Spectrum_5_0_0.praat

Task:

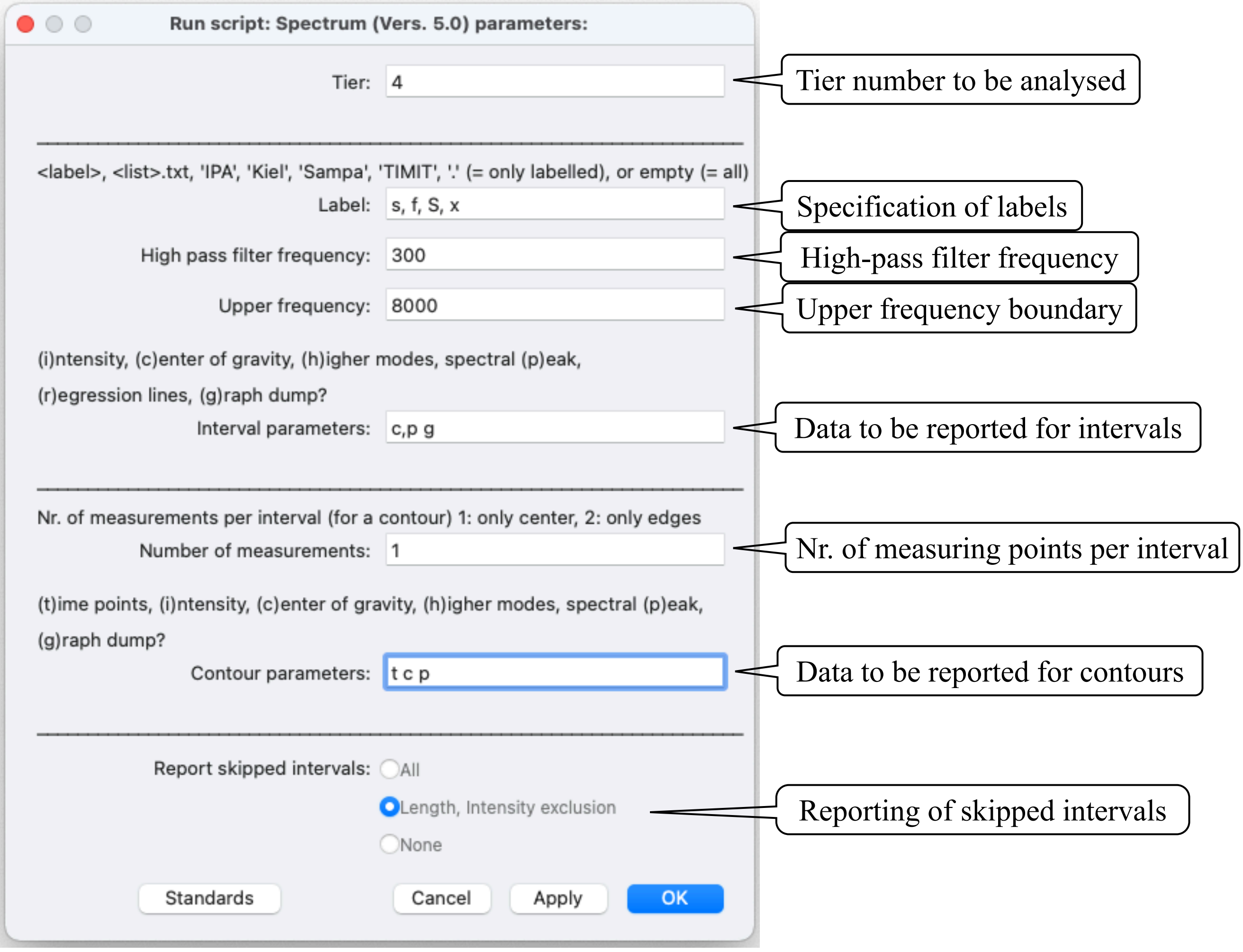
This script opens all TextGrid files in a directory and associated sound files (if they exist), computes the spectral modes, the peak frequency, spectral regression lines of the spectra of the intervals (or whole files) or along the intervals (as a ‘contour’), and writes the results to a text file with the name “spectrum_results_<date>_<time>.txt”. Graphs of the spectra can be generated as well and are stored in separate files.

Remember that the selected upper frequency boundary has a strong affect on the results of the spectral analysis!

This rather complex script which is controlled with many parameters. Some basic parameters are set with a form window, but many more parameters can be easily set inside the script (or integrated into the form window, which has been restricted to fit into a screen with 640+ points vertical resolution).

Content:

Basic Parameters:



Run script: Spectrum (Vers. 5.0) parameters:

Tier: 4

<label>, <list>.txt, 'IPA', 'Kiel', 'Sampa', 'TIMIT', '.' (= only labelled), or empty (= all)

Label: s, f, S, x

High pass filter frequency: 300

Upper frequency: 8000

(i)ntensity, (c)enter of gravity, (h)igher modes, spectral (p)eak, (r)egression lines, (g)raph dump?

Interval parameters: c,p g

Nr. of measurements per interval (for a contour) 1: only center, 2: only edges

Number of measurements: 1

(t)ime points, (i)ntensity, (c)enter of gravity, (h)igher modes, spectral (p)eak, (g)raph dump?

Contour parameters: t c p

Report skipped intervals: ○ All
● Length, Intensity exclusion
○ None

Tier number to be analysed

Specification of labels

High-pass filter frequency

Upper frequency boundary

Data to be reported for intervals

Nr. of measuring points per interval

Data to be reported for contours

Reporting of skipped intervals

Tier number to be analysed:

The number of the interval tier to be analysed. Leaving this field empty to using the tier number 0 will treat whole files as one interval. If no TextGrid files can be found at all, whole files will also be treated as one interval. If single sound files have no TextGrid associated with it (by name), a warning will be issued.

Specification of labels:

The intervals that should be analysed can be specified in several ways (at present, label names may not contain spaces or punctuation symbols!):

<label>:

Giving a label (e.g. f) or a list of labels separated by commas or spaces (e.g. s, ʃ) will only report segments that have this label. This function is case sensitive.

<list>.txt:

Giving a text file (e.g. label_list.txt) will report all segments that are listed on a line-by-line basis in a raw text file (not a Word or Pages file). Note that the extension *.txt* must be given in this field. Example of such a text file:

```
f
s
ʃ
```

'ipa':

Writing ipa in this field will use the IPA notation for fricative segments.

List of IPA fricatives (additional marks ːˑ are possible):

ɸ, β, f, v, θ, ð, s, z, ʃ, ʒ, ʂ, ʐ, ç, ʝ, x, ɣ, χ, ʁ, h, ɦ, ʕ

'kiel':

Writing kiel in this field will use the Kiel-Corpus notation for fricative segments.

List of Kiel-Corpus fricatives:

```
C, S, Z, f, h, s, v, x, z
```

'sampa':

Writing sampa in this field will use the SAMPA notation for fricative segments.

List of SAMPA fricatives (additional mark : is possible)

```
D, S, T, Z, f, h, j, s, v, z
```

'timit':

Writing timit in this field will use the TIMIT-Corpus notation for fricative segments.

List of TIMIT fricatives (upper and lower case are handled):

```
ch, jh, s, sh, z, zh, f, v, th, dh, hh, hv
```

'.':

Using a dot (.) will report values for every labelled interval.

empty:

Leaving this field empty will report values for all labelled and unlabelled intervals.

High-pass filter frequency:

The script performs a high-pass filtering (i.e. attenuating lower frequencies), which is usually done to reduce the influence of any voicing. The script filters the audio files internally but will not save them to disk. If this value is set to 0, no filtering is performed.

Upper frequency boundary:

Upper bound for the spectral computation. The script downsamples internally audio files with a higher frequency but will not save them to disk. In case that this upper boundary is set above the Nyquist frequency of an audio file (i.e. half the sampling rate) the script will issue a warning but will proceed. If this value is set to 0, no downsampling is performed. Remember that the selected upper frequency boundary has a strong affect on the results of the spectral analysis!

Data to be reported for intervals:

By default, the script reports (see **Result file format** below) the file name, name of the interval, starting time of the interval, duration of the interval. Additional data can be reported and/or displayed by specifying letters separated by commas or spaces in this field (e.g. c p) to report data for the interval selected by the specification of labels:

c	**Center** of gravity (CoG)
h	**Higher** modes: Standard Deviation (StDec, SD), Skewness and Kurtosis.
g	**Graphs** of the spectra are displayed and can be stored as files (s.**Graphs** below)
i	**Intensity** of the interval
p	Frequency of the spectral **peak**
r	Spectral **regression** lines; s. **Background** below.

Nr. of measuring points per interval:

The number of data points along an interval/file (a ‘contour’). If this field is left empty or ‘0’ is given, no contour data is generated. ‘1’ will report data at the center of the interval/file, ‘2’ will report data at the left and right edges, ‘3’ at edges and center, etc. Higher numbers will report data at more points. Note that this is the number of data points, not intervals: ’10’ will report data at 0%, 11.11%, 22.22%, …77.78, 88.89%, 100% of the length of the interval/file. Which data to report will be specified in the field **Data to be reported for contours** below. **Note that the values of the contour will not easily match with the data of the interval** (e.g., the CoG of the interval must not be similar to the average CoGs of an interval)! See **Background** below for details.

Data to be reported for contours:

Which data to report and/or displayed can be reported by specifying letters separated by commas or spaces in this field (e.g. c p) to report data for time points along the interval selected by the specification of labels:

c	Center of gravity (CoG)
h	Higher modes: Standard Deviation (StDec, SD), Skewness and Kurtosis.
g	Graphs of the spectra are displayed and can be stored as files (s.**Graphs** below)
i	Intensity of the interval
p	Frequency of the spectral peak

Reporting of skipped intervals:

Handling of segments which are either excluded by the **Specification of labels** or because they do not fulfil the **Minimal length** or **Minimal intensity** (see **More parameters** below) criteria:

All:

All intervals excluded from the computations are reported with File, Label, Start(s) and Duration(ms) only (see **Result file format** below), all other values are set to the missing value symbol (see **More parameters** below). This function can be helpful to see the context of a particular interval analysed, e.g. the listing will show the data for a fricative, but also the labels of the intervals before and after it, which are not fricatives.

Length, Intensity exclusion:

All segments that fulfil the **Specification of labels** are reported, but fail either the length or intensity criteria (see **More parameters** below). Only File, Label, Start[s] and Duration[ms] are reported (see **Result file** below), all other values are set to the missing value symbol (see **More parameters** below).

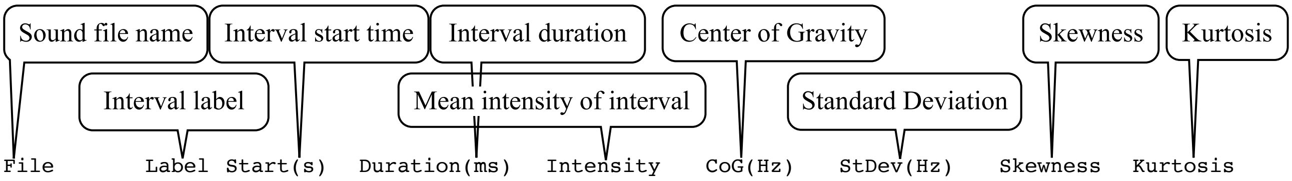
None:

No intervals that are excluded because of any criteria are reported.

Result file format:

The script generates a raw text file with tab-delimited (see **More parameters** to change the delimiter) data and a header line. The file name is of the form “spectrum_<date>_<time>.txt” with <date> of the form ‘yymmdd’ (i.e. 2-digit year, 2-digit month, 2-digit day) and <time> of the form ‘hhmmss’ (i.e. 2-digit hour in 24 hour format, 2-digit minutes, 2-digit seconds). For example, a file with the name ”spectrum_190322_150110.txt” was created on the 22nd of March 2019 at 3pm, 1 minute and 10 seconds. The parameters controlling the computation are listed at the end of every result file.

The result file for a parameter setting to report intensity, CoG and higher modes for intervals with the label s, f, S, x or C on tier 4:



```
File        Label  Start(s)    Duration(ms)   Intensity    CoG(Hz)    StDev(Hz)    Skewness    Kurtosis
                                              _mean(dB)
g071a000    x       1.2279        38.6          64.81      1750.1      1692.12       1.75        2.09
g071a000    f       1.4718       115.3          58.95      3368.6      2481.38       0.52       -1.23
g071a000    C       1.7322        37.2          62.65      3759.4      1266.08       0.88        1.11
g071a000    f       1.9294        56.1          59.61      5643.2      2240.86      -1.24       -0.04
g071a000    x       2.0143        38.3          59.67       941.2       715.86       4.70       30.10
...

Script: Spectrum_5_0_0.praat
Analysis started: 23-Feb-21 10:50:06
Tier: 4
Labels: f,s,S,C,x
High pass filter frequency: 300 Hz
Highest frequency: 8000 Hz
Minimal length: 25 ms
Minimal intensity: 0 dB
Low pitch for intensity computation: 75 Hz
Step rate: 2 ms
Window size: 5 ms
```

More parameters:
Additional parameters can be set underneath the ‘form’ section in the script. These are:

Spectrogram computing parameters

window_length_ms = 5
Size of the window size for the spectrogram computing in milliseconds. A value of 5 ms is typical for a wide band spectrogram.

step_rate_in_ms = 2
Temporal step rate for computing the spectral windows of the spectrogram in milliseconds. A value of 2 ms gives reasonable good temporal resolution and fast computing times.

frequency_step = 20
Frequency resolution in Hertz for each spectral window.

Regression lines presets (see **Background** below)

slope_low_bottom = high_pass_filter_frequency
Lower frequency boundary in Hertz of the ‘Low’ regression line

slope_low_top = 2500
Upper frequency boundary in Hertz of the ‘Low’ regression line

slope_high_bottom = slope_low_top
Lower frequency boundary in Hertz of the ‘High’ regression line

slope_high_top = highest_frequency
Upper frequency boundary in Hertz of the ‘High’ regression line

‘Low’ and ‘High’ regression lines can overlap (i.e. slope_low_top can be higher than slope_high_bottom). It is also possible to compute only one regression line by setting the ‘top’ and ‘bottom’ frequency of the ‘Low’ or ‘Top’ regression line to the same value (e.g. slope_low_bottom = slope_low_top = slope_high_bottom = 300 and slope_high_top to 8000 will compute the regression line between 300 and 8000 Hz).

remove_DC_offset = 1
The DC-component of the spectra (i.e. the intensity at 0 Hz) is removed when this parameter is set to ‘1’; a value of ‘0’ will not remove the DC offset from the data.

Minimal length of intervals to be analysed:

minimal_length_ms = **25**
The minimal length of intervals (in milliseconds) to be analysed. Setting this value to ‘0’ will analyse all selected (by the **Specification of labels**) intervals.

Maximal length of intervals:

max_length = **10.0**
If an interval is longer than this specification (in seconds) a warning will be issued in the Info window but processing will continue.

Minimal intensity of intervals to be analysed:

minimal_intensity = **0**
The minimal average RMS-intensity (in dB) of intervals to be analysed. Setting the **minimal intensity** to ‘0’ will analyse all intervals that are selected by the **Specification of labels**.

low_pitch = **75**
The intensity computation expects a low_pitch value, which is only used for that computation

Maximal number of intervals for reporting the contour

max_number_of_measurements = 20

If the user requests more than this number of measurements for an interval, a *pause* statement is performed to confirm the specified number of measurements.

Contour position in steps (1, 2, 3,…) or percentages (0%, 25%…) of interval length

position_in_percentage = 1

When more than 3 data points are requested for the contour data (for 1 to 3 the terms 'left', 'center' 'right' are used) the data points can be reported in the header as percentage of the whole interval (e.g. 0%, 25.00%, 50.00%, 75.00%, 100%) when this parameter is set to '1' or as index number (e.g. 1, 2, 3, 4, 5) when the parameter is set to '0'.

Positioning of analysis window at the edges of an intervals:

cross_interval_boundary = **1**

If more than 1 contour position is requested, the first and last analysis window is centered on the boundary of an interval.

cross_interval_boundary = **0**

(default) The window stays within the interval (i.e., the left edge of_the first analysis window is positioned on the left edge of the interval, and the right edge of the last windows is on the right edge of the interval). Note that the positions given in percent for the contour parameters refer to the duration of this shortened interval, not to the duration of the actual interval. (The time parameter [like t_t0%] gives the actual location of the middle of the analysis window.)

Directory path:

directory$ = " "

The script handles all sound files in a directory specified by this string. If this string is an empty string, the script will handle all sound files in the directory where the script was started (i.e., the script is placed in the same directory as the sound and TextGrid files). If not a single TextGrid can be found, the whole files will be taken as one interval. If single sound files have no TextGrid associated with it (by name), a warning will be issued.

Directories:

wav_directory$ = directory$
grid_directory$ = directory$
result_directory$ = directory$
support_directory$ = directory$

The script uses internally separate strings for sound, TextGrid, result, support and graphs directories. Users who use separate directories for these can adjust these names in the script.

graph_directory$ = "./Graphs/"

Since the script can generate many individual files for graphs of spectra (see **Graphs** below), the script by default creates a sub-directory of the present directory where the files (as .pdf, .png or ,eps) are stored.

Path name:

path_name = **0** No report of full path name

path_name = **1** Full path name is reported

By default, only the TextGrid or sound file name is listed in the result file. If the full path should be reported, path_name should be set to 1.

Sound file name extensions

sound_ext$ = ".wav"

The default extension for sound files is “.wav”. This parameter can be changed in the script.

Separator symbol:

sep$ = tab$

The (columns of) data in the result file are separated by this symbol. The default is tab$ (tabulator) but users might use e.g. a comma for an csv file.

User feedback:

user_feedback = 1 Gives user feedback (default)

user_feedback = 0 No user feedback

The script reports which file is being handled and the percentage of all files in a directory that have been handled. By setting this switch, any output (other than error and warning messages) will be suppressed. This will decrease processing time, but there is no feedback other than an increasing size of the result file.

Noprogress string:

np_string$ = "" (= empty string) normal PRAAT feedback

np_string$ = "noprogress" No PRAAT feedback (default)

PRAAT itself reports its activity when computing intensity and spectral data. These outputs can take substantial time. (This string is actually positioned where in PRAAT the noprogress is written.)

Dummy data header line:

dummy_data_header = 0 No dummy data line

dummy_data_header = 1 Dummy data line

Statistic programs like JMP decide the type of data for each column on basis of the first data line. To force correct data-type assignment (due to missing data in the first data row) a dummy data line of text, 0 and 0.0 can be generated to force correct data-type assignment (with “Dummy” for strings and “0.0” for numerals).

Duration reporting:

duration_in_ms = 1 Duration is reported in milliseconds (default)

duration_in_ms = 0 Duration is reported in seconds

PRAAT uses seconds for all time data. This can lead to rather confusing number of zeros for segments which are usually short. For this reason, the duration in the result file can be reported in milliseconds.

Symbol for missing values:

missing_value_symbol$ = "NA"

When PRAAT cannot compute a value it uses internally the ‘value’ “undefined”. This script replaces this string by the string given in this field. For a subsequent analysis of the data with JMP, the dot indicate missing values, for R it would be NA.

Graph (draw spectrum) handling:

pause_drawing = 1
After drawing a graph of a spectrum the script pauses and asks for user interaction whether the graph should be saved in a file or not. This parameter can also be changed during this pause to '0', which will not pause anymore until the script has finished (see **Graphs** below for details).

save_graph = 1
After drawing a graph of a spectrum the script pauses and asks for user interaction whether the graph should be saved in a file or not. This parameter can also be changed during this pause to '0', which will not save the graphs to a file but it can be changed (or not) for the next graph (see **Graphs** below for details).

Note that setting both **pause_drawing** and **save_graph** to '0' will stop the drawing of graphs even if they are requested for the intervals or contours.

graph_low_db = 0
Setting the lower boundary of the y-axis of the spectrum to '0' will use the minimal value of the spectrum as lower boundary.

graph_high_db = 0
Setting the upper boundary of the y-axis of the spectrum to '0' will use the maximal value of the spectrum as lower boundary.

Note that the automatic setting of the dB-scale will generate different scalings for each spectrum, which could make comparisons complicated. Using fixed values, on the other hand, requires a careful and appropriate selection of values.

Graph file format

```
if (macintosh)
   graph_file_format$ = "pdf"
else
   graph_file_format$ = "png"
endif
```

PRAAT can save graphs as png or eps (and as pdf on a Macintosh). The script uses pdf for Macintosh as default and png (with 600 dots per inch) otherwise.

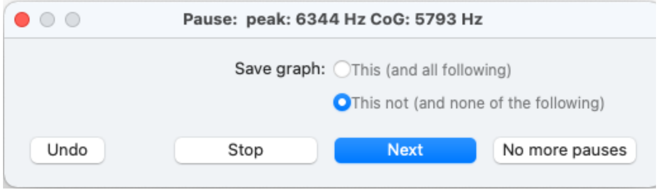
Graph text font size

10
This looks a bit weird, but just write one of the numbers 10, 12, 14, 18, 24 for the font size.

Graphs:

If a ‘graph dump’ (i.e. the drawing of spectra) is requested, the script draws a spectrum, a line for the center of gravity (if it is requested) and the regression lines (if they are requested). In the header of the graph the path (if the parameter **path_name** is set to ‘1’), the file name, the label of the interval, its starting time and its duration in millisecond are written. If requested, the center of gravity, higher modes, spectral peak location and regression lines’ intercepts and slopes are included. (The labelling is somewhat cryptic because only two lines can be written as header.)

Graph pauses after one spectrum is plotted on the screen. The user can choose to save the plot in a graph file or not save the plot. The script will proceed and pause with the next graph plot with the same selection, using the last choice as pre-set. Plots appear in the sequence one interval and then the plots for its contour data before proceeding to the next interval.



Pause: peak: 6344 Hz CoG: 5793 Hz

Save graph: ○ This (and all following)
● This not (and none of the following)

The user can select to proceed to the next plot, that no more pauses should be performed (and the script will proceed with the elected save / no save choice, or the script should stop. The ‘No more pauses’ selection will apply to interval and contour plots as well. The ‘Stop’ will save the last plot (if requested) and exit the script without removing objects from PRAATs object list. (This is done on purpose to allow users to inspect the objects.)

Graph files are by default stored in the sub-directory **Graphs** of the sounds directory, which is automatically created if it does not exist. Graph file names have the format

<file>_<label>_<time>_<index>_<date>_<time of day>.<extension>.

<file> is the name of the sound file from where the data was generated.

<label> is the label of the interval that is analyzed. In case whole files are analyzed (i.e. tier = 0 or no TextGrid files exist), <label> is empty and two underlines will between <file> and <time>.

<time> is the starting time (in seconds) of the spectral window (i.e. the left edge of an interval or the position of the spectral slice for contour data). The decimal point of the time in seconds is represented by an “s” (e.g. ‘3s6147’ is the time 3.6147 seconds). (For whole files, the time is “0”.)

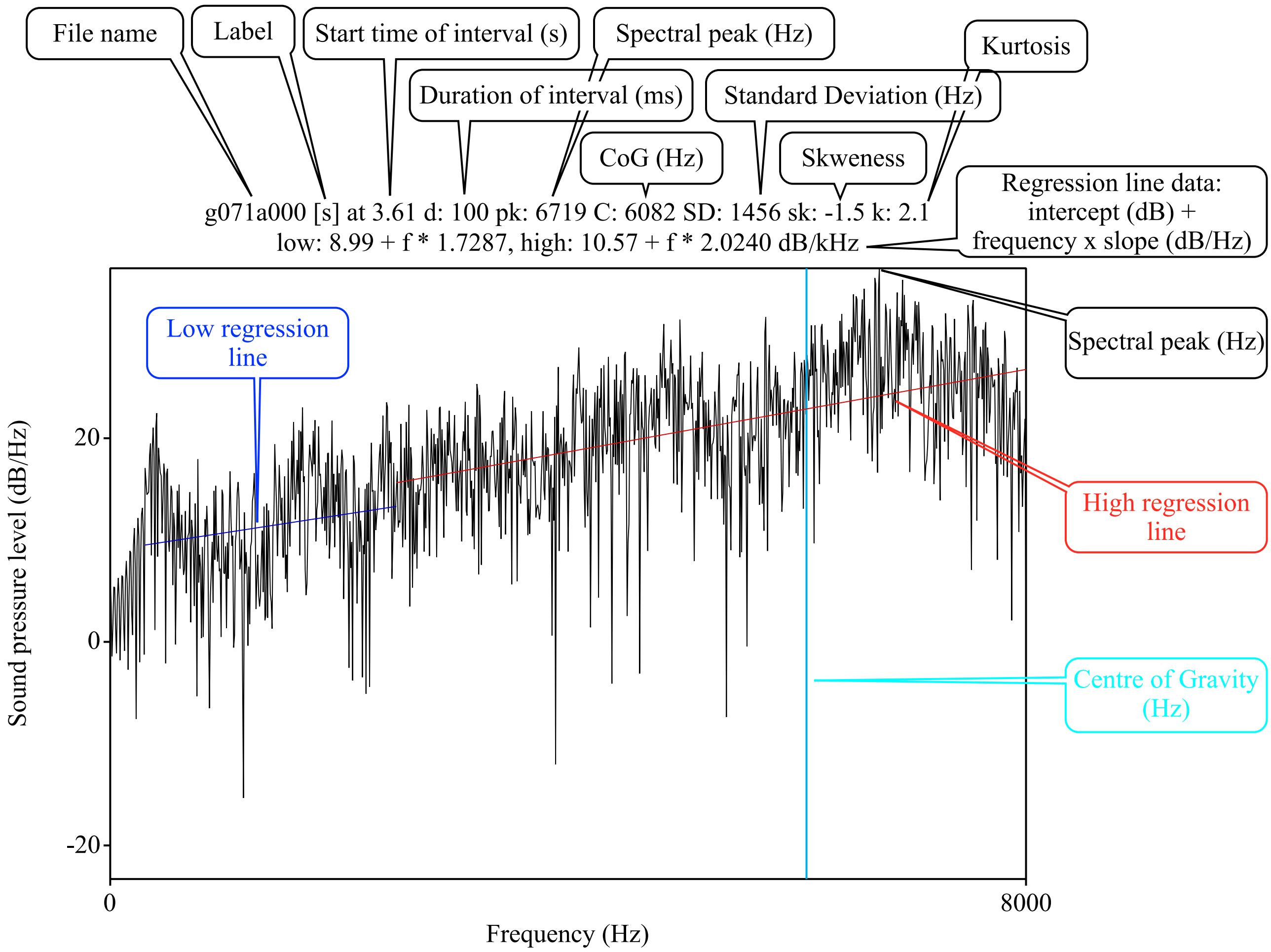
<index> is a number starting from ‘0’ (for interval data) and counting from ‘1’ to ‘number_of_measurements’ for contour data.

<date> is the date of the generation of the file in the form ‘yymmdd’ (i.e. 2-digit year, 2-digit month, 2-digit day)

<time of day> is the time of the generation of the file in the form‘hhmmss’ (i.e. 2-digit hour in 24 hour format, 2-digit minutes, 2-digit seconds)

<extension> is either ‘pdf’, ‘png’ or ‘eps’, specifying the graphics forma of the file.

For example, a file with the name “g071a000_s_3s6147_0_210219_120110” contains the interval spectrum (because the index is “0”) of the sound file “g071a000.wav” for the interval with the label “s”, which starts at “3.6147” (seconds), and was created on the 19th February 2021 at 12pm midday, 1 minute and 10 seconds.


File name
Label
Start time of interval (s)
Spectral peak (Hz)
Kurtosis
Duration of interval (ms)
Standard Deviation (Hz)
CoG (Hz)
Skweness
g071a000 [s] at 3.61 d: 100 pk: 6719 C: 6082 SD: 1456 sk: -1.5 k: 2.1
low: 8.99 + f * 1.7287, high: 10.57 + f * 2.0240 dB/kHz
Regression line data: intercept (dB) + frequency x slope (dB/Hz)
Spectral peak (Hz)
Low regression line
High regression line
Centre of Gravity (Hz)
Sound pressure level (dB/Hz)
20
0
-20
0
8000
Frequency (Hz)

Background:
The script computes the spectral parameter for the specified intervals in the frequency range given by ‘high_pass_filter_frequency’ and ‘upper_frequency’. Note that the results depend on these two parameters and will change if these boundaries are changed.

If the ‘**high_pass_filter_frequency**’ is set to ‘0’. no filtering is performed. Otherwise, the ‘high_pass_filter_frequency’ is used by the PRAAT function

```
Filter (stop Hann band): 0, high_pass_filter_frequency, 100
```

on the original sound file to generate a new sound file.

The script uses internally the ‘**highest_frequency**’ as upper bound for the frequency range. The ‘**upper_frequency**’ given in the initial form window is copied to the ‘highest_frequency’ if ‘upper_frequency’ is not ‘0’, otherwise the ‘highest_frequency’ is set to the Nyqvist frequency of a sound file (i.e., if there are files with different sampling rates in a directory, each file keeps its own sampling rate and the ‘highest_frequency’ can be different for different files). Otherwise, the sound file is assigned a ‘new_rate’ = ‘highest_frequency’ * 2 and is resampled if ‘new_rate’ differs from the ‘rate’ of the sound file with the PRAAT function

```
Resample: new_rate, 50
```

For **intervals**, the **Center of gravity (CoG), Standard Deviation (StDev, SD), Skewness** and **Kurtosis** are computed by extracting the selected intervals with the PRAAT function

```
Extract part: t_left, t_right, "rectangular", 1, “yes”
To Spectrum: "yes"
```

and using the functions

```
cog = Get centre of gravity: 2
stdev = Get standard deviation: 2
skew = Get skewness: 2
kurt = Get kurtosis: 2
```

The **position of the spectral peak** is computed with

```
To Ltas (1-to-1)
peak_freq = Get frequency of maximum: 0, 0, "none"
```

For the **contour** parameters, are computed by computing the spectrogram for the whole file with PRAATs spectrogram function

```
To Spectrogram: window_length, highest_frequency, step_rate, 20, "Gaussian"
```

and then using the functions

```
To Spectrum (slice): <middle position time>
cog = Get centre of gravity: 2
stdev = Get standard deviation: 2
skew = Get skewness: 2
kurt = Get kurtosis: 2

To Ltas (1-to-1)
peak_freq = Get frequency of maximum: 0, 0, "None"
```

Note that the values for the interval can be very different from the individual values from the contour, or the mean of the counter: the size of the spectral window differ and the individual slices of the contour will have different intensities. I.e., the mean of the contour values will not take the different intensities of the slices into account, whereas the interval vales are computed over the whole interval, taking the varying intensities of the interval into account. Additionally, the computation of the interval spectral data uses a window size with an overlap into the neighboring signals. This window overlap has a size of 50% of the interval, although with a decaying weight (I assume it is a Gaussian window – details will be in a separate documentation about window sizes that PRAAT actually uses.)

Regression lines / Slope

Regression lines to approximate the spectral shape are described in

Evers, V., Reetz, H., & Lahiri, A. (1998). Crosslinguistic acoustic categorization of sibilants independent of phonological status. *Journal of Phonetics* **36**, 345–370.

Maniwa, K., Jongman, A., & Wade, T. (2009). Acoustic characteristics of clearly spoken English fricatives. *The Journal of the Acoustical Society of America* **125**, 3962–3973.

To compute regression line coefficients (for intervals only) for parts of the spectrum, the spectrum is selected and then converted to a table.

```
        Tabulate: "no", "yes", "no", "no", "no", "yes"
• • • delete rows that are below and above the frequency range that should be approximated • • •
        To linear regression
        info$ = Info
        low_intercept = extractNumber (info$, "Intercept: ")
        low_slope_hz = extractNumber (info$, "Coefficient of factor freq(Hz): ")
```

By defining only one set of boundaries in the **Regression lines presets** (see **More parameters** above) an overall slope of the spectrum can be computed and used as something that is sometimes called a *spectral tilt*. But note that such a regression line is not the same spectral tilt as it is described in

Sluijter, A. M. C. & van Heuven, V. J. (1996). Spectral balance as an acoustic correlate of linguistic stress. *The Journal of the Acoustical Society of America* **100**, 2471–2485.

Current version and date:
5.0.0, 23-feb-2021

Known problems:
None

Planned extension:
Handling of point tiers
Allowing spaces and interpunctuation in labels
Using more than one tier to specify labels

Contact:
reetz.phonetics@gmail.com