Likelihood preservation and statistical reproduction of searches for new physics

Matthew Feickert, Lukas Heinrich, Giordon Stark
on behalf of the ATLAS collaboration

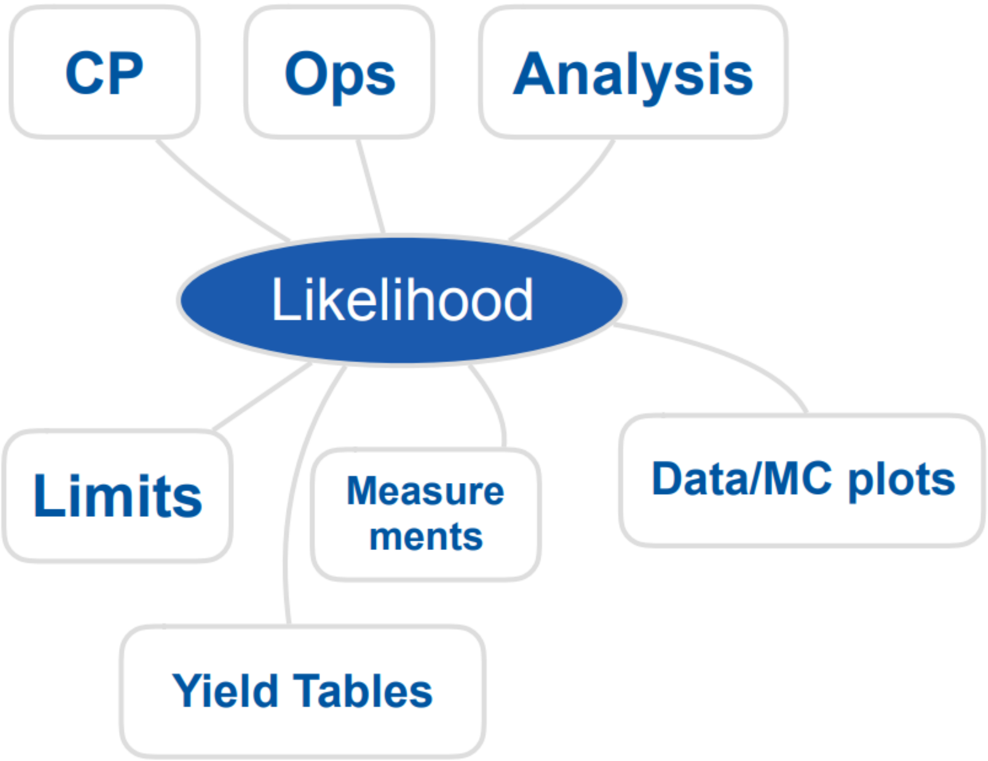
matthew.feickert@cern.ch

CHEP 2019

November 7th, 2019

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
  - Systematic Uncertainties
  - Event Selection
- Unique representation of the analysis to preserve



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.

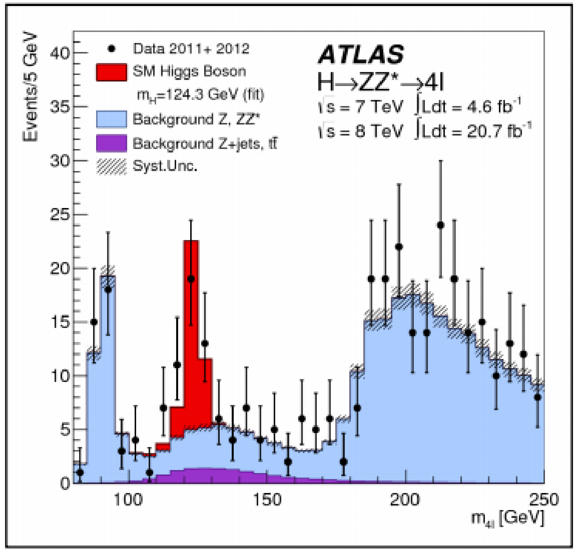
(1st Workshop on Confidence Limits, CERN, 2000)

This hadn't been done in HEP until now

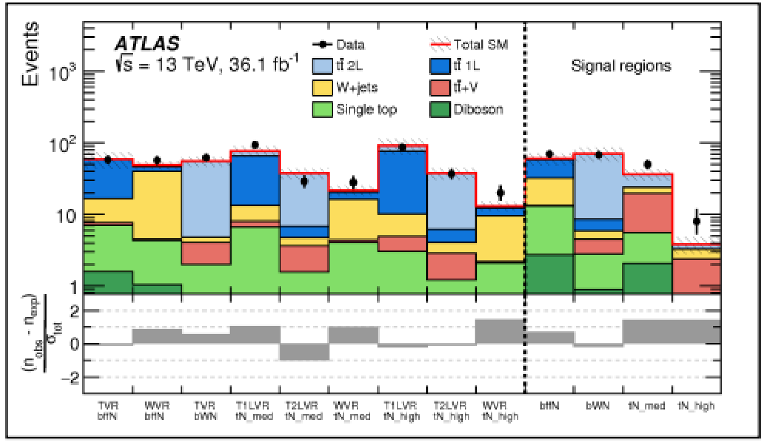
- In an "open world" of statistics this is a difficult problem to solve
- What to preserve and how? All of ROOT?
- Idea: Focus on a single more tractable binned model first

Enter HistFactory

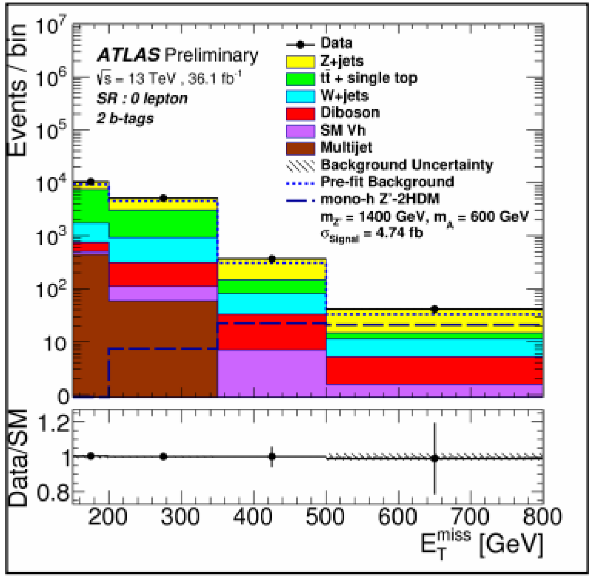
- A flexible p.d.f. template to build statistical models from binned distributions and data
- Developed by Cranmer, Lewis, Moneta, Shibata, and Verkerke (CERN-OPEN-2012-016)
- Widely used by the HEP community for standard model measurements and BSM searches



SM



SUSY



Exotics

HistFactory Template

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

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

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^0_{scb} 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

HistFactory Template

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

This is a mathematical representation! Nowhere is any software spec defined

Until now, the only implementation of HistFactory has been in RooStats+RooFit

- Preservation: Likelihood stored in the binary ROOT format
  - Challenge for long-term preservation (i.e. HEPData)
  - Why is a histogram needed for an array of numbers?
- To start using HistFactory p.d.f.s first have to learn ROOT, RooFit, RooStats
  - Problem for our theory colleagues (generally don't want to)
- Difficult to use for reinterpretation

`pyhf`: HistFactory in pure Python

- First non-ROOT implementation of the HistFactory p.d.f. template
  - DOI 10.5281/zenodo.1169739
- pure-Python library as second implementation of HistFactory
  - `pip install pyhf`
  - No dependence on ROOT!
- Has a JSON spec that fully describes the HistFactory model
  - JSON: Industry standard, parsable by every language, human & machine readable, versionable and easily preserved (HEPData is JSON)
- Open source tool for all of HEP
  - Originated from a DIANA/HEP project fellowship and now an IRIS-HEP supported project
  - Used for reinterpretation in phenomenology paper (DOI: 10.1007/JHEP04(2019)144)
  - Used internally in ATLAS for pMSSM SUSY large scale reinterpretation

Example pyhf JSON spec

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

```
{
    "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
}
```

Live demo time!

Just click the button!

CL_s Example using pyhf CLI

```
$ pyhf cls example.json
{
    "CLs_exp": [
        0.00040903874532508 41,
        0.0032606968023913925,
        0.022552575976539 17,
        0.11898700005707148,
        0.39844667251932997
    ],
    "CLs_obs": 0.05399424662127401 4
}
```

JSON Patch for new signal models

```
{
    "channels": [
        { "name": "singlechannel",
          "samples": [
            { "name": "signal",
              "data": [20.0, 10.0],
              "modifiers": [ { "name": "mu", "type": "normfactor", "data": null} ]
            },
# Rest of the model
}
```

Original model

```
[{
    "op": "replace",
    "path": "/channels/0/samples/0/data",
    "value": [10.0, 6.0]
}]
```

New Signal (JSON Patch file)

```
{
    "channels": [
        { "name": "singlechannel",
          "samples": [
            { "name": "signal",
              "data": [10.0, 6.0],
              "modifiers": [ { "name": "mu", "type": "normfactor", "data": null} ]
            },
# Rest of the model
}
```

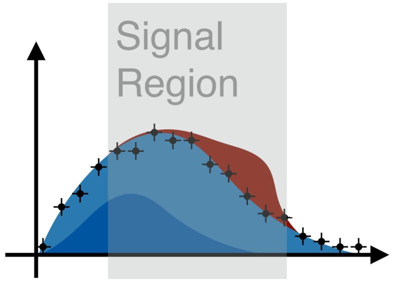
Reinterpretation

JSON Patch for new signal models

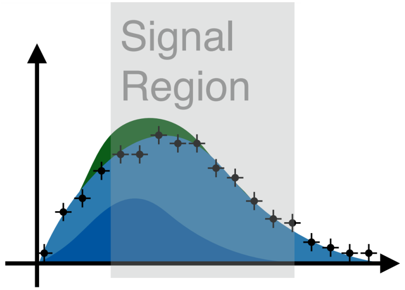
```
$ pyhf cls example.json | jq .CLs_obs
0.053994246621274014

$ 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
```



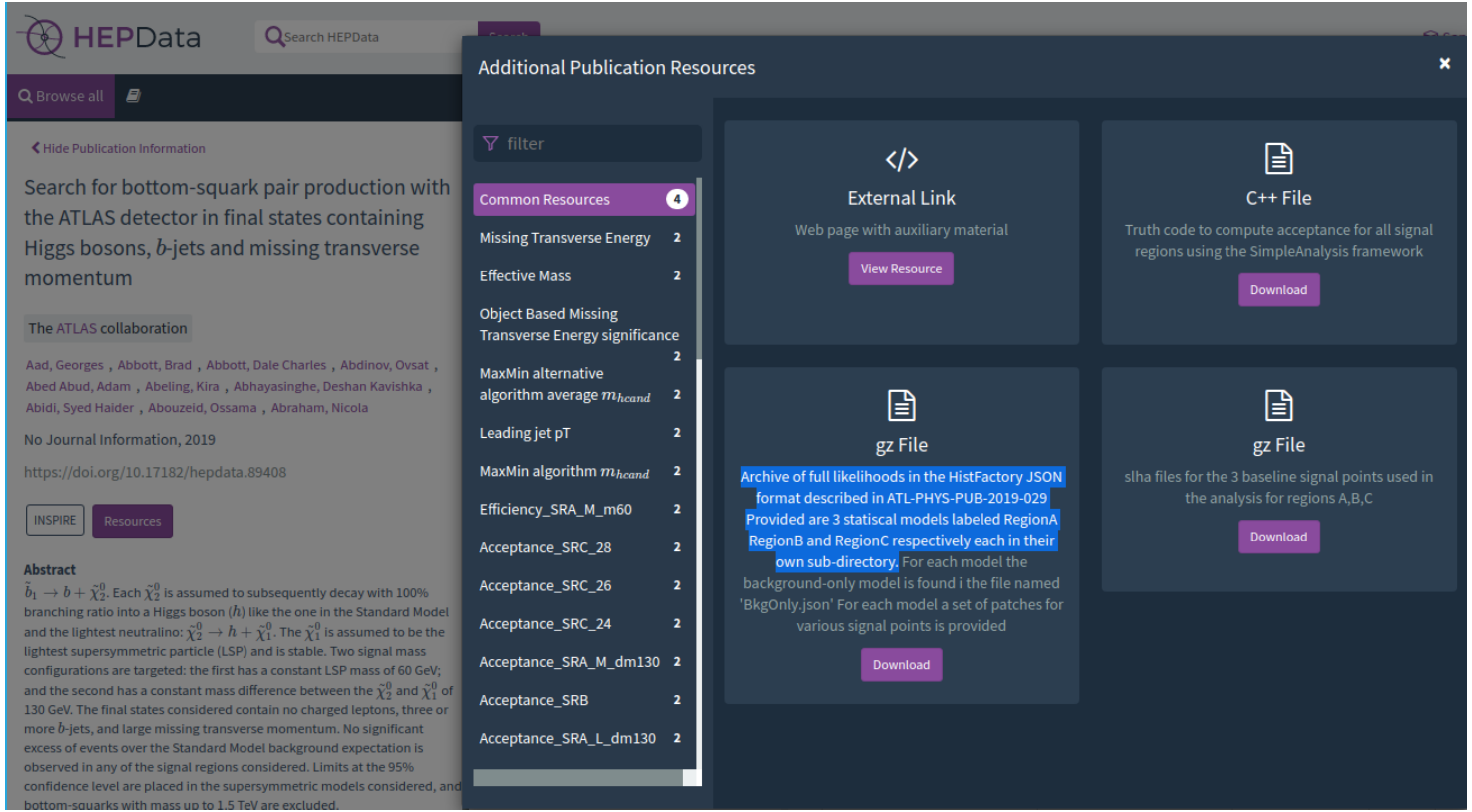
Original analysis (model A)



Recast analysis (model B)

Likelihoods preserved on HEPData

- Background-only model JSON stored
- Signal models stored as JSON Patch files
- Together are able to fully preserve the full model (with own DOI! DOI 10.17182/hepdata.89408.v1/r2)



...can be streamed from HEPData

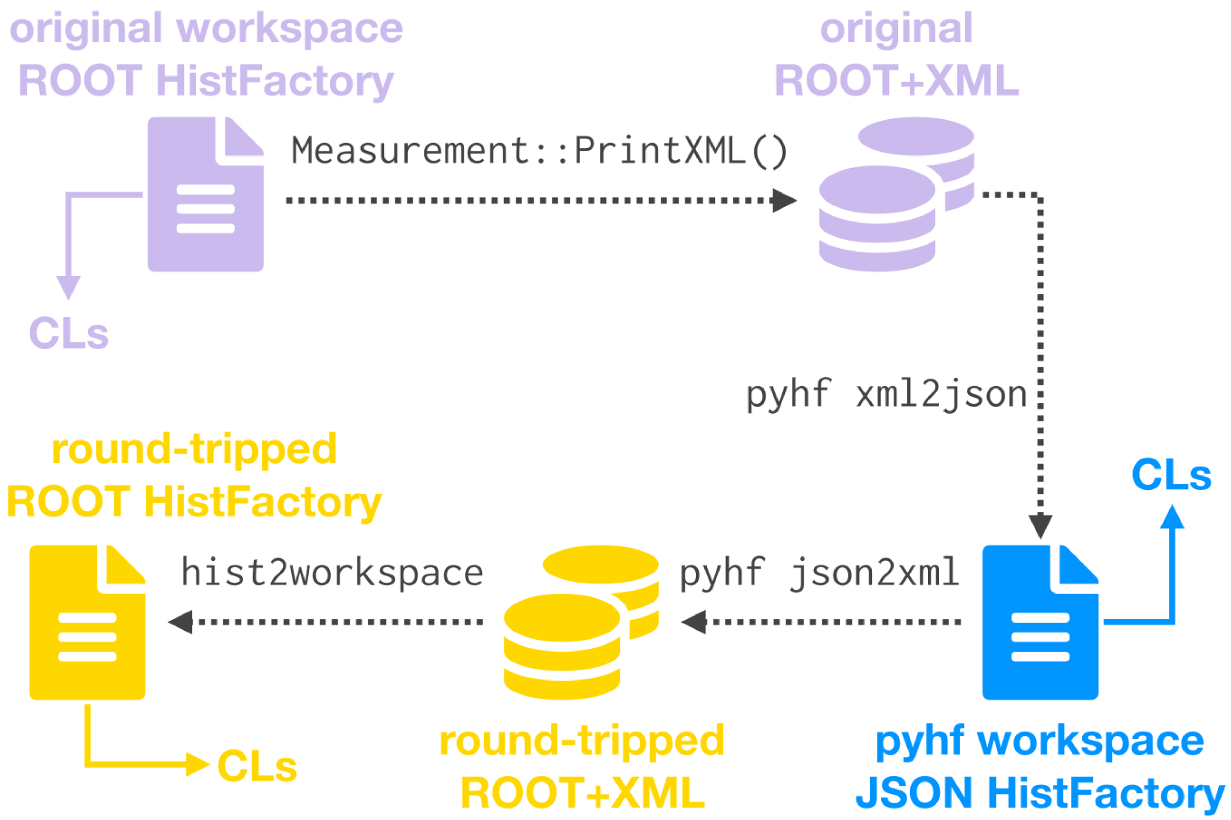
- Background-only model JSON stored
- Signal models stored as JSON Patch files
- Together are able to fully preserve the full model (with own DOI! DOI 10.17182/hepdata.89408.v1/r2)

```
# One signal model
$ curl -sL https://doi.org/10.17182/hepdata.89408.v1/r2 | \
  tar -O -xzv RegionA/BkgOnly.json | \
  pyhf cls --patch <(curl -sL https://doi.org/10.17182/hepdata.89408.v1/r2 | \
    tar -O -xzv RegionA/patch.sbottom_1300_205_60.json) | \
  jq .CLs_obs
0.24443637994463

# A different signal model
$ curl -sL https://doi.org/10.17182/hepdata.89408.v1/r2 | \
  tar -O -xzv RegionA/BkgOnly.json | \
  pyhf cls --patch <(curl -sL https://doi.org/10.17182/hepdata.89408.v1/r2 | \
    tar -O -xzv RegionA/patch.sbottom_1300_230_100.json) | \
  jq .CLs_obs
0.040766026035752724
```

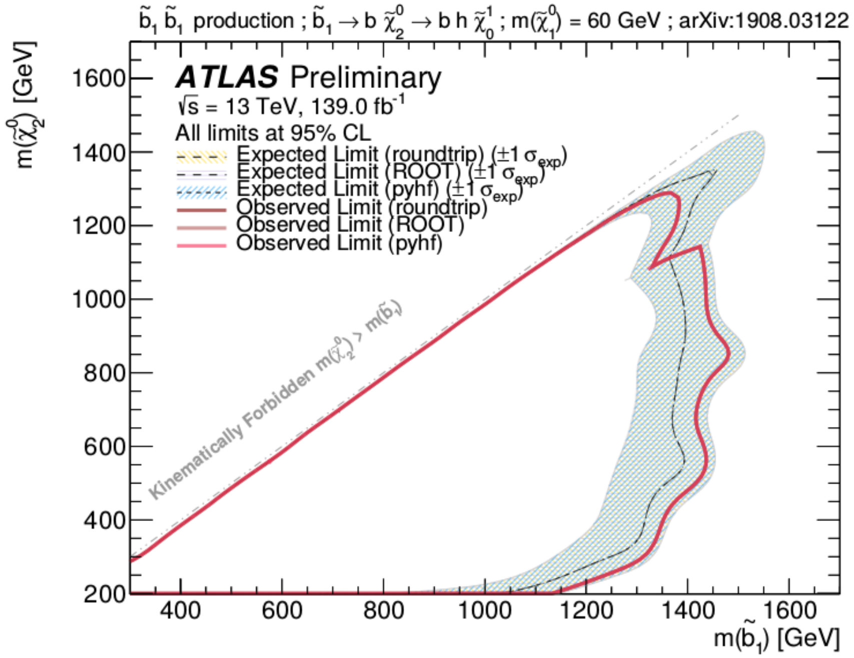
Likelihood serialization and reproduction

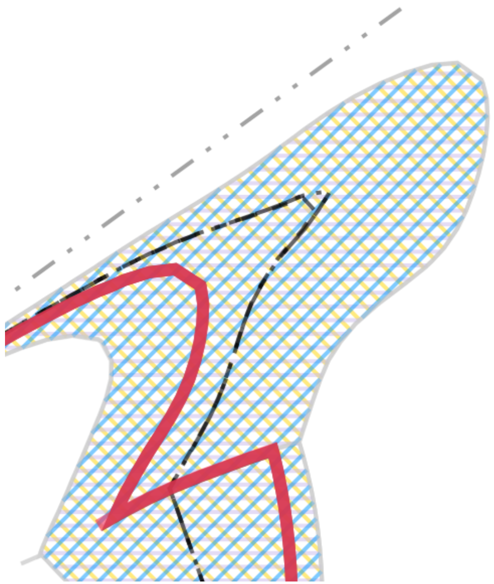
- ATLAS PUB note on the JSON schema for serialization and reproduction of results (ATL-PHYS-PUB-2019-029)
    - Contours: original ROOT+XML, pyhf JSON, JSON converted back to ROOT+XML



Likelihood serialization and reproduction

- ATLAS PUB note on the JSON schema for serialization and reproduction of results (ATL-PHYS-PUB-2019-029)
  - Contours: original ROOT+XML, pyhf JSON, JSON converted back to ROOT+XML
    - Overlay of contours nice visualization of near perfect agreement
  - Serialized likelihood and reproduced results of ATLAS Run-2 search for sbottom quarks (CERN-EP-2019-142) and published to HEPData
  - Shown to reproduce results but faster! **ROOT:** 10+ hours **pyhf:** < 30 minutes





Summary

Through pyhf are able to provide:

- **JSON specification** of likelihoods
  - human/machine readable, versionable, HEPData friendly, orders of magnitude smaller
- **Bidirectional translation** of likelihood specifications
  - ROOT workspaces ↔ JSON
- Independent **pure-Python implementation** of HistFactory + hypothesis testing
- Publication for the first time of the **full likelihood** of a search for new physics

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)

ATLAS PUB Note

ATL-PHYS-PUB-2019-029

5th August 2019

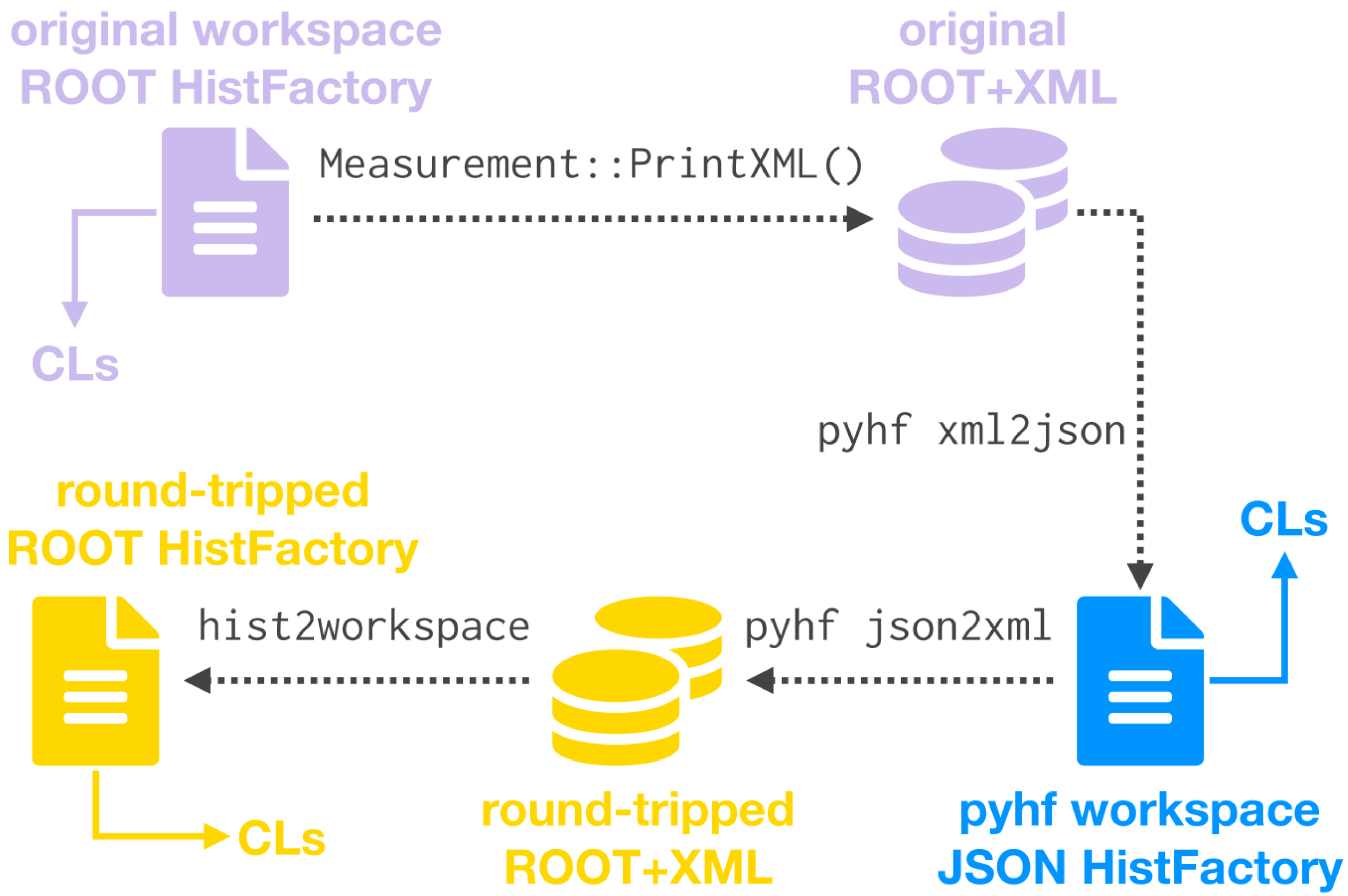
Reproducing searches for new physics with the ATLAS experiment through publication of full statistical likelihoods

The ATLAS Collaboration

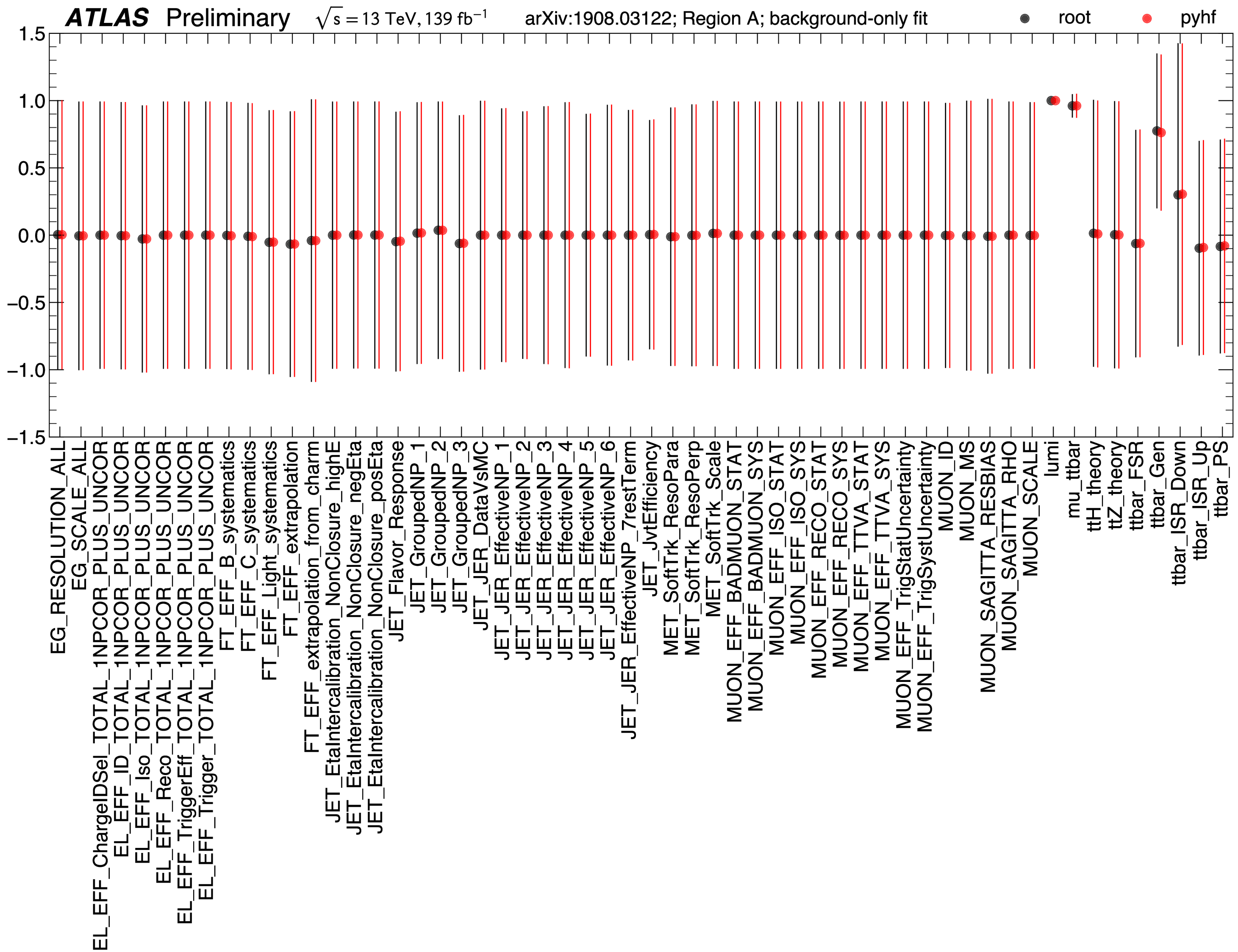
(ATLAS, 2019)

Backup

ROOT + XML to JSON and back



Best-fit parameter values



JSON Patch files for new signal models

```
$ 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 can be streamed from HEPData

```
# One signal model
$ curl -sL https://doi.org/10.17182/hepdata.89408.v1/r2 | \
  tar -O -xzv RegionA/BkgOnly.json | \
  pyhf cls --patch <(curl -sL https://doi.org/10.17182/hepdata.89408.v1/r2 | \
    tar -O -xzv RegionA/patch.sbottom_1300_205_60.json) | \
  jq .CLs_obs
0.24443637909054463

# A different signal model
$ curl -sL https://doi.org/10.17182/hepdata.89408.v1/r2 | \
  tar -O -xzv RegionA/BkgOnly.json | \
  pyhf cls --patch <(curl -sL https://doi.org/10.17182/hepdata.89408.v1/r2 | \
    tar -O -xzv RegionA/patch.sbottom_1300_230_100.json) | \
  jq .CLs_obs
0.040766026035752724
```

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. 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
5. ATLAS collaboration, *Reproducing searches for new physics with the ATLAS experiment through publication of full statistical likelihoods*, 2019
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: HEPData entry*, 2019

The end.