ROOT Tutorial

2020/07/31

Brais López Paredes

About this tutorial

What you should expect

- I will try to *tutor* to the average poll respondent by default (see results)
- But you can ask any questions at any time! I'll try to answer
- I will be solving several "problems" using ROOT
- I will introduce a "problem" and then I'll try to modify the code to solve it
  - Wish me luck, I'm self taught. All the code is at https://gitlab.com/braisLP/root_tutorial
- I expect you to know the (very?) basics of ROOT and C++ already
  - TObjects, read trees, histograms, the existence of makefiles...
- Feel free to criticise my ROOT and C++ usage, always learning
- Please think about how to solve these with your favourite analysis language (R, Matlab, Java, and others) and prepare a tutorial!

A bit of history...

" ROOT is not your enemy "

Quentin Riffard[1]

" (...) this could be an acronym for René Object-Oriented Technology "

René Brun[2]

[1] LZ collaboration meeting, Coimbra, April 2018

[2] Link

Problem 1 (Unmodified code in “Stage 0” folder, solution in “Stage 1” folder)

- The LUV collaboration relies on CEST (Common Element Simulation Technique) to model the response of their detector target material to dark matter recoils. For an input recoil energy, CEST outputs a number of ionisation electrons based on the world average charge yield Qy(E).
- CEST comes in the form of a C++ .h file and an example main.cxx
- ./testCEST outputs the recoil energy and number of electrons to the screen, but we would hate to work with text files (!e.g. ./testCEST > out.txt)
- Suggested solution: create a TTree and fill it with CEST outputs, then write it to a ROOT file. The Makefile has to be modified!
    - A histogram will also allow us to have a quick look at the output

Problem 2 (“Stage 1” folder)

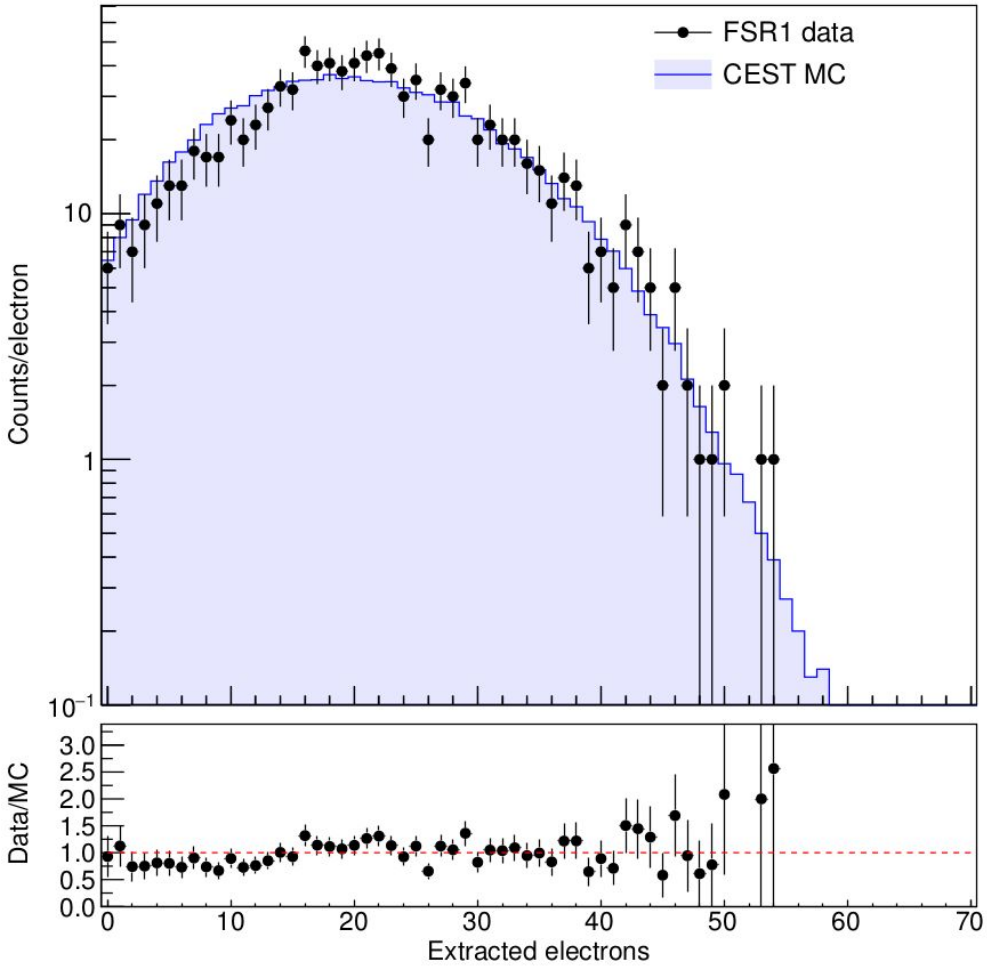
- The CEST collaboration created a default mode in which energy is sampled from a flat spectrum, with a range specified by the user
- LUV collaborators agree that most things in nature do not arise from a flat energy spectrum and want you to do something about it
- We could use built-in C++ to create random distributions to sample from, but since we have ROOT there’s a simpler way, although perhaps slower!
- Suggested solution: create a TF1 that outputs the desired spectrum. Don’t stop at default functions!
    - TF1s can of course be saved in ROOT files, but those based on custom C++ functions get frozen

Problem 3 (“Stage 2” folder)

- The LUV detector has performed especially well and all the group conveners think they have determined the detector/CEST parameters needed to reproduce the data. We have generated some MC with these parameters to compare to the data (output_MC.root, output_data.root)
- We want to see if the histogram of extracted electrons in data and MC agree, so we would like to overlay them on the same canvas
- Since calculating agreement by eye is difficult, we would also like to plot the ratio Data/MC. A legend would be nice too
- Proposed solution: loop over the trees to create the histograms, divide a canvas, plot the overlay on the top and the ratio on the bottom with shared x-axis. Plot the data with uncertainties. Save as a .pdf file

The plot

- You will be asked to move the Y-axis titles a bit closer at the very least
- You cannot drag them with your mouse
- Use histo->GetYaxis()->SetTitleOffset(number);
- Log scale is set on the pad we're drawing on: pad->SetLogy(true);

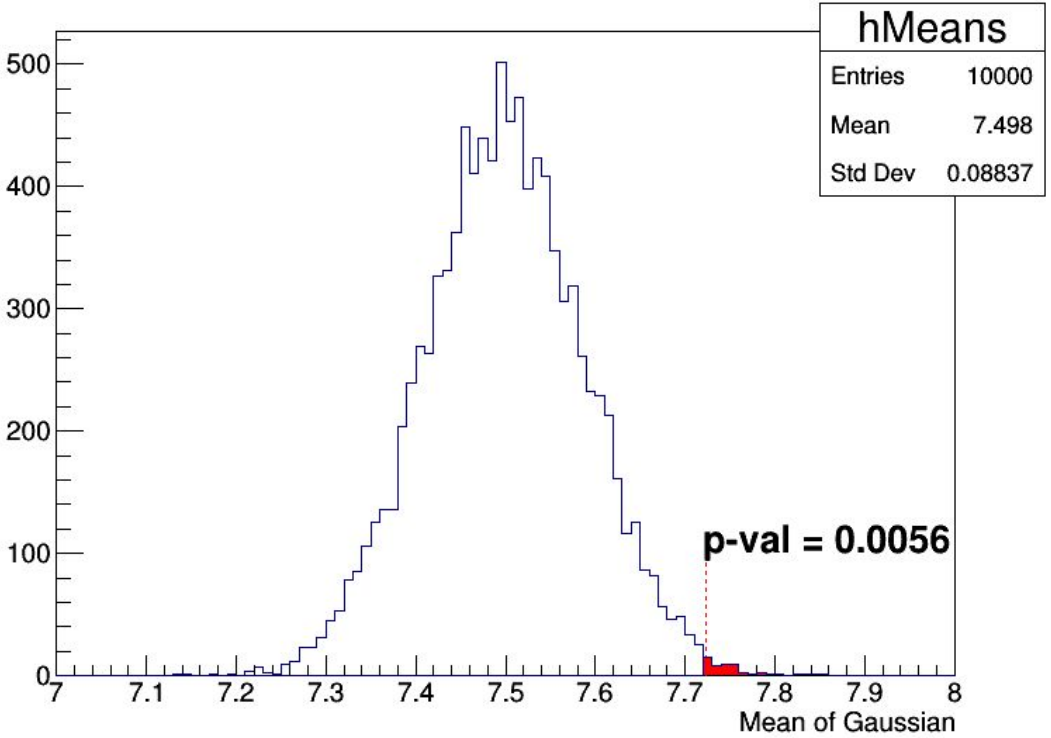


Problem 4 (“Stage 3” folder)

- In an overrunning analysis meeting the collaboration manages to agree that the simulation is not representing the data well
- The simulations convener argues that we should re-run the simulation with a revised spectrum, but a different collaborator whose voice you can’t quite place suggests that we should re-scale the MC energy distribution to save time and CPU.
- Suggested solution: follow the latter suggestion and loop over the data and MC trees and make histograms of the energies of data and MC. Then loop again and make a new MC histogram of extracted electrons, but weighed by the data/MC energy histograms
    - The solution code does other things such as “calculate the probability that the data could be sampled from the MC” and plots it nicely. It uses a gaussian fit for this (and for weight)

The plot

- Definitely not publication quality, but OK for internal consumption
- I didn’t write an example plot comparing the reweighted MC to the data histogram: we can have a look at this on the TBrowser

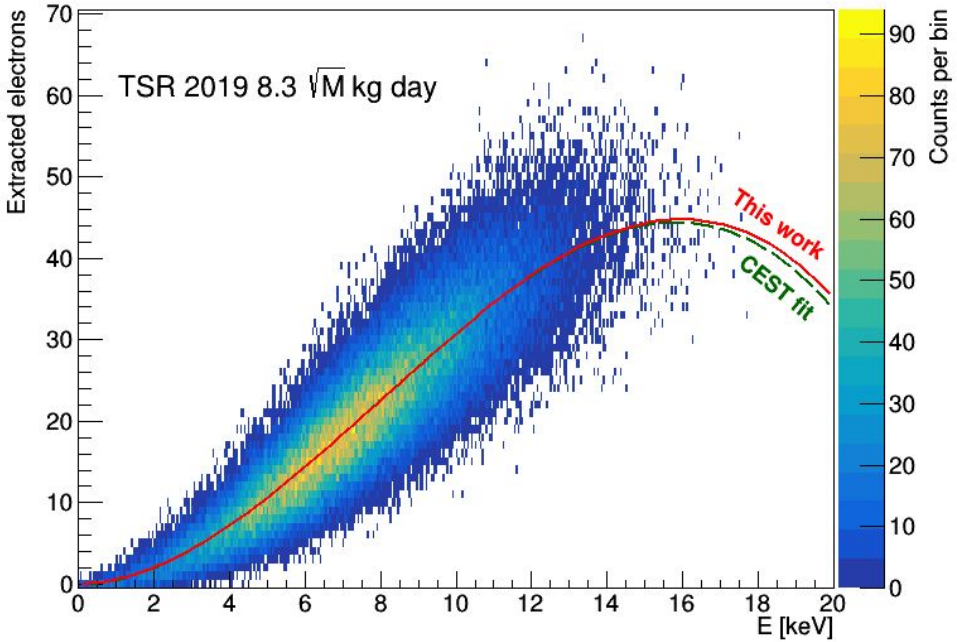


Problem 4 ("Stage 3" folder)

- The CEST collaboration wants to make sure that LUV data is in agreement with world calibration data, so they want to see a plot of recoil energy and number of extracted electrons
- Before making this plot public, the physics coordinator wants to see if re-fitting the CEST model with our data makes a significant difference to the model
- Suggested solution: loop over the data to create a TH2D, then save it to a ROOT file. Create a TF1 to fit the TH2D with the CEST model. Then plot the TH2D with a reasonable colour scheme and overlay the new fit and the original CEST model

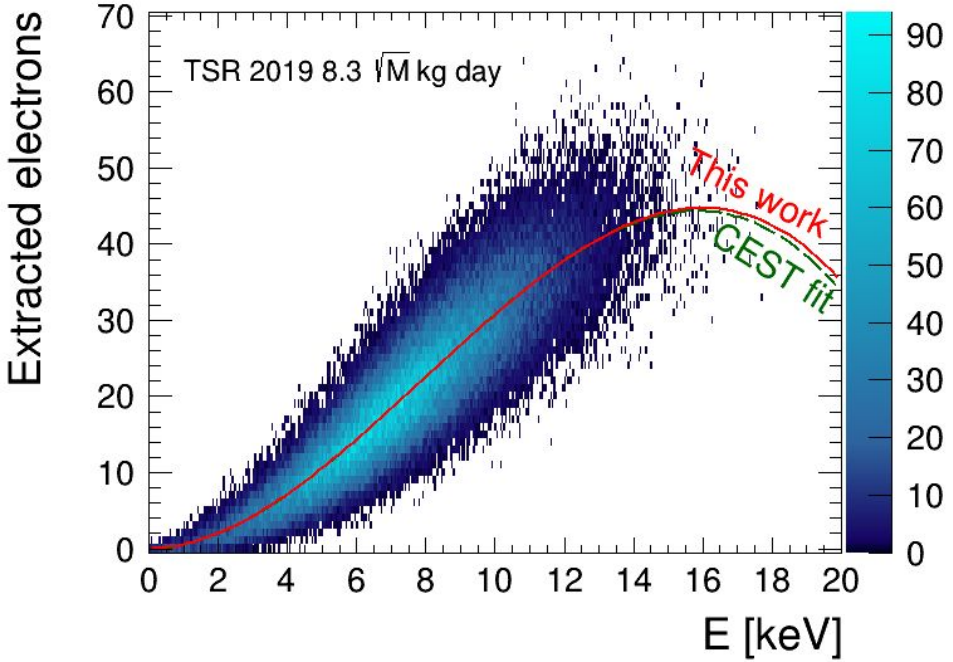
The plot

- This colour scheme is the default
- Some people complain that it’s not intuitive
- Other palettes are possible (e.g. grey or blue scale)



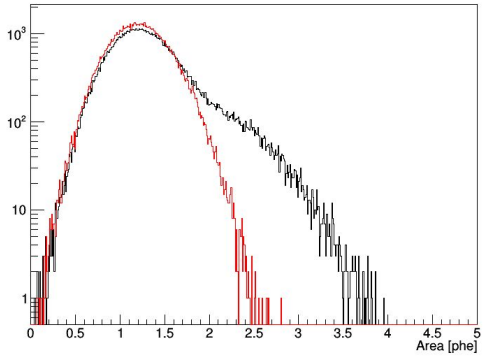
The plot

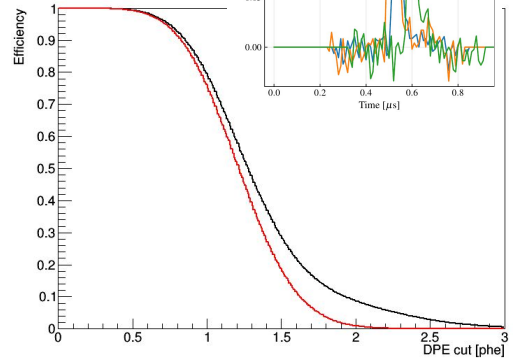
- This is the MDC2 colour scheme SetLZStyle.h, automatically applied on Stage_4/rootlogon.C (this file is executed by default)
- I actually wrote the solution without this, so some more changes would be needed to make it look good

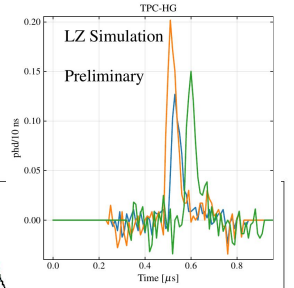


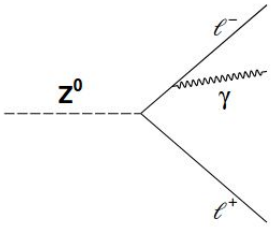
Other examples

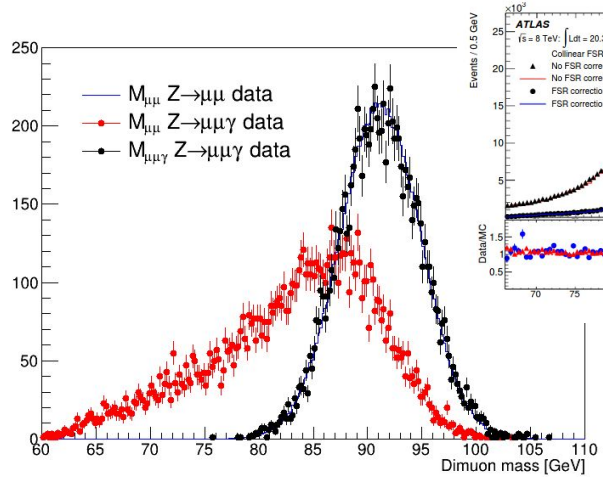
- Back to LZ:
  - DPEareas.C: what's the probability that two dark count coincidences fake an S1 if we take the DPE effect into account?
- Welcome to ATLAS:
  - EMcalibration.C: the Higgs hasn't been discovered, and we don't know if our photon calibration is working. Use Z boson decays to muons (easy to calibrate) and events with FSR photons to find the EM energy scale (use a minimiser)

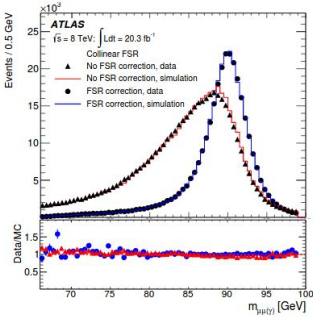












The end

- Thank you for connecting!
- Again, feel free to ask questions or suggest improvements