LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Document Type **LIGO-T030175-00-0-Z**

Quantiles Based Automated Line Detection

Y. Itoh[1], S. Borger[1], S. Mohanty[1], A.M. Sintes[1,2], S. Babak[3]

[1] *Max-Planck-Institute für Gravitationsphysik, Albert-Einstein-Institut, Am Mühlenberg 1, 14476 Golm, Germany*
[2] *Department de Fisica, Universitat de les Illes Balears, E-07122 Palma de Mallorca, Spain*
[3] *Department of Physics and Astronomy, Cardiff University, PO Box 913, Cardiff, CF2 3YB, U.K.*

Distribution of this draft:

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW: http://www.ligo.caltech.edu/

Processed with LaTeX on 2003/08/13

Contents

1 Introduction

Recognizing lines in the amplitude spectral density [1] and measuring their characteristics may give us one of basics for detector characterization and help us improve its sensitivity.

In the amplitude spectral density, generally one could see various spices of lines: ultra thin lines, thin lines, broad lines, drifting lines in frequency with time. They are sometimes isolated and sometimes crowd together.

These lines are often easily found by visual inspection and thus it is an optimal method to use eyes to detect the lines. It is not a good idea in a long run, however, if there are hundreds of lines and tens of channels. Moreover, the results obtained by visual inspection, especially measurements of lines characteristics, possibly depend on human status.

A suitable automated lines detection code on computers runs much more quickly and a clearly defined procedure gives us unambiguous measurement of the lines characteristics. We can look into various channels and detect hundreds of lines within affordable time and also we can implement the method into a online monitor. It is interesting to have a online monitor which detects lines in the channels, since the status of the detector changes second by second and it is possible that we fail to notice many lines appearing which might indicate abnormalities of the detector and/or environment and the reason of them.

A problem associated with automation is that it is difficult to realize criteria for recognizing a line optimized for all the channels and any time chunks. This problem is related with a definition of a line and noise floor (continuous part of the amplitude spectral density. A continuous spectrum in the astrophysical context). Let us show some example. Given that there are many "lines" on a "bump". Then we sometimes say that there is a wide broad line with many notches on it, or we say that the bump defines a noise floor and there are many lines on it. For other example, if there is a drifting line in the data, then is it possible for the code to recognize it as a one line ? Notice that the extent of drift partly depends on time stretch that we take.

There seems no perfectly satisfying solution of the above problem for all our purposes, since the definition of a line and noise floor depend on the very purpose: which frequency band is interesting, what kind of lines we need to know, and so on.

Here we propose an automated lines detection algorithm which provides several parameters which can be freely tuned for various purposes. Also we report results obtained by the application of our algorithm to January data of GEO 600 (the channel is `G1:h`) and the S1 run data. Our purpose here is to detect lines and measure characteristics through all the channels and make a database of the results for later detector characterization and gravitational wave search purposes. The present results are collected into a MySQL[1] database on the web[2]: `"freq_data"` of "frequencies" on
`http://info.geo600.uni-hannover.de/frequencies/phpMyAdmin-2.2.0/`.

We would like to notice that our algorithm is robust in the following sense. We define a noise floor to detect lines and measure their amplitudes etc., however, the noise floor itself is calculated by the amplitude spectral density and should follow it. So if the noise floor tightly follows the original amplitude spectral density, we could not detect lines efficiently. By using running median[3, 4] (see also, [5]), we successfully construct a robust noise floor which does not follow precisely the original spectral density at frequencies where there are lines. Also we have used a running quartile to define a robust criterion for line recognition.

We have confirmed that the analyzed channels, the lines detected by our code are almost always seen by quick visual inspection. Through our experiences we have found that once a channel and time stretch are chosen, our code with some suitably (but not severely) chosen sets of free parameters can detect almost all kind of lines efficiently.

We shall explain our method in detail in Sec. 3. The results are found in the section 4 and appendices A, and B. We mention the analysis result on the S1 data in Sec. 4.2. The definition of various quantities used

[1] The amplitude spectral density is called linear spectral density in Ref [6].

in the method is explained in Sec. 7.

2 Implementation

Our automated line detection code has been written with MatLab[7]. The required toolboxes are Signal Processing Toolbox and Statistical Toolbox. Use is made of the GEO-Tools[8] to load time series of data, calibration information from the data server.

A online monitor has been being developed in the framework of *GEO++*[9].

3 Method

3.1 Overview and free paramter

In a visual inspection, we recognize a potion of the amplitude spectral density over some bins as a line if it is well above the local standard deviation around the noise floor at the bins concerned.

Thus, the basic idea of the method is as follows. First we estimate the amplitude spectral density of the data, then estimate the noise floor by using running median[3]. Next we estimate the running standard deviation by using running interquartile-ranges. Then we detect those lines that are above the standard deviation time some positive number, which is one of the free parameters of the present method. Finally the analysis results are collected on a MySQL[1] database available on the web[2].

There are four important free parameters in our algorithm.

(1) `blocksize_fine` over which the amplitude spectral density is smoothed.
(2) `blocksize_coarse` over which the running median and running quartiles are calculated.
(3) `threshold_broad`: threshold for detecting broad line candidates. A positive number
(4) `threshold_thin`: threshold for detecting lines. A positive number larger than `threshold_broad`.

In general, broad lines have notches on them. To avoid to detect such notches, we smooth the amplitude spectral density using running median calculated over `blocksize_fine` and compare the noise floor with the smoothed amplitude spectral density when we try to detect broad lines. So, `blocksize_fine` should be small enough to follow the rough feature of the amplitude spectral density.

We estimate the noise floor and running standard deviation by using the running median and running quartiles calculated over `blocksize_coarse`. Thus, `blocksize_coarse` should be large enough not to follow the amplitude spectral density precisely and small enough to give the robust estimate of the noise floor and running standard deviation. If `blocksize_coarse` is set to be too large, and if the amplitude spectral density has a convex form, then we would detect very broad bumps.

Then if the smoothed amplitude spectral density over some bins exceed a criterion line defined by (the noise floor) plus (`threshold_broad` times the estimated standard deviation), it is a broad line candidate. Since we use the smoothed amplitude spectral density, we can avoid to detect notches on a line. However, it is possible to detect a broad bump mistakenly which is not a line. Thus we compare the amplitude spectral density with another criterion line defined by (the noise floor) plus (`threshold_thin` times the estimated standard deviation). We then detect isolated thin lines as well as the broad lines which have enough amplitude to be lines. Therefore, the `threshold_thin` has to be larger than `threshold_broad`.

3.2 Step by step description

Herefrom we shall explain step by step our method on automatic line detection and measurement of their characteristics. The choice of free parameters explained above will be shown in Sec. 4.

1. Load from a frame-server a **nSec** continuous seconds worth time series data of a **channel** which starts at `start_time`.

2. DC (and trend) would be removed if necessary.

   Notice that in GEO data , if we do not remove DC, DC has relatively enormous amplitude compared with the other part in the amplitude spectral density. Also the amplitude spectral density often has relatively large amplitude at low frequencies. Since we estimate the amplitude spectral density using window functions to solve a problem caused by the finiteness and discrete sampling of the time series data[6], there should be power (or amplitude) leakage into the neighboring frequencies (bins). Therefore, if the dynamical range of the spectral density is large as may be the case due to DC, care must be taken to keep such a power leakage from masking the true spectral density features. Use a window function which has low sidelobe[6], if necessary.

3. Choose a **frequency resolution, fr**. Note a relationship

$$fr = \frac{f_s}{N}, \tag{1}$$

   where f_s is a sampling frequency of the time series and N is the length of the discrete Fourier transform[6].

4. Estimate an **amplitude spectral density, Asd(f)**, from the time series data. Notice that $Asd(f)$ is called the linear spectral density (LSD) in Ref. [6]. We have used "Welche's method of averaging modified periodogram"[10].

5. Calculate **ENBW** (= effective noise band width, see [6]). The ENBW depends only on the window function and the frequency resolution. See also 7.

   Notice that we need ENBW to calculate the amplitude of the lines. Note the following equations.

$$\mathrm{PS} = \mathrm{ENBW} \cdot \mathrm{PSD}, \tag{2}$$
$$\mathrm{AS} = \sqrt{\mathrm{PS}}, \tag{3}$$

   where PS and PSD stand for the power spectrum and the power spectral density respectively. AS denotes the amplitude spectrum, also called linear spectrum.

6. Calibrate the amplitude spectrum density, if necessary. The calibration information is available from the frame data itself. Then if only either whiting filter or system filter is available, we multiply the absolute value of it by the amplitude spectral density to achieve the calibration. If both of the filters are available, then we multiply the absolute values of the filters and then multiply it by the amplitude spectral density to obtain calibrated amplitude spectral density.

   Notice that some calibration functions are not reliable within some frequency band. So we record only the lines within the frequency band where the calibration information is valid.

7. Calculate running median over `blocksize_fine`, we call it `running_median_fine` from now on. `Blocksize_fine` is a free parameter of the present algorithm. This gives **smoothed Asd** with which we detect broad line features in Asd. `Blocksize_fine` needs to be tuned to make sure that the `running_median_fine` should follow Asd while smears out the notches on the original Asd. Thus the choice of the free parameter `blocksize_fine` depends on what kind of broad lines one is concerned about, or more specifically, how broad the lines which should be detected are.

8. Calculate running median over `blocksize_coarse`. `Blocksize_coarse` is a free parameter of the present algorithm. We call it `running_median_coarse` from now on. We take `Running_median_coarse` as a estimator of a **noise floor** of Asd. In other words we define **noise floor** of Asd by **noise floor** = `running_median_coarse`. We compare the **noise floor** with the original Asd and **smoothed Asd** to detect lines.

   `Blocksize_coarse` need to be tuned to make sure that the **noise floor** "roughly" follows the original Asd. If **noise floor** follows the Asd precisely, we would only detect small notches but miss prominent broad lines. While if **noise floor** does not follow the Asd, we would detect fictions lines.

   - *Note on the step* 8: It is possible that `blocksize_coarse` is taken so wide that there are too many data points within the blocksize and we may need too much memory to compute a running median. In that case, we have resampled **original Asd** at , say, every **coarsing factor** bins, calculate a `running_median_coarse` of resampled Asd, then linearly interpolate to obtain `running_median_coarse` at all of the bins. The **coarsing factor** is a (technical) free parameter.

9. Calculate first and third running quartiles over `blocksize_coarse`. Then the estimated running standard deviation at i-th bin (i:positive integer) is given by

$$erstd_i = (q3_i - q1_i)/1.34, \tag{4}$$

   where $erstd_i, q3_i$, and $q1_i$ are the **estimated running standard deviation at i-th bin**, the **third quartile at i-th bin**, and the **first quartile at i-th bin**. The factor 1.34 is from the case of normal distribution. (Overall constant factor is not important anyway, since we multiply the **estimated running standard deviation**. by a constant number (threshold).)

10. In above steps we explained the quantities to be needed in the present algorithm and how to calculate them. Now we shall explain how to detect lines in the following few steps.

    With the amplitude spectral density (Asd) in hand, suppose that we look into Asd from the lower frequency to higher frequency bin by bin. At the i-th bin, examine a criterion defined as

$$rmf_i \geq rmc_i + thrb \times erstd_i \tag{5}$$

    where rmf_i, rmc_i, and $ersted_i$ are `running_median_fine` **at the i-th bin**, `running_median_coarse` **at the i-th bin**, and **estimated running standard deviation at the i-th bin**. The index i runs from 0 to i_{max} which corresponds to the highest frequency ($i_{max} = N/2$ where N is the length of the DFT). $thrb$ is a positive constant (over whole frequency range) number and denotes `threshold_broad`. `Threshold_broad` is a free parameter of the present algorithm.

    If the equation (5) holds, there is a line **candidate** at the i-th bin. Register the bin. Now we stop at the i-th bin.

11. Now, Move to the next bins. Examin Eq. (5) at each bin. Then suppose that consecutive bins (the i+1,i+2,... j -th bins) are registered in the above step (while the (i-1)-the bin and the (j+1)-th bin are not registered). Then these bins together form a broad line *candidate*. Examine a equation

$$Asd_k \geq rmc_k + thrt \times erstd_k, \tag{6}$$

    where Asd_k and rmc_k are **original Asd at the k-th bin** and `running_median_coarse` **at the k-th bin** (Note that The index k $\in \{i, i+1, ..., j\}$). $thrt$ is a positive constant (over whole frequency range) number and denotes threshold_thin. Now if the equation (6) holds at least at one bin with its index $k \in \{i, i+1, ..., j\}$, then the set of the consecutive bins together is *defined* to form one broad line. Calculate the characteristics of the broad lines, e.g., central frequency of the line, bandwidth, full width half maximum, signal to noise ratio, root mean square amplitude, and so on. See the table in Sec. 4.2 for the quantities which we have recorded.

Quit the operation for the set of bins and move to (j+2)-th bin and examine the step 10.

12. On the contrary, suppose that in the step 10 the i-th bin is registered but the (i-1)-th bin and the (i+1)-th bin are not. Examine Eq. (6)

$$Asd_i \geq rmc_i + thrt \times erstd_i. \tag{7}$$

If this equation holds, then there is an isolated line at the i-th bin. Register the frequency at the bin, calculate the line characteristics, e.g., signal to noise ratio, root mean square amplitude, and so on. Then move to the (i+2)-th bin and examine the step 10.

13. On the contrary again, suppose that i-th bin is not registered in the step 10, (and assume that neither (i-1)-th bin nor (i+1)-th bin is not registered). Examine Eq. (6)

$$Asd_i \geq rmc_i + thrt \times erstd_i. \tag{8}$$

If this equation holds, then there is an isolated line at the i-th bin. Register the frequency at the bin, calculate the line characteristics, e.g., signal to noise ratio, root mean square amplitude, and so on. Then move to the (i+2)-th bin and examine the step 10.

14. *Note on the steps from 10 to 13.*

    - *Note on the steps 10 and 11*: We intend to detect broad lines with these steps. In these steps, we have used running_median_fine instead of the **original Asd**. The reason is as follows. Broad lines in the **original Asd** have often notches and our code sometimes (frequency depending on the threshold_broad) detect many thin lines as well as notches and do not recognize broad lines on which thin lines and notches live. Then to find a cactus (a broad line with many notches on it) not as a bunch of spines but as a cactus, it is found to be helpful to use **smoothed Asd** instead of the **original Asd**.
    - *Note on the steps 10 and 11*: Since broad lines sometimes have relatively small amplitude, we have set threshold_broad relatively small number. This gives us not only broad lines but also a bump. Then the criterion equation (6) in the step 11 helps us select broad lines only.
    - *Note on the step 13.*: In general, it is possible that Eq. (8) holds while Eq. (5) does not. This occurs when $rmf_i >> Asd_i$.

15. Make a database
    The characteristics of the detected lines are collected into the MySQL[1] database.

4 Result

We have analyzed calibrated data for the January data (G1:h). and the S1 data (calibratable channels and G1:DER_H_HP-EP) to investigate the physically meaningful size of the line amplitudes. Here we shall show our results. In the analyais we have used MatLab 6.1 on machines with dual xeon 1.7 GHz and 1 GB memory.

4.1 Application to the January data

The analysis result on the G1:h channel January data is summarized here. Also we show a list of lines database and the graphs of the lines in the appendices A and B.

1. settings

Channel	start time in GPS (sec)	start time in UTC
G1:h	694827000	11-Jan-2002 23:29:47
time stretch (sec)	window	overlap (%)
60	Hanning	50
fr (Hz)	DC removal	calibration
0.1	done	done

   In the `G1:h` January data, we have removed DC with high pass butterworht filter. (We have used the 5th order highpass butterworth with 10 Hz cutoff and *filtfilt* function of *MatLab*.)

   The **hanning window with** 50% **overlap** has been used. Thus we make Fourier transform over $1/fr$ seconds time chunks of the data which has 50 % overlap on the nearest two chunks and take an average of the Fourier transforms of all the chunks in the **nSec** seconds worth data. (See the step 1 in Sec. 3.2.)

   If one wants to measure amplitudes of lines precisely, the HFTxx and HFTxxD windows should be implemented in the code [6].

   We have used *MatLab* pwelch function. Notice that it gives a power spectral density. Take a square root of the power spectral density to obtain an amplitude spectral density.

2. The effective noise band width is 0.15 Hz.

3. free parameters

`blocksize_fine` (Hz)	`blocksize_coarse` (Hz)	coarsing factor
32	1	8
`threshold_broad`	`threshold_thin`	
6	4	

   For January data (`G1:h`), we saw lots of lines which have about 1 Hz bandwidth and have multi-thorns judging by visual inspection. Thus we have set `blocksize_fine` $= 1Hz$.

   In G1:h January data, it seems that some broad lines have width of 10 Hz or so. We have set `blocksize_coarse` 32 Hz to detect such broad lines. Though we have not experimented it seems better to use blocksize of smaller than 50 Hz. If we took, the power lines would affect the noise floor twice.

   We have set **coarsing factor** $= 8$. The so-calculated `running_median_coarse` seems to give a good estimation of noise floor (at least judging by visual inspection).

   We have set `threshold_thin` 6 to detect 100 Hz line and some lines found by quick visual inspection.

4. In the current settings and choice of free parameters, it takes 9 seconds on average. To calculate quartiles, it takes less than 0.2 seconds.

   When we set the frequency resolution finer, say, $1/128Hz$, then the number of data points contained in `blocksize_coarse` (or `blocksize_fine`) becomes bigger. In that case, most of computational burden is from computation of quartiles (, i.e., sort of the data).

5. We have detected 186 lines

4.2 Application to the S1 run

The S1 run started on 23 August 2002 at 15:00 UTC and ended on 09 August 2002 UTC and successfully completed. We have completed the analysis and made a database of the characteristics of the detected lines for the run.

1. settings

Channel	start time in GPS (sec)	start time in UTC
`G1:DER_H_HP-EP`	714321897	25-08-2002 14:44:44
Others	714928813	01-09-2002 15:20:00
time stretch (sec)	window	overlap (%)
60	Hanning	50
fr (Hz)	DC removal	calibration
0.1	No	done

2. The effective noise band width is 0.15 Hz.

3. free parameters

`blocksize_fine` (Hz)	`blocksize_coarse` (Hz)	coarsing factor
32	1	8
`threshold_broad`	`threshold_thin`	
6	4	

   Thus we adopt the same free parameters as the ones taken for the January data. This is because we would like to test to what extent our code with one fixed choice of a parameters set for any channels can work well. As a result, we observe that almost all (a quick visual inspection tells us more than 90 %) lines detected by our code are also confirmed (, i.e., found) by visual inspection.

   For the `G1:PEM_NBR_WIN_S` and `G1:PEM_NBR_WIN_D` channels, however, we have used `blocksize_coarse` = 0.3 Hz, `blocksize_coarse` = 4 Hz, and coarsing factor = 1, since for both channels the amplitude spectral density ranges from 0 Hz to about 60 Hz.

4. The channels of the S1 data which we have analyzed are (`G1:DER_H_HP-EP`) data and the channels for which calibration information was available at the moment when the analysis was done (September 2002). The time series of the data starts at 14:44:44 UTC on 25 August 2002 for `G1:DER_H_HP-EP` channel and at 15:20:00 UTC on 01 August 2002 for the other channels. We took a 1 minutes worth data for each channel. For both of the time the `G1:DER_H_HP-EP` trend data was fairly quiet.

   Here is the list of the channels that we analyzed.

```
G1:DER_H_HP-EP

G1:LSC_BS_PWR-EAST
G1:LSC_MID_EP-P
G1:LSC_MID_EP-P_HP
G1:LSC_MID_FP-MCEI-MCNI
G1:LSC_MID_FP-MCE-MCN
G1:LSC_MID_VIS
G1:LSC_MIC_FP-MMC2B
G1:LSC_MIC_EP
G1:LSC_MIC_VIS
G1:LSC_ML_FP-PZT
G1:LSC_MC1_EP
G1:LSC_MC2_EP

G1:PSL_SL_PWR-LDA
G1:PSL_SL_PWR-AMPL-OUTLP
```

```
G1:SEI_TCIb_ACC-X-ST3
G1:SEI_TCC_SEIS-Y
G1:SEI_TCC_SEIS-X
G1:SEI_TFN_ACC-Z-ST3
G1:SEI_TFN_ACC-Y-ST3
G1:SEI_TFN_ACC-X-ST3
G1:SEI_NBC_SEIS-Y
G1:SEI_NBC_SEIS-X

G1:PEM_NBC_ACOU
G1:PEM_TCC_ACOU-BD
G1:PEM_TFN_ACOU-BD
G1:PEM_TFN_MAG-Z
G1:PEM_TFN_MAG-Y
G1:PEM_TFN_MAG-X
G1:PEM_TCMa_MAG-X
G1:PEM_NBR_WIN-S
G1:PEM_NBR_WIN-D
G1:PEM_NBG_RK-V2
G1:PEM_CBCTR_RK-C-V2
```

5. The result of our analysis for the S1 run is available on the web: `"freq_data"` of “frequencies” on `http://info.geo600.uni-hannover.de/frequencies/phpMyAdmin-2.2.0/`. There one can find the central frequency of the line, the bandwidth and the full width half maximum of the line, the root mean squared amplitude of the line and its unit, signal to noise ratio, and so on.

   Notice that we have first calibrated the amplitude spectral density, then detect lines. Thus, in the database on the web, we have only recorded lines which lie within the range over which calibration information are reliable (The reliable range differs channel by channel). The calibration information including reliable frequency range for each channel is available from the frame data itself.

   Notice that for the S1 data, we have not removed DC.

The following table gives a complete description about the database entries.

column	description
id	Each table entry has its unique Id-number.
channel	The channel name according to the GEO600 naming convention [11].
start_time_UTC	The second at which the analyzed time series begins in the Coordinated Universal Time scale.
start_time_GPS	The second at which the analyzed time series begins in the Global Positioning System scale.
time_stretch_sec	The length of the analyzed time series in seconds.
central_frequency_Hz	The central frequency of the line by calculating the center of the frequency interval that is recognized as a line by the method used.
bandwidth_Hz	The width in Hz of that frequency interval which is recognized as a line by the method used.
FWHM_Hz	The full-width-half-maximum in Hz of the line.
amplitude_rms	The root-mean-square-amplitude of the line which is calculated according to $\sqrt{\frac{1}{2} ENBW \sum_k ASD_k^2}$ where ENBW is the effective noise bandwidth (see column ENBW_Hz), $ASD_k$is the value of the amplitude spectral density in the k -th bin and where k runs over the whole bandwidth of the line.
amplitude_unit	The unit of the root-mean-square-amplitude in the column 'amplitude_rms'.
SNR_dB	The amplitude signal to noise ratio in Decibels calculated according to $20 log_{(10)} \rho$, where ρ is the amplitude signal to ratio calculated according to $\sqrt{\sum_k \frac{ASD_k^2}{N_k^2}}$. Here ASD_k and N_k are the amplitude spectral density and the noise floor in the k-th bin respectively and k runs over the whole bandwidth of the line.
frequency_resolution_Hz	The frequency resolution of the spectrum in Hz.
ENBW_Hz	The effective noise bandwidth in Hz which is necessary to calculate the value of the amplitude spectrum from the amplitude spectral density or vice versa. The ENBW is calculated according to [6].
comments	Here special information to the line and/or channel can be found, e.g. whether the line is a member of a series or whether the channel is calibrated or not. NB: The calibration function is either the whitening filter, the system filter or the product of both according to what is available.
reference	This column contains information connected to the respective table entry. The format is as follows: "author#date#instrument of line search# PSD computation method".

5 Discussion, knownproblems, and summary

5.1 Discussion and knownproblems

1. One known problem of our method is that it is difficult to detect relatively small lines near a broad, steep, large amplitude line, since the running quartile $q3_i$ becomes large near the line. In fact there are

some lines which the code failed to detect. For examle, see the graph in the page 25 which shows two lines around frequency = 3100 Hz. Because of the broad line around frequency = 3117.2 Hz, our code missed one line next to f = 3100 Hz line. One possible way is to use q_1 only and estimate running standard deviation by $2 \times (q2_i - q1_i)/1.34$, where $q2_i$ is the running median at the i-th bin. However, we found that, at least for the S1 run data which we have analyzed, there is little advantage to take this estimator for the standard deviation.

2. Related with the above, at the low frequencies, it is difficult for our code to detect lines. This is because the running median coarse fail to be a good estimator of a noise floor at the low frequencies and the running median fine fail to be a good estimator of a smoothed Asd at the low frequencies. This seems mainly due to the fact that `blocksize_fine` and `blocksize_coarse` are too large compared to the variation of the original Asd.

3. Our code has detected some lines which visual inspection could not confirm. For example, see the graphs in the page 21 which show "lines" around frequencies of 1250.75 Hz and 1251 Hz.

4. There is other kind of lines which our code has detected while visual inspection could not confirm. For example, see the graphs in the page 21 which show "lines" around frequencies of 1095.8 Hz and 1116.9 Hz.

5. For the above two issue, the four free parameters, `blocksize_fine`, `blocksize_coarse`, `threshold_broad`, and `threshold_thin` need to be tuned. Especially, we observe that it seems better to take a smaller value for `blocksize_fine` than present choice (1 Hz).

6. It is interesting to note that our code detect lines whose siginal to noise ratio is about 2. See for example, the graphs in the page 24 which shows lines with frequencies of 2369.75 Hz. This line has signal to ratio of 2.1.

7. In the present algorithm, we detect broad lines through two steps. (See the steps 10 and 11 in Sec. 3.2.) With the first step, it is possible that we mistakenly detect bumps. Thus in the second step we demand that at least one bin in the broad line candidate has a enough amplitude to be a line.

   A possible other way is as follows. For instance, if an Asd around certain bin has *line-like form*, then we detect it as a line. We had once used Q-value as a criterion for *line-like form*, somehow it had not worked well.

8. We do not know how to estimate number of lines which we have failed to detect.

5.2 Summary

We have proposed an automated line search algorithm and shown the analysis reault on the January data and the S1 run data. The algorithm is implemented in the MatLab code. We have used the same parameters set for both January and the S1 data. As a result, we observe that almost all (a quick visual inspection tells us more than 90 %) lines detected by our code are also confirmed (, i.e., found) by visual inspection. Although the four free parameters of the present algorithm need to be tuned, it is encourageous to see that one paticular parameter set can almost successfully work on various channels and different time stretches.

6 Future look

The resulting database may be used as a initial input for a online line detection monitor as well as line tracking codes as a piece of detector characterization pipelines such as rDCR/DCR[4]. Also the resulting database may be used in such a way that in a gravitational wave search people avoid to analyze/use frequency bands where the lines are known to exist from the database.

7 Terminology

1. **ENBW**: Effective noise band width. Also called effective noise-equivalent bandwidth and equivalent noise bandwidth in Ref [6]. ENBW depends only on a frequency resolution (fr) and a window function used. Suppose that we use a window function expressed as a set of N real numbers $\{w_j\}$ with $j = 0..N-1$ in the time domain. N is the length of the DFT. Define two sums by

$$S_1 = \sum_{j=0}^{N-1} w_j, \tag{9}$$

$$S_2 = \sum_{j=0}^{N-1} w_j^2. \tag{10}$$

   Then normalized equivalent noise bandwidth (NENBW) is defined as

$$\mathrm{NENBW} = N\frac{S_2}{S_1^2}, \tag{11}$$

   and effective noise bandwidth (ENBW) is defined as

$$\mathrm{ENBW} = \mathrm{NENBW}\cdot fr = \mathrm{NENBW}\frac{f_s}{N} = f_s\frac{S_2}{S_1^2}. \tag{12}$$

   Here f_s is the sampling frequency of the time series.

2. **quartile**: Imagine that we have a set of data y_i, $i = 1, 2, ..., N$. y_i's are supposed to be ordinable. Sort y_i in ascending order and denote the ordered set $\tilde{y}_i$. Then

$$(first\,quartile) = \tilde{y}_{N/4} \tag{13}$$

$$(second\,quartile) = \tilde{y}_{N/2} = (median) \tag{14}$$

$$(third\,quartile) = \tilde{y}_{3*N/4}. \tag{15}$$

   (This is *NOT* an exact definition. $N/4$ is assumed to be an integer. If not, "$N/4$" is set to be the nearest integer of $N/4$. If the number of data points is large and dense, do not bather.)

   50 % of the sorted data points exist between the *third quartile* and *first quartile*. If the distribution of the data set y_i is normal, then the standard deviation and the quartiles are related via

$$(standard\,deviation) = ((third\,quartile) - (first\,quartile))/1.34. \tag{16}$$

3. **q-value**:

$$(q - value) = \frac{center\ frequency}{full\ width\ half\ maximum} \tag{17}$$

   When **full width half maximum** $= 0$, we use

$$(q - value) = \frac{center\ frequency}{effective\ noise\ band\ width}. \tag{18}$$

4. **root mean squared amplitude**:

   For a thin line at the i-th bin,

$$
\begin{aligned}
&(root\,mean\,squared\,amplitude\,for\,a\,thin\,line) \\
&= \sqrt{\frac{ENBW}{2}} Asd_i.
\end{aligned} \tag{19}
$$

   For a broad line

$$
\begin{aligned}
&(root\,mean\,squared\,amplitude\,for\,a\,broad\,line) \\
&= \sqrt{\sum_{i \in the\,line} \frac{ENBW}{2} (Asd_i)^2}.
\end{aligned} \tag{20}
$$

5. **running median**: Imagine that we have sets of data pair (x_i, y_i), $i = 1, 2, 3, ..., N$. Suppose that we can set both x_i and y_i in order independently. Then the running median at x_j over some blocksize Δ is the median of the set of data $y_{j-k}, y_{j-k+1}, ..., y_{j-1}, y_j, y_{j+1}, ...y_{j+k}$, where the integer k is defined by $x_{j+k} - x_{j-k} \le \Delta \le x_{j+k+1} - x_{j-k-1}$. To calculate running medians of the data points whose indices are either $j \le k$ or $N - j \le k$, suitable padding to the both ends of data should be made.

   The padding we have chosen are zero padding. The resulting running median has the same dimension (length) as the original data set $\{y_i\}$ See also the help of *medfilt1* function of *MatLab*.

6. **running quartile**: ”running” means the same as the running median. See also quartile.

7. **signal to noise ratio**:

   For a thin line at the i-th bin,

$$
(signal\,to\,noise\,ratio\,for\,a\,thin\,line) = \frac{Asd_i}{rmc_i}. \tag{21}
$$

   For a broad line

$$
(signal\,to\,noise\,ratio\,for\,a\,broad\,line) = \sqrt{\frac{\sum_{i \in the\,line} (Asd_i)^2}{\sum_{j \in the\,line} (rmc_j)^2}}. \tag{22}
$$

   Noise floor is always estimated by *running mean coarse*.

8. **width of a line**: Given that consecutive bins are registered as bins on which lines exist. Then the **width of the line** is defined by *largest bin* minus *smallest bin* in the registered bins. That is, *largest frequency* minus *smallest frequency*.

References

[1] *MySQL*, `http://www.mysql.com/`, Especially the chapter 3 on `http://www.mysql.com/documentation/mysql/bychapter/` for the tutorial introduction.

[2] Our analysis result on the S1 data as well as the visual inspection results on the January data are collected in the MySQL[1] frequencies database called `"freq_data"` of “frequencies”. It is found on the web: `http://info.geo600.uni-hannover.de/frequencies/phpMyAdmin-2.2.0/`.

[3] The use of the running median for robust definition of the noise floor is due to S. Mohanty.

[4] S. D. Mohanty, Class. Quant. Grav. **19**, 1513 (2002).

[5] S. D. Mohanty and S. Mukherjee, Class. Quant. Grav. **19**, 1471 (2002).

[6] *Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows*, G. Henzel, A. Rüdiger, and R. Schilling, 2002.

[7] MatLab, a technical computing environment for high-performance numerical computations in linear algebra, is a product of The MathWorks Inc.

[8] *GEO-Tools version 0.1.5*, Martin Hewitson, 2002. See http://www.astro.gla.ac.uk/users/hewitson/downloads.html.

[9] http://www.astro.cf.ac.uk/pub/R.Balasubramanian/geo++/index.htm. *GEO++* is a software centered around C++ which will provide the necessary framework to carry out GEO data analysis and detector characterization tasks. See the above web site for further details.

[10] Welch, P. D. *The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodogram*, IEEE Trans. Audio Electroacoustics, Vol. AU-15 (June 1967), pp 70-73.

[11] *GEO Naming Convention*, https://info.geo600.uni-hannover.de/naming.

A Line list

In each raw of the lists, line id, channel name, (center) frequency, line width (Hz), FWHM (Hz), rms amplitude, unit of Asd, SNR (dB), SNR (Linear) are shown in this order.

```
/* threshold_broad = 4.000000, threshold_thin = 6.000000 */
/* startf = 0.000000, stopf = 8192.000000 */
/* id,channel, */
/* central_frequency_Hz,bandwidth_Hz,FWHM_Hz, */
/* amplitude_rms,amplitude_unit,SNR_dB, */
/* SNR_Linear  */


/* number of lines 186  */
    1, G1:h,    0.00,  0.00,  0.00, 0.0000e+00, m,  0.00,    1.0
    2, G1:h,    8.15,  0.10,  0.10, 7.3920e-17, m, 30.98,   35.4
    3, G1:h,    8.65,  0.10,  0.10, 6.5532e-17, m, 25.20,   18.2
    4, G1:h,   31.85,  1.30,  0.50, 5.2261e-16, m, 19.85,    9.8
    5, G1:h,   43.00,  0.20,  0.10, 4.5011e-16, m, 23.48,   14.9
    6, G1:h,   50.05,  0.30,  0.10, 1.1267e-15, m, 29.56,   30.0
    7, G1:h,  100.10,  0.10,  0.00, 9.8287e-17, m, 10.17,    3.2
    8, G1:h,  150.10,  0.20,  0.10, 1.4699e-16, m, 14.38,    5.2
    9, G1:h,  198.35,  1.30,  1.30, 9.9711e-17, m,  7.18,    2.3
   10, G1:h,  250.20,  0.20,  0.10, 8.3287e-17, m, 15.06,    5.7
   11, G1:h,  450.30,  0.10,  0.00, 1.3062e-17, m,  5.64,    1.9
```

12,	G1:h,	550.40,	0.10,	0.00,	1.6511e-17,	m,	7.96,	2.5
13,	G1:h,	779.40,	1.00,	0.50,	7.5022e-17,	m,	13.05,	4.5
14,	G1:h,	950.75,	0.10,	0.10,	1.1233e-17,	m,	8.41,	2.6
15,	G1:h,	1050.80,	0.10,	0.00,	7.4459e-18,	m,	9.57,	3.0
16,	G1:h,	1095.80,	0.40,	0.40,	8.6222e-18,	m,	6.47,	2.1
17,	G1:h,	1116.90,	0.10,	0.00,	4.8755e-18,	m,	8.18,	2.6
18,	G1:h,	1200.90,	0.20,	0.20,	8.4873e-18,	m,	11.78,	3.9
19,	G1:h,	1250.75,	0.10,	0.10,	5.2439e-18,	m,	10.78,	3.5
20,	G1:h,	1251.00,	0.20,	0.20,	1.1458e-17,	m,	15.82,	6.2
21,	G1:h,	1300.95,	0.30,	0.30,	7.1890e-18,	m,	12.71,	4.3
22,	G1:h,	1350.85,	0.10,	0.10,	3.2720e-18,	m,	10.79,	3.5
23,	G1:h,	1351.10,	0.20,	0.20,	6.4845e-18,	m,	14.97,	5.6
24,	G1:h,	1400.00,	0.10,	0.00,	2.3847e-18,	m,	13.00,	4.5
25,	G1:h,	1401.05,	0.30,	0.30,	7.8990e-18,	m,	17.43,	7.4
26,	G1:h,	1451.25,	0.90,	0.20,	9.2550e-18,	m,	15.92,	6.3
27,	G1:h,	1500.00,	0.10,	0.00,	1.9058e-18,	m,	13.28,	4.6
28,	G1:h,	1501.20,	0.60,	0.30,	9.3689e-18,	m,	18.67,	8.6
29,	G1:h,	1558.65,	3.90,	1.00,	2.7557e-16,	m,	39.86,	98.4
30,	G1:h,	1600.00,	0.20,	0.10,	2.3131e-18,	m,	12.22,	4.1
31,	G1:h,	1601.20,	0.60,	0.30,	4.8022e-18,	m,	14.59,	5.4
32,	G1:h,	1602.10,	0.10,	0.00,	9.3545e-19,	m,	8.78,	2.7
33,	G1:h,	1648.60,	0.10,	0.00,	6.4566e-19,	m,	6.67,	2.2
34,	G1:h,	1649.55,	0.10,	0.10,	1.0945e-18,	m,	8.46,	2.6
35,	G1:h,	1650.35,	0.10,	0.10,	9.9355e-19,	m,	7.68,	2.4
36,	G1:h,	1651.25,	1.10,	0.40,	2.5139e-18,	m,	7.95,	2.5
37,	G1:h,	1652.10,	0.20,	0.20,	1.3461e-18,	m,	8.51,	2.7
38,	G1:h,	1653.00,	0.20,	0.20,	1.1917e-18,	m,	7.47,	2.4
39,	G1:h,	1700.00,	0.20,	0.10,	2.3296e-18,	m,	13.51,	4.7
40,	G1:h,	1701.30,	0.80,	0.30,	5.8856e-18,	m,	16.55,	6.7
41,	G1:h,	1750.50,	0.10,	0.00,	6.0932e-19,	m,	6.32,	2.1
42,	G1:h,	1751.15,	0.70,	0.40,	1.8629e-18,	m,	7.17,	2.3
43,	G1:h,	1800.00,	0.20,	0.10,	2.0893e-18,	m,	12.87,	4.4
44,	G1:h,	1801.35,	0.90,	0.30,	5.0286e-18,	m,	15.19,	5.7
45,	G1:h,	1851.45,	0.70,	0.40,	3.1208e-18,	m,	12.15,	4.0
46,	G1:h,	1900.00,	0.20,	0.10,	1.8794e-18,	m,	11.98,	4.0
47,	G1:h,	1901.45,	0.70,	0.30,	2.8911e-18,	m,	11.36,	3.7
48,	G1:h,	1950.60,	0.10,	0.00,	5.4471e-19,	m,	6.50,	2.1
49,	G1:h,	1951.50,	1.40,	0.40,	3.2627e-18,	m,	10.29,	3.3
50,	G1:h,	1952.35,	0.10,	0.10,	7.1554e-19,	m,	5.85,	2.0
51,	G1:h,	2000.00,	0.20,	0.20,	1.9751e-18,	m,	13.15,	4.5
52,	G1:h,	2001.50,	0.60,	0.30,	3.4830e-18,	m,	14.23,	5.1
53,	G1:h,	2048.10,	0.20,	0.20,	1.8394e-18,	m,	11.84,	3.9
54,	G1:h,	2051.60,	0.40,	0.30,	2.1157e-18,	m,	11.24,	3.6
55,	G1:h,	2100.00,	0.20,	0.10,	1.7928e-18,	m,	11.32,	3.7
56,	G1:h,	2101.65,	0.30,	0.30,	1.8953e-18,	m,	10.57,	3.4
57,	G1:h,	2110.15,	1.50,	0.20,	8.0822e-18,	m,	17.26,	7.3
58,	G1:h,	2151.60,	0.60,	0.30,	1.8889e-18,	m,	9.05,	2.8
59,	G1:h,	2200.00,	0.20,	0.20,	1.5137e-18,	m,	10.58,	3.4

60	G1:h	2201.65	0.50	0.50	1.7411e-18	m	8.83	2.8
61	G1:h	2251.70	0.20	0.20	1.2238e-18	m	8.66	2.7
62	G1:h	2300.00	0.20	0.10	1.7087e-18	m	11.38	3.7
63	G1:h	2301.80	0.20	0.20	1.2063e-18	m	8.49	2.7
64	G1:h	2305.75	0.70	0.70	1.6454e-18	m	7.49	2.4
65	G1:h	2337.90	4.60	1.50	4.7721e-16	m	43.21	144.8
66	G1:h	2369.75	1.10	1.00	1.8365e-18	m	6.50	2.1
67	G1:h	2400.00	0.20	0.20	1.5319e-18	m	10.95	3.5
68	G1:h	2401.75	0.70	0.50	2.0166e-18	m	9.08	2.8
69	G1:h	2432.85	0.50	0.20	3.7677e-18	m	15.80	6.2
70	G1:h	2451.50	0.10	0.00	6.0435e-19	m	7.66	2.4
71	G1:h	2451.90	0.60	0.40	2.8596e-18	m	12.72	4.3
72	G1:h	2500.00	0.20	0.20	1.4651e-18	m	10.88	3.5
73	G1:h	2501.40	0.10	0.00	5.0420e-19	m	6.42	2.1
74	G1:h	2501.85	0.70	0.70	1.7046e-18	m	7.98	2.5
75	G1:h	2546.00	0.10	0.00	5.0872e-19	m	6.72	2.2
76	G1:h	2551.95	0.70	0.60	1.9346e-18	m	9.33	2.9
77	G1:h	2600.00	0.20	0.20	1.3778e-18	m	10.39	3.3
78	G1:h	2601.50	0.10	0.00	5.9140e-19	m	7.81	2.5
79	G1:h	2601.95	0.70	0.60	2.1905e-18	m	10.11	3.2
80	G1:h	2608.65	0.70	0.40	5.6932e-18	m	18.01	7.9
81	G1:h	2651.50	0.10	0.00	5.2112e-19	m	6.85	2.2
82	G1:h	2652.05	0.90	0.50	2.5731e-18	m	10.76	3.5
83	G1:h	2700.00	0.20	0.20	1.3215e-18	m	10.33	3.3
84	G1:h	2702.10	1.20	0.80	2.3699e-18	m	9.00	2.8
85	G1:h	2703.00	0.10	0.00	5.1045e-19	m	6.71	2.2
86	G1:h	2752.05	0.90	0.60	2.8850e-18	m	12.54	4.2
87	G1:h	2799.95	0.10	0.10	1.2543e-18	m	11.31	3.7
88	G1:h	2801.90	3.60	0.40	5.7068e-18	m	11.81	3.9
89	G1:h	2852.15	0.90	0.70	2.6161e-18	m	11.39	3.7
90	G1:h	2900.00	0.20	0.20	1.2798e-18	m	10.48	3.3
91	G1:h	2902.25	0.10	0.10	8.8023e-19	m	8.99	2.8
92	G1:h	2952.25	1.30	0.50	3.4141e-18	m	12.55	4.2
93	G1:h	2999.95	0.10	0.10	1.0600e-18	m	10.62	3.4
94	G1:h	3002.35	1.70	0.40	4.1166e-18	m	12.64	4.3
95	G1:h	3052.20	1.20	0.80	3.1079e-18	m	11.95	4.0
96	G1:h	3100.00	0.10	0.00	8.7775e-19	m	11.46	3.7
97	G1:h	3117.20	4.60	2.00	1.0232e-16	m	34.38	52.3
98	G1:h	3152.40	1.20	0.80	2.2639e-18	m	9.55	3.0
99	G1:h	3199.95	0.10	0.10	1.1586e-18	m	11.21	3.6
100	G1:h	3202.30	1.00	0.90	2.2427e-18	m	9.42	3.0
101	G1:h	3252.25	1.50	0.80	3.2334e-18	m	10.92	3.5
102	G1:h	3287.15	0.10	0.10	9.3400e-19	m	9.56	3.0
103	G1:h	3299.95	0.10	0.10	1.0518e-18	m	10.06	3.2
104	G1:h	3302.50	2.00	0.90	3.1689e-18	m	9.18	2.9
105	G1:h	3352.40	1.20	1.00	2.2703e-18	m	9.11	2.9
106	G1:h	3399.95	0.10	0.10	1.0553e-18	m	10.73	3.4
107	G1:h	3402.50	1.00	0.90	2.0031e-18	m	8.95	2.8

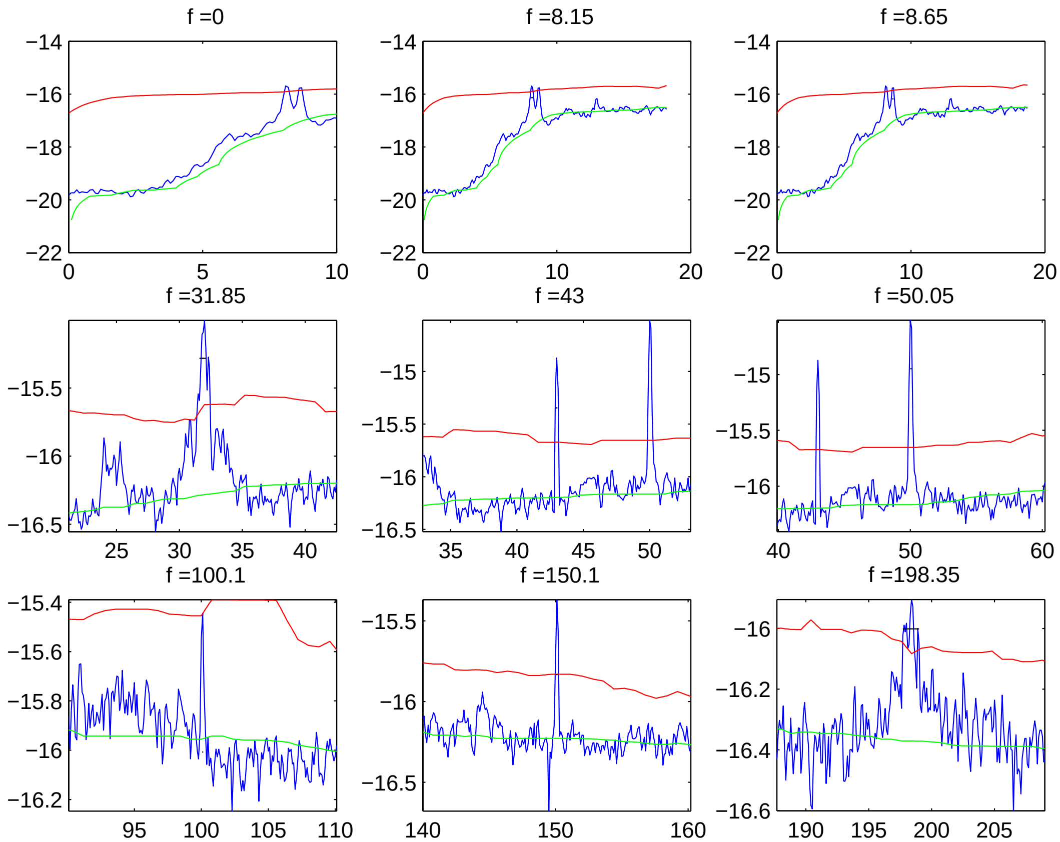
108,	G1:h,	3452.55,	1.10,	0.80,	2.5954e-18,	m,	10.77,	3.5
109,	G1:h,	3500.00,	0.20,	0.20,	1.1441e-18,	m,	9.78,	3.1
110,	G1:h,	3502.50,	1.40,	1.00,	2.8647e-18,	m,	10.55,	3.4
111,	G1:h,	3552.65,	1.10,	1.00,	2.2122e-18,	m,	9.13,	2.9
112,	G1:h,	3600.00,	0.20,	0.10,	1.1267e-18,	m,	8.93,	2.8
113,	G1:h,	3602.80,	1.80,	0.90,	3.7363e-18,	m,	11.30,	3.7
114,	G1:h,	3652.90,	0.10,	0.00,	6.1521e-19,	m,	9.30,	2.9
115,	G1:h,	3699.95,	0.10,	0.10,	9.6100e-19,	m,	9.70,	3.1
116,	G1:h,	3702.80,	0.80,	0.80,	1.4856e-18,	m,	6.90,	2.2
117,	G1:h,	3752.85,	0.70,	0.70,	1.1898e-18,	m,	6.05,	2.0
118,	G1:h,	3800.00,	0.20,	0.20,	1.1500e-18,	m,	9.52,	3.0
119,	G1:h,	3802.85,	1.70,	0.70,	3.2862e-18,	m,	10.90,	3.5
120,	G1:h,	3852.40,	0.10,	0.00,	4.4824e-19,	m,	6.32,	2.1
121,	G1:h,	3852.95,	0.70,	0.60,	1.7023e-18,	m,	8.92,	2.8
122,	G1:h,	3896.40,	5.40,	2.60,	1.0209e-16,	m,	35.50,	59.5
123,	G1:h,	4000.00,	0.20,	0.10,	1.0895e-18,	m,	9.42,	3.0
124,	G1:h,	4003.15,	0.50,	0.50,	8.9416e-19,	m,	4.78,	1.7
125,	G1:h,	4052.50,	0.10,	0.00,	3.7831e-19,	m,	5.44,	1.9
126,	G1:h,	4053.10,	1.00,	0.80,	1.6582e-18,	m,	7.82,	2.5
127,	G1:h,	4094.40,	0.10,	0.00,	6.4566e-19,	m,	9.78,	3.1
128,	G1:h,	4096.20,	0.20,	0.10,	2.8218e-18,	m,	17.56,	7.6
129,	G1:h,	4097.95,	0.10,	0.10,	8.3011e-19,	m,	8.69,	2.7
130,	G1:h,	4099.95,	0.10,	0.10,	8.4609e-19,	m,	8.64,	2.7
131,	G1:h,	4200.00,	0.10,	0.00,	7.5038e-19,	m,	11.23,	3.6
132,	G1:h,	4202.60,	0.10,	0.00,	5.0555e-19,	m,	7.77,	2.4
133,	G1:h,	4203.20,	1.00,	0.90,	1.7522e-18,	m,	8.06,	2.5
134,	G1:h,	4220.00,	0.20,	0.10,	2.0178e-18,	m,	15.38,	5.9
135,	G1:h,	4253.30,	0.60,	0.60,	1.1869e-18,	m,	6.82,	2.2
136,	G1:h,	4300.05,	0.10,	0.10,	7.9742e-19,	m,	9.13,	2.9
137,	G1:h,	4303.30,	1.00,	1.00,	1.5379e-18,	m,	7.40,	2.3
138,	G1:h,	4352.70,	0.10,	0.00,	3.6778e-19,	m,	5.77,	1.9
139,	G1:h,	4353.25,	0.70,	0.60,	1.0818e-18,	m,	6.11,	2.0
140,	G1:h,	4399.95,	0.10,	0.10,	8.2859e-19,	m,	9.43,	3.0
141,	G1:h,	4500.00,	0.10,	0.00,	5.2187e-19,	m,	8.72,	2.7
142,	G1:h,	4502.70,	0.10,	0.00,	4.8060e-19,	m,	7.76,	2.4
143,	G1:h,	4503.40,	1.00,	0.60,	1.8069e-18,	m,	8.83,	2.8
144,	G1:h,	4553.45,	0.70,	0.60,	9.9465e-19,	m,	5.43,	1.9
145,	G1:h,	4600.00,	0.10,	0.00,	6.2452e-19,	m,	10.18,	3.2
146,	G1:h,	4675.65,	5.30,	3.10,	3.3632e-17,	m,	26.39,	20.9
147,	G1:h,	4690.00,	0.10,	0.00,	6.2223e-19,	m,	9.32,	2.9
148,	G1:h,	4700.00,	0.10,	0.00,	6.1567e-19,	m,	9.78,	3.1
149,	G1:h,	4753.80,	0.10,	0.00,	4.4881e-19,	m,	7.69,	2.4
150,	G1:h,	4800.00,	0.10,	0.00,	5.0923e-19,	m,	8.49,	2.7
151,	G1:h,	4803.75,	0.50,	0.50,	1.0719e-18,	m,	6.93,	2.2
152,	G1:h,	4865.70,	0.10,	0.00,	3.6132e-19,	m,	6.02,	2.0
153,	G1:h,	4900.00,	0.10,	0.00,	5.5191e-19,	m,	9.02,	2.8
154,	G1:h,	5000.00,	0.10,	0.00,	4.8856e-19,	m,	8.65,	2.7
155,	G1:h,	5003.85,	0.10,	0.10,	4.5056e-19,	m,	4.94,	1.8

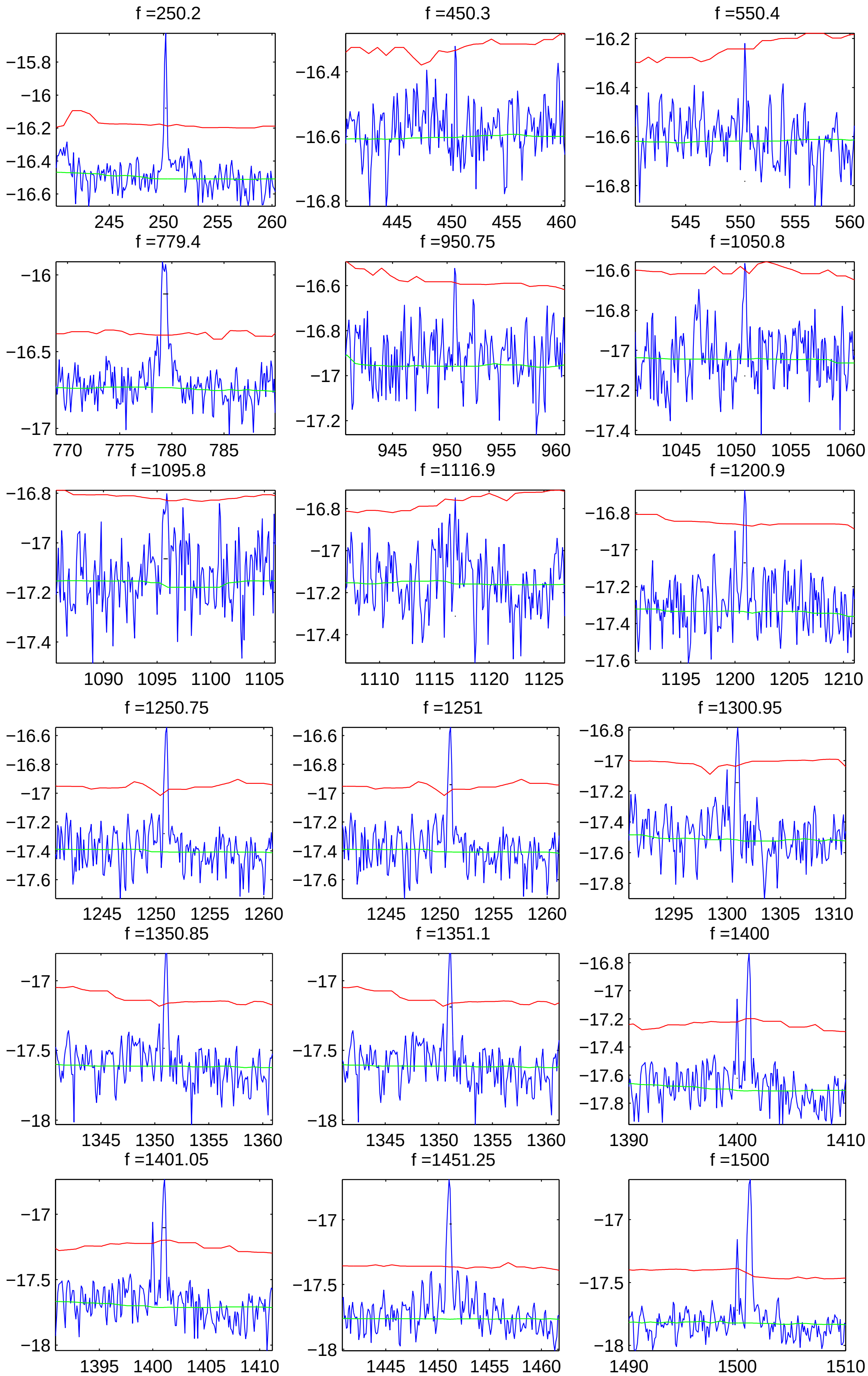
```
156, G1:h, 5100.00,  0.10,  0.00, 3.7565e-19, m,  6.01,    2.0
157, G1:h, 5160.00,  0.10,  0.00, 4.0361e-19, m,  6.63,    2.1
158, G1:h, 5200.00,  0.10,  0.00, 4.3844e-19, m,  7.38,    2.3
159, G1:h, 5217.35,  1.10,  0.70, 3.5859e-18, m, 14.67,    5.4
160, G1:h, 5300.00,  0.10,  0.00, 4.2397e-19, m,  7.52,    2.4
161, G1:h, 5400.00,  0.10,  0.00, 3.4806e-19, m,  5.35,    1.9
162, G1:h, 5454.85,  5.70,  3.50, 5.6910e-17, m, 30.86,   34.9
163, G1:h, 5504.40,  0.10,  0.00, 3.4563e-19, m,  5.41,    1.9
164, G1:h, 5553.95,  0.10,  0.10, 4.7480e-19, m,  6.10,    2.0
165, G1:h, 5554.25,  0.30,  0.30, 5.3287e-19, m,  4.09,    1.6
166, G1:h, 5723.10,  0.10,  0.00, 3.7682e-19, m,  6.44,    2.1
167, G1:h, 5800.00,  0.10,  0.00, 4.1578e-19, m,  7.54,    2.4
168, G1:h, 5900.00,  0.10,  0.00, 3.5030e-19, m,  6.10,    2.0
169, G1:h, 5973.60,  0.10,  0.00, 5.0917e-19, m,  9.11,    2.9
170, G1:h, 6114.80,  0.10,  0.00, 2.9799e-19, m,  4.74,    1.7
171, G1:h, 6142.50,  0.20,  0.20, 1.0439e-18, m, 10.22,    3.2
172, G1:h, 6144.30,  0.20,  0.10, 2.6442e-18, m, 18.28,    8.2
173, G1:h, 6146.05,  0.10,  0.10, 1.1021e-18, m, 12.50,    4.2
174, G1:h, 6163.65,  0.10,  0.10, 8.4620e-19, m, 10.46,    3.3
175, G1:h, 6175.05,  0.10,  0.10, 5.9998e-19, m,  7.40,    2.3
176, G1:h, 6234.35,  5.30,  4.10, 1.4650e-17, m, 20.72,   10.9
177, G1:h, 6300.00,  0.10,  0.00, 3.3356e-19, m,  5.66,    1.9
178, G1:h, 6330.00,  0.10,  0.00, 3.9657e-19, m,  6.80,    2.2
179, G1:h, 6524.60,  0.10,  0.00, 3.2641e-19, m,  5.34,    1.8
180, G1:h, 6592.90,  0.10,  0.00, 3.5242e-19, m,  6.06,    2.0
181, G1:h, 6604.25,  0.10,  0.10, 1.4075e-18, m, 15.06,    5.7
182, G1:h, 6615.65,  0.10,  0.10, 9.7669e-19, m, 11.73,    3.9
183, G1:h, 6627.00,  0.10,  0.00, 4.2812e-19, m,  6.63,    2.1
184, G1:h, 6800.00,  0.20,  0.20, 2.5063e-18, m, 18.55,    8.5
185, G1:h, 7013.60,  5.40,  4.60, 1.3402e-17, m, 20.85,   11.0
186, G1:h, 7049.40,  0.10,  0.00, 3.1080e-19, m,  6.40,    2.1
```

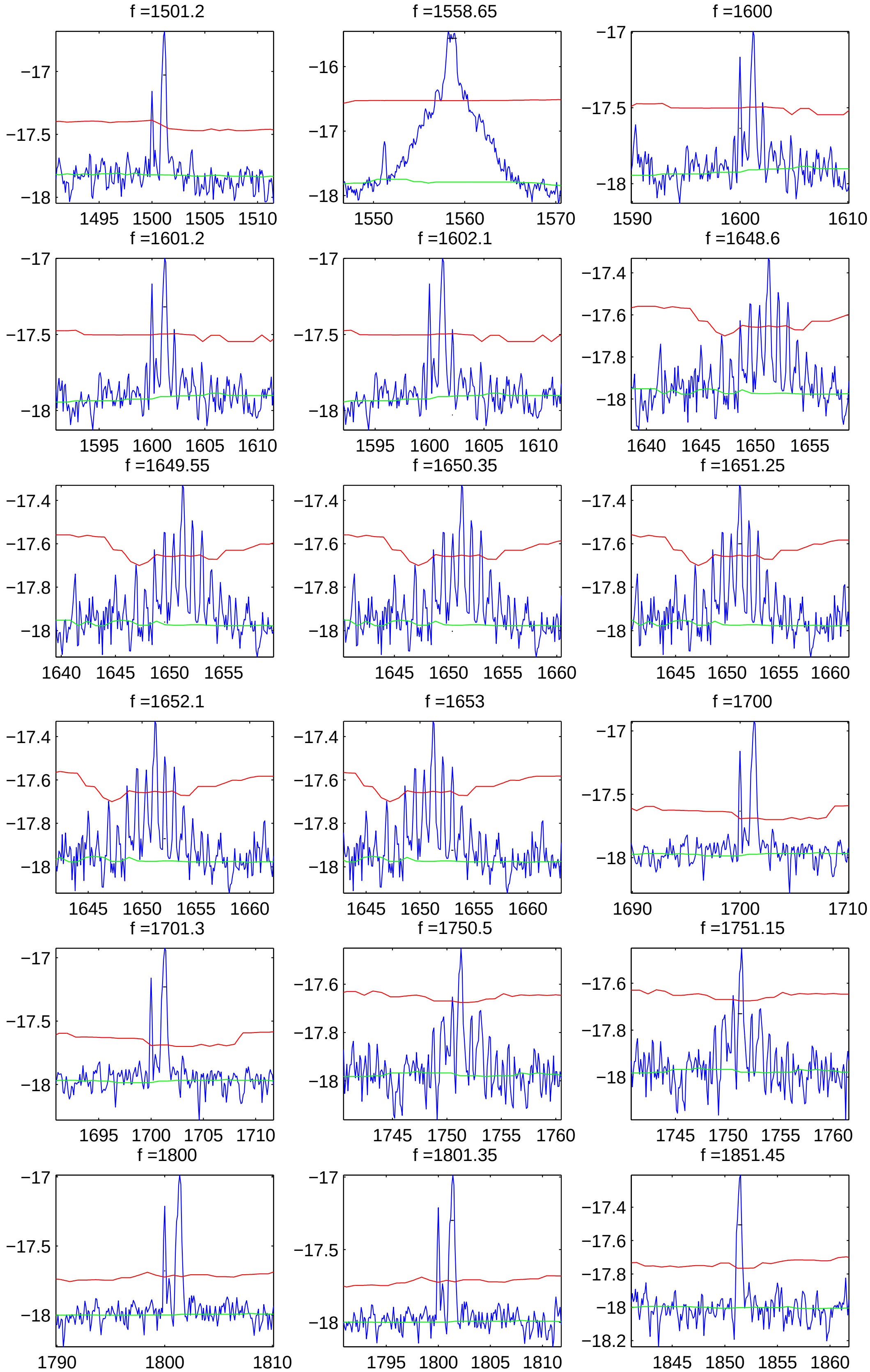
B Graphs

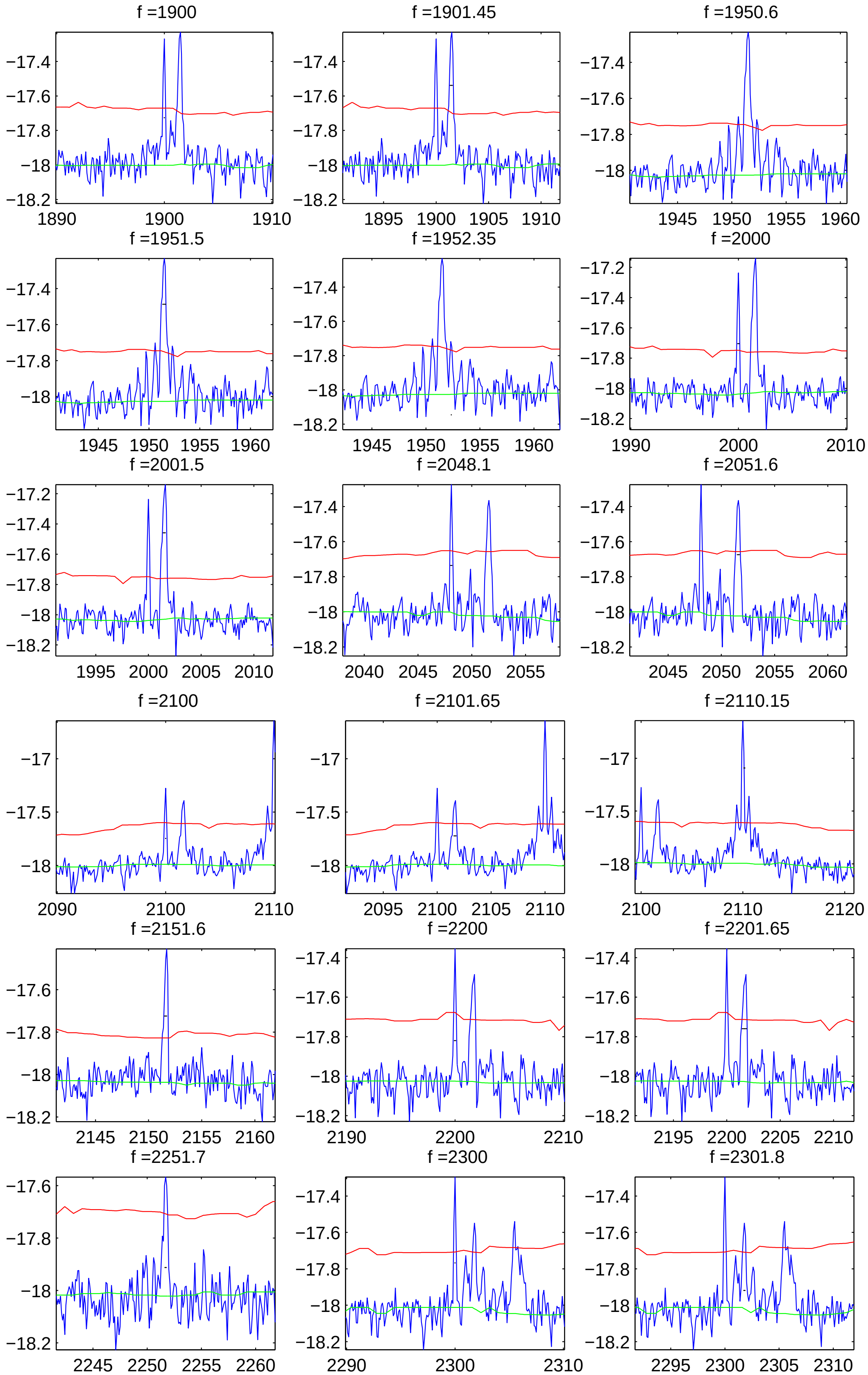
In the 186 graphs, we show lines detected by our code. Title of each figure shows the frequency of the thin line. Green lines are running medians over 32 Hz. Red line is the threshold line. The threshold for lines is running median plus 6 times running standard deviation. The running standard deviation is estimated by the interquartile-ranges calculated over 32 Hz and have value on each bin.

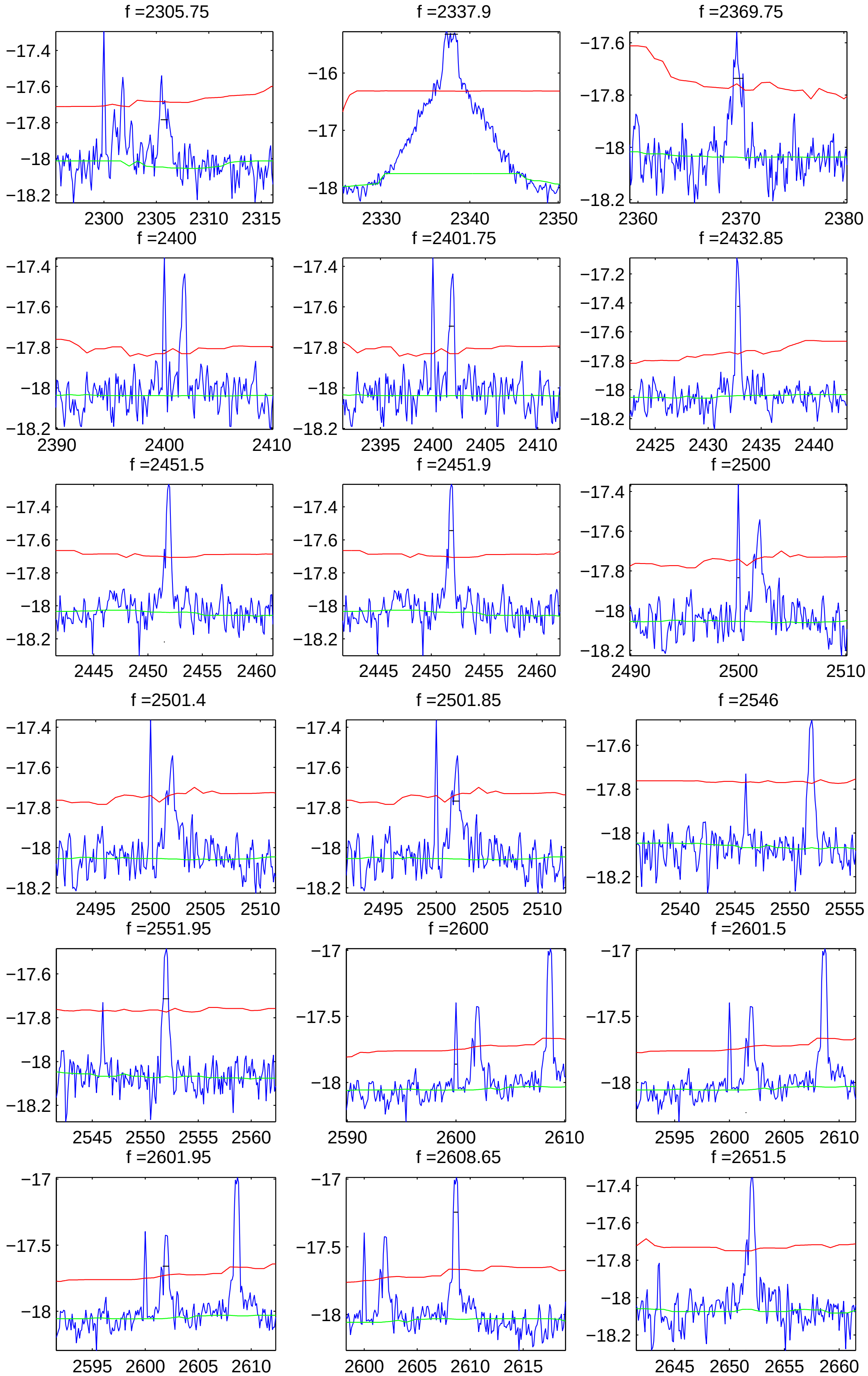
Figure 1: Detected lines

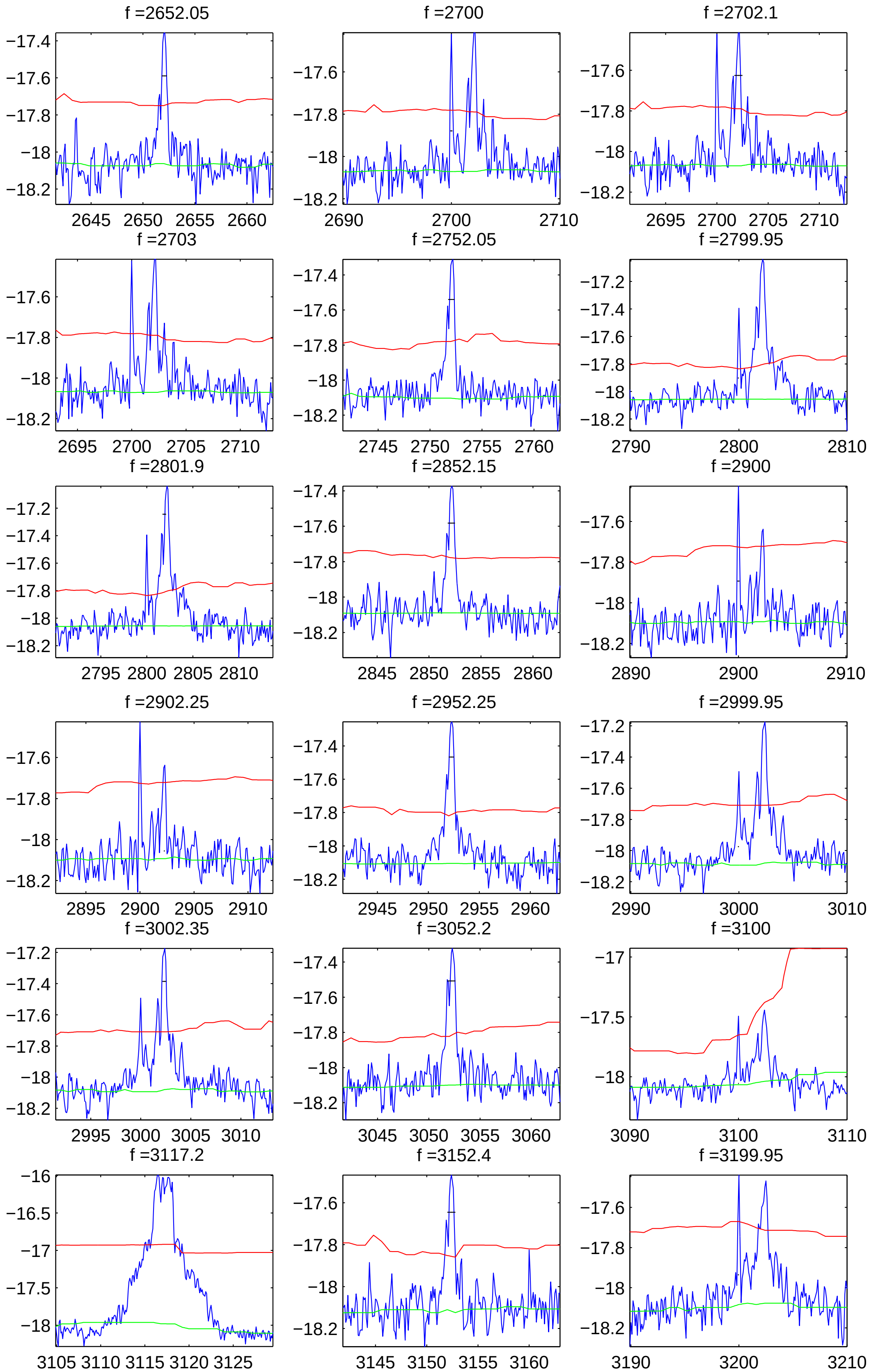


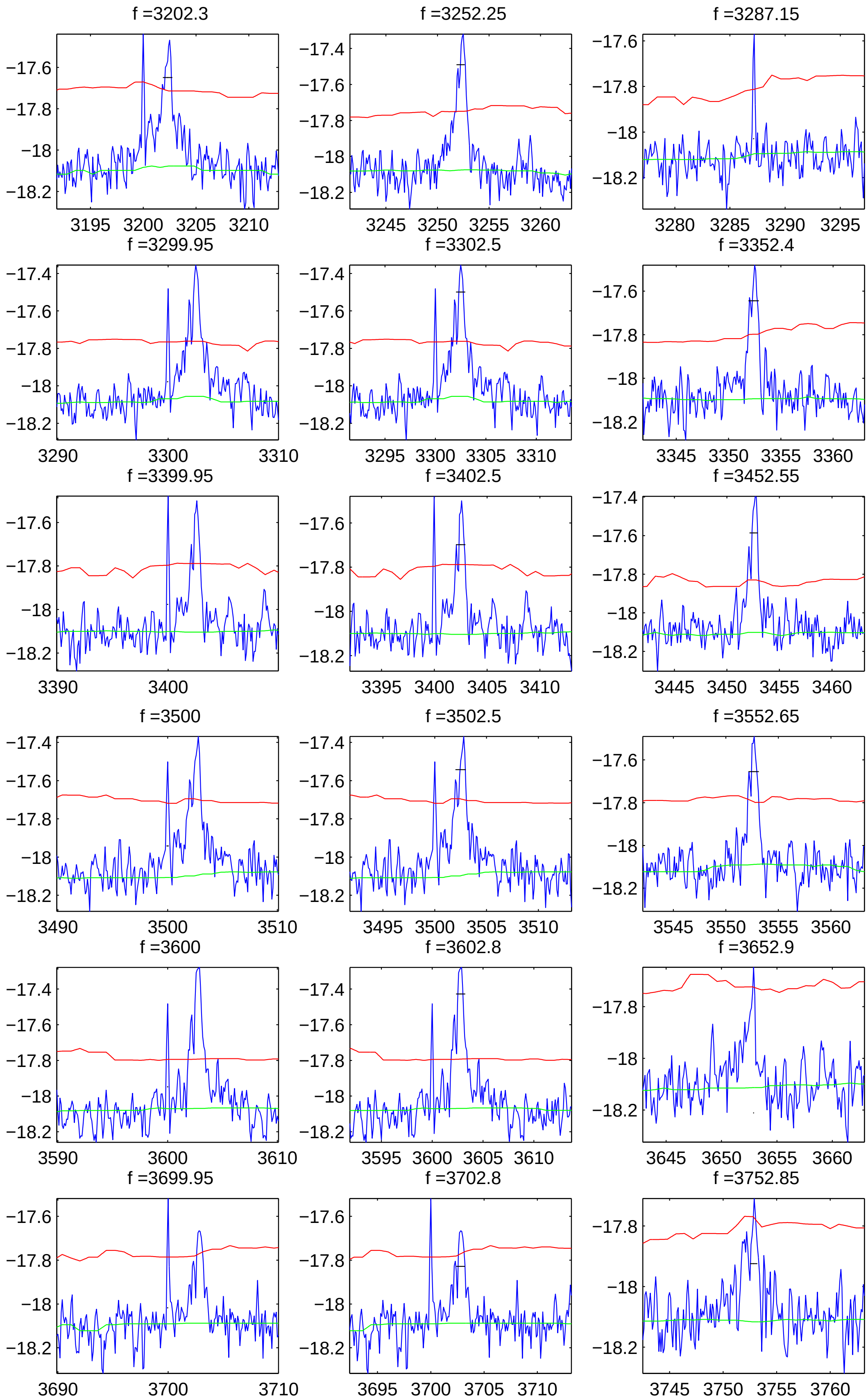

f =250.2
f =450.3
f =550.4
f =779.4
f =950.75
f =1050.8
f =1095.8
f =1116.9
f =1200.9
f =1250.75
f =1251
f =1300.95
f =1350.85
f =1351.1
f =1400
f =1401.05
f =1451.25
f =1500

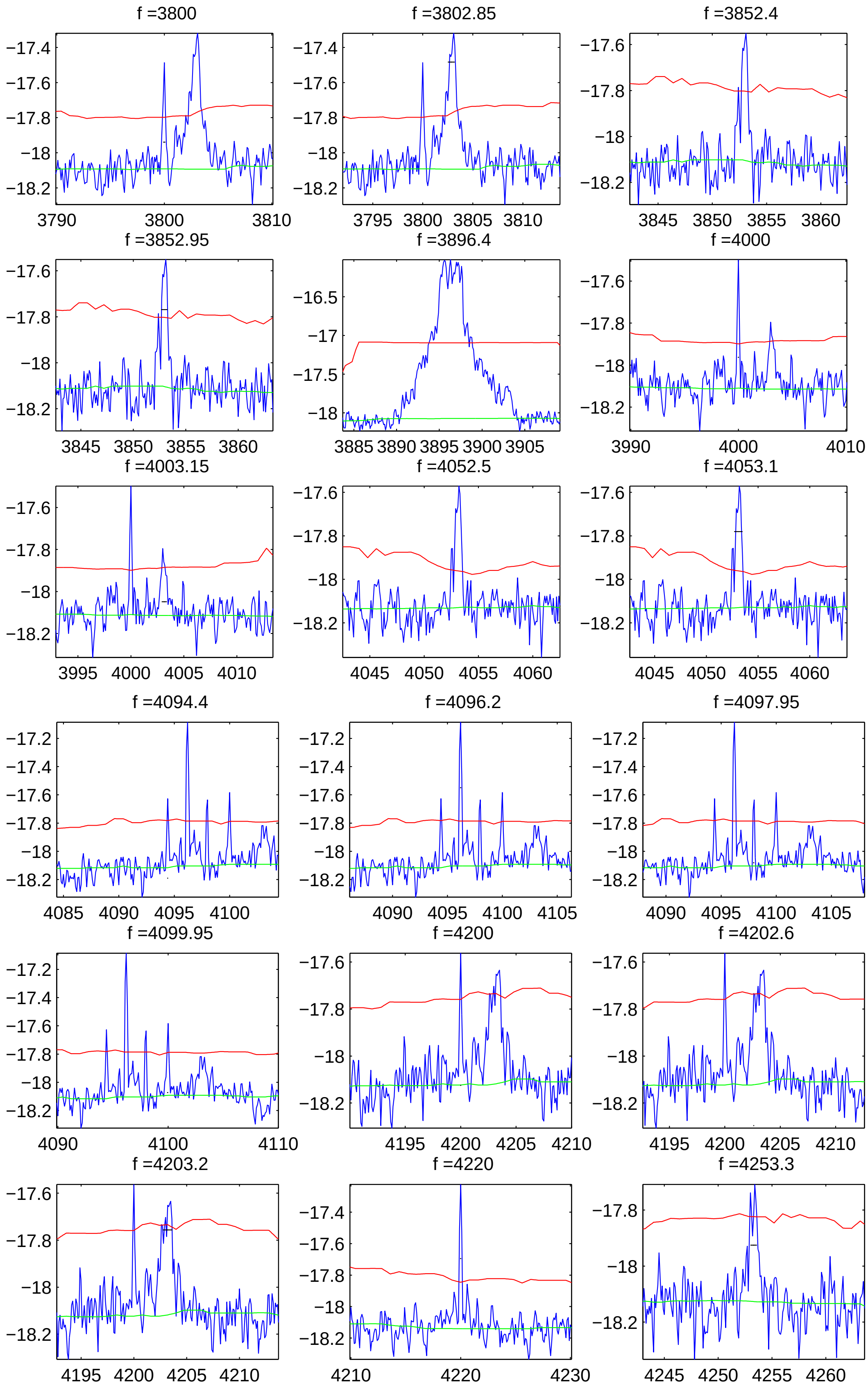

f =1501.2
f =1558.65
f =1600
f =1601.2
f =1602.1
f =1648.6
f =1649.55
f =1650.35
f =1651.25
f =1652.1
f =1653
f =1700
f =1701.3
f =1750.5
f =1751.15
f =1800
f =1801.35
f =1851.45

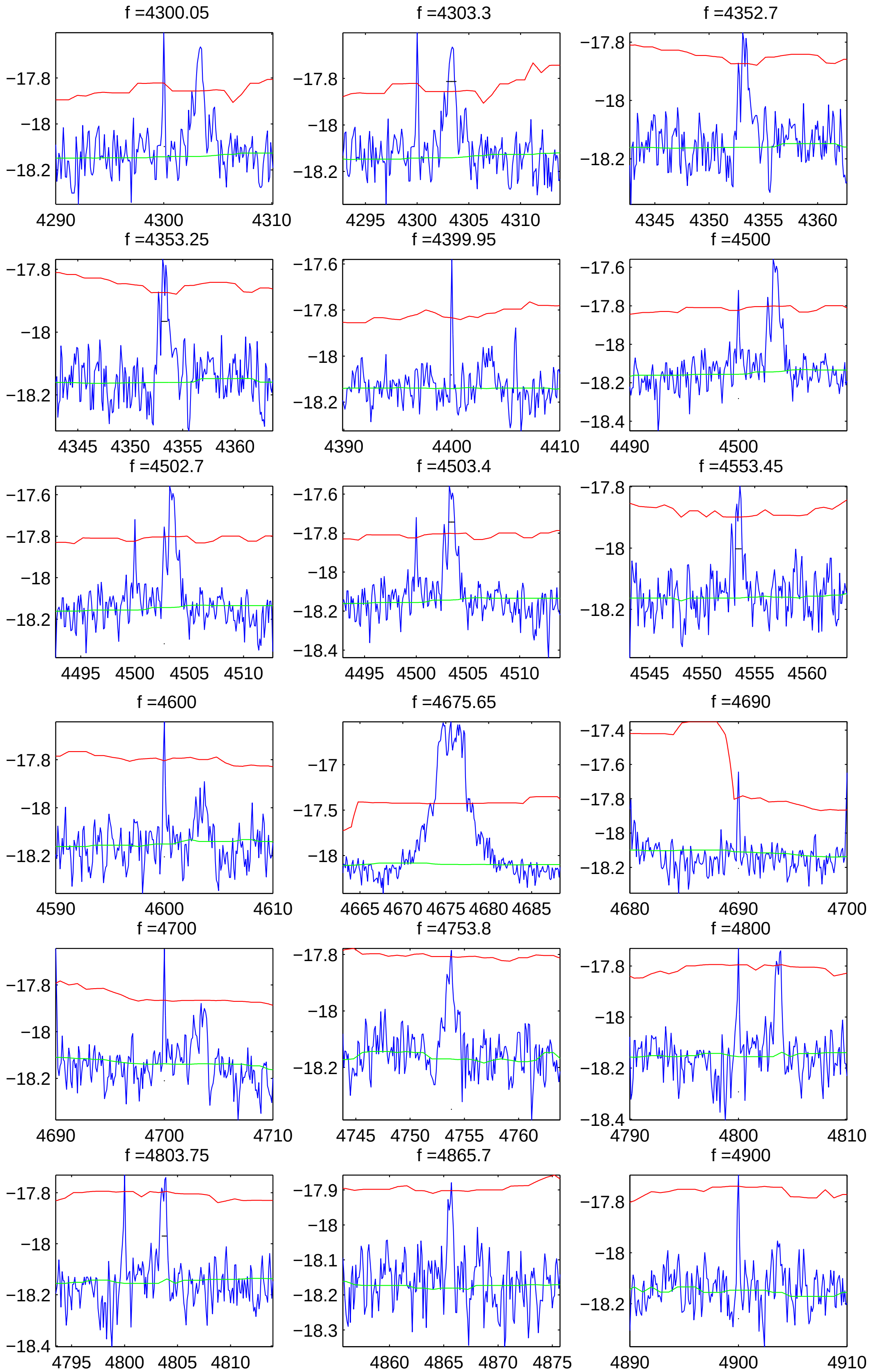

f =1900
f =1901.45
f =1950.6
f =1951.5
f =1952.35
f =2000
f =2001.5
f =2048.1
f =2051.6
f =2100
f =2101.65
f =2110.15
f =2151.6
f =2200
f =2201.65
f =2251.7
f =2300
f =2301.8

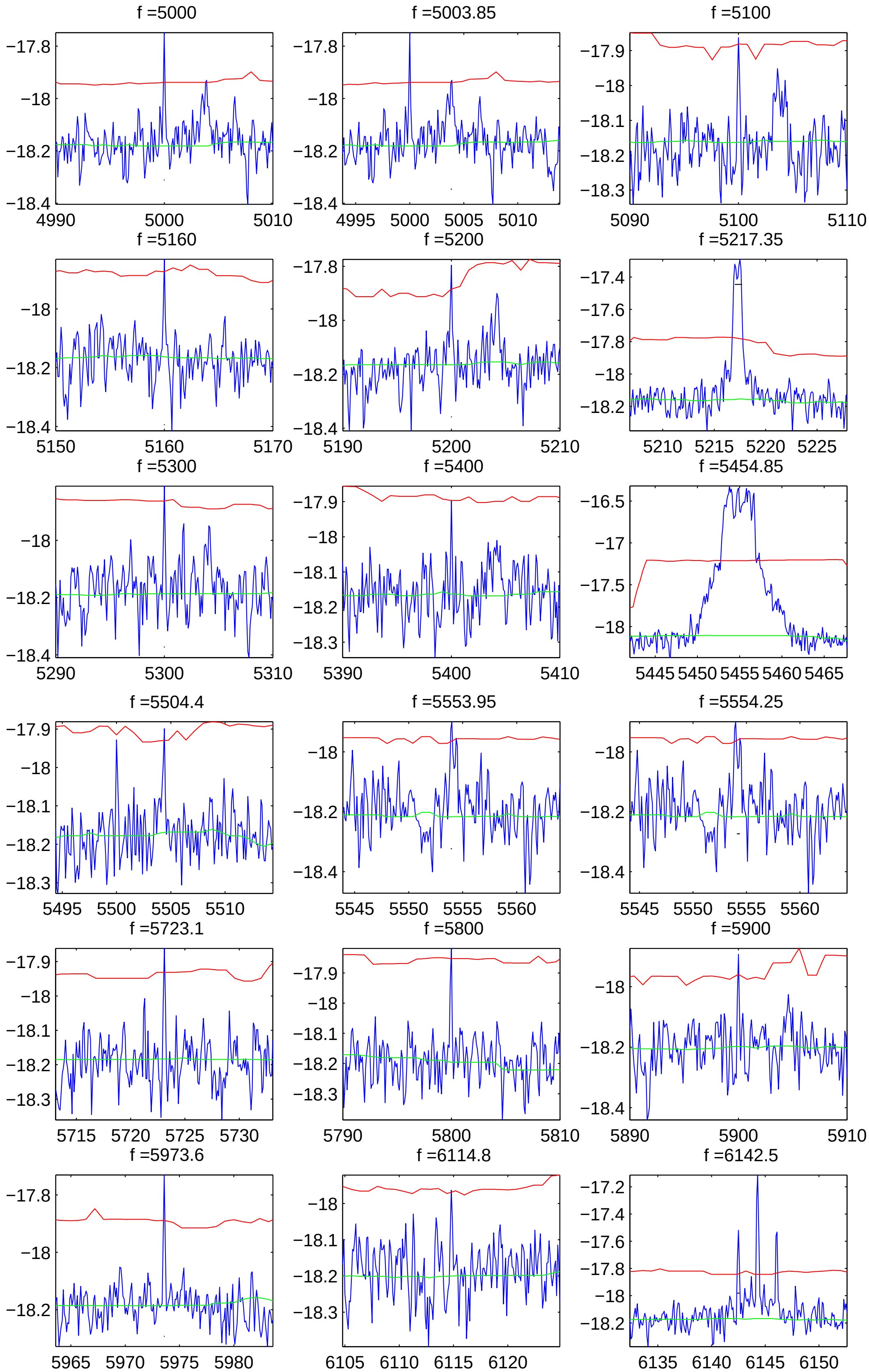

f =2305.75
f =2337.9
f =2369.75
f =2400
f =2401.75
f =2432.85
f =2451.5
f =2451.9
f =2500
f =2501.4
f =2501.85
f =2546
f =2551.95
f =2600
f =2601.5
f =2601.95
f =2608.65
f =2651.5

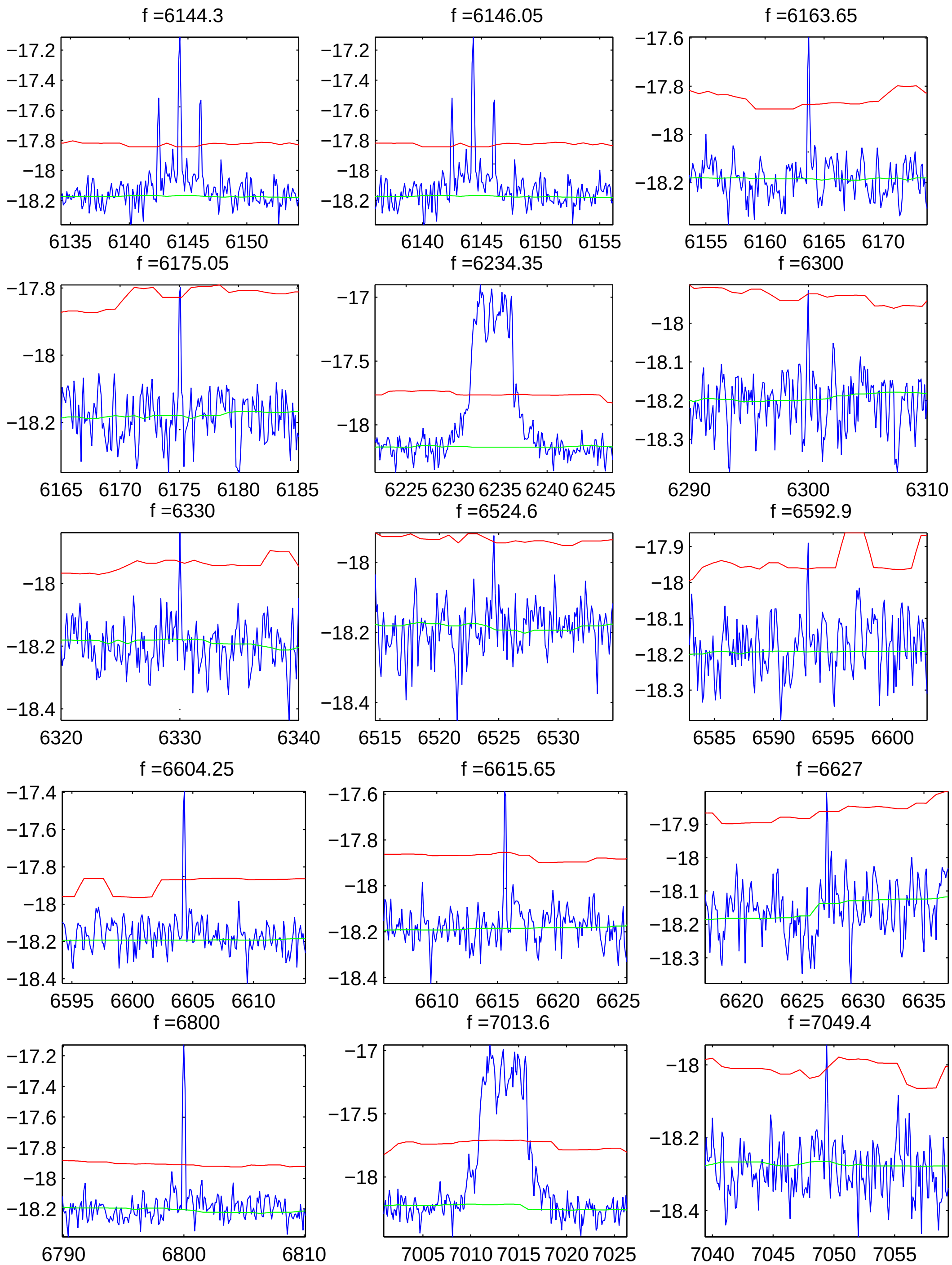

f =2652.05
−17.4 −17.6 −17.8 −18 −18.2
2645 2650 2655 2660
f =2700
−17.6 −17.8 −18 −18.2
2690 2700 2710
f =2702.1
−17.6 −17.8 −18 −18.2
2695 2700 2705 2710
f =2703
−17.6 −17.8 −18 −18.2
2695 2700 2705 2710
f =2752.05
−17.4 −17.6 −17.8 −18 −18.2
2745 2750 2755 2760
f =2799.95
−17.2 −17.4 −17.6 −17.8 −18 −18.2
2790 2800 2810
f =2801.9
−17.2 −17.4 −17.6 −17.8 −18 −18.2
2795 2800 2805 2810
f =2852.15
−17.4 −17.6 −17.8 −18 −18.2
2845 2850 2855 2860
f =2900
−17.6 −17.8 −18 −18.2
2890 2900 2910
f =2902.25
−17.6 −17.8 −18 −18.2
2895 2900 2905 2910
f =2952.25
−17.4 −17.6 −17.8 −18 −18.2
2945 2950 2955 2960
f =2999.95
−17.2 −17.4 −17.6 −17.8 −18 −18.2
2990 3000 3010
f =3002.35
−17.2 −17.4 −17.6 −17.8 −18 −18.2
2995 3000 3005 3010
f =3052.2
−17.4 −17.6 −17.8 −18 −18.2
3045 3050 3055 3060
f =3100
−17 −17.5 −18
3090 3100 3110
f =3117.2
−16 −16.5 −17 −17.5 −18
3105 3110 3115 3120 3125
f =3152.4
−17.6 −17.8 −18 −18.2
3145 3150 3155 3160
f =3199.95
−17.6 −17.8 −18 −18.2
3190 3200 3210


f =3202.3
f =3252.25
f =3287.15
f =3299.95
f =3302.5
f =3352.4
f =3399.95
f =3402.5
f =3452.55
f =3500
f =3502.5
f =3552.65
f =3600
f =3602.8
f =3652.9
f =3699.95
f =3702.8
f =3752.85


f =3800
−17.4 −17.6 −17.8 −18 −18.2
3790 3800 3810
f =3802.85
−17.4 −17.6 −17.8 −18 −18.2
3795 3800 3805 3810
f =3852.4
−17.6 −17.8 −18 −18.2
3845 3850 3855 3860
f =3852.95
−17.6 −17.8 −18 −18.2
3845 3850 3855 3860
f =3896.4
−16.5 −17 −17.5 −18
3885 3890 3895 3900 3905
f =4000
−17.6 −17.8 −18 −18.2
3990 4000 4010
f =4003.15
−17.6 −17.8 −18 −18.2
3995 4000 4005 4010
f =4052.5
−17.6 −17.8 −18 −18.2
4045 4050 4055 4060
f =4053.1
−17.6 −17.8 −18 −18.2
4045 4050 4055 4060
f =4094.4
−17.2 −17.4 −17.6 −17.8 −18 −18.2
4085 4090 4095 4100
f =4096.2
−17.2 −17.4 −17.6 −17.8 −18 −18.2
4090 4095 4100 4105
f =4097.95
−17.2 −17.4 −17.6 −17.8 −18 −18.2
4090 4095 4100 4105
f =4099.95
−17.2 −17.4 −17.6 −17.8 −18 −18.2
4090 4100 4110
f =4200
−17.6 −17.8 −18 −18.2
4195 4200 4205 4210
f =4202.6
−17.6 −17.8 −18 −18.2
4195 4200 4205 4210
f =4203.2
−17.6 −17.8 −18 −18.2
4195 4200 4205 4210
f =4220
−17.4 −17.6 −17.8 −18 −18.2
4210 4220 4230
f =4253.3
−17.8 −18 −18.2
4245 4250 4255 4260


f =4300.05
f =4303.3
f =4352.7
f =4353.25
f =4399.95
f =4500
f =4502.7
f =4503.4
f =4553.45
f =4600
f =4675.65
f =4690
f =4700
f =4753.8
f =4800
f =4803.75
f =4865.7
f =4900


f =5000
−17.8
−18
−18.2
−18.4
4990
5000
5010
f =5003.85
−17.8
−18
−18.2
−18.4
4995
5000
5005
5010
f =5100
−17.9
−18
−18.1
−18.2
−18.3
5090
5100
5110
f =5160
−18
−18.2
−18.4
5150
5160
5170
f =5200
−17.8
−18
−18.2
−18.4
5190
5200
5210
f =5217.35
−17.4
−17.6
−17.8
−18
−18.2
5210
5215
5220
5225
f =5300
−18
−18.2
−18.4
5290
5300
5310
f =5400
−17.9
−18
−18.1
−18.2
−18.3
5390
5400
5410
f =5454.85
−16.5
−17
−17.5
−18
5445
5450
5455
5460
5465
f =5504.4
−17.9
−18
−18.1
−18.2
−18.3
5495
5500
5505
5510
f =5553.95
−18
−18.2
−18.4
5545
5550
5555
5560
f =5554.25
−18
−18.2
−18.4
5545
5550
5555
5560
f =5723.1
−17.9
−18
−18.1
−18.2
−18.3
5715
5720
5725
5730
f =5800
−17.9
−18
−18.1
−18.2
−18.3
5790
5800
5810
f =5900
−18
−18.2
−18.4
5890
5900
5910
f =5973.6
−17.8
−18
−18.2
5965
5970
5975
5980
f =6114.8
−18
−18.1
−18.2
−18.3
6105
6110
6115
6120
f =6142.5
−17.2
−17.4
−17.6
−17.8
−18
−18.2
6135
6140
6145
6150


f =6144.3
−17.2 −17.4 −17.6 −17.8 −18 −18.2
6135 6140 6145 6150
f =6146.05
−17.2 −17.4 −17.6 −17.8 −18 −18.2
6140 6145 6150 6155
f =6163.65
−17.6 −17.8 −18 −18.2
6155 6160 6165 6170
f =6175.05
−17.8 −18 −18.2
6165 6170 6175 6180 6185
f =6234.35
−17 −17.5 −18
6225 6230 6235 6240 6245
f =6300
−18 −18.1 −18.2 −18.3
6290 6300 6310
f =6330
−18 −18.2 −18.4
6320 6330 6340
f =6524.6
−18 −18.2 −18.4
6515 6520 6525 6530
f =6592.9
−17.9 −18 −18.1 −18.2 −18.3
6585 6590 6595 6600
f =6604.25
−17.4 −17.6 −17.8 −18 −18.2 −18.4
6595 6600 6605 6610
f =6615.65
−17.6 −17.8 −18 −18.2 −18.4
6610 6615 6620 6625
f =6627
−17.9 −18 −18.1 −18.2 −18.3
6620 6625 6630 6635
f =6800
−17.2 −17.4 −17.6 −17.8 −18 −18.2
6790 6800 6810
f =7013.6
−17 −17.5 −18
7005 7010 7015 7020 7025
f =7049.4
−18 −18.2 −18.4
7040 7045 7050 7055