

HistFitter

A flexible framework for statistical data analysis

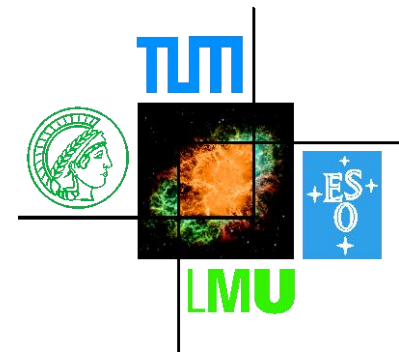
Jeanette Lorenz (LMU Muenchen/Excellence Cluster Universe)

Max Baak (CERN)

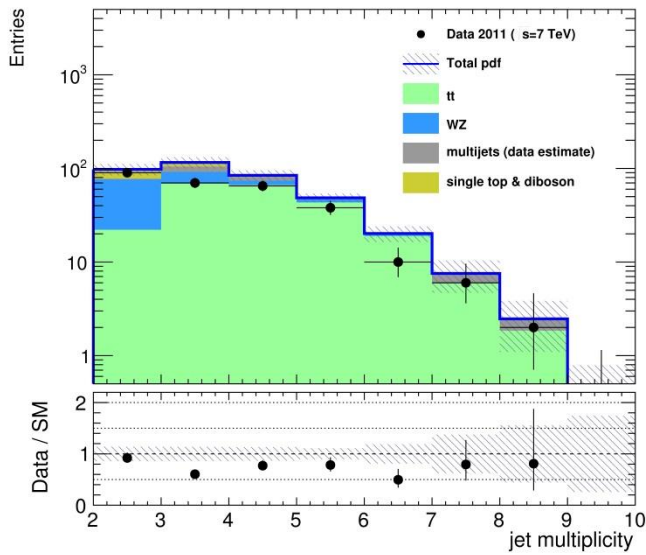
Geert-Jan Besjes (University of Copenhagen)

David Côté (University of Texas at Arlington)

Aleksej Koutsman (TRIUMF)



EPS-HEP Vienna 2015, 24.7.2015

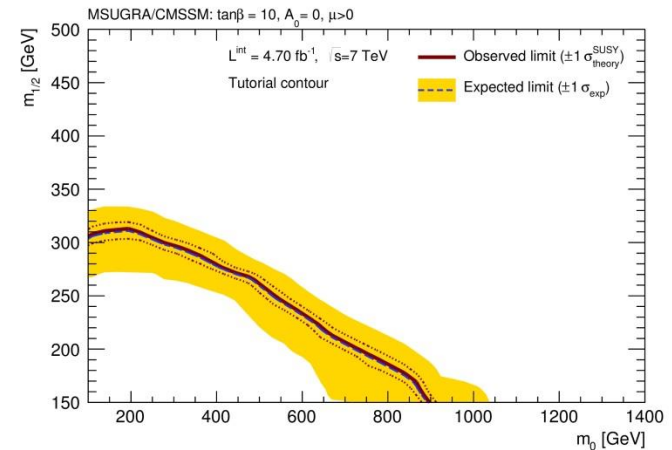
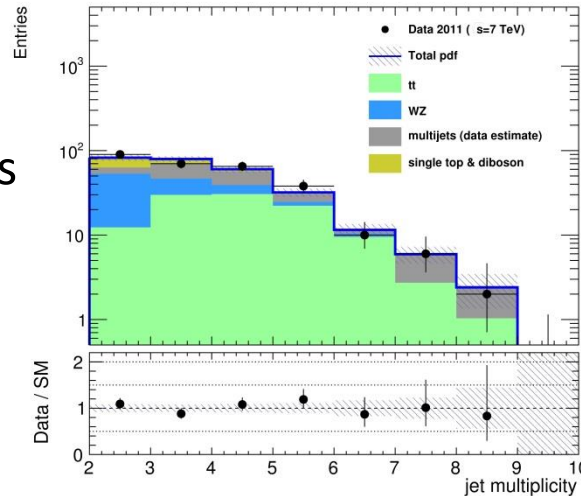


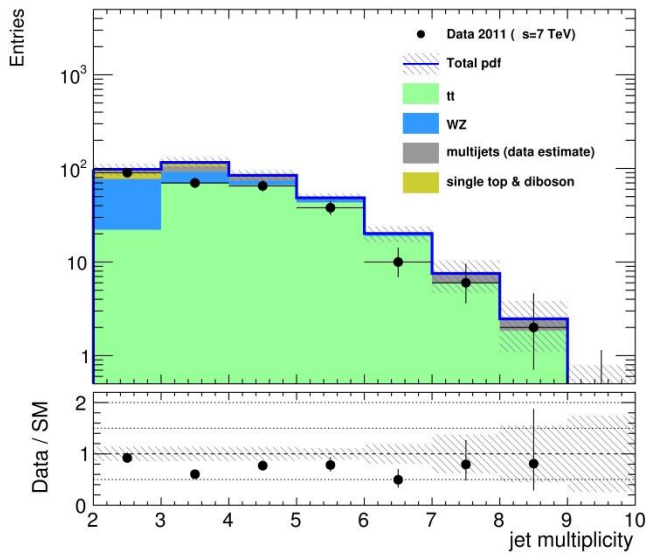
Typical input by a particle physics analysis...

... some complicated steps in between...

... results:

- Kinematic distributions after a likelihood fit
- Interpretations or measurements
- ...



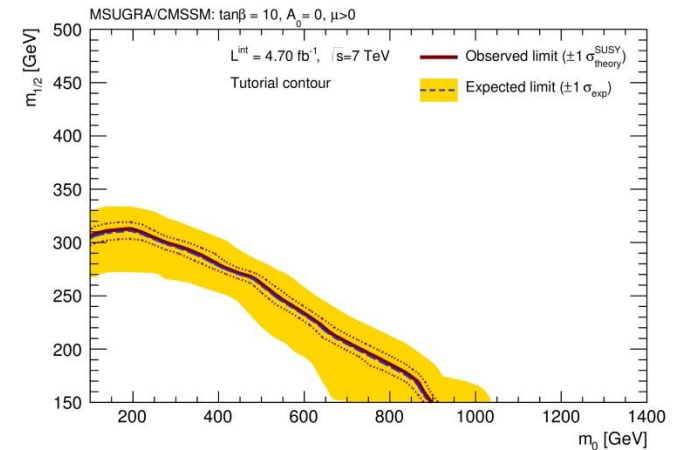
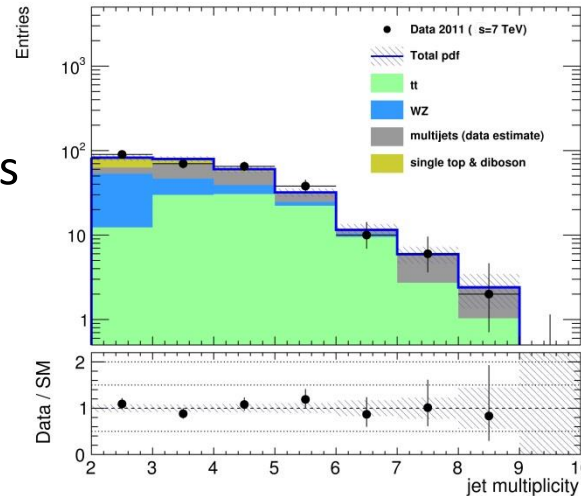


Typical input by a particle physics analysis...

HistFitter

... results:

- Kinematic distributions after a likelihood fit
- Interpretations or measurements
- ...



Overview

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
- Bookkeeping:
HistFitter keeps track of numerous data models - including construction and statistical tests of all of them in an organized way
- Presentation and interpretation:
Easy-to-use tools to present data and interpret results (statistical significances; quality of likelihood fits; tables and plots summarising the results; etc.)

HistFitter used in numerous analyses (e.g. SUSY searches) of the ATLAS Collaboration at the LHC.

Data analysis strategy

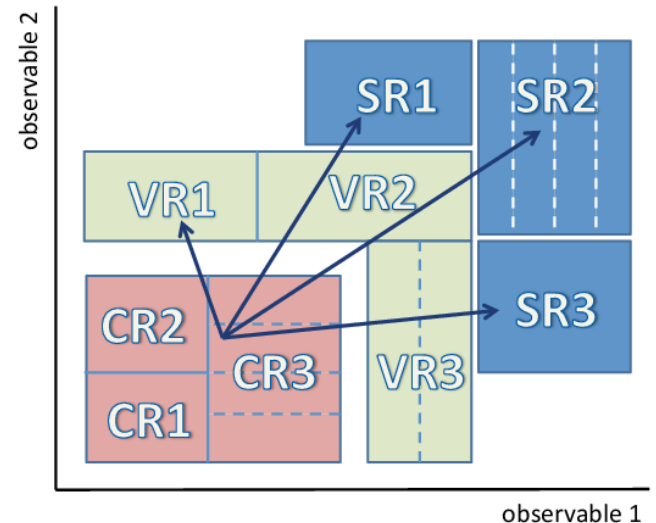
- Particle physics experiments analyze **large** data samples in order to **measure** properties of fundamental particles and to **discover** new physical processes.
- Data interpreted using **external predictions** for background and signal components.



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:** signal-rich region (SR)
- **Control regions:** background-rich region, fit simulated backgrounds to data (CR)
- **Validation regions:** validation of extrapolation (VR)



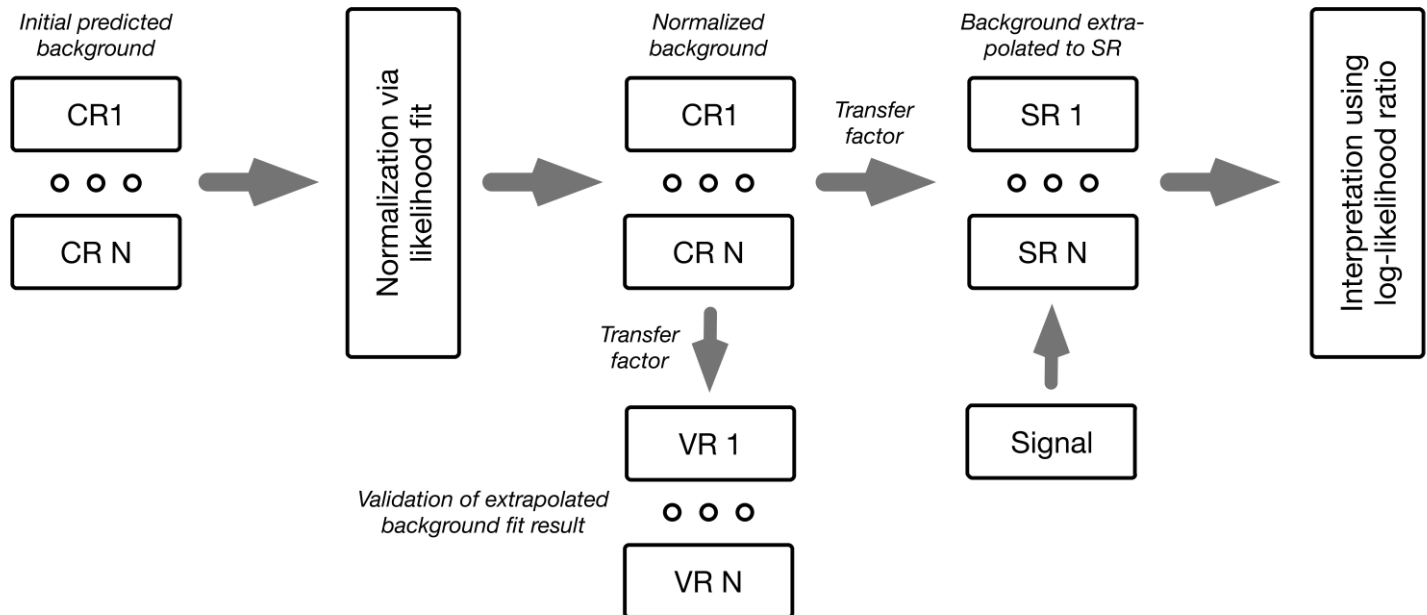
Concept deeply woven into design of HistFitter

Typical analysis strategy with HistFitter

Model represented by a Probability Density Function (PDF):

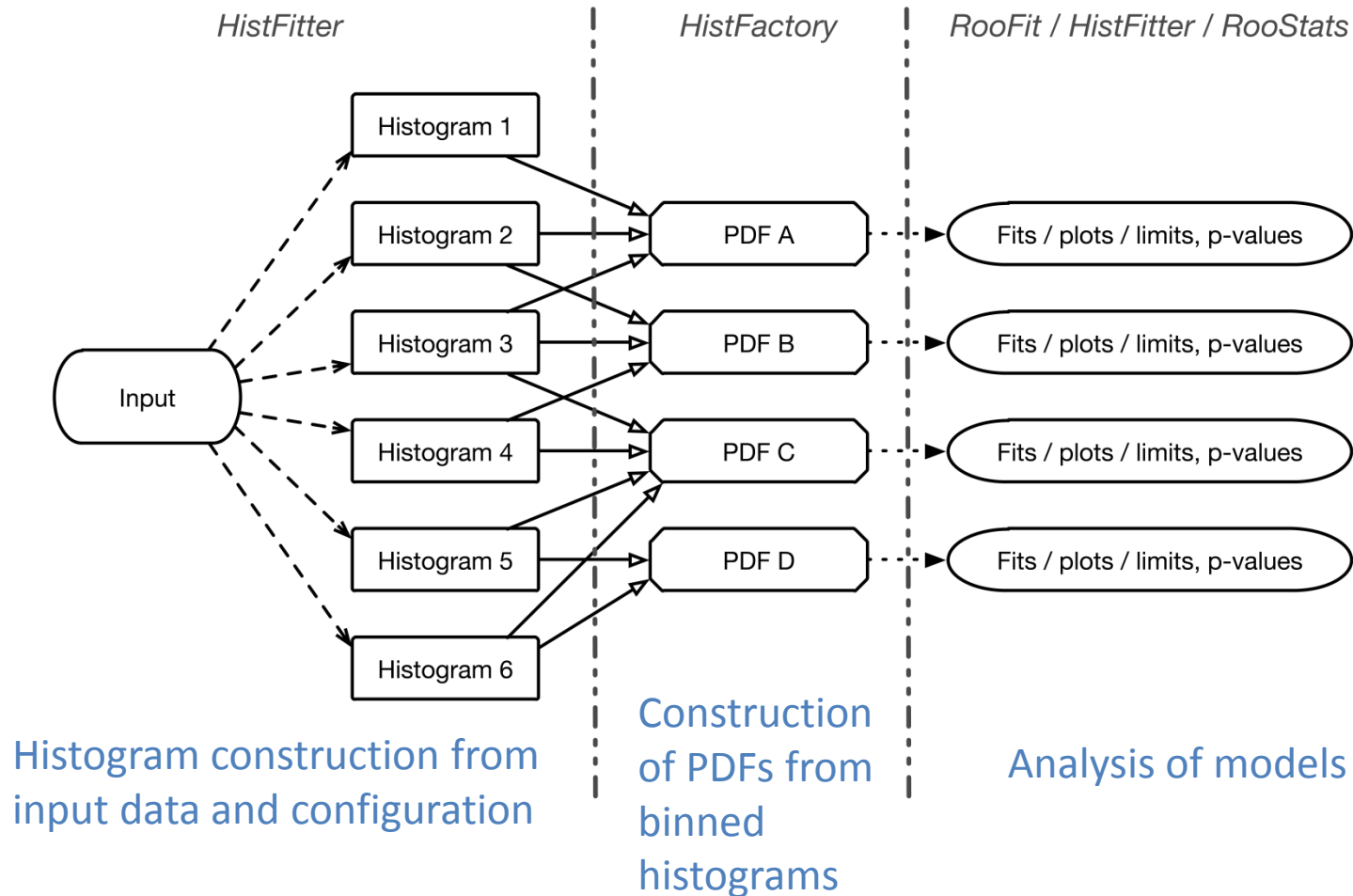
- Include normalisation and nuisance parameters (systematic errors).
- Parameters are adjusted by a likelihood fit.
- Each CR/VR/SR modeled by separate PDF, combined in simultaneous fit.
- PDF parameters can be shared in different regions.

- 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:



Analysis configuration

An analysis – e.g. search for new physics can involve ~10 signal regions, even more control regions and ~ 1000 different models of new physics to test

➡ Substantial **bookkeeping** and **configuration** machinery required to manage all correctly!

HistFitter simplifies this using **one** Python configuration script: implements a **configuration manager** (two related singletons in Python and C++) that have **fit configurations** (fit config)

- Also automatic re-use of shared data to save time and memory

Fit configuration:

- Contains the PDF of a specific statistical model + meta-data (e.g. information on plotting)

PDF:

- Parametric model constructed from binned input histograms using HistFactory.
- Likelihood depending on the **number of observed events** in all regions (n), nuisance parameters parameterizing the impact of **systematic uncertainties** (θ) with their central values θ^0 , signal strength μ_{sig} and predictions b for various **background** sources:

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

- Building block of likelihood mirrored via Python classes: **channels** (control/validation/signal region), **samples** (background/signal/data), **systematics** (uncertainties)

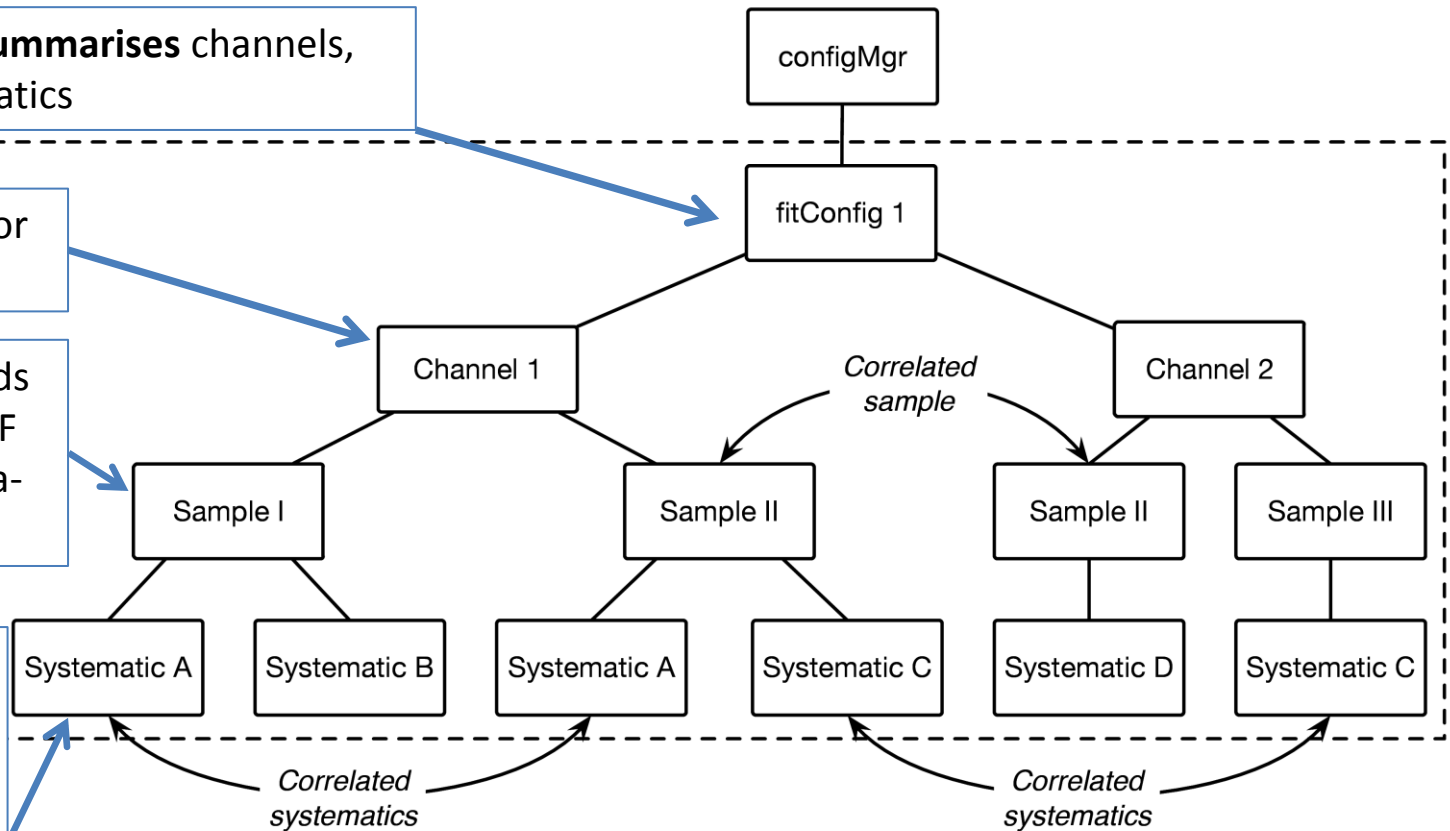
Configuration of a model

A fit configuration **summarises** channels, samples and systematics

Channels unbinned or multi-bin

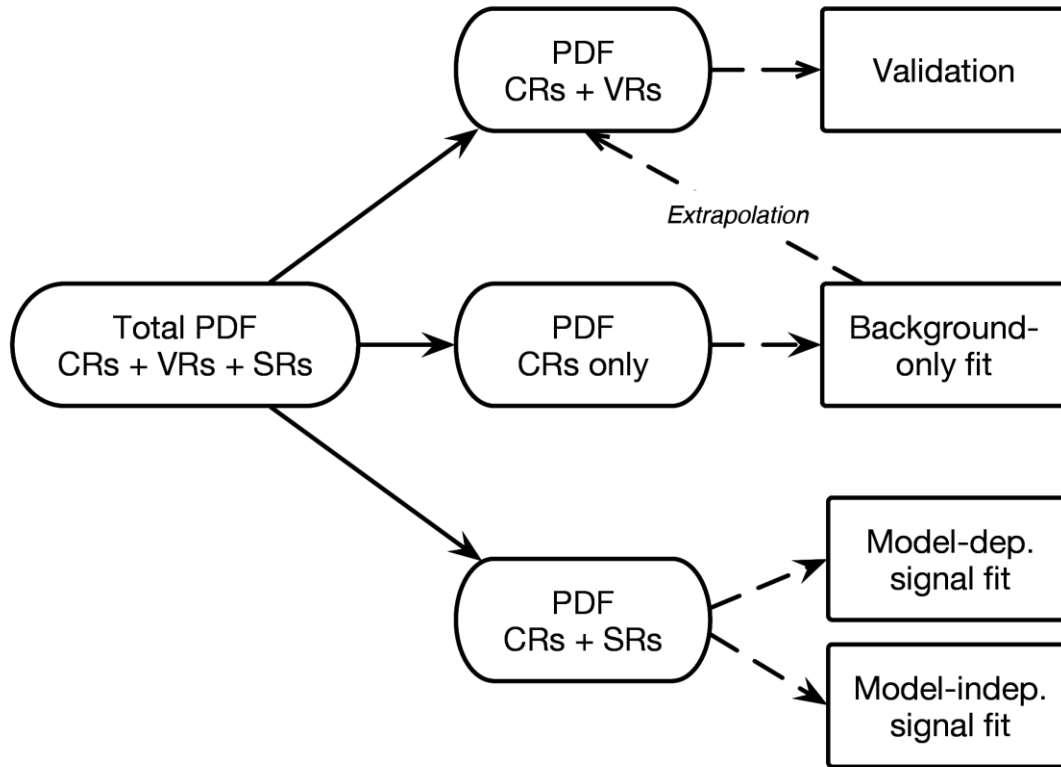
A sample corresponds to component of PDF decorated with meta-data

Systematics typically provided as 1σ variations of nominal histograms – different types available in HistFitter



- Data/MC input as ROOT TTree or TH1 or float number at sample or systematics level.
- Samples and systematics can be correlated between channels.

Performing fits



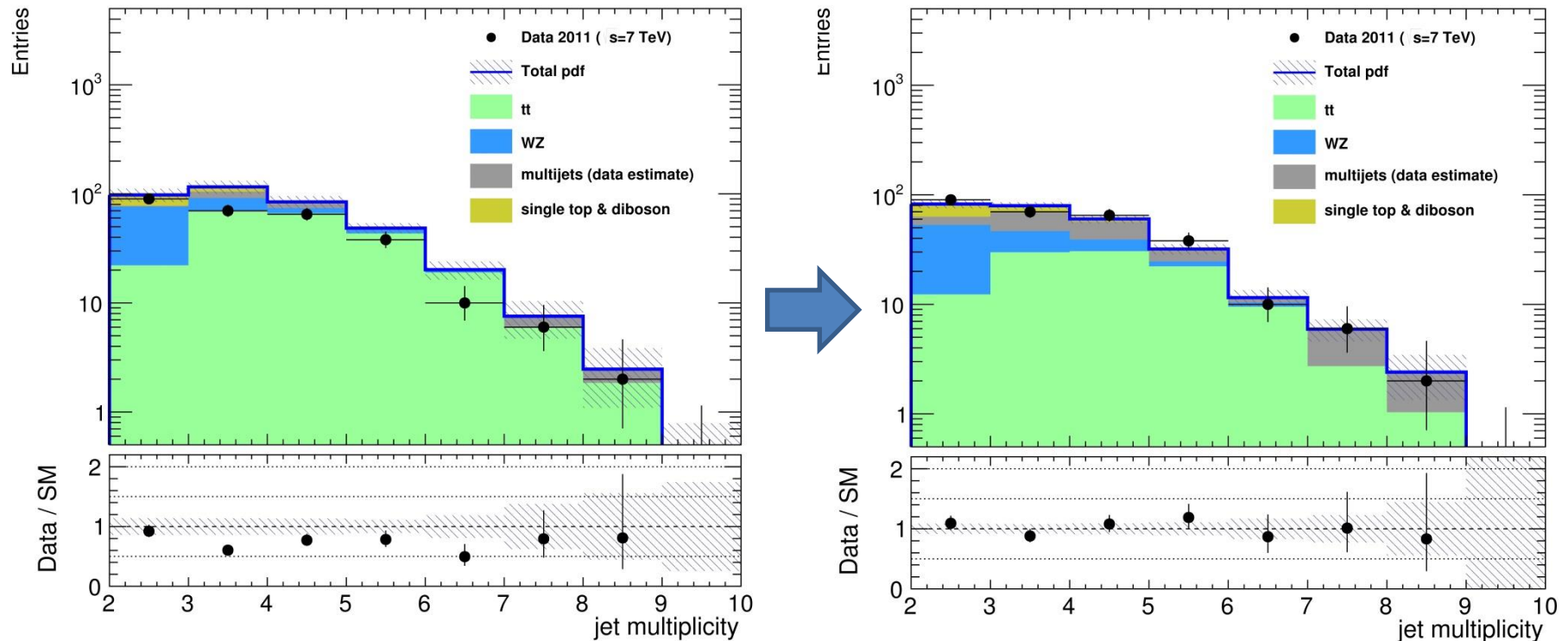
Different fit strategies:

- **Background-only fit:**
Estimate background yields in validation/signal regions
- **Model-dependent signal fit:**
Setting limits on a specific signal model or measure properties of an excess – simultaneous use of multiple signal regions possible (shape fit)
- **Model-independent signal fit:**
model-independent upper limits on # (new physics) events in a certain signal region

| Fit setup | <u>Background-only fit</u> | <u>Model-dependent signal fit</u> | <u>Model-independent signal fit</u> |
|---------------------|----------------------------|-----------------------------------|-------------------------------------|
| Samples used | backgrounds | backgrounds + signal | backgrounds + dummy signal |
| Fit regions | CR(s) | CR(s) + SR(s) | CR(s) + SR |

Presenting results: before/after fit

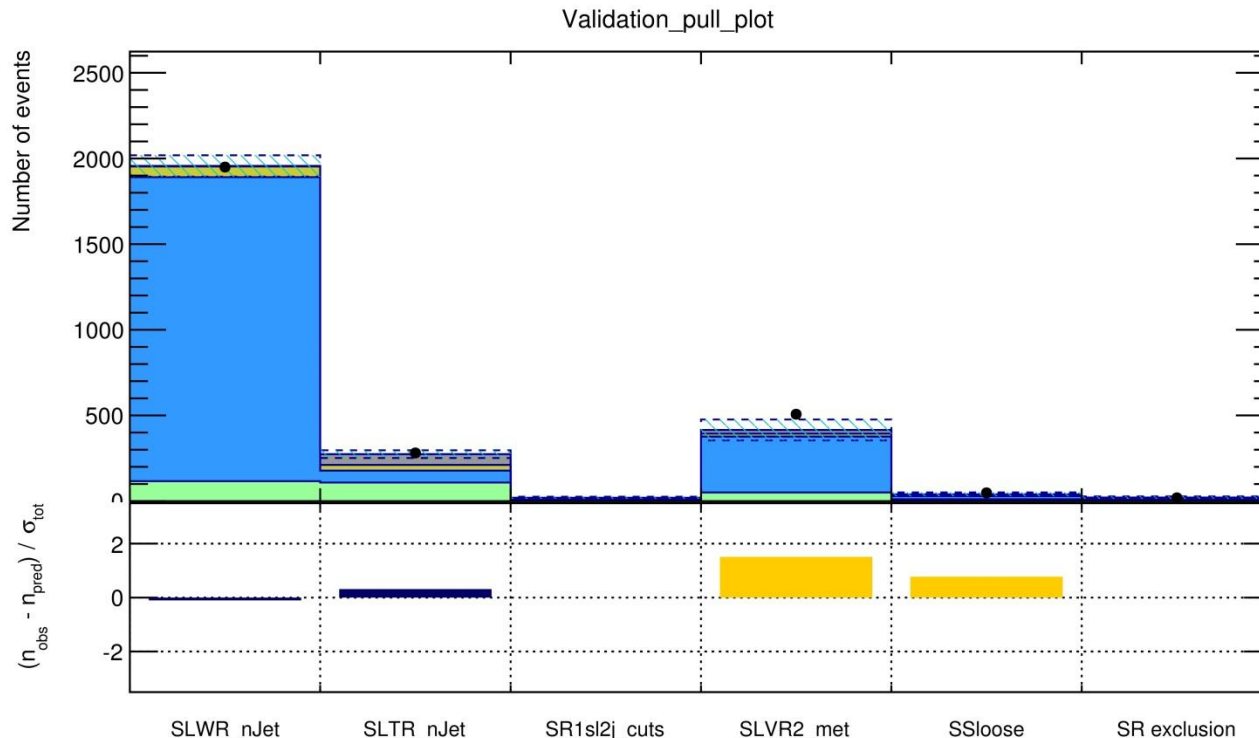
HistFitter includes a collection of tools and functions to aid the presentation of the results:
1. Visualization of fit results in before and after-fit distributions



Example plots: **before** and **after** fit
(Note: example plots without physical meaning)

Presenting results: Validation

HistFitter includes a collection of tools and functions to aid the presentation of the results:
2. Visualization of fit results in pull plots



(Note: example plot without physical meaning)

$$\chi = \frac{n_{\text{obs}} - n_{\text{pred}}}{\sigma_{\text{tot}}}$$
$$\sigma_{\text{tot}} = \sqrt{\sigma_{\text{pred}}^2 + \sigma_{\text{stat, exp}}^2}$$

Pull plot showing data – background expectation in various signal and validation regions.

Presenting results: tables

HistFitter includes a collection of tools and functions to aid the presentation of the results:
 3. Visualization of fit results in yields tables and systematics tables

| Signal Region | SR1 | SR2 |
|---------------------------------|------------------|------------------|
| Observed events | 16 | 19 |
| Fitted bkg events | 19.54 ± 3.93 | 20.47 ± 5.14 |
| Fitted Top events | 4.02 ± 0.96 | 4.32 ± 1.04 |
| Fitted V+jets events | 9.89 ± 1.86 | 10.47 ± 1.91 |
| Fitted other background events | 1.14 ± 0.15 | 1.19 ± 0.16 |
| Fitted QCD events | 4.49 ± 2.72 | 4.49 ± 4.24 |
| MC exp. SM events | 24.85 | 26.32 |
| MC exp. Top events | 8.42 | 9.11 |
| MC exp. V+jets events | 10.82 | 11.55 |
| MC exp. other background events | 1.13 | 1.17 |
| Data-driven exp. QCD events | 4.49 | 4.49 |

data

(Note: example tables without physical meaning)

after fit

| Uncertainty of channel | SR1 | SR2 |
|---|---------------------|---------------------|
| Total background expectation | 19.54 | 20.47 |
| Total statistical ($\sqrt{N_{\text{exp}}}$) | ± 4.42 | ± 4.52 |
| Total background systematic | ± 3.93 [20.14%] | ± 5.14 [25.09%] |
| QCD background | ± 2.66 | ± 4.20 |
| Statistical uncertainties | ± 2.54 | ± 1.86 |
| Jet Energy Scale | ± 1.15 | ± 1.17 |
| Top yield | ± 0.82 | ± 0.88 |
| Renormalization scale (Top) | ± 0.34 | ± 0.39 |
| V+jets yields | ± 0.28 | ± 0.29 |
| Renormalization scale (V+jets) | ± 0.14 | ± 0.03 |

before fit

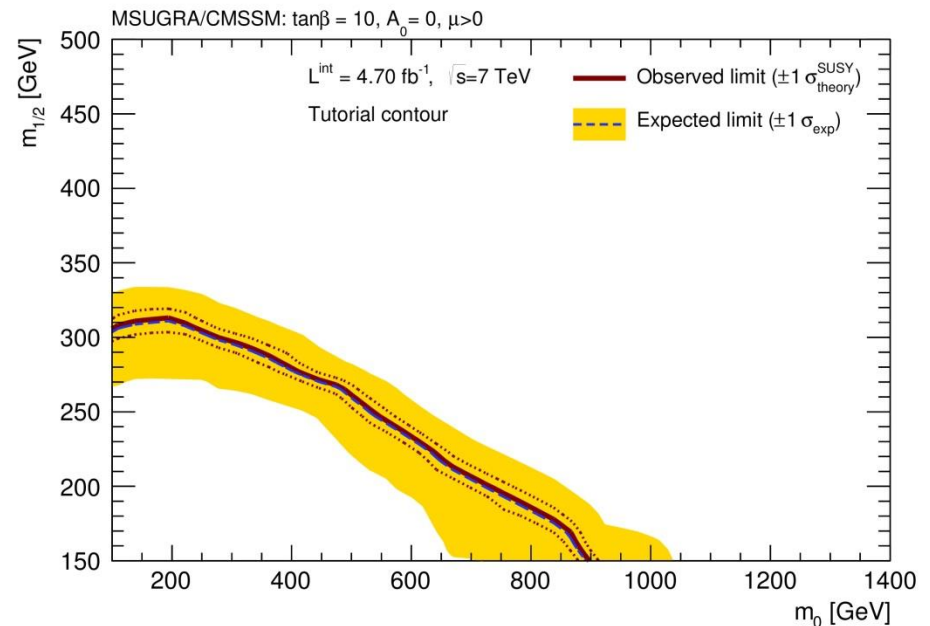
break-down of systematic uncertainties

Interpretation: Testing specific signal models

HistFitter provides various **interpretations/hypothesis tests** through calls to appropriate **Roostats** functions and classes and offers **macros** for presenting the results.

Based on a specific signal model, using the **model-dependent** signal fit:

- **Signal model hypothesis test:**
Testing signal strength of 1 for usually multiple models; HistFitter provides macros for execution and plotting.
- **Signal strength upper limit:**
Testing different signal strengths for a specific model to determine upper limit on cross section.



(Note: example plot without physical meaning)

Model-independent interpretation

Model-independent tests to quantify the agreement data – background-only hypothesis (usually Standard Model)

→ *one-bin counting experiments using the model independent signal fit*

- **Background-only p-value:**

To test the compatibility of the data with the null hypothesis.

- **Model-independent upper limits:**

To set 95 % CL upper limits on the number of events for any kind of new physics and on visible cross-section.

Scripts provided to present results in tables.

| Signal channel | $\langle \epsilon \sigma \rangle_{\text{obs}}^{95} [\text{fb}]$ | S_{obs}^{95} | S_{exp}^{95} | $p(s = 0)$ |
|-----------------------|---|-----------------------|-----------------------|------------|
| Example signal region | 0.72 | 3.4 | $8.9_{-2.7}^{+4.0}$ | 0.50 |

(Note: example table without physical meaning)

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.

Key features:

- **Easy configuration** by using a single user-defined configuration file for an entire analysis.
- **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.

Available on <http://histfitter.web.cern.ch> under a 2-clause BSD license.

Tutorials and examples equally available.

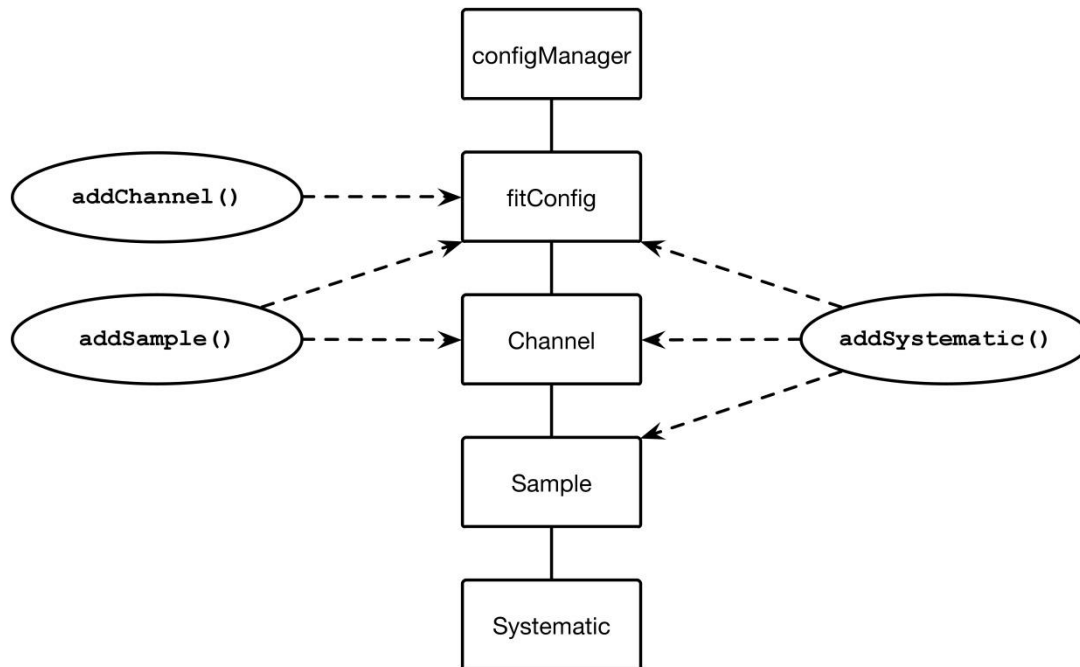
References

- Paper: <http://inspirehep.net/record/1320562>
- Proceeding for ACAT 2014: <http://iopscience.iop.org/1742-6596/608/1/012049/>
- CHEP 2015:
http://histfitter.web.cern.ch/histfitter/Files/HistFitter_CHEP2015_150316.pdf

- Contact: jeanette.miriam.lorenz@cern.ch

BACKUP

Trickle-down mechanism



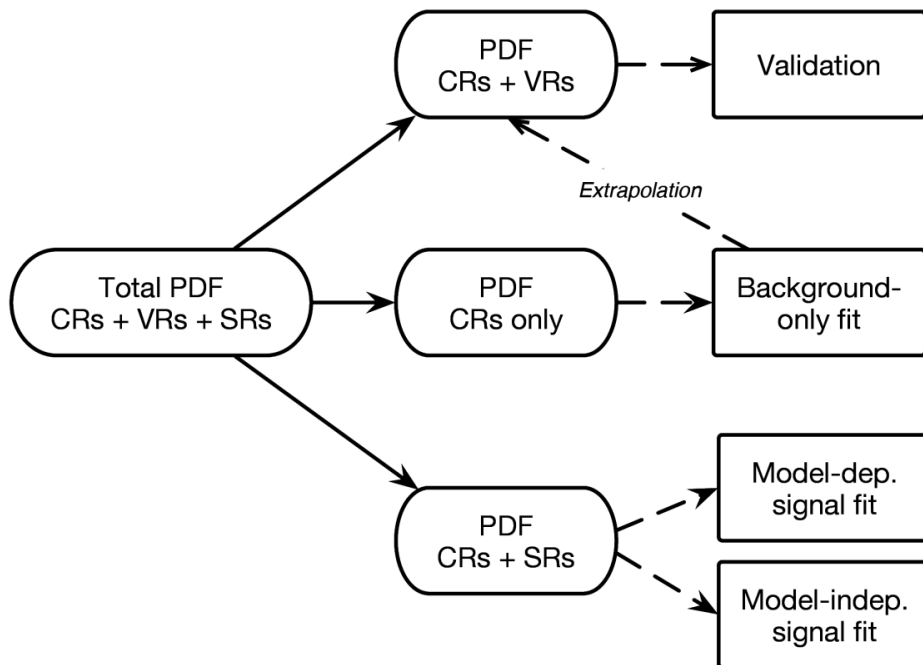
Channels are added to a **fitConfig**.

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.

⇒ A complicated PDF can be described by few lines of code.

Extrapolation



Extrapolation into validation and signal regions:

- Deconstruction of full likelihood containing CRs and VRs/SRs into smaller likelihood only containing CRs for use in the background-only fit.
- Incorporation of fitted parameters after background-only fit into full likelihood.
- 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.
- Using instead the RooExpandedFitResult class allows to extrapolate all parameters, such that a correct evaluation of the uncertainties through error propagation is possible.

Types of systematic uncertainties

Various types available in HistFitter:

Basic systematic methods in HistFactory

| | |
|-------------------------|---|
| <code>overallSys</code> | Uncertainty of the global normalization, not affecting the shape |
| <code>histoSys</code> | Correlated uncertainty of shape and normalization |
| <code>shapeSys</code> | Uncertainty of statistical nature applied to a sum of samples, bin by bin |

Additional systematic methods in HistFitter

| | |
|-------------------------------------|---|
| <code>overallNormSys</code> | <code>overallSys</code> constrained to conserve total event count in a list of region(s) |
| <code>normHistoSys</code> | <code>histoSys</code> constrained to conserve total event count in a list of region(s) |
| <code>normHistoSysOneSide</code> | One-sided <code>normHistoSys</code> uncertainty built from tree-based or weight-based inputs |
| <code>normHistoSysOneSideSym</code> | Symmetrized <code>normHistoSysOneSide</code> |
| <code>overallHistoSys</code> | Factorized normalization shape and uncertainty, described with <code>overallSys</code> and <code>histoSys</code> respectively |
| <code>overallNormHistoSys</code> | <code>overallHistoSys</code> in which the shape uncertainty is modeled with a <code>normHistoSys</code> and the global normalization uncertainty is modeled with an <code>overallSys</code> |
| <code>shapeStat</code> | <code>shapeSys</code> 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.