Single Calculus Chain Tutorial for LALINET

Release 1.0

Gregori Moreira, Fábio Lopes, Diego Gouveia

July, 2017

CONTENTS

CHAPTER

ONE

INTRODUCTION

The **A**ssessment of atmospheric optical **P**roperties during biomass burning **E**vents and **L**ong-range transport of desert dust (APEL) project is an initiative to transfer the knowledge from the EARLINET (The European Aerosol Research Lidar Network) to LALINET (The Latin America Lidar Network). The project has as general objective to assess of the current observation capabilities in South America, by comparison with EARLINET standards and procedures in order to harmonize the standard operation procedures between EARLINET and LALINET. This is the first step towards an operational global lidar network, enabling its contribution to GALION and its support to satellite atmospheric missions, with the required quality.

The APEL project has as project coordinators:

- Dr. Doina Nicolae - head of the National Institute for Research and Development for Optoelectronics (INOE - Bucharest)
- Dr. Eduardo Landulfo that is Head of the Laser Environmental Applications Laboratory (LEAL) at Nuclear and Energy Research Institute (IPEN/CNEN - Brazil)

The Technical officer of APEL is Dr. Anne Grete Straume-Lindner.

Quantitative assessment of the optical properties of aerosol mixtures is currently limited by the observation uncertainties, coming from the instrument and from the retrieval algorithms. A significant expertise was gained in the last 15 years in EARLINET regarding the optimization of the instruments and algorithms, data quality control and retrieval of advanced data products (such as aerosol typing). In the framework of APEL Project, these tools will be implemented to LALINET in order to make the final data products from the two networks comparable, and allow studying the similarities and differences in terms of aerosol loads, transport heights, types and properties.

The APEL project has two specific objectives,

1. The adaptation and implementation of the EARLINET QA/QC program at LALINET stations, in order to achieve homogeneous data products from both networks.
2. Observation of aerosol characteristics at selected LALINET stations, and comparison with EARLINET's climatology

To complete these tasks some actions are needed:

a) Evaluation of the current observation capabilities in South America Lidar community

b) Optimizing the instruments to reach EARLINET's similar operating standards

c) Assessing the aerosol vertical structures, aerosol optical properties and aerosol types at selected LALINET and EARLINET stations

d) Comparison with EARLINET climatology

The QA/QC program developed by EARLINET will be implemented at selected LALINET stations, and the quality of the data products will be assessed. The evaluation will be done at hardware and software level. At hardware level, the quality of the signals will be checked using the specific EARLINET tests described at

http://www.actris.net/Portals/97/deliverables/PU/WP2_D2.2_M12.pdf

At software level, the LALINET data processing algorithms will be compared with the EARLINET Single Calculus Chain [1–4]; the last is a fully automatic evaluation process that can be used for virtually any lidar configuration, and was validated for several EARLINET lidar stations. The SCC is a data processing chain that allows all EARLINET stations to retrieve, in a fully automatic way, the aerosol backscatter and extinction profiles starting from the raw lidar data of the lidar systems they operate.

In this context, the Single Calculus Chain (SCC) basic tutorial for LALINET is just a counterpart of several scientific activities that will take part of the APEL project initiative to transfer the knowledge from the EARLINET (The European Aerosol Research Lidar Network) to LALINET (The Latin America Lidar Network).

According to D'Amico 2015 [2],

> *The Single Calculus Chain (SCC) is an official EARLINET tool. It has been developed to accomplish the fundamental need of any coordinated lidar network to have an optimized and automatic tool providing high-quality aerosol properties. Currently, it has been used by 20 different EARLINET stations which have submitted about 2600 raw data files covering a very large time period. Moreover, more than 5000 SCC optical products (about 3600 aerosol backscatter profiles and 1400 aerosol extinction profiles) have been calculated and used for different purposes like analysis of instrument intercomparisons, air-quality model assimilation experiment, and ongoing long-term comparisons with manually retrieved products. The large usage and the long-term plan for the centralized processing system make the SCC the standard tool for the automatic analysis of EARLINET lidar data.*

The main purpose of this tutorial is to guide the LALINET lidar user to install the necessary Python libraries to convert binary Licel lidar data to NetCDF format, to be use as input to Single Calculus Chain program. Additionally, the tutorial will help the user to upload the NetCDF files to SCC platform. More information about the SCC program can be find in the reference section 6.4.

CHAPTER

TWO

INSTALLING PYTHON

In order to use python for scientific computing and also to convert the Licel raw lidar binary data to NetCDF format is to install the Python program and some libraries you will need to install:

1. The python language
2. A set of modules that allow you to manipulate data, plot, etc.

2.1 Basic Python

If you are using Linux or Mac OS maybe you have python already installed. If you are using Windows you can download python from the following links:

1. Install Python 2.7 in your PC:
    - Windows (https://www.python.org/ftp/python/2.7.13/python-2.7.13.msi);
    - Others operational system (https://www.python.org/download/other/)
2. Install the pip package (This step is only for Windows)
    - https://caiojcarvalho.wordpress.com/2016/03/22/instalando-e-configurando-o-python-3-4-e-o-pip-no-windows/
    - https://pip.pypa.io/en/stable/installing/#do-i-need-to-install-pip
3. Install a Python IDE (Integrated Development Environment)
    - https://www.jetbrains.com/pycharm/ (PyCharm is considered very user friendly).

From there download and run "Python 2.7.13 Windows Installer".

Python has now two active versions, 2.7.13 and 3.6.2. While the 3.* versions are the future of python, right now most modules support the 2.* version, so we will use that applying to convert data to use as input in the SCC program.

2.2 Scientific modules

There are various useful modules for scientific programming with python. Among other:

- Numpy - Allows fast N-dimensional array manipulation (http://numpy.scipy.org/)
- Scipy - A collection open-source software for mathematics, science, and engineering (http://www.scipy.org/)
- Matplotlib - A library for plotting data (http://matplotlib.sourceforge.net/)

While installing all these is relatively easy, it's even easier to install a collection of these and many other useful modules for scientific computing.

A very good collection is Anaconda. Anaconda can be installed in Windows, Linux, and MacOS X, and includes:

- The above mentioned modules
- Spyder - "a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features"
- ipython - "a powerful Python shells" suitable for interactive scientific computing.
- As mentioned before, PyCharm is considered very user friendly integrated development environment.

CHAPTER

THREE

INSTALLING THE CONVERTER (BINARIES FILES TO NETCDF)

After install Python program, the PIP package and the IDE it is necessary install the *atmospheric-lidar* package used to convert the Licel raw binary files to NetCDF file.

1. Open the following folder in your Python IDE: *Scripts/atmospheric-lidar*, as can be seen in figure 3.1.

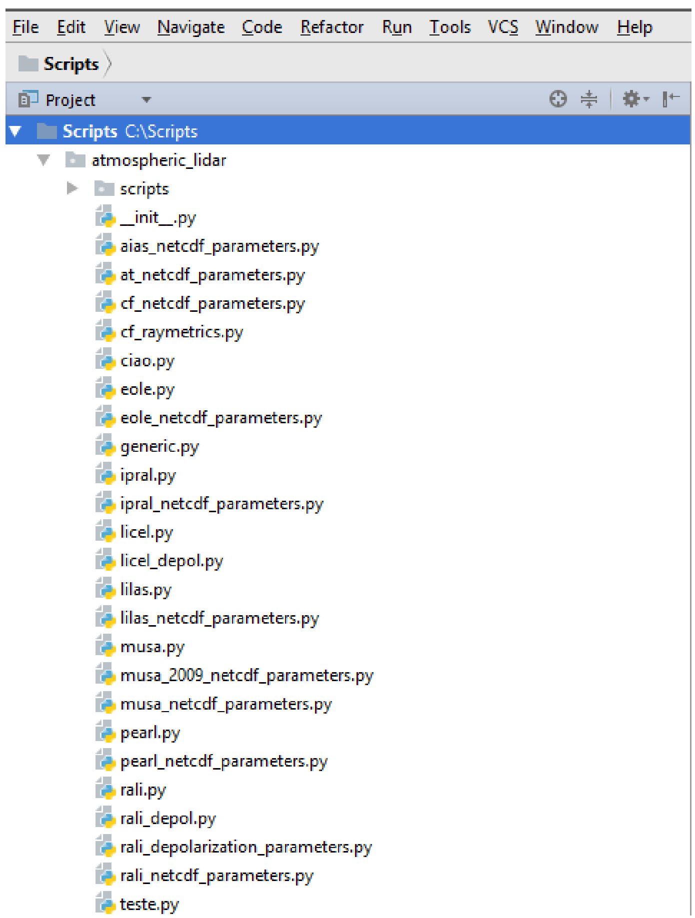

Figure 3.1: Folder of the *atmospheric-lidar* package.

2. Open the Terminal by clicking at the terminal button in the bottom part at the IDE, as shown in the red circle in figure 3.2.

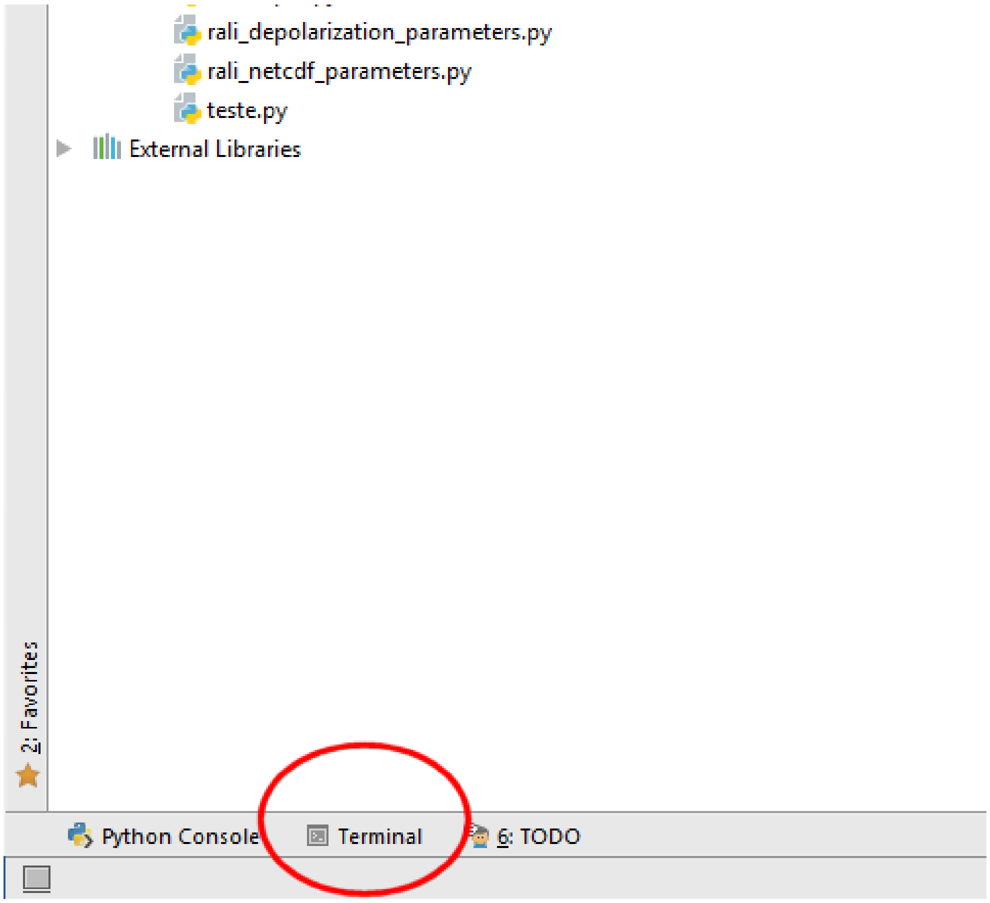

Figure 3.2: Opening the terminal at PyCharm IDE.

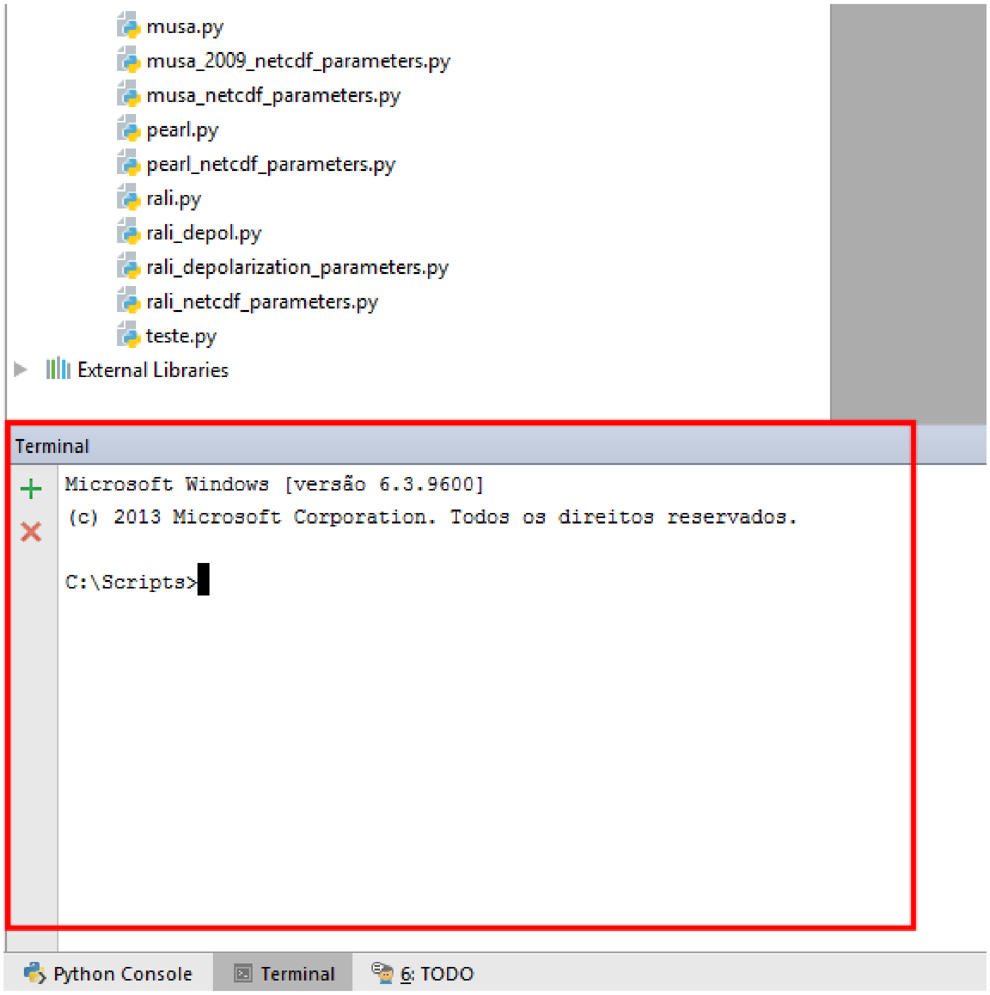

Figure 3.3: Terminal opened at PyCharm IDE.

3. In the PyCharm terminal type the command line: ***pip install atmospheric-lidar*** and press <ENTER>

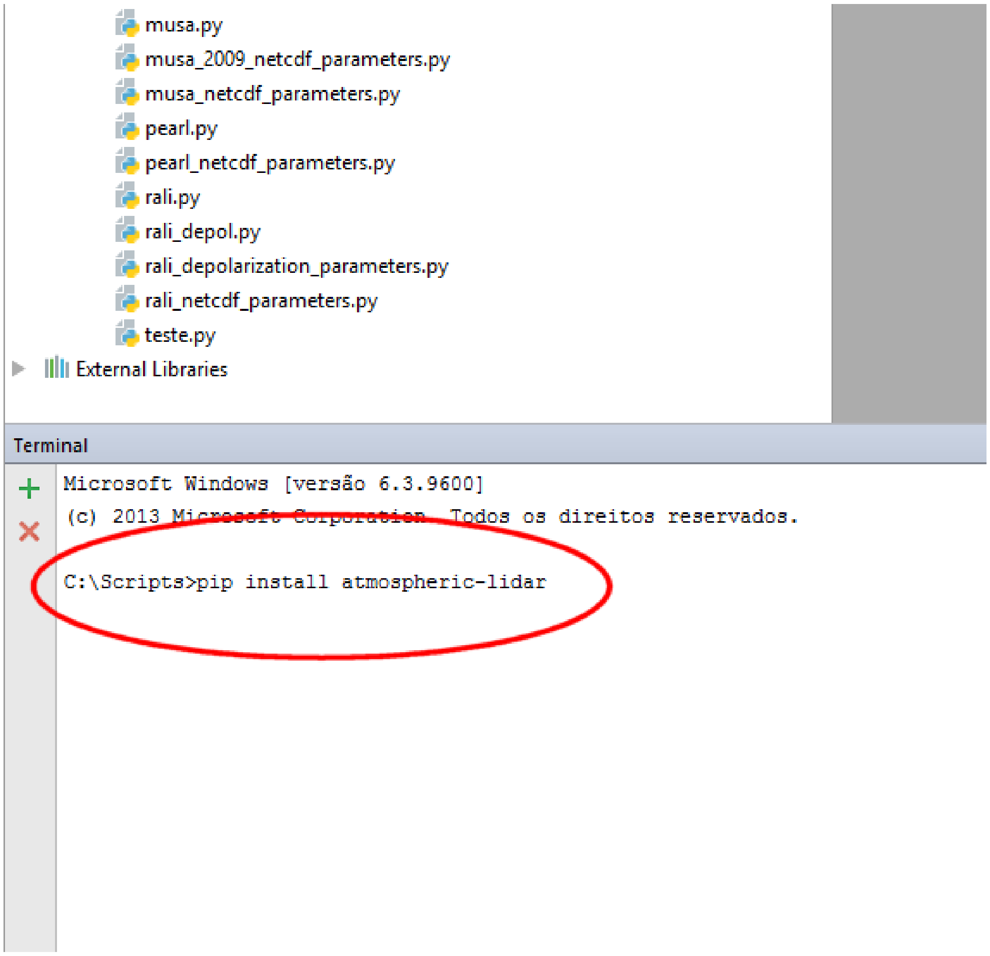

Figure 3.4: Installing the SCC *Atmospheric-lidar* package.

4. **Remember, the *atmospheric-lidar* package just run on the Python version 2.7**
5. **More details about the *atmospheric-lidar* package can be found in the following link:**
6. **https://pypi.python.org/pypi/atmospheric_lidar/**
7. If all packages were correctly installed, all packages from *atmospheric-lidar* were installed and the following message will be shown, according figure 3.5.

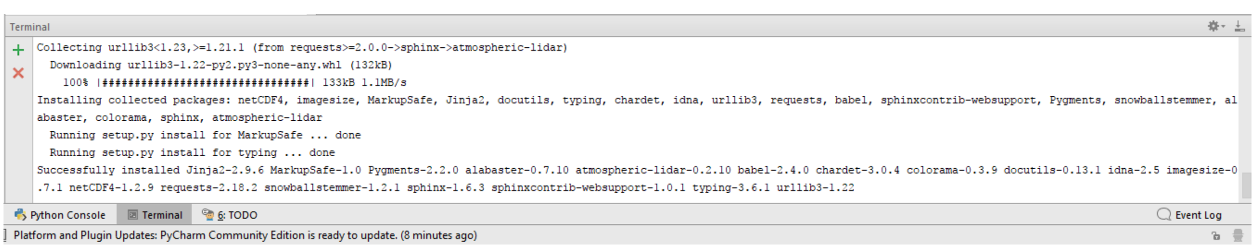

Figure 3.5: Terminal of PyCharm IDE after the *Atmospheric-lidar* package installation.

CHAPTER

FOUR

CREATING AN INPUT FILE

All information presented in this chapter are available in details in the following links:

https://scc-documentation.readthedocs.io/en/latest/file_formats/netcdf_file.html

https://scc-documentation.readthedocs.io/en/latest/_downloads/NetCDF_input_file_v3.pdf

The SCC is composed by three different modules:

- Pre-processing module (ELPP)
- Optical processing module (ELDA)
- Depolarization calibrator module (ELDEC)

To perfom aerosol optical retrievals the SCC needs not only the raw lidar data but also a certain number of parameters to use in both pre-processing and optical processing stages. The SCC gets these parameters looking at two different locations:

- Single Calculus Chain relational database (SCC_DB)
- Input files

The input files have to be submitted to the SCC in NetCDF format. At present the SCC can handle four different types of input files:

- Raw Lidar Data
- Sounding Data
- Overlap
- Lidar Ratio

As already mentioned, the Raw lidar data file contains not only the raw lidar data but also other parameters to use to perform the pre-processing and optical processing. The Sounding Data file contains the data coming from a correlative radiosounding and it is used by the SCC for molecular density calculation. The Overlap file contains the measured overlap function. The Lidar Ratio file contains a lidar ratio profile to use in elastic backscatter retrievals. The Raw Lidar Data file is of course mandatory and the Sounding Data, Overlap and Lidar Ratio files are optional. If Sounding Data file is not submitted by the user, the molecular density will be calculated by the SCC using the ”US Standard Atmosphere 1976´´. If the Overlap file is not submitted by the user, the SCC will get the full overlap height from SCC_DB and it will produce optical results starting from this height. If Lidar Ratio file is not submitted by the user, the SCC will consider a fixed value for lidar ratio got from SCC_DB.

The user can decide to submit all these files or any number of them (of course the file Raw Lidar Data is mandatory). For example the user can submit together with the Raw Lidar Data file only the Sounding Data file or only the Overlap file.

The NETCDF files used as input and read by SCC have certain parameters which are created and modified from a Python routine. Inside *atmospheric-lidar* folder, there are configuration examples ofar several Raymetrics lidar systems belonging to EARLINET. The file *msp1_netcdf_parameters.py* is an example for the São Paulo LALINET lidar station, named as MSPI. There are other files as example, such as, *ipral_netcdf_parameters.py*, *lilas_netcdf_parameters.py*, *musa_netcdf_parameters.py*, and others. If you need to create a file to your own system, the structure below should be used:

```
general_parameters = \
{'System': '\'MSPI\'',        //Lidar system name
 'Laser_Pointing_Angle': 0, // scan angles used in the
    measurement
 'Molecular_Calc': 0,  // molecular density profile option
 'Latitude_degrees_north': -46.7, // Lat coordinate of the
    station
 'Longitude_degrees_east': -23.6, // Long. coordinate of the
    station
 'Altitude_meter_asl': 760.0, // altitude a.s.l. of the station
    (meters)
 'Call sign': "sp"}        //Station name

channel_parameters = \
{'00532.o_an': {'channel_ID': 722, //channel id (provided by SCC)
       'Background_Low': 22500, //Background initial altitude
          (meters)
       'Background_High': 27000, //Background final altitude
          (meters)
       'Laser_Shots': 1200, //# of laser shots of each data
       'LR_Input':1,        //Use the value 1
       'DAQ_Range':500,     // mV range of each channel
       'First_Signal_Rangebin': 1}, // Zero Bin correction

'00532.o_ph': {'channel_ID': 716,
      'Background_Low': 25000,
      'Background_High': 27000,
      'Laser_Shots': 1200,
      'LR_Input':1,
```

```
        'DAQ_Range':0,
        'First_Signal_Rangebin': -1},
```

The mandatory variable of molecular calculation defines the way used by SCC to calculate the molecular density profile. At the moment two options are available:

- US Standard Atmosphere 1976. In this case the value of this variable has to be 0
- Radiosounding profile. In this case the value of this variable has to be 1

For the Lidar ratio variable, the array is required only for lidar channels for which elastic backscatter retrieval has to be performed. It defines the lidar ratio to be used within this retrieval. For this case, two options are available:

- The user can submit a lidar ratio profile. In this case the value of this variable has to be 0.
- A fixed value of lidar ratio can be used. In this case the value of this variable has to be 1.

CHAPTER

FIVE

CONVERTING FILES

1. Open your Python IDE
2. Open the file ***scripts/atmosperic-lidar/conversor2netcdf.py***

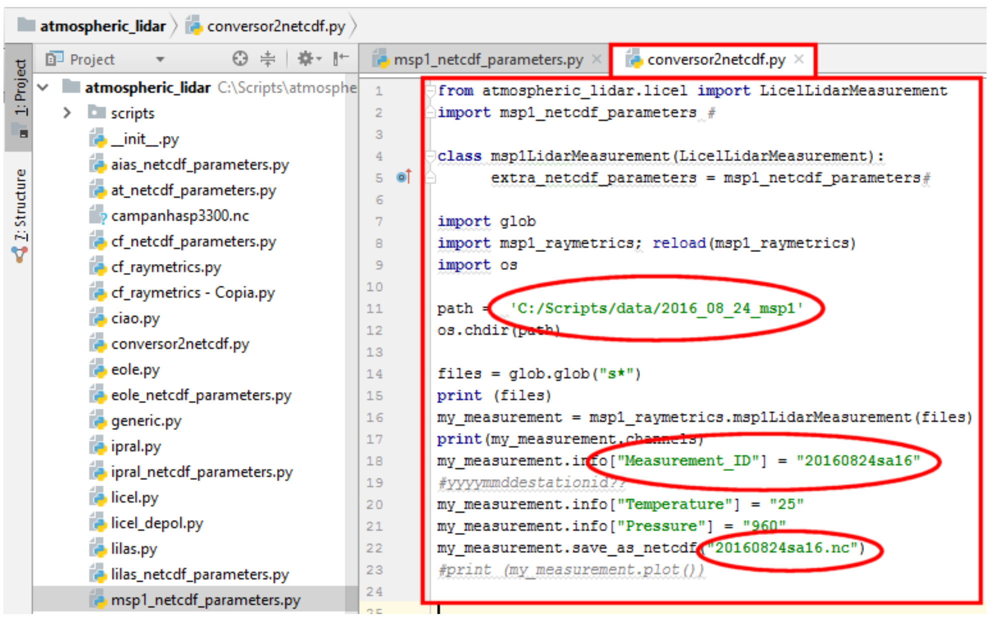

Figure 5.1: File *conversor2netcdf.py* that can be found inside *Atmospheric-lidar* package.

This *conversor2netcdf.py* script generates a NETCDF file from Raymetrics lidar binaries raw data.

3. The main parameters of this script are described below:

```
path = 'C:/Scripts/data/2016_08_24_msp1/'
//This is the path of raw lidar data. In python
   language folders are separated by '/'

files = glob.glob('s*')
//Put the first letter of name of yours binaries files
```

```
my_measurement.info['Measurement_ID'] = '20160824sa16'
//Put the ID of your measurement with the format
   yyyymmddnnhh, where nn is the code of your station. FOr
   S\~{a}o Paulo station the code name is
   \textbf{\textit{sa}}. And hh is associated to the time
   serie of measurements.
Note that SCC does not accept duplicate IDs.

my_measurement.save_as_netcdf('20160824sa16.nc')
//Put the name of NETCDF file that will be generated by
   Python routine.
Note that SCC does not accept files with duplicate names.
```

4. After changing all necessary parameters, you need to run the file *conversor2netcdf.py*.

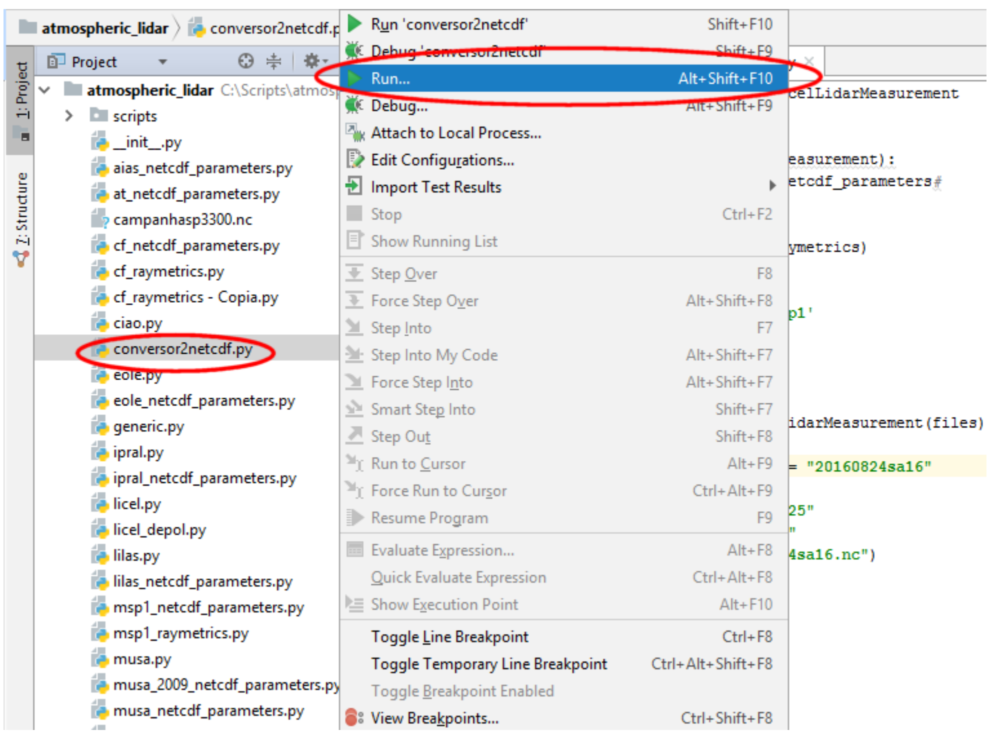

Figure 5.2: Running *conversor2netcdf.py* in PyCharm IDE.

5. If does not occur any problem, the following message will be show:

   'Process finished with exit code 0'

   As can be seen in figure 5.3. The NETCDF file will be created in the same folder of yours binaries data.

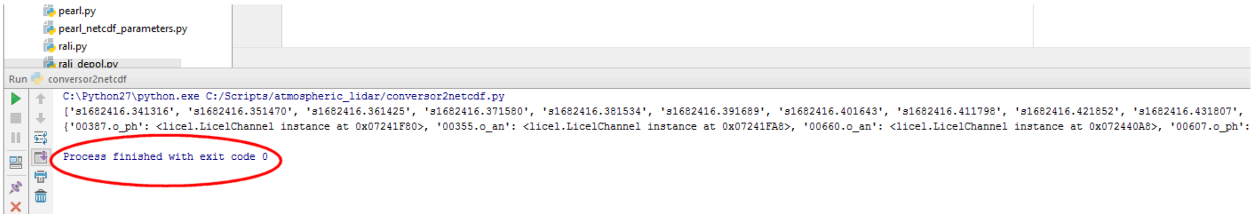

Figure 5.3: Running *conversor2netcdf.py* and creating a NetCDFile in a correct way.

5.1 Checking NetCDF files

All the NetCDF files created using the raw lidar data can be checked if they have the correct format to be used with the EARLINET's Single Calculus Chain.

It is necessary to have the netCDF4 python module installed.

https://pypi.python.org/pypi/netCDF4/

And will be needed the *netcdf_checker.py* script developed by Ioannis Binietoglou provided in the following link:

https://bitbucket.org/iannis_b/scc-netcdf-checker

CHAPTER

SIX

UPLOADING THE NETCDF DATA TO SCC PLATFORM

6.1 Enter in SCC system

1. In order to upload the NetCDF data from the lidar system it is necessary to access the SCC web platform in the following link:
2. https://scc.imaa.cnr.it/
3. To access the system use the general username and password:
   - **username:** *scc_user*
   - **password:** *sccforever!*

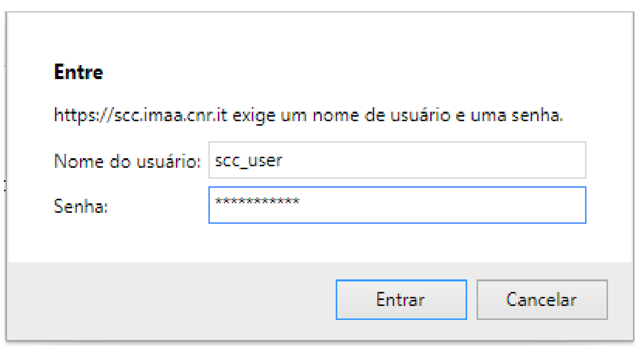

Figure 6.1: SCC login screen.

4. After login, the screen presented in figure 6.2 will be shown,
5. Click on login, in the upper right side of the screen, as can be seen in the red square in figure 6.2.

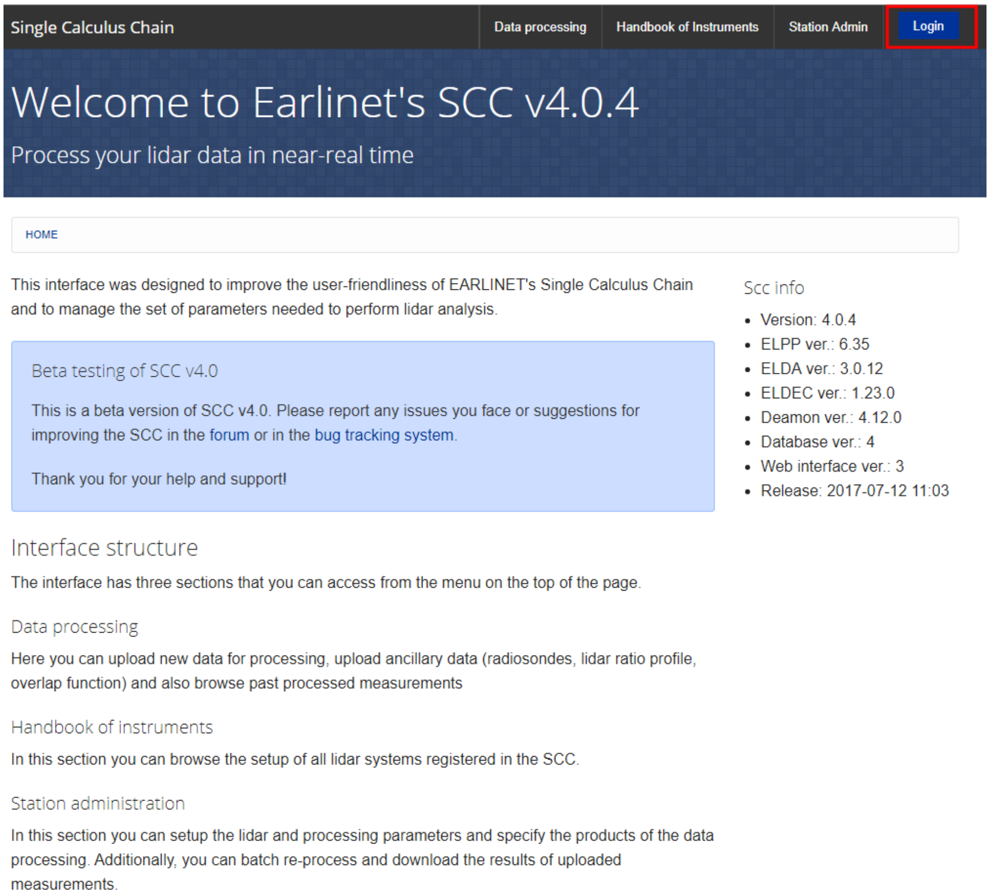

Figure 6.2: SCC home website.

6. In order to request a account to your own LALINET lidar station and obtain a user account, please contact the following researchers

   - Dr. Livio Belegante - livio@inoe.ro
   - Dr. Ioannis Binietoglou - ioannis@inoe.ro

7. To access your account use your username and password provided by the SCC administrators, as can be seen in figure 6.3, and click at submit button.

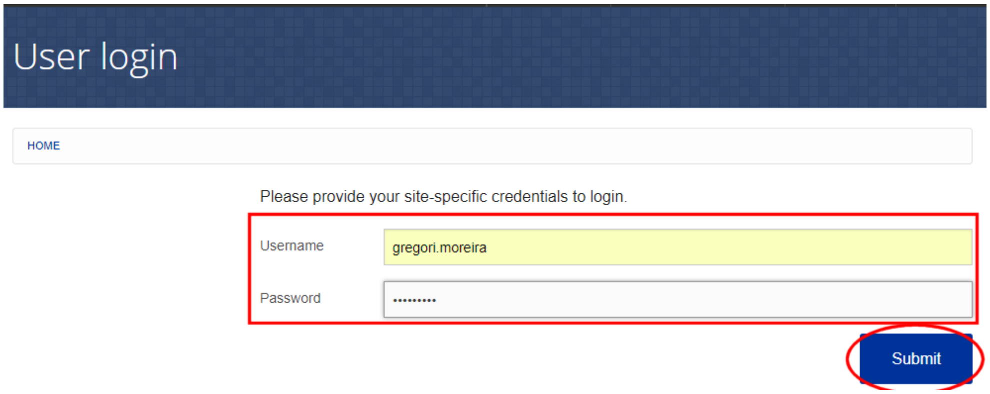

Figure 6.3: SCC login screen.

6.2 Uploading NetCDF data

1. On the next screen click at ***Data processing*** link, the first link in the upper menu, as presented in figure 6.4

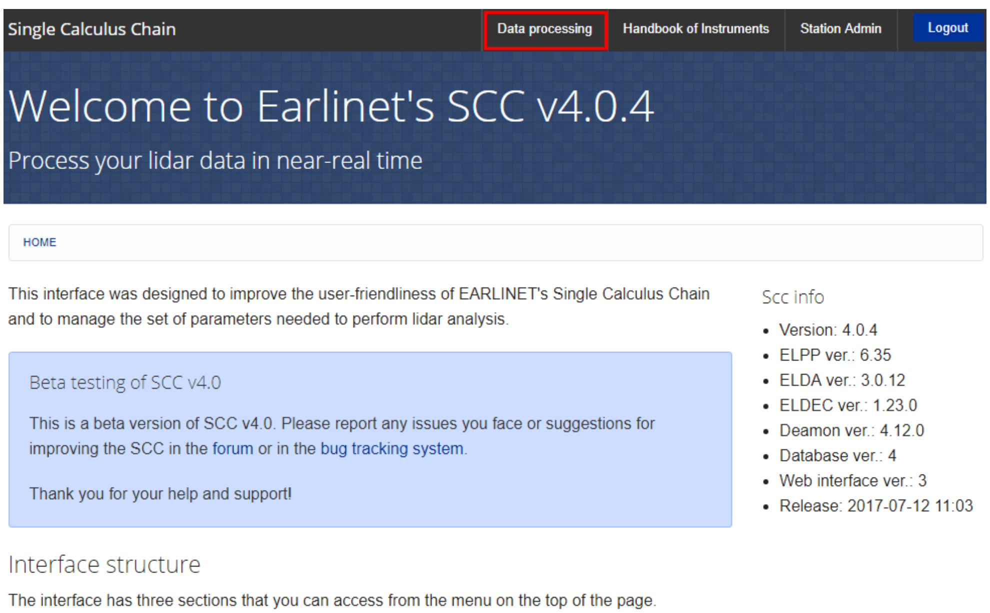

Figure 6.4: Enter at SCC data processing page.

2. It will be shown the *Data processing* screen, as can be seen in figure 6.5.
3. Click in the ***Quick Upload***, inside de Actions menu, in the left side of the screen (See figure 6.5).

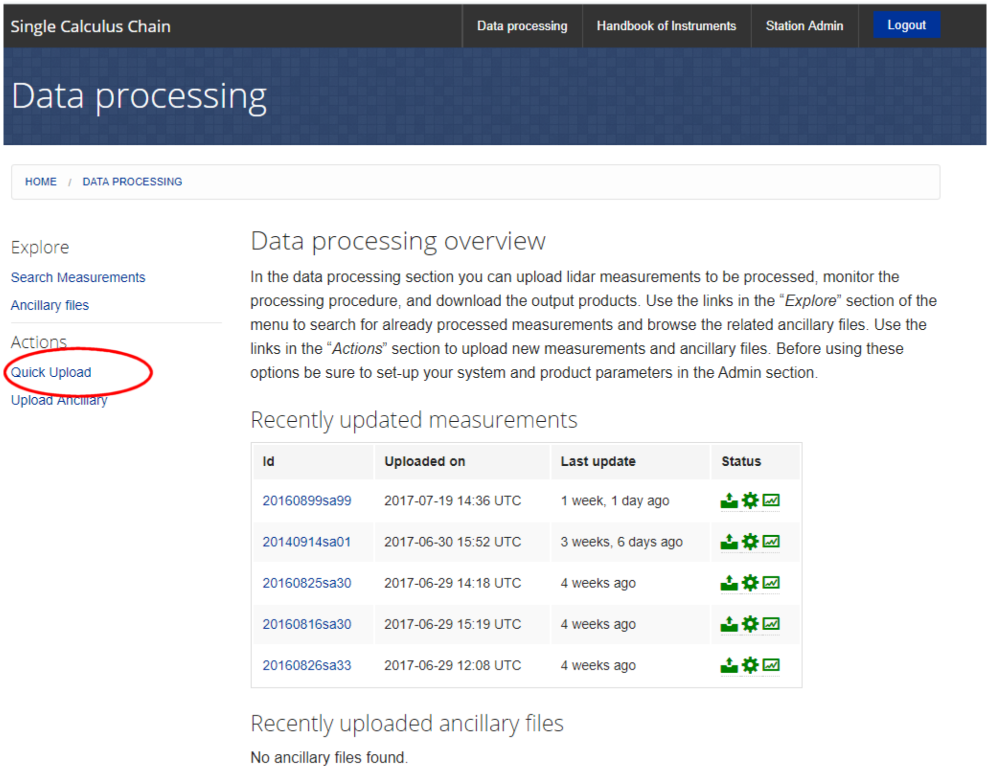

Figure 6.5: Accessing the ***Quick Upload*** inside the Actions menu.

4. Select your lidar system station in the option ***System***, see figure 6.6

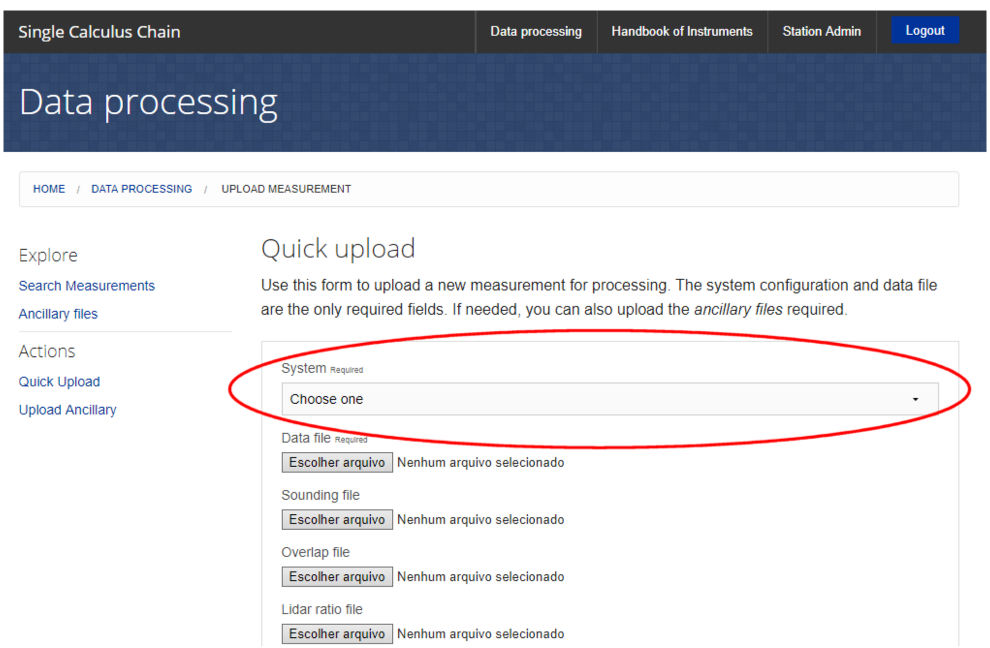

Figure 6.6: Accessing the ***Quick Upload*** page and selecting the lidar system.

5. For example, the code for São Paulo LALINET lidar station is 248, as can be seen in figure 6.7. This code is provided by the SCC administrators.

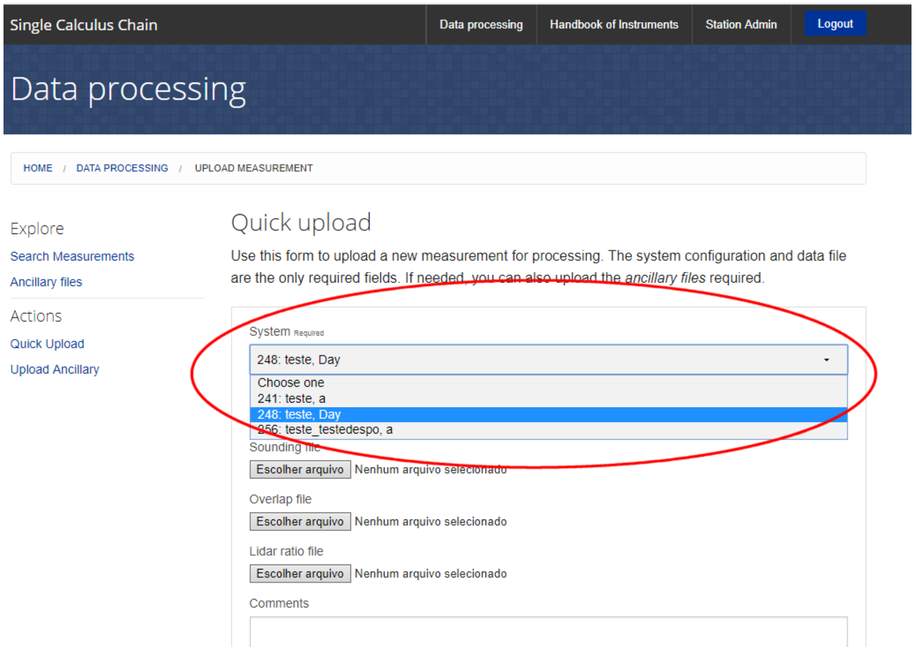

Figure 6.7: Selecting the lidar system.

6. After select your lidar system, click on the bottom ***Select file***, inside *Data File* option. Check figure 6.8.

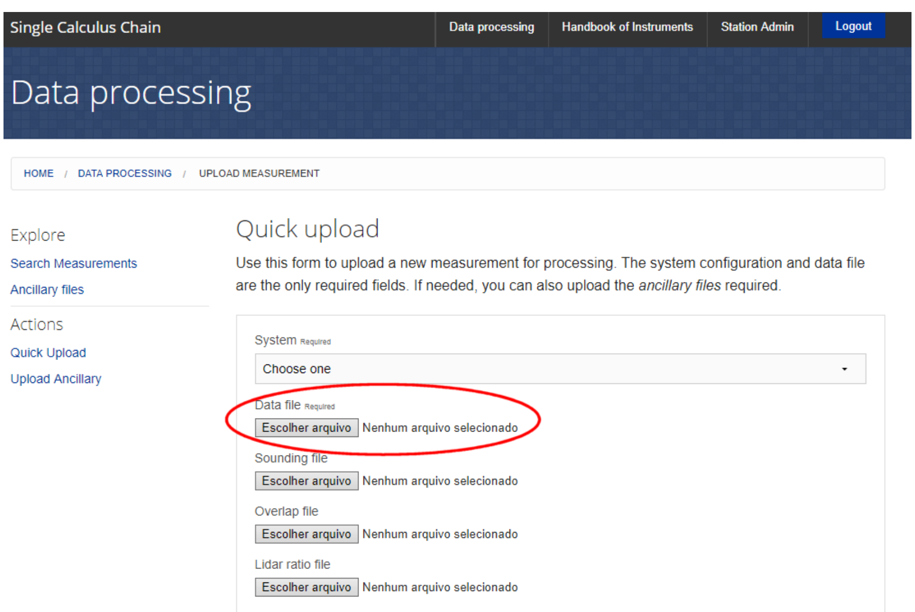

Figure 6.8: Selecting the *data file* option.

7. It will be opened a dialog box. In this box, select the NetCDF file and click on ***Open***

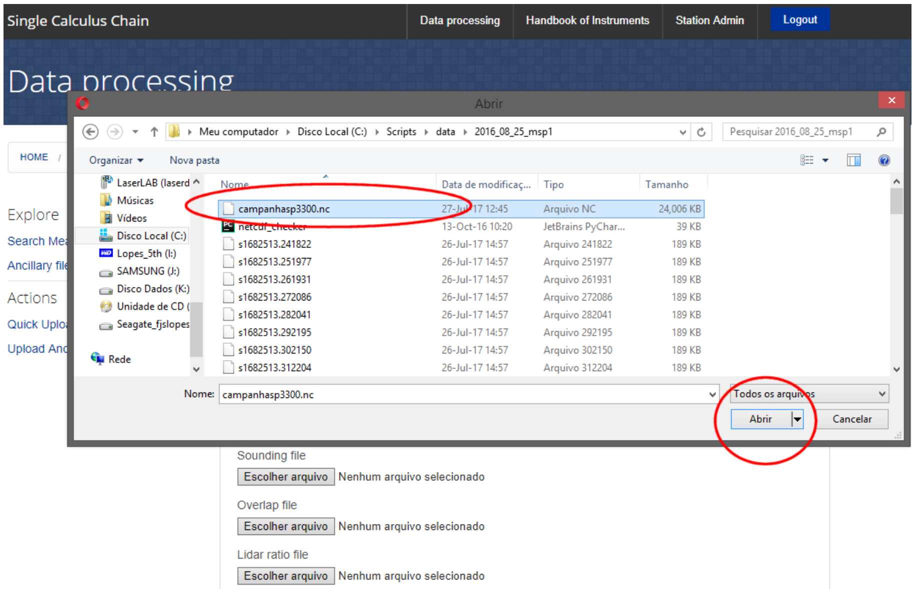

Figure 6.9: Selecting the NetCDF file to SCC platform.

8. It is also possible to upload data for sounding, overlap and lidar ratio information

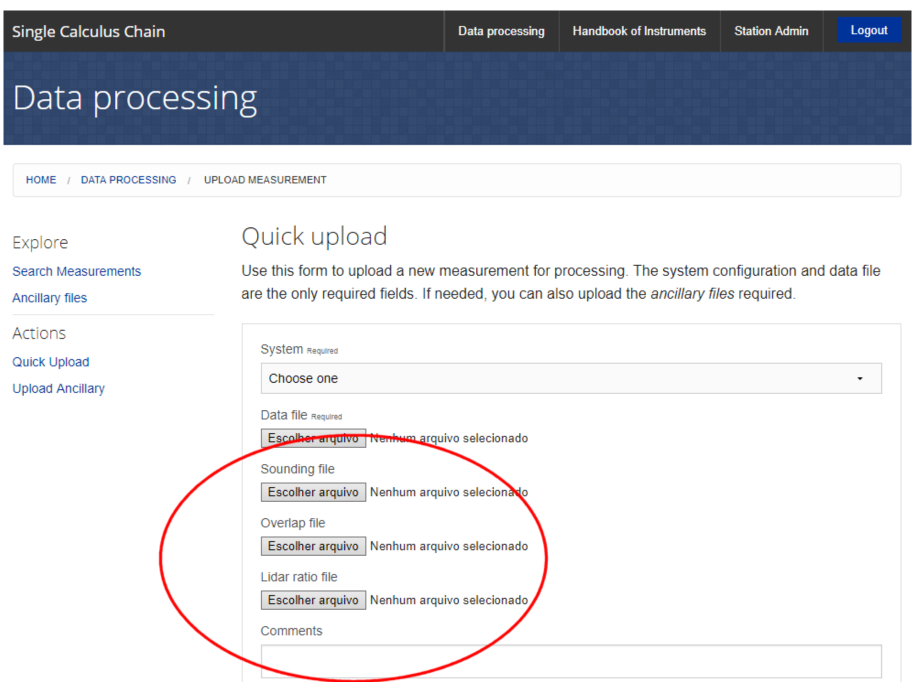

Figure 6.10: Options to upload others NetCDF file to SCC platform.

9. It is optional to classify the measurements

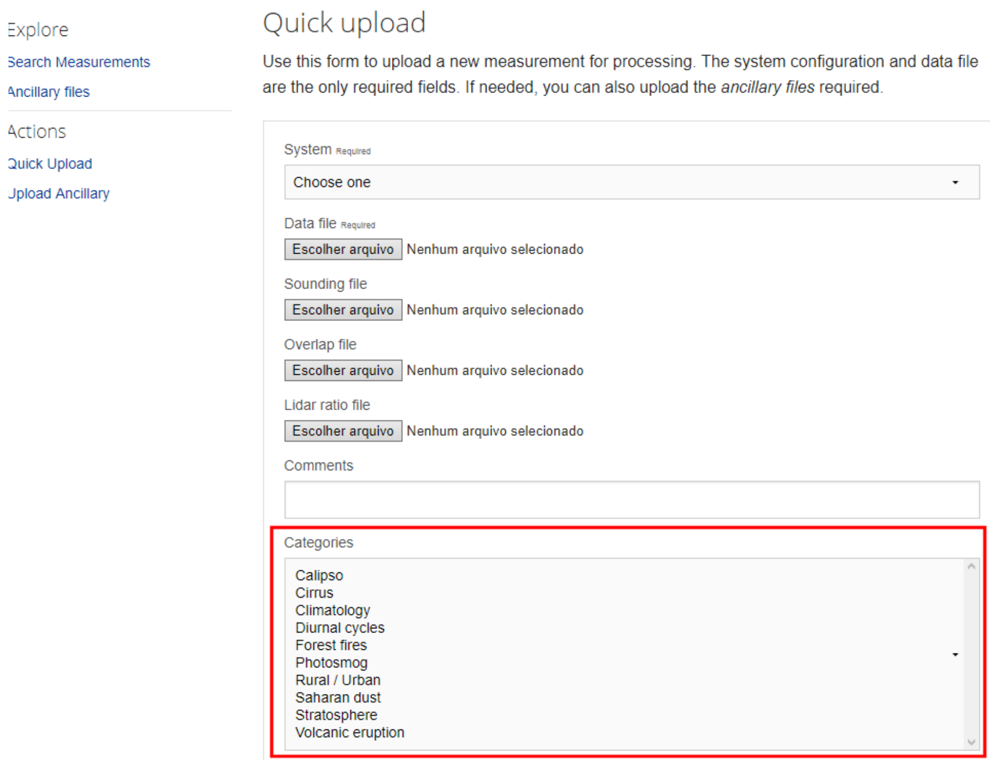

Figure 6.11: Options to to classify the measurements.

10. Click on *submit measurement* button on the right-bottom part of the screen, and wait for the total upload. Check figure 6.12.

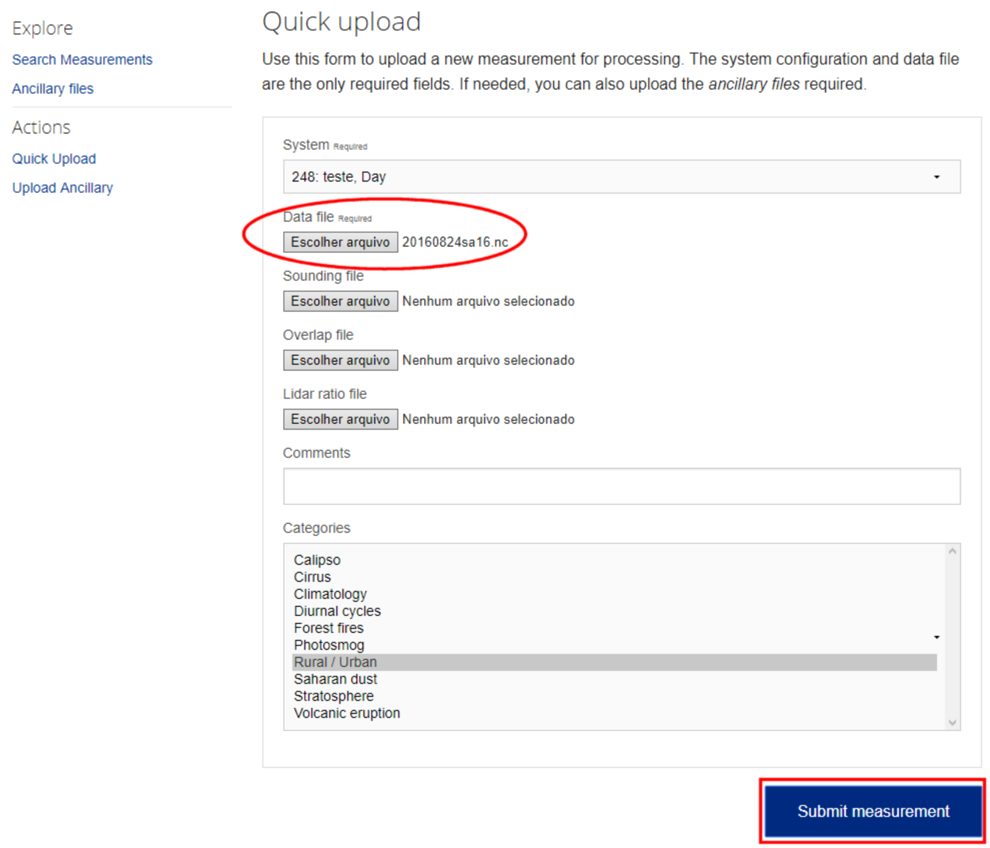

Figure 6.12: Uploading data measurements as NetCDF file to SCC platform.

6.3 Processing NetCDF data

1. If the upload was done correctly, the first icon will turn green and SCC will begin data pre-processing, as shown in figure 6.13.
2. If python input file (See Chapter 4) is configured in agreement to the SCC specifications, pre-processing will be done correctly and the gear icon will turn green, as can be seen in figure 6.14.
3. If process occurs without errors, the small graph icon will turn green as well, as can be seen in figure 6.14.

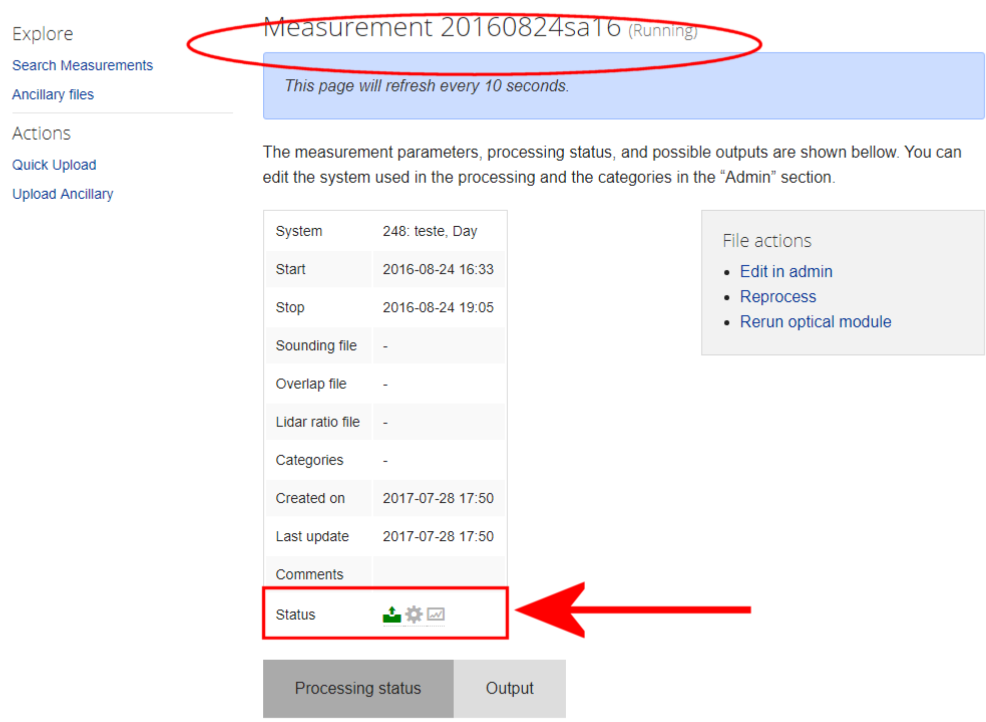

Figure 6.13: NetCDF file uploaded correctly and beginning of pre-processing.

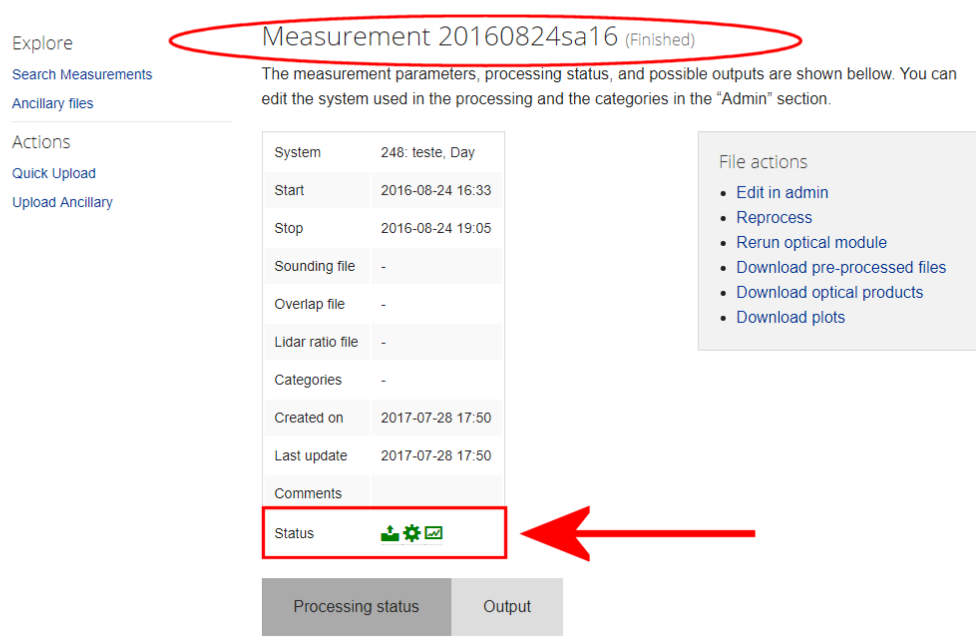

Figure 6.14: Pre-processing and process were done correctly.

4. After all the upload and processing of data are completed, if it is necessary, the SCC products can be downloaded from File actions menu. Check figure 6.15.

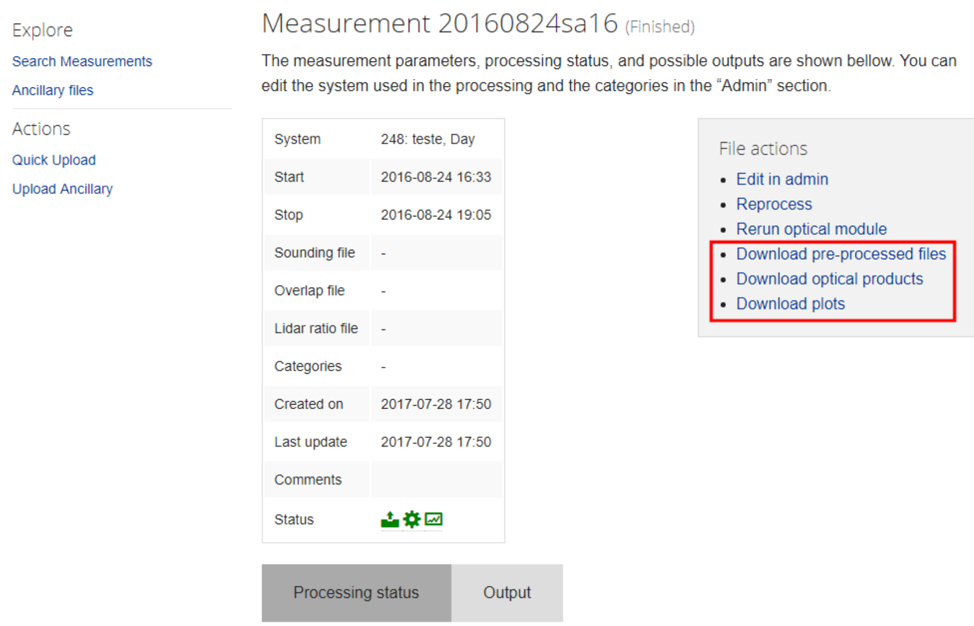

Figure 6.15: Downloading SCC products.

6.4 Python Modules

- Python for Windows https://www.python.org/ftp/python/2.7.13/python-2.7.13.msi

  Python for others operational system https://www.python.org/download/other/

  Pip package: https://pip.pypa.io/en/stable/installing/#do-i-need-to-install-pip

  Numpy: http://numpy.scipy.org/

- Scipy: http://www.scipy.org/
- Matplotlib: http://matplotlib.sourceforge.net/index.html

  Video tutorial for install python and packages: https://www.youtube.com/watch?v=-llHYUMH9Dg

BIBLIOGRAPHY

[1] Welcome to Single Calculus Chain interface's documentation single calculus chain website. http://scc-documentation.readthedocs.io/en/latest/. Accessed: 2017-07-28.

[2] G. D'Amico, A. Amodeo, H. Baars, I. Binietoglou, V. Freudenthaler, I. Mattis, U. Wandinger, and G. Pappalardo. Earlinet single calculus chain - overview on methodology and strategy. *Atmospheric Measurement Techniques*, 8(11):4891–4916, 2015.

[3] G. D'Amico, A. Amodeo, I. Mattis, V. Freudenthaler, and G. Pappalardo. Earlinet single calculus chain – technical – part 1: Pre-processing of raw lidar data. *Atmospheric Measurement Techniques*, 9(2):491–507, 2016.

[4] I. Mattis, G. D'Amico, H. Baars, A. Amodeo, F. Madonna, and M. Iarlori. Earlinet single calculus chain – technical – part 2: Calculation of optical products. *Atmospheric Measurement Techniques*, 9(7):3009–3029, 2016.