pyRinex repository

Introduction

The pyRinex repository is based on the georinex repository which reads RINEX v3 and RINEX v2 and performs batch conversion to NetCDF4/HDF5 in Python or Matlab. Using the NetCDF4/HDF5 format gives far faster access to the observables than parsing a RINEX observation file.

Please consult the Readme file on this page which is from georinex or visit the georinex repository.

Batch conversion of RINEX files using rnx2hdf5

```
$ rnx2hdf5 -h
usage: rnx2hdf5 [-h] [-o OUT] [-v] [-p] [-u USE [USE ...]]
                [-m MEAS [MEAS ...]] [-t TLIM TLIM] [-useindicators] [-strict]
                indir [glob]

example of reading RINEX 2/3 Navigation/Observation file

positional arguments:
  indir                 path to RINEX 2 or RINEX 3 files to convert
  glob                  file glob pattern

optional arguments:
  -h, --help            show this help message and exit
  -o OUT, --out OUT     write data to path or file as NetCDF4
  -v, --verbose
  -p, --plot            display plots
  -u USE [USE ...], --use USE [USE ...]
                        select which GNSS system(s) to use
  -m MEAS [MEAS ...], --meas MEAS [MEAS ...]
                        select which GNSS measurement(s) to use
  -t TLIM TLIM, --tlim TLIM TLIM
                        specify time limits (process part of file)
  -useindicators        use SSI, LLI indicators (signal, loss of lock)
  -strict               do not use speculative preallocation (slow) let us
                        know if this is needed
```

An example run on directory /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/19250 converting all RINEX observation files *.0.19O (Galileo, GPS and COMB) putting the converted output in directory /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/19250/nc is:

```
$ rnx2hdf5 /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250 *0.19O -o /media/amuls/RXTURPBACKUP
INFO:root:opening 217.534199 MByte GPSS2500.19O
INFO:root:opening 217.534199 MByte GPSS2500.19O
INFO:root:opening 217.534199 MByte GPSS2500.19O
saving OBS: /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250/nc/GPSS2500.19O.nc
saving OBS: /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250/nc/GALI2500.19O.nc
INFO:root:opening 308.463966 MByte COMB2500.19O
INFO:root:opening 308.463966 MByte COMB2500.19O
```

```
INFO:root:opening 308.463966 MByte COMB2500.19O
saving OBS: /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250/nc/COMB2500.19O.nc
```

Executing `rnx2hdf5` on daily `RINEX` files is very time consuming (takes hours), the above conversion took about 17 hours.

Python scripts

Script `rnxplot.py`

`rnxplot.py` is a graphical `PyQT5` based program that reads `RINEX` observation files or `NetCDF4/HDF5` files to plot the observables. The preferred way is to previously convert the `RINEX` observation file to the `NetCDF4/HDF5` format as input for `rnxplot.py`.

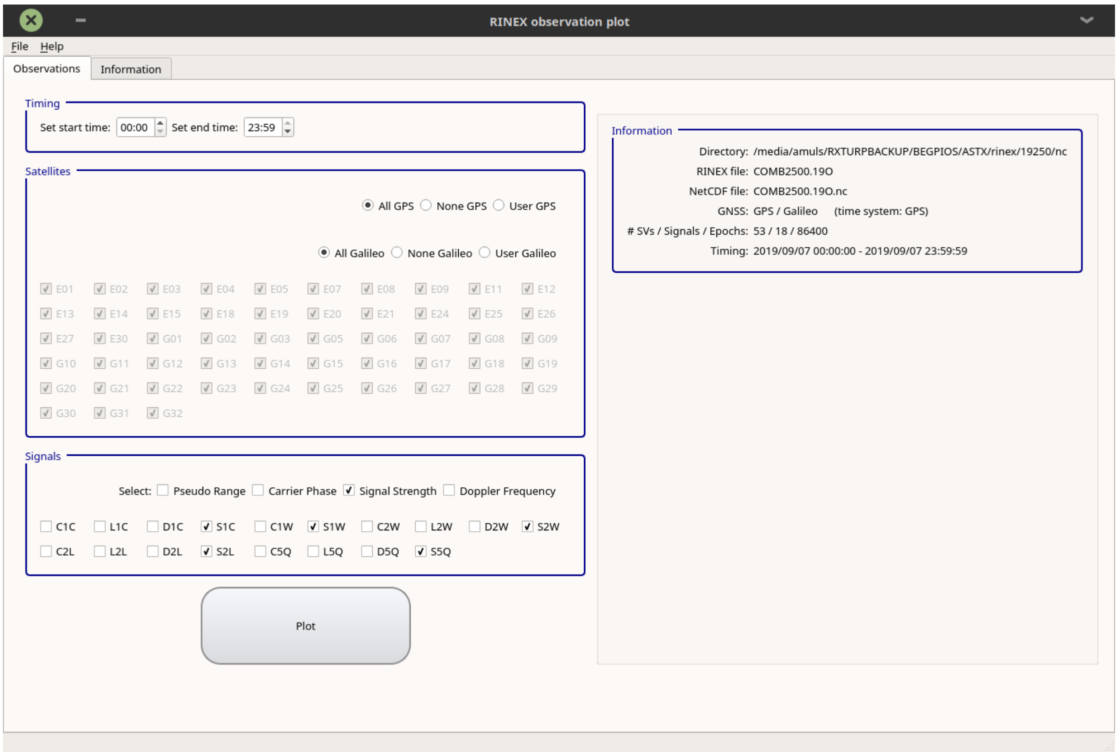
The conversion from `RINEX` to `NetCDF4/HDF5` takes a long time, especially for daily files. This conversion is done using the `georinex` provided file `rnx2hdf5`. The following is an example of converting the daily observation files from an `AsteRx` receiver for Galileo only, GPS only and Galileo+GPS observations. It took about 27 hours to complete.

```
$ rnx2hdf5 /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250 *0.19O -o /media/amuls/RXTURPBACKUP
INFO:root:opening 217.534199 MByte GPSS2500.19O
INFO:root:opening 217.534199 MByte GPSS2500.19O
INFO:root:opening 217.534199 MByte GPSS2500.19O
saving OBS: /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250/nc/GPSS2500.19O.nc
saving OBS: /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250/nc/GALI2500.19O.nc
INFO:root:opening 308.463966 MByte COMB2500.19O
INFO:root:opening 308.463966 MByte COMB2500.19O
INFO:root:opening 308.463966 MByte COMB2500.19O
```

The `RINEX` to `NetCDF4/HDF5` conversion can be done without user interaction and yields a enormous time gain afterwards since reading the `NetCDF/HDF5` files is much faster than reading the `RINEX` observation file during the analysis.

The script `rnxplot.py` is more usable for the interference tests (which have shorter observation times). The following figure shows the start window of `rnxplot.py`, while subsequent figures show plots created.

Running `enxplot.py` also creates comma separated values or `CSV` files stored in the sub directory `csv`, which can be used for further analysis (e.g. using a spreadsheet)



Timing

Set start time: 00:00 Set end time: 23:59

Satellites

● All GPS ○ None GPS ○ User GPS

● All Galileo ○ None Galileo ○ User Galileo

☑ E01 ☑ E02 ☑ E03 ☑ E04 ☑ E05 ☑ E07 ☑ E08 ☑ E09 ☑ E11 ☑ E12
☑ E13 ☑ E14 ☑ E15 ☑ E18 ☑ E19 ☑ E20 ☑ E21 ☑ E24 ☑ E25 ☑ E26
☑ E27 ☑ E30 ☑ G01 ☑ G02 ☑ G03 ☑ G05 ☑ G06 ☑ G07 ☑ G08 ☑ G09
☑ G10 ☑ G11 ☑ G12 ☑ G13 ☑ G14 ☑ G15 ☑ G16 ☑ G17 ☑ G18 ☑ G19
☑ G20 ☑ G21 ☑ G22 ☑ G23 ☑ G24 ☑ G25 ☑ G26 ☑ G27 ☑ G28 ☑ G29
☑ G30 ☑ G31 ☑ G32

Signals

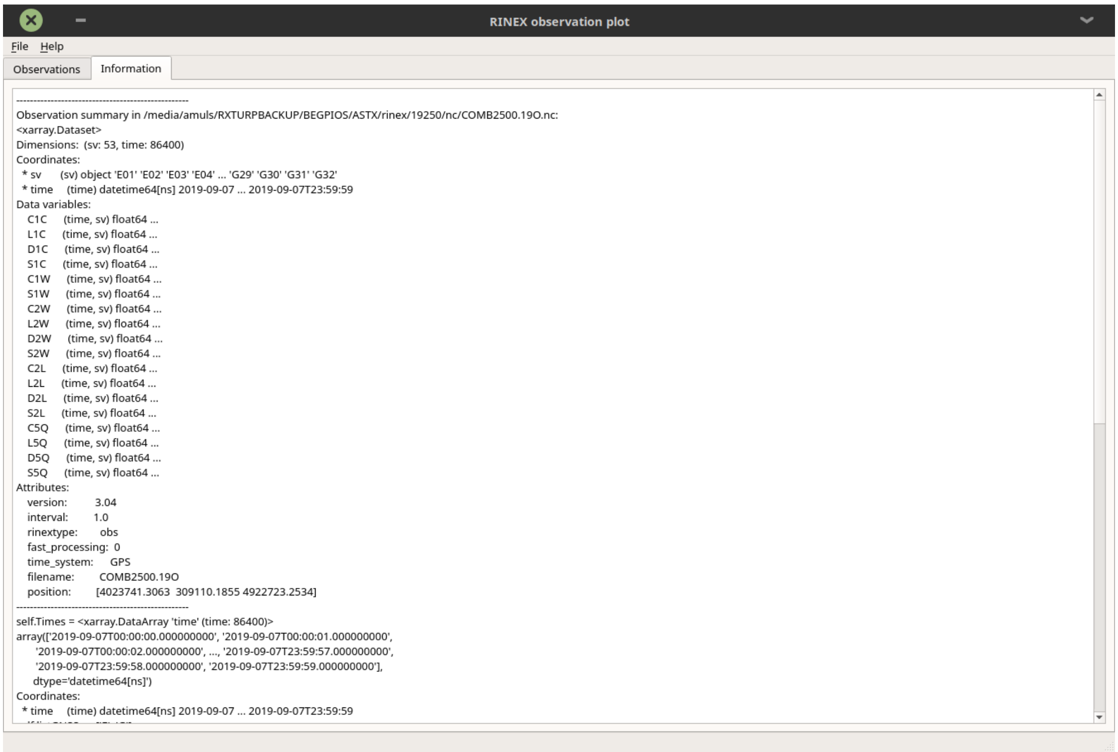
Select: □ Pseudo Range □ Carrier Phase ☑ Signal Strength □ Doppler Frequency

□ C1C □ L1C □ D1C ☑ S1C □ C1W ☑ S1W □ C2W □ L2W □ D2W ☑ S2W
□ C2L □ L2L □ D2L ☑ S2L □ C5Q □ L5Q □ D5Q ☑ S5Q

Information

Directory: /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250/nc
RINEX file: COMB2500.19O
NetCDF file: COMB2500.19O.nc
GNSS: GPS / Galileo (time system: GPS)
SVs / Signals / Epochs: 53 / 18 / 86400
Timing: 2019/09/07 00:00:00 - 2019/09/07 23:59:59

Figure 1: `rnxplot` start window



```
--------------------------------------------------
Observation summary in /media/amuls/RXTURPBACKUP/BEGPIOS/ASTX/rinex/19250/nc/COMB2500.19O.nc:
<xarray.Dataset>
Dimensions:  (sv: 53, time: 86400)
Coordinates:
  * sv       (sv) object 'E01' 'E02' 'E03' 'E04' ... 'G29' 'G30' 'G31' 'G32'
  * time     (time) datetime64[ns] 2019-09-07 ... 2019-09-07T23:59:59
Data variables:
    C1C      (time, sv) float64 ...
    L1C      (time, sv) float64 ...
    D1C      (time, sv) float64 ...
    S1C      (time, sv) float64 ...
    C1W      (time, sv) float64 ...
    S1W      (time, sv) float64 ...
    C2W      (time, sv) float64 ...
    L2W      (time, sv) float64 ...
    D2W      (time, sv) float64 ...
    S2W      (time, sv) float64 ...
    C2L      (time, sv) float64 ...
    L2L      (time, sv) float64 ...
    D2L      (time, sv) float64 ...
    S2L      (time, sv) float64 ...
    C5Q      (time, sv) float64 ...
    L5Q      (time, sv) float64 ...
    D5Q      (time, sv) float64 ...
    S5Q      (time, sv) float64 ...
Attributes:
    version:         3.04
    interval:        1.0
    rinextype:       obs
    fast_processing:  0
    time_system:     GPS
    filename:        COMB2500.19O
    position:        [4023741.3063  309110.1855 4922723.2534]
--------------------------------------------------
self.Times = <xarray.DataArray 'time' (time: 86400)>
array(['2019-09-07T00:00:00.000000000', '2019-09-07T00:00:01.000000000',
       '2019-09-07T00:00:02.000000000', ..., '2019-09-07T23:59:57.000000000',
       '2019-09-07T23:59:58.000000000', '2019-09-07T23:59:59.000000000'],
      dtype='datetime64[ns]')
Coordinates:
  * time     (time) datetime64[ns] 2019-09-07 ... 2019-09-07T23:59:59
```

Figure 2: `rnxplot` information tab

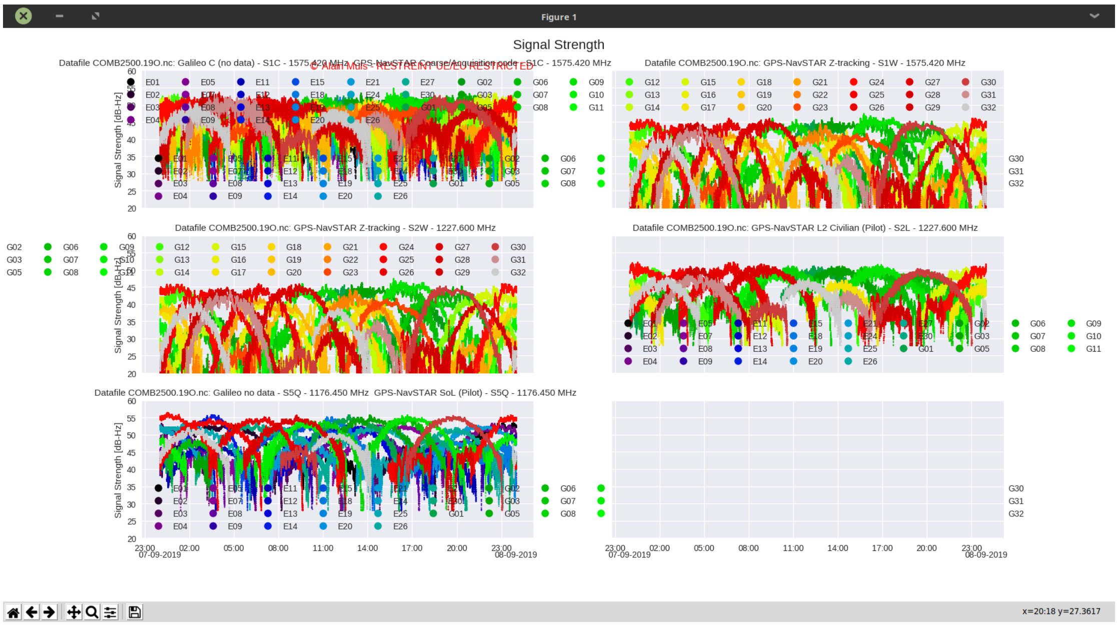


Figure 3: `rnxplot` signal strength all SVs

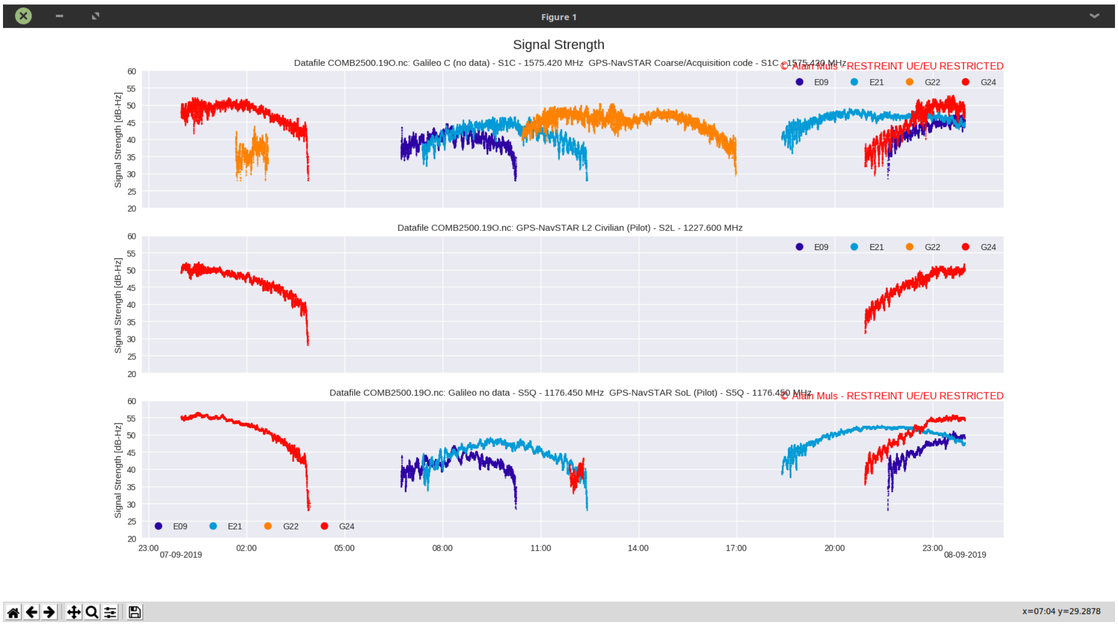


Figure 4: `rnxplot` signal strength selected SVs

Script `rnxdiff.py`

Using the created `CSV` files, `rnxdiff.py` makes a comparison (difference) between observables on 2 different frequencies.

```
$ rnxdiff.py --help
usage: rnxdiff.py [-h] [-d DIR] -f FILES FILES -g GNSS -s SIGNALS SIGNALS
                  [-m MOVAVG]
                  [-l {CRITICAL,ERROR,WARNING,INFO,DEBUG,NOTSET} {CRITICAL,ERROR,WARNING,INFO,DEE

rnxdiff.py compares between similar signals of different navigation services

optional arguments:
  -h, --help            show this help message and exit
  -d DIR, --dir DIR     Directory of SBF file (defaults to .)
  -f FILES FILES, --files FILES FILES
                        Filenames of 2 CSV files to compare
  -g GNSS, --gnss GNSS  GNSS System Name
  -s SIGNALS SIGNALS, --signals SIGNALS SIGNALS
                        Signal names to compare
  -m MOVAVG, --movavg MOVAVG
                        moving average of difference [sec] (defaults 60s)
  -l {CRITICAL,ERROR,WARNING,INFO,DEBUG,NOTSET} {CRITICAL,ERROR,WARNING,INFO,DEBUG,NOTSET}, --log
                        specify logging level console/file (default INFO
                        DEBUG)
```

An example for running a comparison between signal strengths and pseudo-ranges for Galileo

Open Service signals on E1 and E5 is:

```
$ /home/amuls/amPython/pyRinex/am/rnxdiff.py -g Galileo -s S1C S5Q \
  -d ~/RxTURP/BEGPIOS/ASTX/rinex/19250/nc/csv/ \
  -f COMB2500-190-nc-E-S1C.csv  COMB2500-190-nc-E-S5Q.csv

$ /home/amuls/amPython/pyRinex/am/rnxdiff.py -g Galileo -s C5C C5Q \
  -d ~/RxTURP/BEGPIOS/ASTX/rinex/19250/nc/csv/ \
  -f COMB2500-190-nc-E-C1C.csv  COMB2500-190-nc-E-C5Q.csv
```

which produces following plots (plots are saved in a `png` sub-directory). On the pseudo-range plot, the influence of the 1ms clock jumps are clearly visible. The difference and a moving average of the difference are calculated.

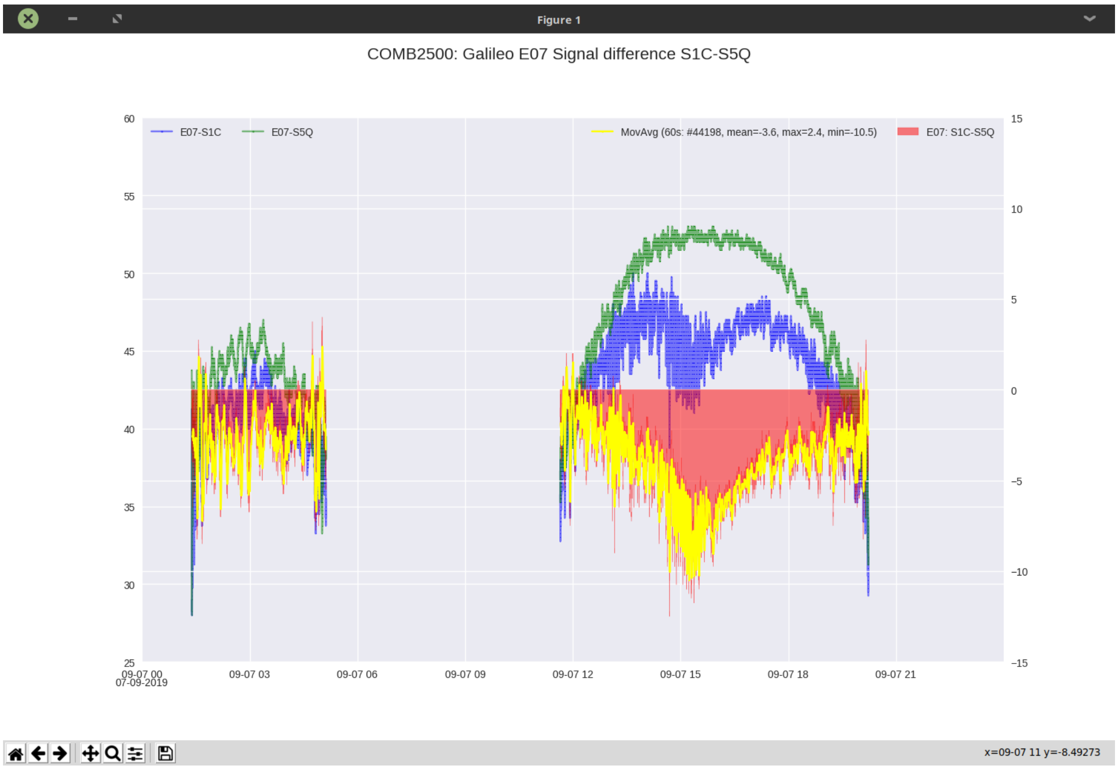


Figure 5: Galileo Signal Strength comparison

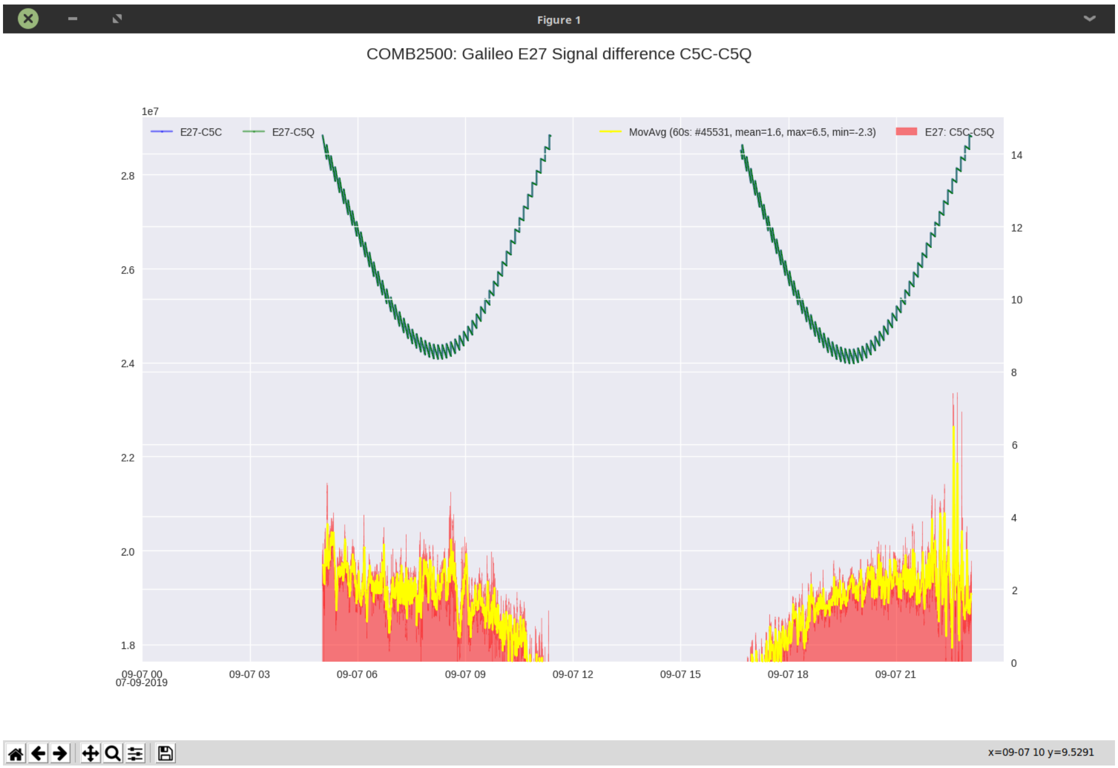


Figure 6: Galileo Pseudo Range comparison