Table of Contents

Introduction

This page briefly describes the analysis framework used for studying variable-R re-clustering in the context of Vector-like Quarks.

In this framework, we inherit base classes from AnalysisTop to build flat ntuples that contain corrected objects (+ systematics) to be used to produce final results.

The re-clustering code is here: github_RC
The AnalysisTop info is here: AnalysisTop_info
You can find the HQTVLQAllHadTools code here: svnweb

The current version (00-00-03) is compatible with AnalysisTop-2.3.37 (as of 13 December 2015).

Below, this Twiki describes briefly the changes in tags for HQTVLQAllHadTools, and how to access the tool from svn (and its dependencies). AnalysisTop must be setup first.

The structure of HQTVLQAllHadTools:

- cmt
  - Makefile
- HQTVLQAllHadTools
  - Header files (for corresponding c++ files)
- python
  - grid scripts (to be used with the AnalysisTop tools for submitting jobs to grid)
- Root
  - c++ scripts (event savers, truth information, tools loader, large-R jet builder, object loader)
- share
  - text files (config file, GRL, PRW, lists of files, etc.)

AnalysisTop-2.3.37

First, we need to setup AnalysisTop. To do this, we will initialize the environment and checkout the AnalysisTop code. Once we have that, we can add our code & other dependencies.

```
# Setup AnalysisTop
mkdir AnalysisTop-2.3.37 && cd AnalysisTop-2.3.37
```

Now, make a file called 'setup.sh' and put the following lines of code in it:

```
export CERN_ACCOUNT=dmarley # change to your CERN username
export RUCIO_ACCOUNT=$CERN_ACCOUNT

setupATLAS
localSetupPandaClient
localSetupRucioClients
localSetupPyAMI
```

Then, execute the script

```
source setup.sh
```

And initialize AnalysisTop

```
rcSetup Top,2.3.37 && rc find_packages && rc compile
```

After the first time you setup AnalysisTop, when you log-in to lxplus (or your cvmfs environment of choice) you only need to type:

```
source setup.sh
rcSetup
```

and everything will be setup for you.

Now that the environment is initialized, let's add our custom code and other dependencies

```
#HQTVLQAllHadTools code
svn co svn+ssh://$CERN_ACCOUNT@svn.cern.ch/reps/atlasphys-top/Physics/Top/PhysAnalysis/Run2/HQTVL
# Jet Uncertainties (for large-R jet uncertainties)
rc checkout_pkg atlasoff/Reconstruction/Jet/JetUncertainties/tags/JetUncertainties-00-09-35 JetUn
# xAOD re-clustering code
git clone https://github.com/kratsg/xAODJetReclustering.git
cd xAODJetReclustering
git checkout tags/v3.141
cd ..
# Boosted Top Tagging
svn co svn+ssh://$CERN_ACCOUNT@svn.cern.ch/reps/atlasoff/Reconstruction/Jet/BoostedJetTaggers/tag
```

Now that we've added our packages, we need to compile everything one last time.

```
rc find_packages && rc compile
```

And we're all set! You can run this over any DAOD you would like (please use p-tags > 2452, and derivations other than EXOT4 and TOPQ* aren't guaranteed to work).

Currently getting an error with a file from here (under investigation):

mc15_13TeV:mc15_13TeV.341177.aMcAtNloHerwigppEvtGen_UEEE5_CTEQ6L1_CT10ME_ttH125_inc.merge.DA

Here is a description of the current tags (use the latest tag!):

- 00-00-04
  - Use! (14 December 2015)
  - Bug fixes and code cleaning for variable-R re-clustering
- 00-00-03
  - Do not use!
  - Added bug fixes to variable-R re-clustering code
- 00-00-02
  - Do not use!
  - Added selection to do trigger studies
    - Empty selection that (only) saves the trigger decisions
- 00-00-01
  - Do not use!
  - Initial tag of the code -- largely a copy/paste of HQTVLQWbXTools

Running HQTVLQAllHadTools

With the environment setup, we can now run over samples and produce flat ntuples with extra variables that are saved in HQTVLQAllHadTools.

To do this, simply type the command:

```
top-xaod HQTVLQAllHadTools/share/vlqwbx-alljets-cuts-varR.txt file.txt
```

where file.txt is a text file that contains the full path to the files you want to run over (see attached file for example).

Your output will a root file called output.root. In that file, you can check to see if anything and everything you wanted is saved correctly. After testing locally in your cvmfs environment, I then suggest you submit jobs to the grid (if needed).

There may be simple changes you want for running the code (that don't affect the code) that can be done just using the config file. Please open this file and see that everything is as you want it to be. The config file is designed for an all-hadronic analysis, and the cutflow and the bottom of the file reflects this.
If you're just testing things, it may be good to set the variable NEvents to some small number > 0 (e.g., 500). Additionally, if you want to run over systematics (or don't) you can change the value for the variable Systematics to Nominal (no systematics) or All (for everything) or even just a comma-separated list of systematics you want to run over.

More information on the config file can be found in the TopxAODStartGuide. (You can even change jet pT, eta cuts; lepton definitions; and much more!)

Variable-R Re-clustering

To perform variable-R re-clustering in AnalysisTop, a lot of "hacks" were performed. Most of this (I believe) is just due to the way AnalysisTop works, and what features users have access to. Of course, I could have missed a lot of things and be doing way more work than needed. Please don't hesitate to let Dan know if there are improvements!

In any case, to do this type of re-clustering, we have to initialize the re-clustering tool for each systematic that you're running over (if just the Nominal case, then only 1 tool is initialized). This is all done in the initialize() function of Root/VLQEventSaverFlatNtuple_varR.cxx.

Once the tool is initialized, and loops over the events begin, we have to make a custom container of jets and store them in the TStore -- the saveEvent() function. This is a bit of a headache, as the only keys that exist in the store appear to be for the jets we don't want (no overlap removal, no JVT cut, no calibrations, etc.). Thus, we make our own container for each kind of event (special containers for the Nominal sample, systematics that affect jets, and 'loose' selection on data). For systematics that don't necessarily affect jets, the nominal jet container is used.

Additionally, this was designed for VLQ events, so we do truth-matching via ghost-association with VLQ partons and their decays (if you run on a different sample, this won't necessarily work -- contact Dan to discuss fixes). The VLQ partons are added to the containers so we can find which jets they were clustered in. A lot of this is hard-coded, and not easily fixed for other signal types (but it's possible to change).

Finally, we also save the subjets in each re-clustered jet. This means that you can access the 4-vector of the subjet to help better understand the jet, or design some substructure variable to improve S/B.

Modifying HQTVLQAllHadTools

If you want to modify the code (and you probably do), here are a few examples on how to do that.

Initially, the code was developed to perform the following 4 tasks:

- Save large-R jet information (Anti-kT R=1.0 trimmed; standard collection) (top and W tagging)
- Save re-clustered jet information (fixed radius)
- Save variable-R re-clustered jet information for = 2*m_top
- Save variable-R re-clustered jet information for = 2*m_Z

For the standard large-R jet information, most of this is done in the file Root/LargeRJetMC15.cxx and then saved in the file Root/VLQEventSaverFlatNtuple_varR.cxx.

All of the re-clustering bits are done in the event saver (VLQEventSaverFlatNtuple_varR). It is setup this way because of how AnalysisTop is configured (if you find a better way, please feel free to share!). If you want to modify the variable-R re-clustering (maybe you only want 1 kind of re-clustering, or you're looking at Higgs or W, then the following examples show you how to modify things.

Modify variable-R re-clustering

Here are some examples on how to modify things.

Reduce Number of Variable-R Re-clustering Instances

Currently, the tool is configured for two kinds of variable-R re-clustering. To change that, you can change one of the booleans in the constructor to false, and you will only have 1 kind of re-clustering. Example:

```
m_rc10 = true; // do re-clustering; change to false if not needed
m_varR_top = true; // do variable-R re-clustering at top mass; change to false if not needed
m_varR_Z = true; // do variable-R re-clustering at Z mass; change to false if not needed
```

make sure that the variable-R re-clustering you keep has the mass scale you want!

Change the Mass Scale (& Other Re-clustering Parameters)

To change the kinds of particles you're running over (or just the scale), I suggest you simply access the event saver header file (HQTVLQAllHadTools/VLQEventSaverFlatNtuple_varR.h) and add the masses you want (the top, W, and Z masses are already there). Then, in the .cxx file (Root/VLQEventSaverFlatNtuple_varR.cxx), change the initialization of the variable-R re-clustering tools in the initialize() function. Each tool is listed twice because if you're running on data, AnalysisTop automatically produces a nominal and loose selection. Otherwise, the tool is initialized in the loop for each systematic (if just Nominal, it is only initialized once). Please check all parameters in the initializations of each re-clustering and make sure the parameters are what you want. List of Parameters:

```
// example code for the variable-R re-clustering at top mass
JetReclusteringTool* varR_top_tool = new JetReclusteringTool("VarR_top_RCJet_tool"+treeName.secon
 top::check(varR_top_tool->setProperty("VariableRMassScale", 2*m_top_mass), "Failed Mass Scale Va
 top::check(varR_top_tool->setProperty("InputJetContainer",   m_customInputJetContainer), "Failed
 top::check(varR_top_tool->setProperty("OutputJetContainer", m_varR_top_outputJetContainer+treeNa
 top::check(varR_top_tool->setProperty("ReclusterRadius", 1.5), "Failed Recluster Radius Variable
 top::check(varR_top_tool->setProperty("VariableRMinRadius", 0.4), "Failed Min Radius Variable R
 top::check(varR_top_tool->setProperty("RCJetPtFrac", m_ptFrac), "Failed ptfrac Variable R top to
 top::check(varR_top_tool->setProperty("RCJetPtMin", m_ptMin_rc),"Failed ptmin Variable R top too
 top::check(varR_top_tool->setProperty("InputJetPtMin", 0), "Failed InputJetPtMin varR top tool")
```

```
 top::check(varR_top_tool->initialize(), "Failed to initialize Variable R top tool"); // initiali
```

(Note: "top::check()" is the function that checks return values)

If you want to add information, for example substructure variables or b-tagging weights on the subjets, these are already shown in the fixed R re-clustering:

```
// substructure
static SG::AuxElement::ConstAccessor<float> RCSplit12("Split12"); // first splitting scale
static SG::AuxElement::ConstAccessor<float> RCSplit23("Split23"); // second splitting scale
static SG::AuxElement::ConstAccessor<float> RCD2("D2"); // energy correlation function
static SG::AuxElement::ConstAccessor<float> RCC2("C2"); // energy correlation function
```

The variable names are then linked to substructure values. You can access and save the values like this:

```
m_reclusteredW_d12[i] = (RCSplit12.isAvailable(*rc_jet)) ? RCSplit12(*rc_jet) : -999.;
```

where rc_jet is the re-clustered jet of interest.

For b-tagging information:

```
double mvx;
const xAOD::Jet* subjet(nullptr);
const xAOD::BTagging* btag(nullptr);
// loop over subjets
for(auto rc_jet_subjet : rc_jet->getConstituents()){
 subjet = static_cast<const xAOD::Jet*>(rc_jet_subjet->rawConstituent());
 btag = subjet->btagging();
 if (btag)
  btag->MVx_discriminant("MV2c20",mvx); // code from TopAnalysis/Root/EventSaverFlatNtuple.cxx
 else
  mvx = -999.;
```

And the subjet will be decorated with its b-tag weight.

VLQ Derivations

If you want to run over VLQ derivations (that are all-hadronic and can be processed with this code), please have a look at the EXOT7 samples produced with the p-tag 2454:

```
rucio list-dids mc15_13TeV:*302*DAOD_EXOT7*p2454
```

We were only interested in the TTS_M* samples (singlet TT pair-production), may investigate the others soon.

Further questions or comments can be directed to Dan.

-- DanielEdisonMarley - 2015-12-14

This topic: Main > VLQRun2_AllHad_varR
Topic revision: r2 - 2015-12-14 - DanielEdisonMarley