

Bachelorprogrammierkurs 2024

ROOT, Uproot, PyRoot

11.04.24

AG Kröninger



ROOT

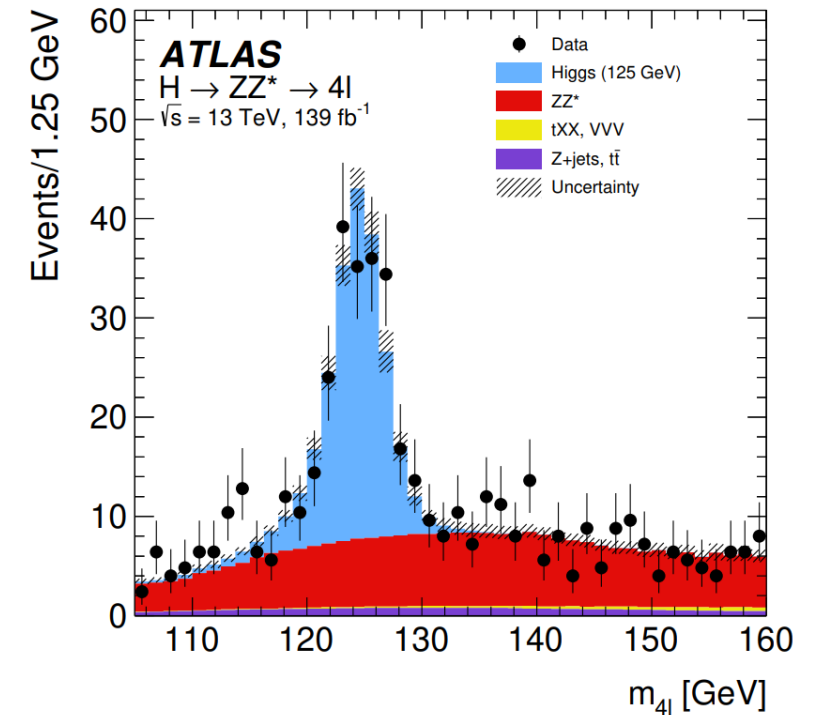
Freies Softwarepaket zur Analyse von Daten

Am CERN entwickelt

Nützlich für große Datenmengen

Standard in der Hochenergiepartikelphysik

→ Findet sich aber auch in der Wirtschaft



ROOT in HEP

ROOT besitzt nützliche Tools für HEP

Datenanalyse bei HEP-Experimenten sehr ROOT lastig

Output von Software häufig im ROOT-Format
→ MC Simulationen, Geant4

ROOT ist ... eher lästig :(

Implementiert in C++

Wenig Nutzen außerhalb von HEP-Experimenten

Umgang mit Root und C++
→ Heute um 13 Uhr

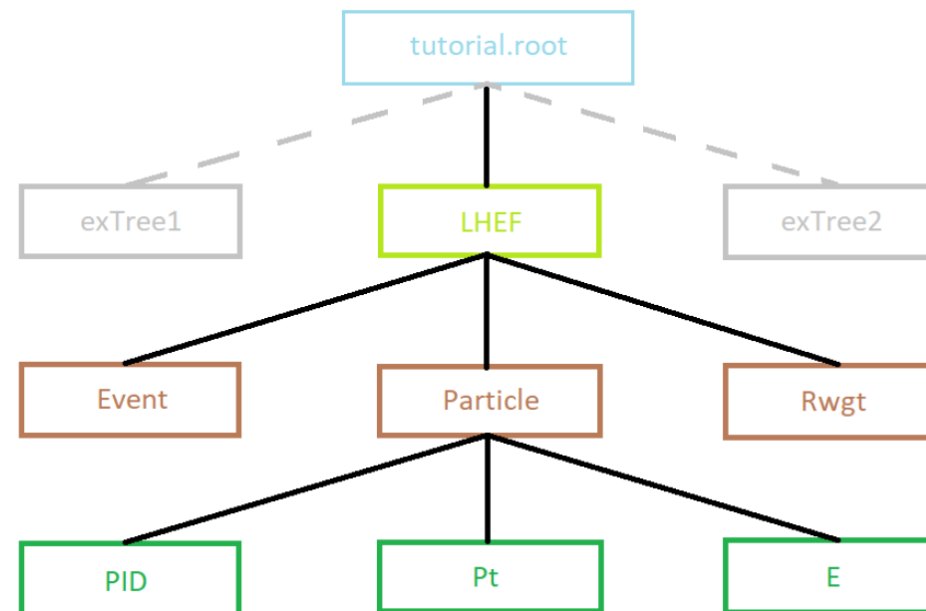
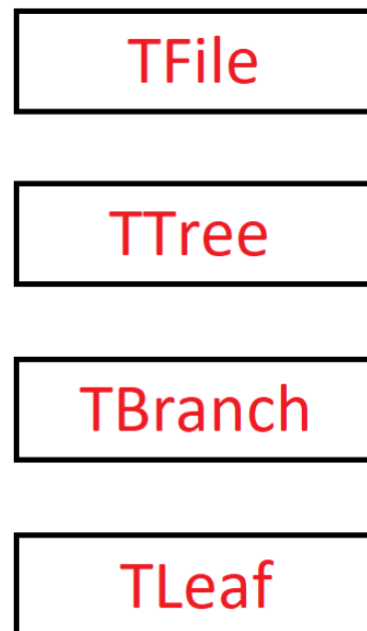


Medizinphysiker & Detektor: Geant4-Output ist im ROOT-Format
Ziel für Heute: Wie gehe Ich mit Dateien im ROOT-Format um?

.root-Files

ROOT hat eigenes Dateiformat (.root) → Nicht leserlich wie .csv Dateien

Strukturierte Informationen über Daten → Wird von ROOT als TFile geladen



.root Files ohne C++

1. TBrowser

→ Nützlich für schnelle Checks

2. Uproot / PyRoot

→ Skripte (aber langsamer als C++)

→ PyRoot: sehr Root ähnlich, nur halt in Python

→ Uproot: sehr Root unähnlich, typische einfache Python Struktur

TBrowser

ROOT Browser zum visualisieren einer .root-Datei

ROOT-Session: just-in-time compilation, C++
→ analog zu ipython

Es gibt eine VSCode extension (nie mit gearbeitet)

TBrowser kann nur innerhalb ROOT-Session geladen werden

```
new TBrowser;
```

TBrowser - .root Datei kann recht viel beinhalten

dataWeightPtBinNoMu	y_truth_pdgId	y_topoetcone40_etaEDParPU_MC_corr	y_E1E2_raw	y_noFF_Rhad1
dataPrescale	y_truth_mother_pdgId	y_topoetcone20_oldPU_corr	y_E0	y_noFF_Eratio
dataWeightThreshold	y2_pt	y_topoetcone40_oldPU_corr	y_E0_raw	y_noFF_e277
dataWeightThresholdNoMu	y2_eta	y_topoetcone20_SCsub_etaEDParPU	y_E1	y_noFF_deltae
acceptEventThreshold	y2_phi	y_topoetcone40_SCsub_etaEDParPU	y_E1_raw	y_maxEcell_E
acceptEventPtBin	y2_e	y_topoetcone20_IsoCorrTool	y_E2	y_maxEcell_gain
evt_lb	y2_convType	y_topoetcone40_IsoCorrTool	y_E2_raw	y_maxEcell_time
evt_runNo	y2_convRadius	y_ptLeak20	y_E3	y_maxEcell_x
evt_eventNo	y_ptcone20	y_ptLeak40	y_E3_raw	y_maxEcell_y
evt_mu	y_ptcone40	y_ptconecore	y_Reta	y_maxEcell_z
evt_npv	y_topoetcone20	evt_centralDensity	y_Rphi	i_n
y_n	y_topoetcone30	evt_fwdDensity	y_weta2	j0_pt
y_passOQ	y_topoetcone40	evt_neflowiso_centralDensity	y_fracs1	j0_eta
y_pt	y_topoetcone20_DDcorr	evt_neflowiso_fwdDensity	y_weta1	j0_phi
y_eta	y_topoetcone40_DDcorr	y_iso_FixedCutLoose	y_emaxs1	j0_e
y_phi	y_ptcone20_TightTTVA_pt500	y_iso_FixedCutTight	y_f1	y_j0_dphi
y_phi_cl	y_ptcone20_TightTTVA_pt1000	y_iso_FixedCutTightCaloOnly	y_wtots1	y_j0_dr
y_e	y_ptvarcone30_TightTTVA_pt500	y_tptcone20	y_Rhad	j1_pt
y_pt_cl	y_ptvarcone30_TightTTVA_pt1000	y_tptcone30	y_Rhad1	j1_eta
y_eta_cl	y_neflowisol20	y_tptcone40	y_Eratio	j1_phi
y_e_cl	y_neflowisol30	y_tetcone20	y_e277	j1_e
y_eta_cl_s2	y_neflowisol40	y_tetcone30	y_deltae	y_j1_dphi
y_eta_cl_s1	y_topoetconecoreConeEnergyCorrection	y_tetcone40	y_noFF_IsEMTight	y_j1_dr
y_convType	y_topoetconecoreConeSCEnergyCorrection	evt_SherpaClass	y_noFF_IsTight	jmin_pt
y_convRadius	y_core57cellsEnergyCorrection	ty_dmin	y_noFF_IsLoose	jmin_eta
y_isTruthMatchedPhoton	y_topoetcone20ptCorrection	y_IsEMTight	y_noFF_Reta	jmin_phi
y_truth_convType	y_topoetcone40ptCorrection	y_IsTight	y_noFF_Rphi	jmin_e
y_truth_convRadius	y_topoetcone20ptLogCorrection	y_IsEMLoose	y_noFF_weta2	y_jmin_dr
y_truth_type	y_topoetcone40ptLogCorrection	y_IsLoose	y_noFF_fracs1	y_jmin_dphi
y_truth_origin	y_neflowisolcoreConeEnergyCorrection	y_IsLoosePrime2	y_noFF_weta1	y_OR_j_dr
y_truth_pt	y_SCsubtraction	y_IsLoosePrime3	y_noFF_emaxs1	y_OR_j_dpt
y_truth_eta	y_topoetcone20_etaEDParPU_corr	y_IsLoosePrime4	y_noFF_f1	y_caloPointing_z
y_truth_phi	y_topoetcone40_etaEDParPU_corr	y_IsLoosePrime5	y_noFF_wtots1	y_caloPointing_z_err
y_truth_e	y_topoetcone20_etaEDParPU_MC_corr	y_E1E2	y_noFF_Rhad	y_convPointing_z

ROOT Session mit .root Datei verknüpfen

```
root -l tutorial.root
```

Überblick:

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

Tree erzeugen:

```
TTree * tree = (TTree*)_file0->Get("alpha");
```

Überlick über Inhalt des Trees:

```
tree->Print("all");
```

Histogramm zeichnen:

```
Tree->Draw("ekin")
```

Cuts hinzufügen:

```
Tree->Draw("ekin", "ekin > 1.0 && ekin < 1.5")
```

Hands-On

Ort: `cd /ceph/groups/e4/users/mwindau/public/BAProgkurs`

swbase?

Ceph?

JupyterLab?

Uproot

Ist nicht abhängig von C++ ROOT



Verwendet numpy und ist sehr python-esque

Hands-On-Tutorial: [BAProgKurs/uproot_tutorial.ipynb](#)