Alternative HEP software approaches at an undergraduate institution

Matt Bellis, *Siena College/Cornell University*

CMS Software R&D Upgrade meeting

6/8/2015

What do we do at Siena?

Undergraduate-only

Joined CMS in 2013 through
Cornell (~3 hour drive)
NSF funded

Previously BaBar, Jefferson Lab

Other current projects: *CoGeNT (dark matter detection), Cosmological calculations with GPUs*

What do we do at Siena?

My students after graduation:

Grad school

- Medical physics (3)
- Engineering
- Business/sports analytics
- Environmental
- *Neutrino physics*
- *LHCb*

Jobs

- Electronics lab tech

Current research students: (2 CS, 3 engineering-focus)

Computing languages

Students learn

- Python
- MATLAB
- *Maybe* R, Java
- ***No C/C++!***

No ROOT

- Little post-Siena use
- No time (even for PyROOT)
- ***Personal choice to commit to alternatives***

My work on CMS

Physics analysis

- Contribute to boosted top $d\sigma/dp_T$ analysis
- Baryon-number violating top decays

Data preservation

- Learning tools for http://opendata.cern.ch/
- New outreach tools

Tools for undergraduates, high-school students, and citizen scientists

CMS Made Simple

Tool for undergrads and outreach efforts

- Data is simplified text (zipped) files
    - 4-vecs + (b-tag, charge, etc)

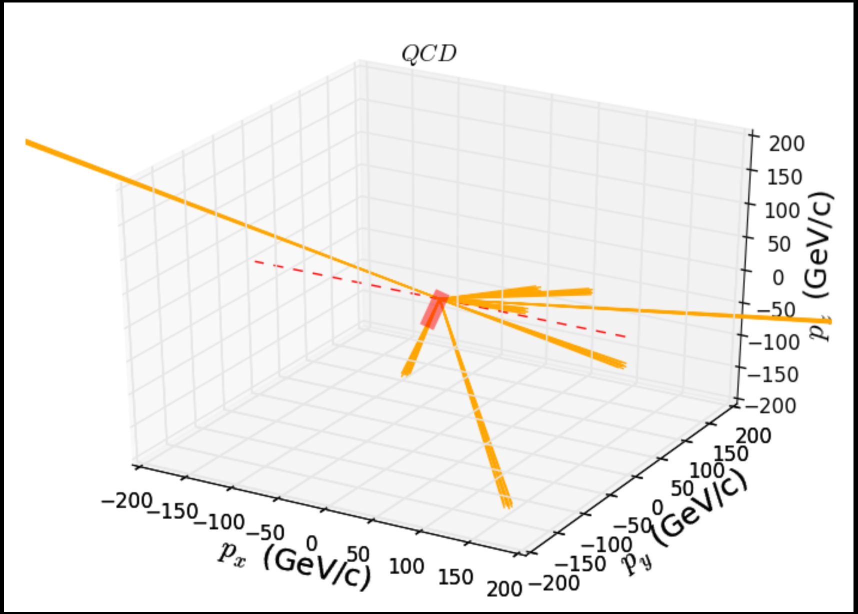
- Python accessors
    - Maintain Python-esque syntax
    - 4-vector viewers
    - *Sacrifice speed for readability*

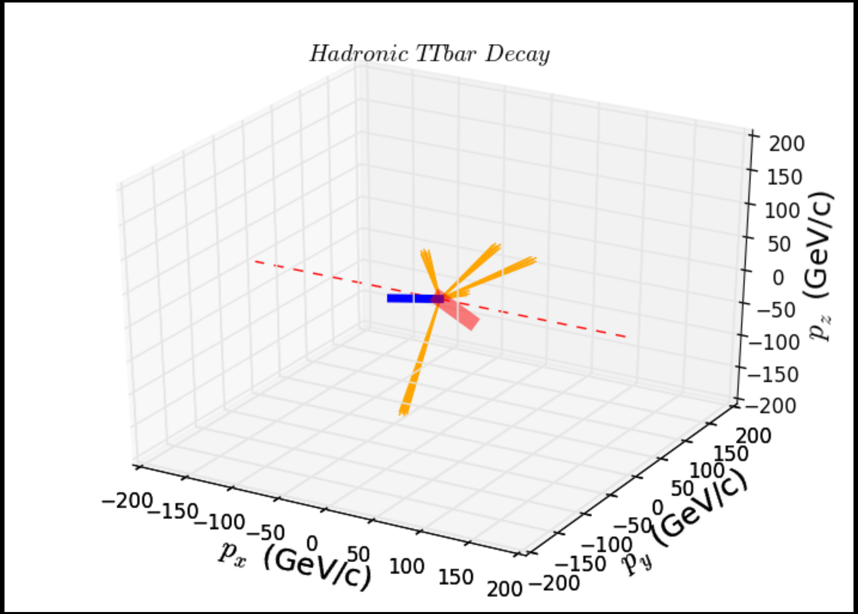
[*https://github.com/mattbellis/CMS-Made-Simple*](https://github.com/mattbellis/CMS-Made-Simple)

CMS Made Simple

```
import cms_made_simple as cms

filename = 'mc_ttbar.zip'

collisions = cms.get_collisions(filename)

for collision in collisions:

    jets,muons,electrons,photons,met = collision

    for jet in jets:
        energy,px,py,pz,btag = jet

    for muon in muons:
        energy,px,py,pz,charge = muon
....
```

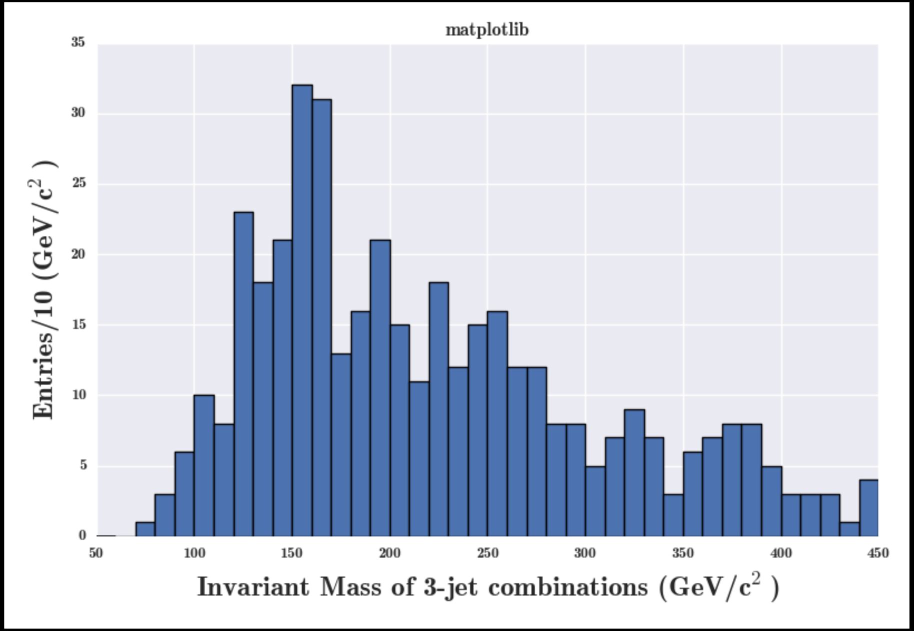
CMS Made Simple

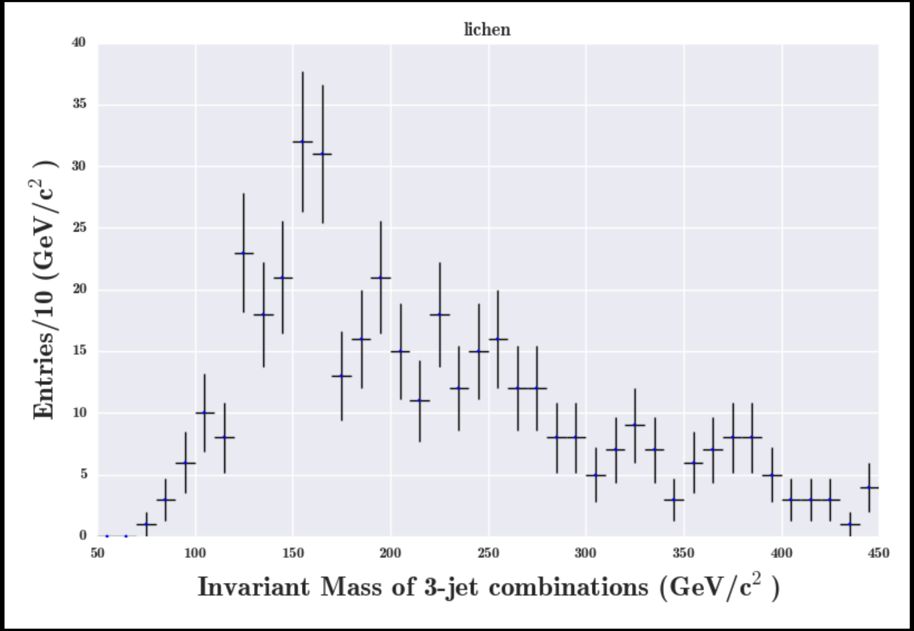




4-vector viewer

lichen





Wrapper to matplotlib.

https://github.com/mattbellis/lichen

iminuit

Python wrapper to Minuit

Written by *Piti Ongmongkolkul*

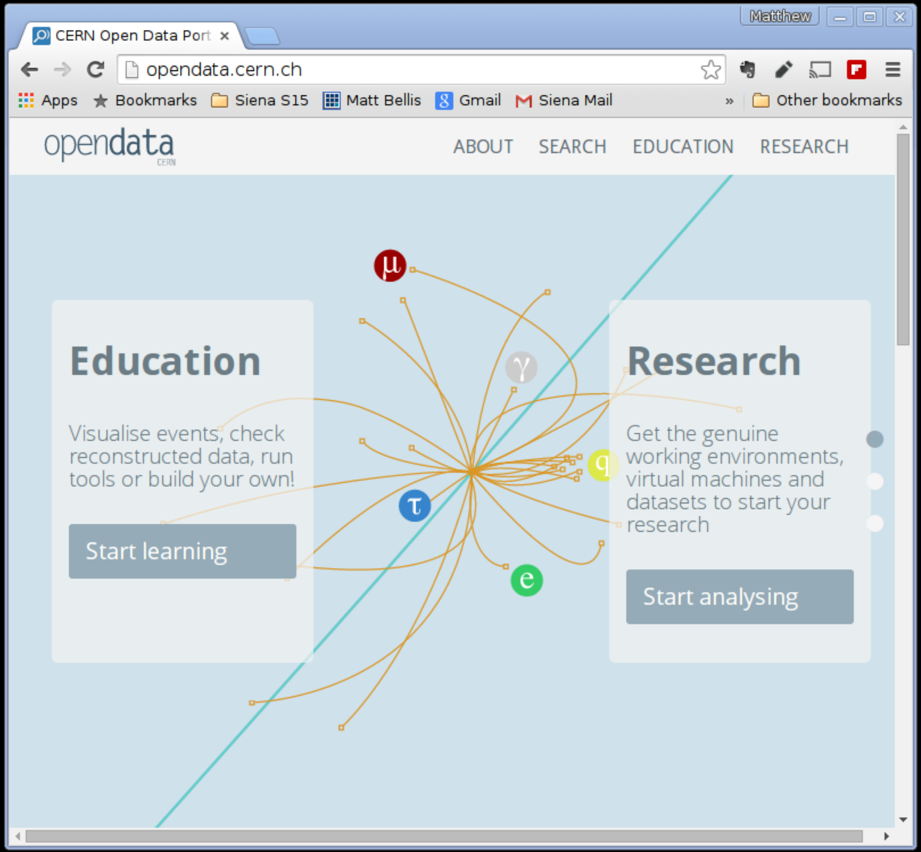
- Caltech grad student, BaBar

Most all of Minuit's functionality

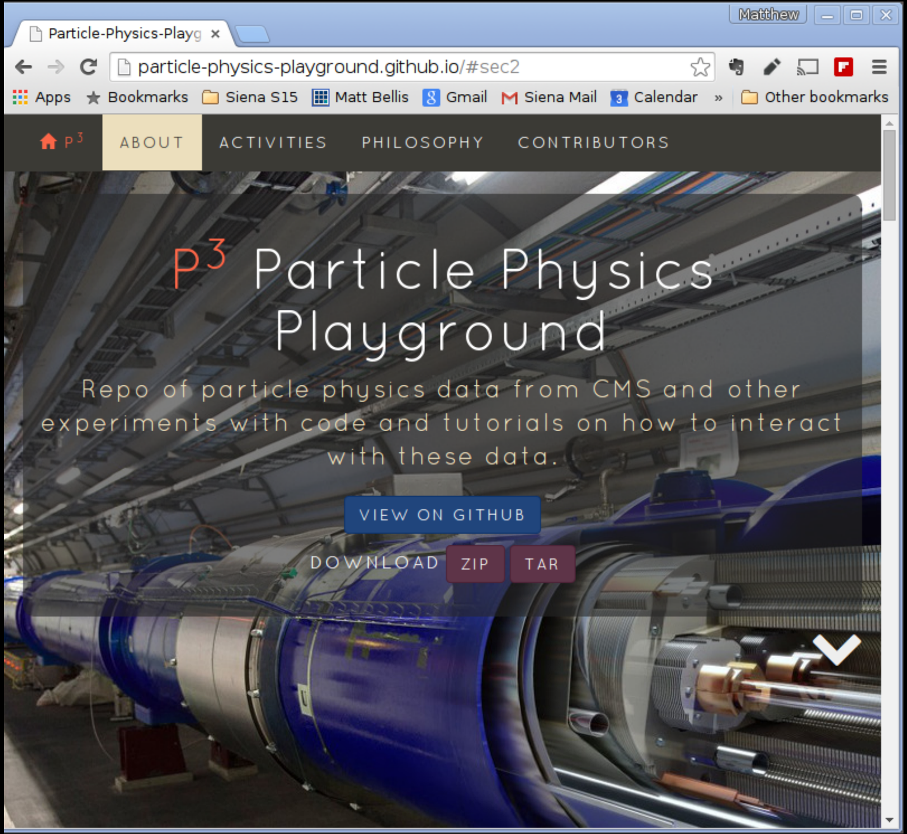
Allows me to teach what is going on under the hood

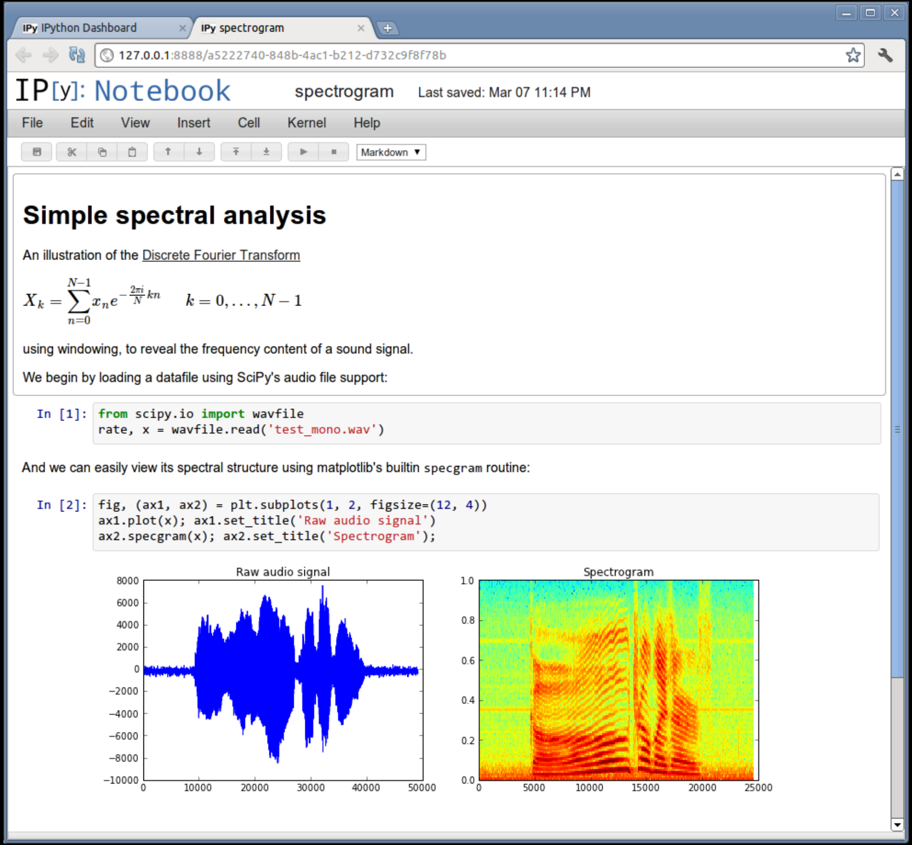
https://github.com/iminuit/iminuit

CERN Open Data Portal



Particle Physics Playground





Particle Physics Playground

nbviewer.ipython.org/g
Particle-Physics-Playground/blob/gh-pages/discovering_particles_1.ipynb
Apps | Google News | Matt Bellis | Gmail | Siena Mail | Calendar | Other Bookmarks

Challenge!

Copy this sample code and use it to calculate the mass of the muons. Make a histogram of this quantity.

Hint!

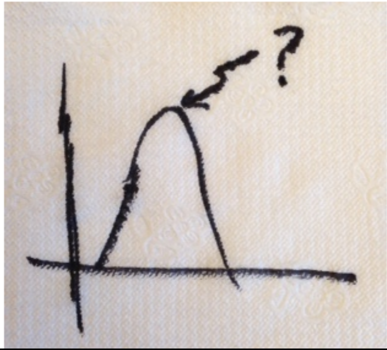
Make sure you do this for all the muons! Each collision can produce differing numbers of muons, so take care when you code this up.

Your histogram should look something like the following sketch. The value of the peak, should be the mass of the muon. Check your answer!

In [4]:

```
from IPython.display import Image
Image(filename='muons_sketch.jpeg')
```

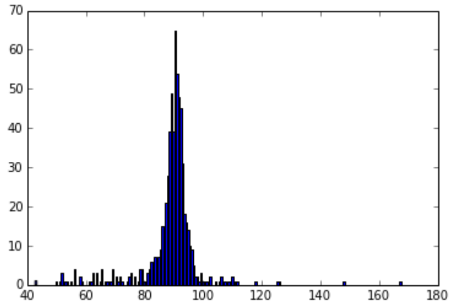
Out[4]:



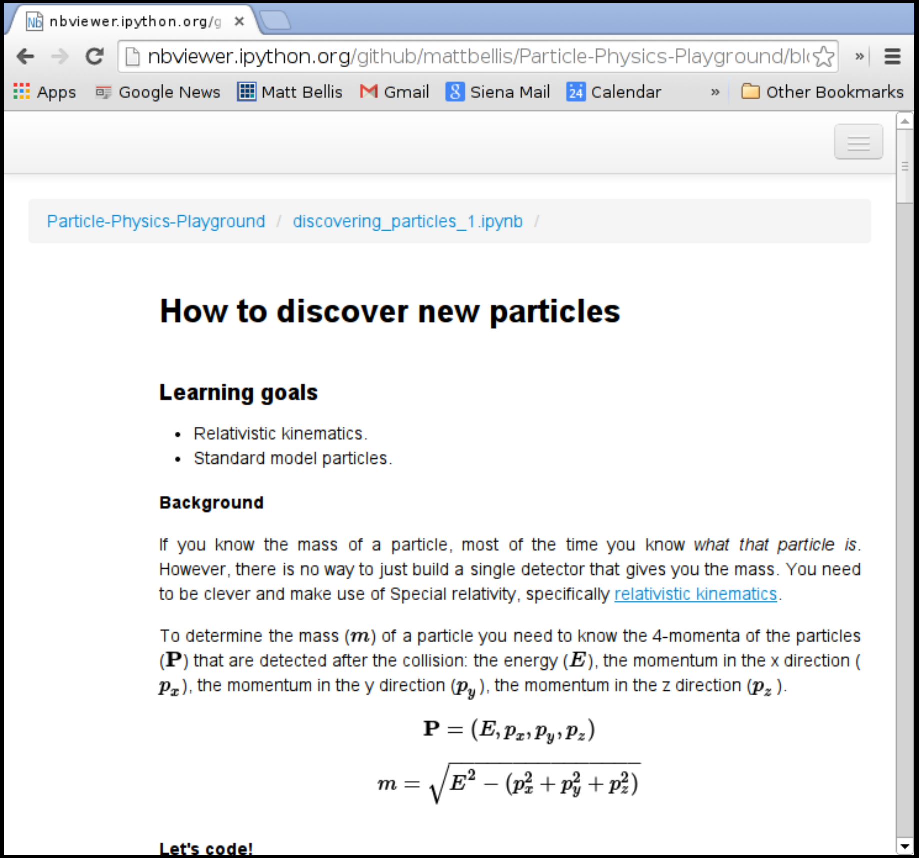
nbviewer.ipython.org/g | https://raw.githubuser | nbviewer.ipython.org/g
nbviewer.ipython.org/github/mattbellis/Siena_College_Amanda_Depoia
Apps | Google News | Matt Bellis | Gmail | Siena Mail | Calendar | Other Bookmarks

```
# To find the mass of a muon


    if len(muons) == 2:

        energy0,px0,py0,pz0,charge0 = muons[0]
        energy1,px1,py1,pz1,charge1 = muons[1]
        masssq = (energy0+energy1)**2 - ((px0+px1)**2 + (py0+py1)**2 +
(pz0+pz1)**2)
        if masssq > 0.0:
            mass.append(np.sqrt(masssq))

fig = plt.figure()
plt.hist(mass,bins=200)

plt.show()
```

Reading in the data....
1000



Particle Physics Playground


nbviewer.ipython.org/github/mattbellis/Particle-Physics-Playground/bl

Apps Google News Matt Bellis Gmail Siena Mail Calendar Other Bookmarks

Particle-Physics-Playground / discovering_particles_1.ipynb /

How to discover new particles

Learning goals

- Relativistic kinematics.
- Standard model particles.

Background

If you know the mass of a particle, most of the time you know *what that particle is*. However, there is no way to just build a single detector that gives you the mass. You need to be clever and make use of Special relativity, specifically relativistic kinematics.

To determine the mass (m) of a particle you need to know the 4-momenta of the particles ($\mathbf{P}$) that are detected after the collision: the energy (E), the momentum in the x direction (p_x), the momentum in the y direction (p_y), the momentum in the z direction (p_z).

$$\mathbf{P} = (E, p_x, p_y, p_z)$$

$$m = \sqrt{E^2 - (p_x^2 + p_y^2 + p_z^2)}$$

Let's code!

Will soon be adding CLEO data, as well as more CMS data.

Have spoken with ATLAS about contributing

Will reach out to BaBar and LHCb

Python and GPUs

Continuum Analytics

- Anaconda distribution

`numba.cuda`

Wrapper to CUDA libraries

Slower than compiled C/CUDA, but….

Can engage my students with Python

Exploring nearest-neighbor density with multidimensional fits

Comments and inquiries are welcome!

Student poster at GPU Tech Conference 2015
``Undergraduate GPU-enabled Research Through Python"
http://on-demand.gputechconf.com/gtc/2015/posters/GTC_2015_Education___Training_01_P5236_WEB.pdf

Student poster at April APS meeting
``CMS Made Simple: A ROOT-less Workflow for Educating Undergraduates about CMS Analysis"
https://cms-mgt-conferences.web.cern.ch/cms-mgt-conferences/conferences/pres_display.aspx?cid=1550&pid=10635

http://particle-physics-playground.github.io/

https://github.com/mattbellis/SCMS-Made-Simple

https://github.com/mattbellis/lichen

https://github.com/iminuit/iminuit

http://ipython.org/notebook.html

http://docs.continuum.io/numbapro/CUDAJit.html

https://github.com/mattbellis/GTC15-Python-and-CUDA