Data analysis in high-energy physics

Search for top—anti-top quark resonances in proton-proton collisions with ATLAS data

CSU Tutorials 2020

Gabriel Palacino

Tuesday, April 21, 2020
Implementing Analysis Code

- AnalysisCode is package where you can implement your analyses, i.e., ZprimeAnalysis
- ZprimeAnalysis has three main running components
  - Initialization: everything that has to be done before looping over all the events (initialize(), histInitialize())
  - Execution: what is done every event. Basically the data analysis code (execute())
  - Finalization: what is done after looping over all the events (finalize(), histFinalize())
- ZprimeAnalysis is a class: you can implement your own functions members of ZprimeAnalysis or you can simply use what’s already there

ZprimeAnalysis.h

```
// these are the functions inherited from Algorithm
virtual EL::StatusCode setupJob (EL::Job& job);
virtual EL::StatusCode fileExecute ();
virtual EL::StatusCode histInitialize ();
virtual EL::StatusCode changeInput (bool firstFile);
virtual EL::StatusCode initialize ();
virtual EL::StatusCode execute ();
virtual EL::StatusCode postExecute ();
virtual EL::StatusCode finalize ();
virtual EL::StatusCode histFinalize ();
```

```
class ZprimeAnalysis : public EL::Algorithm
{
    // put your configuration variables here as public variables.
    // that way they can be set directly from CINT and python.
    public:
    // float cutValue;
    void getLeptons();
    void getJets();
    void getMET();
    void getEventInfo();
};
```
Implementing Analysis Code

- Often we create individual classes for each physics object
  - These classes hold a large amount of information such as MC generator level, detector information and reconstructed object info
  - Since we deal with several objects per event, we store them in "containers"
  - In our case containers are vectors
- We will use C++ structures within AnalysisCode instead of C++ classes for the physics objects
• We can use the new functions to construct the objects (structures) and store them in containers (std::vectors):

```cpp
void getLeptons();
void getJets();
void getMET();
void getEventInfo();
```

• Note that there is a structure for the event information:
  ‣ Although the event itself is not a physics object, we analyse the data on an event-by-event basis
  ‣ We need the event information

Each structure represents a physics object and carries at least its kinematic information using TLorentzVectors.

Physics objects of which more than one is expected can be stored in std::vectors.
The actual implementation of the new functions is done in the source file ZprimeAnalysis.cxx.

- This function will be called every event. Make sure you clear the container or you will end up storing the leptons from all events.
- Set the TLorentzVector
- Is it an electron or a muon
- Does it pass the Tight lepton identification criteria?
- Isolation information
- Fill the lepton container
Implementing Analysis Code

- The functions that fill the containers should be called at the beginning of `execute()`
Why implement structures (or classes) for analysis objects?

```cpp
for(unsigned int i = 0; i < lep_n; i++)
    if(lep_type[i]==11) n_el++; 
    if(lep_type[i]==13) n_mu++; 
if ((lep_flag[i] & 512)) continue;
    h_el->Fill(lv_pt[i]*GeV);
    if (lep_pt[i]<25) continue;
    h_el->Fill(lv_ptcone30[i]/lep_pt[i]);
    if (lep_ptcone30[i]/lep_pt[i] > 0.15) continue;
    nGoodlep++;
    goodlepIndex = i;
}

h_el_n->Fill(n_el);
    h_mu_n->Fill(n_mu);

if(nGoodlep==1){
    for(unsigned int i = 0; i < alljet_n; i++)
    h_jet->Fill(jet_pt[i]*GeV);
    if(jet_pt[i]<25) continue;
    h_jet->Fill(jet_eta[i]);
    if(fabs(jet_eta[i]) > 2.5) continue;
    h_jet->Fill(jet_jvf[i]);
    if(jet_pt[i]<50 && fabs(jet_eta[i]) < 2.4){
    if(jet_jvf[i]<0.5) continue;
    } 
    nGoodjet++;
    goodJetIndex.push_back(i); 
}
```
Event Information: trigger

- Many things can be stored in simple TH1 histograms: trigger information
Does the trigger select other processes?

$Z'(2500 \text{ GeV}) \rightarrow tt$

$WW \rightarrow l\nu qq$

$W^+ \nu l$

$W^– b q q'$

$Z$

\[\begin{array}{c}
\text{events processed} \\
\text{fail e/emu} \quad \text{pass e/emu} \quad \text{fail muon} \quad \text{pass muon} \quad \text{fail both} \quad \text{pass both}
\end{array}\]

\[\begin{array}{c}
\text{events processed} \\
\text{fail e/emu} \quad \text{pass e/emu} \quad \text{fail muon} \quad \text{pass muon} \quad \text{fail both} \quad \text{pass both}
\end{array}\]
Event selection

- The event has one good vertex (hasGoodVertex)
- Exactly one good lepton
  - Tight lepton ID
    - $p_T > 25$ GeV
    - Track isolation: $\text{lep}_\text{ptcone30}/\text{lep}_\text{pt} < 0.15$
    - Calorimeter isolation: $\text{lep}_\text{etcone20}/\text{lep}_\text{pt} < 0.15$
- At least four good jets
  - $p_T > 25$ GeV
  - $|\eta| < 2.5$
  - if $p_T < 50$ GeV and $|\eta| < 2.4$, JVF > 0.5
- At least one b-jets
  - MV1 output $\geq 0.7892$
- MET > 30 GeV
- Transverse mass of the leptonic W: $m_T(\text{lep}, \text{MET}) > 30$ GeV
Event selection

• The event has one good vertex (hasGoodVertex)

• Exactly one good lepton
  › Tight lepton ID
  › $p_T > 25$ GeV
  › Track isolation: $\text{lep}_\text{ptcone30}/\text{lep}_\text{pt} < 0.15$
  › Calorimeter isolation: $\text{lep}_\text{etcone20}/\text{lep}_\text{pt} < 0.15$

• At least four good jets
  › $p_T > 25$ GeV
  › $|\eta| < 2.5$
  › if $p_T < 50$ GeV and $|\eta|<2.4$, JVF > 0.5

• At least one b-jets
  › MV1 output $\geq 0.7892$

• MET $> 30$ GeV
• Transverse mass of the leptonic W: $m_T(\text{lep, MET}) > 30$ GeV
Event selection

- The event has one good vertex (hasGoodVertex)
- Exactly one good lepton
  - Tight lepton ID
    - $p_T > 25$ GeV
    - Track isolation: $\text{lep}_\text{ptcone30}/\text{lep}_\text{pt} < 0.15$
    - Calorimeter isolation: $\text{lep}_\text{etcone20}/\text{lep}_\text{pt} < 0.15$
- At least four good jets
  - $p_T > 25$ GeV
  - $|\eta| < 2.5$
  - if $p_T < 50$ GeV and $|\eta| < 2.4$, JVF $> 0.5$
- At least one b-jets
  - MV1 output $\geq 0.7892$
- MET $> 30$ GeV
- Transverse mass of the leptonic W: $m_T(\text{lep}, \text{MET}) > 30$ GeV
Event selection

- The event has one good vertex (hasGoodVertex)
- Exactly one good lepton
  - Tight lepton ID
  - $p_T > 25$ GeV
  - Track isolation: $\text{lep}_\text{ptcone30}/\text{lep}_\text{pt} < 0.15$
  - Calorimeter isolation: $\text{lep}_\text{etcone20}/\text{lep}_\text{pt} < 0.15$
- At least four good jets
  - $p_T > 25$ GeV
  - $|\eta| < 2.5$
  - if $p_T < 50$ GeV and $|\eta|<2.4$, JVF > 0.5
- At least one b-jets
  - MV1 output $\geq 0.7892$
- MET $> 30$ GeV
- Transverse mass of the leptonic W: $m_T(\text{lep}, \text{MET}) > 30$ GeV
Event selection

- The event has one good vertex (hasGoodVertex)
- Exactly one good lepton
  - Tight lepton ID
    - $p_T > 25$ GeV
    - Track isolation: $\text{lep}_\text{ptcone30}/\text{lep}_\text{pt} < 0.15$
    - Calorimeter isolation: $\text{lep}_\text{etcone20}/\text{lep}_\text{pt} < 0.15$
- At least four good jets
  - $p_T > 25$ GeV
  - $|\eta| < 2.5$
  - if $p_T < 50$ GeV and $|\eta|<2.4$, JVF $> 0.5$
- At least one b-jets
  - MV1 output $\geq 0.7892$
- MET $> 30$ GeV
- Transverse mass of the leptonic W: $m_T(\text{lep}, \text{MET}) > 30$ GeV
Event selection

- The event has one good vertex (hasGoodVertex)
- Exactly one good lepton
  - Tight lepton ID
    - $p_T > 25$ GeV
    - Track isolation: $\text{lep}_\text{ptcone30}/\text{lep}_\text{pt} < 0.15$
    - Calorimeter isolation: $\text{lep}_\text{etcone20}/\text{lep}_\text{pt} < 0.15$
- At least four good jets
  - $p_T > 25$ GeV
  - $|\eta| < 2.5$
  - if $p_T < 50$ GeV and $|\eta|<2.4$, JVF > 0.5
- At least one b-jets
  - MV1 output $\geq 0.7892$
- MET $> 30$ GeV
- Transverse mass of the leptonic W: $m_T(\text{lep}, \text{MET}) > 30$ GeV
Event selection

- The event has one good vertex (hasGoodVertex)
- Exactly one good lepton
- At least four good jets
- At least one b-jets
- MET > 30 GeV
- Transverse mass of the leptonic W: \( m_T(\text{lep, MET}) > 30 \text{ GeV} \)

You can store this in a TH1 as it was done for the trigger information!
Event selection: signal vs. background

- Although the trigger would select event from both processes, the event selection focuses on the signal.
- Some backgrounds won’t be rejected as nicely as WW. For instance, tt.
Event selection efficiency

- Can be defined in terms of absolute or relative efficiency
- **Absolute event selection efficiency**
  - Number of events that passed a given cut $X$ with respect to the total (initial) number of events
  - $\varepsilon_X = \frac{N_X}{N_{\text{total}}}$
- **Relative event selection efficiency**
  - Relative to another cut: often the previous
  - $\varepsilon_X = \frac{N_X}{N_{X-1}}$
- Applying a cut can be considered a binomial process
  - The only possibilities are "pass" or "fail"
- Let’s assume that the uncertainty on the efficiency can be described using binomial errors
  - $\delta \varepsilon = \sqrt{\varepsilon (1-\varepsilon)/N}$
Homework 2 (due Tuesday, April 28)

• Make the object selection plots as in slides 10 to 15 in here for ttbar (9 plots!)
  • Each plot has to be made before the cut on that variable has been applied

• Make an event selection efficiency table for Z’(2500 GeV), ttbar and WW
  ‣ Relative and absolute efficiencies
  ‣ Include binomial errors
Thanks!