
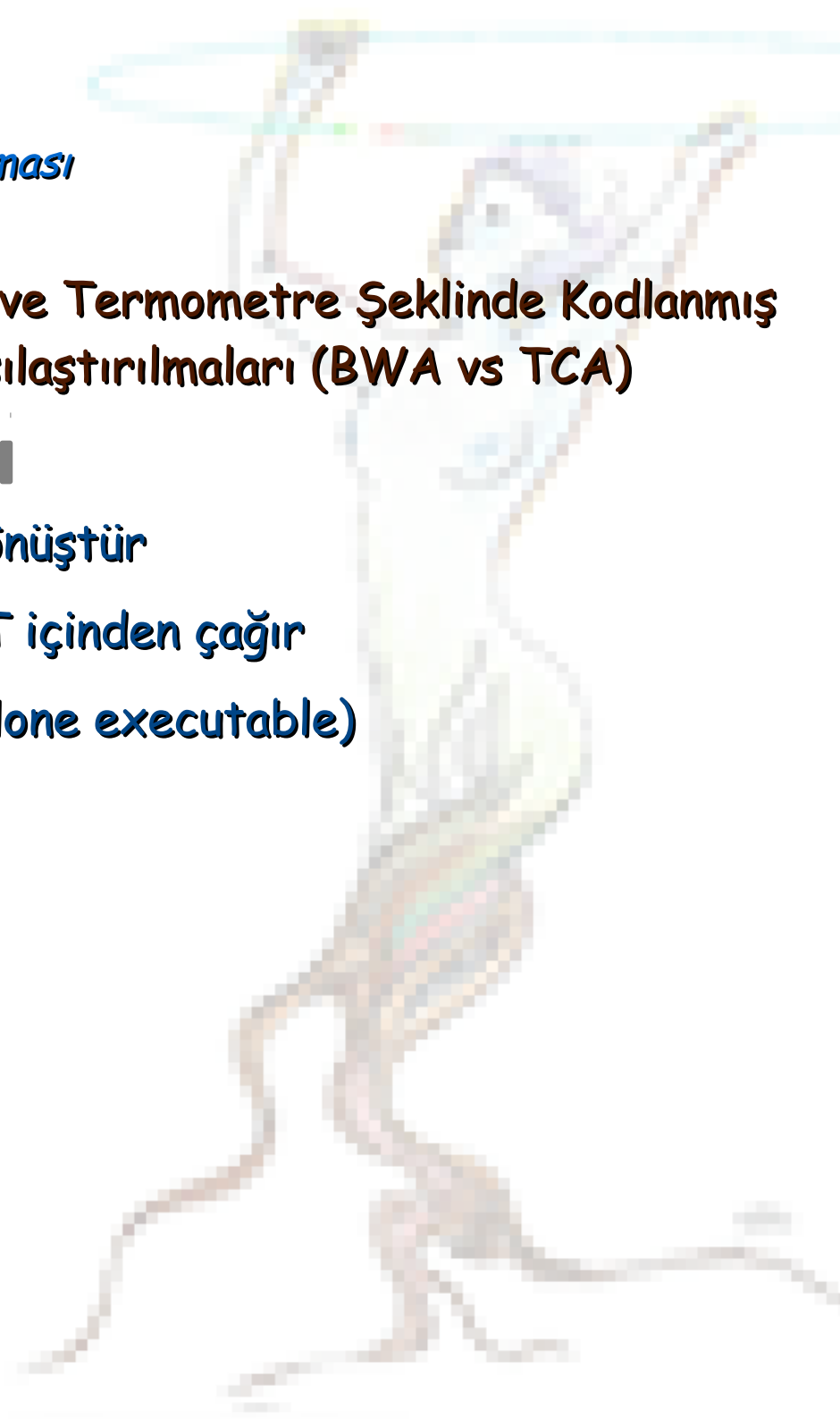


# ROOT Çalışması

## Veri Çevirici Mimarilerinin Karşılaştırılması

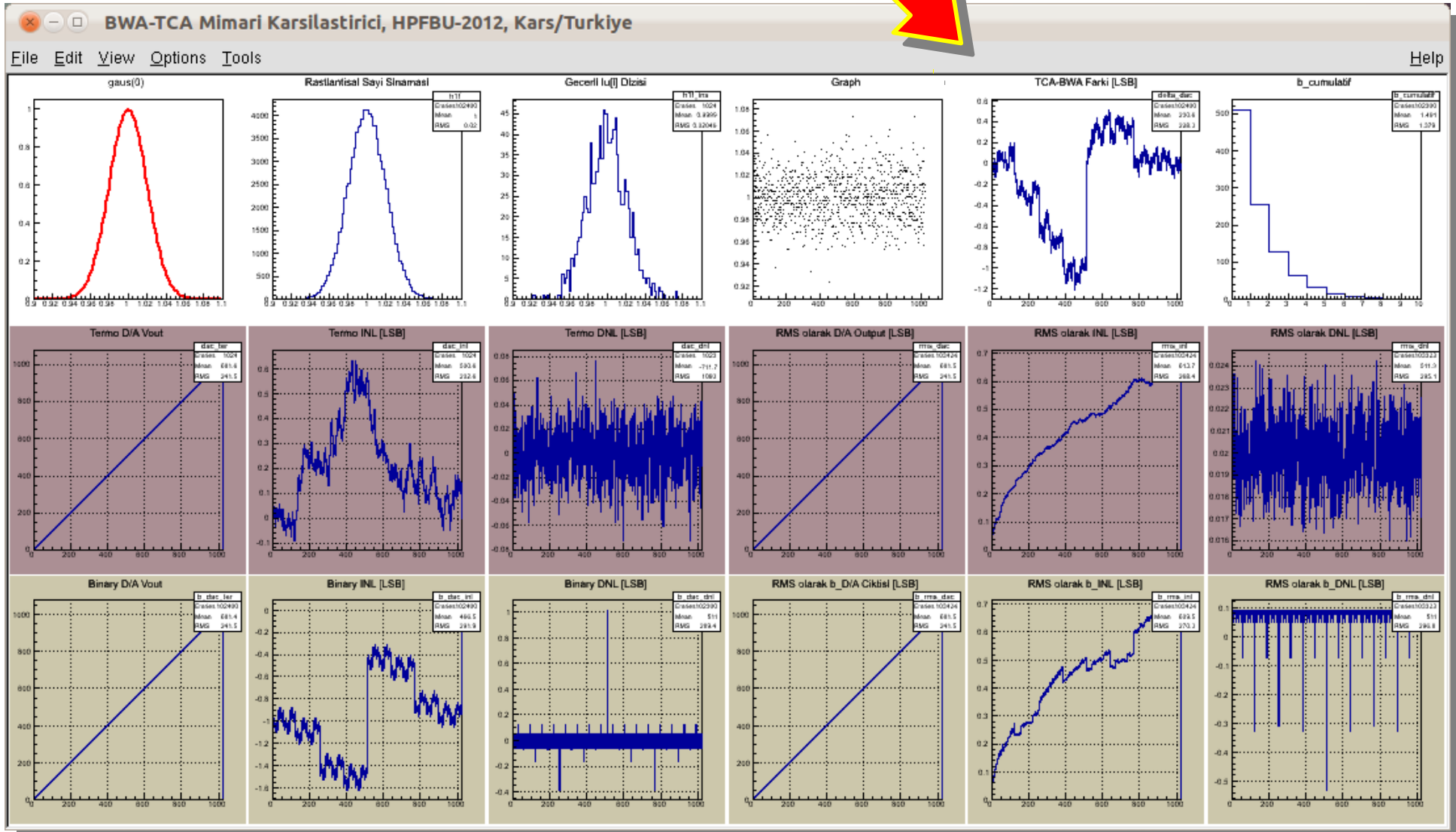
- ◆ İkilik Sisteme Göre Ağırlıklandırılmış ve Termometre Şeklinde Kodlanmış olmak Üzere İki Farklı Mimarinin Karşılaştırılmaları (BWA vs TCA)
- ◆ İsimsiz betik olarak çalıştır 
- ◆ İsimli betik haline (named script) dönüştür
- ◆ Paylaşılmış nesne (.so) üret ve ROOT içinden çağır
- ◆ Derle ve kendi başına kullan (standalone executable)
- ◆ GUI ile etkileşimler



# ROOT Çalışması

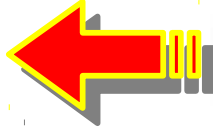
İsimsiz betik olarak çalıştır

- Komut satırında **cd** komutu ile **Root\_Tutorial\_I** isimli dizine git
- Betiği **root -l binter.C** komutu ile çalıştır



# ROOT Çalışması

## Veri Çevirici Mimarilerinin Karşılaştırılması

- ◆ İkilik Sisteme Göre Ağırlıklandırılmış ve Termometre Şeklinde Kodlanmış olmak Üzere İki Farklı Mimarinin Karşılaştırılmaları (BWA vs TCA)
- ◆ İsimsiz betik olarak çalıştır
- ◆ İsimli betik haline (named script) dönüştür 
- ◆ Paylaşılmış nesne (.so) üret ve ROOT içinden çağır
- ◆ Derle ve kendi başına kullan (standalone executable)
- ◆ GUI ile etkileşimler

# ROOT Çalışması

İsimli betik haline (named script) dönüştür

→ Sadece betiğe **isim** vermek **yeterli**:

```
12  ...
13  hibridizasyonu (mimaride hangi bitlerin binary hangil
14  termometre kodlu olacagi) belirler ki bu islev henuz
15
16  Kullanim:
17  Benzetim degiskenlerini ayarlayip, betigi calistirin:
18  > root binter.C
19
20  */
21
22  {
23  gROOT->Reset();
24
25  /*
26  Benzetim degiskenleri kısmi
27  */
28
29  int        cozunurluk           = 10;
30  int        rms_icin_iterasyon_sayisi = 100;
31  int        dagilimdaki_bolme_sayisi = 100;
32  double     belirsizlik          = 0.02;
33  double     merkez               = 1.0;
34  const Int t KYENiLE             = 50;
35
36  ...
37  ...
38  ...
39  ...
40  ...
41  ...
42  ...
43  ...
44  ...
45  ...
46  ...
47  ...
48  ...
49  ...
50  ...
51  ...
52  ...
53  ...
54  ...
55  ...
56  ...
57  ...
58  ...
59  ...
60  ...
61  ...
62  ...
63  ...
64  ...
65  ...
66  ...
67  ...
68  ...
69  ...
70  ...
71  ...
72  ...
73  ...
74  ...
75  ...
76  ...
77  ...
78  ...
79  ...
80  ...
81  ...
82  ...
83  ...
84  ...
85  ...
86  ...
87  ...
88  ...
89  ...
90  ...
91  ...
92  ...
93  ...
94  ...
95  ...
96  ...
97  ...
98  ...
99  ...
100 ...
```

→ Kütük ismi ile betik isminin aynı olduğuna dikkat edin !

→ Çalıştırın: **root -l binter\_namedScript.C**

# ROOT Çalışması

İsimli betik haline (named script) dönüştür

→ Sadece betiğe **isim** vermek **yeterli**, ancak...

```
18 > root binter.C
19
20 */
21
22 {
23   gROOT->Reset();
24
25   /*
26    Benzetim degiskenleri kısmi
27   */
28 }
```

```
18 > root binter.C
19
20 */
21
22 binter_namedScript() {
23   gROOT->Reset();
24
25   /*
26    Benzetim degiskenleri kısmi
27   */
28 }
```

→ isimli betik durumunda **başka** bazı **şeyleri** de düzeltmek gerekebilir:

```
41 int   nokta_sayisi = pow(2.0, cozunurluk);
42 int   son artirilan = 0;
43 double Iu[ pow(2.0, 1.0 * cozunurluk) ]; // 2^cozunurluk tane biri
44
45 /*
46  Hesapların üzerinde görüntüleneceği GUI'nin insası
47 */
48
49 c1 = new TCanvas("c1","BWA-TCA Mimari Karsilastirici, HPFE");
50 pad1 = new TPad("pad1","Birim Akim Kaynagi",0.0,0.0,1.0,1.0);
51 pad1->Draw();
52 pad1->Divide(6, 3, 0.001, 0.001);
```

```
42 int   nokta_sayisi = pow(2.0, cozunurluk);
43 int   son artirilan = 0;
44 double Iu[ pow(2.0, 1.0 * 10) ]; // 2^cozunurluk tane biri
45
46 /*
47  Hesapların üzerinde görüntüleneceği GUI'nin insası
48 */
49
50 TCanvas *c1 = new TCanvas("c1","BWA-TCA Mimari Karsilastirici, HPFE");
51 TPad *pad1 = new TPad("pad1","Birim Akim Kaynagi",0.0,0.0,1.0,1.0);
52 pad1->Draw();
53 pad1->Divide(6, 3, 0.001, 0.001);
```

...ve:


```
92 delta_dac->Reset();
93
94 double dac[nokta_sayisi], inl[nokta_sayisi], dnl[nokta_sayisi], toplam=0.0;
95 double b dac[nokta_sayisi], b inl[nokta_sayisi], b dnl[nokta_sayisi], toplam=0.0;
96 dac_ter = new TH1F("dac_ter", "Termo D/A Vout", nokta_sayisi, 0.0, 1.0);
97 dac_inl = new TH1F("dac_inl", "Termo INL [LSB]", nokta_sayisi, 0.0, 1.0);
```

```
93 delta_dac->Reset();
94
95 double dac[1024], inl[1024], dnl[1023], toplam=0.0;
96 double b dac[1024], b inl[1024], b dnl[1023], toplam=0.0;
97 dac_ter = new TH1F("dac_ter", "Termo D/A Vout", nokta_sayisi, 0.0, 1.0);
98 dac_inl = new TH1F("dac_inl", "Termo INL [LSB]", nokta_sayisi, 0.0, 1.0);
```

→ tekrar çalıştırın ve aynı işi yaptığını görün: **root -l binter\_namedScript.C**

# ROOT Çalışması

## Veri Çevirici Mimarilerinin Karşılaştırılması

- ◆ İkilik Sisteme Göre Ağırlıklandırılmış ve Termometre Şeklinde Kodlanmış olmak Üzere İki Farklı Mimarinin Karşılaştırılmaları (BWA vs TCA)
- ◆ İsimsiz betik olarak çalıştır
- ◆ İsimli betik haline (named script) dönüştür
- ◆ Paylaşılmış nesne (.so) üret ve ROOT içinden çağır 
- ◆ Derle ve kendi başına kullan (standalone executable)
- ◆ GUI ile etkileşimler

# ROOT Çalışması

Paylaşılmış nesne (.so) üret ve ROOT içinden çağır

→ Başlık kütüklerini eklemek gerekli ve ...

```
14      termometre kodlu olacağı) belirler ki bu işlev henüz
15
16      Kullanım:
17      Benzetim değişkenlerini ayarlayıp, betiği çalıştırın:
18      > root binter.C
19
20  */
21
22  {
23      gROOT->Reset();
24
25      /*
26      Benzetim değişkenleri kısmı
27      */
28  }
```

```
19
20  */
21
22  #include<TCanvas.h>
23  #include<TROOT.h>
24  #include<TH1F.h>
25  #include<TF1.h>
26  #include<TGraph.h>
27
28  int binter_sharedObject() {
29
30      gROOT->Reset();
31
32      /*
```

→ bir de işletim sistemine bir **değer dödürmeliyiz**:

```
256      pad1->cd(17);
257      b_rms_inl->Draw();
258      pad1->cd(18);
259      b_rms_dnl->Draw();
260      if (yenile) c1->Update();
261 }
262
263
```

```
265      pad1->cd(18);
266      b_rms_dnl->Draw();
267      if (yenile) c1->Update();
268
269      return 0;
270 }
271
272
```

→ ACLIC ile derleyin ve **binter\_sharedScript\_C.so** isimli yeni bir kütüğün oluştuğunu görün:

**root -l binter\_sharedScript.C++**

# ROOT Çalışması

Paylaşmış nesne (.so) üret ve ROOT içinden çağır

→ Aldığınız hata mesajlarına göre gerekli düzeltmeleri yapın:

```
40
41 int   nokta_sayisi = pow(2.0, cozunurluk);
42 int   son artirilan = 0;
43 double Iu[ pow(2.0, 1.0 * cozunurluk) ]; // 2^cozunurluk tane birim
44
45 /*
46 Hesaplarin uzerinde goruntulenecegi GUI'nin insasi
47 */
48
49 c1 = new TCanvas("c1", "BWA-TCA Mimari Karsilastirici, HPFE");
50 pad1 = new TPad("pad1", "Birim Akim Kaynagi", 0.0, 0.0, 1.0, 1.0);
51 pad1->Draw();
52 pad1->Divide(6, 3, 0.001, 0.001);
53 pad1->cd(7) >SetFillColor(47); pad1->cd(7) >SetGrid();

47
48 int   nokta_sayisi = pow(2.0, cozunurluk);
49 int   son artirilan = 0;
50 double Iu[ (int)pow(2, 10) ]; // 2^cozunurluk tane birim
51
52 /*
53 Hesaplarin uzerinde goruntulenecegi GUI'nin insasi
54 */
55
56 TCanvas *c1 = new TCanvas("c1", "BWA-TCA Mimari Karsilastirici, HPFE");
57 TPad *pad1 = new TPad("pad1", "Birim Akim Kaynagi", 0.0, 0.0, 1.0, 1.0);
58 pad1->Draw();
59 pad1->Divide(6, 3, 0.001, 0.001);
60 pad1->cd(7) >SetFillColor(47); pad1->cd(7) >SetGrid();
```

ve:

```
74 gaus->Draw();
75 if (yenile) c1->Update();
76
77 rms_dnl = new TH1F("rms_dnl", "RMS olarak DNL [LSB]", nokt);
78 rms_inl = new TH1F("rms_inl", "RMS olarak INL [LSB]", nokt);
79 rms_dac = new TH1F("rms_dac", "RMS olarak D/A Output [LSB]", nokt);
80 b_rms_dnl = new TH1F("b_rms_dnl", "RMS olarak b_DNL [LSB]", nokt);
81 b_rms_inl = new TH1F("b_rms_inl", "RMS olarak b_INL [LSB]", nokt);
82 b_rms_dac = new TH1F("b_rms_dac", "RMS olarak b_D/A Ciktisi [LSB]", nokt);
83 b_cumulatif = new TH1F("b_cumulatif", "b_cumulatif", cozunurluk);
84 delta_dac = new TH1F("delta_dac", "TCA-BWA Farki [LSB]", r);
85
86 rms_dnl->Reset();
87 rms_inl->Reset();

81 gaus->Draw();
82 if (yenile) c1->Update();
83
84 TH1F *rms_dnl = new TH1F("rms_dnl", "RMS olarak DNL [LSB]", nokt);
85 TH1F *rms_inl = new TH1F("rms_inl", "RMS olarak INL [LSB]", nokt);
86 TH1F *rms_dac = new TH1F("rms_dac", "RMS olarak D/A Output [LSB]", nokt);
87 TH1F *b_rms_dnl = new TH1F("b_rms_dnl", "RMS olarak b_DNL [LSB]", nokt);
88 TH1F *b_rms_inl = new TH1F("b_rms_inl", "RMS olarak b_INL [LSB]", nokt);
89 TH1F *b_rms_dac = new TH1F("b_rms_dac", "RMS olarak b_D/A Ciktisi [LSB]", nokt);
90 TH1F *b_cumulatif = new TH1F("b_cumulatif", "b_cumulatif", cozunurluk);
91 TH1F *delta_dac = new TH1F("delta_dac", "TCA-BWA Farki [LSB]", r);
92
93 rms_dnl->Reset();
94 rms_inl->Reset();
```

→ **root -l binter\_sharedScript.C++** ile tekrar derlediğinizde bu sefer **binter\_sharedScript\_C.so** isimli yeni bir kütüğün oluştuğunu görün



# ROOT Çalışması

Paylaşılmış nesne (.so) üret ve ROOT için çağır

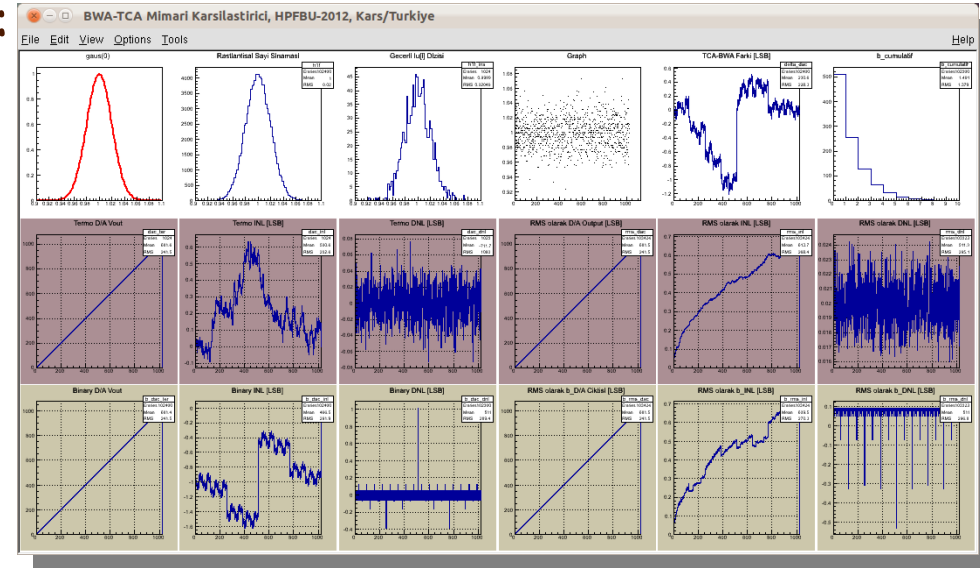
- ROOT'tan ".q" ile çıkın ve tekrar "root -l" ile yeni bir oturum açın
- Bir önceki oturumda yarattığınız **binter\_sharedScript\_C.so** isimli kitaplığı yükleyin:

```
"ROOT [0] .L binter_sharedScript_C.so"
```

- Artık isimli betiğimiz sanki bir ROOT komutuymuş gibi çalıştırılabilir:

```
"ROOT [1] binter_sharedObject()"
```

- Betiğinizin **aynı işi** yaptığını gözleyin:



# ROOT Çalışması

## Veri Çevirici Mimarilerinin Karşılaştırılması

- ◆ İkilik Sisteme Göre Ağırlıklandırılmış ve Termometre Şeklinde Kodlanmış olmak Üzere İki Farklı Mimarinin Karşılaştırılmaları (BWA vs TCA)
- ◆ İsimsiz betik olarak çalıştır
- ◆ İsimli betik haline (named script) dönüştür
- ◆ Paylaşılmış nesne (.so) üret ve ROOT içinden çağır
- ◆ Derle ve kendi başına kullan (standalone executable)
- ◆ GUI ile etkileşimler



# ROOT Çalışması

*Derle ve kendi başına kullan (standalone executable)*

- Uygulamamızı, **ROOT ortamına ihtiyaç duymadan** kendi başına çalışan bir hale getirmek istiyoruz.
- Ana işlevin adını **main** olarak değiştirin (artık C yazıyoruz)
- Daha öncekine benzer şekilde **başlık kütüklerini** ekleyeceğiz
  - Fakat hatırlayın **ROOT çalışma ortamının dışındayız !!**
  - Yani diğerine göre **daha çok başlık kütüğü** eklemeliyiz çünkü bu sefer **C dilinin standart kitaplığından** kullandığımız işlevleri de tanıtmalıyız
- **TApplication** nesnesi yaratıp, olay denetimini (event loop) ona bırakacağız
  - İşletim sistemine **döndürülecek değeri** de bu nesne döndürecek

# ROOT Çalışması

Derle ve kendi başına kullan (standalone executable)

→ ROOT ve C başlıklarını ekleyin, TApplication nesnesini yaratın:

```
13      hibridizasyonu (mimaride hangi bitlerin binary hangil
14      termometre kodlu olacağı) belirler ki bu islev henuz
15
16      Kullanim:
17      Benzetim degiskenlerini ayarlayip, betigi calistirin:
18      > root binter.C
19
20      */
21
22      {
23      gROOT->Reset();
24
25      /*
26      Benzetim degiskenleri kısmi
27      */
28
29      int      cozunurluk      = 10;
30      int      rms_icin_iterasyon_sayisi = 100;
31      int      dagilimdaki_bolme_sayisi = 100;
32      double   belirsizlik     = 0.02;
```

```
18 > root binter.C
19
20 */
21
22 #include<TCanvas.h>
23 #include<TROOT.h>
24 #include<TH1F.h>
25 #include<TF1.h>
26 #include<TGraph.h>
27 #include<TApplication.h>
28
29 #include<stdlib.h>
30 #include<math.h>
31
32 int main(int argc, char **argv) {
33
34     TApplication myApp("app", &argc, argv);
35
36     /*
37     Benzetim degiskenleri kısmi
38     */
```

→ TApplication nesnesini çalıştırıyoruz, olay çevrimini başlatıyoruz:

```
250     pad1->cd(1);
257     b_rms_inl->Draw();
258     pad1->cd(18);
259     b_rms_dnl->Draw();
260     if (yenile) c1->Update();
261 }
262
263
```

```
209     pad1->cd(10);
270     b_rms_dnl->Draw();
271     if (yenile) c1->Update();
272     myApp.Run();
273 }
274 }
275
276
```

→ Standart g++ ile derleyin:

**g++ binter\_standalone.C `root-config --glibs -cflags` -o binter\_standalone**

# ROOT Çalışması

Derle ve kendi başına kullan (standalone executable)

→ Aldığınız hata mesajlarına göre gerekli düzeltmeleri yapın:

```
40
41 int   nokta_sayisi = pow(2.0, cozunurluk);
42 int   son artirilan = 0;
43 double Iu[ pow(2.0, 1.0 * cozunurluk) ]; // 2^cozunurluk t → ← 50 double Iu[ (int)pow(2, 10) ]; // 2^cozunurluk tane birim e
44
45 /*
46 Hesaplarin uzerinde goruntulenecegi GUI'nin insasi
47 */
48
49 c1 = new TCanvas("c1", "BWA-TCA Mimari Karsilastirici, HPFE → ← 56 TCanvas *c1 = new TCanvas("c1", "BWA-TCA Mimari Karsilastir
50 pad1 = new TPad("pad1", "Birim Akim Kaynagi", 0.0, 0.0, 1.0, 1. 57 TPad *pad1 = new TPad("pad1", "Birim Akim Kaynagi", 0.0, 0.0,
51 pad1->Draw();
52 pad1->Divide(6, 3, 0.001, 0.001);
53 pad1->cd(7) >SetFillColor(47); pad1->cd(7) >SetGrid(); 55
54 */
55
```

ve:

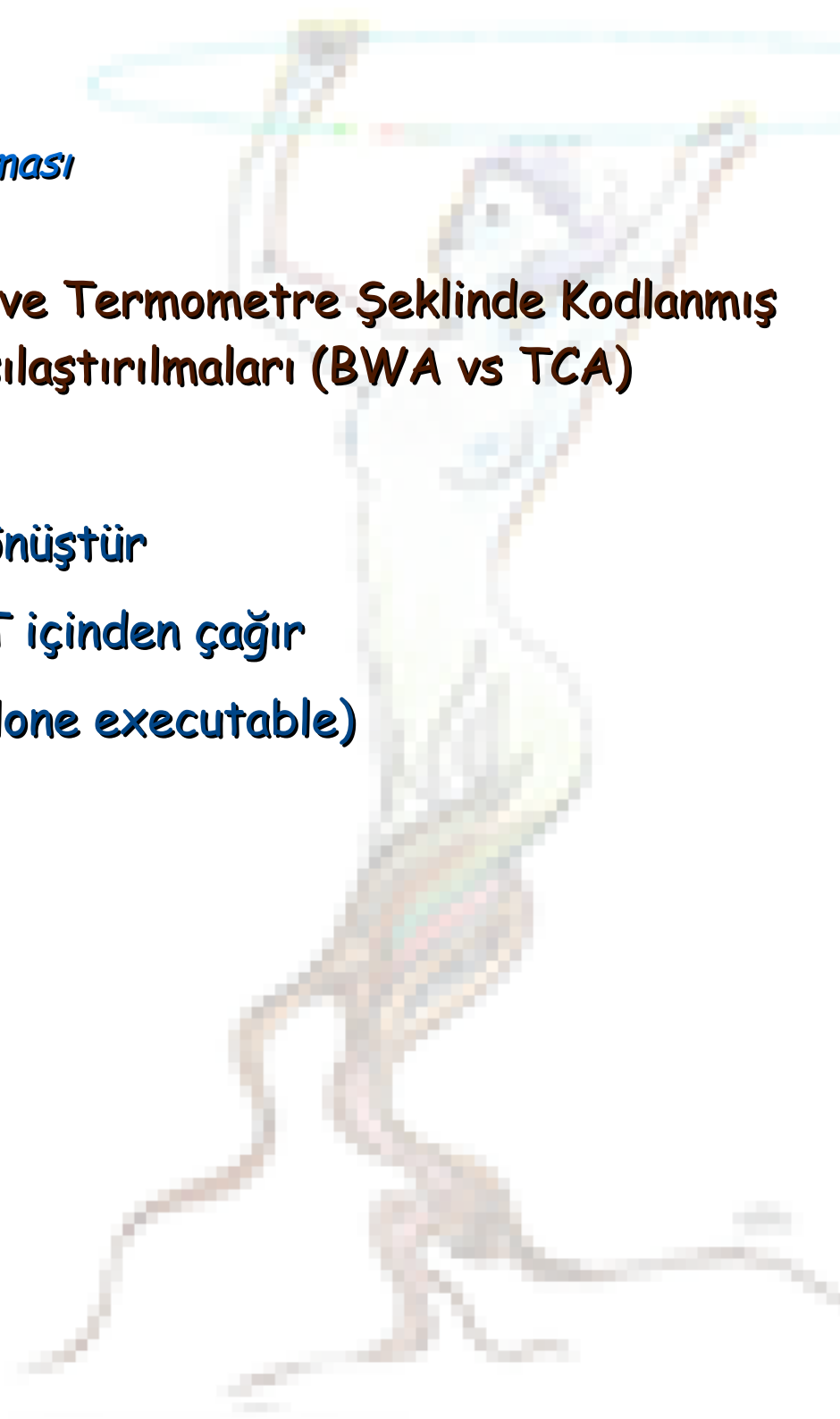
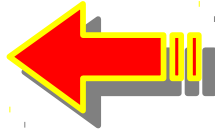
```
74 gaus->Draw();
75 if (yenile) c1->Update();
76
77 rms_dnl = new TH1F("rms_dnl", "RMS olarak DNL [LSB]", nokt → ← 84 TH1F *rms_dnl = new TH1F("rms_dnl", "RMS olarak DNL [LSB]"
78 rms_inl = new TH1F("rms_inl", "RMS olarak INL [LSB]", nokt 85 TH1F *rms_inl = new TH1F("rms_inl", "RMS olarak INL [LSB]"
79 rms_dac = new TH1F("rms_dac", "RMS olarak D/A Output [LSB] 86 TH1F *rms_dac = new TH1F("rms_dac", "RMS olarak D/A Output
80 b_rms_dnl = new TH1F("b_rms_dnl", "RMS olarak b_DNL [LSB]" 87 TH1F *b_rms_dnl = new TH1F("b_rms_dnl", "RMS olarak b_DNL
81 b_rms_inl = new TH1F("b_rms_inl", "RMS olarak b_INL [LSB]" 88 TH1F *b_rms_inl = new TH1F("b_rms_inl", "RMS olarak b_INL
82 b_rms_dac = new TH1F("b_rms_dac", "RMS olarak b_D/A Ciktis 89 TH1F *b_rms_dac = new TH1F("b_rms_dac", "RMS olarak b_D/A
83 b_cumulatif = new TH1F("b_cumulatif", "b_cumulatif", cozur 90 TH1F *b_cumulatif = new TH1F("b_cumulatif", "b_cumulatif",
84 delta_dac = new TH1F("delta_dac", "TCA-BWA Farki [LSB]", r 91 TH1F *delta_dac = new TH1F("delta_dac", "TCA-BWA Farki [LS
85
86 rms_dnl->Reset();
87 rms_inl->Reset();
92
93 rms_dnl->Reset();
94 rms_inl->Reset();
```

→ Tekrar derleyin, **“./binter\_standalone”** ile çalıştırın ve **aynı işi** gerçekleştirdiğini görün

# ROOT Çalışması

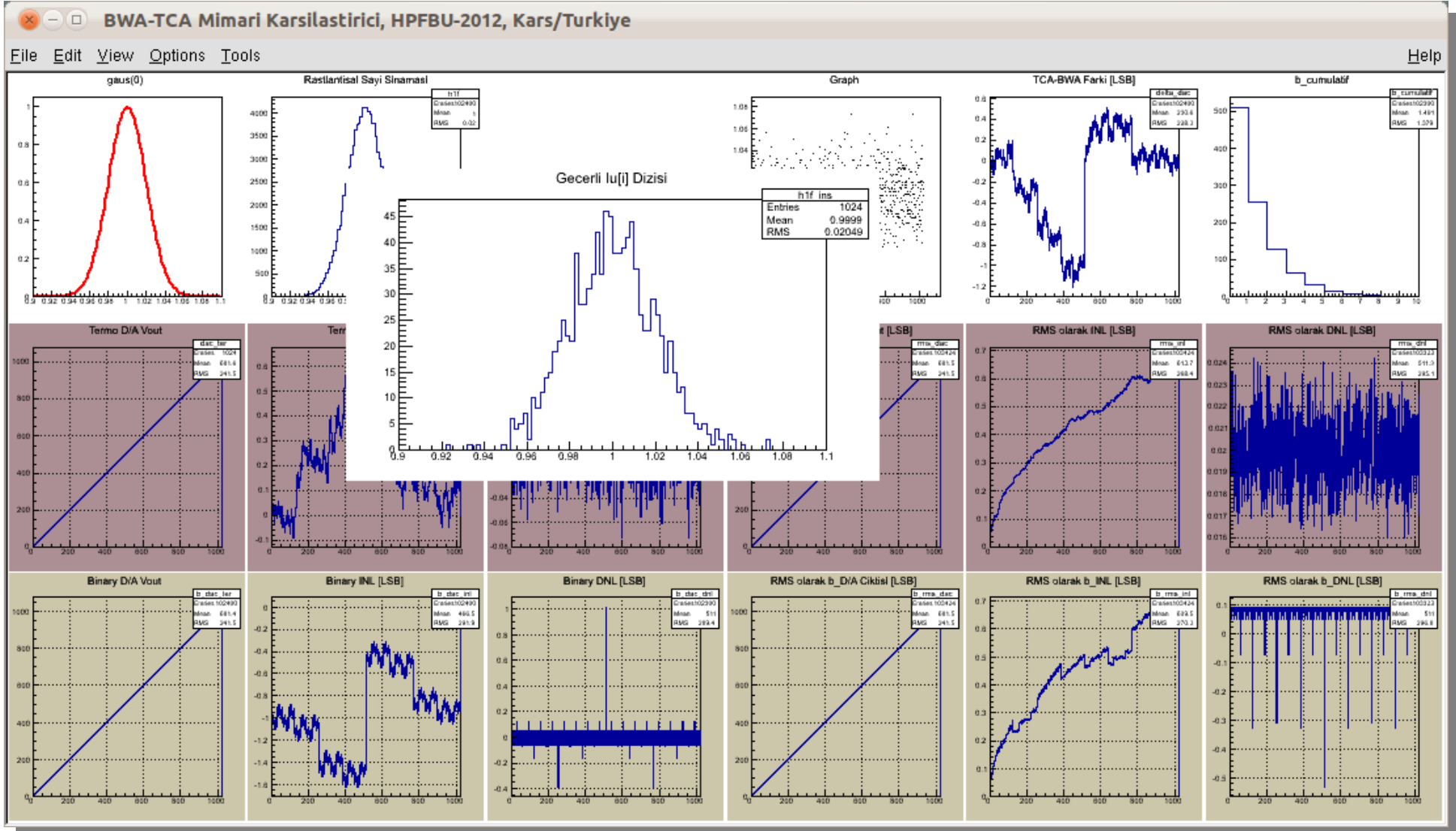
## Veri Çevirici Mimarilerinin Karşılaştırılması

- ◆ İkilik Sisteme Göre Ağırlıklandırılmış ve Termometre Şeklinde Kodlanmış olmak Üzere İki Farklı Mimarinin Karşılaştırılmaları (BWA vs TCA)
  - ◆ İsimsiz betik olarak çalıştır
  - ◆ İsimli betik haline (named script) dönüştür
  - ◆ Paylaşılmış nesne (.so) üret ve ROOT içinden çağır
  - ◆ Derle ve kendi başına kullan (standalone executable)
  - ◆ GUI ile etkileşimler



# ROOT Çalışması

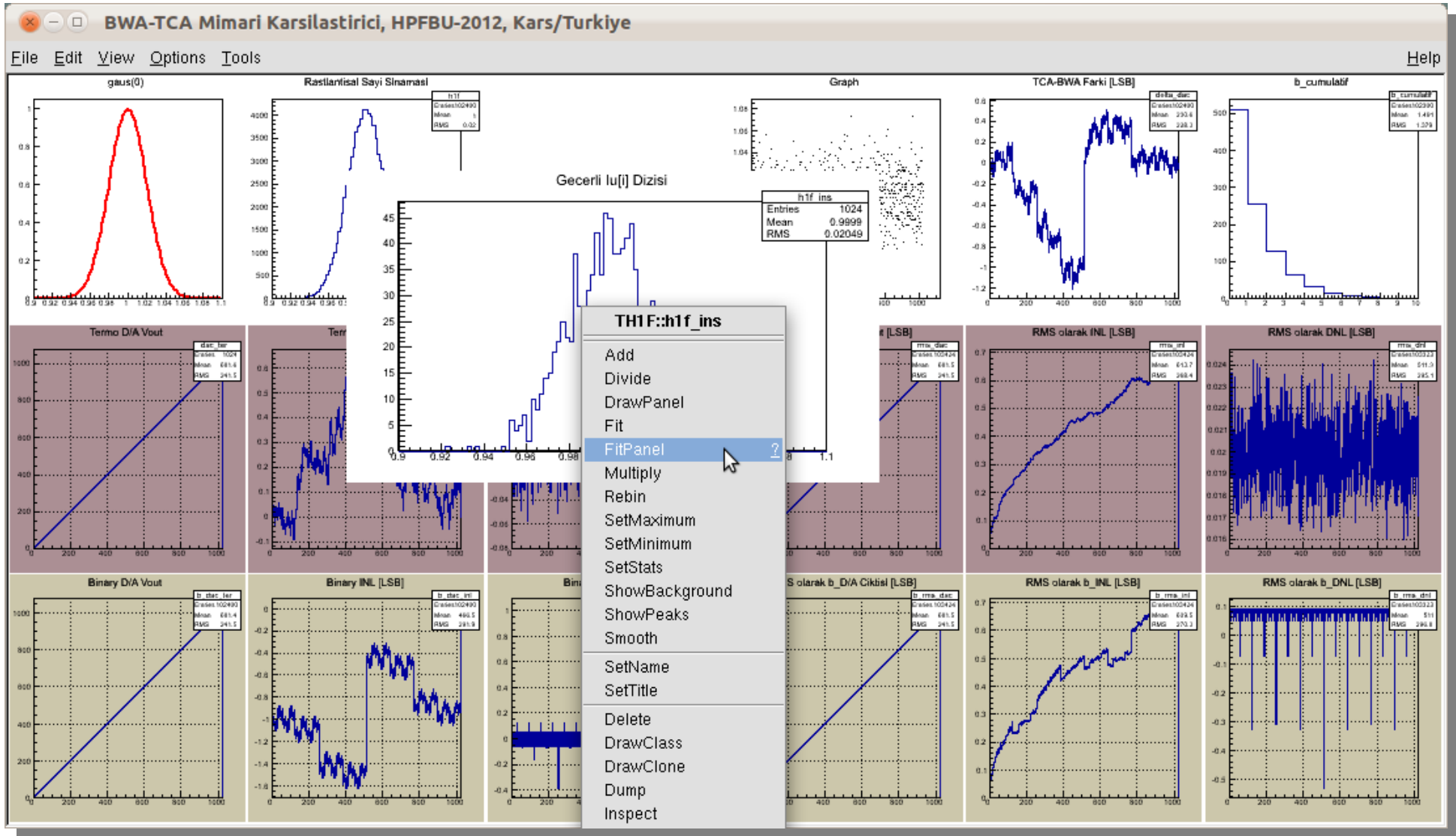
## GUI ile etkileşimler - Uyum eğrisi



→ Geçerli birim akım kaynaklarının tutulduğu **histogramın** içinde bulunduğu **pad'ı büyütün ve öne getirin:**

# ROOT Çalışması

## GUI ile etkileşimler - Uyum eğrisi



→ Histogramın üzerinde sağ tıklayın ve Fit Panel'i seçin



# ROOT Çalışması

GUI ile etkileşimler - Uyum eğrisi



1

2

3

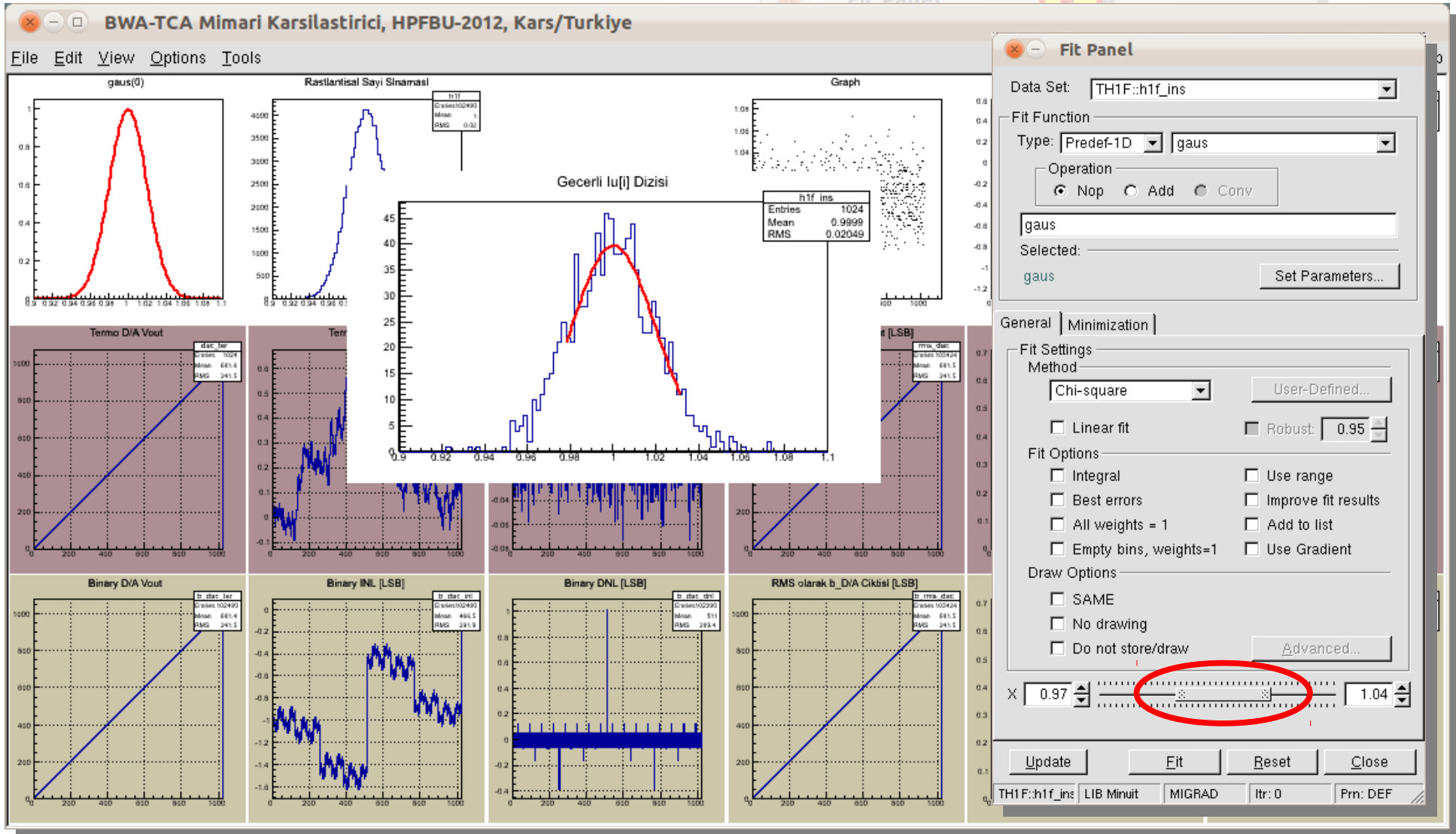
4

5

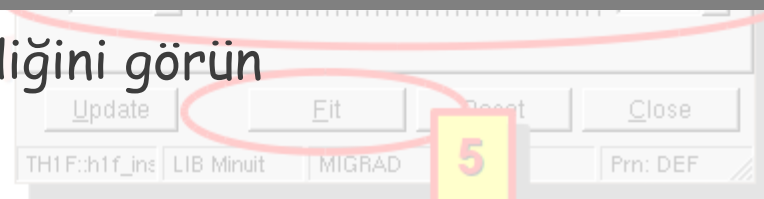
→ Histogramın üzerinde sağ tıklayın ve

# ROOT Çalışması

## GUI ile etkileşimler - Uyum eğrisi

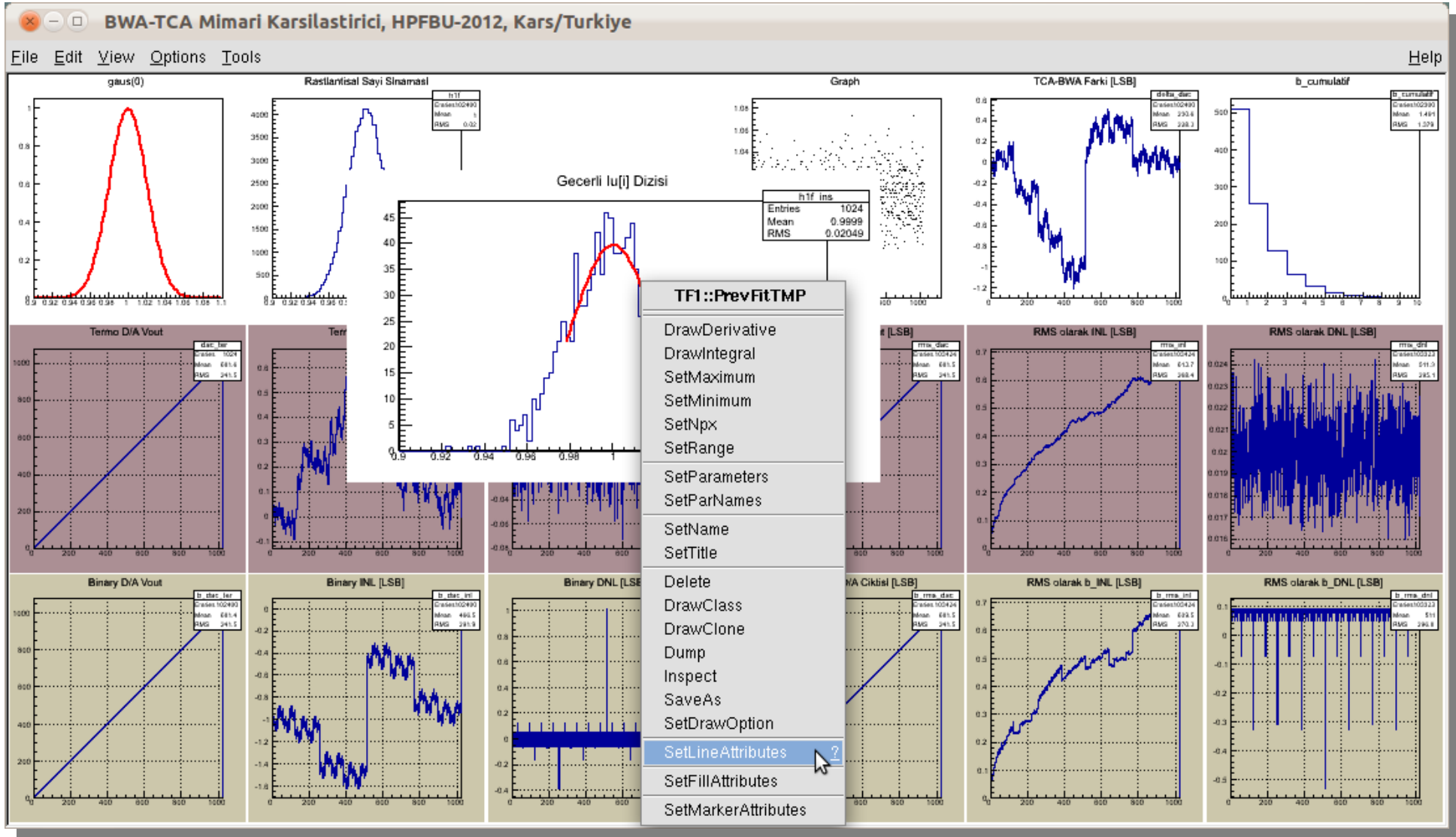


→ Uyum eğrisinin istenen aralıkta belirdiğini görün



# ROOT Çalışması

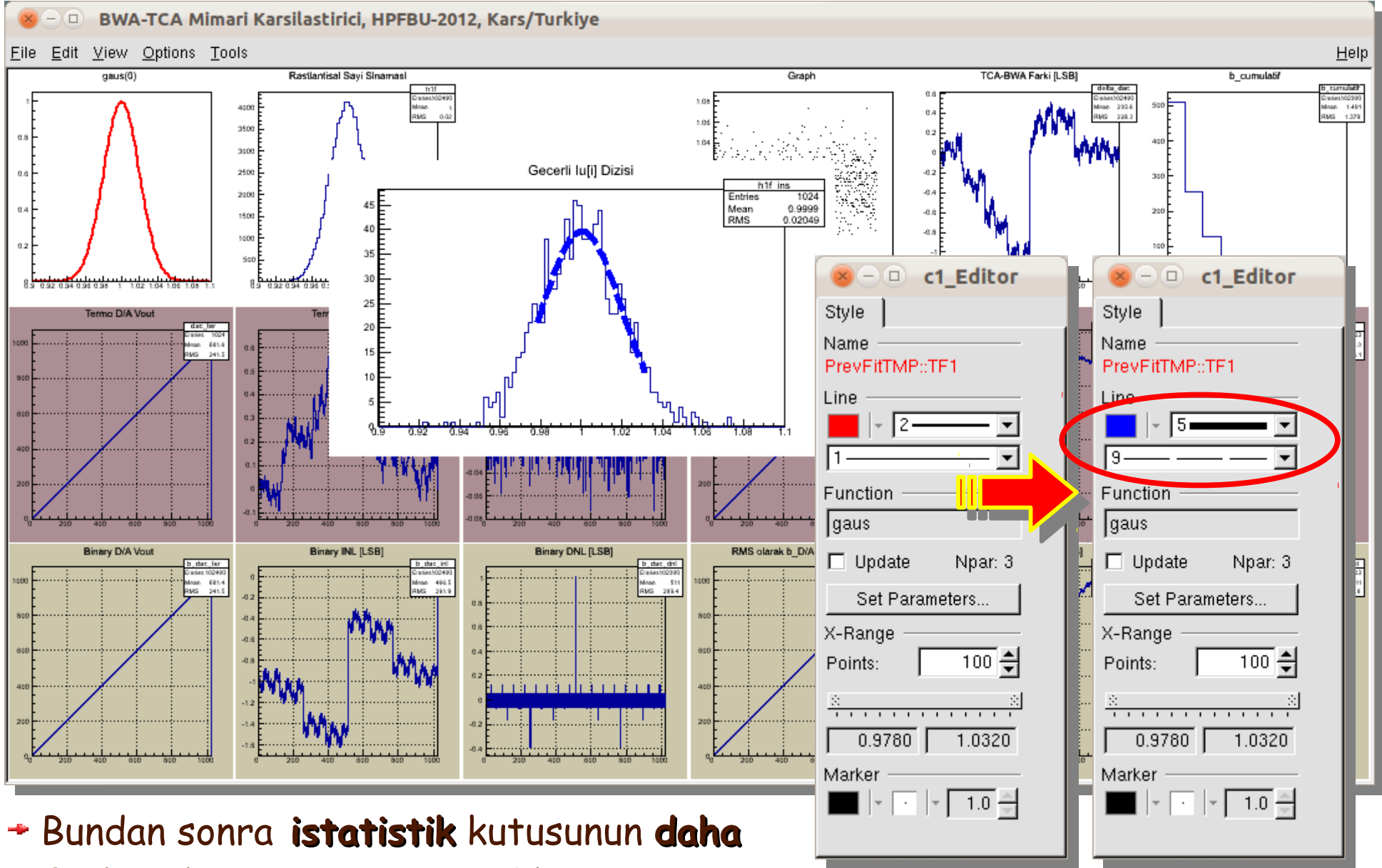
## GUI ile etkileşimler - Görsel



→ Uyum eğrisi üzerinde sağ tıklayın ve **SetLineAttributes**'i seçin

# ROOT Çalışması

## GUI ile etkileşimler - Görsel



→ Bundan sonra **istatistik** kutusunun **daha fazla bilgi** göstermesini sağlayın

# ROOT Çalışması

## Veri Çevirici Mimarilerinin Karşılaştırılması

- ◆ İkilik Sisteme Göre Ağırlıklandırılmış ve Termometre Şeklinde Kodlanmış olmak Üzere İki Farklı Mimarinin Karşılaştırılmaları (BWA vs TCA)
  - ◆ İsimsiz betik olarak çalıştır
  - ◆ İsimli betik haline (named script) dönüştür
  - ◆ Paylaşılmış nesne (.so) üret ve ROOT içinden çağır
  - ◆ Derle ve kendi başına kullan (standalone executable)
  - ◆ GUI ile etkileşimler

