

1) Run on minitrees

At this moment, AnalysisStop has two builders, one designed to be use when you read a latino tree, the other designed to be used when you read a minitree. Each of one set the useful functions for the analysis in case. In both of them a new parameter has been included: *Int SaveHistograms* which values set which histograms will be filled:

If

```
saveHistogram == 0  -> Fill all histograms { general, particular}
saveHistogram == 1  -> Fill only analysis histograms { particular}
saveHistogram == 2  -> Fill only analysis histograms && up-down histograms { particular}
```

Finally, the Loop function accept, now, two more parameters: *float StopRefMass*, *float NeutralinoRefMass* . These parameters set the mass point of the signal you want for analyzing. They are usually initialzited to -1 in AnalysisStop.h which sets some values by default.

To run AnalysisStop on minitrees the script **runAnalysisMiniTrees.C** is needed. This is compiled just by `"/make"` but the second line of the script must be uncommented (the usual issue than one has to uncomment its analysis code after installing the package). The samples must have the word "minitrees" in their name, otherwise, the code will not consider them as minitrees samples.

Ex:

To run with runAnalysis.C

blablabla_latino_T2tt_mStop-250to350.root

To run with runAnalysisMiniTrees.C

blablabla/minitrees/nominal/Stop/T2tt_mStop-250to350.root

ó

blablabla_minitrees_T2tt_mStop-250to350.root

Run interactively:

```
./runAnalysisMiniTrees minitree_name.root systematic
./runAnalysisMiniTrees minitree_name.root systematic SaveHistograms
./runAnalysisMiniTrees minitree_name.root systematic SaveHistograms mstop
./runAnalysisMiniTrees minitree_name.root systematic SaveHistograms mStop mLSP
```

It is convenient to have a file listing all minitrees samples followed by the value of numeric parameters:

Ex: my_list_of_minitree.txt

```
minitrees_T2tt_mStop-250to350.root 2 300 125
minitrees_T2tt_mStop-250to350.root 2 300 175
minitrees_T2tt_mStop-250to350.root 2 300 150
```

Where systematic parameter can be passed when sourcing the file:

source my_list_of_minitree.txt nominal

If the minitree is not so small, one can submit a job using the usual script submit-jobs.py, with two differences:

- the string "minitree" must be present both in the name of the file listing the samples (so that the script knows whether it should use runAnalysis or runAnalysisMiniTrees) and again in the "long name" of the minitree (typically it will be in the name of the directory containing the minitree itself).
- The line containing the minitree name can be followed by the appropriate parameters:
minitree_name.root SaveHistograms mStop mLSP
the "systematic" parameter will be given as usual when running submit-jobs.py

It is known that many studies require the “systematic up-down histograms”. These histograms are built using a scale factor + uncertainty up (down) instead of using the nominal value of the scale factor, in the event weight. So, there is as many histograms as systematic uncertainties you have per two for each variable you want considering.

In AnalysisStop.C we are taking account only one variable, but there is a function to define and book any number of histograms **BookSystematicHistograms()**, a function to fill them **FillSystematicHistograms (int ichannel, int icut)** and a function to save them **SaveSystematicHistograms()**

The function *FillSystematicHistograms (int ichannel, int icut)* does a loop over all systematic event weights producing one histogram scale factor + uncertainty up, one down, for each variable, so you need to have stored or available the right systematic event weights. All these functions work with latino trees and minitrees.

Everything in: <https://github.com/piedraj/AnalysisCMS>