Supplement

gnssrefl: an open source python software package for GNSS interferometric reflectometry applications

Kristine M. Larson

Bonn University

kristinem.larson@gmail.com

The following is an example of how to use *gnssrefl* to measure tides. We use GNSS data from station Mornington Island archived at Geoscience Australia.

1. Station information

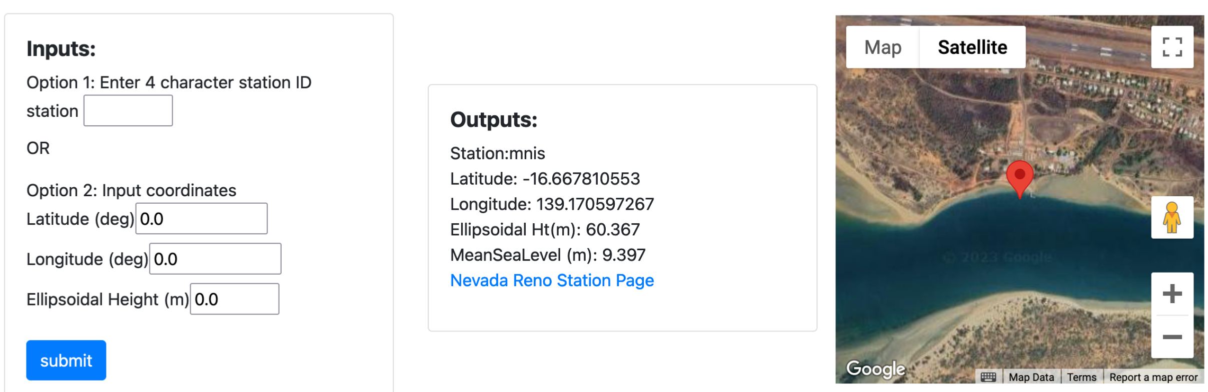
Short marker name: MNIS

Long marker name: MNIS00AUS.

Information (including possibly photographs) about this site can be found at Geoscience Australia:

https://gnss-site-manager.geodesy.ga.gov.au/site/MNIS

You can currently use the https://gnss-reflections.org/geoid API to display a google Earth image of the nearby surroundings of the site. Type the 4-character station name into the API and hit submit. It should return the following:



2. Requirements for installation and operation of gnssrefl

Internet access is required to install gnssrefl and run this example.

The Geoscience Australia GNSS data archive and the GFZ orbit archives must be online (GFZ-orbits).
The goal of this supplement is to show you the commands in the *gnssrefl* software to create results for relative sea level. Documentation for the *gnssrefl* software is available as a pdf and online:

https://gnssrefl.readthedocs.io/en/latest/

gnssrefl can be installed directly from GitHub, from pypi, or via docker. The main difference between the docker and GitHub/pypi installations is that for the latter the user must have a python version between 3.8 and 3.10 installed on their machine. This requirement is due to how python uses fortran bindings in the numpy library. Eventually this python version restriction will be removed.

Below we outline how to use the docker installation to run *gnssrefl*. This means the person using this test case will not have to install python or any of the supporting executables. One disadvantage of the docker is that figures created by the code are not automatically displayed to the screen. However, the code will always tell you where the figure files (png) are being written. Any utility can be used to view them. If individuals prefer to use their own python install, they will need to follow the instructions online:

https://gnssrefl.readthedocs.io/en/latest/pages/README_install.html

Users running DOS may only run *gnssrefl* with the docker image.

3. Run *gnssrefl*

To ensure you will have the same results as provided in *GPS Solutions* and via the *GPS Tool Box*, you need to use the same input files and the same version of the *gnssrefl* code. To simplify this process, we will use the docker image for version 2.5.0 in this example.

3.1 Install the docker

Install the docker appropriate for your local machine.

https://docker.com

Open a terminal window

Install/download the docker image for gnssrefl version 2.5.0

```
docker pull ghcr.io/kristinemlarson/gnssrefl:2.5.0
```

Make a directory where you want the run gnssrefl. cd into it.

```
mkdir rundocker
```

```
cd rundocker
```

Start the docker (type this command on one line):

```
docker run -it -v $(pwd)/refl_code:/etc/gnssrefl/refl_code/ --name gnssrefl
ghcr.io/kristinemlarson/gnssrefl:2.5.0 /bin/bash
```

The code uses environment variables to define where output is stored. When using a dollar sign like $REFL_CODE, we are referring to the environment variable REFL_CODE. For the docker the environment variables are set for you. For local installs, you must define the environment variables every time the software is used; most users set this when they login.

3.2 Check that the docker is working

Type and hit return:

```
query_unr mnis
```

This should return the following to the screen:

```
cp from local directory to Files directory because that is where it goes
Using database  /etc/gnssrefl/refl_code/Files/station_pos_2024.db
XYZ (m): -4624622.6423 3996010.4957 -1817663.547
LLH (deg,deg,m): -16.667811 139.170597 60.367
Sea Level (m):  9.397
```

For coastal sites like mnis, the reported Sea Level value reports the average vertical distance between the GNSS antenna reference point and the ocean. In this example we will expect the average reflector height to be approximately this value. The height given in LLH is the ellipsoidal height. To compute the relevant sea level height, an EGM96 gravity model was used. A station coordinate database from the University of Nevada Reno is also provided in *gnssrefl*. Here the database updated in 2024 is stored locally.

For more detail on installing and using the docker image, please see the online documentation:

https://gnssrefl.readthedocs.io/en/latest/pages/docker_cl_instructions.html

3.2 Translate RINEX files

The first step is to translate two weeks of multi-GNSS data in the RINEX 3 format from the year 2023 and days of year 137-151. To take advantage of multi-GNSS tracking, we explicitly request the gnss orbit option.

```
rinex2snr mnis00aus 2023 137 -doy_end 151 -orb gnss -archive ga
```

This command creates 15 SNR files. To see the names of the files created by this command, type:

```
ls -1 $REFL_CODE/2023/snr/mnis
```

This command returns:

```
mnis1370.23.snr66.gz
mnis1380.23.snr66.gz
mnis1390.23.snr66.gz
mnis1400.23.snr66.gz
mnis1410.23.snr66.gz
mnis1420.23.snr66.gz
mnis1430.23.snr66.gz
mnis1440.23.snr66.gz
mnis1450.23.snr66.gz
mnis1460.23.snr66.gz
mnis1470.23.snr66.gz
mnis1480.23.snr66.gz
mnis1490.23.snr66.gz
mnis1500.23.snr66.gz
mnis1510.23.snr66.gz
```

More details on *rinex2snr*:

https://gnssrefl.readthedocs.io/en/latest/api/gnssrefl.rinex2snr_cl.html

3.3 Assess reflection zones

gnssrefl provides a reflection zone mapping tool called `refl_zones`. However, to display the results (stored in a kml file), one also needs to install Google Earth. To avoid asking the user to do that, for this example we recommend using the online tool hosted at https://gnss-reflections.org/rzones.

From a browser, you could simply enter the station name and hit return. This produces the following API command, which can also be used.

http://gnss-reflections.org/rzones?station=mnis&lat=0.0&lon=0.0&height=0.0&msl=msl&RH=2&freq=1&nyquist=0&srate=30&eang=1&azim1=0&azim2=360&system=gps

This returns a google map with L1 GPS Fresnel (reflection) zones for all azimuths and elevation angle limits of 5-15 degrees.

Our preferred azimuth mask is from 120-240 degrees:

http://gnss-reflections.org/rzones?station=mnis&lat=0.0&lon=0.0&height=0.0&msl=msl&RH=2&freq=1&nyquist=0&srate=30&eang=1&azim1=120&azim2=240&system=gps

There are two advantages to using `refl_zones`. The first is that you can set your own list of elevation angles (up to a maximum of 5 different elevation angles) whereas the online tool has preset elevation angle options. The second advantage is that with Google Earth you can look at reflection zones using imagery on different days. This can be important when the web app is showing imagery for the station when the water levels are at high tide and you would like to see those reflection zones plotted at low tide. It can also be useful to have Google Earth images to see when a site is impacted by ice or ships.

The equivalent *refl_zones* command for mnis would be:

```
refl_zones mnis -azlist 120 240
```

We did not need to input coordinates for mnis because it is in the *gnssrefl* database.

Documentation

https://gnssrefl.readthedocs.io/en/latest/api/gnssrefl.refl_zones_cl.html

3.4 Assess maximum resolvable reflector height

You can use the Geoscience Australia database URL given earlier to confirm that data from MNIS are distributed at a 30 second sampling rate. In this section we are checking to see if 30 second sampling is sufficient to estimate the expected reflector heights ~9.4 meters (with tidal variations superimposed).

Type:

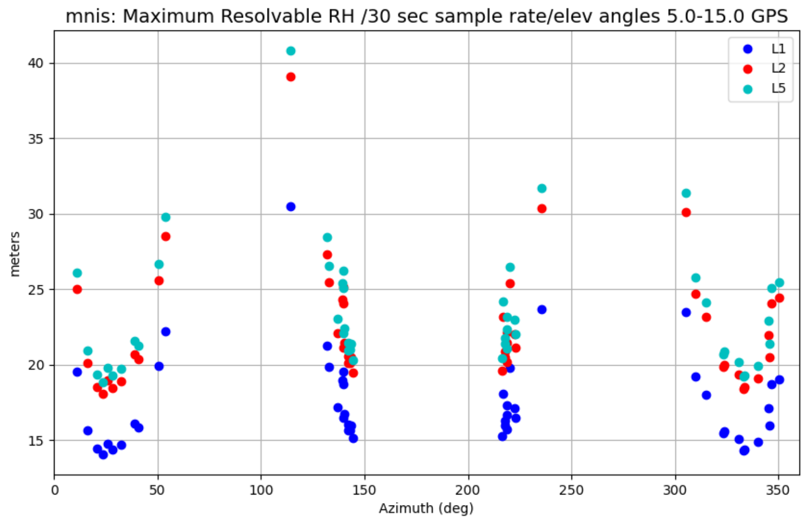
```
max_resolve_RH -h
```

A user can assume the station coordinates are in the *gnssrefl* database or you need to provide them on the command line. We will continue to use minimum and maximum elevation angles of 5 degrees (e1) and 15 degrees (e2) as those are generally a good place to start for water sites. This must be specified as the defaults for *max_resolve_RH* are 5 and 25 degrees.

Type:

```
max_resolve_RH mnis -e1 5 -e2 15
```

This produces a plot and plain text file (the locations of both are printed to the screen):



This plot shows you the maximum resolvable reflector height as previously discussed in *Roesler and Larson* (2018); that paper explains why the maximum resolvable reflector heights vary by azimuth. Three GPS frequencies are shown (L1, L2, L5). The largest RH retrievals (shown in blue) cluster at ~14 meters. This is well beyond the mean sea level value we retrieved earlier (~9.4 meters). If the tidal range is very very large, your L1 RH retrievals at large RH (i.e. low tides) would be aliased. In that case one must use higher sample rate data or exclude that frequency. In principle one could also write a separate piece of code to identify when the RH solutions are compromised by the receiver sampling rate.

Documentation:

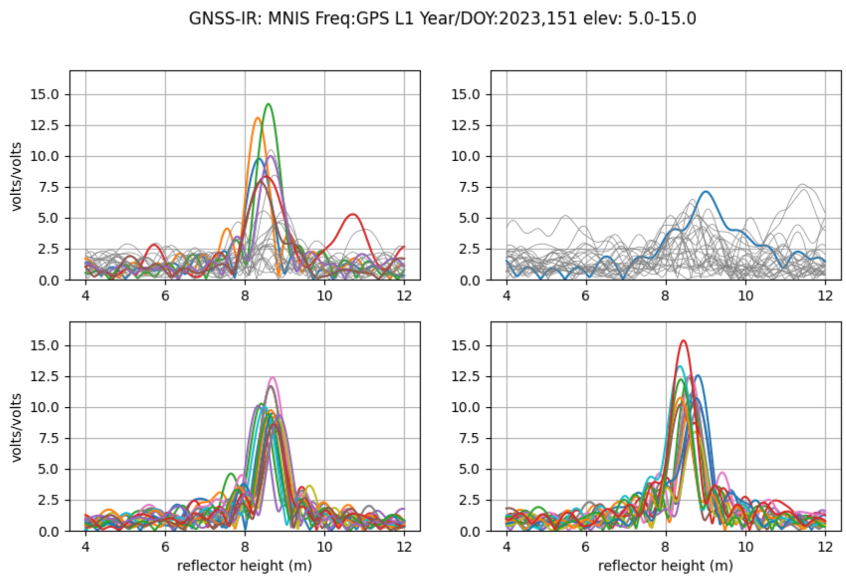
https://gnssrefl.readthedocs.io/en/latest/api/gnssrefl.max_resolve_RH_cl.html

3.5 Take a quick look at the SNR data

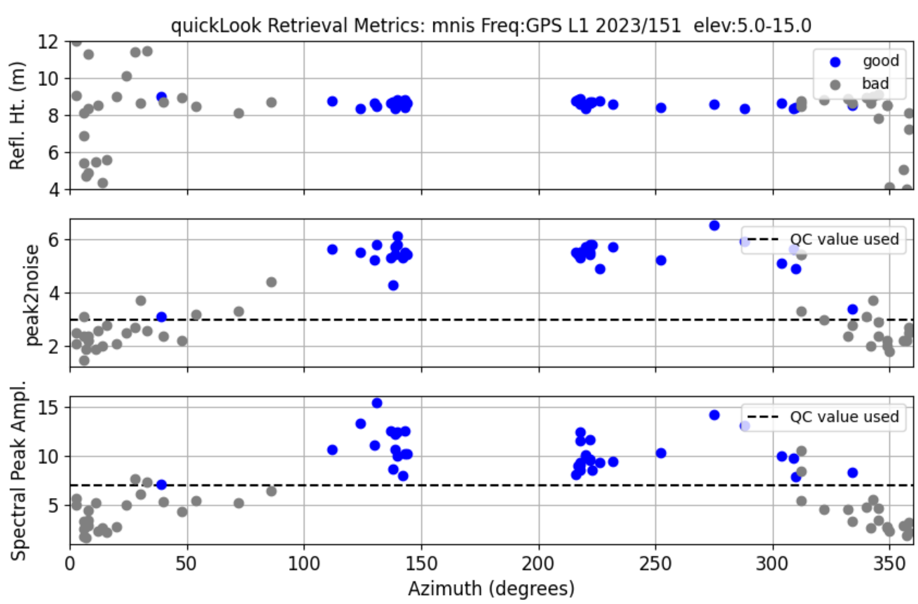
`quickLook` is a utility to confirm that the data from MNIS produce reflector height (RH) values consistent with expectations. The required inputs are simply the station name, year, and day of year. However, we do not want to rely on the defaults for either reflector height range (h1, h2) or elevation angles (e1, e2), so we enter them on the command line. As a starting place for water applications, I always use 5-15 for elevation. For the reflector height constraints, you should have a general idea of the average RH value (which we know to be 9.4 meters) and specify a RH range of about 4 meters on either side of the average. In this case I used 4 and 12.

```
quickLook mnis 2023 151 -h1 4 -h2 12 -e1 5 -e2 15
```

The physical location of the png files created by this code are written to the screen. The first plot created is a compilation of periodograms for the L1 frequency. The four quadrants represent the northwest, northeast etc quadrants. The colored periodograms represent satellite arcs that pass quality control and the gray periodograms represent satellite arcs that failed. As expected, the good periodograms are clustered around a value near 9 meters on the x-axis.



The second plot summarizes the estimated reflector heights and plots them with respect to the azimuth of the satellite arc. This allows you to check that the azimuth choices were reasonably good and that quality control choices (peak to noise and spectral peak amplitude) are acceptable. Reflector heights at good azimuths (shown in blue) are mostly clustered from ~120 to ~300 degrees. Note that there are very few arcs between 250 and 300 degrees.



For additional information, including how to look at different frequencies and how to choose reflector height limits, see the official documentation:

https://gnssrefl.readthedocs.io/en/latest/api/gnssrefl.quickLook_cl.html

Please note that the user sets the quality control values (peak to noise and spectral amplitude peak). This is discussed in the understanding section of the code:

https://gnssrefl.readthedocs.io/en/latest/pages/understand.html

3.6 Set the analysis strategy for daily reflector height estimation

The next step is to save the parameters you wish to use when estimating reflector heights. The only required input is the station name. For defaults, see the documentation:

https://gnssrefl.readthedocs.io/en/latest/api/gnssrefl.gnssir_input.html

The MNIS station operators are tracking multiple GNSS constellations and thus we set the -allfreq flag to true. The ampl setting is for the peak value of the periodogram.

```
gnssir_input mnis -h1 4 -h2 12 -e1 5 -e2 15 -allfreq T -azlist2 120 240 -ampl
5 -delTmax 45
```

Most of these inputs are similar to those we used for *quickLook*. delTmax is the maximum arc length in minutes. Some satellites have very long arcs (i.e. they span a long time period). This is acceptable for snow and soil moisture applications. For a dynamic reflector like the ocean, we have opted to limit satellite arcs to 45 minutes.

The instructions for the gnssir analysis are stored in $REFL_CODE/input/mnis.json. For docker users, this file can be found at /etc/gnssrefl/refl_code/input/mnis.json

3.6 Estimate reflector heights for each day of data you translated

gnssir estimates reflector heights for one or more days. It requires inputs of station name, year, and at least one day of year. Similar to *rinex2snr*, an optional setting allows you to analyze all the data for the two-week period by using the -doy_end command line setting.

```
gnssir mnis 2023 137 -doy_end 151
```

gnssir produces no plots unless you request it (-plt T). As it produces one plot per frequency per day, it is never advised when analyzing multiple days of data. The advantage of the plot is that it allows you see both the periodogram and the SNR data (with direct signal removed). The reflector height results for each day in this example are stored in $REFL_CODE/2023/results/mnis/ddd.txt where ddd is a three-character day of year. More information is provided upon request, -screenstats T

Documentation for *gnssir:*

https://gnssrefl.readthedocs.io/en/latest/api/gnssrefl.gnssir_cl.html

3.7 Compute water levels from reflector heights

The final step is to combine the results from daily reflector height analysis into a longer time series. The `subdaily` module does this and applies more advanced models. The only required inputs are station name and year. If you have multiple years, this can be accommodated with an optional year_end input.

For our test case:

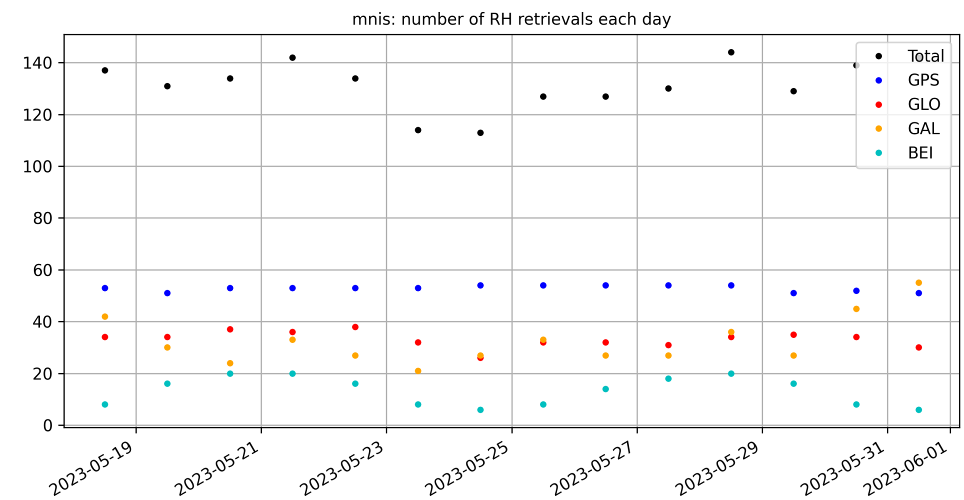
```
subdaily mnis 2023
```

This module produces plots which are meant to provide the user with context about the quality of the retrievals. The user can turn these plots off at the command line. It also writes out plain text files at various stages of processing. The files have self-documenting headers and the user should use those to determine which column has the values you want. In general, the files are updated with new columns as the corrections are applied. This is to allow the user to compare the effects of different corrections.

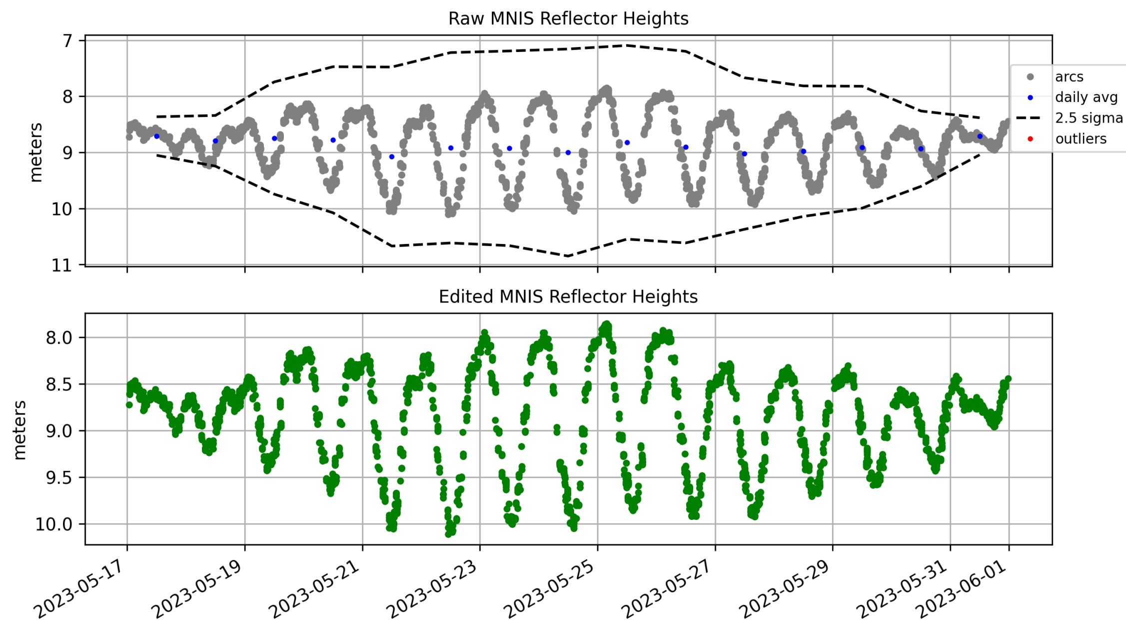
The subdaily code has two parts – and in this supplementary documentation we have labeled them as such. The first part consolidates all the reflector height estimates, computes some statistics, provides context on the quality of the retrieval and removes large outliers. Depending on the simplicity of the site (river, lake, ocean), the user may at this point wish to use this output as is. That can be done by setting -rhdot F.

The second part of the code is meant for users who have significant tidal signatures at their sites. This means the "H dot" correction (Larson et al. 2013) must be calculated and removed. It also removes an inter-frequency bias so that all measurements are defined with respect to the GPS L1 frequency. Now that gross outliers have been removed, more stringent outlier criteria are applied, i.e. three-standard deviation outliers (with respect to a spline) are removed. In the last step a new spline is fit to the improved data and this spline can be output at any interval of the user's choosing. The number of knots per day used in the spline can also be defined by the user. *The number of knots should be reduced for lakes and rivers.*

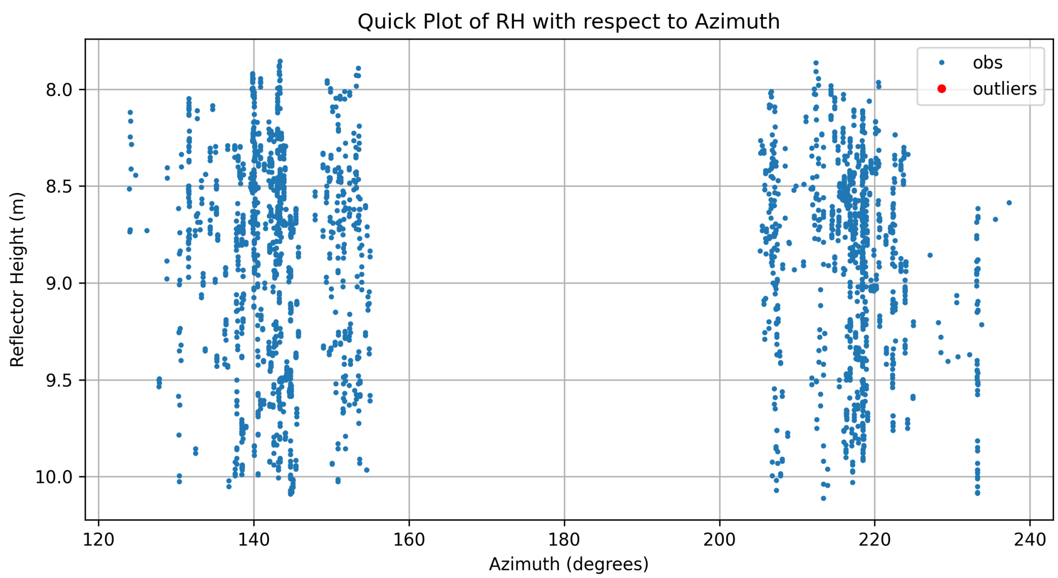
Once the user is satisfied with the analysis strategy, they can also look at using shorter satellite arcs. This can be done most easily by setting an elevation angle list of overlapping elevation angle ranges.



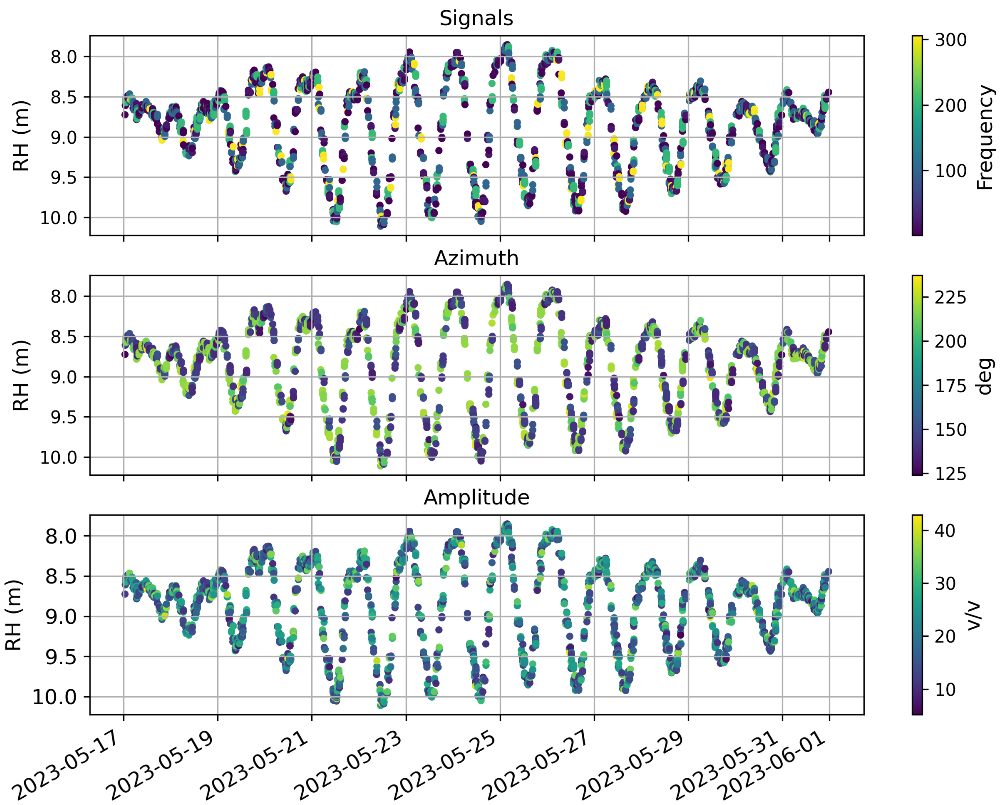
Section 1 Feedback on the number of GNSS signals that have been extracted from the analysis. This will depend on the choices made by the station operator, the number of azimuths, the number of elevation angle bins, the number of available frequencies (e.g. Glonass only has two frequencies while GPS has three), etc.



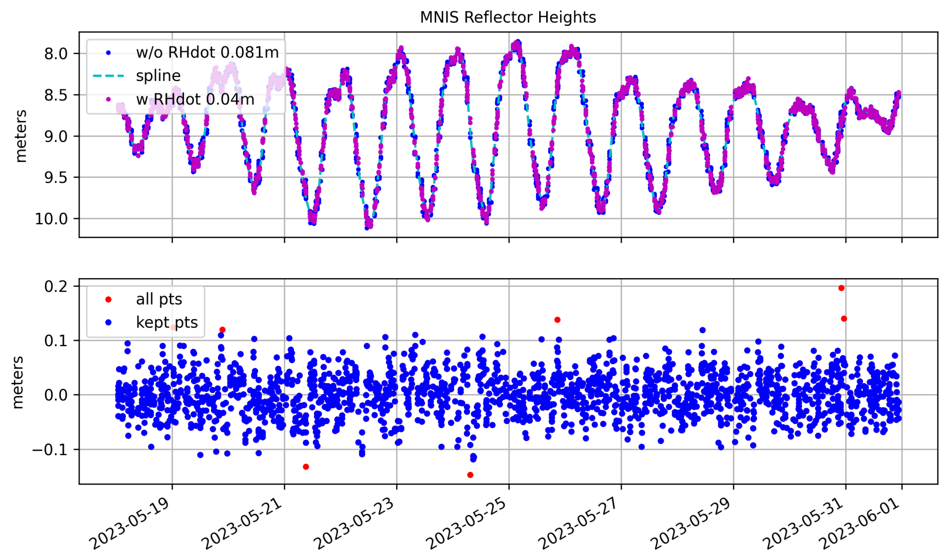
Section 1 Top: raw reflector heights; Below: reflector heights with large outliers removed. In this case the code detected no large outliers.



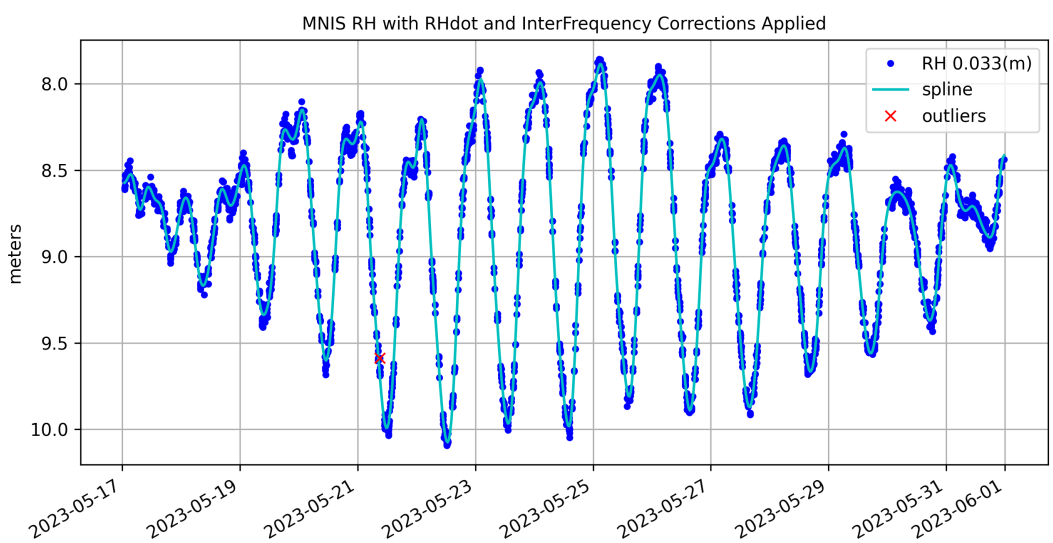
Section 1 Summary of estimated reflector heights with respect to azimuth.



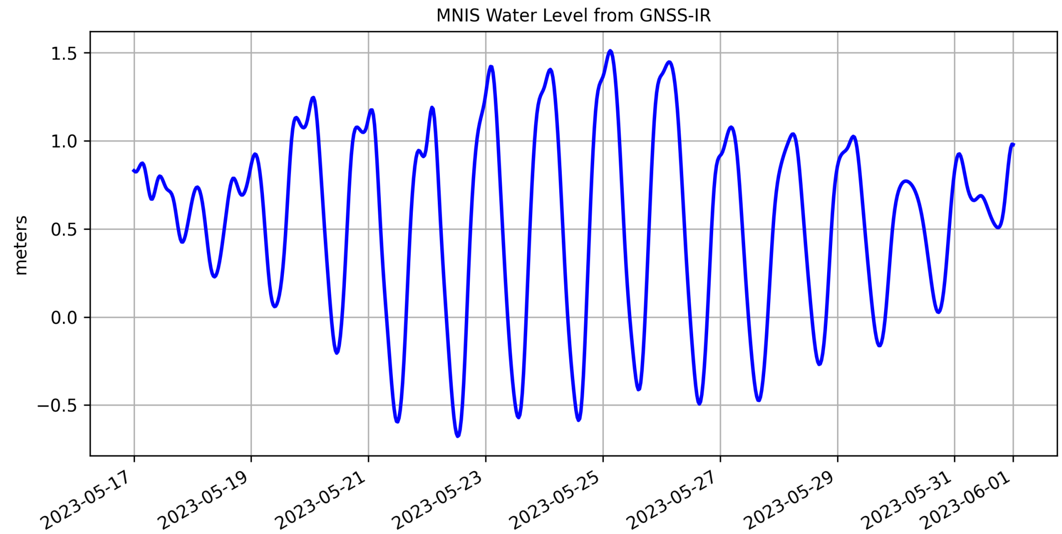
Section 1 Summary of reflector height estimates with respect to constellation frequency (top), azimuth (middle), and spectral amplitude (bottom). Frequencies defined as GPS: 0-99; Glonass: 101-199; Galileo: 201-299, Beidou: 301-399.



Section 2 Top: Reflector height results with and without the RHdot correction. Bottom: residuals to a spline fit for the solution with a RHdot correction. Three standard deviation outliers are shown in red. In both cases the RMS values provided are with respect to a spline fit – and indicate that the standard deviation with respect to that fit was 0.081 meters before the RH dot correction was applied and 0.040 meters afterwards. These RMS values are also printed to the screen.



Section 2 Final series of reflector heights, now with an inter-frequency bias removed. The stated RMS of 0.033 meters is with respect to a spline with 8 knots per day (default).



Section 2 Evenly spaced output from a spline fit to final reflector height solutions. Water level is defined as Hortho (orthometric height) minus reflector height. By default the orthometric height will be based on EGM96 and the given ellipsoidal height. Users can override this value using a more accurate orthometric height and storing it in the gnssir analysis json file.

Plots and text output files are stored in the $REFL_CODE/Files/mnis directory

The physical location of these files on your local machine will be related to where you are running the docker. For example, in the terminal window where you are running the docker type:

```
ls /etc/gnssrefl/refl_code/Files/mnis
```

The location of text files and figure files created from all modules will be printed to the screen.

Documentation:

https://gnssrefl.readthedocs.io/en/latest/api/gnssrefl.subdaily_cl.html

Tide gauge

There is a tide gauge near this GNSS receiver. See the following link for further information:

https://www.qld.gov.au/environment/coasts-waterways/beach/storm/storm-sites/mornington-island

Tide gauge data from the past seven days are easily available:

https://www.data.qld.gov.au/dataset/coastal-data-system-near-real-time-storm-tide-data

For access to tide gauge data from periods other than the last seven days, users must contact the State of Queensland directly.

Acknowledgements
The RINEX files analyzed for this paper were made available by the Geoscience Australia GNSS Data Centre and translated to RINEX 2.11 format using the CRX2RNX (Hatanaka, 2008) and gfzrnx software (Nischan, 2016).

Multi-GNSS orbits were provided by the Deutsches GeoForschungsZentrum (GFZ-orbits, 2023). The MNIS station was installed and is operated by the State of Queensland, Department of Environment and Science.

Updating the code

If you want to use the software at some later date, it is best practice to update your docker:

```
docker pull ghcr.io/kristinemlarson/gnssrefl:latest
```

And then instead of a specific version, you will want to run that latest docker image

```
docker run -it -v $(pwd)/refl_code:/etc/gnssrefl/refl_code/ --name gnssrefl
ghcr.io/kristinemlarson/gnssrefl:latest /bin/bash
```

References

GFZ-orbits (2023) Analysis center of the Multi-GNSS Experiment. https://www.gfz-potsdam.de/en/section/space-geodetic-techniques/projects/mgex/, accessed 2023 July 26.

Hatanaka Y (2008) A compression format and tools for GNSS observation data, *Bull Geograph Survey Inst* Vol. 55(March).

Larson KM, Ray RD, Nievinski FG, Freymueller JT (2013) The Accidental Tide Gauge: A GPS Reflections Case Study from Kachemak Bay, Alaska, doi:10.1109/LGRS.2012.2236075.

Nischan T (2016) GFZRNX – RINEX GNSS Data Conversion and Manipulation Toolbox, GFZ Data Services, https://doi.org/10.5880/GFZ.1.1.2016.002

Roesler CJ, Larson KM (2018) Software Tools for GNSS Interferometric Reflectometry, *GPS Solutions*, *Vol 22:80*, doi:10.1007/s10291-018-0744-8