Search for exotic resonances decaying to WW - semileptonic final state with lepton + neutrino + merged jet (lvJ).

EXO WW semileptonic analysis - run2

Step 1: Dumper - from miniAOD to flat trees (no selection)

The first step of the analysis is to run on miniAOD samples, and produce some flat trees. No selection are applied at this level: the output ntuples just contain the quantities needed for the analysis. Here is the code: https://github.com/lbrianza/RA2_2014

Download and compile for CMSSW >= 8_0_X

Instruction:

```
cmsrel CMSSW_8_0_11
cd CMSSW_8_0_11/src
cmsenv
git cms-init
git cms-merge-topic -u cms-met:CMSSW_8_0_X-METFilterUpdate
git clone git@github.com:lbrianza/RA2_2014.git AllHadronicSUSY
cd AllHadronicSUSY/
git checkout 80x_devel
cd ..
git clone -b Analysis74X git@github.com:cms-edbr/ExoDiBosonResonancesRun2.git ExoDiBosonResonance
scram b -j8
cd AllHadronicSUSY/
```

Download and compile for CMSSW >= 7_6_1

Instruction:

```
export SCRAM_ARCH=slc6_amd64_gcc491
cmsrel CMSSW_7_6_1
cd CMSSW_7_6_1/src
cmsenv
git cms-init
git clone git@github.com:lbrianza/RA2_2014.git AllHadronicSUSY
cd AllHadronicSUSY/
git checkout 76x_devel
cd ..
git clone -b Analysis74X git@github.com:cms-edbr/ExoDiBosonResonancesRun2.git ExoDiBosonResonance
scram b -j8
cd AllHadronicSUSY/
```

Running the code on CMSSW 8_0_X

How to run on MC:

```
 cmsRun TreeMaker/test/runMakeTreeFromMiniAOD_cfg.py global_tag=80X_mcRun2_asymptotic_2016_v3 MC=
```

DATA:

```
cmsRun TreeMaker/test/runMakeTreeFromMiniAOD_cfg.py global_tag=80X_dataRun2_Prompt_v8 MC=False de
```

Running the code on CMSSW 7_6_X

How to run on MC:

```
cmsRun TreeMaker/test/runMakeTreeFromMiniAOD_cfg.py global_tag=76X_mcRun2_asymptotic_v12 MC=True
```

On data

```
cmsRun TreeMaker/test/runMakeTreeFromMiniAOD_cfg.py global_tag=76X_dataRun2_v15 MC=False debug=Fa
```

Run on CRAB3

How to set CRAB3 environment:

```
source /cvmfs/cms.cern.ch/crab3/crab.sh
voms-proxy-init
```

How to run on CRAB3, MC and data (76x):

```
python TreeMaker/test/prod_flatTrees_run2_data_76x.py
python TreeMaker/test/prod_flatTrees_MC76x_25ns_miniAODv2.py
```

List of Samples Run2 - MC76x (incomplete - need to check xsecs)

Signal cross sections and BR:
https://twiki.cern.ch/twiki/bin/viewauth/CMS/ExoDiBosonResonancesRun2#Cross_sections

Step 2: WWTree: Produce ntuples for WW semileptonic final state (preselection)

Run on the flat trees created after step1, and produce the final ntuples that will be used in the analysis. The Ntuplizer is here: https://github.com/lbrianza/WWAnalysisRun2

Instructions for 76x:

```
cd CMSSW_7_6_1/src
cmsenv
git clone -b 76x_devel git@github.com:lbrianza/WWAnalysisRun2
cd WWAnalysisRun2/
make
```

(ignore the warning on the make command)

How to run in local: put the ReducedSelection.root file produced at the step1 in the WWAnalysisRun2 folder, then run

```
 python python/produceWWNtuples.py -o WWTree_test -l mu --ismc 1 -trig 0 --isLocal 1
```

If you want instead to run directly on the files stored in eos:

```
python python/produceWWNtuples.py -n BulkGraviton1000 -o WWTree_BulkGraviton1000 -l mu -w 0.02049
```

Now you have the final tree for the BulkGraviton1000 sample.

There is also a script (python/submit_on_lxbatch_MINIAODv2.py) to run one job for each sample. To produce all the ntuples, simply run with:

```
python python/submit_on_lxbatch_MINIAODv2.py
```

The ntuples will be saved in the output/ folder.

Big ntuples in eos

Before to run the step2 for the all samples, you must re-order a bit the folders containing the root files produced at the step1. In order to do that, go into the directory where you have all the subfolders with the different samples, then run the following three scripts:

```
sh move_file_eos.sh
sh remove_folders_eos.sh
sh split_folders_eos.sh
```

Folders with the ntuple

- Folder with the flat trees (big ntuples - 80x) for run2: /store/caf/user/lbrianza/WWReducedTree_run2/
- Folder with the WWTrees (reduced ntuples) for run2 - 2016 data: /afs/cern.ch/user/l/lbrianza/work/public/WWTree_9jun_80x
- Folder with the WWTrees (reduced ntuples) for the low-mass analysis - 2015 data: /afs/cern.ch/user/l/lbrianza/work/public/WWTree_17feb_jecV7_lowmass/

Step 3: Control plots

Here the instruction to produce control plots for the relevant distributions. First, you must have already produced the ntuples (step2) for the all samples. Once you have all the ntuples, you can download and run the code for the control plots.

Instructions:

```
cmsrel CMSSW_7_1_5
cd CMSSW_7_1_5/src
cmsenv
git clone https://github.com/lbrianza/CVS
cd CVS/VBFHWWlnuJ/
cd NtuplePackage/
source scripts/setup.sh
make; make exe
cd ../KinFitter/
source scripts/setup.sh
make; make exe
cd ..
source scripts/setup.sh
make; make exe
```

NB: at some point, compilation can stop and rootcint shell opens. Press '.q' and go ahead.If it doesn't work, press again '.q' and go ahead.

To run:

```
./bin/DataMCComparisonPlot.exe cfg/DataMCComparison_InputCfgFile/Run2_DataMCComparisonPlot_80x_mu
```

Four files are mandatory in order to produce the control plots:

- Run2_DataMCComparisonPlot_80x_mu.cfg: main cfg file -> here you need to specify the input folder with the ntuples and other stuff such as lumi, reweighting factors, etc..
- Run2_SampleList_80x.txt: it contains the sample list

- Run2_CutList_76x_mu.txt: list of cuts you want to apply
- Run2_Variables_76x.txt: list of variables you want to plot the distributions

Step 4: Run the analysis, produce datacards and extract the limit

Make the analysis: fit on signals, on backgounds, fit on data and extraction of W+jets with alpha method, and extract the limit.

How to run the analysis

Instructions in order to set the environment:

```
cmsrel CMSSW_5_3_13
cd CMSSW_5_3_13/src
cmsenv
git clone https://github.com/lbrianza/EXOVVFitter
cd EXOVVFitter
scramv1 b clean; scramv1 b
```

Before running the code:

- put your ntuples in a folder called WWTree_mu or WWTree_el
- you must add the ntuples containing the same background (e.g. all the single top samples, as s-channel, t-channel, tW-channel.. into a single STop sample). In order to do that, you can use the hadda_mu.sh and hadda_el.sh scripts contained in the repository used to produce the small ntuples at the step2:

https://github.com/lbrianza/WWAnalysisRun2/blob/76x_devel/hadda_mu.sh
https://github.com/lbrianza/WWAnalysisRun2/blob/76x_devel/hadda_el.sh

---

If you modify one of the libraries inside the PDF/ folder, you must recompile it with the following commands: (fo instance, if you modify Util.cxx)

```
cd PDFs/
root -l
gSystem->AddIncludePath("-I$ROOFITSYS/include")
gROOT->ProcessLine(".L Util.cxx+")
```

Now you can run the analysis, example of command:

```
python run-all.py --batchMode --channel mu --jetalgo PuppiAK8_jet_mass_so -s BulkGraviton
```

Look inside the run-all.py file in order to understand the various options.

After that, you will have some folders with plots (e.g: plots_mu_HP) and folders with datacards (e.g: cards_mu_HP).

Extract the limit

Instructions in order to set the environment:

```
cmsrel CMSSW_7_1_5
cd CMSSW_7_1_5/src
cmsenv
git clone https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit.git HiggsAnalysis/CombinedL
cd HiggsAnalysis/CombinedLimit
git fetch origin
```

```
git checkout v5.0.3   # try v5.0.1 if any issues occur
scramv1 b clean; scramv1 b
cd ../..
git clone https://github.com/lbrianza/boostedWWAnalysis
cd boostedWWAnalysis
python Automatic_Setup.py
```

Run on the datacards and produce the limit:

Separate channels:

```
python runLimits_run2exo_new.py -b --computeLimits --channel mu --datacardDIR cards_mu_ExpN_HP/ -
python runLimits_run2exo_new.py -b --computeLimits --channel el --datacardDIR cards_el_ExpN_HP/ -
python runLimits_run2exo_new.py -b --computeLimits --channel mu --datacardDIR cards_mu_ExpN_LP/ -
python runLimits_run2exo_new.py -b --computeLimits --channel el --datacardDIR cards_el_ExpN_LP/ -
```

For the combination, create a new folder (e.g: cards_em). Copy all the datacard folders into this new one:

```
cp -r cards_mu* cards_em/
cp -r cards_el* cards_em/
```

then, combine cards using:

```
sh combCards.sh
```

Now you are ready for the combination:

```
python runLimits_run2exo_new.py -b --computeLimits --channel em --datacardDIR cards_em --makeSMLi
```

Compute p-value, injecting a signal with mu=1:

```
python runLimits_run2exo_new.py --computeLimit --computePvalue 1 --datacardDIR cards_em/ --channe
```

Bias tests

With the following command, you can run the "auto-bias test": This means, for each toy you generate a dataset injecting some signal (the amount of signal is given by the --injectSingalStrenght option, where 1 means for instance you are injecting a signal with xsec equal to 1*value in the datacard), and where the background is generated using the fitting function in the workspace. Then these datasets are re-fitted with the same functions, and hadding the N output you can look at the distribution of the fitted mu

```
python runLimits_run2exo_new.py --computeLimits --datacardDIR cards_em/ --channel em --massPoint
```

"cross-bias toys":

With the following commands instead you generate toys using one set of fitting functions and you fit them with another one. In order to do that, you need to generate two set of datacards, using a different fitting function.

```
 python runLimits_run2exo_new.py --computeLimits --generateOnly --datacardDIR cards_SET1/ --chann
mv cards_SET1/cards_SET1 cards_SET2/
python runLimits_run2exo_new.py --computeLimits --maximumLikelihood --datacardDIR cards_17feb_SET
```

Again, hadding the output of the toys, you can look at the distribution of the fitted mu to see if there is a bias.

W-tagger scale factors

```
python wtagger_doFit_class_run2exo.py -c mu -b --fitwtaggersim
```

electron and muons together:

```
python wtagger_doFit_class_run2exo_em.py -c mu -b --fitwtaggersim
```

Other: Calculate integrated luminosity of a dataset

To do that, just write the dataset name into get_json.sh Then, do:

```
 sh get_json.sh json.txt
python lcr2.py -i json.txt
```

In json.txt, you will have the corresponding json file for that dataset.

Other: Calculate integrated luminosity of a json file

```
mkdir $HOME/.local

export PATH=$HOME/.local/bin:/afs/cern.ch/cms/lumi/brilconda-1.0.3/bin:$PATH

pip install --install-option="--prefix=$HOME/.local" brilws

brilcalc lumi --normtag ~lumipro/public/normtag_file/OfflineNormtagV2.json -i /afs/cern.ch/cms/CA

filterJSON.py --min 256630 /afs/cern.ch/cms/CAF/CMSCOMM/COMM_DQM/certification/Collisions15/13TeV
```

How to generate Pile-up file

```
pileupCalc.py -i /afs/cern.ch/cms/CAF/CMSCOMM/COMM_DQM/certification/Collisions16/13TeV/Cert_2710
```

OBSOLETE INSTRUCTIONS

Download and compile for CMSSW >= 7_4_7

Instruction:

```
export SCRAM_ARCH=slc6_amd64_gcc491
cmsrel CMSSW_7_4_12
cd CMSSW_7_4_12/src
cmsenv
git cms-init
git cms-addpkg CommonTools/PileupAlgos
git remote add nhan-remote https://github.com/nhanvtran/cmssw.git
git fetch nhan-remote puppi-bugfix-for-miniaod:puppi-bugfix-for-miniaod
git merge puppi-bugfix-for-miniaod
git clone git@github.com:lbrianza/RA2_2014.git AllHadronicSUSY
cd AllHadronicSUSY/
git checkout 74x_newEleId
cd ..
git clone -b Analysis74X git@github.com:cms-edbr/ExoDiBosonResonancesRun2.git ExoDiBosonResonance
scram b -j8
cd AllHadronicSUSY/
```

Download and compile for CMSSW < 7_4_7

Instruction:

```
cmsrel CMSSW_7_4_3
cd CMSSW_7_4_3/src
cmsenv
git cms-init
git cms-merge-topic 9003
git cms-addpkg CommonTools/PileupAlgos
git remote add sharper https://github.com/Sam-Harper/cmssw.git
git fetch sharper VIDHEEPV60:VIDHEEPV60
git cherry-pick e9419e5 #get HEEP V6.0 in the local area
git remote add nhan-remote https://github.com/nhanvtran/cmssw.git
git fetch nhan-remote puppi-bugfix-for-miniaod:puppi-bugfix-for-miniaod
git merge puppi-bugfix-for-miniaod
git clone https://github.com/lbrianza/RA2_2014 AllHadronicSUSY
cd AllHadronicSUSY/
git checkout 74x_newEleId
cd ..
git clone -b Analysis74X git@github.com:cms-edbr/ExoDiBosonResonancesRun2.git ExoDiBosonResonance
scram b -j8
cd AllHadronicSUSY/
```

Running the code on CMSSW 7_4_X

With the new miniAODv2, please use CMSSW >=7_4_12

How to run on MC:

```
cmsRun TreeMaker/test/runMakeTreeFromMiniAOD_cfg.py global_tag=74X_dataRun2_Prompt_v2 MC=True deb

cmsRun TreeMaker/test/runMakeTreeFromMiniAOD_cfg.py global_tag=74X_mcRun2_asymptotic_v2 MC=True d
```

How to run on RUN2015D DATA:

```
cmsRun TreeMaker/test/runMakeTreeFromMiniAOD_cfg.py global_tag=74X_dataRun2_Prompt_v2 MC=False de
```

How to run on DATA, if you need to produce miniAOD too (the first command is to produce miniAOD from AOD):

```
cmsDriver.py miniAOD-prod -s PAT --eventcontent MINIAOD --runUnscheduled --data --filein /store/d
cmsRun TreeMaker/test/runMakeTreeFromMiniAOD_cfg.py global_tag=GR_R_74_V9A::All MC=False isHBHEEa
```

Run on CRAB3

How to set CRAB3 environment:

```
source /cvmfs/cms.cern.ch/crab3/crab.sh
voms-proxy-init
```

How to run on CRAB3 (MC, on a single dataset):

```
crab submit -c TreeMaker/test/prod_singleFlatTrees.py
crab status -d RSGraviton1000/crab_RSGraviton1000/
```

How to run on CRAB3 (data, on a single dataset):

```
crab submit -c TreeMaker/test/prod_singleFlatTrees_run2_data_mu.py
crab status -d data_mu/data_mu/
```

How to run on CRAB3, MC and data (25ns, new MINIAOD version):

```
python prod_flatTrees_run2_data_25ns_miniAODv2.py
python TreeMaker/test/prod_flatTrees_MC76x_25ns_miniAODv2.py
```

How to run on CRAB3, MC and data (25ns, old miniaod):

```
python TreeMaker/test/prod_flatTrees_run2_data_25ns.py
python TreeMaker/test/prod_flatTrees_MC74x.py
```

How to run on CRAB3, MC and data (50ns):

```
git checkout 74x_newEleId_50ns
python TreeMaker/test/prod_flatTrees_run2_data.py
python TreeMaker/test/prod_flatTrees_MC74x_50ns.py
```

How to check status on the all datasets:

```
chmod 744 TreeMaker/test/multicrab
./TreeMaker/test/multicrab -c status -d ntuple/
```

How to run on CRAB3 (data, if you need to reproduce miniAOD too):

```
cmsDriver.py miniAOD-prod -s PAT --eventcontent MINIAOD --runUnscheduled --data --conditions GR_R
crab submit -c TreeMaker/test/prod_miniAOD_forData.py
crab submit -c TreeMaker/test/prod_singleFlatTrees_DATA.py
```

-- LucaBrianza - 2015-02-13

This topic: Sandbox > ExoWWSemileptonicRun2
Topic revision: r97 - 2016-11-09 - LucaBrianza