***mausmooth*: Eyeballing made easy**

Francesco Cangemi, Aviad Albert
IfL Phonetik – University of Cologne

Before proceeding to targeted quantitative analyses, researchers working on tone and intonation often need to eyeball pitch contours for several audio files from a large dataset, in order to grasp visible qualitative differences across conditions (or items, or speakers). This stage of preliminary data visualization is often carried out by superposing raw f0 contours in a plot window, using colour-coding for the various conditions. This is often achieved automatically, e.g. using scripted procedures in *Praat* [1]. The readability of such displays is however hindered by the presence of inevitable errors in automatic pitch-tracking procedures. Such errors are especially disruptive when contours are displayed using a semi-continuous representation of f0 (e.g. the usual *curve* drawing method in *Praat*), due to meaningless interpolated stretches. However, a point-by-point representation of f0 (e.g. the *speckles* drawing method in *Praat*) is also ill suited for plotting superposed contours, since it is difficult to ascribe speckles to individual items. Figure 1 provides an example of the reduced readability of superposed contours using raw f0 points.

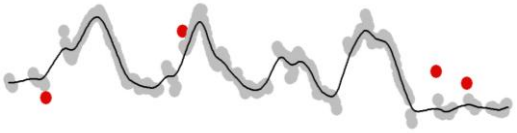
In order to overcome this problem, we introduce *mausmooth* [2], a *Praat* plug-in that provides readable plots of error-free superposed pitch contours (cf. Figure 2). *mausmooth* can be downloaded on the first author's website; it is extensively documented and particularly easy to use, and is thus mainly directed at researchers who do not wish to script their own plotting routines. In a nutshell, for each audio file to be processed, *mausmooth* (i) extracts f0 candidates, (ii) pauses to ask the user a manual correction of the selected f0 candidates, (iii) interpolates and smoothes the selected values. Figure 3 shows the outcome of these three steps: extracted values (grey speckles), points deleted in the manual correction phase (red speckles), output smoothed contour (black curve). Parameters for candidate extraction, selection and smoothing can be customised by the user. Based on the research question, for example, it might be preferable either to retain some microprosodic detail (and thus only perform minimal smoothing) or to focus on major intonational events (and thus filter out smaller perturbations). Figure 4 shows f0 points (grey speckles) for the same utterance, processed using lower (black thick curve) or higher (red thin curve) smoothing values.

The main purpose of the tool is to easily provide readable and error-free plots of f0 contours for several audio files. As such, it complements other f0 stylisation algorithms developed for different purposes, such as *Momel* [3] and *Prosogram* [4]. Figure 5 provides a comparison of the output of the three tools in the stylisation of a test utterance from *SPPAS* [5], with extracted values (black empty circles), *Momel* stylisation (red thin polyline), *Prosogram* stylisation (blue segments), and *Mausmooth* output (grey thick curve). Note *mausmooth*'s resilience to microprosody (cp. *Momel*'s output) and the readability of its smoothed output, especially for superposing multiple utterances (cp. *Prosogram*'s segments).

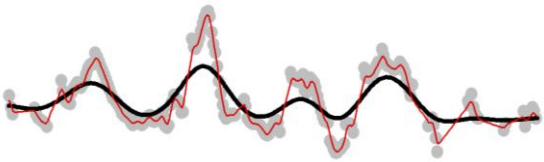
The functionality of *mausmooth* can be expanded to (i) accommodate input from other tools or (ii) provide input for further analyses. As for (i), the manual contribution of the user can be further reduced by filtering out pitch candidates with low reliability. This can naturally be achieved by tweaking the parameters of the *Praat* path-finding algorithm, but also by (i') using intensity and Harmonics-to-Noise Ratio to compute regions of low vs. high pitch intelligibility mass or by (i'') reducing the impact of pitch points near syllable or phone boundaries. Such boundaries can be provided as input using either a manual segmentation or an automatic extraction, for example using syllabification algorithms (e.g. [6]) or forced alignment (e.g. [7]). As for (ii), the smoothed contours can be used as reliable input for further analyses. For example, automatic detection of low turning points (e.g. [8]) is considerably more robust when using smoothed contours rather than raw f0 points.

F1. f0 contours for utterances from two categories (cyan vs. magenta) plotted using *Praat* standard extraction/plotting method.

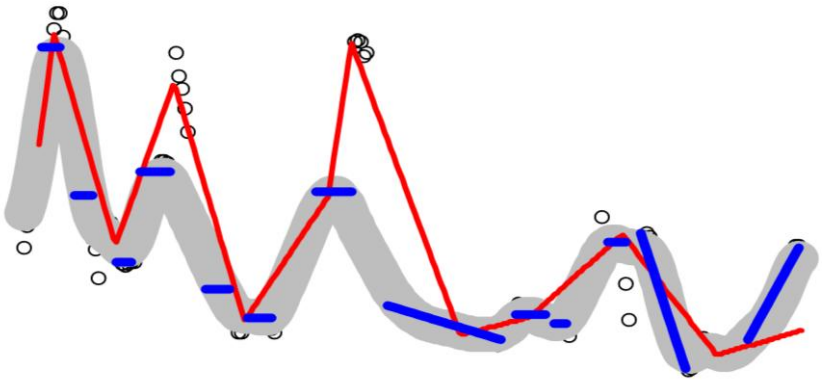
F2. f0 contours from Figure 1 plotted using *Mausmooth*.



F3. A single utterance plotted by *Mausmooth* (black curve); extracted f0 points in grey; manually discarded f0 points in red.



F4. A single utterance plotted using different smoothing values (black thick: lower, red thin: higher); extracted f0 points in grey.



F5. Extracted f0 points (black empty circles) and stylisations from *Momel* (red thin polyline), *Prosogram* (blue segments) and *Mausmooth* (grey thick curve) for an utterance.

References

[1] Boersma, Paul & David Weenink (2016). Praat: doing phonetics by computer [Computer program]. Version 6.0.14, retrieved 11 February 2016 from http://www.praat.org/

[2] Cangemi, Francesco (2015). mausmooth [Computer program]. Version 1.0, retrieved 15 May2016 from http://phonetik.phil-fak.uni-koeln.de/fcangemi.html

[3] Hirst, Daniel & Robert Espesser (1993). Automatic modelling of fundamental frequency using a quadratic spline function. *Travaux de l'Institut de Phonétique d'Aix* 15, 71-85.

[4] Mertens, Piet (2004). The Prosogram: semi-automatic transcription of prosody based on a tonal perception model. in Bernard Bel & Isabelle Marlien (eds.) *Proceedings of Speech Prosody 2004*, Nara (Japan), 23-26 March.

[5] Bigi, Brigitte (2015). SPPAS - Multi-lingual approaches to the automatic annotation of speech. *The Phonetician* 111-112, 55-69.

[6] Origlia, Antonio, Giovanni Abete & Franco Cutugno (2013). A dynamic tonal perception model for optimal pitch stylization. *Computer Speech and Language* 27, 190–208.

[7] Lubbers, Mart, & Francisco Torreira (2016). Praatalign: An interactive Praat plug-in for performing phonetic forced alignment [Computer program]. Version 1.9b, retrieved 25 February 2016 from https://github.com/dopefishh/praatalign/

[8] Cangemi, Francesco (2014). *Prosodic detail in Neapolitan Italian*. Berlin: Language Science Press.