

ValidationsTools

Table des matières

Launching the installation :.....3

Launching the ROOT files creation :.....3

 Reducing size :.....3

 Extracting values :.....3

Annexe : ROOT scripts to modify.....5

Annexe : paths definitions.....5

Annexe : values for ROOT creation.....5

this repository get all the validations tools such as Kolmogorov-Smirnov (KS Tools) or AutoEncoders Tools (AE).

Launching the installation :

```
- git clone https://github.com/archiron/ValidationsTools ValidationsTools
- cd ValidationsTools
- chmod 755 *.sh
then launch : . installValidationsTools.sh
```

This install the library (ChiLib), the release file env (cmsrel \$Release \$Release) and copy the ZEE_Flow/CCA or ZEE_Flow/LLR files (depending of the site you are working on - the CC or the LLR site) onto the \$Release/src/Kolmogorov folder.
\$Release is the release you want working with.

Launching the ROOT files creation :

Talk about the modifications into the createROOTFiles script (nb of files, ...).


From the top of the Tools (i.e. ValidationsTools), then launch :

```
. createROOTFiles.sh
```

This script launch the creation of the ROOT files, using the step[1-4].py scripts.

You can use own but they have to be similar to the existing ones (see 5).

verifying the creation of the wanted RESULT path ? When create it ?

 later create a link to an explanation of the common files used for the ROOT files creation (CommonFiles/path...).

Reducing size :

When ALL ROOT files are created, launch (always from the top folder) :

```
. reduceROOTSize.sh
```

this will reduce the size of the ROOT files (typically from 150/200 Mo to 1.5/2 Mo), keeping the name of the file.

Extracting values :

When all the ROOT files are created and reduced, we need to creates 1 file per histo with all the histo values for each ROOT file.

It is the first job of the extractValues[_init].sh scripts.

It can be launched with : . extractValues_init.sh

and create into the RESULTFOLDER (see 5) a lot of text files. All ROOT files are read, and then for each histo, the values of the histo curve is stored into an array, ROOT file after ROOT file.

Once we have all the ROOT files read, the array is stored into a text file dedicated to this histo.

The name is : RESULTFOLDER + 'histo_' + str(leaf) + '_' + '{:03d}'.format(nbFiles) + '.txt' where leaf is the name of the histo, nbFiles the number of the ROOT files (same as NB_EVTS in rootValues.py - see 5).

Example :

| |
|-------------------------------|
| histo_h_ele_zEff_200.txt |
| histo_h_ele_seedDphi2_200.txt |

Once you have created all those files, you will need to think about comparison. In KS you have a reference release, and one or more release(s) to compare with (see the beginning of this guide).

ANNEXE

In that follow, we have the following notation :

al numeric values (int/float) are in **blue**.

Annexe : ROOT scripts to modify

Here we are talking about the step[1-4].py scripts.

Annexe : paths definitions

RESULTFOLDER : path where you want the created ROOT files are located. It also contain the text files for each histo.

Annexe : values for ROOT creation

For the creation of the ROOT files we need python files with physics and for those files we need some values.

In general we need a lot files and not only one. So, those file went from **Nbegin = 0** to **Nend=200** or **1000**.

For a lot of tests it can be nice to have smaller values such as :

Nbegin = 20

Nend = 23

Inside the python files we nneed to know about the number of events to be used. Most of the « official » runs use **NB_EVTS = 9000**. For some rapid tests we can use :

NB_EVTS = 10