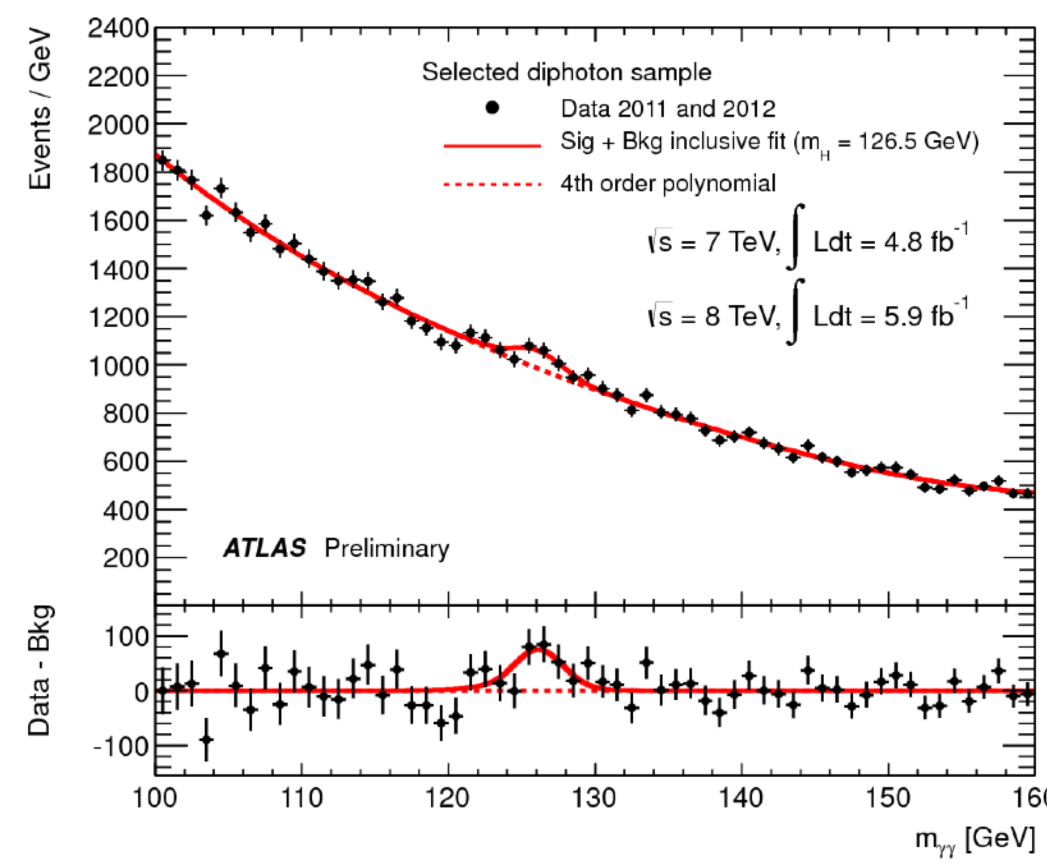
CutLang

an "interpreted" analysis description language

G. Unel / UCI
S. Sekmen / KNU

Veri Çözümlemesi



“ Eski yol ”

```
 // if (Cut(ientry) < 0) continue;
eff->Fill(1);
jmult->Fill(Jet_);
lmult->Fill(Muon_ + Electron_);
for ( int i=0; i<Jet_; i++) {
  jets[i].SetPtEtaPhiM (Jet_PT[i], Jet_Eta[i], Jet_Phi[i], Jet_Mass[i
  jeteta->Fill (jets[i].Eta() );
  jetphi->Fill (jets[i].Phi() );
  jetPT->Fill (jets[i].Pt() );
}
if ( Jet_ != 2) continue;
eff->Fill(2);
MJJ=jets[0]+jets[1];
jjmass->Fill( MJJ.M() );

MET->Fill(MissingET_MET[0]);
if ( MissingET_MET[0] <20 ) continue;
eff->Fill(3);
```

sorunlar….

sorunlar

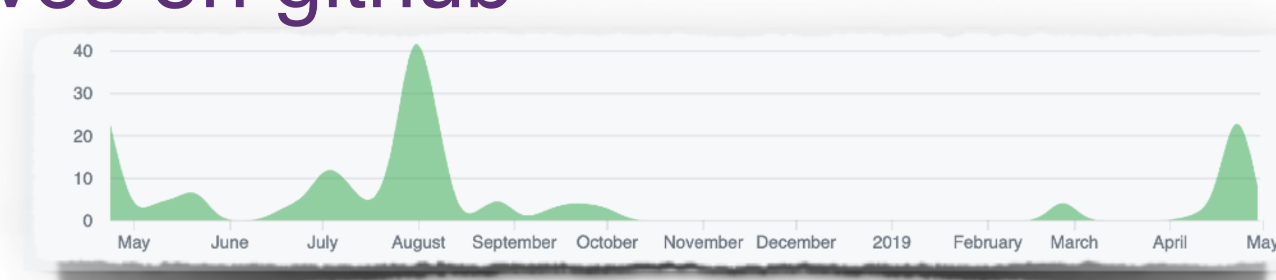
- C++ / python vs
  - öğrenme zorluğu, yenilere kapalı kutu, dediğini gerçekten yapıyor mu?
  - tekrar tekrar aynı döngüler, aynı seçimler, aynı histogramlar…
- Framework ile ilgili
  - fizik analiz algoritması, kodu ve alt yapısı (framework) iç içe geçmiş durumda
- Karşılaştırma
  - tam olarak ne yapılıyor? gruplar-arası karşılaştırma, deneyler arası!?!
  - deney-kuram karşılaştırma, phenomenology - olaybilim
- Tekrarlama
  - 2011'de yapılan bir analizi tekrar etmek = kabus!

Introducing CutLang v2

- Analysis description language (ADL) and runtime interpreter
  - Human readable **text file** to describe the whole analysis
  - Run time interpretation of the ADL file: *No compiling!*
    - ADL: [initializations] [definitions] [objects] [definitions] commands
- Works with multiple input data formats
  - Currently available data formats: LVL0, ATLAS OpenData, CMS OpenData, Delphes, LHCO, FCC, CMSNANOAOD,….
    - more can be easily added…

CutLang implementation

- M*odest* requirements:
  - Pure C++ classes, on top of ROOT LorentzVectors and histograms
  - Linux or Mac, C++ (gcc4.x)
    - ROOT6
    - yacc & lexx (**NEW**)
- Additional tools to help the analyst and the advisor
  - All definitions, cuts and object selections are saved into the output ROOT file
  - Shell & Python scripts for plotting & addition of "user functions" being updated
- The project is opensource and lives on github
  - https://github.com/unelg/CutLang.git



Syntax 1

- The execution order is top to bottom.
  - units are in GeV, comment character is #, mostly case insensitive
- Most mathematical functions are available
  - sin(), sinh(), cos(), cosh(), tan(), tanh(), Hstep(), abs(), sqrt(), ^, *, /, +, -, interval inclusion [] and exclusion][
- Predefined concepts
  - particles are: **ELE**CTRON, **MUO**N, **TAU**, **PHO**TON, **JET**, **F**AT**JET**, **MET**
    - particles are already sorted in decreasing transverse momentum order
  - particle attributes and functions are: charge **q** mass **m**, energy **E**, transverse momentum **pT**, total momentum **P**, pseudorapidity **Eta**, angular distances **dPhi**,…

particle notation

- On the blackboard, we write

jet_1

  - When you type it in latex it is jet_1
  - CL understands *particleName_index* notation:

Highest Pt object	Second Highest Pt object
ELE_0	ELE_1
MUO_0	MUO_1

- On the computer, we write

```
jet[3]
```

  - CL understands *particleName*[*index*] notation:

```
muonsVeto[0]
photons[0]
```

functions & attributes

- Is pseudo rapidity or transverse momentum a property of a particle? of the addition of many particles? is it an attribute? is it a function?
- DO I CARE? no.
  - I only care about the result of my analysis
- However, when I speak or write I might say either of
  - "the mass of a particle set" m ( )
  - "the particle set's mass" { }m ← more natural in Turkish
- CL understands both notations

Meaning	Operator	Operator
Mass of	`m( )`	`{ }m`
Charge of	`q( )`	`{ }q`
Phi of	`Phi( )`	`{ }Phi`
Eta of	`Eta( )`	`{ }Eta`
Absolute value of Eta of	`AbsEta( )`	`{ }AbsEta`
Pt of	`Pt( )`	`{ }Pt`

Syntax 2

- Main keywords:
  - use **select** / **reject** (or **cmd**) to select/reject events
  - use **define** (or **def**) to define constants, functions and composite particles
  - use **histo** to book and fill histograms
  - use **region** (or **algo**) to define independent algorithms
  - use **object** (or **obj**) to define new/composite particle objects
  - use **sort** to sort particles according to a property
  - use **table** to define a table (currently 1D only)
  - use **weight** to define an event weight
  - use **save** to record surviving events
  - use **Union** to define a new set of particles
  - use **Comb** to construct probability combinatorics

A very simple example

- reconstruct Z boson candidate from the first two electrons $Z \to \ell\ell \quad \ell = e, \mu$
- histogram the mass of the candidate

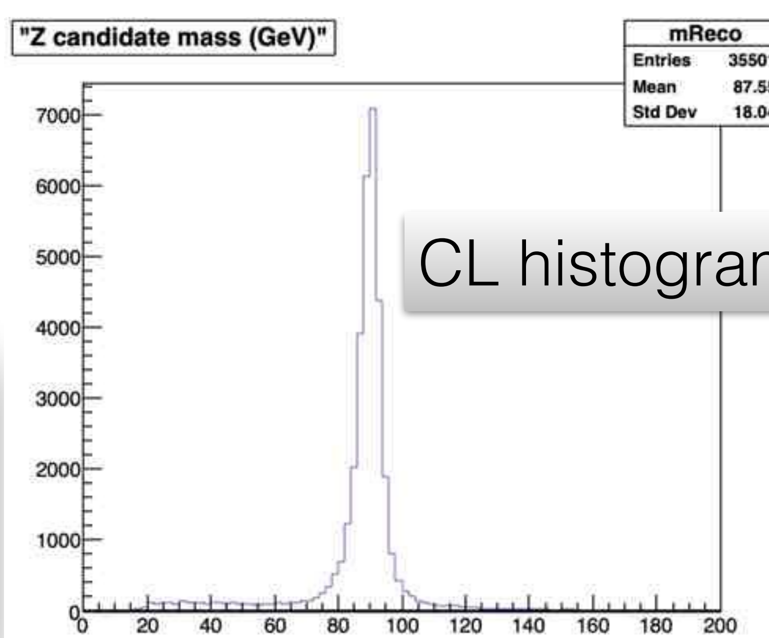
user's ADL file

```
region   test
  select      ALL                # to count all events
  select      Size (ELE) >= 2  # events with 2 or more electrons
  histo       mReco, "Z candidate mass (GeV)", 100, 0, 200, {ELE_0 ELE_1}m
```

CL output

```
            test    Based on 125000 events:
                           ALL :        1 +-         0 evt:    125000
               Size (ELE) >= 2 :   0.284 +-   0.00128 evt:     35501
[Histo] Z candidate mass (GeV) :        1 +-         0 evt:     35501
     --> Overall efficiency  =   28.4 % +-  0.128 %
```

CL histogram



A very simple example

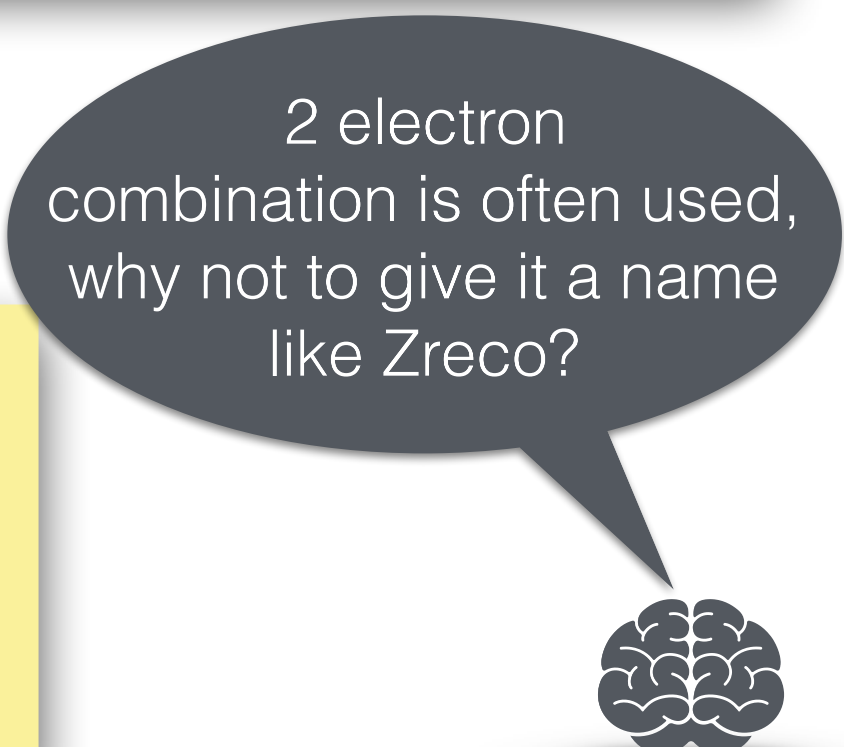
- Additional constraint $Z \rightarrow \ell\ell \quad \ell = e, \mu$
  - the Z candidate should be neutral (q=0)

user's ADL file

```
region   test
  select       ALL                 # to count all events
  select       Size (ELE) >= 2  # events with 2 or more electrons
  histo        h1mReco, "Z candidate mass (GeV)", 100, 0, 200, {ELE_0 ELE_1}m
  select       {ELE[0] ELE[1] }q == 0    # Z is neutral
  histo        h2mReco, "Z candidate mass (GeV)", 100, 0, 200, {ELE_0 ELE_1}m
```

CL output

```
          test    Based on 125000 events:
                              ALL :        1 +-        0 evt:   125000
                   Size (ELE) >= 2 :   0.284 +-  0.00128 evt:    35501
[Histo] Z candidate mass (GeV) :        1 +-        0 evt:    35501
      {ELE[0] ELE[1] }q == 0 : 0.9595 +-  0.00105 evt:    34063
[Histo] Z candidate mass (GeV) :        1 +-        0 evt:    34063
     --> Overall efficiency  =   27.3 % +-  0.126 %
```



A very simple example

- introducing definitions

$Z \to \ell\ell \quad \ell = e, \mu$

user's ADL file

```
define Zreco : ELE[0] ELE[1]

region   test
  select       ALL                  # to count all events
  select       Size (ELE) >= 2  # events with 2 or more electrons
  histo        h1mReco, "Z candidate mass (GeV)", 100, 0, 200, {Zreco}m
  select       {Zreco}q == 0    # Z is neutral
  histo        h2mReco, "Z candidate mass (GeV)", 100, 0, 200, m(Zreco)
```

CL output

```
            test    Based on 125000 events:
                           ALL :      1 +-         0 evt:   125000
                Size (ELE) >= 2 :  0.284 +-   0.00128 evt:    35501
[Histo] Z candidate mass (GeV) :      1 +-         0 evt:    35501
                  {Zreco}q == 0 : 0.9595 +-   0.00105 evt:    34063
[Histo] Z candidate mass (GeV) :      1 +-         0 evt:    34063
     --> Overall efficiency  =   27.3 % +-  0.126 %
```

Are these electrons inside the inner tracker?

A simple example

- introducing derived objects

$$Z \to \ell\ell \quad \ell = e, \mu$$

```
define Zreco : ELE[0] ELE[1]

object goodEle : ELE
  select  Pt(ELE_)        >   10
  select abs({ELE_}Eta)   <   2.4
  select     {ELE_}AbsEta ][  1.442 1.556

define goodZreco : goodEle[0] goodEle[1]

region   test
  select      ALL              # to count all events
  select      Size(ELE)     >= 2  # events with 2 or more electrons
  select      Size(goodEle) >= 2  # events with 2 or more electrons
  histo       h1mReco,     "Z candidate mass (GeV)", 100, 0, 200, {Zreco}m
  histo       h1mgoodReco, "Z candidate mass (GeV)", 100, 0, 200, {goodZreco}m
  select      {Zreco}q == 0       # Z is neutral
  select      {goodZreco}q == 0   # Z is neutral
  histo       h2mReco     , "Z candidate mass (GeV)", 100, 0, 200, m(Zreco)
  histo       h2mgoodReco, "Z candidate mass (GeV)", 100, 0, 200, m(goodZreco)
```

Derived objects

- Further cleaning or refining can be achieved using derived objects
  - Derived objects can be used to derive further refined objects
    - JETS —> goodJETs —> cleanJETs —> verycleanJets …
  - Multiple selection criteria can be applied
  - If all members of a particular class (e.g. jets) are considered, a _ sign might be used
  - The criteria selection line can contain at most 2 different type of objects (e.g. j & p)
  - The whole criteria returns a boolean for the considered pair (j_i and p_j)

```
# jets - no photon
object AK4jetsNOpho : AK4jets
  select dR(AK4jets_, photons_ ) >=0.4 OR {photons_}Pt/{AK4jets_}Pt ][ 0.5 2.0
```

- Analysis algorithms can use the original objects or derived objects

A simple example

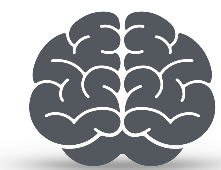
- introducing derived objects

$Z \to \ell\ell \quad \ell = e, \mu$

  - do not use “reject” in object selection

there is a bug in this version

```
CLA v02.02.02   compiled on Sun Feb  2 21:46:31 CET 2020
```



this is not fair, Zreco’s charge should not impact goodZreco selection.

```
define Zreco : ELE[0] ELE[1]

object goodEle : ELE
  select Pt(ELE_)        >   10
  select   {ELE_}AbsEta <   2.4
  select   {ELE_}AbsEta ][  1.442 1.556

define goodZreco : goodEle[0] goodEle[1]

region   test
  select      ALL                # to count all events
  select      Size(ELE)     >= 2  # events with 2 or more electrons
  select      Size(goodEle) >= 2  # events with 2 or more electrons
  histo       h1mReco,     "Z candidate mass (GeV)", 100, 0, 200, {Zreco}m
  histo       h1mgoodReco, "Z candidate mass (GeV)", 100, 0, 200, {goodZreco}m
  select      {Zreco}q == 0         # Z is neutral
  select      {goodZreco}q == 0   # Z is neutral
  histo       h2mReco     , "Z candidate mass (GeV)", 100, 0, 200, m(Zreco)
  histo       h2mgoodReco, "Z candidate mass (GeV)", 100, 0, 200, m(goodZreco)
```

weights

- weights are needed for MC processes
  - simulate the relative importance of certain events
  - simulate the efficiencies (trigger, pileup, vertex, others…)

- Two possbilities
  - via a simple coefficient
  - via a table

```
weight   randWeight    1.123
weight   effWeight effTable( {ELE_0}pT )   # new

histo    h1ept, "E0 pt (GeV)", 100, 0, 2000, {ELE_0}pT
```

```
table    effTable
#        value    min     max
           0.1    0.0    10.0
           0.2   10.0    20.0
           0.4   20.0    50.0
           0.7   50.0    70.0
           0.95  70.0    1000.0



region   test
  select   ALL                        # to
```

A simple example

- introducing multiple regions or algorithms

$Z \to \ell\ell \quad \ell = e, \mu$

```
define Zreco : ELE[0] ELE[1]

object goodEle : ELE
  select Pt(ELE_)        >   10
  select   {ELE_}AbsEta <   2.4
  select   {ELE_}AbsEta ][  1.442 1.556

define goodZreco : goodEle[0] goodEle[1]

algo        preselection
  select      ALL              # to count all events
  select      Size(ELE)     >= 2  # events with 2 or more electrons

algo        testA
  preselection
#  histo       h1mReco,      "Z candidate mass (GeV)", 100, 0, 200, {Zreco}m
  select      {Zreco}q == 0       # Z is neutral
  histo       h2mReco    , "Z candidate mass (GeV)", 100, 0, 200, m(Zreco)

algo        testB
  preselection
  select      Size(goodEle) >= 2  # events with 2 or more electrons
#  histo       h1mgoodReco, "Z candidate mass (GeV)", 100, 0, 200, {goodZreco}m
  select      {goodZreco}q == 0   # Z is neutral
  histo       h2mgoodReco, "Z candidate mass (GeV)", 100, 0, 200, m(goodZreco)
```

Running with multiple regions

- A user defined region can contain another one
  - e.g. SignalRegion containing preselection

- All regions are processed in parallel and saved as TDirectories in the output ROOT file

```
TFile**         histoOut-ex5.root
 TFile*         histoOut-ex5.root
  KEY: TDirectoryFile    preselection;1  preselect
  KEY: TDirectoryFile    testA;1 testA
  KEY: TDirectoryFile    testB;1 testB
```

```
root [2] testA->cd()
(bool) true
root [3] .ls
TDirectoryFile*         testA   testA
 KEY: TText     CLA2cuts;1
  select       ALL
  select       Size(ELE)     >= 2
  select       {Zreco}q == 0
  histo        h2mReco    , "Z candidate mass (GeV)", 100, 0, 200, m(Zreco)
  select       ALL

 KEY: TText     CLA2defs;1
define Zreco : ELE[0] ELE[1]
define goodZreco : goodEle[0] goodEle[1]

 KEY: TText     CLA2Objs;1
object goodEle : ELE
  select    Pt(ELE_)       >   10
  select abs({ELE_}Eta )  <   2.4
  select      {ELE_}AbsEta ][  1.442 1.556

 KEY: TH1F      eff;1   selection efficiencies
 KEY: TNtuple   rntuple;1       run info
 KEY: TH1D      h2mReco;1       "Z candidate mass (GeV)"
```

An example $Z \to \ell\ell \quad \ell = e, \mu$

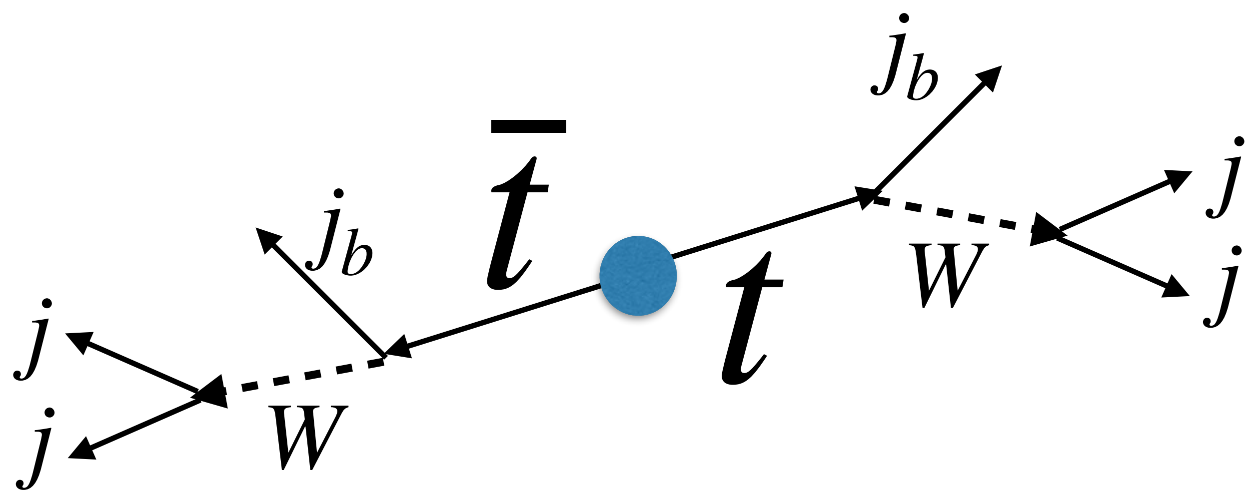
- Introducing optimizers
  - if there are more than 2 electrons, search all possible combinations to find the “best” candidate

```
define Zreco : ELE[0] ELE[1]

object goodEle : ELE
  select Pt(ELE_)       >   10
  select   {ELE_}AbsEta <   2.4
  select   {ELE_}AbsEta ][  1.442 1.556

define goodZreco : goodEle[-1] goodEle[-1]

algo BestZ
  select      ALL                  # to count all events
  select      Size(goodEle) >= 2   # events with 2 or more electrons
  select      {goodZreco}m ~= 91.2 # find the pair yielding mass closest to Z
  select      {goodZreco}q == 0    # Z is neutral
  histo       hZRecoB, "Z candidate mass (GeV)", 100, 0, 200, m(goodZreco)
```

We use negative indices if they are to be determined at run time, using a criterion, such as: **~=**

Repeating the same negative value helps speeding up since $e_i\, e_j = e_j\, e_i$

$t\bar{t}$ Reconstruction example



$$t \rightarrow Wb \rightarrow jjj_b$$

There are 6 jets in the event of which 2 can be b-tagged

+ LOTS of *other jets* from spectator quarks and QCD effects

Which one is which?

with the χ^2 defined as:

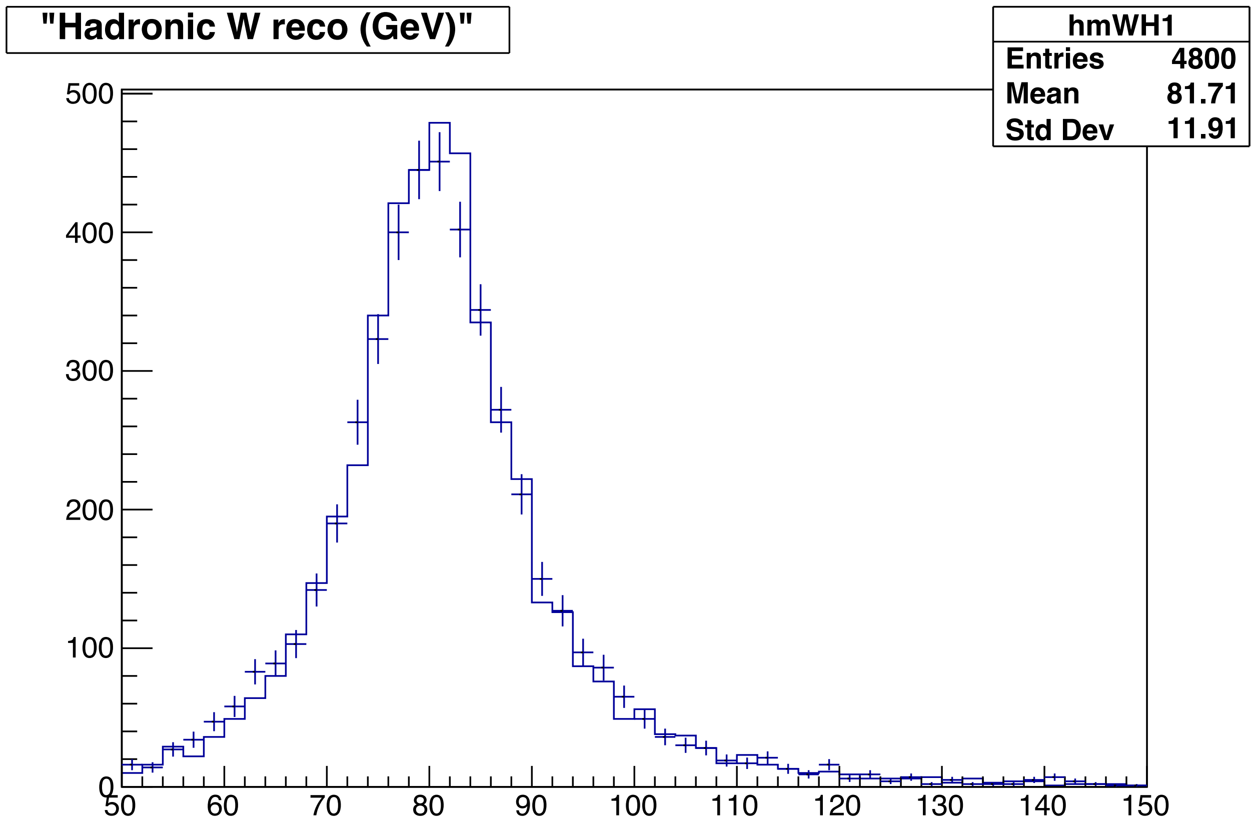
$$\chi^2 = \frac{(m_{b_1 j_1 j_2} - m_{b_2 j_3 j_4})^2}{\sigma^2_{\Delta m_{bjj}}} + \frac{(m_{j_1 j_2} - m_W^{MC})^2}{\sigma^2_{m_W^{MC}}} + \frac{(m_{j_3 j_4} - m_W^{MC})^2}{\sigma^2_{m_W^{MC}}}.$$

$t\bar{t}$ Reconstruction example

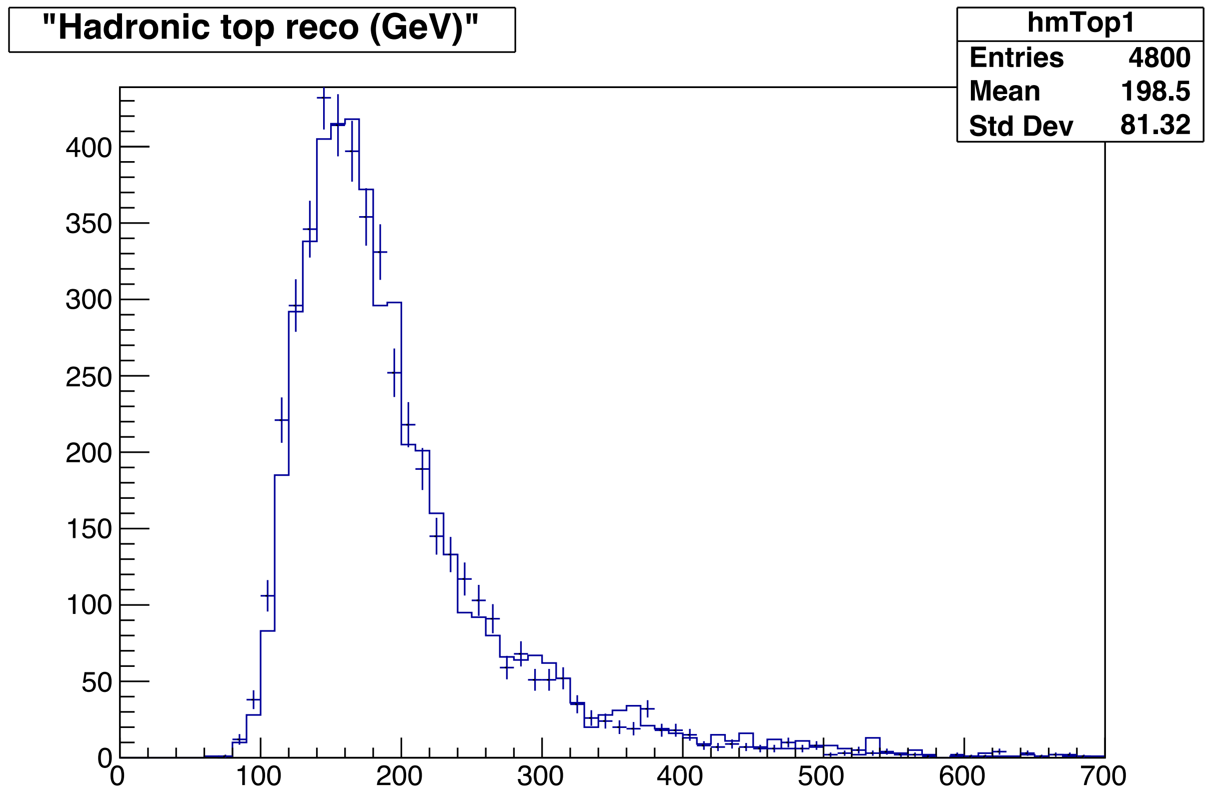
```
define    WH1  : JET[-1] JET[-1]
define    WH2  : JET[-3] JET[-3]
###       chi2 for W finder
define    Wchi2 : (({WH1}m - 80.4)/2.1)^2 + (({WH2}m - 80.4)/2.1)^2

## top quarks without b tagging
define    Top1 : WH1 JET[-2]
define    Top2 : WH1 JET[-4]
define    mTop1 : m(Top1)
define    mTop2 : m(Top2)
###       chi2 for top finder
define    topchi2 : ((mTop1 - mTop2)/4.2)^2

algo besttop
  select    ALL                     # to count all events
  select    Size(JET) >= 6          # at least 6 jets
  select    MET < 100               # no large MET
  select    Wchi2 + topchi2 ~= 0    # find the tops and ws
  histo     hmWH1 , "Hadronic W reco (GeV)", 50, 50, 150, m(WH1)
  histo     hmWH2 , "Hadronic W reco (GeV)", 50, 50, 150, m(WH2)
  histo     hmTop1 , "Hadronic top reco (GeV)", 70, 0, 700, mTop1
  histo     hmTop2 , "Hadronic top reco (GeV)", 70, 0, 700, mTop2
```

, with the χ^2 defined as:

$$\chi^2 = \frac{(m_{b_1 j_1 j_2} - m_{b_2 j_3 j_4})^2}{\sigma^2_{\Delta m_{bjj}}} + \frac{(m_{j_1 j_2} - m_W^{\rm MC})^2}{\sigma^2_{m_W^{\rm MC}}} + \frac{(m_{j_3 j_4} - m_W^{\rm MC})^2}{\sigma^2_{m_W^{\rm MC}}}.$$



reconstructed W bosons



reconstructed top quarks

reference guide

- The Objects

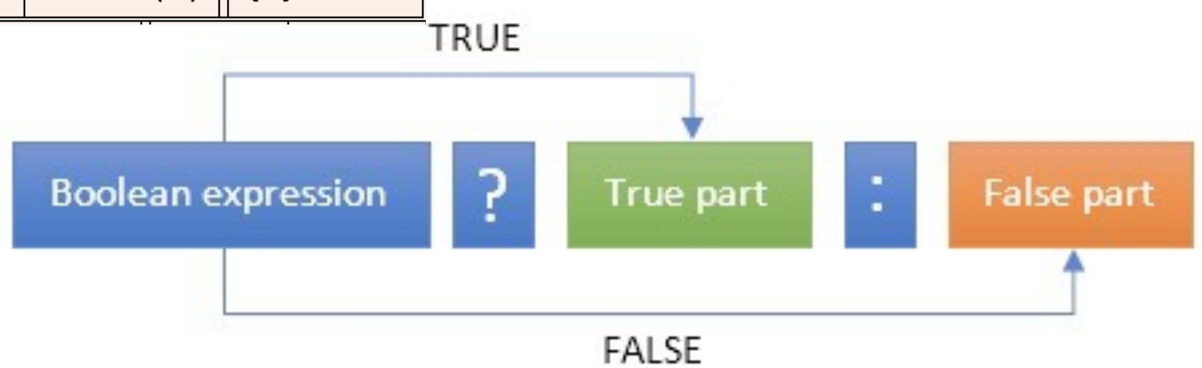
Name	Keyword	Highest Pt object	Second Highest Pt object	$j+1^{th}$Highest Pt object
Electron	ELE	ELE_0	ELE_1	ELE_j
Muon	MUO	MUO_0	MUO_1	MUO_j
Tau	TAU	TAU_0	TAU_1	TAU_j
Lepton	LEP	LEP_0	LEP_1	LEP_j
Photon	PHO	PHO_0	PHO_1	PHO_j
Jet	JET	JET_0	JET_1	JET_j
Fat Jet	FJET	FJET_0	FJET_1	FJET_j
b-tagged Jet	BJET	BJET_0	BJET_1	BJET_j
light Jet	QGJET	QGJET_0	QGJET_1	QGJET_j
neutrino	METV	METV_0	METV_1	METV_j

- Functions

Meaning	Operator
number of	Size( )
tangent	tan()
sinus	sin()
cosinus	cos()
absolute value	abs()
square root	sqrt()
in the interval	[ ]
not in the interval	] [
as close as possible	~=
as far away as possible	!=
usual meaning	+-/*
to the power	^

Meaning	Operator	Operator
Mass of	m( )	{ }m
Charge of	q( )	{ }q
Phi of	Phi( )	{ }Phi
Eta of	Eta( )	{ }Eta
Absolute value of Eta of	AbsEta( )	{ }AbsEta
Pt of	Pt( )	{ }Pt
Pz of	Pz( )	{ }Pz
Energy of	E( )	{ }E
Momentum of	P( )	{ }P
Angular distance between	dR( )	{ }dR
Phi difference between	dPhi( )	{ }dPhi
Eta difference between	dEta( )	{ }dEta

- The ternary function in C notation



User (external) functions

- User defined selection functions are somewhat difficult to incorporate into an interpreter

- Currently we define a user function type and compile it in.
  - CLv2 will provide the means to do this automatically
  - Currently Razor functions are pre-integrated:

```
std::vector<TLorentzVector> fmegajets(std::vector<TLorentzVector> myjets);
double fMR(std::vector<TLorentzVector> j);
double fMTR(std::vector<TLorentzVector> j, TVector2 amet);
double fMTR2(std::vector<TLorentzVector> j, TLorentzVector amet);
```

- Simple functions can be interpreted using CL math functions

```
return sqrt( 2 * lepton.Pt() * pfmet.Pt() * ( 1 - cos( pfmet.Phi() - lepton.Phi() )));
```

```
define MTe : sqrt( 2*{electronsVeto_0}Pt *MET*(1-cos( {METLV_0}Phi - {electronsVeto_0}Phi )))
define MTm : sqrt( 2*{muonsVeto_0}Pt *MET*(1-cos( {METLV_0}Phi - {muonsVeto_0}Phi )))
define mZ : 91.187
```

Combinations

Lets assume we have 5 jets 1 2 3 4 5

we can make 2 hadronic Zs

12 34
12 35
12 45
13 24
13 25
13 45
……

CutLang code to **define** all possibilities, with some cuts:

```
object hZs : COMB( jets[-1] jets[-2] ) alias ahz
     select { ahz }AbsEta < 3.0
     select {jets[-2] }Pt > 2.1
     select {jets[-1] }Pt > 5.1
     select {jets[-1], ahz }dR < 0.6   #--- means a member of hZs a
     select { ahz }m [] 10 200           # does get the paricle num
```

CutLang code to **use** those cuts:

```
region testA
select Count(hZs)  >= 2
```

12 ~~34~~
12 35
12 ~~45~~
13 24
13 25
13 ~~45~~
……

Some combinations are removed because of the selection cuts above.
Lines with 1 remaining Zh are removed since we required at least 2 hadronic Zs

But which combination to use?

12 35
13 24
13 25
……

```
define zham : {hZs[-1]}m
define zhbm : {hZs[-2]}m
define chi2 : (zham - 91.2)^2 + (zhbm - 91.2)^2
….
select chi2  ~= 0
```

Union

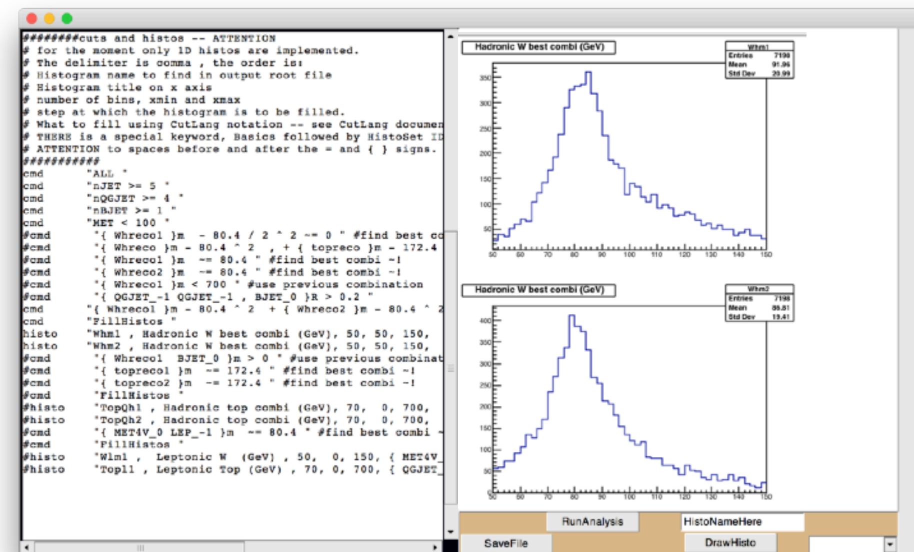
- It is possible to group charged leptons, or derived objects from charged leptons
  - needed in some susy analyses
  - united object requires to be knows
    - check other

```
object goodEle : ELE
  select  Pt(ELE)        >   10
  select abs({ELE}Eta)   <   2.4
  select     {ELE}AbsEta ][  1.442 1.556

object GMUO : MUO
  select  Pt(MUO)        >   10
  select abs({MUO}Eta)   <   2.4


object geps  : Union( MUO , ELE, TAU)            #add all leptons into
object gleps : Union( goodEle , GMUO )           #add all good electron

define Zreco = ELE[-1] ELE[-1]                   #negative indices are to

region   test
  select      ALL                 # to count all events
  select      Size (goodEle)  >= 1  # events with 2 or more electrons
  select      Size (GMUO)  >= 1  # events with 2 or more electrons
  select      Size (gleps) > 2  # events with 2 or more leptons
```

OutLook

- compatibility between CutLang and LHADA is nearly achieved via ADL

- A CutLang Gui is planned
  - edit config file, run, look at histograms…



- Improve CutLang documentation & training guide including a wiki page

- More work on CutLang v2
  - ✓ 2D histograms…
  - ✓ SORT, COMBInation
  - multithreaded version
  - automatic inputfile type and external function inclusion
  - Analysis database preparation

We will have a CERN summer student to work on some of these. CutLang needs you!