Bishoy's bag of tricks
rootpy:
A basic introduction

Bishoy H. Dongwi
The Bishoy school of computational physics

Hampton University, Hampton VA 23668

February 6, 2018

*This work has been supported by DOE awards DE-SC0003884 and DE-SC0013941

What this presentation is not

- Information overload
- Dictate that rootpy or nothing
- Tutorial

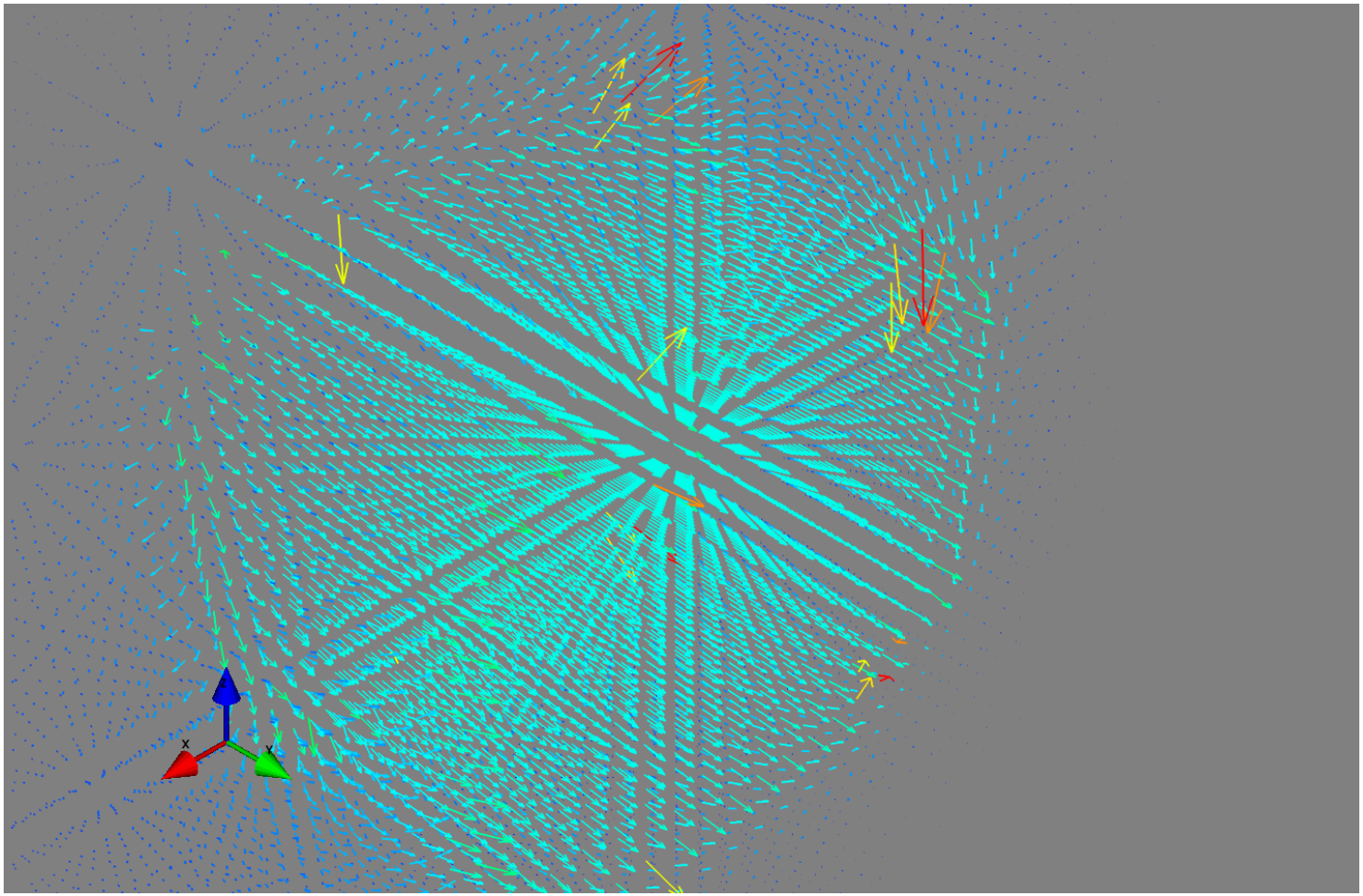
Why python?

- One of the most powerful programming languages
- Very intuitive and easy to learn and use
- Steadily becoming ubiquitous within scientific community

Overview

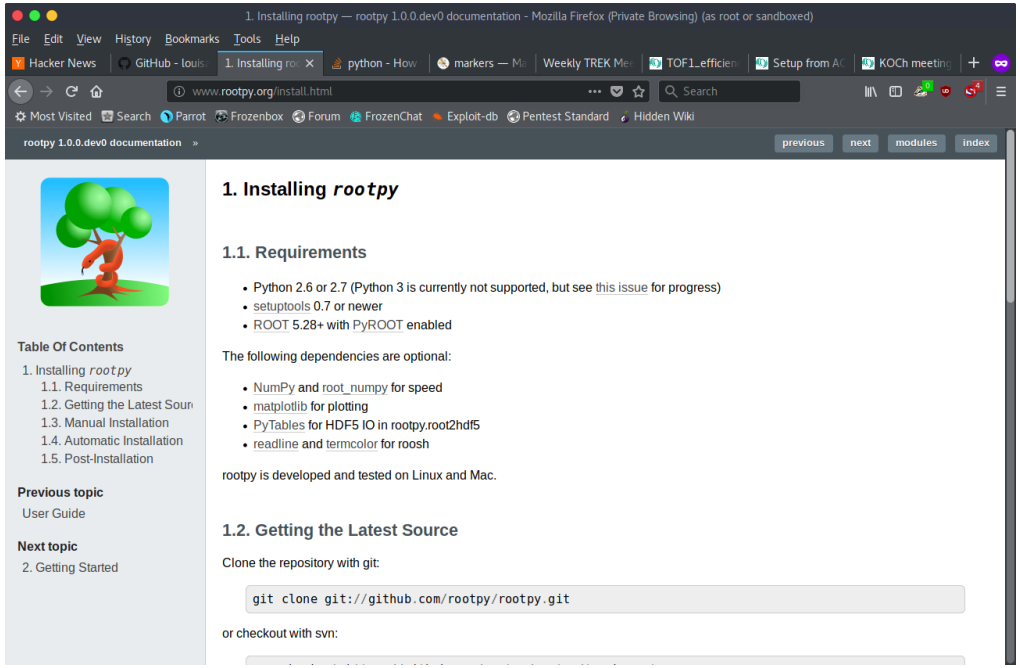
1. My introduction to python
2. rootpy
   - Useful information
3. Using rootpy
   - Advantage of using rootpy

scipy



plot of TOSCA generated 3D magnetic field plot using mayavi viewer & scipy

Introduction to rootpy



Installing rootpy

- **Manual:** python setup.py install –user
  sudo python setup.py install
- **Automatic:** sudo pip install rootpy

Visit: http://www.rootpy.org/

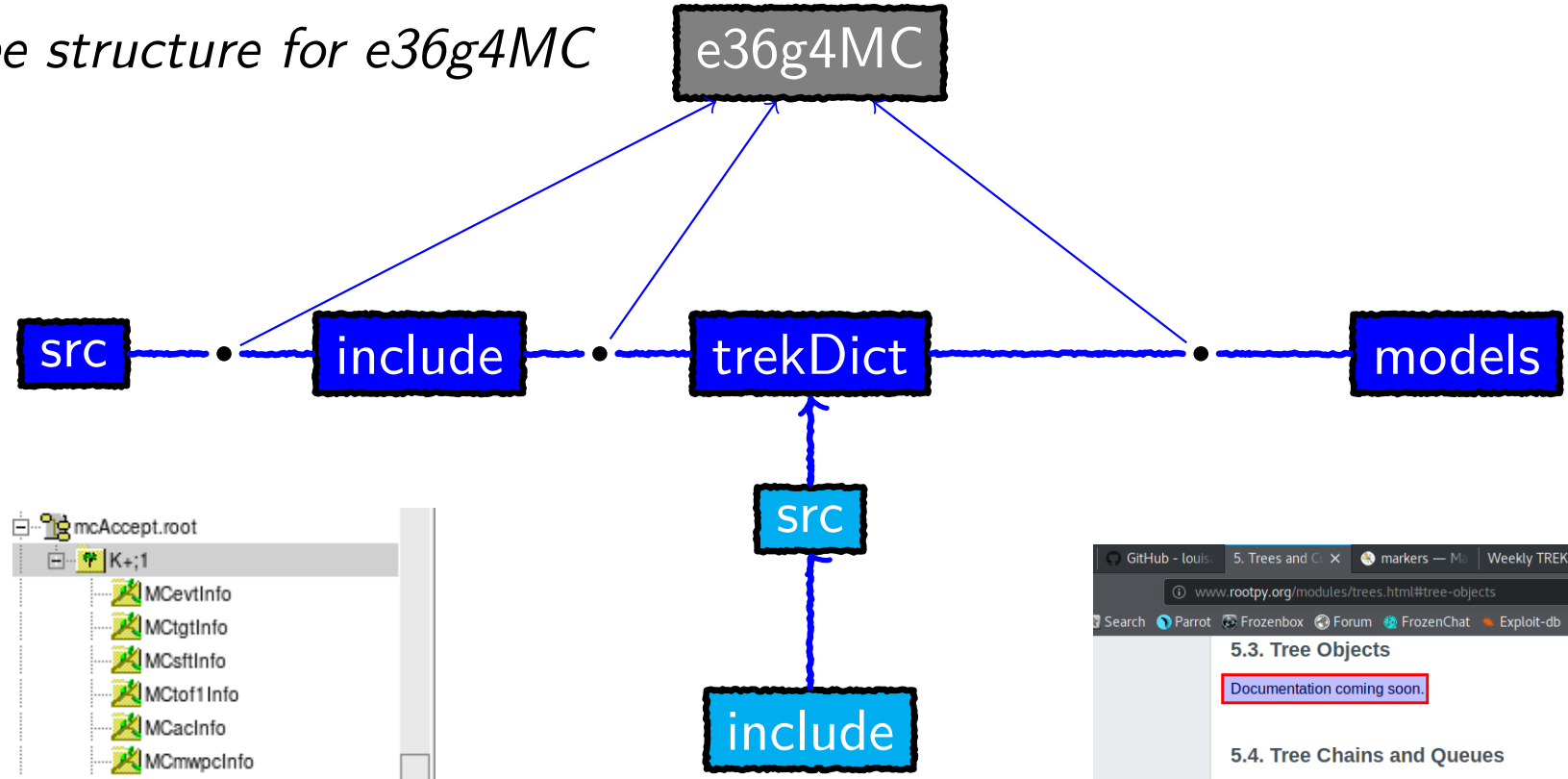
Introduction to rootpy cont...

Additional packagest to install: *root_numpy*

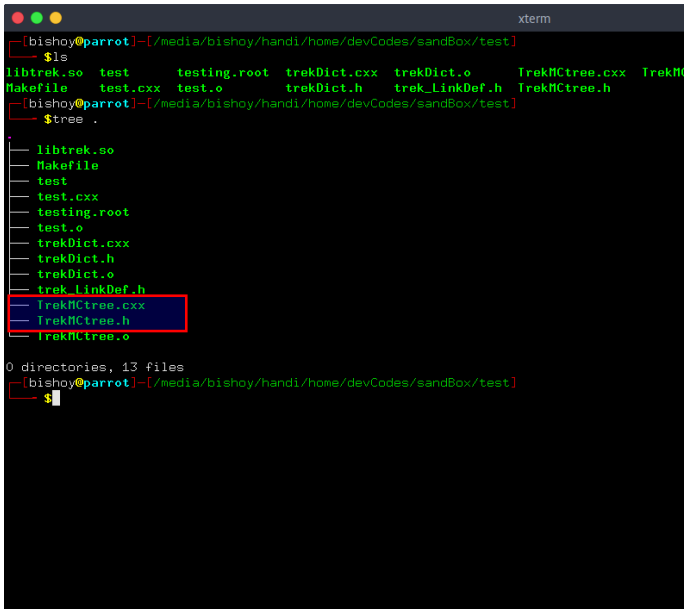
- Python extension module that provides an efficient interface between ROOT and NumPy
- Internals are compiled $C++$
- Handle large amounts of data much faster than pure Python implementations
- Powerful & flexible functions for converting TTrees into structured NumPy arrays
- Convert NumPy arrays back into TTrees
- Convert branches of strings & basic types such as bool, int, float, etc. as well as variable-length & fixed-length multidimensional arrays and 1D or 2D vectors of basic types & strings
- Create columns in the output array that are expressions involving the TTree branches similar to *TTree* :: *Draw*()

Creating ROOT TObjects

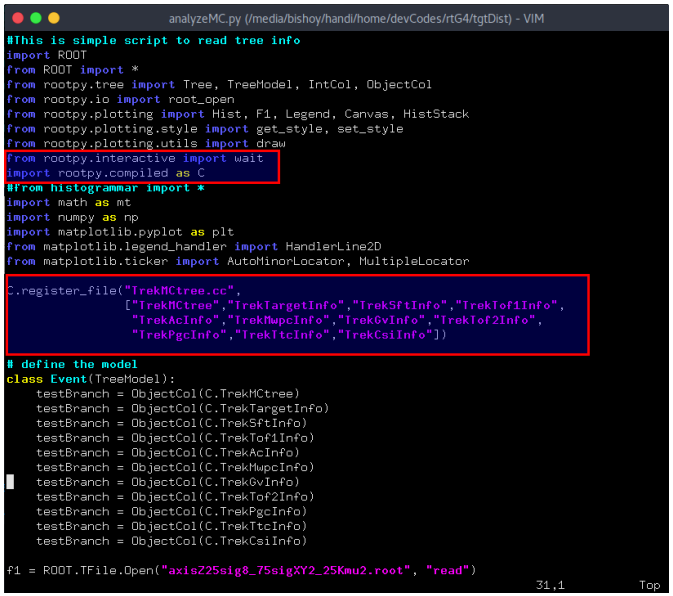
a) Tree structure for e36g4MC



Why I use it



C/C++



rootpy

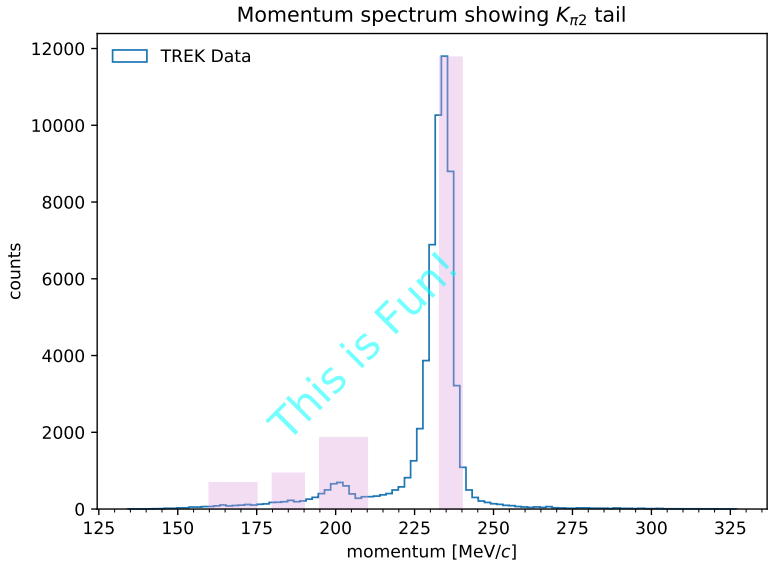
The Zen of Python

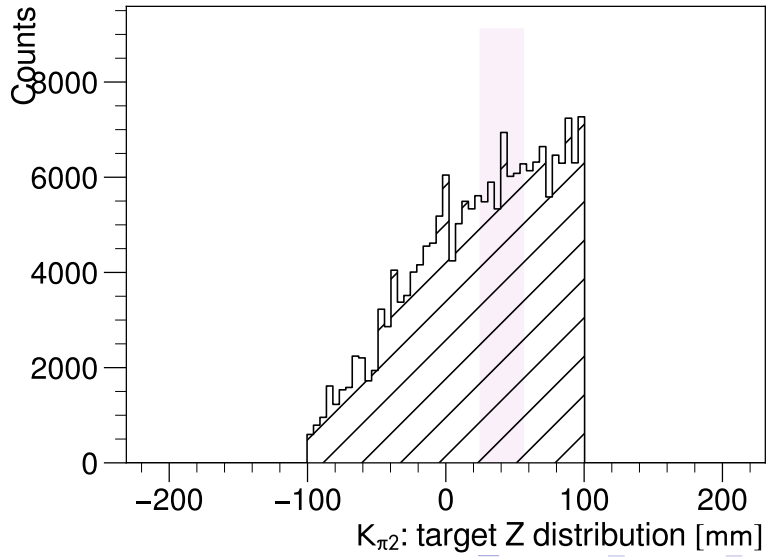
- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Plotting is fun

Using root_numpy

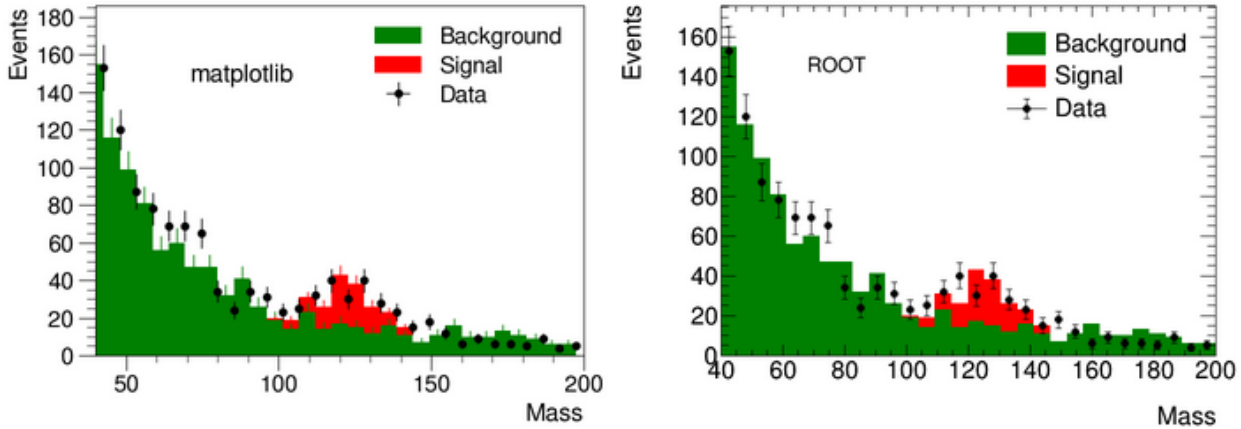
```
#define some array variables for each momentum cut
a1 = data[:,9]
mom = data[:,2]

fil1 = (mom > 199)&(mom < 207)
fil2 = (mom > 179)&(mom < 191)
fil3 = (mom > 199)&(mom < 211)
fil5 = (mom > 184)&(mom < 215)
fil4 = (mom > 229)&(mom < 243)

h_dat1.fill_array((a1[fil5]*10))
h_dat2.fill_array((a1[fil2]*10))
h_dat3.fill_array((a1[fil3]*10))
h_dat4.fill_array((a1[fil4]*10))

#simple test
#Draw canvas:
canvas1 = Canvas(width=700, height=500)
canvas1.SetLeftMargin(0.15)
canvas1.SetBottomMargin(0.15)
canvas1.SetTopMargin(0.10)
canvas1.SetRightMargin(0.05)
h_rat3.Draw()
#wait()
'''
```





Not a summary

- Community-driven initiative to provide pythonic interface with ROOT on top of PyROOT bindings
- Redirect ROOT's messages through Python's logging system
- Plot ROOT histograms/graphs with *matplotlib*
- Manipulate ROOT TTrees as Python arrays and *vice-versa*



GitHub: https://github.com/rootpy/rootpy