ETH zürich

HEP analysis within the Python Ecosystem

Davide Valsecchi (ETH Zurich)

INFN Quasi-Interactive Analysis Workshop

09/01/2025

CMS

PocketCoffea:
Configuration layer for
CMS analyses

coffea

Utilities for general HEP analysis. Processor structure and scaling infrastructure

Awkward Array

Data manipulation layer

uproot

ROOT
Data Analysis Framework

Storage I/O
layer

Boost Histogram

mplhep

Scikit HEP

VECTOR

DASK

matplotlib

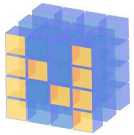
Storage layer

The storage layer is handled by ROOT with TFiles, TTrees and RNTuples.

Uproot allows to read ROOT file in the python environment without having the ROOT lib installed.

It is a lightweight variable implementing ROOT file format reading and converting data to various arrays formats

Arrays



Awkward Array

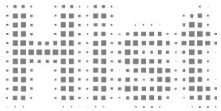
Once read from ROOT files, data is manipulated as arrays, where events are the **rows** and branches are **columns**.

The special nature of HEP data, where each event can contain collection with a different number of elements is not suitable for tabular arrays like numpy ones.

Awkward-array is a library to manipulate this kind of **jagged array**s in a way very close to numpy.

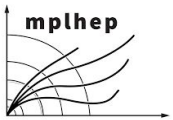
Uproot can convert TTrees to awkward arrays.

Histogram and plotting



Histogramming in Python is handled with the boost-histogram package, for fast and efficient filling.

The **hist** python package is the python binding of the boost lib. This package allows user friendly creation of multidimensional histograms.



Matplotlib is the standard for plotting, with tools such as mplhep to make the creation of HEP typical graphics easier.

Coffea - Columnar Object Framework For Effective Analysis

coffea is a prototype package for pulling together all the typical needs of a high-energy collider physics (HEP) experiment analysis using the scientific python ecosystem.

- It provides utilities to read the CMS/ATLAS data formats with awkward arrays with additional convenient features
- It provides tools for analysis operations: selections, weights, computing corrections, histogramming, skimming, exporting ntuples
- It makes possible to define an analysis workflow once and scale it to larger clusters throught Dask, Spark o WorkQueue

Caveat about packages versions

For the hands-on we will use: coffea 0.7.23, awkward 1.10, uproot 4

These versions are considered **stable**, and there are now newer versions with some upgrades: the main addition is integration of **dask-arrays**, and **dask-histograms**.

The evaluation of operations becomes completely lazy → a task graph is built and then executed using **dask as a scheduler engine**.

In today's hands-on we will use the legacy version of the packages which are full-analysis proof and battle tested. All the operations in today's tutorial will be still completely valid for the most updated version (coffea 2025.1.1, awkward 2a)

Let’s start the Hands-on session:

GitHub repository

git clone https://github.com/valsdav/PyHEPTutorial_INFN_InteractiveAnalysisWorkshop.git