

# Table of Contents

Recreating David's first set of R21 fakes plots:	
<a href="https://indico.cern.ch/event/804144/#19-first-look-at-fake-in-rel21">https://indico.cern.ch/event/804144/#19-first-look-at-fake-in-rel21</a> .....	1
.....	2
<b>FF plotting (single fakes for now)</b> .....	2

# Recreating David's first set of R21 fakes plots: <https://indico.cern.ch/event/804144/#19-first-look-at-fake>

working on branch: ZjetsFF -R21-init (to be merged soon to master)

```
$ pwd  
/afs/cern.ch/work/s/sofen/Fakes/HWWAnalysis/HWWAnalysisCode
```

```
$ git checkout ZjetsFF -R21-init
```

# FF plotting (single fakes for now)

plotting macro :

(<https://gitlab.cern.ch/atlas-physics/higgs/hww/HWWAnalysisCode/blob/master/share/fakefactors/plotting/makeFFPlots.cxx>)

```
bool initializeSamples(std::map<TString, Sample*> samples)

void make2DPlots_numden_and_statunc(std::map<TString, Sample*>& samples)

int
makeEWSubtractionPlots( Sample* sample, bool doElectron, bool doMuon, bool isDijetsFF, bool getAb

void makeFFVsPtPlots(std::map<TString, Sample*>& samples, bool plotAbsoluteVariations=false)

void makeDirs(std::map<TString, Sample*>& samples)

void mainFcn(std::map<TString, Sample*>& samples, const e_PARTICLE part, bool plotAbsoluteVariati

/*****
/*****      MACRO BEGINS HERE      *****/
/*****
int makeFFPlots(
    bool doElectron = true, // run Electron
    bool doMuon = true,    // run Muon
    // TString text_on_plot = "#splitline(id: Tight and new iso full pT){anti-id: Tight and new iso full pT, d0 dropped}",
    //TString text_on_plot = "anti-id: Medium full pT, d0 dropped",
    // TString text_on_plot = "new selections",
    TString text_on_plot = "",
    bool DEBUG = false,    // print out a bunch of messages
    bool make2DNumAndDenStatUncPlots = false, // make 2D plots with numerator and denominator
    bool FlavorSplitMode = false,
    bool doCorrFactors = false,
    bool doSyst_EWsubtr = false,
    bool isDijetsFF = false, // if false assume Zjet. important for EW subtr com
    bool plotAbsoluteVariations = false // if true, plot also the EW up/down variations
)

{

// loading the QFramework lib
// gROOT->ProcessLine(".x $ROOTCOREDIR/scripts/load_libraries.C");
// gROOT->ProcessLine(".L $ROOTCOREBIN/obj/$ROOTCORECONFIG/QFramework/lib/libQFramework.so");
// doesn't really work because of complex c++/root things.
// Use this macro with tqroot -b; .x <thisMacro>
gROOT->SetBatch();

#### some flags got stored -----
g_textOnPlot = text_on_plot;
g_DEBUG = DEBUG;
g_make2DNumAndDenStatUncPlots = make2DNumAndDenStatUncPlots;
g_FlavorSplitMode = FlavorSplitMode;
#### plotstyle using atlas format with 2 digits after decimal
SetAtlasStyle();
gStyle->SetPaintTextFormat("4.2f");

#### canvas format
canvas.SetBottomMargin(0.15);
```



## VBFHWWFakeLeptons < Sandbox < TWiki

```
//std::vector<TString> sample_keys={"v17b_Zjets_newZTagger_WZ1p15_ZZSherpa2p1_FakeWJetsSF_ee", "  
//std::vector<TString> sample_keys={"v17b_ZjetsMU_SherpaWZ1p18_ZZSherpa2p1", "v17b_Zjets_newZTag  
//std::vector<TString> sample_keys={"v17b_Zjets_newZTagger_WZ1p15_fakePosComb", "v17b_Zjets_new  
  
//*** zjet, dijet, dijet-fine-binning **/  
//std::vector<TString> sample_keys={  
// "v17b_Zjets_newZTagger_WZ1p15_fakePosComb",  
// "v17b_NOMINAL",  
// "v17b_NOMINAL_copy"  
//};  
//  
//*****/  
  
// "v17b_Zjets_newZTagger_WZ1p15_EWSuppr", "v17b_Zjets_newZTagger_WZ1p15_EWSuppr4", "v17b_Zjet  
//std::vector<TString> sample_keys={"v17b_Zjets_newZTagger_WZ1p15_fakePosComb", "v17b_Zjets_new  
//std::vector<TString> sample_keys = {"v17b_Zjets_newZTagger_WZ1p15_EWSuppr", "v17b_Zjets_newZT  
//std::vector<TString> sample_keys={  
// "v17b_Zjets_newZTagger_WZ1p15_Javier_nom",  
// "v17b_Zjets_newZTagger_WZ1p15_Javier_100",  
// "v17b_Zjets_newZTagger_WZ1p15_Javier_40",  
// "v17b_Zjets_newZTagger_WZ1p15_Javier_50",  
// "v17b_Zjets_newZTagger_WZ1p15_Javier_60",  
// "v17b_Zjets_newZTagger_WZ1p15_Javier_70",  
// "v17b_Zjets_newZTagger_WZ1p15_Javier_80",  
// "v17b_Zjets_newZTagger_WZ1p15_Javier_90",  
//}; //v17b_Zjets_newZTagger_WZ1p15_EWSuppr5", "v17b_Zjets_newZTagger_WZ1p15_EWSuppr6", "v17b_Z  
//  
  
//*** different binnings **/  
std::vector<TString> sample_keys={  
// "kkWJets20171202NewSkimNew_Mt",  
// "kkWJets20171202NewSkimNew_Mt_MtTrackMet",  
// "kkWJets20171202NewSkimNew_Mt_MtTrackMet_DPhi",  
// "kkWJets20171202NewSkimNew_Mt_MtTrackMet_DPhi_mll",  
// "kkWJets20171202NewSkimNew_Mt_MtTrackMet_DPhi_mll_medZCandId",  
// "kkWJets20171202NewSkimNew_Mt_MtTrackMet_mll_medZCandId",  
// "kkWJets20171202NewSkimNew_Mt_mll_medZCandId",  
// "kkWJets20171202NewSkimNew_mll_medZCandId",  
// "kkWJets20171204FlavComp_Mt",  
// "kkWJets20171204NewEWSuppression_All",  
  
"kkWJets20171205FFBinningA_mcTruthMatch_stdZCandId_Sherpa2p2p1",  
"kkWJets20171205FFBinningA_Mt_stdZCandId_Sherpa2p2p1",  
"kkWJets20171205FFBinningA_mll_Mt_stdZCandId_Sherpa2p2p1",  
"kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_stdZCandId_Sherpa2p2p1",  
"kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_Sherpa2p2p1",  
  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_defaultBins",  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_moreBinsA",  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_moreBinsB",  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_moreBinsC",  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_moreBinsD",  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_moreBinsE",  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_moreBinsF",  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_moreBinsG",  
// "kkWJets20171205FFBinningA_Mt_stdZCandId_moreBinsH",  
// "kkWJets20171205FFBinningA_mll_Mt_stdZCandId",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_stdZCandId",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_defaultBins",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_moreBinsA",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_moreBinsB",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_moreBinsC",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_moreBinsD",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_moreBinsE",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_moreBinsF",  
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_moreBinsG",
```

## VBFHWWFakeLeptons < Sandbox < TWiki

```

// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_stdZCandId_moreBinsH",
// "kkWJets20171205FFBinningA_Mt_medZCandId_defaultBins",
// "kkWJets20171205FFBinningA_Mt_medZCandId_moreBinsA",
// "kkWJets20171205FFBinningA_Mt_medZCandId_moreBinsB",
// "kkWJets20171205FFBinningA_Mt_medZCandId_moreBinsC",
// "kkWJets20171205FFBinningA_Mt_medZCandId_moreBinsD",
// "kkWJets20171205FFBinningA_Mt_medZCandId_moreBinsE",
// "kkWJets20171205FFBinningA_Mt_medZCandId_moreBinsF",
// "kkWJets20171205FFBinningA_Mt_medZCandId_moreBinsG",
// "kkWJets20171205FFBinningA_Mt_medZCandId_moreBinsH",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_defaultBins",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_moreBinsA",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_moreBinsB",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_moreBinsC",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_moreBinsD",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_moreBinsE",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_moreBinsF",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_moreBinsG",
// "kkWJets20171205FFBinningA_mll_Mt_MtTrackMet_DPhi_moreBinsH",
// "v17b_Zjets_newZTagger_WZ1p15_fakePosComb",
// "v17b_Zjets_newZTagger_WZ1p15_binning3",
// "v17b_Zjets_newZTagger_WZ1p15_binning5",
// "v17b_Zjets_newZTagger_WZ1p15_binning8",
// "v17b_Zjets_newZTagger_WZ1p15_binning9",
// "v17b_Zjets_newZTagger_WZ1p15_binning10",
};
/*****/

// For colour setting overrides
std::vector<int> colours{kBlack,kRed,kBlue,kCyan+1,kMagenta,kGray+2,kOrange,kGray,kYellow,kGreen};
std::size_t colourIdx = 0;
// fill map
for (const auto& key : sample_keys) {
    if (allSamples.find(key) == allSamples.end() ) {
        // not found
        std::cout << "ERROR the sample ID specified by you, " << key << ", does not exist in sample keys\n";
        return 1;
    } else {
        samples[key] = allSamples.at(key); // add to map
        samples[key]->color = colours[colourIdx];
        ++colourIdx;
    }
}

/*****/
/*****      DONE adding samples      *****/
/*****/
cout << "\n\nYou will now run this plotting script with the following samples: " << endl;
dumpSamples(samples);

if (!loadSampleReaders(samples)) throw std::runtime_error("Couldn't load TQSampleDataReaders!");
checkThatHistosExist(samples);

// initialize the Samples -- get histos and attach to Sample objects
// (i.e. id and anti-id) histos, as well as the FF histos
if (doElectron) {
    g_PARTICLE = ELECTRON;
    if (!initializeSamples(samples)) throw std::runtime_error("Inside mainFcn :: couldn't initialize Electron Samples");
}
if (doMuon) {
    g_PARTICLE = MUON;
    if (!initializeSamples(samples)) throw std::runtime_error("Inside mainFcn :: couldn't initialize Muon Samples");
}
for (auto& key_sample : samples) {
    Sample* sample = key_sample.second;
    sample->isInitialized=true;
}

```

## VBFHWWFakeLeptons < Sandbox < TWiki

```

}

/** sanity check -- for example make sure the muon histos aren't equal to the electron histos
if (!lookSane(samples, doMuon, doElectron)) {
    cout << "*** Samples don't look sane!" << endl;
    return 1;
}

// Make the directories
makeDirs(samples);

/** NOW EVERYTHING IS SET UP AND LOOKING SANE, READY TO DO WHAT WE WANT TO DO **/

/** DO THE HEAVY WORK: MAKE PLOTS **/
if (doElectron) mainFcn(samples, ELECTRON, plotAbsoluteVariations);
if (doMuon)     mainFcn(samples, MUON, plotAbsoluteVariations);

/** THE FAKE FACTOR HISTOS HAVE BEEN FILLED, NOW SAVE THEM AS ROOT FILES **/
saveAsRootFiles(samples, doElectron, doMuon);

if (doCorrFactors) {

    // Define the correction factor pairs:
    //   For each pair of sample_ID,
    //   a correction factor will be calculated
    //   as ( pair.first / pair.second )

    std::vector< std::pair<TString, TString> > corrFactorPairs = {
                                                                    {"v17b_WjetsAlpgen_nominal", "
                                                                    {"v17b_WjetsAlpgen_SS_nominal
                                                                    // {"v17b_WjetsSherpa2p2p1_no
                                                                    // {"v17b_WjetsSherpa2p2p1_SS
                                                                    // {"v17b_WjetsPowheg_nominal
                                                                    // {"v17b_WjetsPowheg_SS_nomi
                                                                    };

    gSystem->mkdir("./CorrFactorPlots");

    TFile file_out("../corr_factors.root", "RECREATE");

    if (doElectron) computeCorrectionFactors(samples, ELECTRON, corrFactorPairs);
    if (doMuon)     computeCorrectionFactors(samples, MUON, corrFactorPairs);

    file_out.Close();
}

// EW-varied uncertainty histos
if (doSyst_EWsubtr) {
    cout << "\n*****" << endl;
    cout << "*** About to compute the electroweak subtraction variation systematic..." << endl;
    cout << "\n*****" << endl;

    // prepare for systematics
    for (auto& key_sample : samples) {
        Sample* sample = key_sample.second;
        // if (sample->sample_ID != g_NOMINALSAMPLE) continue;
        if ( makeEWSubtractionPlots( sample, doElectron, doMuon, isDijetsFF ) ){
            std::cout << "Something went wrong in makeEWSubtractionPlots... returning 1" << std::endl;
            return 1;
        }
    }
    cout << "\n*** DONE computing EW subtraction variation systematic." << endl;
}

return 0;

```

}

-- SouravSen - 2019-03-03

---

This topic: Sandbox > VBFHWWFakeLeptons

Topic revision: r3 - 2019-03-13 - SouravSen



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback