

# Çözümleme Örneđi

Sezen Sekmen /  
Gökhan Ünel /

HPFBU okulu - Şubat 201

# Giriş

• Kimi SM ötesi modeller (örnek GUT) vektör kuark öneriyorlar :

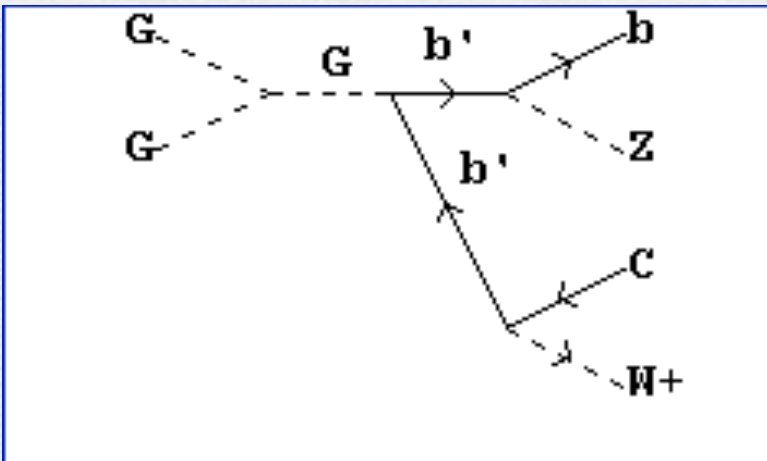
• Bu yeni kuark

• etkileşimleri SM ile aynı değil: Yüksüz Akım (*W ile*)

• kütlesi top'

• 13 TeV LHC ile

• üretim örneği



bozunma ürünlerinden başlayıp keşif yapabilir miyiz?

sinyal son durum:  $2j+3\ell + MET$



# Yapılacak işler

sinyal  $\rightarrow$   $2j+3\ell+MET$

ardalan=gürültü

aynı sondurum  $\rightarrow$   $2j+3\ell+MET$

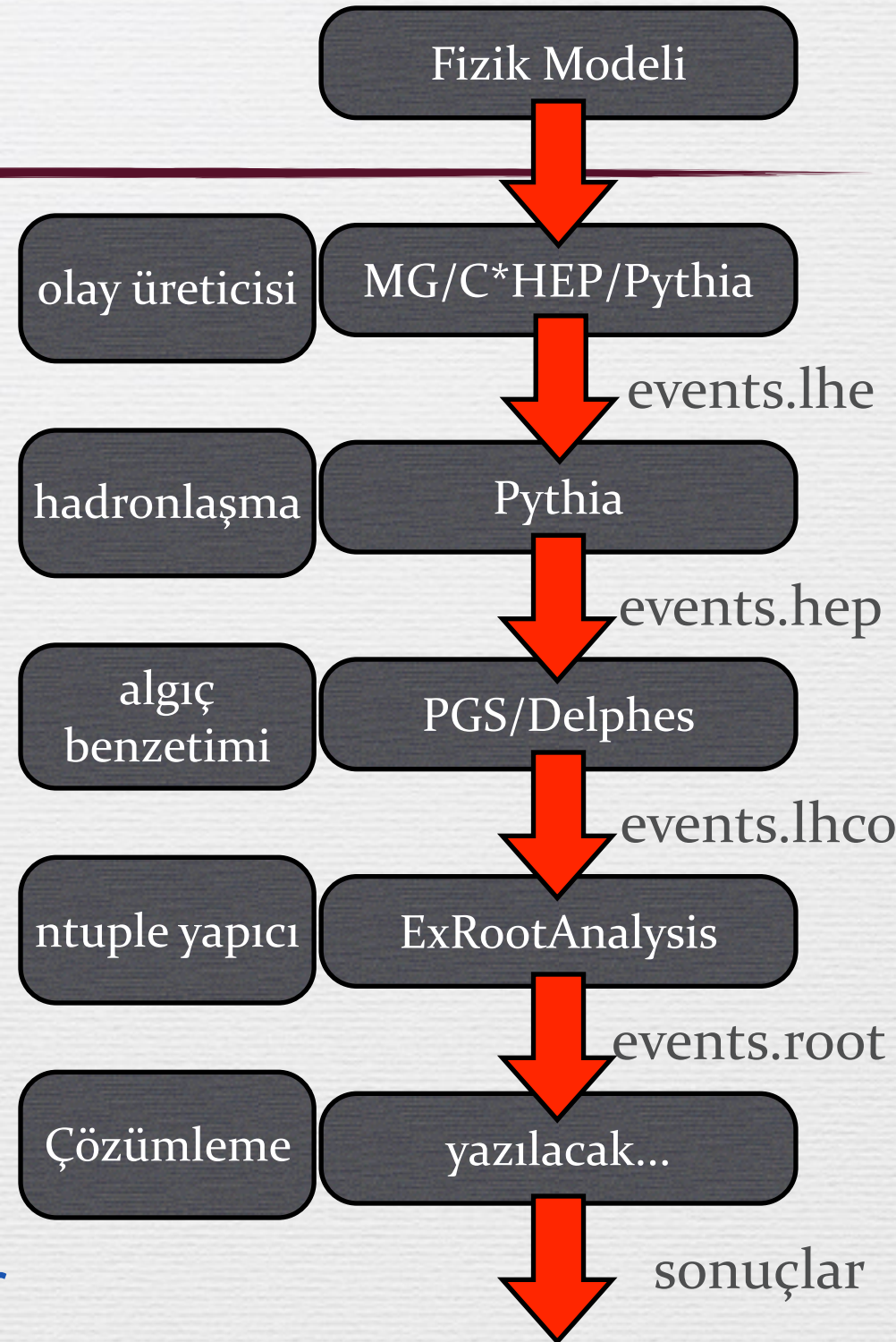
1. Sinyal olaylarını - öğrendiğimiz aletlerle üretelim:

1.1. CompHEP'den, b' sinyali

2. MG ile ardalanı hesaplayalım.

3. Hem sinyali hem ardalanı hadronlaştırıp benzetiminden geçirelim.

4. Bir çözümleme programı yazıp sinyali ardalandan ayırt edebiliyor muyuz bakalım.



# Ardalan

olay üreticisi

• Olayları üretin (örnek: MadGraph v4)

• `cd Events ; ls`

• `unweighted_events.lhe` üretilmiş olmalı

hadronlaşma

• Hadronlaşma yaptırın

• `../bin/run_pythia ; ls`

• `pythia_events.hep` üretilmiş olmalı

algıç benzetimi

• Algıç benzetimi

• `../bin/run_delphes ; ls`

• `delphes_events.lhco` üretilmiş olmalı

ntuple yapıcı

• ROOT dosyasını hazırlayın

• `../..//ExRootAnalysis/ExRootLHColympicsConverter  
pgs_events.lhco sm.root ; ls`

• `sm.root`

Çözümleme

bu dosyayı inceleyip, çözümlelim.

MG5 ile tek  
komutta  
yapılır!



# Çözümleme

• Önce ROOT ile bir çözümleme iskeleti yapalım,

• Laptopla çalışalım ki gece uykumuz kaçınca da işe edebilelim.

• ROOT ile kütüğü yükleyip, yazalım:

• root -l bp

• LHCO->MakeSelector

```
root [3] LHCO->MakeSelector("hpfbu_a")
Info in <TTreePlayer::MakeClass>: Files: hpfbu_a.h and hpfbu_a.C generated from TTree: LHCO
```

• Böylece 2 yeni kütük üretilmiş old

• hpfbu\_a.C ve hpfbu\_a.h

• bunlara daha

```
void hpfbu_a::SlaveBegin(TTree * /*tree*/)
{
    // The SlaveBegin() function is called after the
    // When running with PROOF SlaveBegin() is called
    // The tree argument is deprecated (on PROOF 0.9.1)
    TString option = GetOption();
    std::cout << "This is SlaveBEGIN " << std::endl;
    NJets = new TH1F ("NJets", "Njets", 10, -0.5, 9.5);
    fOutput->Add(NJets);
}
```

SlaveBegin

```
// The return value is currently not used.
```

```
GetEntry(entry);
if (entry<3) {
    std::cout << "Processing entry " << entry << std::endl;
    std::cout << "#j:" << Jet_ << std::endl;
}
NJets->Fill(Jet_);

return kTRUE;
```

```
std::cout << "This is TERMINATE " << std::endl;
NJets->Draw();
TFile tf("out.root", "recreate");
NJets->Write();
tf.Close();
```

5 terminate



# Çözümleme

## Çalıştırılım - sonucu görelim.

1 çizim bekliyoruz

## Sinyalden beklenti:

en az 2 jet, 3 lepton ve

```
if (Jet_ < 2) return kFALSE;  
if ((Electron_ + Muon_) < 3) return kFALSE;  
NJets->Fill(Jet_);
```

kötü tarz. Doğrusu: önce bak sonra kes.

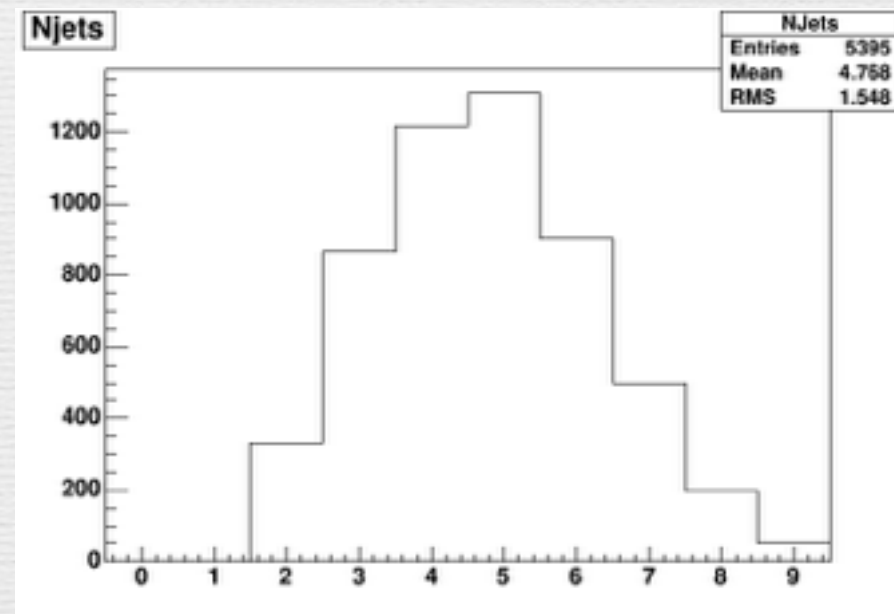
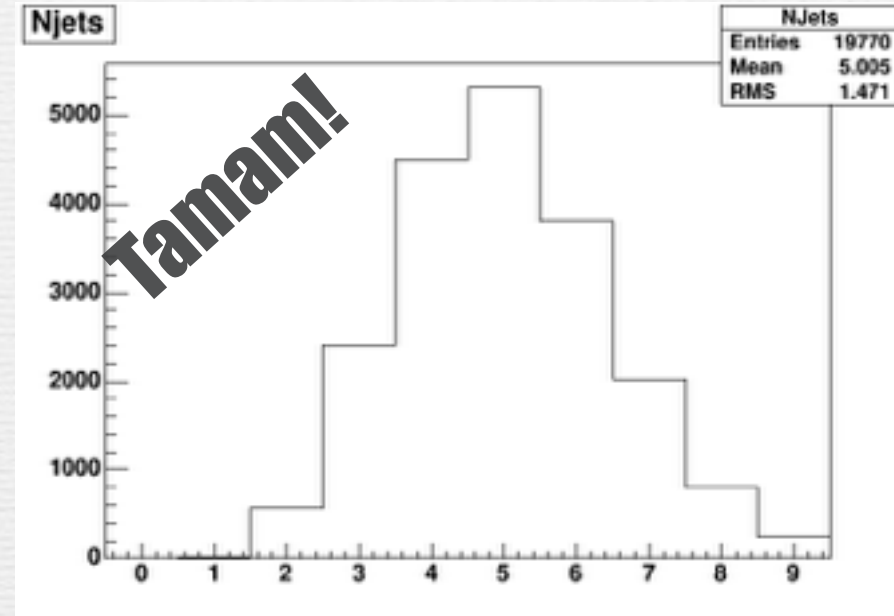
kesimlerin çetelesini tutmak lazım.

## Hangi kesim

Her kesimden sonra kalanları saklayalım: `TH1F *eff;`

```
eff=new TH1F("eff", "cuts", 16, 0.5, 16.5);  
eff->GetXaxis()->SetBinLabel(1,"all");  
eff->GetXaxis()->SetBinLabel(2,"2jets");  
eff->GetXaxis()->SetBinLabel(3,"MET");  
eff->GetXaxis()->SetBinLabel(4,"PTj");
```

```
root.exe bp.root  
root [1] LHC0->Process
```





# Çözümleme



```
NJets->Fill(Jet_);  
eff->Fill(0);  
if (Jet_ < 2) return kFALSE;  
eff->Fill(1);  
if ((Electron_ + Muon_) != 3) return kFALSE;  
eff->Fill(2);
```

2 jet  
tırpanından kurtulan  
olayları sayar.

```
bool bjet_found=false;  
for (k=0; k<Jet_; k++){  
    if (Jet_BTag[k]>0) bjet_found=true;  
}  
if (!bjet_found) return kFALSE;  
eff->Fill(3);
```

3 leptonlu olayları sayar

b-tag jetli olayları sayar

## Daha

### elektron, muon ve jet tanımlarını kullanalım.

1. hpfbu\_a.h içine eklenecek

```
#include <vector>  
#include "hpfParticle.cpp"  
#include "hpfMuon.h"  
#include "hpfElectron.h"  
#include "hpfJet.h"
```

2. particle tanımlarına

```
#include "TLorentzVector.h"  
#ifndef HPF_PARTICLE  
#define HPF_PARTICLE  
class hpfParticle {  
#include "hpfParticle.h"  
#ifndef HPF_ELECTRON  
#define HPF_ELECTRON
```

3. hpfbu\_a.C içine, process kısmına

```
vector <hpfElectron> eles;  
vector <hpfMuon>      muos;  
vector <hpfJet>       jets;  
int k;  
TLorentzVector alv;  
  
for (k=0; k<Electron_; k++){  
    alv.SetPtEtaPhiM(Electron_PT[k], Electron_Eta[k],  
Electron_Phi[k], 0.511/1E3); //in GeV  
    hpfElectron anel (alv);  
    anel.setCharge( Electron_Charge[k]);  
    eles.push_back(anel);  
}
```

# Çözümleme

• Ters yüklü leptonlardan Z ve W yapalım:

•  $2e\ 1\mu\ | \ 2\mu$

•  $3e\ 0\mu\ | \ 3\mu\ 0e \implies$  görece zor: yükler ters ve M

```
hpfParticle Zreco, theLepton;
TLorentzVector nue;
bool charge_problem=false;
switch(muos.size()) {
case 0: { if (abs(eles[0].q()+eles[1].q()+eles[2].q())>2) {charge_problem=true;}
        else { double Mdiff=999999.9;
              if (eles[0].q()*eles[1].q(<0){Zreco=eles[0]+eles[1];
                if (fabs(Zreco.lv().M()-91.2)<Mdiff) { Mdiff=fabs(Zreco.lv().M()-91.2); theLepton=eles[2];} }
              if (eles[0].q()*eles[2].q(<0){Zreco=eles[0]+eles[2];
                if (fabs(Zreco.lv().M()-91.2)<Mdiff) { Mdiff=fabs(Zreco.lv().M()-91.2); theLepton=eles[1];} }
              if (eles[1].q()*eles[2].q(<0){Zreco=eles[1]+eles[2];
                if (fabs(Zreco.lv().M()-91.2)<Mdiff) { Mdiff=fabs(Zreco.lv().M()-91.2); theLepton=eles[0];} }
              }
            break;}
case 3: { if (abs(muos[0].q()+muos[1].q()+muos[2].q())>2) {charge_problem=true;}
        else { double Mdiff=999999.9;
              if (muos[0].q()*muos[1].q(<0){Zreco=muos[0]+muos[1];
                if (fabs(Zreco.lv().M()-91.2)<Mdiff) { Mdiff=fabs(Zreco.lv().M()-91.2); theLepton=muos[2];} }
              if (muos[0].q()*muos[2].q(<0){Zreco=muos[0]+muos[2];
                if (fabs(Zreco.lv().M()-91.2)<Mdiff) { Mdiff=fabs(Zreco.lv().M()-91.2); theLepton=muos[1];} }
              if (muos[1].q()*muos[2].q(<0){Zreco=muos[1]+muos[2];
                if (fabs(Zreco.lv().M()-91.2)<Mdiff) { Mdiff=fabs(Zreco.lv().M()-91.2); theLepton=muos[0];} }
              }
            break;}
case 1: { if (eles[0].q()*eles[1].q(>0) {charge_problem=true;}
        else {Zreco=eles[0]+eles[1]; theLepton=muos[0]; }
        break;}
case 2: { if (muos[0].q()*muos[1].q(>0) {charge_problem=true;}
        else {Zreco=muos[0]+muos[1]; theLepton=eles[0]; }
        break;}
}
if (charge_problem) return kFALSE;
eff->Fill(4);
```



# parçacıklar

☞  $Z = e_1 + e_2$  yaz

```
public:
    hpfpParticle();
    hpfpParticle(TLorentzVector);
    hpfpParticle(TLorentzVector, int);
    ~hpfpParticle();
    hpfpParticle operator+ (hpfpParticle& );
```

.h

```
hpfpParticle hpfpParticle::operator+ (hpfpParticle& p)
{
    hpfpParticle sum;
    sum.setTlv(this->lv()+p.lv());
    sum.setCharge(this->q()+p.q());
    return sum;
}
```

.cpp

Z, W kütlerini bir de b' kütlesini ekleyelim

TH1F \*NJets, \*eff, \*mZ, \*mW, \*mbpw, \*mbpz;

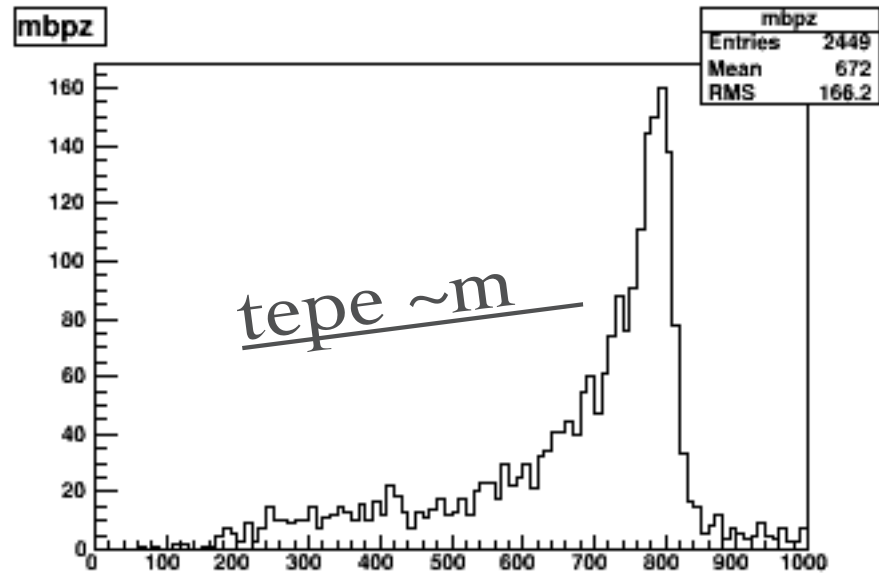
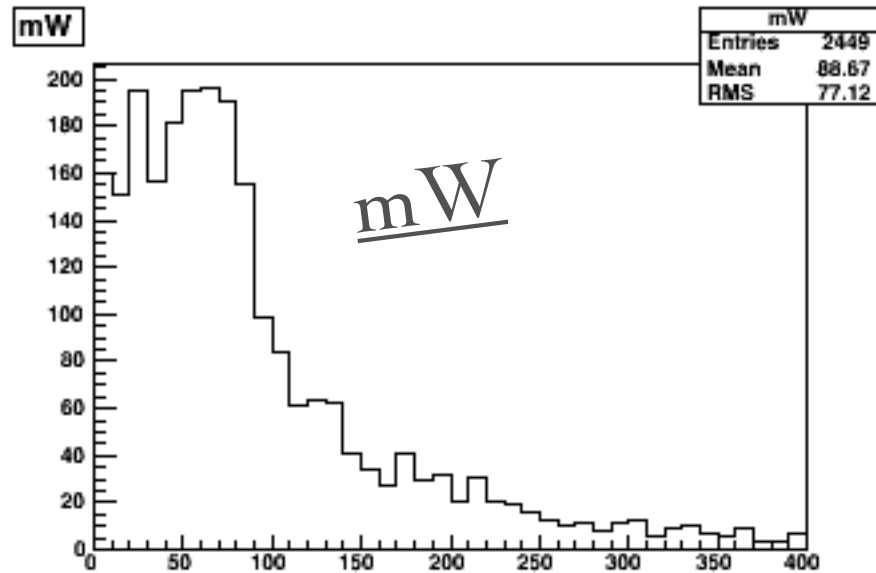
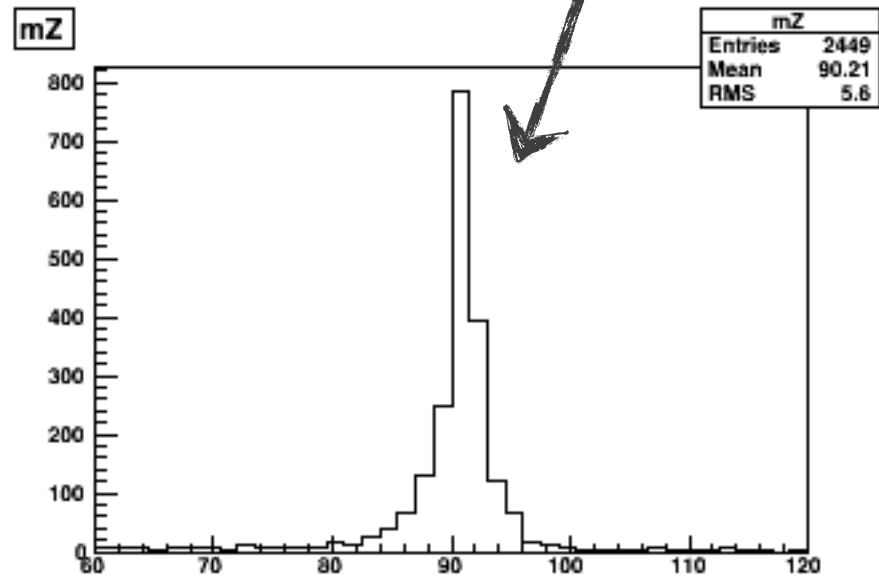
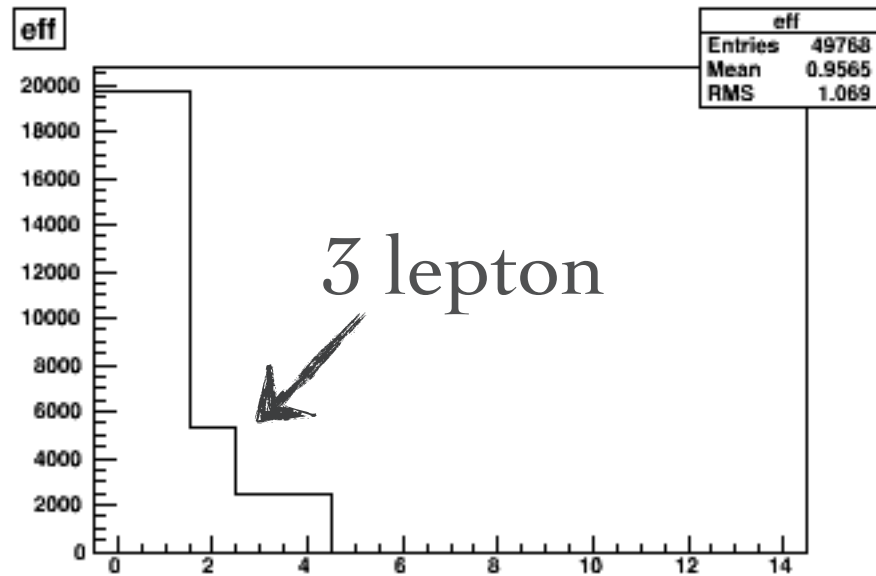
hpfbu\_a.C içine, process kısmına

```
mZ->Fill( Zreco.lv().M());
nue.SetPtEtaPhiM(MissingET_MET[0], theLepton.lv().Eta(), MissingET_Phi[0], 0.0);
hpfpParticle Wreco( (theLepton.lv()+nue) );
Wreco.setCharge(theLepton.q());
mW->Fill( Wreco.lv().M());
WvsMET->Fill( Wreco.lv().M(), MissingET_MET[0]);
```

```
hpfpParticle BPz;
for (k=0; k<jets.size(); k++){
    if (jets[k].bTag() ) {BPz=Zreco+jets[k]; break;}
}
mbpz->Fill( BPz.lv().M());
```

```
std::cout << "This is TERMINATE " << std::endl;
eff->Draw();
TFile tf("
NJets->Write();
eff->Write();
mZ->Write();
mW->Write();
mbpz->Write();
WvsMET->Write();
tf.Close();
```

# Çözümleme





# paralel al

---

```
void runProof(int lite=1){  
    if (lite) {  
        TProof *proof = TProof::Open("");  
    } else {  
        TProof *proof = TProof::Open("pc-atb-srv-01");  
    }  
  
    proof->SetParallel(1); //using only few cores,the rest for other tasks  
  
    TDataSet *set;  
    set = new TDataSet("TTree","LHC0");  
    if (lite) {  
        set->Add("bp.root");  
        // set->Add("tag_1_delphes_events.root");  
    }else{  
        set->Add("root://10.0.1.2//project/analyses/lvl0-1.root");  
        set->Add("root://10.0.1.2//project/analyses/lvl0-2.root");  
    }  
  
    set->Process("hpfbu_a.C");  
}
```

root.exe runProof.C

# Ardalan=

---

- Aradığımız ize benzer iz bırakan
- Ama aradığımız nesneyi
  - bunlardan kurtulmak lazım
- BSM araştırmalarında SM gürültüdür
- Samanı sap'da bulmak gerekli
  - “CUT” = “Tırpan”
- Hangi özellikler kullanılabilir?
  - $b'$  ağır olduğu için bundan çıkan jetler de yüksek  $p$  olması beklenir
  - $M_{b'}$  \_\_\_\_\_



# Çözümleme

Tanımlar

```
void compare() {  
float si_cr=18; //fbarn  
float bg_cr=9500; //fbarn  
float lumi=10; //fbarn-1  
float year=1; // time in years  
TCanvas *ctst = new TCanvas("a1","ctst",960,400);  
  
TFile bg("bp_bg.root");  
TH1F *bgmbp = ((TH1*)bg.Get("mbpz"))->Clone();  
TH1F *bgeff = ((TH1*)bg.Get("eff"))->Clone();  
bgmbp->Scale( (lumi*year*bg_cr)/ bgeff->GetBinContent(1) );  
bgmbp->SetLineColor(2);  
bgmbp->Rebin(2);  
  
TFile si("bp_si.root");  
TH1F *simbp = ((TH1*)si.Get("mbpz"))->Clone();  
TH1F *sieff = ((TH1*)si.Get("eff"))->Clone();  
simbp->Scale( (lumi*year*si_cr)/ sieff->GetBinContent(1) );  
simbp->SetLineColor(4);  
simbp->SetLineStyle(2);  
simbp->Rebin(2);  
  
TH1F *totalbp = new TH1F(*bgmbp);  
totalbp->Add(simbp, bgmbp);  
char aaa[128];  
sprintf (aaa,"#events/%iGeV/%ifb^{-1}",20,lumi);  
totalbp->SetYTitle(aaa);  
totalbp->SetXTitle("M_{Zb} (GeV)");  
totalbp->SetLineColor(1);  
totalbp->SetLineWidth(2.5);  
  
ctst->cd();  
totalbp->Draw("E"); bgmbp->Draw("same"); simbp->Draw("same");  
ctst->SaveAs("final.png");  
}
```

kar

Elimizdeki veri miktarı ve tesir kesitleri belli.

$$N = \mathcal{L} \times \sigma \times \epsilon$$

Işınlık

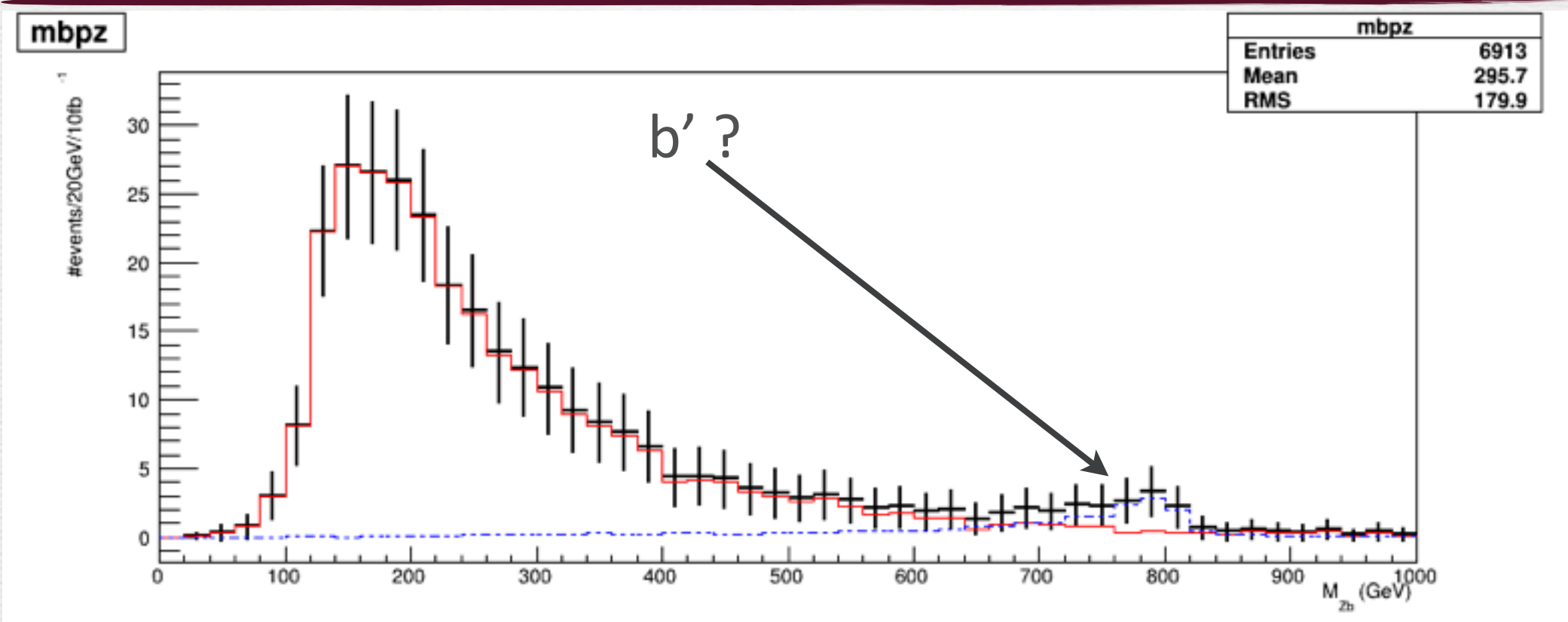
Tesir kesiti

Verim

senelik olay hesabı

toplam olay sayısı

si + sm çizimi



☞ Anlamalı mı yoksa istatistik hata mı?

☞ Ölçütler ( $\sigma$ )

☞  $3\sigma \Rightarrow$  Gözlem,

$$\frac{S}{\sqrt{B}}$$

$$\frac{S}{\sqrt{S+B}}$$

$$2 \times (\sqrt{S+B} - \sqrt{B})$$

$$\sqrt{2 \times \left[ (s+b) \ln \left( 1 + \frac{s}{b} \right) - s \right]}$$

☞ Toplam olay eğrisine poly1+Gauss uydurup sinyal ve ardalın olay sayısını bulalım.



E

biliyoruz...

```
TF1 *total = new TF1("total", "pol1(0)+gaus(2)", 400, 920);
total->SetParameters(15, -0.5, 5, 800, 35);
total->SetFillColor(4);
total->SetLineWidth(2);
total->SetParLimits(2, 0, 10000);
total->SetParLimits(3, 700, 800);
total->SetParLimits(4, 10, 90);
```

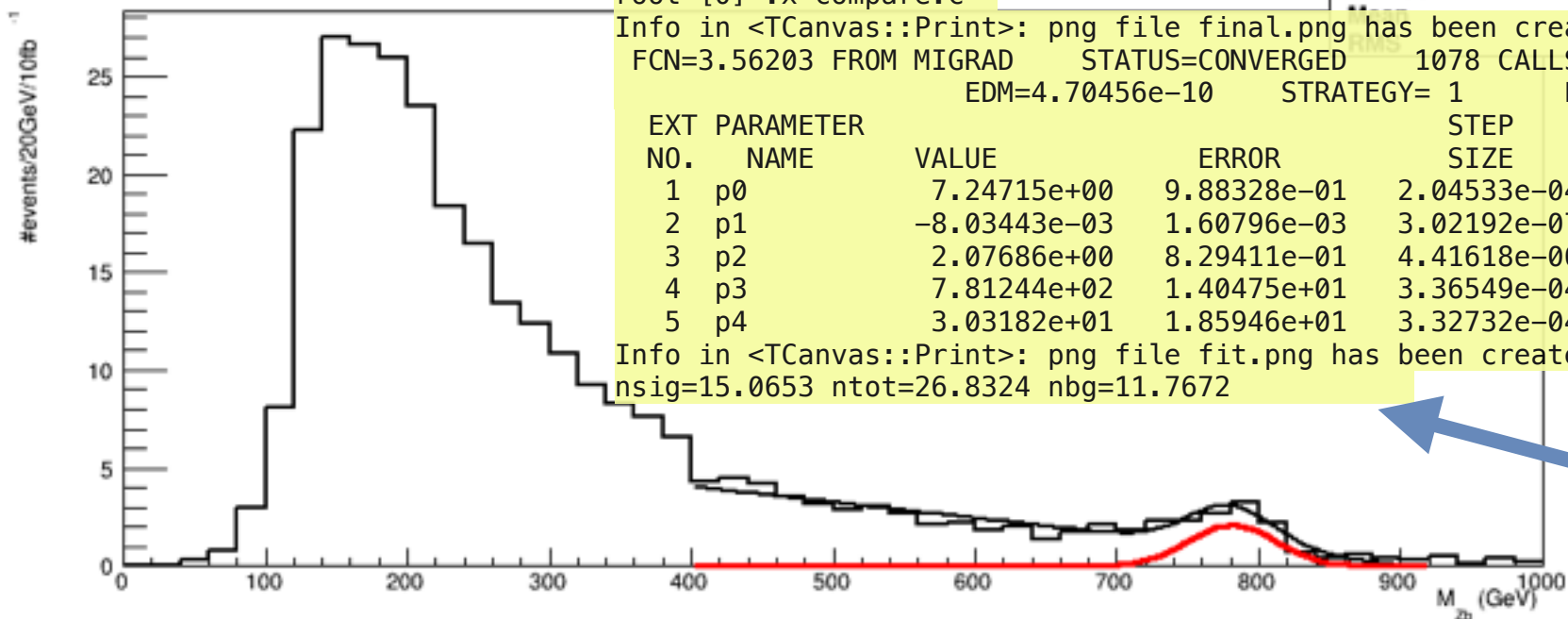
```
totalbp->Fit(total, "WR+");
```

```
Double_t par[5];
total->GetParameters(&par[0]);
```

```
TF1 *f2 = new TF1("f2", "gaus", 400, 920);
f2->SetParameters(&par[2]);
f2->SetLineColor(2);
f2->Draw("same");
ctst->SaveAs("fit.png");
```

```
Int_t ns=2; // #sigmas
Double_t nsig=(f2->Integral(par[3]-ns*par[4], par[3]+ns*par[4]))/10;
Double_t ntot=(total->Integral(par[3]-ns*par[4], par[3]+ns*par[4]))/10;
cout << "nsig=" << nsig << " ntot=" << ntot << " nbg=" << (ntot-nsig);
cout << " SIG=" << sqrt(2*(ntot*log(1+nsig/(ntot-nsig))-nsig)) << endl;
```

mbpz



```
root [0] .x compare.C
```

```
Info in <TCanvas::Print>: png file final.png has been created
```

```
FCN=3.56203 FROM MIGRAD STATUS=CONVERGED 1078 CALLS 1079 TOTAL
EDM=4.70456e-10 STRATEGY= 1 ERROR MATRIX ACCURATE
```

EXT NO.	PARAMETER NAME	VALUE	ERROR	STEP SIZE	FIRST DERIVATIVE
1	p0	7.24715e+00	9.88328e-01	2.04533e-04	-3.75394e-05
2	p1	-8.03443e-03	1.60796e-03	3.02192e-07	-1.17716e-02
3	p2	2.07686e+00	8.29411e-01	4.41618e-06	1.86318e-03
4	p3	7.81244e+02	1.40475e+01	3.36549e-04	5.76554e-05
5	p4	3.03182e+01	1.85946e+01	3.32732e-04	6.11580e-05

```
Info in <TCanvas::Print>: png file fit.png has been created
```

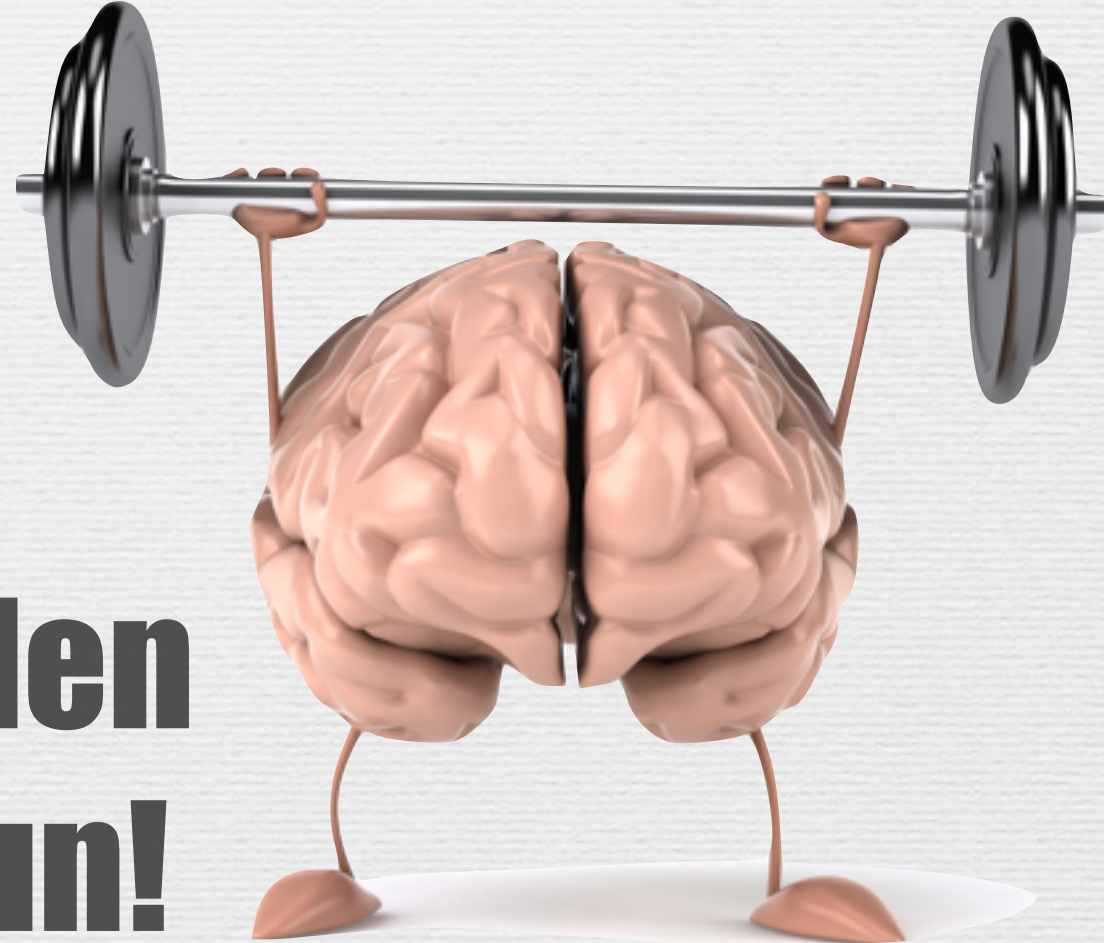
```
nsig=15.0653 ntot=26.8324 nbg=11.7672
```

mbpz	
Entries	6913
...	...
...	...

# Sonuç

• Bu derste öğrendikleriniz PhD ve hatta postdoc çalışmalarınızda bile işinize yarayacak.

• çalışın ve bunları kullanın ki aklınızda kalsın.



**Bahanelerinizden  
Daha Güçlü Olun!**