Manual for using the Spectral Analysis Tool

Luca Rigon, TU Munich, Summer 2023/Winter 2024

Satellite Geodesy Research Facility of the Technical University of Munich

The tool is using a various combinations of command argument sets. Some sample execution sets are given in the following paragraphs, as well as some explanations for the required input parameters and for the code, since there are some adaptation points when the working station is changed, such as folder paths.

Required input parameters:

- **-s** YYYY.MM.DD.hh:mm:ss
- **-session** xxxxxx
- **-type** vgos | sx

E.g. **python main.py -s 2024.02.03.07:30:00 -session q24034 -type sx**

The required parameters for the execution are the **-s** parameter, which indicates the start time for the interested session output, the **-session** parameter, which indicates the information of the session which is to be analyzed (i.e. the filename), and the antenna type for the data to be analysed: this can be either VGOS (**-type vgos**) or S/X – Legacy (**-type sx**). Based on the **-type** parameter, different functions are run and assign different channel allocations.

Additional time parameters:

- **-e** YYYY.MM.DD.hh:mm:ss
- **-d** xxh

E.g. **python main.py -s 2018.06.25.22:24:00 -d 1h -session vt8176 -type vgos**

python main.py -s 2019.05.13.23:56:00 -e 2019.05.14.17:00:00 -session vt9133ws -type vgos

The **-e** and the **-d** parameters indicate the end date & time of the session to be considered, and the duration of the session, respectively. The duration can only be in hours (format: 'xxh'; for example, 02h or 12h). If one of these two is not given, the whole duration of the session will be taken for the analysis.

NB: executing the script without setting any optional parameter (listed below) will result to an empty run of the code.

<u>Optional parameters:</u>

- **-removeCal** 0 (default: 1)
- **-method** max | mean | median (default: max)
- **-GNUplot** YYYY.MM.DD.hh:mm:ss | all
- **-spectrogram** 1
- **-skyplot** all | per_band* | per_channel
- **-sky_clip** xx:xx

E.g. **python main.py -s 2019.05.28.18:00:00 -session vt9148 -type vgos -removeCal 0 -GNUplot 2019.05.28.18:14:00**

python main.py -s 2019.05.13.23:56:00 -d 02h -session vt9133ws -type vgos -spectrogram 1

python main.py -s 2024.02.03.07:30:00 -session q24034 -type sx -skyplot all

python main.py -s 2019.05.13.18:00:00 -session vt9133 -type vgos -skyplot per_channel -sky_clip 0:20

python main.py -s 2024.02.02.17:30:00 -d 02h -session i24033 -type sx -spectrogram 1 -method mean

python main.py -s 2019.05.28.18:00:00 -session vt9148 -e 2019.05.28.22:46:00 -type vgos -removeCal 0 -GNUplot all -spectrogram 1 - skyplot all -sky_clip 50:200 -method median

The **-removeCal** parameter determines whether the registered calibration signals (peaks in the measurements) are to be considered for the analysis or not. Its default is set to True, meaning that the peaks will always be removed unless the input **-removeCal 0** is provided. The frequencies of these calibration signals (hence their indexed location within the datasets) are already provided but differ for different antenna types; hence they shall be adapted to the receiving antenna's calibration signal frequencies and channels which are being considered for the analysis (these can be found within the function *remove_peaks_vgos()* and *remove_peaks_sx()* in the *utils.py* - file).

The **-method** parameter allows to choose on a method for analyzing the data: **-method max** returns the maximum values measured in each channel and is the default parameter in case it is not provided. For a further smoothening of the power measurements **-method median** or **-method mean** can be used; any 'numpy' – function is allowed.

The **-GNUplot** parameter indicates the time at which the spectral data must be visualized as a GNU plot. This data will only be plotted if the indicated time is in the required format (see below), corresponds to the time of an existing scan-file, and is situated within the chosen analysis time interval of the session. Another allowed input is **-GNUplot all**, which will then generate the plot for every scan within the chosen time range. If not provided, no GNUplots will be returned.

The **-spectrogram** parameter determines whether to perform the spectral analysis (and plot the spectrograms, i.e. waterfall plots) of the session. Its default is set to False, meaning that the spectral analysis will only run if the input **-spectrogram 1** is provided.

The **-skyplot** parameter determines whether to visualize the RFI counts spatially as a polar plot or not, for the Azimuth and Elevation values provided in the summary file. Its default is set to False; the skyplot will only be generated for the provided inputs **-skyplot all**, **-skyplot per_band** or **-skyplot per_channel**. When choosing **all**, the analysis and final plot is generated for the values taken over all channels simultaneously. When choosing **per_band**, the data is split into the broader bands (so X upper pol. bands, X lower pol. bands and S upper pol. bands for S/X Legacy data) and the skyplots are generated over those separately. *N.B.: this command does not exist for VGOS data, as it is already separated into the respective spectral bands A,B,C,D; hence, this command is equivalent to **all** in this case. For the **per_channel** setting, every channel is extracted and plotted separately. Multiple parameters can also be selected simultaneously, leading to a simultaneous analysis of e.g. all & per_band data while the data has been loaded. To be correctly parsed, the commands should follow each other separated by a trait, e.g. **all-per_band-per_channel**. The VGOS channel numbers and their respective frequencies are shown in Fig. 1 (channels 9-16 have the same frequencies again, but opposite polarization). They also have fixed bandwidths of 32MHz. The S/X channels as well as their bandwidths may vary, hence in this case the <session>.log – file is required as an input to extract them correctly.

The **-sky_clip** parameter specifies whether to clip the power intensities on a specific range, allowing to visualize better strong disturbances on the skyplot, or to filter them out instead (to be implemented). Input is -**sky_clip clip_min:clip_max**, where these two values specify the range of values on which to clip it. If not provided, no clipping is performed.

	1	2	3	4	5	6	7	8
Band A	3032.4	3064.4	3096.4	3224.4	3320.4	3384.4	3448.4	3480.4
Band B	5272.4	5304.4	5336.4	5464.4	5560.4	5624.4	5688.4	5720.4
Band C	6392.4	6424.4	6456.4	6584.4	6680.4	6744.4	6808.4	6840.4
Band D	10232.4	10264.4	10296.4	10424.4	10520.4	10584.4	10648.4	10680.4

Fig. 1

Multiple optional parameters can be used simultaneously.

The **start**, **end** and the **GNUplot** parameters should have the following format:

%Y.%m.%d.%H:%M:%S (YYYY.mm.dd.HH:MM:SS)

For each session folder, there might be different day of year files. This is handled by listing all the available files in the particular folder of the session and analyzing them regardless checking the time, and in the spec files, this can be observed. For example, the following execution shows this case: **python main_edit.py -s 2019.05.28.23:54:00 -e 2019.05.29.00:12:00 -session vt9148 -spectrogram 1**, and is shown in Fig.2:

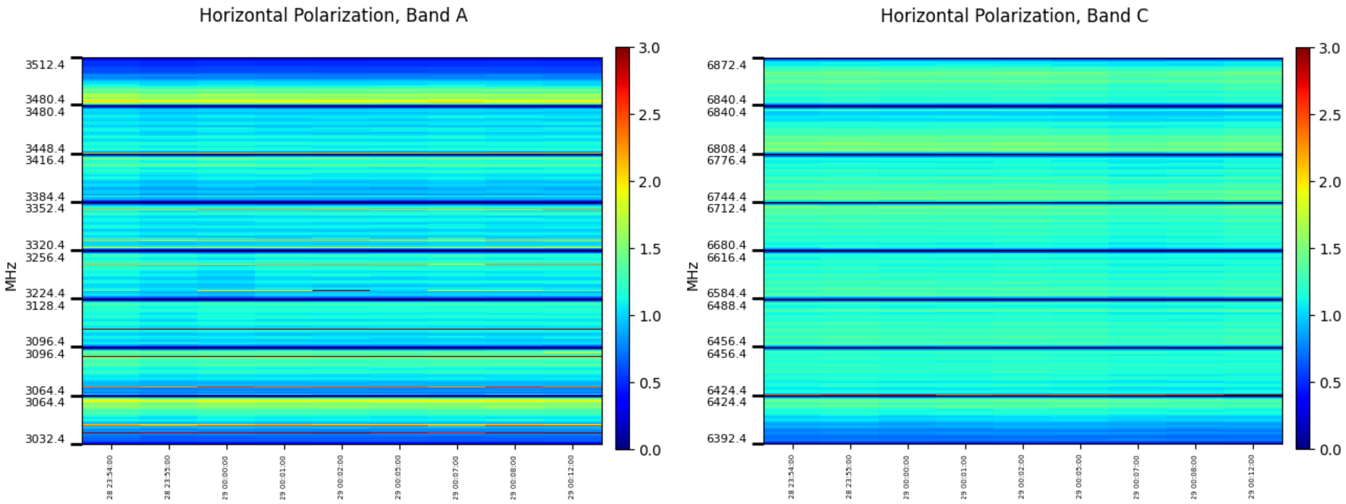

Fig. 2

As can be seen here, if there is another day in the folder, the spectrum results show that in the desired format.

For each spectral band, the scans of the chosen session are extracted and appended into one large dataset, going from the defined beginning time until the end time.

These are then further processed based on the input parameters, returning different types of images. For VGOS data, the possible outputs are:

- One image, or n_s images (where n_s is the number of samples found for the selected interval of the session), containing 16 GNU power-subplots measured at the 8 different frequency ranges, for both 2 polarizations (Horizontal/ Vertical), for each spectral band A, B, C, D (4x1 or 4xn_s images, looking like Fig.3a).
- A spectrogram which shows the measured (maximum value, mean or median, …) power count for all 8 frequency ranges over the whole selected time span, for each polarization, and each spectral band A, B, C, D (4x2 images, see Fig.s2,4).
- One skyplot and/or 16 skyplots which visualize the spatial distribution of the measured counts (maximum value, mean or median, …), according to the corresponding Azimuth and Elevation data provided in the summary file of the session. An arrow is displayed pointing towards the highest disturbance within the selected spectral channel. The frequencies, as well as the intensities, of the 5 highest disturbances over all channels are returned as well. The polar plot is generated for each spectral band A, B, C, D either over all channels taken simultaneously (skyplot input parameter “all”, returning 1x4 images), or over each channel taken individually

(skyplot input parameter “per_channel”, returning 16x4 images). In total, up to 17x4 images can be returned by this parameter; an example image can be seen in Fig.5a.

- A map of the observatory (using the OpenStreetMap API) with N=5 arrows pointing towards the N returned highest disturbances over all channels, for each band A, B, C, D (Fig.5b). The location of the observatory is given as a string and can be manually changed at the beginning of the *run_vgos_analysis.py* script (line 11), and map visualization can manually be removed at line 228 of the same script, by setting the variable **obs_map=False** (1x4 images).

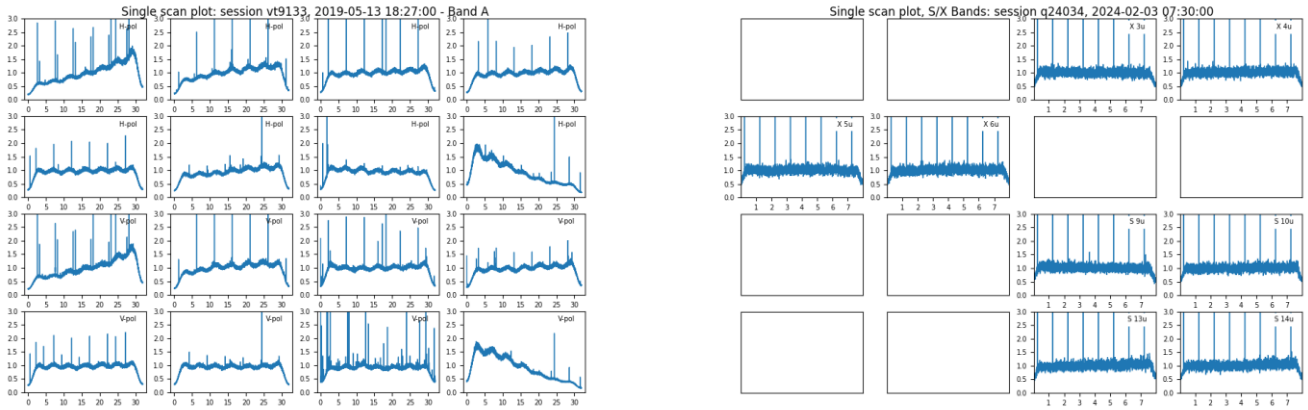

Fig. 3

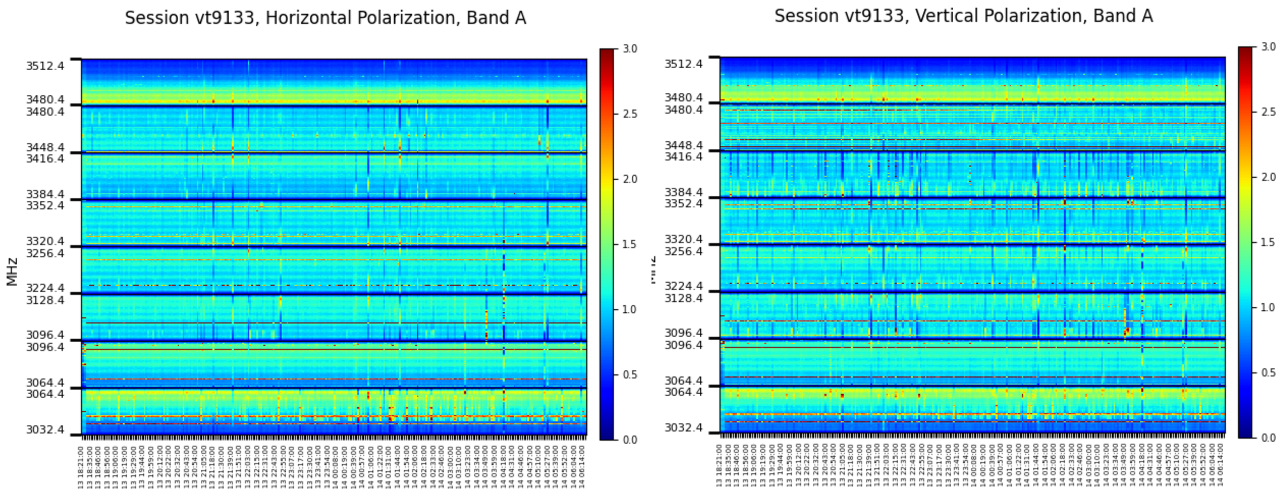

Fig. 4

For S/X Legacy data, the type of outputs remains the same, but the workflow changes a bit, as the channel & frequency allocations change dynamically and have to be automatically extracted by accessing the <session>.log – file. The possible outputs then are:

- One image, or n_s images (where n_s is the number of samples found for the selected interval of the session), containing 16 GNU power subplots measured at the allocated frequency ranges, which can be at most 16, but not necessarily. For non-allocated channels, empty subplots are returned. Channels 1-8 represend X-band upper polarized channels (Xu), and 9-16 represent S-band upper polarized channels (Su). It can occur that some sessions also allocate 2 lower X-polarized channels (Xl), more specifically channel 1 and 8. In that case, these lower bands are localized at positions 9 and 10, whereas the S bands are shifted by two spots, and S15 & S16 are not allocated. In total, 4x1 or 4xn_s images are returned; an example is shown in Fig.3b)

- A spectrogram which shows the measured (maximum value, mean or median, …) power count for all allocated frequency ranges over the whole selected time span, for each spectral band; hence, one spectrogram for the Xu – channels and one for the Su – channels is returned (2 images, see Fig.s2,4)
- One, 2-3, and/or n_c skyplots (where n_c is the total number of allocated channels for the session) which visualize the spatial distribution of the measured counts (maximum value, mean or median, …), according to the corresponding Azimuth and Elevation data provided in the summary file of the session. An arrow is displayed pointing towards the highest disturbance within the selected spectral channel. The frequencies, as well as the intensities, of the 5 highest disturbances over all channels are returned as well. The polar plot is generated either over all channels (and bands) taken simultaneously (skyplot input parameter “all”, returning 1x4 images), over the single bands & polarizations (skyplot input parameter “per_band”, returning either 2 images, one for Xu and one for Su, or 3 images if the lower bands Xl are allocated as well), or over all n_c allocated channels (skyplot input parameter “per_channel”, returning n_c images). In total, up to 4+n_c images can be returned by this parameter; an example image can be seen in Fig.5a
- A map of the observatory (using the OpenStreetMap API) with N=5 arrows pointing towards the N returned highest disturbances over all channels (Fig.5b). The location of the observatory is given as a string and can be manually changed at the beginning of the *run_sx_analysis.py* script (line 12), and map visualization can manually be removed at line 335 of the same script, by setting the variable **obs_map=False**.

For entering for the interested session folder, the following code selection should be updated (found int *utils.py*):

```
def get_session_path(session):
    # Update here for the correct session path where the path is different.
    path = f'Datasets/{session}/'
    return path
```

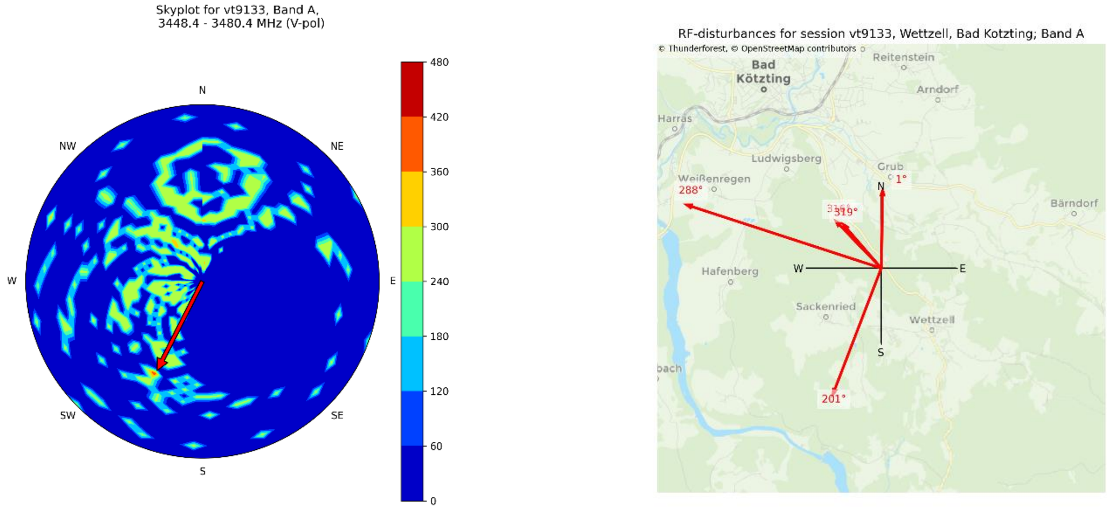

Fig. 5

This code snippet assumes that the given session parameter in the command argument has the same name as the folder containing all the scans to be analyzed. For example, for the above execution scenarios, the folders are appearing like following:

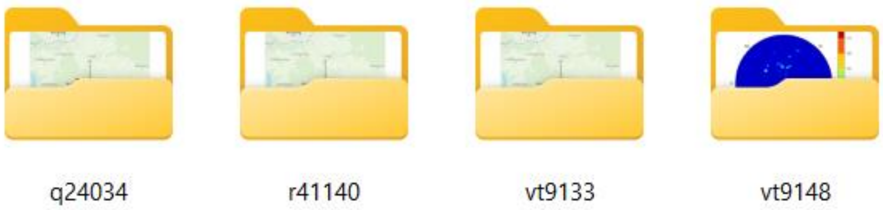

Fig. 6

This path has to be the same for all input files, hence also the session summary file, as well as the log-file, have to be included in this folder.

A results folder (“Figures/*sessionname*”) for the selected session is automatically generated within the current workspace, where the output images are automatically saved. The image names contain the selected parameters, as well as their timestamps.

Here, another important note is the name of the files to be analyzed. From the command arguments, the session parameter is given. For searching the names, the first 6 characters are searched in the dedicated session folder, so that the ‘ws’ or another set of letters attached at the end, indicating different antennas, aren’t creating problems.

For the spectrograms, it was observed that some of the executions are taking a little longer than expected with some sessions when there are many files to be analyzed. At this point, a quick solution can be the adaptation of the split numbers. If there are a lot of records inside a search of interest, the split number can be decreased. (Variable **split_num** at line 13 of the *run_vgos_analysis.py* script, and line 14 of the *run_sx_analysis.py* script). The default value is 160.

For just a quick plotting of the Figures, but without saving them, the figure saving setting can be manually changed to **figsave=False** at line 227 of the *run_vgos_analysis.py* script, and line 334 of the *run_sx_analysis.py* script. Default: **True**.

If more/less disturbances want to be shown on the observatory map, the setting N can be manually changed to the wanted number of disturbances by assigning a new value to the **n_dist** parameter, found at line 166 of the *run_vgos_analysis.py* script, and line 235 of the *run_sx_analysis.py* script. Default is 5.