

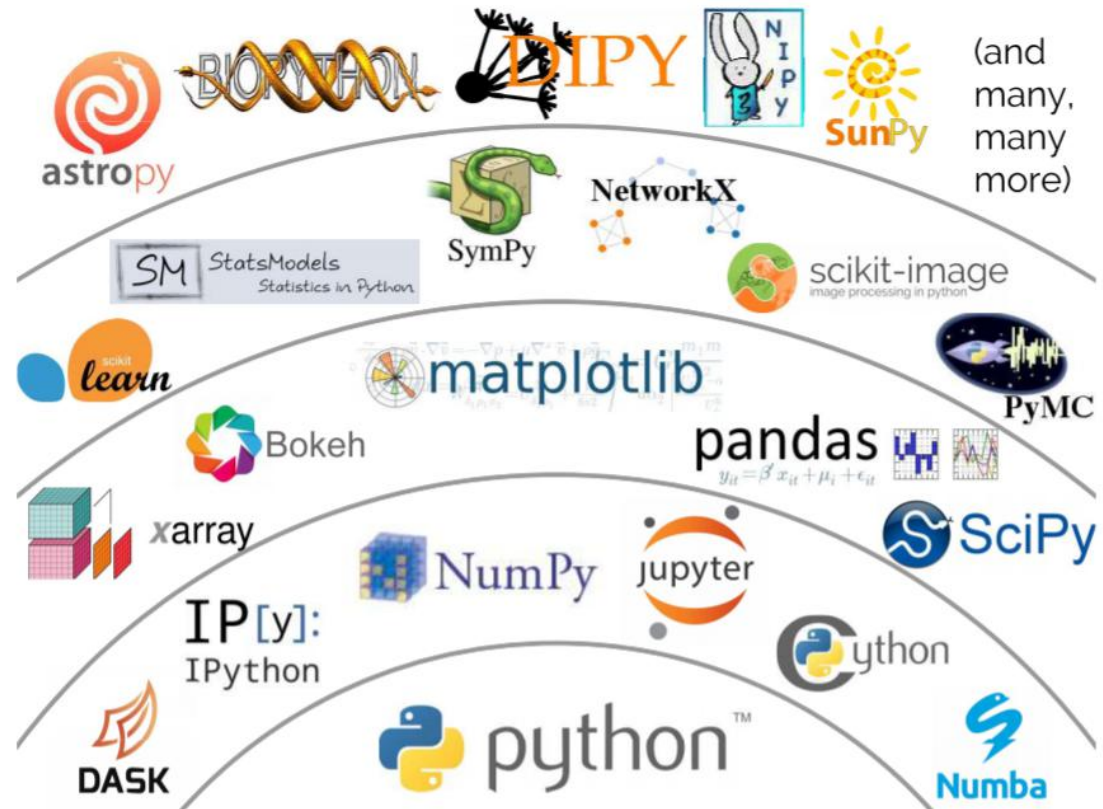
Introduction of Scikit-HEP project

Liu Kai

Python scientific software ecosystem

Including:

- data manipulation,
- visualisation tools,
- packages for statistics
- machine learning, etc.



Motivation of Scikit-HEP

- Traditionally, HEP has been evolving in a rather disjoint ecosystem based on the framework of ROOT (C++)
- For Python, many similar tools available for
 - data manipulation and modeling,
 - fitting
 - statistics and machine learning.
 - **But it is a toolkit rather than a toolset, with bindings to Python**
- Various initiatives exist, which try to **link both HEP and non-HEP worlds**.
- But they mainly tackle(d) specific topics.
- want to do it with more generalised effort, domain-specific oriented.

My personal feelings

- Last year, I taught two young students machine learning
 - ROOT is matured and specific for HEP, but mainly the BDT is incorporated.
 - we used scikit-learn
 - generally used, but mainly in other areas, not HEP
 - not only BDT, many other Alg. available
 - but not straightforward to use ROOT file, which generated by the preSelection/analysis code

Scikit-HEP overview

- The project should neither be seen as a replacement for ROOT nor a replacement for the Python ecosystem based on the SciPy suite



uproot

Minimalist ROOT I/O
in pure Python and Numpy

scikit-hep

Starting point of project.
Contains tools for maths,
kinematics, units, etc.

histbook

Versatile, high-performance
histogram toolkit for Numpy

numpythia

Interface between
PYTHIA and NumPy

Vega
Scope

Minimal viewer of Vega / Vega-Lite
plots in your web browser from
local or remote Python processes

formulate

Easy conversions
between different
styles of expressions

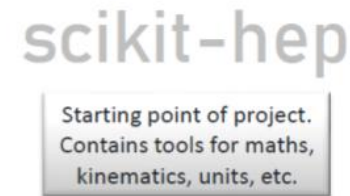
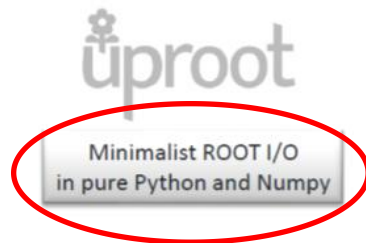
pyjet

Interface between
FastJet and NumPy

Question from Yuhang:

Can you give an example to introduce “cross-talk”?

- expanding the typical set of tools for HEP physicists with common APIs and definitions to ease “cross-talk”



A ROOT file can natively and trivially be read with the pure I/O uproot package [8]. The package strength lies in the fact that it does *not* depend on ROOT, making it installable on virtually any computer via a `pip install uproot` command. It bridges the ROOT and the NumPy-powered worlds. A ROOT TTree object is read as a dictionary of arrays as follows:

```
>>> import uproot
>>> rootfile = "Zmumu.root"
>>> zmumu = uproot.open(rootfile)["events"]
>>> zmumu.arrays(["px1", "px2", "py1", "py2", "M"])
{b'px1': array([-41.19528764,  35.11804977,  35.11804977, ...,  32.37749196,
                32.37749196,  32.48539387]),
```

Question from Xin

- **My question is: what would be the most realistic approach if we want to try on our BES/CEPC analysis? (I checked the root_pandas, it seems it hasn' t been updated for some time...)**
- I personally think this project not started for long and is not matured yet
- For BES/CEPC, the widely used ROOT, although not perfect, maybe our best choice by now——most robust and used/tested worldwide by many many people.
- Maybe finally it is other project succeeded, But the effort they made showing an important direction that worth keeping an eye on.



Scikit-HEP project - welcome!

The *Scikit-HEP project* is a community-driven and community-oriented project with the aim of providing Particle Physics at large with a Python package containing core and common tools. The project started in Autumn 2016 and is under active development.

Question from Amit

- **My Question is: Various initiatives exist or have existed, which try and link both HEP and non-HEP worlds. But they mainly tackle(d) specific topics? what kind of specific topics are linked with HEP and non-HEP world? and What kind of effort can be made to generalize HEP and non-HEP world?**
- For first Question,
 - An example : GWPY used by LIGO experiment whose dependencies: numpy, scipy, cycler, matplotlib, astropy
- For second Question,
 - that is what Scikit-HEP trying to do



Question from Ryuta

- I have a quick look the astropy project (bellow are some pages from the project) <https://www.astropy.org/about.html>, <https://www.astropy.org/team.html> , and this project might be one of a template for the scikit-HEP, though not clearly mentioned. **I would think that one of motivation behind the astropy is the fact that lots of data, such as taken by satellites, can be accessed freely outside of collaborations, and sometimes the softwares are also open as well.** Above is just my guess, but does Scikit-HEP possibly have motivations like this ?
- My personal guess
 - you are right, accessing data is one of the motivation
 - I think the motivation not only limited in the datastructure/data accessing, but also using the matured Alg(such as fitting, machine learning) from other Python projects, which are widely used and matured already.

Summary

- Scikit-HEP, a new toolset of Python packages is introduced
 - Building bridge between different technologies and/or popular packages from the Python scientific software.
- Not fully matured yet, but the direction of doing this is important
- for each experiment/project, using the common tools widely used in the collaboration is a better choice
 - but we should remember, that is not the only choice in the wider scientific area.
 - keep an eye on the new trends....