

The PRAAT script 'Pitch\_contour\_<version>\_<revision>.praat' can be used to compare (normalised) pitch contours of many segments.

The script computes the pitch of all .wav files in one directory/folder. For each file it reports the pitch data of each labelled segment in equidistant intervals (i.e. at time points). The data can be reported in Hertz, ERB,<sup>1</sup> or semitones.<sup>2</sup> The data can be the arithmetic mean (average) or median around the time point or the pitch value at the time point. Additionally, the data can be transformed into z-scores to normalise the pitch values or the means can be subtracted, which allows comparisons between speakers with different pitch ranges. The data is stored in a tab-delimited text file with a unique name 'pitch\_contour\_results\_<date>\_<time>.txt'.

The script reports missing values ("undefined" in PRAAT) with a "NA", but there is a variable "missing\$" at the beginning of the script where this can be changed (e.g. to "."). Likewise, there is a "dummy" variable (set to 0) which can be set to 1, which will generate a dummy line of data to force automatic data-type detection in JMP to set the correct data type.

The script cuts the pitch contour of each labelled segment into a (user-specified) fixed number of equidistant time points. That is, independent of the actual length of a segment in milliseconds each segment has a length of '100%' and by cutting it into e.g. 10 segments leads to 11 time points (at 0%, 10%, 20%, ..., 90%, 100%; see Figure 1); cutting it into e.g. 4 segments leads to 5 time points (0%, 25%, 50%, 75%, 100%). The pitch values are taken at these time points, or as a mean or median centred around this point in time (see Figure 1). Note, however, that the pitch algorithm of PRAAT moves with a fixed window size (= frame) with a fixed step rate (default: 5 ms) across the signal; i.e., there is actually not a pitch value for every point in time but rather for a certain stretch of time the same pitch value is used. Taking the value 'at a point in time' is done in PRAAT by interpolating (i.e. averaging) the value from the neighbouring frames. As a consequence, taking the median or mean will often lead to the same pitch value as taking it at a point in time. Be also aware that the first and last part of a segment is crossed the segment boundary for the centred mean or median (see Figure 1). Note further that the pitch algorithm of PRAAT takes some time to 'warm up' and it is not advisable to have the segment boundaries at the edges of a .wav file (i.e., do NOT cut the words you are interested in out of a continuous recording into separate .wav files, but label them in a recording and analyse the whole file(s).)

Using semitones or ERB values allows the comparison of speaker with very different pitch ranges and is the preferred method. Using z-scores additionally normalises each segment by transforming the pitch values to a distribution around the a mean of '0' and the standard deviation of '1' by the formula:

$$z = \frac{F_0(\text{at time point}) - F_0\text{mean}(\text{of segment})}{F_0\text{stdev}(\text{of segment})}$$

Using z-score transformed semitone values to compare pitch contours is an often applied method but can lead to misleading results if a contour is rather flat (cf. <https://sites.google.com/site/tonemodelling/anaposts/z-transformdoesnotworkforpitchcontoursflat>). The normalisation by subtracting the mean only seems to be a more appropriate method, but I suggest to apply this method only to ERB or semitone scaled data (and not to Hertz values directly):

$$z = F_0(\text{at time point}) - F_0\text{mean}(\text{of segment})$$

---

<sup>1</sup> Equivalent Rectangular Bandwidth

<sup>2</sup> With respect to 1 Hertz. The absolute semitone values depend on this reference frequency (1 Hertz), but relations between values are independent of the reference frequency and give identical contours / relations.

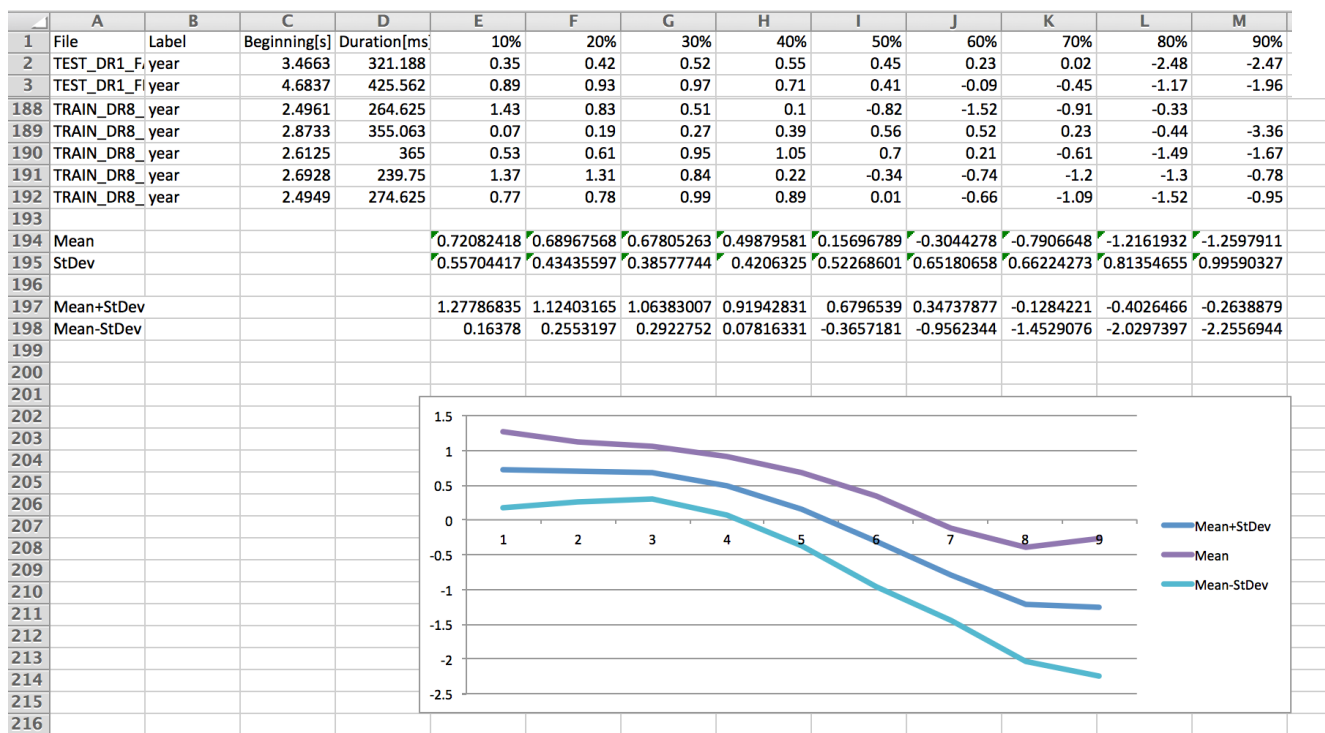
The tab-delimited result file has the form of:<sup>3</sup>

File	Label	Beginning[s]	Duration[ms]	0%	25%	50%	75%	100%	(z_score normalizing at position in ERB) <sup>4</sup>
SA2	don	0.8267	167.500	NA	-0.87	0.01	0.75	1.90	
SA2	ask	0.9942	255.187	-1.86	-0.38	1.66	NA	NA	
SA2	me	1.3250	120.000	1.98	0.81	0.22	-0.98	-1.63	

- File: Name of the .wav file of which the pitch was computed.
- Label: Label of the analysed segment.
- Beginning: Beginning of the segment in seconds (helpful to locate a segment in a file, especially if there are more than one identical label per file).
- Duration: Duration of the segment in milliseconds.
- 0%: Pitch data at the beginning (= left edge) of the segment.
- 25%: Pitch data at 25% of the duration of the segment.
- 50%: Pitch data at the middle of the segment.
- 75%: Pitch data at 75% of the duration of the segment.
- 100%: Pitch data at the end (= right edge) of the segment.
- (z\_score normalizing at position in ERB): Information about the selected method and units.
- A data value 'NA' indicates a missing value.<sup>5</sup>

Example of post-processing the data (here: z-scores of ERBs):

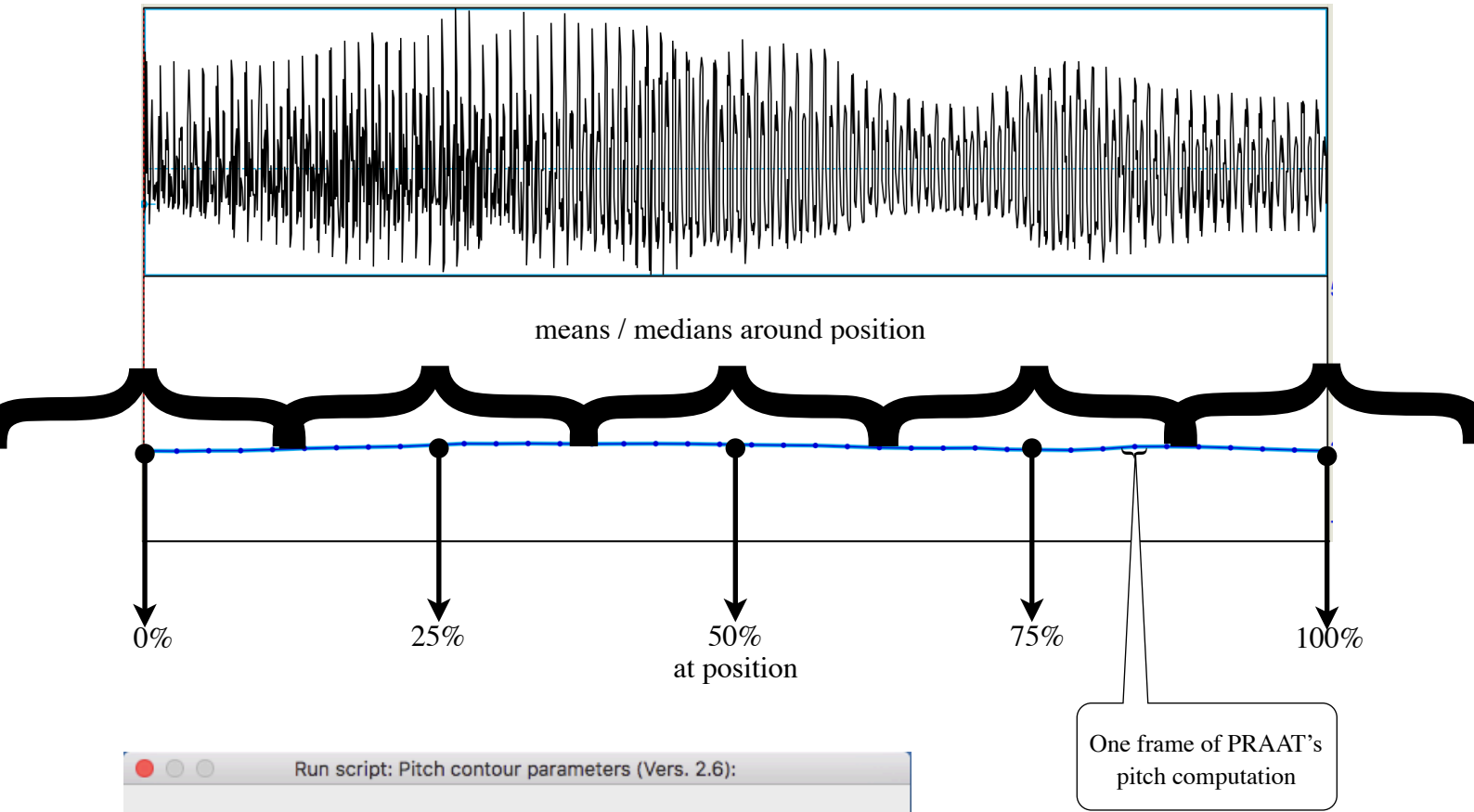
After selecting one segment from all files and putting the data into Excel, computing the mean and standard deviation and displaying differences between mean and st.dev. in a graph, the average pitch contour of (in this case: 191 items) can be displayed (with st.dev. range):



<sup>3</sup> An dummy data-line after the header can be added by setting the variable "dummy" to 1 at the beginning of the script. This dummy data-line will be in the form  
dummy dummy 0 0 0 0 0 0

<sup>4</sup> The string "at position in ERB)" is part of the previous row and is displayed separately here only for better readability.

<sup>5</sup> This character can be changed by assigning a different string to the variable "missing\$" at the beginning of the script.



Run script: Pitch contour parameters (Vers. 2.6):

Leave the directory path empty if you want to use the current directory.

Which tier should be analyzed?  
 tier:

---

Number of intervals:

Unit:  Hertz  
 ERB  
 semitones

Method:  mean around position  
 median around position  
 at position

Normalizing:  none  
 z\_score  
 subtract mean

Fast processing (no screen output)?  
 fast

---

step rate:

F0 range:

- Path for .TextGrid, .wav, and result files
- Data of labelled Segments in this tier will be reported
- Number of intervals (i.e. n+1 time points)
- Units pitch values
- Method to compute pitch values around or at a point in time
- Normalizing method
- 'Fast' computation suppresses activity messages, which makes computation much faster, but there can be no visible activity for several minutes.
- Frame advance rate (in sec) of PRAAT's pitch algorithm
- Lower and upper pitch boundaries (in Hertz) of PRAAT's pitch algorithm