

pyhf: pure-Python implementation of HistFactory with tensors and automatic differentiation

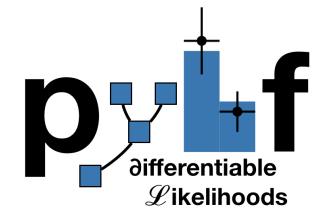
Matthew Feickert

(University of Illinois at Urbana-Champaign)

matthew.feickert@cern.ch

Tools for High Energy Physics and Cosmology 2020 Workshop

November 3rd, 2020



pyhf team



Lukas Heinrich

CERN



Matthew Feickert

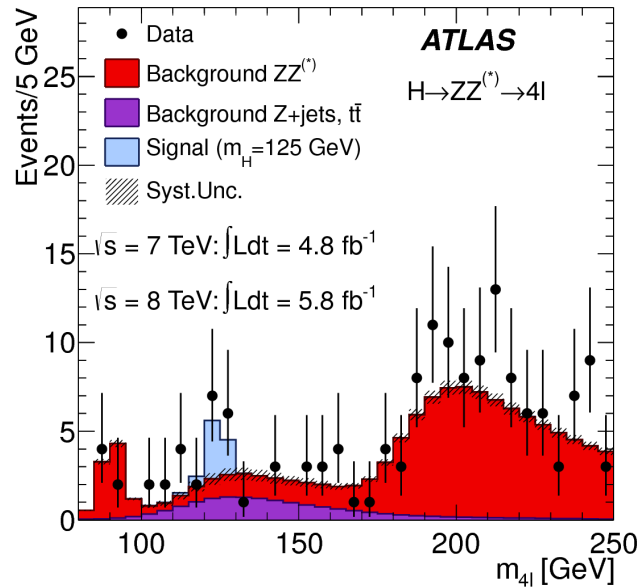
Illinois



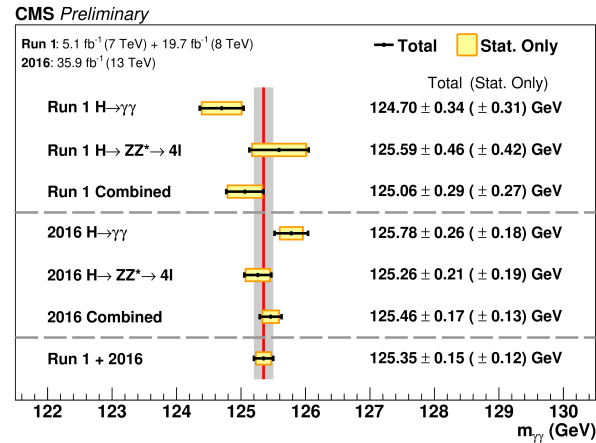
Giordon Stark

UCSC SCIPP

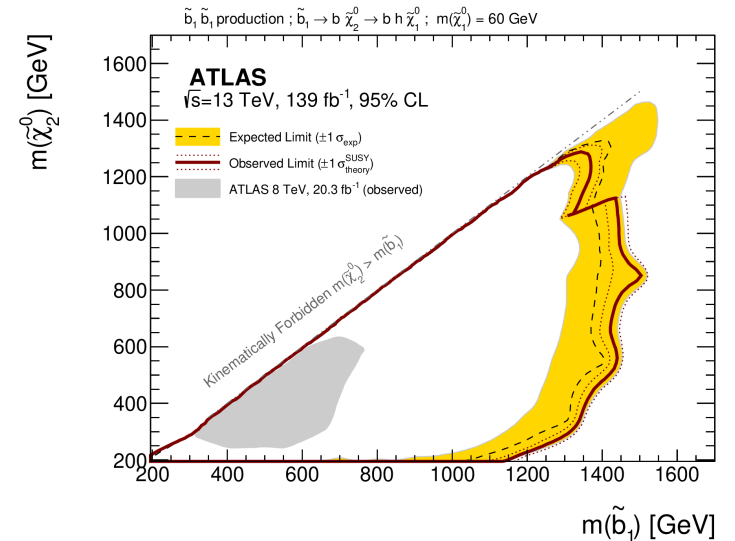
Goals of physics analysis at the LHC



Search for new physics



Make precision measurements

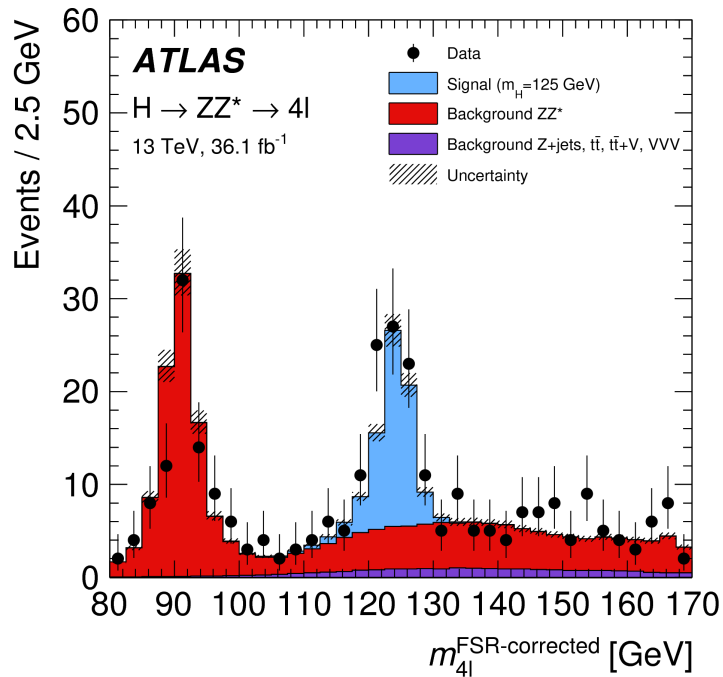


Provide constraints on models through setting best limits

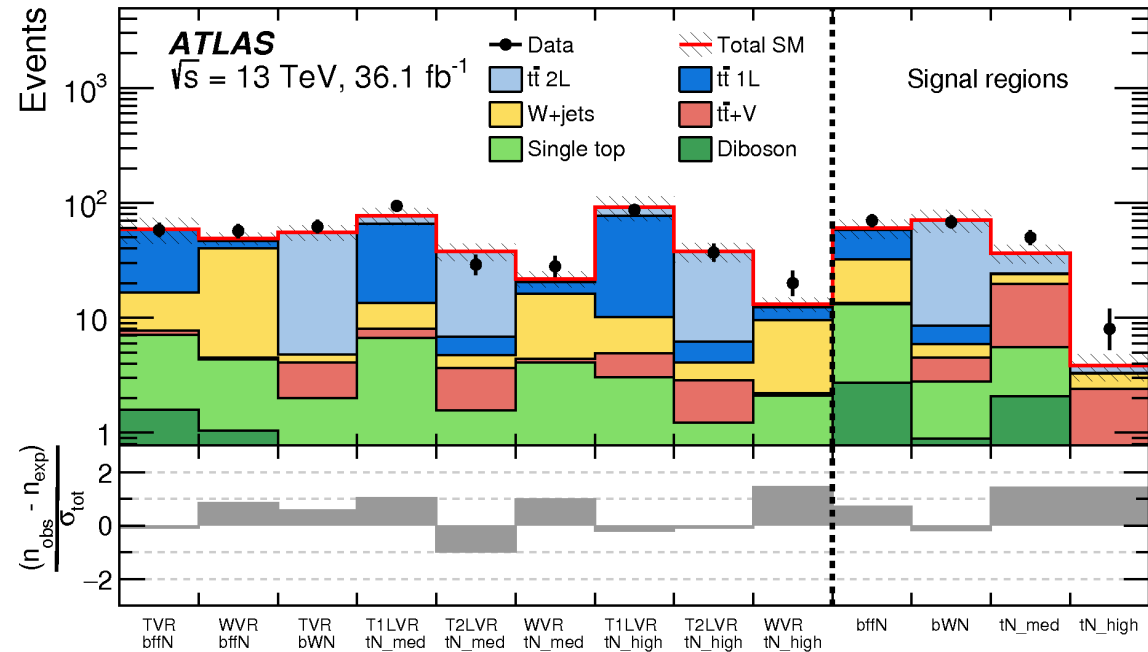
- All require **building statistical models** and **fitting models** to data to perform statistical inference
- Model complexity can be huge for complicated searches
- **Problem:** Time to fit can be **many hours**
- **Goal:** Empower analysts with fast fits and expressive models

HistFactory Model

- A flexible probability density function (p.d.f.) template to build statistical models in high energy physics
- Developed in 2011 during work that lead to the Higgs discovery [CERN-OPEN-2012-016]
- Widely used by the HEP community for **measurements of known physics** (Standard Model) and **searches for new physics** (beyond the Standard Model)



Standard Model



Beyond the Standard Model

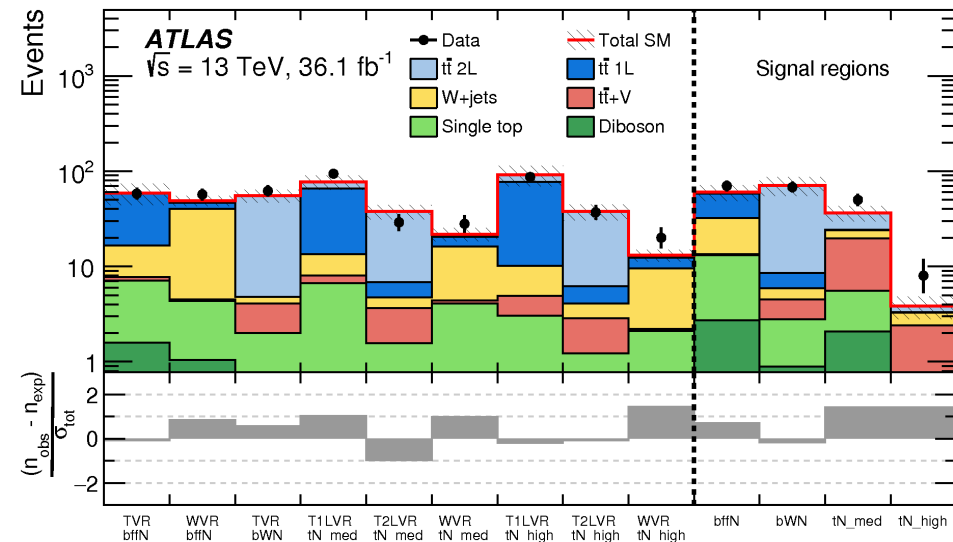
HistFactory Template

$$f(\text{data}|\text{parameters}) = f(\vec{n}, \vec{a}|\vec{\eta}, \vec{\chi}) = \prod_{c \in \text{channels}} \prod_{b \in \text{bins}_c} \text{Pois}(n_{cb}|\nu_{cb}(\vec{\eta}, \vec{\chi})) \prod_{\chi \in \vec{\chi}} c_{\chi}(a_{\chi}|\chi)$$

Use: Multiple disjoint **channels** (or regions) of binned distributions with multiple **samples** contributing to each with additional (possibly shared) systematics between sample estimates

Main pieces:

- Main Poisson p.d.f. for simultaneous measurement of multiple channels
- Event rates ν_{cb} (nominal rate ν_{scb}^0 with rate modifiers)
- Constraint p.d.f. (+ data) for "auxiliary measurements"
 - encode systematic uncertainties (e.g. normalization, shape)
- \vec{n} : events, \vec{a} : auxiliary data, $\vec{\eta}$: unconstrained pars, $\vec{\chi}$: constrained pars



Example: **Each bin** is separate (1-bin) **channel**, each **histogram** (color) is a **sample** and share a **normalization systematic** uncertainty

HistFactory Template

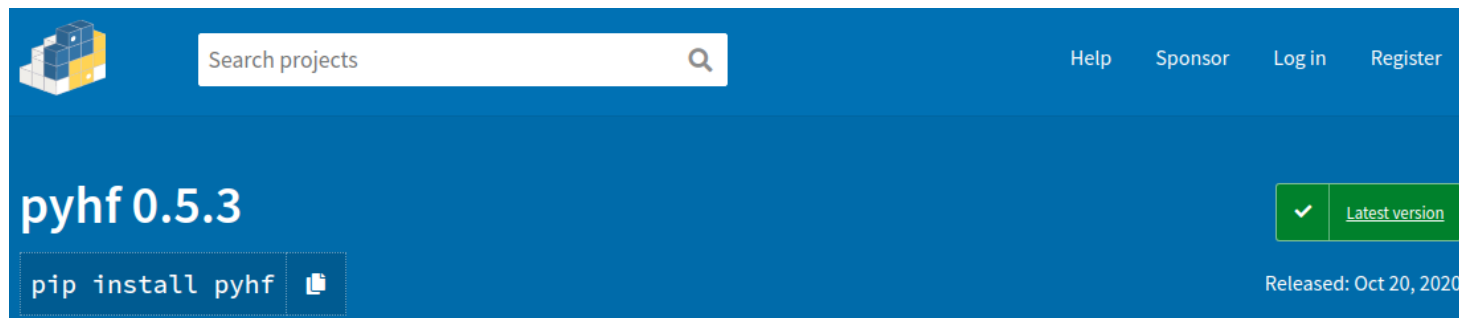
$$f(\vec{n}, \vec{a} | \vec{\eta}, \vec{\chi}) = \prod_{c \in \text{channels}} \prod_{b \in \text{bins}_c} \text{Pois}(n_{cb} | \nu_{cb}(\vec{\eta}, \vec{\chi})) \prod_{\chi \in \vec{\chi}} c_{\chi}(a_{\chi} | \chi)$$

Mathematical grammar for a simultaneous fit with

- multiple "channels" (analysis regions, (stacks of) histograms)
- each region can have multiple bins
- coupled to a set of constraint terms

This is a mathematical representation! Nowhere is any software spec defined
Until now (2018), the only implementation of HistFactory has been in ROOT

pyhf: HistFactory in pure Python



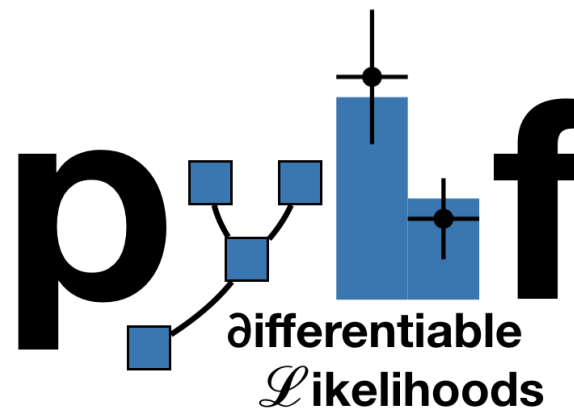
The screenshot shows the PyPI page for the package 'pyhf'. At the top left is the PyPI logo. To its right is a search bar with the text 'Search projects' and a magnifying glass icon. Further right are links for 'Help', 'Sponsor', 'Log in', and 'Register'. Below the search bar, the package name 'pyhf' and version '0.5.3' are displayed in large white text. To the right of the version is a green checkmark icon and the text 'Latest version'. Below the package name and version is a button with the text 'pip install pyhf' and a small icon of a terminal window. At the bottom right, it says 'Released: Oct 20, 2020'.

pyhf: HistFactory in pure Python

- First non-ROOT implementation of the HistFactory p.d.f. template
 - DOI [10.5281/zenodo.1169739](https://doi.org/10.5281/zenodo.1169739)
- pure-Python library as second implementation of HistFactory

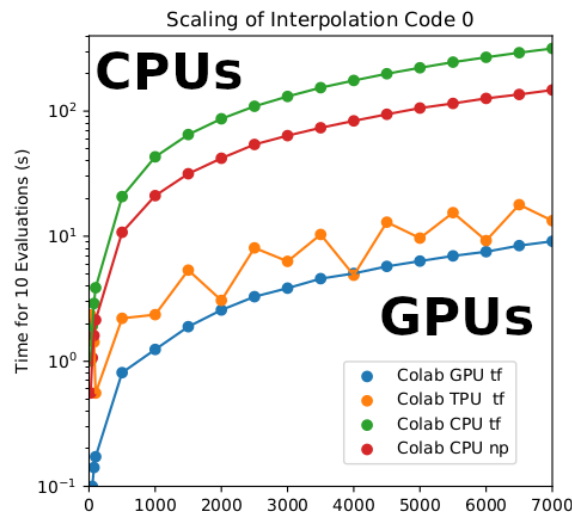
- `$ pip install pyhf`
- No dependence on ROOT!

- Open source tool for all of HEP
 - [IRIS-HEP](#) supported Scikit-HEP project
 - Used for reinterpretation in phenomenology paper (DOI: [10.1007/JHEP04\(2019\)144](https://doi.org/10.1007/JHEP04(2019)144)) and `SModelS`
 - Used in ATLAS SUSY groups and for internal pMSSM SUSY large scale reinterpretation
 - Maybe your experiment too!



Machine Learning Frameworks for Computation

- All numerical operations implemented in **tensor backends** through an API of n -dimensional array operations
- Using deep learning frameworks as computational backends allows for **exploitation of auto differentiation (autograd) and GPU acceleration**
- As huge buy in from industry we benefit for free as these frameworks are **continually improved** by professional software engineers (physicists are not)



- Show hardware acceleration giving **order of magnitude speedup** for some models!
- Improvements over traditional
 - 10 hrs to 30 min; 20 min to 10 sec

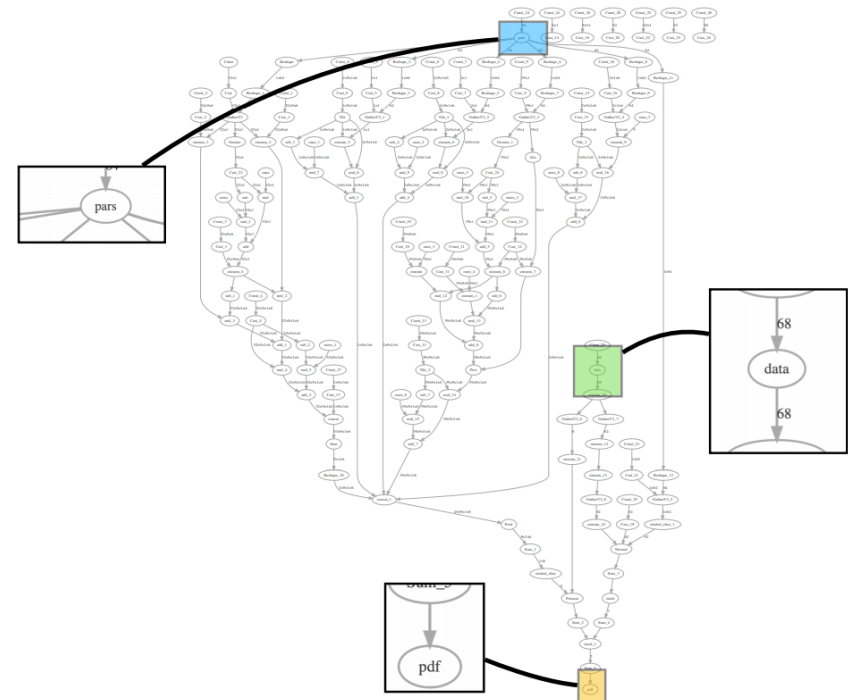
Automatic differentiation

With tensor library backends gain access to **exact (higher order) derivatives** — accuracy is only limited by floating point precision

$$\frac{\partial L}{\partial \mu}, \frac{\partial L}{\partial \theta_i}$$

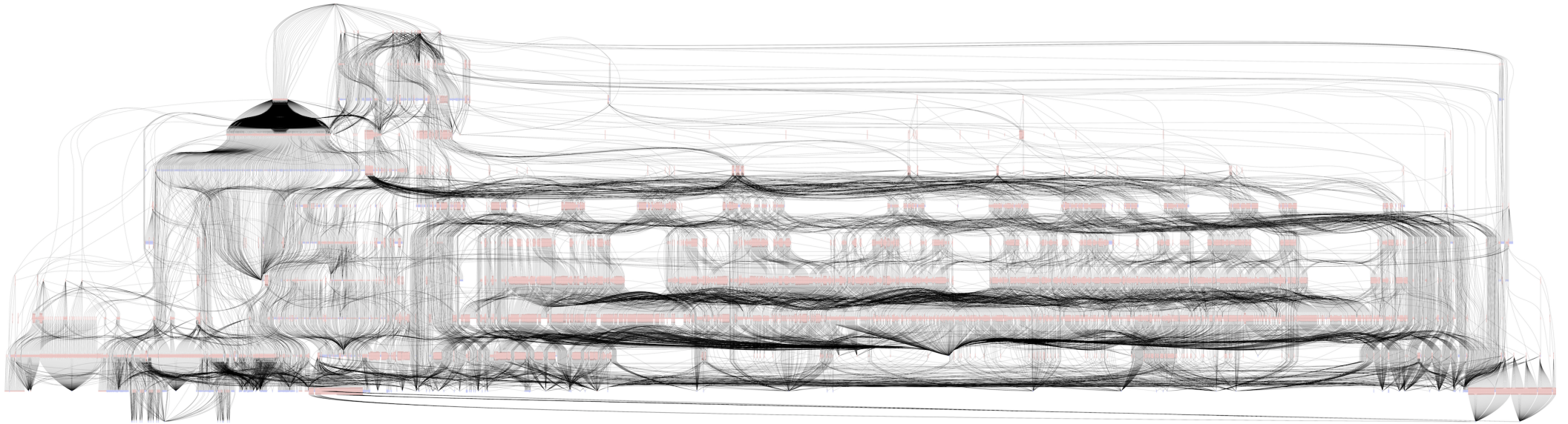
Exploit **full gradient of the likelihood** with **modern optimizers** to help speedup fit!

Gain this through the frameworks creating **computational directed acyclic graphs** and then applying the chain rule (to the operations)



Tensor backends offer a computational advantage

For visual comparison: the computational graph of the Higgs discovery analysis from the C++ framework. Image courtesy of Kyle Cranmer.



JSON spec fully describes the HistFactory model

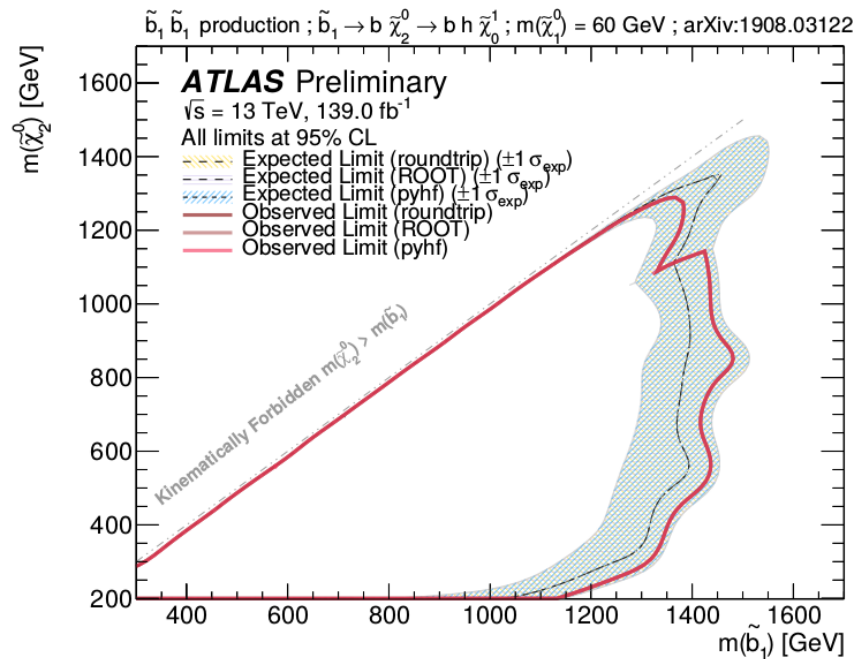
- Human & machine readable **declarative** statistical models
- Industry standard
 - Will be with us forever
- Parsable by every language
 - Highly portable
 - Bidirectional translation with ROOT
- Versionable and easily preserved
 - JSON Schema [describing HistFactory specification](#)
 - Attractive for analysis preservation
 - Highly compressible

```
{
  "channels": [ # List of regions
    { "name": "singlechannel",
      "samples": [ # List of samples in region
        { "name": "signal",
          "data": [20.0, 10.0],
          # List of rate factors and/or systematic uncertainties
          "modifiers": [ { "name": "mu", "type": "normfactor", "data": null } ]
        },
        { "name": "background",
          "data": [50.0, 63.0],
          "modifiers": [ { "name": "uncorr_bkguncrt", "type": "shapesys", "data": [5.0, 12.0] } ]
        }
      ]
    }
  ],
  "observations": [ # Observed data
    { "name": "singlechannel", "data": [55.0, 62.0] }
  ],
  "measurements": [ # Parameter of interest
    { "name": "Measurement", "config": { "poi": "mu", "parameters": [] } }
  ],
  "version": "1.0.0" # Version of spec standard
}
```

JSON defining a single channel, two bin counting experiment with systematics

ATLAS validation and publication of likelihoods

ATLAS Note	
Report number	ATL-PHYS-PUB-2019-029
Title	Reproducing searches for new physics with the ATLAS experiment through publication of full statistical likelihoods
Corporate Author(s)	The ATLAS collaboration



(ATLAS, 2019)

New open release allows theorists to explore LHC data in a new way

The ATLAS collaboration releases full analysis likelihoods, a first for an LHC experiment

9 JANUARY, 2020 | By Katarina Anthony



Explore ATLAS open likelihoods on the HEPData platform (Image: CERN)

(CERN, 2020)

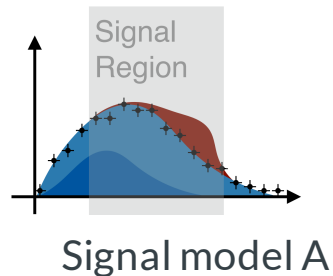
JSON Patch for signal model (reinterpretation)

JSON Patch gives ability to **easily mutate model**

Think: test a **new theory** with a **new patch!**

(c.f. [Lukas Heinrich's RECAST talk from Snowmass 2021 Computational Frontier Workshop](#))

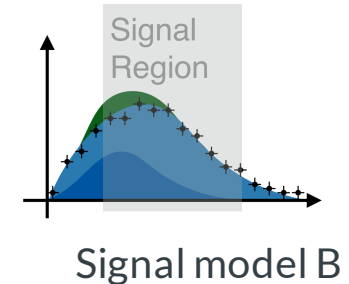
Combined with RECAST gives powerful tool for **reinterpretation studies**



```
# Using CLI
$ pyhf cls example.json | jq .CLs_obs
0.053994246621274014

$ cat new_signal.json
[{"op": "replace",
 "path": "/channels/0/samples/0/data",
 "value": [10.0, 6.0]}]

$ pyhf cls example.json --patch new_signal.json | jq .CLs_obs
0.3536906623262466
```



Likelihoods preserved on HEPData

- `pyhf` pallet:
 - Background-only model JSON stored
 - Hundreds of signal model JSON Patches stored together as a `pyhf` "patch set" file
- Fully preserve and publish the full statistical model and observations to give likelihood
 - with own DOI! DOI [10.17182/hepdata.90607.v3/r3](https://doi.org/10.17182/hepdata.90607.v3/r3)

HEPData Search HEPData

Search

Q Browse all

Hide Publication Information

Search for direct production of electroweakinos in final states with one lepton, missing transverse momentum and a Higgs boson decaying into two b -jets in (pp) collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector

The ATLAS collaboration

Aad, Georges, Abbott, Brad, Abbott, Dale Charles, Abed Abud, Adam, Abeling, Kira, Abhayasinghe, Deshan Kavishka, Abidi, Syed Haider, Abouzeid, Ossama, Abraham, Nicola, Abramowicz, Halina

PhD Thesis, 2020.

<https://doi.org/10.17182/hepdata.90607.v2>

INSPIRE Resources

Abstract

The results of a search for electroweakino pair production $pp \rightarrow \tilde{\chi}_1^{\pm} \tilde{\chi}_2^0$ in which the chargino ($\tilde{\chi}_1^{\pm}$) decays into a W boson and the lightest neutralino ($\tilde{\chi}_1^0$), while the heavier neutralino ($\tilde{\chi}_2^0$) decays into the Standard Model 125 GeV Higgs boson and a second $\tilde{\chi}_1^0$ are presented. The signal selection requires a pair of b -tagged jets consistent with those from a Higgs boson decay, and either an electron or a muon from the W boson decay, together with missing transverse momentum from the corresponding neutrino and the stable neutralinos. The analysis is based on data corresponding to 139 fb^{-1} of $\sqrt{s} = 13$ TeV pp collisions provided by the Large Hadron Collider and recorded by the ATLAS detector. No statistically significant evidence of an excess of events above the Standard Model expectation is found. Limits are set on the direct production of the electroweakinos in simplified models, assuming pure wino cross-sections. Masses of $\tilde{\chi}_1^{\pm} / \tilde{\chi}_2^0$ up to 740 GeV are excluded at 95% confidence level for a massless $\tilde{\chi}_1^0$.

Additional Publication Resources

filter

Common Resources 4

- dataMC_VR_onLM_nomct 2
- dataMC_VR_onMM_nomct 2
- dataMC_VR_onHM_nomct 2
- dataMC_VR_offLM_nomct 2
- dataMC_VR_offMM_nomct 2
- dataMC_VR_offHM_nomct 2
- dataMC_SRRM_mct 2
- dataMC_SRRM_mct 2
- dataMC_SRLM_mct 2
- dataMC_SRRM_nombb 2
- dataMC_SRRM_nombb 2
- dataMC_SRLM_nombb 2

Observed limit 1lbb 2

Observed limit 1lbb (Up) 2

Observed limit 1lbb (Down) 2

Expected limit 1lbb 2

Upper limits 1Lbb 2

External Link
web page with auxiliary material
View Resource

C++ File
C++/ROOT-inspired pseudo-code to emulate the signal selection efficiency using the provided reinterpretation material
Download

Text File
Example SLHA file
Download

gz File
Archive of full likelihoods in the HistFactory JSON format described in CERN-EP-2019-188. For each signal point the background-only model is found in the file named BkgOnly.json. All jsonpatches are contained in the file patchset.json. Each patch is identified in patchset.json by the metadata field "name": "CIN2_Wh_hbb_m1_m2" where m1 is the mass of both the lightest chargino and the next-to-lightest neutralino (which are assumed to be nearly mass degenerate) and m2 is the mass of the lightest neutralino.
Download

```
$ tree pyhf-pallet
pyhf-pallet
├── BkgOnly.json
├── patchset.json
└── README.md

0 directories, 3 files
```

...can be used from HEPData

- pyhf pallet:
 - Background-only model JSON stored
 - Hundreds of signal model JSON Patches stored together as a [pyhf "patch set" file](#)
- Fully preserve and publish the full statistical model and observations to give likelihood
 - with own DOI! DOI [10.17182/hepdata.90607.v3/r3](https://doi.org/10.17182/hepdata.90607.v3/r3)

```

# pyhf pallet for the SUSY EWK 1Lbb analysis
$ pyhf contrib download https://doi.org/10.17182/hepdata.90607.v3/r3 1Lbb-pallet && cd 1Lbb-pallet

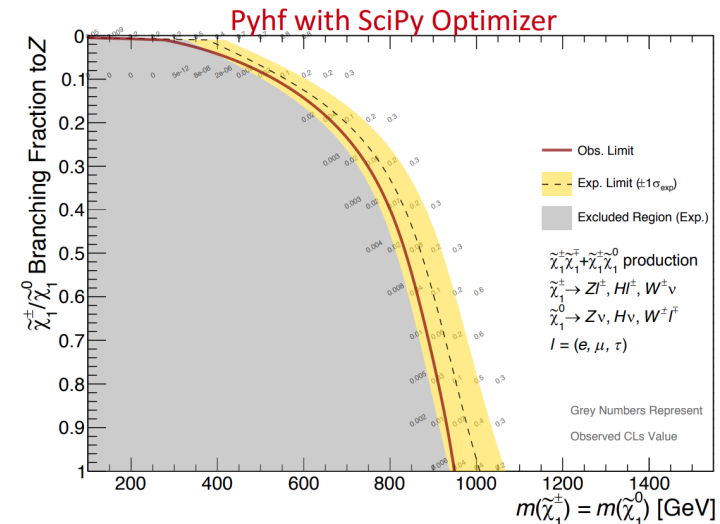
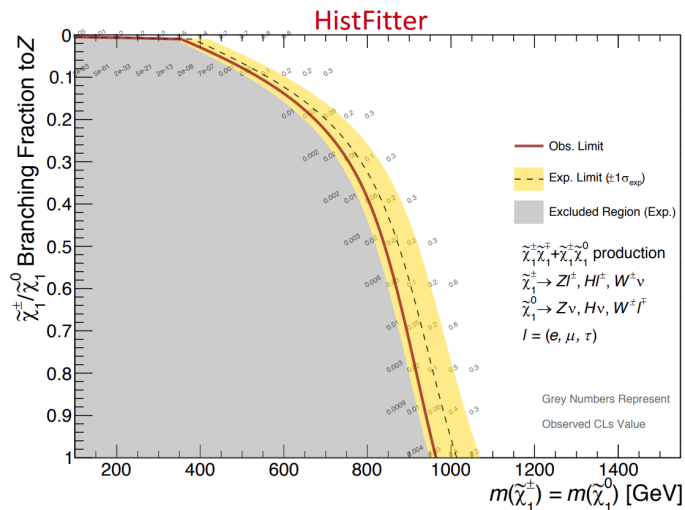
# verify patchset is valid
$ pyhf patchset verify BkgOnly.json patchset.json
All good.

# signal model: m1 = 900, m2 = 300 (chain CLI API output)
$ cat BkgOnly.json | \
  pyhf cls --patch <(pyhf patchset extract --name C1N2_Wh_hbb_900_300 patchset.json) | \
  jq .CLs_obs
0.5004165245329418

# new signal model: m1 = 900, m2 = 400 (use serialized CLI API output)
$ pyhf patchset extract --name C1N2_Wh_hbb_900_400 --output-file C1N2_Wh_hbb_900_400_patch.json patchset.json
$ pyhf cls --patch C1N2_Wh_hbb_900_400_patch.json BkgOnly.json | jq .CLs_obs
0.5735007268333779
```

Rapid adoption in ATLAS...

- **Five** ATLAS analyses with full likelihoods published to HEPData
- ATLAS SUSY will be continuing to publish full Run 2 likelihoods
- direct staus, [doi:10.17182/hepdata.89408](https://doi.org/10.17182/hepdata.89408) (2019)
- sbottom multi-b, [doi:10.17182/hepdata.91127](https://doi.org/10.17182/hepdata.91127) (2019)
- 1Lbb, [doi:10.17182/hepdata.92006](https://doi.org/10.17182/hepdata.92006) (2019)
- 3L eRJR, [doi:10.17182/hepdata.90607](https://doi.org/10.17182/hepdata.90607) (2020)
- ss3L search, [doi:10.17182/hepdata.91214](https://doi.org/10.17182/hepdata.91214) (2020)



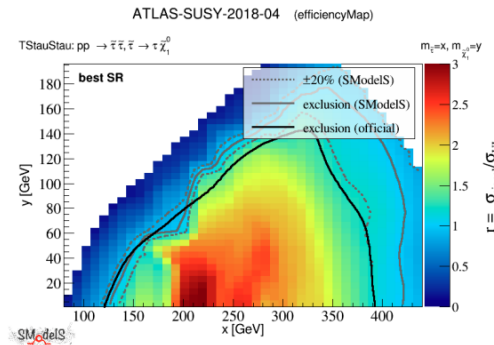
...and by theory

- `pyhf` likelihoods discussed in
 - [Les Houches 2019 Physics at TeV Colliders: New Physics Working Group Report](#)
 - [Higgs boson potential at colliders: status and perspectives](#)
- `SModelS` team has implemented a `SModelS/pyhf` interface [[arXiv:2009.01809](#)]
 - tool for interpreting simplified-model results from the LHC
 - designed to be used by theorists
 - `SModelS` authors giving [tutorial later today!](#)
- Have produced three comparisons to published ATLAS likelihoods: [ATLAS-SUSY-2018-04](#), [ATLAS-SUSY-2018-31](#), [ATLAS-SUSY-2019-08](#)
 - Compare simplified likelihood (bestSR) to full likelihood (`pyhf`) using `SModelS`

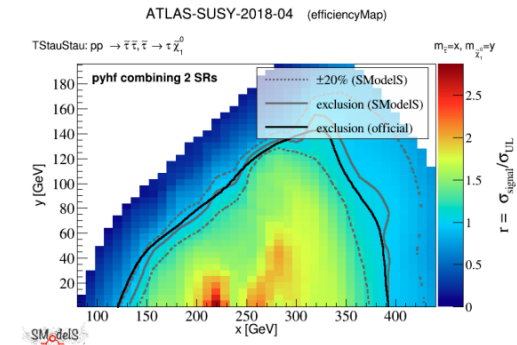
Validation & impact

Gaël Alguero, SK, Wolfgang Waltenberger, [arXiv:2009.01809](#)

- ATLAS-SUSY-2018-04: TStauStau



Best SR: over exclusion



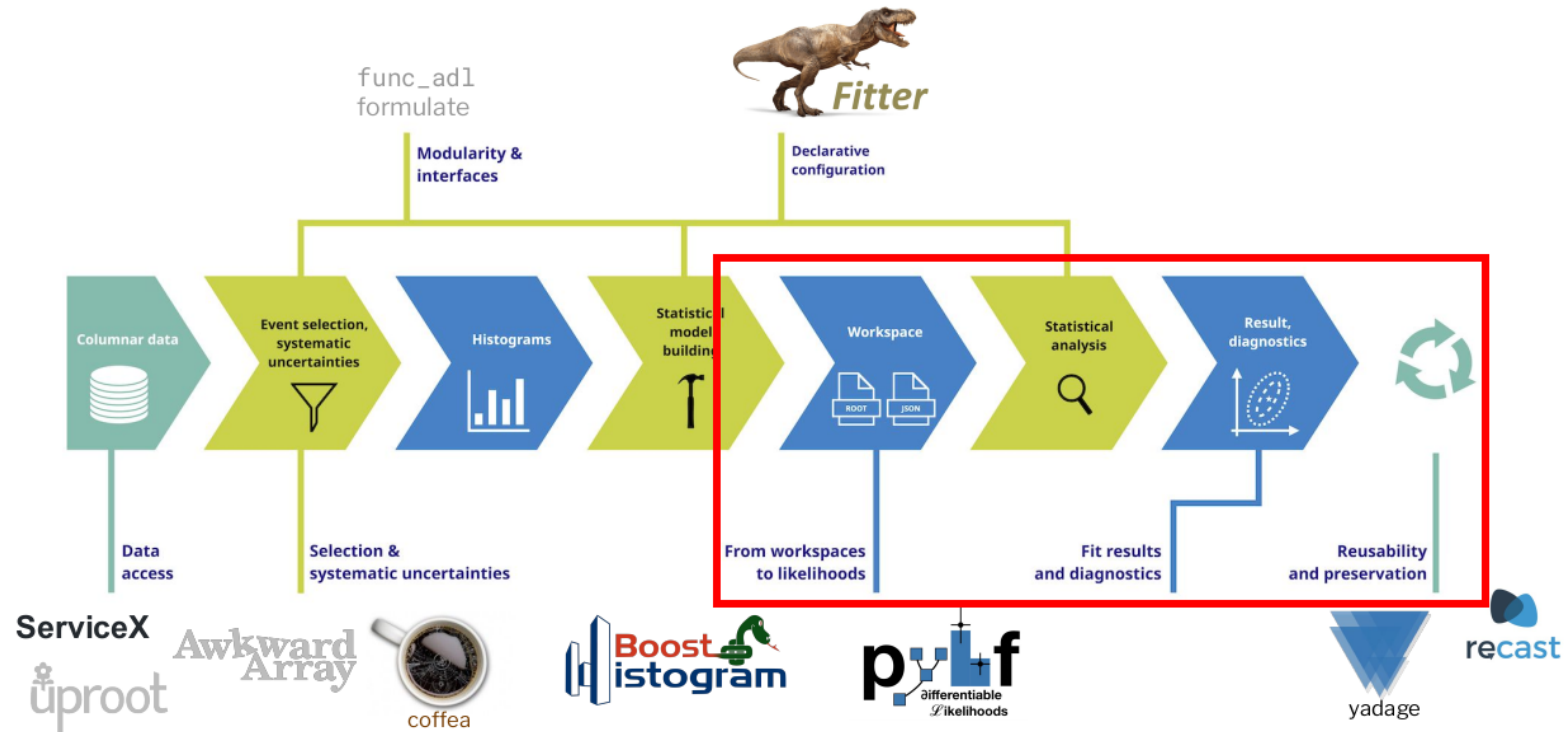
Full likelihood: very good agreement with official ATLAS result

The remaining small difference is probably due to the (interpolated) $A \times \epsilon$ values from the simplified model efficiency maps not exactly matching the “true” ones of the experimental analysis.

S. Kraml - Feedback on use of public likelihoods - 24 Sep 2020

[Feedback on use of public Likelihoods](#), Sabine Kraml
(ATLAS Exotics + SUSY Reinterpretations Workshop)

Core part of IRIS-HEP Analysis Systems pipeline



- Accelerating fitting (reducing time to **insight** (statistical inference)!) (`pyhf` + `cabinetry`)
- Flexible schema great for open likelihood **preservation**
 - Likelihood serves as high information-density summary of analysis
- An enabling technology for **reinterpretation** (`pyhf` + RECAST)

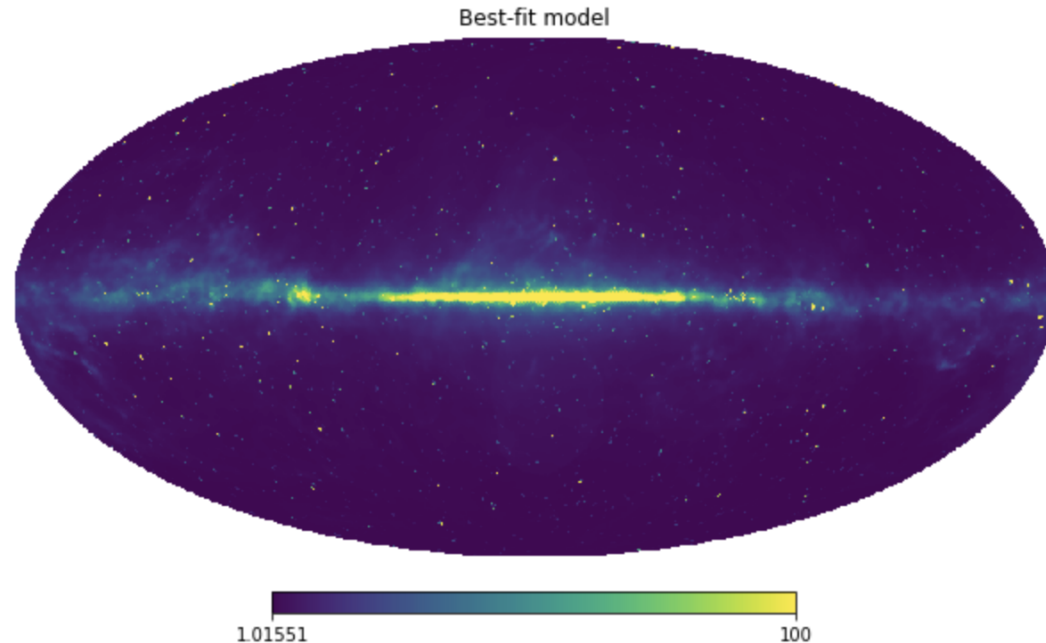
Use in analysis outside of particle physics

- Public data from [Fermi Large Area Telescope \(LAT\)](#) analyzed by L. Heinrich et al.
- The LAT is a high-energy gamma-ray telescope – the gamma-ray photons come from extreme cosmological events
- Can represent the photons counts in the LAT as a binned model
 - Here full-sky map visualized with [healpy](#)'s Mollweide projection
 - Think: 2d histogram with special binning

```
In [6]: m = pyhf.Model(spec, poiname = 'mu_dm')
bestfit = pyhf.optimizer.minimize(
    lambda theta,data,m: -m.logpdf(theta, data),data,m,
    init_pars = [1]*5,
    par_bounds = [[0,20]]*5
)
print(bestfit)

[ 0.89713789  0.43737808  1.0913045   0.75777142  13.3680226 ]
```

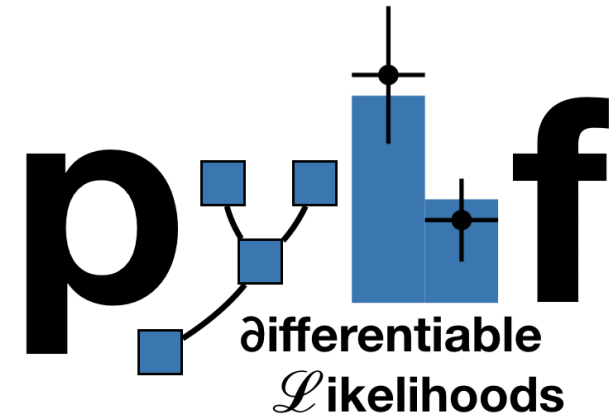
```
In [7]: hp.mollview(m.expected_data(bestfit), max=100, title='Best-fit model', )
```



Summary

pyhf provides:

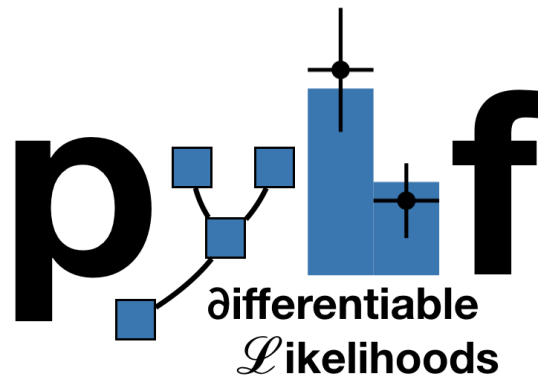
- **Accelerated** fitting library
 - reducing time to insight/inference!
 - Hardware acceleration on GPUs and vectorized operations
 - Backend agnostic Python API and CLI
- Flexible **declarative** schema
 - JSON: ubiquitous, universal support, versionable
- Enabling technology for **reinterpretation**
 - JSON Patch files for efficient computation of new signal models
 - Unifying tool for theoretical and experimental physicists
- Project in growing **Pythonic HEP ecosystem**
 - [Openly developed on GitHub](#) and welcome contributions
 - [Comprehensive open tutorials](#)
 - Ask us about Scikit-HEP and IRIS-HEP!



Thanks for listening!

Come talk with us!

www.scikit-hep.org/pyhf



HistFactory Template (in more detail)

$$f(\vec{n}, \vec{a} | \vec{\eta}, \vec{\chi}) = \prod_{c \in \text{channels}} \prod_{b \in \text{bins}_c} \text{Pois}(n_{cb} | \nu_{cb}(\vec{\eta}, \vec{\chi})) \prod_{\chi \in \vec{\chi}} c_{\chi}(a_{\chi} | \chi)$$

$$\nu_{cb}(\vec{\eta}, \vec{\chi}) = \sum_{s \in \text{samples}} \underbrace{\left(\sum_{\kappa \in \vec{\kappa}} \kappa_{scb}(\vec{\eta}, \vec{\chi}) \right)}_{\text{multiplicative}} \left(\nu_{scb}^0(\vec{\eta}, \vec{\chi}) + \underbrace{\sum_{\Delta \in \vec{\Delta}} \Delta_{scb}(\vec{\eta}, \vec{\chi})}_{\text{additive}} \right)$$

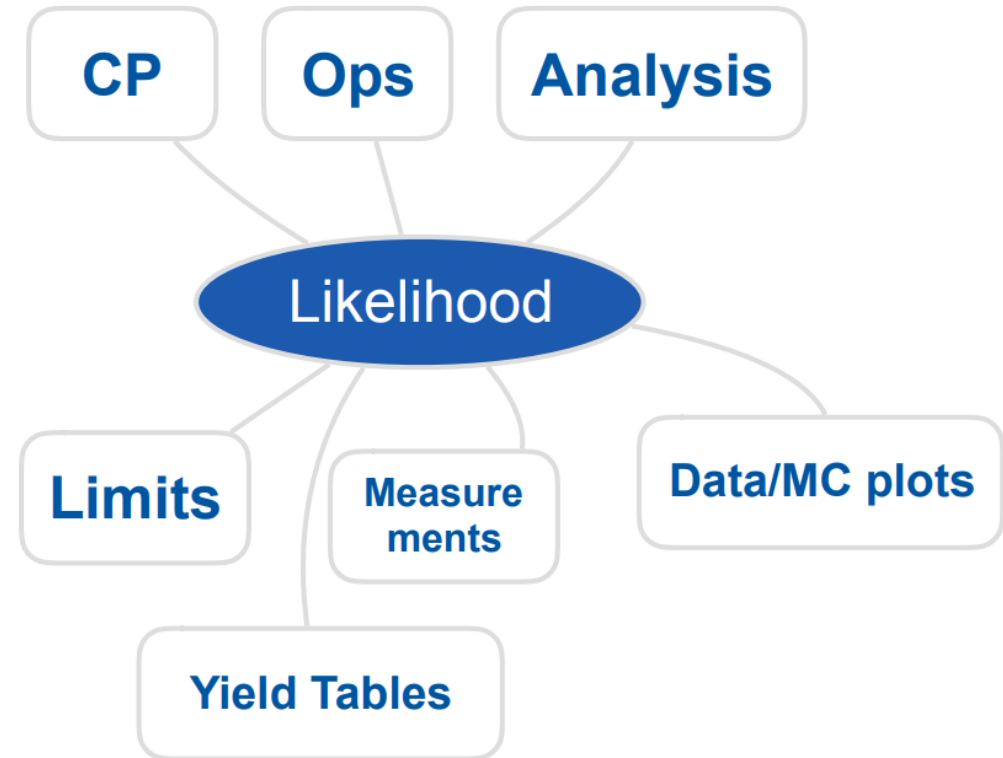
Use: Multiple disjoint **channels** (or regions) of binned distributions with multiple **samples** contributing to each with additional (possibly shared) systematics between sample estimates

Main pieces:

- Main Poisson p.d.f. for simultaneous measurement of multiple channels
- Event rates ν_{cb} from nominal rate ν_{scb}^0 and rate modifiers κ and Δ
- Constraint p.d.f. (+ data) for "auxiliary measurements"
 - encoding systematic uncertainties (normalization, shape, etc)
- \vec{n} : events, \vec{a} : auxiliary data, $\vec{\eta}$: unconstrained pars, $\vec{\chi}$: constrained pars

Why is the likelihood important?

- High information-density summary of analysis
- Almost everything we do in the analysis ultimately affects the likelihood and is encapsulated in it
 - Trigger
 - Detector
 - Combined Performance / Physics Object Groups
 - Systematic Uncertainties
 - Event Selection
- Unique representation of the analysis to reuse and preserve



Full likelihood serialization...

...making good on [19 year old agreement to publish likelihoods](#)

Massimo Corradi

It seems to me that there is a general consensus that what is really meaningful for an experiment is *likelihood*, and almost everybody would agree on the prescription that experiments should give their likelihood function for these kinds of results. [Does everybody agree on this statement, to publish likelihoods?](#)

Louis Lyons

Any disagreement? [Carried unanimously. That's actually quite an achievement for this Workshop.](#)

[\(1st Workshop on Confidence Limits, CERN, 2000\)](#)

This hadn't been done in HEP until 2019

- In an "open world" of statistics this is a difficult problem to solve
- What to preserve and how? All of ROOT?
- Idea: Focus on a single more tractable binned model first

References

1. F. James, Y. Perrin, L. Lyons, *Workshop on confidence limits: Proceedings*, 2000.
2. ROOT collaboration, K. Cranmer, G. Lewis, L. Moneta, A. Shibata and W. Verkerke, *HistFactory: A tool for creating statistical models for use with RooFit and RooStats*, 2012.
3. L. Heinrich, H. Schulz, J. Turner and Y. Zhou, *Constraining A_4 Leptonic Flavour Model Parameters at Colliders and Beyond*, 2018.
4. A. Read, *Modified frequentist analysis of search results (the CL_s method)*, 2000.
5. K. Cranmer, *CERN Latin-American School of High-Energy Physics: Statistics for Particle Physicists*, 2013.
6. ATLAS collaboration, *Search for bottom-squark pair production with the ATLAS detector in final states containing Higgs bosons, b-jets and missing transverse momentum*, 2019
7. ATLAS collaboration, *Reproducing searches for new physics with the ATLAS experiment through publication of full statistical likelihoods*, 2019
8. ATLAS collaboration, *Search for bottom-squark pair production with the ATLAS detector in final states containing Higgs bosons, b-jets and missing transverse momentum: HEPData entry*, 2019

