Root Tutorial

Dr. Rakitha Beminiwattha
Louisiana Tech University

Prerequisites

- We will use the virtual machine (VM) or ifarm interactive terminal
- Files required for the tutorial are available at Hands-On-Root in https://drive.google.com/drive/folders/1yFj_sr8VfnpUWPJvbgNnxoDJLfnbLNji?usp=sharing
- Download the root file HandsOn_remollout_10k.root into ~/softwares/remoll directory (in the VM)
- Download the file basicRootScript.C into ~/softwares/remoll/analysis directory

Open A Root File

1. Open a terminal using Alt+Crtl+T
2. Goto ~/softwares/remoll
3. Now we will open the root file using

```
./build/reroot HandsOn_remollout_10k.root
```

reroot is root compiled with remoll libraries

4. Now we will learn basic operations of root to access data stored in the root file

Histograms

- The most straightforward way to access data is to view histograms of data stored in the root file
- Method 1: type

  ```
  TBrowser b
  ```

  This will open the file browser

- Then we can click on any tree element to view as a histogram
- Method 2: using Root command line

View Tree Elements using Root Command Line

- Plot hit.p (momentum) of electrons hitting detector 28

```
T->Draw("hit.p","hit.pid==11 && hit.det==28")

T->Draw("hit.p/GeV","hit.pid==11 && hit.det==28")
```

- Plot hit.p (momentum) of electrons hitting detector 28 weighted by the rate so higher rate events gets higher weights and lower rate events gets a lower weight,

```
T->Draw("hit.p","(hit.pid==11 && hit.det==28)*rate",”hist”)
```

- Plot hit.r (radius of hit) of electrons hitting detector 28 weighted by the rate so higher rate events gets higer weights and lower rate events gets a lower weight,

```
T->Draw("hit.r","(hit.pid==11 && hit.det==28)*rate",”hist”)
```

- Plot 2D histogram to see correlation between momentum and radius at the detector 28

```
T->Draw("hit.r:hit.p","(hit.pid==11 && hit.det==28)*rate",””)

T->Draw("hit.r:hit.p","(hit.pid==11 && hit.det==28)*rate",””)
```

- How do we automate these steps? Use a root script

Root Scripting: Histogram Declarations

`basicRootScript.C`is our template script, we will create histograms of hit radius, xy 2D distribution and source vertex of these hits

1. Declare 1D histograms for radius and source vertex

   ```
   TH1D *r

   TH1D *sourceZ

   TH1D *rRate //for rate weighted radial distribution
   ```

2. Declare `2D histograms for XY distribution`

   ```
   TH2D *hXY

   TH2D *hXYrate //for rate weighted XY distribution
   ```

Root Scripting: Histogram definitions

- Let’s define their parameters and create them initHisto() routine

```
 r = new TH1D("r","radial distribution;r[mm]",200,500,1500);

 sourceZ = new TH1D("sourceZ","initial vertex for hit ;z position
[mm]",5000,-5500,-3500);

 hXY = new TH2D("hXY","2D hit distribution;x [mm];y
[mm]",200,-2100,2100,200,-2100,2100);
```

- Define rate weighted histograms

```
 rRate = new TH1D("rRate","rate weighted distribution;r[mm]",200,500,1500);

 hXYrate = new TH2D("hXYrate","rate weighted 2D hit distribution;x [mm];y
[mm]",200,-2100,2100,200,-2100,2100);
```

Root Scripting: Filling Histograms

- Let’s fill these histograms with data from the Tree in at the `processOne(...)` routine

```
r->Fill(hit->at(j).r);

sourceZ->Fill(hit->at(j).vz);

hXY->Fill(hit->at(j).x,hit->at(j).y);

rRate->Fill(hit->at(j).r,rate);

hXYrate->Fill(hit->at(j).x,hit->at(j).y,rate);
```

Root Scripting: Post Processing

- Scale rate weighted histograms if we have used chain of root files (more than one root filed linked) in the `void scale()` routine

```
rRate->Scale(1./nFiles);

hXYrate->Scale(1./nFiles);
```

Make Histograms Canvases

- Let’s create few canvases

```
TCanvas *c1 = TCanvas();

c1->Divide(1,2)

c1->cd(1)

r->DrawCopy()

c1->cd(2)

rRate->DrawCopy()
```

Save Output into a Root File for Later access

- Output written in this step can be accessed later in a root file “`basicRootScript.root`”

```
r->Write();

sourceZ->Write();

hXY->Write();

rRate->Write();

hXYrate->Write();
```

Save Output into a Root File for Later access

- Output written in this step can be accessed later in a root file "`basicRootScript.root`"
- This file name is set in the routine `void initHisto()`

```
string foutNm = Form("basicRootScript.root");
```

- You can access the saved histograms using the command

```
root basicRootScript.root or

./build/reroot basicRootScript.root
```

How to Execute the Script

1. Load the script `basicRootScript.C`

```
.L analysis/basicRootScript.C
```

2. Execute the script

```
basicRootScript(“HandsOn_remollout_10k.root”)
```