

pyhf Users and Developers Workshop

2023: Community Notes

Users Session

Introduction to pyhf

Anything you would like to see? Anything confusing? Need clarification?

-

Shared Tooling

What are areas shown that tooling around pyhf has been built that you have also built, or would like to reuse?

-

Common Problems (Wednesday Morning)

What are problems or issues with pyhf that you've identified that you feel are not unique to your situation and are affecting others? What are things that are missing from pyhf that make your analyses harder to do? What are APIs that you keep reaching for, but find does not exist? What are things that are undocumented that make it hard to get started? What are areas of the docs that are not clear?

- No option for correlated nuisance parameters.
- (From Luis's talk) No way to have correlated shapesys across samples (keeping the uncorrelated bin-by-bin part). See [scikit-hep/pyhf#1967](https://github.com/scikit-hep/pyhf/issues/1967).
- [Fabian's Belle II talk](#) had good summaries of feedback in these areas. (Can we explore more here?)
 - Support for multiple-POI fits (existing GitHub Issue for adding API — <https://github.com/scikit-hep/pyhf/issues/179>)
 - Alex: Technically you can already do this
 - Fabian: The idea here is that it would be nice to add this to the API so that it could be much easier when making impact plots.
 - Alex: If you want to do a 2-D limit then this is a fundamentally different thing, which needs to get set up. This is also covered in the Cabinetry discussion ([LINK HERE](#)) that shows people how to do things.

- Giordon: Most of the time you're thinking about doing 2D limits with 2 Poles, but if you want to generalize this to multiple n Poles then this isn't clear.
- Alex: I think we do not do hypothesis tests / limits in multi-dimensional parameter space, so typically we would just look at likelihood contours there
- Fabian: In higher dimensions it doesn't necessarily make sense, but there are times when you want to be able to interface with the systematics and
- Common Naming Scheme for Plots/Inference Algorithms/Statistical Procedures
 - Carsten: We need to get a common set of terms for these plots
 - https://atlas-stats-doc-dev.web.cern.ch/atlas-stats-doc-dev/recommendations/rec_diagnostics_checks/
 - <https://github.com/iris-hep/as-user-facing/tree/master/fit-visualization>
 - Fabian/Carsten: Need to make sure that this starts with the software and then take this to the experiments as even within experiments (e.g. ATLAS) there is disparity in what these things are called between physics groups.
 - Alex: Nominally the ATLAS Stats Committee is also working to try to get names for these things.
 - "Impacts" / "uncertainty components" / "parameter ranking" / "breakdown": check out <https://arxiv.org/abs/2307.04007> for a recently proposed method with some nice theoretical properties (which is also very computationally cheap in Gaussian approximation as it does not require extra fits)
 - Fabian: Make a page in the docs that is just for these most important docs and references
 - Perhaps use <https://github.com/hepstats> and static documentation built under this org? (Giordon owns it)
- Belle II: Arbitrary functional constraints between parameters
 - There are existing GitHub Issues there
 - Interest in multi-dimensional constraint terms, which is technically out of scope for HistFactory (v1) so would require quite a bit of extension and discussion.
 - From the feedback though this was a "this would be great to have" and so understood if
 - Carsten: There's an open PR for editing constraint terms (which Giordon worked on and which did not fully work yet)
 - <https://github.com/scikit-hep/pyhf/pull/1829>
 - Carsten would suggest 4 options: Normal, Poisson, Log Normal, and None and that all of these should be possible to apply to all modifiers.

- Alex: Not sure if this works in ROOT in HistFactory. Should have some follow up on this. Carsten is interested in following up on this.
- Belle II: Functionality of grouped impact calculations
 - This is essentially a feature request, and can be followed up on in the dev session
 - This goes back also to the naming of the statistical summaries as well
- Belle II: Pruning of stat. error (bin-by-bin pruning)
 - Issue: <https://github.com/scikit-hep/pyhf/issues/662>
- Belle II: pre-/post-fit plots out of the pyhf model
 - While this exists in cabinetry people might not know about this, or they might want this in the pyhf-level API
 - Fabian: This is related to “Extend cabinetry to other input formats (pandas data-frames or python histograms)”
 - Alex: The reason for the passing of the config is for a specific key and so you can even use a pseudo-config to do this
 - Fabian: Maybe this would be an important thing to add this into the documentation for cabinetry so that
 - Lorenz: It might be a knowledge problem too, so in situations it might make sense to try to direct users in the pyhf documentation to the cabinetry documentation for visualization: Take the view that cabinetry is the visualization engine for pyhf (which is true anyway :))
 - Matthew: If people really want this in the pyhf-level API, how much work would it take to sync this between pyhf and cabinetry? At least a minor release from pyhf.
 - [Alex] I do not have strong views here of where code should live, happy to migrate things around as needed. I am wondering if this is potentially mostly a documentation issue or if there are actual concerns around installing a second dependency to get plots working.
- Belle II: Separate the hypotest and inference part (use of hepstats)
 - Is this an interface issue? Is this a feature or integration issue?
 - Giordon: There isn't much inference tooling that could be pulled out to a general inference package
- Belle II: Some documentation/introduction to asymmetric uncertainties. Good/Bad work examples
 - The idea here for “bad” work examples is to create a pathological examples where there are problems in how it is handled and then have a study which steps through how to address this and get to a solution on how to address this.
 - Example: A systematic that drives negative yields
 - Giordon: Do people know that you can zero out negative yields?
 - Alex: if this does not come up in the error message in the traceback then this really needs to be added.

- Giordon: not in the traceback - we would need to add a way to do this
<https://github.com/scikit-hep/pyhf/blob/b4fec1c35b840508fb9adcfbecedfe0f0e69cb47/src/pyhf/pdf.py#L598>) basically checking the yields at the crashing of a minimization to see if anything was negative and providing as a suggestion [easy to do and I think there's a GitHub issue for this]
 - [Alex] I just realized I have an issue for it already:
<https://github.com/scikit-hep/pyhf/issues/2093>
 - Vangelis: for reference, here is my messy utility function to fix a workspace unphysical zero or nan values:
[zeroless_ws](#)
- Alex: An important thing is that if a tool fixes everything for you then it magically fixes it for you and then you don't know how or what is happening and this leads to bad statistical modeling habits. So crashing is good here as it makes people take responsibility.
 - Fabian: It might be good to have the ability to do this, but that these things should be not be on by default and then telling people to turn these things on if they know what they're doing.
 - Matthew: I think I might more really want them to do it themselves. I want to tell people what they're doing wrong, but I do not want to be responsible for fixing other people's modeling mistakes.
- Belle II: More information for the correct procedure for toys
 - Alex: Perhaps controversial take, but we shouldn't do this as you're now opening the door to statistical advice and you shouldn't do that.
 - Assuming this might be related to
<https://github.com/scikit-hep/pyhf/discussions/2377>
 - Giordon: this is true. There are some things that pyhf can improve here but it is probably not documentation-related. More about not allowing toys to fail so harshly, and being able to understand why some toys fail by providing that information.
 - Suggestions how to deal with non closure:
 - No. This is not something that pyhf can help with.
 - Matthew: We currently have
<https://scikit-hep.org/pyhf/examples/notebooks/toys.html> and this is as much documentation as I want in the library for it. We should consider giving more examples in the pyhf user guide.
 - Giordon: Is this easy or hard to run toys at the moment?
 - Matthew: It would be cool to show how to run toys in a distributed manner with Dask.

- See <https://github.com/alexander-held/distributed-cabinetry-agc-demo/blob/main/demo.ipynb> for an example
 - Giordon has an old PR for this (Executor-based workflow <https://github.com/scikit-hep/pyhf/pull/1158>).
- Belle II: Include some of the outreach tutorial as examples, hard to find in all of the talks/tutorial the right example (sort them more by topic)
 - Part of the benefit from putting more “mature”/“advanced” examples together (think Vagelis’s talk from the Monday session) would be to also provide clear explanations and summaries of why you would want to do particular inference tasks (e.g. a generic fit, vs. a hypothesis test).
 - Would be good to both extend the documentation in the user guide as well as to collect these more advanced examples centrally.
- Volker: Parameters and parsets are pretty confusing. Better documentation would be great here.
 - <https://github.com/scikit-hep/pyhf/issues/1024>
 - Will want to give focus on documentation during developer portion of the workshop in addition to code
-

New Ideas (Wednesday Afternoon)

What are ideas that you had from presentations at the workshop that you think are important to pursue?

- Custom modifiers [PR #1991](#)
- Vangelis: Multiple workarounds so far to get features working that aren’t implemented
 - In cases where there is 1 systematic / uncertainty but in each region of the analysis it is given with a different weight. Work around: Make them separate nuisance parameters with different names, and then doing special histogram renaming. When discussed with Alex, how to combine systematics was brought up so that there could be 1 systematic that can represent the combination of multiple systematics
 - Relevant example: Parton Distribution Function (PDF) scale uncertainties
 - New features: (possibly) Additional ways of post processing the workspace, specifically systematic combination
 - Reference discussion on IRIS-HEP Slack: <https://iris-hep.slack.com/archives/C01ETQ190F4/p1699374460957609>
 - In the situation in which the types of systematics are different it could be not-easy to implement things

- Addition of an overall normsys like in HistFitter
Called: overallNormSys or normHistoSys (reference: <https://arxiv.org/abs/1410.1280>, page 18)
 - So this would be something that would need to be implemented in Cabinetry probably, given that this isn't something that is part of the HistFactory standard, but an extension.
 - More info: [1],[2] (ATLAS internal)
- Workspace Manipulations/Post-processing
 - See Vangelis' first feature request above (how to combine systematics)
 - Volker: Remove modifiers in a given region: <https://github.com/scikit-hep/pyhf/issues/2374> (low-hanging fruit)
 - Use case: In combination there are workspaces that need to be cleaned up by removing modifiers that might not need to be there (e.g. the modifier should only affect the signal, but in the workspace that had been converted from another tool the modifier then affects the signal and the background)
 - Could also affect situations in which there are huge analyses where it could be advantageous to do reductions
 - Example: In Higgs → WW there were some situations of over 1 million histograms
 - Volker: Also useful to remove modifiers by the scale/level of impact
 - Should be an open issue on this in Cabinetry: <https://github.com/scikit-hep/cabinetry/issues/311>
 - Should pruning happen during building (at cabinetry level) or at runtime.
 - This is maybe something that isn't necessarily clear RE: what is technically feasible vs. what is perhaps preferable (prune at runtime).
 - Belle II: Have had real situation (similar to H→WW in ATLAS) where trying to build the non-pruned workspace would have taken way too long (unfeasibly long, not many minutes, but many many hours)
 - To be more memory efficient: Could also try to do modifier reduction by removing the least impactful
- Interest in being able to do extrapolation of post-fit results to another
 - Vangelis: <https://github.com/scikit-hep/cabinetry/discussions/431>
 - Fabian: It is important that this isn't user implemented but that this exists in a library (either pyhf or cabinetry) that can be validated so that this significantly reduces the work on the user. Otherwise this can be quite a lot of work and take a long time to validate that this is done correctly.
 - [Alex from outside] how useful is the current cabinetry API here / what other things would be needed?
 - Vangelis: In this particular case (as the above API allows this to happen), but maybe this could be added to the docs to be made more clear and advertise this.

[\[https://github.com/scikit-hep/cabinetry/discussions/431#discussioncomment-6801649\]](https://github.com/scikit-hep/cabinetry/discussions/431#discussioncomment-6801649)

- This is shown in the notebook talk that Vaneglis gave on Monday at the workshop. This represents on and off 2 years of work, which is mostly from trying to understand what TRexFitter and HistFitter would be doing and then translating this into pyhf and cabinetry.
- Matthew: What is more preferable? Higher level APIs or a collection of “recipes” for common analysis tasks.
 - Everyone: Recipes are way more preferable
 - Giordon: (internal notes): TensorViewer / parameter stitching
- Peter: In the JSON representation you have different channels, but it might be useful in the serialization to unroll everything. [Giordon: remove the channel boundaries between bins]
 - Matthew/Giordon: Not fully clear on how this is useful in a serialization format.
 - Carsten: Unclear on how to apply the overall sys as this is for a specific region, as if you concatenate you can end up with two histosys on the same channel with the same name. Beyond this, everything should be fine.
- Backends Use Poll:
 - Numpy:
 - PyTorch: 1 (Volker)
 - Tensorflow:
 - Jax:
- Optimizer Use Poll:
 - Scipy: 1
 - Minuit: 7
- Lukas:
 - Extended constraint terms
 - With custom modifiers might be able to have more complex constraint terms
 - Use case: Background estimation with Gaussian processes you could have a multi-bin constraint with a correlation between the bins
 - Carsten: If you have this with custom modifiers then perhaps this could allow you to do unfolding.
- Adoption wise:
 - Carsten: In ATLAS there's a lot of variation. In SUSY there's lots of use and variations. In Exotics there are some, but not a lot. In Higgs, and Standard Model this is super rare, as things can get very custom very fast (e.g. analytical shapes). As this seems pretty static, more motivation for HS3 for groups to compare.

Developers Session

Birds of a Feather (BoF) Topic Suggestion

- Documentation
 - Sign-Up: Giordon, Lorenz
- Optimization (Speed, Efficiency)
 - Sign-Up:
- Minimization
 - Sign-Up:
- Autodifferentiation
 - Sign-Up: Alex, Matthew, Rishabh, Lorenz
- Adoption of HS³
 - Sign-Up: Giordon, Carsten, Alex, Matthew, Rishabh
 - [Alex] specifically here we should discuss whether it is desirable to converge on a single JSON representation for HistFactory workspaces that is aligned between HS3 and pyhf: this goes further than having interoperability (e.g. via a converter script) and is more aimed at stopping proliferation of very similar formats
- Visualization tools
 - Sign-Up: Carsten, Lorenz, Alex, Volker, Giordon
- Tensor backend support: Array API standard or Keras v3.0?
 - Sign-Up: Giordon
 - c.f. <https://github.com/scikit-hep/pyhf/issues/2249>

Adoption of HS³

- <https://github.com/scikit-hep/pyhf/pull/2397>
- Draft v0.2 of HS3 is effectively released this week and there is an implementation in ROOT v6.30 as of last week which is waiting for a patch
 - Julia implementation is in draft status, zfit is committed to moving forward
- Giordon's PRs that are relevant:
 - <https://github.com/scikit-hep/pyhf/pull/1978> (schema migration)
 - <https://github.com/scikit-hep/pyhf/pull/1829> (configurable constraints per-modifier)
 - <https://github.com/hep-statistics-serialization-standard/hep-statistics-serialization-standard/discussions>
- Relevant HS3 Information for moving forward:
 - <https://github.com/hep-statistics-serialization-standard/hep-statistics-serialization-standard/discussions/30> (differences to pyhf JSON)
 - Current example:
https://github.com/root-project/root/blob/master/tutorials/roofit/rf515_hfJSON.json

Visualization Tools

- Workspace Explorers:
 - Extend the backend to rely on pyhf more
 - All work is done so far by Volker on free time
 - Talk about reduction of overlap of effort in general (e.g. with Abe's implementation)
- Cabinetry visualizations
 - Not fully complete set of visualizations (but good)
 - Biggest problem: Customization of visualizations is very hard. Expose matplotlib code to interface with, but this requires that people know
 - Carsten has the ability to write in TikZ via Python (generically cool), though wants to know if there could be other viz backends
 - Alex/Matthew: No. Alex: Interface is hard.
 - What tools are used for publications?
 - matplotlib + mplhep: Matthew, Vangelis, Giordon, Mason, Rishabh
 - ROOT: Alex, Giordon (<https://github.com/kratsg/takeovertheworld>), Rishabh, Mason
 - gnuplot: Giordon
 - TikZ: Carsten (https://gitlab.cern.ch/cburgard/RooFitUtils/-/blob/master/python/pgfplotter.py?ref_type=heads), Giordon
 - d3/triple: Giordon
- Vangelis
 - SUSY exclusion contour plotting functions: <https://gitlab.cern.ch/-/snippets/2927>
- Giordon:
 - Reproducible contour plots (from TGraph) <https://gist.github.com/kratsg/4ff8cb2ded3b25552ff2f51cd6b854dc>

Automatic Differentiation

Developer Sprints Ideas

- Add new examples to the docs
- Explore adopting Keras v3.0 as a new tensorlib system
 - C.f. <https://github.com/scikit-hep/pyhf/issues/2249>
- Explore adopting array-api-compat as a new tensorlib system
 - C.f. <https://github.com/scikit-hep/pyhf/issues/2249>