Performance of Trigger counter for WC test

Toshiyuki Gogami

3Dec2014

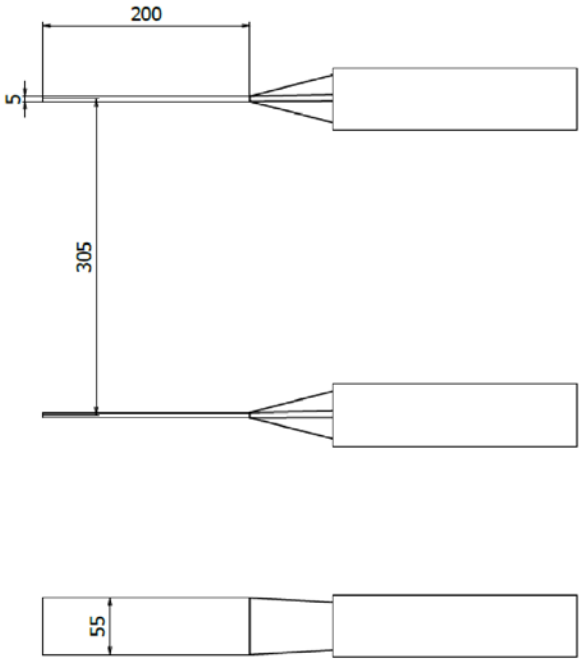
Contents

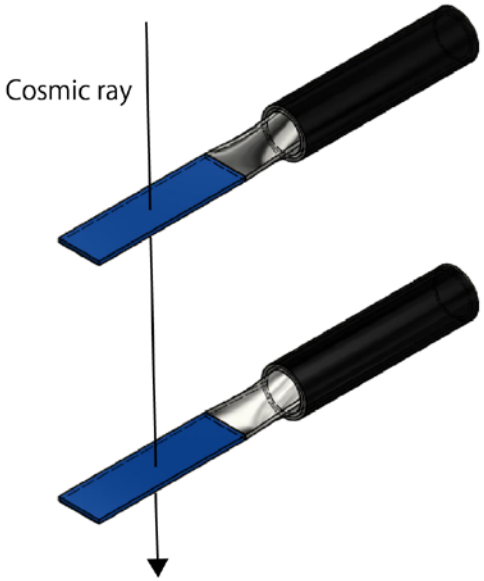
- Experimental setup
- Timing resolution

Goal

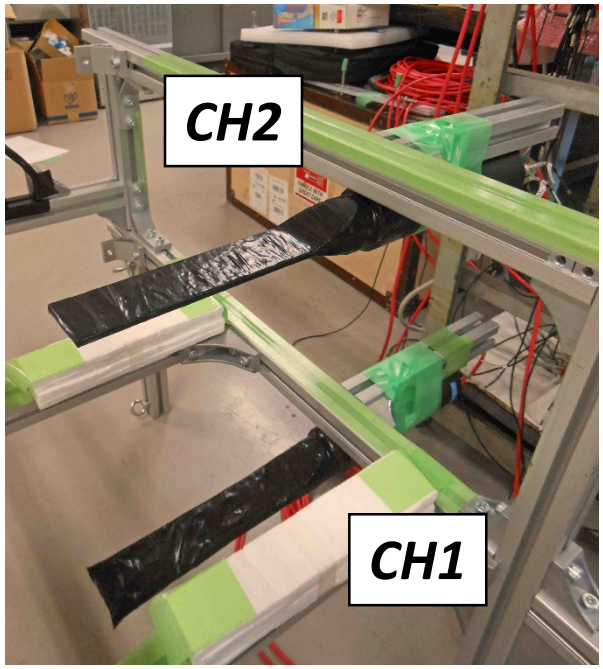
Check performance of trigger counters for WC window test which will be performed in a few weeks.

Experimental setup





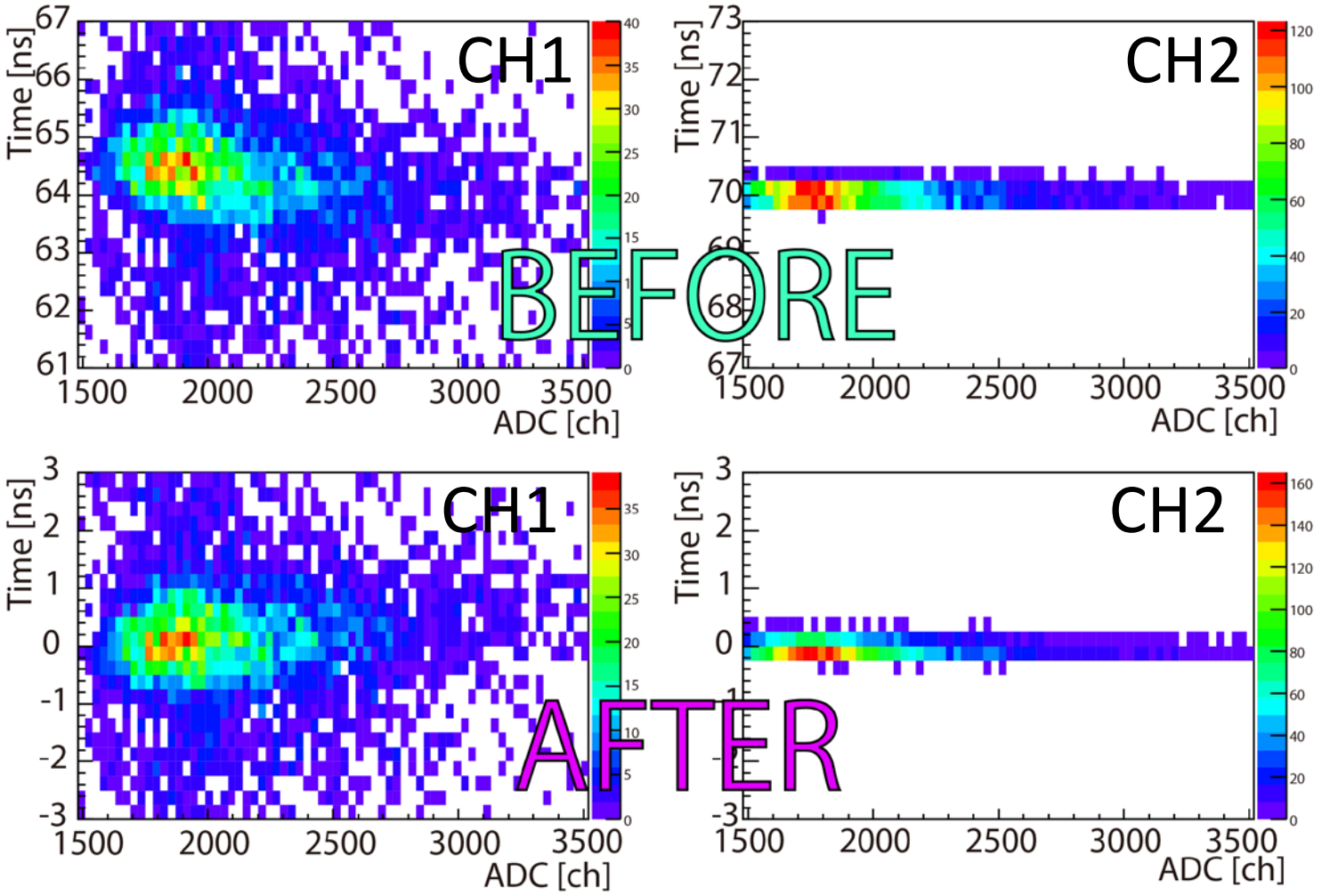
Unit: mm



Channel	ID	HV [V]	Remarks
1	H7195 (RD7241)	2300	Lower
2	H7195 (RD7198)	2300	Upper

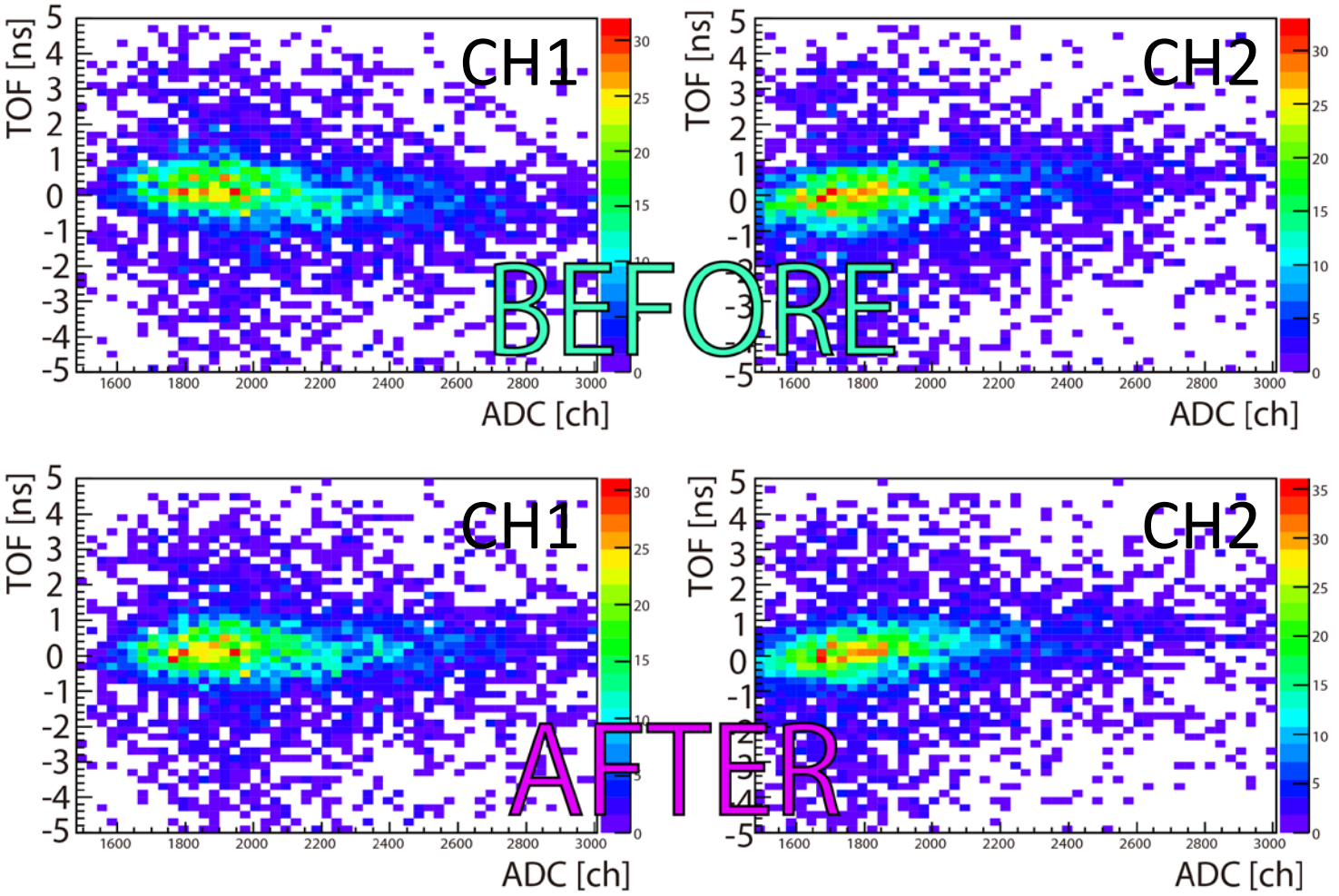
One night data
12/2 – 12/3
(RUN335-run338)

ADC vs. TDC
(Before and after correction)



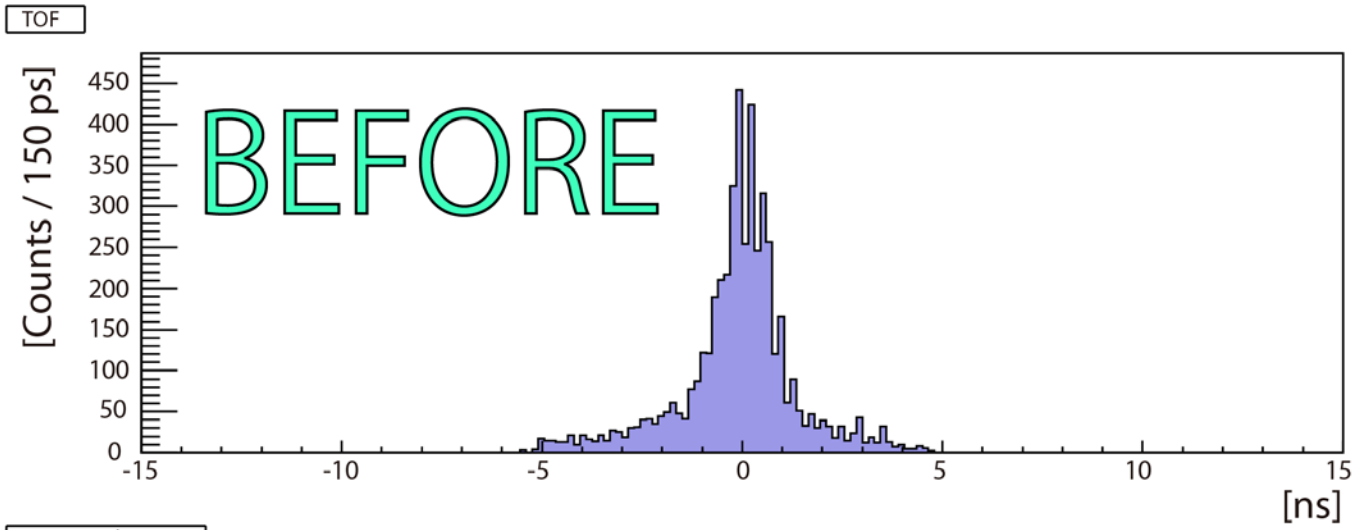
TOF vs. ADC
(Before and after correction)

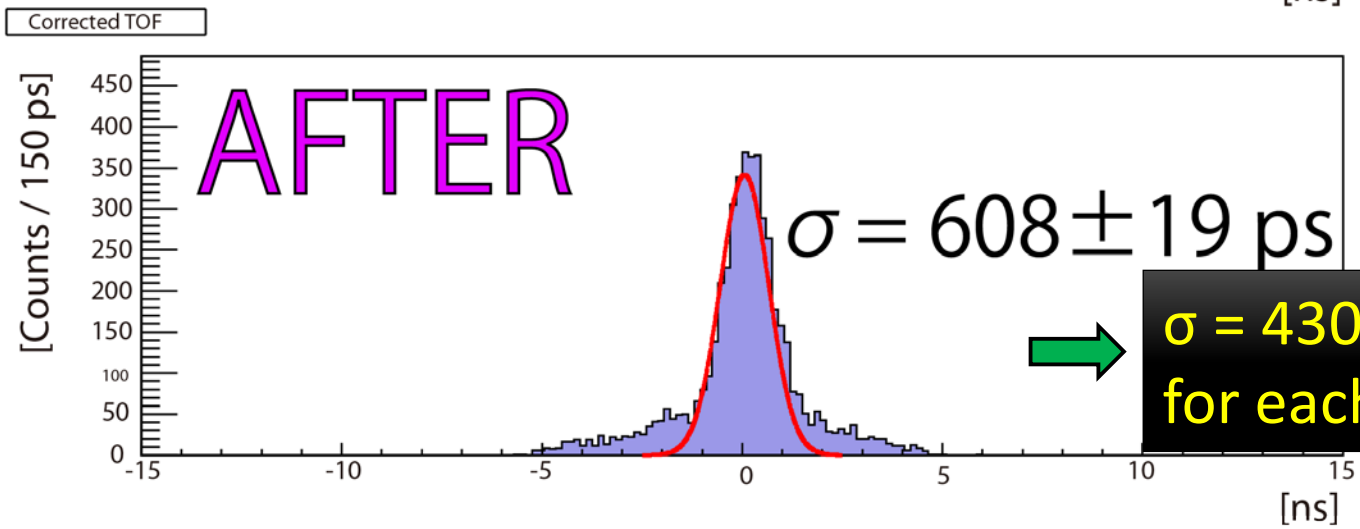
One night data
12/2 – 12/3
(RUN335-run338)



One night data
12/2 – 12/3
(RUN335-run338)

TOF distribution
Before and after correction





$\sigma = 430 \pm 14$ ps
for each scintillator

Summary

Plastic scintillation detectors which will be used as a trigger for WC window test was tested with cosmic ray @3F experimental room in Kyoto Univ.

TOF resolution: ~ 600 ps

→Timing resolution: 430 ps


なすびの志ば漬
椅出し中
Backup

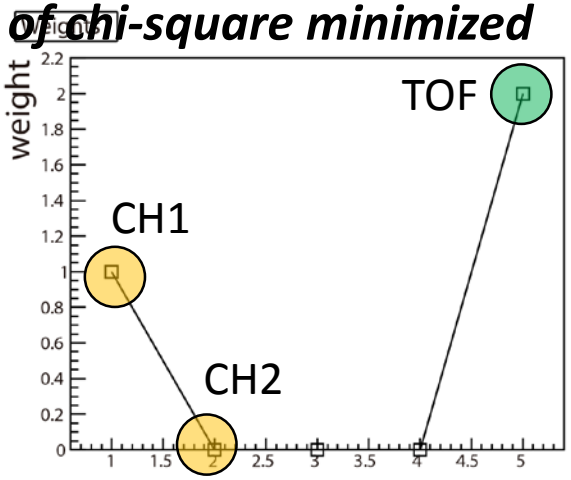
Reference

Performance of E05 TOF detector →

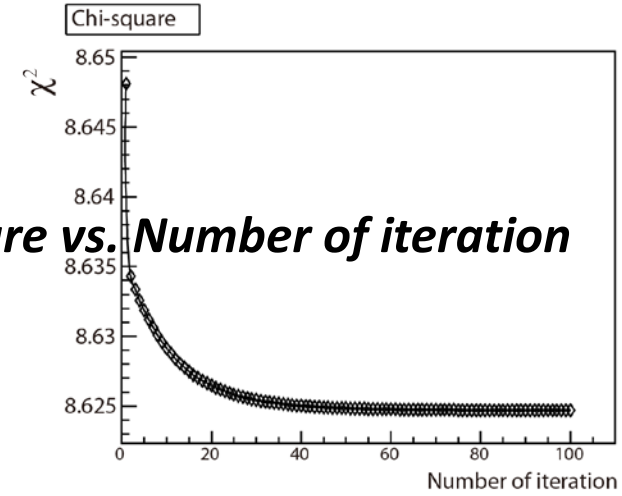
http://www-nh.scphys.kyoto-u.ac.jp/~gogami/s-2s/meeting/2014/gogami_S-2Smeeting(2014_10_16).pdf

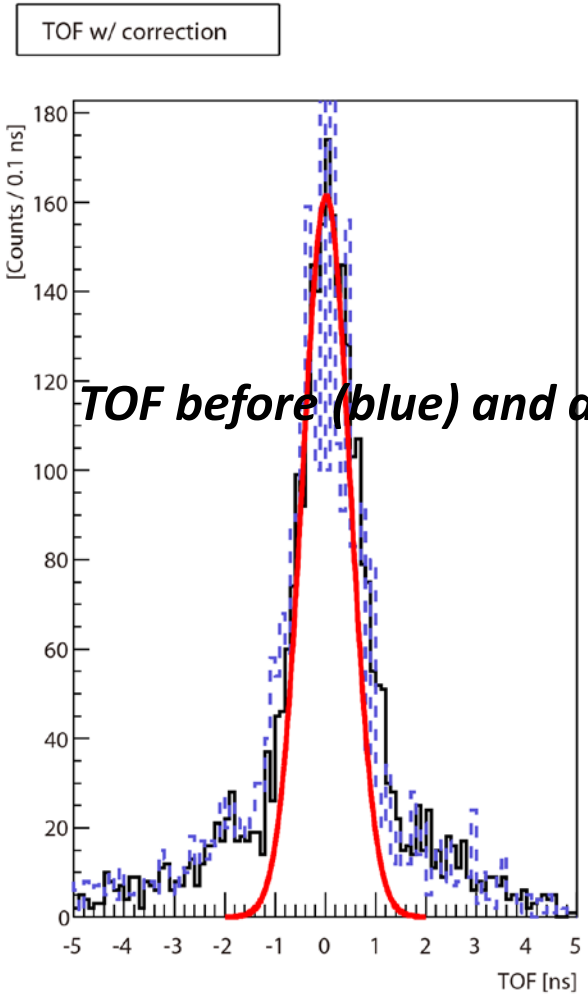
Pulse height correction

Weights of chi-square minimized



Chi-square vs. Number of iteration





TOF before (blue) and after correction

adc_tof2.cc (1)

```
/*
 "adc_tof2.cc"

 Toshi Gogami , 13Oct2014
*/

void adc_tof2(){
 // ====== General conditions =======
 gROOT->SetStyle("Plain");
 gStyle->SetOptStat(0);
 //gStyle->SetOptFit(1011);

 // ====== Open ROOT file ============
 //TFile* f1 = new TFile("toftest_20141013.root");
 //TFile* f1 = new TFile("run46.dat.root");
 //TFile* f1 = new TFile("toftest_run335-run337.root");
 TFile* f1 = new TFile("toftest_run335-run338.root");
 //TFile* f1 = new TFile("run7.dat.root");
 TTree* t1 = (TTree*)f1->Get("tree");

 // ====== Cut conditions ============
 /*
  TCut cut1 = "adc1>500 && adc1<2000.0 && abs(time1-67.0)<5.0";
  TCut cut2 = "adc2>500 && adc2<2000.0 && abs(time2-74.5)<5.0";
  TCut cut3 = "adc3>500 && adc3<2000.0 && abs(time3-71.0)<5.0";
  TCut cut4 = "adc4>500 && adc4<2000.0 && abs(time4-67.0)<5.0";
 */
 TCut cut1 = "adc1>1500 && adc1<4000.0 && abs(time1-64.1)<5.0";
 TCut cut2 = "adc2>1200 && adc2<4000.0 && abs(time2-70.0)<5.0";
 TCut cut3 = "abs(ctof)<10.0";
 TCut cut4 = "";
```

```
// ====== Create histograms =========
 //double amin = 0.0 , amax = 3000.0;
 double amin = 1000.0 , amax = 4000.0;
 double tmin = -100.0 , tmax = 100.0;
 int abin = 100;
 int tbin = 800;
 // ~~~ ADC1 vs. TDC1~~~~
 TH2F* hist1 = new TH2F("hist1","",
                                                    abin,amin,amax,
                                                    tbin,tmin,tmax);
 t1->Project("hist1","time1:adc1",cut1&&cut2&&cut3&&cut4);
 //t1->Project("hist1","time1:adc1",cut1&&cut2&&cut3);
 hist1->GetXaxis()->SetRangeUser(1500.0,3500.);
 //hist1->GetYaxis()->SetRangeUser(60.0,75.);
 //hist1->GetYaxis()->SetRangeUser(62.0,72.);
 //hist1->GetYaxis()->SetRangeUser(60.0,70.);
 hist1->GetYaxis()->SetRangeUser(61.0,67.);
 hist1->GetXaxis()->SetTitle("ADC [ch]");
 hist1->GetYaxis()->SetTitle("Time [ns]");
 // ~~~ ADC2 vs. TDC2~~~~
 TH2F* hist2 = (TH2F*)hist1->Clone("hist2");
 //hist2->GetYaxis()->SetRangeUser(71.0,78.);
 //hist2->GetYaxis()->SetRangeUser(65.0,75.0);
 hist2->GetYaxis()->SetRangeUser(67.0,73.0);
 t1->Project("hist2","time2:adc2",cut1&&cut2&&cut3&&cut4);
 // ~~~ ADC1 vs. corrected TDC1~~~~
 TH2F* hist1_ = (TH2F*)hist1->Clone("hist1_");
 //hist1_->GetYaxis()->SetRangeUser(-5.0,5.0);
 hist1_->GetYaxis()->SetRangeUser(-3.0,3.0);
 t1->Project("hist1_","ctime1:adc1",cut1&&cut2&&cut3&&cut4);
 // ~~~ ADC2 vs. corrected TDC2~~~~
 TH2F* hist2_ = (TH2F*)hist2->Clone("hist2_");
 //hist2_->GetYaxis()->SetRangeUser(-5.0,5.0);
 hist2_->GetYaxis()->SetRangeUser(-3.0,3.0);
 t1->Project("hist2_","ctime2:adc2",cut1&&cut2&&cut3&&cut4);
```

```
/*
 // ~~~ ADC3 vs. TDC3~~~~
 TH2F* hist3 = (TH2F*)hist1->Clone("hist3");
 hist3->GetYaxis()->SetRangeUser(65.0,74.);
 t1->Project("hist3","time3:adc3",cut1&&cut2&&cut3&&cut4);
 // ~~~ ADC4 vs. TDC4~~~~
 TH2F* hist4 = (TH2F*)hist1->Clone("hist4");
 hist4->GetYaxis()->SetRangeUser(62.0,70.);
 t1->Project("hist4","time4:adc4",cut1&&cut2&&cut3&&cut4);
 */

 // ~~~ Time Of Flight ~~~~

 TH1F* h_tof = new TH1F("h_tof","TOF",200,-15.0,15.0);
 //h_tof->GetYaxis()->SetRangeUser(62.0,70.);
 t1->Project("h_tof","tof",cut1&&cut2&&cut3&&cut4);
 h_tof->GetXaxis()->SetTitle("[ns]");
 h_tof->GetYaxis()->SetTitle("[Counts / 150 ps]");
 h_tof->SetFillColor(9);

 TH1F* h_tof_ = (TH1F*)h_tof->Clone("h_tof_");
 h_tof_->SetTitle("Corrected TOF");
 t1->Project("h_tof_","ctof",cut1&&cut2&&cut3&&cut4);

 double ha,hb;
 double hmax;
 ha = h_tof ->GetMaximum();
 hb = h_tof_->GetMaximum();
 hmax = ha;
 if(ha<hb) hmax = hb;
 h_tof ->SetMaximum(hmax*1.10);
 h_tof_->SetMaximum(hmax*1.10);
```

adc_tof2.cc (2)

```
// ~~~ ADC1 vs. TOF~~~~
 double tofmin = -10.0 , tofmax = 10.0;
 int tofbin = 100;
 TH2F* hist5 = new TH2F("hist5","",
                                                    abin,amin,amax,
                                                    tofbin,tofmin,tofmax);
 t1->Project("hist5","tof:adc1",cut1&&cut2&&cut3&&cut4);
 hist5->GetYaxis()->SetRangeUser(-5.0,5.0);
 //hist5->GetXaxis()->SetRangeUser(500.0,1500.);
 hist5->GetXaxis()->SetRangeUser(1500.0,3000.);
 hist5->GetXaxis()->SetTitle("ADC [ch]");
 hist5->GetYaxis()->SetTitle("TOF [ns]");
 // ~~~ ADC2 vs. TOF~~~~
 TH2F* hist6 = (TH2F*)hist5->Clone("hist6");
 t1->Project("hist6","tof:adc2",cut1&&cut2&&cut3&&cut4);

 // ~~~ ADC1 vs. corrected TOF~~~~
 TH2F* hist5_ = (TH2F*)hist5->Clone("hist5_");
 t1->Project("hist5_","ctof:adc1",cut1&&cut2&&cut3&&cut4);

 // ~~~ ADC2 vs. corrected TOF~~~~
 TH2F* hist6_ = (TH2F*)hist5->Clone("hist6_");
 t1->Project("hist6_","ctof:adc2",cut1&&cut2&&cut3&&cut4);
 /*
 // ~~~ ADC3 vs. TOF~~~~
 TH2F* hist7 = (TH2F*)hist5->Clone("hist7");
 t1->Project("hist7","tof:adc3",cut1&&cut2&&cut3&&cut4);
 // ~~~ ADC4 vs. TOF~~~~
 TH2F* hist8 = (TH2F*)hist5->Clone("hist8");
 t1->Project("hist8","tof:adc4",cut1&&cut2&&cut3&&cut4);
 */
```

```
TCanvas* c1 = new TCanvas("c1","c1");
 c1->Divide(2,2);
 //t1->Draw("time1:adc1");
 c1->cd(1);hist1->Draw("colz");
 c1->cd(2);hist2->Draw("colz");
 c1->cd(3);hist1_->Draw("colz");
 c1->cd(4);hist2_->Draw("colz");
 //c1->cd(3);hist3->Draw("colz");
 //c1->cd(4);hist4->Draw("colz");

 TCanvas* c2 = new TCanvas("c2","c2");
 c2->Divide(1,2);
 c2->cd(1);h_tof->Draw();
 c2->cd(2);h_tof_->Draw();


 // ===== Fitting =======
 //TF1* func1 = new TF1("func1","gaus",-3.0,3.0);
 //TF1* func1 = new TF1("func1","[0]*exp(x)",-3.0,3.0);
 TF1* func1 = new TF1("func1","[0]*exp(-0.5*pow((x-[1])/[2],2.0))",-2.5,2.5);
 func1->SetParameters(80.0,0.0,0.48);
 //func1->FixParameter(1,0.);
 func1->SetNpx(3000);
 //h_tof->Fit("func1","","",-1.5.1.5);
 //h_tof->Fit("func1","","",-0.8.1.8);
 //h_tof->Fit("func1","","",-1.0.1.0);
 h_tof->Fit("func1","BN","",-0.9,0.9);
 func1->SetLineColor(2);
 func1->SetLineWidth(2);
 //func1->SetLineStyle(7);
 func1->Draw("same");
 double width = 0.0;
 double widther = 0.0;
 width = func1->GetParameter(2);
 width = width*1.0e+3; // ns --> ps
 widther = func1->GetParError(2);
 widther = widther*1.0e+3; // ns --> ps
 cout << " Corrected TOF width (sigma) = "
     << width << " +/- "
     << widther << " ps" << endl;
 cout << " --> single counter: "
     << width/sqrt(2.0) << " +/- "
     << widther/sqrt(2.0) << " ps" << endl;
```

```
 TCanvas* c3 = new TCanvas("c3","c3");
 c3->Divide(2,2);
 c3->cd(1);hist5->Draw("colz");
 c3->cd(2);hist6->Draw("colz");
 c3->cd(3);hist5_->Draw("colz");
 c3->cd(4);hist6_->Draw("colz");
 // c3->cd(3);hist7->Draw("colz");
 //c3->cd(4);hist8->Draw("colz");


 // ===== Print ======
 //c1->Print("toftest_adcVStdc_3Dec2014.eps","eps");
 //c1->Print("toftest_adcVStdc_3Dec2014.png","png");
 //c2->Print("toftest_tof_3Dec2014.eps","eps");
 //c2->Print("toftest_tof_3Dec2014.png","png");
 //c3->Print("toftest_adcVStof_3Dec2014.eps","eps");
 //c3->Print("toftest_adcVStof_3Dec2014.png","png");
}
```

```
/*
 Processing adc_tof2.cc...
 FCN=136.597 FROM MIGRAD   STATUS=CONVERGED     89 CALLS          90 TOTAL
 EDM=1.17707e-10   STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                          STEP         FIRST
 NO.   NAME      VALUE          ERROR         SIZE      DERIVATIVE
 1  p0        3.41618e+02  8.52499e+00  3.58102e-02  -8.54110e-07
 2  p1        5.86524e-02  1.51946e-02  8.65149e-05  -9.08946e-04
 3  p2        6.07689e-01  1.90831e-02  7.98488e-05  -3.48313e-04

 Corrected TOF width (sigma) = 607.689 +/- 19.0831 ps
 --> single counter: 429.701 +/- 13.4938 ps

 hyperdragon3:~/dragon/ana/s2s/TOFtest2014/ROOT/
 Toshiyuki Gogami , 9Dec2014
*/
```