山东大学 SHANDONG UNIVERSITY

Department of Physics, Shandong University

Setup and C1C1 analysis

Chengxin Liao
liaocx@ihep.ac.cn

Feb, Wed 26, 2025

Setup

First thing you need to do is check the computing platform introduction and familiar with linux, more detail can find here http://afsapply.ihep.ac.cn/cchelp/en/

Linux tool: https://missing.csail.mit.edu/2020/

I think every rookie in computer field have to see this

In short is a file can autoexcute when users log in

Here should be your username

Second thing is modify your own computing environment

1. create a file called **.bashrc and .bash_profile** at you home dir, full path is here: /afs/ihep.ac.cn/users/c/chengxinliao
more detail: https://www.digitalocean.com/community/tutorials/bashrc-file-in-linux

2. your work dir at: /publicfs/atlas/atlasnew/SUSY/users/chengxinliao (This is my work dir🤓)

3. you can cd to work dir can creat a file called **setup_initiate.sh**

Setup

Some tools need to know:

based on we have LLM, program can be easy for us, but we still need have basic cognition about these tool

1. C/Cpp/python, just a tool, as a rookie, you probably don't need to know the detail, but you have to build a frame about it

2. Latex: I think I don't need to introduce it

3. GNU make & CMake: maybe you don't need to know how to write, but at least you need to know why we need this file

4. Git & Github: code version !!!(Introduce in Missing semester, website link at page 1 in this sildes)

5. VPN(25105 19981 30693 36947 36825 20010 26159 20160 20040)

Great class I recommend:

UCB CS61a(python and structure): https://cs61a.org

Stanford CS106L(Cpp): https://web.stanford.edu/class/cs106l/

UCB CS189(ML): https://eecs189.org

Actually I don't want to put too much burden on you, so just get a superficial understanding will be cool

Hope you have a wonderful journey in these courses

Details about setup file

.bashrc full file can check: /afs/ihep.ac.cn/users/c/chengxinliao/.bashrc

```
export PATH=/cvmfs/common.ihep.ac.cn/software/hepjob/bin:$PATH
export PATH=/cvmfs/container.ihep.ac.cn/bin:$PATH
export PATH=$PATH:/afs/ihep.ac.cn/users/c/chengxinliao/.local/bin
```

C and Cpp include/lib path

```
export  C_INCLUDE_PATH=/publicfs/atlas/atlasnew/SUSY/users/chengxinliao/MiniAnalysis/include:/publicfs/atlas/atlasnew/SUSY/users/chengxinliao/MiniAnalysis/src:$C_INCLUDE_PATH
export CPLUS_INCLUDE_PATH=/publicfs/atlas/atlasnew/SUSY/users/chengxinliao/MiniAnalysis/include:/publicfs/atlas/atlasnew/SUSY/users/chengxinliao/MiniAnalysis/src:$CPULS_INCLUDE_PATH

export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase
alias setupATLAS='source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh’
alias setupATLAS6='source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh -c slc6’

source /publicfs/atlas/atlasnew/SUSY/users/chengxinliao/setup_initiate.sh
lsetup cmake

alias liao='cd /publicfs/atlas/atlasnew/SUSY/users/chengxinliao'
```

Help you cd to work dir, remember change name

Details about setup file

setup_initiate.sh full detail can check: /publicfs/atlas/atlasnew/SUSY/users/chengxinliao/setup_initiate.sh

setupATLAS
lsetup "views LCG_102b_ATLAS_22 x86_64-centos9-gcc11-opt”
unset PYTHONHOME
export Dir_MiniAnalysis="/publicfs/atlas/atlasnew/SUSY/users/chengxinliao/MiniAnalysis/”
export Dir_BOOST="/cvmfs/sft.cern.ch/lcg/releases/Boost/1.72.0-7bbce/x86_64-centos7-gcc8-opt”

.bash_profile

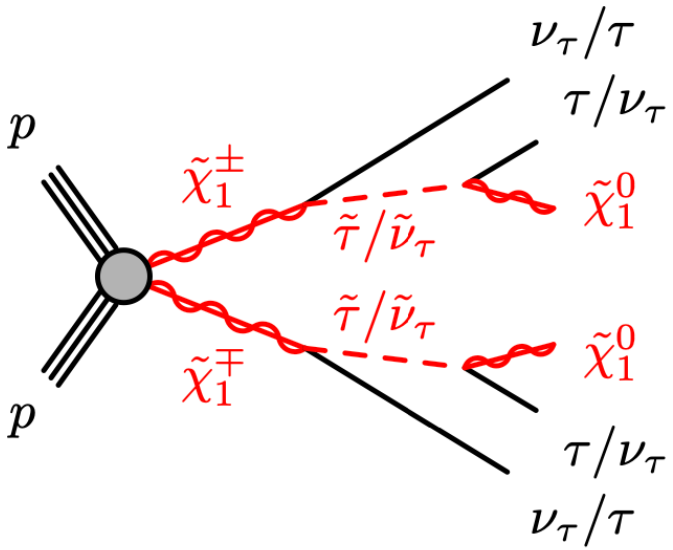
source ~/.bashrc

About C1C1 analysis

process of analysis

selecte powerful var to separate sig and bkg

pre-selection -> MC_modeling check -> kinematic distribution -> CutCount and define signal region–> bkg estimation and error analysis(data-driven, ABCD method)



== 2 medium tau(OS)
b-jet veto
Z/H veto(M_tautau) > 120GeV
asymmetric di-tau trigger
MET < 150GeV

Signal choose: $m(\tilde{\chi}_1^{\pm}, \tilde{\chi}_1^0) = (200, 50), (300, 150), (387.5, 237.5), (425, 275)\ \ GeV$

slimmer

Code can be find at /publicfs/atlas/atlasnew/SUSY/users/chengxinliao/Run2-C1C1/easycode/slimmer.cxx

slimmer actually just a code that you can apply cut

Loading bkg sample and signal

```
std::map<std::string,std::set<std::string>> bkg_dict={
    {"Vqq",{"Sh_2211_Vqq"}},
    {"Wjets",{"Sh_2211_W","Sherpa221_W"}},
    {"Zjets",{"Sh_2211_Z","Sherpa221_Z"}},
    {"MultiBoson",{"Sherpa_VV"}},
    {"TopQuark",{"single_top","multi_t","ttbar","ttV","ttX","tVV","tZ"}},
    {"Higgs",{"Higgs"}},
    {"Data",{"data","Data"}},
//    {"QCD",{"QCD","qcd"}}
};
for(auto bkg_set : bkg_dict){
    //bkg_set.second is the set of keywords
    for(auto keyw : bkg_set.second){
        if(proc.find(keyw) != string::npos){
            //if the proc contains the keyword, then set the proc to the background name
            proc = bkg_set.first;
            break;
        }
    }
}
```

Process the name of file, if you don't get it, can check the log file

```
//get the full name of input file
    std::string outFullName = getOutName();
LOG(INFO) << "outFullName: " << outFullName ;
//initialize a string var to store the type the data
std::string proc = "null";
//check if the input file name contains "XAMPP", if so, get the substring after "_"
if (outFullName.find("XAMPP") != string::npos){
    proc = outFullName.substr(outFullName.find("_"));
}else {
    //or use the whole name
    proc = outFullName;
}
LOG(INFO) << "current process name: " << proc;
```

In the output of slimmer, the log file

```
^[[36m[INFO | 11:15:34 | main.cxx:45] ********** Analysis Start **********
^[[0m^[[36m[INFO | 11:15:34 | MiniAnasrc.cxx:68] Files to analyze: /publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000484.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000919.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000231.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000429.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000798.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000065.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000664.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000761.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000474.root,/publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP/user.chzhu.23926555.XAMPP._000929.root
^[[0m^[[36m[INFO | 11:15:34 | MiniAnasrc.cxx:101] Set the output file name as user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP.0.root
^[[0m^[[36m[INFO | 11:15:34 | MiniAnasrc.cxx:113] Select analysis: slimmer
^[[0m^[[36m[INFO | 11:16:13 | AnaConfigReader.h:89] Make Ntuple Option: On
^[[0m^[[36m[INFO | 11:16:13 | SingleTreeRunner.h:25] Running 1 trees with 1 thread
^[[0m^[[36m[INFO | 11:16:15 | SingleTreeLooper.h:115] *********Start cutting tree Staus_Nominal*********
^[[0m^[[36m[INFO | 11:16:15 | slimmer.cxx:18] outFullName: user.chzhu.XAMPP.HadHad.Prod_V4_Syst16e.SherpaZ21_Ztautau_XAMPP.0.root
^[[0m^[[36m[INFO | 11:16:15 | slimmer.cxx:25] current process name: _V4_Syst16e.SherpaZ21_Ztautau_XAMPP.0.root
^[[0m^[[36m[INFO | 11:16:15 | slimmer.cxx:59] currentTree name: Staus_Nominal
^[[0m^[[36m[INFO | 11:16:45 | SingleTreeLooper.h:128] *********End of cutting Staus_Nominal, cleaning all objs*********
^[[0m^[[36m[INFO | 11:16:45 | main.cxx:52] ********** Analysis End. Total run time is: 1m, 10s, 740ms **********
^[[0m
```

```
    if(proc.find("_") != string::npos){
        //if the proc contains "_", then get the substring between "Stau" and "_"
        proc = proc.substr(proc.find("Stau"), (proc.rfind("_") - proc.find("Stau") ));
    }else{
        //if the proc does not contain "_", then get the substring between "Stau" and "."
        proc = proc.substr(proc.find("Stau"), (proc.rfind(".") - proc.find("Stau") ));
    }
```

slimmer

Here set the input vars, which must can be find in branch

```
vector<string> varNames = {
    "totalWeight",
    "bVeto", "MET", "bNumber",
    "nBaseTau", "nBaseEle", "nBaseMuon", "nBaseLep",
    "nTaus", "nEles", "nMuons", "nLeps", "nJets", "nTTaus","nMTaus","nLTaus","nVLTaus",
        "mergedRunNumber", "mlumiBlock",
    "Minvtt", "dRtt", "dPhitt", "MT2tt", "OS2TAU","dEtatt",
    "tau1_pt","tau1_eta","tau1_phi","mT_tau1",
    "tau2_pt","tau2_eta","tau2_phi","mT_tau2",
    "Meff",
};
```

```
oTree->Branch("MET", &Var["MET"]);
oTree->Branch("Minvtt", &Var["Minvtt"]);
oTree->Branch("dRtt", &Var["dRtt"]);
oTree->Branch("dPhitt", &Var["dPhitt"]);
oTree->Branch("MT2tt", &Var["MT2tt"]);
oTree->Branch("dEtatt", &Var["dEtatt"]);
oTree->Branch("tau1Pt", &Var["tau1_pt"]);
oTree->Branch("tau1Phi", &Var["tau1_phi"]);
oTree->Branch("tau1Mt", &Var["mT_tau1"]);
oTree->Branch("tau2Pt", &Var["tau2_pt"]);
oTree->Branch("tau2Phi", &Var["tau2_phi"]);
oTree->Branch("tau2Mt", &Var["mT_tau2"]);
oTree->Branch("meff", &Var["Meff"]);
oTree->Branch("nJets", &Var["nJets"]);
oTree->Branch("nTTaus", &Var["nTTaus"]);
oTree->Branch("nMTaus", &Var["nMTaus"]);
oTree->Branch("nLTaus", &Var["nLTaus"]);
oTree->Branch("nVLTaus", &Var["nVLTaus"]);
oTree->Branch("Weight_mc", &Var["totalWeight"]);
```

generate new branch in the root file of result

slimmer

Create the branch and Calculate the var you gone put into branch

```
//
  Var["MET"] =  tree->MetTST_met / 1000;
Var["mcChannelNumber"] = tree->mcChannelNumber;
Var["evtNum"] = tree->eventNumber;
// weight
if(tree->mcChannelNumber == 0){     //表示Data, totalWeight为1
    Var["totalWeight"] = 1;
        Var["mergedRunNumber"] = tree->runNumber;
        Var["mlumiBlock"] = tree->lumiBlock;
    // Var["totalWeight"] *= 0.782692393;//purity
}else{
        Var["mergedRunNumber"] = tree->RandomRunNumber;
        Var["mlumiBlock"] = tree->RandomLumiBlockNumber;
}
//caculate extraweight
if(tree->mcChannelNumber != 0 ){
    double extraWei = tree->GenWeight*tree->muWeight*tree->TauWeight*tree->EleWeight*tree->MuoWeight*tree->JetWeight;
    Var["totalWeight"] = MetaDB::Instance().getWeight(PhyUtils::getCompaign(Var["mergedRunNumber"]), tree->mcChannelNumber, tree->SUSYFinalState, extraWei);
    // Var["totalWeight"] *= -1 ;//用于ABC时的SM本底
}
```

```
oTree->Branch("MET", &Var["MET"]);
oTree->Branch("Minvtt", &Var["Minvtt"]);
oTree->Branch("dRtt", &Var["dRtt"]);
oTree->Branch("dPhitt", &Var["dPhitt"]);
oTree->Branch("MT2tt", &Var["MT2tt"]);
oTree->Branch("dEtatt", &Var["dEtatt"]);
oTree->Branch("tau1Pt", &Var["tau1_pt"]);
oTree->Branch("tau1Phi", &Var["tau1_phi"]);
oTree->Branch("tau1Mt", &Var["mT_tau1"]);
oTree->Branch("tau2Pt", &Var["tau2_pt"]);
oTree->Branch("tau2Phi", &Var["tau2_phi"]);
oTree->Branch("tau2Mt", &Var["mT_tau2"]);
oTree->Branch("meff", &Var["Meff"]);
oTree->Branch("nJets", &Var["nJets"]);
oTree->Branch("nTTaus", &Var["nTTaus"]);
oTree->Branch("nMTaus", &Var["nMTaus"]);
oTree->Branch("nLTaus", &Var["nLTaus"]);
oTree->Branch("nVLTaus", &Var["nVLTaus"]);
oTree->Branch("Weight_mc", &Var["totalWeight"]);
```

slimmer

About add Cut

registerCut function definition at: /publicfs/atlas/atlasnew/SUSY/users/chengxinliao/MiniAnalysis/src/Cutflow.cxx

```
mCutflow->registerCut("baseline",[&] {return fabs(Var["totalWeight"]) < 1000 ; });
//mCutflow->registerCut("10 < MET < 150 GeV", [&] {return (Var["MET"] <=150 )&&(Var["MET"] >=10 );},
mCutflow->registerCut("60 < MET < 150 GeV", [&] {return (Var["MET"] <=150 )&&(Var["MET"] >=60 );}, "MET",80,0,400,[&]{return Var["MET"];});
mCutflow->registerCut("trigPass", [&] {return Var["TriggerPass"];});
mCutflow->registerCut("120 < Minvtt GeV", [&] {return (Var["Minvtt"] >= 120) ;},"Minvtt",80,0,800,[&]{return Var["Minvtt"];});
mCutflow->registerCut("B veto", [&] {return Var["bVeto"];},"bNumber",3,0,3,[&]{return Var["bNumber"];});
mCutflow->registerCut("80 < MT2tt",[&] {return (Var["MT2tt"] >= 80);},"MT2tt",160,0,160, [&]{return Var["MT2tt"] ;});//EE //从本底来看可以小部分除去W+jets(Nb)，可以出去小部分Di-Boson(dEta)从而消去部分统计误差，这部分由Di-BOSON引起
//mCutflow->registerCut(" 15 < MT2tt < 35 GeV",[&] {return (Var["MT2tt"] <= 35)&&(Var["MT2tt"] >= 15);},"MT2tt",160,0,160, [&]{return Var["MT2tt"] ;});//BC
//mCutflow->registerCut("85 < MT2tt",[&] {return (Var["MT2tt"] >= 85);},"MT2tt",160,0,160, [&]{return Var["MT2tt"] ;});

mCutflow->registerCut("== 2 medium tau", [&] {return Var["nMTaus"] == 2;},"NMTaus",4,0,4,[&]{return Var["nMTaus"];});


//mCutflow->registerCut(">= 1 tight tau", [&] {return Var["nTTaus"] >= 1;},"NTTaus",3,0,3,[&]{return Var["nTTaus"];});//CDF
//draw SS and OS
mCutflow->registerCut("OS", [&] {return Var["OS2TAU"];},"OS2TAU",2,0,2,[&]{return Var["OS2TAU"];});//not the biggest one


// mCutflow->registerCut("== 2 very loose tau", [&] {return Var["nVLTaus"] == 2;},"NVLTaus",4,0,4,[&]{return Var["nVLTaus"];});
//mCutflow->registerCut("<= 2 medium tau", [&] {return Var["nMTaus"] <= 2;},"NMTaus",4,0,4,[&]{return Var["nMTaus"];});//ABE
//

// mCutflow->registerCut("no light lepton", [&] {return Var["nLeps"] == 0;},"NLep",4,0,4,[&]{return Var["nLeps"];}); //light lepton veto
mCutflow->registerCut("1.4 < dPhi  ", [&] {return Var["dPhitt"] >= 1.4;},"dPhitt",16,0,3.2,[&]{return Var["dPhitt"];});
//mCutflow->registerCut("1.55 < dPhi  ", [&] {return Var["dPhitt"] >= 1.55;},"dPhitt",16,0,3.2,[&]{return Var["dPhitt"];});

mCutflow->registerCut("dR < 2.6",[&] {return Var["dRtt"] <= 2.6;},"dRtt",45,0,4.5, [&]{return Var["dRtt"] ;});
//mCutflow->registerCut("dR < 3.2",[&] {return Var["dRtt"] <= 3.2;},"dRtt",45,0,4.5, [&]{return Var["dRtt"] ;});

//mCutflow->registerCut("80 < MT2tt GeV",[&] {return Var["MT2tt"] >= 80;},"MT2tt",160,0,160, [&]{return Var["MT2tt"] ;});//AD
```

First definition

Second definition

```
// if disabled histogram fill
std::shared_ptr<CutIterator> Cutflow::registerCut(std::string cutname, std::function<bool()> cut) {
    checkCutName(cutname);
    std::shared_ptr<CutIterator> newCut(new CutIterator(cut, this->willFillHist));
    newCut->setDirectory(this->histFolder);
    newCut->name = cutname;
    CutQueue.emplace_back(newCut);
    return newCut;
}

// if enabled histogram fill
std::shared_ptr<CutIterator> Cutflow::registerCut(std::string cutname, std::function<bool()> cut, std::string histname, int nBin, double binStart,
                                                  double binEnd, std::function<double()> fillPosition, bool useOverflow) {
    checkCutName(cutname);
    std::shared_ptr<CutIterator> newCut(new CutIterator(cut, this->willFillHist));
    newCut->setDirectory(this->histFolder);
    newCut->set_Before_Hist(histname, nBin, binStart, binEnd, fillPosition, useOverflow);
    std::string histname_N_1 = histname + "_N_1";
    newCut->set_N_1_Hist(histname_N_1, nBin, binStart, binEnd, fillPosition, useOverflow);
    newCut->name = cutname;
    CutQueue.emplace_back(newCut);
    return newCut;
}
```

First definition will return the N plot
Second definition will return the N plot and N-1 plot

slimmer

About run :

Please ref the code of wenyi in run.sh

```
#python3 /publicfs/atlas/atlasnew/SUSY/users/rabiahameed/jinwenyi/MiniAnalysis/python/subJob.py -a slimmer -i /publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/Data -s -l 10 -n NOM_data_pSR_R1

#python3 /publicfs/atlas/atlasnew/SUSY/users/rabiahameed/jinwenyi/MiniAnalysis/python/subJob.py -a slimmer -i /publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM -s -l 10 -n NOM_SM_pSR_R1

#python3 /publicfs/atlas/atlasnew/SUSY/users/rabiahameed/jinwenyi/MiniAnalysis/python/subJob.py -a slimmer -i /publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/C1C1/ -s -l 10 -n NOM_C1C1_test_R1

#python3 /publicfs/atlas/atlasnew/SUSY/users/rabiahameed/jinwenyi/MiniAnalysis/python/subJob.py -a slimmer -i /publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM -s -l 10 -n NOM-SM-pSR

python3 /publicfs/atlas/atlasnew/SUSY/users/jinwenyi/Run2-C1C1/MiniAnalysis/python/subJob.py -a slimmer -i /publicfs/atlas/atlasnew/SUSY/users/jinwenyi/DS_Tutorial/inputs/NOM/SM/ -s -l 10 -n test_run
```

subJob.py is a scripts help you submit job
-a means analysis
-I means input dir
-n means output dir
other argument can check the subJob.py

CutCount

Full code address: /publicfs/atlas/atlasnew/SUSY/users/chengxinliao/Run2-C1C1/CutCount

Method Principle: CutCount is a method to define your signal region, it can calculate the significance and yields of different cut combination, below setting is combination of input

```
void CutCount::setCutSteps() {
    // addStep("nTTaus", {0, 1}, '>');
    addStep("MET", {30,50,55}, '>');
    //addStep("MT2tt",{40,50,60},'>');
    addStep("dRtt",{2.4,2.6,},'<');
    addStep("dPhitt",{0,0.5,1,1.2,1.4,1.5},'>');
    // addStep("Minvtt",{120,125,130,135,140,145,150,155},'>');
    //addStep("Meff",{0,300,400,500,600,700},'>');

    // addStep("tau1_pt",{95, 100, 110, 120, 130, 140, 150},'>'); // trigger requirement is enough for tau1Pt,tau2Pt
    // addStep("tau2_pt",{60, 70, 80, 90, 100},'>');
    //addStep("MTsum",{0, 200, 250, 300, 350},'>');
}
```

MET > 30
dRtt < 2.4 ……..
dPhitt > 0.5

More details can check MiniAnalysis/include/CutCountRunner.h

```
^[[36m[INFO | 09:58:55 | main.cxx:45] ************************* Analysis Start *************************
^[[0m^[[36m[INFO | 09:58:55 | MiniAnasrc.cxx:68] Files to analyze: ../preCUT.root
^[[0m^[[36m[INFO | 09:58:55 | MiniAnasrc.cxx:101] Set the output file name as preCUT.root
^[[0m^[[36m[INFO | 09:58:55 | MiniAnasrc.cxx:113] Select analysis: CutCount
^[[0m^[[36m[INFO | 09:59:34 | CutCountRunner.cxx:37] Read Bkg Sample:MultiBoson_Nom
^[[0m^[[36m[INFO | 09:59:39 | CutCountRunner.cxx:37] Read Bkg Sample:Higgs_Nom
^[[0m^[[36m[INFO | 09:59:43 | CutCountRunner.cxx:37] Read Bkg Sample:Wjets_Nom
^[[0m^[[36m[INFO | 09:59:44 | CutCountRunner.cxx:37] Read Bkg Sample:Zjets_Nom
^[[0m^[[36m[INFO | 09:59:55 | CutCount.cxx:7] 100000 entry of 133204 entries. E.T.A. for this tree: 3s, 791ms
^[[0m^[[36m[INFO | 10:00:00 | CutCountRunner.cxx:37] Read Bkg Sample:TopQuark_Nom
^[[0m^[[31m[ERROR | 10:00:01 | CutCountRunner.cxx:34] No tree named QCD_Nom is found in the target processed trees. This tree is skiped
^[[0m^[[36m[INFO | 10:00:01 | CutCountRunner.cxx:43] Read Signal Sample:Stau_300p0_100p0_Nom
^[[0mFilter
^[[36m[INFO | 10:00:01 | CutCountRunner.cxx:53] Prepare CutCount pools...
^[[0m^[[36m[INFO | 10:00:01 | CutCountRunner.cxx:66] The CouCount will run 6 * 2 * 3 = 36 steps
```

36 combinations
based on operator you choose

Grid setting I suggest ref wenyi's code and fine tuning

CutCount

You also can add the filter to your result

```
void CutCount::setResultFilters() {
    std::cout<<"Filter"<<std::endl;
    addYieldsFilter("totalBkg", 3);   //totalBkg
    addYieldsFilter("Zjets_Nom",1);
    addRelErrorFilter("MultiBoson_Nom",0.5);
    addRelErrorFilter("totalBkg", 0.2);
    //setZnFilter(0.5); // > 0.5
}
```

Before you start to set cut, you have to told complier where can find the var

```
void Looper_CutCount::setVariables() {
    //setVar("nTTaus", [&] { return tree->nTTaus; });
    setVar("MET",  [&] { return tree->MET; });
    //setVar("MT2tt",[&] { return tree->MT2tt; });
    //setVar("Meff", [&] { return tree->meff; });
    setVar("dRtt", [&] {return tree->dRtt;   });
    // setVar("dEtatt", [&] {return tree->dEtatt; });
    setVar("dPhitt", [&] {return tree->dPhitt; });
    //setVar("Minvtt", [&] {return tree->Minvtt; });
    //setVar("tau1Pt", [&] {return tree->tau1Pt; });
    //setVar("tau2Pt", [&] {return tree->tau2Pt; });
    setWeight([&] { return tree->Weight_mc; });
   //setVar("MTsum", [&] {return tree->MTsum; });
}
```

So, maybe for now, you already familiar with structure of CutCount
Let's try to run it 🗣️

CutCount

Before start to run, thers are some file we need to prepare

```
build  CMakeLists.txt  config  CutCount.cxx  preCUT.root  run  showZn.py  subJOB.sh
 jobs:0 /publicfs/atlas/atlasnew/SUSY/users/chengxinliao/Run2-C1C1/CutCount
```

preCUT.root contain the result of your pre-selection

```
KEY: TTree    Stau_200p0_50p0_Nom;1   SmallTree for fast analysis
KEY: TH1F     Minvtt_N_1;1    Minvtt_N_1
KEY: TH1F     bNumber_N_1;1   bNumber_N_1
KEY: TH1F     MT2tt_N_1;1     MT2tt_N_1
KEY: TH1F     NMTaus_N_1;1    NMTaus_N_1
KEY: TH1F     NTTaus_N_1;1    NTTaus_N_1
KEY: TH1F     OS2TAU;1        OS2TAU
KEY: TH1F     OS2TAU_N_1;1    OS2TAU_N_1
KEY: TH1F     Minvtt;1        Minvtt
KEY: TH1F     dRtt;1  dRtt
KEY: TH1F     dPhitt;1        dPhitt
KEY: TH1F     MT2tt;1 MT2tt
KEY: TH1F     NLeps;1 NLeps
KEY: TH1F     tau1_pt;1       tau1_pt
KEY: TH1F     tau1_eta;1      tau1_eta
KEY: TH1F     tau1_phi;1      tau1_phi
KEY: TH1F     mT_tau1;1       mT_tau1
KEY: TH1F     tau2_pt;1       tau2_pt
KEY: TH1F     tau2_eta;1      tau2_eta
KEY: TH1F     tau2_phi;1      tau2_phi
KEY: TH1F     mT_tau2;1       mT_tau2
KEY: TH1F     NVLTaus;1       NVLTaus
KEY: TH1F     NLTaus;1        NLTaus
KEY: TH1F     NMTaus;1        NMTaus
KEY: TH1F     NTTaus;1        NTTaus
KEY: TH1F     NJets;1 NJets
KEY: TH1F     bNumber;1       bNumber
KEY: TH1F     MET;1   MET
KEY: TH1F     dEtatt;1        dEtatt
KEY: TH1F     Meff;1  Meff
KEY: TTree    Stau_300p0_150p0_Nom;1  SmallTree for fast analysis
KEY: TTree    Stau_387p5_237p5_Nom;1  SmallTree for fast analysis
KEY: TTree    Stau_425p0_275p0_Nom;1  SmallTree for fast analysis
KEY: TTree    Data_Nom;1      SmallTree for fast analysis
KEY: TTree    Higgs_Nom;1     SmallTree for fast analysis
KEY: TTree    TopQuark_Nom;1  SmallTree for fast analysis
KEY: TTree    QCD_Nom;1       SmallTree for fast analysis
KEY: TH1F     NVLTaus_N_1;1   NVLTaus_N_1
KEY: TTree    Wjets_Nom;1     SmallTree for fast analysis
KEY: TTree    Zjets_Nom;1     SmallTree for fast analysis
KEY: TTree    MultiBoson_Nom;1        SmallTree for fast analysis
```

Here is your result after passing pre-selection

```
C1C1_Stau_200p0_50p0.root   ggHiggs.root          MG5Py8_tZ.root                  QCD.root              Sherpa221_Zee.root          Sherpa221_ZnunuMJJCVBV.root  ttHiggs.root
C1C1_Stau_300p0_150p0.root  MG5_aMCatNLO_Py8_ttV.root  PowHegPy8_single_top_s_chan.root  Sherpa221_Wenu.root   Sherpa221_Zll2jets.root     Sherpa221_Znunu.root         VBFHiggs.root
C1C1_Stau_387p5_237p5.root  MG5_aMCatNLO_Py8_tVV.root  PowHegPy8_single_top_t_chan.root  Sherpa221_Wlnu2jets.root  Sherpa221_Zmumu.root  Sherpa221_Ztautau.root       VHiggs.root
C1C1_Stau_425p0_275p0.root  MG5Py8_multi_t.root   PowHegPy8_single_top_Wt_chan_incl.root  Sherpa221_Wmunu.root  Sherpa221_ZnunuMJJBF.root   Sherpa_VV.root
data.root                   MG5Py8_ttX.root       PowHegPy8_ttbar_incl.root       Sherpa221_Wtaunu.root     Sherpa221_ZnunuMJJCFBV.root  Sherpa_VVV.root
```

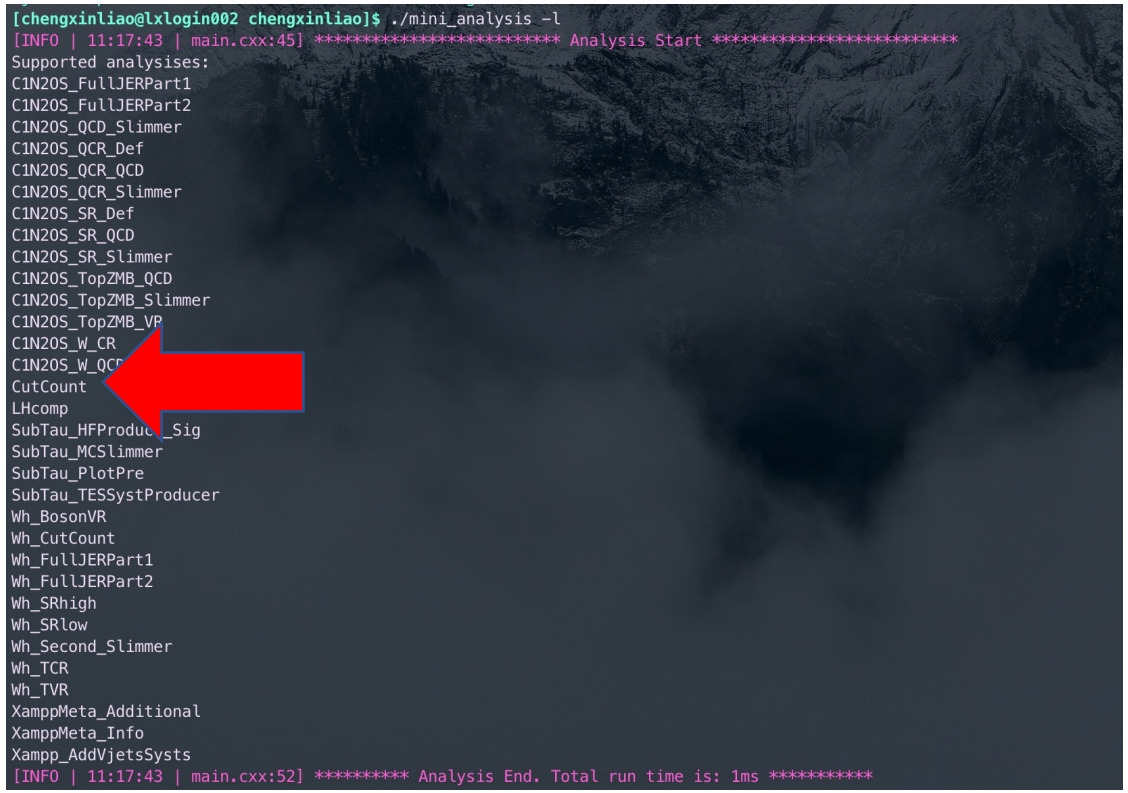
You can merge them with hadd command
more detail about hadd please check here
https://root.cern/doc/v632/hadd_8cxx.html

Please try to merge these file and get your own preCUT.root
acctully that's the only file you need to prepare

CutCount

```
build  CMakeLists.txt  config  CutCount.cxx  preCUT.root  run  showZn.py  subJOB.sh
 jobs:0 /publicfs/atlas/atlasnew/SUSY/users/chengxinliao/Run2-C1C1/CutCount
```

Then you can complie the file manully or by running subJOB.sh file
after you complie with no error, you will get a file called mini_analysis in build dir, and if you already learn something about CMake, you will how to handle it. But if you still not, please please check setup page!!!



subJOB.sh also contain the cmake step which is simple
if you already cmake manully, please comment these lines

```
#mkdir build
#cd build/
#rm -rf *
#cmake ../
#make -j 16
```

BTW, remember delete the run folder before you run, or change the output folder in case there is a warning, because your output folder alread exist

CutCount

Just use hep_sub to submit you job to computing platform, then you can see that

```
cutResults.csv  end.csv  mini_analysis  myeasylog.log  showZn.py
 jobs:0 /publicfs/atlas/atlasnew/SUSY/users/chengxinliao/Run2-C1C1/CutCount/run
```

cutResult.csv contain the raw result based on you cut, and end.csv will show the topZn cut combination

	Zn	Signal	totalBkg	MET	MT2tt	dRtt	dPhitt	Minvtt
6261	0.926522	3.74506 +- 0.710651	6.96156 +- 1.30889(309)	75	70	3.0	1.55	145
6262	0.926522	3.74506 +- 0.710651	6.96156 +- 1.30889(309)	75	70	3.0	1.55	146
6263	0.926522	3.74506 +- 0.710651	6.96156 +- 1.30889(309)	75	70	3.0	1.55	147
6251	0.917609	3.74506 +- 0.710651	7.09055 +- 1.31106(320)	75	70	3.0	1.50	147
6250	0.917609	3.74506 +- 0.710651	7.09055 +- 1.31106(320)	75	70	3.0	1.50	146
6249	0.916294	3.74506 +- 0.710651	7.11004 +- 1.3112(321)	75	70	3.0	1.50	145
6189	0.914288	3.53707 +- 0.690708	6.48258 +- 1.28865(299)	75	70	2.8	1.40	146
6190	0.914288	3.53707 +- 0.690708	6.48258 +- 1.28865(299)	75	70	2.8	1.40	147
6188	0.912906	3.53707 +- 0.690708	6.50207 +- 1.2888(300)	75	70	2.8	1.40	145

I suggest that sig and bkg yields need to large than 3, and bkg uncertainty is less thann 20%

then use the Cut you choose to slimmer.cxx and check the N-1 distribution to see if the result can meet you requirement