HistFitter

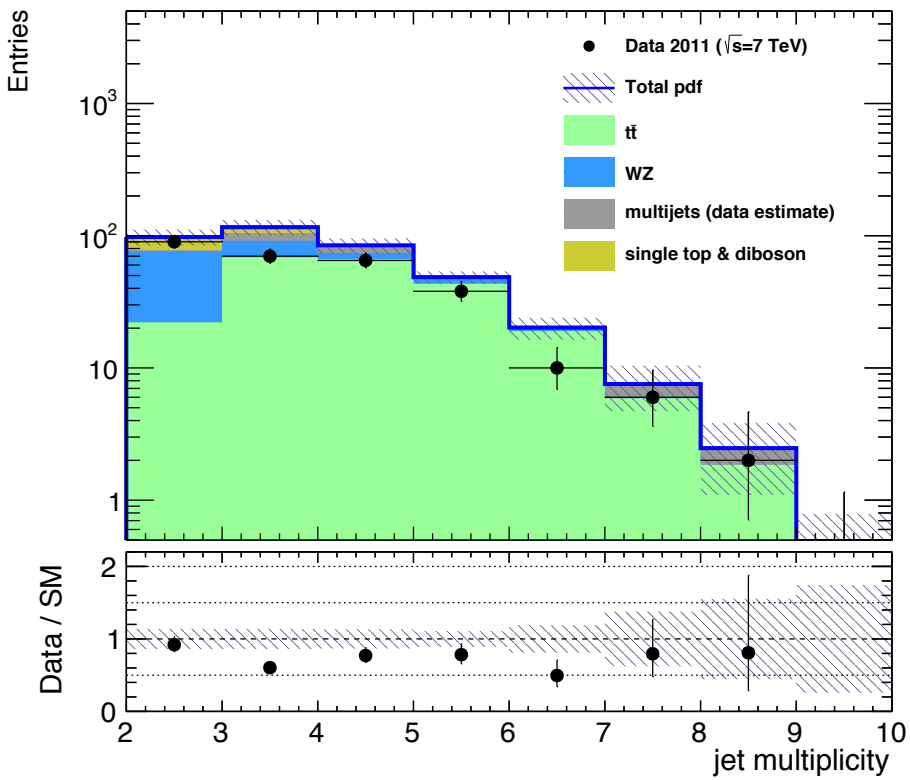
A flexible framework for statistical data analysis

Geert-Jan Besjes (*Niels Bohr Institute, Univ. of Copenhagen*),

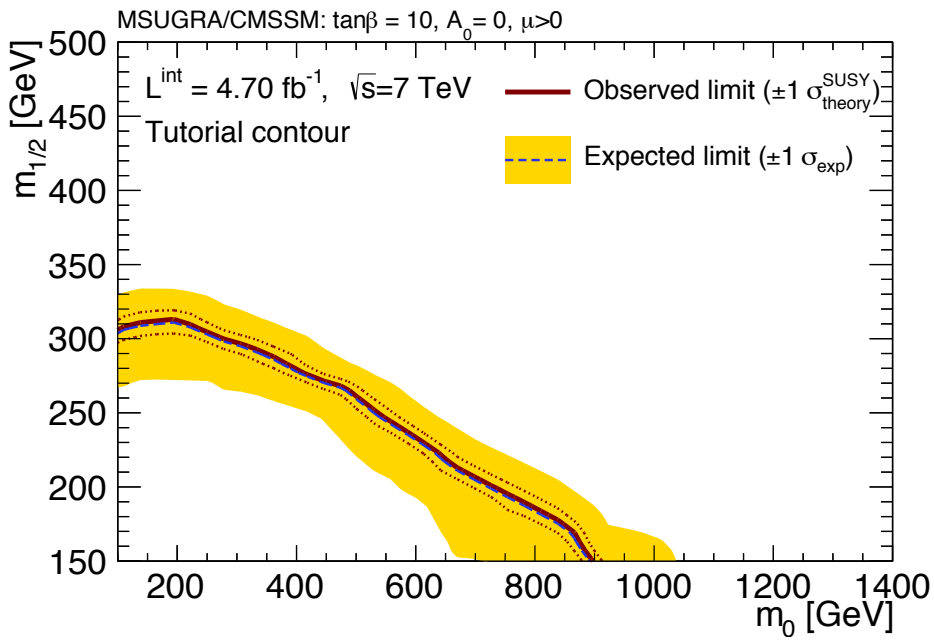
Max Baak (*CERN*), David Côté (*Univ. of Texas*), Alex Koutsman (*TRIUMF*), Jeanette Lorenz (*LMU München*), Dan Short (*Univ. of Oxford*)

CHEP 2015, Okinawa
16 April 2015

Motivation



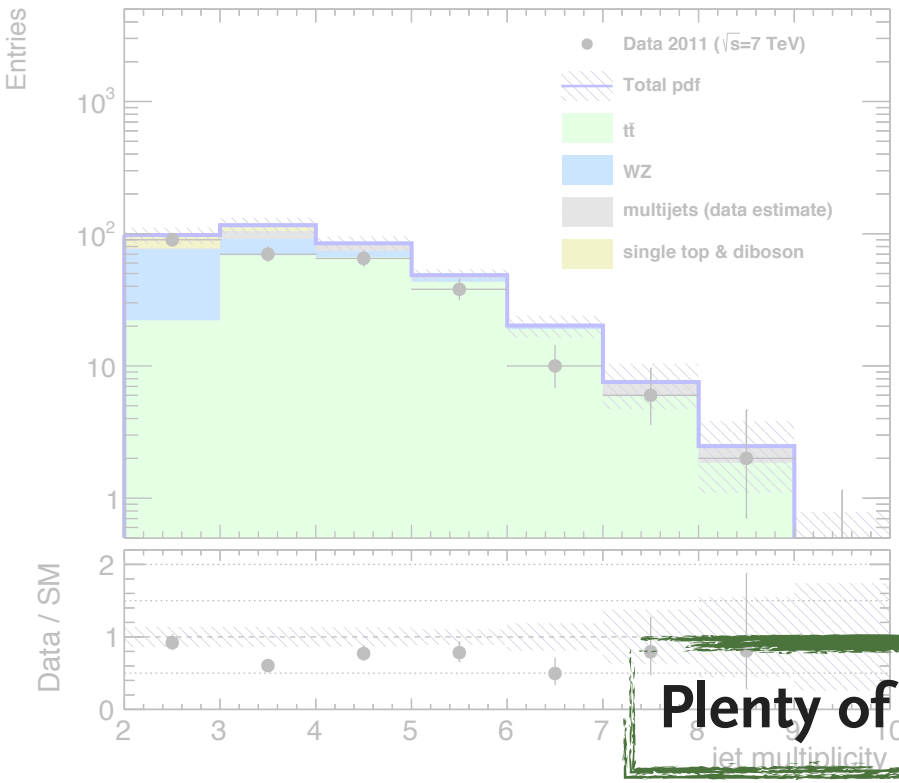
Calibrated measurements

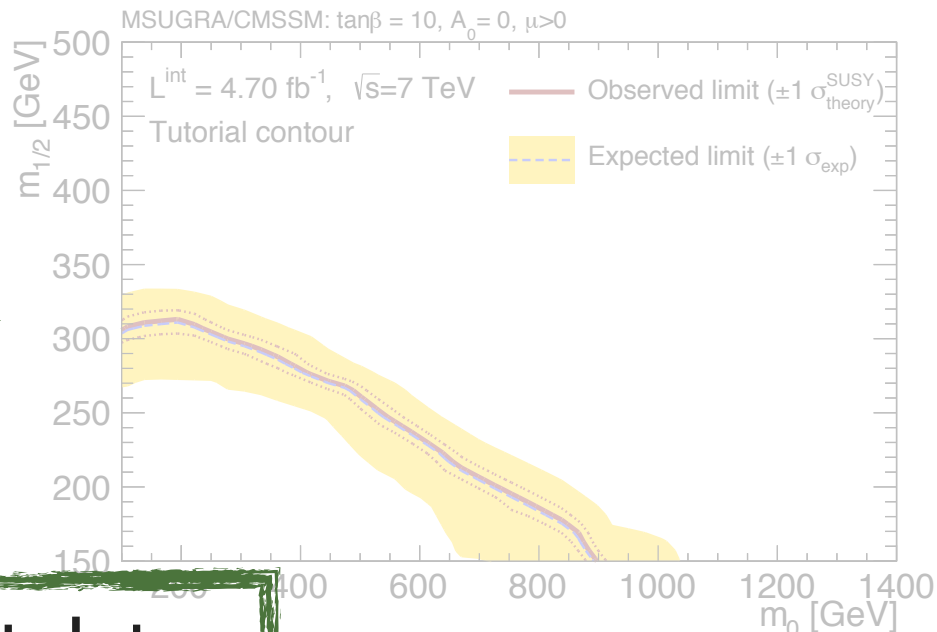


Results / interpretation

Note: dummy figures!

Motivation





Plenty of complicated steps

Calibrated measurements

Results / interpretation

Note: dummy figures!

Overview

Software framework for statistical data analysis

- Built on top of the HistFactory/RooFit (construction of models) and RooStats (statistical tests) software packages
- Divided in Python scripts and C++ core for intensive calculations

Extends RooFit/RooStats in four areas:

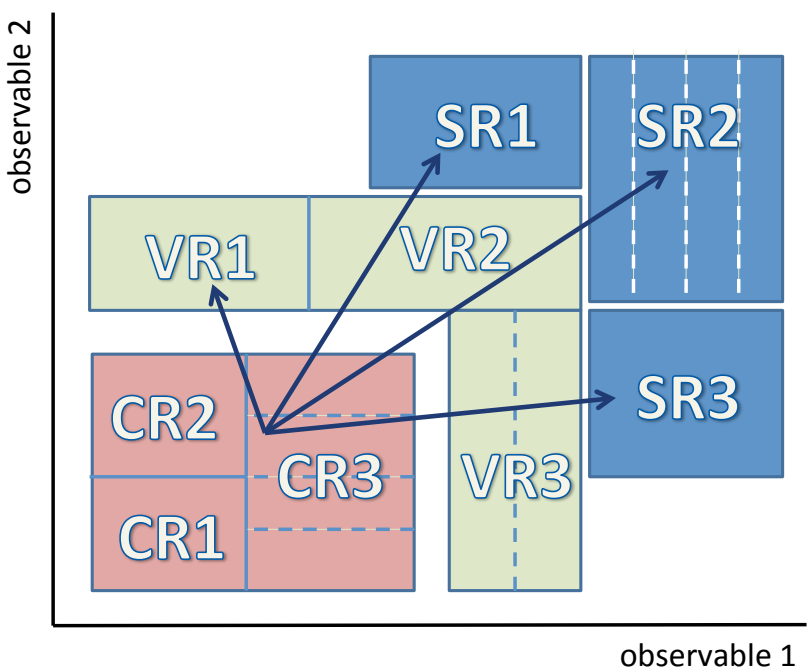
1. ***Easy configuration***: complete statistical analyses, using a single configuration file
2. ***Analysis strategy***: concepts of analysis control, validation and signal regions deeply woven into the design
3. ***Bookkeeping***: automatic management of multiple configurations, statistical tests, underlying data, etc.
4. ***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.)

Used in (almost) all searches for supersymmetry in the ATLAS collaboration

Analysis strategy

- Particle physics: analyse large datasets to **measure** properties of known particles and **discover** evidence of new particles
- Dataset must be compared to (externally provided) predictions for **background** and **signal**

- **HistFitter configures and builds models that describe the data and provides (access to) tools for both presentation and statistical interpretation**



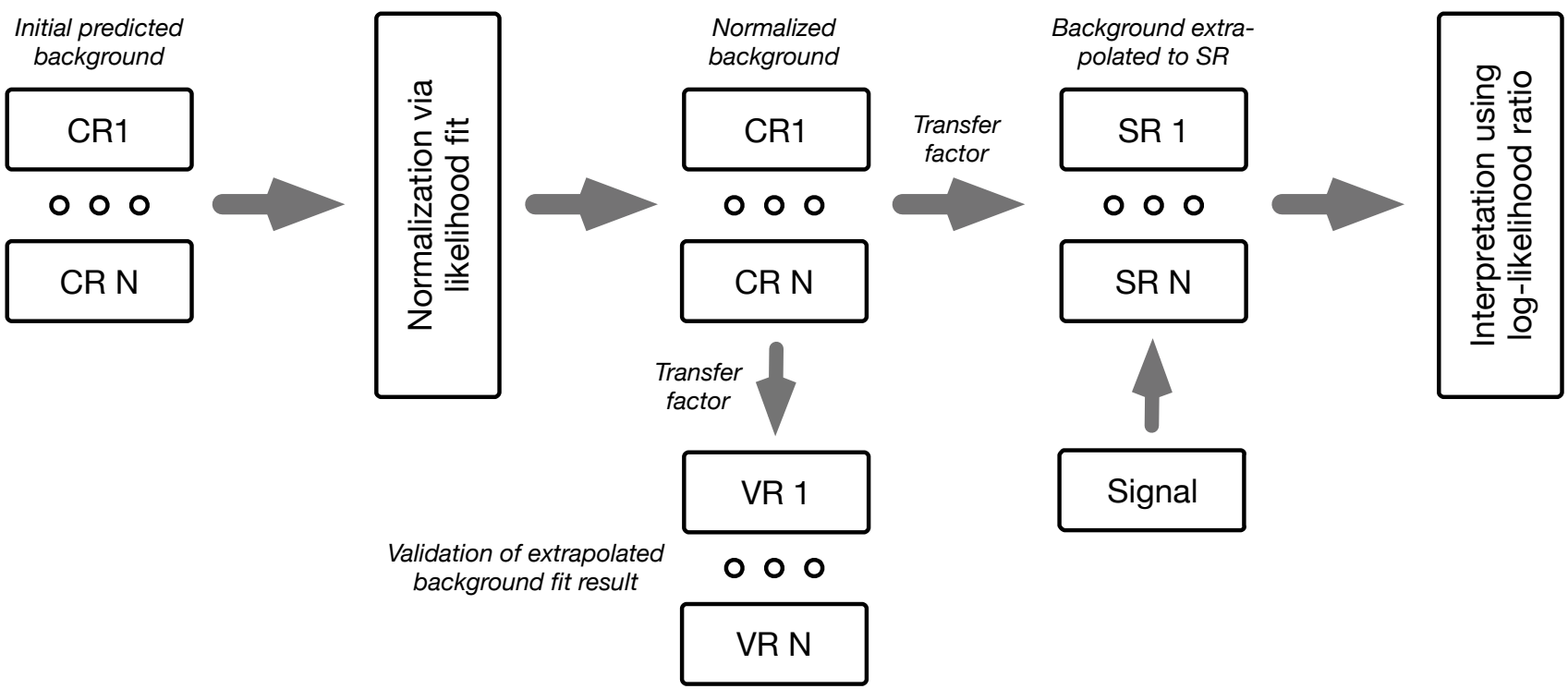
Models are based on signal, control and validation regions:

- **signal region**: events of interest (SR)
- **control region**: background-rich region; constrains background predictions (CR)
- **validation region**: region without expected signal used to validate background predictions (VR)

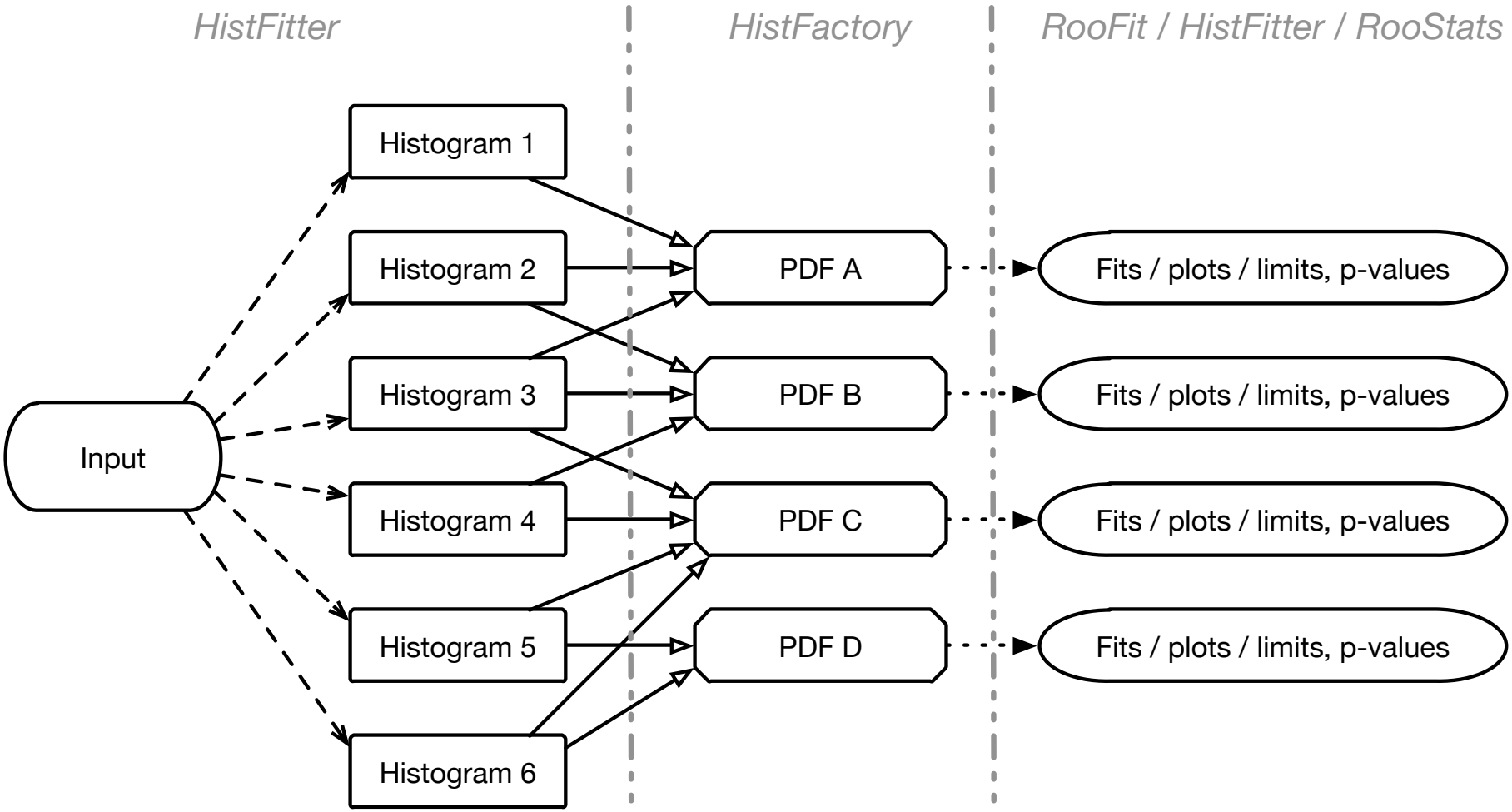
Analysis strategy (cont.)

Regions described using **probability density functions** (PDF)

- these include **normalisations and nuisance parameters** (systematic errors);
- parameters adjusted by **likelihood minimisations**;
- every region modelled by its own PDF; can be combined in **simultaneous fit**;



Processing sequence



1. Histogram creation, based on input data and configuration
2. PDF creation
3. Interpretation and presentation

Analysis configuration

- Managing all fits and all regions can be complex: a search for new particles can involve ~10 signal regions, ~5 control regions and ~1000 models to test
- Hard to manage this all correctly and set up non-trivial tools!

- HistFitter simplifies this using **one** Python configuration script: implements a **configuration manager** (two related singletons in Python and C++) that have **fit configurations**
- Automatic re-use of shared data to save time and memory

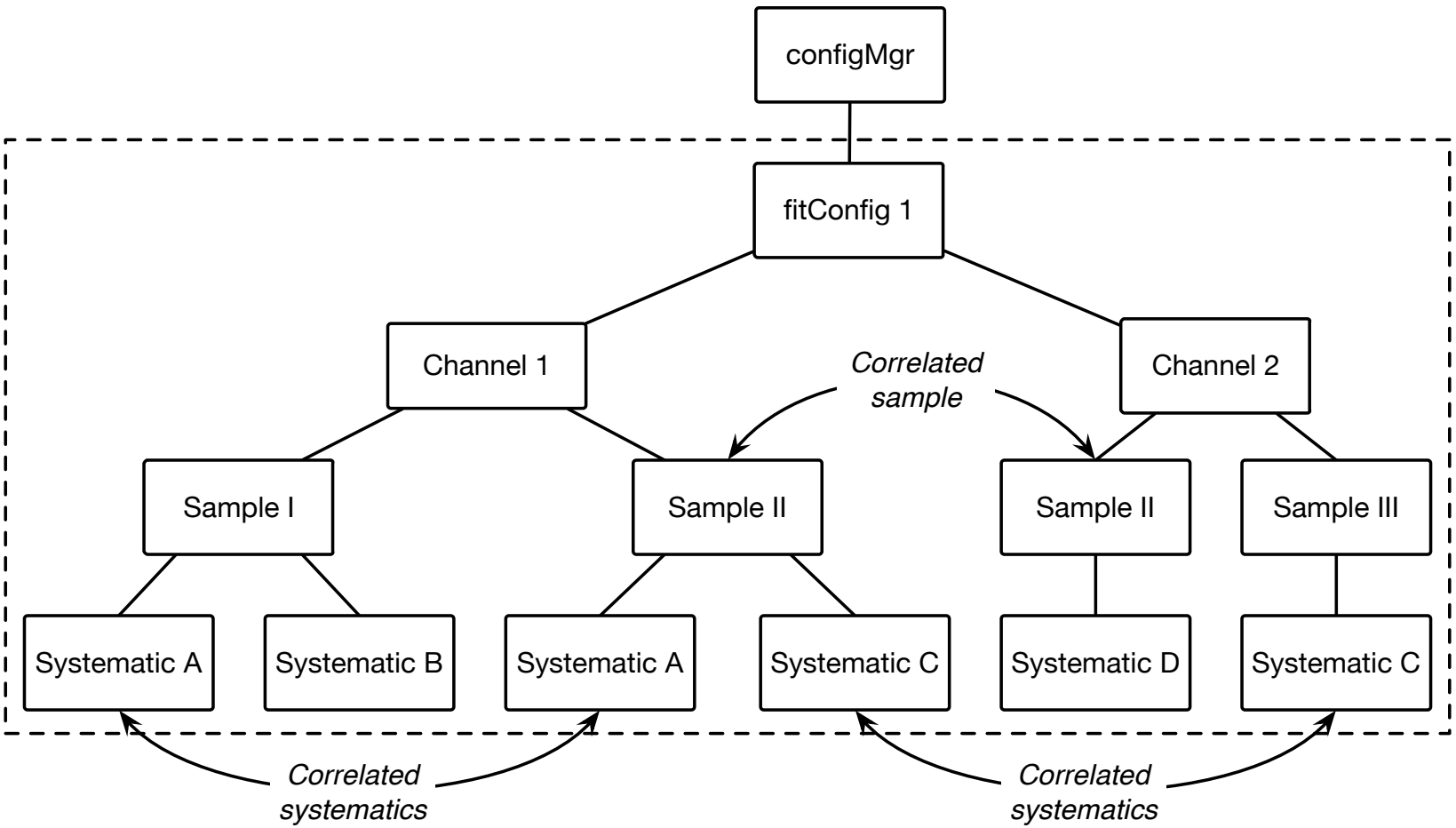
- **Likelihood** described by Poisson distributions and additional contraints:
  $L(\mathbf{n}, \theta^0 | \mu_{\text{sig}}, \mathbf{b}, \theta) = P_{\text{SR}} \times P_{\text{CR}} \times C_{\text{syst}}$
- Depends on **observed events** n, **expected background** b and parameters θ for **systematic uncertainties** (constrained by C)

- Building blocks of the likelihood function correspond to Python classes:
  - regions (called **channels**), **samples** and **systematics**
  - common building blocks can be shared

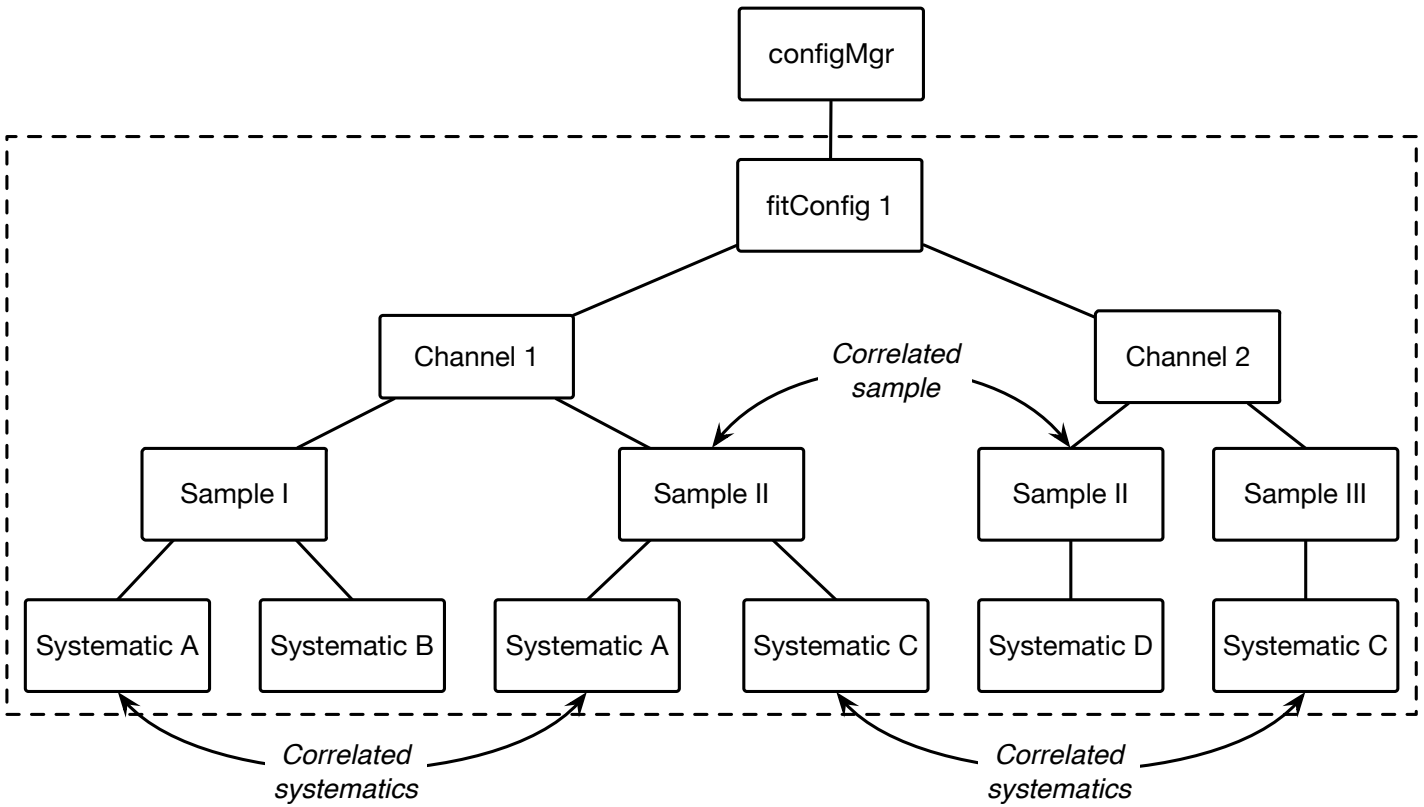
Analysis configuration

Regions described using **probability density functions** (PDF)

- these include **normalisations and nuisance parameters** (systematic errors);
- parameters adjusted by **likelihood minimisations**;
- every region modelled by its own PDF; can be combined in **simultaneous fit**;



Analysis configuration



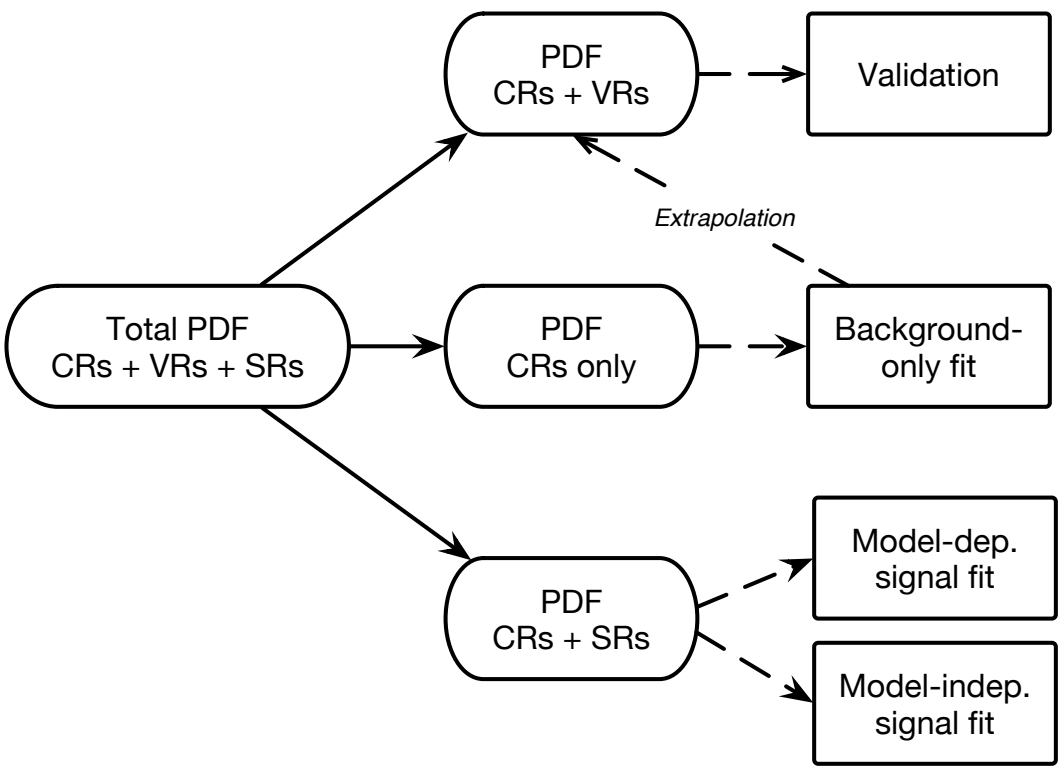
Samples

- Component of PDF with meta-data
- Input: ROOT TTree, ROOT TH1 histograms or floating point numbers
- Can be correlated between multiple channels

Systematics

- Fluctuations w.r.t. nominal histograms (typically 1 sigma)
- Input: ROOT TTree, ROOT TH1 histograms or floating point numbers
- Can be correlated between different channels and/or samples
- Several HistFactory types available (plus extended and composite types)

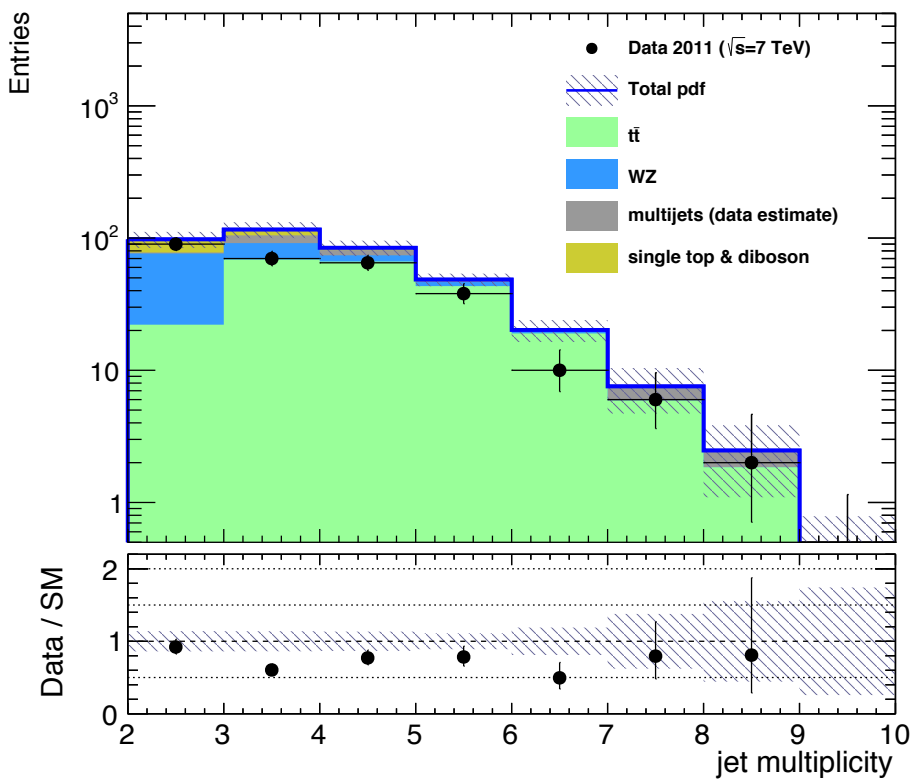
Performing fits



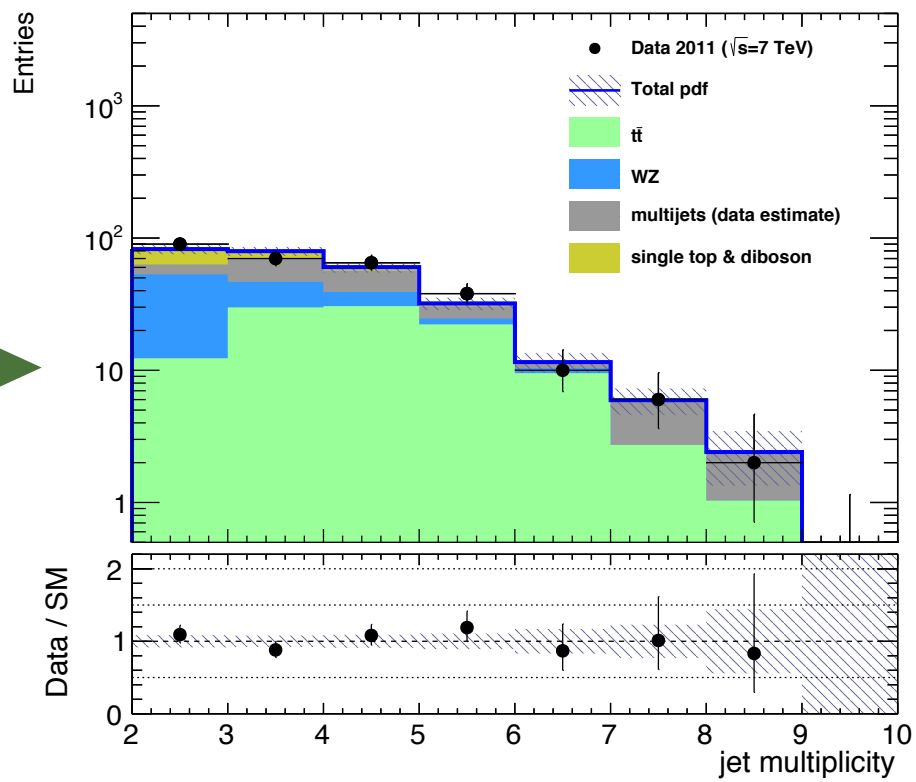
- **Background-only**: estimate background contributions in SR(s) and/or VR(s)
  Performed using backgrounds in CR, extrapolated to other regions
- **Model-dependent:** perform a hypothesis test (e.g. reject a particular alternative model) or measure properties
  Performed using backgrounds and a signal model in CR+SR
- **Model-independen**t: calculates model-independent upper limits on #events in SR allowed
  Performed using backgrounds and a dummy signal in the CR+SR

Extrapolation relies on a special region-aware `RooExpandedFitResult` class to propagate fit result from one likelihood function to another

Presenting results: before/after fit plots



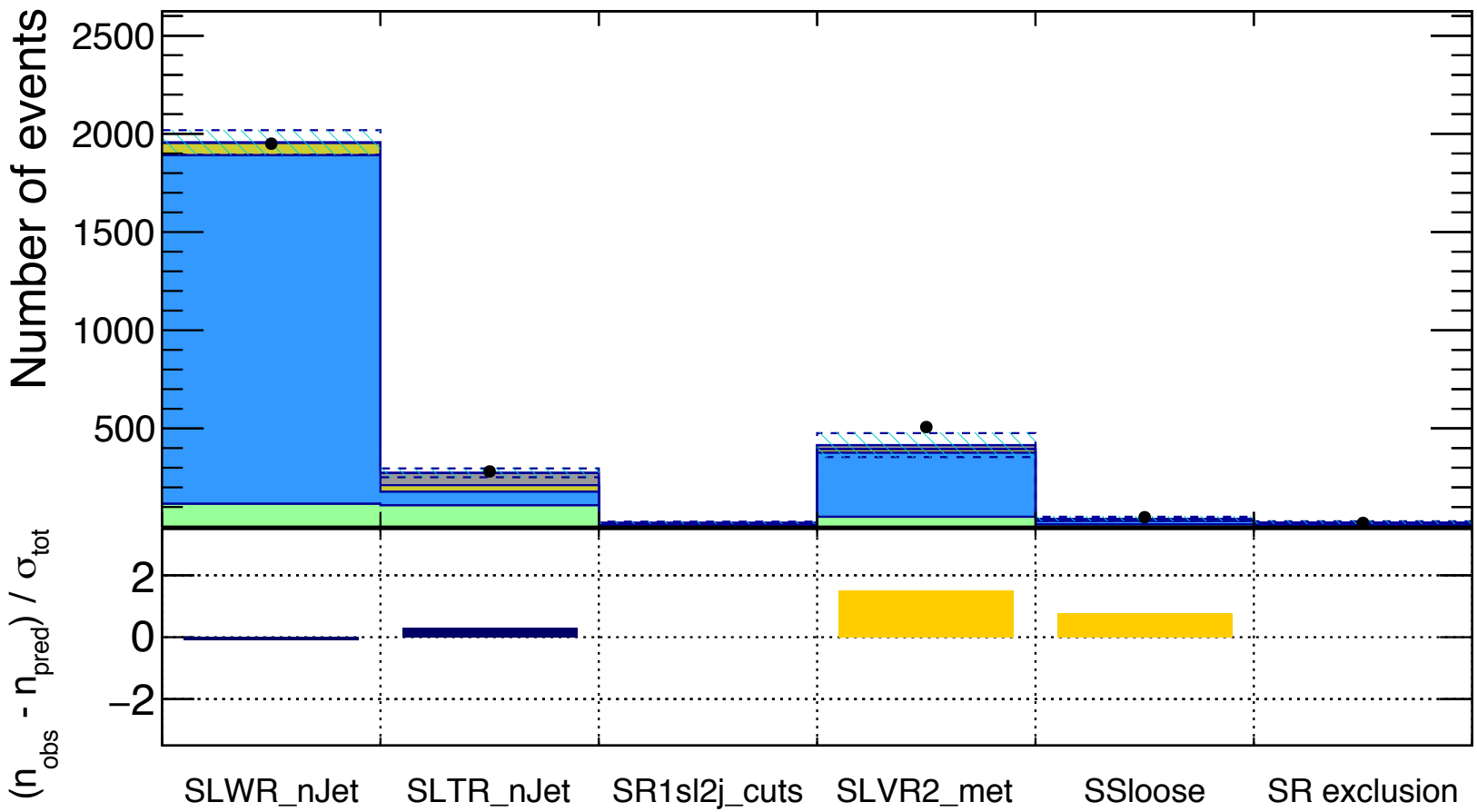
Before fit



After fit

Note: dummy figures!

Presenting results: validation plots



Pull plot in various one-bin validation regions;
pulls shown relative to total systematic and statistical error

Note: dummy figure!

Presenting results: result 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

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

Tables with event yields and breakdown of systematic uncertainties

Note: dummy tables!

Interpreting results: testing models

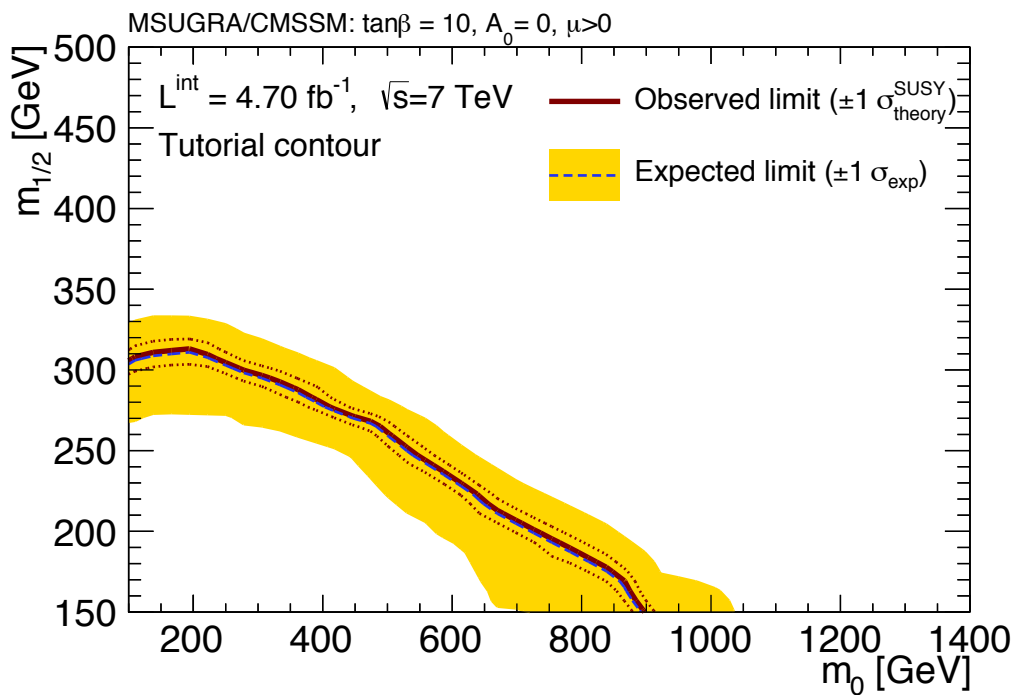
Several interpretations/hypothesis tests available (RooStats-based) and macros to interpret and present the results.

- **Hypothesis tests**: test a specific alternative (signal) model

  HistFitter provides tools to run over sets of models and graphically present the results.

- **Upper limit calculation**s: performs repeated hypothesis tests

  Determines 95% CL cross-section upper limit for specific signal models



Note: dummy figure!

Interpreting results: model-independent

Model-independent tests quantify the agreement with the background-only hypothesis (e.g. "the Standard Model is true") and are always performed as one-bin counting experiments; no shape can be assumed

- **Background-only p-value**:
  test the compatibility of the data with the null hypothesis

  HistFitter provides tools to run over sets of models and graphically present the results.

- **Model-independent upper limit**:
  how many extra events in the signal region are allowed?

  Determines 95% CL cross-section upper limit and allowed extra events N_{BSM}

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

Note: dummy table!

Summary

Presented **HistFitter, a software framework** for statistical data analysis

- easy modelling of complex configurations
- builds complex PDFs and performs statistical tests and fits using HistFactory, RooFit and RooStats

Key features:

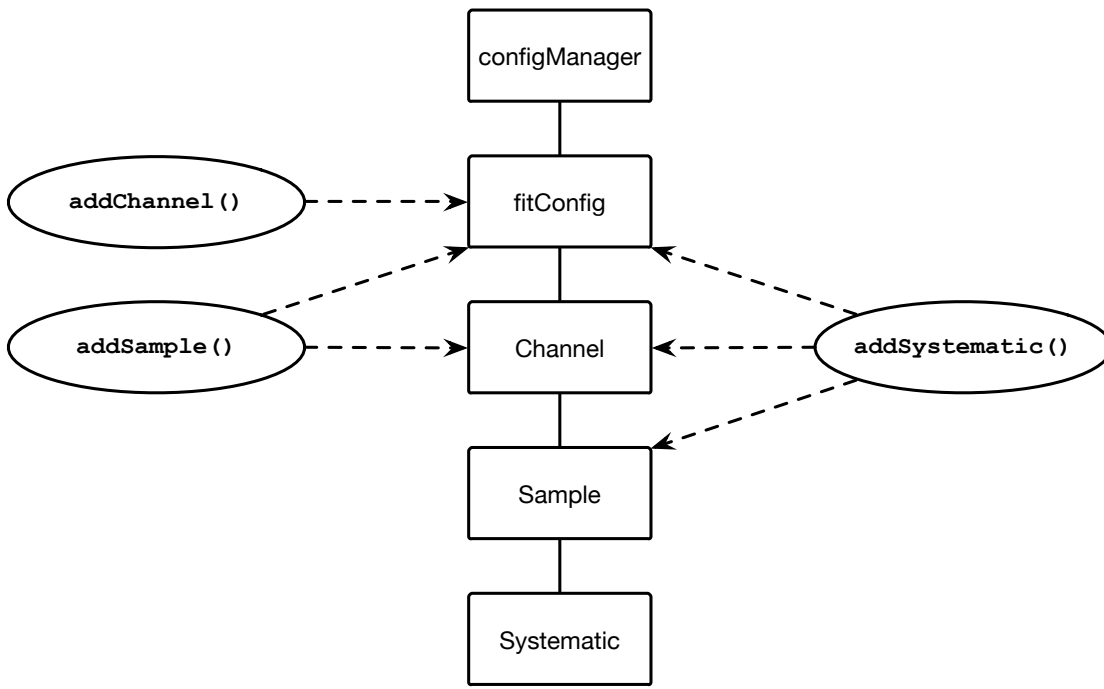
1. ***Easy configuration***: single user-defined configuration file for an entire analysis
2. ***Essential concepts of regions built in***: control, validation and signal regions deeply woven into the design and extrapolation of results is automatically taken care of
3. ***Bookkeeping***: automatic management of multiple configurations, statistical tests, underlying data, etc.
4. ***Presentation and interpretation***: large set of easy-to-use tools to present data and interpret results available

Available on http://histfitter.web.cern.ch under a 2-clause BSD license, also including links to a tutorial and examples

Backup

Easy extending of fit configurations

- `Channels` (regions) added to a `fitConfig` (describes analysis)
- `Samples` (e.g. W background) can be added to either a `fitConfig` or a `Channel`; those added to `fitConfig` also added to all dependent `Channels`
- Similar for `Systematic` (e.g. W background theoretical uncertainty; jet energy scale; etc.): added to either `fitConfig`, `Channel` or `Sample` and propagated downwards



Easy to build and extend complicated likelihood functions in a simple python file!

Systematic types

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 inputs
`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

Additional systematic types: combination of basic HistFactory methods and optional HistFitter keywords: `norm`, `OneSide` and/or `Sym`.

Systematic objects can be built with tree-based, weight-based, floating-point or histogram input methods in all cases.