

IRIS-HEP Fellowship Proposal

Project: Hist Serialization and Interactivity

Project Area: Analysis Systems

Aman Goel

aman.goel185@gmail.com | University of Delhi, India

Mentor: Henry Schreiner

henryfs@princeton.edu | Princeton University

1 Background

Hist is a powerful Histogramming tool for analysis based on boost-histogram (the Python binding of the Histogram library in Boost). It is a friendly analysis-focused project designed for Python 3.7+ that uses boost-histogram as a backend to do the work, but provides plotting tools, shortcuts, and new ideas.[1] It provides various features for histogramming, such as:

- Augmentation of axes with names
- Augmentation by class with simpler construction
- QuickConstruct
- UHI+ implementation
- Quick plotting routines
- Stacks
- Extended histogram features
- New modules
- In-notebook representation[2]

There are several other improvements that can be made to the library and new features that can be implemented which would help in its advancement.

Additionally, the uproot-browser is a proposed library that would be based on textual and can view histograms. It would have a tree-like browser on the left, a plotext based plot viewer on the right and a potential control box below the plot.

Its aim is to enable a user to browse and look inside a ROOT file, completely via the terminal. The current base has a command-line utility to view the contents of a ROOT file as a tree and can be developed further into a fully functional library.

2 Proposed Project

This project proposes the development of the Hist library further via the implementation of new features, bug fixes, test coverage and documentation. This would help in improving the performance, usability, accessibility and scope of functionality of the library. Further, it proposes the development of the uproot-browser into a fully functional library to help achieve its aim.

The plan of action and the features that would be implemented are as follows:

1. Documentation Improvement and Coverage

- Add documentation for undocumented features
- Complete porting of *boost-histogram* documentation
- Increase coverage and polish documentation to improve user experience
- Would address issue #265

2. Interpolator Integration

- Current interpolation function works via *SciPy*[3]
- Direct implementation via a class would be better
- Add support for user-designed interpolators
- Add shortcuts for common interpolators
- Would address issue #165

3. Development of uproot-browser

- Add functionality to the base of *uproot-browser*[4]
- Implement as a *textual*[5] app with a tree-like browser and a plot viewer
- Eventual implementation of a control box below the plot
- Would help to browse *ROOT* files via terminal

4. Serialization and Interchangeable Formats

- Implementation of serialization, possibly via a JSON like standard format
- Would help improve in interaction and interoperability with *ROOT*
- Serialization through *Aghast* would help solve issue #2
- Would be described in *UHI* and implemented in *Hist/boost-histogram*
- Would address discussion #351

5. Yoda (and HEPData) ASCII File Formats Support

- Serialization support would extend to integrating with *YODA*
- Would help read Histogram data manually as an ASCII file
- Possible implementation of a general reader if all types can't be integrated
- Extend support to *HEPData* YAML files if feasible
- Would address issue #218

6. Improved Reprs

- Improve Repr Speed of 2D Histograms
- Reprs could use an embedded bitmap
- Implement Reprs with better details and information
- Provide configurable options to give a better output
- Would address issues #112, #198 and #209

7. Fill from Awkward Arrays

- *Awkward Arrays* would be taken in `.fill()` intelligently
- `ak.flatten` can be used to merge all of the numbers in an *Awkward Array* into a 1-dimensional array
- `ak.fields` and `ak.unzip` can be used to generically get field names and field arrays in the same order
- Would address issue #120

8. Statistical Functions

- Implementation of something similar to *TEfficiency* from *ROOT*[6]
- Used to handle efficiency histograms
- Would provide statistical methods for calculating frequentist and Bayesian confidence intervals
- Would compute the calculation of efficiencies and their uncertainties

3 Timeline

Time Zone: Indian Standard Time - UTC +5:30

Commitments: I do not have any prior commitments or any planned vacations during the duration of the fellowship. I would be able to work full time and continuously devote 5-6 hours each day (~ 40 hours per week) for the entire period. My work hours are flexible, and I will be present for all teleconferences/discussions, if any, by adjusting my time to that which would suit the mentor.

Exams: I would not have any written or practical exams this semester since this is the last semester of my undergraduate degree and the fellowship itself would account for all the academic credits.

Week(s)	Plan of Action
1 - 2	<ul style="list-style-type: none">- General bug fixes and maintenance- Fix bugs such as issues #337, #330 and #260- Implement optional <i>SciPy</i> requirement to address issue #358- Discuss the plan of action with the mentor and the community
3	<ul style="list-style-type: none">- Work on the implementation of <i>Documentation Improvement and Coverage</i>
4	<ul style="list-style-type: none">- Work on the implementation of <i>Interpolator Integration</i>- Write tests for the functions and document the progress
5 - 7	<ul style="list-style-type: none">- Work on the <i>Development of uproot-browser</i>- Write tests and document the progress
8 - 11	<ul style="list-style-type: none">- Work on the implementation of <i>Serialization and Interchangeable Formats</i>- Write tests and document the progress- Clean up previous code and fix bugs, if any- Keep buffer time in case of changes in the timeline
12 - 13	<ul style="list-style-type: none">- Add <i>Yoda (and HEPData) ASCII File Formats Support</i>- Write tests and document the progress
14	<ul style="list-style-type: none">- Work on the implementation of <i>Improved Reprs</i>- Write tests and document the progress
15 - 16	<ul style="list-style-type: none">- Work on the implementation of <i>Fill from Awkward Arrays</i>- Write tests and document the progress
17 - 18	<ul style="list-style-type: none">- Work on the implementation of <i>Statistical Functions</i>- Write tests and document the progress
19 - 20	<ul style="list-style-type: none">- Buffer Period and Wrap up- Work on Improving <i>Documentation</i>- Clean up code and fix bugs, if any

4 Student Background

About: I am Aman Goel, a senior from the University of Delhi, India currently pursuing a B.Tech in Information Technology and Mathematical Innovations with a minor in Management and Economics. I have been passionate about technology, space and physics since I was a kid. I am proficient in Python and volunteered in Stanford University's Code in Place[7] 2021 course. I have also previously worked with Defence Research and Development Organisation, India.[8]

Relevance: I was first initially interested in the Hist project in its early developmental stages. It lies at the intersection of physics, technology and mathematics, which is an ideal scenario for me. I had the wonderful opportunity to work on this project as an IRIS-HEP fellow[9] in summer 2021 under the mentorship of Henry Schreiner. I would love to work on Hist further, develop the uproot-browser and contribute to the HEP community while it helps me to learn and grow.

I also was a speaker at PyHEP 2021[10] in which we presented the Hist project and then helped organise a sprint for Hist at SciPy 2021.[11] My prior experience with Matplotlib, statistical analysis, Linux, git and open source would aid me to pursue the project too. I believe I have the relevant experience, skill-set and passion to work on the project.

Furthermore, I want to pursue my higher education in the field of physics and computing and I believe this fellowship would guide me and help me take a step towards the correct path.

References

- [1] Hist 2.5.3.dev9 gf29581f. (n.d.). Retrieved from <https://hist.readthedocs.io/en/latest/>
- [2] Scikit-Hep. (n.d.). Scikit-hep/hist: Histogramming for analysis powered by boost-histogram. Retrieved from <https://github.com/scikit-hep/hist>
- [3] Interpolation (scipy.interpolate)¶. (n.d.). Retrieved from <https://docs.scipy.org/doc/scipy/reference/interpolate.html>
- [4] Henryiii. (n.d.). Henryiii/uproot-browser. Retrieved from <https://github.com/henryiii/uproot-browser>
- [5] Textualize. (n.d.). Textualize/textual: Textual is a TUI (Text User Interface) framework for Python inspired by modern web development. Retrieved from <https://github.com/Textualize/textual>
- [6] TEfficiency Class Reference. (n.d.). Retrieved from <https://root.cern.ch/doc/master/classTEfficiency.html>
- [7] Code in Place Application. (n.d.). Retrieved from <https://codeinplace.stanford.edu/>
- [8] Defence Research and Development Organisation - DRDO, Ministry of Defence, Government of India. (n.d.). Retrieved from <https://www.drdo.gov.in/>
- [9] Aman Goel - IRIS-HEP Fellow. (2021, October 21). Retrieved from <https://iris-hep.org/fellows/amangoel185.html>
- [10] PyHEP 2021 (virtual) Workshop. (n.d.). Retrieved from <https://indico.cern.ch/event/1019958/contributions/4430375/>
- [11] Sprints. (n.d.). Retrieved from <https://www.scipy2021.scipy.org/sprints>