PitchMendR: A Tool for the Diagnosis and Treatment of F0 Irregularities

Thomas Sostarics[1], *Jennifer Cole*[1]

[1]Northwestern University, Department of Linguistics

tsostarics@u.northwestern.edu, jennifer.cole1@northwestern.edu

Abstract

Irregularities in F0 tracking such as sudden jumps or the halving/doubling of F0 often arise from consonantal perturbations, voice quality modulations, or environmental noise. These irregularities are typically visually apparent to the researcher, but fixing such errors is a time-intensive process even with algorithms that provide heuristic assessments of potential errors. In this paper we describe PitchMendR: an R-based interactive visualization tool to rapidly identify and fix irregularities. We discuss the main features of the tool and a proof-of-concept analysis of how it can be used to reduce noise in a dataset for statistical modeling.

Index Terms: pitch, intonation, pitch tracking, f0

1. Introduction

Reliable F0 tracking is crucial for speech research, especially for the phonetics and phonology of intonation. Irregularities in F0 measurement, whether from the F0 extraction algorithm or from disturbances in a recording itself, add noise to time-series analyses and can make identification of a meaningful signal from a limited set of data more difficult. While advances have been made in creating effective algorithms for extracting F0 samples [1,2] and for flagging potential irregularities [3], there is still room for improvement on what to do with the remaining irregularities. In this paper, we describe an R-based graphical user interface (GUI) for rapidly visualizing and "mending" pitch contours containing irregularities in F0.

We focus our attention on two types of F0 irregularities: octave-jump errors (also known as pitch halving/doubling) and perturbation errors, with examples of each shown in Fig. 1.

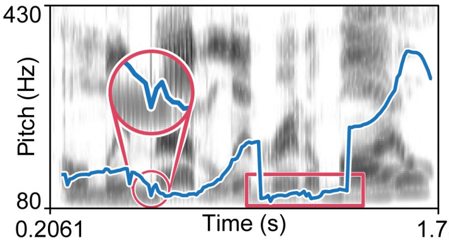


Figure 1: *Spectrogram and pitch contour of "Only Madelyn ran a mile?" Circled: perturbation from a transient environmental noise; Boxed: halving error.*

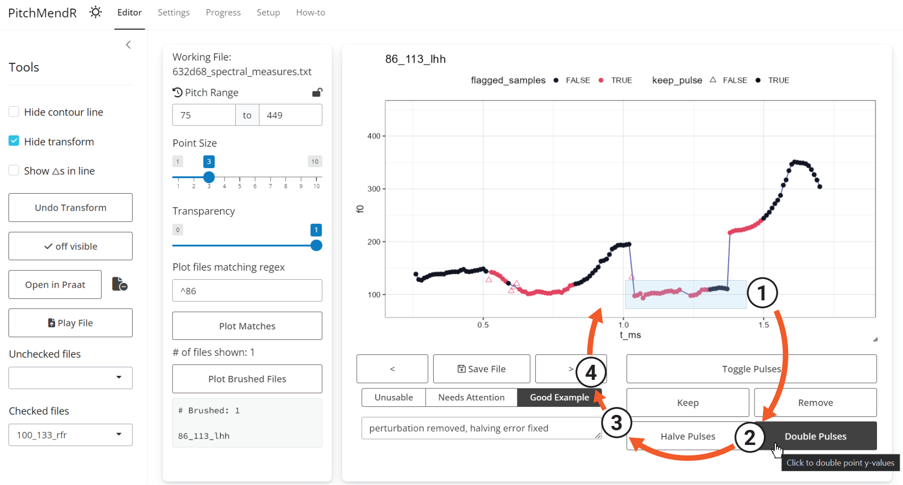
Octave-jump errors occur when the measured F0 samples are a harmonic or subharmonic of the F0, e.g., produced 110Hz may be measured as 220Hz or 55Hz. While some instances of octave-jump errors are faithful to the acoustic signal, such as genuine instances of waveform period doubling from non-modal phonation [4-6]. While these errors can be mitigated by adjusting the parameters used in the F0 extraction algorithm [7], they can nonetheless remain in large datasets, making it difficult to identify meaningful changes in F0 for intonation. Perturbation errors are discontinuities or distortions in the F0 contour due to segmental transitions or transient events in the acoustic signal. For example, the release from an oral stop can cause slight distortions in the pitch contour (see also [8] for nasal-vowel perturbations). Non-linguistic transient disturbances such as environmental noises can also perturb the F0 contour. These kinds of perturbations are particularly common in production studies where the recording environment cannot be controlled, such as in online studies where people can participate from home.

While materials for a production study can often be phonologically controlled to be amenable to F0 extraction, such as by prioritizing sonorant segments, this is not always an option. In work on the role of intonation on adjectival scales [9], some adjectives are amenable to F0 tracking (such as *brilliant*) while others present a challenge (as in *ecstatic*). Similarly, while some perturbation errors might be fixed with smoothing methods such as running-median smoothing, it appears inevitable that some F0 irregularities will remain. When it comes to working with the errors, the slowest but most thorough option involves annotating the extracted F0 time-series and selectively removing problematic samples or files. A faster option is to use an automated method such as [3] to flag potential errors, then use a heuristic to decide whether a file is likely unusable for an analysis based on the number of potential errors (e.g., remove files with >2 errors). While convenient, this can reduce the amount of usable data for an analysis and may require more data to be collected to supplement the lost data. In settings where data is difficult to acquire (e.g., in a fieldwork setting), operationalizing an exclusion criterion this way may make already-scarce data even harder to come by.

A compromise between the two approaches would be to manually inspect only those files that have been identified as potentially unusable, allowing the researcher to make the final determination on how to handle such cases. Even so, the number of problematic files can be quite large depending on the scale of the study; for a production study of 40 participants with 144 trials each, even 10% of files flagged as potentially unusable would still yield 576 files to inspect. Yet, errors are nonetheless visually apparent and can be readily confirmed with additional comparison to the audio signal.

2. PitchMendR

We introduce an application to fill the gap between the identification of F0 measurement errors and annotating, removing, or fixing such errors; a screenshot is shown in Fig. 2. PitchMendR is an open-source Shiny app written in R [10,11] that can be used to quickly plot time-series values. The plotted contours are interactive and individual measurement points or groups of points can be selected and marked for removal. Where there are clear cases of pitch halving or doubling, these values can be multiplied/divided by two to correct for the error.



□ Hide contour line
☑ Hide transform
□ Show △s in line

Unchecked files:

Checked files: 100_133_rfr

Working File: 632d68_spectral_measures.txt

Pitch Range: 75 to 449

Point Size: 3 (1–10)

Transparency: 1 (0–1)

Plot files matching regex: ^86

# of files shown: 1

```
# Brushed: 1

86_113_lhh
```

perturbation removed, halving error fixed

Figure 2: *Screenshot of interface using the example from Fig. 1. Regions in red are flagged as potential errors, and points marked for removal do not contribute to the sample-to-sample contour. Points can be selected directly from the plot (1) to be marked for removal, halved, or doubled (2). Tools are provided to make annotations (3) before moving to the next file (4).*

While developed with F0 values in mind, any time-series data can be used (e.g., formant trajectories, voice quality measures like H1*-H2*, or intensity). The app includes a reworked implementation of the algorithm in [3] which runs an estimated 37 times faster, allowing for datasets to be quickly flagged for potential errors. Points can then be colored by whether they're flagged or not, or alternatively, colored by different parts of the dataset (e.g., intensity, label from a TextGrid interval, or experimental condition). The GUI also provides a notepad to record observations and a series of buttons to quickly tag files with predetermined values (e.g., Unusable or Good Example). The app is available through an R package available at https://github.com/tsostarics/PitchmendR which can be run locally on a user's machine (henceforth the local version) or through a web-hosted version with more limited functionality (henceforth the web version), which is available with a demo file containing 36 contours at https://tsostarics.shinyapps.io/PitchMendR/.

Importantly, the output of the app is **non-destructive**: it does not change the original F0 values, meaning no data is lost or modified in the annotation process. Rather, a collection of new columns that describe the transformations needed to create the "mended" contours from the original F0 values are added to the table of data. A numeric transformation column is used to record multiplicative factors for correcting doubling/halving errors (where a factor of 1 indicates no correction needed) and a separate binary column is used to record annotations of whether a point should be kept or removed. The mended contours are thus equal to the original F0 values times the multiplicative factor and then filtered to retain only those points not explicitly marked for removal.

This non-destructive annotation process helps guard against academic dishonesty: it would be suspicious for a large dataset to have no irregularities without mentioning any annotation process. Researchers using PitchMendR can thus share not only the original F0 values, but also the columns containing the transformation values and their notes to reproduce the mended contours—all recorded in the same dataset. In the process, the criteria for how errors are handled can be made explicit in the same way criteria for other types of phonetic annotations are. In cases where there is researcher disagreement about how one class of error should be handled, visualizing the contours with and without these non-destructive transformations can serve as a basis for discussion and adjudication.

PitchMendR is intended as a streamlined **annotation tool** to supplement workflows with Praat and R. Additionally, the focus of this app is on working with large datasets of pitch contours that have ***already been extracted.*** That is, it is not a tool for resynthesis (c.f. PSOLA in Praat [12]) nor a tool for preprocessing pitch extraction/measurement. It is also not an analysis tool, meaning that students and researchers will still need to learn and use valuable skills in signal processing and data handling (i.e., smoothing should be thoughtfully done by the researcher, not quickly by a button). Students working as research assistants can be trained to use the app, though we do not recommend "naïve" annotation done without a firm understanding of phonetics related to the voice source and especially the effects of voice quality on octave-jump errors.

The local version of PitchMendR allows users to play audio files from within the app when inspecting an individual file, or even interface with Praat, allowing the user to open audio files and TextGrids for further inspection using a single "Open in Praat" button without needing to navigate additional menus. For security reasons, the web app does not have this functionality and is primarily restricted to uploading a spreadsheet, making annotations, then downloading the annotated spreadsheet containing new columns. The app also keeps track of which files have already been inspected, allowing users to keep track of their progress across annotation sessions.

2.1. Proof of Concept

We use recordings elicited from an imitation paradigm to show how PitchMendR can be used to reduce non-meaningful variability and avoid data loss. In this study, speakers listened to an auditory model sentence such as "Only Oliver rode away?" with different resynthesized pitch contours and were tasked with imitating the pitch contour they heard with a new sentence such as "Only Harmony ran a mile?" or "Only Madelyn made a move?". The experiment had 144 trials, which were split evenly between three broad tune classes (falls, rises, and rise-fall-rise), each of which had four distinct (but potentially not linguistically contrastive) trajectories. For this paper we focus only on the imitations of the four rising pitch trajectories (=48 trials per participant). We show data from four speakers (two identified as male, two female), with two having many files initially flagged as unusable (21 and 18 out of 48) and two with fewer flagged files (5 and 2).

Using PitchMendR to annotate and repair F0 contours is not the only way to reduce the prevalence of F0 irregularities in a dataset. Although different F0 sampling algorithms may be differentially robust to different types of errors, all algorithms will produce **some** errors. Fig. 3 shows extracted F0 samples for 192 rising contours using Praat's raw autocorrelation algorithm [12] and STRAIGHT's algorithm as implemented in VoiceSauce [2,13] using speaker-specific pitch ranges.

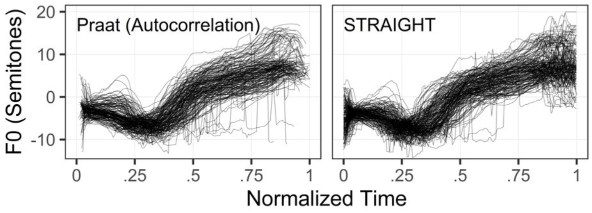


Figure 3: *F0 Contours from two different algorithms. Both algorithms show some amount of noise from abrupt jumps and evident halving errors.*

In the remainder of this section, we will show how variability in sample-to-sample F0 differences (henceforth: F0 jumps) can be reduced through the annotation process. We will begin with a visual comparison of (1) the original F0 contours compared to the mended contours, then (2) how variability in jumps as shown through the first derivative of F0 with respect to time is reduced. Finally, we will show how the results of statistical modeling using generalized additive models (GAMs) and clustering methods can change depending on the dataset.

Fig. 4 shows the original pitch contours (from Praat's autocorrelation output, extracted with PraatSauce [14]) and the mended pitch contours using PitchMendR. All contours in Fig. 4 are smoothed using five-point running median smoothing, though one can observe that some errors persist even after smoothing. Of the 192 rising contours, 46 were flagged as containing enough errors to exceed the exclusion criterion described in [3]; for this dataset, this would be a loss of 24% of the data—nearly one participant's worth of rising contour trials.

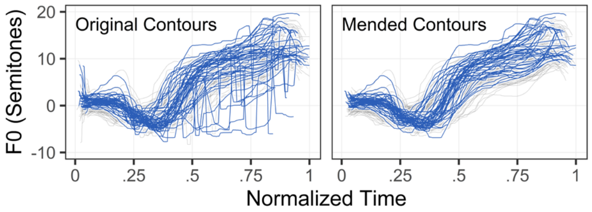


Figure 4: *Smoothed time-normalized original vs. mended pitch contours. F0 is shown as semitones from speakers' median pitch. Files initially flagged as unusable are shown in blue.*

In the context of the experiment, the pitch excursions of these rises are quite large: upwards of two octaves for some speakers (e.g., a rise from 110Hz to 440Hz). Given the scale of these excursions, we can hypothesize that speakers may shift into a higher register or adopt different laryngeal strategies to reach these high pitch targets, which will likely incur a change in voice quality (see [6, 15] regarding voice quality). In line with this prediction, we can observe from the left side of Fig. 4 that most of these errors are likely halving errors, which can be corrected by doubling the frequency values in PitchMendR. Most of the corrected perturbation errors were due to transient environmental noises (e.g., mouse clicks), oral stops (the /d/ in *Madelyn made*), or frication (the /h/ in *Only Harmony*). Glottalization of the utterance initial vowel also tended to cause F0 jumps. Using PitchMendR, all but one file was recoverable.

Whereas Fig. 4 suggests that the overall shapes of the rising contours are less noisy than before, we might also consider whether the variability in F0 jumps has also been reduced. Thus, we turn from F0 over time to the **change in** F0 over time—i.e., the first derivative of F0 with respect to time—which we normalize to the sampling period. For example, if a pair of adjacent samples have a change in F0 of 4 semitones, but they are separated by 40ms (4 times the sampling period of 10ms), then the normalized F0 jump is 1 semitone. At issue now is the variability of these F0 jumps in three datasets: the **original** dataset with no files removed, the dataset remaining **after removing files** flagged as unusable, and the **mended** dataset. Fig. 5 shows the change in F0 for each dataset.

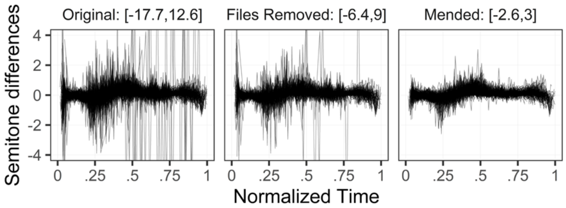


Figure 5: *F0 jumps as change in semitones over 10ms, labeled by dataset (range of F0 jumps in brackets). The y-axis is zoomed in to [-4, 4], meaning that more extreme F0 jumps go beyond the plot's bounds.*

Fig. 5 shows that while the majority of extreme F0 jumps in the original dataset can be eliminated by simply removing those files, there remain some extreme F0 jumps. The mended dataset not only includes the files that would have been removed, but also shows reduced variability in F0 jumps. The reduced variability seen in the time-series from Fig. 4 (F0 over time) and Fig. 5 (change in F0 over time) may should also affect statistical analyses using these datasets. For our analyses, recall that this experiment had four distinct rising trajectories that participants imitated. We model the imitated rising contours shown in Fig. 4 using GAMs [16,17] for each speaker and trajectory using separate models using each of the three datasets. We restrict our discussion to one speaker due to space limitations. Fig. 6 shows the model predicted contours for the speaker with the greatest number of files flagged as unusable.

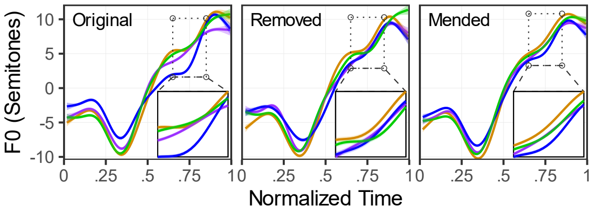


Figure 6: *GAM model predictions for the four rising trajectories from one speaker. The inlaid panel is zoomed in on the region from 65% to 85% of the normalized time.*

From Fig. 6 we can observe that generally the model is quite robust in identifying the overall shape of the rising contours. However, there are slight differences in the model predictions when comparing the across the four trajectories. Fig. 7 shows the difference between two trajectories for this speaker; regions where these trajectories do not significantly differ from one another would yield a difference at or near zero.

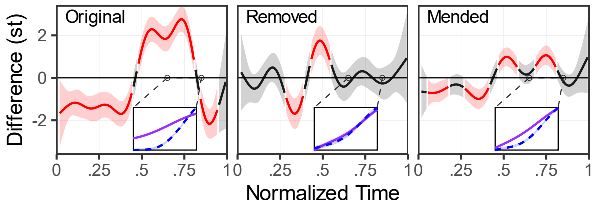


Figure 7: *Differences between two selected trajectories with 95% confidence intervals. The inlaid panel shows the two contours between 65% and 85% of the normalized time. The differences can be read as the solid purple line* ***minus*** *the dashed blue line.*

We can also investigate how the mending process affects the results of clustering analyses using F0 contours, such as those in [18,19]. One possibility is that the number of optimal clusters may vary depending on the amount of data available and the variability within the dataset. Fig. 8 shows the optimal number of clusters (assessed via the Calinski-Harabasz index criterion) for each of our illustrative datasets via K-means clustering for longitudinal data [20]. Here, each contour has been downsampled to use 30 equally spaced points for each contour. Based on the results in Fig. 8, we can observe that omitting files with an automated criterion did not change the number of clusters nor their apparent shape compared to the original dataset containing F0 irregularities. However, when these files are recovered in the mended dataset, we can see that the added number of observations for this sample of participants shows an additional cluster.

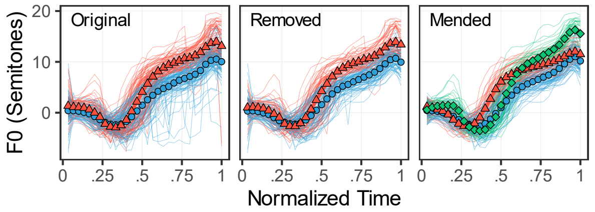


Figure 8: *Optimal clusters for each dataset, with the average contour within each cluster overlaid.*

3. Discussion

Mending pitch contours with PitchMendR provides two benefits. First, we were able to recover a large amount of data that would otherwise have been discarded from the analysis dataset. Second, we reduced the variability in the F0 jumps between adjacent samples. By doing so, we not only improve our statistical power by virtue of having more data at our disposal, but the mended dataset is also less noisy. These improvements are seen even when the contours to be modeled are smoothed, as seen in Fig. 4. Moreover, as seen in Fig. 7 and Fig. 8, depending on the decisions made when addressing F0 irregularities, the results of the statistical analysis can change in small or substantial ways.

While mending contours can help overall, it is nonetheless an annotation process that takes time to carry out rigorously. Thus, it is likely most fruitful to focus annotation efforts on files where automated methods have detected a high number/likelihood of F0 irregularities. It should nonetheless be noted that the goal of such efforts is not necessarily to coerce **all** recordings into a mended form: sometimes recordings do need to be thrown out for one reason or another.

The distinctions between some Intonational features can be subtle, and so while PitchMendR can help with removing distortions from environmental perturbations, care should also be taken not to remove **too many** points. A key example of this lies in regions of a contour containing high curvature, which require many points to adequately depict the curve through a series of line segments between discretized points. While tone-sequence models for intonational phonology, such as Autosegmental-Metrical theory [21], focus on the linear sequence of tones and their corresponding (relative) F0 targets, it has also been shown that intonational features can differ in the overall **shape** between points [22]. For example, the L+H* and L*+H pitch accents in English [23] and German [24] differ not only in the alignment of the accentual peak (early vs. late) but also in whether the shape is “domed” or “scooped.” Misuse of PitchMendR by removing too many points along these domed/scooped trajectories may limit one’s ability to reliably characterize their curved shapes. Again, we note that because the output of PitchMendR is non-destructive, cases where care has not been taken in the annotation process can be straightforwardly identified through comparison of the original and mended contours, which exist in the same dataset.

4. Conclusions

Despite a variety of F0 extraction algorithms and smoothing techniques to minimize the prevalence of F0 measurement irregularities, there is not currently a good way to quickly remove the irregularities that remain in a dataset. We have introduced an open-source tool called PitchMendR that capitalizes on how irregularities are typically visually salient. Using a responsive GUI, researchers can quickly visualize, identify, and repair pitch contours that contain octave-jump or perturbation errors. These “mended” contours are derived from a series of non-destructive transformations, which help guard against data loss as well as careless annotations.

We also compared the variability of F0 contours across three datasets from an imitation task: a dataset with no measures taken to treat irregularities, a dataset where many files were removed after being flagged with automated methods, and a dataset where PitchMendR was used to address the irregularities. We showed reduced variability (i.e., noise) in the data while also recovering a large amount of data in the mending process that would normally have been lost when using automated exclusion criteria. Finally, we showed that the results of statistical methods, such as GAM modeling and clustering analyses, can vary depending on how irregularities are handled within the dataset. We conclude that PitchMendR can be a valuable tool for researchers working with F0 data, helping to recover hard-earned data that would otherwise be discarded.

5. Acknowledgements

This research is supported by NSF BCS-1944773. Code for analyses and figures available at https://osf.io/687b2/.

6. References

[1] S. Strömbergsson.“Today”'s Most Frequently Used F0 Estimation Methods, and Their Accuracy in Estimating Male and Female Pitch in Clean Speech.” in *Proceedings INTERSPEECH 2016*, San Francisco, United States. 2016, pp. 525-529.

[2] H. Kawahara, A. d. Cheveigné, H. Banno, T. Takahashi, and T. Irino. “Nearly defect-free F0 trajectory extraction for expressive speech modifications based on STRAIGHT,” in *Proceedings INTERSPEECH 2006*, Lisbon, Portugal, Sep. 2006, pp. 537-540.

[3] J. Steffman and J. Cole.“An automated method for detecting F0 measurement jumps based on sample-to-sample differences.” *JASA Express Letters* vol. 2, no. 11, 2022.

[4] P. A. Keating, M. Garellek, and J. Kreiman, “Acoustic properties of different kinds of creaky voice.” in *Proceedings of the* $1^{8t}h$ *International Congress of Phonetic Sciences*, vol. 18, 2015, pp. 2–7.

[5] Y. Huang, “Articulatory properties of period-doubled voice in mandarin,” in *Proceedings of Speech Prosody 2022*, 2022, pp. 545–549.

[6] B. Roubeau, N. Henrich, and M. Castellengo, “Laryngeal vibratory mechanisms: the notion of vocal register revisited,” *Journal of voice*, vol. 23, no. 4, pp. 425–438, 2009.

[7] E. Keelan, C. Lai, and K. Zechner, “The importance of optimal parameter setting for pitch extraction.” *Proceedings of Meetings on Acoustics*, vol. 111, no. 1, 2011.

[8] Y. Xu, “Effects of tone and focus on the formation and alignment of f0 contours,” *Journal of phonetics*, vol. 27, no. 1, pp. 55–105, 1999.

[9] A. Göbel and E. Ronai, “On the meaning of intonational contours: a view from scalar inference,” in *Semantics and Linguistic Theory*, vol. 33, in press.

[10] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2022. [Online]. Available: https://www.R-project.org/.

[11] W. Chang, J. Cheng, J. Allaire, C. Sievert, B. Schloerke, Y. Xie, J. Allen, J. McPherson, A. Dipert, and B. Borges, *shiny: Web Application Framework for R*, 2023, R package version 1.8.0. [Online]. Available: https://CRAN.R-project.org/package=shiny

[12] P. Boersma and D. Weenink, *Praat: doing phonetics by computer [Computer program]*, 2022, version 6.2.14, retrieved 24 May 2022. [Online]. Accessed Available: http://www.praat.org/.

[13] Y.-L. Shue, P. Keating, C. Vicenik, and K. Yu, “Voicesauce: A program for voice analysis,” in *Proceedings of the 17th International Congress of Phonetic Sciences*, vol. 17, 2011, pp. 1846–1849.

[14] J. Kirby, *PraatSauce: Praat-based tools for spectral analysis [Computer program]*, 2018, version 0.2.6. [Online]. Available: github.com/kirbyj/praatsauce.

[15] J. H. Esling, S. R. Moisik, A. Benner, and L. Crevier-Buchman, *Voice quality: The laryngeal articulator model*. Cambridge University Press, 2019, vol. 162.

[16] S. Wood, *Generalized Additive Models: An Introduction with R*, 2nd ed. Chapman and Hall/CRC, 2017.

[17] G. L. Simpson, *gratia: Graceful ggplot-Based Graphics and Other Functions for GAMs Fitted using mgcv*, 2023, R package version 0.8.1. [Online]. Available: https://gavinsimpson.github.io/gratia/.

[18] J. Cole, J. Steffman, S. Shattuck-Hufnagel, and S. Tilsen, “Hierarchical distinctions in the production and perception of nuclear tunes in American English,” *Laboratory Phonology*, vol. 14, no. 1, 2023.

[19] C. Kaland, “Contour clustering: A field-data-driven approach for documenting and analysing prototypical f0 contours,” *Journal of the International Phonetic Association*, vol. 53, no. 1, pp. 159–188, 2023. doi:10.1017/S0025100321000049.

[20] C. Genolini, X. Alacoque, M. Sentenac, and C. Arnaud, “kml and kml3d: R packages to cluster longitudinal data,” *Journal of Statistical Software*, vol. 65, no. 4, pp. 1–34, 2015.

[21] J. B. Pierrehumbert, “The phonology and phonetics of English intonation,” Ph.D. dissertation, Massachusetts Institute of Technology, 1980.

[22] K. Iskarous, J. Steffman, and J. Cole, “American English pitch accent dynamics: A minimal model,” in *Proceedings of the 20th International Congress of Phonetic Sciences*, R. Skarnitzl and J. Volín, Eds. Guarant International, 2023, pp. 1469–1473.

[23] J. Barnes, A. Brugos, N. Veilleux, and S. Shattuck-Hufnagel, “On (and off) ramps in intonational phonology: Rises, falls, and the tonal center of gravity,” *Journal of Phonetics*, vol. 85, p. 101020, 2021.

[24] K. Zahner-Ritter, M. Einfeldt, D. Wochner, A. James, N. Dehé, and B. Braun, “Three kinds of rising-falling contours in German wh-questions: Evidence from form and function,” *Frontiers in Communication*, vol. 7, p. 58, 2022.