METHODOLOGY
European Journal of Research Methods for the Behavioral and Social Sciences

PsychOpen GOLD

Tutorial

Check for updates

Extracting Vocal Characteristics and Calculating Vocal Synchrony Using Praat and R: A Tutorial

Désirée Schoenherr[1], Alisa Shugaley[1], Franziska Roller[1], Lukas A. Knitter[1], Bernhard Strauss[1], Uwe Altmann[1]

[1] *University Hospital Jena, Institute of Psychosocial Medicine, Psychotherapy and Psychooncology, Jena, Germany.*

Methodology, 2023, Vol. 19(3), 283–302, https://doi.org/10.5964/meth.9375

Received: 2022-04-27 • **Accepted:** 2023-07-12 • **Published (VoR):** 2023-09-29

Handling Editor: Katrijn Van Deun, Tilburg University, Tilburg, The Netherlands

Corresponding Author: Bernhard Strauss, Jena University Hospital, Institute of Psychosocial Medicine, Psychotherapy and Psychooncology, Stoystr. 3, 07743 Jena, Germany. Tel: 0049 36419398020. E-mail: desireeschoenherr@gmx.de

Supplementary Materials: Data [see Index of Supplementary Materials]

Abstract

In clinical research, the dependence of the results on the methods used is frequently discussed. In research on nonverbal synchrony, human ratings or automated methods do not lead to congruent results. Even when automated methods are used, the choice of the method and parameter settings are important to obtain congruent results. However, these are often insufficiently reported and do not meet the standard of transparency and reproducibility. This tutorial is aimed at researchers who are not familiar with the software Praat and R and shows in detail how to extract acoustic features like fundamental frequency or speech rate from video or audio files in conversations. Furthermore, it is presented how vocal synchrony indices can be calculated from these characteristics to represent how well two interaction partners vocally adapt to each other. All used scripts as well as a minimal example, can be found on the Open Science Framework and Github.

Keywords

reproducibility, vocal synchrony, Praat pitch extraction, nonverbal synchrony, speech rate

Human behavior can be categorized into linguistic (e.g., speech), extra-linguistic (e.g., body movements) and paralinguistic features (e.g., fundamental frequency) (Tonti & Gelo, 2016). With regard to interaction patterns in psychotherapy, research is currently focusing increasingly on nonverbal behavior (extra-linguistic and paralinguistic features). The advantages of recording paralinguistic features are that they can be measured automatically and in a non-invasive manner and can provide decisive information for diagnostics and process evaluation. In addition to efforts to identify nonverbal correlates of mental disorders for diagnostic purposes, more and more attention is being paid to investigating correlations between therapeutic success and the nonverbal synchrony of patient and therapist (Altmann et al., 2020; Galbusera et al., 2018; Paulick et al., 2018; Ramseyer & Tschacher, 2011; Schoenherr, Paulick, Strauss, et al., 2019b). Nonverbal synchrony refers to the interrelation of behavior, emotions or other nonverbal aspects of communication (Altmann et al., 2020; Paxton, 2015).

In recent years, movement synchrony has been established as a therapy-relevant factor for the prediction of therapeutic alliance and outcome (Altmann et al., 2020; Lutz et al., 2020; Paulick et al., 2018; Schoenherr, Paulick, Strauss, et al., 2019b; Schoenherr et al., 2021). However, different approaches within the research groups led to inconsistent findings in some cases. Furthermore, the scripts were not accessible to a large number of people, which greatly limited the transparency and replicability of the results. Some methodological studies pointed out that especially the parameter settings and methods used were significantly related to the found results (Luehof, 2019; Schoenherr, Paulick, Strauss, et al., 2019a; Schoenherr, Paulick, Worrack, et al., 2019). In a number of disciplines, there was a call for reproducibility of study results, so in psychology (Munafò et al., 2017). Since then, scripts and packages are increasingly published which allow replicability and provide the greatest possible transparency of the procedures of different working groups (Kleinbub & Ramseyer, 2018).

Although the emotional state of a person can be determined on the basis of vocal characteristics and these are subject to special attention in psychotherapy (Banse & Scherer, 1996), only few studies exist with regard to the influence of paralinguistic characteristics within psychotherapy. Vocal synchrony indicates how often the interaction partners refer to each other on a paralinguistic level (e.g., pitch alterations, speech rate adaption). Studies show a positive influence of vocal synchrony on alliance or outcome (Imel et al., 2014; Pérez-Rosas et al., 2017; Rocco et al., 2017; Spivack, 1996; Wieder & Wiltshire, 2020), a negative impact of vocal synchrony on outcome (Reich et al., 2014) or no effects on empathy (Gaume et al., 2019). Previous studies on the influence of vocal synchrony are mostly single case analyses with limited generalizability (Tomicic et al., 2015). One reason for this could be that the evaluation of audio files from the psychotherapy context is often time-consuming and cannot be compared to classical phonetic analyses. Additionally, researcher used different methodologies or vocal synchrony operationalizations, for example Reich et al. (2014) used correlations of aggregated acoustical features of speaker turns to predict depression whereas Imel et al. (2014) evaluated synchrony as correlations of random effects of a mixed model predicting acoustic features by the speaker or empathy values. Therefore, also the methodology might lead to different results. Unfortunately, only a few scripts for vocal synchrony calculation are available so far which complicates comparisons.

In recent years, more and more methods have been developed that allow an automated or semi-automated determination of different parts of speech within a conversation. A completely manual segmentation of the speech parts is extremely time-consuming and costly, but has shown the most valid results so far. Fully automated methods have higher segmentation error rates. Recent studies of semi-automated methods show a significant reduction in error rates due to a learning phase preceding the algorithms in which manual coding of the speech parts is performed (Fürer et al., 2020).

Aim of the Tutorial

This paper offers a tutorial for the segmentation, annotation and transcription of audio tracks, script-based extraction of paralinguistic features and calculation of an exemplary vocal synchrony score. This allows achieving comparable results between research groups, to perform replications or to transparently reproduce the procedure. In contrast to common phonetic analysis in Praat, the tutorial provides scripts that have been specially developed for the interaction context. Thus, they go beyond the previous procedures and can be of great use for communication scientists and especially for psychotherapy process researchers.

Overview Process of Feature Extraction

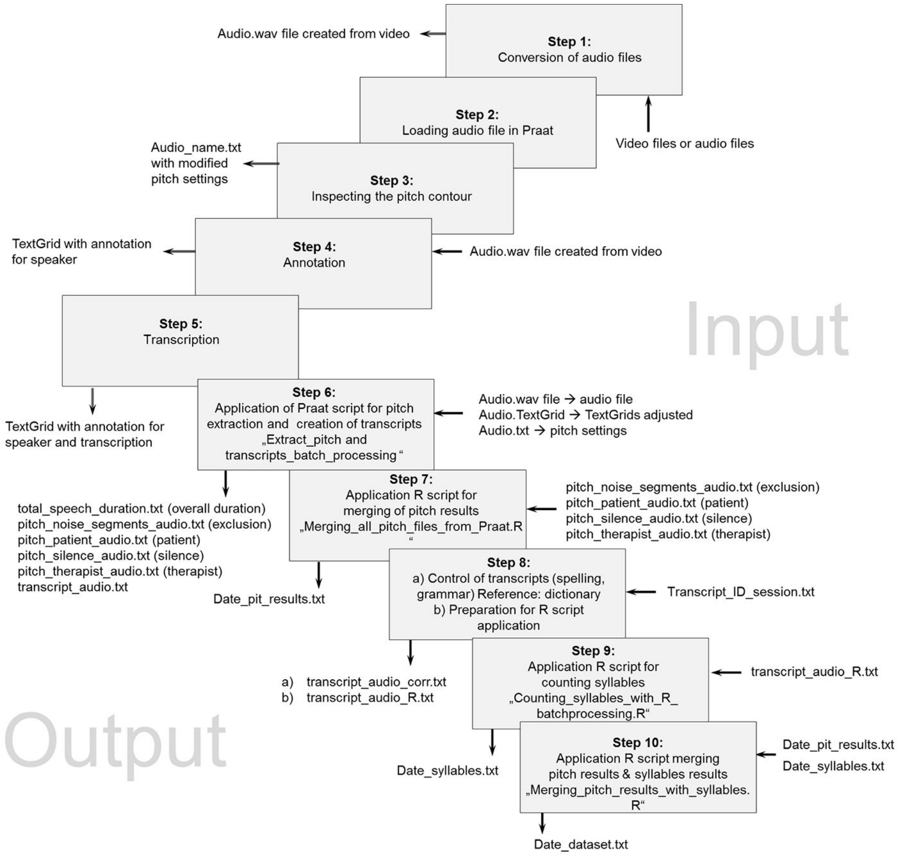
The corresponding steps to build a dataset containing syllable counts and pitch features are summarized in Figure 1. An overview about the steps is also provided in Table A1 in Appendix A of Schoenherr et al. (2023).

Required Software and Scripts

The current section shows which software and scripts are needed. If your input data are videos, you need the VLC Player to separate video and audio files. To segment, annotate and transcribe the audio files and for the extraction of vocal features, Praat is needed. Praat is open source and free software, which can be used mainly for speech analysis or labelling and segmentation (Boersma, 2001; Boersma & Weenink, 2007). It can be downloaded from https://www.fon.hum.uva.nl/praat/. Make sure to download the latest version of Praat for your operating system. To merge datasets and calculate vocal synchrony, R is needed. Please install the freeware software R (Version 4.0 or higher), it can be downloaded from (https://cran.r-project.org/bin/windows/base/). If you want to test the inter-rater reliability of the segmentation and annotation, first the application Easy-DIAg has to be downloaded and installed (https://sourceforge.net/projects/easydiag/).

Figure 1

Data Processing Steps From Audio File to Dataset Containing Pitch Features and Syllables of Each Speaker Turn


Audio.wav file created from video
Step 1:
Conversion of audio files
Step 2:
Loading audio file in Praat
Video files or audio files
Audio_name.txt
with modified
pitch settings
Step 3:
Inspecting the pitch contour
TextGrid with annotation
for speaker
Step 4:
Annotation
Audio.wav file created from video
Step 5:
Transcription
Input
Step 6:
Application of Praat script for pitch extraction and creation of transcripts „Extract_pitch and transcripts_batch_processing"
Audio.wav file → audio file
Audio.TextGrid → TextGrids adjusted
Audio.txt → pitch settings
TextGrid with annotation for speaker and transcription
Step 7:
Application R script for merging of pitch results „Merging_all_pitch_files_from_Praat.R"
pitch_noise_segments_audio.txt (exclusion)
pitch_patient_audio.txt (patient)
pitch_silence_audio.txt (silence)
pitch_therapist_audio.txt (therapist)
total_speech_duration.txt (overall duration)
pitch_noise_segments_audio.txt (exclusion)
pitch_patient_audio.txt (patient)
pitch_silence_audio.txt (silence)
pitch_therapist_audio.txt (therapist)
transcript_audio.txt
Date_pit_results.txt
Step 8:
a) Control of transcripts (spelling, grammar) Reference: dictionary
b) Preparation for R script application
Transcript_ID_session.txt
Step 9:
Application R script for counting syllables „Counting_syllables_with_R_batchprocessing.R"
transcript_audio_R.txt
a) transcript_audio_corr.txt
b) transcript_audio_R.txt
Output
Date_syllables.txt
Step 10:
Application R script merging pitch results & syllables results „Merging_pitch_results_with_syllables.R"
Date_pit_results.txt
Date_syllables.txt
Date_dataset.txt

Note, that Matlab (https://de.mathworks.com/downloads/) is required for the usage of this application. By installing the application, it is stored as an application in Matlab. More details can be found in the EasyDIAg manual which is also downloaded by installing the application. Furthermore, ELAN, open source freeware software is needed (https://www.softpedia.com/get/Multimedia/Video/Other-VIDEO-Tools/ELAN.shtml). All used scripts in this tutorial are publicly available at Schoenherr (2020) and Schoenherr (2022).

Preparing Audio Files for Praat (Step 1–3)

Before voice frequency and other prosodic parameter can be extracted, the conversion of the files, loading the files in Praat and inspecting the pitch signal referring to noise are important preprocessing steps.

Conversion (Step 1)

Often the conversation is recorded with a video. In this case the audio tracks must be separated from the consisting video file first. After the VLC Media Player has been opened, click on *Media* in the menu and then on *Convert/Save*. This opens a window, where you can click the *Add* button to select the appropriate video to the list for conversion. Then select the *Convert/Save* button at the bottom of the window. In the newly opened window, the source of the video is now readable and the conversion settings and the destination have to be entered. Select *Settings > Convert* and *Profile > Audio-CD*. The checkbox *Display the Output* and *Deinterlace* are not necessarily required. Finally click on *Browse* (at *Destination file*), save the file name and the file type (preferably .wav) and click on the *Start* button. The VLC media player starts the conversion. It is important not to close the window until the conversion is done (see timeline), otherwise, the audio track will not represent the full length of the video. Make sure that you convert all files to a coherent file format to ensure comparability of the audio signal (recommended sampling frequency 44100 Hz).

Adding Files to Praat (Step 2)

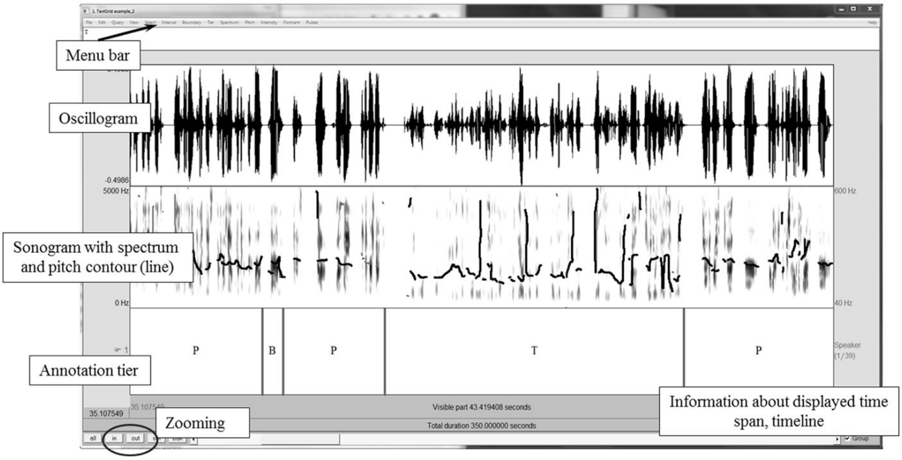
Starting Praat will open two windows: the *Praat Objects* window (main window) and the *Praat Picture* window. The *Praat Picture* window is not needed for our purposes. To create an annotated audio file, you have to open the sound file by clicking on *Open > Read from file* and select the previously converted file (file type .wav). Next, a so called TextGrid is created which contains the manual annotations who are speaking. With annotation, we mean a labelling of the speaker for each speaker turn. These annotations are needed to differentiate between the acoustic features of different persons of the audio file. Such TextGrid is created by activating the command *New > Create TextGrid*. This will bring up another window, where *Start time* and *End time* are used to enter the length (in seconds) for creating the TextGrid. In the field *All tier names* the name of the annotation line can be specified (e.g., Speaker). The field *Which of these are point tiers?* remains empty. Afterwards confirm with *Ok*. The sound file and the created TextGrid should now be available under *Objects*. Before further processing, the sound file should first be converted into a mono signal because it saves space. Language is mostly recorded as mono signal (one channel); thus, there is no need for a stereo signal (with two channels). To accomplish it, mark the sound file and select *Convert > Convert to mono* from the commands shown on the right-hand screen. Next, select the new mono-converted object and the TextGrid and press *View & Edit*. This opens the actual annotation/transcription window. When closing this window, only the TextGrid has to be marked and saved via *Save > Save as text file*.

Inspection of the Pitch Signals and Adjusting the Pitch Settings (Step 3)

In the annotation window, the audio file can be viewed, the speaker parts can be labeled (segmented) and the speech can be transcribed. Figure 2 shows such a window including an example audio track. In the tab toolbar (menu bar) different options can be selected, which determine what should be displayed in the particular areas. In the upper part of the editor window the oscillogram should be displayed and in the area below the sonogram of the audio file. The sonogram can be deactivated or activated via *Spectrum > Show spectrogram*. For our annotation and transcription, we do not necessarily need the spectrogram and by deactivating it computational time is reduced. Additionally, the sonogram shows the fundamental frequency profile f0 (Praat: blue line; Figure 1: black line). For long sequences it might be necessary to zoom into this area first (via the *in* button in the lower-left corner of the editor window) to make the spectrogram visible because Praat shows this only for a segment of 10 seconds. If no blue line is shown, it must be activated in the tab *Pitch > Show pitch*.

Figure 2

Three-Part Editor Window With Oscillogram, Sonogram With Pitch Contour and Annotation Tier


1. TextGrid example_2
File Edit Query View Select Interval Boundary Tier Spectrum Pitch Intensity Formant Pulses Help
Menu bar
Oscillogram
-0.4986
5000 Hz
600 Hz
Sonogram with spectrum and pitch contour (line)
0 Hz
40 Hz
P
B
P
T
P
Speaker (1/39)
Annotation tier
35.107549
Visible part 43.419408 seconds
Total duration 350.000000 seconds
Information about displayed time span, timeline
Zooming
all
in
out
Group

The pitch signal may need to be adjusted if the audio contains too much (background) noise, visible octave jumps in the fundamental frequency or a displayed very high tuning frequency. Otherwise the pitch feature extraction is biased containing some pitch signals which are not from the speakers. Basic noise is no problem if it does not change the fundamental frequency. Typically, audio files that are recorded using microphones in contrast to be extracted from video files do not need these setting changes. Figure 3A shows a drastically changed fundamental frequency because a noise line is still visible in the lower part of the spectrogram. This noise line can either be modified by adjusting the pitch settings or, if this fails, it can be coded as an exclusion segment. Sometimes it can be advisable to conduct a noise reduction with the software Audacity or exclude very noisy audio files. Octave jumps can be recognized by the fact that the fundamental frequency does not make audibly large jumps, but a rapid drop can be seen in the pitch signal. Again, the pitch settings should be changed so that the jumps disappear (see Figure 3B).

To adjust the pitch settings click on *Pitch > Advanced pitch settings.* In the new opening window the parameters *Voicing threshold*, *Octave cost*, *Octave-jump cost* should be increased step by step until the pitch signal appears appropriate. At first, start modifying Octave cost. If after some upward steps no improvement of the pitch signal is visible, Octave-jump cost can be increased step by step. If this still does not clear the signal from octave jumps or noise, try to increase the Voicing threshold slowly. It has to be decided how high the pitch parameters should be set, because with higher values the noise decreases, but also the overall displayed pitch signal decreases. Please note that for each new opened audio file the pitch settings have to be reset to the default setting (Voicing threshold: 0.45, Octave cost: 0.01, Octave-jump cost: 0.35) to avoid distortion of the pitch signal. This can be done by clicking *Pitch > Advanced pitch settings > Standards > OK.* It is also required to document any changes in the pitch settings. With respect to the analysis, altered pitch settings have to be stored in a .txt file with space as field separator string in the following order: voicing threshold, octave cost, octave-jump cost (e.g., 0.5 0.05 0.40). The .txt file has thereby been named equally to the audio file and TextGrid file. In case of the problem with selectively very high voice frequencies (Figure 4), it only can be decided by personal sighting whether the person is really talking that high or background noise is the reason. If the high pitch signal is caused by background noise, the segment should be declared as an exclusion segment.

Figure 3

Examples of Background Noise and Diplophony as Indicators of Altered Pitch Contour

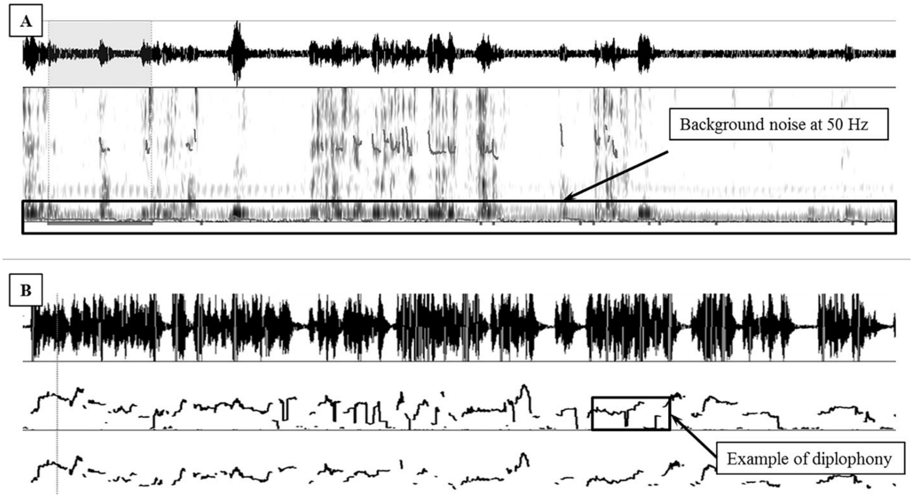
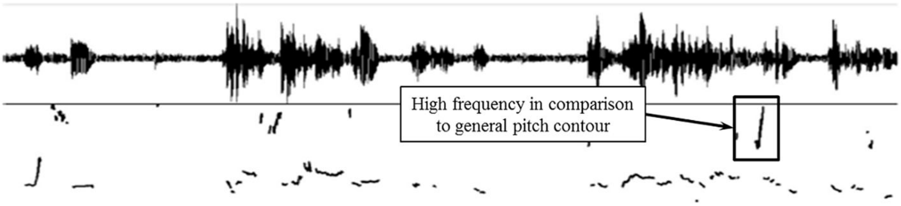


Figure 4

Example of a High Frequency in Pitch Contour



Segmentation, Annotation & Transcription (Step 4–5)

Due to the fact that most audio files contain speaker turns of both interacting persons (e.g., therapist and patient), an annotation of different speaker turns is necessary so that the measurements of acoustic features can be assigned to the concerning person. The processing time of one file strongly depends on the audio quality. In our experience the time needed to annotate, segment and transcribe a file varies between 6 to 8 times of the length of the audio file (e.g., 3 to 4 hours coding time for a 30-minute audio file).

Segmentation and Annotation Using Coding Rules (Step 4)

Segmentation means that the different speaker turns have to be separated in the sense that a boundary is set indicating the start of a speaker turn and the end of a speaker turn. To begin segmentation, click on the yellow area in the lower part of the TextGrid editor window. With the tabulator key you can play or pause the audio track. In parallel, the segment boundaries can be set with *ENTER*. By clicking into particular segments (selected segment appears yellow) and pressing the tab key the individual segment can be listened to. Segment boundaries can be moved via drag and drop to the desired position. If a particular section (independent of the segment boundaries) is to be listened to, the relevant area in the oscillogram or sonogram can be marked (hold left mouse button) and press the tab key to start the playback. Segments can be further subdivided by clicking with the left mouse button at a position between the segment boundaries in the oscillogram or sonogram and then clicking into the appearing circle (on the line between sonogram and tier) or pressing *ENTER*. Boundaries can be deleted with *ALT + Backspace* or via *Boundary > Remove*. Below the title of the tier, an indication of the segment number is given in brackets. The input field for annotation and transcription can be found below the menu bar. To annotate the audio track, click on a segment in the annotation area. Then enter the appropriate speaker code (e.g. T = therapist, P = patient, S = silence, B = overlapping speech or noise segments) in the input field below the menu bar. In order for the R script to run later, each segment must contain a code. In Table B1 in Appendix B of Schoenherr et al. (2023), the coding rules used by our working group are exemplarily presented. We decided that breaks longer than 1.5 seconds are rated as silence, i.e., long pauses in accordance with Watts (1989). The TextGrid can be saved by pressing *CTRL + S*, or by choosing the TextGrid in the Praat Objects window and then selecting *Save > Save as textfile*. The file extension of the TextGrid must always be .TextGrid.

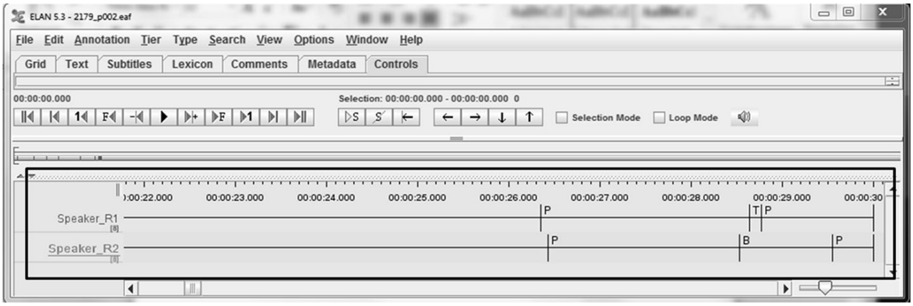
Investigation of Inter-Rater-Reliability

We recommend the annotation by multiple raters to reduce bias to one rater and rating time. Furthermore, the raters should be trained referring to the coding rules at the beginning and the inter-rater-agreement should be investigated as quality check. Common methods to investigate inter-rater agreement are based on event sequential data, that is, for a known event, ratings from different raters are conducted and compared to each other. The events have a known onset and offset and only the rating which event took place is done by the rater. That would mean, that segments are pre-defined and only annotations have to be set by the rater. However, our segmentation and annotation are based on timed-event sequential data, which means that also the start and end of an event is coded by the rater. Therefore, not only the rating of the event (annotation of specific speaker code) but also the start and end of the event (segmentation) might vary between different raters. The inter-rater reliability of this kind of data can be examined using the Matlab application EasyDIAg (Holle & Rein, 2015).

We recommend that the different raters of the study annotate a subset of the study material (e.g., first 30 seconds of $N = 10$ audios) so that annotations of all raters are available for each audio file. We show the calculation of inter-rater-reliability using two different raters. First, annotations of different raters have to be saved within the concerning annotation file. Therefore, import the TextGrid file of audio file 1 into ELAN by opening ELAN and clicking *File > Import > Praat-TextGrid file...* Please ensure that encoding is set to UTF-16. If TextGrid file is displayed, rename Speaker by clicking on the text with the right mouse key and choosing *Speaker > Change attributes of Speaker* and change the tier name to Speaker_R1. After this, the annotation of the second rater has to be imported by *File > Merge transcriptions ....* There, you choose the TextGrid of the second rater for audio file 1 as second source and click *Next*, after this *Finish*. After importing the second annotation line, please change the name of the tier to Speaker_R2. Both annotation lines appear (see Figure 5). Please ensure that the annotations have exactly the same length, if not trim the annotation lines by clicking on the last annotation with the right mouse key, choose *Modify Annotation Time* and set the end to an equal time point (we recommend 30 seconds). By that way you also have a visual impression about the concordance of speaker turn boundaries.

Figure 5

Two Annotation Lines (Black Rectangle: Annotations of Speaker_R1 and Annotations of Speaker_R2) Shown in ELAN



Save file as ELAN file (extension .eaf). Repeat the procedure for all audio files. After this, all ELAN files containing the different annotations of two raters each have to be exported. Therefore, click *File > Export Multiple Files as ... > Tab delimited Text ....* Define *New Domain* by choosing all .eaf files and hit *OK*. In the next window, you set checks to *Speaker_R1* and *Speaker_R2*. Additionally, checks should be set to *Include file path column*, *Begin Time*, *End Time*, *msec*. After this, the exported file has to be imported in the EasyDIAg application. Open the application and import the ELAN file by *File > Import ELAN File*, choose *Overlap* (we used 51) and hit *Update*. The application returns the values kappa, positive agreement, kappa max and raw agreement. We recommend using kappa. It can be interpreted similarly to Cohen's kappa.

Transcription for the Measurement of Speech Rate and Word Count (Step 5)

In order to measure paralinguistic parameters such as speech rate, a transcription of the conversation must be made. Speech parts in the existing tier (e.g., called *Speaker*) have been annotated, select *Tier > Duplicate tier*. This duplicates the entire boundaries of the *Speaker* tier, which facilitates further transcription of the speaker parts. Before starting the transcription, the duplicated tier must be renamed (e.g., *Text*) by clicking *Tier > Rename tier*. Alternatively, you can already enter the name of the second tier in the function window *Duplicate tier*. Now all segments can be replayed one at a time and the speech can be transcribed literally. For this, use the input field below the menu bar, just as for annotation. It is important to avoid quotation marks (e.g., in case of direct speech within the talk) during transcription (a colon can be used instead); otherwise, the Praat script will not work correctly. Other punctuation marks are not necessary; however, they facilitate proof-reading. Transcription rules of our working group are displayed in Table B2 in Appendix B of Schoenherr et al. (2023).

Feature Extraction with Praat (Step 6–7)

A script was used to extract the median, the 5% and 95% quantile of the pitch signal and the transcriptions for each speaker turn. To ensure a smooth compilation of the Praat script, a TextGrid file and, if necessary, a file with the modified pitch settings should be available for each audio file. These should be named equally except for the file extension. Please make sure that the file name does not contain any additional dots. Select the script in the Praat Objects window under the tab *Praat > Open Praat Script*. This will open a script editor that shows the script. Make sure that .txt files are written in UTF-8 encoding standard by choosing *Praat > Preferences > Text writing preferences* and set the encoding standard to UTF-8. Otherwise, the .txt file is not readable by R. Please save the script to your results folder. The output of the script will be written to the folder where the script was saved. By pressing *CTRL + R* or the *Run* button the complete script is executed. After starting the script, a window opens, where you have to enter the path of the files to be analyzed. It is important that the folder path ends with a backslash. Afterwards, all audio files in the folder are analyzed (with TextGrids and pitch setting files if necessary). The script computes six files for each audio file:

- pitch_patient_audioname.txt
- pitch_therapeut_audioname.txt
- pitch_silence_audioname.txt
- pitch_noise_segments_audioname.txt
- transcript_audioname.txt
- total_speech_duration_audioname.txt

All pitch files are semicolon-separated text files containing: name of the file; number of the segment, start of the segment in seconds, end of the segment in seconds, median pitch, minimum pitch, maximum pitch, 5% quantile pitch, 95% quantile pitch (header included). The standard value is 600 Hz. We used a lower ceiling value (400 Hz) because otherwise, background noise would have biased the pitch calculation. After conducting the script, check the output pitch values and make sure that values have a reasonable range; male voices average f_0 (100–125 Hz) and f_0 range (70–200 Hz); female adults average f_0 (180–220 Hz) and f_0 range (140–400 Hz); see Biemans (2000). If not, the pitch ceiling can be adjusted in the script by altering the value in Line 21

```
pitch_ceil = 400
```

of the Praat script

```
Extract_pitch and transcripts_batch_processing.
```

Troubleshooting

Sometimes errors within the segmentation, annotation and transcription process may lead to errors while compiling the Praat script. Some possible error messages are presented in Schoenherr et al. (2023), Table C1 in Appendix C. Additionally, the cause and solution of the problem are displayed.

Merging Results With R (Step 7)

The Praat script results in separate files with acoustic features for each audio file and each annotation. For merging all pitch results of one person (different annotations) and afterwards merging pitch results of all persons into one dataset, first all files have to be stored in a list by using

```
nam <- list.files(getwd(), "pitch").
```

Afterwards, a for loop loads all files, checks which annotation was made and builds a coherent dataset. Within the loop the ID is extracted, in our case it is a five-digit number within the audio name. If this is not the case for your data, please adapt the script found in Schoenherr (2020) and Schoenherr (2022).

Counting Syllables With R (Step 8–9)

Counting syllables manually is very time-consuming and prone to mistakes. Therefore, we imported the transcripts in R and realized counting using the R package *sylly*. Before doing so, transcripts have to be proofread and prepared for R analysis.

Proofreading and Preparation of the Transcript (Step 8)

Transcripts extracted by the Praat script are saved as text files (in Step 5). A line in the transcript file contains the segment number and the transcribed text. After the line break, the number and transcribed text of the next segment follows. Each transcript should be proofread, if possible, by a person who did not create the TextGrid in order to correct typos or spelling mistakes due to personal blind spots. During proofreading, it should be ensured that all numbers are written out in numerical words and not in Arabic numerals, otherwise, the number of syllables cannot be extracted correctly. Furthermore, polysyllabic words such as “okay” should not be abbreviated as “o.k.” or “ok”. In addition, attention should be given to a uniform coding of the interjections. For example, “uh-huh” / “mm-hmm” (meaning consent or yes) or “uh-uh” / “m-m” (meaning no) are uniformly transcribed as “mh mh” and a monosyllabic “hm” only as “mh”. If words are pronounced in dialect or short forms, they should be replaced by the grammatically correct form. It is a useful practice to save the corrected transcripts in a separate directory (preferably with the ending “_corr.txt”). In the further procedure in preparation for syllable counting in R, all double spaces must be removed from the corrected transcripts (*CTRL + H*, then type two spaces in *Search*, and one space in *Replace*). It is advisable to save the document in a separate folder under the same file name, but replace the extension “_corr.txt” with “_R.txt”.

Extraction of Syllables With R (Step 9)

Counting syllables with R is realized by using the R package *sylly.* An additional language support package has to be installed and loaded which refers to the language of the transcripts (e.g., for English transcripts use *sylly.en*, for German *sylly.de*, for Spanish *sylly.es*). Note that German transcripts have to be opened with Notepad++ and resaved with substituted umlauts. Otherwise, R would not be able to count syllables correctly. To use the R code, the packages `Hmisc`, `base` and `readtext` are needed, therefore they have to be installed, e.g.,

```
install.packages(“readtext”)
```

and loaded, e.g.,

```
library(readtext).
```

The transcript has to be read with

```
transcript=readtext(“path/transcript_audioname.txt”).
```

Afterwards, the text is split into lines containing the segment number and text of the concerning segment:

```
list_lines = string.break.line(transcript$text)
lines_text = unlist(list_lines)
```

Next, an ID variable is extracted from the audio name. In our case, every audio name file starts with a 5 digits ID of the patient (e.g., 12345).

```
ID = as.numeric(gsub("[^0-5]", "", “transcript_audioname.txt”))
```

Afterwards, a loop processes each line of the transcript, extracting the syllables for each segment. The loop starts with k = 2 because in the first line the header is displayed; script available at Schoenherr (2020) and Schoenherr (2022).

The resulting table `res_all` contains three columns: ID, segment, number of syllables. Results can be exported with:

```
currentDate <- Sys.Date()
txtFileName <- paste(currentDate,"_syllables.txt",sep = "")
write.table(res_all, file = txtFileName, dec= ".", sep = ”;”, row.names = F, col.names = T)
```

The script with complete batch processing of all transcript files in a folder and can be downloaded from Schoenherr (2020) and Schoenherr (2022).

Creating a Dataset With All Features (Step 10)

Pitch results and numbers of syllables can be merged using the R package `plyr`. First, both datasets have to be loaded using the function `read.table`:

```
pitch = read.table(file = "path/Date_pit_results.txt", dec = ".",
sep = ";", header = T, colClasses = c(ID = "numeric", segment = "numeric",
start = "numeric", end = "numeric", med_pit = "numeric", P5_pit = "numeric",
P95_pit = "numeric"), stringsAsFactors = F)

syllables = read.table(file = "path/Date_syllables.txt",dec = ".",sep = ";",
header=T, colClasses = c(ID="numeric", segment = "numeric",
N_syllables = "numeric"), stringsAsFactors = F)
```

Using a full merging of pitch results and numbers of syllables by ID and segments, both datasets are combined.

```
total = join(dataset_pitch, dataset_syllables, by = c("ID", "segment"), type = "full")
```

After combining, the duration of a segment and the speech rate within a segment can be calculated.

```
total$dur = total$End-total$Start
total$speed = total$N_syllables/total$dur
```

The dataset total contains eleven columns: ID, segment, start (in seconds), end (in seconds), median pitch, 5% quantil pitch, 95% quantil pitch, class (annotation: patient/therapist/noise segment or silence), number of syllables, duration of the segment (in seconds), speech rate (syllables/second). Please save the dataset afterwards using the write command.

Calculating Vocal Synchrony

We define vocal synchrony in accordance to Reich et al. (2014) as a linear relationship that is the correlation of aggregated acoustical features of speaker turns. It is important to note that we only use speaker turns that are not separated by other units, that is, speaker turns with coded silences or overlaps are not used for the computation of synchrony. In contrast to the computation of movement and physiological synchrony, a computation of vocal synchrony without time lag is not possible, since an interaction consists of alternating speaker turns, that is, cannot take place completely simultaneously. This is an important difference. Our construct of vocal synchrony most closely describes the strength of linear time-lagged dependence of paraverbal features in successive speaker turns. Thus, it can be considered a very global measure of vocal synchrony.

The required packages for the script are: `psych` and `utils`. Please, install these packages and load them using the library command. As the first step, the data is loaded using:

```
data_sync = read.table("path/Date_dataset.txt", header = TRUE).
```

After this, some preprocessing took place:

```
#transform variable as.factor
data_sync$class = as.factor(data_sync$class)
#calculate range of pitch
data_sync$range = data_sync$P95_pit-data_sync$P5_pit
#get vector ID variables
ID=unique(data_sync$ID)
```

Next, only speaker turns are used, which are not interrupted by silence or noise segments, to ensure that the pitch of the patient and the therapist is not altered by these segments. Thereby, speaker turn combinations where the patient speaks first and the therapist answers are labeled as patient leading and speaker turn combinations where the therapist speaks and the patient answers are labeled as therapist leading; script available at Schoenherr (2020) and Schoenherr (2022).

Example Dataset

As example file we used the file “6829-68769.1580-141083” from the EMRAI synthetic diarization corpus which is publicly available at Github (Edwards et al., 2018). The example audio includes a 2-person interaction. With this example, we applied all steps to calculate vocal synchrony. Note that our main aim with the example was to include all possible annotation codes and an example of modification of pitch settings. Therefore, the analyses are provided as a simple demonstrative illustration of the application of the method.

Conclusion

This tutorial showed how to extract vocal synchrony indices from a video or audio file. The tutorial increases the objectivity and transparency of the procedure and enables other working groups to replicate our results. It should be emphasized that the annotation and transcription were performed in a standardized way by coding rules and its reliability can be investigated by inter-rater agreement of the segmentation. In addition, the extraction of pitch parameters, the speech rate and the calculation of vocal synchrony was performed automatically by scripts.

Previous scripts are based on small parts of speech and analyze formants or set boundaries based on spoken consonants (Park & Seong, 2018; Winn, 2020). These types of scripts are very helpful for linguists, but are unsuitable for conversations especially in the psychotherapeutic context. If one is only interested in specific words within a psychotherapy context and wants to investigate, for example, whether psychological disorders can already be recognized by the sound of certain phonemes, the scripts we propose are unsuitable, since they aggregate the complete language within different speaker changes and do not make it possible to look at individual syllables, phonemes or formants.

However, if aggregated acoustic features should be extracted for speaker parts, our script offers the possibility to do this based on freeware software Praat and R which makes it accessible to a wide range of researchers.

Funding: The current study was funded by the German Research Foundation (funding number GZ: STR 306/28-3).

Acknowledgments: The authors have no additional (i.e., non-financial) support to report.

Competing Interests: The authors have declared that no competing interests exist.

Author Contributions: DS wrote the Praat and R scripts. DS and AS wrote the manuscript to the largest part. FR wrote and edited some parts of the manuscript. UA, BS and LK edited and corrected the manuscript. All authors read and approved the final manuscript.

Ethics Statement: The local Ethics Committee approved this study (4189-08/14).

Data Availability: The Praaat and R scripts to run vocal synchrony analysis; the datasets generated and/or analysed during the current study; and tables containing dataset building steps overview, coding & transcription rules, and possible error messages are freely available and can be found in the Supplementary Materials.

Supplementary Materials

The supplementary materials provided are the Praaat and R scripts to run vocal synchrony analysis; the datasets generated and/or analysed; and tables containing dataset building steps overview, coding & transcription rules, and possible error messages (see Schoenherr, 2020, Schoenherr, 2022, and Schoenherr et al., 2023).

Index of Supplementary Materials

Schoenherr, D. (2020). *vocal_synchrony* [Praat and R scripts, datasets]. OSF. https://osf.io/dzvpt/

Schoenherr, D. (2022). *vocal_synchrony* [Praat and R scripts, datasets]. GitHub. https://github.com/SchoenherrD/vocal_synchrony

Schoenherr, D., Shugaley, A., Roller, F., Knitter, L. A., Strauss, B., & Altmann, U. (2023). *Supplementary materials to "Extracting vocal characteristics and calculating vocal synchrony using Praat and R: A tutorial"* [Tables containing: dataset building steps overview, coding & transcription rules, possible error messages]. PsychOpen GOLD. https://doi.org/10.23668/psycharchives.13253

References

Altmann, U., Schoenherr, D., Paulick, J., Deisenhofer, A.-K., Schwartz, B., Rubel, J., Stangier, U., Lutz, W., & Strauss, B. (2020). Associations between movement synchrony and outcome in patients with social anxiety disorder: Evidence for treatment specific effects. *Psychotherapy Research, 30*(5), 574–590. https://doi.org/10.1080/10503307.2019.1630779

Banse, R., & Scherer, K. R. (1996). Acoustic profiles in vocal emotion expression. *Journal of Personality and Social Psychology, 70*(3), 614–636. https://doi.org/10.1037/0022-3514.70.3.614

Biemans, M. (2000). *Gender variation in voice quality.* Netherlands Graduate School of Linguistics.

Boersma, P. (2001). Praat, a system for doing phonetics by computer. *Glot International, 5*(9), 341–345.

Boersma, P., & Weenink, D. (2007). *Praat: Doing phonetics by computer* (Version 6.0.26) [Computer program]. http://www.praat.org/

Edwards, E., Brenndoerfer, M., Robinson, A., Sadoughi, N., Finley, G. P., Korenevsky, M., Axtmann, N., & Suendermann-Oeft, D. (2018, September). *A free synthetic corpus for speaker diarization research.* International Conference on Speech and Computer. Springer, Cham, Switzerland.

Fürer, L., Schenk, N., Roth, V., Steppan, M., Schmeck, K., & Zimmermann, R. (2020). Supervised speaker diarization using random forests: A tool for psychotherapy process research. *Frontiers in Psychology, 11*, Article 1726. https://doi.org/10.3389/fpsyg.2020.01726

Galbusera, L., Finn, M. T., & Fuchs, T. (2018). Interactional synchrony and negative symptoms: An outcome study of body-oriented psychotherapy for schizophrenia. *Psychotherapy Research, 28*(3), 457–469. https://doi.org/10.1080/10503307.2016.1216624

Gaume, J., Hallgren, K. A., Clair, C., Schmid Mast, M., Carrard, V., & Atkins, D. C. (2019). Modeling empathy as synchrony in clinician and patient vocally encoded emotional arousal: A failure to replicate. *Journal of Counseling Psychology, 66*(3), 341–350. https://doi.org/10.1037/cou0000322

Holle, H., & Rein, R. (2015). EasyDIAg: A tool for easy determination of interrater agreement. *Behavior Research Methods, 47*(3), 837–847. https://doi.org/10.3758/s13428-014-0506-7

Imel, Z. E., Barco, J. S., Brown, H. J., Baucom, B. R., Baer, J. S., Kircher, J. C., & Atkins, D. C. (2014). The association of therapist empathy and synchrony in vocally encoded arousal. *Journal of Counseling Psychology, 61*(1), 146–153. https://doi.org/10.1037/a0034943

Kleinbub, J. R., & Ramseyer, F. (2018). *rMEA synchrony in Motion Energy Analysis (MEA) time-series* [R package version 1.0.0.9012]. R Foundation.

Luehof, S. (2019). *Automatic analysis of synchrony in dyadic interviews* [Master's Thesis, Utrecht University]. Utrecht University Student Theses Repository. https://studenttheses.uu.nl/handle/20.500.12932/33670

Lutz, W., Prinz, J. N., Schwartz, B., Paulick, J., Schoenherr, D., Deisenhofer, A.-K., Terhürne, P., Boyle, K., Altmann, U., & Strauß, B. (2020). Patterns of early change in interpersonal problems and their relationship to nonverbal synchrony and multidimensional outcome. *Journal of Counseling Psychology, 67*(4), 449–461. https://doi.org/10.1037/cou0000376

Munafò, M. R., Nosek, B. A., Bishop, D. V., Buon, K. S., Chambers, C. D., Sert, N. P. d., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., & Ioannidis, J. P. (2017). A manifesto for reproducible science. *Nature Human Behaviour, 1*, Article 0021. https://doi.org/10.1038/s41562-016-0021

Park, J., & Seong, C. (2018). The implementation of children's automated formant setting by Praat scripting. *Phonetics and Speech Sciences, 10*(4), 1–10. https://doi.org/10.13064/KSSS.2018.10.4.001

Paulick, J., Deisenhofer, A.-K., Ramseyer, F., Tschacher, W., Boyle, K., Rubel, J., & Lutz, W. (2018). Nonverbal synchrony: A new approach to better understand psychotherapeutic processes and drop-out. *Journal of Psychotherapy Integration, 28*(3), 367–384. https://doi.org/10.1037/int0000099

Paxton, A. (2015). *Coordination: Theoretical, methodological, and experimental perspectives* [Doctoral thesis, University of California, Merced]. eScholarship. https://escholarship.org/uc/item/5tx5s7zh

Pérez-Rosas, V., Mihalcea, R., Resnicow, K., Singh, S., & An, L. (2017, July 30–August 4). *Understanding and predicting empathic behavior in counseling therapy.* Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada.

Ramseyer, F., & Tschacher, W. (2011). Nonverbal synchrony in psychotherapy: Coordinated body movement reflects relationship quality and outcome. *Journal of Consulting and Clinical Psychology, 79*(3), 284–295. https://doi.org/10.1037/a0023419

Reich, C. M., Berman, J. S., Dale, R., & Levitt, H. M. (2014). Vocal synchrony in psychotherapy. *Journal of Social and Clinical Psychology, 33*(5), 481–494. https://doi.org/10.1521/jscp.2014.33.5.481

Rocco, D., Gennaro, A., Salvatore, S., Stoycheva, V., & Bucci, W. (2017). Clinical mutual attunement and the development of therapeutic process: A preliminary study. *Journal of Constructivist Psychology, 30*(4), 371–387. https://doi.org/10.1080/10720537.2016.1227950

Schoenherr, D., Paulick, J., Strauss, B., Deisenhofer, A.-K., Schwartz, B., Rubel, J., Lutz, W., Stangier, U., & Altmann, U. (2019a). Identification of movement synchrony: Validation of windowed cross-lagged correlation and –regression with peak-picking algorithm. *PLoS One, 14*(2), Article e0211494. https://doi.org/10.1371/journal.pone.0211494

Schoenherr, D., Paulick, J., Strauss, B., Deisenhofer, A.-K., Schwartz, B., Rubel, J., Lutz, W., Stangier, U., & Altmann, U. (2019b). Nonverbal synchrony predicts premature termination of psychotherapy for social phobic patients. *Psychotherapy (Chicago, Ill.), 56*(4), 503–513. https://doi.org/10.1037/pst0000216

Schoenherr, D., Paulick, J., Worrack, S., Strauss, B., Rubel, J., Schwartz, B., Deisenhofer, A.-K., Lutz, W., Stangier, U., & Altmann, U. (2019). Quantification of nonverbal synchrony using linear time series analysis methods: Lack of convergent validity and evidence for facets of synchrony. *Behavior Research Methods, 51*(1), 361–383. https://doi.org/10.3758/s13428-018-1139-z

Schoenherr, D., Strauss, B., Paulick, J., Deisenhofer, A.-K., Schwartz, B., Rubel, J., Boyle, K., Lutz, W., Stangier, U., & Altmann, U. (2021). Movement synchrony and attachment related anxiety and avoidance in social anxiety disorder. *Journal of Psychotherapy Integration, 31*(2), 163–179. https://doi.org/10.1037/int0000187

Spivack, N. (1996). *Measuring mutual influence in the analytic discourse* [Doctoral thesis, Adelphi University]. ProQuest Dissertations Publishing.

Tomicic, A., Martinez, C., & Krause, M. (2015). The sound of change: A study of the psychotherapeutic process embodied in vocal expression. Laura Rice's ideas revisited. *Psychotherapy Research, 25*(2), 263–276. https://doi.org/10.1080/10503307.2014.892647

Tonti, M., & Gelo, O. C. G. (2016). Rate of speech and emotional-cognitive regulation in the psychotherapeutic process: A pilot study. *Research in Psychotherapy, 19*(2), 102–112. https://doi.org/10.4081/ripppo.2016.232

Watts, R. J. (1989). Taking the pitcher to the 'well': Native speakers' perception of their use of discourse markers in conversation. *Journal of Pragmatics, 13*(2), 203–237. https://doi.org/10.1016/0378-2166(89)90092-1

Wieder, G., & Wiltshire, T. J. (2020). Investigating coregulation of emotional arousal during exposure-based CBT using vocal encoding and actor–partner interdependence models. *Journal of Counseling Psychology, 67*(3), 337–348. https://doi.org/10.1037/cou0000405

Winn, M. B. (2020). Manipulation of voice onset time in speech stimuli: A tutorial and flexible Praat script. *The Journal of the Acoustical Society of America, 147*(2), 852–866. https://doi.org/10.1121/10.0000692

Methodology is the official journal of the European Association of Methodology (EAM).

leibniz-psychology.org

PsychOpen GOLD is a publishing service by Leibniz Institute for Psychology (ZPID), Germany.