

running the gnssrefl code

Kristine M. Larson

<https://kristinelarson.net>

<https://github.com/kristinemlarson>

kristinem.larson@gmail.com

Where did gnssrefl come from?

- PBO H₂O was a GPS research group based in Boulder that made environmental products (soil moisture, snow, and vegetation) from data collected by the Plate Boundary Observatory GPS network:
 - Fortran for everything computationally intensive
 - c-shell scripts to fetch files
 - lots and lots of plain text files
 - Matlab for environmental models
 - Web programmer (Evan Crouch) to make products available



PBO H₂O

Using GPS reflection data to study the water cycle

Where did gnssrefl come from?

- PBO H₂O ran until 2017
 - With NASA support, we gave the PBO H₂O code to JPL
 - My Fortran-based RINEX translation codes were/are public.
 - I took a very tiny part of the code - the reflector height periodogram code - translated it to Matlab for the cryosphere community.
 - Carolyn Roesler and I published Matlab code for reflection zones & Nyquist calculations.

GPS Solutions (2018) 22:80
<https://doi.org/10.1007/s10291-018-0744-8>

GPS TOOL BOX

Software tools for GNSS interferometric reflectometry (GNSS-IR)

Carolyn Roesler¹ · Kristine M. Larson¹ 

Why did I write gnssrefl?

- Effort to take the knowledge gained from PBO H2O (and tide gauge work done in parallel with Chalmers University and PSMSL) and provide it in an accessible way to the community.
- Open source python hosted by GitHub
- Fortran is embedded into the python for speed
- multi-GNSS by design (GPS, Glonass, Galileo, Beidou)
- no shell scripts
- reflection zones via web app and python

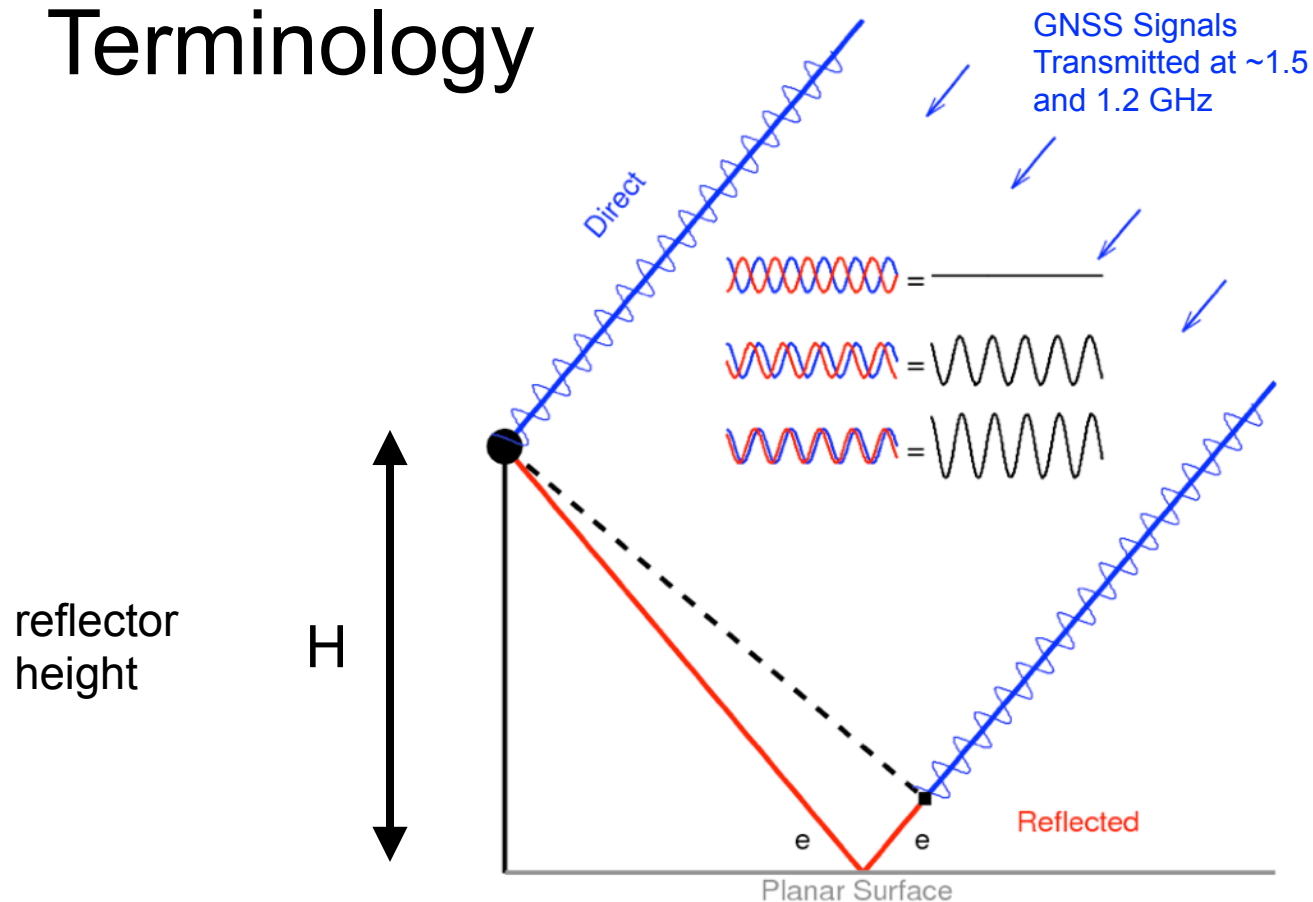
gnssrefl features

- queries GNSS archives and orbits for you
 - precise, rapid, and ultra multi-GNSS
 - more than 15 data archives
- Originally RINEX 2.11
 - RINEX 3
 - NMEA
- Soil moisture, snow accumulation, water levels.
- readthedocs documentation with use cases**
- Originally Linux only**
 - Dockers
 - Jupyter Notebooks (will not be discussed here)

** huge assistance from Tim Dittmann and Kelly Enloe

Felipe has gone through some of the theoretical background for GNSS-IR

Terminology

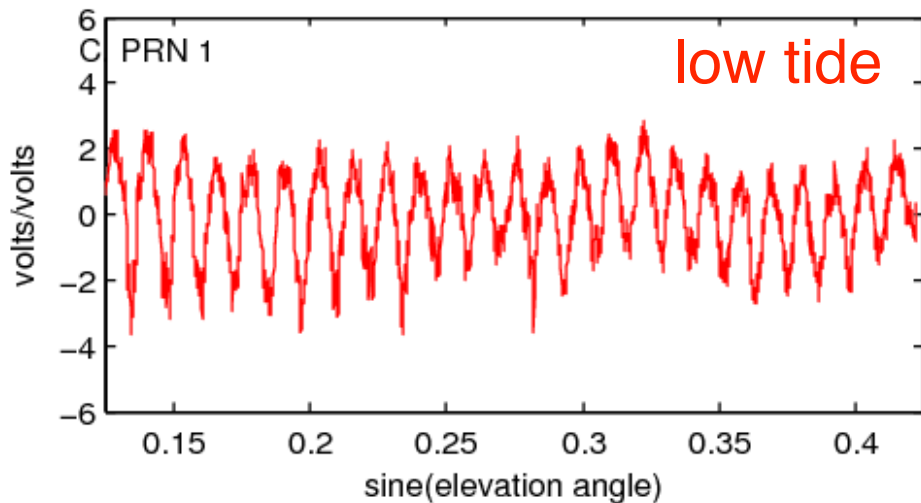
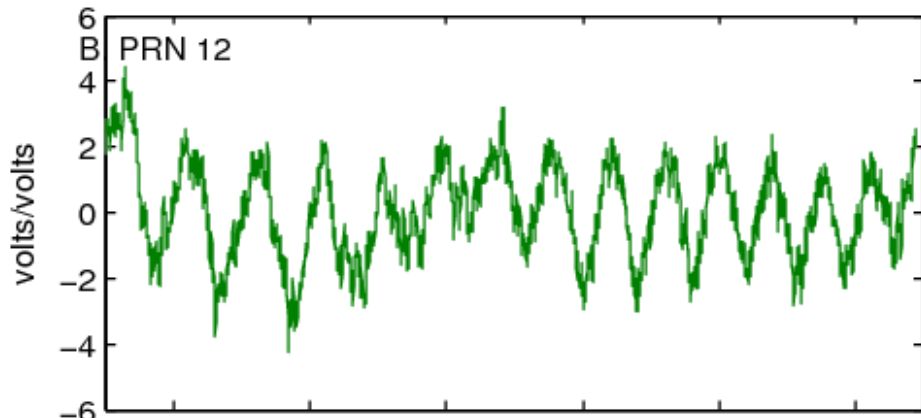
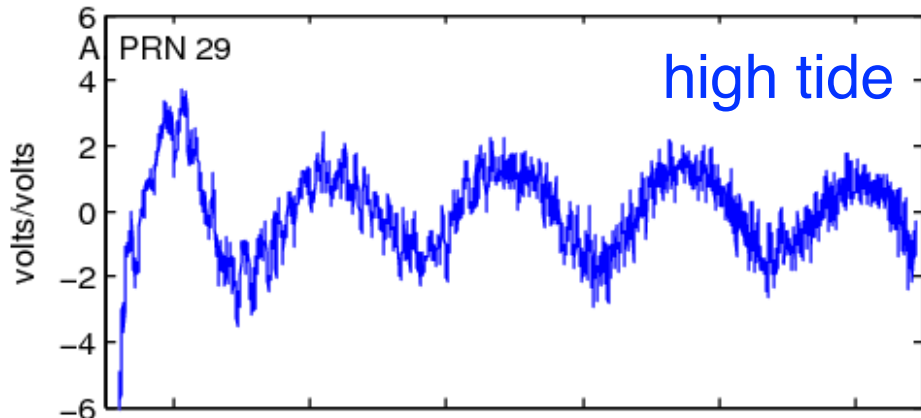


- The interference pattern has a frequency directly related to H , the distance between the reflecting surface and the antenna.
- The interference pattern can be retrieved from SNR data.
- Footprint of GNSS-IR method depends on H .
- Reflection surface also has an impact on the interference pattern, primarily in how strong the reflection is.

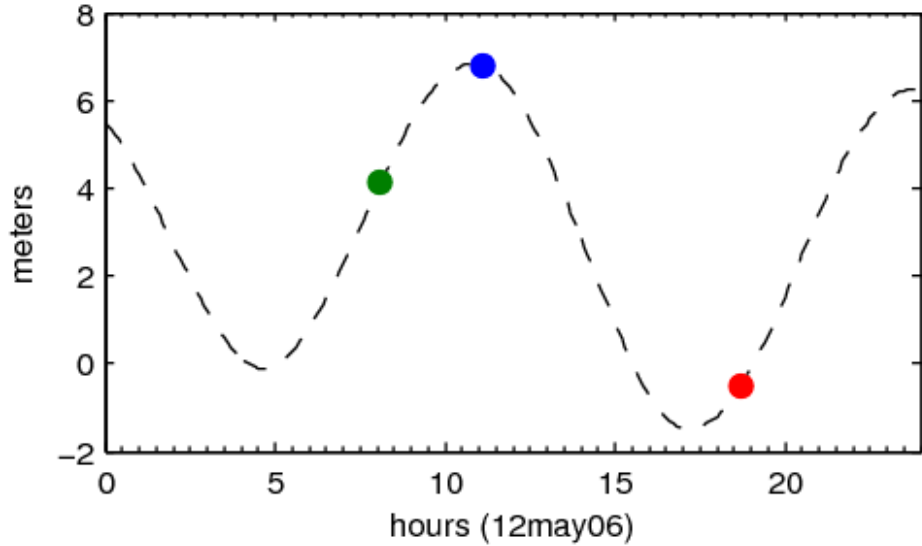
Visually what are we doing?



Raw SNR Data



Seldovia Tide Records



We use a periodogram of these SNR data to estimate H . That is why you will sometimes see me showing you periodograms,

Extract the dominant frequency from rising and setting arcs

- What is the best way to do that?
- Can you use all sites?
- Can you use the same data as geodesists and surveyors?
Data from low-cost navigation chips?
- What is the fastest way to do this?
- Which GNSS frequencies should be used?
- How should you pick your reflection mask?
- Quality control
- What do you do with the H estimates afterwards?
- Note: I am going to show you examples using the command line version; you can also use a python script to call this code.

Basic Steps

- Make sure that your site can be used for GNSS-IR
- You need GNSS data (SNR)
 - your own data or you can pick one of our supported archives
 - Must be compliant (header info, SNR data, follows community naming rules)
- You need orbits
 - GPS or multi GNSS
- Define the region where you think GNSS reflections will be useful.
- Isolate each rising and setting GNSS satellite arc in that region
- Find the frequency of each (useful) interference pattern - relate that to H.
- Quality Control: Make sure you trust those H values.
- Repeat, repeat, repeat
- Turn into environmental products (here we only discuss water)

Basic Steps

- Make sure that your site can be used for GNSS-IR
 - are you close enough to the water to be able to measure it?
 - is the sampling rate of the receiver high enough for the given water surface?
 - did the owners of the GNSS site collect the appropriate data?

If you can't answer yes to these questions, you really should stop.

Example from South Africa

- Are you close enough to the water to be able to measure it?
- it depends
- Photographs are great - but don't use photographs to answer this question!



GNSS-IR Reflection Zone Mapping

Station Location

Input 4 character station name: (uses Nevada Reno database)

OR

Input coordinates: Lat. (deg) Lon. (deg) EllipseHt (m)

Reflection Height (meters)

Use Mean Sea Level Set Reflector Ht. Value

Frequency

L1 L2 L5

Compute Nyquist (this takes a few seconds)

no yes rcvr sample rate (sec)

Elevation Angles (degrees)

5,10,15 5,10,15,20,25 5,7,10,12 5,6,7 10,15,20 5,7,10

Azimuth Angles (degrees)

If needed, you can cross North, e.g. -90 to 90

Start (deg) End (deg)

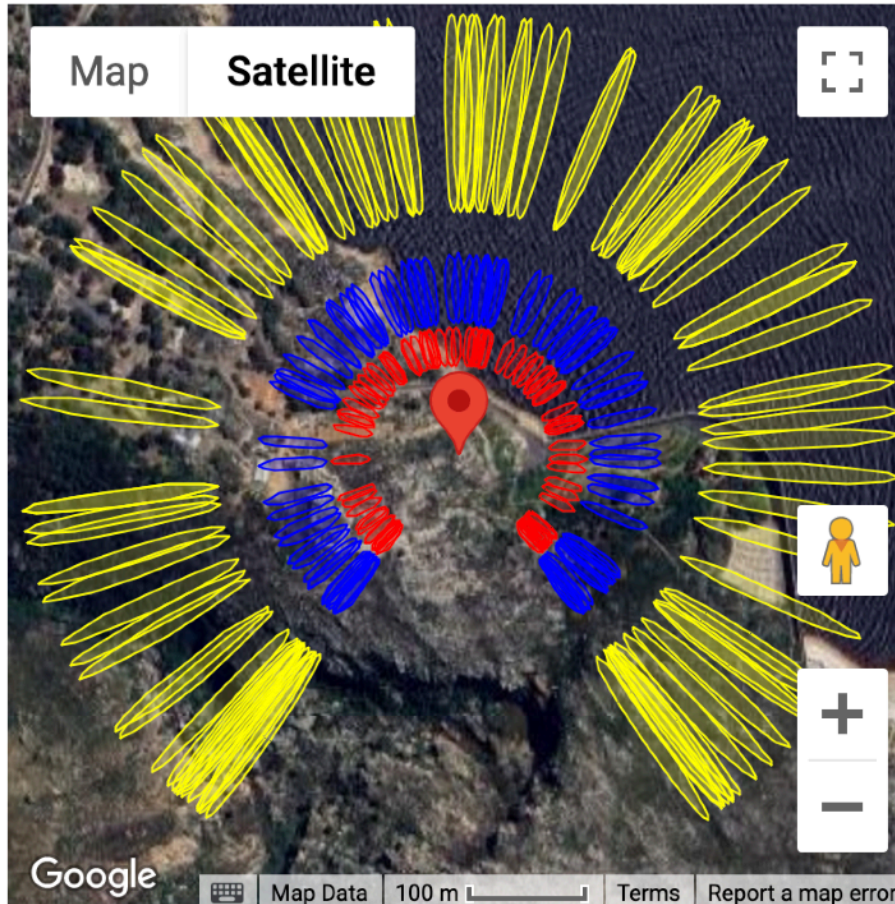
Constellation

GPS Galileo Glonass Beidou(MEO)

<https://gnss-reflections.org/rzones>

That reservoir in South Africa is ~80 meters away from the GNSS station. This would normally mean you could not use it for GNSS-IR.

BUT, it is 20-30 meters above the height of the water surface (depending on how full the reservoir is). So don't assume. Calculate the reflection zones.

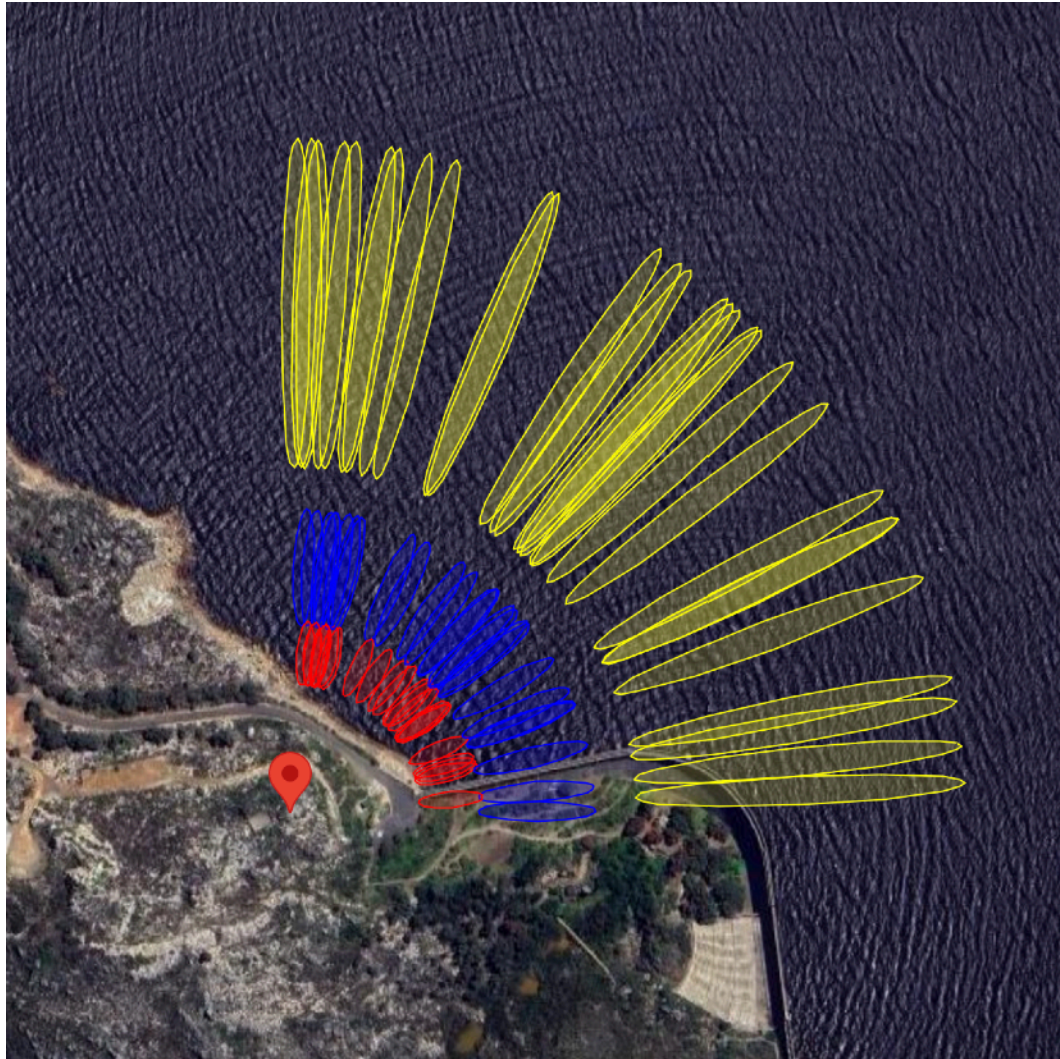


this is for
 $H = 25$ meters
L1 GPS

So looks pretty
good so far.

It might work!

This is for L2 GPS and 25 meters



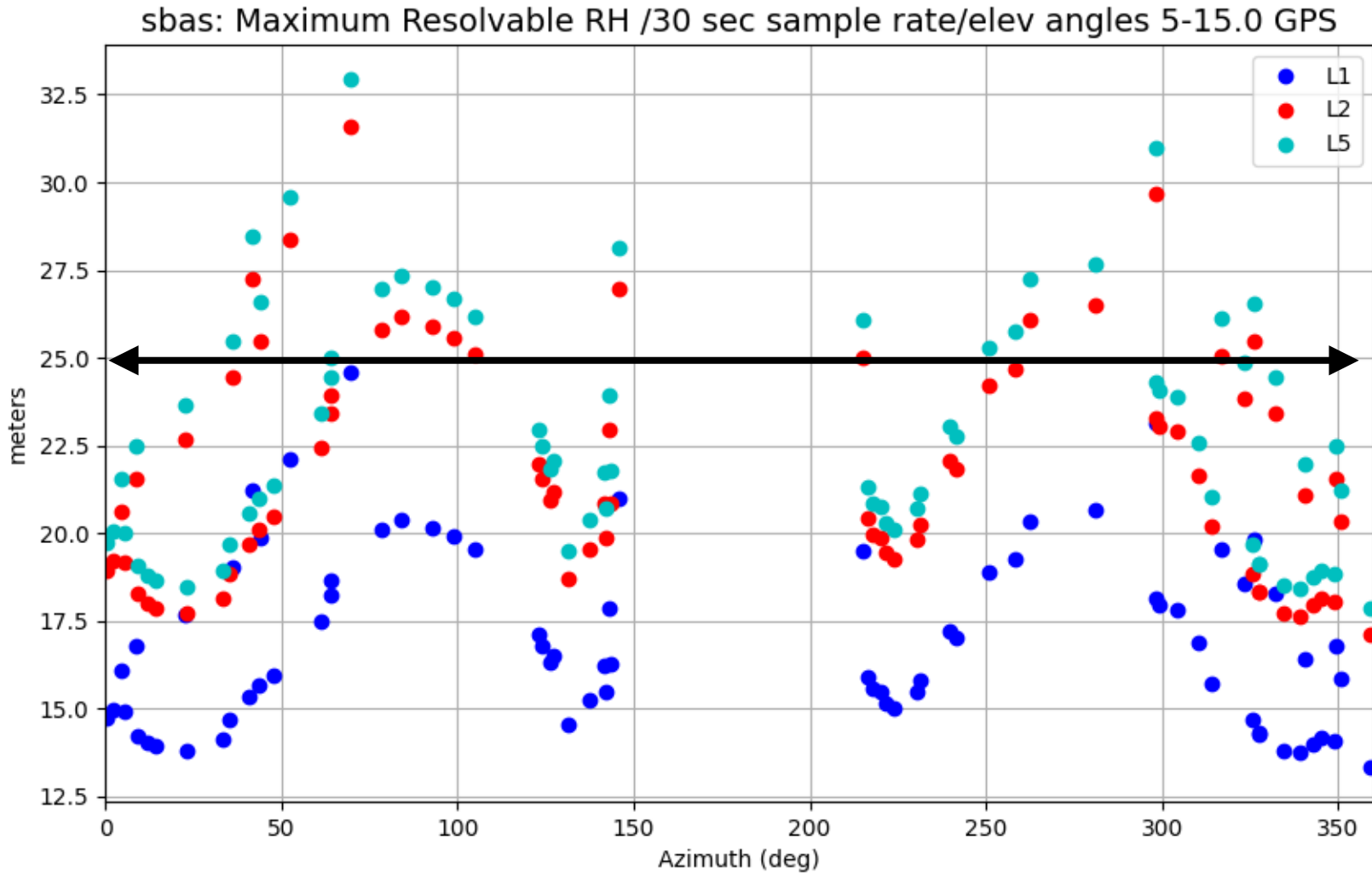
so azimuth mask
should be further
modified

- is the sampling rate of the receiver high enough for the given water surface?
- did the owners of the GNSS site collect the appropriate data?
- Maybe and No

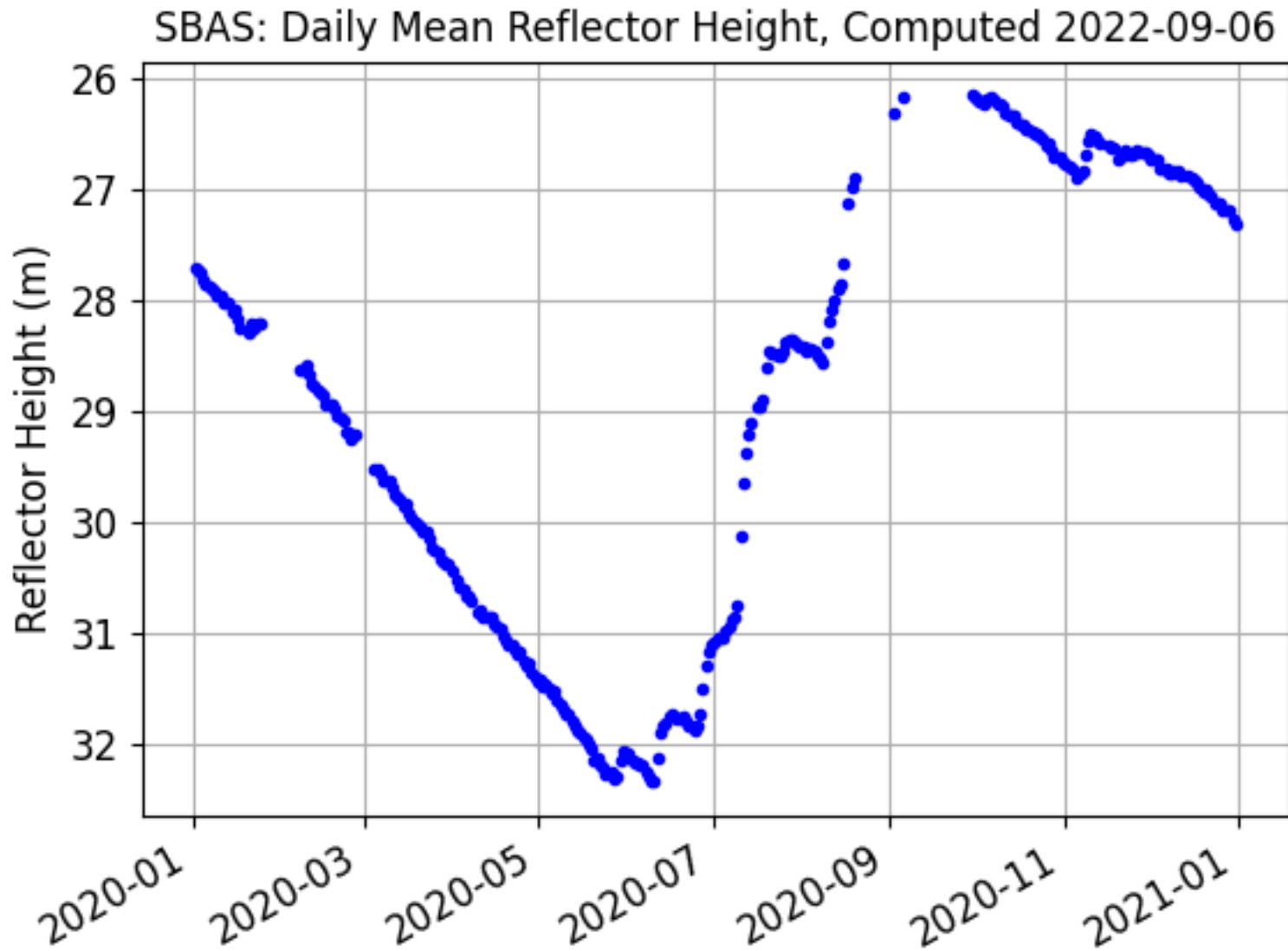
1-second data are collected for this site - for surveyors I assume - but there are no SNR data in that file. And I think it is L1 only.

The 30-second data preferred by geodesists are available with SNR data. But they aren't the high-quality signals.

30 seconds will not work. But these scale, so if you could have 15 second files with SNR data, it should mostly work.

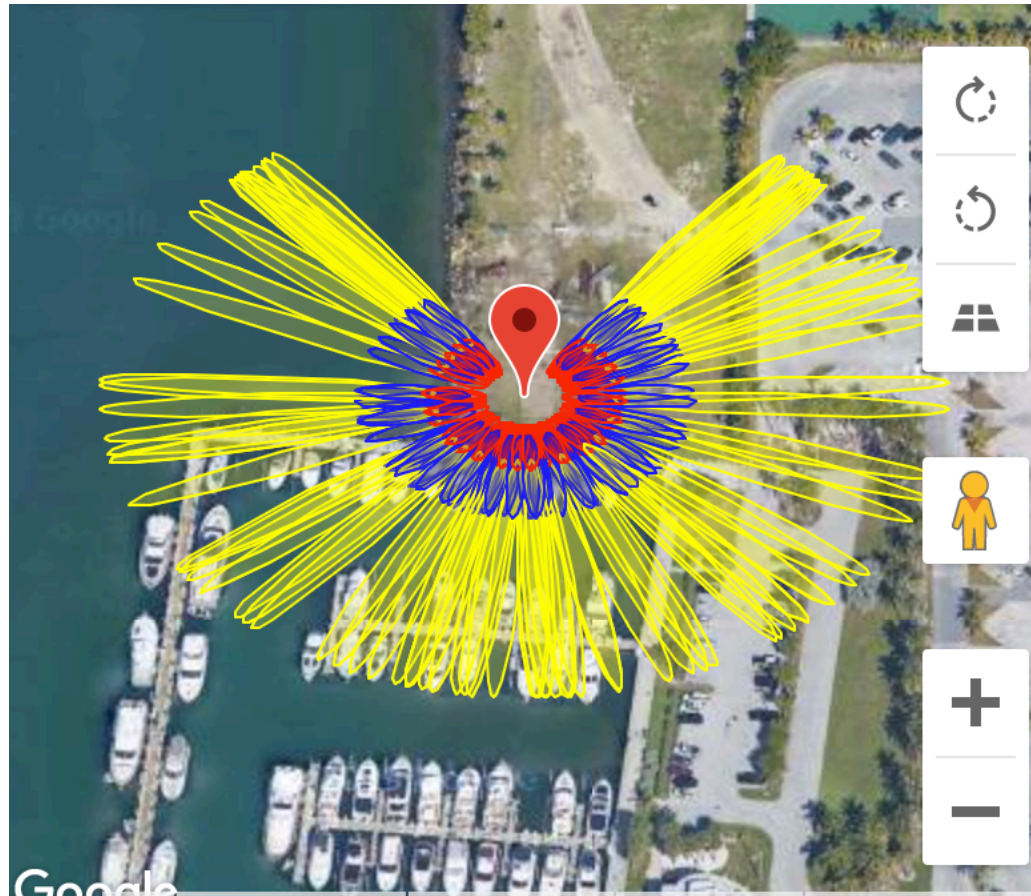


But using 15 second L1 and L2C you can do pretty well.



Example from Florida



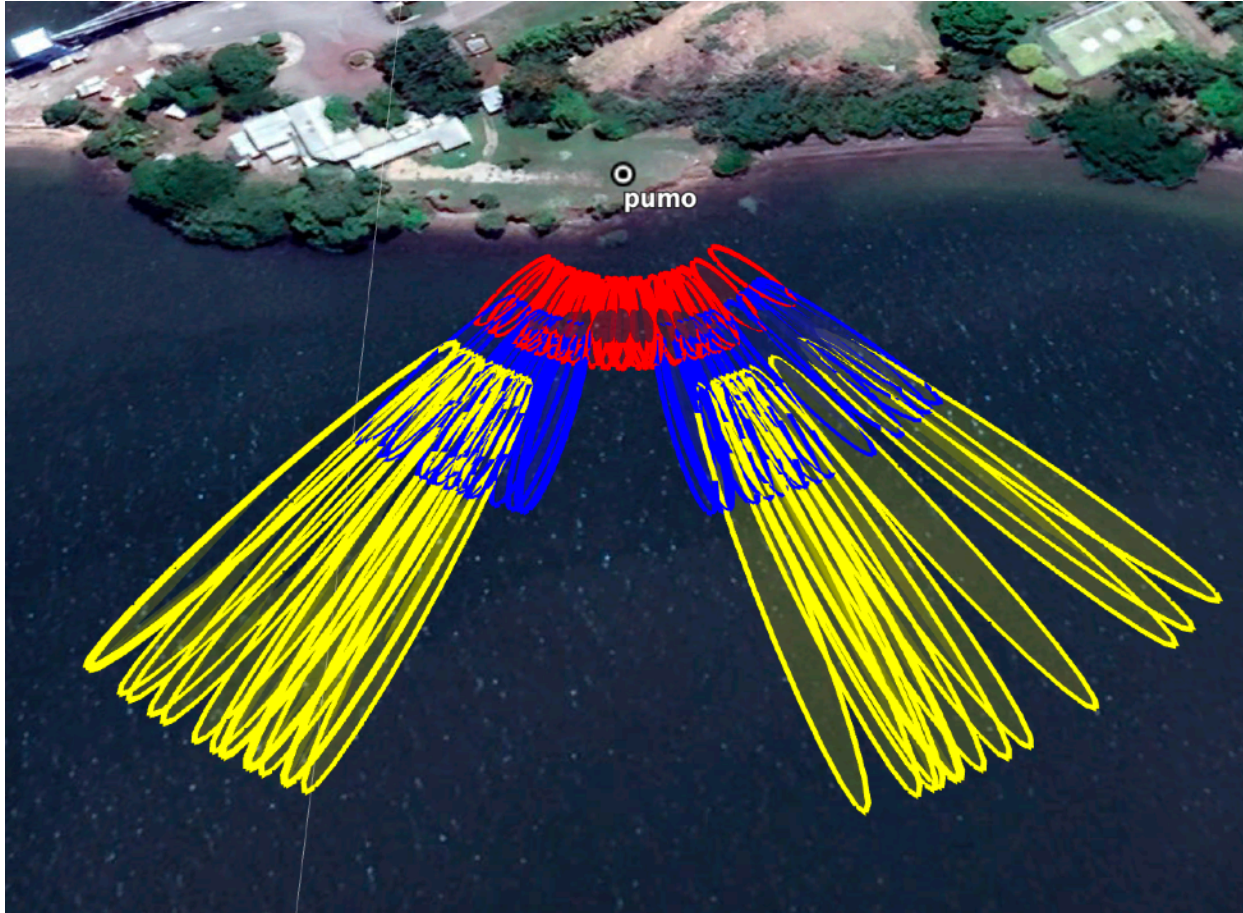


Ellipsoidal Height(m): -24.5
Reflection Ht. (m) : 3.45
Elevation Angles (deg) : 5,10,15
Azimuth Angles (deg) : 0 to 360

Example for Costa Rica, PUMO



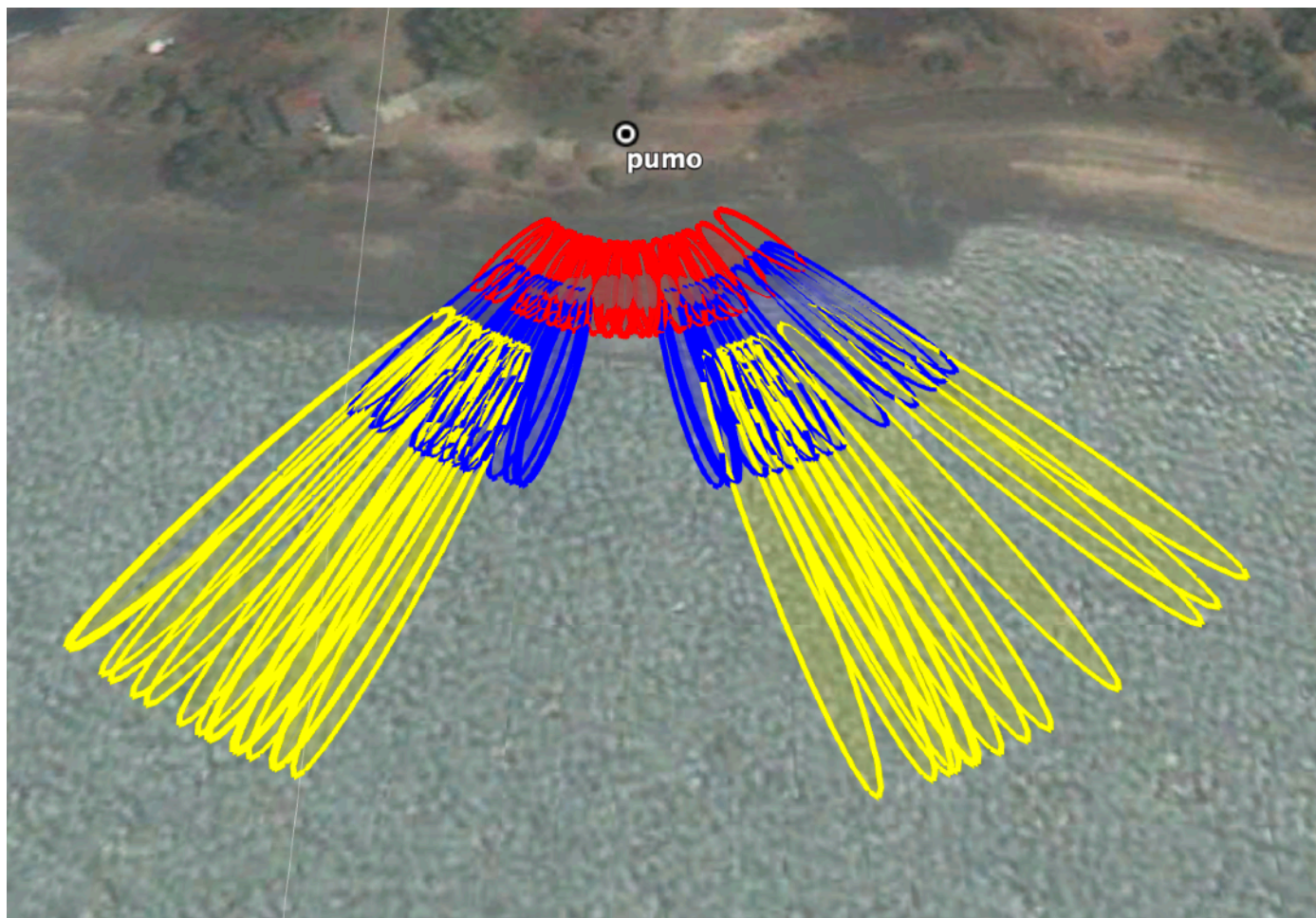
At first ... it looks promising



ellipses represent 5, 10, 15 degree elevation angles

[Kristine M. Larson](#), 2024 GNSS-IR Short Course

But then ... reality sets in



Basic Steps

- You need GNSS observation data.
 - your own data or you can pick one of our supported archives
 - Must be compliant (**header info**, **SNR data**, follows community naming rules)
- You need orbits
 - GPS or multi-GNSS

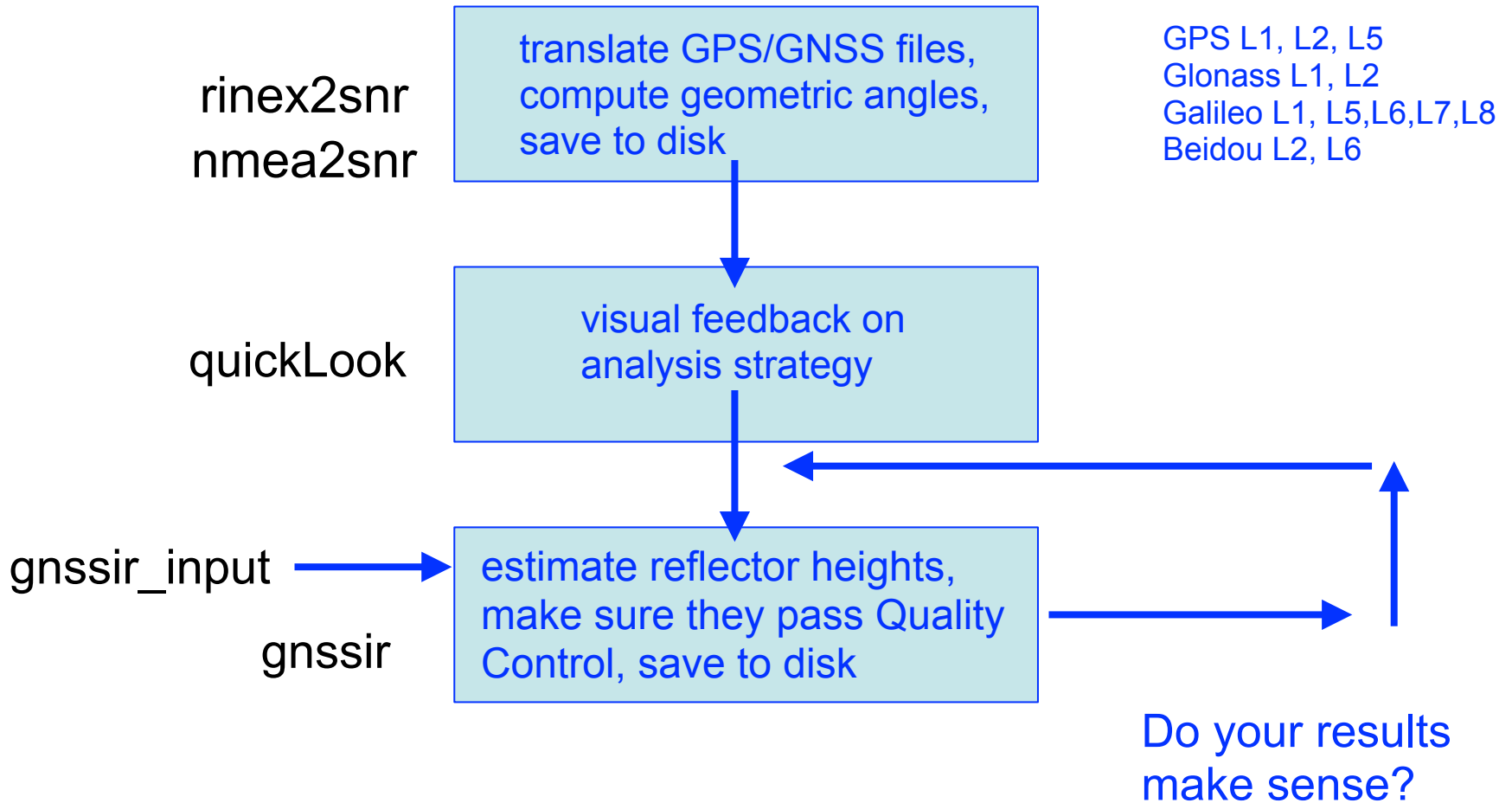
Basic Steps

- YOU Define the region where you think GNSS reflections will be useful.
- YOU decide which frequencies to use and QC values.
- THE CODE Isolates each rising and setting GNSS satellite arc in that region.
- THE CODE Finds the frequency of each (useful) interference pattern - relates that to H based on your choices.

Basic Steps

- Make sure you trust those H values.
- Repeat, repeat, repeat

gnssrefl software building blocks



gnssrefl

rinex2snr

translate GPS/GNSS files,
compute geometric angles,
save to disk



gnssir

estimate reflector heights,
make sure they pass Quality
Control, save to disk



snowdepth
daily_avg

vwc
vwc_input, daily_avg

subdaily
daily_avg

this course

gnsrefl uses environment variables

If you are using the docker or the Jupyter notebooks, you don't have to set the environment variables. They ARE still there - you just don't need to set them.

- **Settings:**
 - With github or pypi, these environment variables should be stored in your .bashrc
 - Using the docker, these are defined for you.
 - Using the Jupyter notebooks, these are defined for you.
- **EXE** - this is where support executables, e.g. CRX2RNX and gfzrnx, are stored.
- **REFL_CODE** - main directory for everything related to gnsrefl.
- **ORBITS** - where the GNSS orbits will be stored. This can be the same as REFL_CODE.

GPS/GNSS data

- We use the community standards, RINEX 2.11, RINEX 3, NMEA
- Your file **MUST** have SNR data in it. Why?
- Your file **MUST** have coordinates in the header. Why?
- RINEX 2.11: lowercase filenames and naming conventions used by geodetic archives (e.g. station names are four characters).
 - Example, station p041 day of year 30 and year 2020 is called p0410300.20o
 - If the file is Hatanaka compressed, p0410300.20d
- RINEX 3: filenames are very complicated. >>> Read the online documentation for details
- NMEA: ssss1520.23.A for station ssss, day of year 152 and year 2023
- Can you use your own data? Yes. But you still need to follow the rules.
- We suggest: practice with our highlighted use cases first. Then try new data.

step 1: rinex2snr

picks up and translates RINEX files, picks up orbits, computes geometric angles, and extracts SNR data

- Main input - GPS/GNSS RINEX files
 - your own RINEX files
 - or the code will download them from a long list of international geodetic archives
- The output is called a SNR file. Unfortunately these files have non-intuitive file endings.
- Some defaults
 - Before 17 May 2021, GPS orbits. After this time, GFZ multi-GNSS rapid orbits
 - All elevation angles < 30 degrees
 - If you don't specify an archive, it tries unavco, sopac, and sonel
- `rinex2snr -h` and the online GitHub documentation for more information

Examples

```
rinex2snr p041 2020 132
```

would produce this file:

```
$REFL_CODE/2020/snr/p041/p0411320.20.snr66
```

```
rinex2snr p041 2020 132 -archive sopac
```

produces the same file, but you wouldn't have to set up an Earthscope account

```
rinex2snr mchl00aus 2024 1 -archive cddis
```

produces SNR data for a RINEX 3 data file stored at CDDIS

Examples for Local Data

```
rinex2snr p041 2020 132 -nolook T
```

Assumes the RINEX 2.11 file is stored locally**

```
nmea2snr ABCD 2020 132
```

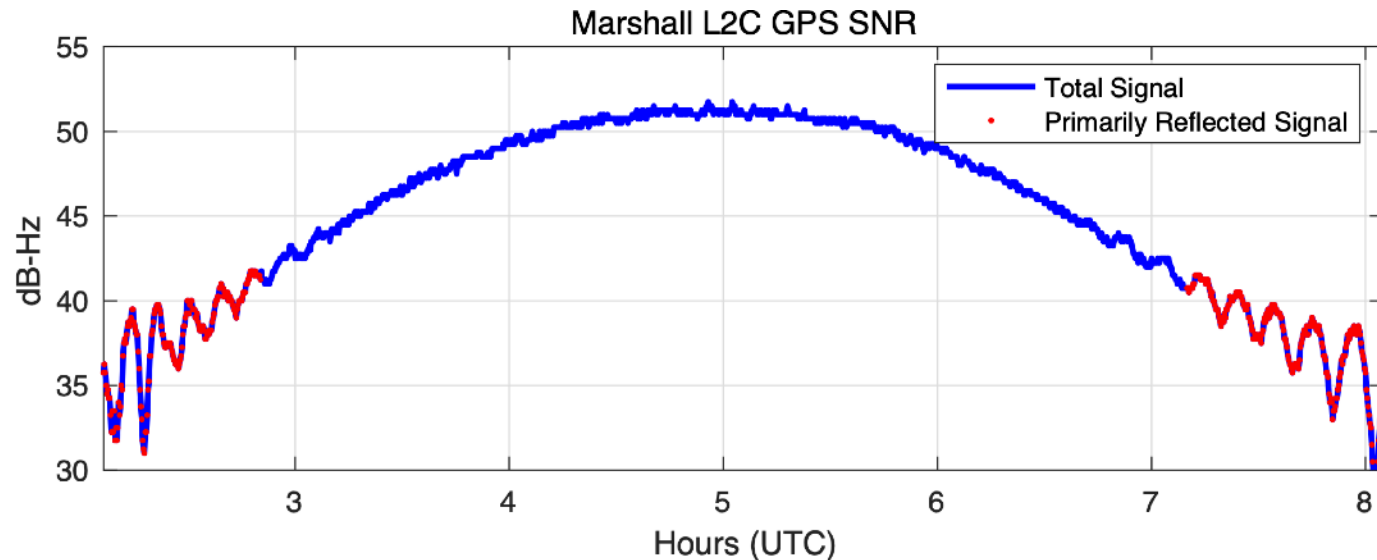
Translates a NMEA file stored locally. nolook is not required as there are no archives for NMEA data, so by definition you will never be “looking”**

** see documentation for allowed locations of NMEA files

What is stored in a SNR file?

This is for one satellite pass

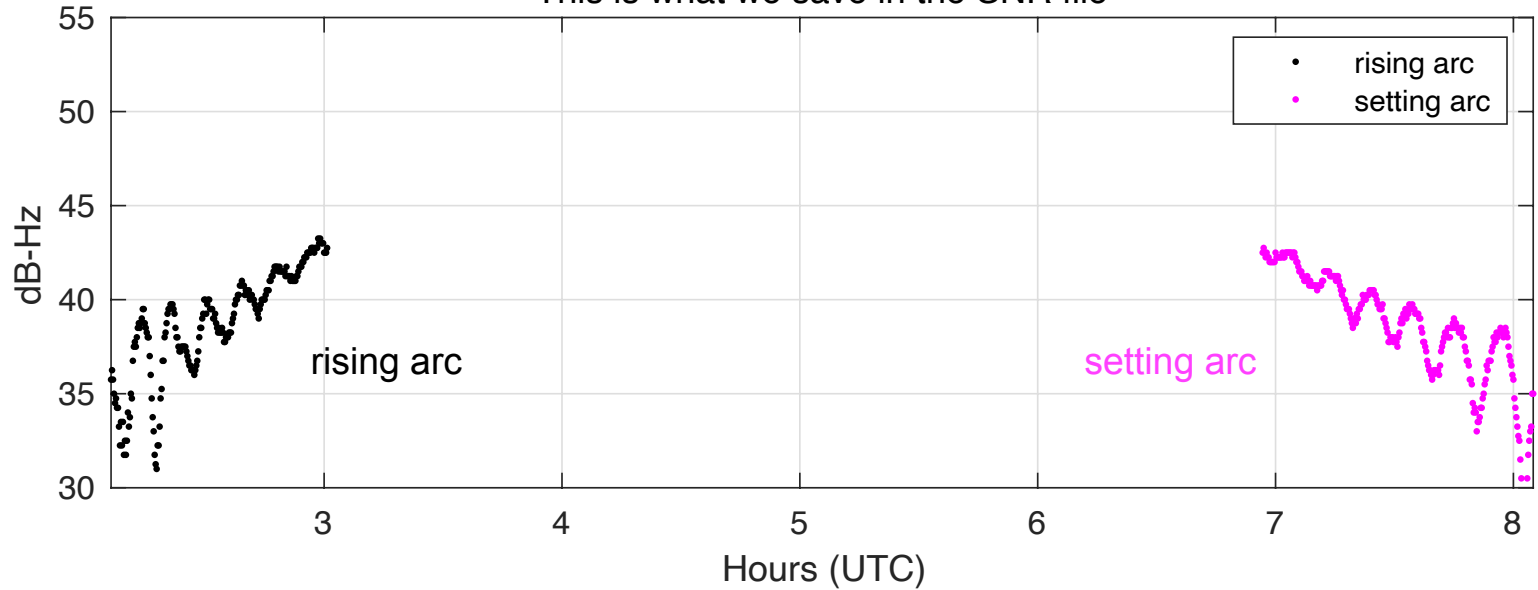
Satellite Number
Time
Elevation Angle
Azimuth Angle
SNR data



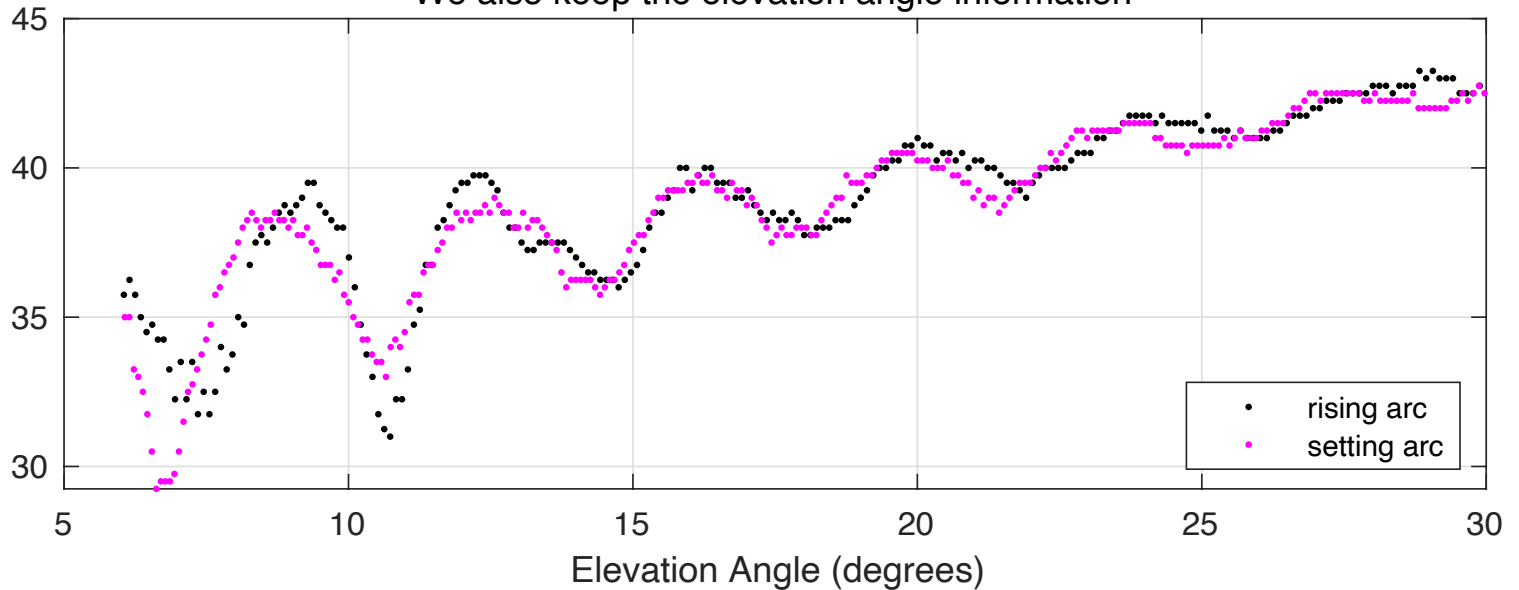
This is what SNR data look like in the time domain

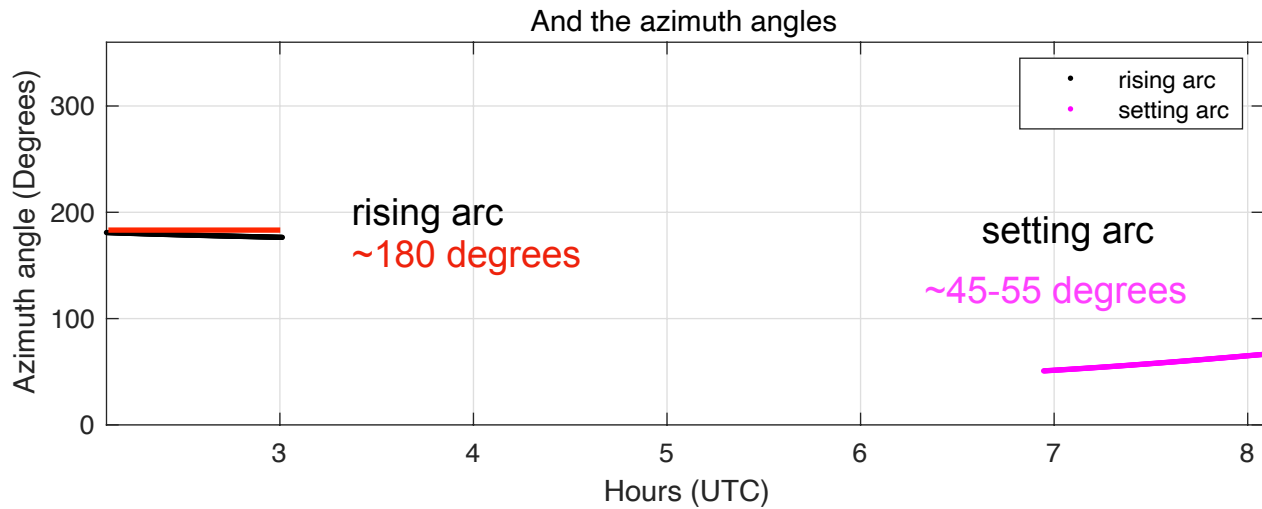
We really only want the red part - the low elevation angles -where we can see reflected signals. Fitting out that overall behavior is called “removing the direct signal.” We use a lower order polynomial. You can control that polynomial order.

This is what we save in the SNR file

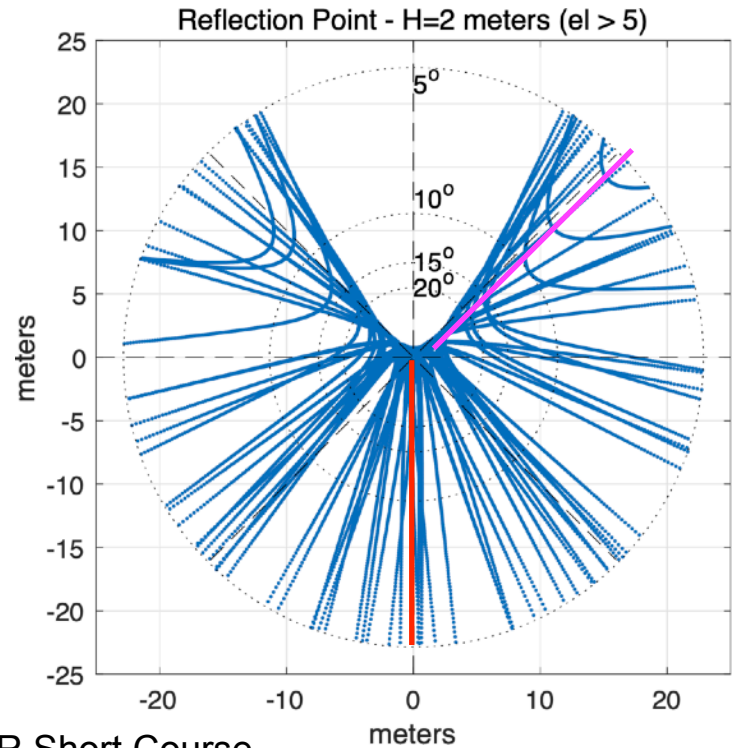


We also keep the elevation angle information





You can have no more than two rising arcs and two setting arcs per day (for some satellites and locations, there will be only one rising and one setting arc).



Where do files live?

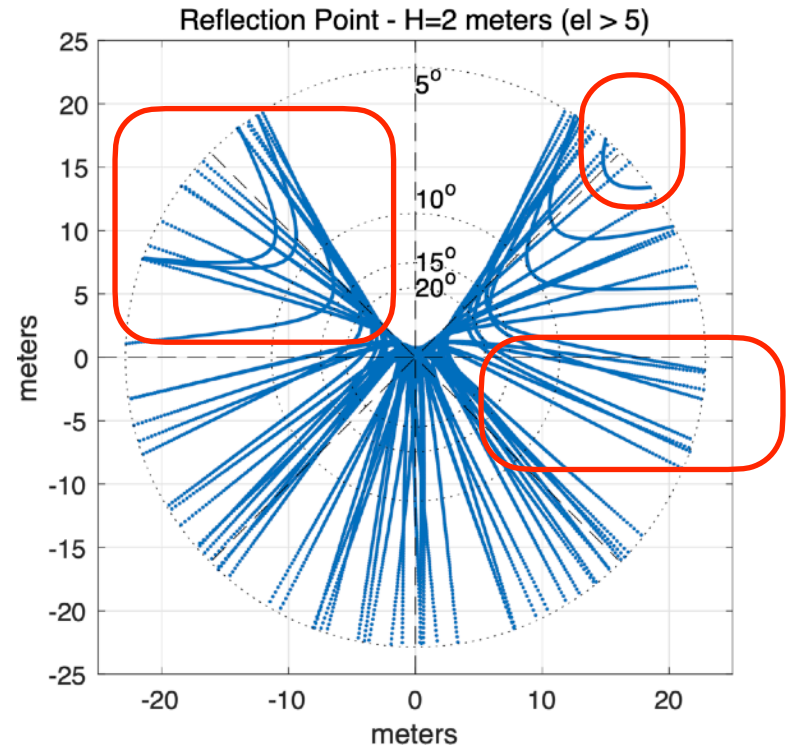
File structure for station abcd in the year YYYY (last two characters YY), doy DDD:

- REFL_CODE/input/abcd.json - instructions for gnssir analysis, refraction files
- REFL_CODE/YYYY/snr/abcd/abcdDDD0.YY.snr66 - SNR files
- REFL_CODE/YYYY/results/abcd/DDD.txt Lomb Scargle analysis
- REFL_CODE/YYYY/phase/abcd/DDD.txt phase analysis
- REFL_CODE/Files/ - various output files and plots will be placed here
- ORBITS/YYYY/nav/autoDDD0.YYn - GPS broadcast orbit file
- ORBITS/YYYY/sp3/ - sp3 files of orbits - these use names from the archives.

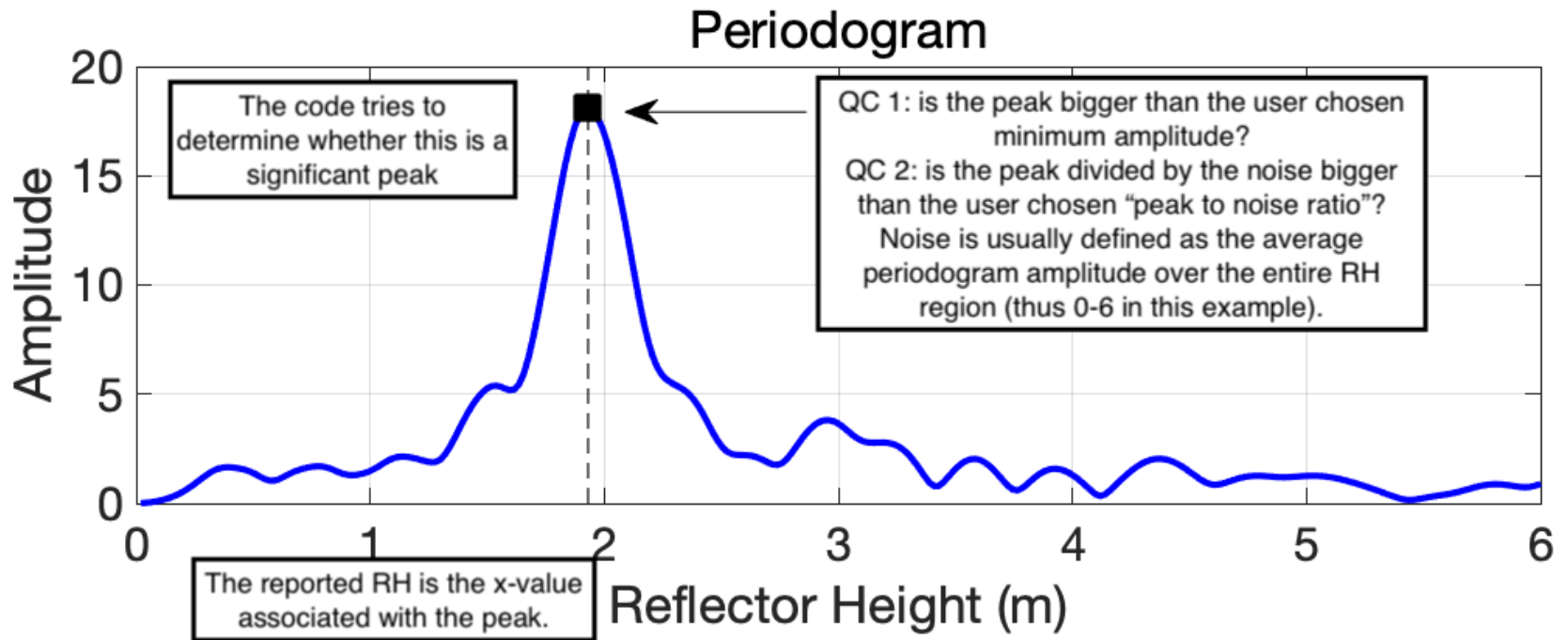
REFL_CODE and ORBITS can point to the same place

Step 2. quickLook

- For each geographic quadrant (northwest, southeast, etc), it checks all satellites to see if it can find
 - a rising or setting arc
 - computes a periodogram to find the dominant frequency (reflector height is on the x-axis)
 - defaults are currently
 - 5-25 degrees elevation angle
 - reflector heights >0.5 & < 8 meters
 - L1 GPS frequency
 - Quality Control metrics also displayed



Quality Control

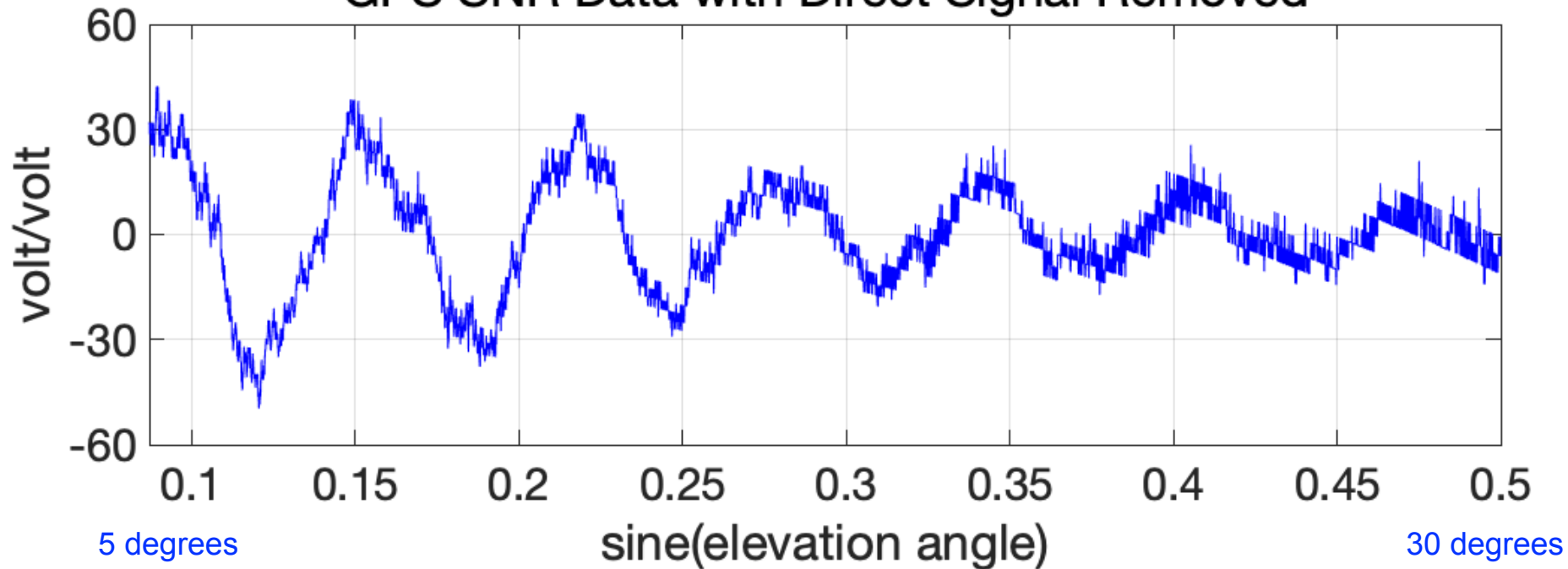


The reflector height zone cannot be too small as it is used for QC.

Why isn't there one special number for quality control?

- Different surfaces and elevation angle masks lead to different peak amplitudes.
- Different receivers (and frequencies) have different quality SNR data (as we will see for L2P vs L2C).
- Peak to noise is a simple metric - it works pretty well, but misses complexity from things like double peaks.
- The “noise” metric should be computed in a consistent way appropriate for your surface.
- QC was easier for PBO H₂O because all the stations were ~2 meters above the reflector. With tides, it is bit trickier. We are still working on making it easier for users.

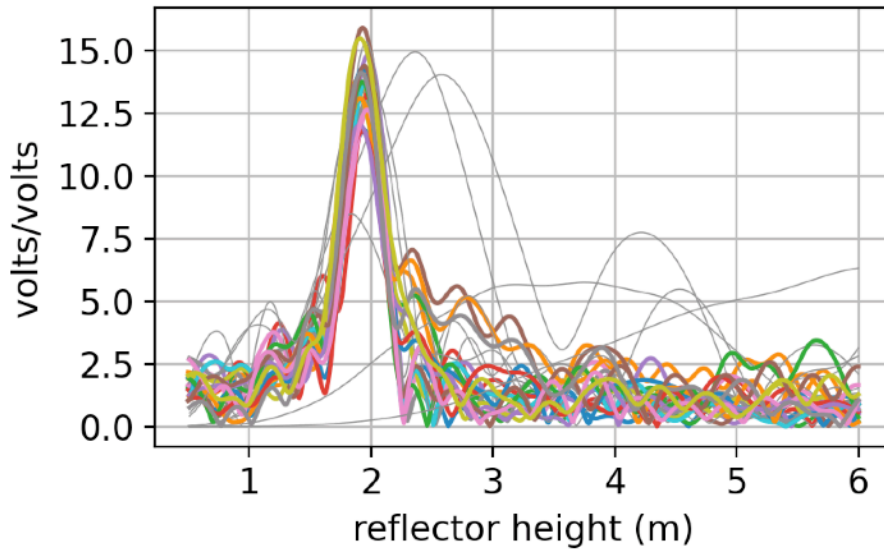
GPS SNR Data with Direct Signal Removed



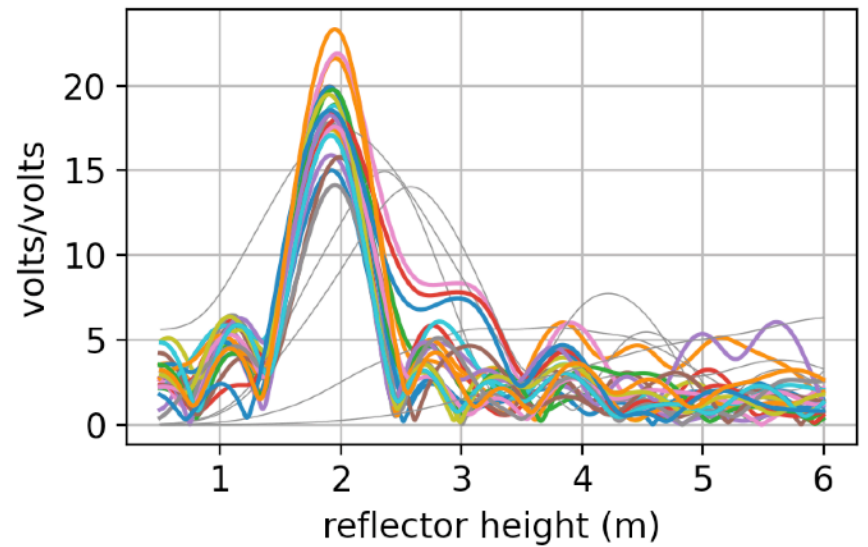
A spectral peak computed for this region will be smaller than the spectral peak using this region

Example periodograms for SNR data

Using elevation angles 5-25 degrees



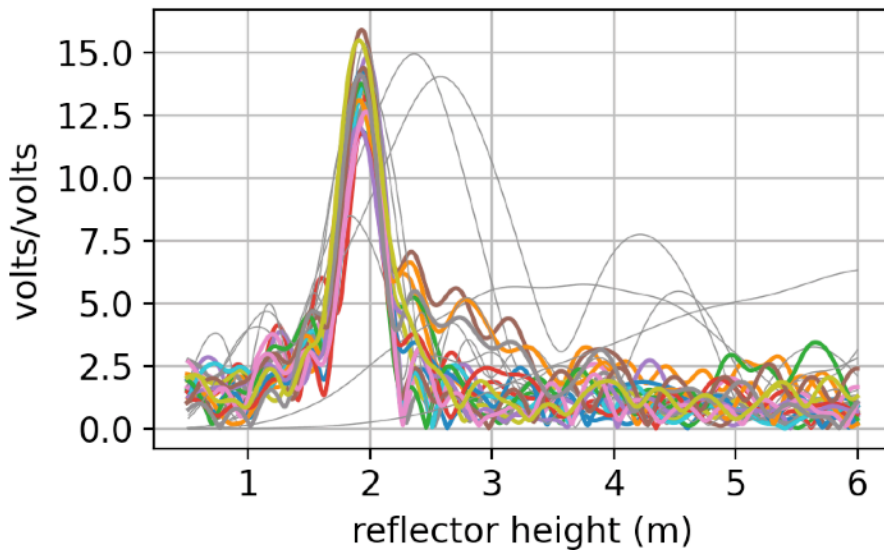
Using elevation angles 5-15 degrees



What do the colors mean?

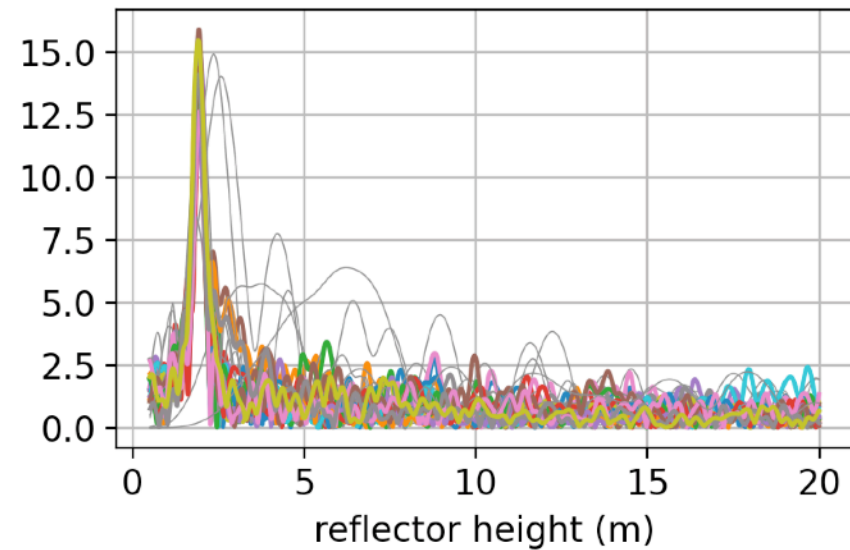
Your choices control the Peak to Noise Ratio

RH 0.5 to 6 meters (original default)



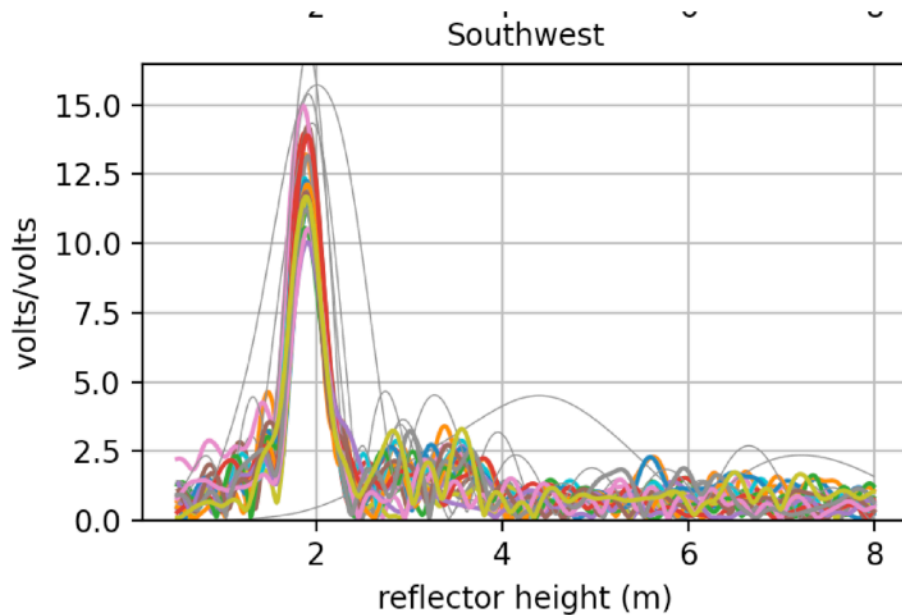
Background noise will be larger here,
so peak2noise of ~6

RH 0.5 to 20 meters



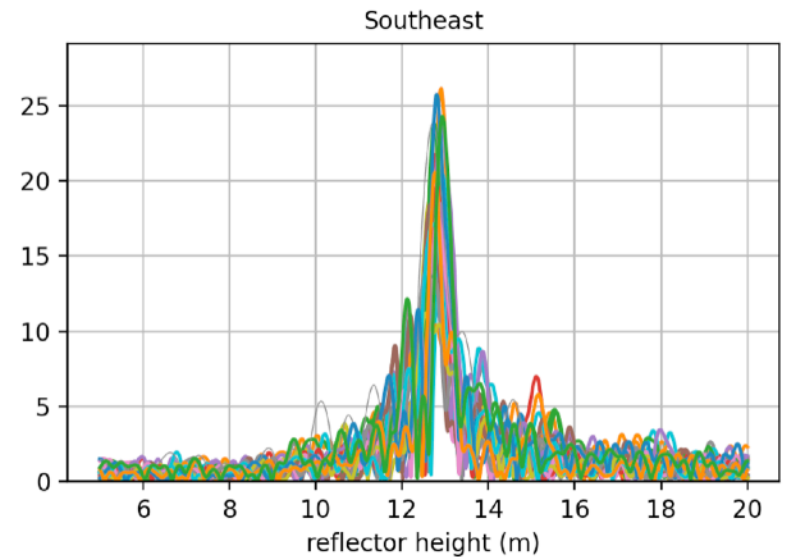
Background noise will be
smaller, so peak2noise of ~15

Peak amplitudes depend on the surface



here the x-axis tells you the reflector height value is ~1.9 meters.
The different colors are different satellites

PBO H2O site - bare soil



in this case the antenna is 13 meters
above the Greenland ice sheet

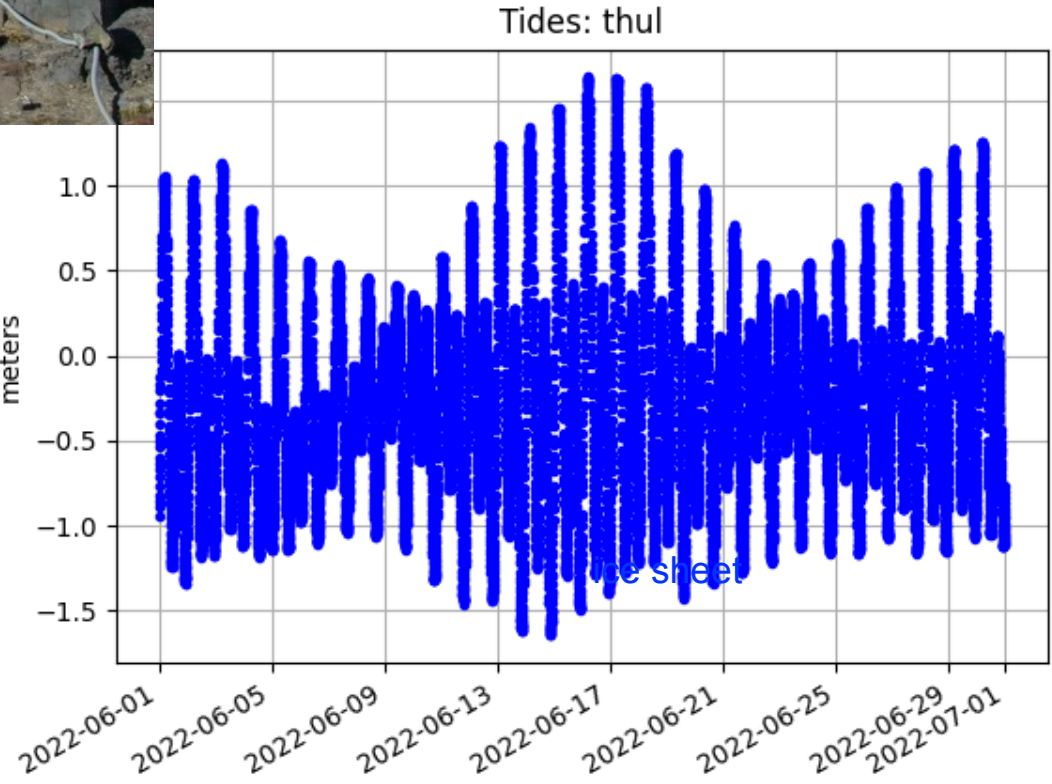
ice sheet

And sure enough, these are quite different surfaces

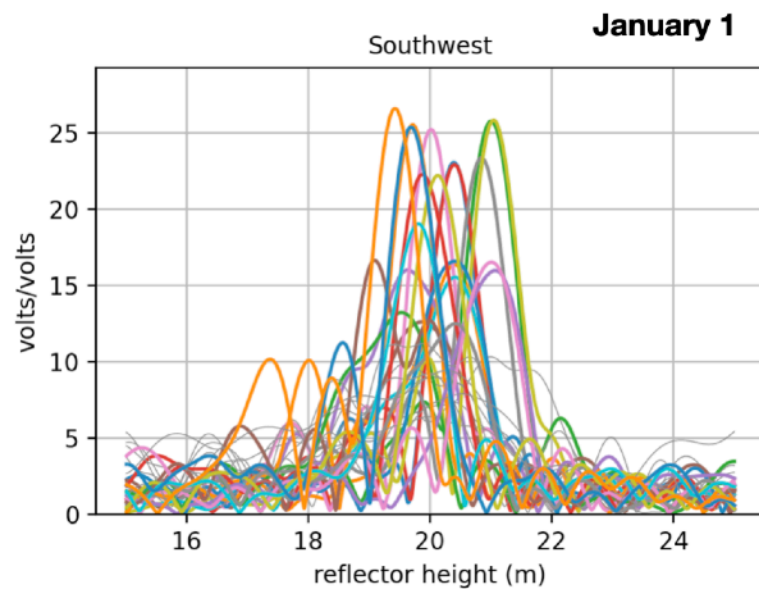
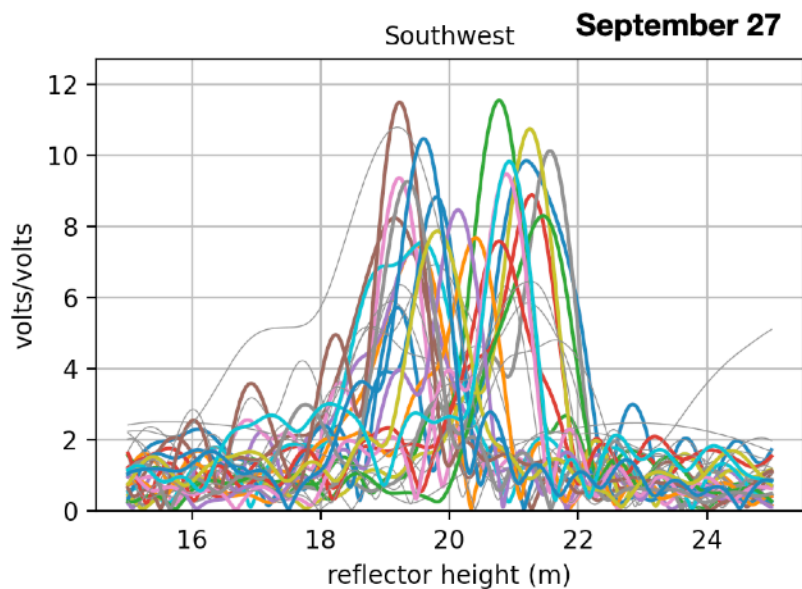




Surface examples for another site in Greenland

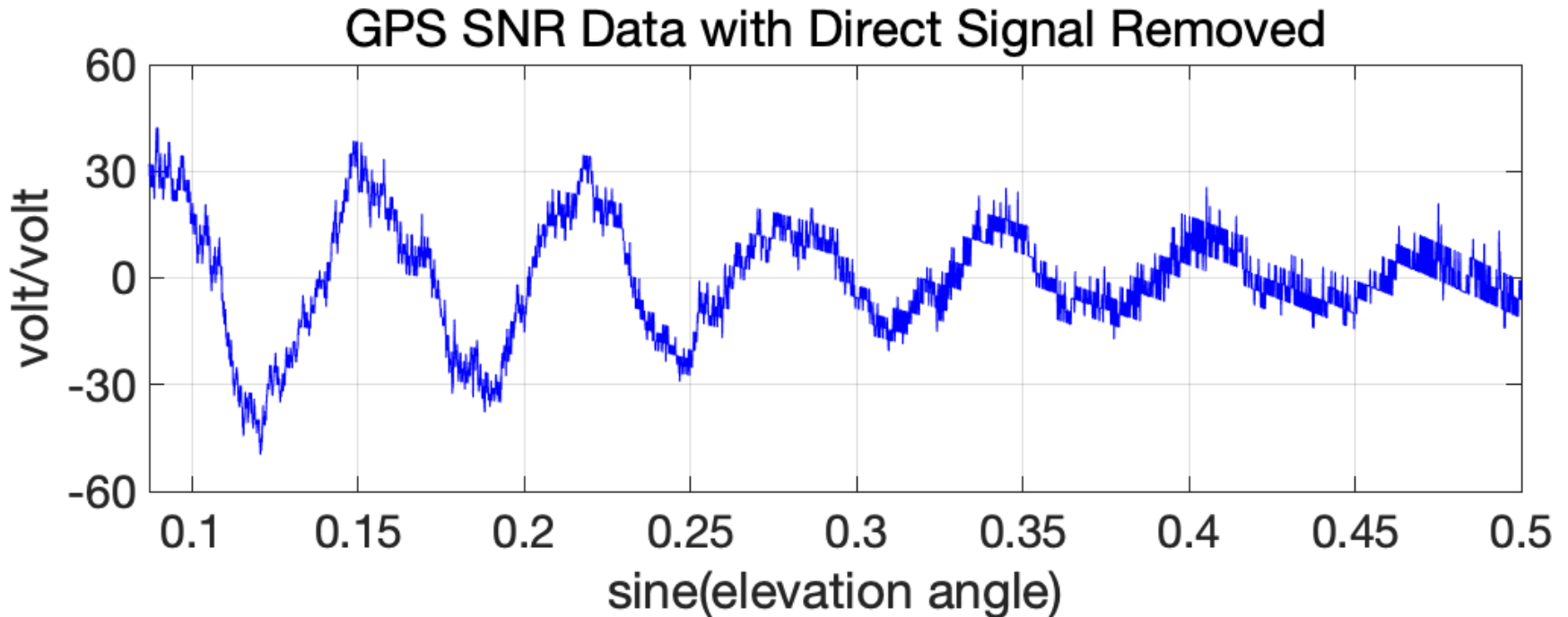


How might September in northern Greenland be different than in January?



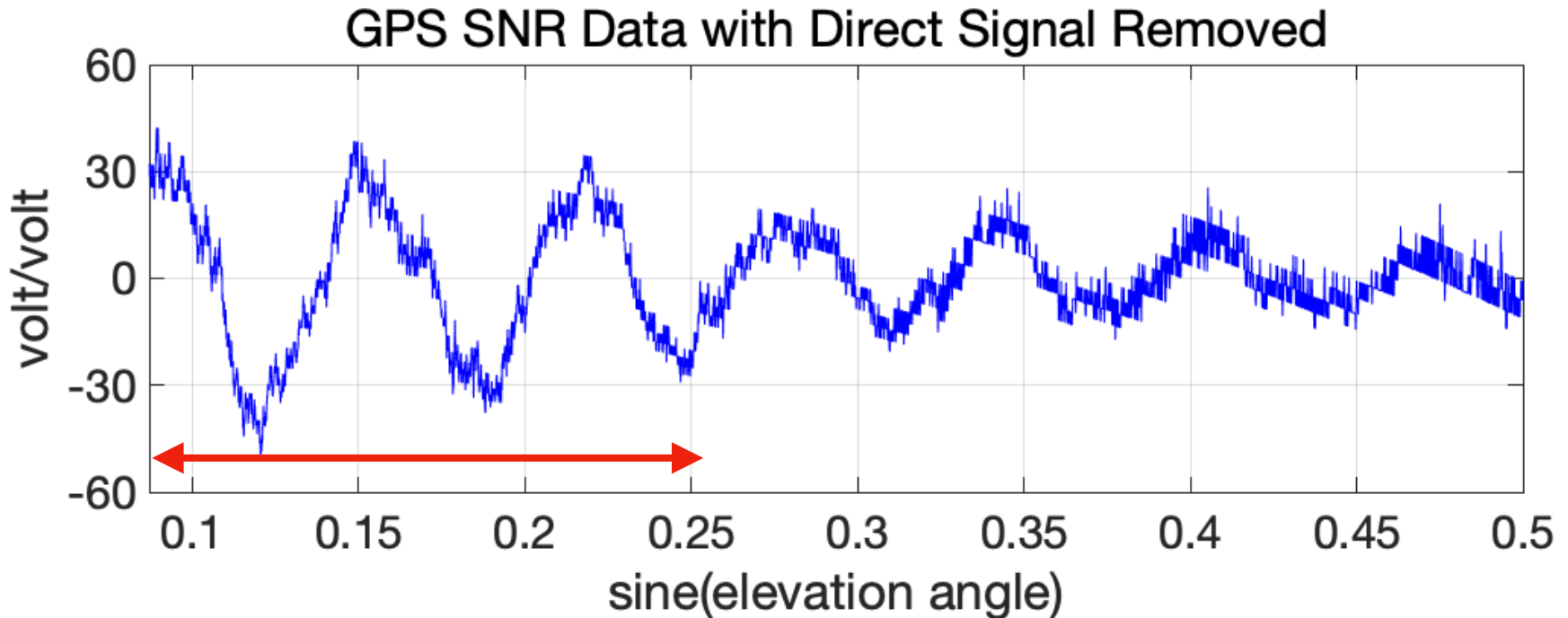
These are for a site that measures water reflections in Thule Greenland. Why are the peaks at different x-axis values?

hidden quality control



This arc has data between 5 and 30 degrees. The surface is planar and a periodogram will give reasonable results.

hidden quality control



What if you compared it with an arc that only had data from the red period ? For some stations and applications this would be fine. But for snow and soil moisture, it would not be fine.

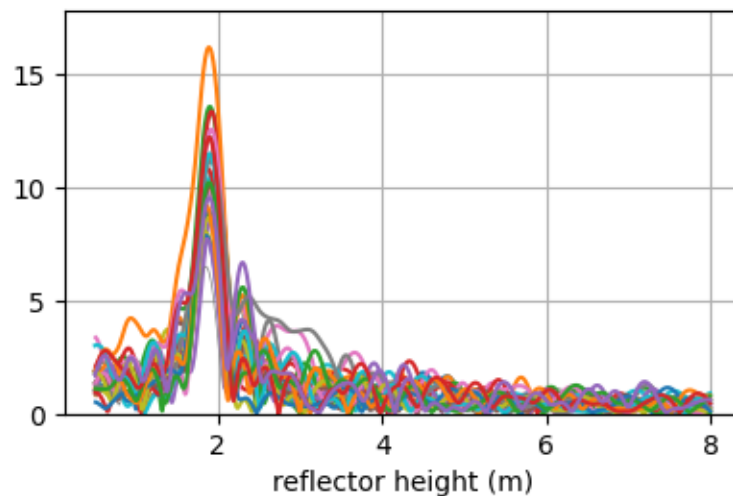
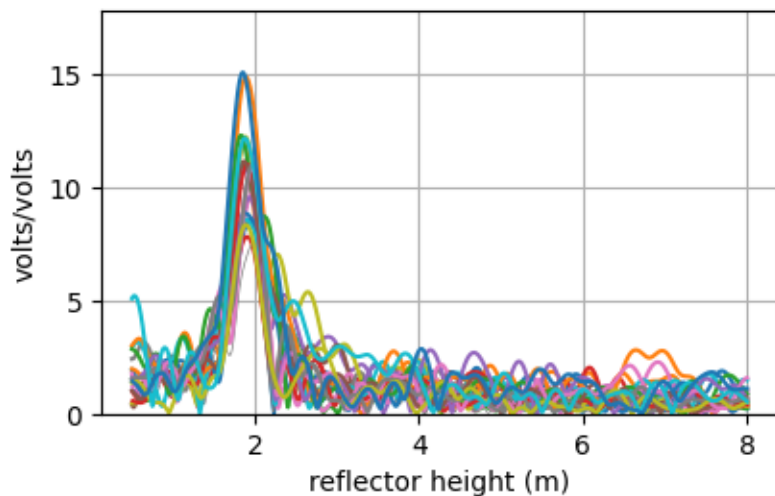
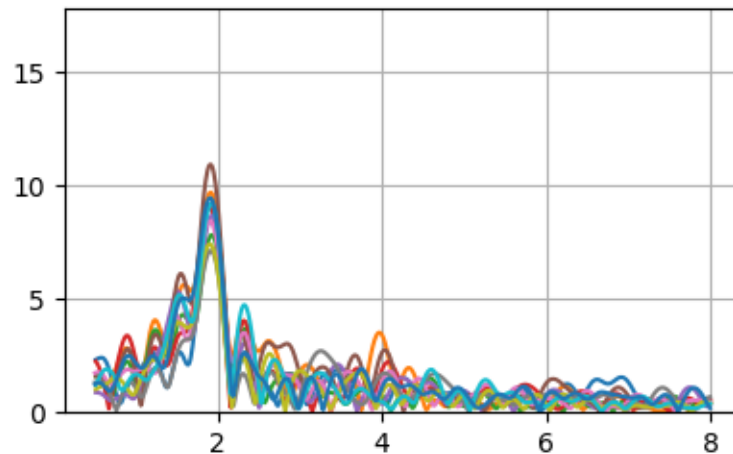
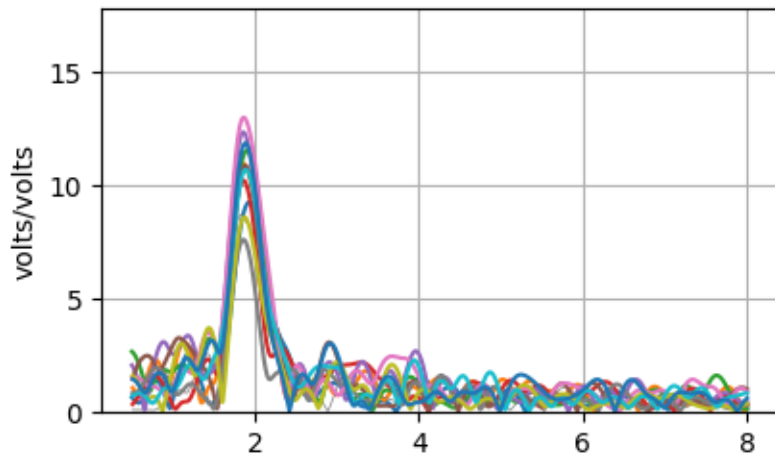
If you ask for arcs between 5 and 25 degrees, the code requires each arc to be within “ediff” degrees of those limits (i.e. between 7 and 23). You can change that value, but 2 degrees is the default.

First we'll try out gnsrefl for a site on soil.
This gives you some perspective on how
water applications are different

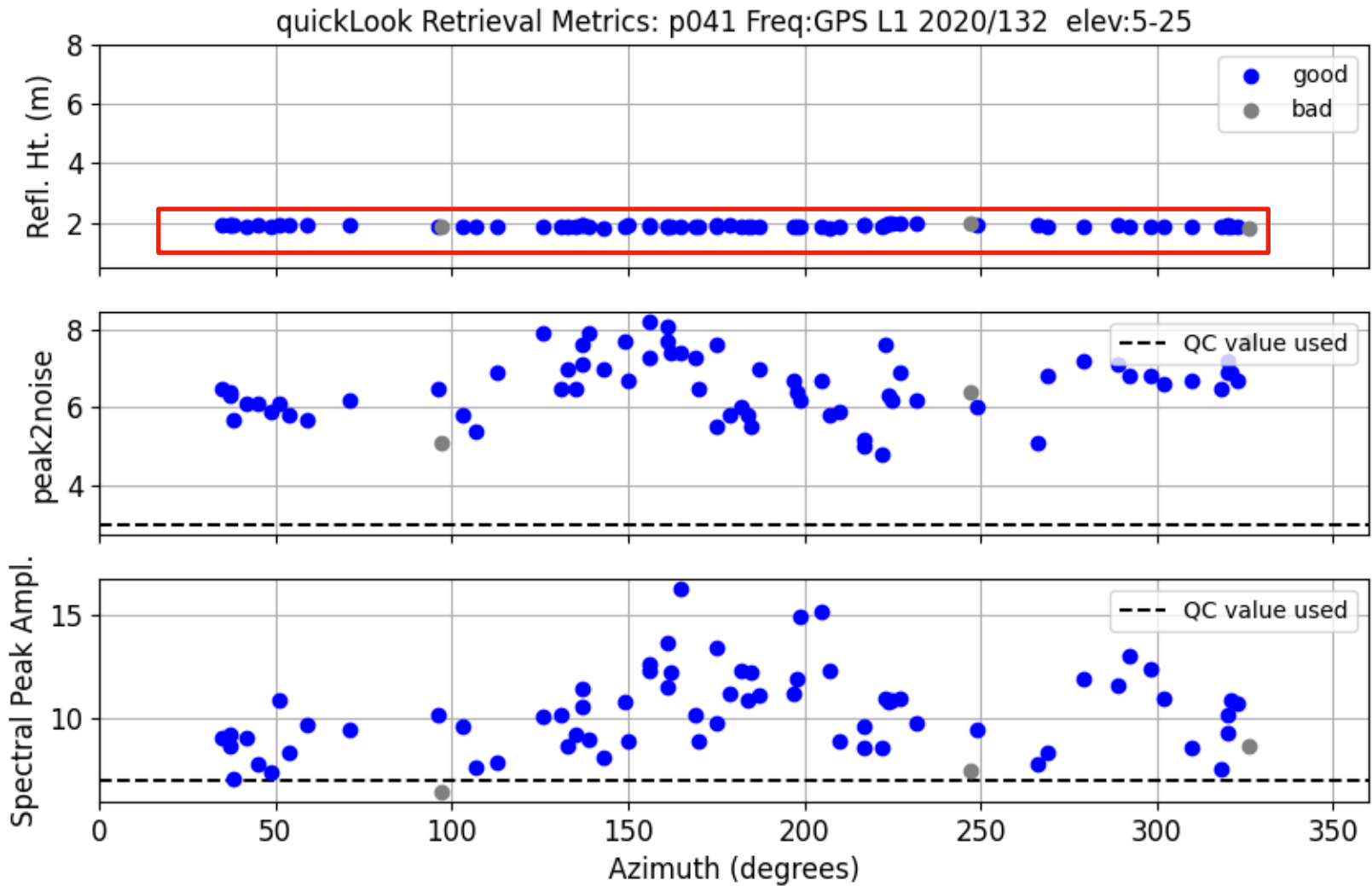


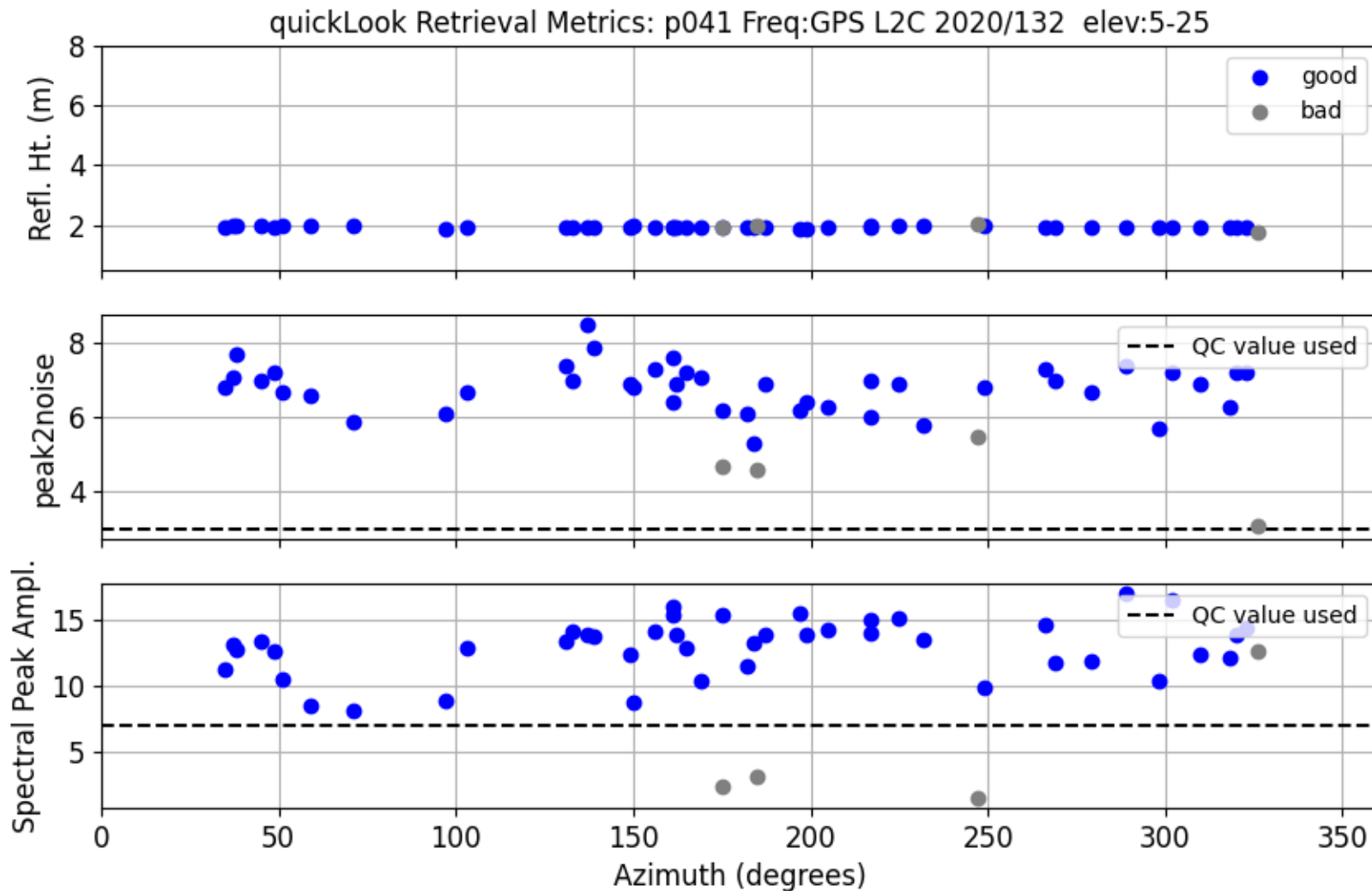
station
p041

GNSS-IR: P041 Freq:GPS L1 Year/DOY:2020,132 elev: 5-25



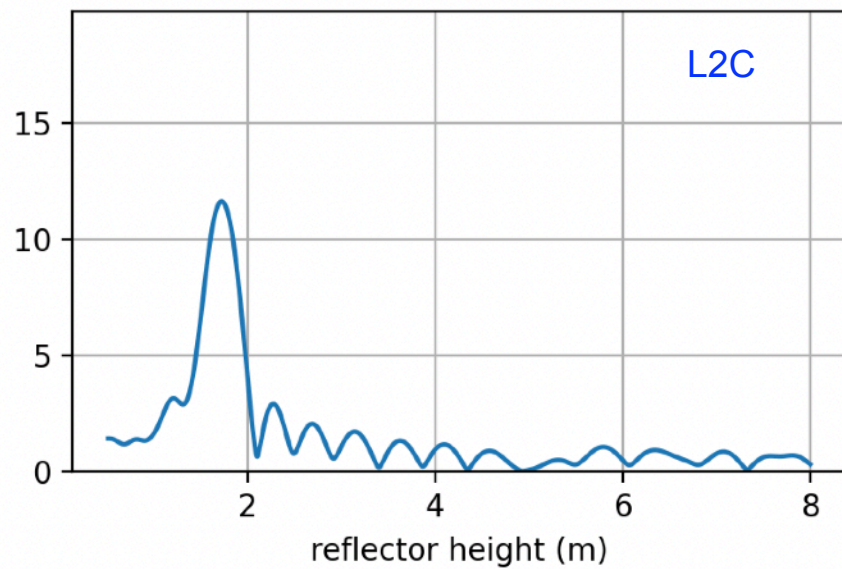
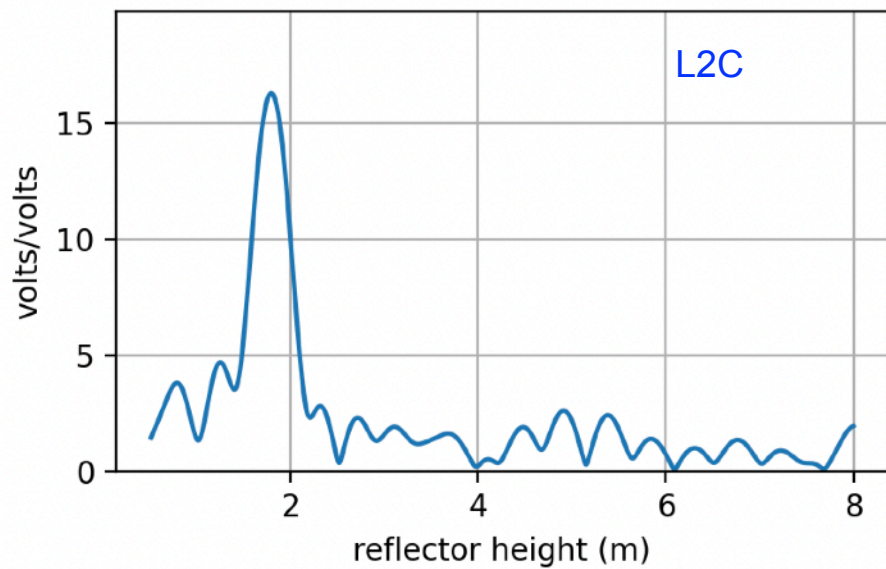
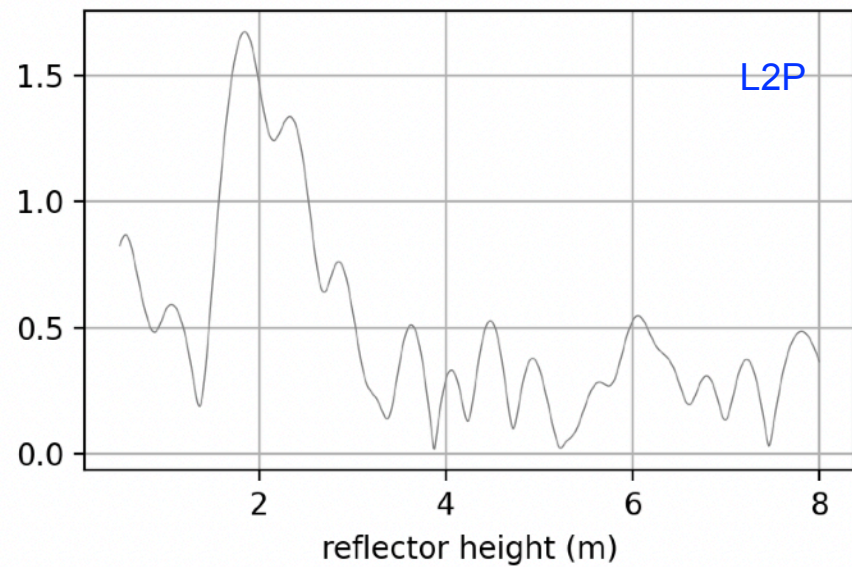
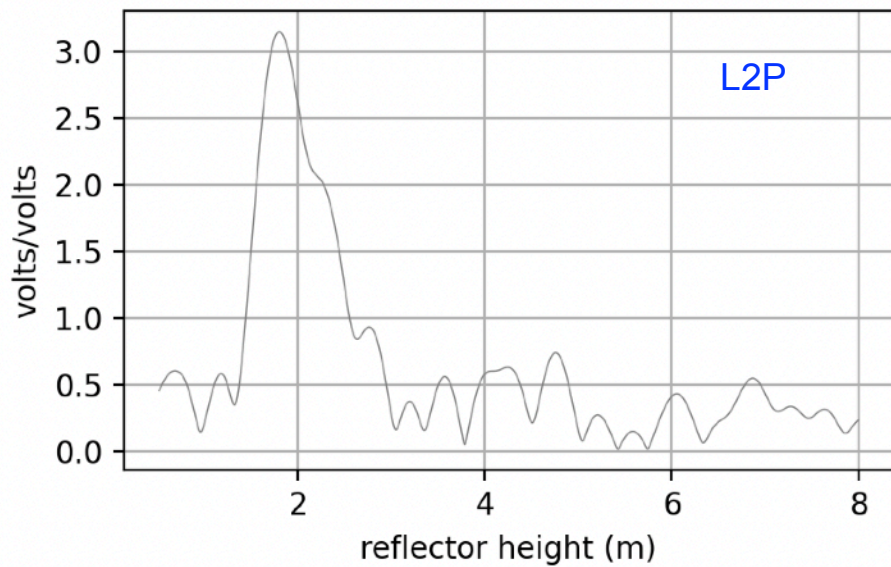
quickLook summary



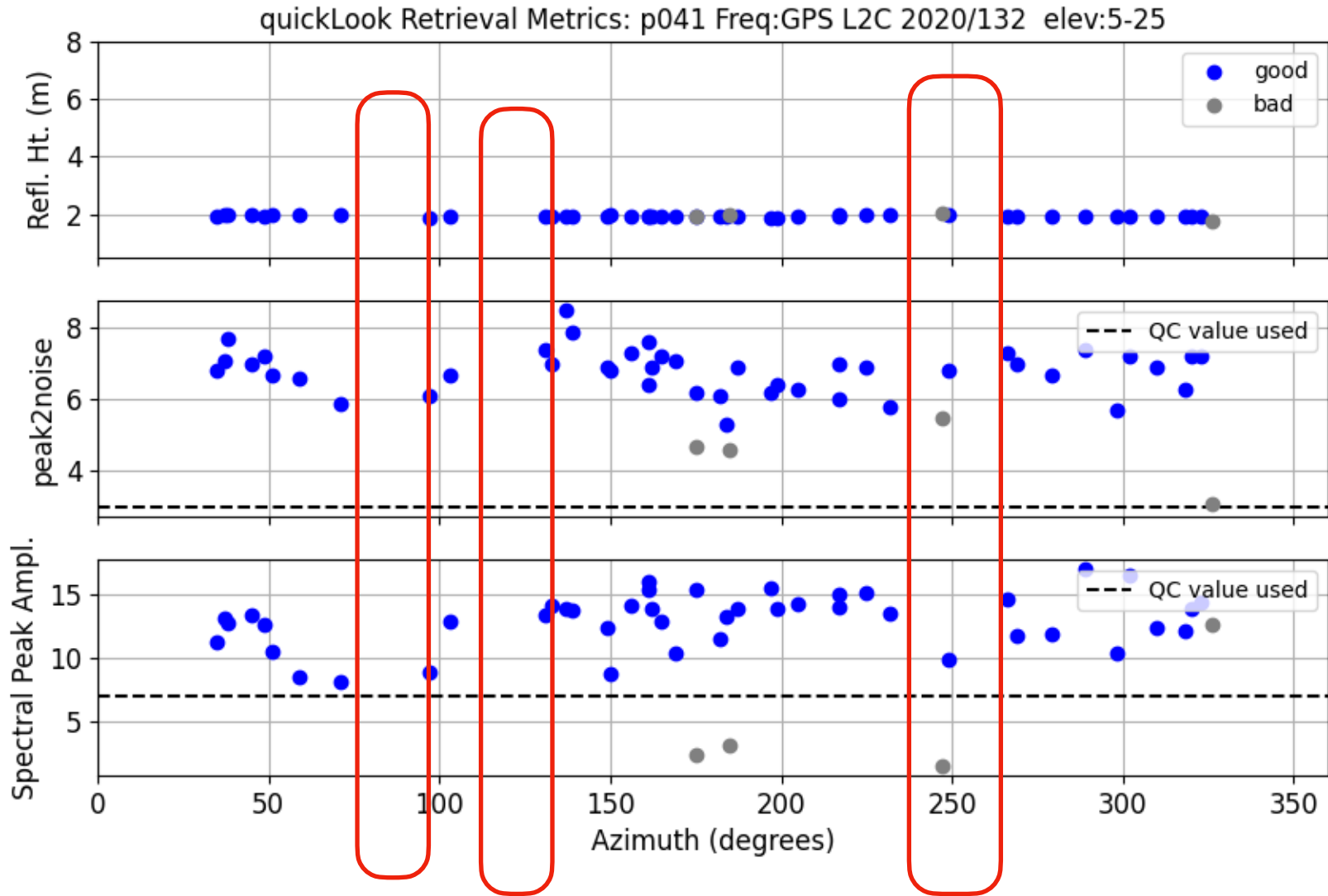


L1 vs L2C

- For this particular receiver, the L1 results look very similar to L2C.
- This will NOT always be the case.



what causes these empty spaces?



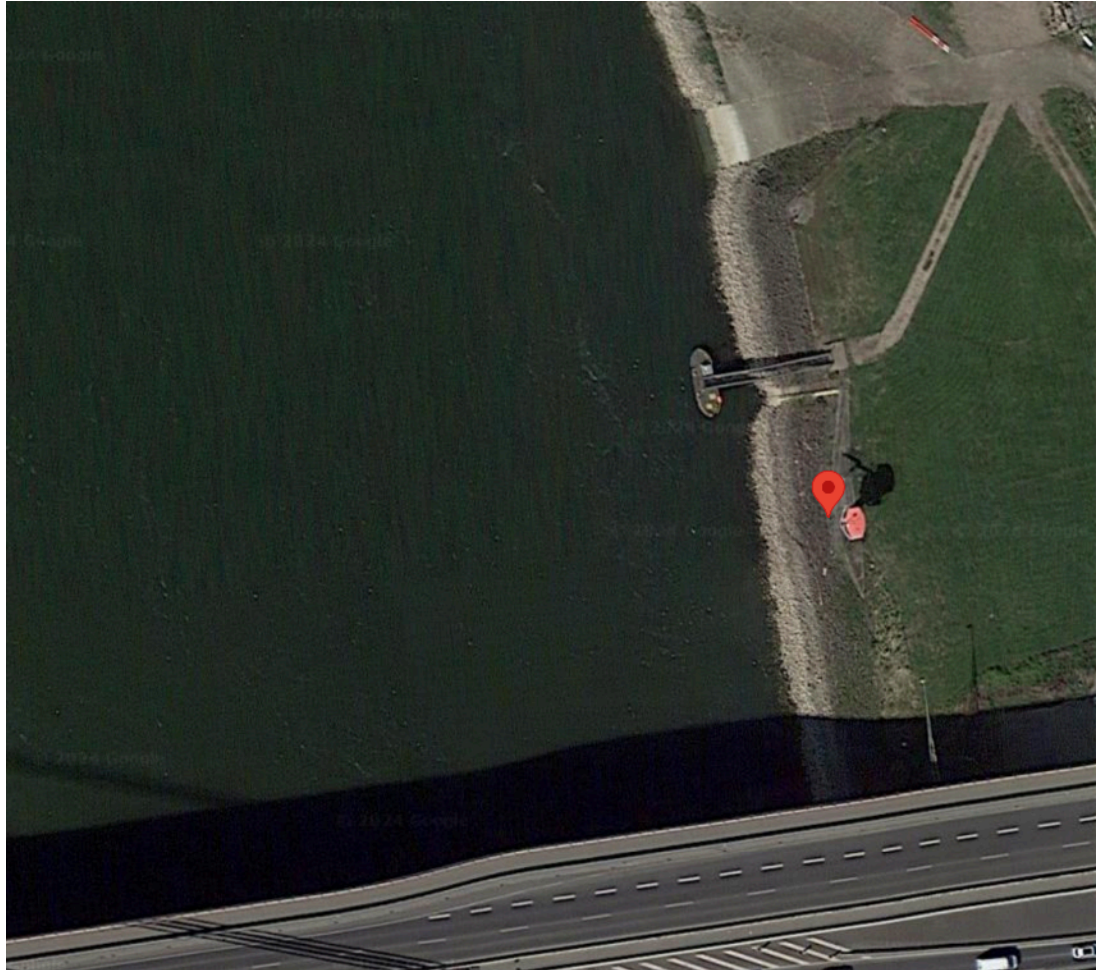
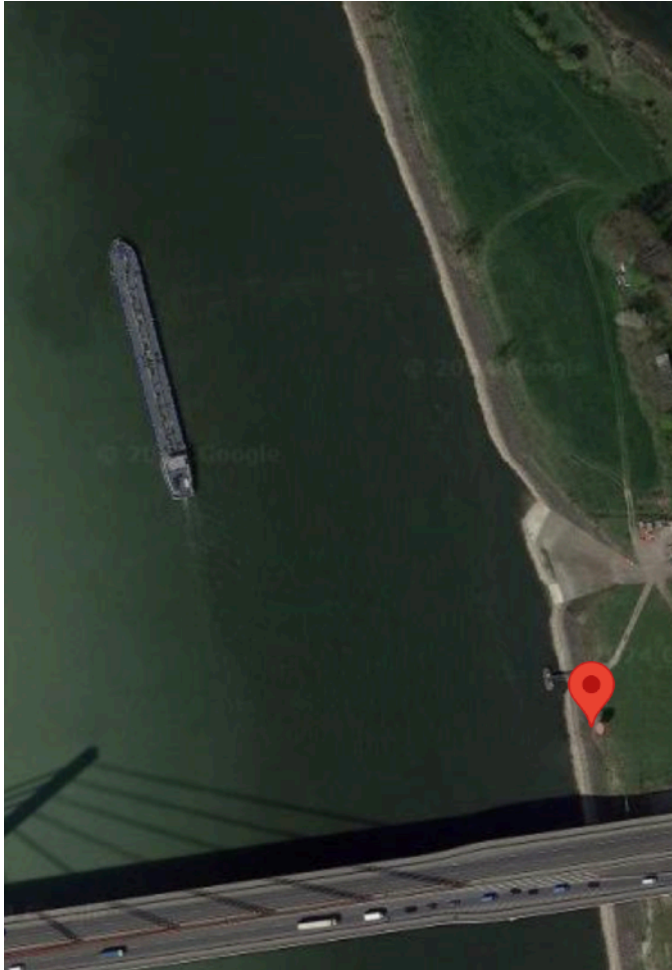
Additional Requirements for GNSS-IR



Wesel on the Rhine River



views from Google Earth





Use reflection web app

I used RH of 10 meters
(we will need to check that)

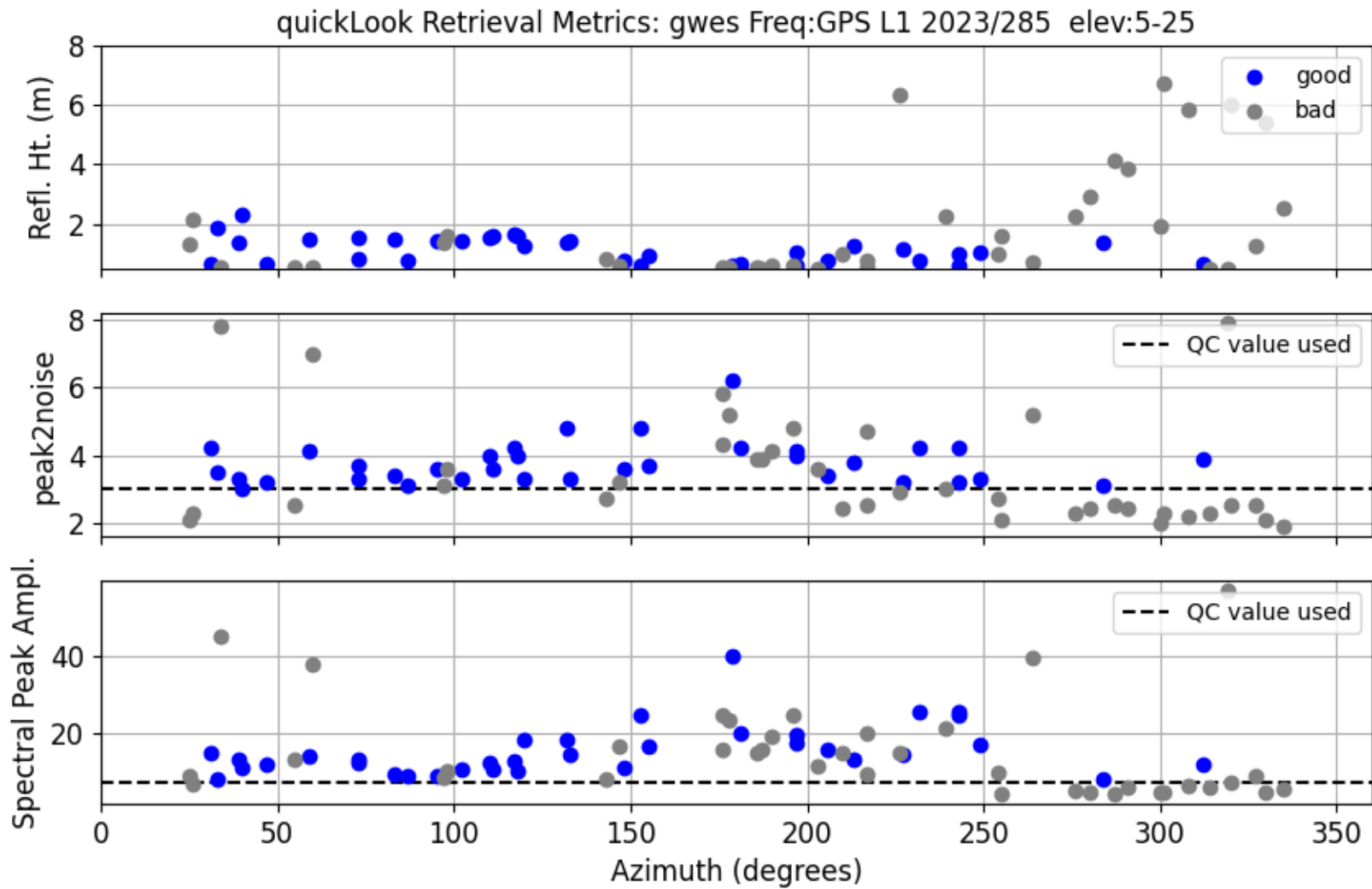
Azimuths 250 to 360
(will also need to check that)

Default is 5, 10, 15 elevation
angles, which is usually pretty
good for water.

Check max_resolve_RH

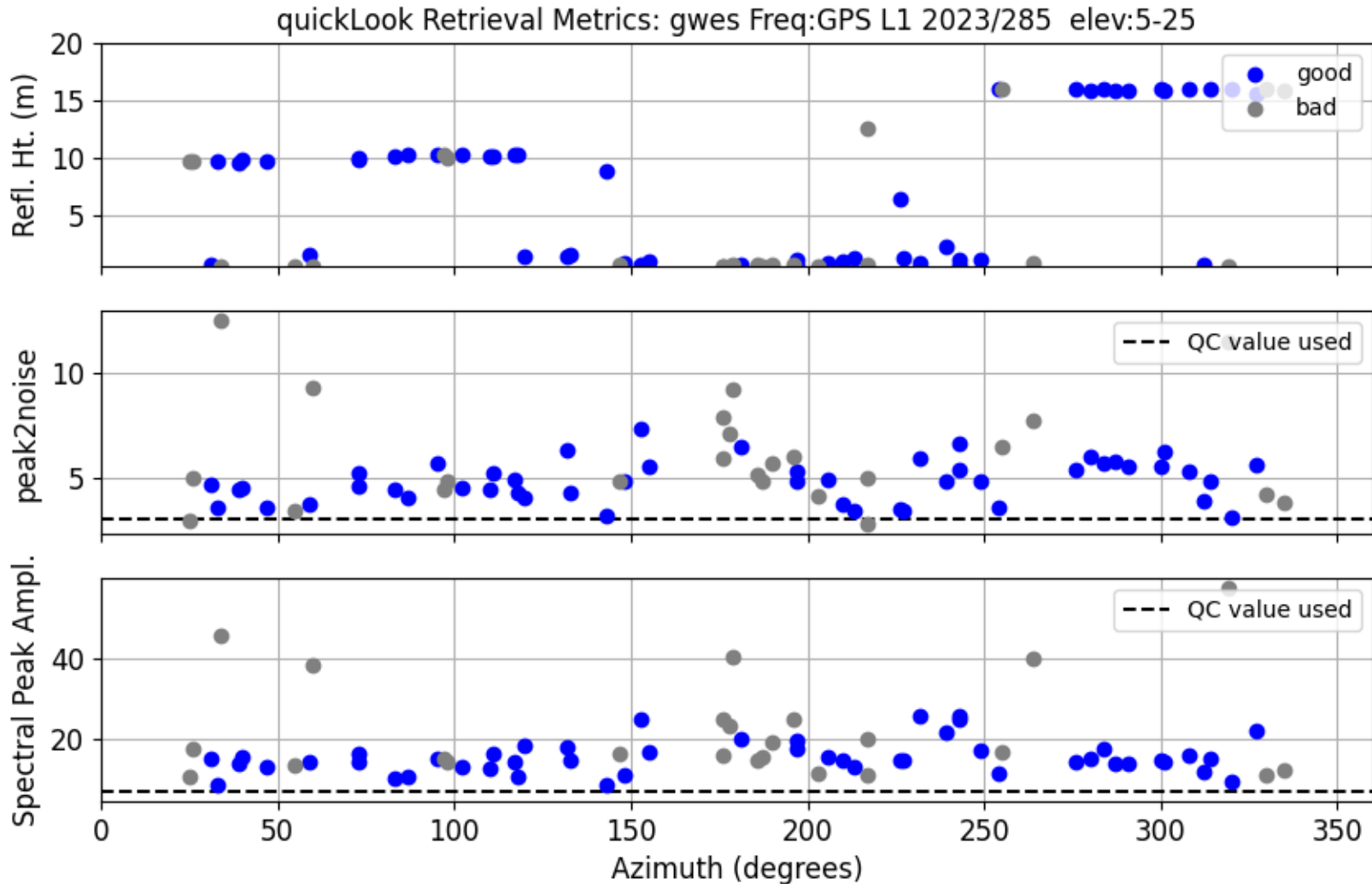
get started

- `rinex2snr gwes00deu 2023 285 -archive bfg -samplerate 15`
- `quickLook gwes 2023 285`



where should the signal be?

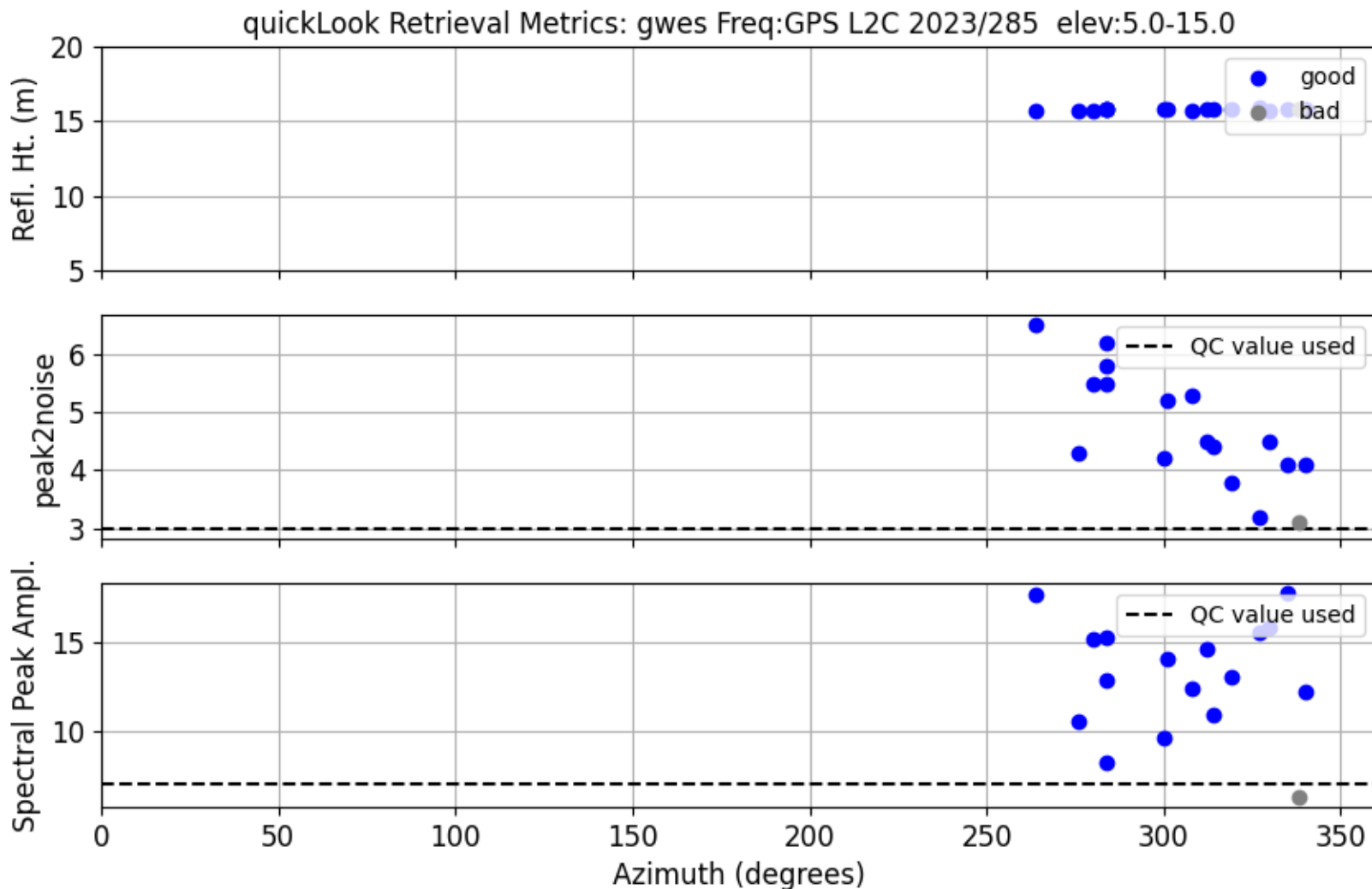
- quickLook gwes 2023 285 -h2 20



not RH=10
as I thought

quickLook gwes 2023 285 -fr 20 -h1 5 -h2 20 -azim1 260 -azim2 340 -e1 5 -e2 15

Note: peak2noise below



You need to save your gnssir analysis strategy

Thanks to the group at Nevada Reno, gnssrefl has a 20,000 station database. If your site is in that database, as this one is, and *if you are happy with the defaults*, you can say:

```
gnssir_input gwes
```

You can also input the lat/long/ellipsoidal height at the command line. It does not have to be super precise - as these are only used to correct the refraction error.

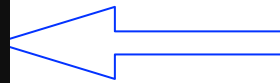
This is what we used for quickLook

```
quickLook gwes 2023 285 -fr 20 -h1 5 -h2 20 -azim1 260 -azim2  
340 -e1 5 -e2 15
```

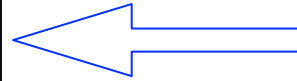
gnssir_input tries to use similar inputs (azimuths are done differently)

```
gnssir_input gwes -h1 5 -h2 20 -azlist2 260 340 -e1 5 -e2 15  
-peak2noise 3 -frlist 1 20 5 101 102 201 205 206 207
```

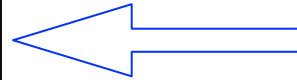
```
"station": "gwes",
"lat": 51.646149169,
"lon": 6.606783068000027,
"ht": 73.0572263,
"Hortho": 28.3472,
"minH": 5.0,
"maxH": 20.0,
"e1": 5.0,
"e2": 15.0,
"NReg": [
  5.0,
  20.0
],
"PkNoise": 3.0,
"polyV": 4,
"pele": [
  5,
  30
],
"ediff": 2.0,
"desiredP": 0.005,
"azval2": [
  260.0,
  340.0
],
```



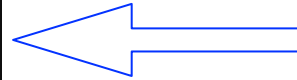
Recovers all valid RH between 5 and 20 meters.
Elevation angles - 5 to 15 degrees



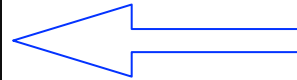
QC Noise region is 5 to 20 meters (pretty big)



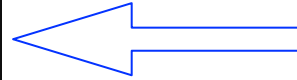
QC Peak to noise is 3



pele: Direct signal model uses 5-30 degr elev. angles



RH precision is 0.005 meters



use azimuths 260-340

Example of some of what is written by gnssir_input

Reminder: Quality Control

- You can set the required periodogram amplitude to zero and rely entirely on peak2noise.
- Setting amplitude QC can be tricky because they have a receiver dependence.
- peak2noise values vary by surface and how big the noise region.
- When all the antennas were 2 meters above the ground, it was easy to have one peak2noise. With tides, we need to rethink how to do this.

now run the gnssir code

gnssir gwes 2023 285

```
% gnssrefl, https://github.com/kristinemlarson, 2.5.2
% Phase Center corrections have NOT been applied
% year, doy, RH, sat,UTCtime, Azim, Amp, emin0, emax0,NumOf,freq,rise,EdotF, PkNoise DelT MJD refr-appl
% (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17)
% m hrs deg v/v deg deg values hrs min 1 is yes
2023 285 16.145 6 7.704 301.89 11.83 5.02 14.98 105 1 1 0.43770 3.48 26.00 60229.321007 1
2023 285 16.224 7 2.994 284.33 9.19 5.07 15.00 260 1 -1 -1.20314 3.98 64.75 60229.124734 1
2023 285 16.225 8 1.333 276.21 13.94 5.03 14.96 109 1 1 0.45568 3.28 27.00 60229.055544 1
2023 285 15.985 10 18.854 264.13 14.19 5.07 14.98 143 1 -1 -0.61307 3.31 35.50 60229.785590 1
2023 285 16.135 15 13.660 284.91 14.32 5.03 14.92 106 1 1 0.44375 3.86 26.25 60229.569178 1
2023 285 16.175 16 22.654 287.34 11.41 5.08 14.98 105 1 1 0.44161 3.73 26.00 60229.943912 1
2023 285 16.370 17 5.415 314.50 14.52 5.06 14.97 106 1 1 0.44639 3.36 26.25 60229.225602 1
2023 285 16.144 18 15.167 284.81 13.99 5.09 14.99 223 1 -1 -1.00987 4.71 55.50 60229.631933 1
2023 285 16.225 20 10.100 291.15 11.75 5.05 14.93 109 1 1 0.45750 3.58 27.00 60229.420822 1
2023 285 16.034 26 21.498 280.19 17.22 5.06 14.96 108 1 1 0.45350 4.80 26.75 60229.895741 1
2023 285 16.130 31 20.542 308.08 12.20 5.06 14.93 101 1 1 0.42543 3.27 25.00 60229.855903 1
```

What if you think you should have better results from gnssir?

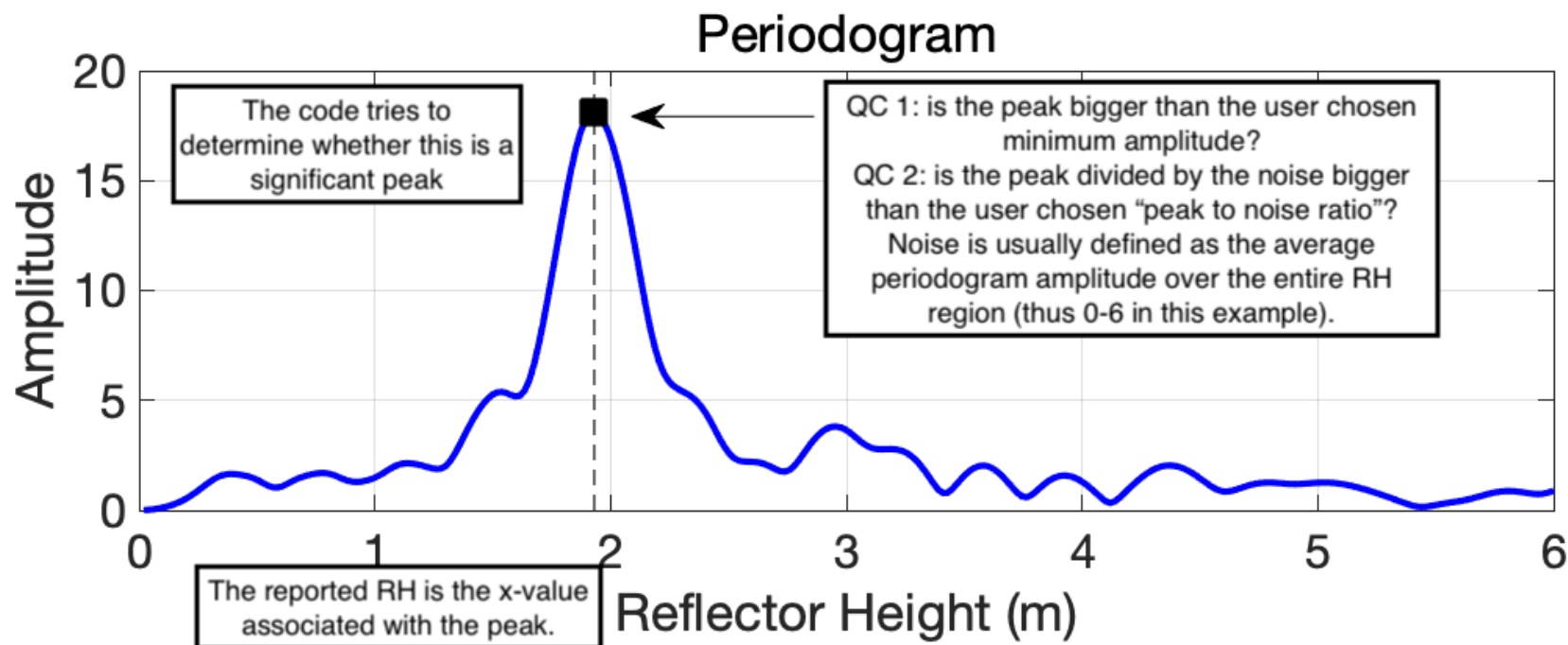
- try `-screenstats T` when you run *gnssir*
- It will try to tell you why your track was rejected, was the amplitude too small? Was the arc too long? Was the arc too short?
- use `-plt T` if you want to see plots

In gnssir there are actually two sets of numbers for the x-axis.

-h1 and -h2 tell the code which RH values are allowed. It will compute RH values out to the maximum value (h2).

-nr1 and -nr2 tell the code the min and max RH value for the noise computation.

The defaults are to treat these the same, but you can control them in *gnssir*



Can you change the strategy in *gnssir* on the command line?

- It isn't illegal (at least for some parameters), but that does defeat the purpose of having a strategy
- If you want to compare strategies, make two sets of analysis instructions using the `-extension` flag. That way your results will be put into different directories.

Multiple days and years

```
gnssir gwes 2023 285 -doy_end 300
```

```
gnssir gwes 2023 285 -year_end 2024 -doy_end 15
```


What do you do next?

It depends on what you are trying to measure.

For things that *do not* have significant sub-daily behavior, lakes and rivers, we use **daily_avg**.

For tides, we use the **subdaily** module.