Useful Commands (CMSSW, RCT, wisconsin scripts, etc)

RCT Commands & Links

- RCT - Basic info on RCT, shift schedules
- PatternTest - Instructions for running test to check our RCT crates and links to GCT source cards
- RCTDataQualityMonitoring - Instructions for creating DQM plots
- HWGuideRegionalCaloTrigger - Useful hardware information and links
- RCTOps - Info on power, cooling, rack computers, configuration
- RCTChecklist - Checklist for being on-call
- DCS - Slow control (DCS) information
- DCSSoftware - Info on setting up DCS
- VplusJets - Info on setting up DCS

CMSSW project and analyzer

- `scramv1 project CMSSW CMSSW_X_Y_Z` (or `cmsrel CMSSW_X_Y_Z`)
- `cd CMSSW_X_Y_Z/src`
- `cmsenv`
  - (`eval `scramv1 runtime -sh``)
- `cmscvsroot CMSSW`
- `cvs co [path/of/CMSSW/files]`
- `scramv1 b`

- `cd CMSSW_X_Y_Z/src`
- `mkdir Analysis`
- `cd Analysis`
- `mkedanlzr AnalyzerName`
- `cd Analyzer`
- `scramv1 b`
- `cmsRun analyzername_cfg.py`

Submitting to Wisc Grid

- Get a grid proxy:
  - `voms-proxy-init --valid 48:00`
- Farmout the analysis jobs:
  - `farmoutAnalysisJobs --input-files-per-job=5 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/hep.wisc.edu/data4/PA_TFC/store/C WenuChowder ~/CMSSW_1_6_8/ ~/CMSSW_1_6_8/src/Analysis/analysis.cfg`
- In a scratch folder on login machine, merge results of farmed out analysis
  - `mergeFiles --use-hadd --copy-timeout=60 WenuChowder.root /pnfs/hep.wisc.edu/data5/uscms01/grogg/WenuChowder-analysis`
  - Use options `--cache-dir=/scratch/grogg/[alreadyCopiedFileDirectory] --reuse-cache-files` to only add new files
- Modify files in dChache:
  - `grid-xterm-uwhep`
- To copy to/from dcache:
  - `srmcp file://localhost//scratch/grogg/[localFileName].root srm://cmssrm.hep.wisc.edu:8443/pnfs/hep.wisc.edu/data5/uscms01/grogg/[sameOrNewName]`
- To manage files in dcache
  - `gsissh -p 222 cmsgrid02.hep.wisc.edu`
  - `cd /pnfs/hep.wisc.edu/store/user/grogg`

- To look at dCache files using root :
  - `root dcap://cmsdcap.hep.wisc.edu:22125//pnfs/hep.wisc.edu/data5/uscms01/grogg/[fileName]`

        - root dcap://cmsdcap.hep.wisc.edu:22125//pnfs/hep.wisc.edu/data4/PA_TFC/store/[path/to/fil
            - Events->Show(0)
        - edmDumpEventContent [path/to/root/file]

Tunneling Commands

- Tunnel into online machines within CERN:
    - ssh -ND 1080 -l grogg cmsusr1.cern.ch
- Tunnel into online machines from outside CERN:
    - ssh -t grogg@lxplus.cern.ch -L 1080:localhost:1081 "ssh -1 -ND 1081 grogg@cmsusr1"
- Tunnel into pcwiscms05 (for DCS)
    - ssh -Y grogg@lxplus.cern.ch -L 60001:pcwiscms05.cern.ch:3389

CVS

- CVS startup commands for online (pattern tests)
    - kinit grogg@CERNNOSPAMPLEASE.CH
    - export CVSROOT=:ext:grogg@isscvs.cern.ch:/local/reps/tridas
    - export CVS_RSH=ssh
    - -OR-
    - export CVSROOT=:gserver:isscvs.cern.ch:/local/reps/tridas
- To update CVS in a directory checked out under the old CVS version
    - Change the CVS/Root file in the directory to refer to the new version
        - :ext:@isscvs.cern.ch:/local/reps/tridas
    - See RCTCodeRepository and use the ssh method
- Wisconsin CVS
    - export CVSROOT=/afs/hep.wisc.edu/cms/CVSRepository/
- CERN cvs from wisconsin
    - cmscvsroot CMSSW (doesn't work anymore)
    - export CVSROOT=:gserver:cmscvs.cern.ch:/cvs_server/repositories/CMSSW
    - kinit grogg@CERNNOSPAMPLEASE.CH

SVN

For notes (on lxplus):

```
svn co -N svn+ssh://svn.cern.ch/reps/tdr2 tdr2
cd tdr2
svn update utils
svn update -N [papers|notes]
svn update [papers|notes]/XXX-YY-NNN
```

- cd tdr2
- eval `notes/tdr runtime -sh`
- cd [note/path]
- tdr --style=pas b EWK-10-012

OR

- tdr --style=an b AN-11-136

CRAB

- Good stuff: WorkBookCRABTutorial

```
source /afs/cern.ch/cms/LCG/LCG-2/UI/cms_ui_env.sh
```

```
cd CMSSW_X_Y_Z/src

cmsenv

source /afs/cern.ch/cms/ccs/wm/scripts/Crab/crab.sh

cd /path/to/run/from/

cp $CRABPATH/crab.cfg .
```

OR

```
cp $CRABPATH/full_crab.cfg .
```

Edit the crab.cfg file as needed

```
crab -create [-cfg crabfilename.cfg]

crab -submit

crab -status [-c crab_0_directory]

crab -getoutput [-c crab_0_directory]

crab -publish
```

DBS

Need to run `cmsenv` in a CMSSW area first

```
dbs search --query='find dataset where dataset like *cosmics*/RECO'

#find datasets containing events from run number 108741
dbs search --query='find dataset where run = 108741'


#find for all datasets the files of a given Run:Lumi
dbs search --query='
find file,dataset where run=109011 and lumi=19
```

Other stuff

- Access other AFS spaces
  - `klog -pr grogg -cell hep.wisc.edu`
  - `klog -pr grogg -cell cern.ch`
- Update things in rctts
  - `xsudo -u rctts`
- Find files of a certain size, while skipping a directory (scratch0):
  - `find . -path './scratch0' -prune -o -path './images' -prune -o -size +1000k`
- Basement printer:
  - 32-SB02-HP
  - 137.138.213.14
- Making a root class based off of NTuples:
  - `TChain T("nameOfNTuple");`
  - `T.Add("nameOfFiles*");`
  - `T->MakeClass("nameOfCFile");`
  - To compile and run:
    - Put the `Loop()` function into the constructor (in header file), and run using `root nameOfCFile.C+`

- To *really* quit root: .qqqqqq (six q's)

```
source /afs/cern.ch/user/k/klukas/public/Sharing/root2matplot_setup.sh
source /afs/hep.wisc.edu/src/root2matplot/root2matplot_setup.sh
```

- To see event content of EDM file:
  - `edmFileUtil -P -f dcap://cmsdcap.hep.wisc.edu:22125/pnfs/hep.wisc.edu/data4/PA_TFC/...`
- myTree->GetListOfLeaves()->Print()
- To log in to dcs computers
  - rdesktop -f cerntscms01.cern.ch -n dkdjf
- Using screen (for scping, cases with nohup not an option)
  - `screen`
  - [command to run]
  - ctrl+A d to detach
  - `screen -r` to reattach

Standalone EWK DQM

Job submission

- For old 31X files
  - Need to change akt5... to antikt5... in jets

```
farmoutAnalysisJobs --input-files-per-job=3 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=3 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
```

- For new 7TeV production

```
farmoutAnalysisJobs --input-files-per-job=3 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=3 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
```

- For Summer09 reprocessing

```
farmoutAnalysisJobs --input-files-per-job=3 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=3 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
```

- To merge the resultant files

```
mergeFiles --copy-timeout=30  --use-hadd EMEnriched_Pt80to170_31X.root /pnfs/hep.wisc.edu/store/u
mergeFiles --copy-timeout=30  --use-hadd EMEnriched_Pt80to170_reRecoPreprod.root /pnfs/hep.wisc.e

mergeFiles --copy-timeout=30  --use-hadd BCtoE_Pt80to170_31X.root /pnfs/hep.wisc.edu/store/user/g
mergeFiles --copy-timeout=30  --use-hadd BCtoE_Pt80to170_reRecoPreprod.root /pnfs/hep.wisc.edu/st
```

W->enu + jets work

Recent job submission

```
 farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
 farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
 farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
```

```
 farmoutAnalysisJobs --input-files-per-job=1 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
 farmoutAnalysisJobs --input-files-per-job=1 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
 farmoutAnalysisJobs --input-files-per-job=1 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h

 farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
 farmoutAnalysisJobs --input-files-per-job=6 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
 farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
```

```
farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he

farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he

farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
farmoutAnalysisJobs --input-files-per-job=2 --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/he
```

For minbias MC and Data:

```
farmoutAnalysisJobs --input-files-per-job=5  --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
farmoutAnalysisJobs --input-files-per-job=5  --input-dir=dcap://cmsdcap.hep.wisc.edu:22125/pnfs/h
```

Running NTuple Code

CMSSW_3_7_X

```
cmsrel CMSSW_3_7_0_patch4
cd CMSSW_3_7_0_patch4/src
cmsenv
kinit grogg@CERN.CH
cvs co UserCode/grogg
## For jet cleaning
cvs co -r V01-09-01-06 CondFormats/JetMETObjects
cvs co -r V00-05-08 JetMETAnalysis/JetUtilities
cvs co -r V02-00-04-02 JetMETCorrections/Configuration
cvs co -r V03-00-10-01 JetMETCorrections/Modules
## PF2PAT
addpkg PhysicsTools/PFCandProducer
addpkg PhysicsTools/PatAlgos
## Conversion rejections
cvs co -r V00-05-03 RecoEgamma/EgammaTools
scram b -j 4
```

CMSSW_3_8_X

```
cmsrel CMSSW_3_8_2_patch1
cd CMSSW_3_8_2_patch1/src
cmsenv
kinit grogg@CERN.CH
cvs co UserCode/grogg
## PF2PAT
addpkg PhysicsTools/PatAlgos
## Jet Cleaning
## OLD cvs co -r V00-05-08 JetMETAnalysis/JetUtilities
cvs co -r V03-01-02 CondFormats/JetMETObjects
```

```
cvs co -r V00-08-01 JetMETAnalysis/JetUtilities
cvs co -r V05-00-09 JetMETCorrections/Modules
cvs co -r V02-01-01 JetMETCorrections/Algorithms
cvs co -r V03-01-01 JetMETCorrections/Configuration

scramv1 b -j 4

cmsrel CMSSW_3_8_6
cd CMSSW_3_8_6/src
cmsenv
kinit grogg@CERN.CH
cvs co UserCode/grogg
addpkg DataFormats/PatCandidates                        V06-01-06
addpkg PhysicsTools/PatAlgos                            V08-00-45
addpkg PhysicsTools/PatExamples                         V00-04-23
addpkg PhysicsTools/SelectorUtils                       V00-02-27
addpkg PhysicsTools/UtilAlgos                           V08-02-01
cvs co -r V00-08-01 JetMETAnalysis/JetUtilities
```

CMSSW_3_9_X

On a UW login machine:

```
cmsrel CMSSW_3_9_9
cd CMSSW_3_9_9/src
cmsenv
kinit grogg@CERN.CH
cvs co UserCode/grogg
cvs co -r  V00-03-19  RecoEgamma/ElectronIdentification
cvs co -r  V03-01-21  CondFormats/JetMETObjects
cvs co -r  V02-02-03  JetMETCorrections/Algorithms
cvs co -r  V05-00-17  JetMETCorrections/Modules
cvs co -r V03-02-07  JetMETCorrections/Configuration
checkdeps -a
```

Running the Code

Analysis is done in several steps:

- patTuples from AOD -> ntuples from patTuples -> reduced ntuples (skimming and trimming) -> plots from ntuple -> (fits from plots/further reduced ntuple)

The making of the ntuples is based off of code described here: https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideEWKVPlusJets (Kalanand's code is more up to date. It creates ntuples directly from AOD)

PatTuples

For data, PatTuples are made using http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/VJetsAnalysis_cfg.py?view=markup For MC, PatTuples are made using http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/VJetsAnalysis_mc_cfg.py?view=log

- Data is run using crab: http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/crab_data_399.cfg?view=log
- MC can be run using two options
  - crab: http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/crab_mc_399.cfg?view=log

- farmoutAnalysisJobs (for data at UW): http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/farmoutCommands.txt?view=log
- The crab file or farmout lines (copy and paste individual lines into terminal) need to be modified to suit the needs of a sample being run on.
- Crab needs to be run from a scratch area on one of the login machines

To test either locally, change the PoolSource file names to an AOD file (at UW), and PoolOutputModule output file name to something stored on the local scratch space (the patTuples are very large files). Don't forget to change back before submitting crab or farmout jobs. Farmout jobs take `$inputFileNames` and `$outputFileName`. Crab jobs have empty input files, and any desired output name (no slashes).

There is HLT filtering at this stage, but the total event counts before and after the filter are stored. You can also remove the filter by commenting out the line `process.hltHighLevel *`

PatTuple files go to dcache and can then be run on to create ntuples

Ntuples

For data, ntuples are made using
http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/WenuJetsAnalysis_Data_cfg.py?view=log
For mc, ntuples are made using
http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/WenuJetsAnalysis_MC_cfg.py?view=log

Both data and MC can be submitted using farmoutAnalysisJobs. Again, to test locally, change the PoolSource to one of that patTuples made in the previous step. Change `$outputFileName` in the line `process.VplusJets.HistOutFile = cms.string('$outputFileName')` to desired output name.

This is where the code in UserCode/grogg/src is used. Two configuration files in the /python directory are used.

- Make W candidates from electrons on met: http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/python/WenuCollections_cfi.py?view=log
  - Type of electron and/or MET can be specified
- Set the inputTags for the main code: http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/python/VplusJets_cfi.py?view=markup&sortby=
  - Jet collections, W candidate types, HLT paths, and cut specified here (cut values and MET tags are not really used anymore)
  - Some of these are overwritten in the test file, because they change depending on data/MC or local/farmed out

Skim/trim

The skimming/trimming/merging is done all at once.

- Ntuples stored on dcache need to be copied over to a local scratch area using merge files (syntax specified in http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/farmoutCommands.txt?view=log)
- In /test open root
  - `.L BranchSelector`
  - `mergeFiles()`
- The branch selector/skimmer needs to be modified to ones needs http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/grogg/test/BranchSelector.cpp?view=log
  - This will loop over whatever samples you want, and add branches with event weight and a number (e.g. W MC is "0" and Z MC is "1")

        - ◊ Desired cross section must be specified. If any events were skimmed out at the pattuple stage, these need to be accounted for:
            - · Merge just the histogram in all the ntuple files: `hadd -T sampleName_hist.root [sampleName]`
            - · `root sampleName_hist.root` //open file with just numEvents histogram
            - · `numEvents->GetBinContent(6) / numEvents->GetBinContent(1)` //get the final number of events divided by the initial number. The cross section needs to be multiplied by this number
    - ♦ Add/remove branches as necessary by setting the branch status to 0 or 1 (all are set to zero initially, so only those explicitly set to 1 are kept).
    - ♦ Skimming is done on this line: `TTree *newTree = oldtree->CopyTree("hltAccept==1 && (PfW_electron_pt>10 || CaloW_electron_et > 10)")`
        - ◊ In this case only events that pass the HLT and have a gsf ("Calo") or pf electron above a pt of 10 make it into the final ntuple (the inital and final number of events are printed after merging)

Plotting

- Code is in `UserCode/grogg/macros`
- Base class for running over MC and data: `WenuJetsBase.C`
    - ♦ This is compiled when root starts using a line in rootlogon.C: `gROOT->ProcessLine(".x WenuJetsBase.C+");`
- Sub classes for various tasks include WenuJetsAnalyzer.C, WenuJetsFit.C, WenuJetsJetStudies.C, WenuJetsRooUnfold.C, WenuJetsAbcd.C
- The lines to run each of these tasks is given above the `Loop()` function in each file, e.g. `nohup root -q -b "WenuJetsAnalyzer.C+(0, \"WJets\", 0)" > Wlog &`
    - ♦ Some are run by MC/data type, some need to run over all MC/data at once, and this is indicated by a name, e.g., "WJets", or "all"
- Additional macros are available for plotting from the results of these codes, as well as doing fitting
- Basic plots can be made using Jeff's rootplot program: e.g., `rootplot DataInclusive_36pb_gsfElec.root DataInclusive_36pb_pfElec.root --path=".*W_mt*"`
- Unfolding documentation can be found at RooUnfoldVJets

Fitting

- Code is in `UserCode/grogg/macros`
- TemplateFunctionFit.C is a root macro to do W+jets fitting using Mt and b-tagging together
    - ♦ Warning: There is probably more information and options than necessary (some is left over from older versions of the code and not used)
    - ♦ Files for the PDFs: VecbosBtagPdf.cc, RooCruijffPdf.h
    - ♦ Files for initial parameterization and tests: makeRooDataSetFile.C, testBtagFit.C, testCruijffFit.C
- Needs files made from running WenuJetsFit.C -- One file with histograms for parameterization and comparison, one file with flat ntuple for creating a RooDataSet
    - ♦ Most of the selection and variables are set up in WenuJetsFit.C, including all cuts except for the id/isolation (inverted versions used for background estimation), the Mt distribution and the b-tagging distributions
    - ♦ Easiest (fastest) is to run each MC and dataset separately and use `hadd` to merge them together into one file for the fitter to use
        - ◊ Expected ntuple name is of the format `ntuple_allFitYourChosenNameHere.root` and the histogram file `allFitYourChosenNameHere.root` (can easily change expected format and file path in TemplateFunctionFit.C)
    - ♦ Can run either from these files directly (set redoRooDataSet=true), or a RooDataSet file can be made first (quicker if running fits multiple times)
        - ◊ `nohup root -q -b "makeRooDataSetFile.C+( \"YourChosenNameHere\", \"1\", false, true )" >& datasetlog1&` to make a RooDataSet where

`YourChosenNameHere` is descriptive part of the file name, `1` is the jet bin (repeat for all bins), false is for inclusive/exclusive, and true is for data/mc
- ◊ Cuts can be made at this stage, provided the variables are available in the ntuple (can make signal or background selections)

- Only one jet bin is fit at a time (can be inclusive or exclusive counting)
- `nohup root -q -b "testCruijffFit.C+( \"YourChosenNameHere\", \"0\", \"20\", false, true )" >& testfitlog0&` to set inital cruijff parameters for 0 jet bin (repeat for all bins)
  - ♦ Initial parameters can be changed in the function `void fitCruijff(TString speciesName, RooWorkspace* wspace, bool doubleC = false)`
  - ♦ Performs a cruijff fit to each species (W+jets, top, others) and creates a text file of the parameters that is read by TemplateFunctionFit.C
- `nohup root -q -b "TemplateFunctionFit.C+( \"YourChosenNameHere\", \"0\", \"20\", false, true )" >& datafitlog0&` to perform Mt and b-tag fits to data (or data-like MC), same options as testCruijffFit.C
- There are several options set by boolean in the code, such as whether to to recreate a RooDataSet, do b-tagging, do splots (functionality is questionable), do pull plots
- Some parameters can be changed, such as the minimum jet pt cut and W mt cut -- some are more hard coded than others as the code evolved, or depend on cuts made at the WenuJetsFit.C level
- New variables that are desired in the RooDataSet must be added to the ntuple that is made by running WenuJetsFit.C and then explicitly added to the RooDataSet in TemplateFunctionFit.C or makeRooDataSetFile.C
- Some comments are available throughout the code to explain each section

Running OLD Plotting Code

*** THIS IS OLD ***

- Use root on login01, the Summer09 datasets are in `/scratch/grogg`
- To make basic plots run the following from /afs/hep.wisc.edu/user/grogg/CMSSW_3_1_4/src/grogg/:
  - ♦ `.L WenuJetsTree.C+`
  - ♦ `.x WenuJetsObjects.C+(0, "all", 0)`
    - ◊ First argument is the tree used (should always be 0)
    - ◊ Second argument is the name given to the run, should describe what datasets were used an any special settings or cuts
    - ◊ Third argument is number of events to run over, 0 if running over all events
- For electron specific plots (more detailed efficiency, S/B, ID and isolation)
  - ♦ `.L WenuJetsTree.C+`
  - ♦ `.x WenuJetsElectrons.C+(0, "all", 0)`
    - ◊ Same parameters as above
- For plots comparing generator and reco objects
  - ♦ `.L WenuJetsTree.C+`
  - ♦ `.x WenuJetsRecoGen.C+(0, "all", 0)`
    - ◊ Same parameters as above
- For background specific plots (ABCD, matrix, and file for fitting)
  - ♦ `.L WenuJetsTree.C+`
  - ♦ `.x WenuJetsBackgrounds.C+(0, "all", 0)`
    - ◊ Same parameters as above
- For fitting:
  - ♦ First run WenuJetsBackgrounds.C, which outputs `fitFile-*.root`, where * is the name given in 2nd parameter
  - ♦ `.x WenuJetsRooDataFit.C("fitFile-*.root", "1", 15, "w_Mt", 5, 20, false)`
    - ◊ (string) fileName
    - ◊ (int) number of jets
    - ◊ (double) lower limit to fit
    - ◊ (string) variable type ("w_Mt" or "MET")

◊ (int) number of pseudodatasets to run over,
◊ (int) electron pt cut used
◊ (bool) whether to vary the proportion of W events relative to background

Strange Errors

Do cmsRun, get this: `python encountered the error: 'tuple' object has no attribute 'find'`

- Check for a comma after a proces.[stuff] entry

Log

My work log is KiraGroggLog

-- KiraGrogg - 04 Nov 2008

This topic: Sandbox > KiraGroggSandbox
Topic revision: r56 - 2011-08-17 - unknown