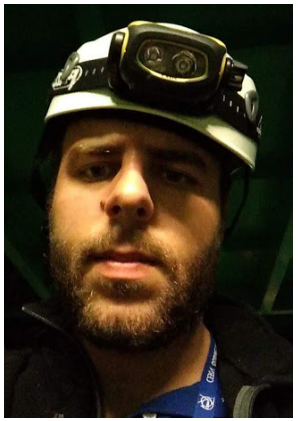
# My SWAN experience as an ALICE analyzer

Nicolò Jacazio
INFN - Bologna


11/10/2019

# About me

*“Ask not what your SWAN can do for you, ask what you can do for your SWAN”*

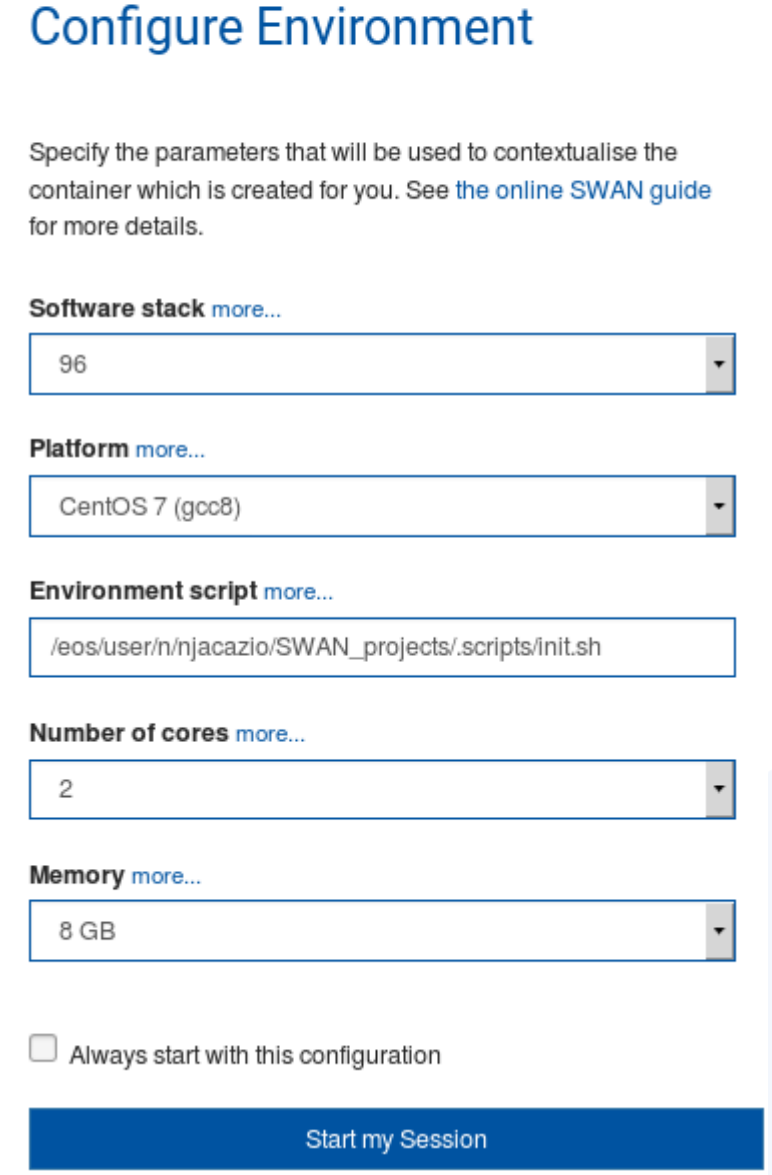


https://about.me/jacazio

- I’m a research fellow at INFN CNAF in Bologna (Italy), I’m part of the ALICE Collaboration
- I’m involved in several projects including **physics analyses**, **code development** for the detector upgrade, **detector performance studies**..
- I used SWAN on a very low level, so far it worked quite nicely and now I’m trying to return the favor

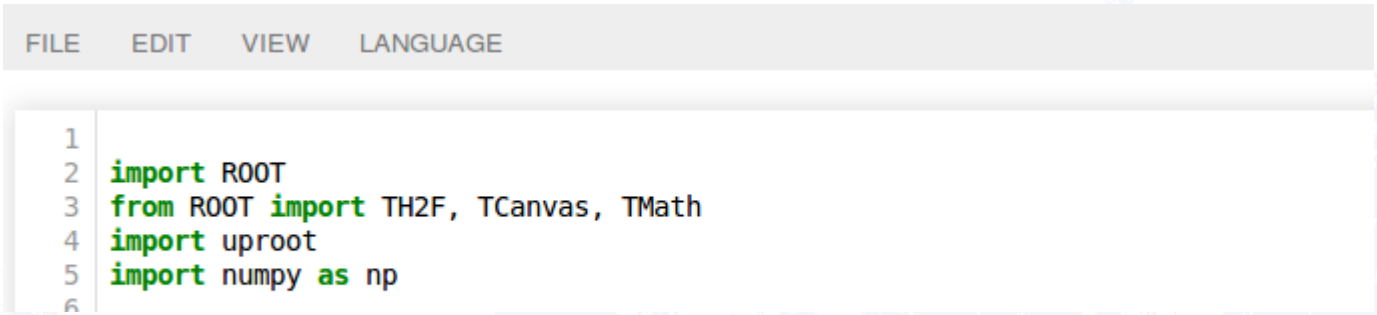
# How SWAN works for me

- Highly **customizable**, very **versatile**, **fast**
  - » Customization is very simple
  - » **Machine specs are OK** for most of purposes but could be limiting for some cases
  - » **Custom init script** useful to setup experiment software



# How SWAN works for me

- **Jupyter notebook interface is the key**
  - » **Python+root** together are quite powerful
  - » TTrees can be converted in pandas data frames and be used handily
  - » Convenient input for ML studies

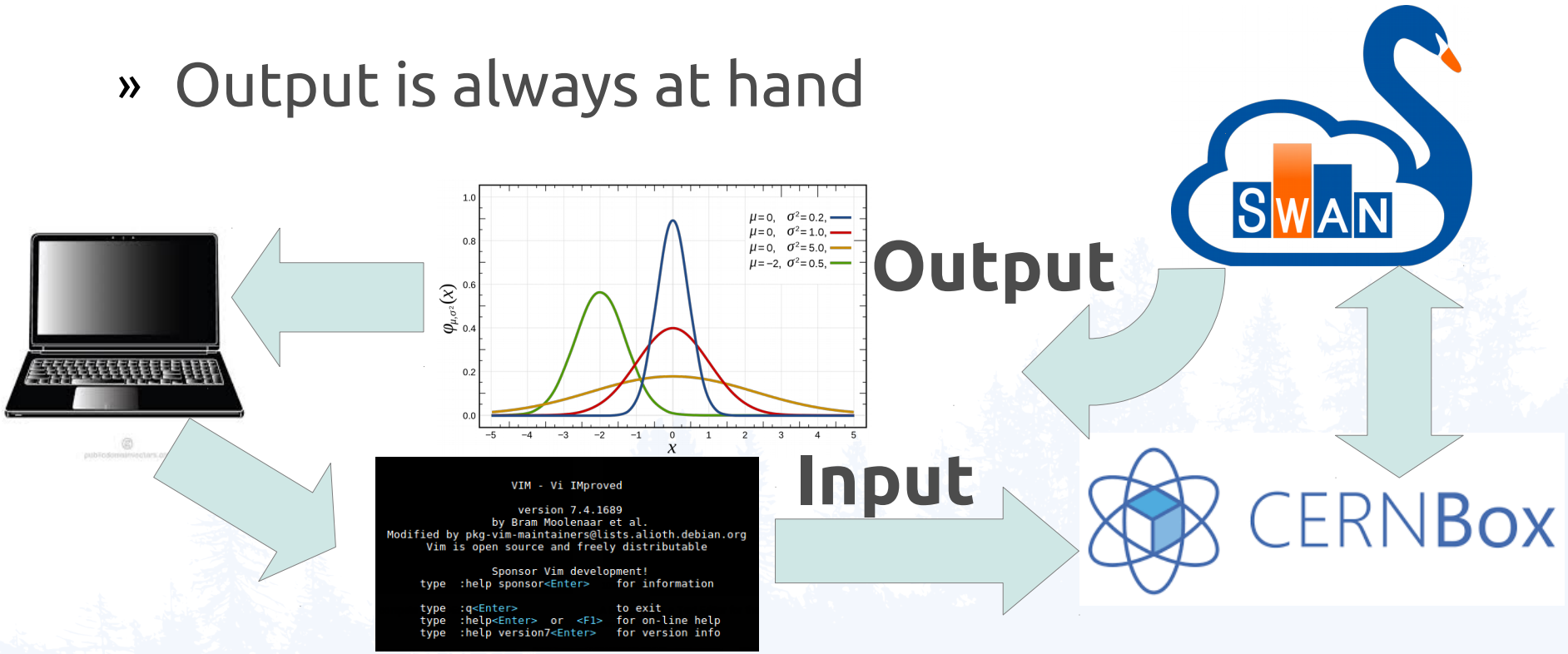


# How SWAN works for me

- **No installation required**
  - » Preset and **ready for use**
  - » Useful for newcomers and **students**!
  - » Many **python libraries** are already installed
  - » Several **ML utilities** ready for use
- Usage with **cernbox and EOS** is great
  - » Develop locally, run your code **both on-line and offline**
  - » Output is always at hand

# How SWAN works for me

- Usage with **cernbox and EOS** is great
  - » Develop locally, run your code **both on-line and offline**
  - » Output is always at hand



# How SWAN works for me

- **Isolated environment**
  - » Scripta manent (especially on EOS)
  - » **Easy to share** with colleagues
- Very useful for **testing and developing** new software
  - » Many small projects instead of few larger ones
- Useful for **physics analyses** as well
  - » Useful to prepare physics plots, read/write hepdata, toy MC

# Summary

- SWAN is a very useful service for analysers
- It fits perfectly most of the needs of the daily work
- The quality and stability of the service is very good
- So far my experience is very positive without any major complain
- For sure more people could profit of this


Thank you!

# ML with Keras

- **Works straight away!**

In [1]:

```
import numpy
import matplotlib.pyplot as plt
import numpy as np
from keras.models import Sequential
from keras.models import model_from_yaml
from keras.layers import Dense, Activation, Dropout
from keras.utils import plot_model
from ROOT import TTree, TFile, TH1F, TCanvas, TH2F, TH1D
import os
```

```
Using TensorFlow backend.
```

```
Welcome to JupyROOT 6.18/00
```

+

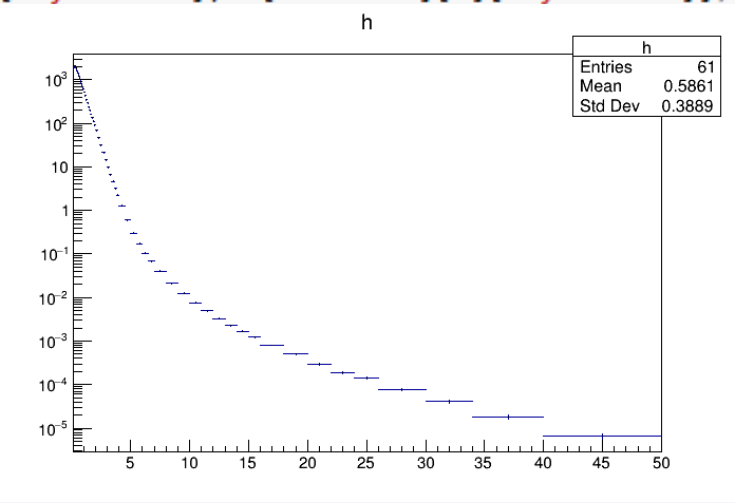
More code magic

# Reading HEPData

In [44]:
```
import yaml
from ROOT import TH1F, TCanvas, gPad
from numpy import array

```

In [47]:
```
with open("HEPData-ins1657384-v1-Table_1.yaml", 'r') as stream:
    try:
        Origcontent = yaml.safe_load(stream)
        x = []
        y = []
        content = Origcontent["independent_variables"][0]
        for i in content["values"]:
            x.append(i["low"])
            if i == content["values"][-1]:
                x.append(i["high"])
        x = array(x)
        content = Origcontent["dependent_variables"][0]
        for i in content["values"]:
            y.append([i["value"], i["errors"][0]["symerror"], i["errors"][1]["symerror"]])
        h = TH1F("h", "h", len(x) - 1, x)
        hs = TH1F("hs", "hs", len(x) - 1, x)
        for i in enumerate(y):
            h.SetBinContent(i[0]+1, i[1][0])
            hs.SetBinContent(i[0]+1, i[1][0])
            h.SetBinError(i[0]+1, i[1][1])
            hs.SetBinError(i[0]+1, i[1][2])
    except yaml.YAMLError as exc:
        print(exc)
```



# Writing HEPData

```python
"""Utilities for converting to hepdata"""


from hepdata_lib import Submission
from hepdata_lib import Table
from hepdata_lib import RootFileReader
from ROOT import TFile, gPad
import os
# from __future__ import print_function
from hepdata_lib import Variable, Uncertainty

submission = Submission()  # Create new submission
```

+

More code magic