ProSelecta: Re-e-wind

Luke Pickering[1⋆], **Patrick Stowell**[2†]

1 Rutherford Appleton Laboratory, Harwell Campus, Didcot OX11 0QX
2 Physics Department, University of Sheffield, Western Bank, Sheffield S10 2TN

⋆ luke.pickering@stfc.ac.uk , † p.stowell@sheffield.ac.uk

Abstract

Interpreting the wealth of data that modern particle physics experiments produce requires a lot of models. As particle physicists, we must predict what backgrounds we expect to observe in order to make new measurements or discoveries. New and improved elementary particle interaction models must still be able to accurately predict the growing corpus of existing measurements. Preserving the ability to rapidly and correctly test the predictions of interaction models is therefore critical for XXX. As modern measurements get more complicated, the process. ProSelectaprovides a minimal environment to write standalone interaction simulation processors that facilitate correct analysis preservation by allowing experimentalists to publish executable signal definitions and observable projection operators for their measurements.

Received Date
Accepted Date
Published Date

Contents

1 Introduction

The production of data from High Energy Physics (HEP) experiments is expensive and almost exclusively tax-payer funded. The analysis of said data is often deeply involved and thus the resulting measurements hold significant value. The use of simulation in data analysis is near ubiquitous in modern HEP measurements. As simulations and analyses get ever more sophisticated, the ability to rapidly and automatically reinterpret sets of existing measurements becomes more important, both to provide constraints on the predictions from new models and to search for anomalies that are only apparent in the context of improved model predictions. Reproducing the measurement from scratch is often impractical for a variety of reasons and thus preserving the ability to correctly predict historic measurements is a critical part of modern HEP measurements.

There are various techniques for releasing measurements and preserving HEP analyses in use and the main tradeoff is usually between the amount and complexity of input required for reinterpretation and the ease of use and the correctness of the preserved analysis. The simplest type of release might be a single measured parameter value with uncertainties contained in a publication. The most complex might constitute terbytes of recorded event data, the computational tools required to run the experiment simulation, and event selection code that depends on experiment-specific analysis frameworks. Most modern approaches lie between these two extremes, with the simplest case difficult to correctly reinterpret and the complex case being completely infeasible for individual users to use. One common approach in HEP, often used for measurements of particle scattering processes, is to publish histograms of properties for some class of signal event alognside detailed descriptions of the signal selection criteria. Community efforts such as RIVET [1] and NUISANCE [2] are then able to implement reinterpretation tools from the data and descriptions in the publication. Such efforts have been broadly effective and are heavily relied-upon within the relevant communities but do exhibit a number of downsides: The community tools take on *de facto* responsibility for the measurement implementations, including any incorrectness in said implementations; if important details are missed at the time of publication, it may not be possible to correctly implement a measurement; if the original analysers themselves do not provide an implementation alongside their publication, then either measurements are not well preserved or the preservation burden falls on the community; analysis implementers must learn the relevant parts of the specific community tool, which may be a non-trivial effort; and implementations are defined within a specific tool and usage in other analysis tools may require reimplementation from the original publication.

ProSelecta was designed in this context to mitigate these percieved downsides by standardising a processing environment that provides a minimal set of functions that can be further composited to define standalone event selection and property projection operations. ProSelecta should be used as part of the measurement workflow to define the signal selection and measured property projection functions. Analysers can then directly publish standalone, executable versions of their simulation analysis subroutines alongside the measurement; no lossy translation to prose and back required. Community re-analysis frameworks can then use the provided ProSelecta tools or, alternatively, implement their own version of the environment, to automatically run these published code snippets over vectors of simulated HEP events to produce correct predictions for the measurement in question. Thus, ProSelecta supports standardising, streamlining, and simplifing robust reinterpretation workflows that are central to modern HEP progress.

2 The Problem Domain

Particle physics measurements .

This allows measurement makers to publish the code used to define what they measured explicitly in a way that model-builders wishing to check if their model can predict the measurement can execute directly. Eliding a huge class of communication issues where the analysis is written in code, translated to English and then back to code.

GAMBIT, CONTUR, NUISANCE, GENIE Tuning.

3 The Solution Domain

Like ADL

containers for each measurement

HEPData

RIVET

NUISANCE

3.1 The ProSelecta Approach

The scope of ProSelecta is limited to defining Selections (filters) and Projections of some simulation particle interactions in a standalone source code file, colloquially called ’a snippet’. There are other *moving parts* to making a complete prediction of a given measurement, but the authors believe that this is one that hasn’t yet been solved generally and ergonomically and it can be. Fully automating measurement prediction.

The reference implementation of ProSelecta has been developed in the context of the NUISANCE framework, but can be used to execute C++ or Python analysis processors for any HepMC3 event stream.

Minimal set of functions, declaratively named so that reimplementation of snippets in other languages or frameworks is simple.

common bits in particle physics – event streams.

handles Physical real particles, incoming/outgoing, not internal particles, these can be retrieved using standard facilities of the event format.

4 The ProSelecta Environment

Currently defined in the context of C++ and HepMC3, with python bindings. The design of the framework is meant to be extended to other languages or event formats. A Lua/HepMC3 prototype exists, for example.

Note. Snippets cannot #include

4.1 Events

4.2 Cuts

4.3 Particles

4.4 Units

A simple system of units is included which is defined with the leadings units as c=1, MeV=1, and cm=1. This means that events should be converted to MEV before being passed through functions written against proselecta.

4.5 Vector Operations

Included as such elementary operations are critical for many projection operators and allow to work on 3-vector and 4-vector objects that are stored as simple arrays. In context of HepMC3::FourVector

4.6 Missing Datum

A magic missing datum constant is defined to flag values as explicitly invalid or uncalcuable. This allows explicit differentiation between an invalid return values from a projection operator and domain errors encountered by standard library math functions.

Example

4.7 Community Functions

5 ProSelecta Tools

6 ProSelecta Workflows

The ProSelecta Interpreter:

Compiled Snippets:

7 Conclusion

Alternative approaches such as ADLs (require), containerize it all, and RIVET (). Thus, ProSelectaisn't a complete analysis preservation solution, it tackles the same problem as an ADL, but doesn't require a new DSL and associated interpreter to be maintained. The authors hope that it becomes foundational software to ensure the longevity of world particle physics data.

Analysis preservation is tricky but critical. ProSelecta provides a minimal set of functions for writing MC processors in both C++ and Python.

Acknowledgments

The authors would like to thank Joshua Isaacson, for alpha testing and supporting the ProSelecta project in development and Andy Buckley and Chris Gutshow for providing comments on this manuscript.

Funding information L.P. is supported by The Royal Society under grant number: URF\R1\211661.

- Patrick Grants
- Sheffield CG STFC?

Authors are required to provide funding information, including relevant agencies and grant numbers with linked author's initials. Correctly-provided data will be linked to funders listed in the Fundref registry.

A A Worked ProSelecta Example

MINERVA 3D

B A Word On Code Safety

It is not feasible to be fully safe when attempting to execute arbitrary, user-supplied code. A lot of the recent progress in particle physics is built on trusting [1] software developed and self-regulated within the community. No automated tools can yet replace an expert user reading and understanding some third-party analysis code before running it. However, as ProSelecta encourages users to run snippets from third parties, some simple mitigation measures are included in the interpreter. These mitigations should limit the damage from malicious code snippets by limiting the complexity of the snippet that can be run and the ability of any contained code to interact with the the filesystem or the network. ProSelecta will refuse to run snippets which: 1) contain pre-processor directives, 2) are longer than 10000 characters, or 3) contain the token `system`. These restrictions can be disabled at build time by individual users if required. The authors welcome comments and further discussions on how to improve the safety of ProSelecta.

References

[1] C. Bierlich *et al.* *Robust independent validation of experiment and theory: Rivet version 4 release note*, SciPost Phys. Codebases **36** 10.21468 (2024), doi:10.21468/SciPostPhysCodeb.36.

[2] Stowell, P. *et al.* *NUISANCE: a neutrino cross-section generator tuning and comparison framework*, JINST **12** P01016 (2017), doi:10.1088/1748-0221/12/01/P01016.

[1]the word *trust* here is used in the social sense, rather than the information security sense.