

h TopCommonDileptonEventSaver.h 5.98 KiB

```

1  #ifndef TOPCOMMONDILEPTONEVENTSAVER_H_
2  #define TOPCOMMONDILEPTONEVENTSAVER_H_
3
4  #include "TopAnalysis/EventSaverFlatNtuple.h"
5  #include "TRandom3.h"
6  #include "TH2.h"
7  #include "TopEventSelectionTools/PlotManager.h"
8  #include "TopCommonDilepton/TTbarNNLORecursiveRew.h"
9
10 class TopCommonDileptonEventSaver : public top::EventSaverFlatNtuple {
11
12 public:
13
14     ///-- Default - so root can load based on a name --///
15     TopCommonDileptonEventSaver();
16
17     ///-- Default - so we can clean up --///
18     //~TopCommonDileptonEventSaver();/=0;
19
20     ///-- Exposition of base class initializers (to avoid [-Woverloaded-virtual] hidden warnings --///
21     using top::EventSaverFlatNtuple::initialize;
22
23     ///-- Run once at the start of the job --///
24     void initialize(std::shared_ptr<top::TopConfig> config, TFile* file, const std::vector<std::string>& extraB
25
26     ///-- For parton level objects --///
27     void initializePartonLevelEvent();
28     void addCustomPartonLevelVariables();
29     void saveTruthEvent(); //naming enforced by AnalysisTop
30
31     ///-- For particle level object --///
32     void initializeParticleLevelBranches();
33     void addCustomParticleLevelVariables();
34     void saveParticleLevelEvent(const top::ParticleLevelEvent& plEvent); //naming enforced by AnalysisTop
35
36     ///-- Run for every event (in every systematic) that needs saving --///
37     void initializeRecoLevelBranches();
38     void addCustomRecoLevelVariables(const std::vector<std::string>& extraBranches);
39     void saveEvent(const top::Event& event); //naming enforced by AnalysisTop
40
41     ///-- Observable calculating functions --///
42     float cos_theta_helicity(TLorentzVector top, TLorentzVector parent_t, TLorentzVector ttbar, TLorentzVector
43     float cos_theta_transverse(TLorentzVector top, TLorentzVector parent_t, TLorentzVector ttbar, TLorentzVecto
44     float cos_theta_raxis(TLorentzVector top, TLorentzVector parent_t, TLorentzVector ttbar, TLorentzVector lep
45     float cos_phi(TLorentzVector lpos, TLorentzVector lneg, TLorentzVector top, TLorentzVector tbar, TLorentzVe
46
47     //skip saving nominal branch - useful for splitting ntuples by systematics if need be
48     int isBranchStored(top::TreeManager const *treeManager, const std::string &variableName);
49     bool m_skipNominal;
50
51     bool m_doNeutrinoWeighter;
52     bool m_doEllipseMethod;
53     bool m_doSonnenchein;
54     bool m_store_weights_reco;
55     bool m_store_weights_particle;
56     bool m_store_weights_parton;
57
58     TTbarNNLORecursiveRew* m_ttbarNNLO_Reweighter;
59
60
61 private:
62
63     ///-- Steering --///
64     bool is_MC;
65     std::shared_ptr<top::TopConfig> m_config;
66
67
68     ///-- Custom reco level variables --///
69     bool m_passSelection;
70
71     std::vector<float> m_lep_pt;
72     std::vector<float> m_lep_eta;
73     std::vector<float> m_lep_phi;
74     std::vector<float> m_lep_e;
75     std::vector<int> m_lep_pdgid;
76     bool m_fakeEvent;
77     std::vector<int> m_lep_true_type;

```

```

78 std::vector<int>    m_lep_true_origin;
79 std::vector<char>  m_lep_true_isPrompt;
80 std::vector<char>  m_lep_true_isChargeFl;
81 std::vector<int>    m_lep_true_pdgid;
82 float m_dilep_m;
83 std::vector<int> m_jet_isbtagged_DL1r_85;
84 std::vector<float> m_ljet_pt;
85 std::vector<float> m_ljet_eta;
86 std::vector<float> m_ljet_phi;
87 std::vector<float> m_ljet_e;
88 std::vector<float> m_bjet_pt;
89 std::vector<float> m_bjet_eta;
90 std::vector<float> m_bjet_phi;
91 std::vector<float> m_bjet_e;
92 float m_met_met;
93 float m_met_phi;
94 float m_t_pt;
95 float m_t_eta;
96 float m_t_phi;
97 float m_t_m;
98 float m_tbar_pt;
99 float m_tbar_eta;
100 float m_tbar_phi;
101 float m_tbar_m;
102 float m_ttbar_pt;
103 float m_ttbar_eta;
104 float m_ttbar_phi;
105 float m_ttbar_m;
106 int m_top_reco_method;
107 float m_cosphi;
108 float m_cos_kaxis_p;
109 float m_cos_kaxis_m;
110 float m_cos_naxis_p;
111 float m_cos_naxis_m;
112 float m_cos_raxis_p;
113 float m_cos_raxis_m;
114 std::vector<float> m_generator_weights;
115
116
117 ///-- Custom particle level variables --///
118 std::vector<float> m_particle_lep_pt;
119 std::vector<float> m_particle_lep_eta;
120 std::vector<float> m_particle_lep_phi;
121 std::vector<float> m_particle_lep_m;
122 std::vector<float> m_particle_lep_pdgid;
123 std::vector<float> m_particle_jet_pt;
124 std::vector<float> m_particle_jet_eta;
125 std::vector<float> m_particle_jet_phi;
126 std::vector<float> m_particle_jet_m;
127 std::vector<float> m_particle_jet_btagged;
128 float m_particle_met_ex;
129 float m_particle_met_ey;
130 float m_particle_met_phi;
131 float m_particle_met_met;
132 float m_particle_t_pt;
133 float m_particle_t_eta;
134 float m_particle_t_phi;
135 float m_particle_t_m;
136 float m_particle_tbar_pt;
137 float m_particle_tbar_eta;
138 float m_particle_tbar_phi;
139 float m_particle_tbar_m;
141 float m_particle_ttbar_eta;
142 float m_particle_ttbar_phi;
143 float m_particle_ttbar_m;
144 float m_particle_cosphi;
145 float m_particle_cos_kaxis_p;
146 float m_particle_cos_kaxis_m;
147 float m_particle_cos_naxis_p;
148 float m_particle_cos_naxis_m;
149 float m_particle_cos_raxis_p;
150 float m_particle_cos_raxis_m;
151 std::vector<float> m_particle_generator_weights;
152
153
154 ///-- Custom parton level variables --///
155 float m_parton_weight_nnlo;
156 float m_parton_l_pt;
157 float m_parton_l_eta;

```

```
158 float m_parton_l_phi;
159 float m_parton_l_m;
160 float m_parton_l_pdgid;
161 float m_parton_lbar_pt;
162 float m_parton_lbar_eta;
163 float m_parton_lbar_phi;
164 float m_parton_lbar_m;
165 float m_parton_lbar_pdgid;
166 float m_parton_t_afterFSR_pt;
167 float m_parton_t_afterFSR_eta;
168 float m_parton_t_afterFSR_phi;
169 float m_parton_t_afterFSR_m;
170 float m_parton_tbar_afterFSR_pt;
171 float m_parton_tbar_afterFSR_eta;
172 float m_parton_tbar_afterFSR_phi;
173 float m_parton_tbar_afterFSR_m;
174 float m_parton_ttbar_pt;
175 float m_parton_ttbar_eta;
176 float m_parton_ttbar_phi;
177 float m_parton_ttbar_m;
178 float m_parton_cosphi;
179 float m_parton_cos_kaxis_p;
180 float m_parton_cos_kaxis_m;
181 float m_parton_cos_naxis_p;
182 float m_parton_cos_naxis_m;
183 float m_parton_cos_raxis_p;
184 float m_parton_cos_raxis_m;
185 bool m_parton_is_tau;
186 std::vector<float> m_parton_generator_weights;
187
188
189 //ClassDef(top::TopCommonDileptonEventSaver, 1);
190 ClassDef(TopCommonDileptonEventSaver, 0);
191 };
192
193 //}
194
195 #endif
196
```