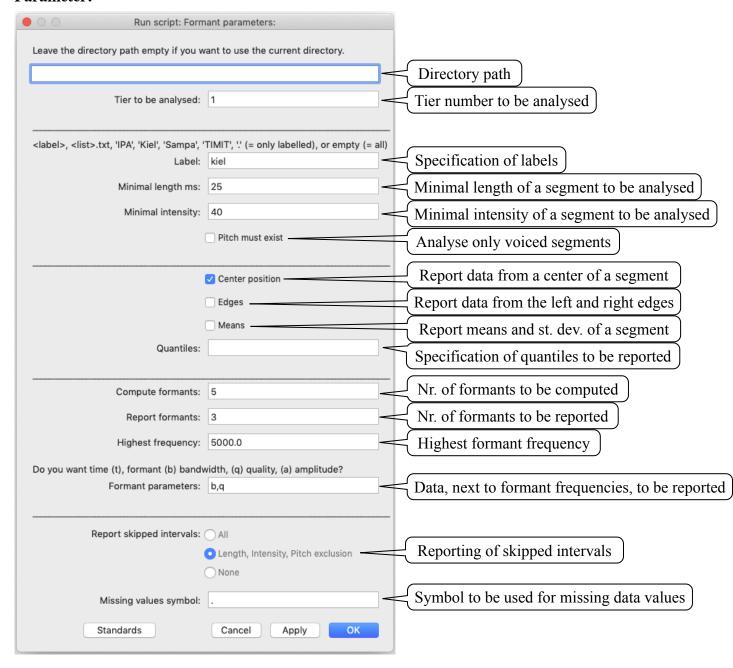## Formants_4_0.praat

Task:
This script opens all sound files in a directory and associated TextGrids (if they exist), computes formants, bandwidths, amplitudes and Q-values at the center and/or edges of all segments, which can be specified in various ways, together with means and quantiles and writes the results to a text file with the name "formant_results_<date>_<time>.txt". The length of the segment, and the pitch and intensity values are reported along with the formant data. Sound files without an associated TextGrid file will be treated as one segment on its own.

There are more parameters that control the behaviour of the script than displayed in the form window (the size of the form window fits into a screen with 900+ points vertical resolution). These parameters can be set below the form statement in the script (see the **Programming**) section.

Parameter:

Please read the **Programming** section for more internal parameters that can easily be changed.

Directory path:

The script handles all sound files in a directory specified in this field. If this field is left empty, 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).

Tier number to be analysed:

The number of the interval tier to be analysed.

(There is a not well tested functionality to handle point tiers, which will be improved in a later version.)

Specification of labels:

The segments (= intervals in PRAAT terminology) that should be analysed can be specified in several ways (in case no TextGrid file is found, this field is ignored):

<label>:

Giving a label (e.g. a:) or a list of labels separated by commas or spaces (e.g. i:,I u:,U) 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:

```
a
a:
ae
o
```

'ipa':

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

List of IPA vowels (additional marks ː ˑ ̃ ̥ ̩ ̯ are possible):

a, ɑ, æ, ɐ, ɒ, œ, e, ɛ, ə, u, ʊ ,ʉ ,i ,ɨ ,ɪ ,ɔ ,o ,ø

'kiel':

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

List of Kiel-Corpus vowels:

```
@, 2:, 6, 9, a, a:, E, e:, E:, I, i:, O, o:, U, u:, Y, y:
```

'sampa':

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

List of SAMPA vowels (additional marks : ' ` ~ % are possible)

```
A, {, 6, Q, E, @, 3, I, O, 2, 9, &, U, }, V, Y
```

'timit':

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

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

```
aa, ae, ah, ao, aw, ax, axr, ay, eh, er, ey, ih, ix, iy, ow, oy, uh, uw, ux
```

'.':
Using a dot (.) will report values for every labelled segment.

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

Minimal length of a segment to be analysed:
The minimal length of a segment (in milliseconds) to be analysed. Setting this value to '0' will analyse all selected (by the **Specification of labels**) segments.

Minimal intensity of a segment to be analysed:
The minimal RMS-intensity (in dB or as quantile of the whole file) of a segment to be analysed. A value greater equal 1 is interpreted as an absolute dB value (e.g. '50' means a minimal average RMS-amplitude of 50 dB for a segment). A value between 0 and 1 is interpreted as a quantile of the whole file (e.g. '0.1' will use the 10% quantile of all RMS values of the whole file as lower boundary for a segment). Setting the **Minimal intensity** to '0' will analyse all segments that are selected by the **Specification of labels**.

Analyse only voiced segments:
If this field is checked, only intervals where PRAAT can compute a mean pitch will be reported.

Report data from a center of a segment:
The middle of each segment specified by **Specification of labels** will be reported. If no TextGrid file is found, the center of a sound file will be used.

Report data from the left and right edges:
The edges each segment specified by **Specification of labels** will be reported. If no TextGrid file is found, the center of a sound file will be used.
(There is a parameter **do_not_cross_segment_boundary** in the script that controls the positioning of the analysis window at these edges. If set to 1 (the default), the analysis window is positioned to stay inside a segment. If this parameter is set to 0, the analysis windows will be centred on the edges of a segment. In case the analysis window will go beyond the beginning or end of a sound file, the window is positioned to be completely inside a sound file. The exact location of the center of an analysing window can be reported with the **time** request, see below.)

Report means and st. dev. of a segment:
Setting this switch will report the mean, standard deviation and median of each segment selected by **Specification of labels**. If no TextGrid file is found, the whole sound file will be used.

Specification of quantiles to be reported:
Quantiles of a segment will be reported by specifying values separated by commas or spaces. Values can be given between 0 and 1 (as fraction of a the whole range) or as percentages between 1 and 100. The program will always makes the reporting symmetrical to the lower and upper quantile range. Specifying quantiles will always force the median to be reported. For example, a specification like 0.05,10 will report the 5%, 10%, 50%, 90% and 95% quantiles. If no TextGrid file is found, the whole sound file will be used.

Nr. of formants to be computed:
Number of formants that should be used for the LPC-analysis. '5' is a usual value when the highest frequency for analysis is set to 5,000 Hz.

Nr. of formants to be reported:
Number of formants that should be reported in the output. Often, only the first two formants are used for further investigation and there is no need to report, for example, all five computed formants. It is advisable to report at least one more formant than are needed for an investigation to be able to spot 'shifted' formants (e.g., if the analysis added an additional formant between F1 and F2; in that case, the values for F3 are the real F2 values).

Highest formant frequency:
Upper bound for the formant computation. Usually, this is set to 5,000 Hz (PRAAT recommends 5,500 Hz for female voices, but I do not know of any documented investigation supporting that – my own research (to be published) shows a disadvantage for 5,500 Hz).

Data, next to formant frequencies, to be reported:
By default, the script reports (see **Result file** below) the file name, name of the segment, starting time of the segment, duration of the segment, the mean intensity of a segment, its mean pitch and the data selected. Up to 6 additional data can be reported by specifying letters separated by commas or spaces in this field (e.g. b,q) to report data for the particular point in time where a measurement takes place:

a Amplitude of the formant in dB. This value is normalised by the intensity of the signal at the selected time point. (The intensity is subtracted from the computed formant intensity and and arbitrary value of 80 dB is added. This will provide values that are reasonably compatible between different amplitude levels of a signal, although the values should not be treated with care. The arbitrary value of 80 dB was chosen to shift the dB values into a comfortable range (a 16-bit quantizised recording would be at 92 dB, but be experience, most recording are about 12 dB below that on the average; furthermore).

b (3 dB) Bandwidth of a formant. A smaller bandwidth indicates a 'sharper' formant peak.

i Intensity of the signal.

p pitch of the signal.

q Quality (= formant frequency divided by dormant bandwidth: $Q = F/B$) of a formant. A smaller Q-value indicates a 'sharper' formant peak independent of frequency.

t Time the center of the analysis window in seconds (absolute time in the sound file). Note that inspecting the same time point in PRAAT interactively usually gives slightly different results (and might change with the exact position and size of the signal window) since PRAAT computes the data on the fly and the positions of the (25 ms) analyses windows change. Consequently, different parts of the signal go into one analysis window depending on where the left edge of the signal display window (*Edit window* in PRAAT terms) is.

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**, **Minimal intensity** or **Pitch existence** criteria:

All:
All segments excluded from the computations are reported with File, Label, Start(s) and Duration(ms) only (see **Result file** below), all other values are set to the missing value symbol

(see below). This function can be helpful to see the context of a particular segment analysed, e.g. the listing will show the data for a vowel, but also the labels of the segments before and after it, which are not vowels.

Length, Intensity, Pitch exclusion:
All segments that fulfil the **Specification of labels** are reported, but fail either the length, intensity of pitch criteria. 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 below).

None:
No segments that are excluded because of any criteria are reported.
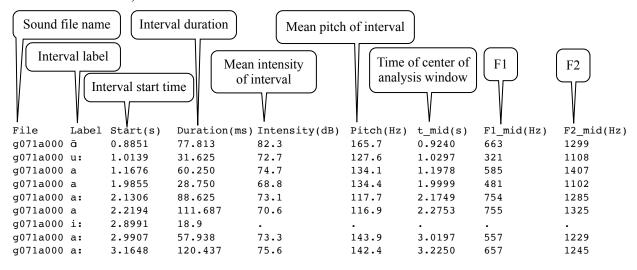
Symbol for missing values:
When PRAAT cannot compute a value it uses internally the string "- undefined -". This script replaces this string by the string given in this field. For a subsequent analysis of the data with JMP, the dot indicate missing values, for R it would be NA.

Result file:
The script generates a raw text file with tab-delimited data and a header line. The file name is of the form "formants_results_<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 "formants_results_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 2 formants with their time and reporting of excluded segments (due to duration, pitch or intensity) will look like (Note that the i: is excluded since it is shorter than 25 ms):

Sound file name, Interval label, Interval start time, Interval duration, Mean intensity of interval, Mean pitch of interval, Time of center of analysis window, F1, F2

```
File      Label  Start(s)  Duration(ms) Intensity(dB)  Pitch(Hz)  t_mid(s)  F1_mid(Hz)  F2_mid(Hz)
g071a000  ɑ̄      0.8851    77.813       82.3           165.7      0.9240    663         1299
g071a000  u:     1.0139    31.625       72.7           127.6      1.0297    321         1108
g071a000  a      1.1676    60.250       74.7           134.1      1.1978    585         1407
g071a000  a      1.9855    28.750       68.8           134.4      1.9999    481         1102
g071a000  a:     2.1306    88.625       73.1           117.7      2.1749    754         1285
g071a000  a      2.2194    111.687      70.6           116.9      2.2753    755         1325
g071a000  i:     2.8991    18.9         .              .          .         .           .
g071a000  a:     2.9907    57.938       73.3           143.9      3.0197    557         1229
g071a000  a:     3.1648    120.437      75.6           142.4      3.2250    657         1245

Analysis started: 15-Apr-20 10:25:10
Tier: 4
Labels: ipa
Minimal length: 25 ms
Minimal intensity: 40 dB
Pitch must exist: No
Formants computed: 5
Highest formants frequency: 5000 Hz
Window size: 25.6 ms
Pre-emphasis: 50
```

Information for programming:
Some parameters can be set underneath the 'form' section in the script. These are:

Directories:
The script uses internally separate strings for sound, TextGrid, result and support directories. Users who use separate directories for these can adjust these names in the script (or put them into PRAATs 'form' window).

Sound file names:
The default extension for sound files is ".wav". This parameter can be changed in the script.

Positioning of analysis window at the edge of a segment:
do_not_cross_segment_boundary = 0   The window is centred on the boundary
do_not_cross_segment_boundary = 1   The window stays within the segment (i.e., the left or right edge of the analysis window is on the edge)

Frequency unit:
The units that should be used for the analysis.
unit = 1   "Hertz"
unit = 2   "Bark"

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.
user_feedback = 0   Gives user feedback
user_feedback = 1   No user feedback

Noprogress string:
PRAAT itself reports its activity when computing pitch, intensity and formants. All these outputs can take substantial processing time, sometimes longer than the actual computation time. (This string is actually positioned where in PRAAT the   noprogress   is written.)
np_string$ = ""   (= empty string) normal PRAAT feedback
np_string$ = "noprogress"   No PRAAT feedback

Dummy data header 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.
dummy_data_header = 0   No dummy data line
dummy_data_header = 1   Dummy data line with "Dummy" for strings and "0.0" for numerals

Duration reporting:
duration_in_ms = 1   Duration is reported in milliseconds
duration_in_ms = 0   Duration is reported in seconds

Formant computing parameters:
pre_emphasis = 50.0
time_step = 0.0
window_length_ms = 25.6

Pitch computing parameters:
low_pitch = 75
high_pitch = 600

Current version and date:
4.0, 15-apr-2020

Known problems:
None

Planned extension:
Handling of point tiers

Contact:
reetz.phonetics@gmail.com