GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

BSc Intro 2022 ROOT Intro

Wael Alkakhi, Ishan Pokharel, Chris Scheulen, Sreelakshmi Sindhi

II. Physikalisches Institut, Georg-August-Universität Göttingen

2022-03-09

Today's Agenda

- **Intro**
  - Setting up
  - Interactive ROOT sessions
  - ROOT via Terminal
- **Tutorial**
  - Opening TFiles
  - Creating TChains
  - Accessing Variables from TBranches
  - Setting up Event Loops
  - Filling Histograms (TH1/TH2)
  - Writing Histograms to TFile
  - General Notes

First Things First: Setting Up

1. SSH into lxplus:
   `$ ssh -XY <MY_USERNAME>@lxplus.cern.ch`
   - Flags for allowing graphical stuff (X11-Forwarding)
   - Hope you didn't forget your password here...
2. setup ATLAS environment: `setupATLAS`
3. Load up ROOT:
   `$ lsetup "root <version>"`
   - For looking up available versions: Use `$ showVersions root`
   - Here: We will use version **6.20.06-x86_64-centos7-gcc8-opt**
   - Later: Ask your ~~doctor~~ supervisor, which version is right for you

Congrats! Now you are ready to ROOT!

Interactive Mode

- Simplest way to run ROOT: Interactive mode
  - Similar to python console

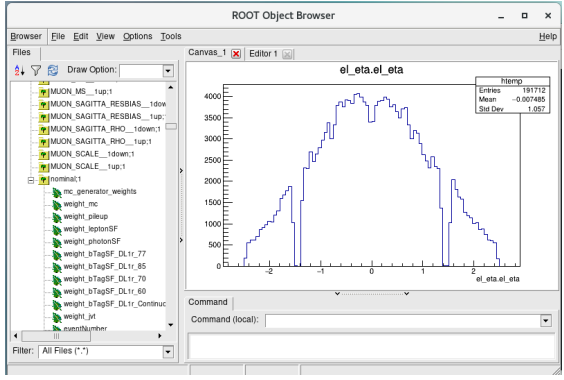
→ Just type `root` for this
  - Not ideal for running a full analysis (You'd have to remember every command in the right order...)
  - Great for learning/debugging though
  - If you want to save many histograms: Batch-mode is your friend (Flag `-b` for this)
  - Annoyed by all the info? Flag `-l` to start quietly.
- Now you can try out ROOT (and C++) commands
- Stuck in interactive mode with no way out?! `.q` to quit or `.help` to get help!

Stuff to Try Out

- Define variables and use them (in C++ syntax)
  - `double pi = 3.14`
  - `int r = 10`
  - `std::string welcome = "Welcome to ROOT!"` (← Note how some C++ stuff works as well!)
  - `2 * pi * r`
- Write a function (Multi-line expressions in brackets work!)
  - `std::cout << "Hello, World!" << std::endl`
  - ```
    [0] int doubling(int a) {
    [1] return 2 * a;
    [2] }
    ```
  - You need to include semicolons in multi-line functions (like in normal C++)!
    ```

TBrowser

- Nice way of browsing through ROOT files
- Can show histograms, TTrees, etc.
- Also possible to modify attributes (e.g. Histogram display options)
- A note on TTrees, TBranches and TLeaves:
  - Event data organized in TTrees
  - TTrees hold TBranches
  - Branches have TLeaves with the event variables (one variable per leaf, one or more leafs per branch)



Looking at TFiles without TBrowser

- You can load TFiles when starting ROOT:
  `$ root <filepath>`
  → TFile gets handle (normally: `_file0`)
  – Alternative while in ROOT (used later today):
    `[0] TFile* file = new TFile("<filepath>", "READ")`
    `[1] TTree* tree = (TTree*)file->Get("<treename>")`
- Now you can look at the tree contents:
  – `[0] nominal->Print()` prints tree contents of nominal
  – `[1] nominal->Show(10)` prints out all variables of the 10^{th} entries in nominal
  – `[2] nominal->Scan("jet_pt:jet_eta")` prints out values for jet_pt and jet_eta of entries
  – `[3] nominal->Scan("jet_pt:jet_eta", "", "colsize=XX")` prints out values for jet_pt and jet_eta of entries and specifies range of characters to print
  – `[4] nominal->MakeClass("Myclass")` generates a C++ class to reproduce the nominal TTree
    (helpful for figuring out variable types)

Next up: Tutorial

Creating an EventLoop with selection cuts and plotting/fitting the resulting histograms!

Tipps

1. Memory Management in ROOT
   - Has its own memory management system that is different than a specific programming language's garbage collector.
   - If a file is open (TFile), then all objects (TObjects) are 'owned' by this file. Important to free up the object to the global register to be used later (`SetDirectory(0)`)
   - Always open a file, access the object and IMMEDIATELY close the file.
2. `TBrowser` over ssh connection is very slow to respond
   - Familiarise and use `tree->Print(), tree->Scan()` for quick review.
   - Get Event related information from `tree->MakeClass("myClass")`.
   - If coding in C++, make sure the exact same object type is being used.
3. Environment setup
   - Include `setupATLAS` and other ATLAS-specific setups (`lsetup`) in a bash script. Create an alias to set it up in your `\.bashrc`
   - Bash `alias` is your friend!