Long Term Reproducibility and Sustainability of Scientific Software
(Letter of Interest for Snowmass 2021 Computational Frontier)

Matthew Feickert[1], Giordon Stark[2], Steven Gardiner[3], Yu-Dai Tsai[3]

[1] *University of Illinois at Urbana-Champaign*
[2] *SCIPP, UC Santa Cruz*
[3] *Fermi National Accelerator Laboratory*

September 1, 2020

Abstract

This Early Career Letter of Interest focuses on the importance of having reproducible and sustainable software for improving the physics of the HL-LHC era, but is not limited to only larger collaborations. The existing efforts in particle physics to improve reproducibility and sustainability are mentioned, and known upcoming challenges are discussed. Recommendations for improving the outlook of the field's software are given.

Introduction

Improving the reproduciblity and sustainability of software in the broader high energy physics (HEP) community is a challenge that affects all areas but disproportionately impacts early career researchers. The code that is developed for analyses is created and maintained at great time and expense to the researchers who write it, yet it is rarely developed in a manner that it is reusable between analyses, distributable, and properly credited. Instead, in most circumstances, beyond a few macros or scripts the code is largely abandoned once a publication is produced. Additionally, many of the scientists who maintain the code "graduate out" of the collaboration and that expert knowledge is usually lost.

With Run 2 of the LHC complete, final analysis of the data ongoing, and LHC Run 3 and the High Luminosity LHC (HL-LHC) on the near horizon it is crucial to consider how code written at the LHC experiments will be sustained and preserved so that it can be both reused to replicate previous analyses for new signals (c.f. RECAST [1] and REANA) as well iterated upon to be useful tools for future analyses well into the HL-LHC era. This applies not only at the level of the experiment analysis frameworks (i.e. `Athena` [2] and `CMSSW` [3]) but also at the level of the code for individual analyses. To facilitate these goals, there should be an additional push within the borader community to adopt and adhere to Findable, Accessible, Interoperable, and Reusable (FAIR) principles for software and data products [4].

Current State

The state of research and work on improving the reproducibility and sustainability of scientific software is well established and active outside of particle physics [4, 5]. In recent years, is has also started to be taken more seriously inside the community as well [6]. The RECAST framework [1] has been used in ATLAS to successfully perform reinterpretation analyses [7, 8] and in the case of [8] extend the scope of the original work to provide additional coverage in theory space. These results are important examples of reusable software maximizing the scientific potential of the LHC data, as noted in [8]:

> For a small fraction of the effort necessary to resurrect a published result, RECAST was used to improve the scope and depth of the ATLAS search program. As the time required to double the luminosity collected by LHC experiments increases to years, the importance of a reliable and easy-to-use tool such as RECAST will only grow. In the case of [long lived particle (LLP)] searches, such a protocol is critical as no other tool allows for high fidelity reinterpretations of published results.

Additionally, publication and preservation of data products, such as likelihoods, falls under the same scope as the software that generated them, given that they are necessary components of reinterpretation

studies. CMS has been publishing simplified likelihoods since 2017 [9] and has been using the CERN Analysis Preservation Portal, which implements FAIR data practices, for preservation of results. ATLAS has started using `JSON` as a serialization standard for the publication of full likelihoods from LHC Run 2 publications [10, 11] and as of August 2020 has published four full likelihoods to HEPData [12–15].

Challenges

Being able to predict what technologies will be available and survive into the future beyond five years is a near impossible task. It is therefore important to not attach all success and goals to individual products or implementations, but rather to concepts and more generic serializations and implementations that can be easily read or ported into future languages, implementations, and technologies. For example, the concept of the n-dimensional numerical array, largely popularized in `NumPy` [16] which had succeeded in unifying interfaces between existing array libraries, has become so ubiquitous in the Pythonic computing world that in August of 2020 the Consortium for Python Data API Standards was formed to develop common "API standards for arrays (a.k.a. tensors) and dataframes." [17] It is therefore important to consider that any analysis tool that offers a Pythonic API should provide one that is consistent with the Python Data API Standards. Similarly, serialization schemes change rapidly and so adopting a singular binary file format is problematic, as this requires the preservation of the reading and writing of that format indefinitely into the future. Serialization schemes that are plain text and have been used ubiquitously across fields (e.g. `JSON`) offer potential choices for serialization standards as they offer high portability and long term support from industry and other communities. Additionally, if these standards were to be dropped, their heavy use in industry and other fields would require there to be clear transnational standards to new serializations and backwards compatibility.

Additionally, containerization technologies should be considered carefully. Docker is the leading containerization technology in terms of use, however, as one of the first widespread implementations of Linux containers it has weaknesses and can present some security concerns. While alternative implementations of container technologies (i.e. Singularity, Shifter, Podman) exist, they largely all have the ability to build containers from `Dockerfiles` — Docker's build instruction set. Podman has even largely adopted the Docker API, allowing for near seamless switching from Docker to Podman to the extent that `docker` can even be aliased to `podman` in most situations. This means that containers that can be created from `Dockerfiles` could have a higher probability of being reproducible in their build content in a future beyond Docker.

There are inherent sociological challenges to making the field of particle physics more open, reproducible, and sustainable. Among them is requiring the field to change its practices *en masse* while many people have devoted huge amounts of their work and careers to bring the field to its current state of success. The adoption of sustainable practices requires a change in both workflows and thinking about how software is produced, maintained, and distributed. Some of these changes may prove harder for the field to adopt than others given the desire to carry forward social momentum from Run 2 of the LHC. Additionally, community attitudes toward people who develop and maintain software will need to change as well: much greater importance should be ascribed to their efforts with corresponding incentives and recognition. This could additionally require new career paths within particle physics to be created to allow for the retention of physicists with software development expertise, as well as the rise of hiring research software engineers (known as RSEs) to lead software projects that physicists would contribute to.

Conclusions and Goals

It is crucial for the computational challenges that the field will face in future runs of the LHC and at successor accelerators and experiments that the software used for processing and analysis of experimental data be openly developed and have long term sustainability and reproducibility at the forefront of their design. These issues will disproportionately affect early career researchers, who will predominantly be the ones to develop and maintain all software used in future experiments. It is our recommendation to establish an inter-experimental working group to focus on the the largest-impact actions that can be moved forward to provide enough traction in the field to allow for reproducible and sustainable software practices to become research norms. These contributions would culminate in a white-paper that would describe the current state of the field and serve to educate the Snowmass 2021 community on progress in the area. While this document has focused on scientific software for collider experiments, similar issues should be explored in other subfields, and we encourage our colleagues from across all areas of HEP to participate in preparation of the white-paper.

References

[1] K. Cranmer and I. Yavin, "RECAST: Extending the Impact of Existing Analyses," *JHEP* **04** (2011) 038, arXiv:1010.2506 [hep-ex].

[2] ATLAS Collaboration, "Athena," Apr., 2019. https://doi.org/10.5281/zenodo.2641996.

[3] CMS Collaboration, "CMSSW," 2020. https://github.com/cms-sw/cmssw.

[4] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, *et al.*, "The FAIR guiding principles for scientific data management and stewardship," *Sci. Data* **3** (2016) 160018.

[5] L. A. Barba, "Terminologies for Reproducible Research," arXiv:1802.03311 [cs.DL].

[6] X. Chen *et al.*, "Open is not enough," *Nature Phys.* **15** no. 2, (2019) 113–119.

[7] **ATLAS** Collaboration, "RECAST framework reinterpretation of an ATLAS Dark Matter Search constraining a model of a dark Higgs boson decaying to two b-quarks." ATL-PHYS-PUB-2019-032, 2019. https://cds.cern.ch/record/2686290.

[8] **ATLAS** Collaboration, "Reinterpretation of the ATLAS Search for Displaced Hadronic Jets with the RECAST Framework." ATL-PHYS-PUB-2020-007, 2020. https://cds.cern.ch/record/2714064.

[9] **CMS** Collaboration, "Simplified likelihood for the re-interpretation of public CMS results," Geneva, Jan, 2017. https://cds.cern.ch/record/2242860.

[10] **ATLAS** Collaboration, "Reproducing searches for new physics with the ATLAS experiment through publication of full statistical likelihoods," Geneva, Aug, 2019. https://cds.cern.ch/record/2684863.

[11] M. Feickert, "Likelihood Publication and Preservation: Snowmass 2021 Computational Frontier Workshop," Aug., 2020. https://doi.org/10.5281/zenodo.3978653.

[12] **ATLAS** Collaboration, "Search for bottom-squark pair production with the ATLAS detector in final states containing Higgs bosons, b-jets and missing transverse momentum," 2019. https://doi.org/10.17182/hepdata.89408.

[13] **ATLAS** Collaboration, "Search for direct stau production in events with two hadronic τ-leptons in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector," 2019. https://doi.org/10.17182/hepdata.92006.

[14] **ATLAS** Collaboration, "Search for chargino-neutralino production with mass splittings near the electroweak scale in three-lepton final states in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector," 2019. https://doi.org/10.17182/hepdata.91127.

[15] **ATLAS** Collaboration, "Search for direct production of electroweakinos in final states with one lepton, missing transverse momentum and a Higgs boson decaying into two b-jets in (pp) collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," 2020. https://doi.org/10.17182/hepdata.90607.v2.

[16] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: a structure for efficient numerical computation," *Computing in Science & Engineering* **13** no. 2, (2011) 22.

[17] R. Gommers, "Announcing the Consortium for Python Data API Standards," Aug, 2020. https://data-apis.org/blog/announcing_the_consortium/.