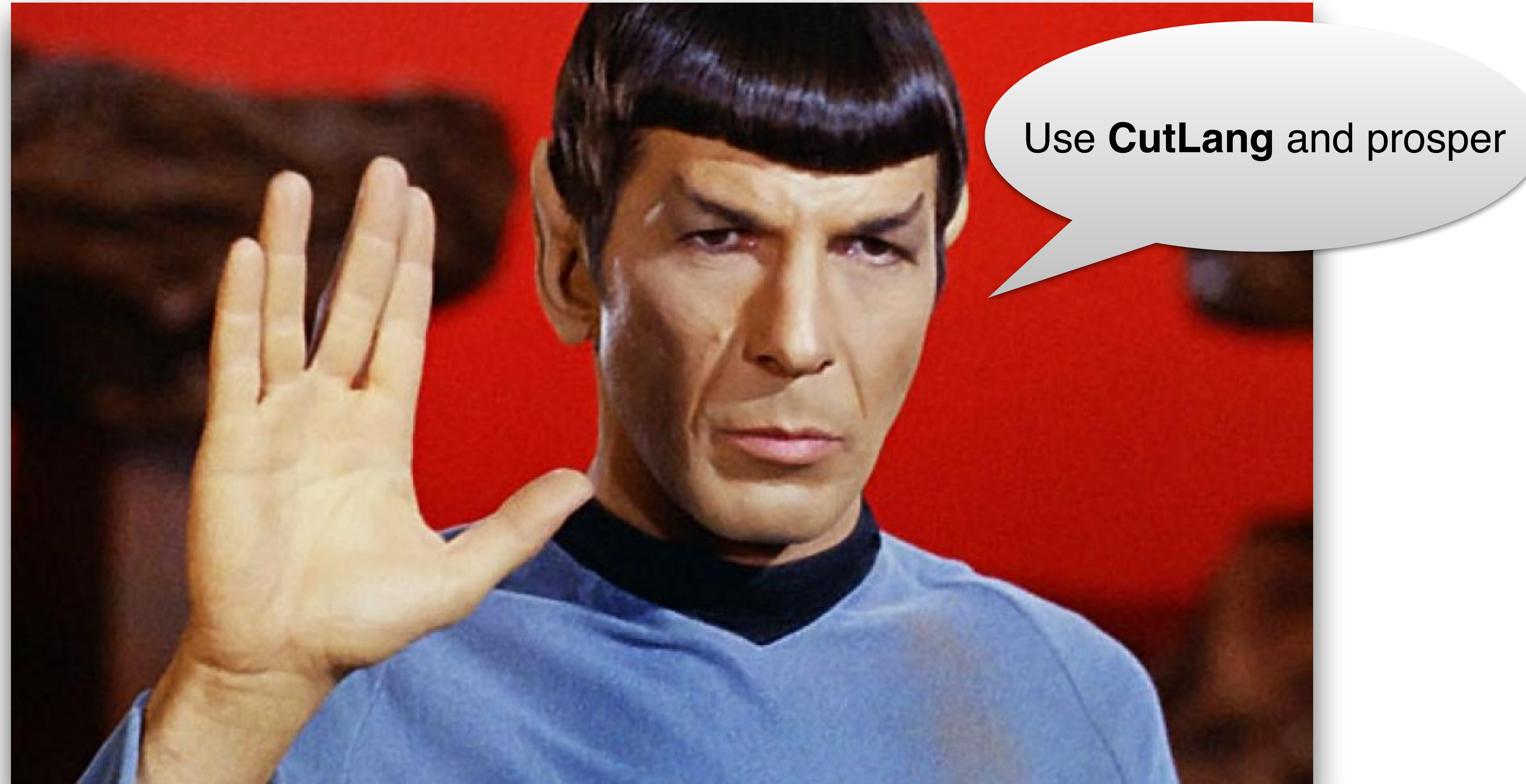
Tablet computers
Tractor beams
Flip (and wearable) communicators
Replicators (3D printer)
Voice interface computers (hello Siri)

Bluetooth headsets (Uhura had one first)
Google Glass
Portable memory (USB sticks)
Focused ultrasound technology
Biometric data tracking & identity verification

GPS
Automatic doors
Big screen displays
Real-time universal translators
Teleconferencing



Did StarTrek also hint modern data analysis?

S. Sekmen (KNU) & G. Unel (UCI)

reinterpretation workshop May 2018

I wish we had a similar data analysis tool…

Introducing “CutLang”

- A dedicated cut based analysis description language and runtime interpreter
  - Human readable text file to describe the whole analysis
    - follows closely the LHADA principles
    - Back and forth convertible to XML, editable in XML
  - Run time interpretation of the text-based analysis. No compiling!

- Works with multiple input data formats
  - Currently available data formats:
    - LVL0, ATLAS OpenData, CMS OpenData, Delphes, LHCO, FCC
    - more can be easily added…
  - Events from ROOT file(s) given in command line are TChain’ed

- Documentation
  - arXiv paper: internals, user manual, how to run examples etc. arXiv:1801.05727
  - Web page: user manual, examples & source code tgz: https://cutlang.hepforge.org

CutLang details

- Interpreted language
  - Commands are read from input file, interpreted and evaluated at run-time
  - No more compilation, forgotten { } ; issues, wrong loop variables etc.
  - Very simple syntax for cuts, histograms, objects, definitions, etc…
- Very *modest* requirements
  - Linux or Mac, C++ (gcc4.x) & ROOT (5 or 6)
  - Pure C++ classes, on top of ROOT LorentzVectors and histograms
  - Analysis can be run on a single core or on a PROOF farm
- Helps the analyst and the advisor
  - Shell & Python scripts available for plotting & addition of “user functions”
  - All definitions, cuts and object selections are saved into output ROOT file

Basics

- The execution order is top to bottom. (Units in GeV, comment is #)
  - Particles between { } are added, + is not used.

- There are 3 basic keywords: **cmd**, **def**, and **histo**
  - a command (cmd) is either a cut or an instruction to fill histograms

```
cmd        "ALL "
cmd        "nJET >= 5 "
cmd        "nQGJET >= 4 "
cmd        "nBJET >= 1 "
cmd        "MET < 100 "
```

  - a definition (def) helps the physicist to construct variables

```
def        "mLL : { LEP_1 LEP_0 }m"
def        "qLL : { LEP_1 LEP_0 }q"
def        "Zreco : LEP_0 LEP_1 "
def        "dR(LL,J0) : { Zreco , JET_0 }R "
```

  - a histo contains the definition of a (currently) 1D histogram, ROOT/Paw style

```
cmd       "FillHistos "
histo     "ZPTcon2   , Z PT after Z Mass Window (GeV)     , 360,  0, 1800, 3, { LEP_1 LEP_0 }Pt  "
histo     "ZETAcon2  , Z Eta after Z Mass Window          , 50,  -5,    5, 3, { LEP_1 LEP_0 }Eta "
histo     "ZPHIcon0  , Z Phi after preselection cuts      , 52,  -5,    5, 1, { LEP_1 LEP_0 }Phi "
```

Predefined Objects & Functions

- The Objects

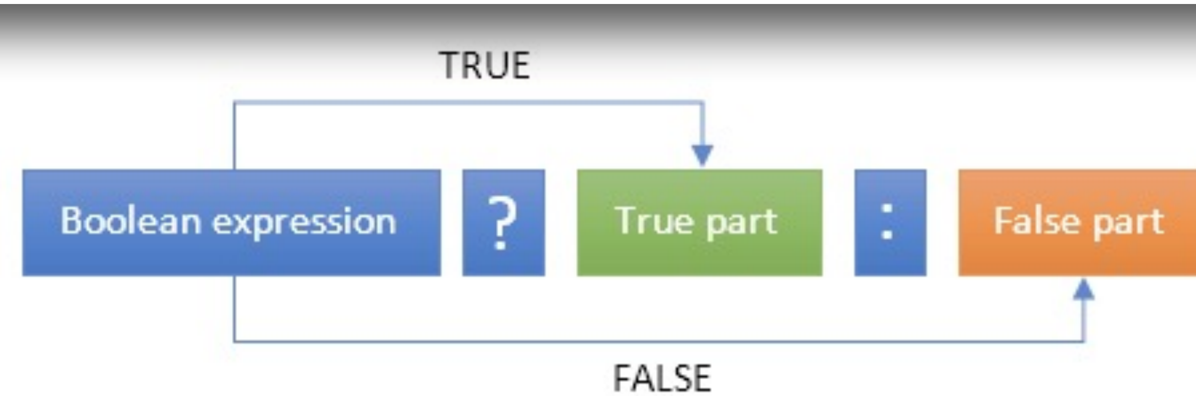
Name	Keyword	Highest Pt object	Second Highest Pt object	j^{th}Highest Pt object
electron	ELE	ELE_0	ELE_1	ELE_j
muon	MUO	MUO_0	MUO_1	MUO_j
lepton	LEP	LEP_0	LEP_1	LEP_j
photon	PHO	PHO_0	PHO_1	PHO_j
jet	JET	JET_0	JET_1	JET_j
b-tagged Jet	BJET	BJET_0	BJET_1	BJET_j
light Jet	QGJET	QGJET_0	QGJET_1	QGJET_j
neutrino	NUMET	NUMET_0	NUMET_1	NUMET_j
missing ET	METLV	METLV_0	N/A	N/A

- Functions

Function	Returned quantity
nELE	number of electrons
nMUO	number of muons
nPHO	number of photons
nLEP	number of leptons (electrons or muons, trigger dependent)
nJET	number of jets
nBJET	number of *b*-tagged jets
nQGJET	number of light jets
SumHTJET	sum of all the jets transverse momenta
METMWT	sum of the leptonically reconstructed W boson's transverse mass and missing transverse energy
MWT	transverse mass of leptonically reconstructed *W* boson
MET	missing transverse energy
ALL	all events
LEPsf	inclusion of lepton MC scale factors
FillHistos	filling histograms defined afterwards

Returned quantity	Function	Argument
Mass of	{ }m	a particle
Charge of	{ }q	a particle
Phi of	{ }Phi	a particle
Eta of	{ }Eta	a particle
Transverse momentum of	{ }Pt	a particle
Axial momentum of	{ }Pz	a particle
Total momentum of	{ }P	a particle
Energy of	{ }E	a particle
Angular distance between	{ }R	two comma separated particles

- The **ternary** function as the cherry on the cake



A very simple example

- reconstruct Z boson candidate from two leptons
- histogram the mass of the candidate

```
minpte  = 15.0  # min pt of electrons
minptm  = 15.0  # min pt of muons
minptj  = 15.0  # min pt of jets
maxetae = 2.47  # max pseudorapidity of electrons
maxetam = 2.5   # max pseudorapidity of muons
maxetaj = 5.5   # max pseudorapidity of jets

TRGm = 0 #     muon Trigger Type: 0=dont trigger, 1=1st trigger (data) 2=2nd trigger (MC)
TRGe = 2 # electron Trigger Type: 0=dont trigger, 1=1st trigger (data) 2=2nd trigger (MC)

###### DEFINITIONS
def        "mLL : { LEP_1 LEP_0 }m"
def        "qLL : { Zreco }q"         #note the nested definition usage
def        "Zreco : LEP_0 LEP_1 "

###### Very simple Reconstruction
cmd       "ALL "            # to count all events
cmd       "nLEP == 2 "      # events with only leptons
cmd       "mLL [] 70 120 " # a mass window for Z candidate
cmd       "qLL == 0 "       # reconstructed object should be neutral
cmd       "FillHistos "
histo     "Zlm , Leptonic Z best combi (GeV), 50, 50, 150,  mLL  "
```

A less simple example

- Scan all lepton pairs to find the pair yielding closest Z boson mass
- Negative indices mean “to be automatically determined per event”
    - These are to be used in conjunction with **optimizer** operators: **~=, !=**

```
minpte  = 15.0  # min pt of electrons
minptm  = 15.0  # min pt of muons
minptj  = 15.0  # min pt of jets
maxetae = 2.47  # max pseudorapidity of electrons
maxetam = 2.5   # max pseudorapidity of muons
maxetaj = 5.5   # max pseudorapidity of jets

TRGm = 0 #     muon Trigger Type: 0=dont trigger, 1=1st trigger (data) 2=2nd trigger (MC)
TRGe = 2 # electron Trigger Type: 0=dont trigger, 1=1st trigger (data) 2=2nd trigger (MC)

###### USER DEFINITIONS
def      "mLL : { Zreco }m"
def      "qLL : { Zreco }q"         #note the nested definition usage
def      "Zreco : LEP_-1 LEP_-1 "

cmd      "ALL "              # to count all events
cmd      "nLEP >= 2 "        # events with only leptons
cmd      "mLL ~= 90     "    # central mass for Z candidate
cmd      "qLL == 0 "         # reconstructed object should be neutral
cmd      "FillHistos "
histo    "Zlm1 , Leptonic Zreco (GeV), 50, 50, 150,  mLL "
```

$t\bar{t}$ Reconstruction example

```
def        "Wh1 : JET_-1 JET_-1 "  # -1 as index: search to match a condition
def        "Wh2 : JET_-3 JET_-3 "  # anohter W boson, hadronic reco
def        "mTop1 : { JET_-1 JET_-1 JET_-2 }
def        "mTop2 : { JET_-3 JET_-3 JET_-4 }
def        "mWh1 : { JET_-1 JET_-1 }m "  # m
def        "mWh2 : { JET_-3 JET_-3 }m "  # m

#######cuts and histos -- ATTENTION#######
# for the moment only 1D histos are impleme
# The delimiter is comma , the order is:
# Histogram name to find in output root fil
# Histogram title on x axis # number of bins, xmin and xmax
# step at which the histogram is to be filled.
# What to fill using CutLang notation -- see CutLang documentation for details
# THERE is a special keyword, Basics followed by HistoSet ID and step at which the histogram is to filled.
# ATTENTION to spaces before and after the = and { } signs. DO not deviate from the examples.
##################################################################
cmd       "ALL "
cmd       "nJET >= 6 "
cmd       "nLEP == 0 "
cmd       "MET < 100 "
cmd       "mTop1 - mTop2 / 4 ^ 2 +  mWh1 - 80.4 / 2 ^ 2  + mWh2 - 80.4 /2 ^ 2 ~= 0 " #find best combi
cmd       "FillHistos "
histo     "Whm1 , Hadronic W best combi (GeV), 50, 50, 150,  mWh1  "
histo     "Whm2 , Hadronic W best combi (GeV), 50, 50, 150,  mWh2  "
histo     "TopQh1 , Hadronic top combi (GeV), 70,  0, 700,  mTop1  "
histo     "TopQh2 , Hadronic top combi (GeV), 70,  0, 700,  mTop2  "
histo     "TopQh2 , Hadronic top combi (GeV), 70,  0, 700,  mTop2  "
histo     "WbR1 , Angular distance between W1 and bjet, 50,  0, 10,  { Wh1 , JET_-2 }R "
cmd       "{ Wh1 , JET_-2 }R > 0.2 "
histo     "TopQh1a , Hadronic top combi (GeV) after angular cut, 70,  0, 700,  mTop1  "
```

, with the χ^2 defined as:

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

Running with multiple regions

- CutLang allows definition of multiple selection regions via the keyword **algo**
  - Typically this is used to test different techniques or define signal and control regions (SR, CR1, CR2…)
- A user defined region can contain another one
  - e.g. SR containing a preselection
- All regions are processed in parallel and saved as directories in the output ROOT file
  - e.g. **arXiv:1704.03848** has 6 regions

```
######## EVENT SELECTION
algo __preselection__
cmd   "ALL "                                  # to
cmd   " nPHOtight >= 0 "                     # event
cmd   "{ PHOtight_0 }Pt > 150 "               # sel
cmd   " MET / HT ^ 0.5 > 8.5 "                 # se
cmd   " nJETsr  =< 1 "
cmd   "{ JET_0 , METLV_0 }dPhi > 0.4 "         # se
cmd   "  nMUOclean == 0 "                     # selec
cmd   "  nELEclean == 0 "                     # selec

algo __SRI1__
__preselection__
cmd      "MET > 150 "
algo __SRI2__
__preselection__
cmd      "MET > 225 "
algo __SRI3__
__preselection__
cmd      "MET > 300 "

algo __SRE1__
__preselection__
cmd      "MET [] 150 225 "
algo __SRE2__
__preselection__
cmd      "MET [] 225 300 "
```

Derived objects

- Further cleaning or refining can be achieved using derived objects
  - Derived objects can further be used to derive more refined objects
  - Multiple selection criteria can be applied
- Analysis algorithms can use the original objects or refined objects

```
######## USER OBJECTS
obj  "JETclean : JET "
cmd  "{ JET_ , ELE_ }dR >= 0.9 "
cmd  "{ JET_ , MUO_ }dR >= 1.0 "
cmd  "{ JET_ }Pt >= 20 "

obj  "MUOclean : MUO "
cmd  "{ MUO_ , JETclean_ }dR >= 1.4 "

obj  "ELEclean : ELE "
cmd  "{ ELE_ , JETclean_ }dR >= 0.4 "

obj  "PHOtight : PHO "
cmd  "{ PHO_ }Eta [] -1.37 1.37 "

obj  "JETsr : JETclean "
cmd  "{ JETclean_ , ELE_ }dR >= 0.4 "
cmd  "{ JETclean_ }Pt > 50 "
cmd  "{ JETclean_ }Eta [] -4.5 4.5 "

algo __preselection__
cmd   "ALL "                                  # to c
cmd   " nJET   >= 4 "                      # events wit
cmd   " nJETclean  >= 4 "
cmd   " nPHOtight >= 0 "                        # events
#cmd   "{ PHOtight_0 }Pt > 150 "                  # sel
#cmd   "{ PHOtight_0 , METLV_0 }dPhi > 0.4 "
cmd   " MET / HT ^ 0.5 > 8.5 "                   # sel
cmd   " nJETsr >= 4 "                          # reject
```

Notes:

1. an object without an index refers to all members of that class (JET_: all jets)
2. new object names should start with basic object types

User (external) functions

- User defined selection functions should not be called "external"
  - data access methods are framework specific
  - variable names and types are framework specific
    - "download & use" not possible in any software framework
- Best we can "currently" do is to define a user function and compile it "in".
  - CutLang provides the means to do just this

```
NGU-mbp13-tr:scripts ngu$ ./adduserfunction.py myCut

Created the user function analysis_core/dbx_myCut.h.
Please edit and recompile.
```

```
#ifndef DBX_MYCUT_H
#define DBX_MYCUT_H

class dbxCutmyCut : public dbxCut {
 public:
      dbxCutmyCut: dbxCut("}myCut"){}
      dbxCutmyCut(std::vector<int> ts, std::vector<int> is, in

      bool select(AnalysisObjects *ao){
          float result;
          result=calc(ao);
          return (Ccompare(result) );
      }
      float calc(AnalysisObjects *ao){
         float retval;

         // *********************************
         // Write your own code here
         // *********************************

         return retval;
}
private:
};

#endif
```

- A future project: enlarge current(+-/*^) operators to include more mathematical functions to reduce the need for user functions.

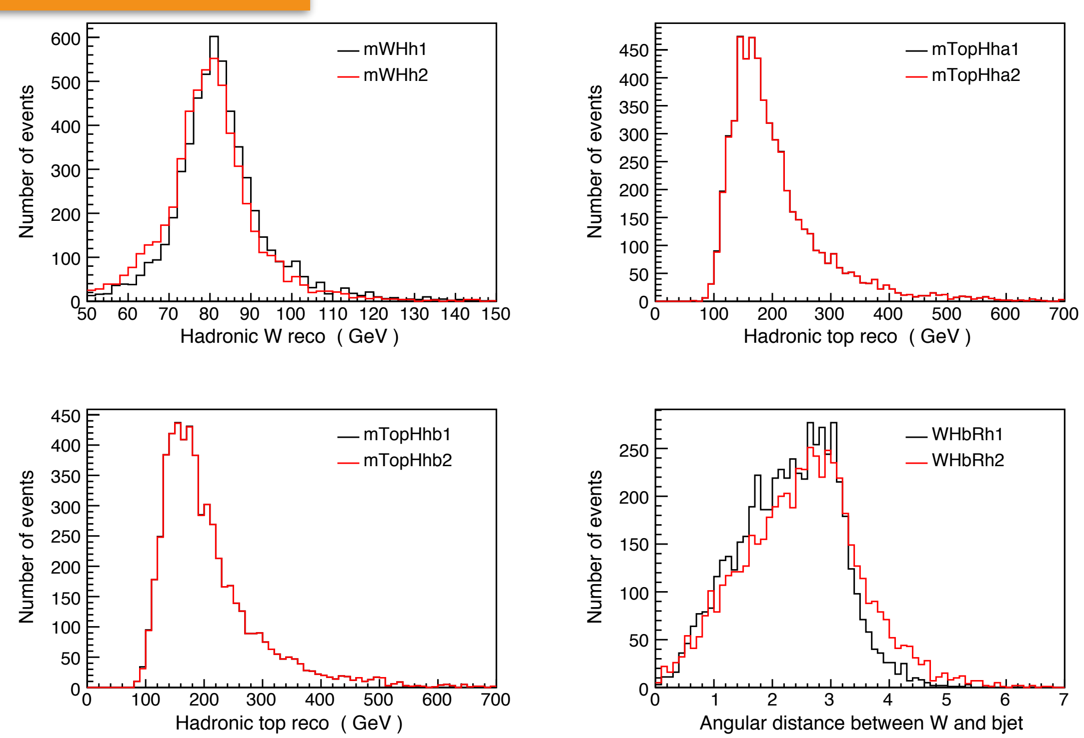
Usage, input & output files

```
./CLA.sh ROOTfile_name ROOTfile_type
```

ATLASOD (ATLAS open data),
LVLO, FCC, LHCO, Delphes,
CMSOD (CMS open data)

```
optional command line arguments:
    -i|--inifile
    -e|--events
    -h|--help
    -v|--verbose
```

./showall.sh 2




```
histoOut-CLA.root
```

```
root [1] .ls
TFile**    histoOut-CLA.root
 TFile*    histoOut-CLA.root
  KEY: TDirectoryFile   BP_1;1  BP_1
  KEY: TDirectoryFile   BP_2;1  BP_2
  KEY: TDirectoryFile   BP_3;1  BP_3
```

```
root [2] .ls
TDirectoryFile* BP_2 BP_2
 KEY: TText CLA2defs;1
WH1 : JET_-1 JET_-1
WH2 : JET_-11 JET_-11
mWH1 : { WH1 }m
mWH2 : { WH2 }m
mTopH1 : { WH1 JET_-2 }m
mTopH2 : { WH2 JET_-4 }m
WHbR1 : {WH1 , JET_-2 }R
WHbR2 : {WH2 , JET_-4 }R
Wchi2 : { WH1 }m - 80.4 / 2.1 ^ 2 + { WH2 }m - 80.4 / 2.1 ^ 2
topchi2 : mTopH1 - mTopH2 / 4.2 ^ 2

 KEY: TText CLA2cuts;1
cmd1 : ALL
cmd2 : nJET >= 6
cmd3 : MET < 100
cmd4 : topchi2 + Wchi2 ~= 0
cmd5 : FillHistos
cmd6 : WHbR1 > 0.6
cmd7 : WHbR2 > 0.6
cmd8 : FillHistos
```

```
# arxiv:1704.03848, ATLAS monophoton

info analysis
# Details about experiment
  experiment ATLAS
  id EXOT-2016-32
  publication Eur.Phys.J. C77 (2017) no.6, 393
  sqrtS 13.0
  lumi 36.1
  arXiv 1704.03848
  hepdata https://www.hepdata.net/record/ins1591
  doi 10.1140/epjc/s10052-017-4965-8


# FUNCTIONS

function METoverSqrtSumET
  arg MET
  arg scalarHT
  code ATLASEXOT1704.0384_functions.h

# OBJECT SELECTIONS

object photons
# Delphes photons
  take Delphes_Photon
  select PT > 10
  select |Eta| < 2.37

object tightphotons
  take photons
  select |eta| < 1.37
  select |eta| > 1.52 and |eta| < 2.37

object electrons
  take Delphes_Electron
  select PT > 7
  select |eta| < 2.7
```

```
object muons
  take Delphes_Muon
  select PT > 6
  select |eta| < 2.7

object jets
# Delphes jets
  take Delphes_Jet
  select PT > 20

object cleanjets
  take jets
  apply dR(Eta, Phi, electrons.Eta, electrons.
  reject dRje < 0.2

object cleanelectrons
  take electrons
  apply dR(Eta, Phi, cleanjets.Eta, cleanjets.
  reject dRlj < 0.4

object cleanmuons
  take muons
  apply dR(Eta, Phi, cleanjets.Eta, cleanjets.
  reject dRlj < 0.4

object verycleanjets
  take cleanjets
  apply dR(Eta, Phi, photons.Eta, photons.Phi)
  reject dRje < 0.4

object jetsSR
  take verycleanjets
  select PT > 30
  select |Eta| < 4.5

object MET
  take Delphes_MissingET

object scalarHT
  # scalar sum of jets, electrons, photons, mu
  take Delphes_scalarHT
```

```


# VARIABLES

variable METoverSqrtSumET
  apply METoverSqrtSumET(MET, scalarHT)

# EVENT SELECTION

cut preselection
# Pre-selection cuts
  select photons[0].PT > 150
  select dphi(photon[0].Phi, MET.Phi) > 0.4
! select METoverSqrtSumET > 8.5
  reject jetsSR.size > 1
  reject dphi(jetsSR.Phi, MET.Phi) > 0.4
  select cleanmuons.size == 0
  select cleanelectrons.size == 0

cut SRI1
  select preselection
  select MET.PT > 150

cut SRI2
  select preselection
  select MET.PT > 225

cut SRI3
  select preselection
  select MET.PT > 300

cut SRE1
  select preselection
  select MET.PT > 150 and MET.PT < 225

cut SRE2
  select preselection
  select MET.PT > 225 and MET.PT < 300
```

```
# PLEASE PAY ATTENTION TO SPACE BEFORE AND AFTER = SIGN
# format is " variable = value "

# info analysis
#   experiment ATLAS
#   id EXOT-2016-32
#   publication Eur.Phys.J. C77 (2017) no.6, 393
#   sqrtS 13.0
#   lumi 36.1
#   arXiv 1704.03848
#   hepdata https://www.hepdata.net/record/ins1591328
#   doi 10.1140/epjc/s10052-017-4965-8

######## GENERIC OBJECT THRESHOLDS
minptp  = 10.0  # min pt of photons
minpte  =  7.0  # min pt of electrons
minptm  =  6.0  # min pt of muons
minptj  = 20.0  # min pt of jets

maxetap = 2.37  # max pseudorapidity of photons
maxetae = 2.70  # max pseudorapidity of electrons
maxetam = 2.70  # max pseudorapidity of muons
maxetaj = 4.50  # max pseudorapidity of jets

######## OBJECT SELECTION
obj  "JETclean : JET "
cmd  "{ JET_ , ELE_ }dR >= 0.9 "
cmd  "{ JET_ , MUO_ }dR >= 1.0 "
cmd  "{ JET_ }Pt >= 20 "

obj  "MUOclean : MUO "
cmd  "{ MUO_ , JETclean_ }dR >= 1.4 "

obj  "ELEclean : ELE "
cmd  "{ ELE_ , JETclean_ }dR >= 0.4 "

obj  "PHOtight : PHO "
cmd  "{ PHO_ }AbsEta ][  1.37 1.52 "

obj  "JETsr : JETclean "
cmd  "{ JETclean_ , PHO_ }dR >= 0.4 "
cmd  "{ JETclean_ }Pt > 50 "
cmd  "{ JETclean_ }AbsEta < 4.5 "

######## EVENT SELECTION
algo __preselection__
cmd   "ALL "                             # to count all events
cmd   " nPHOtight >= 0 "               # events with 1 or more tight photons
cmd   "{ PHOtight_0 }Pt > 150 "          # select photons[0].PT > 150
cmd   " MET / HT ^ 0.5 > 8.5 "            # select METoverSqrtSumET > 8.5
cmd   " nJETsr  =< 1 "
cmd   "{ JET_0 , METLV_0 }dPhi > 0.4 "    # select dphi(jetsSR.Phi, MET.Phi) > 0.4
cmd   "  nMUOclean == 0 "              # select cleanmuons.size == 0
cmd   "  nELEclean == 0 "              # select cleanelectrons.size == 0

algo __SRI1__
__preselection__
cmd      "MET > 150 "
algo __SRI2__
__preselection__
cmd      "MET > 225 "
algo __SRI3__
__preselection__
cmd      "MET > 300 "

algo __SRE1__
__preselection__
cmd      "MET [] 150 225 "
algo __SRE2__
__preselection__
cmd      "MET [] 225 300 "
```

LHADA version


```
# arxiv:1605.03814 ATLAS JetMET

info analysis
# Details about experiment
  experiment ATLAS
  id SUSY-2013-15
  publication Eur. Phys. J. C(2016) 76: 392
  sqrtS 13.0
  lumi 3.2
  arXiv 1605.03814
  hepdata http://hepdata.cedar.ac.uk/view/ins1304456
  doi 10.1140/epjc/s10052-016-4184-8

function Meff
  arg jetsSR
  arg MET
  code ATLASSUSY1605.03814_functions.h

function dphijNjle3METmin
  arg jetsSR
  arg MET
  code ATLASSUSY1605.03814_functions.h

function dphijNjgt3METmin
  arg jetsSR
  arg MET
  code ATLASSUSY1605.03814_functions.h

function METovermeffNJ
  arg jetsSR
  arg njets
  arg MET
  code ATLASSUSY1605.03814_functions.h

function METoversqrtHT
  arg jetsSR
  arg MET
  code ATLASSUSY1605.03814_functions.h

function aplanarity
  arg jetsSR
  code ATLASSUSY1605.03814_functions.h


# OBJECT SELECTIONS
object jets
# Delphes jets
  take Delphes_Jet
  select PT > 20
  select |Eta| < 2.8

object cleanjets
  take jets
  apply dR(Eta, Phi, electrons.Eta, electrons.Phi) dRje
  reject dRje < 0.2

object bjets
# b-tagging jets
  take jets
  select BTag = 1
  select PT > 50
  select |Eta| < 2.5

object muons
# Muons
  take Delphes_Muon
  select PT > 10
  select |Eta| < 2.7
  select IsolationVarRhoCorr < 0.1
  #select isol(src=tracks, dR=0.4, reliso=true)<0.1

object cleanmuons
  take muons
  apply dR(Eta, Phi, cleanjets.Eta, cleanjets.Phi) dRlj
  reject dRlj < 0.4

object electrons
# loose electrons
  take Delphes_Electron
  select PT > 10
  select |Eta| < 2.47

object cleanelectrons
  take electrons
  apply dR(Eta, Phi, cleanjets.Eta, cleanjets.Phi) dRlj
  reject dRlj < 0.4

object verycleanelectrons
  take cleanelectrons
  apply dR(Eta, Phi, cleanelectrons.Eta, cleanelectrons.Phi) dRee
  reject (dRee < 0.05 and (PT < cleanelectrons.PT)

object MET
  take Delphes_MissingET

object jetsSR
  take cleanjets
  select PT > 50

# EVENT VARIABLES

variable Meff
  apply Meff(jetsSR, MET)

variable dphijNjle33METmin
  apply dphijNjle3METmin(jetsSR, MET)

variable dphijNjgt3METmin
  apply dphijNjgt3METmin(jetsSR, MET)

variable METoversqrtHT
  apply METoversqrtHT(jetsSR, MET)

variable METovermeff2j
  apply METovermeffNJ(jetsSR, 2, MET)

variable METovermeff4j
  apply METovermeffNJ(jetsSR, 4, MET)

variable METovermeff5j
  apply METovermeffNJ(jetsSR, 5, MET)

variable METovermeff6j
  apply METovermeffNJ(jetsSR, 6, MET)

variable aplanarity
  apply aplanarity(jetsSR)


# EVENT SELECTION

cut preselection
# Pre-selection cuts
  select MET.PT > 200
  reject cleanmuons.PT > 10
  reject verycleanelectrons.PT > 10
  select jetsSR.size > 0

cut 2jl
  select preselection
  select jetsSR.size >= 2
  select jetsSR[0].PT > 200
  select jetsSR[1].PT > 200
  select dphijNjle3METmin > 0.8
  select METoversqrtHT > 15
  select Meff > 1200

cut 2jm
  select preselection
  select jetsSR.size >= 2
  select jetsSR[0].PT > 300
  select jetsSR[1].PT > 50
  select dphijNjle3METmin > 0.4
  select METoversqrtHT > 15
  select Meff > 1600

cut 2jt
  select preselection
  select jetsSR.size >= 2
  select jetsSR[0].PT > 200
  select jetsSR[1].PT > 200
  select dphijNjle3METmin > 0.8
  select METoversqrtHT > 20
  select Meff > 2000

cut 4jt
  select preselection
  select jetsSR.size >= 4
  select jetsSR[0].PT > 200
  select jetsSR[1].PT > 100
  select jetsSR[2].PT > 100
  select jetsSR[3].PT > 100
  select dphijNjle3METmin > 0.4
  select dphijNjgt3METmin > 0.2
  select aplanarity > 0.04
  select METoverMeff4j > 0.2
  select Meff > 2200

cut 5j
  select preselection
  select jetsSR.size >= 5
  select jetsSR[0].PT > 200
  select jetsSR[1].PT > 100
  select jetsSR[2].PT > 100
  select jetsSR[3].PT > 100
  select jetsSR[4].PT > 50
  select dphijNjle3METmin > 0.4
  select dphijNjgt3METmin > 0.2
  select aplanarity > 0.04
  select METoverMeff5j > 0.25
  select Meff > 1600

cut 6jm
  select preselection
  select jetsSR.size >= 6
  select jetsSR[0].PT > 200
  select jetsSR[1].PT > 100
  select jetsSR[2].PT > 100
  select jetsSR[3].PT > 100
  select jetsSR[4].PT > 50
  select jetsSR[5].PT > 50
  select dphijNjle3METmin > 0.4
  select dphijNjgt3METmin > 0.2
  select aplanarity > 0.04
  select METoverMeff6j > 0.25
  select Meff > 1600

cut 6jt
  select preselection
  select jetsSR.size >= 6
  select jetsSR[0].PT > 200
  select jetsSR[1].PT > 100
  select jetsSR[2].PT > 100
  select jetsSR[3].PT > 100
  select jetsSR[4].PT > 50
  select jetsSR[5].PT > 50
  select dphijNjle3METmin > 0.4
  select dphijNjgt3METmin > 0.2
  select aplanarity > 0.04
  select METoverMeff6j > 0.2
  select Meff > 2000
```

CutLang version


```
# PLEASE PAY ATTENTION TO SPACE BEFORE AND AFTER = SIGN
# format is " variable = value "

# info analysis
#  experiment ATLAS
#  id SUSY-2013-15
#  publication Eur. Phys. J. C(2016) 76: 392
#  sqrtS 13.0
#  lumi 3.2
#  arXiv 1605.03814
#  hepdata http://hepdata.cedar.ac.uk/view/ins1304456
#  doi 10.1140/epjc/s10052-016-4184-8

######## GENERIC OBJECT THRESHOLDS
minptp  = 10.0  # min pt of photons
minpte  = 10.0  # min pt of electrons
minptm  = 10.0  # min pt of muons
minptj  = 20.0  # min pt of jets

maxetap = 2.37  # max pseudorapidity of photons
maxetae = 2.47  # max pseudorapidity of electrons
maxetam = 2.70  # max pseudorapidity of muons
maxetaj = 2.80  # max pseudorapidity of jets

######## USER DEFINITIONS
def "Meff : MET + HT "  #Meff is simple
def "Jsr0MEtFi : { JETsr_0 , METLV_0 }dPhi "
def "Jsr1MEtFi : { JETsr_1 , METLV_0 }dPhi "
def "Jsr2MEtFi : { JETsr_2 , METLV_0 }dPhi "
def "Meff4j : MET + { JETsr_0 }Pt + { JETsr_1 }Pt + { JETsr_2 }Pt + { JETsr_3 }Pt "
def "Meff5j : Meff4j + { JETsr_4 }Pt "
def "Meff6j : Meff5j + { JETsr_5 }Pt "

######## OBJECT SELECTION
obj  "JETclean : JET "
cmd  "{ JET_ , ELE_ }dR >= 0.2 "

# How to do bjets?

obj  "MUOclean : MUO "
cmd  "{ MUO_ , JETclean_ }dR >= 1.4 "
#cmd  "{ MUO_ }IsolationVarRhoCorr < 0.1"

obj  "ELEclean : ELE "
cmd  "{ ELE_ , JETclean_ }dR >= 0.4 "

obj  "ELEveryclean : ELE "
cmd  "{ ELE_ , JETclean_ }dR >= 0.4 "

obj  "JETsr : JETclean "
cmd  "{ JETclean_ }Pt > 50 "

######## EVENT SELECTION
algo __preselection__
cmd   "ALL "                          # to count all events
cmd   "MET > 200 "
cmd   "nPHOtight >= 0 "
cmd   "nMUOclean <= 1 "         # Reject evt if there is a muon with pT > 10
cmd   "nMUOclean == 0 ? ALL : { MUOclean_0 }Pt < 10 "          # Reject evt if there is a muon wi
cmd   "nELEveryclean <= 1 "         # Reject evt if there is a muon with pT > 10
cmd   "nELEveryclean == 0 ? ALL : { ELEveryclean_0 }Pt < 10 "     # Reject evt if there is an elec
cmd   "nJETsr > 0 "

algo __2jl__
__preselection__
cmd   "nJETsr >= 1 "
cmd   "{ JETsr_0 }Pt > 200 "
cmd   "nJETsr >= 2 "
#cmd   "Jsr0MEtFi - Jsr1MEtFi < 0 ? Jsr0MEtFi > 0.8 : Jsr1MEtFi > 0.8 "
#cmd   "nJETsr >= 3 "
#cmd   "Jsr0MEtFi - Jsr1MEtFi < 0 ?  Jsr0MEtFi - Jsr2MEtFi < 0 ? Jsr0MEtFi > 0.8 : Jsr2MEtFi > 0.8 : Jsr1MEtFi - Jsr2MEtFi < 0 ? Jsr1MEtFi > 0.8 : Jsr2MEtFi > 0.8 "
cmd   " Ex1 > 0.8  "
cmd   "{ JETsr_1 }Pt > 200 "
cmd   "MET / HT ^ 0.5 > 15 "  # make sure we use JETsr in HT
cmd   "Meff > 1200 "

algo __2jm__
__preselection__
cmd   " nJETsr >= 2 "
cmd   "{ JETsr_0 }Pt > 300 "
cmd   "{ JETsr_1 }Pt > 50 "
cmd   "Jsr0MEtFi - Jsr1MEtFi < 0 ? Jsr0MEtFi > 0.4 : Jsr1MEtFi > 0.4 "
cmd   " MET / HT ^ 0.5 > 15 "
cmd   " Meff > 1600 "

algo __2jt__
__preselection__
cmd   " nJETsr >= 2 "
cmd   "{ JETsr_0 }Pt > 200 "
cmd   "{ JETsr_1 }Pt > 200 "
cmd   "Jsr0MEtFi - Jsr1MEtFi < 0 ? Jsr0MEtFi > 0.8 : Jsr1MEtFi > 0.8 "
cmd   " MET / HT ^ 0.5 > 20 "
cmd   " Meff > 2000 "

algo __4jt__
__preselection__
cmd   " nJETsr >= 4 "
cmd   "{ JETsr_0 }Pt > 200 "
cmd   "{ JETsr_1 }Pt > 100 "
cmd   "{ JETsr_2 }Pt > 100 "
cmd   "{ JETsr_3 }Pt > 100 "
cmd   "Jsr0MEtFi - Jsr1MEtFi < 0 ? Jsr0MEtFi > 0.4 : Jsr1MEtFi > 0.4 "
cmd   "{ JETsr_-1 , METLV_0 }dPhi ~ 0.0 "
cmd   "{ JETsr_-1 , METLV_0 }dPhi > 0.2 "
#cmd   "aplanarity > 0.04 "
cmd   " MET / Meff4j > 0.2 "
cmd   " Meff > 2200 "

algo __5j__
__preselection__
cmd   " nJETsr >= 5 "
cmd   "{ JETsr_0 }Pt > 200 "
cmd   "{ JETsr_1 }Pt > 100 "
cmd   "{ JETsr_2 }Pt > 100 "
cmd   "{ JETsr_3 }Pt > 100 "
cmd   "{ JETsr_4 }Pt > 50 "
cmd   "Jsr0MEtFi - Jsr1MEtFi < 0 ? Jsr0MEtFi > 0.4 : Jsr1MEtFi > 0.4 "
cmd   "{ JETsr_-1 , METLV_0 }dPhi ~ 0.0 "
cmd   "{ JETsr_-1 , METLV_0 }dPhi > 0.2 "
#cmd   " aplanarity > 0.04 "
cmd   " MET / Meff5j > 0.25 "
cmd   " Meff > 1600 "

algo __6jm__
__preselection__
cmd   " nJETsr >= 6 "
cmd   "{ JETsr_0 }Pt > 200 "
cmd   "{ JETsr_1 }Pt > 100 "
cmd   "{ JETsr_2 }Pt > 100 "
cmd   "{ JETsr_3 }Pt > 100 "
cmd   "{ JETsr_4 }Pt > 50 "
cmd   "{ JETsr_5 }Pt > 50 "
cmd   "Jsr0MEtFi - Jsr1MEtFi < 0 ? Jsr0MEtFi > 0.4 : Jsr1MEtFi > 0.4 "
cmd   "{ JETsr_-1 , METLV_0 }dPhi ~ 0.0 "
cmd   "{ JETsr_-1 , METLV_0 }dPhi > 0.2 "
#cmd   " aplanarity > 0.04 "
cmd   " MET / Meff6j > 0.25 "
cmd   " Meff > 1600 "

algo __6jt__
__preselection__
cmd   " nJETsr >= 6 "
cmd   "{ JETsr_0 }Pt > 200 "
cmd   "{ JETsr_1 }Pt > 100 "
cmd   "{ JETsr_2 }Pt > 100 "
cmd   "{ JETsr_3 }Pt > 100 "
cmd   "{ JETsr_4 }Pt > 50 "
cmd   "{ JETsr_5 }Pt > 50 "
cmd   "Jsr0MEtFi - Jsr1MEtFi < 0 ? Jsr0MEtFi > 0.4 : Jsr1MEtFi > 0.4 "
cmd   "{ JETsr_-1 , METLV_0 }dPhi ~ 0.0 "
cmd   "{ JETsr_-1 , METLV_0 }dPhi > 0.2 "
#cmd   " aplanarity > 0.04 "
cmd   " MET / Meff6j > 0.2 "
cmd   " Meff > 2000 "
```

Only 1 user function is needed: **aplanarity**

```
#  experiment ATLAS
#  id SUSY-2013-15
#  publication Eur. Phys. J. C(2016) 76: 392
#  sqrtS 13.0
#  lumi 3.2
#  arXiv 1605.03814
```

Compatibility

	Rivet			MadAnalysis 5			CheckMATE
Description	#evt	tot.eff	rel.eff	#evt	tot.eff	rel.eff	tot.eff
2jl cut-flow	31250	1	-	31250	1	-	
Pre-sel+MET+pT1	28592	0.91	0.91	28626	0.92	0.92	
Njet	28592	0.91	1	28625	0.92	1	
Dphi_min(j,MET)	17297	0.55	0.6	17301	0.55	0.6	
pT2	17067	0.55	0.99	17042	0.55	0.99	
MET/sqrtHT	8900	0.28	0.52	8898	0.28	0.52	
m_eff(incl)	8896	0.28	1	8897	0.28	1	
2jm cut-flow	31250	1	-	32150	1	-	1
Pre-sel+MET+pT1	28472	0.91	0.91	28478	0.91	0.91	0.91
Njet	28472	0.91	1	28477	0.91	1	0.91
Dphi_min(j,MET)	22950	0.73	0.81	22889	0.73	0.8	0.73
pT2	22950	0.73	1	22889	0.73	1	0.73
MET/sqrtHT	10730	0.34	0.47	10710	0.34	0.47	0.33
m_eff(incl)	10630	0.34	0.99	10609	0.34	0.99	0.32
2jt cut-flow	31250	1	-	31250	1	-	
Pre-sel+MET+pT1	28592	0.91	0.91	28626	0.92	0.92	
Njet	28592	0.91	1	28625	0.92	1	
Dphi_min(j,MET)	17297	0.55	0.6	17301	0.55	0.6	
pT2	17067	0.55	0.99	17042	0.55	0.99	
MET/sqrtHT	5083	0.16	0.3	5098	0.16	0.3	
Pass m_eff(incl)	4861	0.16	0.96	4889	0.16	0.96	

CutLang

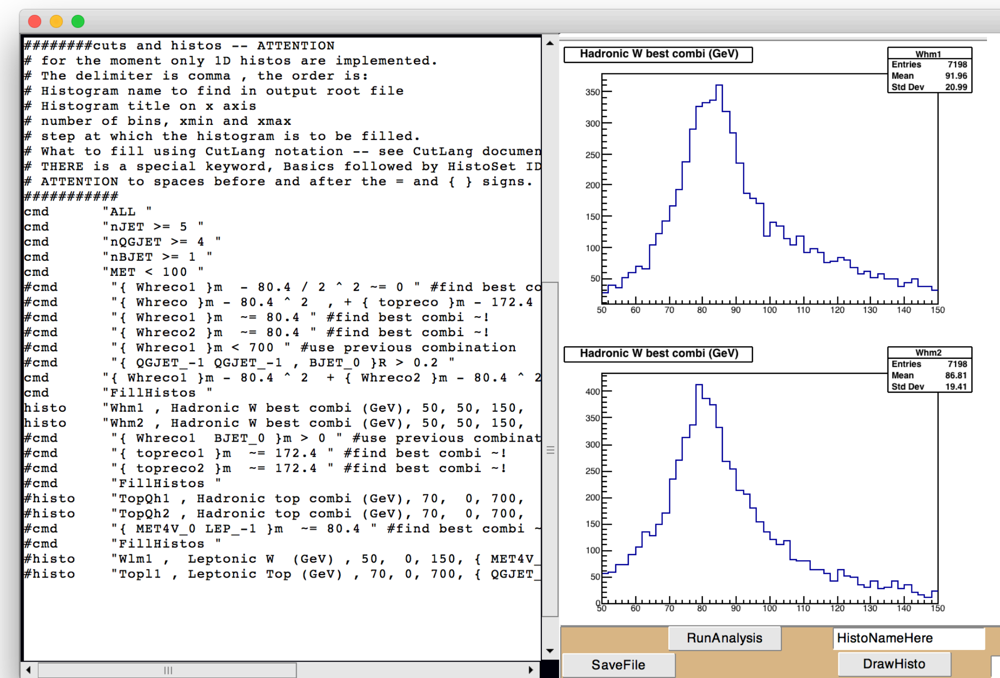
#evt	total eff.	rel. eff.
31250	1.000	-
28431	0.91	0.91
28430	0.91	1.00
16661	0.53	0.59
16381	0.52	0.98
8159	0.26	0.50
8156	0.26	1.00
31250	1.000	-
28301	0.91	0.91
28300	0.91	1.00
22441	0.72	0.79
22441	0.72	1.00
10043	0.32	0.45
9896	0.32	0.99
31250	1.000	-
28431	0.91	0.91
28430	0.91	1.00
16661	0.53	0.59
16381	0.52	0.98
4375	0.14	0.27
4132	0.13	0.94

First reactions

- An earlier version of CutLang is being used in an ATLAS Exotics analysis, the feedback is so far positive:
- Students
  - Very happy not to deal with C++ syntax, pointers, ROOT etc.
  - Very easy to keep track of various analysis variations such as control regions, cut optimizations etc.
  - Very easy to test ideas, disable a cut or swap two cuts, add a histogram etc.
- Advisors
  - Can follow easily what the students are doing
  - Analysis algorithms, object definition thresholds are automatically synchronized to program output
  - Full algorithm including thresholds is saved into output ROOT files

OutLook

- Extensive timing tests between “hard coded” and configurable analysis options
  - CutLang currently takes ~5s for 37000 ttbar cms opendata events on a single core of my 3GHz, i7 laptop from 2014.
- A CutLang Gui is underway
  - edit config file, run, look at histograms…



- Improve CutLang documentation & training guide including a wiki page
- Improve compatibility between CutLang and LHADA
  - one or more input file converters could be written
- Apply to more analyses to get more feedback
  - CutLang internal algorithms could be improved and expanded to cover more cases

We’ll have a CERN summer student to work on some of these projects.
Any other volunteers?

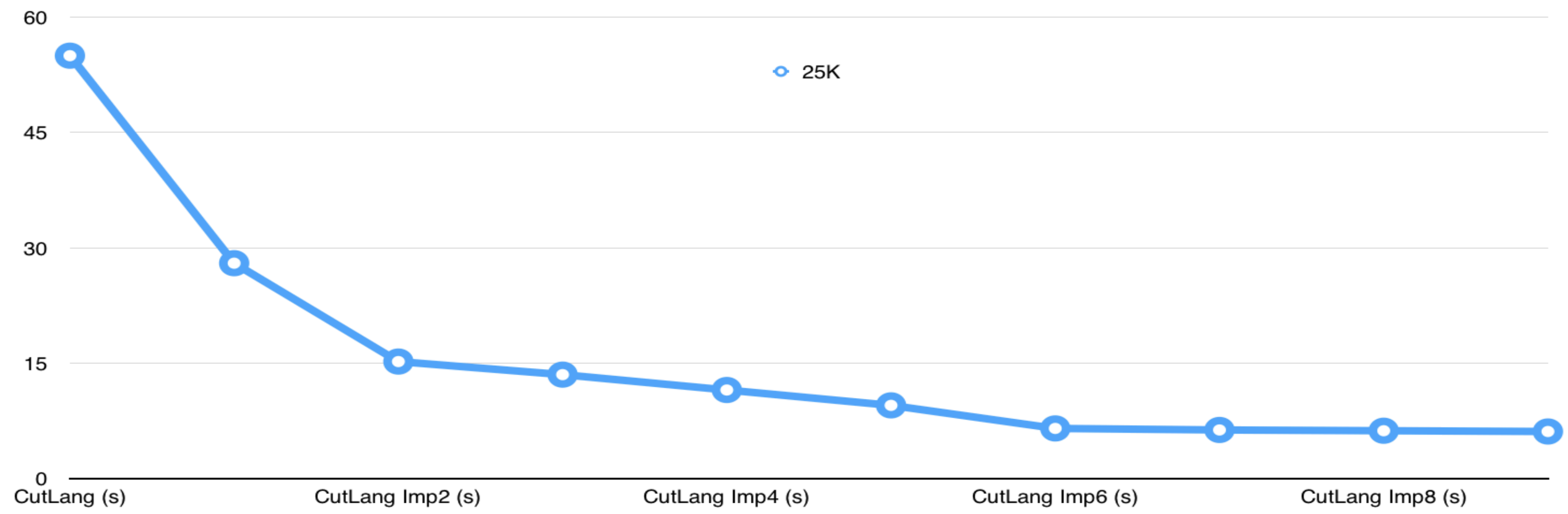
fin

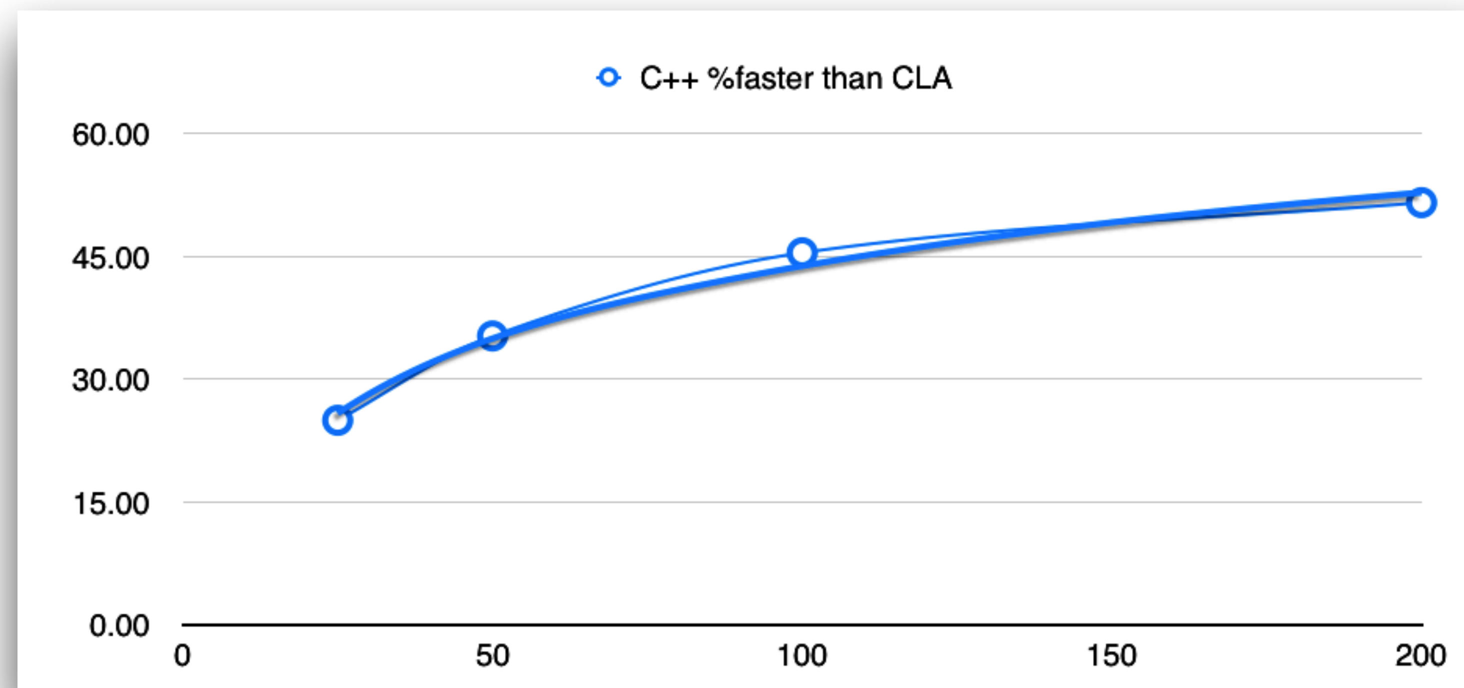
backup slides

Debugging & speeding

ATLAS hadronic ttbar tests

	25K	50K	100K
Sezen (s)	12	24	45.7
Sezen imp (s)	3.1	4.7	7.9
CutLang (s)	55	106	210
CutLang Imp (s)	28	55	108
CutLang Imp2 (s)	15.2	29	56.2
CutLang Imp3 (s)	13.5	25.8	49.7
CutLang Imp4 (s)	11.5	21.6	41.9
CutLang Imp5 (s)	9.5	17.5	33.5
CutLang Imp6 (s)	6.5	11.5	21.6
CutLang Imp7 (s)	6.3	11.1	20.8
CutLang Imp8 (s)	6.2	11.0	20.4
CutLang Imp9 (s)	6.1	10.9	20.1
ratio best	1.9677419	2.319144	2.544300
%slow	103.226	136.170	163.291
%faster	49.180	56.881	60.697
	25	50	100





ATLAS wz tests e- channel

	500K
Sezen (s)	6.3
CutLang Imp (s)	6.9
ratio	1.10
%faster	8.70

For example, the one step and two step top quark reconstructions requiring one line and two lines to implement in the *CutLang* language take about 40 to 70 lines of standard analysis code in C++.

Internal workings

- All predefined cuts in the library are derived from a generic class.
  - They each have a function to be evaluated: f(x)
  - They each have a limit value or an interval of values: a
  - They each have an operator: <=

  } f(x)<=a

- Logical operators possible in both C and Fortran syntax
  - Recall: AND is a multiplication and OR is an addition.
- Evaluation
  - For each event the function result is compared to the limit values
  - Each cut is then written as an arithmetical expression 1*((0)+(1))*1
  - Which is converted to **R**everse **P**olish **N**otation for numerical evaluation using the Shunting Yard algorithm
  - Resulting number is a Boolean for deciding to drop the event, or not.
  - Function value can also be extracted for histogramming