CutLang

Analysis description language and runtime interpreter

G. Ünel (U.C. Irvine),
S. Sekmen (Kyungpook Nat. U.)

What is it?

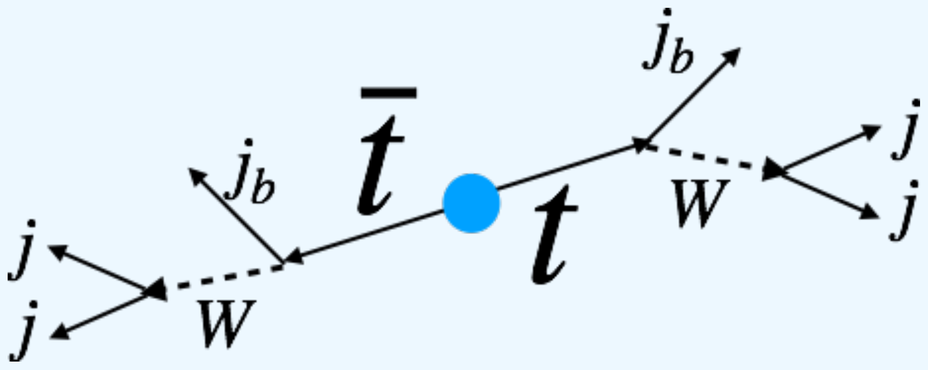
- A declarative language that aims to provide a clear, human readable way to unambiguously define analyses in high energy particle physics,
- An interpretation framework for the language
- Can be used by experimentalists and phenomenologists (actively used in an ATLAS exotics analysis)

Why CutLang?

- A practical approach to writing analyses.
- Facilitates the abstraction, design, visualization, validation, combination, reproduction, interpretation and overall communication of analyses.

A typical LHC top pair reconstruction analysis

How to build an analysis for reconstructing two hadronically decaying top quarks from their decay products?



- Define criteria for jets that can come from top quarks
- Count events
- Make sure there is no large missing transverse energy
- Make sure there are at least 6 clean jets
- Find the correct jet combinations to give the best W and top masses
- Plot W, top candidate invariant masses, angular distance between W and b quark jet candidates

$t\bar{t}$ analysis with CutLang, the language

The object section helps refining physics objects

The **obj** keyword defines a new particle class

New object sets can be defined by applying selection criteria on previously known objects

Object name without the index refers to the whole particle set (loop implied)

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

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

Custom event variables can be defined

Negative indices show indices to be found via a χ^2 search

```
### EVENT VARIABLE DEFINITIONS
def "WH1 : JET_-1 JET_-1 " # 1st W boson
def "WH2 : JET_-11 JET_-11 " # 2nd W boson
def "mWH1 : {WH1 }m " # 1st W boson mass
def "mWH2 : {WH2 }m " # 2nd W boson mass
def "mTopH1 : {WH1 JET_-2 }m " # 1st top mass
def "mTopH2 : {WH2 JET_-4 }m " # 2nd top mass
def "WHbR1 : {WH1 , JET_-2 }dR " # dR(top1, jet)
def "WHbR2 : {WH2 , JET_-4 }dR " # dR(top2, jet)
def "topchi2 : mTopH1 - mTopH2 / 4.2 ^ 2 "                    #top chi2
def "Wchi2 : ((mWH1 - 80.4) / 2.1)^2 + ((mWH2 - 80.4) / 2.1)^2 " #W chi2
```

The **algo** keyword defines algorithms such as multiple signal and background regions

Event selection can be done for multiple regions

The **cmd** keyword precedes the steps of the analysis

The **histo** keyword defines histograms for plotting distributions

χ^2 minimisation

Mass window selection

Multiple algorithms are executed in parallel over the same events

Selection on angular distance

```
## EVENT SELECTION

algo      preselection
cmd       "ALL "              # to count all events
cmd       "nJET >= 6 "        # events with 6 or more jets
cmd       "MET < 100 "        # fully hadronic events should have small MET

algo     singlestep
preselection
cmd       "topchi2 + Wchi2 ~= 0 " # reconstruct two hadronic Ws in the event
cmd       "mWH1 [] 50 120 "
cmd       "mWH2 [] 50 120 "
cmd       "{WH1 , JET_-2 }dR > 0.6 "
cmd       "{WH2 , JET_-4 }dR > 0.6 "
histo     "mWHh1 , Hadronic W2 reco (GeV), 50, 50, 150, mWH1 "
histo     "mWHh2 , Hadronic W1 reco (GeV), 50, 50, 150, mWH2 "
histo     "mTopHh1, Hadronic top1 reco (GeV), 70, 0, 700, mTopH1 "
histo     "mTopHh2, Hadronic top2 reco (GeV), 70, 0, 700, mTopH2 "

algo     doublestep
preselection
cmd       "Wchi2 ~= 0 "
cmd       "topchi2 ~= 0 "     # reconstruct two hadronic Ws in the event
cmd       "mWH1 [] 50 120 "
cmd       "mWH2 [] 50 120 "
cmd       "{WH1 , JET_-2 }dR > 0.6 "
cmd       "{WH2 , JET_-4 }dR > 0.6 "
histo     "mWHh1 , Hadronic W1 reco (GeV), 50, 50, 150, mWH1 "
histo     "mWHh2 , Hadronic W2 reco (GeV), 50, 50, 150, mWH2 "
histo     "mTopHh1, Hadronic top1 reco (GeV), 70, 0, 700, mTopH1 "
histo     "mTopHh2, Hadronic top2 reco (GeV), 70, 0, 700, mTopH2 "
```

CutLang tools for writing an analysis:

- Predefined and user defined functions
- Comparison operators and thresholds
- Logical operators
- Ternary operator
- χ^2 minimisation
- Multiple event selection regions
- User defined variables
- Derived analysis object sets
- Histogramming

CutLang output:

- ROOT file with all algorithm results and analysis histograms in separate directories.
- Cutflows, event counts and selection efficiencies as text output.
- Surviving events in ntuple format (upcoming).

CutLang, the interpreter

- CutLang is written in C++ based on ROOT-6 classes.
- CutLang can utilize multiple ROOT input data formats such as
  - ATLAS & CMS Open Data, DELPHES, FCC, etc…
- CutLang is a run time interpreter, and requires no compiling.
- CutLang is available on MacOS and Linux platforms.
- External (user) functions can be downloaded and merged.
- New input data types can be integrated.
- Lex/Yacc based new parser is being beta-tested.

References

Source code, user manual and example root files are available at `https://cutlang.hepforge.org`
Reference paper at arXiv:1801.05727, Comput.Phys.Commun. 233 (2018) 215-236