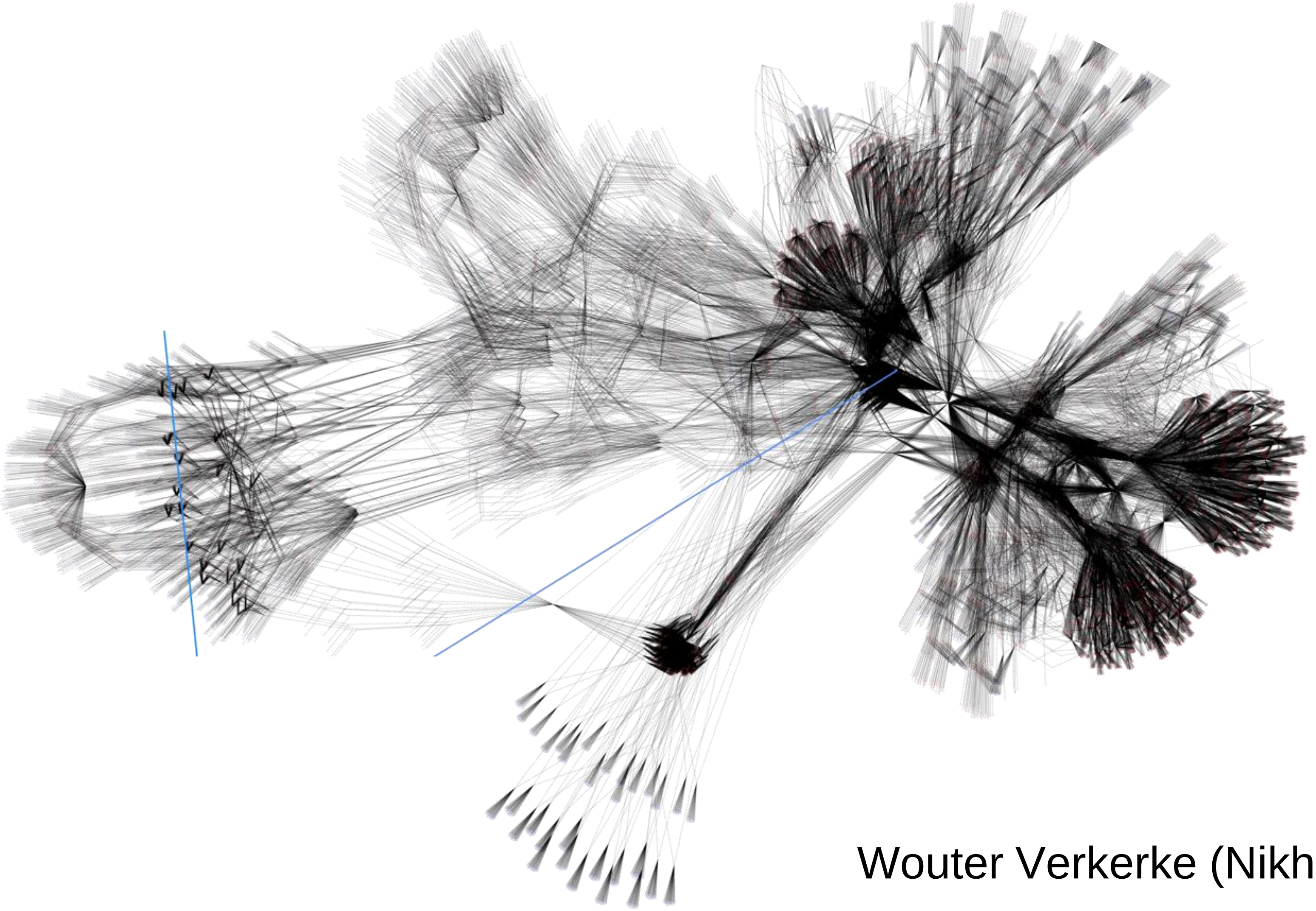
Statistical analysis tools for the Higgs discovery and beyond



Wouter Verkerke (Nikhef)

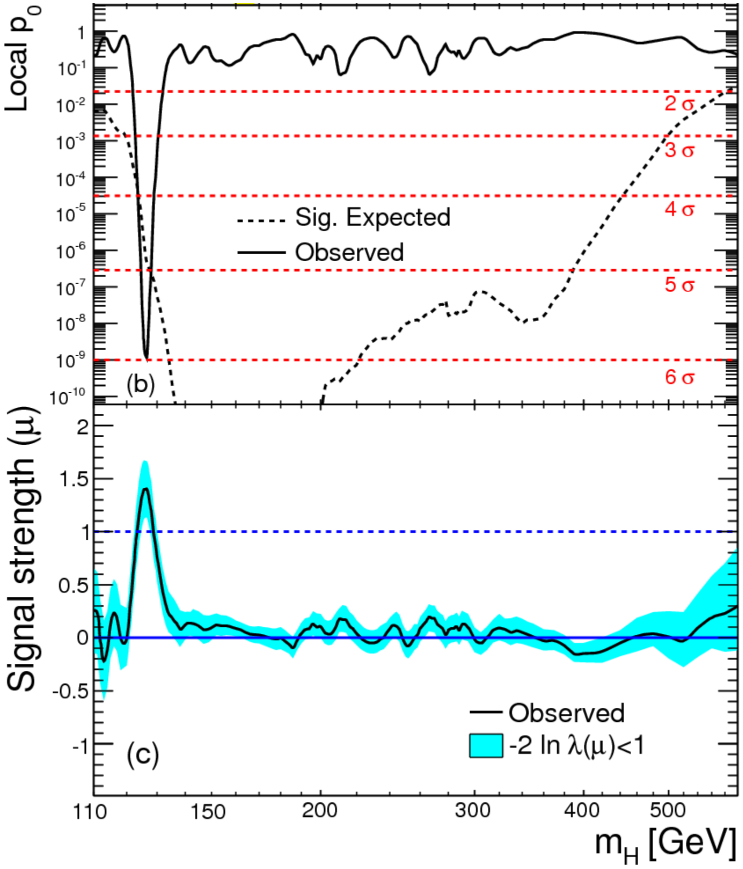
What do you want to know?

- Physics questions we have…
  - Does the (SM) Higgs boson exist?
  - What is its production cross-section?
  - What is its boson mass?

- Statistical tests construct probabilistic statements: p(theo|data), or p(data|theo)
  - Hypothesis testing (discovery)
  - (Confidence) intervals
    Measurements & uncertainties

- Result: *Decision* based on tests

“As a layman I would now say: I think we have it”



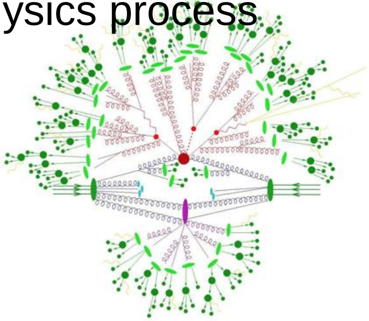
Wouter Verkerke, NIKHEF

All experimental results *start* with the formulation of a model

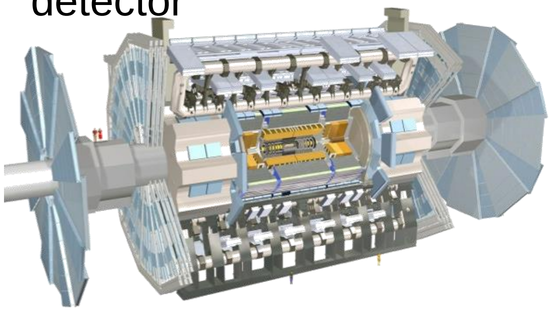
- Examples of HEP physics models being tested
  - SM with m(top)=172,173,174 GeV → Measurement top quark mass
  - SM with/without Higgs boson → Discovery of Higgs boson
  - SM with composite fermions/Higgs → Measurement of Higgs coupling properties
- Via chain of physics simulation, showering MC, detector simulation and analysis software, a physics model is reduced to a statistical model

The HEP analysis workflow illustrated

Simulation of ‘soft physics’ physics process

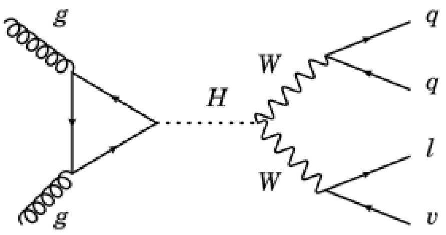


Simulation of ATLAS detector

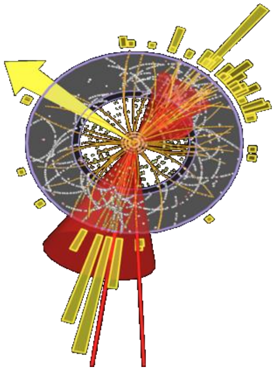


LHC data

Simulation of high-energy physics process



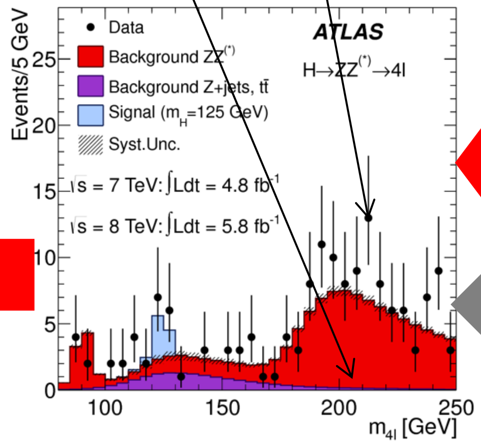
Reconstruction of ATLAS detector



Analysis Event selection

$P(m_{4l}|SM[m_H])$

Observed m_{4l}

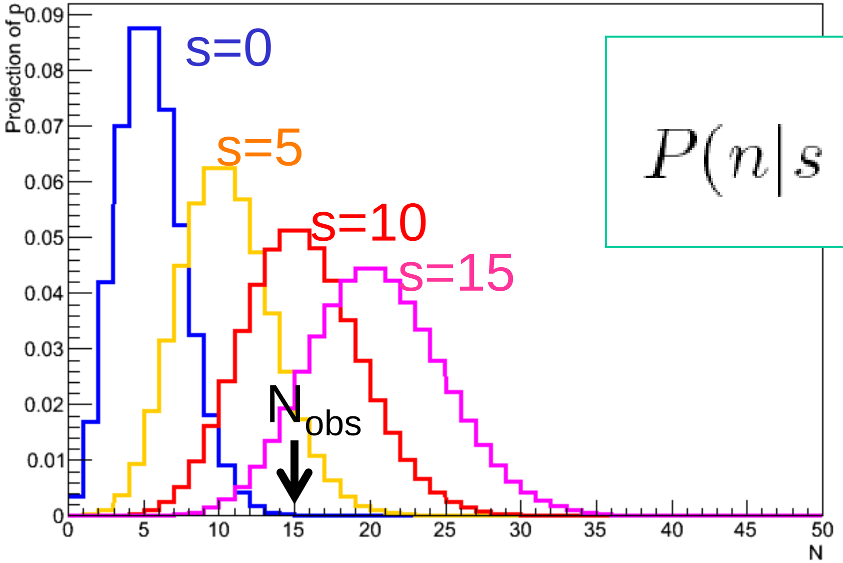


prob(data|SM)

Wouter Verkerke, NIKHEF

All experimental results start with the formulation of a model

- Examples of HEP physics models being tested
  - SM with m(top)=172,173,174 GeV → Measurement top quark mass
  - SM with/without Higgs boson → Discovery of Higgs boson
  - SM with composite fermions/Higgs → Measurement of Higgs coupling properties
- Via chain of physics simulation, showering MC, detector simulation and analysis software, a physics model is reduced to a statistical model
- A statistical model defines p(data|theory) for all observable outcomes
  - Example of a statistical model for a counting measurement with a known background



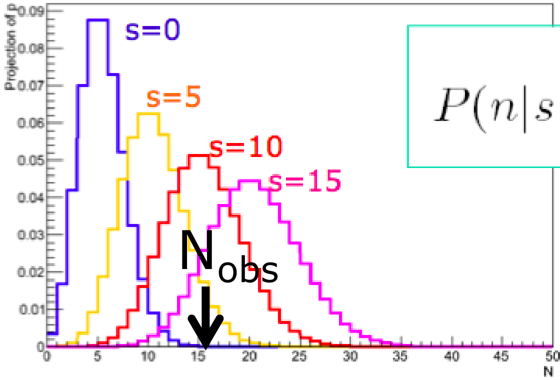
$$P(n|s+b) = \frac{(s+b)^n}{n!} e^{-(s+b)}$$

NB: b is a constant in this example

Definition: the Likelihood is P(observed data|theory)

Everything starts with the likelihood

- **All** fundamental statistical procedures are based on the likelihood function as ‘description of the measurement’



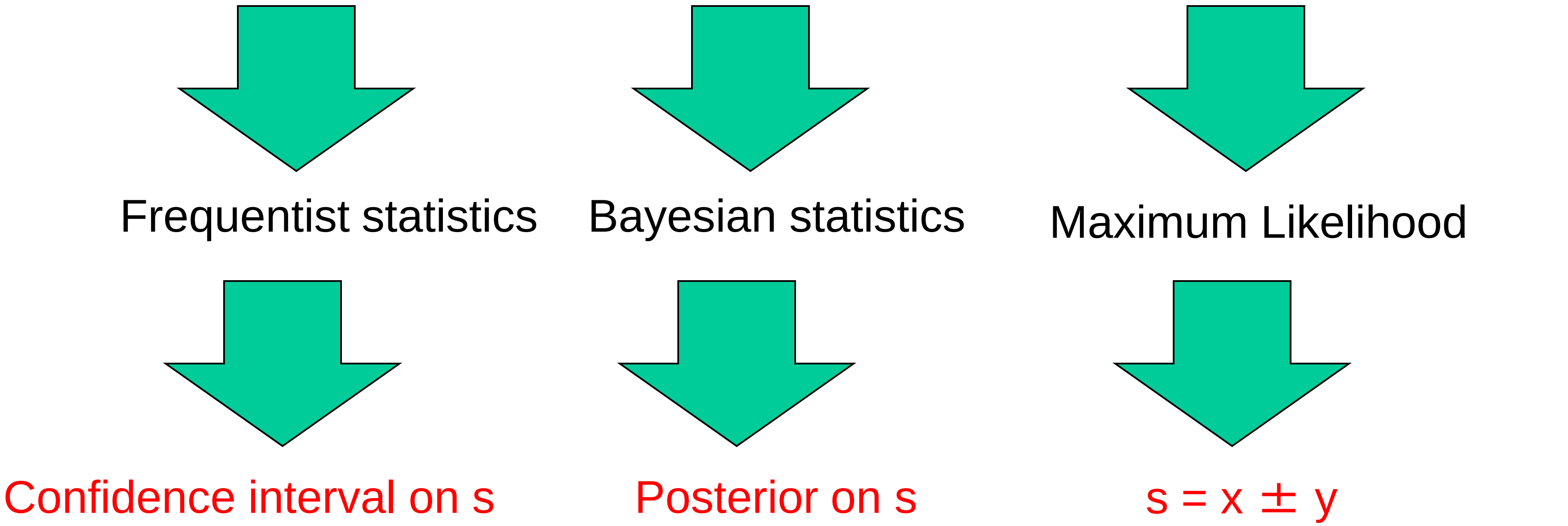
$$P(n|s+b) = \frac{(s+b)^n}{n!} e^{-(s+b)}$$

NB: b is a constant in this example

Definition: the Likelihood is P(observed data|theory)

e.g. L(15|s=0)

e.g. L(15|s=10)



Frequentist statistics

Bayesian statistics

Maximum Likelihood

Confidence interval on s

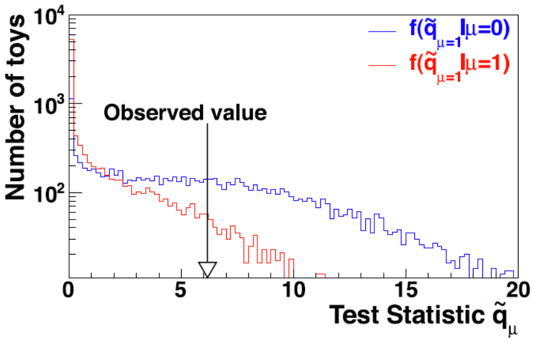
Posterior on s

s = x ± y

Everything starts with the likelihood

Frequentist statistics

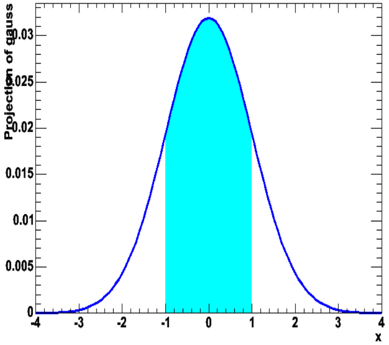
$$/_{m}(\vec{N}_{obs}) = \frac{L(\vec{N}\,|\,m)}{L(\vec{N}\,|\,\hat{m})}$$



Confidence interval
or p-value

Bayesian statistics

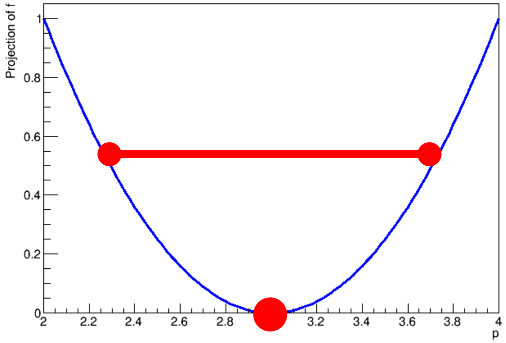
$$P(m) \mu L(x\,|\,m) \times p(m)$$



Posterior on s
or Bayes factor

Maximum Likelihood

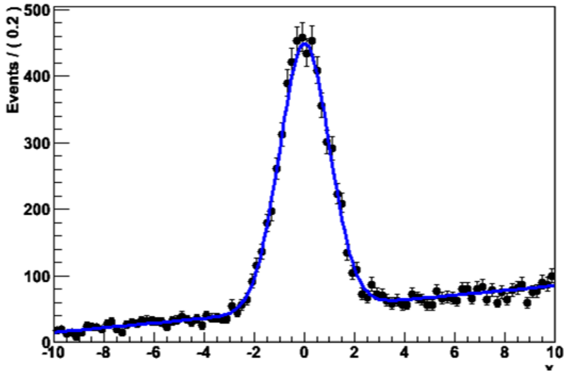
$$\left.\frac{d \ln L(\vec{p})}{d\vec{p}}\right|_{p_i=\hat{p}_i} = 0$$



s = x ± y

How is Higgs discovery different from a simple fit?

Gaussian + polynomial

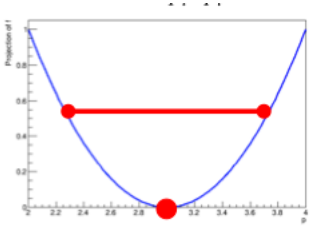


ROOT TH1

ROOT TF1

$$L(\vec{N}\,|\,\mu,\vec{\theta}) = \prod_i Poisson(N_i\,|\,f(x_i,\mu,\vec{\theta})$$

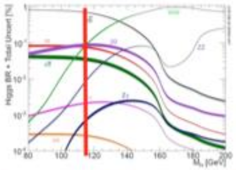
"inside ROOT"



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

μ = 5.3 ± 1.7

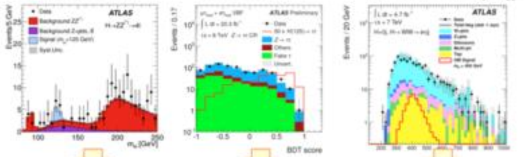
Higgs combination model



H→ZZ→llll

H→ττ

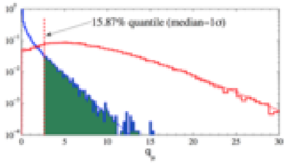
H→WW→μνjj



$L(\vec{N}_{ZZ}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...)$ $L(\vec{N}_{\tau\tau}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{\tau\tau},...)$ $L(\vec{N}_{WW}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{WW},...)$

$$L(\vec{N}_{ZZ},\vec{N}_{\tau\tau},\vec{N}_{WW}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...)\cdot\prod Poisson(N^i_{\tau\tau},...)\cdot\prod Poisson(N^i_{WW},...)\cdot...$$

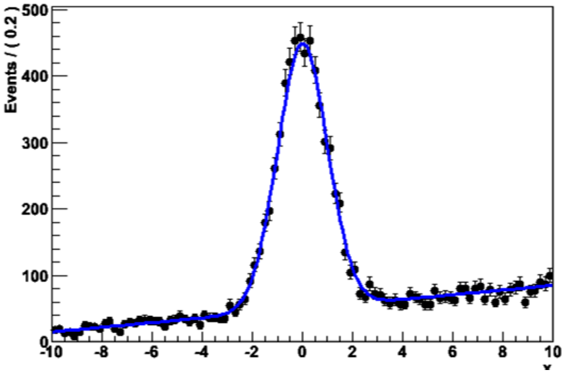
$$\lambda_\mu(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}) = \frac{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}\,|\,\mu,\hat{\hat{\vec{\theta}}})}{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}\,|\,\hat{\mu},\hat{\vec{\theta}})}$$



$$p(H_\mu) = \int_{\lambda_{obs}}^{\infty} f(\lambda\,|\,H_\mu)\,d\lambda = ...$$

How is Higgs discovery different from a simple fit?

Gaussian + polynomial

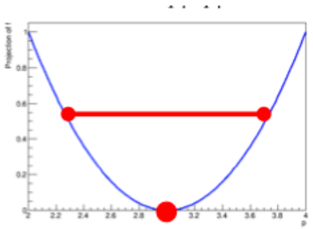


ROOT TH1

ROOT TF1

$$L(\vec{N}\,|\,\mu,\vec{\theta}) = \prod_i Poisson(N_i\,|\,f(x_i,\mu,\vec{\theta})$$

"inside ROOT"



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

μ = 5.3 ± 1.7

Likelihood Model orders of magnitude more complicated. Describes

- O(100) signal distributions

- O(100) control sample distr.

- O(1000) parameters representing syst. uncertainties

$$L(\vec{N}_{ZZ},\vec{N}_{\tau\tau},\vec{N}_{WW}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},\ldots)\cdot\prod Poisson(N^i_{\tau\tau},\ldots)\cdot\prod Poisson(N^i_{WW},\ldots)\cdot\ldots$$

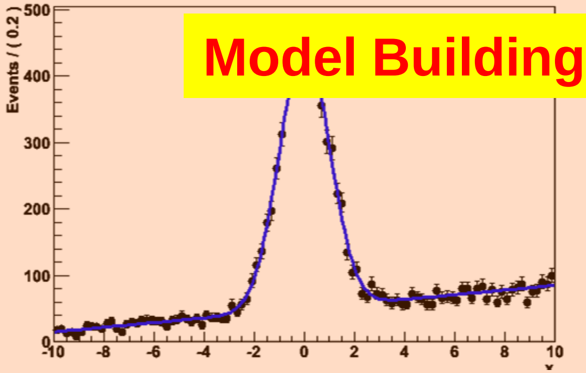
Frequentist confidence interval construction and/or p-value calculation not available as 'ready-to-run' algorithm in ROOT

How is Higgs discovery different from a simple fit?

Gaussian + polynomial

Higgs combination model

Model Building phase (formulation of L(x|H))

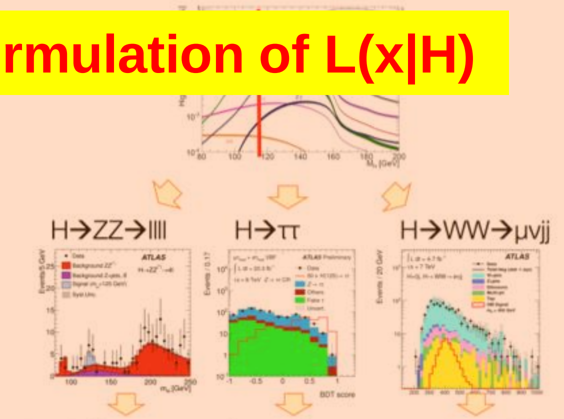


ROOT TH1

ROOT TF1

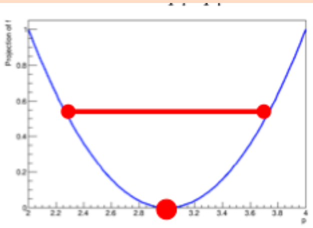
$$L(\vec{N}\,|\,\mu,\vec{\theta}) = \prod_i Poisson(N_i\,|\,f(x_i,\mu,\vec{\theta}))$$

"inside ROOT"



$L(\tilde{N}_{ZZ}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...)$ $L(\tilde{N}_{\tau\tau}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{\tau\tau},...)$ $L(\tilde{N}_{WW}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{WW},...)$

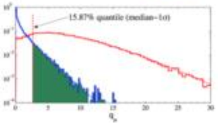
$$L(\tilde{N}_{ZZ},\tilde{N}_{\tau\tau},\tilde{N}_{WW}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...)\cdot\prod Poisson(N^i_{\tau\tau},...)\cdot\prod Poisson(N^i_{WW},...)\cdot...$$



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

μ = 5.3 ± 1.7

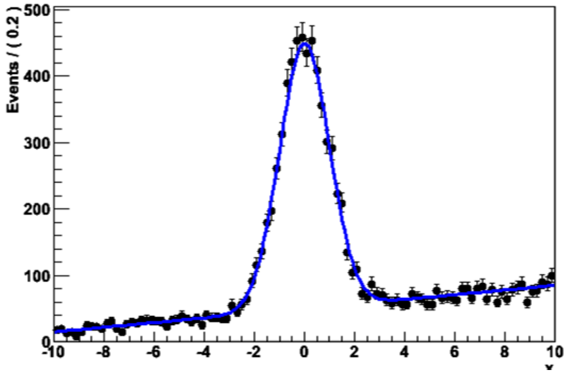
$$\lambda_\mu(\tilde{N}_{ZZ},\tilde{N}_{WW},\tilde{N}_{\tau\tau}) = \frac{L(\tilde{N}_{ZZ},\tilde{N}_{WW},\tilde{N}_{\tau\tau}\,|\,\mu,\hat{\hat{\theta}})}{L(\tilde{N}_{ZZ},\tilde{N}_{WW},\tilde{N}_{\tau\tau}\,|\,\hat{\mu},\hat{\theta})}$$



$$p(H_\mu) = \int_{\lambda_{obs}}^{\infty} f(\lambda\,|\,H_\mu)\,d\lambda = ...$$

How is Higgs discovery different from a simple fit?

Gaussian + polynomial



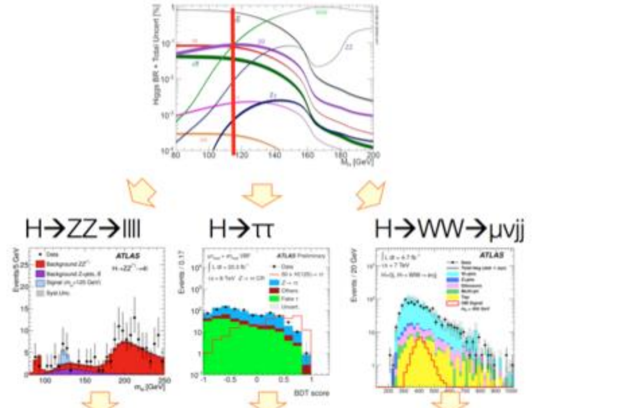
ROOT TH1

ROOT TF1

$$L(\vec{N}\,|\,\mu,\vec{\theta}) = \prod_i Poisson(N_i\,|\,f(x_i,\mu,\vec{\theta})$$

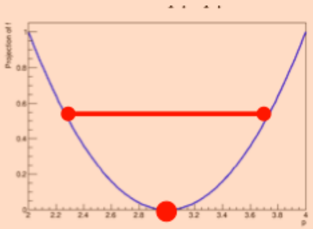
"inside ROOT"

Higgs combination model



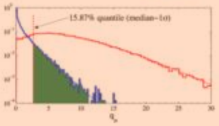
$$L(\vec{N}_{ZZ}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...)\quad L(\vec{N}_{\tau\tau}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{\tau\tau},...)\quad L(\vec{N}_{WW}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{WW},...)$$

$$L(\vec{N}_{ZZ},\vec{N}_{\tau\tau},\vec{N}_{WW}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},...)\cdot\prod Poisson(N^i_{\tau\tau},...)\cdot\prod Poisson(N^i_{WW},...)\cdot...$$



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

$$\lambda_\mu(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}) = \frac{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}\,|\,\mu,\hat{\hat{\theta}})}{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}\,|\,\hat{\mu},\hat{\theta})}$$



$$p(H_\mu) = \int_{\lambda_{obs}}^{\infty} f(\lambda\,|\,H_\mu)\,d\lambda = ...$$

Model Usage phase (use L(x|H) to make statement on H)

How is Higgs discovery different from a simple fit?

Gaussian + polynomial

Higgs combination model

Design goal:

Separate **building of Likelihood model** as much as possible from statistical analysis **using the Likelihood model**

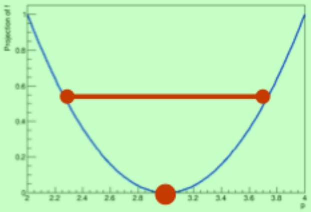
→ More modular software design

→ ‘Plug-and-play with statistical techniques

→ Factorizes work in collaborative effort

R(

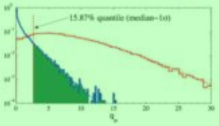
“in



ML estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

μ = 5.3 ± 1.7

$$\lambda_\mu(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau}) = \frac{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau} \mid \mu,\hat{\hat{\theta}})}{L(\vec{N}_{ZZ},\vec{N}_{WW},\vec{N}_{\tau\tau} \mid \hat{\mu},\hat{\theta})}$$



$$p(H_\mu) = \int_{\lambda_{obs}}^{\infty} f(\lambda \mid H_\mu)\, d\lambda = \ldots$$

Wouter Verkerke, NIKHEF

The idea behind the design of RooFit/RooStats/HistFactory

- Modularity, Generality and flexibility

- Step 1 – Construct the likelihood function L(x|p)

RooFit, or RooFit+HistFactory

- Step 2 – Statistical tests on parameter of interest p

  Procedure can be Bayesian, Frequentist, or Hybrid),
  but always based on L(x|p)

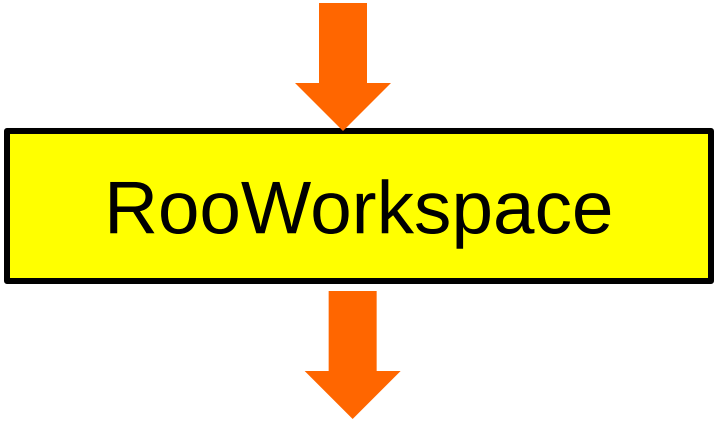
RooStats

- Steps 1 and 2 are conceptually separated,
  and in Roo* suit also implemented separately.

The idea behind the design of RooFit/RooStats/HistFactory

- Steps 1 and 2 can be ‘physically’ separated (in time, or user)
- Step 1 – Construct the likelihood function *L(x|p)*

RooFit, or RooFit+HistFactory



RooWorkspace

Complete description of likelihood model, persistable in ROOT file

(RooFit pdf function)

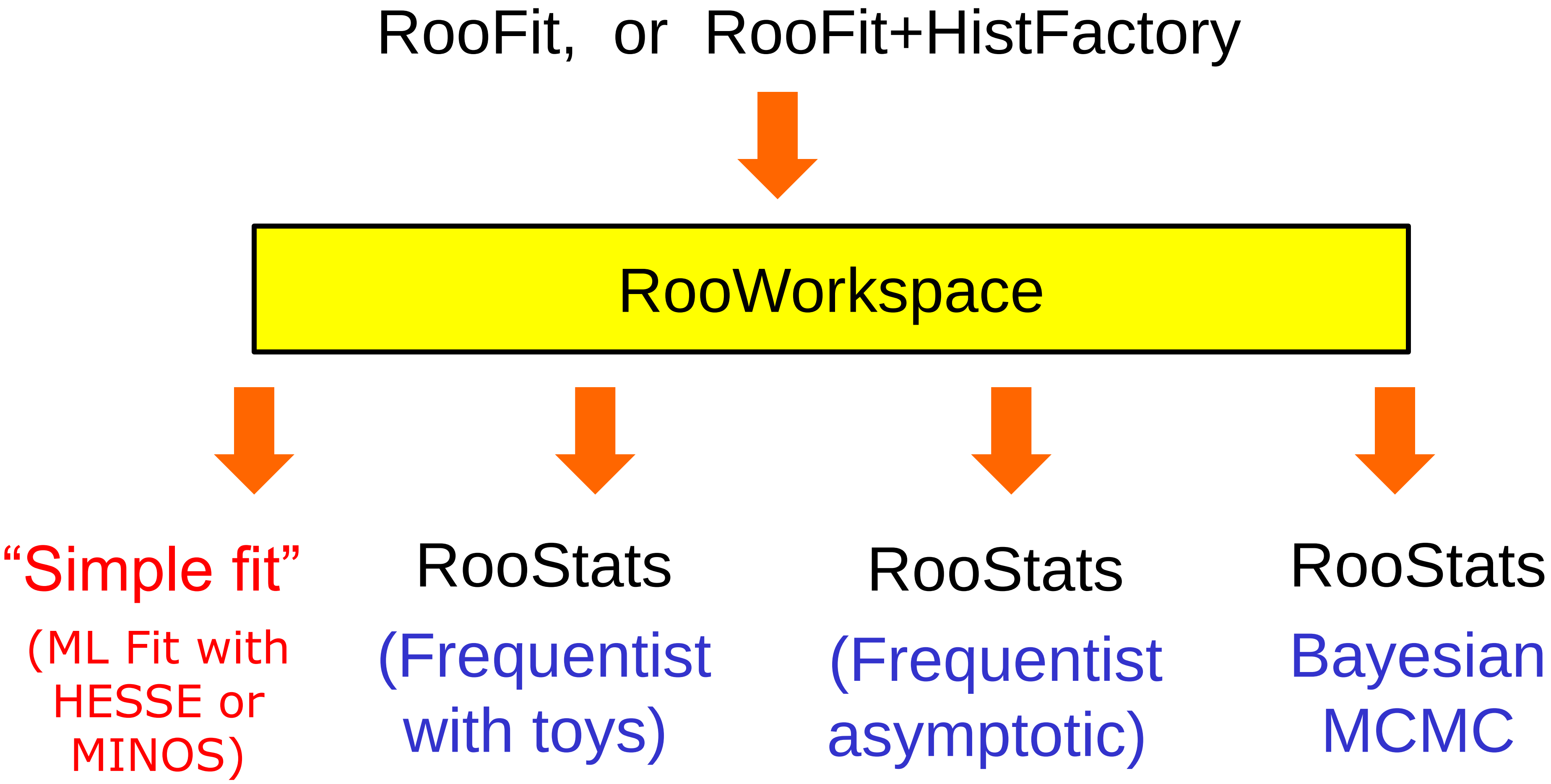
Allows full introspection and a-posteriori editing

- Step 2 – Statistical tests on parameter of interest *p*

RooStats

The benefits of modularity

- Perform different statistical test on exactly the same model



RooFit, or RooFit+HistFactory

RooWorkspace

"Simple fit"

(ML Fit with HESSE or MINOS)

RooStats

(Frequentist with toys)

RooStats

(Frequentist asymptotic)

RooStats

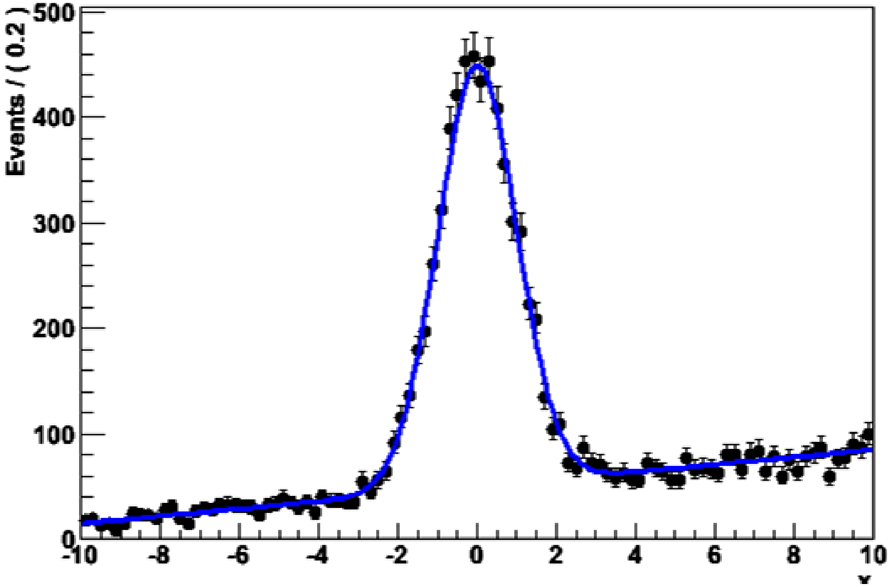
Bayesian MCMC

RooFit

WV + D. Kirkby - 1999

RooFit – Focus: coding a probability density function

- Focus on one practical aspect of many data analysis in HEP:
  How do you formulate your p.d.f. in ROOT
  - For ‘simple’ problems (gauss, polynomial) this is easy
  - But if you want to do unbinned ML fits, use non-trivial functions, or work with multidimensional functions you quickly find that you need some tools to help you



- The RooFit project started in 1999 for data modeling needs for BaBar collaboration initially, publicly available in ROOT since 2003

RooFit core design philosophy

- Mathematical objects are represented as C++ objects

Mathematical concept		RooFit class
variable	x	`RooRealVar`
function	$f(x)$	`RooAbsReal`
PDF	$f(x)$	`RooAbsPdf`
space point	$\vec{x}$	`RooArgSet`
integral	$\int_{x_{\min}}^{x_{\max}} f(x)dx$	`RooRealIntegral`
list of space points		`RooAbsData`

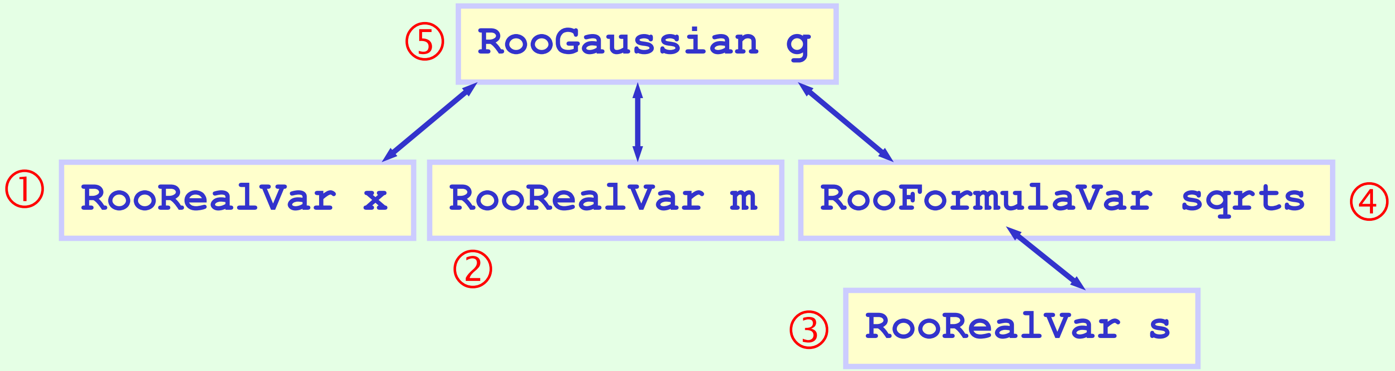
Data modeling – Constructing composite objects

- Straightforward correlation between mathematical representation of formula and RooFit code

Math

$$gauss(x, m, \sqrt{s})$$

RooFit diagram



⑤ RooGaussian g

① RooRealVar x

② RooRealVar m

RooFormulaVar sqrts ④

③ RooRealVar s

RooFit code

```
① RooRealVar x(“x”,”x”,-10,10) ;
② RooRealVar m(“m”,”mean”,0) ;
③ RooRealVar s(“s”,”sigma”,2,0,10) ;
④ RooFormulaVar sqrts(“sqrts”,”sqrt(s)”,s) ;
⑤ RooGaussian g(“g”,”gauss”,x,m,sqrts) ;
```

New feature for LHC

RooFit core design philosophy

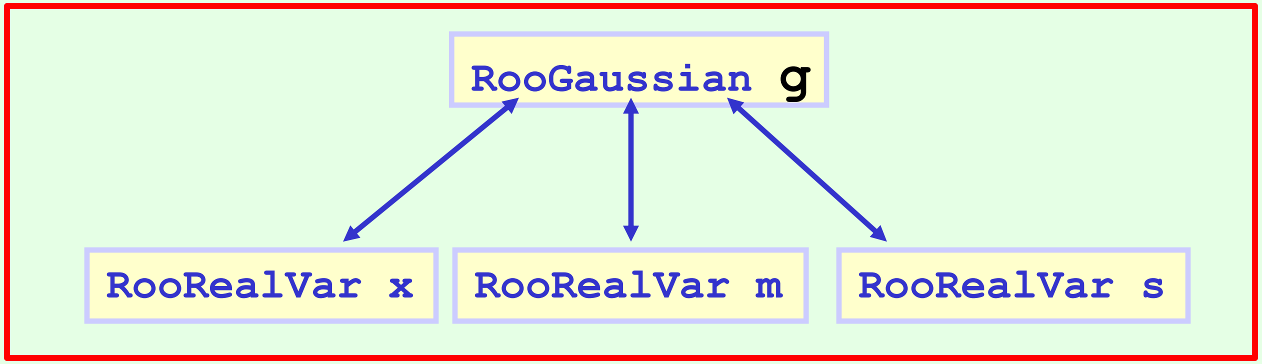
- A special container class owns all objects that together build a likelihood function

Math

$$\text{Gauss}(x,\mu,\sigma)$$

RooFit diagram

RooWorkspace (keeps all parts together)



RooFit code

```
RooRealVar x(“x”,”x”,-10,10) ;
RooRealVar m(“m”,”y”,0,-10,10) ;
RooRealVar s(“s”,”z”,3,0.1,10) ;
RooGaussian g(“g”,”g”,x,m,s) ;
RooWorkspace w(“w”) ;
w.import(g) ;
```

Populating a workspace the easy way – “the factory”

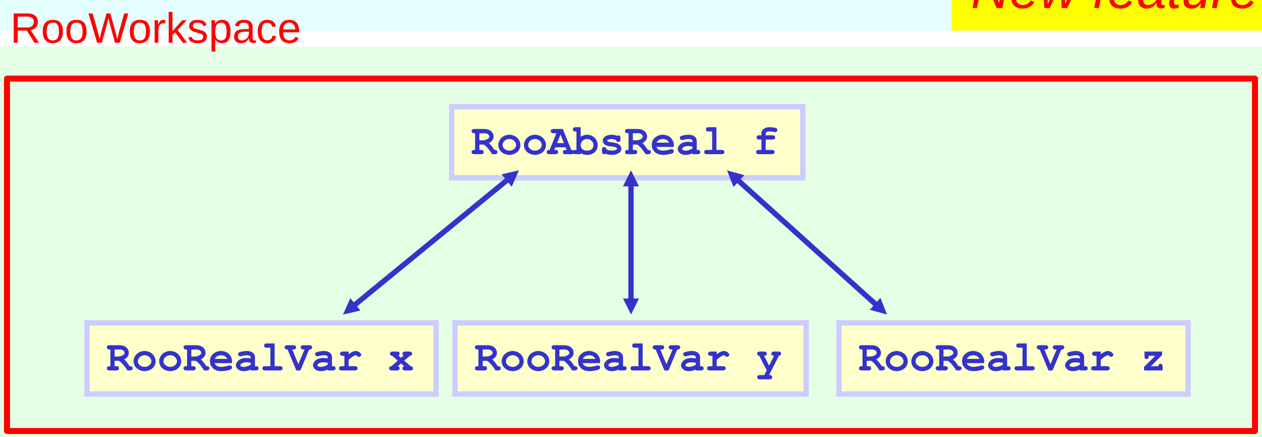
- The factory allows to fill a workspace with pdfs and variables using a simplified scripting language

Math

$$\mathrm{Gauss}(x,\mu,\sigma)$$

New feature for LHC

RooFit diagram

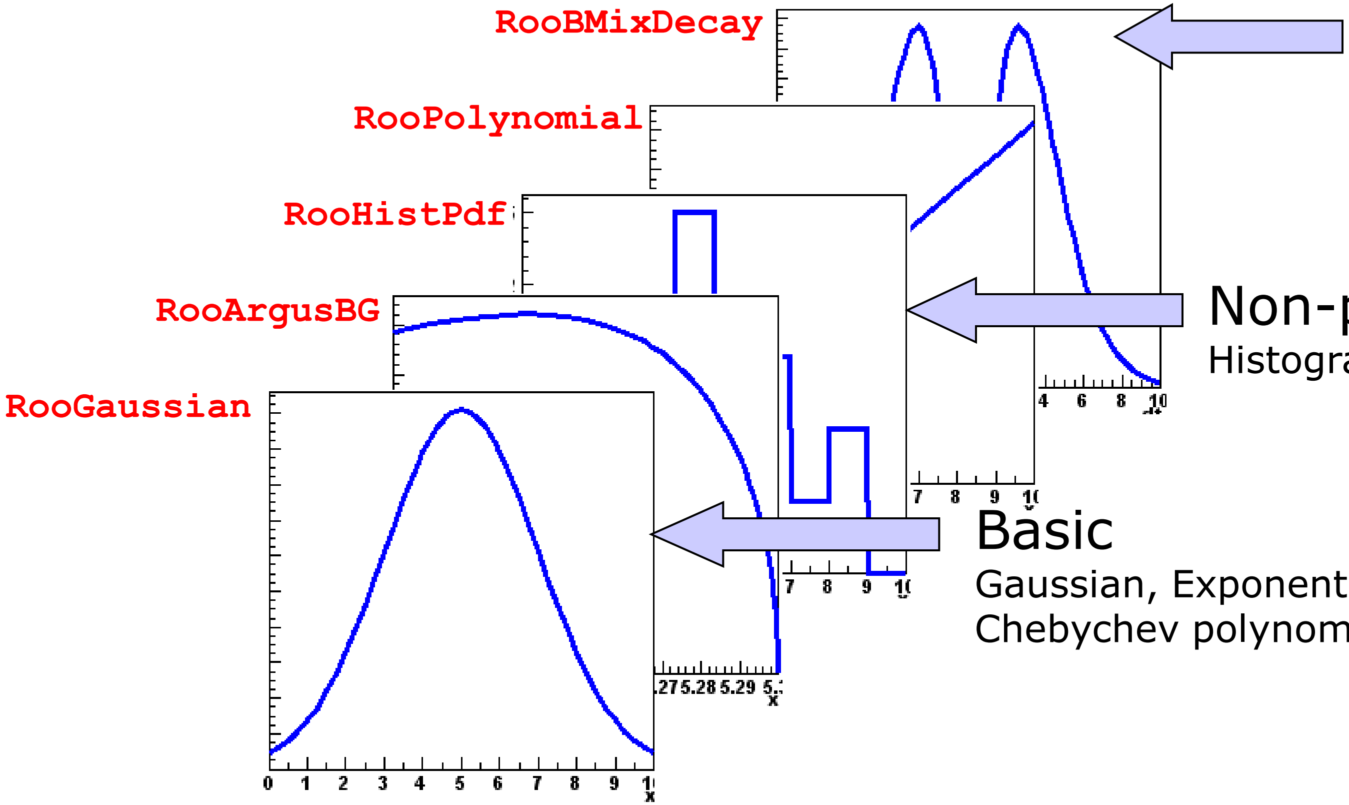


RooFit code

```
RooWorkspace w(“w”) ;
w.factory(“Gaussian::g(x[-10,10],m[-10,10],z[3,0.1,10])”);
```

Model building – (Re)using standard components

- RooFit provides a collection of compiled standard PDF classes



Physics inspired

ARGUS,Crystal Ball,
Breit-Wigner, Voigtian,
B/D-Decay,….

Non-parametric

Histogram, **KEYS**

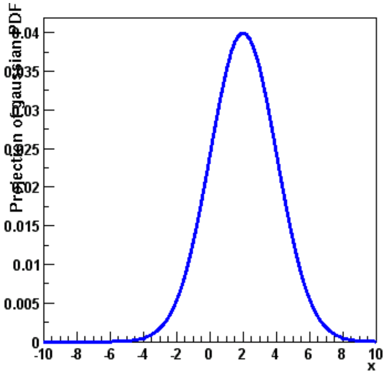
Basic

Gaussian, Exponential, Polynomial,…
Chebychev polynomial

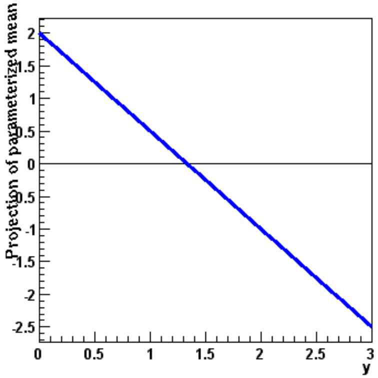
Easy to extend the library: each p.d.f. is a separate C++ class

Model building – (Re)using standard components

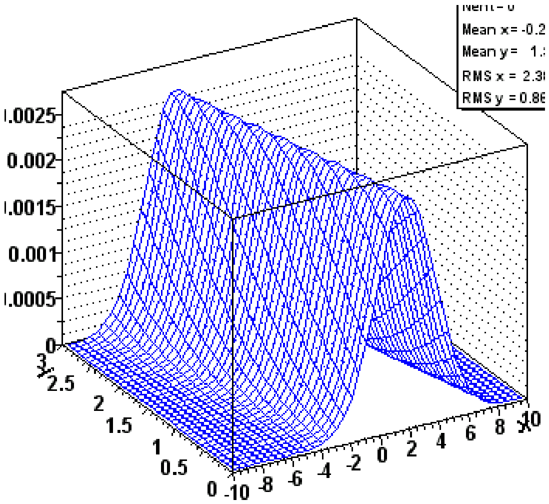
- Library p.d.f.s can be adjusted on the fly.
  - Just plug in *any function expression* you like as input variable
  - Works universally, even for classes you write yourself



$g(x;m,s)$



$m(y;a_0,a_1)$



$g(x,y;a_0,a_1,s)$

```
RooPolyVar  m(“m”,y,RooArgList(a0,a1)) ;
RooGaussian g(“g”,”gauss”,x,m,s) ;
```

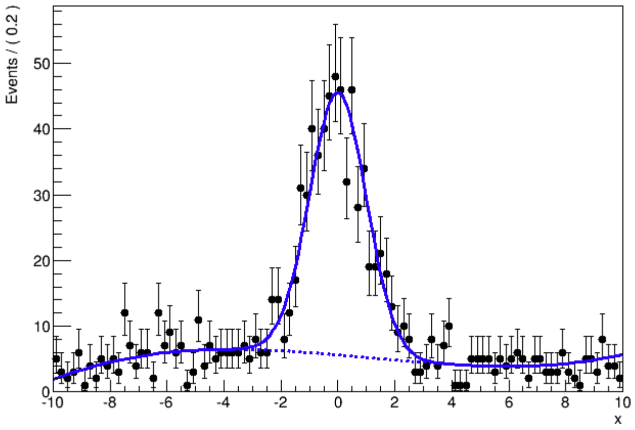
- Maximum flexibility of library shapes keeps library small

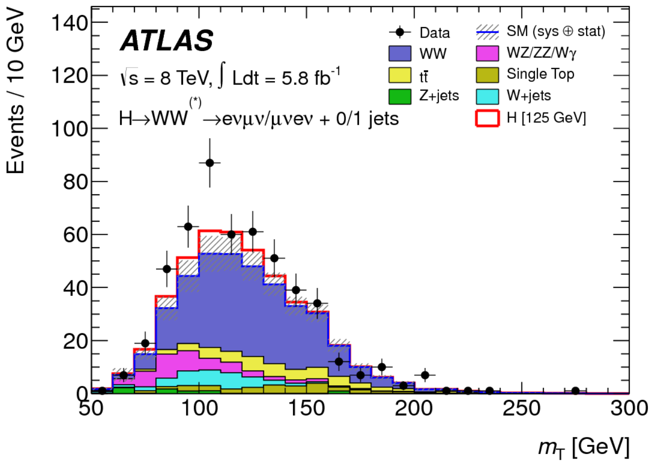
From empirical probability models to simulation-based models

- Large difference between B-physics and LHC hadron physics is that for the latter distributions usually don’t follow simple analytical shapes

Unbinned analytical probability model

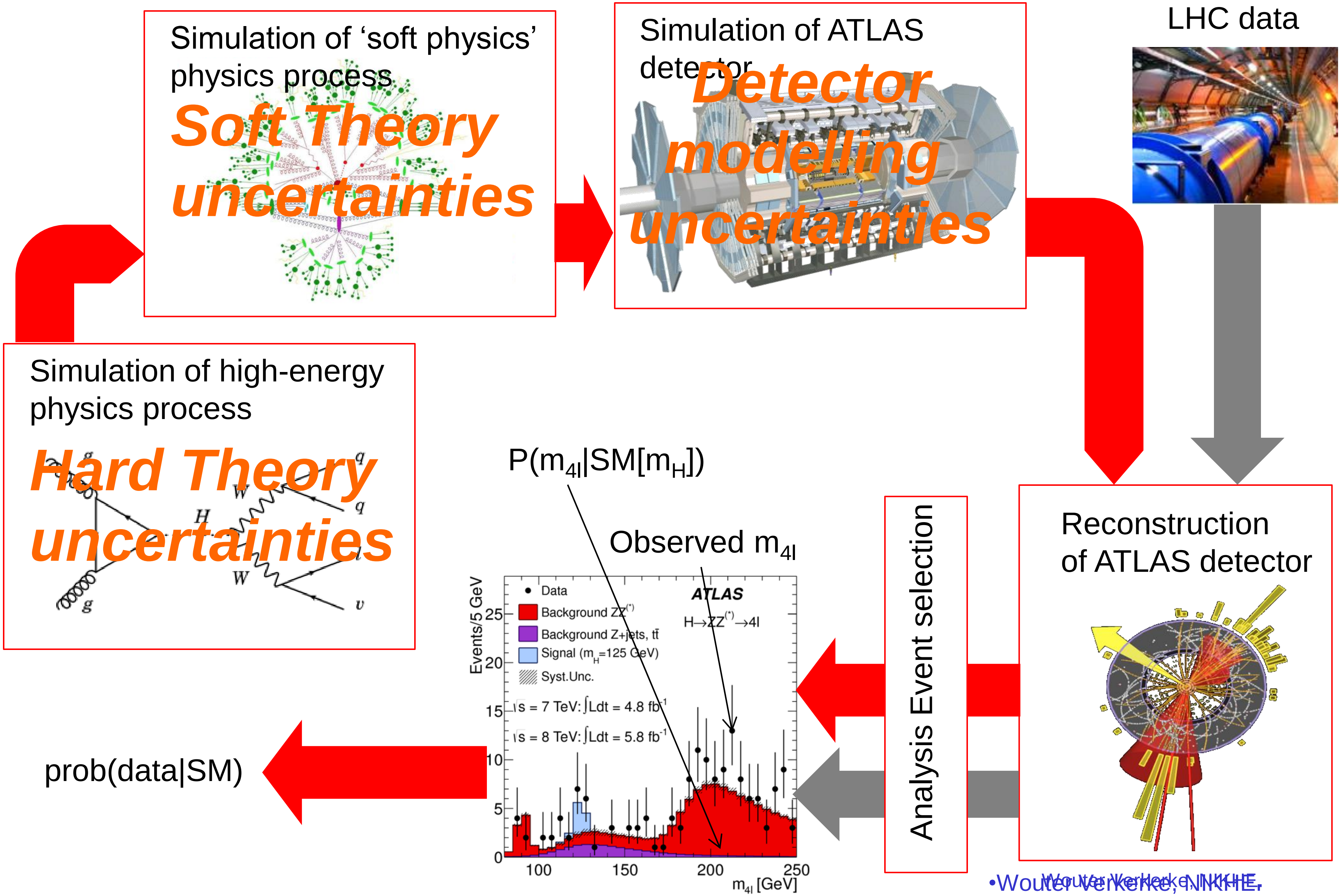
(Geant) Simulation-driven binned template model





- But concept of simulation-driven template models can also extent to systematic uncertainties. Instead of empirically chosen ‘nuisance parameters’ (e.g. polynomial coefs) construct degrees of freedom that correspond to known systematic uncertainties

The HEP analysis workflow illustrated



Simulation of ‘soft physics’ physics process

Soft Theory uncertainties

Simulation of ATLAS detector

Detector modelling uncertainties

LHC data

Simulation of high-energy physics process

Hard Theory uncertainties

Reconstruction of ATLAS detector

Analysis Event selection

$P(m_{4l}|SM[m_H])$

Observed m_{4l}

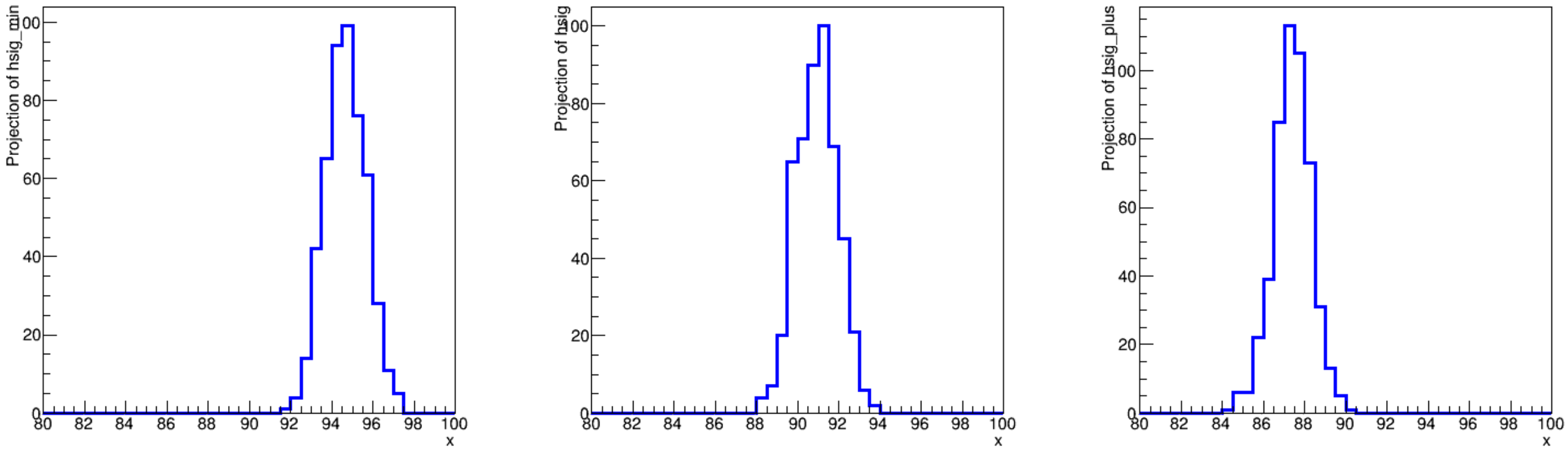
prob(data|SM)

g
q
W
q
H
l
W
g
v
Events/5 GeV
● Data
Background $ZZ^{(*)}$
Background Z+jets, $t\bar{t}$
Signal (m_H=125 GeV)
Syst.Unc.
ATLAS
$H\rightarrow ZZ^{(*)}\rightarrow 4l$
$\sqrt{s}$ = 7 TeV: $\int Ldt$ = 4.8 fb^{-1}
$\sqrt{s}$ = 8 TeV: $\int Ldt$ = 5.8 fb^{-1}
0
5
10
15
20
25
100
150
200
250
m_{4l} [GeV]

Wouter Verkerke, NIKHEF

Modeling of shape systematics in the likelihood

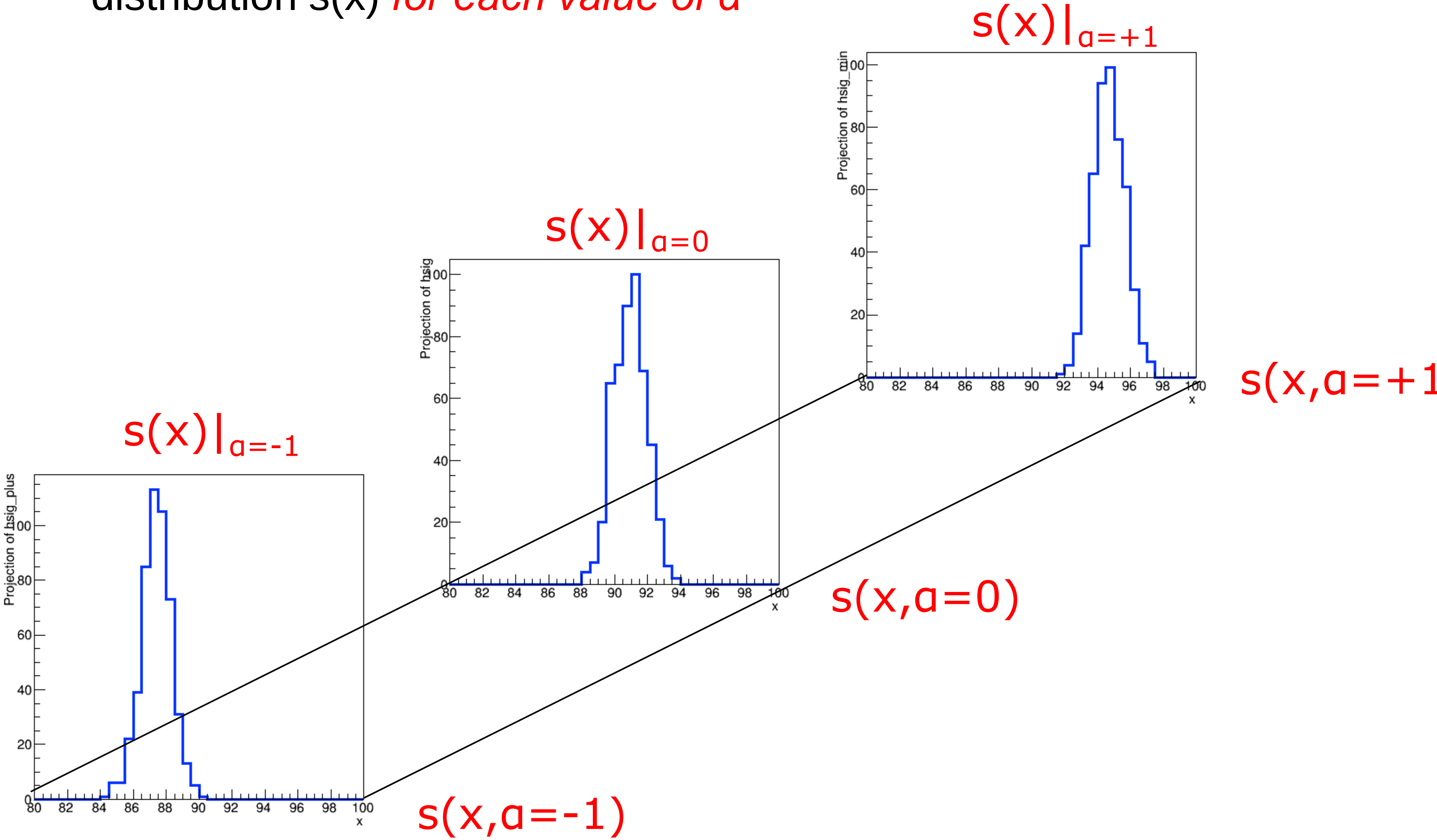
- Effect of *any* systematic uncertainty that affects the shape of a distribution can in principle be obtained from MC simulation chain
  - Obtain histogram templates for distributions at '+1σ' and '-1σ' settings of systematic effect



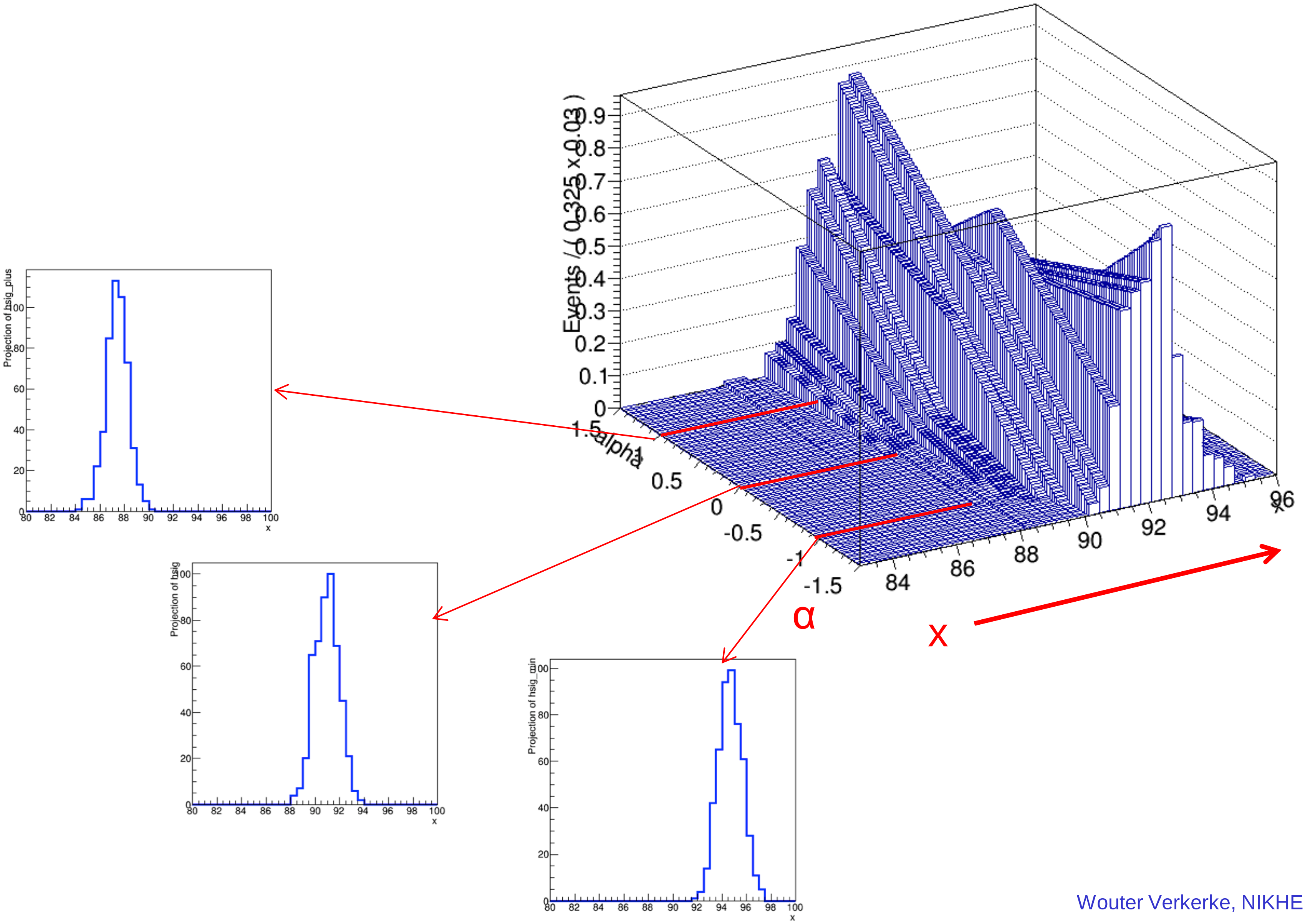
- Challenge: construct an empirical response function based on the interpolation of the shapes of these three templates.

Need to interpolate between template models

- Need to define ‘morphing’ algorithm to define distribution s(x) *for each value of α*



Visualization of bin-by-bin linear interpolation of distribution

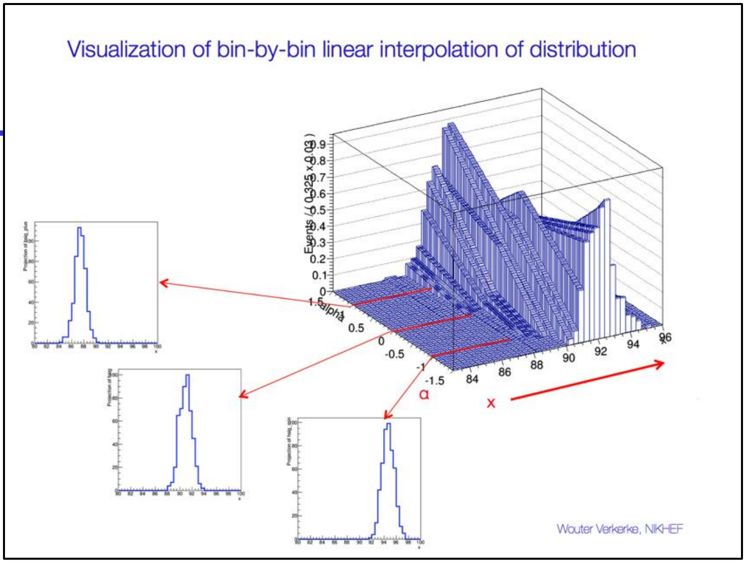


Example 2 : binned L with syst

- Example of template morphing systematic in a binned likelihood

$$s_i(\alpha,\ldots) = \begin{cases} s_i^0 + \alpha\cdot(s_i^+ - s_i^0) & \forall\ \alpha > 0 \\ s_i^0 + \alpha\cdot(s_i^0 - s_i^-) & \forall\ \alpha < 0 \end{cases}$$

$$L(\vec{N}\,|\,\alpha,\vec{s}^{\,-},\vec{s}^{\,0},\vec{s}^{\,+}) = \prod_{bins} P(N_i\,|\,\underbrace{s_i(\alpha,s_i^-,s_i^0,s_i^+)}_{})\cdot\underbrace{G(0\,|\,\alpha,1)}_{}$$



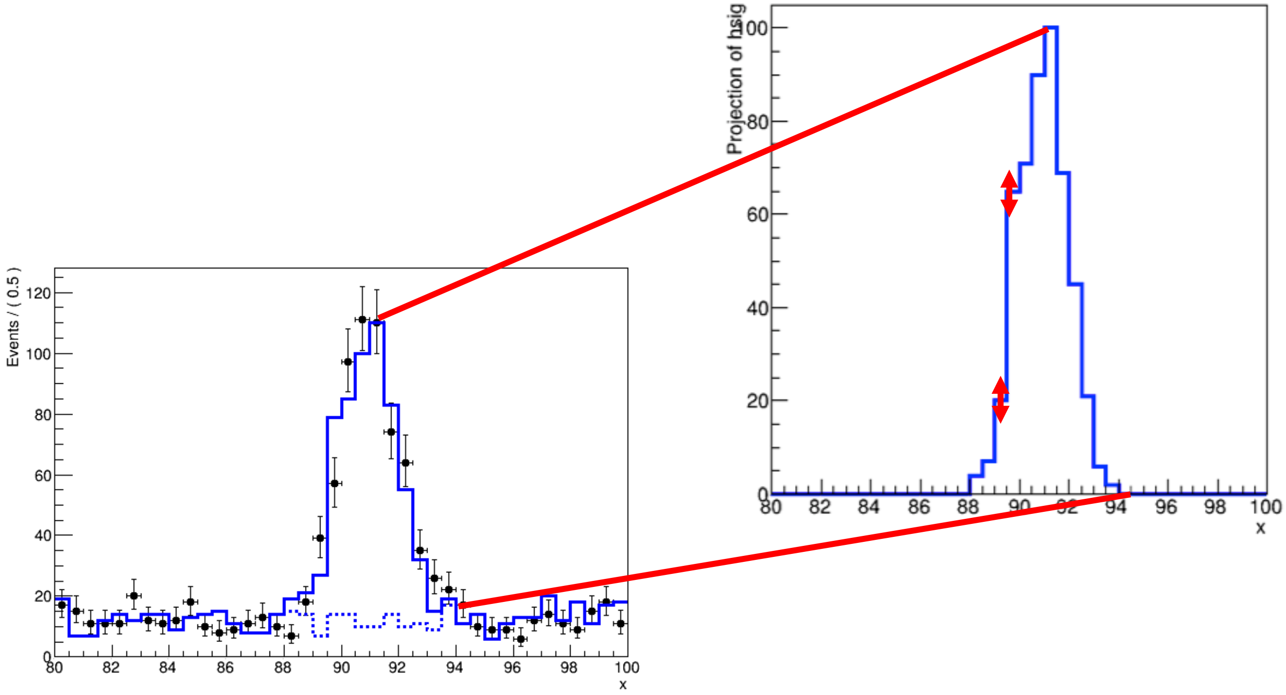
Visualization of bin-by-bin linear interpolation of distribution

Wouter Verkerke, NIKHEF

```
// Import template histograms in workspace
 w.import(hs_0,hs_p,hs_m) ;

 // Construct template models from histograms
 w.factory(“HistFunc::s_0(x[80,100],hs_0)”) ;
 w.factory(“HistFunc::s_p(x,hs_p)”) ;
 w.factory(“HistFunc::s_m(x,hs_m)”) ;

 // Construct morphing model
 w.factory(“PiecewiseInterpolation::sig(s_0,s_,m,s_p,alpha[-5,5])”) ;

 // Construct full model
 w.factory(“PROD::model(ASUM(sig,bkg,f[0,1]),Gaussian(0,alpha,1))”) ;
```

Other uncertainties in MC shapes – finite MC statistics

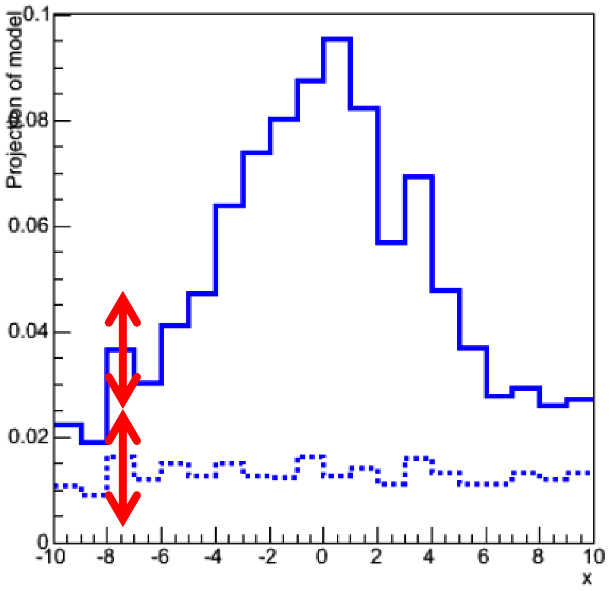
- In practice, MC distributions used for template fits have finite statistics.



- Limited MC statistics represent an uncertainty on your model → how to model this effect in the Likelihood?

Other uncertainties in MC shapes – finite MC statistics

- Modeling MC uncertainties: *each MC bin has a Poisson uncertainty*
- Thus, apply usual ‘systematics modeling’ prescription.
- For a single bin – exactly like original counting measurement



Fixed signal, bkg MC prediction

$$L_{bin-i}(\mu) = Poisson(N_i \mid \mu \cdot \tilde{s}_i + \tilde{b}_i)$$

Signal, bkg
MC nuisance params

$$L_{bin-i}(\mu, s_i, b_i) = Poisson(N_i \mid \mu \times s_i + b_i)$$

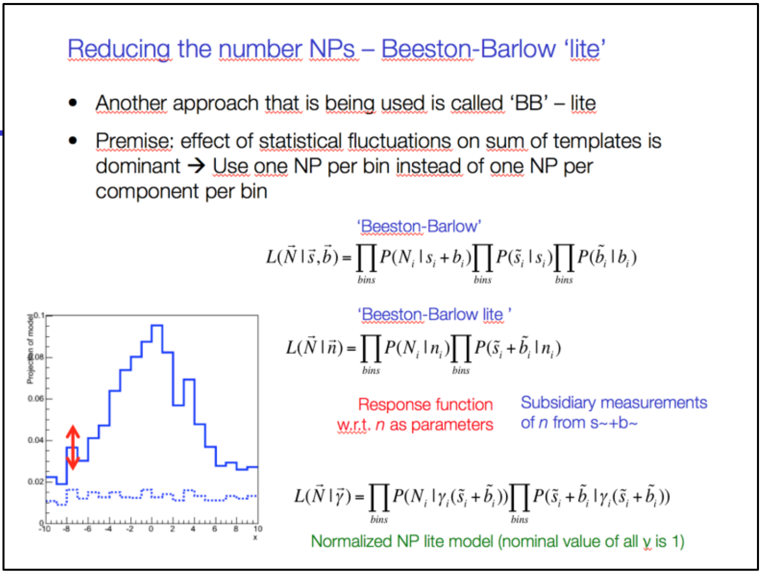
$$\times Poisson(N_i^{MC-s} \mid s_i)$$

$$\times Poisson(N_i^{MC-b} \mid b_i)$$

Subsidiary measurement for signal MC
(‘measures’ MC prediction s_i with Poisson uncertainty)

Code example – Beeston-Barlow

- Beeston-Barlow-(lite) modeling of MC statistical uncertainties



Reducing the number NPs – Beeston-Barlow ‘lite’

- Another approach that is being used is called ‘BB’ – lite
- Premise: effect of statistical fluctuations on sum of templates is dominant → Use one NP per bin instead of one NP per component per bin

‘Beeston-Barlow’

$$L(\vec{N}|\vec{s},\vec{b}) = \prod_{bins} P(N_i | s_i + b_i) \prod_{bins} P(\tilde{s}_i | s_i) \prod_{bins} P(\tilde{b}_i | b_i)$$

‘Beeston-Barlow lite ’

$$L(\vec{N}|\vec{n}) = \prod_{bins} P(N_i | n_i) \prod_{bins} P(\tilde{s}_i + \tilde{b}_i | n_i)$$

Response function w.r.t. *n* as parameters

Subsidiary measurements of *n* from s~+b~

$$L(\vec{N}|\vec{\gamma}) = \prod_{bins} P(N_i | \gamma_i(\tilde{s}_i + \tilde{b}_i)) \prod_{bins} P(\tilde{s}_i + \tilde{b}_i | \gamma_i(\tilde{s}_i + \tilde{b}_i))$$

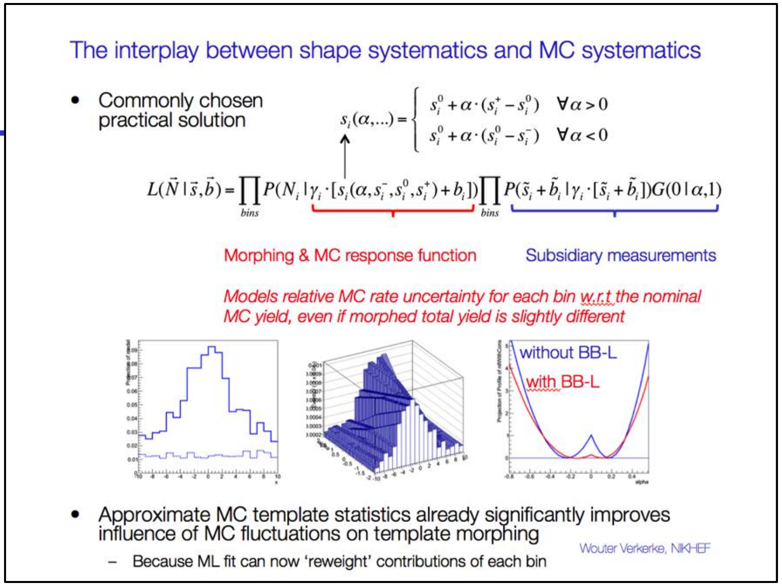
Normalized NP lite model (nominal value of all γ is 1)

$$L(\vec{N}|\vec{g}) = \underbrace{\tilde{O}_{bins} P(N_i | g_i(\tilde{s}_i + \tilde{b}_i))}_{} \underbrace{\tilde{O}_{bins} P(\tilde{s}_i + \tilde{b}_i | g_i(\tilde{s}_i + \tilde{b}_i))}_{}$$

```
// Import template histogram in workspace
 w.import(hs) ;

// Construct parametric template models from histograms
// implicitly creates vector of gamma parameters
 w.factory(“ParamHistFunc::s(hs)”) ;

 // Product of subsidiary measurement
 w.factory(“HistConstraint::subs(s)”) ;

 // Construct full model
 w.factory(“PROD::model(s,subs)”) ;
```

Code example: BB + morphing

- Template morphing model with Beeston-Barlow-lite MC statistical uncertainties



The interplay between shape systematics and MC systematics

- Commonly chosen practical solution

$$s_i(\alpha,\ldots)=\begin{cases} s_i^0+\alpha\cdot(s_i^+-s_i^0) & \forall\alpha>0\\ s_i^0+\alpha\cdot(s_i^0-s_i^-) & \forall\alpha<0\end{cases}$$

$$L(\vec{N}|\vec{s},\vec{b})=\prod_{bins}P(N_i|\gamma_i\cdot[\underbrace{s_i(\alpha,s_i^-,s_i^0,s_i^+)+b_i]}_{\text{Morphing \& MC response function}})\prod_{bins}\underbrace{P(\tilde{s}_i+\tilde{b}_i|\gamma_i\cdot[\tilde{s}_i+\tilde{b}_i])G(0|\alpha,1)}_{\text{Subsidiary measurements}}$$

Models relative MC rate uncertainty for each bin w.r.t the nominal MC yield, even if morphed total yield is slightly different

- Approximate MC template statistics already significantly improves influence of MC fluctuations on template morphing
  - Because ML fit can now ‘reweight’ contributions of each bin

Wouter Verkerke, NIKHEF

$$s_i(\alpha,\ldots)=\begin{cases} s_i^0+\alpha\cdot(s_i^+-s_i^0) & \forall\alpha>0\\ s_i^0+\alpha\cdot(s_i^0-s_i^-) & \forall\alpha<0\end{cases}$$

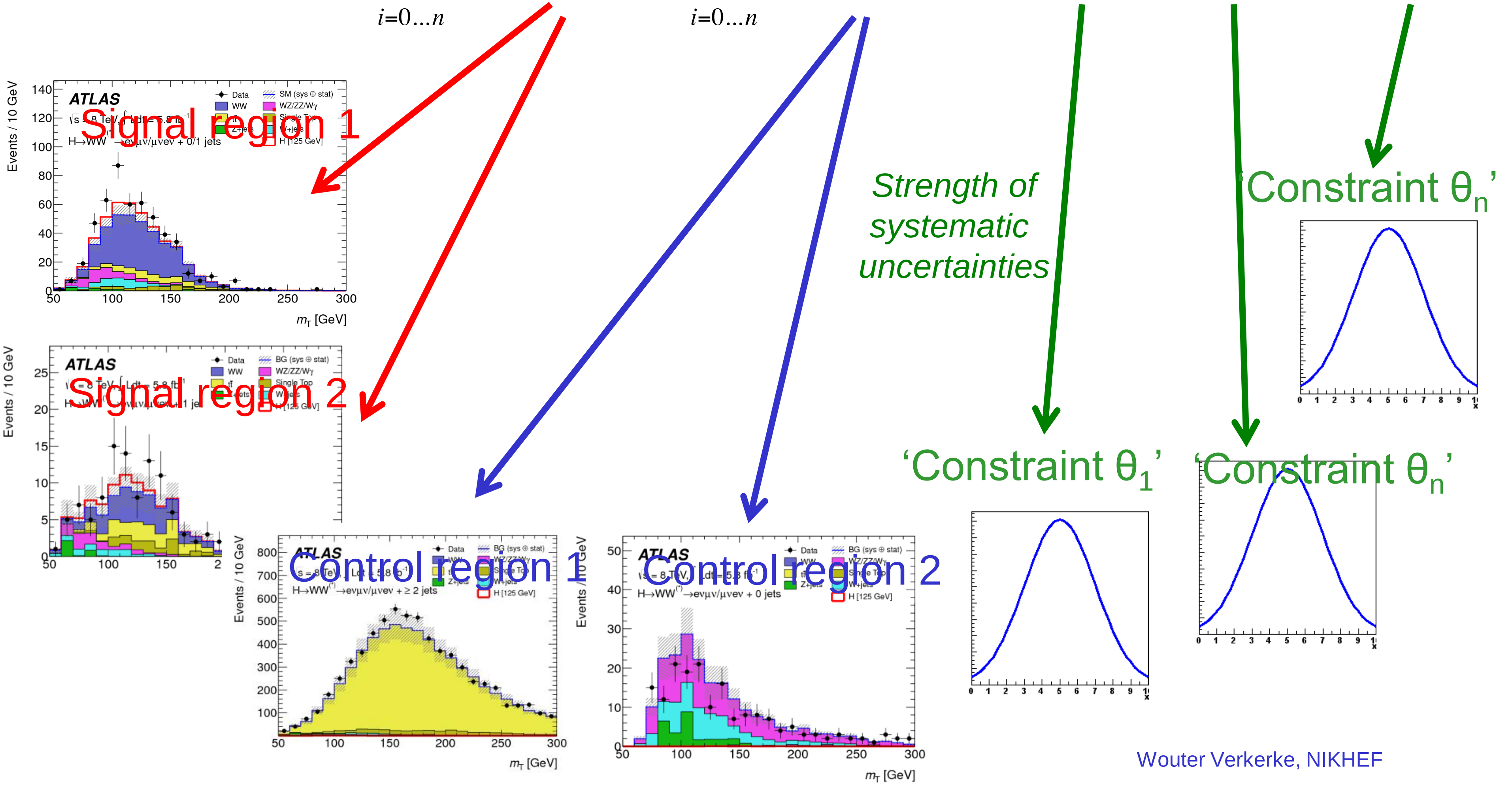
$$L(\vec{N}|\vec{s},\vec{b})=\prod_{bins}P(N_i|\gamma_i\cdot[\underbrace{s_i(\alpha,s_i^-,s_i^0,s_i^+)}+b_i])\prod_{bins}\underbrace{P(\tilde{s}_i+\tilde{b}_i|\gamma_i\cdot[\tilde{s}_i+\tilde{b}_i])G(0|\alpha,1)}$$

```
// Construct parametric template morphing signal model
w.factory(“ParamHistFunc::s_p(hs_p)”) ;
w.factory(“HistFunc::s_m(x,hs_m)”) ;
w.factory(“HistFunc::s_0(x[80,100],hs_0)”) ;
w.factory(“PiecewiseInterpolation::sig(s_0,s_,m,s_p,alpha[-5,5])”) ;

// Construct parametric background model (sharing gamma’s with s_p)
w.factory(“ParamHistFunc::bkg(hb,s_p)”) ;

// Construct full model with BB-lite MC stats modeling
w.factory(“PROD::model(ASUM(sig,bkg,f[0,1]),
           HistConstraint({s_0,bkg}),Gaussian(0,alpha,1))”) ;
```

The structure of an (Higgs) profile likelihood function

- Likelihood describing Higgs samples have following structure

$$L_{H\rightarrow X}(x\,|\,\mu,\vec{\theta}) = \prod_{i=0\ldots n} L_{phys}(x\,|\,\mu,\vec{\theta})\cdot \prod_{i=0\ldots n} L_{control}(x\,|\,\mu,\vec{\theta})\cdot L_{sub}(\theta_1)\cdot L_{sub}(\theta_{_1})\cdot\cdots\cdot L_{sub}(\theta_n)$$



Signal region 1

Signal region 2

Control region 1

Control region 2

Strength of systematic uncertainties

'Constraint θ_n'

'Constraint θ_1'

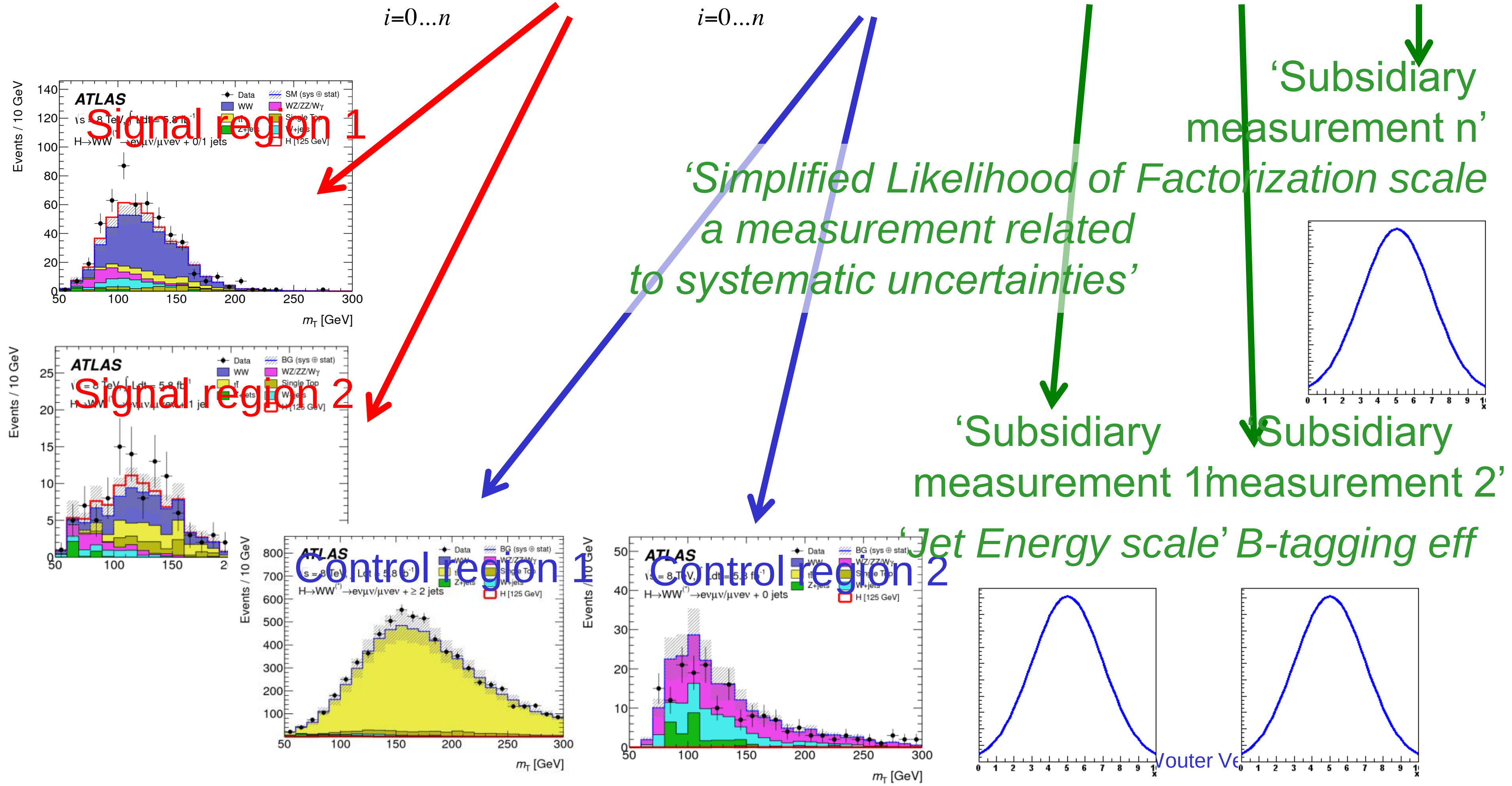
'Constraint θ_n'

ATLAS
Events / 10 GeV
m_T [GeV]
Data
WW
WZ/ZZ/Wγ
$t\bar{t}$
Single Top
Z+jets
W+jets
H [125 GeV]
SM (sys ⊕ stat)
BG (sys ⊕ stat)
$\sqrt{s}$ = 8 TeV, $\int L dt$ = 5.8 fb^{-1}
H→WW$^{(*)}$→eνμν/μνeν + 0/1 jets
H→WW$^{(*)}$→eνμν/μνeν + 1 jet
H→WW$^{(*)}$→eνμν/μνeν + ≥ 2 jets
H→WW$^{(*)}$→eνμν/μνeν + 0 jets

The structure of an (Higgs) profile likelihood function

- A simultaneous fit of physics samples and (simplified) performance measurements

$$L_{H\rightarrow X}(x\,|\,\mu,\vec{\theta}) = \prod_{i=0\ldots n} L_{phys}(x\,|\,\mu,\vec{\theta}) \cdot \prod_{i=0\ldots n} L_{control}(x\,|\,\mu,\vec{\theta}) \cdot L_{sub}(\theta_1) \cdot L_{sub}(\theta_{\ 1}) \cdot\cdots\cdot L_{sub}(\theta_n)$$



Signal region 1

Signal region 2

Control region 1

Control region 2

'Subsidiary measurement n'

'Simplified Likelihood of Factorization scale a measurement related to systematic uncertainties'

'Subsidiary measurement 1'

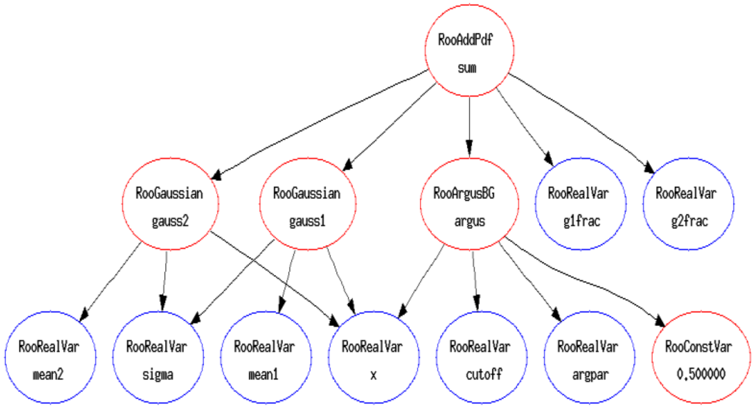
'Subsidiary measurement 2'

'Jet Energy scale' *B-tagging eff*

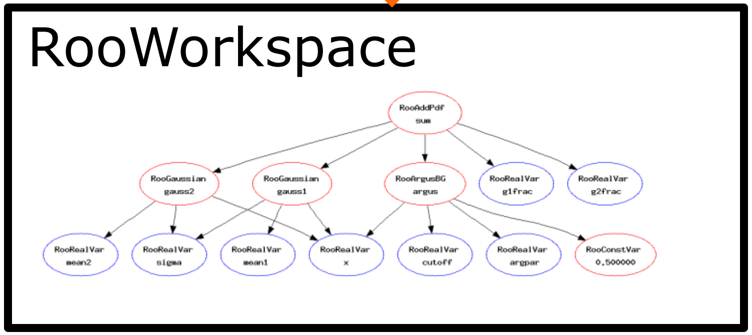
The Workspace

The workspace

- The workspace concept has revolutionized the way people share and combine analysis
  - Completely factorizes process of building and using likelihood functions
  - You can give somebody an analytical likelihood of a (potentially very complex) physics analysis in a way to the easy-to-use, provides introspection, and is easy to modify.



```
RooWorkspace w(“w”) ;
w.import(sum) ;
w.writeToFile(“model.root”) ;
```

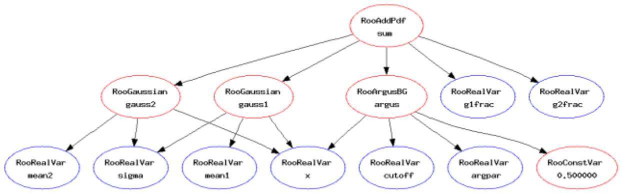


model.root

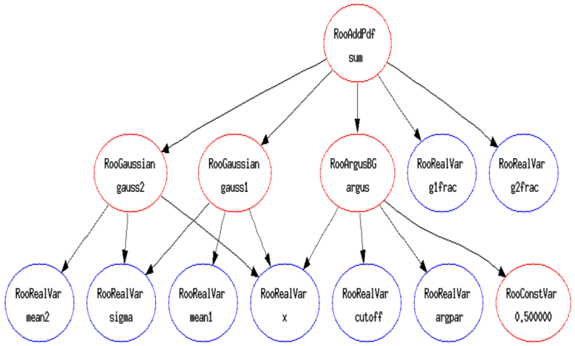
Using a workspace

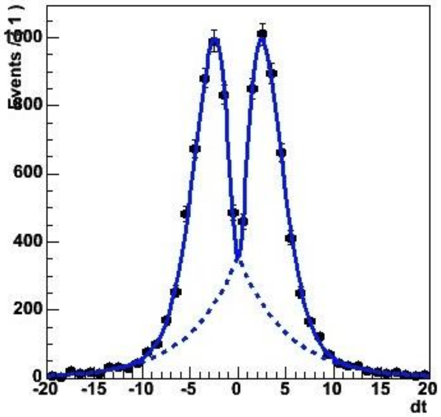
```
// Resurrect model and data
TFile f(“model.root”) ;
RooWorkspace* w = f.Get(“w”) ;
RooAbsPdf* model = w->pdf(“sum”) ;
RooAbsData* data = w->data(“xxx”) ;
```

RooWorkspace



```
// Use model and data
model->fitTo(*data) ;
RooPlot* frame =
         w->var(“dt”)->frame() ;
data->plotOn(frame) ;
model->plotOn(frame) ;
```

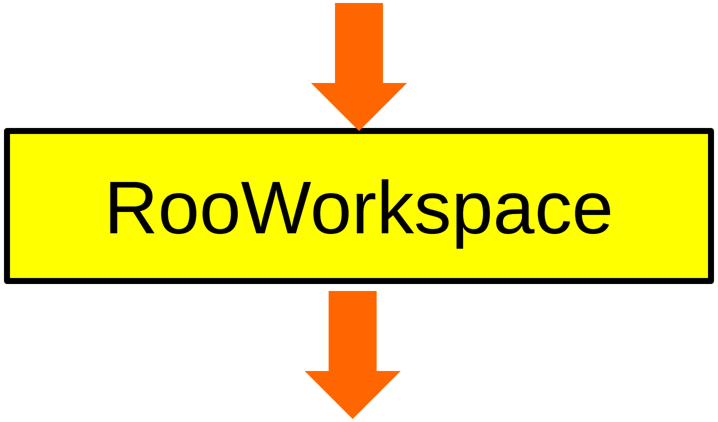




The idea behind the design of RooFit/RooStats/HistFactory

- Step 1 – Construct the likelihood function *L(x|p)*

```
RooWorkspace w(“w”) ;
w.factory(“Gaussian::sig(x[-10,10],m[0],s[1])”) ;
w.factory(“Chebychev::bkg(x,a1[-1,1])”) ;
w.factory(“SUM::model(fsig[0,1]*sig,bkg)”) ;
w.writeToFile(“L.root”) ;
```



RooWorkspace

Complete description of likelihood model, persistable in ROOT file

(RooFit pdf function)

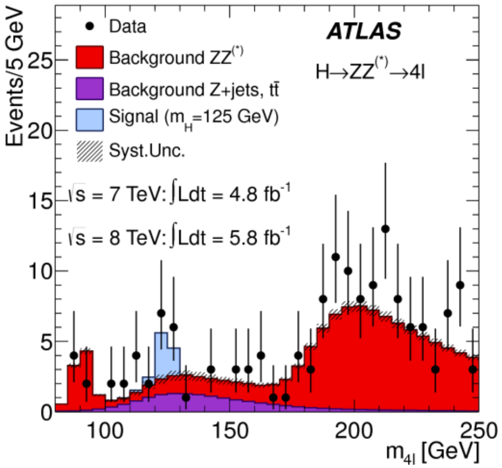
Allows full introspection and a-posteriori editing

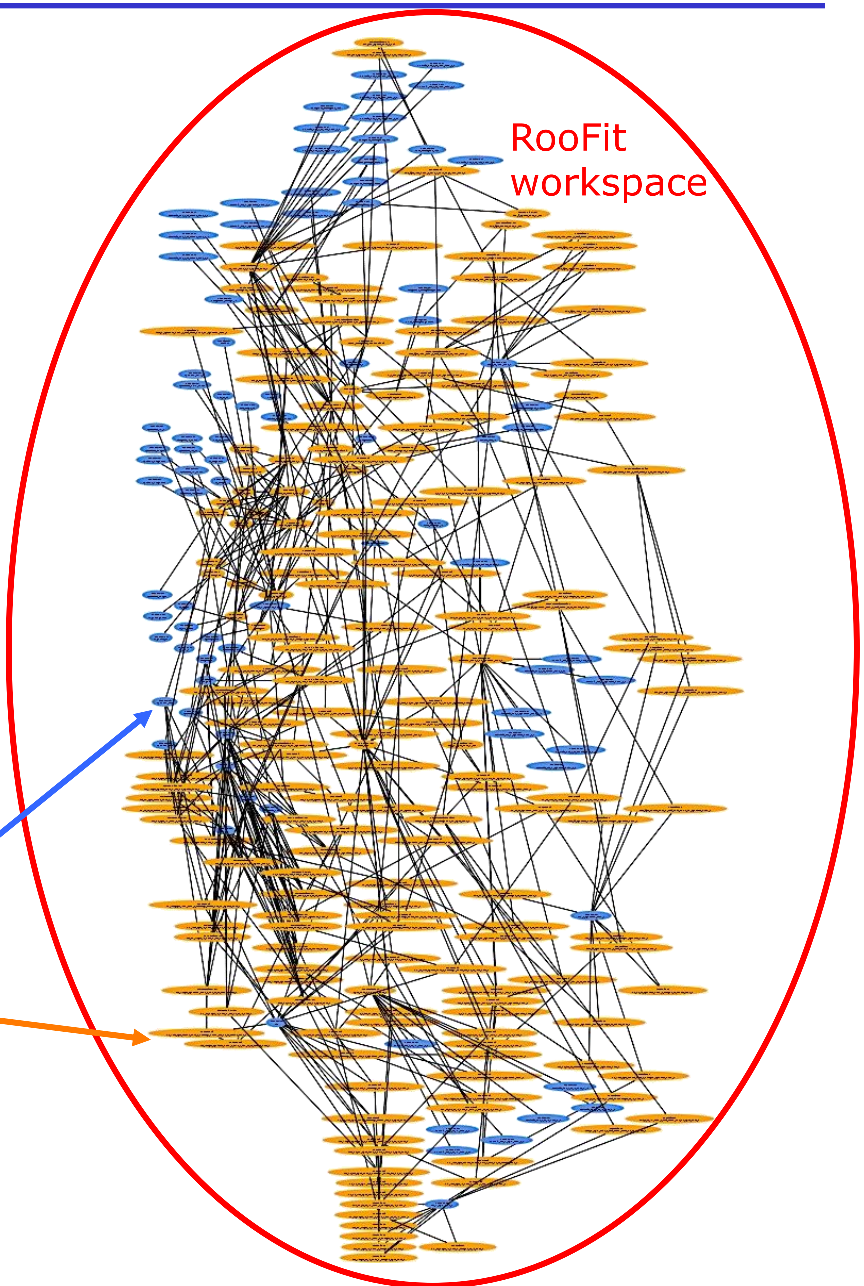
- Step 2 – Statistical tests on parameter of interest *p*

```
RooWorkspace* w=TFile::Open(“L.root”)->Get(“w”) ;
RooAbsPdf* model = w->pdf(“model”) ;
pdf->fitTo(data) ;
```

Example RooFit component model for realistic Higgs analysis

Likelihood model describing the ZZ invariant mass distribution *including all possible systematic uncertainties*





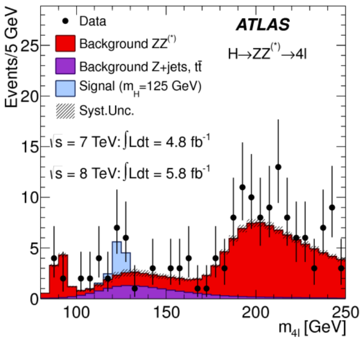
variables

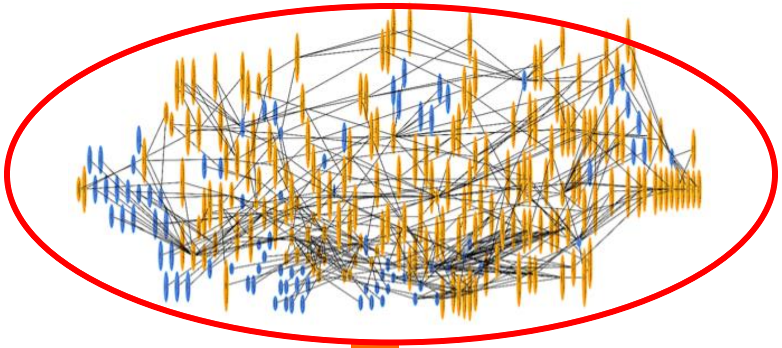
function objects

Graphical illustration of function components that call each other

Analysis chain identical for highly complex (Higgs) models

- Step 1 – Construct the likelihood function *L(x|p)*





RooWorkspace

Complete description of likelihood model, persistable in ROOT file

(RooFit pdf function)

Allows full introspection and a-posteriori editing

- Step 2 – Statistical tests on parameter of interest *p*

```
RooWorkspace* w=TFile::Open(“L.root”)->Get(“w”) ;
RooAbsPdf* model = w->pdf(“model”) ;
pdf->fitTo(data,
           GlobalObservables(w->set(“MC_GlObs”),
           Constrain(*w->st(“MC_NuisParams”) ;
```

Workspaces power collaborative statistical modelling

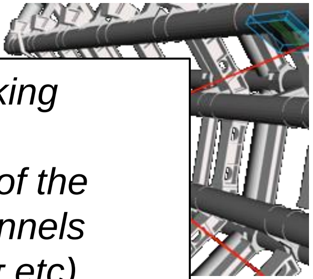
- Ability to persist complete(*) Likelihood models has profound implications for HEP analysis workflow
  - (*) Describing signal regions, control regions, and including nuisance parameters for all systematic uncertainties)
- Anyone with ROOT (and one ROOT file with a workspace) can re-run any entire statistical analysis out-of-the-box
  - About 5 lines of code are needed
  - Including estimate of systematic uncertainties
- Unprecedented new possibilities for cross-checking results, in-depth checks of structure of analysis
  - Trivial to run variants of analysis (what if 'Jet Energy Scale uncertainty' is 7% instead of 4%). Just change number and rerun.
  - But can also make structural changes a posteri. For example, rerun with assumption that JES uncertainty in forward and barrel region of detector are 100% correlated instead of being uncorrelated.

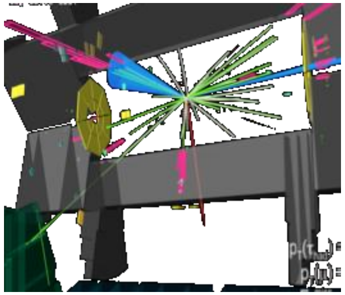
Collaborative statistical modelling

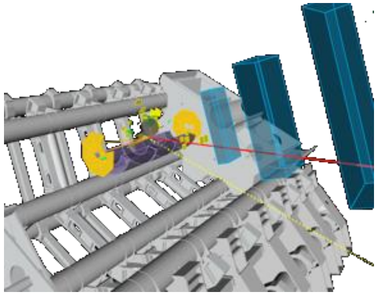
- As an experiment, you can effectively build a library of measurements, of which the full likelihood model is preserved for later use
  - Already done now, experiments have such libraries of workspace files,
  - Archived in AFS directories, or even in SVN….
  - Version control of SVN, or numbering scheme in directories allows for easy validation and debugging as new features are added
- Building of combined likelihood models greatly simplified.
  - Start from persisted components. No need to (re)build input components.
  - No need to know how individual components were built, or are internally structured. Just need to know meaning of parameters.
  - Combinations can be produced (much) later than original analyses.
  - Even analyses that were never originally intended to be combined with anything else can be included in joint likelihoods at a later time

Higgs **discovery** strategy – add everything together

H→ZZ→llll H→ττ H→WW→μνjj



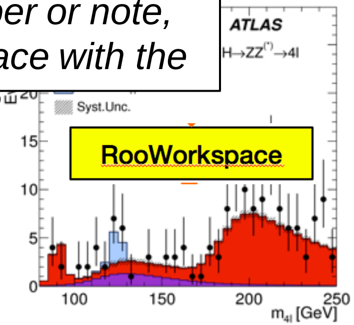




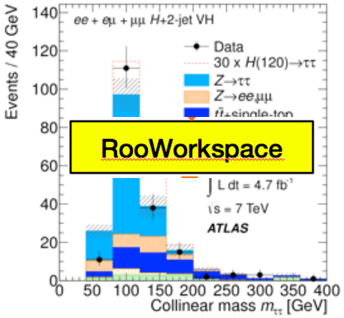
+…

Dedicated physics working groups define search for each of the major Higgs decay channels (H→WW, H→ZZ, H→ττ etc).

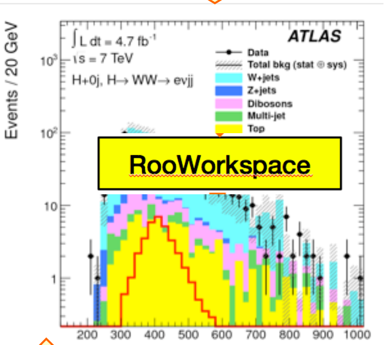
Output is physics paper or note, and a RooFit workspace with the full likelihood function



RooWorkspace

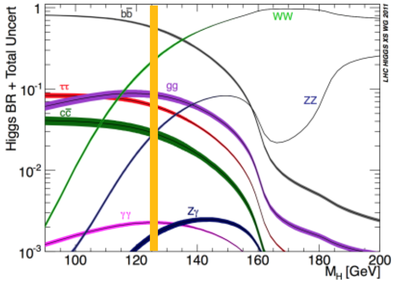


RooWorkspace



RooWorkspace

Assume SM rates



RooWorkspace

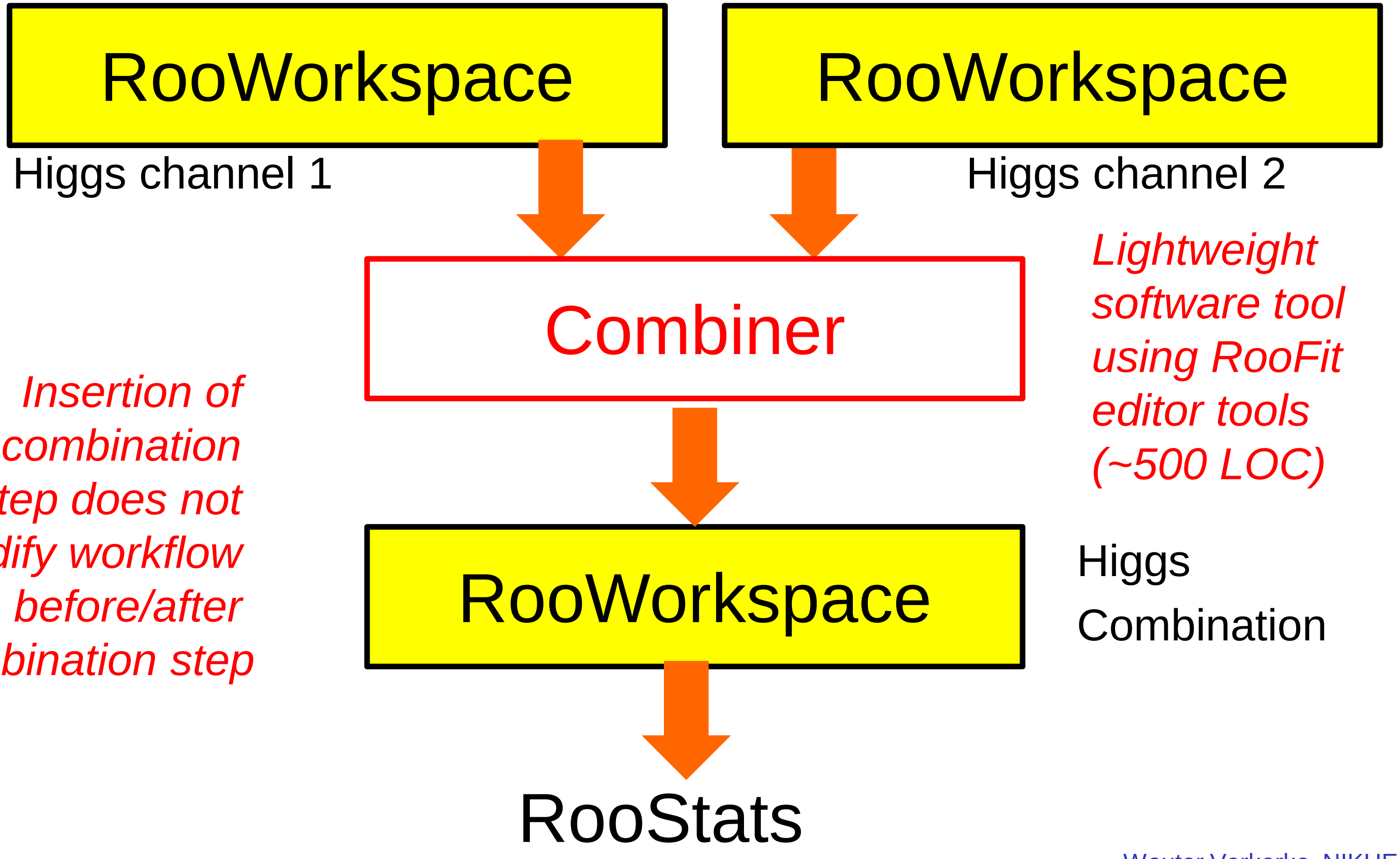
$$L(m,\vec{q}) = L_{H\rightarrow WW}(m_{WW},\vec{q})\cdot L_{H\rightarrow gg}(m_{gg},\vec{q})\cdot L_{H\rightarrow ZZ}(m_{ZZ},\vec{q})\cdot\cdots$$

A small dedicated team of specialists builds a combined likelihood from the inputs. Major discussion point: naming of parameters, choice of parameters for systematic uncertainties (a physics issue, largely)

The benefits of modularity

- Technically very straightforward to combine measurements

RooFit, or RooFit+HistFactory



RooWorkspace

Higgs channel 1

RooWorkspace

Higgs channel 2

Combiner

Lightweight software tool using RooFit editor tools (~500 LOC)

Insertion of combination step does not modify workflow before/after combination step

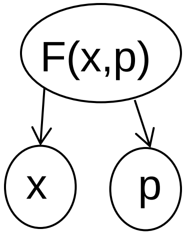
RooWorkspace

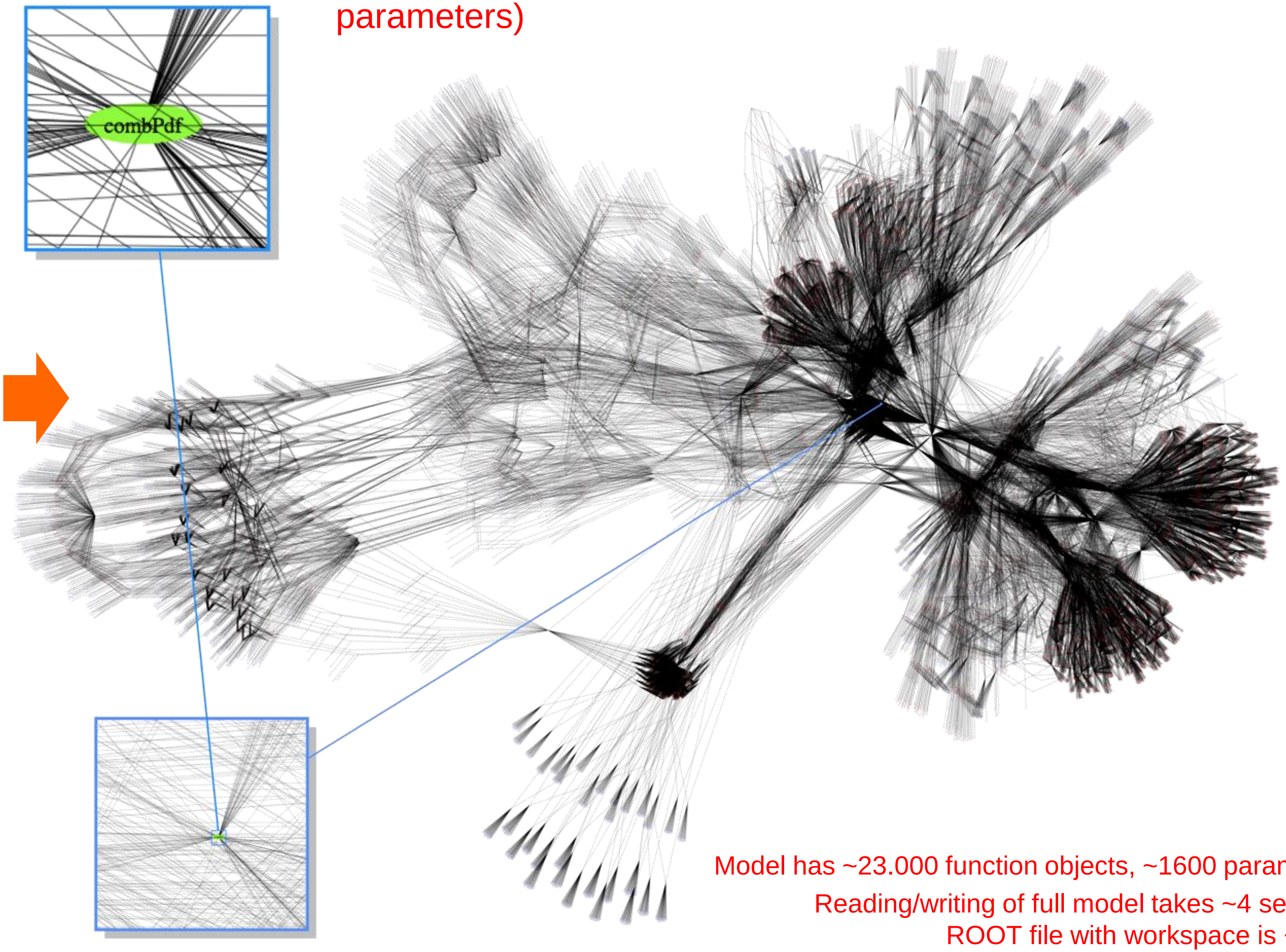
Higgs Combination

RooStats

Workspace persistence of *really* complex models works too!

Atlas Higgs combination model (23.000 functions, 1600 parameters)





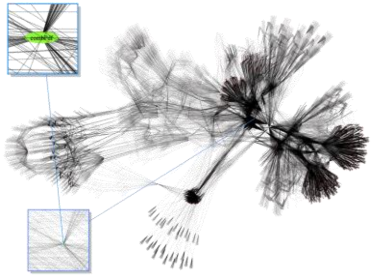
Model has ~23.000 function objects, ~1600 parameters

Reading/writing of full model takes ~4 seconds

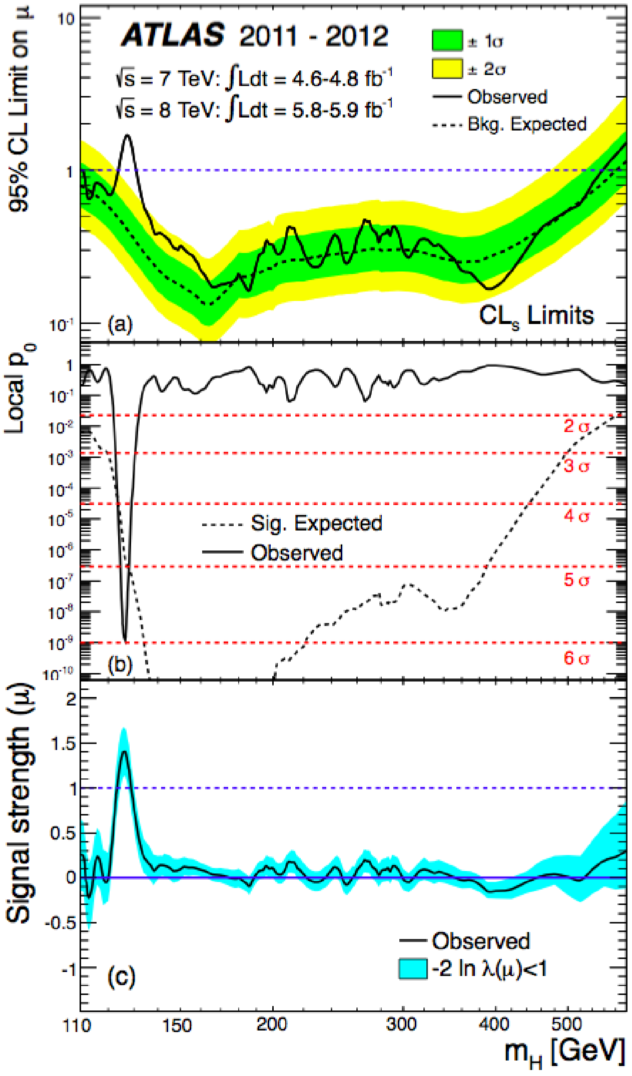
ROOT file with workspace is ~6 Mb

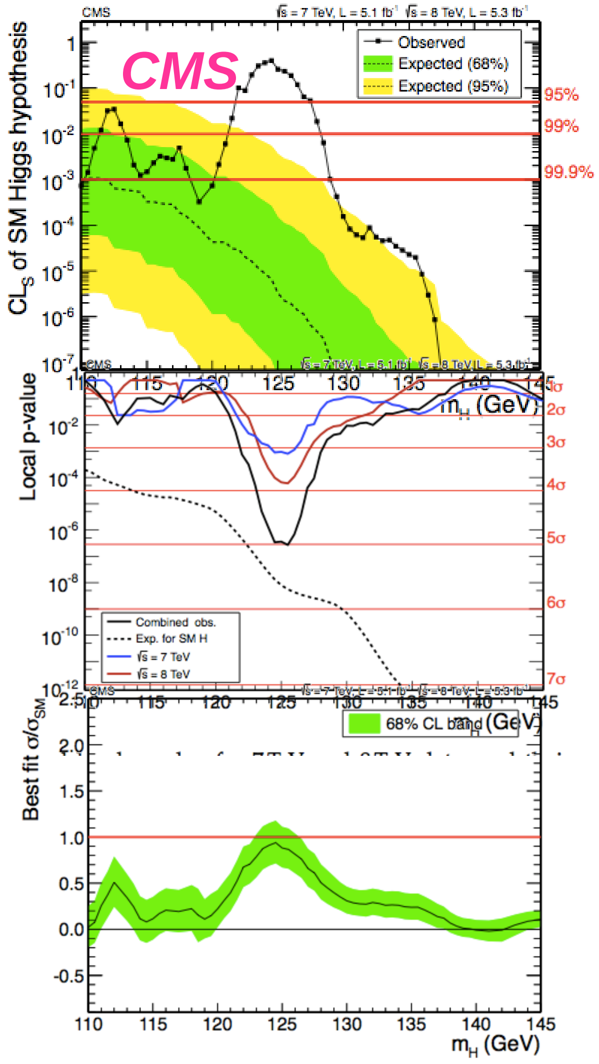
With these combined models the Higgs discovery plots were produced…

$L_{ATLAS}(\mu,\theta)$ =



Neyman construction with profile likelihood ratio test

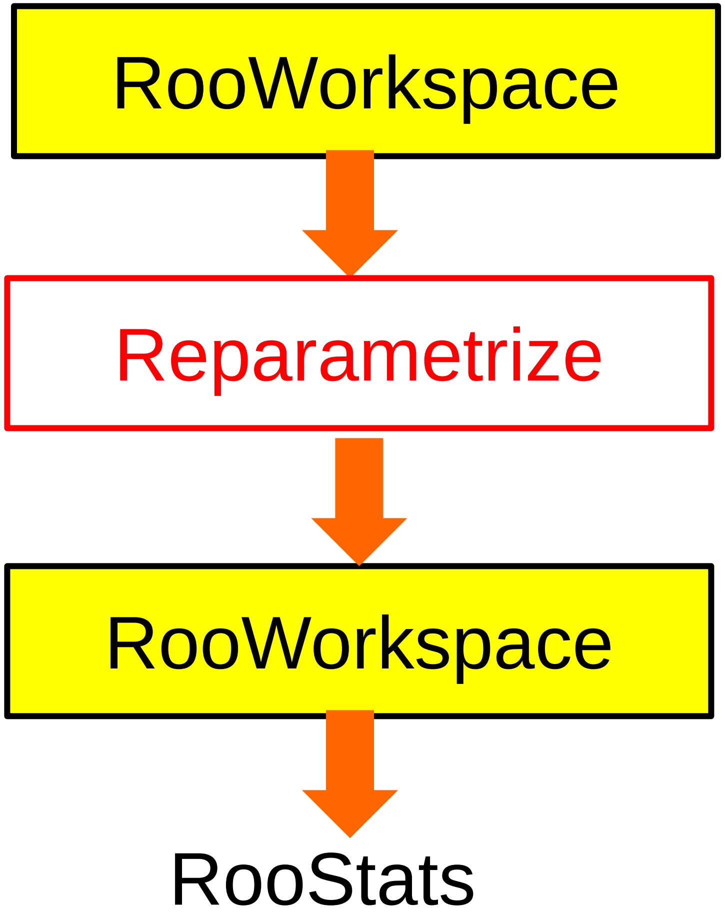




More benefits of modularity

- Technically very straightforward to reparametrize measurements

RooFit, or RooFit+HistFactory



Standard
Higgs combination

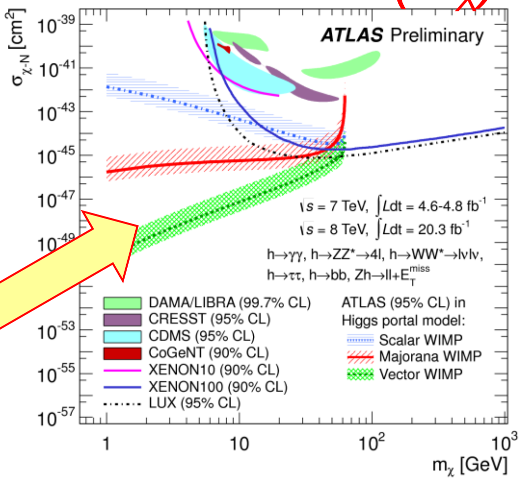
*Reparametrization step does **not** modify workflow*

Lightweight software tool using RooFit editor tools

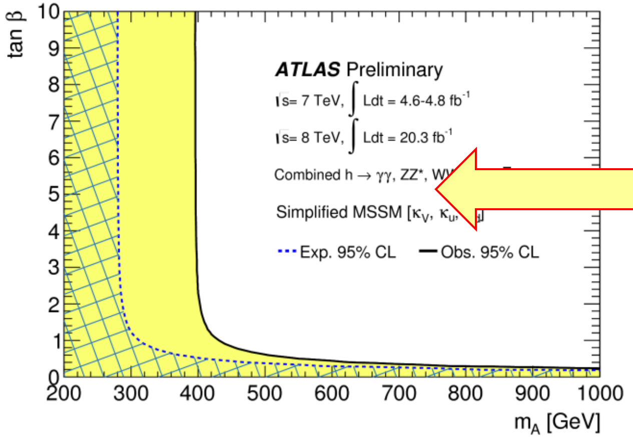
BSM
Higgs combination

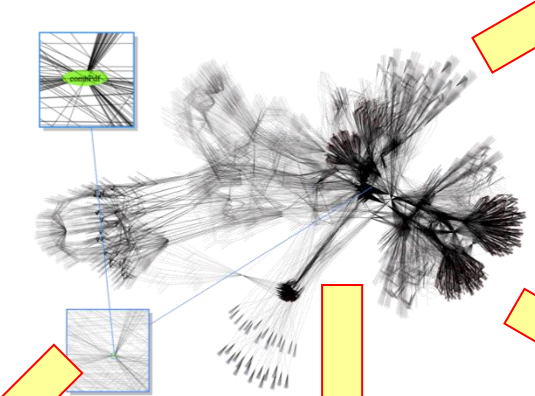
BSM Higgs constraints from reparametrization of SM Higgs Likelihood model

Portal model (m_χ)

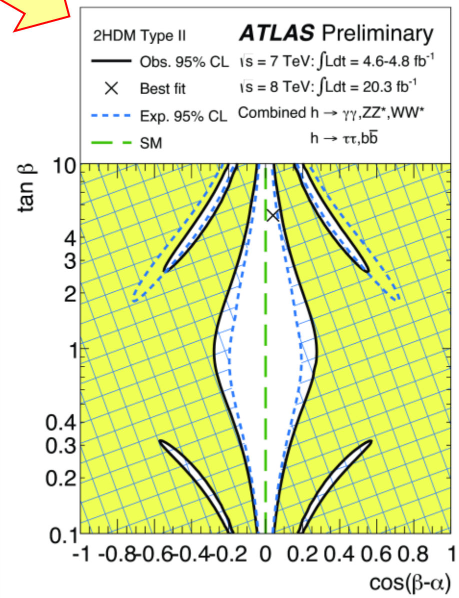


Simplified MSSM (tanβ,m_A)

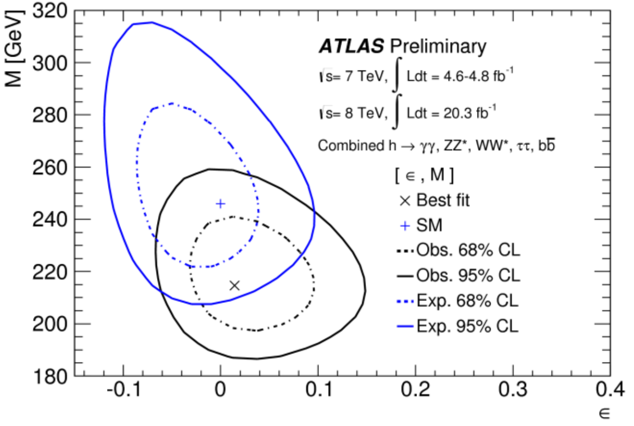




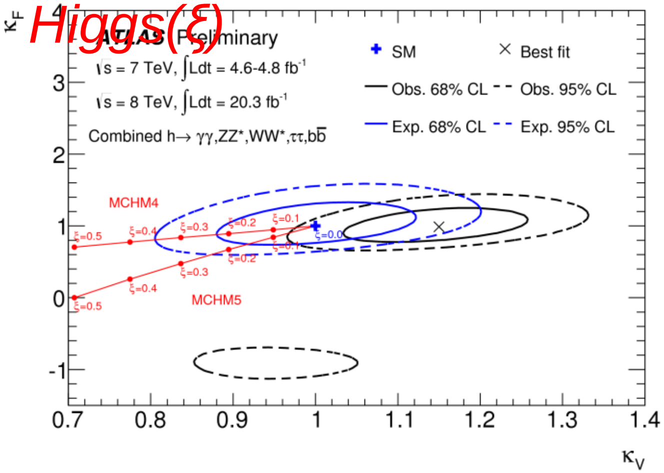
Two Higgs Double Model (tanβ,cos(α-β))



Imposter model(M,ε)



Minimal composite Higgs(ξ)



(ATLAS-CONF-2014-010)

An excursion – Collaborative analyses with workspaces

- *How can you reparametrize existing Higgs likelihoods in practice?*
- Write functions expressions corresponding to new parameterization

$$\sigma(gg \rightarrow H) * \mathrm{BR}(H \rightarrow \gamma\gamma) \quad \sim \quad \frac{\kappa_F^2 \cdot \kappa_\gamma^2(\kappa_F, \kappa_V)}{0.75 \cdot \kappa_F^2 + 0.25 \cdot \kappa_V^2}$$

```
w.factory(“expr::mu_gg_func(‘(KF2*Kg2)/
                            (0.75*KF2+0.25*KV2)’,
                            KF2,Kg2,KV2) ;
```

- Import transformation in workspace, edit *existing* model

```
w.import(mu_gg_func) ;
w.factory(“EDIT::newmodel(model,mu_gg=mu_gg_gunc)”) ;
```

HistFactory

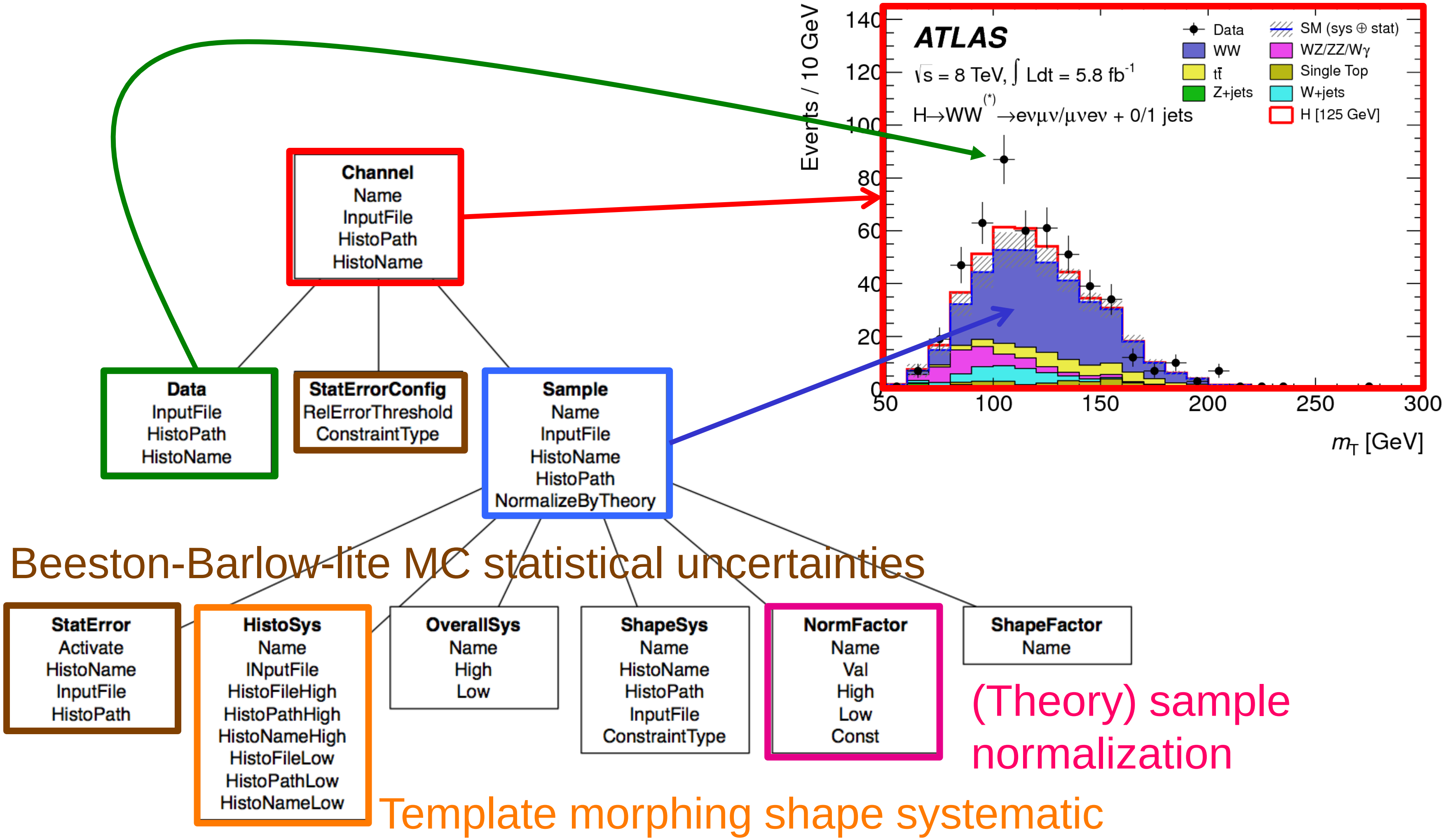
K. Cranmer, A. Shibata, G. Lewis, L. Moneta, W. Verkerke (2010)

HistFactory – structured building of binned template models

- RooFit modeling building blocks allow to easily construct likelihood models that model shape and rate systematics with one or more nuisance parameter
  - Only few lines of code per construction
- Typical LHC analysis required modeling of 10-50 systematic uncertainties in O(10) samples in anywhere between 2 and 100 channels → Need structured formalism to piece together model from specifications. This is the purpose of HistFactory
- HistFactory conceptually similar to workspace factory, but has much higher level semantics
  - Elements represent physics concepts (channels, samples, uncertainties and their relation) rather than mathematical concepts
  - Descriptive elements are represented by C++ objects (like roofit), and can be configured in C++, or alternively from an XML file
- HistFactory builds a RooFit (mathematical) model from a physics model.

HistFactory elements of a channel

- Hierarchy of concepts for description of one measurement channel



Channel
Name
InputFile
HistoPath
HistoName

Data
InputFile
HistoPath
HistoName

StatErrorConfig
RelErrorThreshold
ConstraintType

Sample
Name
InputFile
HistoName
HistoPath
NormalizeByTheory

Beeston-Barlow-lite MC statistical uncertainties

StatError
Activate
HistoName
InputFile
HistoPath

HistoSys
Name
INputFile
HistoFileHigh
HistoPathHigh
HistoNameHigh
HistoFileLow
HistoPathLow
HistoNameLow

OverallSys
Name
High
Low

ShapeSys
Name
HistoName
HistoPath
InputFile
ConstraintType

NormFactor
Name
Val
High
Low
Const

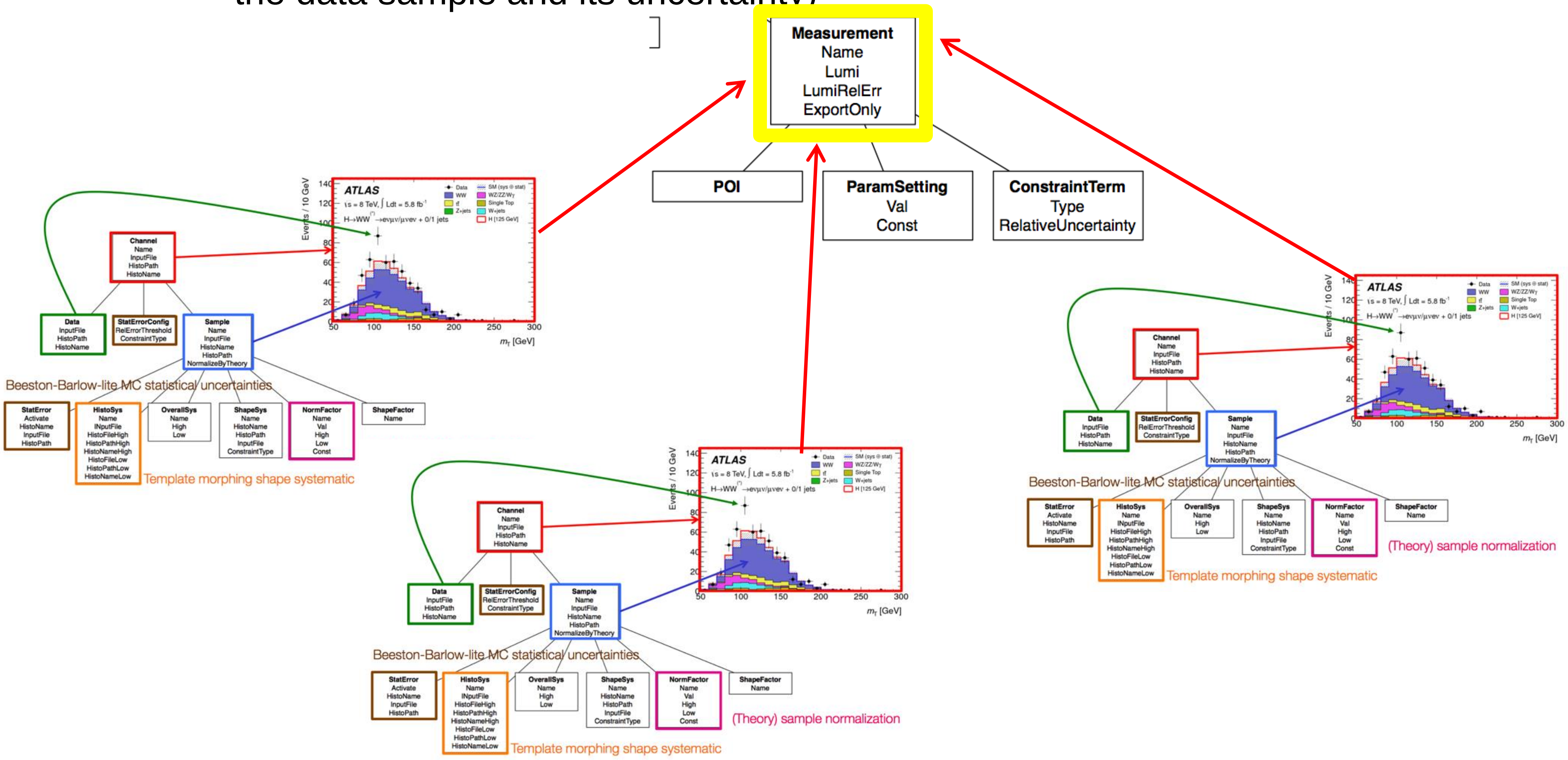
ShapeFactor
Name

(Theory) sample normalization

Template morphing shape systematic

HistFactory elements of measurement

- One or more channels are combined to form a measurement
  - Along with some extra information (declaration of the POI, the luminosity of the data sample and its uncertainty)

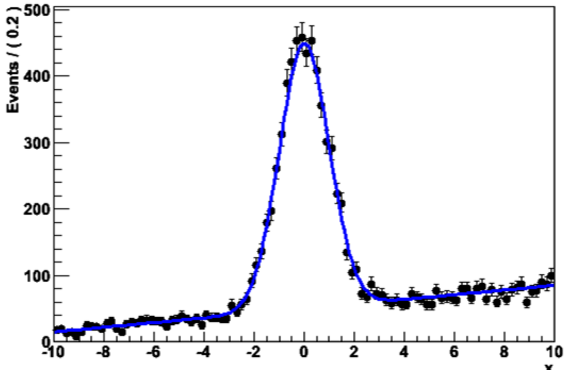


Once physics model is defined, one line of code will turn it into a RooFit likelihood

Wouter Verkerke, NIKHEF

How is Higgs discovery different from a simple fit?

Gaussian + polynomial



ROOT TH1

ROOT TF1

$$L(\vec{N}\,|\,\mu,\vec{\theta}) = \prod_i Poisson(N_i\,|\,f(x_i,\mu,\vec{\theta}))$$

"inside ROOT"

Maximum Likelihood estimation of parameters μ,θ using MINUIT (MIGRAD, HESSE, MINOS)

μ = 5.3 ± 1.7

Likelihood Model orders of magnitude more complicated. Describes

- **O(100) signal dis utions**
- **O(100) ntro mple distr.**
- **O(1000 eters esenting syst. uncertainties**

✓

$$L(\vec{N}_{ZZ},\vec{N}_{\tau\tau},\vec{N}_{WW}\,|\,\mu,\vec{\theta}) = \prod Poisson(N^i_{ZZ},\ldots)\cdot\prod Poisson(N^i_{\tau\tau},\ldots)\cdot\prod Poisson(N^i_{WW},\ldots)\cdot\ldots$$

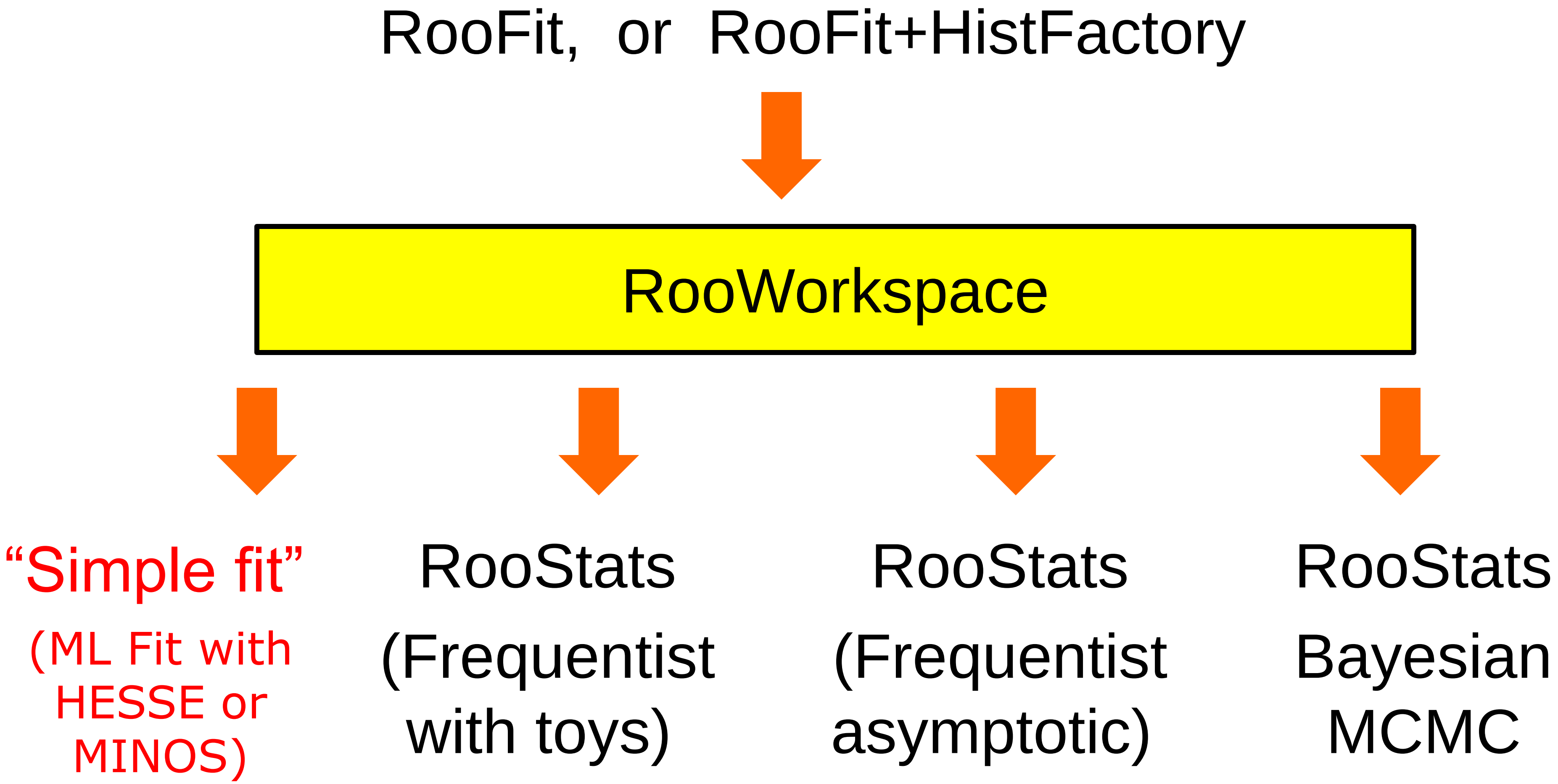
Frequentist confidence interval construction and/or p-value calculation not available as 'ready-to-run' algorithm in ROOT

RooStats

K. Cranmer, L. Moneta, S. Kreiss, G. Kukartsev, G. Schott, G. Petrucciani, WV - 2008

The benefits of modularity

- Perform different statistical test on exactly the same model



RooFit, or RooFit+HistFactory

RooWorkspace

“Simple fit” (ML Fit with HESSE or MINOS)

RooStats (Frequentist with toys)

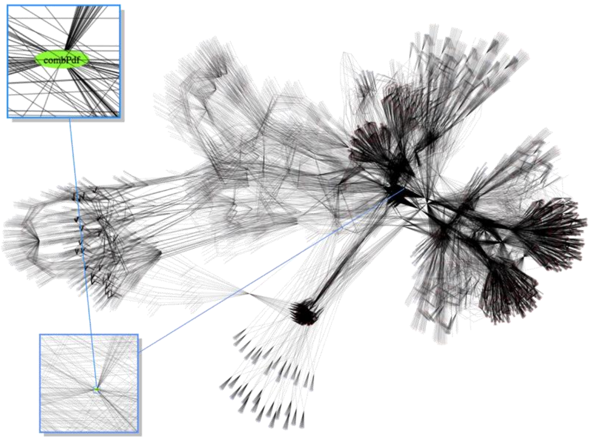
RooStats (Frequentist asymptotic)

RooStats Bayesian MCMC

Maximum Likelihood estimation as simple statistical analysis

- Step 1 – Construct the likelihood function *L(x|p)*

```
RooWorkspace w(“w”) ;
w.factory(“Gaussian::sig(x[-10,10],m[0],s[1])”;
w.factory(“Chebychev::bkg(x,a1[-1,1])”) ;
w.factory(“SUM::model(fsig[0,1]*sig,bkg)”) ;
w.writeToFile(“L.root”) ;
```



RooWorkspace

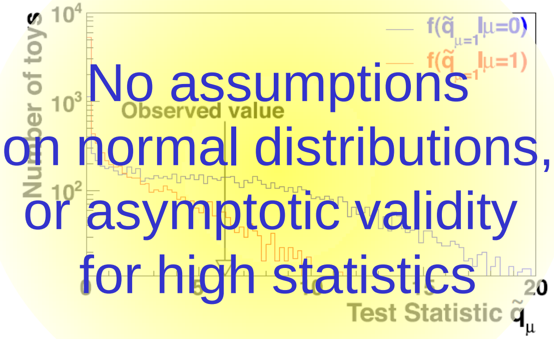
- Step 2 – Statistical tests on parameter of interest *p*

```
RooWorkspace* w=TFile::Open(“L.root”)->Get(“w”) ;
RooAbsPdf* model = w->pdf(“model”) ;
pdf->fitTo(data) ;
```

The need for fundamental statistical techniques

Frequentist statistics

$$\lambda_{\mu}(\vec{N}_{obs}) = \frac{L(\vec{N}\,|\,\mu)}{L(\vec{N}\,|\,\hat{\mu})}$$

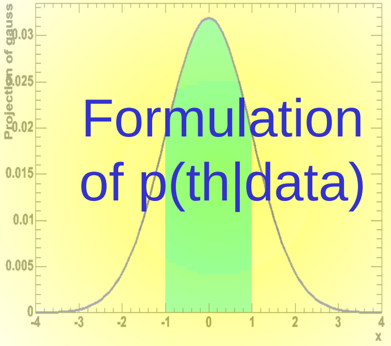


No assumptions
on normal distributions,
or asymptotic validity
for high statistics

Confidence interval
or p-value

Bayesian statistics

$$P(\mu) \propto L(x\,|\,\mu) \times p(\mu)$$

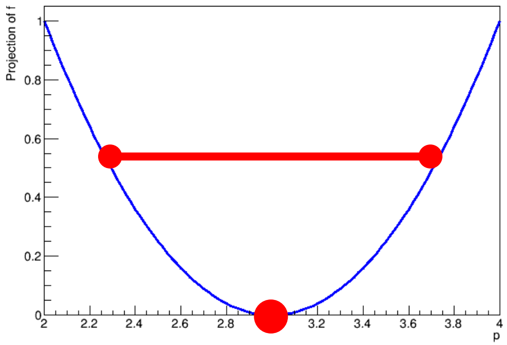


Formulation
of p(th|data)

Posterior on s
or Bayes factor

Maximum Likelihood

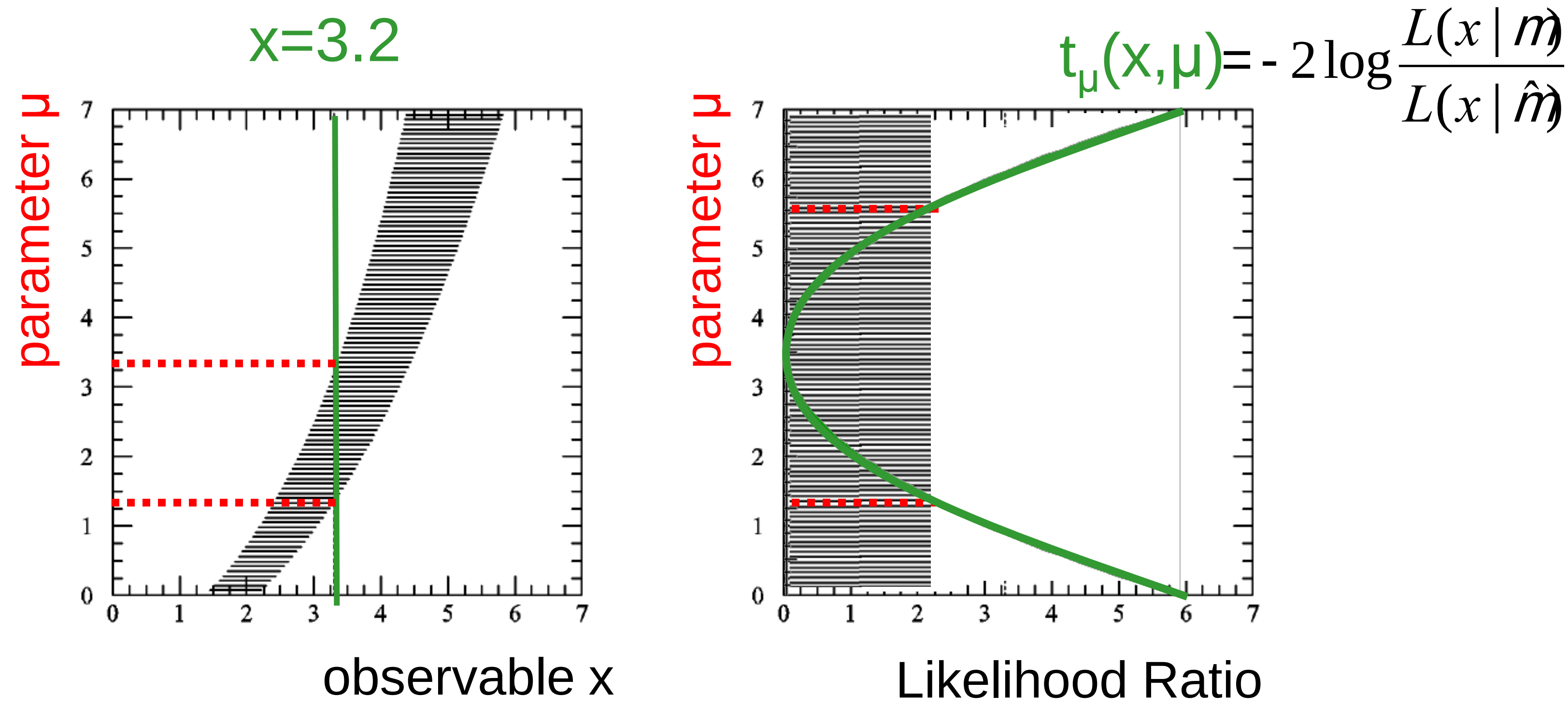
$$\left.\frac{d\ln L(\vec{p})}{d\vec{p}}\right|_{p_i=\hat{p}_i} = 0$$



s = x ± y

But fundamental techniques can be complicated to execute…

- Example of confidence interval calculation with Neyman construction
  - Need to construct ‘confidence belt’ using toy MC. Intersection observed data with belt defined interval in POI with guaranteed coverage



$$t_{\mu}(x,\mu) = -2\log\frac{L(x\,|\,m)}{L(x\,|\,\hat{m})}$$

- Expensive, complicated procedure, *but completely procedural once Likelihood and parameter of interest are fixed*
  → Can be wrapped in a tool that runs effectively ‘out-of-the-box’

Running RooStats interval calculations ‘out-of-the-box’

- Confidence intervals calculated with model
  - ‘Simple Fit’

```
RooAbsReal* nll = myModel->createNLL(data) ;
RooMinuit m(*nll) ;
m.migrad() ;
m.hesse() ;
```

  - Feldman Cousins (Frequentist Confidence Interval)

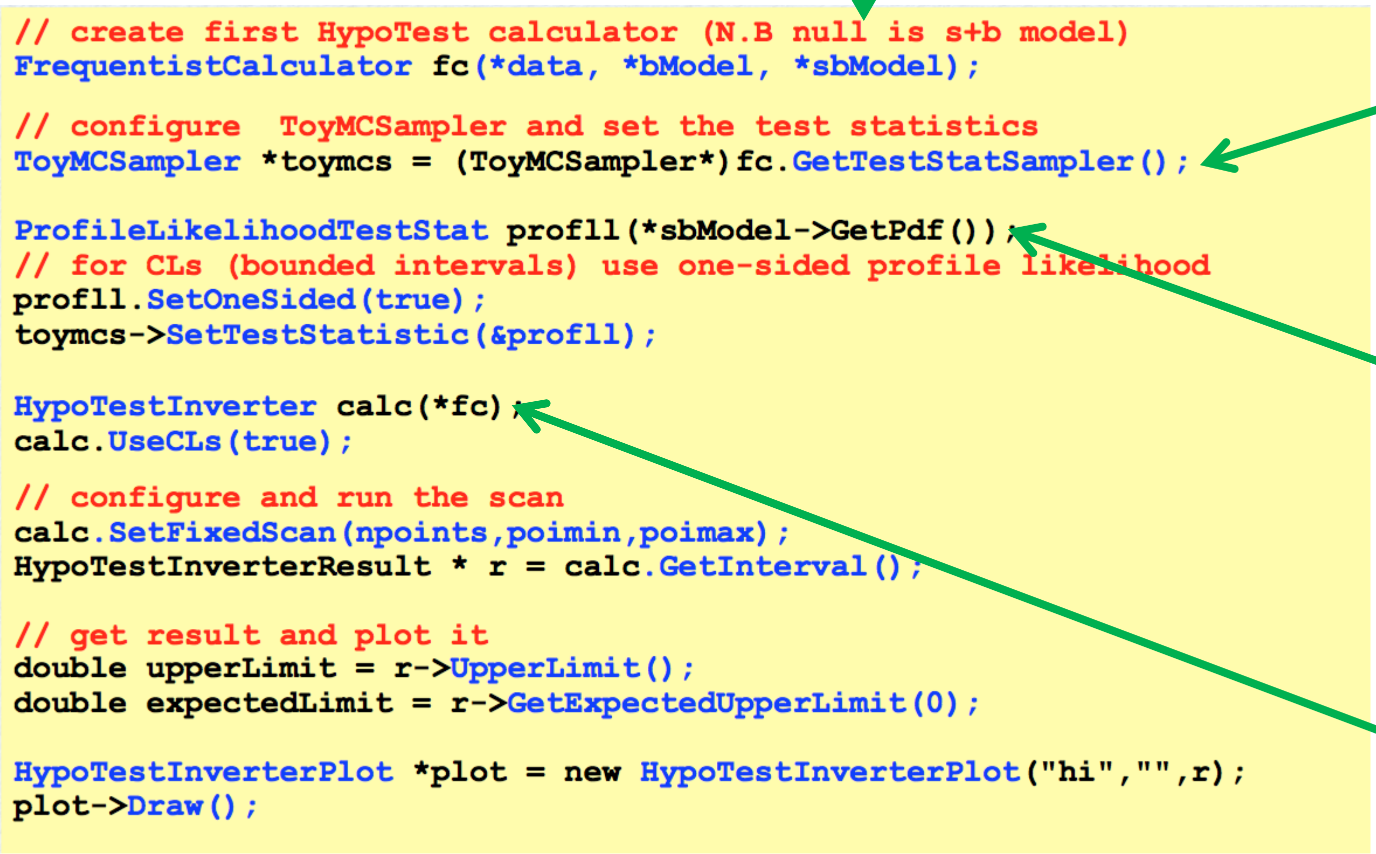
```
FeldmanCousins fc;
fc.SetPdf(myModel);
fc.SetData(data); fc.SetParameters(myPOU);
fc.UseAdaptiveSampling(true);
fc.FluctuateNumDataEntries(false);
fc.SetNBins(100); // number of points to test per parameter
fc.SetTestSize(.1);
ConfInterval* fcint = fc.GetInterval();
```

  - Bayesian (MCMC)

```
UniformProposal up;
MCMCCalculator mc;
mc.SetPdf(w::PC);
mc.SetData(data);  mc.SetParameters(s);
mc.SetProposalFunction(up);
mc.SetNumIters(100000); // steps in the chain
mc.SetTestSize(.1); // 90% CL
mc.SetNumBins(50); // used in posterior histogram
mc.SetNumBurnInSteps(40);
ConfInterval* mcmcint = mc.GetInterval();
```

But you can also look ‘in the box’ and build your own

Tool to calculate p-values for a given hypothesis $\int_{q_{\mu,obs}}^{\infty} f(q_\mu \mid \mu')dq_\mu$



```
// create first HypoTest calculator (N.B null is s+b model)
FrequentistCalculator fc(*data, *bModel, *sbModel);

// configure  ToyMCSampler and set the test statistics
ToyMCSampler *toymcs = (ToyMCSampler*)fc.GetTestStatSampler();

ProfileLikelihoodTestStat profll(*sbModel->GetPdf());
// for CLs (bounded intervals) use one-sided profile likelihood
profll.SetOneSided(true);
toymcs->SetTestStatistic(&profll);

HypoTestInverter calc(*fc);
calc.UseCLs(true);

// configure and run the scan
calc.SetFixedScan(npoints,poimin,poimax);
HypoTestInverterResult * r = calc.GetInterval();

// get result and plot it
double upperLimit = r->UpperLimit();
double expectedLimit = r->GetExpectedUpperLimit(0);

HypoTestInverterPlot *plot = new HypoTestInverterPlot("hi","",r);
plot->Draw();
```

$f(q_\mu \mid \mu')$

Tool to construct test statistic distribution

$q_\mu(\mu')$

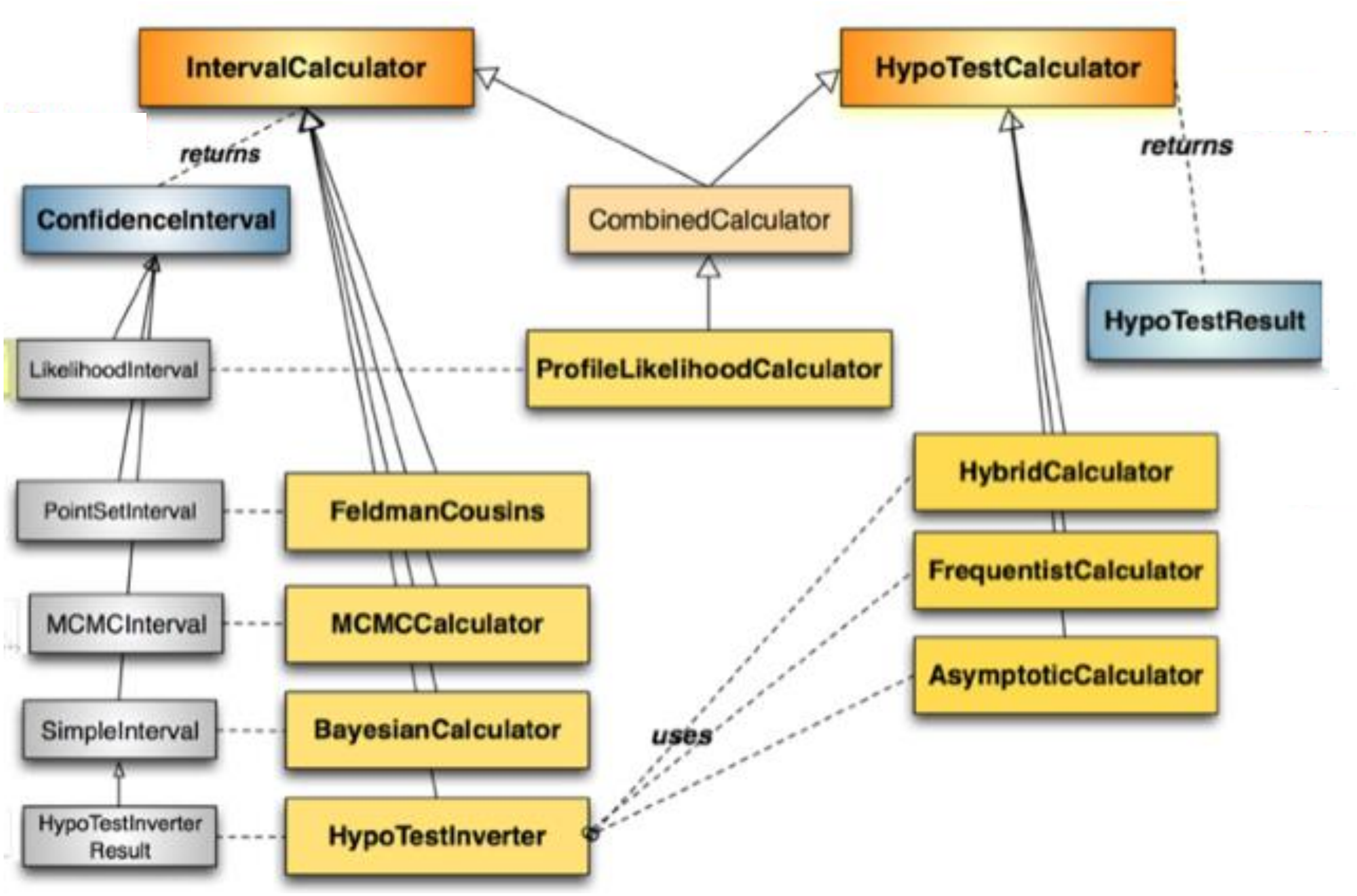
The test statistic

to be used for the calculation of p-values

Tool to construct interval from hypo test results

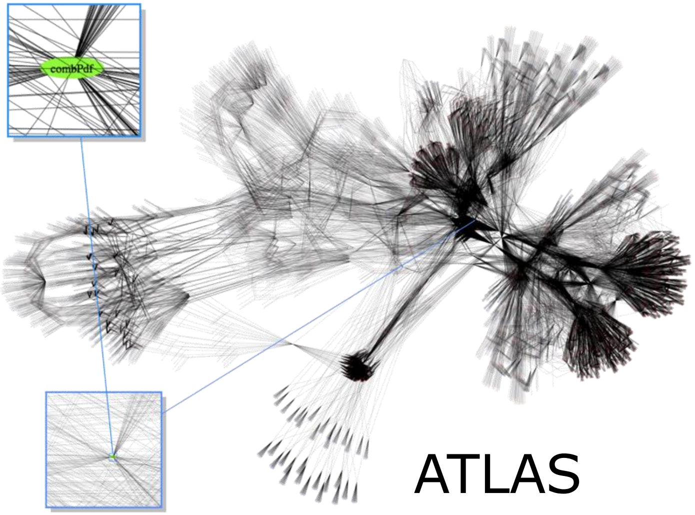
Offset advanced control over details of statistical procedure (use of CLS, choice of test statistic, boundaries…)

RooStats class structure

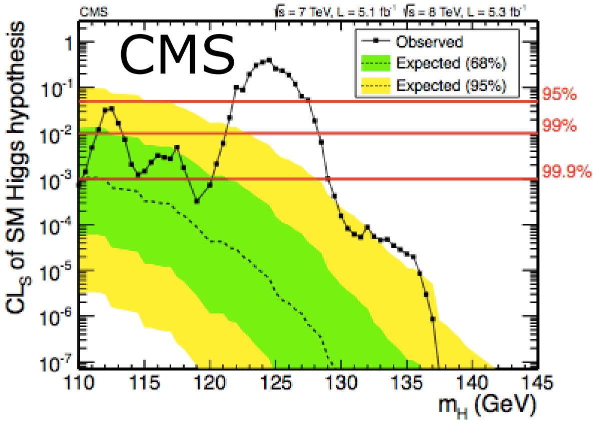


Summary

- RooFit and RooStats allow you to perform advanced statistical data analysis
  - LHC Higgs results a prominent example



- RooFit provides (almost) limitless model building facilities
  - Concept of persistable model workspace allows to separate model building and model interpretation
  - HistFactory package introduces structured model building for binned likelihood template models that are common in LHC analyses
- Concept of RooFit Workspace has completely restructured HEP analysis workflow with 'collaborative modeling'
- RooStats provide a wide set of statistical tests that can be performed on RooFit models
  - Bayesian, Frequentist and Likelihood-based test concepts



The future - physics

- Many more high-profile RooFit/RooStats full likelihood combinations in the works
  - Combination of ATLAS and CMS Higgs results
  - CMS/LHC combination of rare B-decays
- But many more combinations are easily imaginable & feasible
  - Combination across physics domains (e.g. SUSY and Higgs, or Exotics and Higgs) → reparametrization allows to constrain parameters of BSM physics models that have features in both domains (e.g. 2 Higgs Doublet Models)
  - Incorporation of more sophisticated models for detector performance measurements (now often simple Gaussians).

    Many ideas ongoing (e.g eigenvector diagonalization of calibration uncertainties across p_T bins → less parameters with correlated subsidiary measurement), modeling of correlated effects between systematic uncertainties (e.g. Jet energy scales and flavor tagging)

The future - computing

- Technical scaling and performance generally unproblematic
  - MINUIT has been shown to still work with 10.000 parameters, but do you really need so much detail?
  - Persistence works miraculously well, given complexity of serialization problem
  - Algorithmic optimization of likelihood calculations works well
  - Likelihood calculations trivially parallelizable. But more work can be done here (e.g. portability of calculations to GPUs, taking advantage of modern processor architectures for vectorization)
  - Bayesian algorithms still need more development and tuning
- But physicists are very good and pushing performance and scalability to the limits
  - Generally, one keep adding features and details until model becomes ‘too slow’
  - But if every Higgs channel reaches this point on its own, a channel combination is already ‘way too slow’ from the onset
  - Need to learn how to limit complexity → Prune irrelevant details from physics models, possibly a posteriori. Work in progress, some good ideas around
- Looking forward to LHC Run-2