E4/E5 Programming Course 2025

Introduction to ROOT & PyRoot

Aaron van der Graaf (TU Dortmund)

20-03-2025

What is “ROOT” ?

- ROOT is a Data analysis package
  - Developed by CERN in the 90th → from particle physicists for particle physicists
  - Based mostly on C++, but also some Fortran while Python becoming more and more important
  - Very powerful, but rough to get started with
- Advantages of ROOT:
  - Extremely fast reading and writing of Data
  - Large number of features: fitting of complex functions, plotting, and much more …
- Disadvantages of ROOT:
  - Bad documentation → luckily LLMS such as ChapGPT are a good help here
  - Machine Learning is somewhat outdated and should not be directly performed in Root

Installing ROOT

More Details: https://root.cern/install/

- On workstations (no installation required):

```
setupATLAS
showVersions root
lsetup “root recommended”
```

- In einer Linux-distribution

Fedora: `dnf install root python3-root root-notebook`
CentOS: `yum install epel-release && yum install root`
ArchLinux: `pacman -Syu root`
Gentoo: `emerge sci-physics/root`

- Via Conda:

```
conda config --set channel_priority strict
conda create -c conda-forge --name <my-environment> root
conda activate <my-environment>
```

Work interactive with ROOT (TBrowser):

- Easiest way to get started:

```
    root
    new TBrowser
```

- As this is a GUI, X11 forwarding is needed if used remotely via SSH

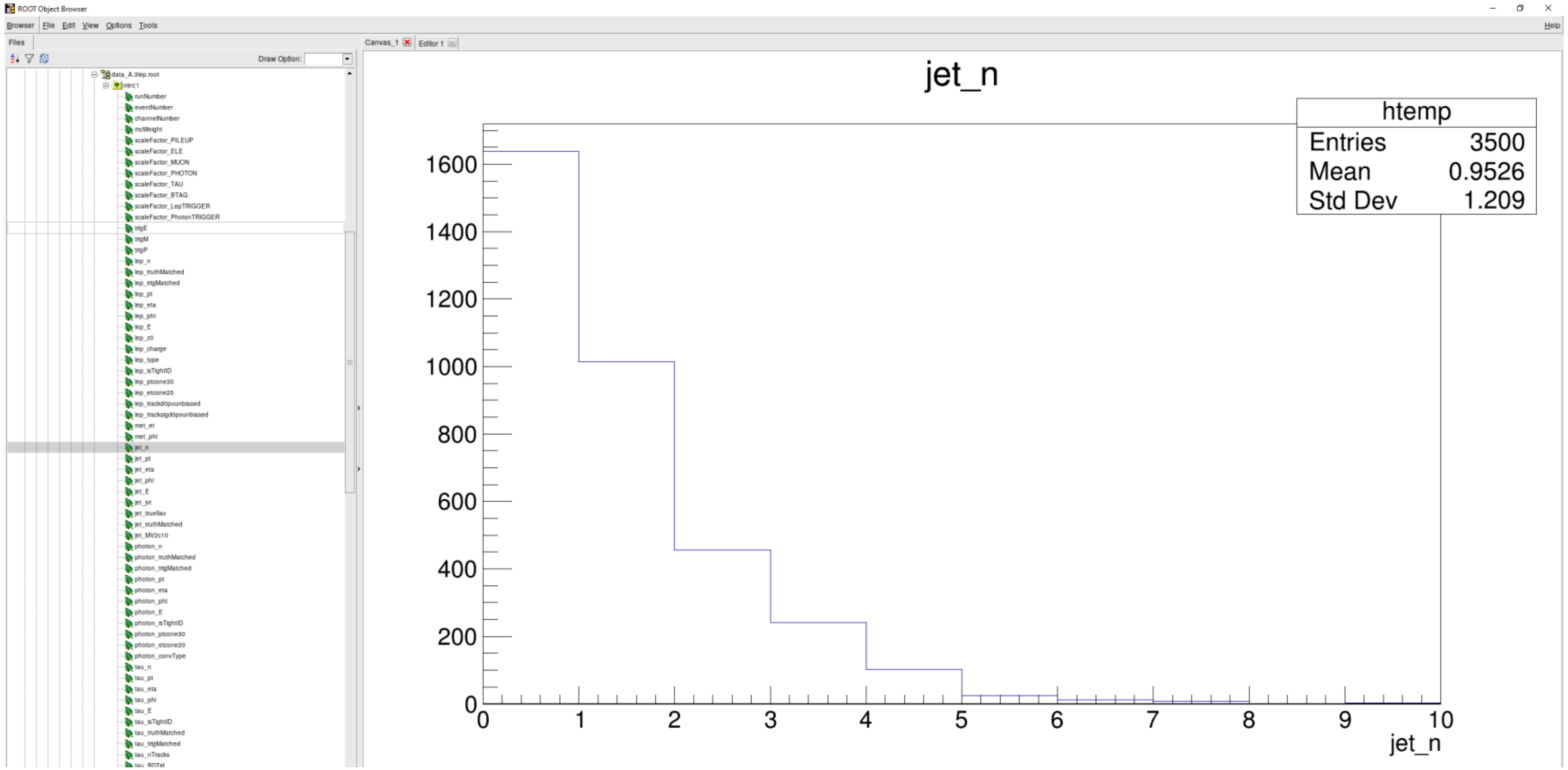
- Very helpful to have a quick check on a root file

- Alternatively: **VScode Ext. Root file viewer**

- However, the TBrowser has several **shortcomings**

Data for this tutorial: [ATLAS Open Data (3 leptons, 13 TeV)]()

`/ceph/e4/users/avdgraaf/public/ProgrammingCourse_Root/3lep`



Work interactive with ROOT (shell):

- Inspecting ROOT files via the ROOT C++ Interpreter without using the Tbrowser
- Open a ROOT File:

```
root data_A.3lep.root
```

- Or:

```
root
TFile* _file0 = TFile::Open("data_A.3lep.root")
```

Object class Variable name Class function call Root file name

- Inspect available trees via ls():

```
_file0->ls()
```

- Grab TTree of interest and insepect available branches via Print():

```
TTree *mini = (TTree*) _file0->Get("mini")
```

Object class Variable name

```
mini->Print()
```

Syntax is not easy and rather hard to remember, don’t be shy to look up these commands in the future

Work interactive with ROOT (shell):

All of these commands should also work inside C++/Root scripts

- Let’s plot a branch using Draw():

```
mini->Draw("lep_pt")
```

- Draw again, defining number of bins and the range:

```
mini->Draw("lep_pt>>h1(50,0,200000)")
```

N bins   Lower and upper range boundary

- Now we want to apply some selection by only plotting leptons with an $|\eta| < 0.5$:

```
mini->Draw("lep_eta")
mini->Draw("lep_eta", "lep_eta > -0.5 && lep_eta < 0.5")
```

Selection requirements, can be long and complex, allowing quick checks → **Very Powerful**

```
mini->Draw("lep_pt>>h1(50,0,200000)", "lep_eta > -0.5 && lep_eta < 0.5")
```

- Lets try something similar, check what happens to N_{jets} if we apply a higher p_T^{jet} cut of e.g. 50 GeV

```
mini->Draw("jet_n")
mini->Draw("jet_pt>>h1(50,0,200000)")
mini->Draw("jet_n", "jet_pt > 50000")
```

Read and write ROOT files via PyRoot

- **PyRoot** in combination with **RDataFrame** is a modern and powerful alternative → [Documentation]
- Will be illustrated in the following in a jupyter notebook
- Goal of using PyRoot:
    - Translate data quickly into an easier format
        - ➢ Numpy arrays
        - ➢ RDataFrames
    - Translate root file into better format for python, e.g. for Machine Learning in python
        - ➢ .CSV or .hdf5 are common formats
- The shown jupyter notebook can found here:

`/ceph/e4/users/avdgraaf/public/ProgrammingCourse_Root/3lep/Data/PyRoot_Tutorial.ipynb`

➢ Copy the notebook and the data_A.3lep.root into your home dir: `cp PyRoot_Tutorial.ipynb ~/.`

Practice Tasks

- Try using the shown concepts:
    - Open a TBrowser and inspect the tree of a root file and its branches
    - Try to use the VSCode extension Root file viewer to inspect some of the root files
    - Start an interactive Root session and try plotting some branches as histograms
        - Use the custom options for settings the histogram bins
        - Apply some selection to the events and replot a given branch
    - Run the presented Jupyter Notebook and try add some code:
        - Try plot some other distributions
            - Plot the absolute value of the missing transverse energy (*met_et*)
        - Apply some selection, select events with max. 2 jets
            - Plot the *jet_pt* and find the event with the highest *jet_pt* and print its *eventNumber*