IRIS-HEP Fellowship Proposal

Vladimir Ovechkin

Duration: June 2020 - September 2020
WBS: Analysis Systems (AS)
Project: Integrating MadAnalysis and pyhf into RECAST_cli
Funding Period: 12 weeks in Summer/Fall 2019

RECAST[1, 2] is a framework for reinterpretation of analyses prepared at the LHC using computational workflows (referred to as yadage workflows). These workflows can be run locally on the researcher's computer or on REANA, a cloud-native distributed application. It consists of two services: a public “Frontend” and private “Control Plane" web-services for managing reinterpretation requests from the phenomenology community. For the execution of a reinterpretation, a new candidate signal model must be generated and passed to a suitable implementation of the event selection and statistical inference logic, which is expressed as a workflow to be processed locally or on a cloud-based workflow service such as REANA[3].

RECAST-workflow[4] aims to replicate this system for truth level reinterpretations, which run significantly faster with lower accuracy. Furthermore, RECAST-workflow has the added capability of generating new workflows using encapsulated “subworkflows” for each step of the workflow process mentioned above (generation, selection, analysis). RECAST-workflow maps subworkflow input and output file formats to generate valid combinations of subworkflows that are runnable complete workflows. These combinations can then be filtered by common inputs (i.e. analysis id).

The benefit of this new system is the ability to quickly construct new truth-level reinterpretations to determine which regions of phase space would be interesting for a full reinterpretation that is much more computationally expensive and difficult to use. RECAST-cli (command line interface for RECAST-workflow) has been previously developed to provide a user interface for creating and executing new workflows. In RECAST-workflow’s current state, the truth reinterpretation workflows only run on the user’s local machine using RECAST-cli. The goal of this project is three fold: add new options for the selection step of RECAST-workflow besides Rivet (e.g. MadAnalsyis), add alternative statistical tools (e.g. pyhf), and to run the workflows in REANA in the cloud.

Under the local supervision of Shih-Chieh Hsu (UW) and technical guidance from Lukas Heinrich (CERN fellow) and Alex Schuy(1st year PhD at UW), Vladimir Ovechkin will work on expanding the RECAST-workflow options for statistical analysis and ensuring the workflows can be run on REANA. Vlad’s work will be made available within the public software repositories of RECAST [5] (including comprehensive user manuals).

Task 1: Create and Test MadAnalysis5 Subworkflow for RECAST-workflow

- **Week 1:** Study MadAnalysis5 (Ma5) using tutorials referenced on Ma5 homepage. Specifically, document system dependencies and install procedures for running in normal and expert modes to assist in creation of Docker image later. Confirm knowledge by running analysis on sample data to build plots and get upper confidence limits.

- **Week 2 - 3:** Build Ma5 Docker image using documented dependencies from before. Also, include a python script in this image that can run the analysis process automatically without user inputs. Test this image by building and running from the command line with sample data to reproduce the same results from weeks 1 and 2.

- **Week 4 - 5:** Add subworkflow to RECAST-cli by creating yadage workflow for running Ma5 Docker image and specify input and output file interfaces. Run full end-to-end workflow (from generation of data to analysis) using a workflow generated by RECAST that includes this subworkflow to reproduce results from before.

Task 2: Create and Test Pyhf Subworkflow for RECAST-workflow

- **Week 6 - 7:** Follow through pyhf tutorials and implement simple analysis to understand components of pyhf JSON workspaces. Create plots for several different selections of particles.

- **Week 8 - 9:** Complete subworkflow by finishing incomplete python script for pyhf Docker image. Run this script from inside a docker container to generate the same results from the previous weeks. Then, test with end-to-end workflow to verify subworkflow is compatible with the rest of RECAST-cli.

Task 3: Run End-to-End workflow in REANA cluster using RECAST-cli

- **Week 10 - 11:** Read online instructions for deploying jobs to REANA cluster to simultaneously run several analyses with different parameters at once and document this process.

Task 4: Write Implementation Documents for RECAST-cli and RECAST-workflow

- **Week 12:** Write documentation in readthedocs.io and link to it from the github. All code will be publicly available in recast-hep github organization: https://github.com/recast-hep

References

[1] K. Cranmer, I. Yavin, “RECAST: Extending the Impact of Existing Analyses”, JHEP 1104:038, 2011.

[2] K. Cranmer and L. Heinrich, “Analysis Preservation and Systematic Reinterpretation within the ATLAS experiment,” J.Phys.Conf.Ser. 1085 (2018) no.4, 042011.

[3] REANA: http://reanahub.io/

[4] Schuy, Alex, “Extending RECAST for Truth-Level Reinterpretations”, arXiv:1910.10289 2019.

[5] RECAST: https://github.com/recast-hep/recast-workflow