The New ROOT Interface: Jupyter Notebooks

Enric Tejedor for the ROOT team
CERN

CHEP
10/10/2016
**Notebook**: A web-based interactive computing interface and platform that combines code, equations, text and visualisations.

Many supported languages: Python, Haskell, Julia, R ... One generally speaks about a “kernel” for a specific language.

In a nutshell: an “interactive shell opened within the browser”
Access TTree in Python using PyROOT and fill a histogram

Loop over the TTree called "events" in a file located on the web. The tree is accessed with the dot operator. Same holds for the access to the branches: no need to set them up - they are just accessed by name, again with the dot operator.

```python
import ROOT
h = ROOT.TH1F("TracksPt", "Tracks;Pt [GeV/c];#", 128, 0, 64)
for event in T.events:
    for track in event.tracks:
        h.Fill(track.Pt())
c = ROOT.TCanvas()
h.Draw()
c.Draw()
```
• ROOT has been fully integrated with Jupyter notebooks

• Notebook features appealing to ROOT:
  – **Sharing**: scientists can share their results (code, plots, text) in the form of notebooks
  – **Teaching**: runnable tutorials and exercises, combining code and explanations
  – **Reproducibility**: a notebook contains results and the code that led to them
Two language flavours (a.k.a. kernels) are available:

- Powered by the ROOT C++ interpreter
- Via PyROOT

New in Jupyter!
C++ and Python can be mixed in the same notebook
  - Thanks to the ROOT type system

Interleave Python with C++: the %%%cpp magic

In [1]:
```python
import ROOT

Welcome to JupyROOT 6.07/03
```

Thanks to its interpreter and type system, entities such as functions, classes and variables, created in a C++ cell, can be accessed from within Python.

In [2]:
```cpp
class A {
public:
  A() { cout << "Constructor of A!" << endl; }
};
```

In [3]:
```python
a = ROOT.A()

Constructor of A!
```

%%python also available in C++ notebooks
Both of the presented flavours (C++, Python) allow to **inline ROOT graphics** in a notebook.

Two modes: static image and **JavaScript visualisation**
- Activate JSROOT mode with `%jsroot on`
- Interact with your plot: zoom, modify axis, inspect bins, …
• TMVA: machine learning toolkit in ROOT
  – Recently integrated with Jupyter as well: `%jsmva on`
  – JSROOT plots for input variables
  – Visualisation of neural networks and decision trees, DNN designer
  – Interactive training: stop a server computation
  – HTML output formatting
Interactive Machine Learning (II)

Decision trees

Interactive training

HTML output

<table>
<thead>
<tr>
<th>DataSetInfo</th>
<th>Correlation matrix (Signal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataSetInfo</td>
<td>Correlation matrix (Background)</td>
</tr>
<tr>
<td>DataSetFactory</td>
<td>Dataset: tmva_class_example</td>
</tr>
<tr>
<td>TFHandler_MLP</td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td>myvar1</td>
</tr>
<tr>
<td></td>
<td>myvar2</td>
</tr>
<tr>
<td></td>
<td>var3</td>
</tr>
<tr>
<td></td>
<td>var4</td>
</tr>
</tbody>
</table>

Training Network
Elapsed time for training with 6000 events : 4.45 sec

MLP
Dataset: tmva_class_example Evaluation of MLP on training sample (6000 events)
Elapsed time for evaluation of 6000 events : 0.0187 sec
Creating xml weight file: tmva_class_example/weights/TMVAClassification_MLP.weights.xml
Creating standalone class: tmva_class_example/weights/TMVAClassification_MLP.class.C
Write special histos to file: TMVA.root:/tmva_class_example/Method_MLP/MLP
Follow some simple instructions in:
https://root.cern.ch/how/how-create-rootbook
and…

$ root --notebook

This command:
1. Starts a local notebook server
2. Connects to it via the browser

Since 6.06

Provides a ROOT C++ kernel and the rest of ROOTbook goodies
SWAN: Data analysis “as a service”

https://swan.cern.ch

**Interface:** Jupyter Notebooks

**Goals:**

- Analysis **only with a web browser**
  - Platform independent ROOT-based data analysis
- Calculations, input and results **“in the Cloud”**
  - Easy sharing of scientific results: plots, data, code
- Centrally-distributed **software**: CVMFS
  - Integration with other analysis ecosystems: R, Python, …
Gallery of notebooks at swan.web.cern.ch

“Notebookised” tutorials at root.cern

Click to open in SWAN!
• ROOT integrated with Jupyter notebooks
  – C++ and Python notebook flavours
  – Inline graphics
  – JSROOT interactive visualisation
  – TMVA interactive features
  – Other goodies: tab completion, language mixing, …

• All available in the next ROOT release (6.08)

• Accessible online thanks to SWAN
  – https://swan.cern.ch
Try It Out! - ROOT Binder

http://mybinder.org/repo/cernphsft/rootbinder

ROOT is a framework for data processing, born at CERN, at the heart of high-energy physics research. Every day, thousands of physicists use ROOT applications to analyze petabytes of data or to perform simulations.

Try a ROOTbook now: choose your favourite language!

Anonymous access

View, Create and Run ROOTbooks!