ICTP-NCP School on LHC Physics | (smr 2632)

Prepared by: Taimoor Khurshid & Muhammad Ahmad

Course material: From CMSDAS

Introduction

Welcome to the 4th School on LHC Physics exercises. The purpose of these exercises is to become familiar with the basic software tools required to perform physics analysis at the school. Please run and complete these exercises. Questions for each exercise are in red font. A large amount of additional information about these exercises is available in the twikis that we reference. Please remember that twikis evolve but aim to provide the best information at any time. If, at any time problems are encountered please ask.

Exercise 1 - Cut and Paste

Note: This exercise is designed to run only on **lxs** as the scripts are present there. Elsewhere, one can certainly copy the script and try. Your login names and passwords are provided to you.

Please login as follows:

your_user_name@lxs.ncp.edu.pk

enter your password.

Please cut and paste the following and then hit return

```
./runThisCommand.py "asdf;klasdjf;kakjsdf;akjf;aksdljf;a"
```

The response should be your username followed by alphanumeric string of characters unique to your username, for example for a user named taimoor:

```
success: taimoor gnvzbbe
```

QUESTION 1 - Post the alphanumeric string of characters unique to your username.

If the command is executed without any cut and paste:

```
user@lxs01>./runThisCommand.py
```

the result will likely be:

```
Error: You must provide the secret key
```

Pasting incorrectly, the result will likely be:

```
Error: You didn't paste the correct input string
```

Exercise 2 - Simple Edit Exercise

Note: This exercise is designed to run only on **lxs** as the scripts are present there. Elsewhere, one can certainly copy the script and try.

Please note: in the following

if don't find the "editThisCommand.py" in your home you can copy it from

```
/scratch/t/taimoor/LHC_SCHOOL_4/course_material
```

The purpose of this exercise is to ensure that the user can edit files.

```
cp /scratch/t/taimoor/LHC_SCHOOL_4/course_material/editThisCommand.py .
```

Then open `editThisCommand.py` and make sure that the 11th line has # (hash character) as the first character of the line. If not, explicitly change the following three lines:

```
# Please comment the line below out by adding a '#' to the front of
# the line.
raise RuntimeError, "You need to comment out this line with a #"
```

to:

```
# Please comment the line below out by adding a '#' to the front of
# the line.
#raise RuntimeError, "You need to comment out this line with a #"
```

Save the file and execute the command:

```
user@lxs01> ./editThisCommand.py
```

If this is successful, the result will be:

```
success:  taimoor 0x468263E
```

QUESTION 2 - Paste the line beginning with "success" into the form provided.

If the file has not been successfully edited, an error message will result such as:

```
Traceback (most recent call last):
   File "./editThisCommand.py", line 11, in ?
```

```
    raise RuntimeError, "You need to comment out this line with a #"
RuntimeError: You need to comment out this line with a #
```

Exercise 3 - Setup a release area CMSSW_5_3_13

```
mkdir YOURWORKINGAREA
cd YOURWORKINGAREA
cmsrel CMSSW_5_3_13
cd CMSSW_5_3_13/src/
cmsenv
```

Note: The directory YOURWORKINGAREA/CMSSW_5_3_13/src is called your WORKING DIRECTORY.

Exercise 4 - Find data in the DAS (Data Aggregation Service)

In this exercise we will locate the MC dataset **RelValZMM** and the collision dataset **/SingleMu/Run2012A-recover-06Aug2012-v1/AOD** using the **Data Aggregation Service**. Also remember DAS is an improved (faster) database access service previously known as DBS (Dataset Bookkeeping Seystem).

Go to the url https://cmsweb.cern.ch/das/ and type in the space provided:

```
dataset release=CMSSW_5_3_6 dataset=/RelValZMM/*CMSSW_5_3_6*/GEN-SIM-
RECO
```

This will search for a dataset, processed with release `CMSSW_5_3_6`, which is named like `/RelValZMM*/*/GEN-SIM-RECO`. The syntax for searches is found https://cmsweb.cern.ch/das/faq, with many useful common search patterns under "CMS Queries".

For this query, hit enter on the dataset name and after a few seconds another page will appear.

QUESTION 4.1 - What is the size of this data set? Click on "Sites" to get a list of sites hosting this data. Is this data at FNAL?

Click on the link "Files" to get a list of the root files in this dataset. One of the files it contains should look like this:

```
        '/store/relval/CMSSW_5_3_6-START53_V14/RelValZMM/GEN-SIM-
RECO/v2/00000/08C1D822-F629-E211-A6B1-003048679188.root',
.................................................
```

If you want to know the name of the dataset from the name of a file, one can go to DAS and type `dataset`

```
file=/store/relval/CMSSW_5_3_6-START53_V14/RelValZMM/GEN-SIM-
   RECO/v2/00000/08C1D822-F629-E211-A6B1-003048679188.root
```

in the command line and hit "Enter".

Now we will locate a collisions dataset skim using the keyword search, which is sometimes more, convenient if you know the dataset you are looking for. In https://cmsweb.cern.ch/das/, type

```
dataset=/SingleMu/Run2012A-recover-06Aug2012-v1/AOD
```

and hit Enter. Answer the following question:

QUESTION 4.2 - What release was this dataset collected in? (If you see more than one release, just answer one)

Having set your CMSSW environment one can also search for the dataset

/SingleMu/Run2012A-recover-06Aug2012-v1/AOD

by invoking the DAS command in your `WORKING DIRECTORY` following the directions https://cmsweb.cern.ch/das/faq

```
curl -k https://cmsweb.cern.ch/das/cli > das_client.py
```

clicking on "DAS Command Line Tool". Use your browser to open the "CLI" macro and download it, following the instructions to change the name to `das_client.py`. Remember to `chmod a+x das_client.py`. The query we're interested in is:

```
./das_client.py --query="dataset= /SingleMu/Run2012A-recover-06Aug2012-
v1/AOD | grep dataset.name" --format=plain
```

Result:

```
./das_client.py --query="dataset= /SingleMu/Run2012A-recover-06Aug2012-
v1/AOD | grep dataset.name" --format=plain

Showing 1-10 out of 1 results, for more results use --idx/--limit
options  /SingleMu/Run2012A-recover-06Aug2012-v1/AOD
```

More information about accessing data in the https://cmsweb.cern.ch/das/faq can be found in
https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookDataSamples

Some links below will help you keep up to date with different physics datasets:

https://twiki.cern.ch/twiki/bin/viewauth/CMS/PdmV2012Analysis

https://twiki.cern.ch/twiki/bin/view/CMS/PdwgMain

https://twiki.cern.ch/twiki/bin/view/CMS/PhysicsSecondaryDatasets

https://twiki.cern.ch/twiki/bin/view/CMS/PhysicsCentralSkims

Exercise 5 - EDM (Event Data Model framework) standalone utilities - `edmFileUtil`, `edmDumpEventContent`, `edmProvDump`, `edmEventSize`

The overall collection of CMS software, referred to as CMSSW, is built around a Framework, an Event Data Model (EDM see link: https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookCMSSWFramework#EdM), and Services needed by the simulation, calibration and alignment, and reconstruction modules that process event data so that physicists can perform analysis. The primary goal of the Framework and EDM is to facilitate the development and deployment of reconstruction and analysis software. The CMS Event Data Model (EDM) is centered around the concept of an Event. An Event is a C++ object container for all RAW and reconstructed data related to a particular collision.To understand what is in a data file and more, several EDM utilities are available. In this exercise, one will use three of these EDM utilities. They will be very useful at CMSDAS and after. More about these EDM utilities can be found at WorkBookEdmUtilities(see link: https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookEdmUtilities). These together with the CVS web interface from CMSSW (see link: http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/CMSSW) and the CMS LXR Cross Reference(see link: http://cmslxr.fnal.gov/lxr/source/) are very useful to understand and write CMS code.

edmFileUtil

First we will use the `edmFileUtil` to find the **physical file name** (PFN) where the file is actually stored at your site, given the **logical file name** (LFN) which is an alias that can be used in CMSSW at any site.

Use `edmFileUtil` to find the **physical file name** (PFN) corresponding to the **logical file name** (LFN) from the AOD (see Twiki: https://twiki.cern.ch/twiki/bin/view/CMS/AOD) data file located above.

To do this execute

```
edmFileUtil -d /store/data/Run2012A/SingleMu/AOD/recover-06Aug2012-
v1/0000/10816437-A0E0-E111-B70E-D8D3855BBDC4.root
```

If you are working on lxs this will return:

```
rfio:/dpm/ncp.edu.pk/home/cms/store/data/Run2012A/SingleMu/AOD/recover-
06Aug2012-v1/0000/10816437-A0E0-E111-B70E-D8D3855BBDC4.root
```

edmDumpEventContent

Next we will use edmDumpEventContent to dump a summary of the products that are contained within the file we're interested in.

Use edmDumpEventContent to see what class names etc. to use in order to access the objects in the AOD data file located above. If you want to look at a specific object (say, PFJets) then exectue

```
 edmDumpEventContent --all --regex pfJet
/scratch/t/taimoor/LHC_SCHOOL_4/course_material/LHCSchool_Data_536.root
```

The output of `edmDumpEventContent` has information divided into four variable width columns. The first column is the C++ class `type of the data`, the second is `module label`, the third is `product instance label` and the fourth is `process name`. More information is available at https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookCMSSWFramework#ModularEvtContent

QUESTION 5.1 - How many modules produce products of type `vector` ? What are their names?

NOTE: Instead of the above, try without the option `--regex pfJet` . This will dump the entire event content - a file with many lines. For this reason we'll send the output to a file called `EdmDumpEventContent.txt` with a UNIX pipe command.

To do this execute

```
edmDumpEventContent
/scratch/t/taimoor/LHC_SCHOOL_4/course_material/LHCSchool_Data_536.root
> EdmDumpEventContent.txt
```

edmProvDump

To aid in understanding the full history of an analysis, the framework accumulates provenance for all data stored in the standard ROOT output files. Using the command

edmProvDump one can print out all the tracked parameters used to create the data file. For example, one can see which modules were run and the CMSSW version used to make the RECO(see link: https://twiki.cern.ch/twiki/bin/view/CMS/RECO) file. In executing the command below it is important to follow the instructions carefully, otherwise a large number of warning messages may appear. The ROOT warning messages can be ignored.

To do this execute

```
edmProvDump
/scratch/t/taimoor/LHC_SCHOOL_4/course_material/LHCSchool_Data_536.root
> EdmProvDump.txt
```

NOTE: `EdmProvDump.txt` is a very large file of the order of 10000-20000 lines. Open and look at this file and locate "`Processing History`" (about 20-40 lines from the top).

QUESTION 5.2 - Which version of CMSSW_?_?_? was used to produce the data?

edmEventSize

Finally we will execute `edmEventSize` to determine the size of different branches in the data file. Further details may be found here:
https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideEdmEventSize

`edmEventSize` isn't actually a 'Core' helper function (anyone can slap 'edm' on the front of a program in CMSSW). You can use edmFileUtil to get a PFN from an LFN (as shown above) so you could combine the call

Execute

```
edmEventSize -v
/scratch/t/taimoor/LHC_SCHOOL_4/course_material/LHCSchool_Data_536.root
> EdmEventSize.txt
```

Open and look at file `EdmEventSize.txt` and locate the line containing the text `recoPFJets_ak5PFJets__RECO.`. There are two numbers following this text that measure the plain and the compressed size of this branch.

QUESTION 5.3 - What are these two numbers?

Exercise 6 - Make a Physics Analysis ToolKit (PAT)Tuple (MC and Data sample)

Exercise 6 is on PAT, the Physics Analysis ToolKit. Analyzing physics data at CMS is a very complicated task involving multiple steps, sharing of expertise, cross checks and comparing different analysis. To maximize physics productivity, CMS has developed a collection of analysis tools called the PAT. PAT is a high-level analysis layer enabling the development of common analysis efforts across and within Physics Analysis Groups with easy access to the algorithms developed by Physics Objects Groups (POGs) in the framework of the CMSSW offline software. Its goal is to fulfill the needs of most CMS analyses, providing both ease-of-use for the beginner and flexibility for the advanced user. Its modular structure allows maximum flexibility by factorizing tasks in independent and highly configurable modules. About 80% of the physiscs analysis requires PAT to be used. Hence it is important to know about this tool. More information about PAT can be found in SWGuidePAT (see link: https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuidePAT)

In these two exercises we would make a PATtuple (see https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookPATTupleCreationExercise for more info) using a python configuration file.

We will plot the `Pt` distribution of the PAT muons. We will use files from the MC sample **RelValZMM** and the collision data skim **/SingleMu/Run2012A-recover-06Aug2012-v1/AOD**. The corresponding PATtuples will be called `simple_PAT_MC.root` and `simple_PAT_data.root`.

As a general reminder, you may be working on **lxs** and the data sample used in the exercise might become unavailable later and the exercise might not work. As a result, when trying to access file `/store/relval/CMSSW_5_3_6-START53_V14/RelValZMM/GEN-SIM-RECO/v2/00000/08C1D822-F629-E211-A6B1-003048679188.root` from **RelValZMM** MC sample **or** data file `/store/data/Run2012A/SingleMu/AOD/recover-06Aug2012-v1/0000/10816437-A0E0-E111-B70E-D8D3855BBDC4.root` from the collisions dataskim **/SingleMu/Run2012A-recover-06Aug2012-v1/AOD**, you might get an error indicating data unavailable. To avoid this scenario for the purpose of this exercise, we have made a small set of datafiles from these and copied them to **scratch** area called `/scratch/t/taimoor/LHC_SCHOOL_4/course_material` which anyone on **lxs** can access.

The MC file `/scratch/t/taimoor/LHC_SCHOOL_4/course_material /LHCSchool_RelValZMM536.root` consists of 500 events made from the dataset **`RelValZMM`** in `CMSSW_5_3_6` and the datafile

`/scratch/t/taimoor/LHC_SCHOOL_4/course_material /LHCSchool_Data_536.root` consists of 5000 events made from the collisions dataskim **/SingleMu/Run2012A-recover-06Aug2012-v1/AOD**.

In the directory `YOURWORKINGAREA/CMSSW_5_3_13/src` follow these steps:

1. Copy the following files

   simple_PAT_MC_cfg.py

   `simple_PAT_data_cfg.py`.

   The number of events have been set up to 1000 `process.maxEvents.input=1000` to run over 1000 events only, one can change it to -1 to run over all the events in the sample.

2. Now run the following command:

```
cmsRun simple_PAT_MC_cfg.py
```

When this command is executed it produces output that looks like in the file "simplePAT_MC_536.txt" present in your home directory.

3. Now open the root file just created called **`simple_PAT_MC.root`**.

```
root -l
```

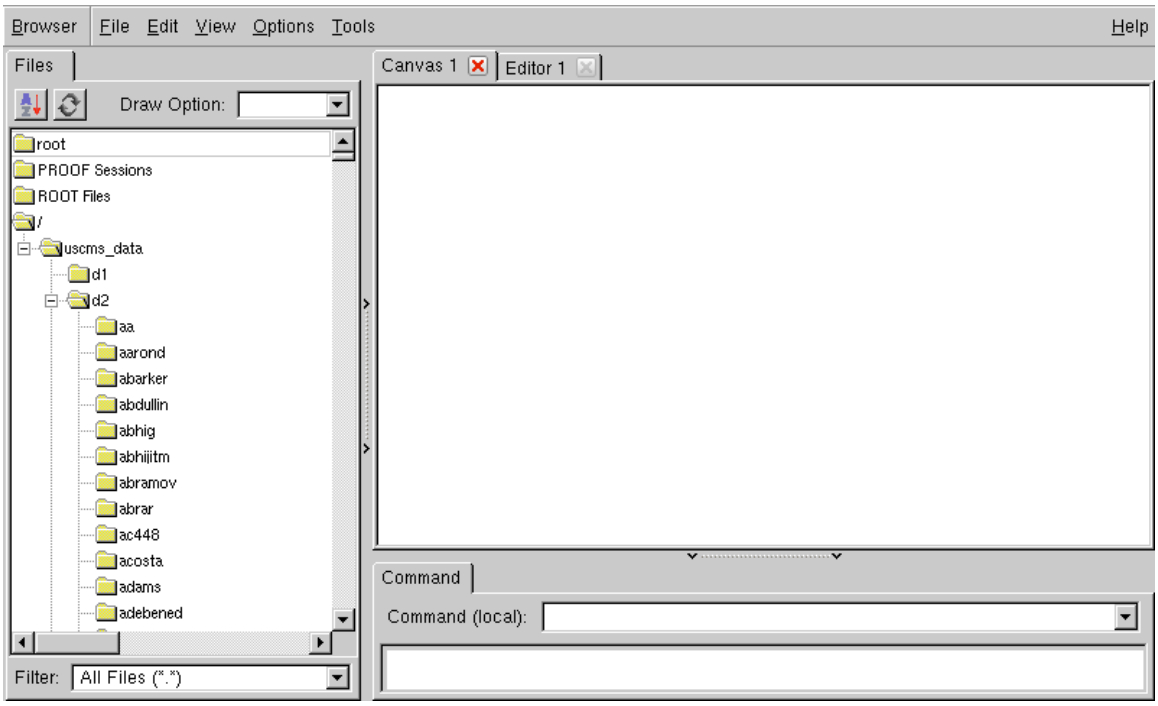
On the ROOT prompt type the following:

```
gSystem->Load("libFWCoreFWLite.so");
AutoLibraryLoader::enable();
gSystem->Load("libDataFormatsFWLite.so");
gROOT->SetStyle ("Plain");
gStyle->SetOptStat(111111);
TBrowser b;
TFile::Open("simple_PAT_MC.root");
```

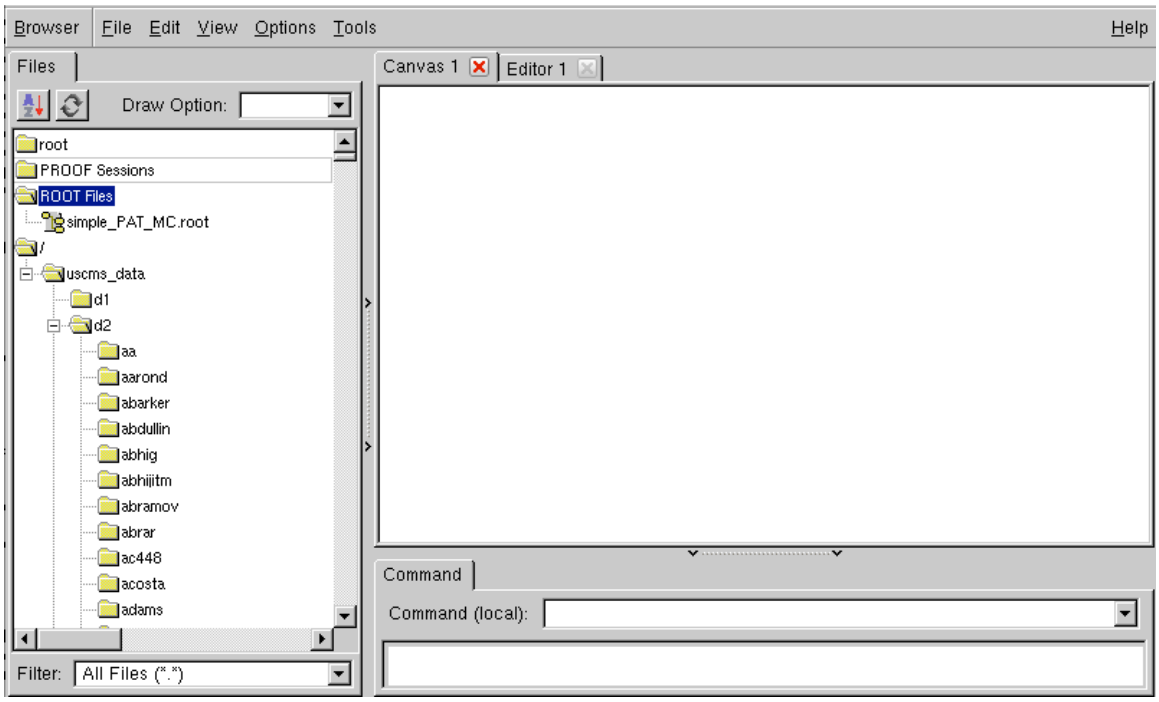
To be able to use the member functions of a CMSSW data class from within ROOT requires ROOT to have a 'dictionary' for that class. To obtain that dictionary, it is necessary to load the proper library into ROOT. The first three lines of the code above do exactly that. More information is at WorkBookFWLiteExamples (see link: https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookFWLiteExamples). Note that gROOT->SetStyle ("Plain"); sets a plain white background for all the plots in ROOT.

NOTE: If the rootlogon.C is created in the home area, and the above four lines of code are in that file, the dictionary will be obtained, and all the plots will have a white background automatically upon logging in to ROOT.

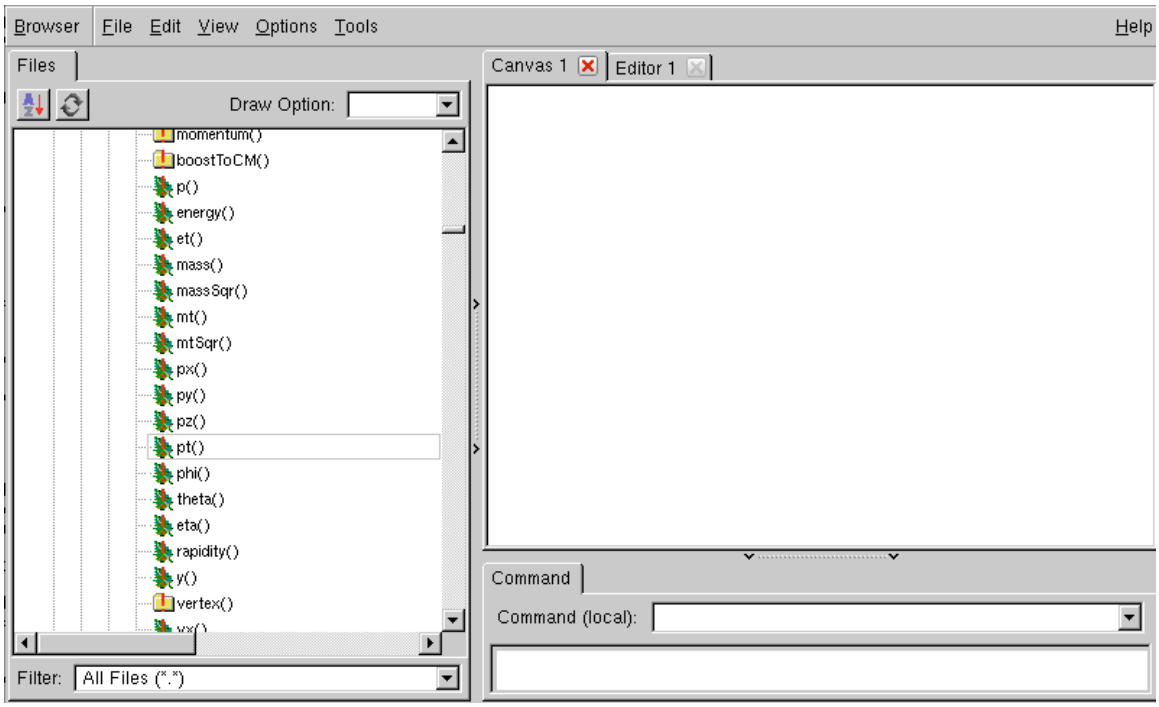
Now a ROOT browser window opens and looks like this:



In this window click on ROOT Files on the left menu and now the window looks like this:



Click simple_PAT_MC.root, then Events, then patMuons_cleaPatMuons__PAT and then patMuons_cleanPatMuons__PAT.obj. A window appears that looks like this:



Scroll a long way down the file (not too fast) and click on pt(). A PAT Muon Pt distribution will appear. These muons have been produced in the Z to mumu interactions as the name of the data sample implies.

QUESTION 6.1 - What is the mean value of the muon pt() for the MC data?

Note: To exit ROOT simply type .q in the command line.

4. Similarly run the following command:

```
cmsRun simple_PAT_data_cfg.py
```

Open the output file created simple_PAT_data.root and following the same steps as above, answer the following question:

QUESTION 6.2 - What is the mean value of the muon pt() for the collision data?

Create your GIT account:

Go to the following link and create your account:
https://github.com/

test user: `lhcncp74`
`password: lhcncp74_sk`

Then on lxs:

```
git  config  -- global  user.github  "lhcncp74"

git  config  -- global  user.email "user_name@mail.com"
```