

Introduction to Analysis Example Tutorial

P. Skubic (OU)

Email: pskubic@ou.edu

Information about the University of Oklahoma (OU)

- *The Princeton Review* consistently ranks OU among the best in the nation in terms of academic excellence and cost for students.
- The Department of Physics and Astronomy has four research groups:
 - Particle physics
 - Atomic physics [Center for Quantum Research and Technology (CQRT)]
 - Condensed matter physics [CQRT]
 - Astronomy and Astrophysics
- We would welcome applications from ASP2024 participants (see: <http://www.nhn.ou.edu>)

Typical data analysis tasks in particle physics

- Task 1: create files containing simulated data
- Task 2: analyze simulated data
- Task 3: collect real data from detector
- Task 4: analyze real data
- Task 5: compare simulation with real data
 - If there is good agreement, limits can be set on existence of new physical states (i.e. particles)
 - If there is disagreement, further study is needed
 - Possible mistake?
 - Possible new discovery?
- We will illustrate tasks 1, 2, (tutorial steps 1 and 3) and task 4 (tutorial step 2) today

Notes

- In particle physics software tools such Madgraph or Isajet (event generators) and GEANT (to simulate our detector response) are used in Step 1
 - We will use a simple random generator of a Gaussian distribution in Root
- Typically (almost) the same reconstruction software is used for Step 4 and Step 2
- Root is a powerful tool to read and analyze large amounts of data

Root Documentation

- Web page: <http://root.cern.ch>
- It is useful to click on: **Documentation** and then select: **Reference Manual**
- From there you can look at the documentation and source code for all the Root classes in any version of Root

Step 1: Create simulated data by running Root

- Contents of execution script: **run-root.sh**

```
#!/bin/bash
```

```
root -b < run-root.C > root.out
```

This command executes Root in batch mode using macro **run-root.C** and routes output to file **root.out**

Step 1: Create simulated data by running Root with macro **run-root.C**

- Create TFile 0 for “run 0” (t00.root)
- Create TTree object (“t0”) to store data in Root
 - Generate 100 “events” each with Gaussian distributed “Energy”
 - Fill TTree branches for each event
- Write TFile 0
- Close TFile 0
- Repeat above steps to create TFile 1 for “run 1” (t01.root)

Step 2: Analyze real data on the grid and with Root

- First we will run a Root macro to read di-muon events and fill a TTree with associated variables such as energy and transverse momentum
- The macro also determines the invariant mass of the muon pairs.
- Finally, we will examine the invariant mass with a Root TBrowser to determine the Z-peak mass

readEvents.C

- Macro to calculate the invariant mass of the first muon pair in each event and then plot the invariant mass in a histogram.
- Only looks at events which contain at least two muons where both muons have transverse momentum, $p_T > 20$ GeV.
- The two selected muons must have opposite charge.

Z-boson Plot

- A Z-boson is a particle that only lives for a very short time before decaying. We can observe a Z-boson by looking at its decay products. The decay modes of the Z are here

<http://pdg.lbl.gov/2014/tables/rpp2014-sum-gauge-higgs-bosons.pdf>

- It decays to two muons or two electrons 3.4% of the time.

More Information

- ROOT website: <http://root.cern.ch/>
- Tutorials:
<http://root.cern.ch/root/html/tutorials/>
- ATLAS Z cross-section legacy paper:
<http://arxiv.org/pdf/1010.2130v1.pdf>

Step 3: make TSelector

While running Root:

```
TFile f("t00.root"); //open file  
t0->MakeSelector("s0", "=legacy"); //create TSelector  
f.Close(); //close file  
.q
```

This creates two files with code: `s0.C` and `s0.h`

We will modify these files to add a histogram of the Energy variable and use them to process the simulated data on the Grid

Conclusion

- After completing Steps 1 - 3 you are in principle ready to scale up and make TTree's with hundreds of variables and create and analyze thousands of files
- If time permits you can try adding your own features to the existing example by adding variables and histograms, etc.
- Good luck and have fun!!