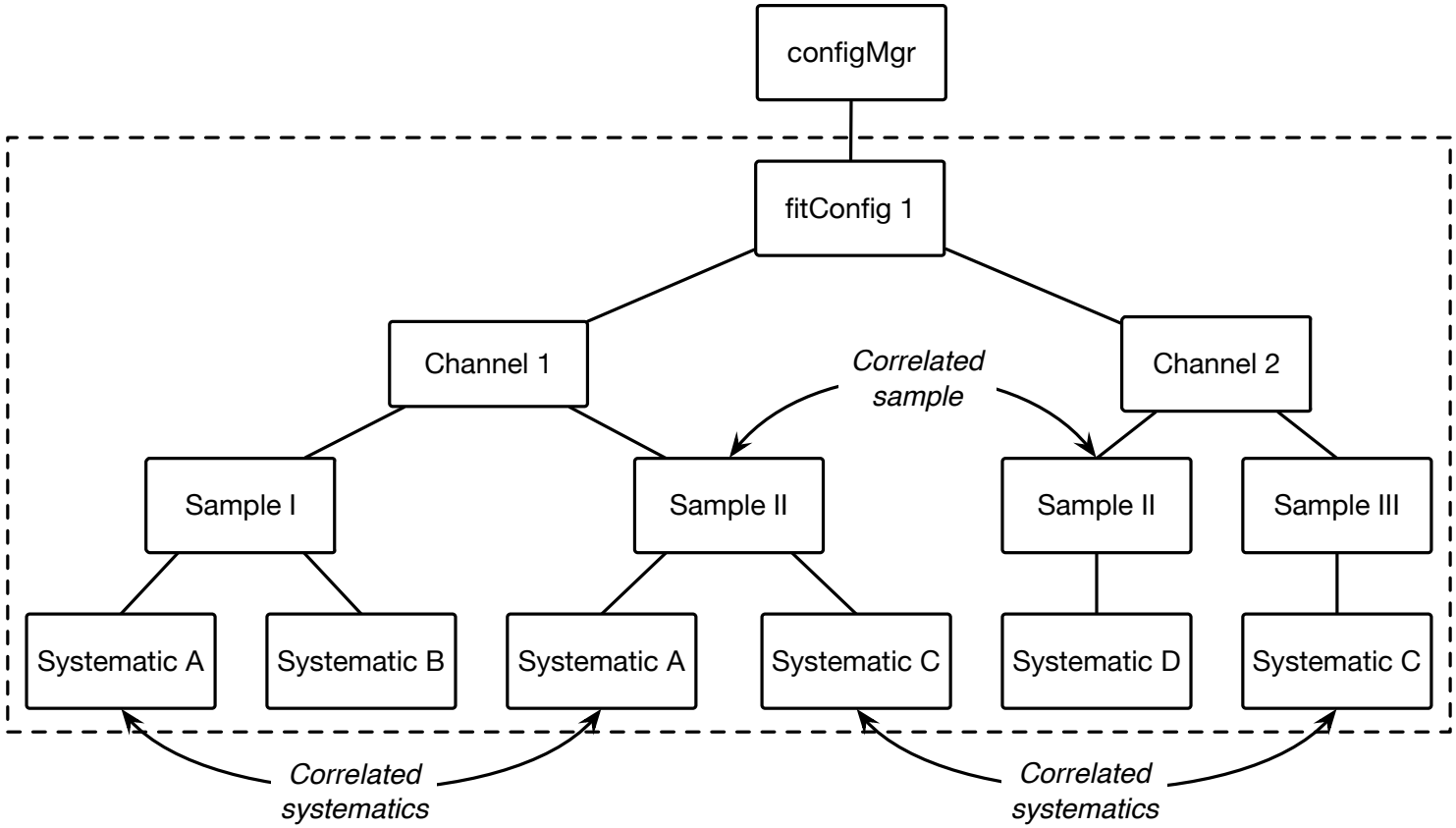
The likelihood thus has multiple building blocks:

- Control, validation, signal regions: called `channel` in HistFitter context
- Signal and background processes: called `samples`
- `Systematic` uncertainties, including statistical, theoretical and experimental uncertainties

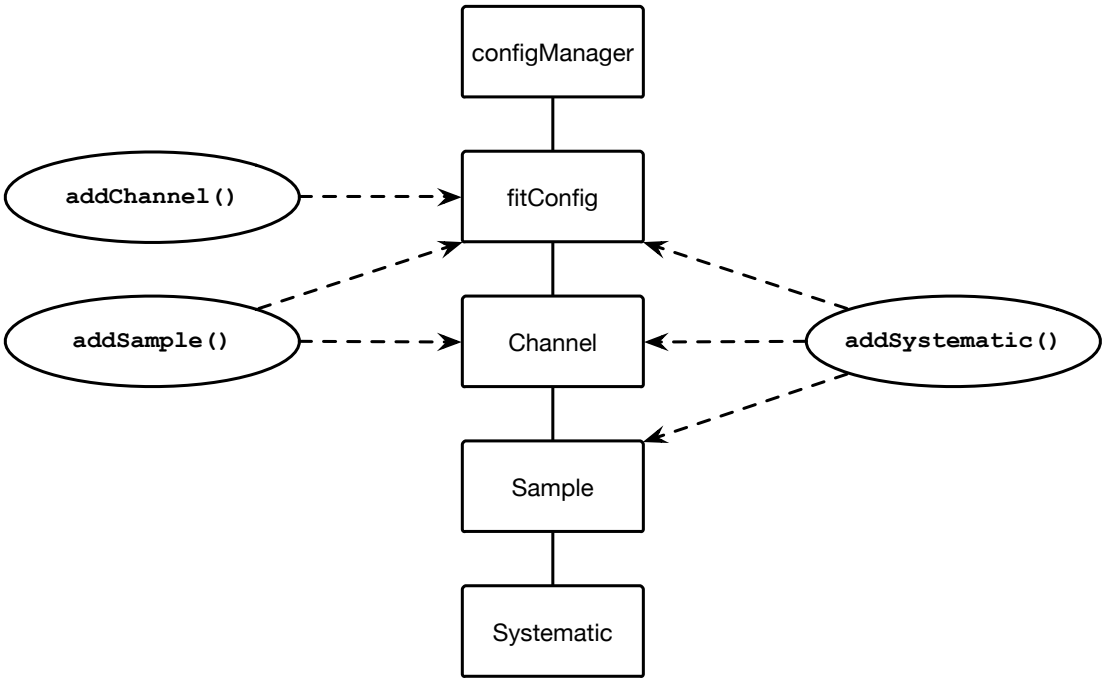
Original HistFactory classes mirrored and extended in HistFitter (in Python)

Fit configuration through `fitConfig` object

The `fitConfig` objects summarize channels, samples and systematics together with links to input histograms (representing the input data)



“Trickle-down” mechanism



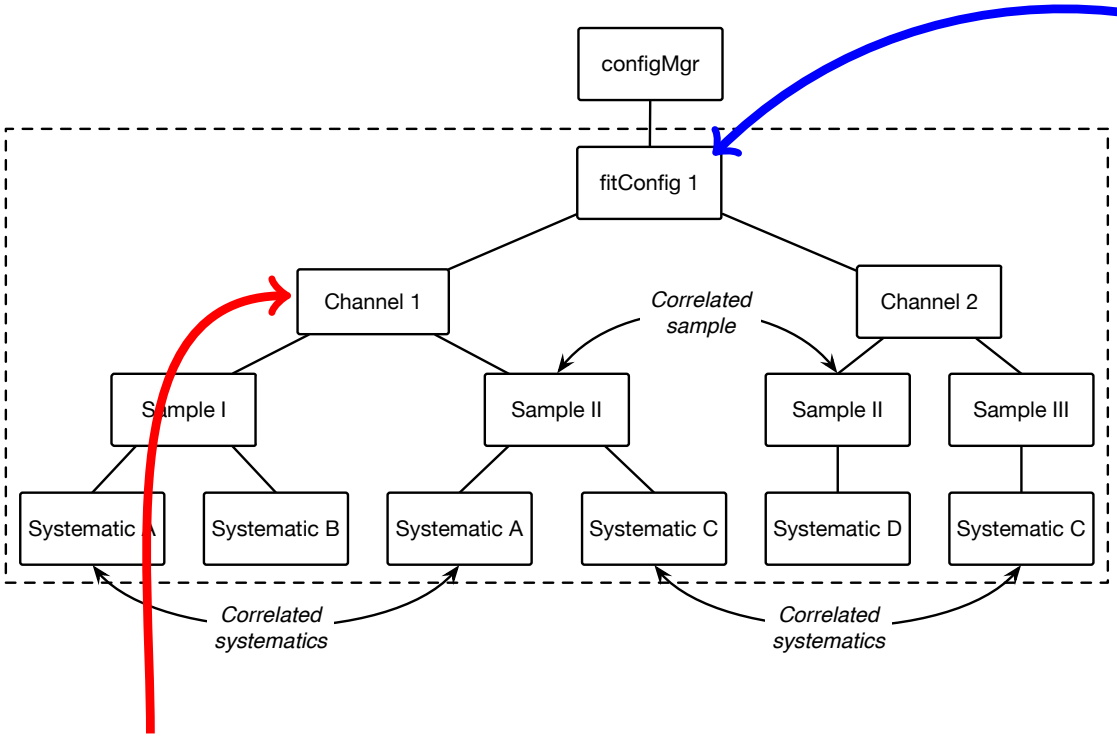
`Channels` are added to a `fitConfig`.

Samples can be added to either a `fitConfig` or a `channel`. If adding to `fitConfig` also added to all depending `channels`.

Similar for **systematics**. Can be added to `fitConfig` and then propagated to all `channels` and `samples`. Or added to a `channel` and then propagated to all depending `samples`. Or just added to a specific `sample`.

$\Rightarrow$ A complicated PDF can be described by few lines of code.

Further properties of the fit configuration

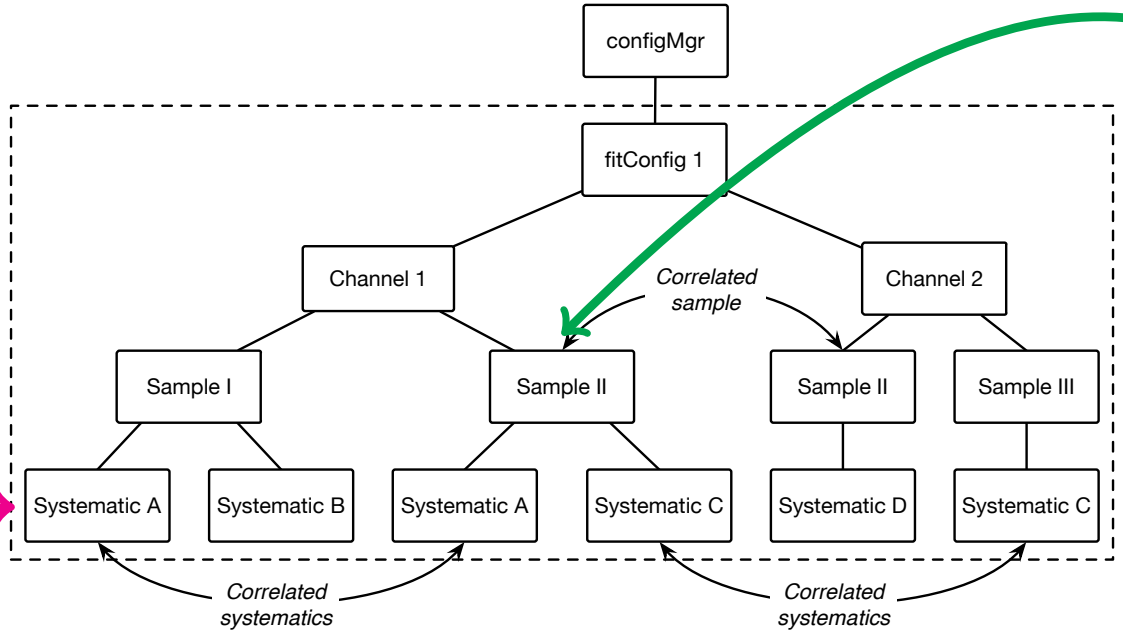


Basic fit configuration can be cloned and extended to more complicated configurations

e.g. adding signal regions and signal to a background-only model only containing control regions.

- Channels have either one bin or derive from multi-bin histograms.
- Property set: the channel is a control, validation or signal region.

Further properties of the fit configuration



`Sample`:

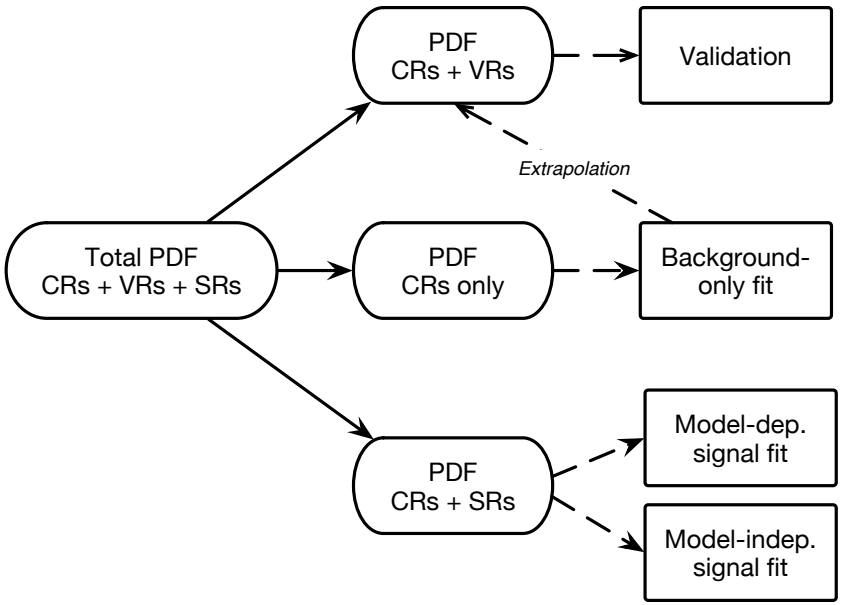
- Corresponds to a component of RooFitPDF decorated with HistFitter meta-data.
- Input either ROOT TTree, ROOT TH1 histograms or raw float numbers.
- Samples can be correlated between multiple channels by setting the same name.

`Systematic` class object:

- Systematic uncertainties typically provided as 1σ up and down variations of a nominal histogram.
- Input either ROOT TTree, ROOT TH1 histograms or raw float numbers.
- Systematic uncertainties can be correlated between different samples and channels by assigning them the same name.
- Different types available (differing in the constraint parametrization and interpolation/extrapolation type).

$\rightarrow$ HistFitter extends here the types present in HistFactory (`overallSys`, `histoSys`, `shapeSys`) by further composite and extended types.

Fit strategy



Fit setup	Background-only fit	Model-dependent signal fit	Model-ind ependent signal fit
Samples used	backgrounds	backgrounds + signal	backgrounds + dummy signal
Fit regions	CR(s)	CR(s) + SR(s)	CR(s) + SR

HistFitter provides different fit strategies:

- **Background-only fit:**
  - To estimate background yields in validation and signal regions.
  - Model only in control regions fitted, extrapolation to validation and signal regions.
  - No signal included.
- **Model-dependent signal fit:**
  - To set exclusion limits on a specific signal model in absence of an excess in the signal regions or to measure its properties in case of an excess.
  - Control and signal regions used simultaneously in the fit and a signal sample included in all regions.
  - Simultaneous use of multiple signal regions possible (and of binned histograms in them); thus signal constrained in multiple signal regions and/or histogram bins (“shape fit”) $\rightarrow$ usually sensitivity increase.
- **Model-independent signal fit:**
  - To obtain model-independent upper limits on the number of events beyond the expected number of events in a certain signal region.
  - Control and signal regions used, but signal only present in the unbinned signal region

Extrapolation and error propagation

Extrapolation into validation and signal regions:

- For background-only fit only likelihood in control regions required and used in fitting.
- However, construction of full likelihood containing control and validation/signal regions.
- Deconstruction of full likelihood into smaller likelihood only containing control regions for use in the background-only fit.
- Incorporation of fitted parameters after background-only fit into full likelihood describing control and (validation)signal regions.
- Evaluation of the extrapolated uncertainty in validation/signal regions through standard error propagation.

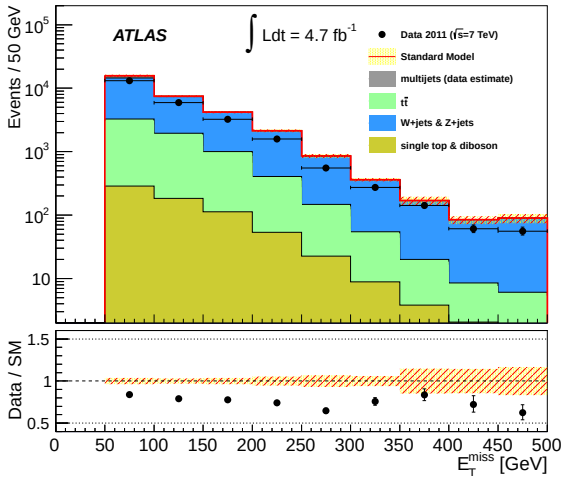
Extrapolation into signal and validation regions particularly rigorous in HistFitter due to use of `RooExpandedFitResult` class:

- Standard `RooFitResult` contains only the parameters used in the background-only fit in the control regions.
- Instead using the `RooExpandedFitResult` class allows to extrapolate all parameters used in the background-only fit and the parameters not used, such that a correct evaluation of the uncertainties through error propagation is possible.

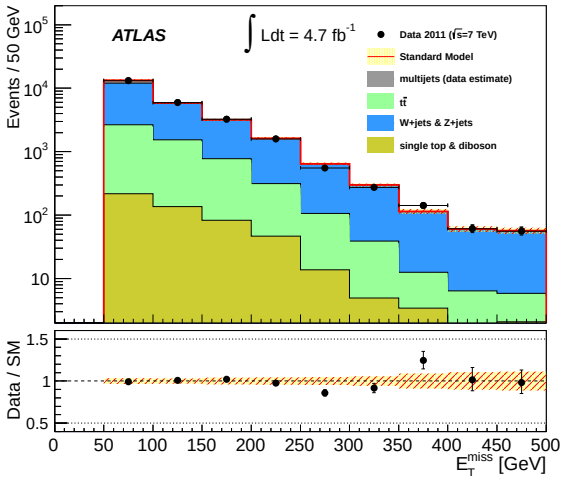
Presentation of results

HistFitter includes a collection of tools and functions to aid the presentation of the results:

1. Visualization of fit results in before and after-fit distributions and in pull plots.



(before the fit)

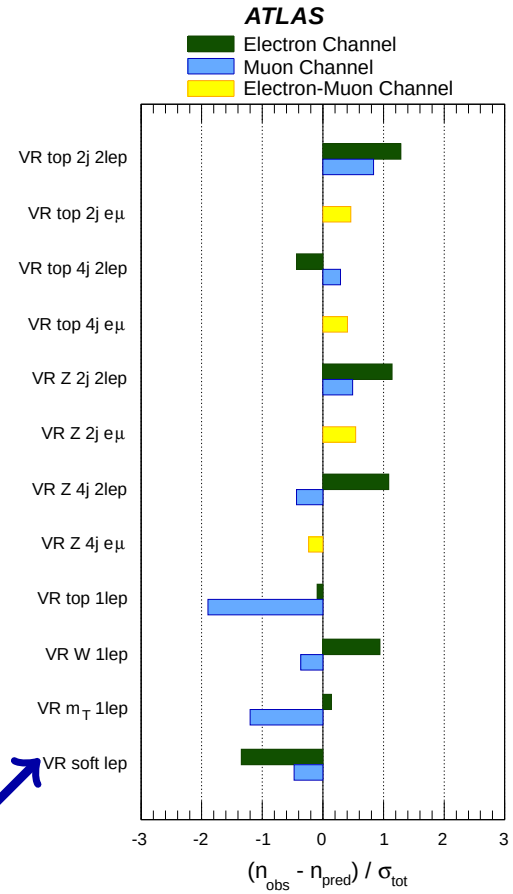


(after the fit)

Pull plot:

$$\chi = \frac{n_{\text{obs}} - n_{\text{pred}}}{\sigma_{\text{tot}}}$$

$$\sigma_{\text{tot}} = \sqrt{\sigma^2_{\text{pred}} + \sigma^2_{\text{stat, exp}}}$$



Presentation of results

HistFitter includes a collection of tools and functions to aid the presentation of the results:

2. Scripts for producing event yields and uncertainty tables.

Process	Signal Region				
	SR-A tight	SR-B tight	SR-C tight	SR-D tight	SR-E tight
$t\bar{t}$+single top	0.2 ± 0.2 (0.1)	0.3 ± 0.3 (0.2)	2.0 ± 1.5 (1.2)	2.4 ± 1.7 (1.4)	4.2 ± 4.7 (3.0)
Z+jets	3.3 ± 1.5 (4.0)	2.0 ± 1.3 (2.1)	2.0 ± 1.0 (5.6)	0.9 ± 0.6 (3.4)	3.4 ± 1.6 (2.3)
W+jets	2.2 ± 1.0 (1.9)	1.0 ± 0.6 (0.8)	1.5 ± 1.3 (2.7)	2.4 ± 1.4 (2.5)	2.8 ± 1.9 (1.5)
Multi-jets	0.00 ± 0.02 (0.01)	0.00 ± 0.07 (0.02)	0.00 ± 0.03 (0.01)	0.0 ± 0.3 (0.1)	0.5 ± 0.4 (0.9)
Di-bosons	1.8 ± 0.9 (2.0)	1.8 ± 0.9 (1.9)	0.5 ± 0.3 (0.5)	2.2 ± 1.1 (2.2)	2.5 ± 1.3 (2.5)
Total	$7.4 \pm 1.3 \pm 1.9$	$5.0 \pm 0.9 \pm 1.7$	$6.0 \pm 1.0 \pm 2.0$	$7.8 \pm 1.0 \pm 2.4$	$13 \pm 2 \pm 6$
Data	1	1	14	9	13

Signal Region	SR3b	SR0b	SR1b	SR3Llow	SR3Lhigh
Observed events	1	14	10	6	2
Total expected background events	2.2 ± 0.8	6.5 ± 2.3	4.7 ± 2.1	4.3 ± 2.1	2.5 ± 0.9
Systematic uncertainties on expected background					
Fake-lepton background	± 0.6	$^{+1.5}_{-1.2}$	$^{+1.2}_{-0.8}$	± 1.6	< 0.1
Theory unc. on dibosons	< 0.1	± 1.5	± 0.3	± 0.4	± 0.4
Jet and $E_{\mathrm{T}}^{\mathrm{miss}}$ scale and resolution	± 0.1	± 0.7	± 0.4	± 0.4	± 0.3
Monte Carlo statistics	± 0.1	± 0.5	± 0.2	± 0.4	± 0.4
b-jet tagging	± 0.2	± 0.5	± 0.1	< 0.1	± 0.1
Theory unc. on $t\bar{t}V$, $t\bar{t}H$, tZ and $t\bar{t}t\bar{t}$	± 0.4	± 0.3	± 1.7	± 1.0	± 0.6
Trigger, luminosity and pile-up	< 0.1	± 0.1	± 0.1	± 0.1	± 0.1
Charge-flip background	± 0.1	± 0.1	± 0.1	–	–
Lepton identification	< 0.1	± 0.1	< 0.1	± 0.1	± 0.1

Two methods available for the systematics tables:

- Method 1: Calculating the uncertainty propagated to the background prediction by a specific parameter.
- Method 2: Excluding a (or multiple) specific parameters from the fit and refit - the impact of the parameter is then given by $\sigma_{\eta_1} = \sqrt{\left(\sigma_{\mathrm{tot}}^{\mathrm{nominal}}\right)^2 - \left(\sigma_{\mathrm{tot}}^{\eta_1 = C}\right)^2}$

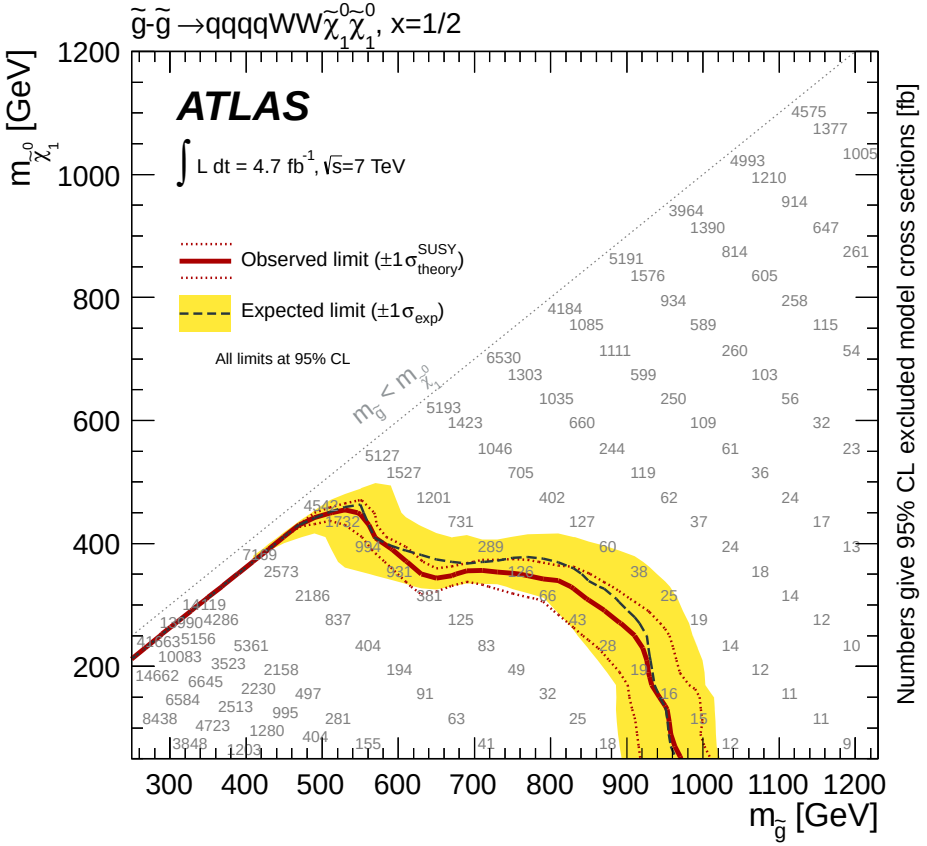
Interpretation signal model dependent

Hypothesis tests for different assumptions and models

HistFitter provides various interpretations/hypothesis tests through calls to the appropriate RooStats functions and classes and offers macros for interpreting the results in plots and tables.

Based on a specific signal model:

- **Signal model hypothesis test:**
  - Exclusion limits for a certain signal model.
  - Often multiple hypothesis tests for different signal models, e.g. scanning over a certain parameter of a new physics.
  - Using model-dependent signal limit fit.
  - HistFitter executes the hypothesis test(s), collects the results, can convert the information into histograms and provides plotting macros for those.
- **Signal strength upper limit:**
  - Using model-dependent signal limit fit.
  - Multiple hypothesis tests for a certain signal model, testing different signal strengths, to determine upper limit on 95% CL excluded upper limit.



Interpretation signal model independent

Interpretations without the dependency on a specific signal model:

- **Model-independent upper limit:**
  - To calculate the 95% CL upper limit on the number of events for any kind of new physics, so without a specific model dependency.
  - Using the model-independent fit configuration and an upper limit scan as in the signal strength upper limit.
  - HistFitter includes a script for calculating of these upper limits including a presentation of them in a table.
- **Background-only hypothesis test (discovery p-value):**
  - Significance of an excess of events in the signal region: probability that a background-only experiment is more signal-like than observed.

Signal channel	$\langle\sigma_{\mathrm{vis}}\rangle^{95}_{\mathrm{obs}}$[fb]	S^{95}_{rmobs}	S^{95}_{exp}	$p(s=0)$
SR3b	0.19	3.9	$4.4^{+1.7}_{-0.6}$	0.50
SR0b	0.80	16.3	$8.9^{+3.6}_{-2.0}$	0.03
SR1b	0.65	13.3	$8.0^{+3.3}_{-2.0}$	0.07
SR3Llow	0.42	8.6	$7.2^{+2.9}_{-1.3}$	0.29
SR3Lhigh	0.23	4.6	$5.0^{+1.6}_{-1.1}$	0.50

Summary

Presented the software framework HistFitter which is tailored for statistical analysis.

...programmable framework to build and test data models of nearly arbitrary complexity.

...starting from user-defined input configuration file, and by using HistFactory, RooStats, RooFit, the tool constructs and fits PDFs and provides interpretations by statistical tests

Innovative features:

- Modular configuration interface with trickle-down mechanism which eases the construction of complicated PDFs.
- Built-in concepts of control, validation and signal regions with a particular rigorous statistical treatment for the extrapolation.
- Designed and providing the bookkeeping to work with multiple signal models at once and thus provides an additional level of abstraction.
- Sizable collection of tools and options for presenting end results with a publication-style quality.

Backup

Different possibilities of implementing systematic uncertainties

Various types available in HistFitter:

Basic systematic methods in HistFactory	
`overallSys`	uncertainty of the global normalization, not affecting the shape
`histoSys`	correlated uncertainty of shape and normalization
`shapeSys`	uncertainty of statistical nature applied to a sum of samples, bin by bin
Additional systematic methods in HistFitter	
`overallNormSys`	overallSys constrained to conserve total event count in a list of region(s)
`normHistoSys`	histoSys constrained to conserve total event count in a list of region(s)
`normHistoSysOneSide`	one-sided normHistoSys uncertainty built from tree-based or weight-based inpu ts
`normHistoSysOneSideSym`	symmetrized normHistoSysOneSide
`overallHistoSys`	factorized normalization shape and uncertainty, described with `overallSys` and `histoSys` respectively
`overallNormHistoSys`	overallHistoSys in which the shape uncertainty is modeled with a normHistoSys and the global normalization uncertainty is modeled with an overallSys
`shapeStat`	shapeSys applied to an individual sample

Sub-set of the systematic methods available in HistFitter. The methods are specified by a string argument containing a combination of basic HistFactory methods and optional HistFitter keywords: `norm`, `OneSide` and/or `Sym`. Systematic objects can be built with Tree-based, weight-based, Float or histogram input methods in all cases.