Data & Analysis Preservation: status update

Maxim Potekhin
Nuclear and Particle Physics Software Group

Brookhaven National Laboratory

PHENIX DAP Meeting
09/30/2021

Overview

- HEPData
- Website
- PISA
- New tools
- Workflow capture/REANA discussion

HEPData

- Ongoing but now at a slower pace than before - people are busy
- An interesting issue of correlated error bars in PG202
    - Not supported natively on HEPData
    - Can it be mitigated by a supplementary note on Zenodo?

Website

- Fixed a small bug on the run display pages
- Added 5 new conferences (thanks Gabor) - now covering 2017 - now at 66 conferences referenced on the website - on both “keywords” and “conferences” pages

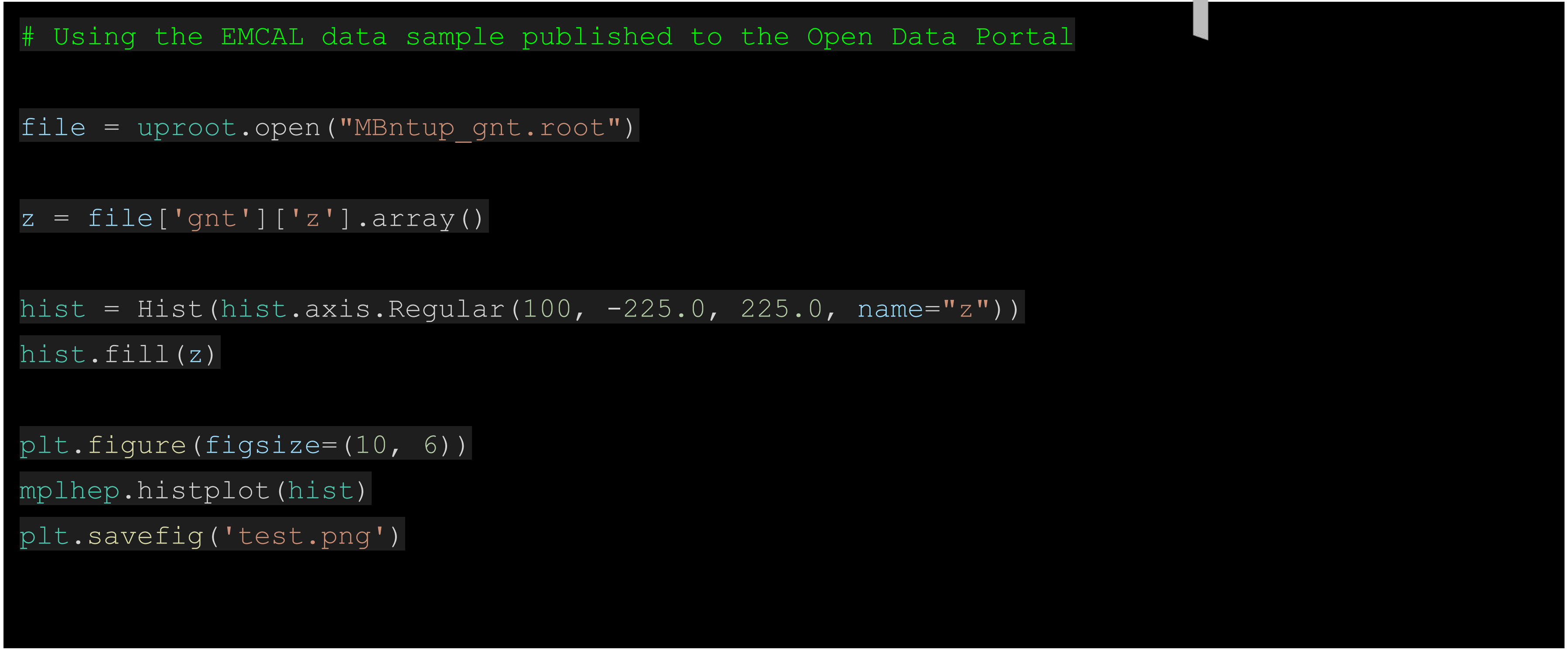
PISA

- Zhiyan Wang has discovered an issue with handling of the VTX dead channel maps in pisaToDST.C when running simulations for Run 15. They will talk to Takashi.
- It would be optimal to produce a note on dead channels to improve documentation, possibly add to the website.

New tools: uproot

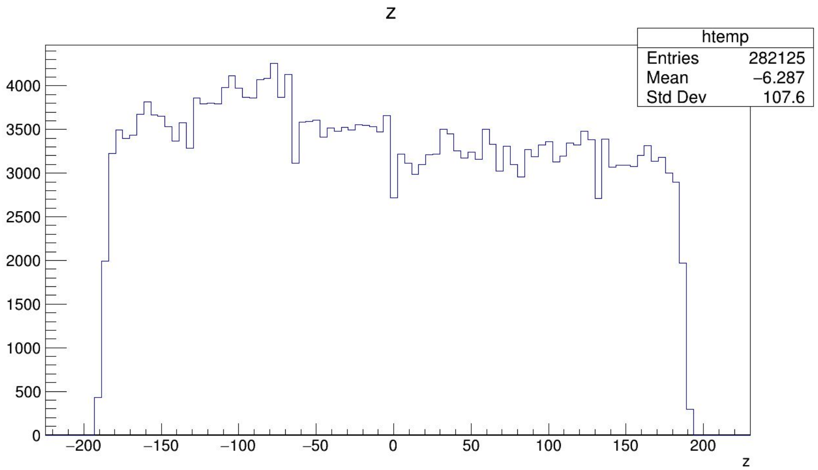
- Uproot is a Python package for ROOT I/O
- Could be interesting for grad students since it’s a state-of-the-art component with links to tools and packages like
  - Numpy
  - Pandas
  - Awkward array
  - Various statistical packages, ML etc
- Joining the Python ecosystem is a huge benefit for people starting their careers both within and outside academia
- Can make an intro at the next PHENIX School

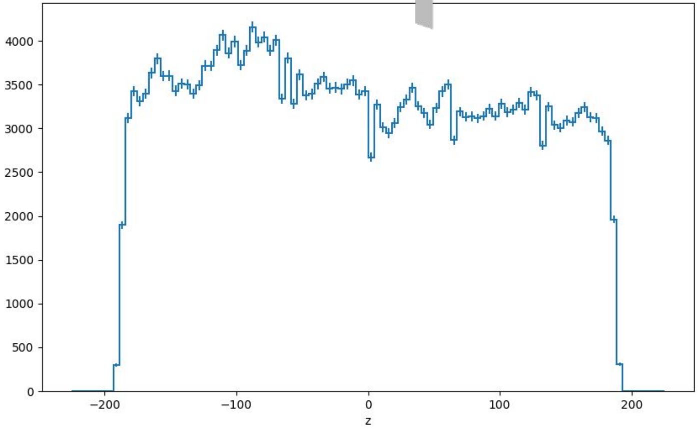
A histogramming example



```
# Using the EMCAL data sample published to the Open Data Portal

file = uproot.open("MBntup_gnt.root")

z = file['gnt']['z'].array()

hist = Hist(hist.axis.Regular(100, -225.0, 225.0, name="z"))
hist.fill(z)

plt.figure(figsize=(10, 6))
mplhep.histplot(hist)
plt.savefig('test.png')
```

Uproot histograms and plots

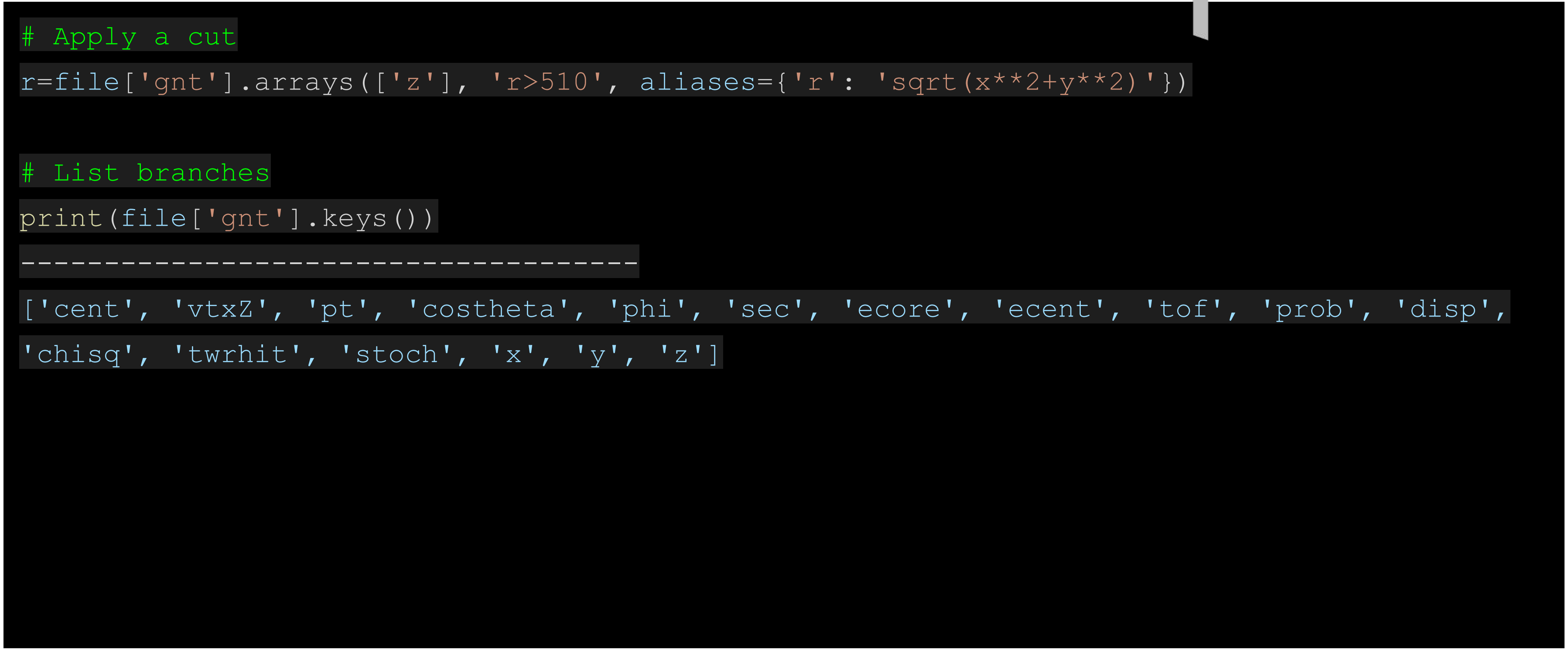




An example using Gabor’s EMCAL Ntuples:

- ROOT plot on the left
- Uproot on the right

Applying cuts, browsing



```
# Apply a cut
r=file['gnt'].arrays(['z'], 'r>510', aliases={'r': 'sqrt(x**2+y**2)'})

# List branches
print(file['gnt'].keys())
-------------------------------------
['cent', 'vtxZ', 'pt', 'costheta', 'phi', 'sec', 'ecore', 'ecent', 'tof', 'prob', 'disp',
'chisq', 'twrhit', 'stoch', 'x', 'y', 'z']
```

A thought on workflow diagrams (Niv’s work)

- Kudos - this is the right thing to do… Considerable complexity is evident...
- To make an impact the workflow chart - the graph - should be mapped to software
- Is this easy or difficult?
  - Adding tags to blocks on the diagram manually, inserting corresponding tags as comments in the code (making it discoverable) - this solution handles N-to-M mapping
- We could use a declarative description of the graph using XML/YAML/JSON/Python and generate the graphic automatically - also including mapping to software components

REANA

- Let’s document and test components of Niv’s analysis
- Should we add this material to Zenodo and the website?
- REANA testing to follow
- Can use this work to create another Open Data entry