ROOT::Reading and Cutting a Tree

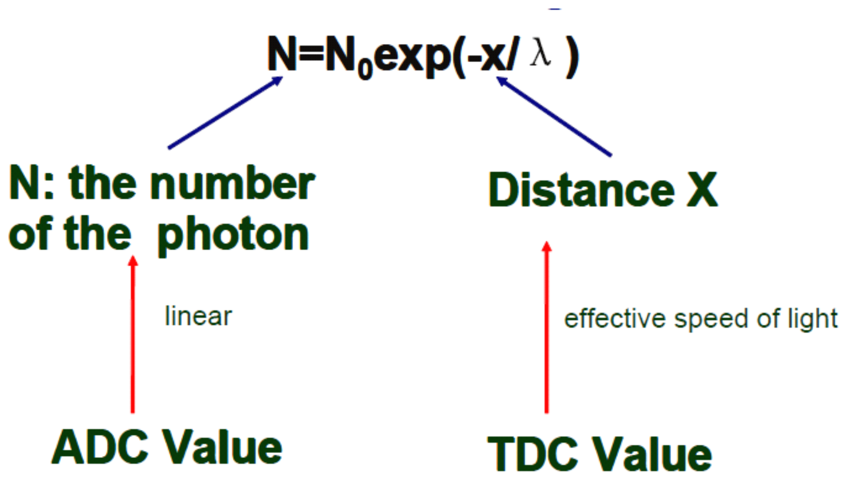
For this section we will learn how to plot 2-Dimension histograms and make some cuts on them. Furthermore, we will use the resonable function to fit the ADC histograms after making cuts. If you have time, it is better to learn how to use TGraph to plot the cutting points and fit them by some functions .

Firstly, download file"511.root" from the website. You will use the data stored in this file to do analysis.

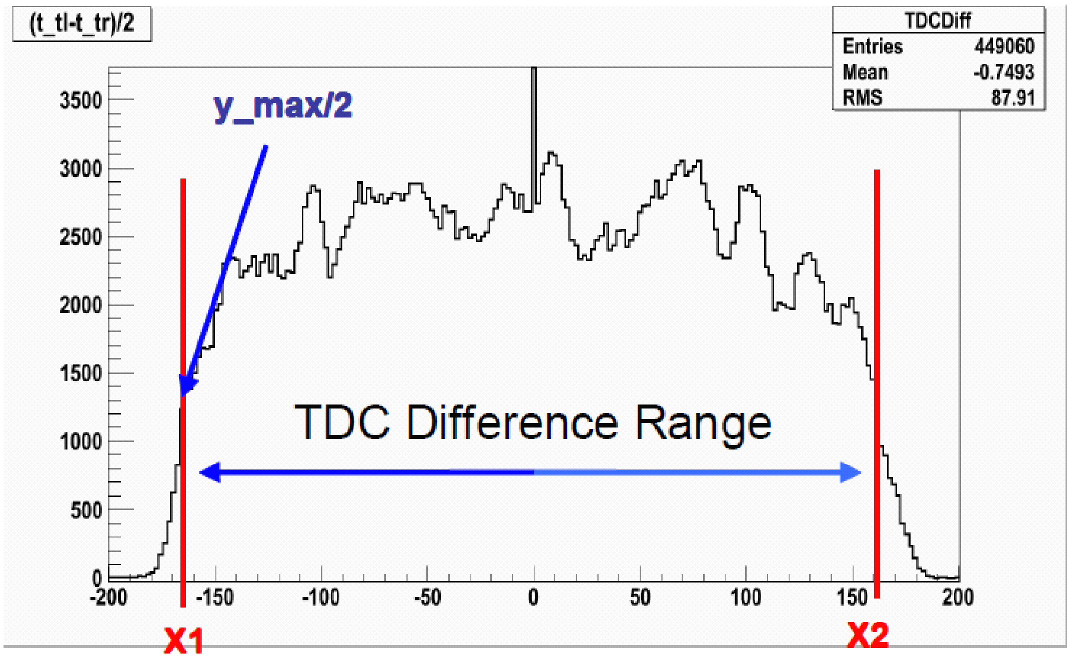
1) How to read a Tree? (Hint, you can write a named script) please read the Tree section on the website http://root.cern.ch/drupal/content/users-guide

If you do not get it after reading, you can see the reference code at page 10.

2) What is the attenuation length? What kind information (from electronic equipment) do you need to get the attenuation length of the scintillator?



Here, we can calculate the effective speed of light with TDC difference value.

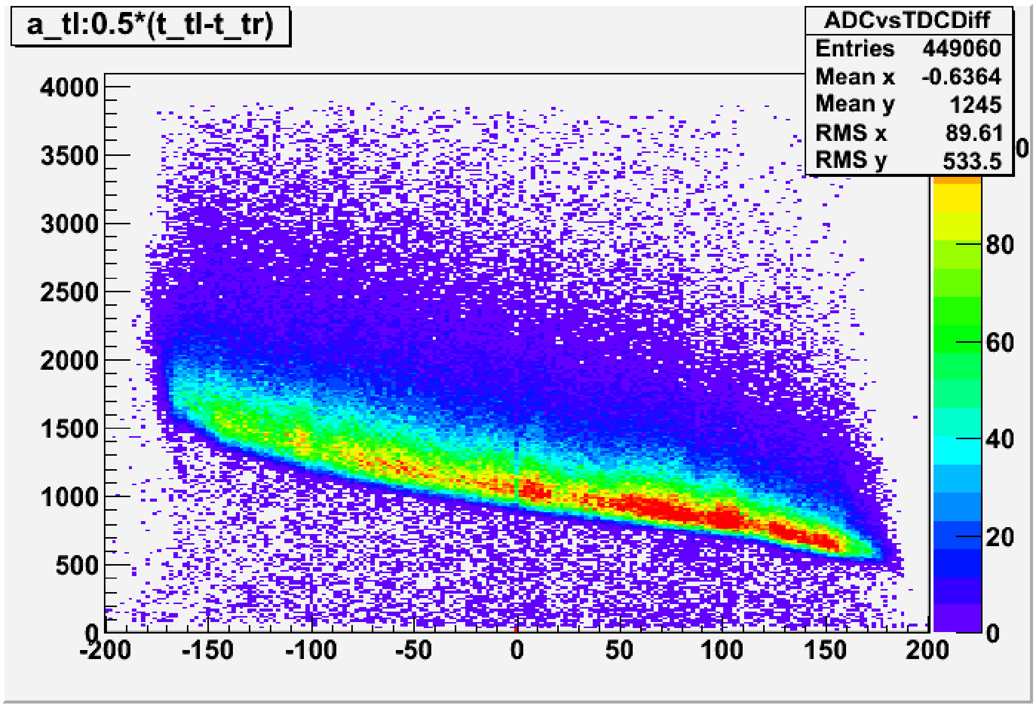


Picture1

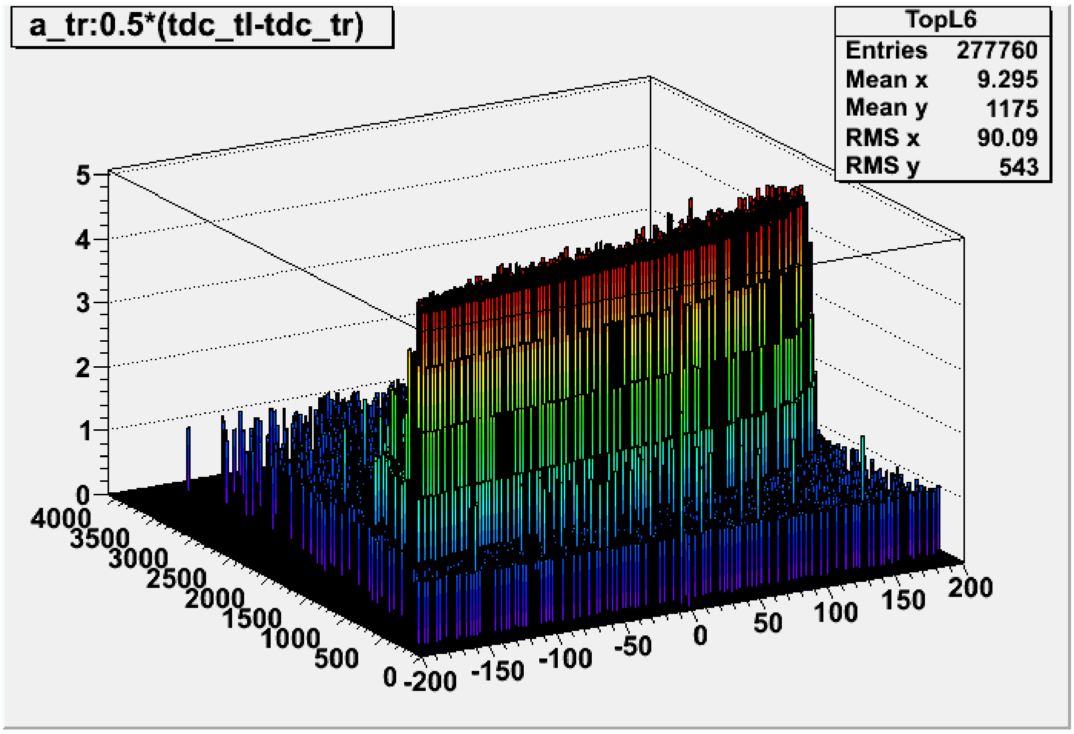
We can measure the length of scintillator with unit "cm". Furthermore, we know that the TDC Difference divided by two is the time of photons going through the whole sintillator. With this information, we can calculate the effect speed of light by the formula $V_{eff}=S/t$. From picture1, you can see that the unit of X axis is bin. For this experiment, the electronic set up makes one bin is equal to 25ps. If we know the TDC difference range with bin unit, then we will know time(t). The distance between two red lines is the TDC difference.

Now we need to identify the red lines' positions. Firstly, we need to find the maximum number of this distribution(y_max), and then find two x values corresponding to this y_max/2. Secondly, calculate the difference between X_1 and X_2, and this difference is the TDC difference range, shows on picture1. You can compare what number you get form this histogram with that you did before by using source method. Also, we know the relationship between "cm" and "bin".

Then, we need to plot the histogram of ADC versus TDC difference, which shows on picture2(a). From picture2 (b), Does the meaning of Z axis and color make sense for you?



Picture2 (a)



Picture 2(b)

t1->Draw("a_tl:0.5*(t_tl-t_tr)","sub-range","COLZ"); // The OPTIONAL“sub-range” : with it, you can plot the histogram in the cutting range. And another OPTIONAL “COLZ”: only for

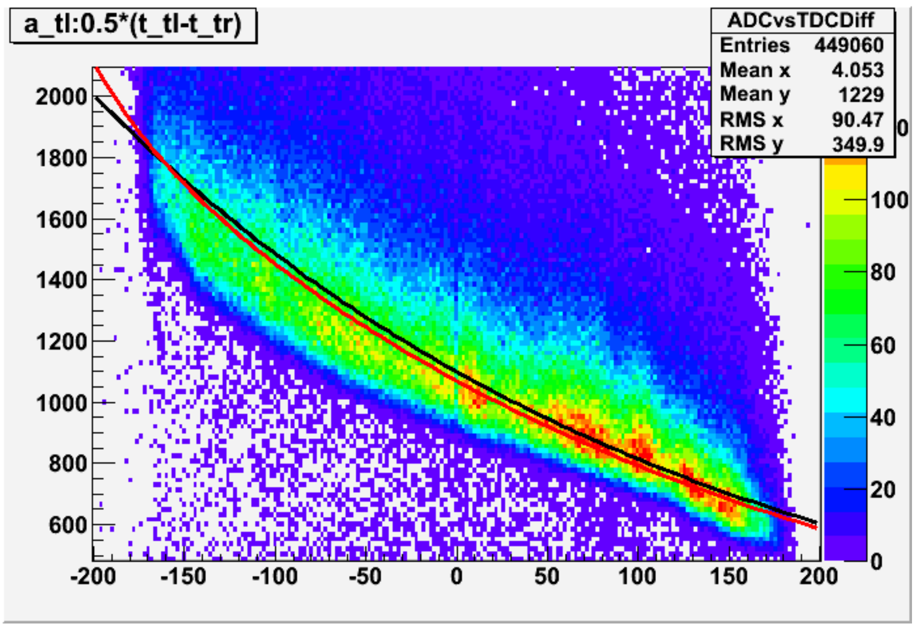
2-Dimensional histograms showed with color. Here, t1 is the name of the data tree.

3) Now, with this histogram, how can you get the attenuation length? What kind of fitting do you need to do?
Firstly, by eye-fitting method, we can try to write the fitting function and fit it on this 2-Dimensional histogram to see if it fits well . The fitting shows on picture3. If not, please think about what method will fit well.

TF1 *f = new TF1("f","[0]*exp(x*[1])+[2]*exp(x*[3])",-200.5,200.5);

f.SetParameters(1070,-0.003,0.4,-0.03);// It is interesting to play with the parameters.

f->Draw("same");



Picture3

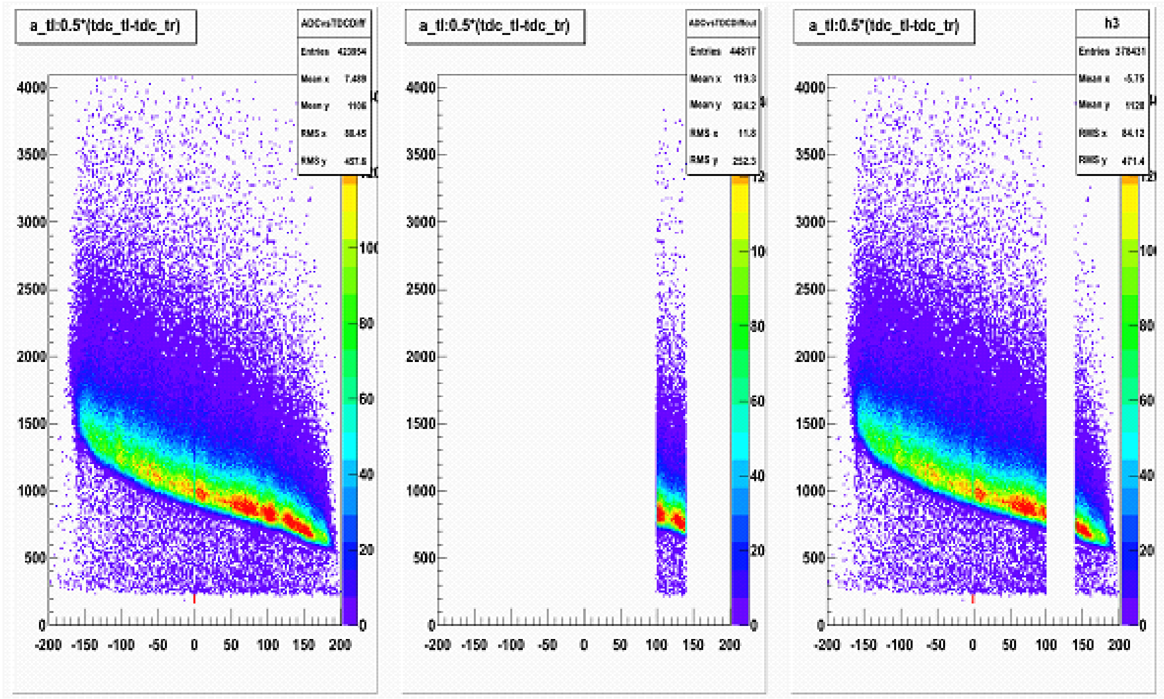
If you are satisfied with the fitting result, how do you calculate the attenuation length from these parameters?

Secondly, if you are not satisfied with the fitting result, you may need to cut on the TDC difference, as you did before for choosing three points to calculate the attenuation length. Now you can choose more points (remember to consider the statistic problems). And you get the idea about cutting range then. For example:

t1->Draw("a_tl:(t_tl-t_tr)/2>>ADCvsTDCDiffcut(201,-400.5,400.5,256,-0.5,4095.5)","(t_tl-t_tr)/2>100 && (t_tl-t_tr)/2<140","COLZ"); // With this cutting, you choose one point x= 120 bin, and the cutting range is ±20 bins.

Usually, after cutting, we need to make sure that the cutting is

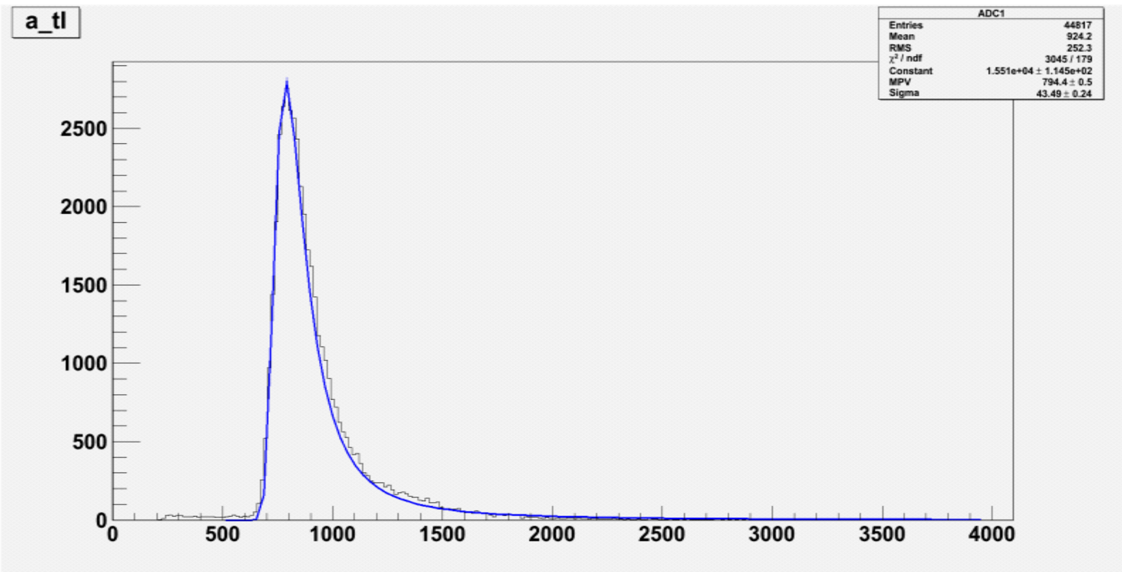
really active. So we can plot histograms called "Original", "Cut" and "Anti-cut", which shows on picture4. Please see reference code.



Picture4

4) After cutting, we need to find the ADC value of the cutting slit. Plot the ADC histogram and determine what kind of function do you use to fit the ADC histogram, Why?

t1->Draw("a_tl>>ADCtl(256,-0.5,4095.5)","(t_tl-t_tr)/2>100 && (t_tl-t_tr)/2<140"));



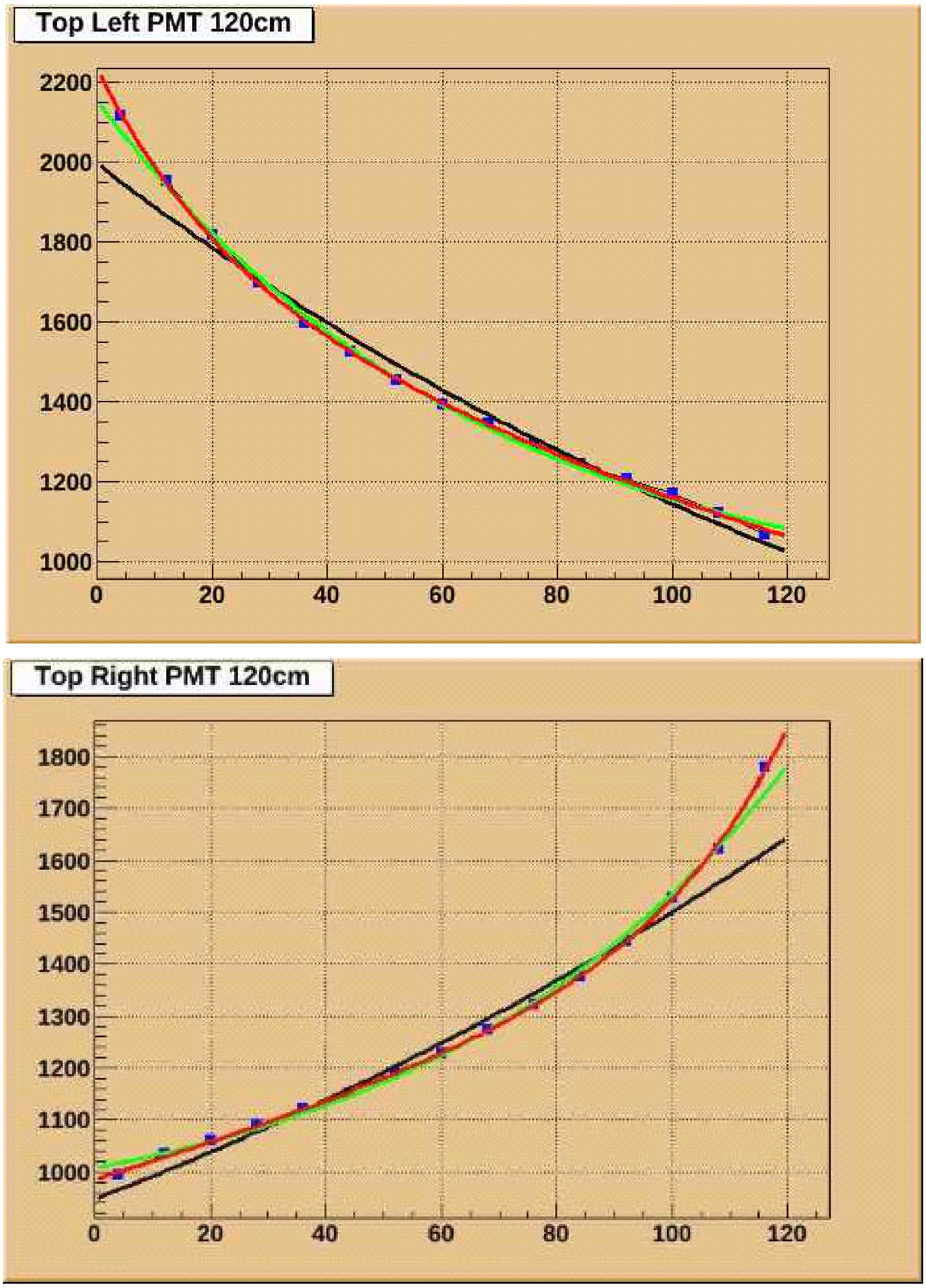
Picture5

Try right clicking on the distribution and fit by landau, which shows on picture5. After fitting, you will get the fitting parameters. We will use the fitting parameter "MPV" and its error bar and "x" value (bin number). For example, here x = 120 bins, we get MPV = 794.4 and its error bar is 0.5. However, there is only one point, you need use the same method to get more points.

5) After you get more points, we can use TGraph to plot point and fit them by exponential functions. Does one exponential function fit well? If not what about two exponential function? If it fit well, do you know the physics reason behind it?

For example, see Picture6, the blue points are the cutting points. And the functions I used here are:

- $A_1 exp(x/\lambda_1)$
- $A_2 exp(x/\lambda_2)$+A3exp(x/λ_3)
- $A_4 exp(x/\lambda_4)+y_0$



Picture6

We can compare the fitting attenuation length with the factory parameters of the scintillator.

6) How many attenuation length do you get from the best fitting. Try to think about the physics meaning of them?

Reference Code

1)Readrootfile.c

```
void tree1r() {
//read the Tree generated by tree1w and fill two histograms
//note that we use "new" to create the TFile and TTree
objects,
//to keep them alive after leaving this function.
TFile *f = new Tfile("511.root"); // If 511.root is not in the
current directory, you need to put the right path here.
TTree *t2 = (Ttree*)f->Get("t1");
  // t1 is the tree's name, you can change t2
//to what name you like.
Double_t a_tl = 0, a_tr = 0, tdc_tl = 0, tdc_tr = 0;
t2->SetBranchAddress("a_tl",&a_tl);
t2->SetBranchAddress("a_tr",&a_tr);
t2->SetBranchAddress("t_tl",&t_tl);
t2->SetBranchAddress("t_tr",&t_tr);
//create two histograms
//raw data plotting
TH1F *ADC
= new TH1F("ADC","ADC distribution",256,-0.5,4095.5);
TH2F *ADC_vs_TDCDiff = new
```

```
TH2F("ADC_vs_TDCDiff","ADC vs TDCDiff",201,-
400.5,400.5,256,-0.5,4095.5);
//read all entries and fill the histograms
Int_t nentries = (Int_t)t2->GetEntries();
for (Int_t i=0; i<nentries; i++) {
// If you like you can add cutting condition here and plot
//histograms
//
  t2->GetEntry(i);
  ADC->Fill(a_tl);
ADC_vs_TDCDiff->Fill((tdc_tl-tdc_tr)/2,a_tl);
}
//We do not close the file. We want to keep the generated
histograms
//we open a browser and the TreeViewer
if (gROOT->IsBatch()) return;

new TBrowser ();
t2->StartViewer();
//In the browser, click on "ROOT Files", then on "511lab.root"
//You can click on the histogram icons in the right panel to
draw
```

```
//them in the TreeViewer, follow the instructions in the Help.
}
```

2)cut.c

```
{
t1->Draw("a_tl:0.5*(t_tl-t_tr)>>ADCvsTDCDiff(201,-
200.5,200.5,256,-.5,4096.5)","","COLZ");
TH2F*h1 = (TH2F*)gROOT->FindObject("ADCvsTDCDiff");
t1->Draw("a_tl:0.5*(t_tl-t_tr)>>ADCvsTDCDiffcut(201,-
200.5,200.5,256,-.5,4096.5)","(t_tl-t_tr)/2>100 && (t_tl-
t_tr)/2<140","COLZ");
TH2F*h2 = (TH2F*)gROOT-
>FindObject("ADCvsTDCDiffcut");
TH2F*h3 = h1->Clone("h3");
TCanvas*myCanvas = new TCanvas();
myCanvas->Divide(3);
myCanvas->cd(1);
  h1->Draw("COLZ");
        myCanvas->cd(2);
      h2->Draw("COLZ");
      myCanvas->cd(3) ;
      h3->Add(h2,-1) ;
```

```
    h3->Draw("COLZ");
    gStyle->SetPalette(1) ;
}
```

3)TGraph.c

```
  void gerrors() {
    //Draw a graph with error bars
TCanvas *c1 = new
TCanvas("c1","landau",200,10,700,500);
c1->SetFillColor(42);
c1->SetGrid();
c1->GetFrame()->SetFillColor(21);
c1->GetFrame()->SetBorderSize(12);
  const Int_t n = 15;
Float_t x[n]   =
{4,12,20,28,36,44,52,60,68,76,84,92,100,108,116};
Float_t y[n]   =
{2116.85,1953.67,1817.31,1698.55,1598.46,1524.53,1453.3
5,1392.64,1345.48,1293.26,1245.56,1206.01,1168.88,1124.
16,1067.11};
Float_t ex[n] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

```
Float_t ey[n] =
{11.723,11.142,8.607,6.991,7.681,5.578,6.391,5.328,6.940,
5.866,5.250,5.282,4.991,4.946,4.963};
TGraphErrors *gr = new TGraphErrors(n,x,y,ex,ey);
gr->SetTitle("Top Left PMT 120cm");
gr->SetMarkerColor(4);
gr->SetMarkerStyle(21);
gr->Draw("ALP");//"ALP",plese see the Draw Options of Root
c1->Update();
//Fit Function
        Double_t par[8];
TF1 *g1       = new TF1("g1","expo",0.0,120.0); // "expo" An
Exponential with 2 parameters: f(x) = exp(p0+p1*x)
TF1 *g2       = new TF1("g2","expo+[2]",0.0,120.0);//f(x) =
exp(p0+p1*x)+p2, p2 is the offset parameter by checking the
cutting activity
TF1 *g3 = new TF1("g3","expo(0)+expo(2)",0.0,120.0);//f(x) =
exp(p0+p1*x)+exp(p2+p3*x)
     g2->SetLineColor(3);
     g3->SetLineColor(2);
     gr->Fit(g1,"R"); //"R",Use the range specified in the
function range
```

```
    gr->Fit(g2,"R+");
    gr->Fit(g3,"R+");
    g1->GetParameters(&par[0]);
    g2->GetParameters(&par[2]);
    g3->GetParameters(&par[5]);
}
```