TROPPO LoRa Tools

Author: Marco Rainone
Document Version: 1.0
Date: 2020/08/19

Index

1 Introduction

This document describes the tools developed for the "TROPPO LoRa" project, TROPospheric Personal Observatory using LoRa signals.

2 List of python3 library used

library	info	website
array	Efficient arrays of numeric values	https://docs.python.org/3/library/array.html
calendar	General calendar-related functions	https://docs.python.org/3/library/calendar.html
collections	Container datatypes	https://docs.python.org/3/library/collections.html
csv	Reading and Writing of CSV Files	https://docs.python.org/3/library/csv.html
datetime	Basic date and time types	https://docs.python.org/3/library/datetime.html
errno	Standard errno system symbols	https://docs.python.org/3.1/library/errno.html
folium	folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the Leaflet.js library. Manipulate your data in Python, then visualize it in a Leaflet map via folium.	https://pypi.org/project/folium/
ftplib	FTP protocol client	https://docs.python.org/3/library/ftplib.html
gc	Garbage Collector interface	https://docs.python.org/3/library/gc.html

library	info	website
geopy.distance	geopy is a Python client for several popular geocoding web services. geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.	https://pypi.org/project/geopy/
getopt	C-style parser for command line options	https://docs.python.org/3/library/getopt.html
json	JSON encoder and decoder	https://docs.python.org/3/library/json.html
matplotlib.pyplot	Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.	https://pypi.org/project/matplotlib/
numpy	NumPy can be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.	https://pypi.org/project/numpy/
os	Miscellaneous operating system interfaces	https://docs.python.org/3/library/os.html
pandas	Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive.	https://pypi.org/project/pandas/

library	info	website
plotly	plotly.py is an interactive, open-source, and browser-based graphing library for Python	https://pypi.org/project/plotly/
staticmap	A small, python-based library for creating map images with lines and markers.	https://pypi.org/project/staticmap/
sys	System-specific parameters and functions	https://docs.python.org/3/library/sys.html
time	Time access and conversions	https://docs.python.org/3/library/time.html
wget		https://pypi.org/project/python3-wget/
zipfile	Work with ZIP archives	https://docs.python.org/3/library/zipfile.html

3 List of programs

3.1 allgtwttn868.py

Program to generate the list of TTN 868 MHz gateways

It connects via ftp to TTN, reads the list of gateways and creates the csv file gtwttn-EU_863_870.csv, which contains the list of 868 MHz gateways.

The command must be run periodically, for example every 1 or 2 days, to ensure that the gateway list is up to date.

3.1.1 Program launch

```
$ python3 allgtwttn868.py
```

```
$ ls -l
total 48388
...
-rw-rw-rw- 1 root root   876841 May  5 09:07  gtwttn-EU_863_870.csv
…
```

3.2 dist-dev-gtwttn.py

The program analyzes the log downloaded from ttnmapper, filters the ttnmapper report and prepares it for subsequent processing.

3.2.1 Program launch

Inputs:

1. a ttnmapper report file name (example: rfsee_drivetest_unit_4.txt)
2. the distance from the device (transmitting node) to the gateway in km, to find the closest gateway (example: 20 means 20km)
3. case sensitivity flag in the gateway name. Indicating "no", the gateway name is checked in the gtwttn-EU_863_870.csv file without taking into account the upper / lower case of the letters.
4. output directory to be used to save the generated report in csv format

The program generates an output file with the same name as the ttnmapper report. The file extension is .csv.

For example, if the program parses the rfsee_drivetest_unit_4.txt file, the program generates a csv report named rfsee_drivetest_unit_4.csv.

To simplify the analysis, the records are sorted by decreasing distance between the device and gateway: the records with the greatest distance are placed first and those with shorter distance follow.

Record fields in the final report

1. time — time to receive message
2. distance — distance in km between device and gateway
3. nodeaddr — TTN Device ID
4. lat; lon — device coordinates in degrees and fractions
5. gwaddr — TTN Gateway ID
6. gtw_lat — TTN gateway latitude (as provided by the Gateway owner)
7. gtw_lon — TTN gateway longitude

Example of the records in the resulting csv file:

```
time;distance;nodeaddr;lat;lon;gwaddr;gtw_lat;gtw_lon
2020-03-25
09:52:55;4858;rfsee_drivetest_unit_4;52.0894;5.1035;008000000000A889;10.0;20.0
2020-04-08
…
13:58:42;71;rfsee_drivetest_unit_4;52.691;5.7636;0000024B080309C2;52.3364;4.8878
2020-04-10
14:02:19;65;rfsee_drivetest_unit_4;52.66;5.6891;0000024B080309C2;52.3364;4.8878
2020-04-10
14:02:29;64;rfsee_drivetest_unit_4;52.6579;5.6861;0000024B08030916;52.3363;4.887
4
2020-04-10
...
```

3.2.2 Example

The program:

- analyzes the file ./test/rfsee_drivetest_unit_4.txt,
- removes the records where the distance between device and gateway is less than 20Km and
- saves the resulting report in the directory ./test

```
python3 dist-dev-gtwttn.py -i ./test/rfsee_drivetest_unit_4.txt -d 20 -c "no" -o ./test
```

```
$ python3 dist-dev-gtwttn.py -i ./test/rfsee_drivetest_unit_4.txt -d 20 -c "no" -o ./test
                  time  distance               nodeaddr      lat     lon
gwaddr  gtw_lat  gtw_lon
0    2020-03-25 09:52:55      4858  rfsee_drivetest_unit_4  52.0894  5.1035
008000000000A889  10.0000  20.0000
1    2020-04-08 13:33:24      4858  rfsee_drivetest_unit_4  52.0874  5.1165
```

```
008000000000A889  10.0000  20.0000
2    2020-04-08 13:33:35      4858  rfsee_drivetest_unit_4  52.0878  5.1158
008000000000A889  10.0000  20.0000
3    2020-04-08 13:33:45      4858  rfsee_drivetest_unit_4  52.0876  5.1156
008000000000A889  10.0000  20.0000
4    2020-04-08 13:33:55      4858  rfsee_drivetest_unit_4  52.0875  5.1157
008000000000A889  10.0000  20.0000
..                   ...       ...                     ...      ...     ...
...      ...      ...
962  2020-04-07 06:43:06        20  rfsee_drivetest_unit_4  52.0645  4.8181
0000024B080E0FFD  52.0856   5.1092
963  2020-04-10 14:32:32        20  rfsee_drivetest_unit_4  52.3276  5.3460
mjs-gateway-3  52.1437   5.3643
964  2020-04-10 14:32:42        20  rfsee_drivetest_unit_4  52.3250  5.3445
mjs-gateway-3  52.1437   5.3643
965  2020-04-10 14:32:53        20  rfsee_drivetest_unit_4  52.3227  5.3426
0000024B08031D2D  52.3574   5.6329
966  2020-04-10 14:44:43        20  rfsee_drivetest_unit_4  52.1791  5.1805
0000024B08030954  52.0101   5.0537

[967 rows x 8 columns]
root@4b22874cd103:/home/tropo# ls test
rfsee_drivetest_unit_4.csv  rfsee_drivetest_unit_4.txt
root@4b22874cd103:/home/tropo#
```

3.3 rsigra-near.py

The program executes these steps:

1. processes the csv generated by dist-dev-gtwttn.py (ex: rfsee_drivetest_unit_4.csv)
2. analyzes the data provided by Integrated Global Radiosonde Archive (IGRA)
3. identifies the radiosondes that are closeser to the midpoint between devices and TTN gateways
4. automatically downloads the troposonde archives with minimum distance

Igra site:

https://www.ncdc.noaa.gov/data-access/weather-balloon/integrated-global-radiosonde-archive

ftp://ftp.ncdc.noaa.gov/pub/data/igra

3.3.1 Program launch

Inputs:

1. the output of dist-dev-gtwttn.py csv filename (example: rfsee_drivetest_unit_4.csv)
2. output directory to save the generated report in csv format

and the radiosonde archives downloaded from IGRA site.

3.3.2 Example:

python3 rsigra-near.py -i ./test/rfsee_drivetest_unit_4.csv -o ./test

```
The program inputs are:
   1. the csv ./test/rfsee_drivetest_unit_4.csv
   2. the ./test output directory

For each line of the rfsee_drivetest_unit_4.csv file, the program
identifies the codes of the radiosondes closer to the midpoint
between the device and the TTN gateway.
A list is created with the unique radiosonde codes (eg:
['TSM00060760', 'NLM00006260']) and the logs of the radiosondes
are automatically downloaded from the IGRA site.
```

python3 rsigra-near.py -i ./test/rfsee_drivetest_unit_4.csv -o ./test

```
... download: igra2-station-list.txt ...
                    time  distance                nodeaddr
lat     lon             gwaddr  gtw_lat  gtw_lon
0    2020-03-25 09:52:55      4858  rfsee_drivetest_unit_4
52.0894  5.1035  008000000000A889  10.0000  20.0000
1    2020-04-08 13:33:24      4858  rfsee_drivetest_unit_4
52.0874  5.1165  008000000000A889  10.0000  20.0000
2    2020-04-08 13:33:35      4858  rfsee_drivetest_unit_4
..                   ...       ...                    ...      ..
.     ...                ...      ...      ...
965  2020-04-10 14:32:42        20  rfsee_drivetest_unit_4
52.3250  5.3445     mjs-gateway-3  52.1437   5.3643
966  2020-04-10 14:32:53        20  rfsee_drivetest_unit_4
52.3227  5.3426  0000024B08031D2D  52.3574   5.6329
967  2020-04-10 14:44:43        20  rfsee_drivetest_unit_4
52.1791  5.1805  0000024B08030954  52.0100   5.0530

[968 rows x 8 columns]
row index: 0...
row index: 1...
row index: 2...
....
row index: 965...
row index: 966...
```

```
row index: 967...
N. radiosonde identificate: 2
['TSM00060760' 'NLM00006260']
/pub/data/igra
/
/pub/data/igra/derived/derived-por
ftp://ftp.ncdc.noaa.gov/pub/data/igra/derived/derived-por
... download: TSM00060760-drvd.txt.zip ...
... download: NLM00006260-drvd.txt.zip ...
Number of radiosonda files downloaded: 2
```

3.4 inpnear.py

The program executes these steps:

1. the user manually enters some parameters and the program generates a csv file containing only one record, compatible with the format generated by dist-dev-gtwttn.py
2. analyzes the data provided by Integrated Global Radiosonde Archive (IGRA)
3. identifies the radiosondes that are closesr to devices and TTN gateways
4. automatically downloads the radiosonde archives with minimum distance

3.4.1 Program launch

python3 inpnear.py

```
inpnear.py -o <path output csv>
Example:
inpnear.py -o ./data/result.csv
Store result data in ./data/result.csv file
```

3.4.2 Example

Suppose we want to analyze an event that took place on the following date and time:

2020-02-16 14.00.00

the transmitting node device has this identifier: **device_01**

and is positioned at these coordinates:

(lat, lon): (45.6093, 13.6034)

the gateway has this identifier: **gateway_01**

positioned at these coordinates:

(lat, lon): (45.9103, 13.9445)

The final report is **tst20200216.csv**, saved in the data subdirectory.

Run:

python3 inpnear.py -o ./test/tst20200216.csv

enter the following 5 parameters:

1. time of the event: (2020-02-16 14:00:00)

```
time (year-month-day hour:min:sec Example: 2020-03-25 09:52:55) ?
2020-02-16 14:00:00
  Input: [2020-02-16 14:00:00] OK (y/n) ? y
```

2. TTN device ID

```
node ID string ? device_01
  Input: [device_01] OK (y/n) ? y
```

3. device coordinates

```
Node position coordinates (latitude and longitude in degrees, ex:
45.6573 13.7694):
 ? 45.6093 13.6034
  Coordinates: [45.6093, 13.6034] OK (y/n) ? y
Coordinates: [45.6093, 13.6034]
```

4. TTN gateway ID

```
gateway ID string ? gateway_01
  Input: [gateway_01] OK (y/n) ? y
```

5. gateway coordinates

```
Gateway position coordinates (latitude and longitude in degrees, ex:
45.6573 13.7694):
 ? 45.9103 13.9445
  Coordinates: [45.9103, 13.9445] OK (y/n) ? y
Coordinates: [45.9103, 13.9445]
```

At this point, the program connects to the IGRA site and downloads the list of radiosondes

```
access to ftp://ftp.ncdc.noaa.gov/pub/data/igra ...
-------------------- ftp.ncdc.noaa.gov/pub/data/igra
get radiosonde list: igra2-station-list.txt ...
```

and proceeds to identify the radiosonde that is closest to the midpoint between device and the gateway

```
N. radiosonde identificate: 1
['ITM00016045']
```

finally, it generates the ./test/tst20200216.csv report file

```
                  time distance   nodeaddr      lat      lon      gwaddr
gtw_lat  gtw_lon        rs_id   rs_lat   rs_lon rs_distance
0  2020-02-16 14:00:00  42.7018  device_01  45.6093  13.6034  gateway_01
45.9103 13.9445  ITM00016045  45.9806  13.0592          60
```

3.5 map-rsigra.py

The program input is a csv file generated by inpnear.py or rsigra-near.py.

Generates an html geographic map with the locations of the device, gateway and igra radiosonde.

3.5.1 Program launch

python3 map-rsigra.py -i <log TTN events> -o <out dir>

3.5.2 Example

python3 map-rsigra.py -i test/budnag-20190828.csv -o test

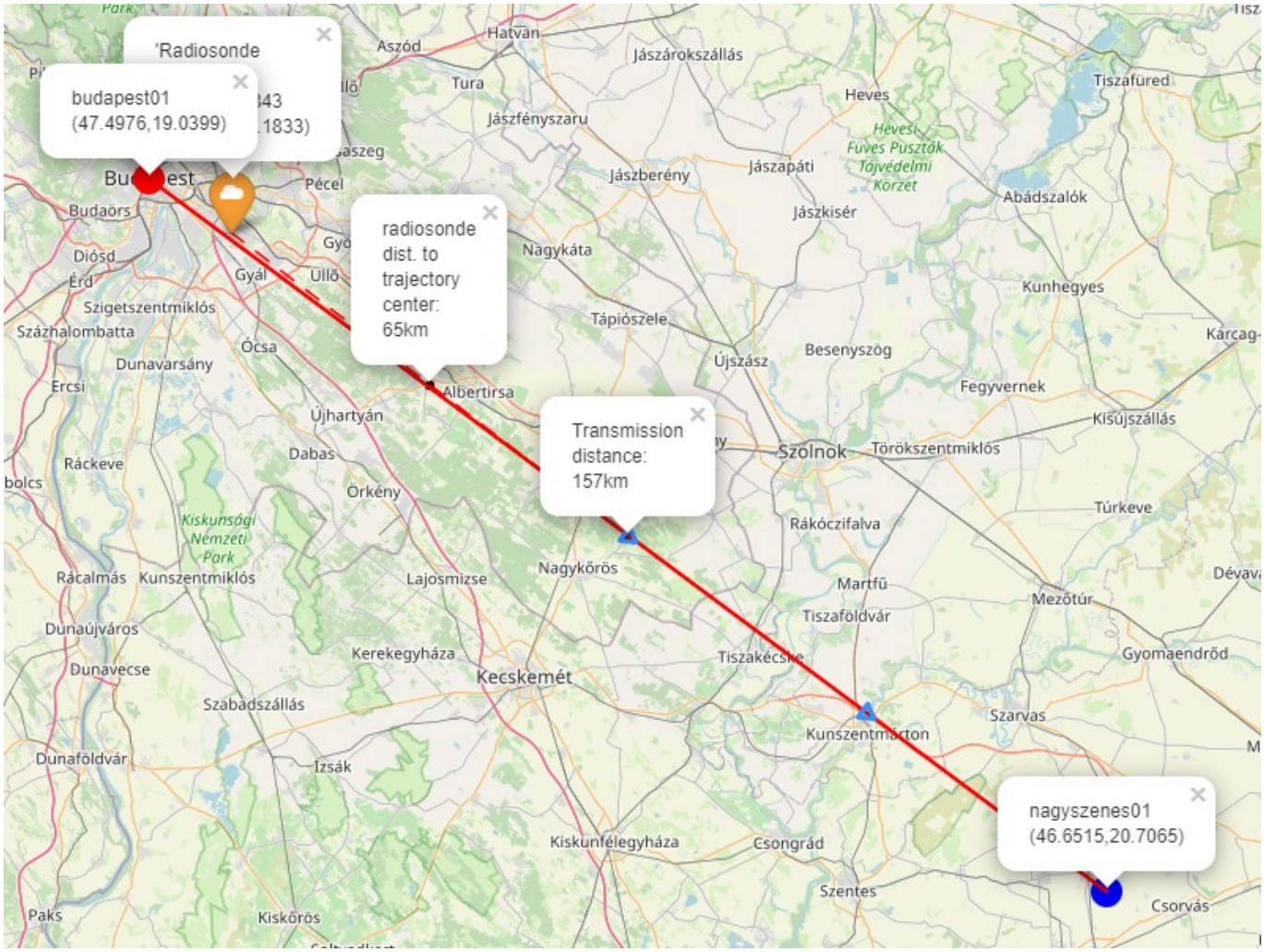
The program reads the budnag-20190828.csv file in the test subdirectory.

The map-budnag-20190828.html map is generated in the same test directory.

Note: By default, pop-up texts are normally displayed as soon as the page is opened.

If necessary, you can resize the map and possibly close any pop-ups that you don't want to appear.

Map generated by map-rsigra.py



3.6 get-rsigra.py

The input is the code of an igra radiosonde and the program downloads the file with the "derived" data

3.6.1 Program launch

python3 get-rsigra.py -i <code ID radiosonda> -o <output dir>

3.6.2 Example

From the analysis of rfsee_drivetest_unit_4.csv, one of the radiosondes closer to the positions listed in the file has this identifier: **'NLM00006260'**

to download the NLM00006260 data in the test directory, run:

python3 get-rsigra.py -i NLM00006260 -o ./test

```
$ python3 getrs05.py -i NLM00006260 -o ./test
Station string find: NLM00006260
Station search: NLM00006260
search string: [nlm00006260] ...
... download: igra2-station-list.txt ...
/pub/data/igra
/
/pub/data/igra/derived/derived-por
ftp://ftp.ncdc.noaa.gov/pub/data/igra/derived/derived-por
Found radiosonda: [NLM00006260]
... download: NLM00006260-drvd.txt.zip ...
/pub/data/igra/derived/derived-por
Number of files downloaded: 1
#NLM00006260
./test/NLM00006260-drvd.txt
./test/NLM00006260-drvd.idx
```

the NLM00006260-drvd.txt.zip file is downloaded into the test subdirectory, containing the 'derived' data log of the radiosonde probe code **NLM00006260**

3.7 graph-rsigra-day.py

According to the radiosonde ID and a date, the program processes and generates the html graphs of the slopes of N and M

3.7.1 Program launch

inputs:

1. path of radiosonde log file
2. time, in the year month day hour min format

The program extracts the radiosonde data acquired on the date provided and generates the html graphs of the slopes of N and M as a function of the height above the ground.

3.7.2 Example

Assume the following IGRA radiosonde file:

./test/NLM00006260-drvd.txt.zip

The program analyzes the radiosonde data and generates:

- the csv file with radiosonde acquisitions, and the calculation of N, M, slopeN, slopeM.

  NLM00006260-drvd-202002160000.csv

- The html graphs:

  1. slNH-NLM00006260-202002160000.html
  2. slMH-NLM00006260-202002160000.html

python3 graph-rsigra-day.py -i ./test/NLM00006260-drvd.txt.zip -t "2020 02 16 00 00"

```
python3 graph-rsigra-day.py -i ./test/NLM00006260-drvd.txt.zip -t
"2020 02 16 00 00"
dateSearch: [['2020', '02', '16', '00', '00']][202002160000]
nameZipIgraLog[NLM00006260-drvd][NLM00006260]
fpIdxIgraLog[/home/tropo/test/NLM00006260-drvd.idx]
operazione indicizzazione ...
#NLM00006260
/home/tropo/test/NLM00006260-drvd.txt
/home/tropo/test/NLM00006260-drvd.idx
... read file indice
... end read indice
start search time in log ...
search_time: [2020-02-16 00:00:00]
                    date    tm_epoch  pos_header   pos_data  n_rec
```

```
2148  2020-02-16 00:00:00  1581811200   295524582  295524740     41
... end search in log
Differenza di tempo in ore: 0
/home/tropo/test/NLM00006260-drvd.txt
/home/tropo/test/NLM00006260-drvd-202002160000.csv
#NLM00006260 2020
     HGHT    N          M  deltaN  deltaH    slopeN_H
0       2  319   319.314     NaN     NaN         NaN
1      35  317   322.495    -2.0    33.0  -60.606061
2     678  302   408.446   -15.0   643.0  -23.328149
3     701  302   412.057     0.0    23.0    0.000000
4    1204  285   474.028   -17.0   503.0  -33.797217
5    1375  280   495.875    -5.0   171.0  -29.239766
6    1849  264   554.293   -16.0   474.0  -33.755274
7    2075  253   578.775   -11.0   226.0  -48.672566
8    2190  235   578.830   -18.0   115.0 -156.521739
9    2285  217   575.745   -18.0    95.0 -189.473684
10   2349  215   583.793    -2.0    64.0  -31.250000
11   2491  212   603.087    -3.0   142.0  -21.126761
12   2951  209   672.307    -3.0   460.0   -6.521739
13   3425  202   739.725    -7.0   474.0  -14.767932
14   3743  190   777.651   -12.0   318.0  -37.735849
15   4305  182   857.885    -8.0   562.0  -14.234875
16   4318  182   859.926     0.0    13.0    0.000000
17   4491  177   882.087    -5.0   173.0  -28.901734
18   5331  165  1001.967   -12.0   840.0  -14.285714
19   5493  162  1024.401    -3.0   162.0  -18.518519
20   5568  160  1034.176    -2.0    75.0  -26.666667
21   6079  150  1104.403   -10.0   511.0  -19.569472
22   7210  130  1261.970   -20.0  1131.0  -17.683466
23   8375  113  1427.875   -17.0  1165.0  -14.592275
24   9096  103  1531.072   -10.0   721.0  -13.869626
25   9207  102  1547.499    -1.0   111.0   -9.009009
26   9232  102  1551.424     0.0    25.0    0.000000
27   9390  100  1574.230    -2.0   158.0  -12.658228
28  10052   93  1671.164    -7.0   662.0  -10.574018
29  10402   89  1722.114    -4.0   350.0  -11.428571
30  11090   82  1823.130    -7.0   688.0  -10.174419
31  11229   81  1843.953    -1.0   139.0   -7.194245
32  11545   77  1889.565    -4.0   316.0  -12.658228
33  11784   75  1925.088    -2.0   239.0   -8.368201
34  12026   73  1961.082    -2.0   242.0   -8.264463
35  12512   68  2032.384    -5.0   486.0  -10.288066
36  12754   64  2066.378    -4.0   242.0  -16.528926
37  13435   57  2166.295    -7.0   681.0  -10.279001
38  13516   56  2178.012    -1.0    81.0  -12.345679
39  15609   39  2489.613   -17.0  2093.0   -8.122312
40  16031   37  2553.867    -2.0   422.0   -4.739336
root@4fbc06fd454d:/home/tropo#
```

3.8 graph-rsigra-interval.py

Inputs:

1. path of a radiosonde log file
2. time in year month day hour min format
3. number of days of the radiosonde log to be analyzed

The program extracts the data of the radiosonde acquisitions closer to the date provided and generates html graphs of N, M and their slopes as a function of the height H in kilometers.

The number of traces in the graph depends on the number of launches of the radiosonde carried out on the dates and times in the specified time interval in days.

3.8.1 Example

Suppose we need to process the acquisition log of the GMM00010184 radiosonde, previously downloaded from the get-rsigra.py program in the test subdirectory (in test we find the GMM00010184-drvd.txt.zip archive) for 7 days.

The graph-rsigra-interval.py program processes the log contained in the archive ./test/GMM00010184-drvd.txt.zip.

For all launches made during the 7 days specified, it generates:

- a series of csv files with the radiosonde acquisitions and the calculation of N, M, slopeN e slopeM.
- The html graphs:
  1. slNH-GMM00010184-202002160000-007days.html
  2. slMH-GMM00010184-202002160000-007days.html

run the program with these parameters:

```
python3 graph-rsigra-interval.py -i
test/GMM00010184-drvd.txt.zip -t "2020 02 16 00 00"
-d 7
```

```
$ python3 graph-rsigra-interval.py -i test/GMM00010184-drvd.txt.zip -t "2020 02
16 00 00" -d 7
            dateSearch: [['2020', '02', '16', '00', '00']][202002160000]
            nameZipIgraLog[GMM00010184-drvd][GMM00010184]
            fpIdxIgraLog[/home/tropo/test/GMM00010184-drvd.idx]
            operazione indicizzazione ...
            #GMM00010184
```

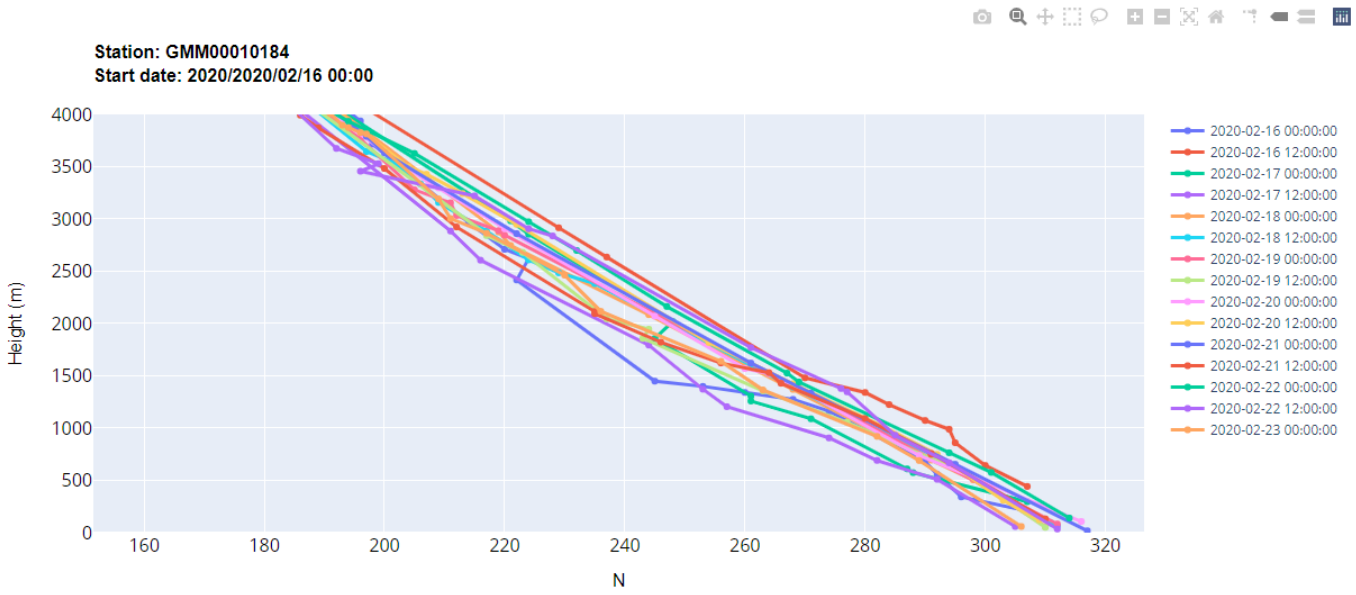
```
/home/tropo/test/GMM00010184-drvd.txt
/home/tropo/test/GMM00010184-drvd.idx
... read file indice
... end read indice
start search time in log ...
search_time: [2020-02-16 00:00:00]
                    date    tm_epoch  pos_header   pos_data  n_rec
3744  2020-02-16 00:00:00  1581811200   308438922  308439080     54
3745  2020-02-16 12:00:00  1581854400   308447288  308447446     48
….
3756  2020-02-22 00:00:00  1582329600   308528370  308528528     56
3757  2020-02-22 12:00:00  1582372800   308537040  308537198     53
3758  2020-02-23 00:00:00  1582416000   308545254  308545412     64
... end search in log
n. righe risIdx: 15
Differenza di tempo in ore: 0
/home/tropo/test/GMM00010184-drvd.txt
#GMM00010184 2020
/home/tropo/test/GMM00010184-20200216000000.csv
    HGHT    N        M  deltaN  deltaH    slopeN_H
1     67  312  322.519    -3.0    65.0  -46.153846
2    207  307  339.499    -5.0   140.0  -35.714286
3    342  296  349.694   -11.0   135.0  -81.481481
...
17  3717  198  781.569    -4.0   180.0  -22.222222
18  3938  196  814.266    -2.0   221.0   -9.049774
19  5261  166  991.977   -30.0  1323.0  -22.675737
/home/tropo/test/GMM00010184-20200216120000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1    442  307  376.394   -16.0   440.0 -36.363636
2    643  300  400.951    -7.0   201.0 -34.825871
3    860  295  430.020    -5.0   217.0 -23.041475
…..
8   1481  270  502.517   -10.0   140.0 -71.428571
9   2636  237  650.852   -33.0  1155.0 -28.571429
10  2916  229  686.812    -8.0   280.0 -28.571429
11  4661  180  911.777   -49.0  1745.0 -28.080229
/home/tropo/test/GMM00010184-20200217000000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1    296  307  353.472   -13.0   294.0 -44.217687
2    574  288  378.118   -19.0   278.0 -68.345324
3    610  287  382.770    -1.0    36.0 -27.777778
…..
7   1857  245  536.549   -16.0   550.0 -29.090909
8   2019  248  564.983     3.0   162.0  18.518519
9   2854  224  672.078   -24.0   835.0 -28.742515
10  2983  221  689.331    -3.0   129.0 -23.255814
11  4211  188  849.127   -33.0  1228.0 -26.872964
/home/tropo/test/GMM00010184-20200217120000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1     59  305  314.263    -2.0    57.0 -35.087719
2    510  292  372.070   -13.0   451.0 -28.824834
3    691  282  390.487   -10.0   181.0 -55.248619
…....
7   1794  244  525.658    -9.0   422.0 -21.327014
8   2606  216  625.142   -28.0   812.0 -34.482759
9   2885  211  663.945    -5.0   279.0 -17.921147
10  4043  186  820.751   -25.0  1158.0 -21.588946
```

```
/home/tropo/test/GMM00010184-20200218000000.csv
   HGHT    N        M  deltaN  deltaH   slopeN_H
1    59  310  319.263    -3.0    57.0 -52.631579
2   505  298  377.285   -12.0   446.0 -26.905830
3   694  291  399.958    -7.0   189.0 -37.037037
...
6  2747  221  652.279   -23.0   662.0 -34.743202
7  2876  219  670.532    -2.0   129.0 -15.503876
8  4617  179  903.869   -40.0  1741.0 -22.975302
/home/tropo/test/GMM00010184-20200218120000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1     91  311  325.287    -4.0    89.0 -44.943820
2    725  292  405.825   -19.0   634.0 -29.968454
3    959  283  433.563    -9.0   234.0 -38.461538
…….
10  3336  206  729.752    -3.0   178.0 -16.853933
11  3529  202  756.053    -4.0   193.0 -20.725389
12  3645  197  769.265    -5.0   116.0 -43.103448
13  4258  184  852.506   -13.0   613.0 -21.207178
/home/tropo/test/GMM00010184-20200219000000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1     83  312  325.031    -3.0    81.0 -37.037037
2    710  291  402.470   -21.0   627.0 -33.492823
3   1384  269  486.288   -22.0   674.0 -32.640950
…..
8   3155  211  706.335    -1.0   122.0  -8.196721
9   3279  205  719.803    -6.0   124.0 -48.387097
10  3766  196  787.262    -9.0   487.0 -18.480493
11  4316  183  860.612   -13.0   550.0 -23.636364
/home/tropo/test/GMM00010184-20200219120000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1     51  310  318.007    -3.0    49.0 -61.224490
2    650  295  397.050   -15.0   599.0 -25.041736
3    684  292  399.388    -3.0    34.0 -88.235294
……….
9   2681  223  643.917   -14.0   636.0 -22.012579
10  2841  217  663.037    -6.0   160.0 -37.500000
11  2852  217  664.764     0.0    11.0   0.000000
12  4500  178  884.500   -39.0  1648.0 -23.665049
/home/tropo/test/GMM00010184-20200220000000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1    107  316  332.799    -4.0   105.0 -38.095238
2    739  289  405.023   -27.0   632.0 -42.721519
3   1409  268  489.213   -21.0   670.0 -31.343284
……..
7   2908  220  676.556   -25.0   837.0 -29.868578
8   3297  209  726.629   -11.0   389.0 -28.277635
9   3320  208  729.240    -1.0    23.0 -43.478261
10  4133  188  836.881   -20.0   813.0 -24.600246
…….
/home/tropo/test/GMM00010184-20200222120000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1     35  312  317.495    -1.0    33.0 -30.303030
2    668  294  398.876   -18.0   633.0 -28.436019
3    932  285  431.324    -9.0   264.0 -34.090909
4   1347  277  488.479    -8.0   415.0 -19.277108
……….
11  3529  199  753.053     3.0    72.0  41.666667
```

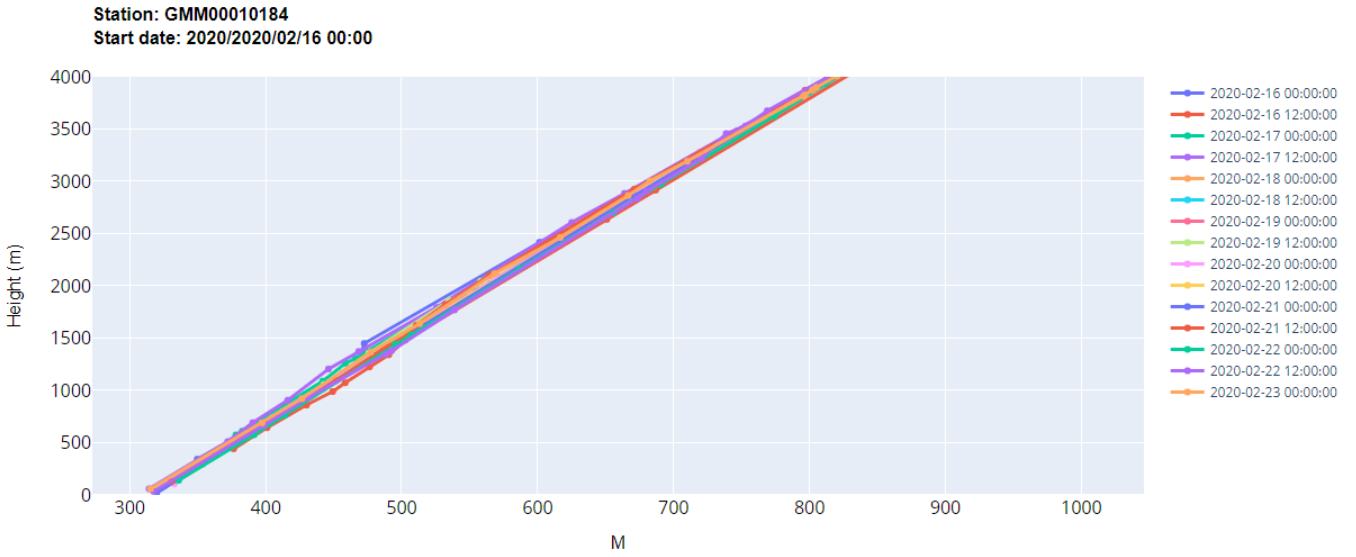
```
12  3675  192  768.975    -7.0   146.0 -47.945205
13  5061  166  960.577   -26.0  1386.0 -18.759019
/home/tropo/test/GMM00010184-20200223000000.csv
    HGHT    N        M  deltaN  deltaH   slopeN_H
1     59  306  315.263    -2.0    57.0 -35.087719
2    690  289  397.330   -17.0   631.0 -26.941363
3    921  282  426.597    -7.0   231.0 -30.303030
4   1365  263  477.305   -19.0   444.0 -42.792793
5   1639  256  513.323    -7.0   274.0 -25.547445
6   2117  236  568.369   -20.0   478.0 -41.841004
7   2463  230  616.691    -6.0   346.0 -17.341040
8   2864  217  666.648   -13.0   401.0 -32.418953
9   3004  211  682.628    -6.0   140.0 -42.857143
10  3192  209  710.144    -2.0   188.0 -10.638298
11  3814  197  795.798   -12.0   622.0 -19.292605
12  3826  196  796.682    -1.0    12.0 -83.333333
13  3874  194  802.218    -2.0    48.0 -41.666667
14  3898  193  804.986    -1.0    24.0 -41.666667
15  4008  191  820.256    -2.0   110.0 -18.181818
root@f82dee987f29:/home/tropo#
```

3.8.2 Reports generated with the parameters of the example

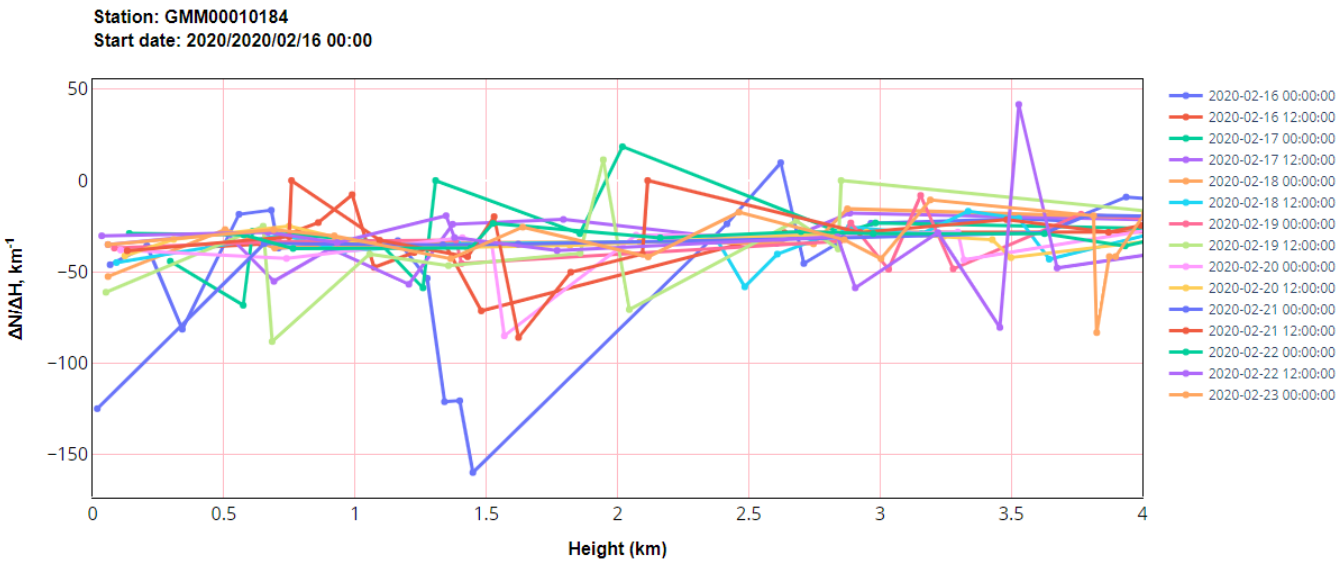
File reNH-GMM00010184-202002160000-007days.html



file reMH-GMM00010184-202002160000-007days.html



file slNH-GMM00010184-202002160000-007days.html



End of file.