



VECTOR

IRIS-HEP Fellowship 2022

Vector: Constructors, documentation, and benchmarks

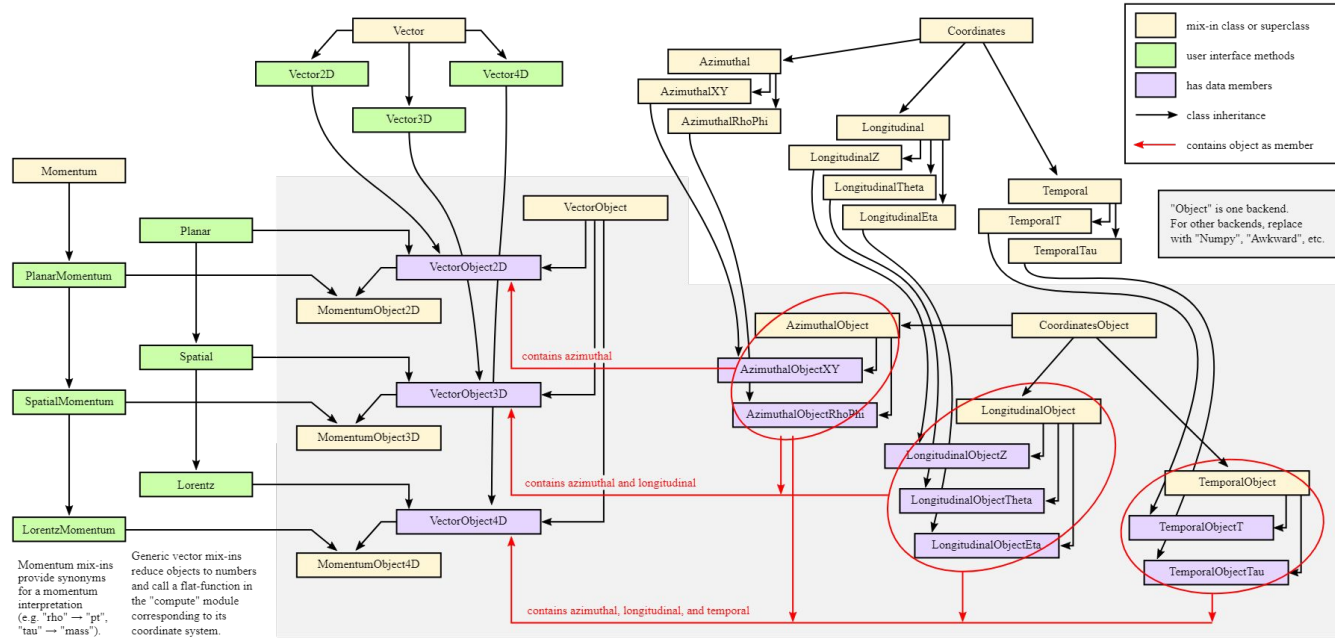
Saransh Chopra (University of Delhi)

Mentors: Henry Schreiner (Princeton University), Jim Pivarski (Princeton University)

Vector

- Vector allows working with 2D, 3D, and Lorentz vectors, to solve common physics problems in a NumPy-like way.
- Uses names and conventions set by ROOT's TLorentzVector and Math::LorentzVector, henryiii/hepvector, and coffea.nanoevents.methods.vector
- Users can create vectors in a variety of coordinate systems, including Cartesian, cylindrical, spherical, and any combination of these with time or proper time for Lorentz vectors.
- Comes loaded with 3 + 2 backends; a pure Python object backend, a NumPy backend, an Awkward Array backend, an Object-Numba, and an Awkward-Numba backend to leverage JIT (Just In Time) compiled calculations on vectors.

An amazing workshop by Mason Proffitt showing how Vector integrates with other HEP packages - <https://indico.cern.ch/event/1076231/contributions/4560398/>



<https://github.com/scikit-hep/vector/blob/main/docs/api/inheritance.svg>

Areas of improvement

1. Constructors

The user-facing API of Vector does not expose its true API, which are the Python classes. The current API provides wrapper functions (named as *obj*, *arr*, and *awk*) to users. These functions are also displayed in the `__repr__` method of the classes making them even more ambiguous.

2. Documentation

Vector lacks proper user as well as API documentation and tutorials. Currently, there are no detailed explanations in the existing tutorials, and most of the functions and classes do not contain docstrings.

3. Benchmarks

Scikit-HEP packages, including Vector, lack benchmarks. Benchmarking would be a valuable addition to the Vector project, which could then be propagated to all Scikit-HEP packages in the future. Also, benchmarking against ROOT would be interesting!

Goals

1. Constructors
Expose the inner constructors and classes to the user API.
2. Documentation
Improve the existing API documentation, add tutorials, and add documentation for new features.
3. Benchmarks
Develop a benchmark suite for Vector which will then be propagated to other Scikit-HEP packages.
4. Misc
Bug fixes, chore jobs, minor feature additions, etc.

Summarising: Preparing Vector for its first major release!



Thank you!