IRIS-HEP Fellow Project Proposal
HS3 Support for Combine: Provide HS3 Support for the Combine Statistical Analysis Tool

Hannah Havel
Northern Illinois University
April 30, 2025

Background

The Combine tool is widely used across the CMS collaboration for statistical inference, offering a standardized way to share models and perform statistical analysis. It is important in making complex statistical analysis more accessible and user-friendly. The Combine analysis software is built around the Root, RooFit, and RooStats packages [1]. They allow for the modeling of 'event data' distributions, but the current structure for publishing models presents challenges–the use of binary files makes sharing and interoperability difficult, especially when trying to interact with different fitting frameworks. The HEP Statistics Serialization Standard (HS3) defines a human- and machine-readable format for statistical models and results used in HEP. It aims to enable publishing likelihoods, improve interoperability between frameworks, and reduce dependence on legacy software [2].

Integrating HS3 into Combine would address the drawbacks of using Combine alone. The "full" approach starts by generating a RooFit workspace from the datacard, the file that describes everything needed for a statistical analysis, using the text2workspace.py script, which combines the datacard and ROOT file. An attempt is made to convert the workspace to JSON (HS3), but this conversion will fail for several custom Combine classes, such as ProcessNormalization [3]. To fix this, an importer and exporter are written and implemented for the class, and text2workspace.py is updated to accept JSON input. Using the importer, the JSON file is converted back into a RooFit workspace to run the fit, allowing a comparison between the traditional method and the JSON-based approach.

An initial assessment of costs and benefits revealed that adopting HS3 would enable publishing models in a non-binary format and eliminate the need for maintaining backward compatibility with custom PDFs [3]. However, the current format is not necessarily more readable than a traditional datacard and would require additional tooling to improve usability and interoperability with other fitting tools would still require significant adaptation effort.

However, a hybrid approach would still offer the same pros as the "full" approach, while decreasing the number of classes that are necessary to implement importers and exporters, and reducing the amount of code that must be modified in the class that builds the physical model. Instead of fully replacing workspaces, this approach requires implementation of importers and exporters for only selected parts of the model. This is the approach I will work on during my time as an IRIS-HEP Fellow.

Timeline

- **Weeks 1-2:** Become familiar with HS3, Combine, and Combine packages, and get set up on LXPLUS. Implement the lines of code from the post 6th CAT hackathon on GitLab to explore the idea of replacing datacard histogram pointers with HS3 JSON pointers to minimize code changes and avoid binary files.
- **Weeks 3-4:** Understand simple datacards without shapes. Run basic examples to validate existing JSON compatibility.
- **Week 5:** Increase the complexity of data cards by turning on shapes ** FAKE. Determine if new importers and exporters are necessary, and if so, how they should be implemented.
- **Weeks 6-8:** Extend support for realistic shape-containing datacards (TH1 and RooDataHist in ROOT files). Update text2workspace.py to accept HS3 JSON shape inputs alongside .root and .csv formats.
- **Weeks 9-10:** Implement additional importers/exporters to support PhysicsModel components in JSON. Continue to refine the “hybrid” approach workflow.
- **Weeks 10-12:** Make steps toward the “full” approach of integrating HS3 with Combine. Also, assess what other directions are possible given the outcomes of previous weeks.

References

[1]: https://arxiv.org/abs/2404.06614
[2]: https://github.com/hep-statistics-serialization-standard/hep-statistics-serialization-standard
[3]: Project Introduction Presentation
[4]: https://cms-analysis.github.io/HiggsAnalysis-CombinedLimit/latest/