CMS public statistical models

Aliya Nigamova (UHH)

Reinterpretation & OpenMAPP mini-workshop

LPSC Grenoble | 18/06/2024

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

CMS full statistical model release

- The first public CMS statistical model is out!
- Citable public release on new-cds.cern.ch

Release contents:

DOI *(Cite this version - v1.0)*

DOI 10.17181/c2948-e8875

- `Combine` datacards (statistical model)
- documentation (`Combine` version, instructions to reproduce main results)
- Systematic uncertainties description (html)

[DOI: 10.17181/2cp5k-ggn24]

Significance Calculation

To calculate the observed significance assuming the mass of the Higgs boson mH=125.5 GeV run,

```
combine 125.5/comb.txt --mass 125.5 -M Significance
```

The output will be:

```
 <<< Combine >>>
 <<< v9.2.1 >>>
>>> Random number generator seed is 123456
>>> Method used is Significance

 -- Significance --
Significance: 4.87557
Done in 1.76 min (cpu), 1.76 min (real)
```

We can look at partial combinations by using the `combineCards.py` script with the datacards for individual decay channel analyses. For the observation using only the Higgs to diphoton and Higgs to four-lepton datacards, run

```
combineCards.py 125.5/comb_hgg.txt 125.5/comb_hzz.txt > 125.5/comb_hgg_hzz.txt
combine 125.5/comb_hgg_hzz.txt --mass 125.5 -M Significance
```

Files

Files (3.0 MB)

Name	Size	Download all
cms-h-observation-public-v1.0.tar.gz md5:0114d88d1fa0bb9213e7e48908d413f4	3.0 MB	Download

CAT-23-001 is submitted to CSBS

- Paper submitted CSBS: arxiv:2404.06614
- Paper contents:
  - Detailed description and examples of statistical model constructed in combine
  - Description of common statistical analysis routines
  - Command-line examples to run the commonly used methods
- Accompanied by pre-compiled containerized Combine release v9.2.0: [docs]

```
$ docker run [--platform linux/amd64] --name combine -it gitlab-registry.cern.ch/cms
  ↪ -cloud/combine-standalone:v9.2.0
```

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH (CERN)

CERN-EP-2024-078
2024/04/11

CMS-CAT-23-001

The CMS statistical analysis and combination tool: COMBINE

The CMS Collaboration*

arXiv:2404.06614v1 [physics.data-an] 9 Apr 2024

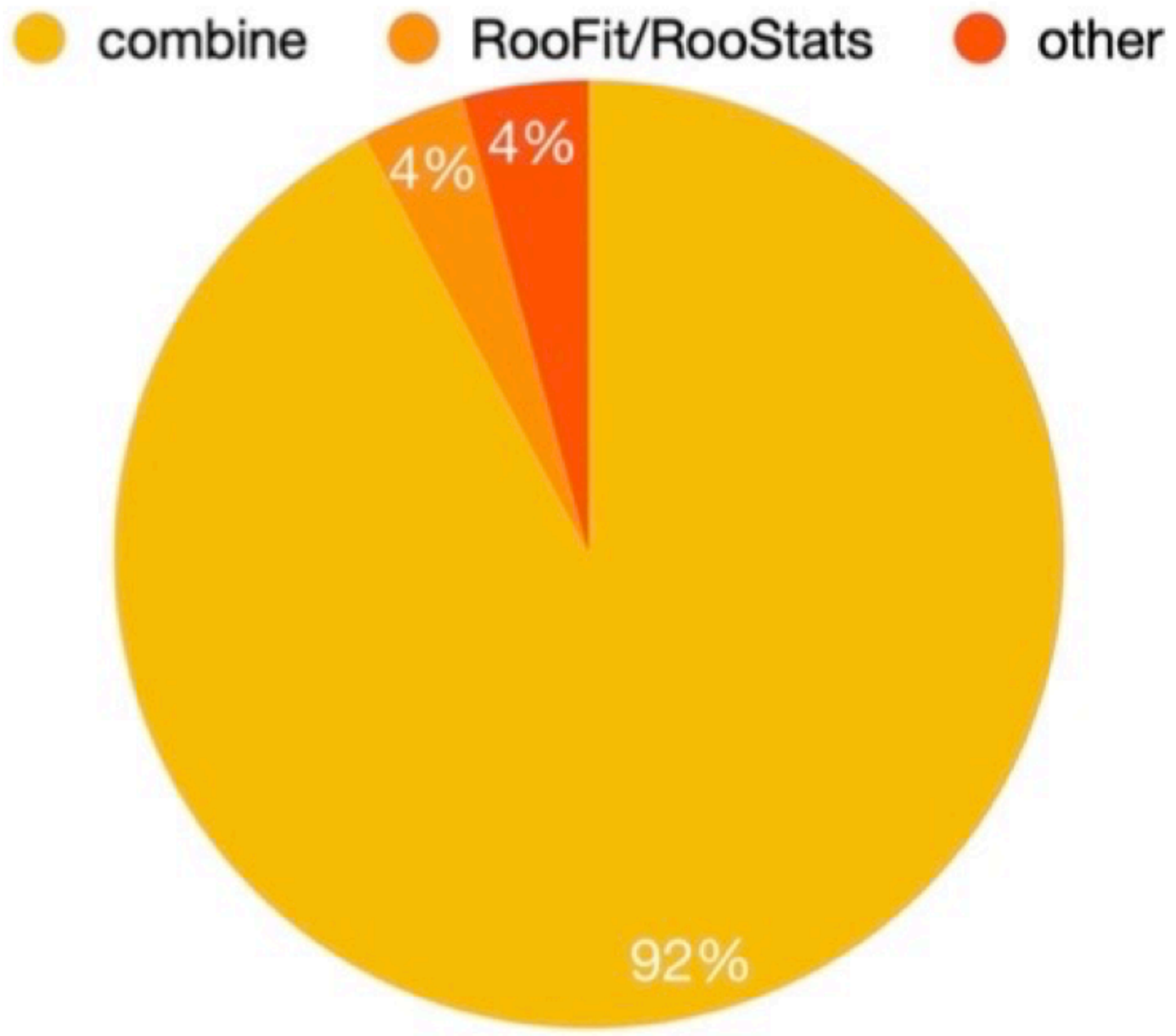
Abstract

This paper describes the COMBINE software package used for statistical analyses by the CMS Collaboration. The package, originally designed to perform searches for a Higgs boson and the combined analysis of those searches, has evolved to become the statistical analysis tool presently used in the majority of measurements and searches performed by the CMS Collaboration. It is not specific to the CMS experiment, and this paper is intended to serve as a reference for users outside of the CMS Collaboration, providing an outline of the most salient features and capabilities. Readers are provided with the possibility to run COMBINE and reproduce examples provided in this paper using a publicly available container image. Since the package is constantly evolving to meet the demands of ever-increasing data sets and analysis sophistication, this paper cannot cover all details of COMBINE. However, the online documentation referenced within this paper provides an up-to-date and complete user guide.

Submitted to Computing and Software for Big Science

Introduction to Combine

- Command-line interface to RooStats/RooFit methods, and even more:
  - Builds statistical (counting, parametric unbinned and binned, template-based) models
  - Powerful for combinations, scales well with model complexity
  - Provides workflow for statistical procedures recommended by CMS Statistics Committee
  - Provides tool extensive tool-set for validation
- Supported with extensive documentation and tutorials: https://cms-analysis.github.io/HiggsAnalysis-CombinedLimit/latest/



From Statistics Committee Questionnaires 2021-2022

Statistical models built with Combine

- Factorize into primary and auxiliary components

$$p(\vec{x}, \vec{\mu}, \vec{\nu}) \rightarrow \mathrm{p_k}(\vec{\mathrm{x}}_\mathrm{k}, \vec{\mu}, \vec{\nu})$$

$$p(\vec{x}, \vec{y}; \vec{\Phi}) = p(\vec{x}; \vec{\mu}, \vec{\nu}) \prod_k p_k(y_k; \nu_k)$$

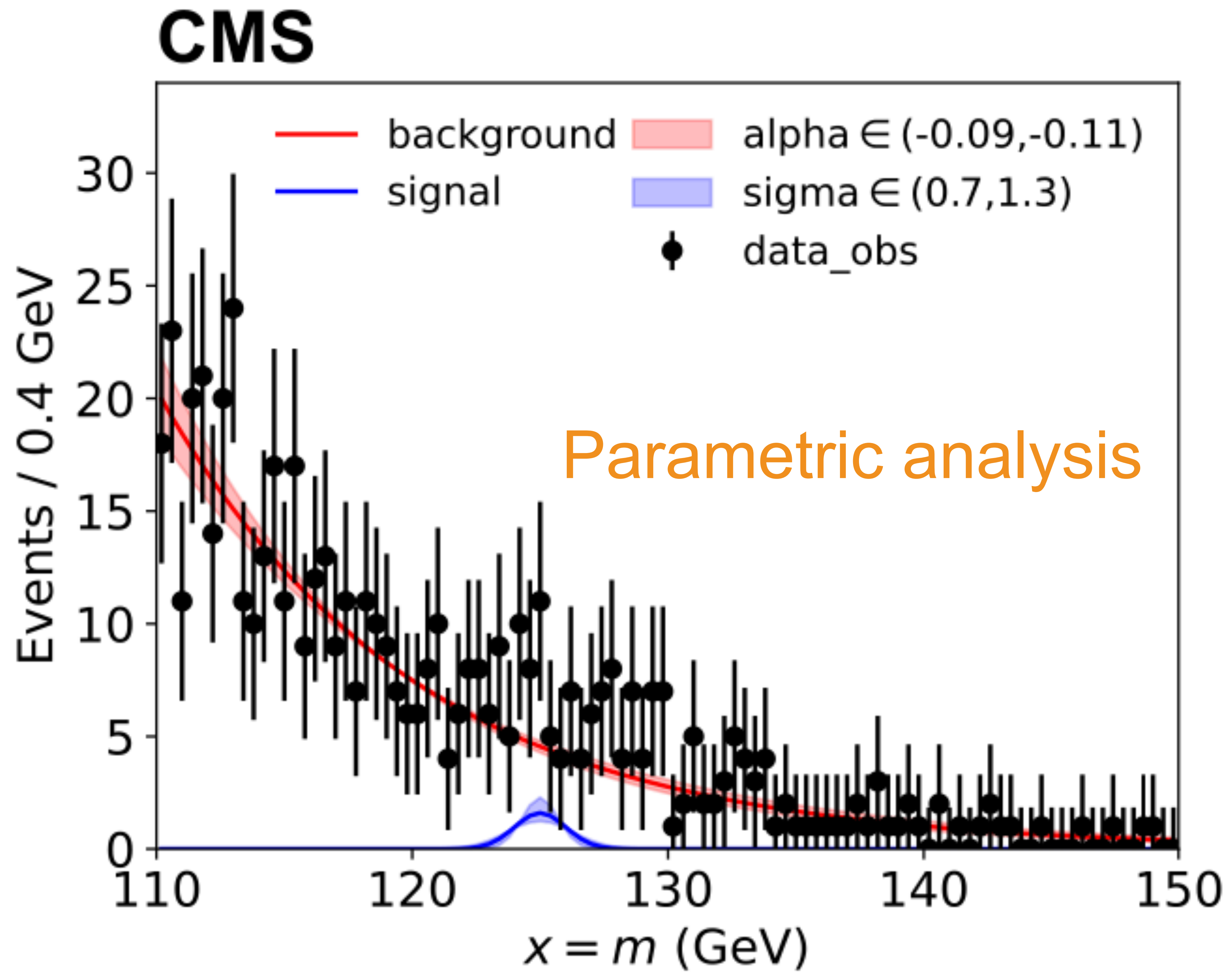
  - $\vec{\Phi} = (\vec{\mu}, \vec{\nu})$ - model parameters; $\vec{x}$ - primary observables; $\vec{y}$ - global observables
  - Likelihood function is constructed by evaluating $p(\vec{x}, \vec{y}; \Phi)$ on a dataset

$$\mathcal{L}(\vec{\Phi}) = \prod_d p(\vec{x}_d; \vec{\mu}, \vec{\nu}) \prod_k p_k(y_k; \nu_k).$$

  - Combine implements a custom class (RooFit based) to build the likelihood

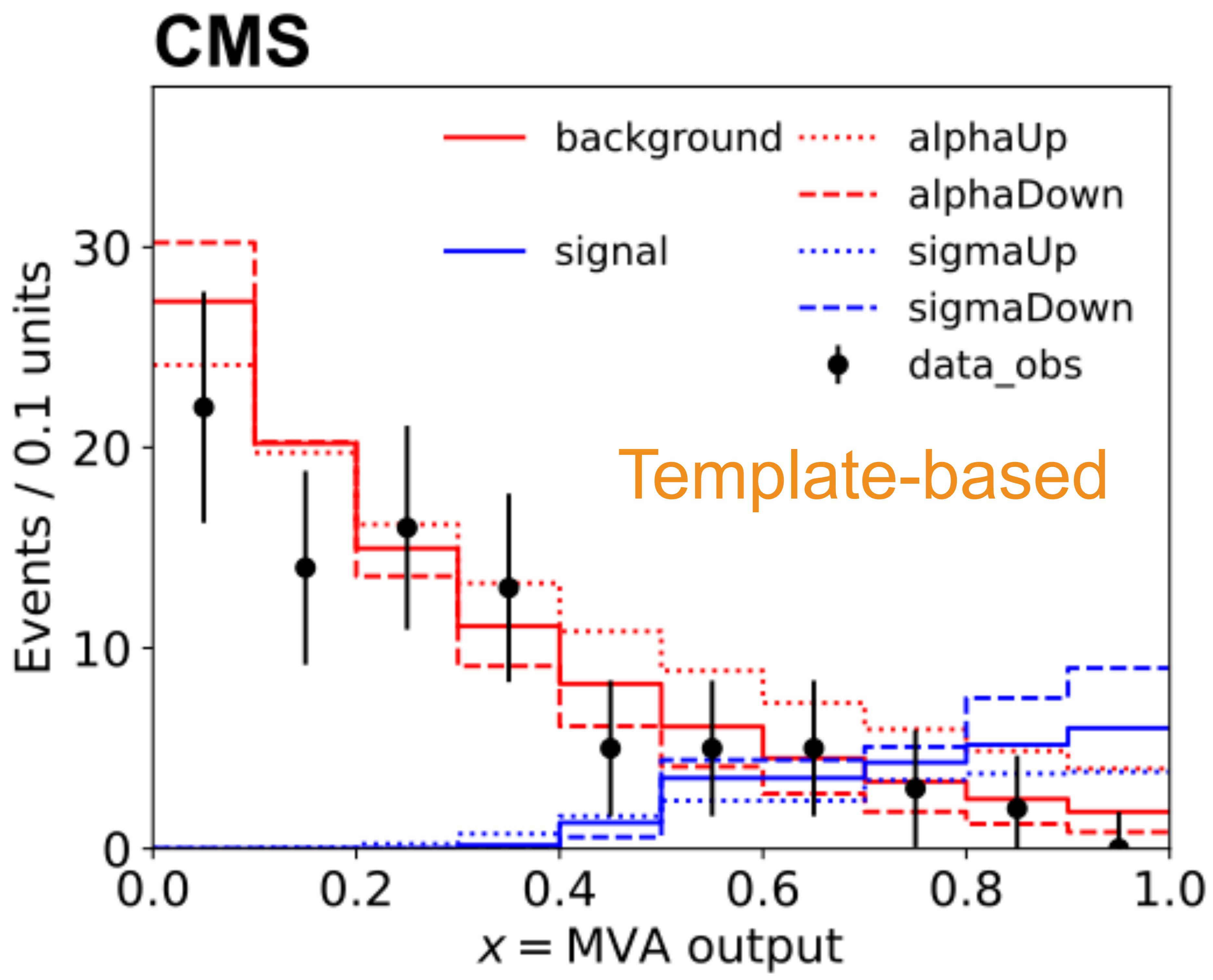
Supported statistical models

$$\mathcal{L}(\vec{\Phi}) = \prod_d p(\vec{x}_d; \vec{\mu}, \vec{\nu}) \prod_k p_k(y_k; \nu_k)$$



$$p(\vec{x}; \vec{\mu}, \vec{\nu}) = \sum_p \frac{\lambda_p(\vec{\mu}, \vec{\nu}) f_p(x; \vec{\mu}, \vec{\nu})}{\sum_p \lambda_p(\vec{\mu}, \vec{\nu})}$$

x - observable can be both binned and unbinned (`RooDataSet`)



$$p(x; \vec{\mu}, \vec{\nu}) = \prod_{b=1}^{N_B} \mathcal{P}\left(n_b; \lambda_b(\vec{\mu}, \vec{\nu})\right)$$

Counting analyses: $\mathcal{P}(n; \lambda) = \lambda^n \frac{e^{-\lambda}}{n!}$

Supported constraint terms

$$\mathcal{L}(\vec{\Phi}) = \prod_d p(\vec{x}_d; \vec{\mu}, \vec{\nu}) \prod_k p_k(y_k; \nu_k)$$

Uncertainty type	Directive	Inputs	Multiplicative factor, $f(\nu)$	$p(y;\nu)$	Default values
Log-normal	lnN	kappa	κ^ν	$\mathcal{N}(y;\nu,1)$	$\nu = y = 0$
Asymmetric log-normal	lnN	kappaDown, kappaUp	$(\kappa^{\text{Down}})^{-\nu}$ if $\nu < -0.5$, $(\kappa^{\text{Up}})^{\nu}$ if $\nu > 0.5$, $e^{\nu K(\kappa^{\text{Down}},\kappa^{\text{Up}},\nu)}$ otherwise.*	$\mathcal{N}(y;\nu,1)$	$\nu = y = 0$
Log-uniform	lnU	kappa	κ^ν	$\mathcal{U}(y, 1/\kappa, \kappa)$	$\nu = y = \frac{1}{2}(\kappa + 1/\kappa)$
Gamma	gmN	N, alpha†	ν/N	$\mathcal{P}(y;\nu)$	$\nu = N+1, y = N^\ddagger$

$^\star K(\kappa^{\text{Down}}, \kappa^{\text{Up}}, \nu) = \frac{1}{8}\left[4\ln(\kappa^{\text{Up}}/\kappa^{\text{Down}}) + \ln(\kappa^{\text{Up}}\kappa^{\text{Down}})(48\nu^5 - 40\nu^3 + 15\nu)\right]$ ensures that the multiplicative factor and its first and second derivatives are continuous for all values of ν, and reduces to a log-normal for $\kappa^{\text{Down}} = 1/\kappa^{\text{Up}}$.

†The rate value for the affected process must be equal to $N\alpha$.

‡The default value for the nuisance parameter is set to the mean of a gamma distribution with parameters $\kappa = N+1$, $\lambda = 1$, as defined in Ref. [20].

Combine datacard structure

➡ Combine datacard - human readable configuration file for statistical models

Example for simple counting experiment:

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin              ch1
observation      0
# ----------
bin              ch1    ch1    ch1
process          ppX    WW     tt
process          0      1      2
rate             1.47   0.64   0.22
# ----------
lumi     lnN     1.11   1.11    1.11
xs       lnN     1.20    -       -
nWW      gmN 4    -     0.16     -
```

$$\mathcal{L}(\vec{\Phi}) = \prod_d p(\vec{x}_d; \vec{\mu}, \vec{\nu}) \prod_k p_k(y_k; \nu_k)$$

Defines primary $\prod_d p_d(\vec{x}_d, \vec{\mu}, \vec{\nu})$ and auxiliary $\prod_k p_k(\nu_k, y_k)$ likelihood components

Physics Model

- Defines how parameters of interest scale the signal processes defined in the datacard

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin             ch1   ch1   ch1
process         ppX   WW    tt
process         0     1     2
rate            1.47  0.64  0.22
# ----------
lumi     lnN    1.11  1.11   1.11
xs       lnN    1.20   -      -
nWW      gmN 4   -    0.16    -
```

$$p(n,\vec{y};r,\vec{\nu}) = \frac{\lambda(r,\vec{\nu})^n}{n!} e^{-\lambda(r,\vec{\nu})} \frac{1}{2\pi} e^{-(\nu_{\mathtt{lumi}}-y_{\mathtt{lumi}})^2} e^{-(\nu_{\mathtt{xs}}-y_{\mathtt{xs}})^2} \frac{(\nu_{\mathtt{nWW}})^{y_{\mathtt{nWW}}}}{y_{\mathtt{nWW}}!} e^{-\nu_{\mathtt{nWW}}}$$

$$\lambda(r,\vec{\nu}) = r\, 1.47\, (1.11)^{\nu_{\mathtt{lumi}}} (1.2)^{\nu_{\mathtt{xs}}} + 0.22\, (1.11)^{\nu_{\mathtt{lumi}}} + 0.64\, (1.11)^{\nu_{\mathtt{lumi}}} \frac{\nu_{\mathtt{nWW}}}{0.64}.$$

The default physics model scales all processes marked as signal with linear parameter ***r*** - signal strength

Physics Model

- For example: multi-signal model to scale different signal processes with separate POIs

```
imax 2
jmax 2
kmax *
------------
shapes *  dijet  FAKE
shapes *  incl   FAKE
------------
bin            incl dijet
observation    166  8
------------
bin                 incl    incl    incl    dijet   dijet   dijet
process             ggH_hgg qqH_hgg bkg     ggH_hgg qqH_hgg bkg
process              -1     0       1       -1      0        1
rate                21      1.6     140     0.4     0.95     3.2
------------
QCDscale_ggH lnN    1.12    -       -       1.12    -        -
pdf_gg       lnN    1.08    -       -       1.08    -        -
pdf_qqbar    lnN     -      1.025   -       -       1.025    -
bg_incl      lnN     -      -       1.05    -       -        -
```

Implement your own model using the classes defined in python/PhysicsModel.py

```
def getHiggsSignalYieldScale(self,production,decay, energy):
  if production == "ggH": return ("r_ggH" if "ggH" in self.modes else 1)
  if production == "qqH": return ("r_qqH" if "qqH" in self.modes else 1)
  if production in [ "WH", "ZH", "VH" ]: return ("r_VH" if "VH" in self.modes else
  ↪ 1)
  if production == "ttH": return ("r_ttH" if "ttH" in self.modes else ("r_ggH" if
  ↪ self.ttHasggH else 1))
  raise RuntimeError, "Unknown production mode '%s'" % production
```

```
$ text2workspace.py <datacard.txt> -P HiggsAnalysis.CombinedLimit.<PythonFile>:<
  ↪ modelInstance> [--PO <options>]
```

Physics Model (make your own)

- Simple process scaling:
  - By default single POI assigned to all processes marked as signal in the datacards (`“r”`)
  - Multiple POIs can be assigned with [multiSignalModel] :

```
text2workspace.py -P HiggsAnalysis.CombinedLimit.PhysicsModel:multiSignalModel  --PO verbose
--PO 'map=.*/sig1:r_sig1[1,0,10]' --PO ‘map=.*/sig2:r_sig2[1,0,20]' datacard.txt -o ws.root
```

  - Mapping: `--PO 'map=bin/process:parameter'`

- More complicated: Interference, κ, EFT (signal processes are parametrised)
  - Use existing model [location]:

```
text2workspace.py datacard.txt  -P HiggsAnalysis.CombinedLimit.PythonFile:modelName
```

    e.g for Higgs couplings κ-model: `-P HiggsAnalysis.CombinedLimit.HiggsCouplings:c7` [link]

  - Create your own model in CombinedLimit/python directory; example given here on a model with interference: [tutorial], [code]

Constructing likelihood function with Combine

1. Define the statistical model with datacard + provide inputs for shape analysis (details in backup)

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin             ch1   ch1   ch1
process         ppX   WW    tt
process         0     1     2
rate            1.47  0.64  0.22
# ----------
lumi    lnN     1.11  1.11   1.11
xs      lnN     1.20   -      -
nWW     gmN 4    -    0.16    -
```

2. Apply physics model

\+

```
$ text2workspace.py <datacard.txt> -P HiggsAnalysis.CombinedLimit.<PythonFile>:<
  ↪ modelInstance> [--PO <options>]
```

=

RooFit Workspace

Constructing likelihood function with Combine

```
$ text2workspace.py <datacard.txt> -P HiggsAnalysis.CombinedLimit.<PythonFile>:<
  ↪ modelInstance> [--PO <options>]
```

➡ RooFit Workspace:

```
RooWorkspace(w) w contents

variables
---------
(lumi,lumi_In,nWW,nWW_In,n_obs_binch1,r,xs,xs_In)

p.d.f.s
-------
SimpleGaussianConstraint::lumi_Pdf[ x=lumi mean=lumi_In sigma=1 ] = 1
RooProdPdf::modelObs_b[ pdf_binch1_bonly ] = 0.360595
RooProdPdf::modelObs_s[ pdf_binch1 ] = 0.08291
RooProdPdf::model_b[ modelObs_b * nuisancePdf ] = 0.0632726
RooProdPdf::model_s[ modelObs_s * nuisancePdf ] = 0.014548
RooPoisson::nWW_Pdf[ x=nWW_In mean=nWW ] = 0.175467
RooProdPdf::nuisancePdf[ lumi_Pdf * xs_Pdf * nWW_Pdf ] = 0.175467
RooPoisson::pdf_binch1[ x=n_obs_binch1 mean=n_exp_binch1 ] = 0.08291
RooPoisson::pdf_binch1_bonly[ x=n_obs_binch1 mean=n_exp_binch1_bonly ] = 0.360595
SimpleGaussianConstraint::xs_Pdf[ x=xs mean=xs_In sigma=1 ] = 1
```

```
functions
--------
RooAddition::n_exp_binch1[ n_exp_binch1_proc_ppX + n_exp_binch1_proc_WW + n_exp_binch1_proc_tt ] = 2.49
RooAddition::n_exp_binch1_bonly[ n_exp_binch1_proc_WW + n_exp_binch1_proc_tt ] = 1.02
ProcessNormalization::n_exp_binch1_proc_WW[ thetaList=(lumi) asymmThetaList=() otherFactorList=(nWW) ] = 0.8
ProcessNormalization::n_exp_binch1_proc_ppX[ thetaList=(lumi,xs) asymmThetaList=() otherFactorList=(r) ] = 1.47
ProcessNormalization::n_exp_binch1_proc_tt[ thetaList=(lumi) asymmThetaList=() otherFactorList=() ] = 0.22

datasets
--------
RooDataSet::data_obs(n_obs_binch1)

named sets
----------
ModelConfig_GlobalObservables:(lumi_In,xs_In,nWW_In)
ModelConfig_NuisParams:(lumi,xs,nWW)
ModelConfig_Observables:(n_obs_binch1)
ModelConfig_POI:(r)
ModelConfig_bonly_GlobalObservables:(lumi_In,xs_In,nWW_In)
ModelConfig_bonly_NuisParams:(lumi,xs,nWW)
ModelConfig_bonly_Observables:(n_obs_binch1)
ModelConfig_bonly_POI:(r)
POI:(r)
globalObservables:(lumi_In,xs_In,nWW_In)
nuisances:(lumi,xs,nWW)
observables:(n_obs_binch1)
```

➡ Ready for statistical inference: estimate significance, CI, extract limits

Extracting results with Combine

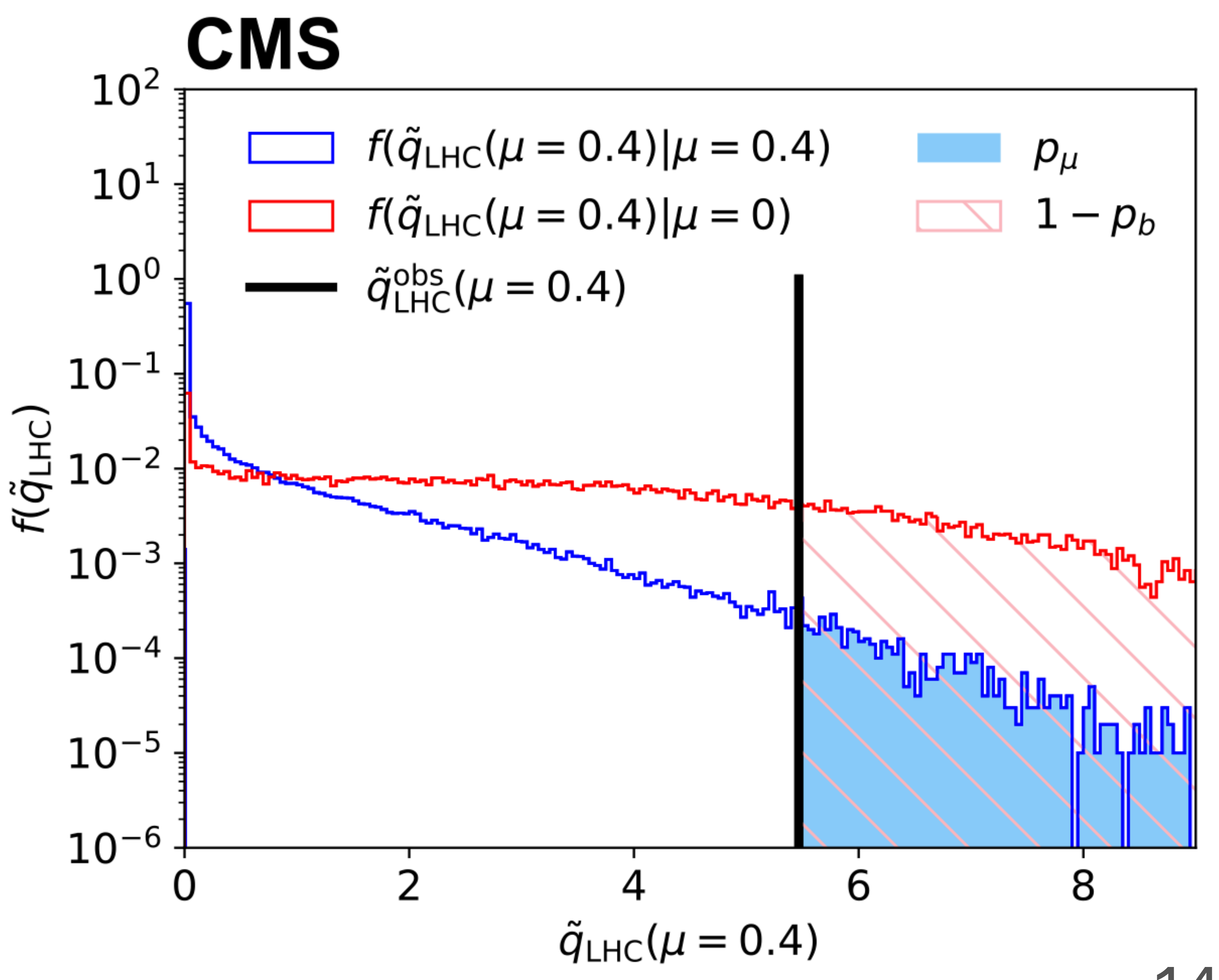
[Documentation]

```
$ combine <datacard.[txt|root]> -M <Method>
```

- AsymptoticLimits: limits calculated according to the asymptotic formulas in arxiv:1007.1727, valid for large event counts
- Significance: simple profile likelihood approximation for calculating significances.
- HybridNew: compute modified frequentist limits with toys, significance/p-values and confidence intervals with several options, `--LHCmode LHC-limits` is the recommended one

- **MultiDimFit**: perform maximum likelihood fit, with multiple POIs, estimate CI from likelihood scans

$$\tilde{q}_\mu = \begin{cases} -2\log\left(\frac{\mathcal{L}(\mu)}{\mathcal{L}(\mu=0)}\right) & \hat{\mu} < 0 \\ -2\log\left(\frac{\mathcal{L}(\mu)}{\mathcal{L}(\hat{\mu})}\right) & 0 < \hat{\mu} < \mu \\ 0 & \mu < \hat{\mu} \end{cases}$$



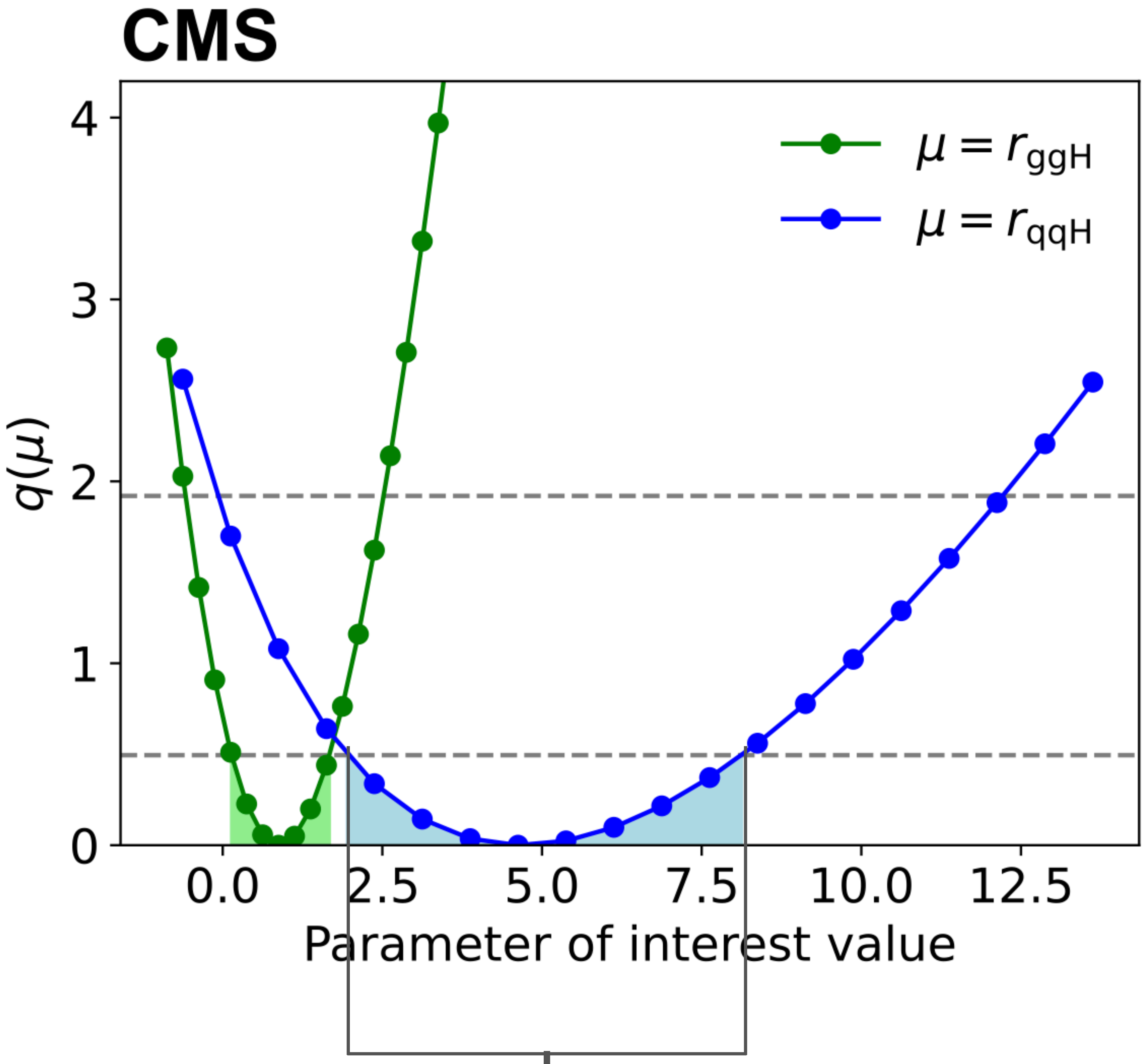
Extracting results with Combine

[Documentation]

```
$ combine <datacard.[txt|root]> -M <Method>
```

- AsymptoticLimits: limits calculated according to the asymptotic formulas in arxiv:1007.1727, valid for large event counts
- Significance: simple profile likelihood approximation for calculating significances.
- HybridNew: compute modified frequentist limits with toys, significance/p-values and confidence intervals with several options, `--LHCmode LHC-limits` is the recommended one

- **MultiDimFit**: perform maximum likelihood fit, with multiple POIs, estimate CI from likelihood scans



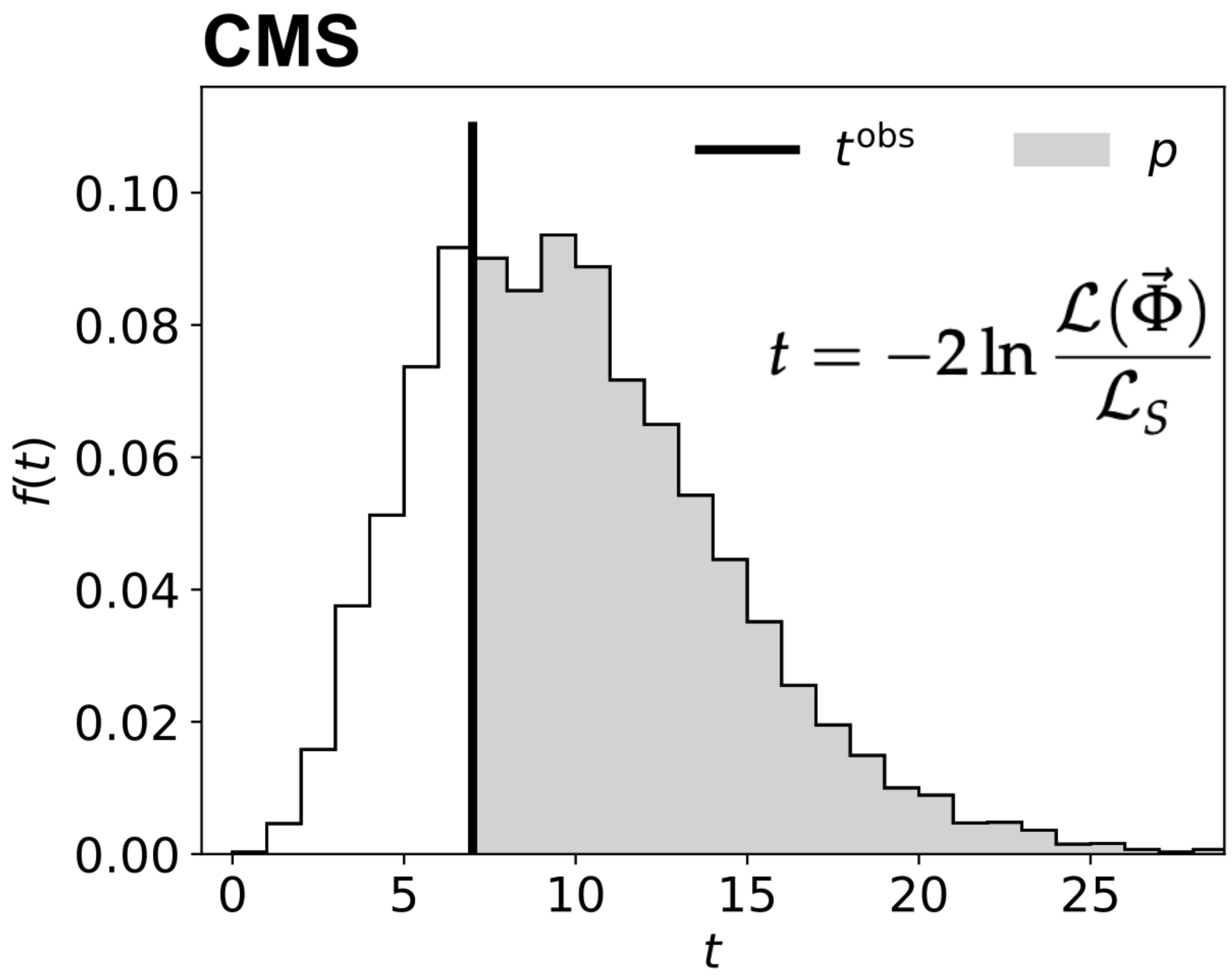
$$\Delta^{\pm}\mu = |\mu^{\pm} - \hat{\mu}|$$

Diagnostics tools

[Documentation]

```
$ combine <datacard.[txt|root]> -M <Method>
```

- **GoodnessOfFit**: perform a goodness of fit test for models including shape information using several GOF estimators (AD, KS, **saturated - recommended by SC)**
- Impacts: evaluate the shift in POI from $\pm\sigma_{\text{postfit}}$ variation for each NP
- ChannelCompatibiltyCheck: check how consistent are the individual channels of a combination are
- GenerateOnly: generate random or asimov toy datasets for use as input to other methods

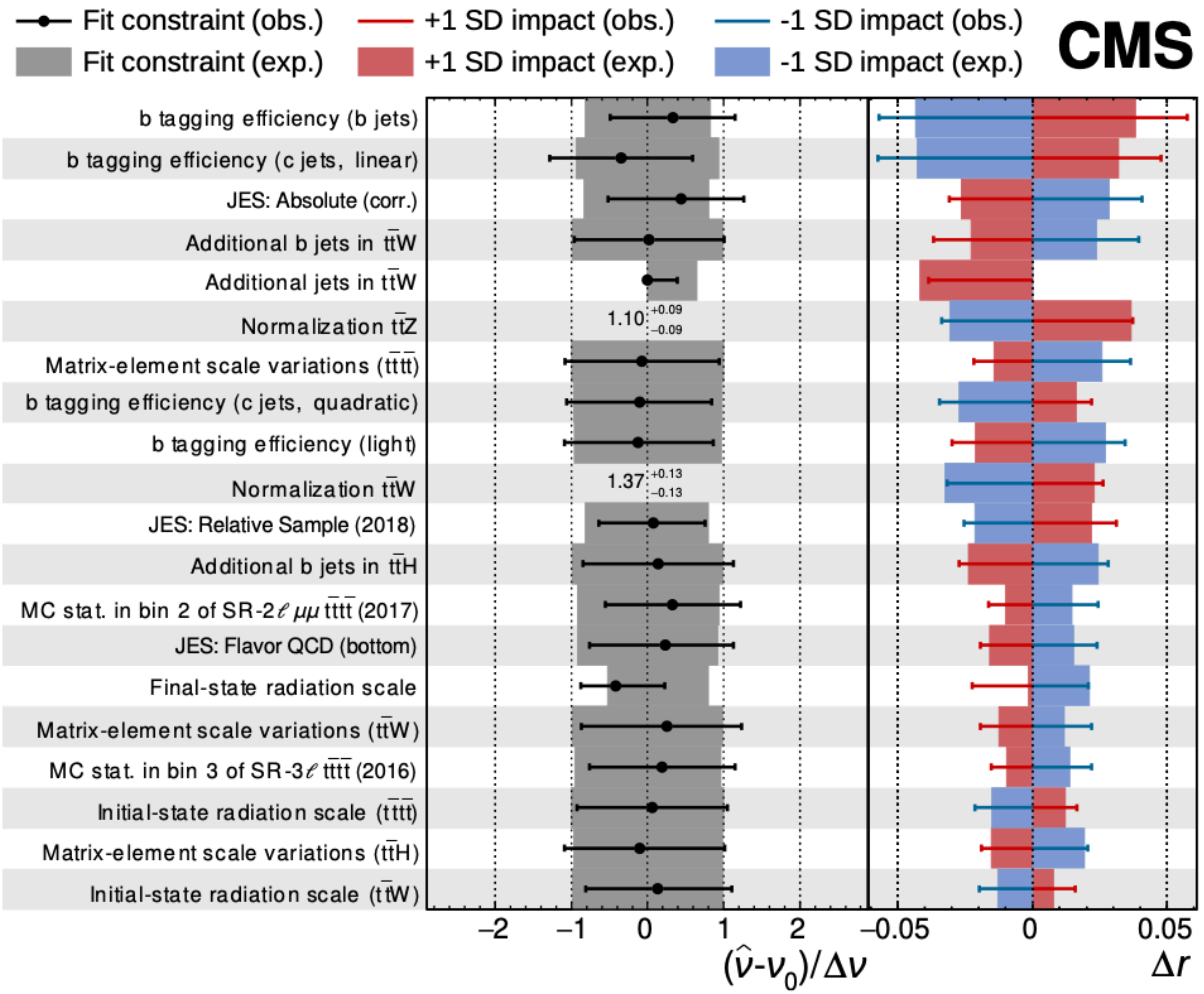


MLE and diagnostics tools

[Documentation]

```
$ combine <datacard.[txt|root]> -M <Method>
```

- GoodnessOfFit: perform a goodness of fit test for models including shape information using several GOF estimators (AD, KS, saturated - recommended by SC)
- **Impacts**: evaluate the shift in POI from $\pm\sigma_{\text{postfit}}$ variation for each nuisance parameter
- ChannelCompatibiltyCheck: check how consistent are the individual channels of a combination are
- GenerateOnly: generate random or asimov toy datasets for use as input to other methods



Combine documentation

[Combine documentation]

Combine v10.0.X Search

Introduction

These pages document the RooStats / RooFit - based software tool used for statistical analysis within the CMS experiment - Combine. Note that while this tool was originally developed in the Higgs Physics Analysis Group (PAG), its usage is now widespread within CMS.

Combine provides a command-line interface to many different statistical techniques, available inside RooFit/RooStats, that are used widely inside CMS.

The package exists on GitHub under https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit

For more information about Git, GitHub and its usage in CMS, see http://cms-sw.github.io/cmssw/faq.html

The code can be checked out from GitHub and compiled on top of a CMSSW release that includes a recent RooFit/RooStats, or via standalone compilation without CMSSW dependencies. See the

- Installation (w/ CMSSW, standalone, using LCG, conda, latest release) [link]
- Setting up the analysis (counting, template based, parametric) [link]
- Running Combine [link]
- Underlying statistics
  - Likelihood definition: [link]
  - How the fits are performed (profiling, marginalization, confidence intervals): [link]
  - Statistical tests (test statistic, GOF): [link]
- Tutorials: main features (template-based model), parametric, unfolding

First CMS public statistical model

Back to the released model [DOI: 10.17181/2cp5k-ggn24]

- Corresponds to the CMS Higgs Run 1 combination of 5 main Higgs channels [10.1016/j.physletb.2012.08.021]
- Accompanied by the command-line instructions to reproduce the main results:

1. Combine channels

```
combineCards.py 125.5/comb_hgg.txt 125.5/comb_hzz.txt > 125.5/comb_hgg_hzz.txt
combine 125.5/comb_hgg_hzz.txt --mass 125.5 -M Significance
```

2. Calculate the significance

```
combine 125.5/comb.txt --mass 125.5 -M Significance
```

The output will be:

```
 <<< Combine >>>
 <<< v9.2.1 >>>
>>> Random number generator seed is 123456
>>> Method used is Significance

 -- Significance --
Significance: 4.87557
Done in 1.76 min (cpu), 1.76 min (real)
```

3. Measure the signal strength

We can measure the signal strength of the Higgs boson (`r`) and its uncertainty using `Combine`,

```
combine 125.5/comb.txt -m 125.5 -M MultiDimFit --algo singles --setParameterRanges r=0.2,1.5
```

and the output will be:

```
 <<< Combine >>>
 <<< v9.2.1 >>>
>>> Random number generator seed is 123456
>>> Method used is MultiDimFit
Set Range of Parameter r To : (0.2,1.5)
Doing initial fit:

 --- MultiDimFit ---
best fit parameter values and profile-likelihood uncertainties:
   r :    +0.785   -0.204/+0.218 (68%)
Done in 1.86 min (cpu), 1.86 min (real)
```

First CMS public statistical model

Back to the released model [DOI: 10.17181/2cp5k-ggn24]

- Corresponds to the CMS Higgs Run 1 combination of 5 main Higgs channels [10.1016/j.physletb.2012.08.021]
- Accompanied by the command-line instructions to reproduce the main results:

Apply κ model to measure for interpretation [HiggsCouplings_ICHEP12:cVcF]

```
text2workspace.py -P HiggsAnalysis.CombinedLimit.HiggsCouplings_ICHEP12:cVcF 125.5/comb.txt -m 125.5 -o comb_kVkF.root
```

Note that since the discovery, the Physics Model `HiggsCouplings_ICHEP12:cVcF` has evolved to use make use of higher precision theoretical calculations, but for the discovery analysis, this is the model that was used.

We can measure these parameters and their uncertainties with,

```
combine comb_kVkF.root -m 125.5 -M MultiDimFit --algo singles
```

```
 --- MultiDimFit ---
best fit parameter values and profile-likelihood uncertainties:
   CV :    +0.946   -0.120/+0.113 (68%)
   CF :    +0.497   -0.170/+0.203 (68%)
Done in 3.09 min (cpu), 3.09 min (real)
```

More public statistical models are coming, at some point it should follow most of the CMS publications

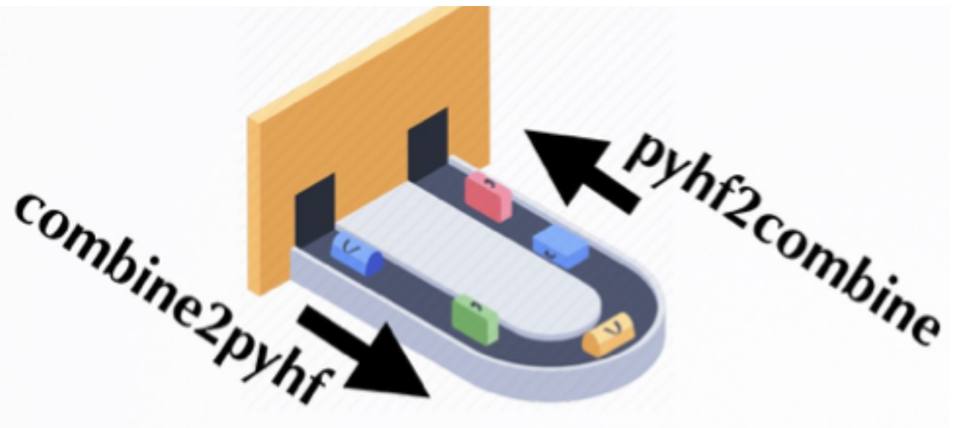
How Combine statistical models can be used

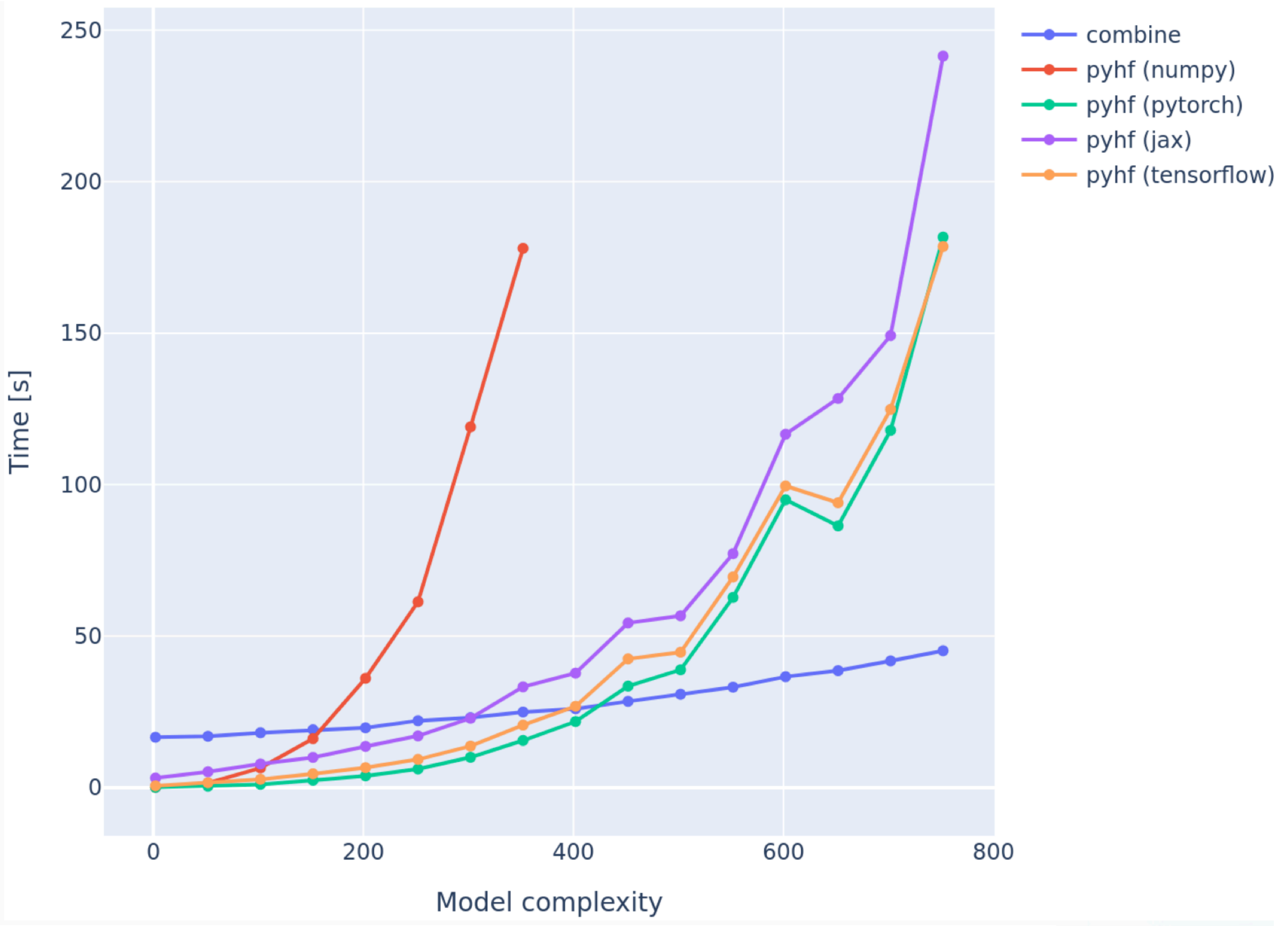
- ✓ Combinations
  - Trivial, but one should be careful with the correlation scheme for nuisance parameters → CMS will be following common conventions for NP naming, and making sure that they are documented in every statistical model release (automating this process)
- ✓ (Re)Interpretations
  - Redefine parameters of interest $\overrightarrow{\Phi} = (\mu, \vec{\nu}) \rightarrow (\overrightarrow{\mu'}, \vec{\nu})$, apply different Physics Model, implement your own (discussed earlier) → support and examples are available
- ◉ (?) Replace PDFs, e.g. update the signal shape with a new prediction, update signal uncertainties
  - Technically possible, but can be tricky estimate because most analyses use complex observables. Workflow management tools might help here (REANA, snakeMake, Luigi) → WIP in CMS

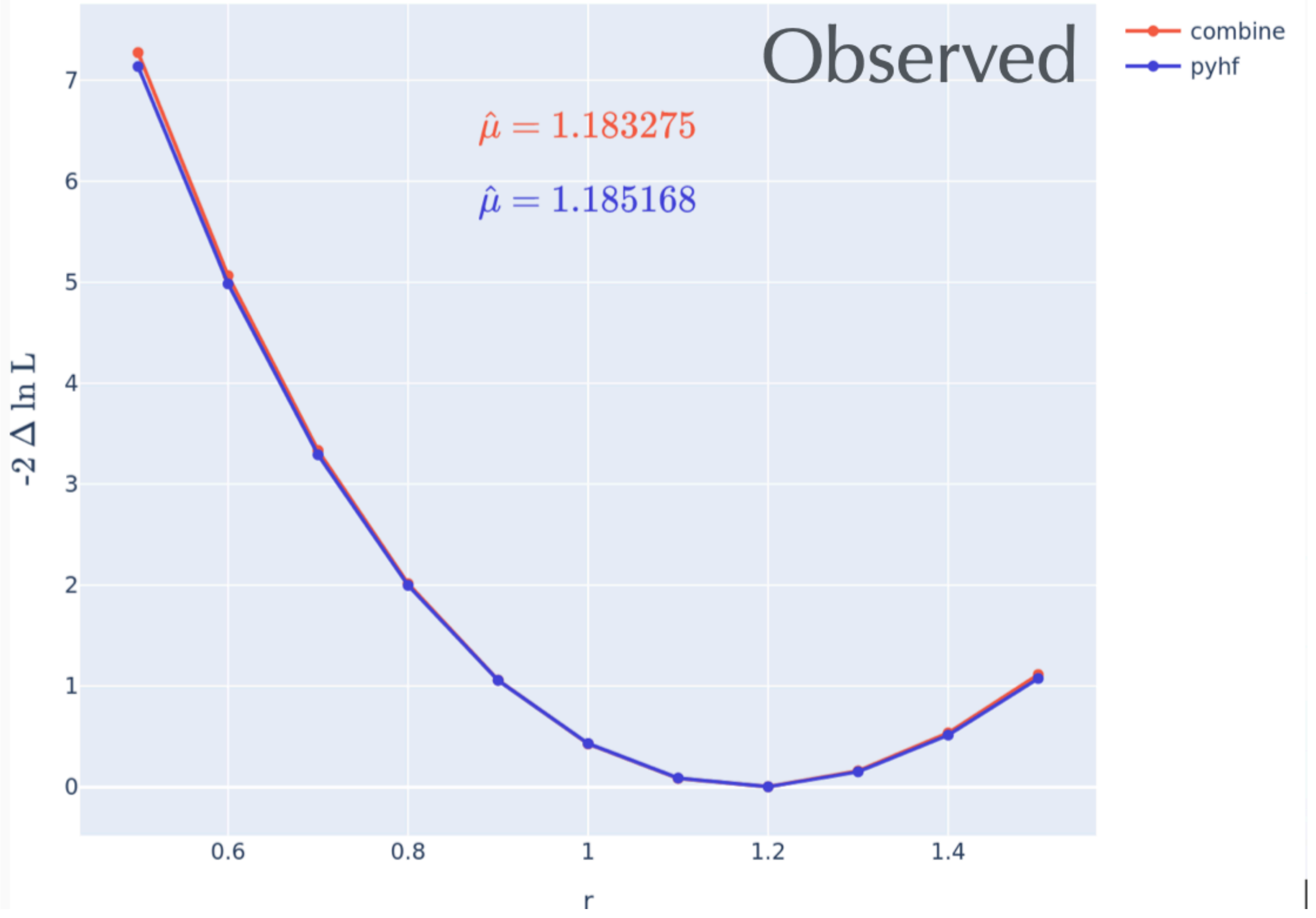
Next steps: Interaction with other statistical frameworks

[K. Skovpen]

- Should be able support interoperability with other statistical frameworks (e.g. pyHF)
- Combine <—> pyHF conversion tool was developed and extensively validated in the context of ATLAS + CMS tttt EFT combination [github]
- In contact with HS3 developers [link], and in principle we should be able to apply follow HS3 standards
  - Might be complicated because of the custom classes in Combine, but they could be included in future ROOT versions (tbd)







Summary

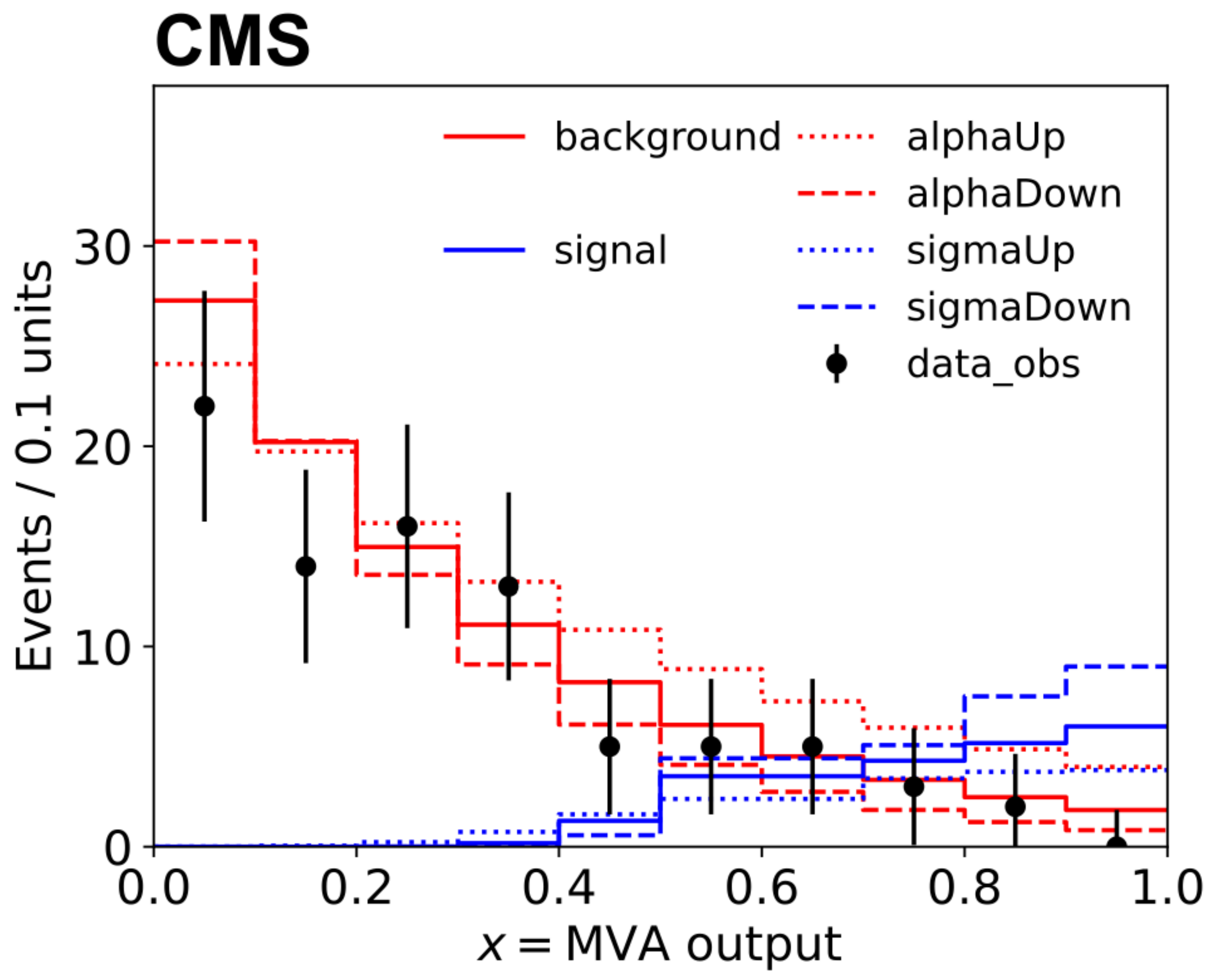
- CMS released the first statistical model implemented in Combine and in process of releasing more statistical models
- Combine covers most of the statistical inference analyses:
  - Parameter estimation, BSM searches, maximum likelihood unfolding
- Provides an extensive toolset for statistical inference
- Effort to make Combine usable outside of CMS: standalone containers, improving documentation, making sure that it is compatible with the latest ROOT version
- In future planning to improve compatibility with other formats compatible with pyHF and HS3

Thank you!

Backup

Template analysis datacard structure

```
imax 1
jmax 1
kmax 4
# ---------------
shapes * * template-analysis-datacard-input.root $PROCESS
  ↪ $PROCESS_$SYSTEMATIC
# ---------------
bin          ch1
observation 85
# ------------------------------
bin                ch1         ch1
process            signal      background
process            0           1
rate               24          100
# --------------------------------
lumi      lnN    1.1        1.0
bgnorm    lnN    -          1.3
alpha   shape    -           1    # uncertainty in the background template.
sigma   shape    0.5         -    # uncertainty in the signal template.
```



$$\mathcal{L}(\vec{\Phi}) = \prod_{d} p(\vec{x}_d; \vec{\mu}, \vec{\nu}) \prod_{k} p_k(y_k; \nu_k).$$

```
shapes <process> <channel> <file> <histogram> [<
  ↪ histogram_systematic_uncertainty_variation>]
```

Standardizing datacards: documentation

Datacard:

```
CMS_zz4l_bkgMELA     param 0 1 [-3,3]
CMS_zz4l_mean_e_sig  param 0 0.004
CMS_zz4l_mean_m_sig  param 0 0.004
CMS_zz4l_n_sig_1_7   param 0 0.01
CMS_zz4l_n_sig_1_8   param 0 0.01
CMS_zz4l_n_sig_2_7   param 0 0.05
CMS_zz4l_n_sig_2_8   param 0 0.05
CMS_zz4l_n_sig_3_7   param 0 0.05
CMS_zz4l_n_sig_3_8   param 0 0.05
CMS_zz4l_sigma_e_sig param 0 0.2
CMS_zz4l_sigma_m_sig param 0 0.2
BR_hzz               lnN    1.02            1.02
CMS_eff_e            lnN    1.111           1.111
CMS_eff_m            lnN    -               -
CMS_hzz2e2mu_Zjets   lnN    -               -
CMS_hzz4e_Zjets      lnN    -               -
CMS_hzz4mu_Zjets     lnN    -               -
QCDscale_VH          lnN    1.015           -
QCDscale_VV          lnN    -               -
```

	class	description
BR_hzz	branching_ratios	uncertainty on the branching ratio of higgs to Z bosons
CMS_zz4l_bkgMELA	custom	shape uncertainties jet energy scale and resolution modifying the background shape
CMS_hzz4mu_Zjets	custom	uncertainty on irreducible Z+jets background split in different channels
CMS_hzz4e_Zjets	custom	uncertainty on irreducible Z+jets background split in different channels
CMS_hzz2e2mu_Zjets	custom	uncertainty on irreducible Z+jets background split in different channels
pdf_hzz4l_accept	custom	acceptance uncertinty derived in h->4l analysis
CMS_zz4l_sigma_m_sig	custom	shape uncertainty on signal width parameter split for e and mu channels
CMS_zz4l_n_sig_3_8	custom	shape uncertainty on signal normalisation parameter in different categories
CMS_zz4l_sigma_e_sig	custom	shape uncertainty on signal width parameter split for e and mu channels
CMS_zz4l_n_sig_2_8	custom	shape uncertainty on signal normalisation parameter in different categories
CMS_zz4l_n_sig_2_7	custom	shape uncertainty on signal normalisation parameter in different categories
CMS_zz4l_n_sig_1_8	custom	shape uncertainty on signal normalisation parameter in different categories
CMS_zz4l_n_sig_1_7	custom	shape uncertainty on signal normalisation parameter in different categories
CMS_zz4l_mean_m_sig	custom	shape uncertainty on signal mean parameter split for e and mu channels
CMS_zz4l_mean_e_sig	custom	shape uncertainty on signal mean parameter split for e and mu channels
CMS_zz4l_n_sig_3_7	custom	shape uncertainty on signal normalisation parameter in different categories
CMS_eff_e	electron_efficiency	efficiency uncertainty for electrons for all years.