Web-based interactive analysis environment (Wise)

Tao Lin <lintao@ihep.ac.cn>

May 10th - 15th , 2018

JUNO workshop @WHU

Outline

- Introduction and design
- Jupyter
- SWAN
- Summaries and Plans

Introduction

- Physics analysis in JUNO is a challenge.
  - PB-scale data every year.
  - Time correlated events.
- Two steps in analysis procedures (general analysis model)
  - Using framework to process data containing Event Data Model.
  - Using standalone program to process user data, fit data, produce plots.
- For physics users, the second step is quite important. Need the ability to analyze data interactively.
  - Easy to access or get data.
  - Fitting data and tuning parameters.
  - Visualization, or interactive detector simulation.

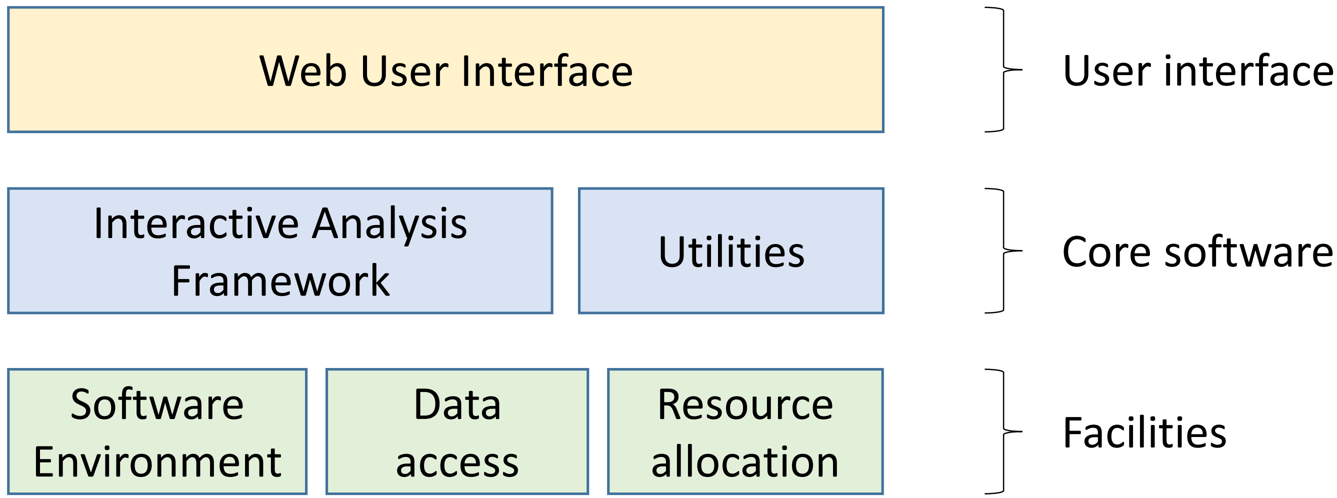
Challenges in interactive analysis

- Computing powers.
  - The current computing mode is that users develop in login node, and submit jobs to computing nodes.
  - The ability to do interactive analysis in login node is limited.
  - Not possible to use multiple nodes to accelerate data processing in interactive analysis.
  - Not easy to download the event data to local, run software locally.
- Software functionalities.
  - Not easy to access existing services in framework.
- New analysis technologies.
  - Machine learning and deep learning are popular.
  - Different software stacks. Such as CUDA, Python+TensorFlow.
  - How to help users?

“Analysis as a Service”

- To improve the abilities, “Analysis as a Service” is proposed by several communities. Such as
  - Jupyter and JupyterHub.
  - CERN’s SWAN (Service for Web based ANalysis)
- Adopt a cloud-based solution for JUNO analysis:
  - Elastic computing power and storage.
  - Flexible software configuration.
  - Integration with framework, new technologies.
  - Thin client. Just a web browser.
- Wise: Web-based interactive analysis environment
  - Simplify the analysis within this environment.
  - Not start from scratch, but based on existing Jupyter, SWAN and so on.

Designs



Need to consider:

- Web-based software and data
- Resource allocation and management
- Interactive analysis framework
- Monitoring and validation utilities

Based on following:

- ■ Jupyter: UI and kernel
- ■ SWAN: Architecture.
- ■ ROOT6: Libraries.
- ■ SNiPER: Framework

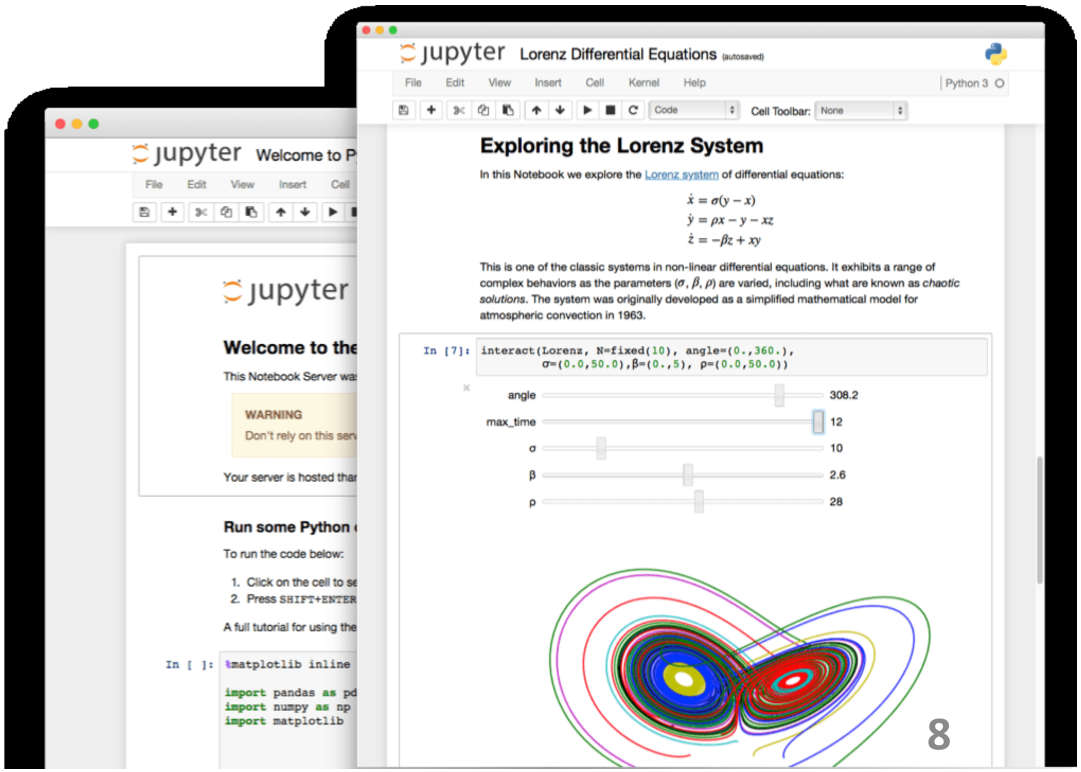
Jupyter

Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

https://jupyter.org/

Jupyter notebook interface

- Web-based.
- It combines code, text, figures in the same document.
  - Multiple programming language
  - Markdown support.
  - Big data integration
- Widely used in
  - Python community
  - Data analysis
  - Machine learning



Jupyter notebook

Currently in use at

Google
Microsoft
IBM
Bloomberg
O'REILLY
ANACONDA
rackspace the #1 managed cloud company
SOUNDCLOUD
Quantopian
NetApp
software carpentry
hhmi janelia Research Campus
<CODE NEURO>
N-Site LLC
COCALC
BRYN MAWR COLLEGE
CAL POLY SAN LUIS OBISPO
Berkeley UNIVERSITY OF CALIFORNIA
The University Of Sheffield.
THE GEORGE WASHINGTON UNIVERSITY WASHINGTON, DC
CLEMSON UNIVERSITY
MICHIGAN STATE UNIVERSITY
Northwestern University
NYU
NASA
AYASDI
The Data Incubator

Jupyter notebook

Simple spectral analysis

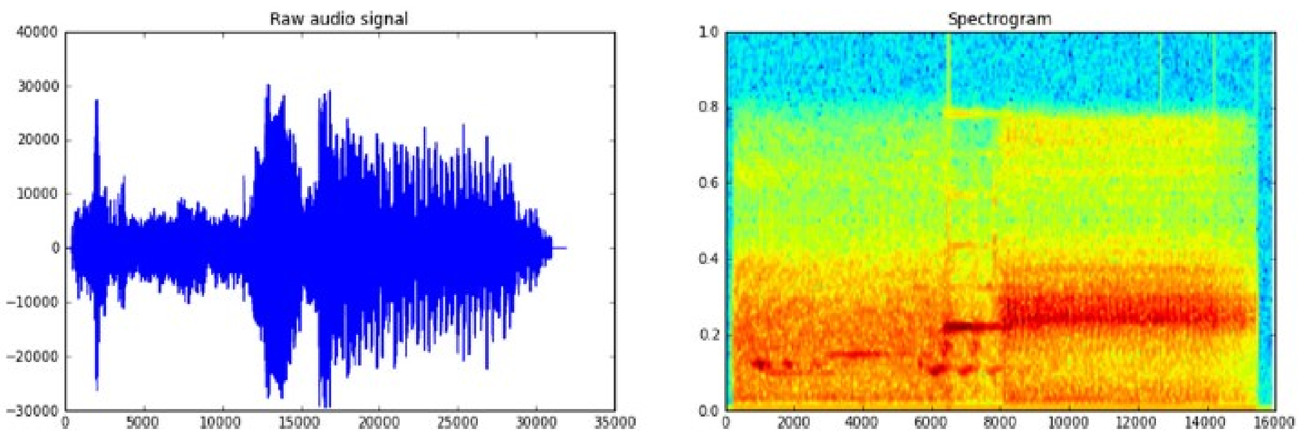
An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n exp^{\frac{-2\pi i}{N}kn} \quad k = 0, \ldots, N-1$$

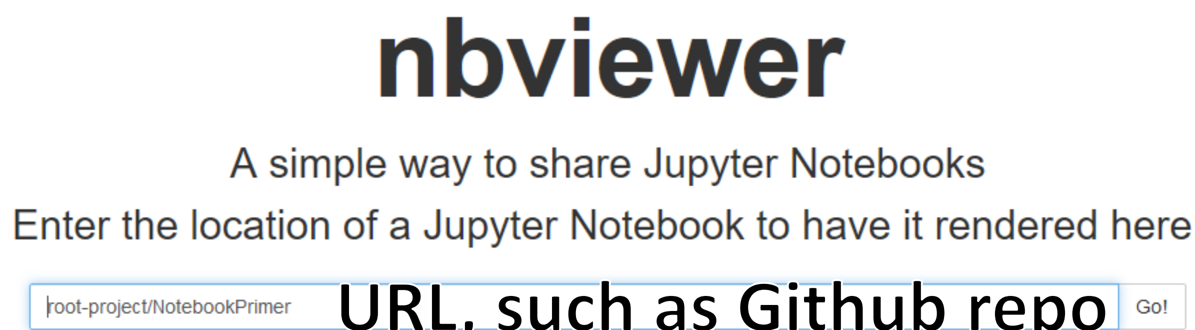
In [2]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

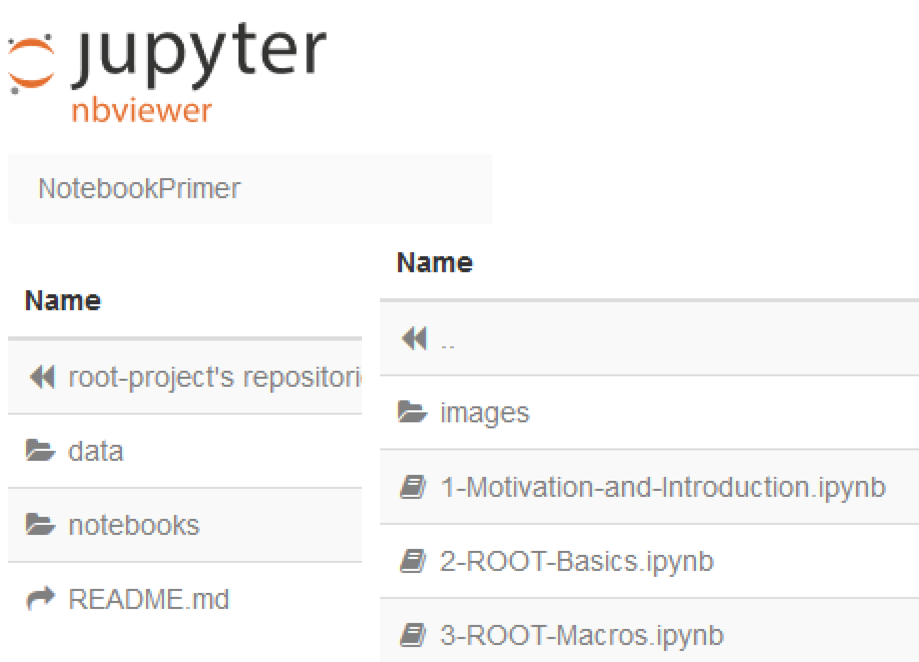
In [5]:
```
fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```

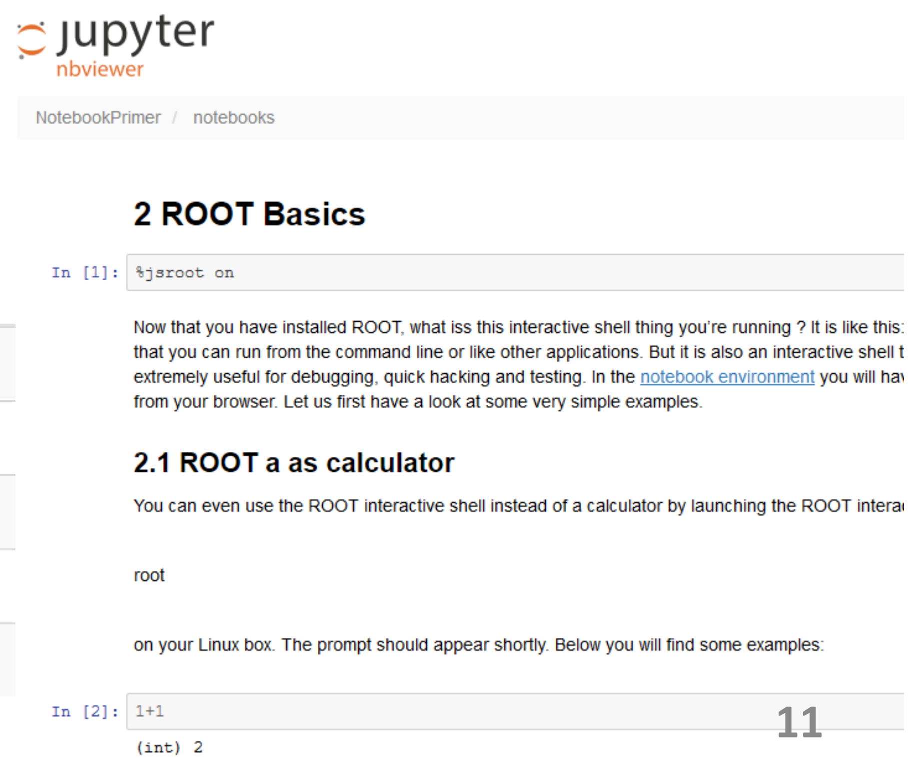


nbview: online viewer



URL, such as Github repo





SWAN

SWAN (Service for Web based ANalysis) is a platform to perform interactive data analysis in the cloud.

https://swan.web.cern.ch/

SWAN[#]

- A platform to perform interactive data analysis in the cloud.
- Analyze data without the need to install any software
- Jupyter notebook interface as well as shell access from the browser
- Use CERNBox as your home directory and synchronize your local user storage with the cloud
- Access experiments' and user data in the CERN cloud
- Share your work with your colleagues thanks to CERNBox
- Document and preserve science - create catalogues of analyses: encourage reproducible studies and learning by example

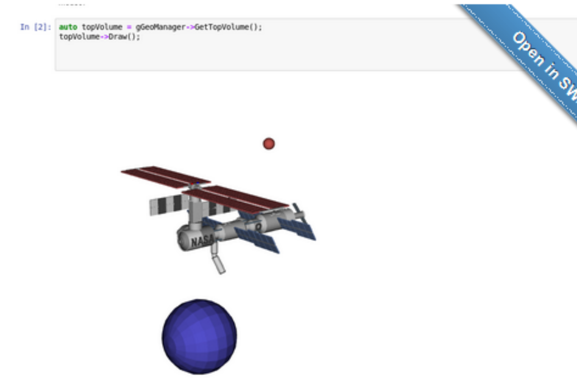
 # https://swan.web.cern.ch/

Examples

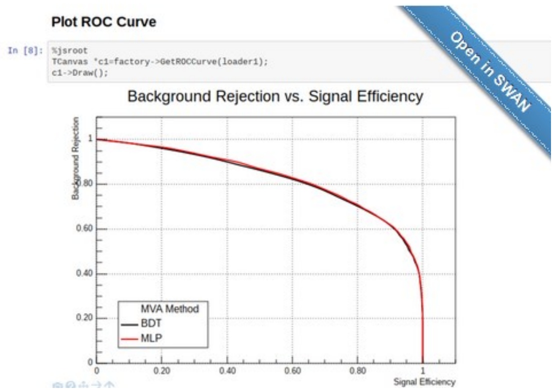
Simple ROOTbook (Python)

Simple I/O

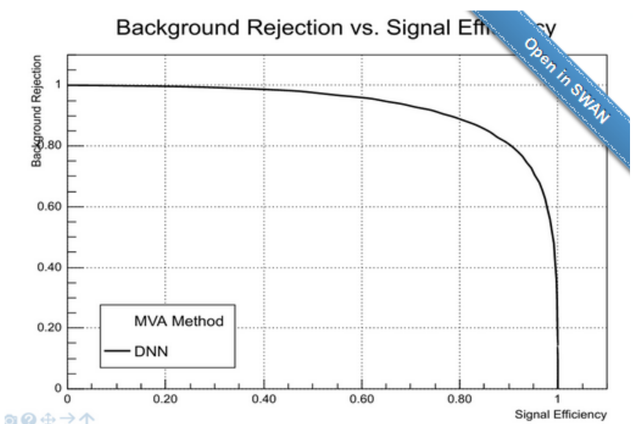
3D Visualisation



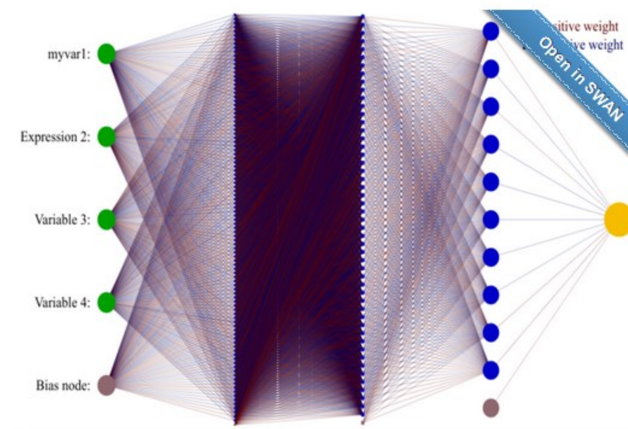
TMVA Basics



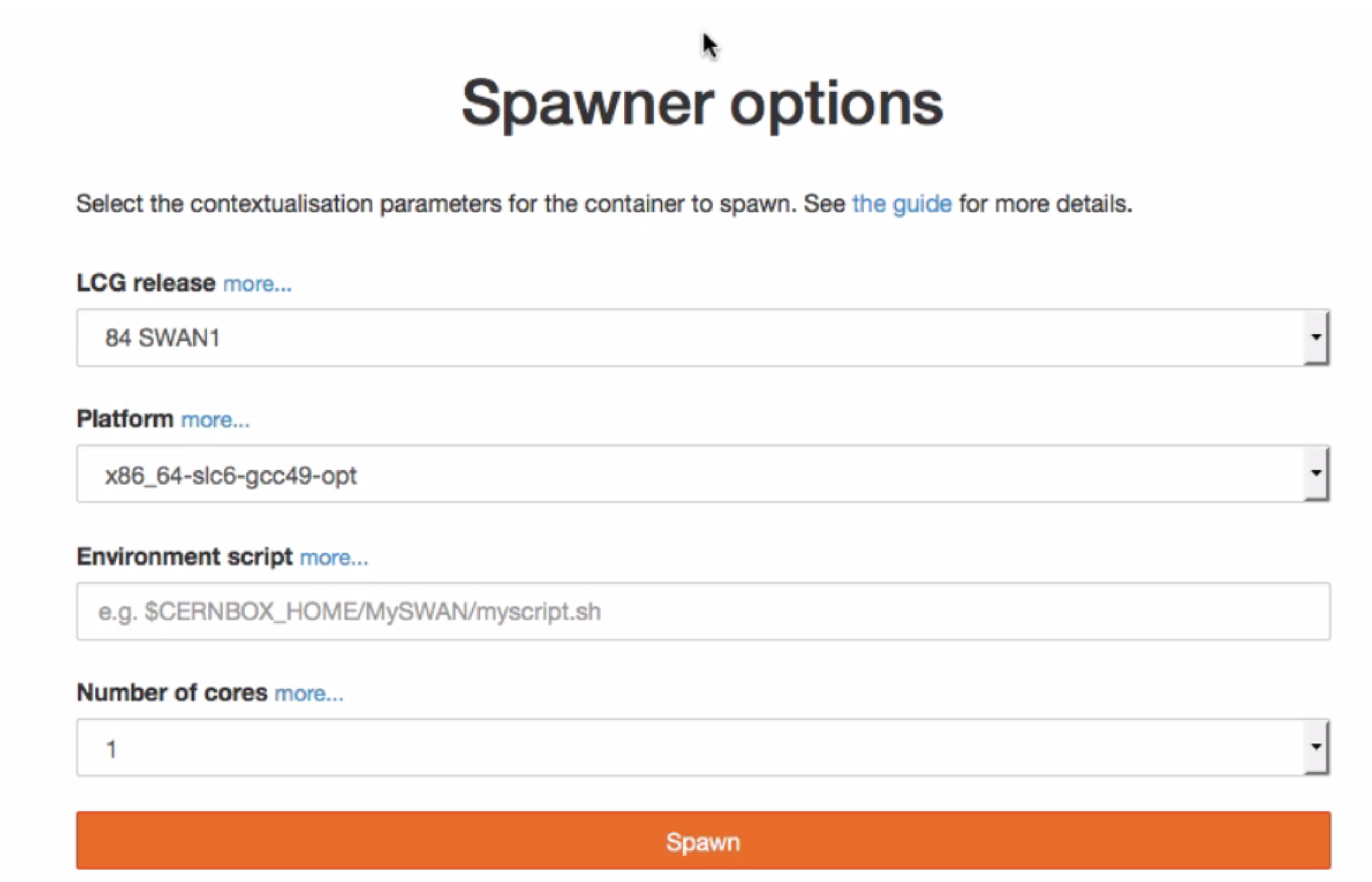
DNN



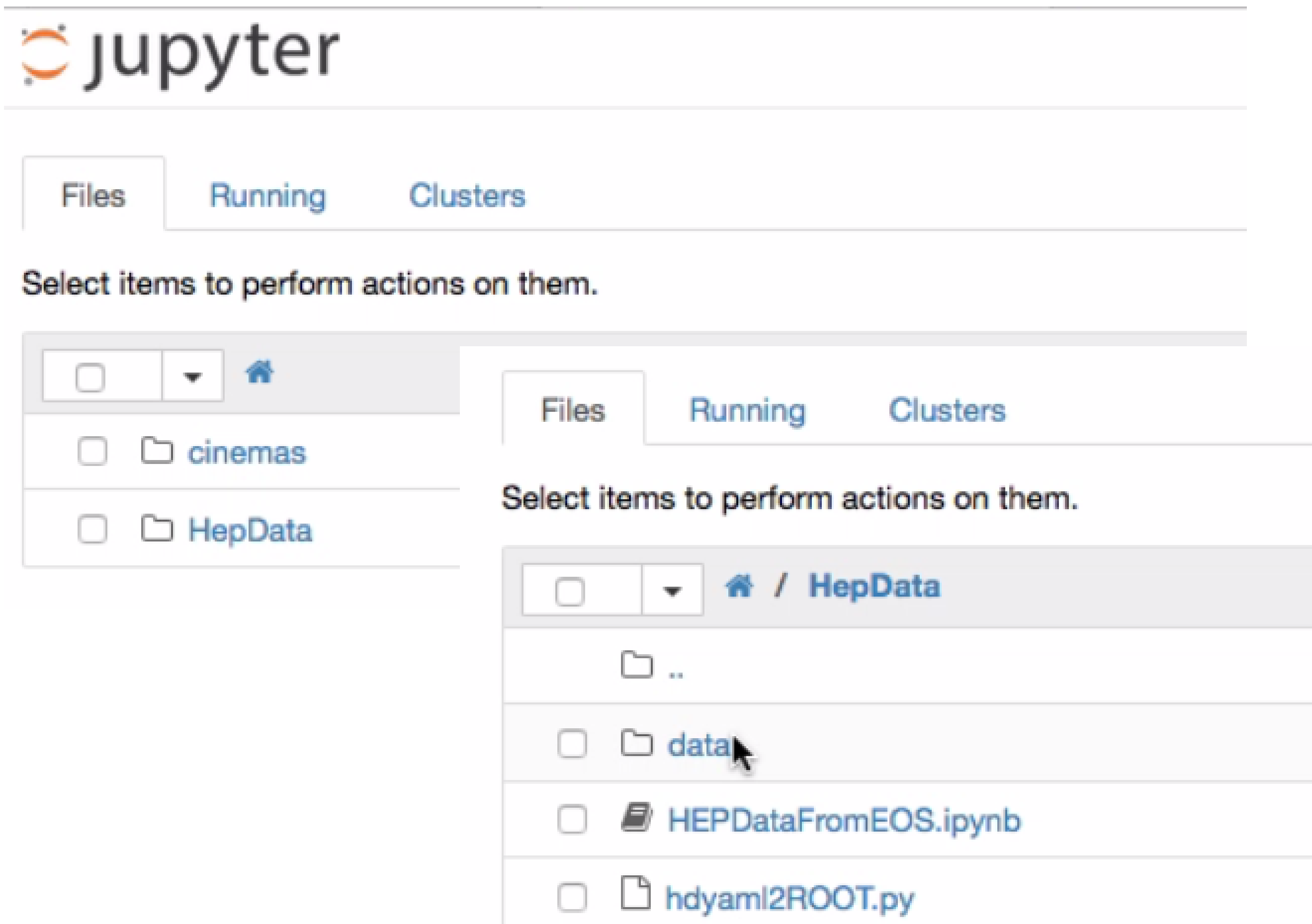
TMVA in Jupyter



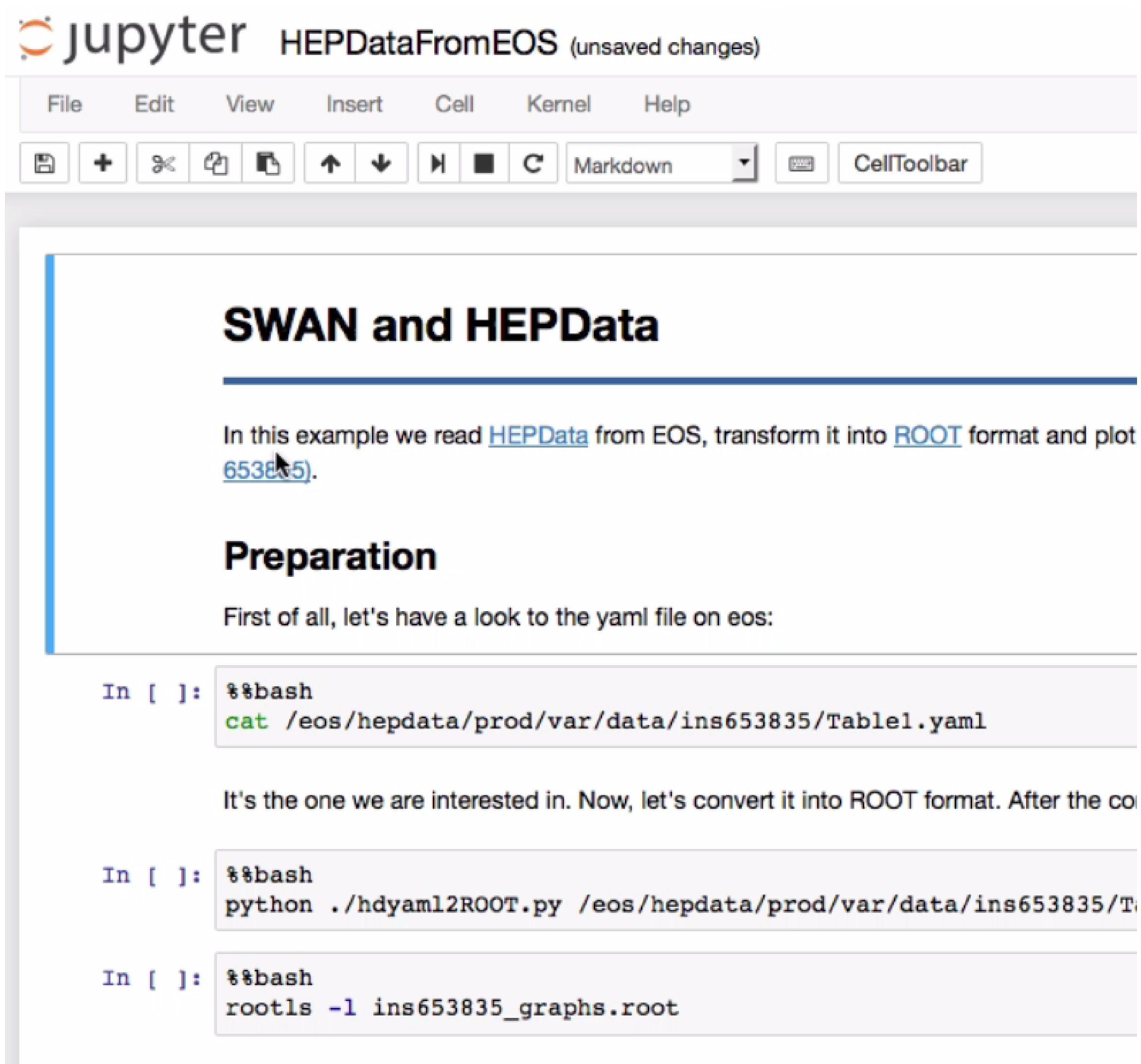
demo: choose container



demo: file explorer



demo: jupyter notebook



demo: ROOT

jupyter HEPDataFromEOS (unsaved changes)

File Edit View Insert Cell Kernel Help

Markdown CellToolbar

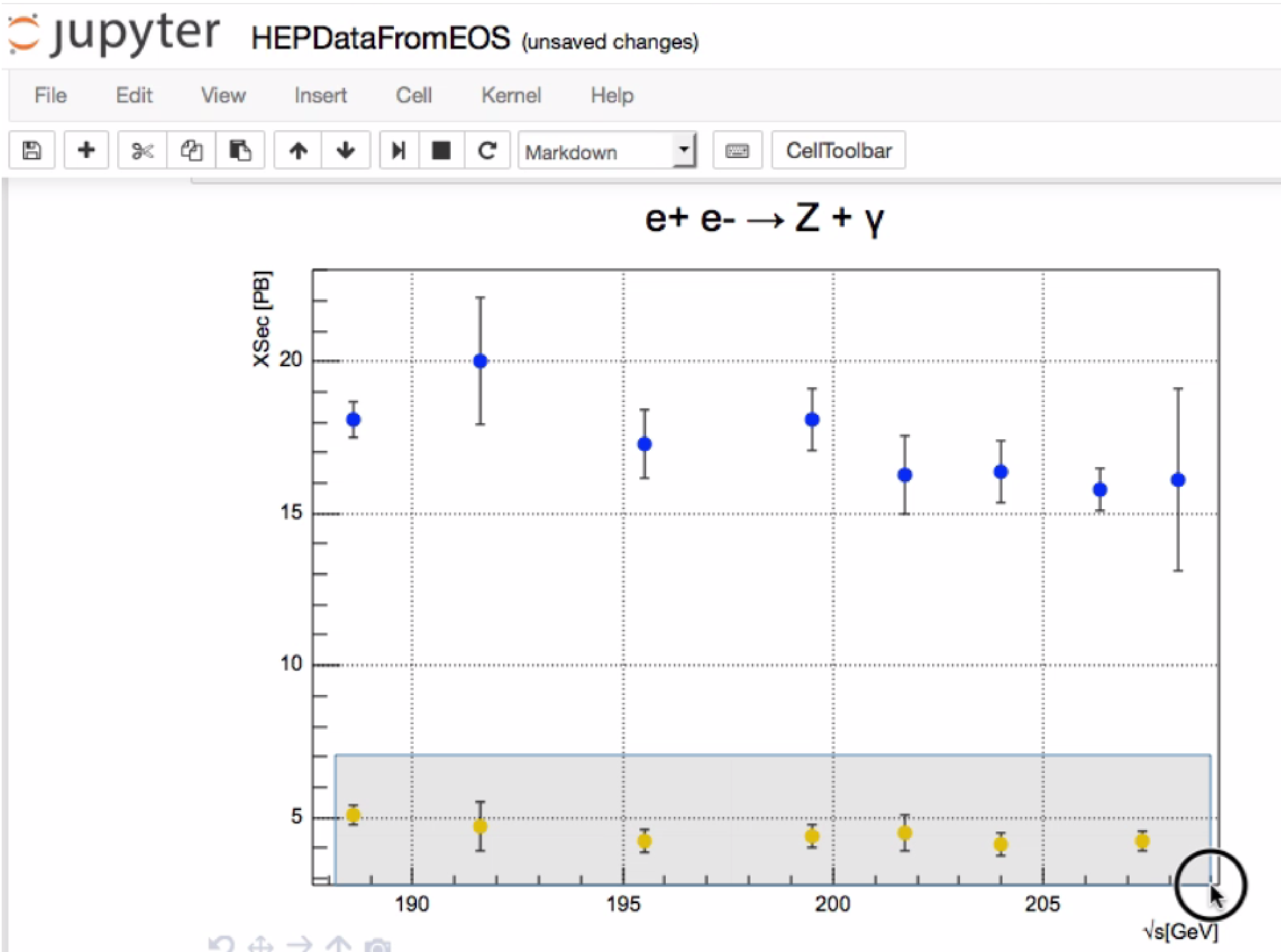
```
g2 = iFile.g2
```

Here the canvas, the entity which in ROOT holds graphics primitives, is cr
added to a TMultiGraph, which will then be drawn.

In [7]:
```
c = ROOT.TCanvas()
for g in [g1,g2]:
    g.SetMarkerStyle(ROOT.kFullCircle)
    g.SetMarkerSize(1)
g1.SetMarkerColor(ROOT.kOrange)
g1.SetTitle("Z #rightarrow #nu #nu}")
g2.SetMarkerColor(ROOT.kAzure)
g2.SetTitle("Z #rightarrow had")

mg = ROOT.TMultiGraph()
mg.SetTitle("e^{+} e^{-} #rightarrow Z + #gamma;#sqrt{s
mg.Add(g1)
mg.Add(g2)
```

demo: interactive



demo: save results

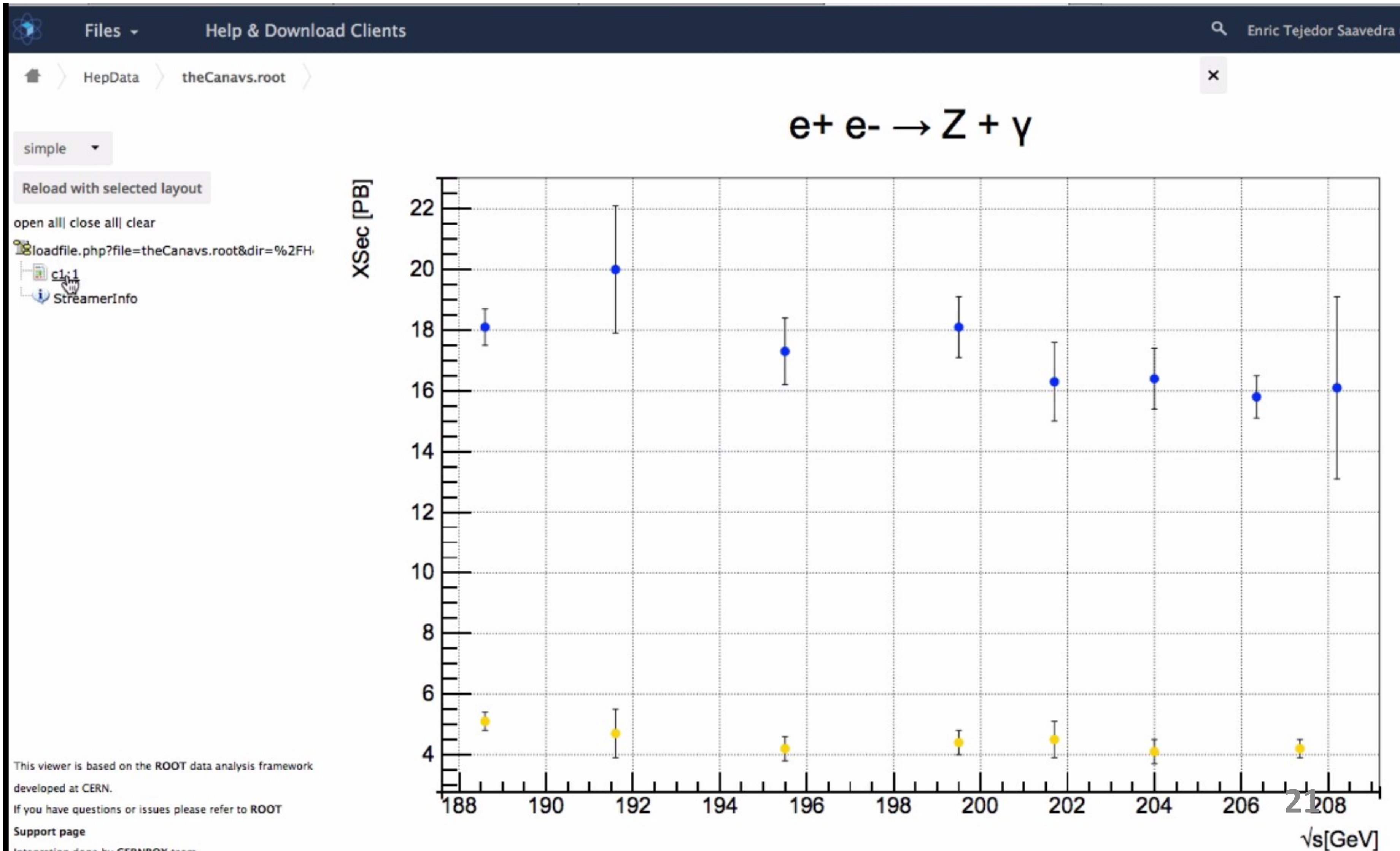
We now save our plot in pdf format on disk.

In [9]:
```
c.Print("myStudy.pdf")
```

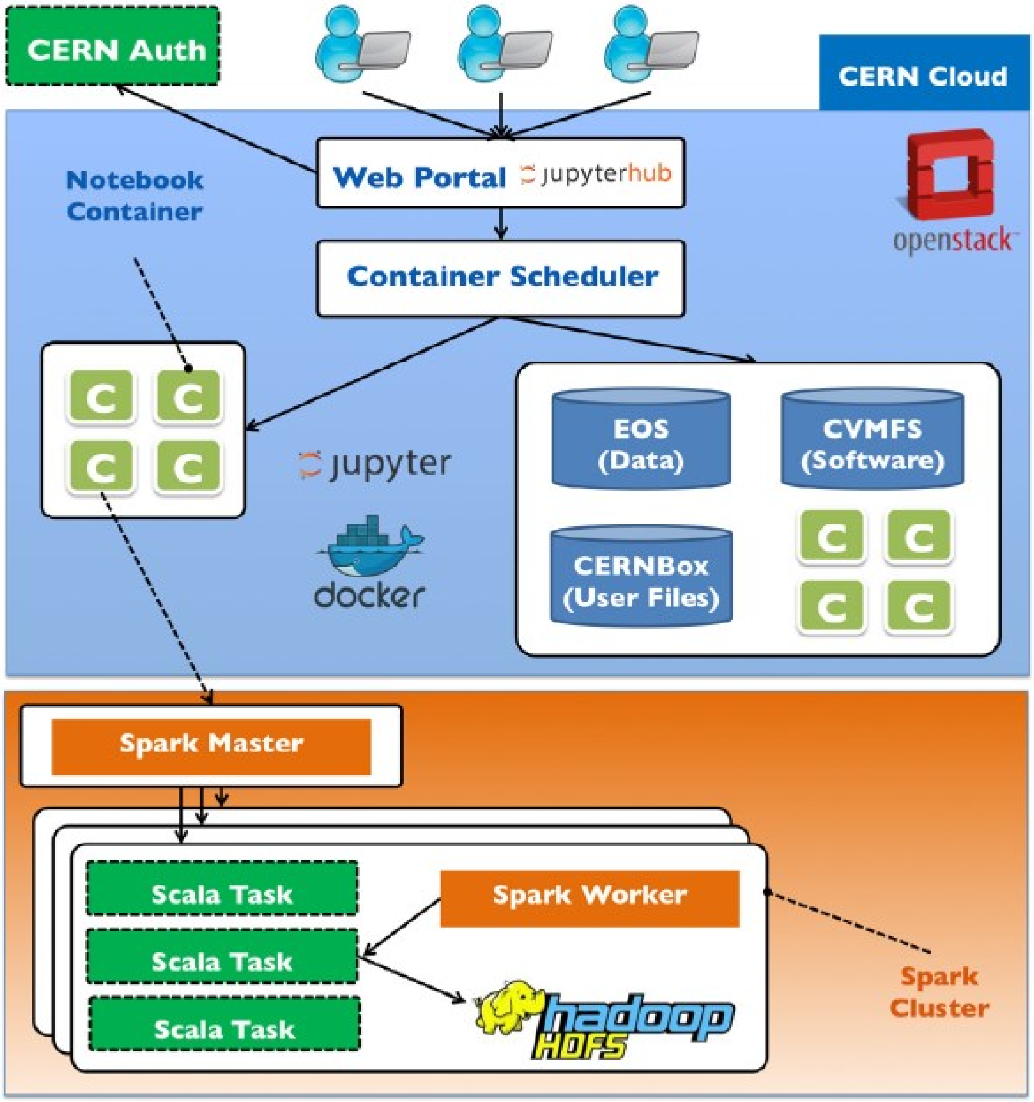
```
Info in <TCanvas::Print>: pdf file myStudy.pdf has been created
```

In [10]:
```
f=ROOT.TFile("theCanavs.root","RECREATE")
c.Write()
f.Close()
```

Demo: result in CERNBox



Architecture of SWAN



Authentication and security

Web portal

Virtualized infrastructure

Containers

Software distribution

Analysis software

Storage

Asynchronous data processing

- Spark
- HTCondor

Note: only available for CERN.

Summaries and Plans

- We propose a new project called Wise. It is "Analysis as a Service" for JUNO.
  - Based on Jupyter and SWAN.
  - Extend the software functionalities, such as SNiPER framework support.
- A preliminary design
  - User interface is based on web.
  - A core software layer is used to integrate Jupyter and SNiPER.
  - Underlying layer to supply software distribution, data access, computing resource allocation and so on.
- Plans
  - Setup a prototype before December in 2018.