Rethinking final analysis stages

Team: Kyle Cranmer and Alexander Held on behalf of Analysis Systems
Institution: New York University

Overview

An **overhaul of established approaches** to analyses at the LHC is needed to meet the challenge of handling an **order of magnitude more data** expected from the High-Luminosity LHC.

Existing software in use by the LHC to perform binned template fits is typically in the form of **monolithic frameworks**, and often **not available for use outside** the experiments.

IRIS-HEP approaches the challenge with a **modular workflow** focused on **well-defined interfaces**. It is **containerized** for analysis preservation and reusability. The modular nature allows for natural integration of tools developed within IRIS-HEP and beyond.

Modularity & interfaces

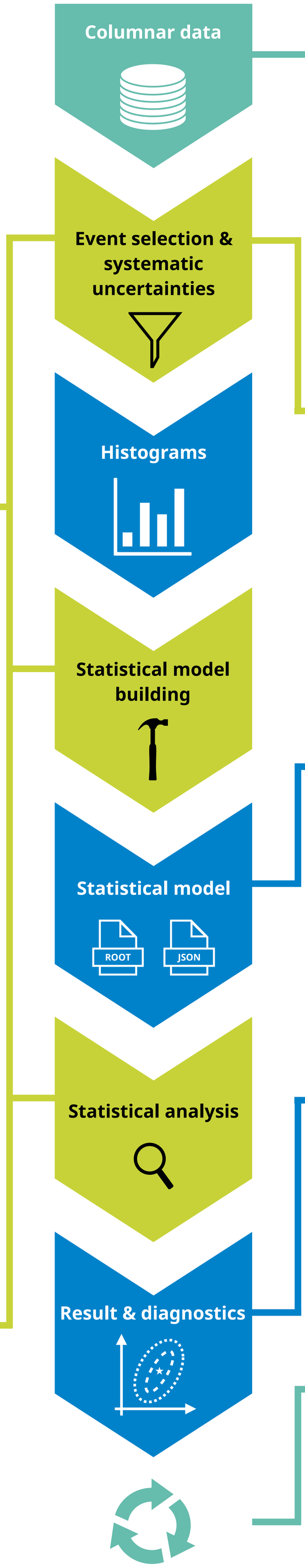
IRIS-HEP **investigates** the **performance and usability** of existing software for various parts of the workflow (🔗1). The studies compare the use of an established monolithic framework (`TRExFitter`, 🔗2) to approaches that make use of `pyhf` (🔗3) and tools developed in the `FAST-HEP` (🔗4) project.

To accommodate novel analysis methods, the full workflow is envisioned to be **end-to-end differentiable**.

```
General:
  Luminosity: 139.0
  Label: "My fit"

Fit:
  POI: "Signal_norm"
  FitType: "Asimov"

Samples:
  - Name: "ttH"
    Type: "Signal"
    Path: "samples/ttH.root"
    Color: "#FFAA55"
 - Name: "ttbar"
    Type: "Background"
    Path: "samples/ttbar.root"
    Color: "#F55EE33"

Channels:
  - Name: "Signal region"
    Variable: "jet_pt"
    Bins: [0, 25, 50, 75, 100, 150, 200]

NormFactors:
- Name: "Signal_norm"
  Nominal: 1
  Samples: "ttH"

Systematics:
- Name: "Luminosity"
  OverallDown: -0.02
  OverallUp: 0.02
  Samples: all
  Type: “normalization"

...
```

Declarative configuration

Different stages of the workflow require similar information. The user specifies this information in a configuration file. The declarative format makes the configuration **highly readable** and **intuitive** to use. The format can be serialized for example as JSON and YAML and easily parsed for the relevant steps in the workflow. A possible design of such a configuration file is being investigated in 🔗5.



Data access

The typical **dataset size** at this stage is **multiple terrabyte**, and can scale **up to petabyte** depending on how much filtering was already applied. `ServiceX` from the **DOMA focus area** provides relevant parts of the dataset on demand. The scalability and intelligent caching are crucial for fast turnaround times.

The connection between existing analysis frameworks and `ServiceX` is being explored in 🔗6.

Selection & systematic uncertainties

Event selection, columnar operations and the various kinds of processing performed in this step make it the most **compute-intensive** part of the workflow. A wide range of packages with IRIS-HEP involvement, including `coffea` (🔗7), enter at this stage. To achieve modularity, a common ground for a selection language is needed.

The **SSL focus area** allows IRIS-HEP to **benchmark** realistic analysis examples at scale.

From models to likelihoods

The so-called workspace **serializes** all information needed to build the **likelihood function** for subsequent inference. While they have traditionally been `ROOT`-based, `pyhf` (🔗3) now provides a python-based alternative for workspaces in the `HistFactory` (🔗8) scheme, easily serializable as JSON.

Extensions to the definition of the workspace are planned in order to accommodate **novel analysis methods**.

Fit results and diagnostics

Fit results are typically presented with a small set of common visualizations used for diagnostics. We have developed **user stories** for these visualizations to facilitate the development of a **common declarative format** to specify them. This promotes **modularity** and allows for easier **preservation** of the results. For details, see 🔗9.

Reusability and preservation

Analysis reusability and preservation are **guiding design principles**. The modular containerized workflow is of great use to achieve these goals. Well-defined schemas and interfaces further help preserve analyses in a common format.

This project is supported by National Science Foundation under Cooperative Agreement OAC-1836650. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

🔗1 github.com/alexander-held/template-fit-workflows
🔗2 gitlab.cern.ch/TRExStats/TRExFitter
🔗3 github.com/scikit-hep/pyhf
🔗4 fast-hep.web.cern.ch/fast-hep/
🔗5 github.com/alexander-held/TRExFitter-config-translation
🔗6 github.com/kyungeonchoi/ServiceXforTRExFitter
🔗7 github.com/CoffeaTeam/coffea
🔗8 cdsweb.cern.ch/record/1456844
🔗9 github.com/iris-hep/as-user-facing