pyhf Roadmap:

2019 into 2020

(for the dev team)
Matthew Feickert
matthew.feickert@cern.ch

2019 IRIS-HEP Institute Retreat

September 12th, 2019

ILLINOIS

pyhf
differentiable
Likelihoods

Collaborators

Lukas Heinrich

CERN

Giordon Stark

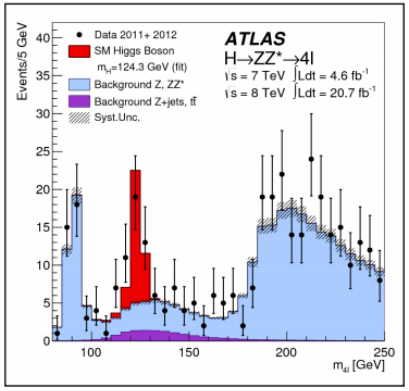
UCSC SCIPP

Kyle Cranmer

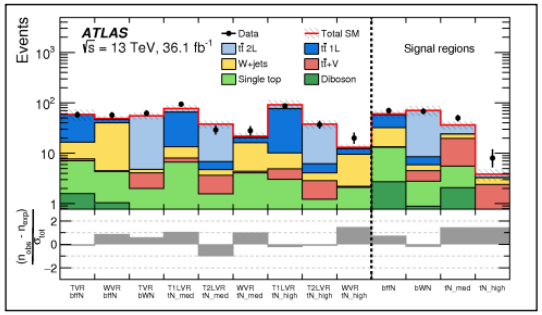
NYU

HistFactory

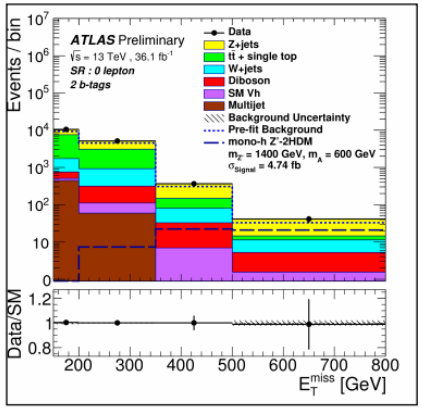
- A flexible p.d.f. template to build statistical models from binned distributions and data
- Developed by Cranmer, Lewis, Moneta, Shibata, and Verkerke [1]
- Widely used by the HEP community for standard model measurements and BSM searches



SM



SUSY



Exotics

HistFactory Template

$$\mathcal{P}\left(n_c, x_e, a_p | \phi_p, \alpha_p, \gamma_b\right) = \prod_{c \in \text{channels}} \left[\text{Pois}\left(n_c | \nu_c\right) \prod_{e=1}^{n_c} f_c\left(x_e | \vec{\alpha}\right)\right] G\left(L_0 | \lambda, \Delta_L\right) \prod_{p \in \mathbb{S}+\Gamma} f_p\left(a_p | \alpha_p\right)$$

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 bins observed in all channels
- Constraint p.d.f. (+ data) for "auxiliary measurements"
  - encoding systematic uncertainties (normalization, shape, etc)

`pyhf`: HistFactory in pure Python

- First non-ROOT implementation of the HistFactory p.d.f. template
  - DOI 10.5281/zenodo.1169739
  - pure-Python library
    - `pip install pyhf`
  - machine learning frameworks as computational backends
    - `pip install pyhf[tensorflow]`
- Alternative choice to ROOT-based HistFactory to use in the analysis pipeline of HistFitter + HistFactory + RooStats
  - Project scope: HistFactory
  - Not a replacement for HistFitter or RooStats
- Open source tool for all of HEP
  - Originated from a DIANA/HEP project fellowship
  - Not experiment specific (though designed by ATLAS physicists)
  - Used for reinterpretation in phenomenology paper [2] and gaining interest in ATLAS

Likelihood serialization and ATLAS sbottom reproduction PUB note

Makes good on 19 year old agreement to publish likelihoods

(JSON likelihoods should be on HEPData anyday)

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

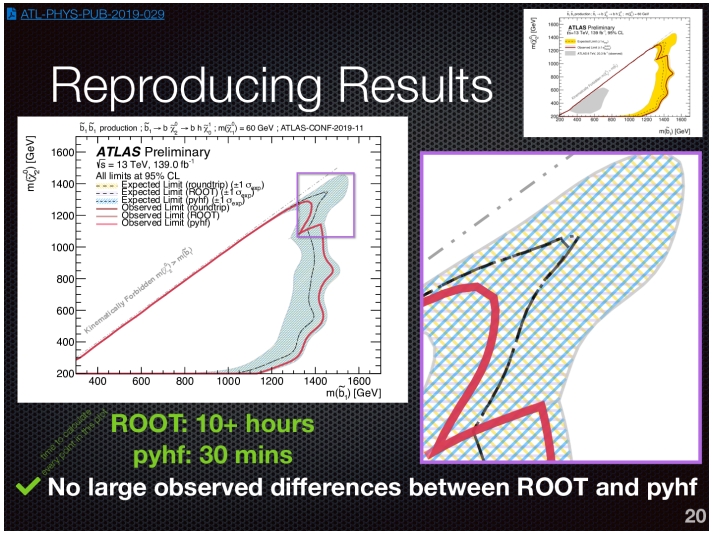
ATL-PHYS-PUB-2019-029

5th August 2019

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

The ATLAS Collaboration

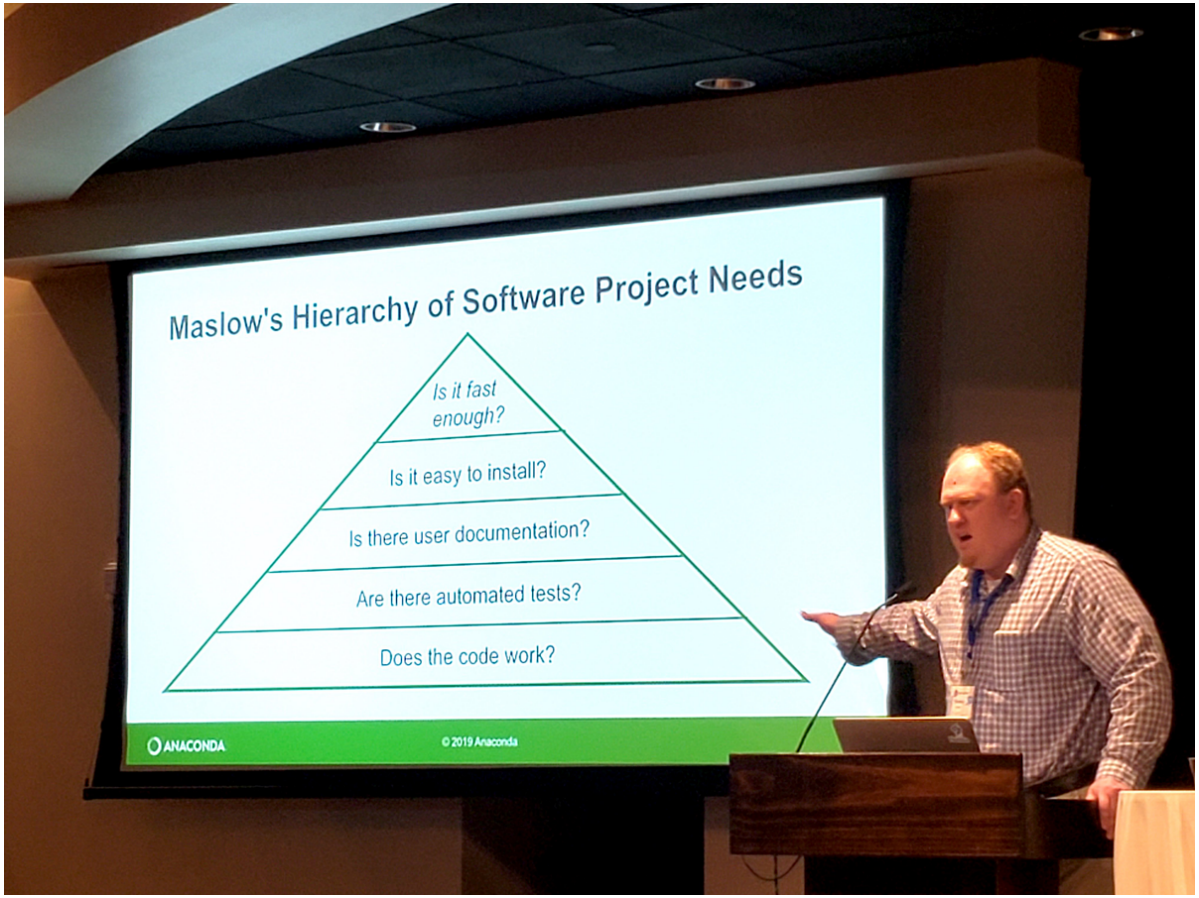
(ATLAS, 2019)



Given at DPF 2019 by Giordon Stark

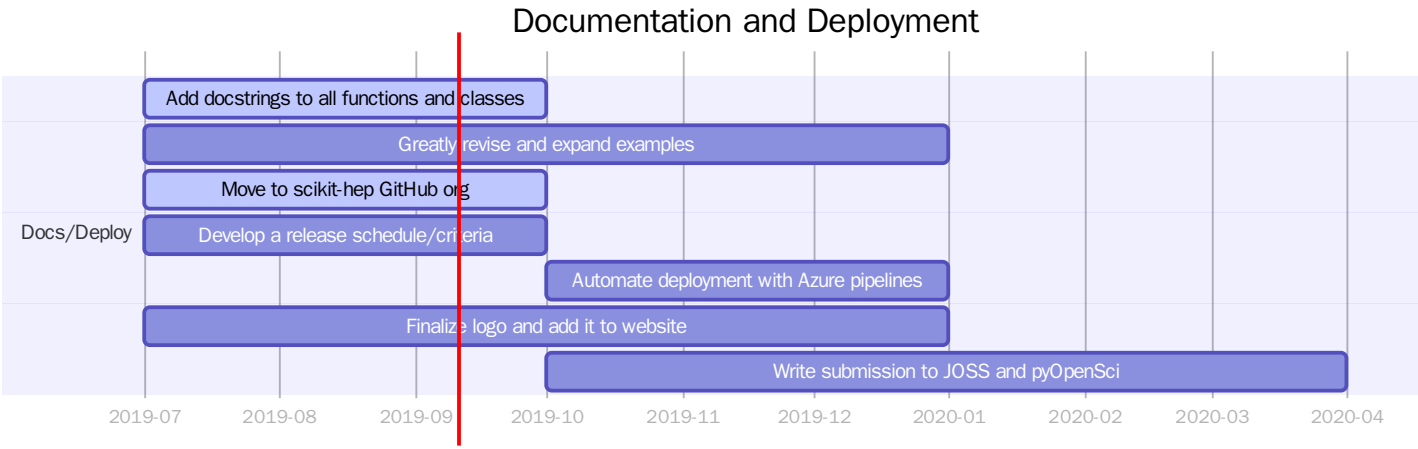
Roadmap (Issue #561)

Loosely follow Seibert's Heirarchy of Needs

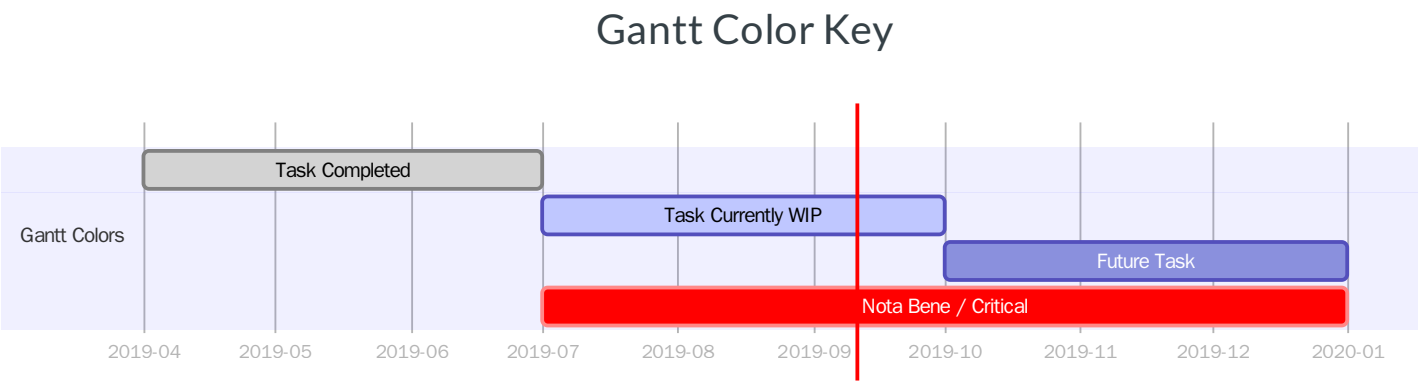


Stan Seibert (Anaconda) at SciPy 2019

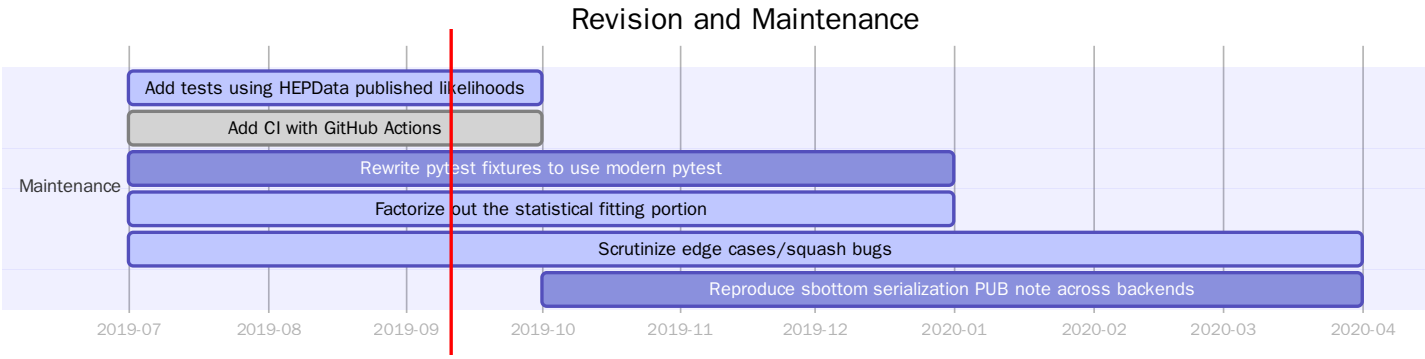
Roadmap



- Most common question/request core devs get over email is for examples

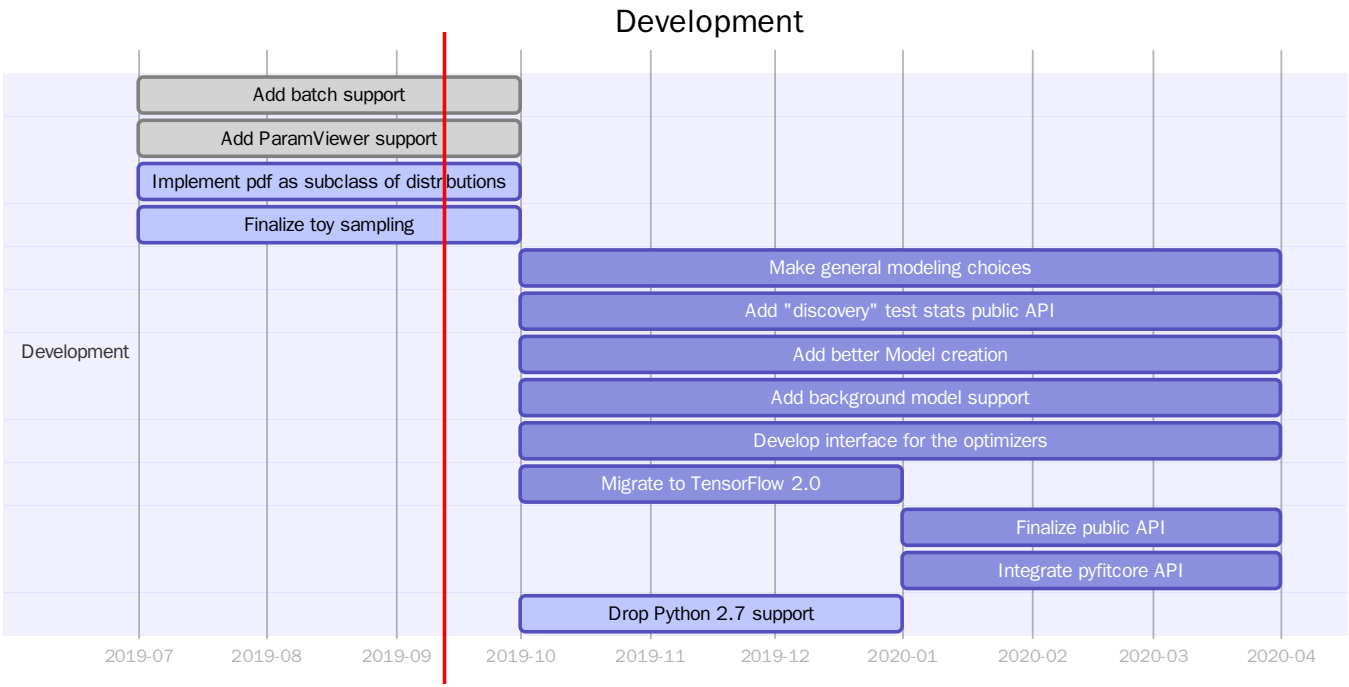


Roadmap



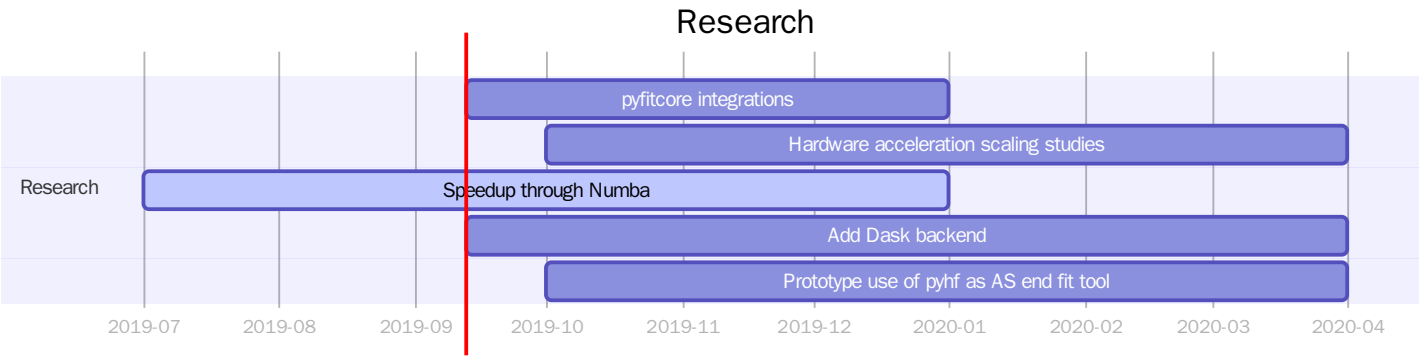
- Now that we have published likelihoods important to integrate them into testing
- Good opportunity to examine real physics case performance of all backends
- Update testing framework to take advantage of continued improvements
- With addition of GitHub Actions CI and Henry's use of Azure Pipelines for deployment reconsidering continued use of Travis CI
    - Probable to drop CI in Travis but keep CD with Travis in near future

Roadmap



- Clear, reliable public API
- Toys for when asymptotics fail in low stats regime
  - Most requested feature
- pyfitcore and zfit integrations

Roadmap

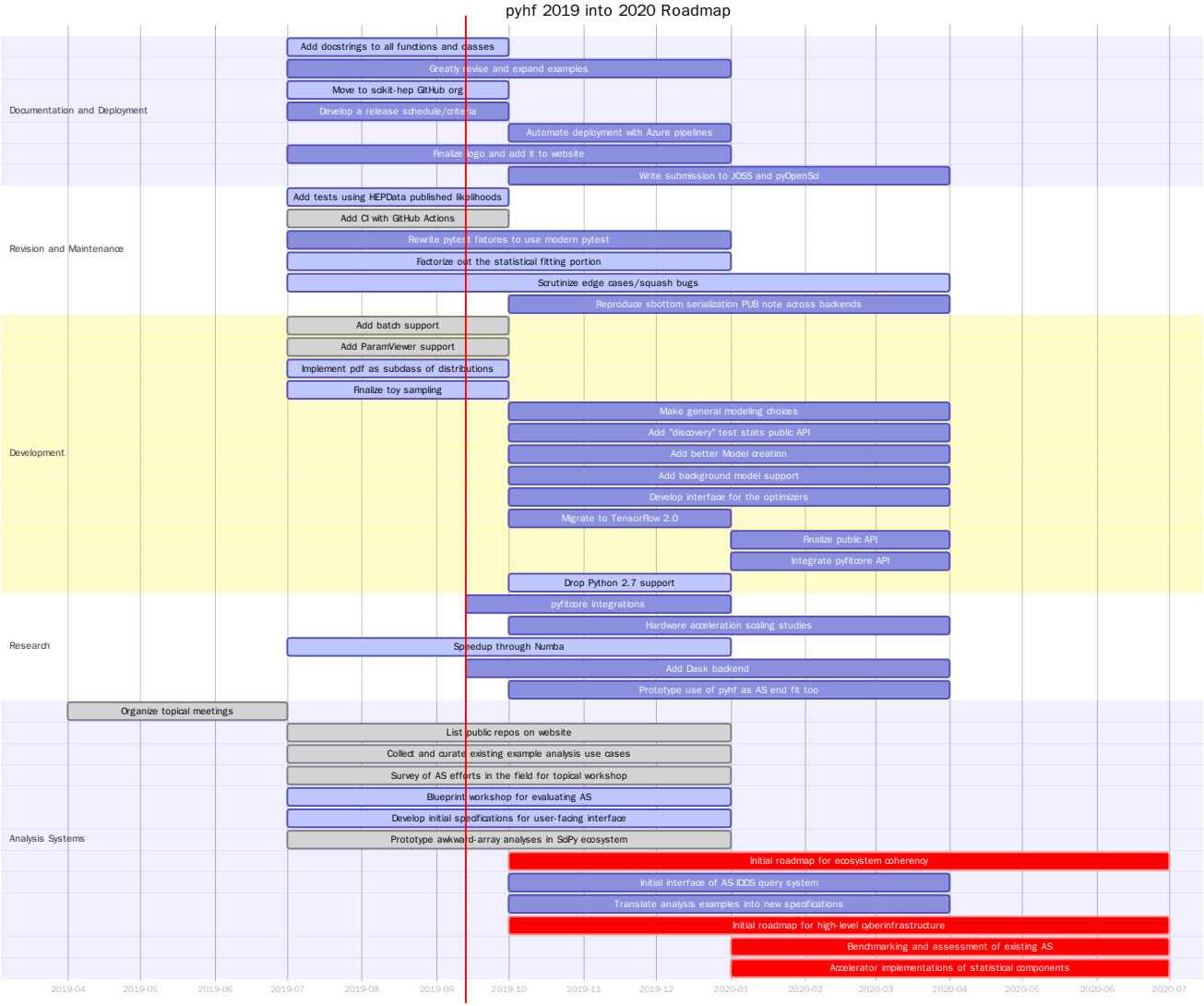


- Hardware acceleration and scaling studies highly important
- Access to Hardware-Accelerated Learning (HAL) at NCSA at Illinois
  - Ideal cluster for benchmarking scaling and performance
- Scaling with Dask could add additional benefit

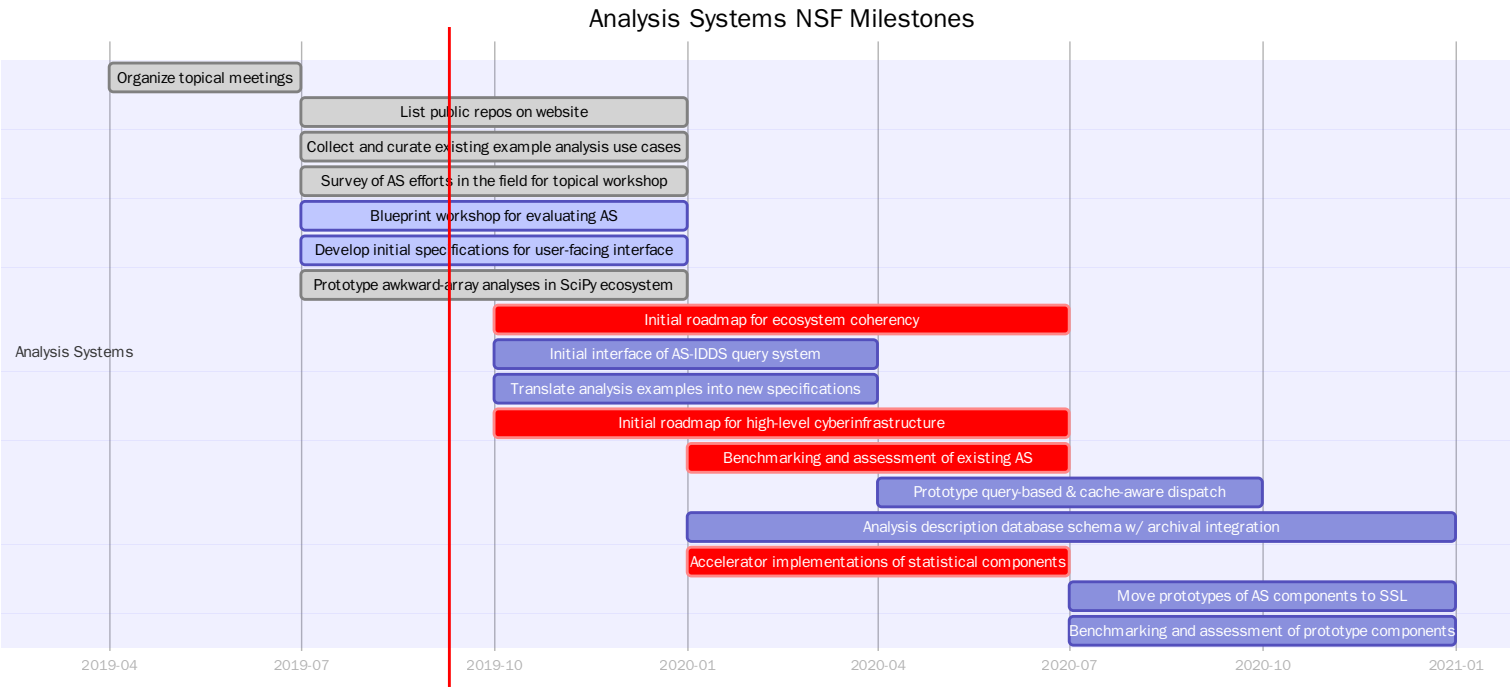
Hardware

- 16 IBM AC922 nodes
  - IBM 8335-GTH AC922 server
    - 2x 20-core IBM POWER9 CPU @ 2.4GHz
    - 256 GB DDR4
  - 4x NVIDIA V100 GPUs
    - 5120 cores
    - 16 GB HBM 2
  - 2-Port EDR 100 Gb/s IB ConnectX-5 Adapter
- 1 IBM 9006-22P storage node
  - 72TB Hardware RAID array, NFS-mounted on all nodes via IB EDR
- Storage upgrade TBD

Roadmap + Analysis Systems

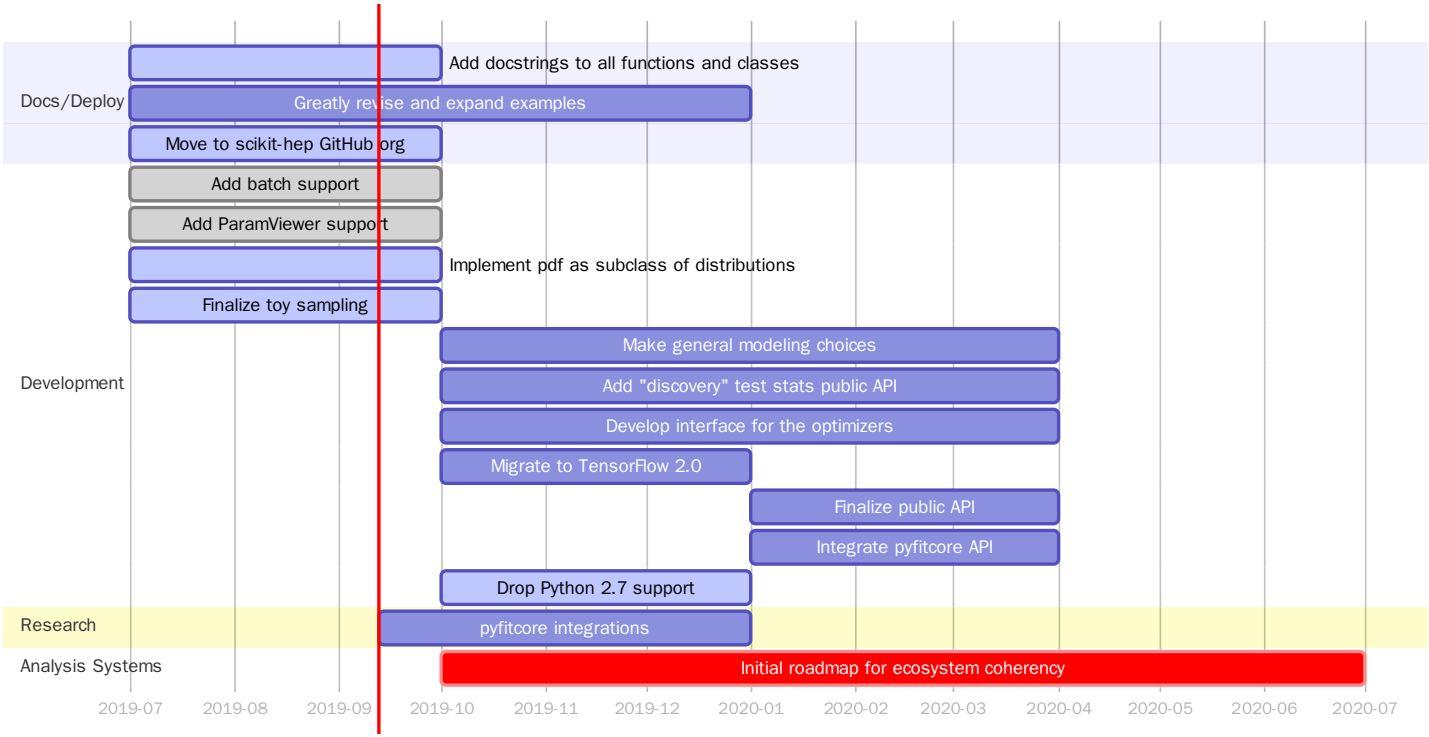


Analysis Systems Milestones



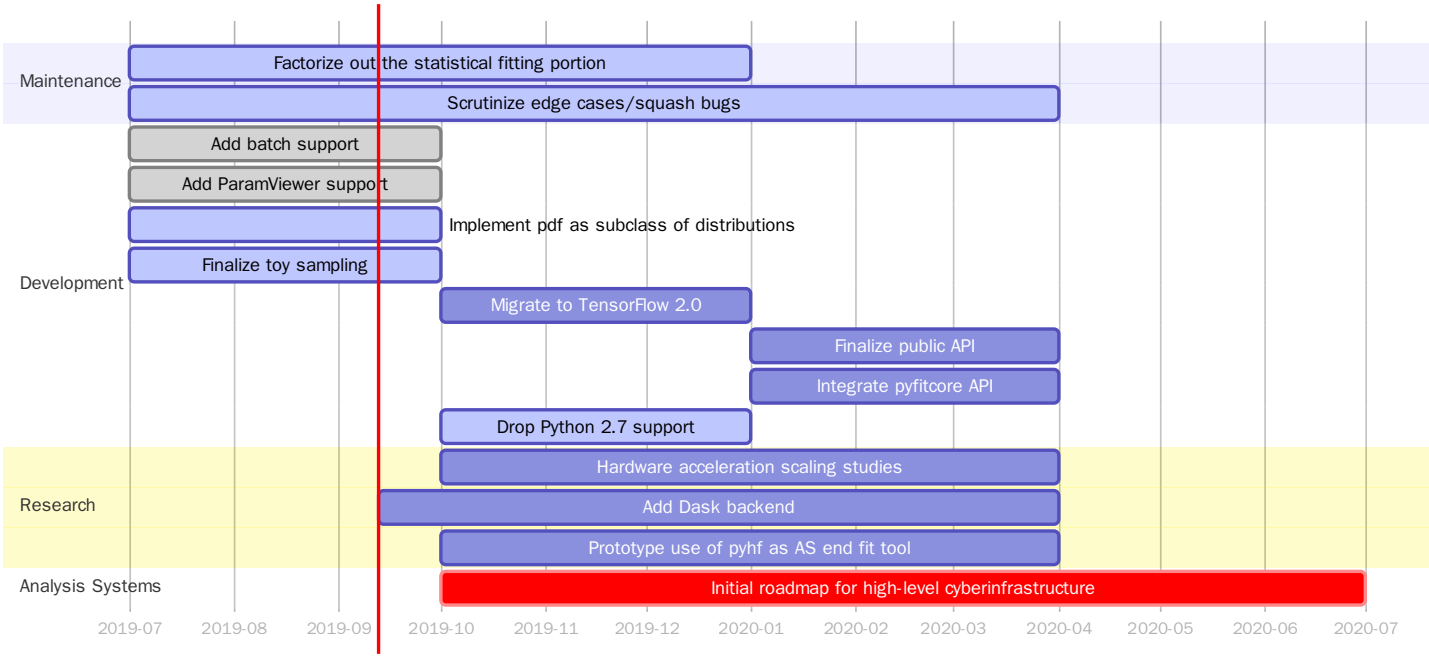
- Initial roadmap for ecosystem coherency (2019-Q4 -> 2020-Q3)
- Initial roadmap for high-level cyberinfrastructure (2019-Q4 -> 2020-Q3)
- Benchmarking and assessment of existing AS (2020-Q1 -> 2020-Q3)
- Accelerator implementations of statistical components (2020-Q1 -> 2020-Q3)

Initial roadmap for ecosystem coherency



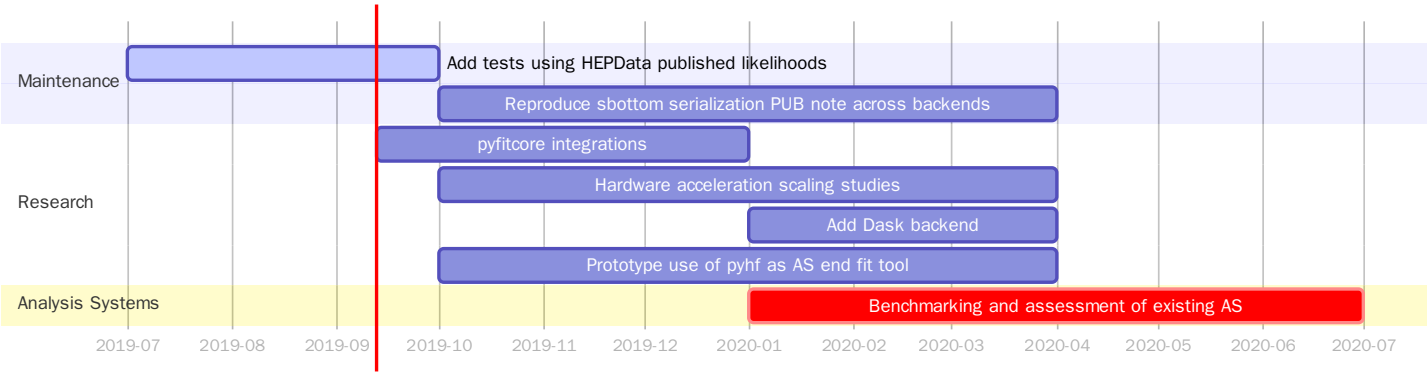
- pyfitcore integration and stable API critical

Initial roadmap for high-level cyberinfrastructure



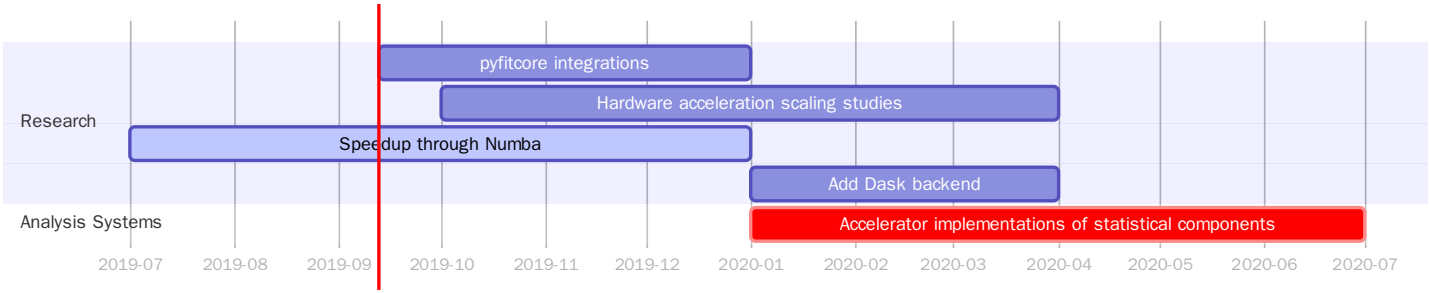
- Fitting as a service on a cluster
- Similar idea to ATLAS ML Platform (IRIS-HEP SSL) but for fits

Benchmarking and assessment of existing AS



- Case study: Use pyhf in concert with other parts of AS to replicate entire ATLAS analysis (DAOD -> plot)
- Proposal: Matthew's thesis analysis (ATLAS-CONF-2018-052)

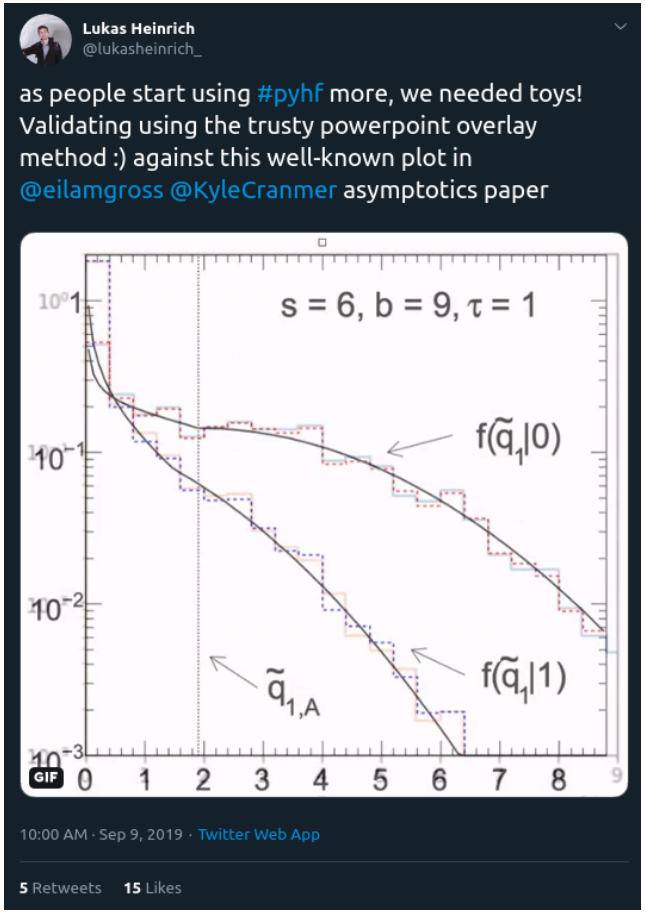
Accelerator implementations of statistical components

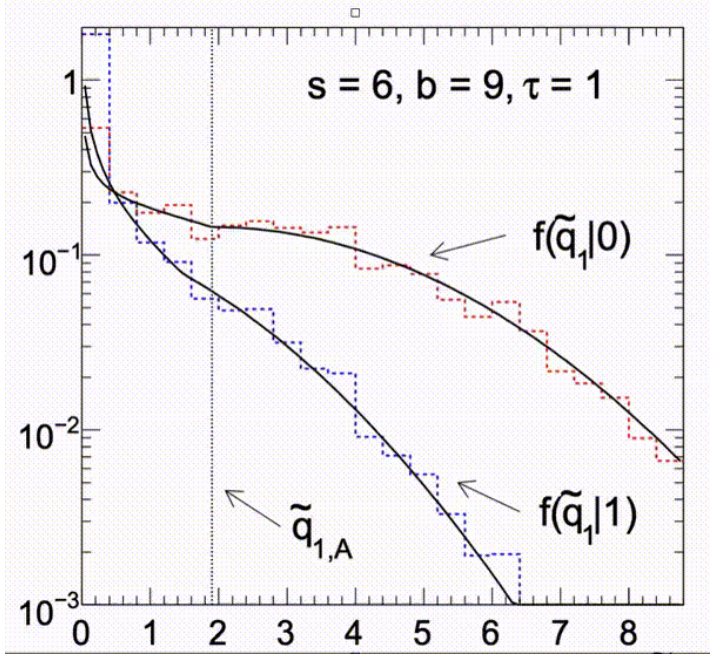


- Given pyhf research goals should be well on track

Upcoming Release v0.1.3

Toy Sampling

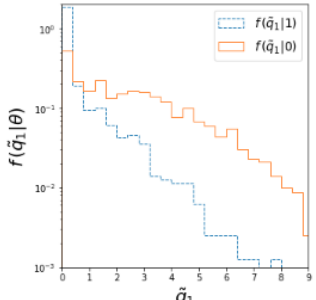




In just a few lines of code are able to reproduce Figure 5b of arXiv:1007.1727!

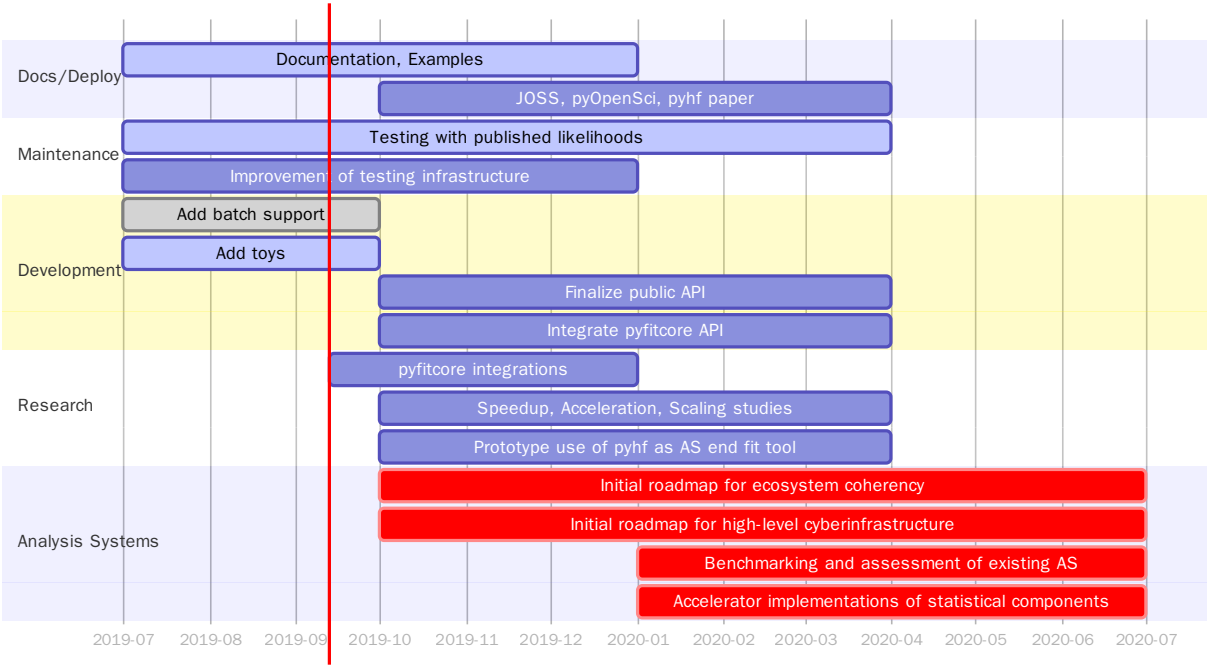
...in just a few lines

```
In [1]: import pyhf
        import numpy as np
        import matplotlib.pyplot as plt

In [2]: np.random.seed(0)

In [3]: model = pyhf.simplemodels.hepdata_like([6], [9], [np.sqrt(9)])

        signal_pars = model.config.suggested_init()
        signal_pars[model.config.poi_index] = 1.0

        bkg_pars = model.config.suggested_init()
        bkg_pars[model.config.poi_index] = 0.0

        signal_pdf = model.make_pdf(signal_pars)
        bkg_pdf = model.make_pdf(bkg_pars)

        sample_shape = (10000,)

        signal_sample = signal_pdf.sample(sample_shape)
        bkg_sample = bkg_pdf.sample(sample_shape)

In [4]: def q_mu_tilde(poi_test, data, pdf):
            return pyhf.utils.hypotest(
                poi_test, data, pdf, qtilde=True, return_test_statistics=True
            )[1][0]

In [5]: signal_qtilde = np.asarray([q_mu_tilde(1.0, sample, model) for sample in signal_sample])
        bkg_qtilde = np.asarray([q_mu_tilde(1.0, sample, model) for sample in bkg_sample])

In [6]: fig, ax = plt.subplots(figsize=(5, 5))

        ax.hist(
            signal_qtilde,
            bins=np.linspace(0, 10, 26),
            histtype="step",
            density=True,
            linestyle="dashed",
            label=r"$f\,(\tilde{q}_1|1)$",
        )
        ax.hist(
            bkg_qtilde,
            bins=np.linspace(0, 10, 26),
            histtype="step",
            density=True,
            label=r"$f\,(\tilde{q}_1|0)$",
        )

        ax.set_xlim(0, 9)
        ax.set_ylim(1e-3, 2)
        ax.semilogy()
        ax.set_xlabel(r"$\tilde{q}_1$", fontsize=20)
        ax.set_ylabel(r"$f\,(\tilde{q}_1|\theta)$", fontsize=20)
        ax.legend(loc="best", fontsize=14);
```



Summary

- pyhf development roadmap laid out for rest of the year
- Institute retreat is well needed ground for discussions on pyhf in pyfitcore
  - Discussions here will be critical for API design
- Discussions on alignment with Analysis Systems milestones important for roadmap changes



Backup

Backup Slide

Recent Talks:

- Poster at SciPy 2019 (July, 8-14th, 2019)

Upcoming Talks:

- Talk at PyHEP 2019 (October 16-18th, 2019)
- Talk at CHEP 2019 (November 4-8th, 2019)
- Poster at CHEP 2019 (November 4-8th, 2019)

References

1. 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.
2. L. Heinrich, H. Schulz, J. Turner and Y. Zhou, *Constraining A_4 Leptonic Flavour Model Parameters at Colliders and Beyond*, 2018.

The end.