`praatpicture`: A library for making flexible Praat Picture-style figures in R

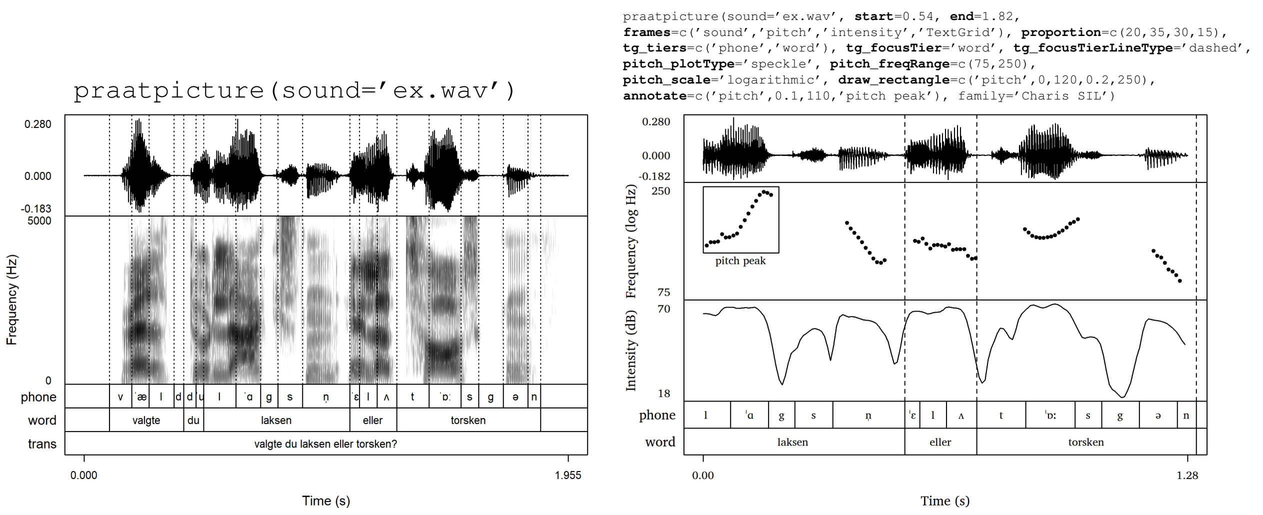
Rasmus Puggaard-Rode

Institute for Phonetics and Speech Processing, LMU Munich

`r.puggaard@phonetik.uni-muenchen.de`

Introduction. The plotting utility available through Praat (Boersma and Weenink 2023), usually accessed through the Praat Picture window of the graphical user interface (GUI), is ubiquitous in phonetics. Praat Picture is a flexible tool which can produce a wide variety of figures, although its most common application is probably plotting one or more acoustic signals which are time-aligned with annotations written in the `.TextGrid` file format; indeed, Praat Picture is undoubtedly the most widely used method for producing this very common style of figure. Praat Picture can either be used with the GUI, which limits the tool's flexibility quite a bit, or with scripts written in Praat's specialized custom scripting language.
The software environment R (R Core Team 2023), which is much more general-purpose than Praat, is used by many phoneticians for a big portion of their processing and analysis pipeline, and increasingly also for preparing manuscripts and presentations using the RMarkdown and Quarto formats (Xie 2015). This presents a need for a similarly flexible plotting utility that can visualize acoustic signals with time-aligned annotations in R, allowing phoneticians to keep as much as possible of their workflow in one software environment.
This paper introduces an R library, `praatpicture`, which aims to fill this gap. The purpose of `praatpicture` is to produce figures of acoustic signals with time-aligned annotations that by default resemble their counterparts in Praat as much as possible, and to allow for at least the same degree of flexibility as plotting in Praat, while relying on signal processing tools that are already available in R. `praatpicture` relies on base R graphics tools, which presents some advantages over Praat, including the ability to resize figures dynamically (i.e. without regenerating figures with new size parameters), and the ability to use any font available to the system. Version 0.6.0 of `praatpicture` is currently available from GitHub (`https://github.com/rpuggaardrode/praatpicture`).

Usage and options. The core function of the library is `praatpicture()`, which only takes one obligatory argument, `sound`, giving the name of a sound file with the `.wav` extension. Calling `praatpicture()` with just one argument will produce a very common figure format: a waveform, a spectrogram, and annotations with dotted lines in the various figure components indicating the locations of annotation boundaries (see the left panel of **Figure 1**). (This assumes that there is a file with the `.TextGrid` extension and the same base name as the `.wav` file in the same directory, but the `make_TextGrid()` function also allows users to create time-aligned annotations interactively in R.) In the following, I give a brief and incomplete overview of the options available to users of the package, some of which are visualized with accompanying code in the right panel of **Figure 1**; argument names which cooccur in the text and in **Figure 1** are given in parentheses in the text and are bolded in the figure.
The user can control the size of individual plot components (`proportion`) and which portion of a sound file to plot (`start`, `end`). Using Praat's terminology, the user can control which annotation tiers to plot (`tg_tiers`) which boundaries to show throughout all figure components (`tg_focusTier`), and the appearance of these boundaries (`tg_focusTierLineType`). In addition to waveforms, spectrograms, and annotations, `praatpicture` can also plot pitch tracks, formant tracks, and intensity tracks (`frames`). The user can control how derived signals are generated – i.e., which window shape and size to use, dynamic range, pitch floor and ceiling, how many formants to calculate, etc. – and how they are visualized – i.e., which frequency range to show (e.g. `pitch_freqRange`), whether pitch and formants should be 'speckled' or 'drawn' (e.g. `pitch_plotType`), which frequency scale to use (e.g. `pitch_scale`), which colors should be used for plotting individual plot components, etc. – with largely the same arguments as those available in Praat. Several options are available for highlighting parts of a figure (e.g. `draw_rectangle`, `annotate`).
A sister function to `praatpicture()`, called `emupicture()`, is available for users of the EMU Speech Database Management System (Winkelmann, Harrington, and Jänsch 2017) who wish to plot annotated signal data directly from an EMU database. No such plotting utility has previously been available. The library also offers the function



```
praatpicture(sound='ex.wav')
```

```
praatpicture(sound='ex.wav', start=0.54, end=1.82,
frames=c('sound','pitch','intensity','TextGrid'), proportion=c(20,35,30,15),
tg_tiers=c('phone','word'), tg_focusTier='word', tg_focusTierLineType='dashed',
pitch_plotType='speckle', pitch_freqRange=c(75,250),
pitch_scale='logarithmic', draw_rectangle=c('pitch',0,120,0.2,250),
annotate=c('pitch',0.1,110,'pitch peak'), family='Charis SIL')
```

Figure 1: *Two examples of figures generated with* `praatpicture` *and the code used to generate them.*

`talking_praatpicture()` for creating video files of figures with embedded audio, and `praatanimation()` for easily creating `praatpicture()`-based animations.

Implementation. Spectrograms are generated in R using the `phonTools` package (Barreda 2023). Other derived signals are generated using the `wrassp` package in R (Winkelmann, Bombien, et al. 2023). Pitch is calculated using the `ksvF0()` function, formants are calculated using the `forest()` function, and root-mean-squared intensity is calculated using the `rmsana()` function. In all cases, default parameters are set to emulate those in Praat as much as possible, such as e.g. using Gaussian-like window shapes across the board. When results still differ, it is because the underlying algorithms are not identical.
`.TextGrid` files are read into R using the `rPraat` package (Bořil and Skarnitzl 2016), optionally converting to Praat's special character formatting using a custom script. It is also possible to plot pitch, formant, and intensity with `praatpicture()` using values calculated in Praat, if the signals are saved from Praat using the same base file name as the `.wav` file in the same directory; these are then also read into R using `rPraat`. Alternatively, any other software can be used to calculate these signals, as long as they are stored in the Simple Signal File Format (SSFF).

Conclusion. `praatpicture` provides an opportunity for phoneticians who use R (and potentially EMU-SDMS) to keep more of their workflow in R, by allowing users to make familiar-looking figures in a general-purpose software environment without necessarily relying on the plotting and signal processing tools in Praat. Using base R graphics tools to produce these figures arguably has a number of advantages in terms of flexilibity. `praatpicture` currently has most of the same options as Praat does in terms of producing figures with time-aligned acoustic signals and annotations. The library is still in development, so existing features will be augmented and more features will be added over time.

References.

Barreda, Santiago (2023). "`phonTools`. Tools for phonetic and acoustic analyses". (Version 0.2–2.2). URL: `https://CRAN.R-project.org/package=phonTools`.

Boersma, Paul and David Weenink (2023). "Praat. Doing phonetics by computer". (Version 6.4.01). URL: `https://fon.hum.uva.nl/praat/`.

Bořil, Tomáš and Radek Skarnitzl (2016). "Tools `rPraat` and `mPraat`. Interfacing phonetic analyses with signal processing". In: *Text, speech, and dialogue*. Ed. by Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala. Cham: Springer, pp. 367–374. DOI: `10.1007/978-3-319-45510-5_42`.

R Core Team (2023). "R. A language and environment for statistical computing". (Version 4.3.2). URL: `https://R-project.org`.

Winkelmann, Raphael, Lasse Bombien, Michel Scheffers, and Markus Jochim (2023). "`wrassp`. Interface to the ASSP library". (Version 1.0.4). URL: `https://CRAN.R-project.org/package=wrassp`.

Winkelmann, Raphael, Jonathan Harrington, and Klaus Jänsch (2017). "EMU-SDMS. Advanced speech database management and analysis in R". In: *Computer Speech & Language* 45, pp. 392–410. DOI: `10.1016/j.csl.2017.01.002`.

Xie, Yihui (2015). *Dynamic documents with R and* `knitr`. Boca Raton: CRC Press.