Formants_5_0_0.praat

Task:
This script opens all sound files in a directory and associated TextGrids (if they exist), computes formants' means, standard deviations and percentiles of intervals (or the whole files), and at time points along the intervals/file (i.e. a 'contour') the formants with their bandwidths, amplitudes, Q-values, and additionally the front values as z-scores or with their means subtracted. Intervals can be specified in various ways, and the results to a text file with the name "formant_results_<date>_<time>.txt". The length of the segment, 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 behavior of the script than displayed in the form window (the size of the form window fits into a screen with 640 points vertical resolution). These parameters can be set below the form statement in the script (see the **Programming**) section.

Parameters:

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

Tier number to be analyzed:
The number of the interval tier to be analyzed. Setting this value to '0' will treat whole files as intervals.

Number of formants to compute:
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.

Number 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).

Specification of labels:
The intervals that should be analyzed 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 (not 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 ːˑ̃ � are possible diacritics and attributes):
```
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.

Analyse only voiced segments:
If this field is checked, only intervals where PRAAT can compute a mean pitch will be reported.
(Note that a mean pitch might exist even though the interval is essentially voiceless; see
Pitch_notes.pdf for more information.)

Interval parameters:
A list of single letters, separated by commas or spaces, which specify what data should be reported
for the intervals specified by the **Specification of labels**. If no TextGrid file is found, the center of a
sound file will be used.

i   Mean intensity of the interval

p   Mean pitch of the interval (see *Pitch_notes.pdf* for the interpretation of mean pitch values).

m   Mean and standard deviation for the formant centre frequencies of the intervals

<numerical value>   Quantiles (actually percentiles) of the formant centre frequencies of
the intervals (more than one value can be specified, separated by commas or spaces). The
numerical values are percentiles between 0 (Minimum) and 100 (Maximum), where 50 is the
Median. The script always adds the symmetrical values to the lower and upper quantile
range. For example, a specification like 5,10 will report the 5%, 10%, 90% and 95%
percentiles of the formant values within a particular interval.

Number of time points for an interval:
The script can report 'contour' data (see **Contour parameters** below), that is, data along time
points within an interval. '1' time point will only report data at the center of an interval, '2' will
report only data from its left and right edges, '3' left edge, center, and right edge, etc. With 4 or
more time points, the header for the data in the result file will be in the form
<data type>_t<percentage>%
(e.g. "f_t10%" for the formant frequency at 10% of the duration of an interval).
The "t" in front of the percentage indicates that the percentage is a "time" percentage to make it
distinctive from the quantile percentages of an **Interval parameter**.
Note that the number of time points is different from a number of intervals. That is, value of '4' will
report data at 0%, 33.33%, 66.67% and 100% of the duration of an interval.
(There is a parameter **cross_interval_boundary** in the script that controls the positioning of the
analysis window at the edges of an interval. If this parameter is set to 0, the analysis windows will
be centered <u>on</u> the edges of a segment. (But in case the analysis window will go beyond the
beginning or end of a sound file, the window is not repositioned and the data might be 'undefined')
If this parameter is set to 1 (<u>not</u> the default), the analysis window is positioned to stay inside an
interval. The exact location of the center of an analyzing window can be reported with the **time**
request.)

Contour parameters:
By default, the script reports (see **Result file** below) the file name, label of the interval, starting time of the interval, and duration of the interval. Additional data can be reported by specifying letters separated by commas or spaces about the interval in the **Interval** parameters field. In the **Contour parameters** field data for the particular points where a measurement takes place within an interval:

a   Amplitude of the formant in dB. This value is normalized by the intensity of the signal at the selected time point. (The intensity is subtracted from the computed formant intensity and arbitrary value of 80 dB is added.) This provides 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 quantized recording would be at 92 dB, but be experience, most recording are about 12 dB below that on the average). Details of the procedure are given at the end of the **Programming** section below.

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

i   Intensity of the signal.

p   Pitch of the signal.

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

s   Subtract mean: the mean formant frequency of the interval is subtracted from the formant frequency at the reported time point.

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.

z   z-score: the mean formant frequency of the interval is subtracted from the formant frequency at the reported time point and divided by the standard deviation of the interval.

Reporting of skipped intervals:
Handling of segments which are either excluded by the **Specification of labels** or because they do not fulfill the **Minimal length**, **Minimal intensity** or **Pitch existence** criteria:

All:
All intervals excluded from the computations are reported with File, Label, Start(s) and Duration(ms) only, all other values are set to the missing value symbol (see **Result file** below). This function can be helpful to see the context of a particular segment analyzed, 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 fulfill the **Specification of labels** are reported, but fail either the length, intensity or pitch criteria. Only File, Label, Start[s] and Duration[ms] are reported, all other values are set to the missing value symbol (see **Result file** below).
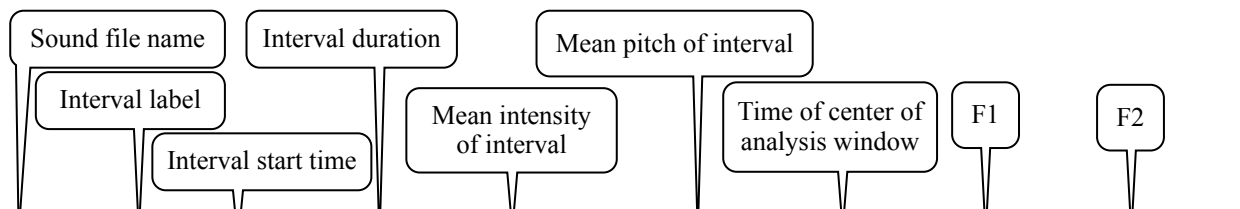
None:
No segments that are excluded because of any criteria are reported.

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 at the middle of an interval and reporting of excluded intervals (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 duration   Mean pitch of interval

Interval label   Mean intensity of interval   Time of center of analysis window   F1   F2

Interval start time

```
File       Label  Start(s)   Duration(ms) Intensity(dB) Pitch(Hz)  t_mid(s)   F1_mid(Hz)  F2_mid(Hz)
g071a000   ɑ̃     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

Script: Formant_5_0_0.praat
Analysis started: 5-Jan-21 12:41:07
Tier: 4
Labels: ipa
Minimal length: 25 ms
Minimal intensity: 0 dB
Pitch must exist: No
Analysis crosses interval boundaries: Yes
Formants computed: 5
Highest formants frequency: 5000 Hz
Step rate: 5 ms
Window size: 25.6 ms
Pre-emphasis: 50 Hz
```

Example of the header line in the output for a maximal number of parameters selected (only two formants, only 2.5% / 97.5% percentiles, and 4 contour positions). Note that the sequence of the specification of the parameters is not matching the output sequence, which always comes in a fixed order (but will lack columns, if less parameters are requested).

For better readability, the output is broken down in chunks of maximally four items. The real output is in one long line for each interval. The contour parameters are given in this example in percentage of the interval (e.g. t33.33%) and not as a position (e.g. t2), which is controlled by the parameter 'position_in_percentage' (see Programming section). Additionally, the contour percentages/ positions are indicated by the 't' prior to the percentage or position to separate them von the quantiles of the formant values for the whole interval. That is, "F1_33.33%" would be the 33.33% percentile of all F1 values in one interval, whereas "F1_t33.33%" is the F1 value at 33.33% of the total interval duration.

1st: basic information: File name, label, beginning and duration of interval
```
File            Label           Start(s)          Duration(ms)
```

2nd: average of the intensity and pitch of the interval
```
Intensity_mean(dB)              Pitch_mean(Hz)
```

3rd: average and standard deviation of all formants of the interval
```
F1_mean(Hz)  F1_stdev(Hz)    F2_mean(Hz)       F2_stdev(Hz)
```

4th: quantile information about all formants of the interval
```
F1_2.50%(Hz) F1_97.50%(Hz)   F2_2.50%(Hz)      F2_97.50%(Hz)
```

5th: time points of the contour of the interval
```
t_t0%(s)     t_t33.33%(s)    t_t66.67%(s)      t_t100.00%(s)
```

6th: intensity at these time points of the interval
```
i_t0%(dB)    i_t33.33%(dB)   i_t66.67%(dB)     i_t100.00%(dB)
```

7th: pitch values at these time points of the interval
```
p_t0%(Hz)    p_t33.33%(Hz)   p_t66.67%(Hz)     p_t100.00%(Hz)
```

8th: center frequencies of F1 at these time points of the interval
```
F1_t0%(Hz)   F1_t33.33%(Hz)  F1_t66.67%(Hz)    F1_t100.00%(Hz)
```

9th: bandwidths of F1 at these time points of the interval
```
B1_t0%(Hz)   B1_t33.33%(Hz)  B1_t66.67%(Hz)    B1_t100.00%(Hz)
```

10th: qualities of F1 at these time points of the interval
```
Q1_t0%       Q1_t33.33%      Q1_t66.67%        Q1_t100.00%
```

11th: amplitudes of F1 at these time points of the interval
```
A1_t0%(dB)   A1_t33.33%(dB)  A1_t66.67%(dB)    A1_t100.00%(dB)
```

12th: mean F1 subtracted from F1s at these time points of the interval
```
sF1_t0%(Hz)  sF1_t33.33%(Hz) sF1_t66.67%(Hz)   sF1_t100.00%(Hz)
```

13th: z-scores of F1s at these time points of the interval
```
zF1_t0%(z)   zF1_t33.33%(z)  zF1_t66.67%(z)    zF1_t100.00%(z)
```

14th: center frequencies of F2 at these time points of the interval
```
F2_t0%(Hz)   F2_t33.33%(Hz)  F2_t66.67%(Hz)    F2_t100.00%(Hz)
```

15th: bandwidths of F2 at these time points of the interval
```
B2_t0%(Hz)   B2_t33.33%(Hz)  B2_t66.67%(Hz)    B2_t100.00%(Hz)
```

16th: qualities of F2 at these time points of the interval
```
Q2_t0%       Q2_t33.33%      Q2_t66.67%        Q2_t100.00%
```

17th: amplitudes of F2 at these time points of the interval
```
A2_t0%(dB)   A2_t33.33%(dB)  A2_t66.67%(dB)    A2_t100.00%(dB)
```

18th: mean F2 subtracted from F2s at these time points of the interval
```
sF2_t0%(Hz)  sF2_t33.33%(Hz) sF2_t66.67%(Hz)   sF2_t100.00%(Hz)
```

19th: z-scores of F2s at these time points of the interval
```
zF2_t0%(z)   zF2_t33.33%(z)  zF2_t66.67%(z)    zF2_t100.00%(z)
```

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

Units for formant measurement:
unit$ = "hertz" (default); another possible value is "bark".

Formant computing parameters:
step_rate_in_ms = 5   (step rate for the analysis window in milliseconds)
pre_emphasis = 50.0   (in Hertz)
window_length_ms = 25.6   (window length in milliseconds)

Minimal length of a interval to be analyzed:
minimal_length_ms = 25   (default)
The minimal length of an interval (in milliseconds) to be analyzed. Setting this value to '0' will analyze all selected intervals (as selected by the **Specification of labels**).

Maximal length in seconds of a interval to be considered to be a analyzed
max_length = 1.0   (default)
This parameter raises a warning if unusual long intervals should be analyzed.

Minimal intensity for an interval to be analyzed in dB
minimal_intensity = 0 (default)
The minimal RMS-intensity (in dB) of an interval to be analyzed. 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 an interval). "0" means "no restrictions" (although 0 dB does not mean 'no amplitude', which would be minus infinity)

Position reporting in steps (1, 2, 3,…) or percentage (t0%, t25%…) of interval
position_in_percentage = 1   (default; the data header will be in the form "f_t25.00%")
position_in_percentage = 0   (the data header will be in the form "f_t2")

Maximal number of intervals for reporting the contour
max_number_of_measurements = 20 (default)
This parameter raises a warning if unusual many parameters should be reported

Pitch computing parameters:
low_pitch = 75
high_pitch = 600

Positioning of analysis window at the edges of an intervals:
cross_interval_boundary = 1   (default) 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   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.)

Reference amplitude to compute formant amplitudes
arbitrary_db = 80   (default)
The LPC-analysis using (accurate) root-solving methods to convert the internally computed reflection coefficients to formant center and bandwidth frequencies cannot provide the amplitude of a formant peak (less accurate Fourier-smoothing methods can provide amplitudes). To still provide amplitude values I use the following script part:

  call SelectObject intensity_obj
  intensity = Get value at time: mid, "Cubic"
  call SelectObject lpc_obj
  slice_obj = 'np_string$' To Spectrum (slice): mid, 20, 0, pre_emphasis
  ltas_obj = 'np_string$' To Ltas (1-to-1)
  amplitude = Get value at frequency: f'i_formant'_'i_pos',"Cubic"
  amplitude -= intensity + arbitrary_db

The arbitrary value of 80 dB was chosen to shift the dB values into a comfortable range (a 16-bit quantized recording would be at 92 dB, but be experience, most recording are about 12 dB below that on the average).

Directories:
directory$ = ""(default)
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. Note that these directory names have to end with a slash ("/").

Directory path:
path_name = 0(default; the full directory path is <u>not</u> reported in the result file)
path_name = 1(the full directory path <u>is</u> reported in the result file)

Extension of sound file names:
sound_ext$ = ".wav"   (default)

Separator for columns in the result file:
sep$ = tab$ (default; e.g. = "," for csv files)

User feedback:
user_feedback = 0   (default; no user feedback)
user_feedback = 1   (gives user feedback)
The script can report in the info window which file is being handled and the percentage of all files in a directory that have been handled. By setting this switch to "0", any output (other than error and warning messages) will be suppressed.

Noprogress string:
np_string$ = "noprogress" (default; no processing information from PRAAT)
np_string$ = ""   PRAAT will indicate processing
PRAAT reports its activity when computing pitch, intensity, or formants. All these outputs can take substantial processing time, sometimes longer than the actual computation time.

Dummy data header line:
dummy_data_header = 0   (default; no dummy data line)
dummy_data_header = 1   (dummy data line with "dummy" for strings and "0.0" for numerals)

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 potentially 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.

Duration reporting:
duration_in_ms = 1   (default; duration is reported in milliseconds)
duration_in_ms = 0   (duration is reported in seconds)

Symbol for missing values:
missing_value_symbol$ = "NA"   (default)
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".

Current version and date:
5.0.0, 5-jan-2021

Known problems:
None

Planned extension:
Handling of point tiers
More than one tier to select intervals/points
Allow spaces in label names

Contact:
reetz.phonetics@gmail.com