

Zprime4Lepton8TeV

Recipe for Z' decaying into 4 leptons at 8 TeV

This twiki contains all the procedure of Z' decaying into 4 leptons at 8 TeV (EXO-14-006) including technical details, performed by Hwidong Yoo (hdyoo@cern.ch). All based machineries are available at xte.physics.purdue.edu/data/HWIDONGYOO. All recipes start from this working directory.

Documents

CADI: [EXO-14-006](#)

AN note: [2014/060](#)

Pre-approval talk: [link](#)

Physics motivation

Non-typical Z' search: A baryonic Z' resonance model in SUSY framework

B. Barger and H.S. Lee, Phys. Rev. D 85, 055030 (2012)

Dataset

8 TeV Full dataset: Jan22 rereco

JSON: [/afs/cern.ch/cms/CAF/CMSCOMM/COMM_DQM/certification/Collisions12/8TeV/Reprocessing/Cert_190456-208686_8TeV_22Jan2013ReReco_Collisions12_JSON_MuonPhys.txt](https://afs.cern.ch/cms/CAF/CMSCOMM/COMM_DQM/certification/Collisions12/8TeV/Reprocessing/Cert_190456-208686_8TeV_22Jan2013ReReco_Collisions12_JSON_MuonPhys.txt)

Summer12 MC samples (see details in AN note)

Benchmark signal MC generation

LHE production

Use CalcHep 3.4.1

```

cd calchep_3.4.1/newProd
# directories Calchep750 to Calchep3000 (10 mass bins)
# in each directory, calchep_batch is the macro to run the batch input
card (calchep is for single interactive calculation)
# batch_file_* is input card for each channel (ex. batch_file_4m is input
card for 4 muon channel)
# in models, there are model files: vars4.mdl and vars9.mdl contain the
detailed parameter sets
# phiMass0p*ZpM directory contains the LHE output for different M(phi)
scenario (* = 05 - 45 => 5% to 45% of Z' mass)

./run.csh # run each channel in 10 mass bins
# change the channel in the script if wanted
# 10 interactive jobs with 12k events take approximately < 1 hour in
lxplus machine (easy to produce interactively)
# other scripts are only for the convenience of organization of output.

```

Produce LHE file using the CalcHep and the LHE file will be input for your signal MC production.

MC production

```

# current signal sample, M(phi) = 50 GeV, is done by Suneel
Under investigation

```

Ntuple production

Ntuple production is based on the DY ntuple production. In this version, more variables for high pt muon id/iso and HEEP id/iso are included. Also there are preselection briefly to reduce the size of ntuples. See more details in ZprimeAnalyzer.cc

```

cd CMSSW_5_3_11/src/Phys/ZprimeAnalyzer/ # this directory is for ntuple
production
# ZprimeAnalyzer: EDMAalyzer for ntuple production
# basic config: python/ZprimeAnalyzer_cfi.py
# PU reweighting config: python/PUreweight2012_cff.py
cd ntuples/allProd/ # directory for all data/background MC ntuple
production
multicrab -create -submit -cfg MultiCrabConfig_*.cfg # see each config
file what for
cd ntuples/Modell/ # directory for all signal MC ntuple production
# use submit.csh to submit crab jobs with existing configuration

```

Location of latest ntuples for this analysis:

- **Data/background MC:** [eos/cms/store/group/phys_smp/hdyoo/Analysis2/eighthRound_02102014/](https://eos.cms/store/group/phys_smp/hdyoo/Analysis2/eighthRound_02102014/)
- **Signal:** [afs/cern.ch/work/h/hdyoo/Zprime/ntuples/eighthRound_02102014](https://afs.cern.ch/work/h/hdyoo/Zprime/ntuples/eighthRound_02102014)

Analysis machinery setup

```
cd analysisCodes/preapproval # latest version of macro set
# each directory name shows which analysis steps are
# "rootfiles" directory contains all necessary setup files to run ROOT
macros for this analysis
cd rootfiles
# eosChain.pl: create "chain_*.C" file as a ROOT TChain input of ntuples
(list of ntuples for each data/mc samples), which files are located in eos
directory
# createChain.pl: same as eosCHain.pl but files stored in scratch
# SetupTree.h,C: setup class,declare parameters to run macro => add/modify
properly if necessary
./eosChain.pl DYMuMu_M20 => create chain_DYMuMu_M20.C in same directory
(see the perl script for more details)
```

Event selection

Single Kinematics

```
cd Gen
root -b -q GenInfo.C++ # produce histograms to get the comparison between
leading and subleading leptons pt from signal generator information
root makePlots.C # produce plots in AN note
```

Muon id

Follow up the high pt muon id recommended by muon POG: [link](#)

Require only Tracker muon (and drop other muon detector information criteria) in case that two muons are close each other: same recipe as boosted Z -> mumu paper (see more details in AN note)

Selections are included in the macros, for example

```

if( fabs(muon_eta[j]) < _etaCut
    && muon_trackerLayers[j] > 5
    && muon_pixelHits[j] > 0
    && _muIso < 0.10
    && fabs(muon_dxyVTX[j]) < 0.2 && fabs(muon_dzVTX[j]) < 0.5 )
{
    if( muon_cktpt[j] > _ptCutLeading ) {
        // GLB muons
        if( muon_type[j] == 0 || muon_type[j] == 1 ) {
            if( muon_muonHits[j] > 0 && muon_nMatches[j] > 1 &&
muon_cktptError[j] / muon_cktpt[j] < 0.3 ) {
                muGLBLeading.push_back(tmpMu);
                if( muon_charge[j] > 0 ) muPlus.push_back(tmpMu);
                if( muon_charge[j] < 0 ) muMinus.push_back(tmpMu);
            }
        }
    }
    else if( muon_cktpt[j] > _ptCut ) {
        // GLB muons
        if( muon_type[j] == 0 || muon_type[j] == 1 ) {
            if( muon_muonHits[j] > 0 && muon_nMatches[j] > 1 &&
muon_cktptError[j] / muon_cktpt[j] < 0.3 ) {
                muGLB2ndLeading.push_back(tmpMu);
                if( muon_charge[j] > 0 ) muPlus.push_back(tmpMu);
                if( muon_charge[j] < 0 ) muMinus.push_back(tmpMu);
            }
        }
    }
}
}

```

Muon Iso

Use relative tracker isolation < 0.1 same as Z⁰->mumu analysis but remove 2nd lepton contribution

```

cd muTrkIso
root -b -q muonIso.C++ # produce output of histograms
root effMuon.C # produce comparison plot among original trk iso, PF iso,
and trk iso with a veto of 2nd lepton in AN note

```

Electron id

```

for( int j = 0; j < nelec; j++ ) {
    if( fabs(elec_etaSC[j]) < 1.47 ) isEB = true;
    else isEE = true;
    double _modTrkIso = elec_modIsoPtTrks[j];
    double _modEMHADIso = elec_modIsoEMHADDepth1[j];
    for( int k = 0; k < nTT; k++ ) {

```

```

        if( track_pT[k] < _ptCut ) continue;
        double tmpDr = deltaR(track_eta[k], track_phi[k], elec_gsfEta
[j], elec_gsfPhi[j]);
        if( tmpDr < 0.3 ) {
            _modTrkIso -= track_pT[k];
            if( _modTrkIso < 0 ) _modTrkIso = 0;
        }
    }
    bool isElecSel = false;
    if( fabs(elec_etaSC[j]) < _etaCut
    && elec_ecalDriven[j] == 1
    && fabs(elec_dPhiIn[j]) < 0.06
    && fabs(elec_dEtaIn[j]) < 0.02
    && elec_HoverE[j] > -99 && elec_HoverE[j] < 0.1 // default:
0.05
    && (elec_mHits[j] >= 0 && elec_mHits[j] <= 1)
    && _modTrkIso < 5 // iso
    ) {
        if( isEB ) {
            if( //fabs(elec_dEtaIn[j]) < 0.007
            //(elec_25over55[j] > 0.90 || elec_15over55[j] > 0.83)
            _modEMHADIso < (2.0+0.05*elec_et[j]) // iso
            && fabs(elec_dxyVTX[j]) < 0.02 ) {
                if( elec_et[j] > _ptCutLeading ) {
                    elecEBLeading.push_back(tmpElec);
                    if( elec_charge[j] > 0 ) elPlus.push_back(tmpElec);
                    else elMinus.push_back(tmpElec);
                    isElecSel = true;
                }
            }
            else if( elec_et[j] > _ptCut ) {
                elecEB2ndLeading.push_back(tmpElec);
                if( elec_charge[j] > 0 ) elPlus.push_back(tmpElec);
                else elMinus.push_back(tmpElec);
                isElecSel = true;
            }
        }
    }
    else if( isEE ) {
        bool isPassEMH = false;
        // elec iso
        if( elec_et[j] < 50. ) {
            if( _modEMHADIso < (2.5) ) isPassEMH = true;
        }
        else {
            if( _modEMHADIso < (2.5+0.05*(elec_et[j]-50.)) ) isPassEMH
= true;
        }

        if( isPassEMH
            && elec_sigmaIEtaIEta[j] > -99 && elec_sigmaIEtaIEta[j]
< 0.03
            && fabs(elec_dxyVTX[j]) < 0.05 ) {
            if( elec_et[j] > _ptCutLeading ) {

```

```

        elecEELeading.push_back(tmpElec);
        if( elec_charge[j] > 0 ) elPlus.push_back(tmpElec);
        else elMinus.push_back(tmpElec);
        isElecSel = true;
    }
    else if( elec_et[j] > _ptCut ) {
        elecEE2ndLeading.push_back(tmpElec);
        if( elec_charge[j] > 0 ) elPlus.push_back(tmpElec);
        else elMinus.push_back(tmpElec);
        isElecSel = true;
    }
}
}
}
}

```

Merged electron/HEEP id modification study

Need this treatment to recover inefficiency by the merged electron

```

// check additional gsfTrack in a cone (dR < 0.30)
bool isMoreGsfTrack = false;
if( isElecSel && elec_et[j] > 500. ) {
    for( int l = 0; l < nTT; l++ ) {
        if( track_pT[l] < _ptCut ) continue;
        double dR = deltaR(elec_etaSC[j], elec_phiSC[j], track_eta
[1], track_phi[1]);
        //cout << "track = " << l << " " << track_pT[l] << " " <<
track_eta[1] << " " << track_phi[1] << " " << track_charge[1] << " " <<
endl;
        if( dR < 0.25 && elec_eoverp[j] > 1.5 ) isMoreGsfTrack =
true;
    }
    if( isMoreGsfTrack ) nMerged++;
}

```

```

cd preapproval/countMerged
root -b mergedElecAllch.C++ # produce histogram for fraction of merged
electron
root fracMerge.C # produce a plot in AN Note

cd preapproval/inEffElec/
root -b cutOpt.C++ # produce histograms for HEEP id modification study
root makePlots.C
root -b invMergedElec++ # produce histograms for merged electron studyroot
EoverP.C

```

Electron iso

```
cd preapproval/elecIso
root -b elecIso.C++ # produce histograms for electron iso
root effElec.C # produce the figure in AN note
```

Efficiency

Produce various muon/electron id efficiency and event selection efficiency

```
cd preapproval/accEff
root -b evtEff(ZZ).C++ # produce histograms for cut-flow of event
selection efficiency
root compEvtEff.C # produce plots
root -b lepEff.C++ # produce histograms for lepton id efficiency as a
function of lepton pt
root compSeqEff.C
root compLepPt.C
root compIso.C
root seqEff(ZZ).C++ # produce histograms for cut-flow event selection
efficiency
root -b AccEff.C++ # produce histograms for event selection efficiency as
a function of Z' mass
root finalEff.C # produce plots in AN note
```

Background estimation

Control plots

```
cd preapproval/ControlPlots
root -b ControlPlots.C++ # produce histograms for comparison between data
and mc as a function of 4 lepton inv. mass
root getControlPlots.C # produce plot in AN note
```

Mass resolution

```
cd preapproval/Gen
root -b MassResolution.C++ # produce mass resolution histograms
root massRes.C # fit the resolution plots
root combRes.C # fit and produce figure in AN note
```

Systematic uncertainties

To produce efficiency systematic uncertainty

```
cd preapproval/accEff
root GetSyst.C # then print out the efficiency difference between signal
and ZZ MC
```

Limit calculation

Use HiggsCombined package in CMSSW: [twiki](#)

```
setenv SCRAM_ARCH slc5_amd64_gcc472
cmsrel CMSSW_6_1_1 ### must be >= 6.1.1, as older versions have bugs (6.2.
X and 7.0.X are NOT supported either)
cd CMSSW_6_1_1/src
cmsenv
git clone https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit.git
HiggsAnalysis/CombinedLimit

cd CMSSW_6_1_1/src/HiggsAnalysis/CombinedLimit/preapproval_final/ #
working directory used for preapproval results
```

Produce limit plots

Need to produce limit plots for both data and mc samples with fine bin size to produce datacard of HiggsCombined package

```
cd analysisCodes/preapproval/Limits/
root -b limit.C++ # to produce all necessary histograms for datacard and
limit calculation
```

Background fit

In current analysis (presented in preapproval), we set the background rate to 0 in the limit calculation because the background is negligible. Therefore this background fit is not used in current analysis.

The main motivation for this background fit is to estimate backgrounds in $M(4l) > 1$ TeV region because we have almost no statistics from SM backgrounds in this region. So we extrapolate the distribution of backgrounds from low mass region ($M(4l) < 500$ GeV) where we observe good agreement between data and mc.

```
cd analysisCodes/preapproval/Limits/
root fitPlotBg.C
```

Produce datacard

We need large number of datacard as an input for limit calculation and therefore we need to automatize it.


```
cd analysisCodes/preapproval/Limits/  
root prodDataCard.C  
# you can find the datacard in each channel in dataCard directory
```

Run limit calculation

Need to carefully test the option of the method and then produce all channels

In currently analysis, we use "BayesianToyMC" method (same as MarkovChainMC, but simpler option), recommended by experts.

```
# for mumumumu channel as an example (same for other channels)  
cd CMSSW_6_1_1/src/HiggsAnalysis/CombinedLimit/preapproval_final/  
cmsenv  
./calc.csh  
# high mass calculation is done quickly relatively and low mass  
calculation takes longer time  
# need to adjust the option (for example, reduce the integration range  
rMax, if the signal strength is smaller)  
# example output  
-- BayesianToyMC --  
Limit: r < 0.677481 @ 95% CL  
Done in 1.86 min (cpu), 1.86 min (real)  
mean expected limit: r < 0.676587 +/- 4.86033e-05 @ 95%CL (200 toyMC)  
median expected limit: r < 0.676539 @ 95%CL (200 toyMC)  
68% expected band : 0.675963 < r < 0.677278  
95% expected band : 0.675415 < r < 0.678158  
# there is signal strength as a limit and 1/2 sigma expected band  
  
# model-independent calculation  
combine -M BayesianToyMC ../../dataCard/zprime_mumumumu_ind.txt --rMax 3 -  
t 200 -i 100000 > & outputInd_mumumumu.txt &
```

Limit plot

```
cd analysisCodes/preapproval/Limits/  
# When you finish to calculate the limits for all channels and mass bins,  
you need extract the limit results from the output and make the input  
files as the files in inputLimits/  
root drawLimit.C # produce the figure in AN note  
root calcGridLimit.C # produce the table in AN note (and paper)
```