RMT: Rhythm and Melody Tools for Prosody Computation

Dafydd Gibbon

Bielefeld University, Germany

gibbon@uni-bielefeld.de

Abstract

Prosodic computational literacy is an important goal for students of acoustic phonetics, especially those from endangered language communities in less affluent countries. There are several excellent ‘off-the-shelf’ packages for prosody computation, including *Praat*, *ProsodyPro*, *Prosogram*, *ProZed*, *Winpitch*, and many convenient Praat scripts. However, experiments typically require small hybrid intersections of functionalities of these packages together with spreadsheets, R, Praat scripting or Python. Python was chosen in order to enable non-hybrid, seamless embedding of small tools into larger systems for exploratory research, because of scalability, and because of the availability of extensive Python libraries to support in-depth insight into filters and transformations rather than using ready-made complex functionalities. A design criterion for the toolkit is overall coherence and clarity of structure. The tools cover the analysis of speech signal annotations, and a modulation-theoretic approach to the demodulation of speech signal amplitude modulation and frequency modulation. Comparison of results is enabled by provision of distance measurement, hierarchical clustering techniques and SVM classification. The approach has been evaluated in practice in a range of publications and in teaching.

Index Terms: computational prosody, prosody course, rhythm tools, melody tools, pitch models, rhythm models, prosody vector clustering, Python

1. Introduction

1.1. Prosody analysis tools: objectives

The tools in the *Rhythm and Melody Toolkit* (RMT) are educational, not product-oriented, and provide support for computational literacy training in the analysis of speech rhythm and melody. The tools are not an ad hoc collection, but have a coherent theoretically motivated architecture in the form of Speech Modulation and Demodulation Theory (SMDT) for relating rhythm analysis and melody analysis (Figure 1), and thus the terminology is slightly different from the familiar textbook terms in Source-Filter or Exemplar or Analysis by Synthesis theories. SMDT models prosody production as frequency modulation (FM, i.e. the melodies of tones, pitch accents and intonation) and amplitude modulation (AM, i.e. sonority and rhythms of syllable, word and phrase components) of a carrier signal, and perception as demodulation of the FM and AM speech signal components (for an overview cf. [1]). The goal is not primarily engineering efficiency but phonetic understanding of speech with a theoretical foundation for physiological interpretation in speech production and perception.

Many excellent ‘off-the-shelf’ custom tools for prosody computation are currently available, mainly for F0 analysis, and with broad functionalities, including Praat [2], ProsodyPro [3], Prosogram [4], MoMel [5], ProZed [6], WinPitch [7], TextGrid Tools [8]. These tools are oriented towards the ‘phonetic consumer’: it is generally the functionality which interests the user rather than the mechanism. RMT is an open source educational practice kit rather than a mature and robust product: where there is a choice of algorithms, for example, the simplest is chosen, such as the absolute signal and smoothing for AM demodulation, and the Average Magnitude Difference Function for FM demodulation (see Section 3).

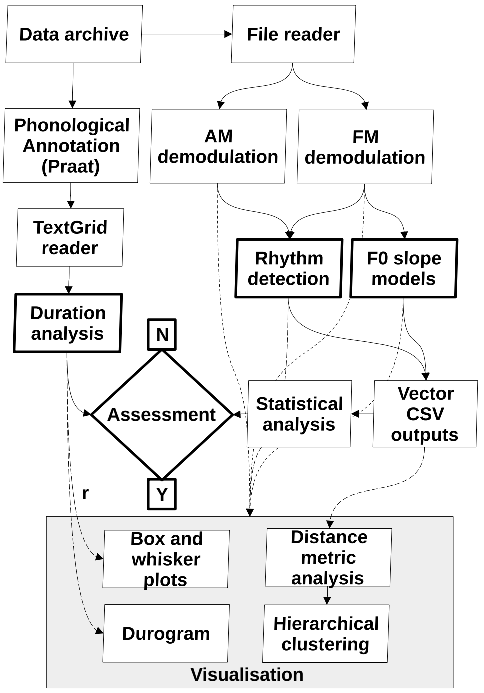

Figure 1: *Functional architecture of the toolkit. Dotted lines show paths to visual output, bold-lined boxes mark core tools.*

Scripts for Praat, MatLab and R are used in phonetics mainly for specific tasks (but cf. the general tool in [3]). Python is chosen for implementing RMT in order to facilitate use in larger computational linguistic and local language contexts within the SMDT framework. In addition to understanding the algorithmic mechanisms, RMT tool output is evaluated by comparison with annotation-based linguistic phonetic results as sources of independent evidence (cf. [1, 9]).

1.2. Specification

Modulation theory is 100 years old, as old as radio broadcasting, and the terminology (including the familiar terms ‘AM’ and ‘FM’) comes from radio engineering. One way of describing the AM and FM operations on unmodulated and modulated carrier signals is as follows:

Unmod. carrier: $C = Acos(2\pi ft + \phi)$

Mod. carrier: $C_{MOD} = A_{AM}Acos(2\pi(f + A_{FM})t + \phi)$ The FM and AM information signals are represented by A_{FM} and A_{AM}, respectively. For FM the amplitude of A_{FM} is added to the carrier and for AM the amplitude of the frequency-modulated carrier is multiplied by the scaled and raised amplitude of modulating signal A_{AM}. The task of FM and AM demodulation is to extract both A_{FM} and A_{AM} from C_{MOD}. The phase component ϕ is not considered here.

Modulation theoretic analysis was introduced to phonetics by Traunmüller [10] as a model of perception, and has been applied directly or indirectly since then in various contexts, including clinical applications; cf. for example [11, 12, 13, 14, 15, 16, 17, 1, 18]. Most of these studies have been concerned with the demodulation of AM and the use of forms of spectral analysis to analyse and compare different speech rhythms effected by AM, but the same procedure has been applied to FM in order to characterise rhythmic patterns of the slow modulations of fundamental frequency which characterise tones, pitch accents and intonation. Modulation of high frequency (HF) harmonics, such as vowel formants, is not dealt with. The low frequency (LF) switching AM of on-off voicing is also not dealt with explicitly, though voicing has to be dealt with at least implicitly for FM demodulation. The tools have been used extensively for teaching and research and their output has figured in a range of publications, which provide functional validation of the tools in operation [19, 20, 1, 21, 9, 22, 23].

The following areas are covered by the tools:

1. user-definable parameter settings in a configuration file;
2. speech signal input: mono WAV (previously normalised, e.g. with SOX, to mono 16kHz and amplitude range -1...0...1), input as an array with sampling frequency;
3. annotation analysis and visualisation: Praat TextGrid input with TextGrid annotation of interval durations, and descriptive statistics and visualisation;
4. bandpass filtering for AM demodulation (amplitude envelope extraction);
5. bandpass filtering for FM demodulation and smoothing (frequency envelope extraction, F0 estimation, 'pitch' extraction);
6. LF spectral analysis of rhythm frequencies (LF spectrum and LF spectrogram);
7. data comparison with distance networks, hierarchical cluster analysis and Support Vector Machines.

2. Annotation mining and visualisation

The annotation analysis tools generate statistical tables and visualisations from Praat TextGrid annotations, which have at least one tier and at least one labelled time-stamp on each tier, which consists of start and end time-stamps and a text label (variants also exist):

$\langle\langle TIER = \langle start = ts_1, end = ts_2, text = label\rangle^+\rangle^+\rangle$ Each annotation is a pair $\langle label, interval\rangle$, where the interval is a pair $\langle start, end\rangle$, and in a (complete) sequence of annotations the end of one annotation is identical to the $start$ of the next. An annotation tuple can be uniquely identified as a quintuple $\langle filename, tiername, label, start, end\rangle$.

The annotation can be created manually, or (semi)automatically [24], and an RMT tool extracts the annotation as a two-dimensional Python array (effectively a tier list of time-stamped label lists) and exports the array to a CSV file for further processing in Python (or R, other statistics packages, spreadsheets). Vectors of statistical values and their visualisations are generated, containing descriptive statistics, duration regularity indices and metrics such as the $rPVI$ and $nPVI$ [25]. The metrics measure one component of rhythm, (ir)regularity, under the counterfactual assumptions that rhythms are binary alternations and that other properties such as frequency are irrelevant. They are nevertheless useful, e.g. as heuristic benchmarks for evaluating results of other methods.

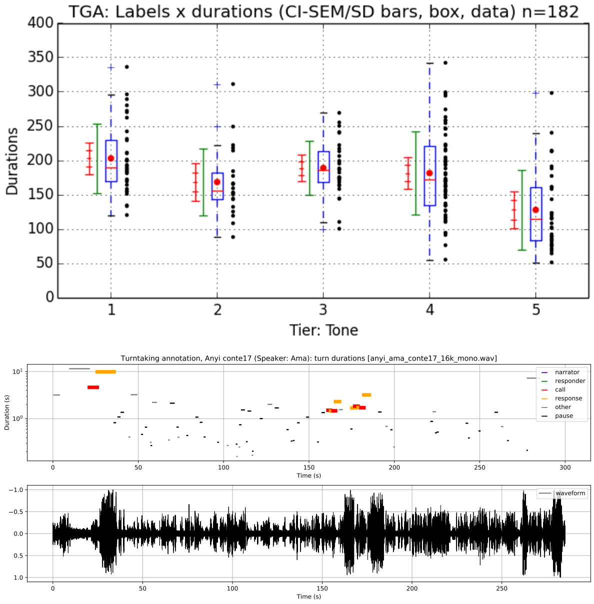

Figure 2: *Upper: Mandarin tone × duration in story reading, automatically generated from a Praat TextGrid file. Lower: Durogram of a long-term 5 min (300 s) interactive story-telling session in Ega (ISO 693-3 ega) automatically derived from a Praat TextGrid file. Colours represent participants, y-axis height and x-axis length represent turn durations.*

Tiers with limited vocabulary can be visualised for the purpose of initial 'eyeballing' and hypothesis development (cf. Figure 2, upper panel) using enhanced box-and-whisker plots. The plot shows tone properties from a reading of a Mandarin translation of *The North Wind and the Sun*, enabling a visual heuristic: if the boxes do not overlap, the elements are distinct. In this case the neutral tone (labelled as '5') is clearly dissimilar to Tone 1 and Tone 3 and is considerably different from Tone 2 and Tone 4.

More complex syntagmatic duration patterns, for example of turn distribution in dialogues, can be visualised as shown in the novel discourse patterning durogram in Figure 2, a timeline scatter plot which is not statistically derived but reflects parallel participant turn annotations directly. With $time \times duration$ axes, the label interval durations of different elements (in this case, turn-takers in a dialogue) are aligned along the time axis in different colours. Turn durations are shown by the height and length of the interval mark. The durogram provides initial exploratory information for hypothesis formation in later confirmatory studies.

The annotation-based tools are used for 'blackbox validation' [26] of the SMDT tools (cf. Figure 1) by providing heuristic estimates of average rhythm unit duration, and thus also of average frequency, for comparison with the rhythm formant frequencies found by spectral analysis [1].

3. Demodulation

3.1. Amplitude demodulation

FM and AM time functions demodulated by an RMT tool are shown in Figure 3. The upper panel shows four time functions: demodulated AM, a phonetic correlate of the phonological sonority curve (varying line); demodulated FM, i.e. F0 estimation, a phonetic correlate of intonation, pitch accent and tone (irregular dotted line); linear model of F0 contour direction, here declination (diagonal line); 2^{nd} order polynomial model of contour shape (curved dotted line) [27]; cf. also [28] on regression modelling of tone. The lower panel is discussed in the next subsection, on FM demodulation.

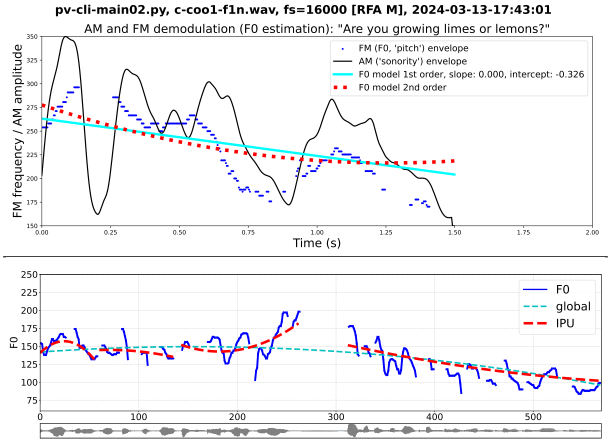

Figure 3: *Upper panel: AM and FM speech demodulation for a reading of 'Do you grow limes or lemons' (IVIE corpus). Lower panel: 2-level regression model hierarchy (2nd order, domains: IPU, dyad, see text; Aix-MARSEC corpus J0104G).*

In RMT, speech AM demodulation is analogous to a simple 'crystal set' demodulation procedure for AM radio:

1. tuning (as a bandpass Butterworth filter, 300–2000 Hz),
2. full-wave rectification of the signal (by a diode bridge in radio, in RMT as the absolute value of the signal),
3. LF smoothing of the resulting AM envelope.

Some related studies use the absolute Hilbert transform [17, 18]; there is little practical difference between the two procedures, however. For an application, see Section 4.

3.2. Frequency demodulation

Demodulation of FM in speech is the functional term for F0 estimation and is not exactly analogous to radio FM demodulation, where the carrier frequency is constant. There are many F0 estimation methods, all of which measure either the periods of F0 cycles in the time domain or harmonic patterns in the frequency domain. Some use straightforward comparison of neighbouring signal intervals, such as autocorrelation (AC) or Average Magnitude Difference Function (AMDF), some use phonatory, auditory or neural models.

In line with the educational aim, simpler models such as AC or AMDF are preferred for initial study. AMDF, a 50 year old [29] but not obsolete variant of AC, is still widely used, especially in musical tone identification and realtime applications, and forms the core of many other improved frequency estimation algorithms such as EAMDF (Extended AMDF) [30, 31, 32]. Other models can be added to RMT as required. Informal comparisons with other methods have shown only minor differences [19].

The basic assumption of AC and AMDF is that the signal shape is repeated after one F0 cycle, and measurements of such repetitions step through the signal, yielding the F0 contour. Starting at a reference interval (e.g. 10...20 ms), similarities with following intervals can be measured until the most similar signal shape is found; the distance from the starting point is the lag or delay T and the frequency estimate is $1/T$. The core of the AMDF implementation can be expressed rather straightforwardly in 'pure Python':

```
frame=signal[framestart:framestop]
movwindows=zip(movwindowrange,
                    movwindowrange+framelength)
diffsums=[np.sum(np.abs(
                frame-signal[winstart:winstop]))
           for winstart,winstop in movwindows]
f0=1/((np.argmin(diffsums)+f0diffoffset)/fs)
```

All speech F0 estimation procedures are error-prone in varying degrees. AMDF errors include (1) effects of noise, (2) weak fundamental and stronger 2^{nd} harmonic, (3) creaky voice, (4) imprecision by tapering (size reduction of the comparison interval towards the end of the reference interval). The RMT F0 tool includes a number of standard measures to ameliorate issues #1, #2 and #4:

1. signal centre-clipping to reduce high frequency noise;
2. tuning to ensure that the relevant frequency range is emphasized (Butterworth band-pass filtering depending on the F0 range, e.g. 70–200 Hz for a deep male voice),
3. taper reduction by continuing comparisons to one frame beyond the reference window;
4. AMDF (actually a subset of EAMDF);
5. non-linear F0 smoothing with a moving median window;
6. regression modelling of the contour.

More complex and more accurate algorithms such as RAPT, REAPER or YIN [33, 34, 35] use different and/or additional operations, can be included as RMT modules, and are recommended for product level deployment (cf. also GitHub for implementations of other algorithms of varying quality). Hybrid neural machine learning models [36] have also been developed for engineering applications but are phonetically non-transparent and require extensive training data, and are thus not directly suitable for the present educational purpose.

The lower panel of Figure 3 in fact uses a RAPT-based RMT tool and shows two levels of FM demodulation as a correlate of hierarchical intonation: a Question-Answer adjacency pair and its constituents, with two levels of 2^{nd} order regression modelling yielding a falling-rising model for the Question and a falling model for the Answer, and an overall holistic rising-falling model for the Question-Answer pair, signifying a form of prosodic cohesion and alignment between questioner and answerer. The temporal contour spans are extracted in this case from a Praat annotation of interpausal units (IPUs), but automatic pause detection also can be used.

4. Rhythm bars: *time* and *frequency*

The term ‘rhythm’ is notoriously ambiguous in linguistics and phonetics. In some phonologies, rhythm is a numerical encoding of structure, based on exocentric headed relations between a ‘strong’ and a ‘weak’ child node of a parent node in a tree graph, with linearisation filters (‘grids’). In linguistic phonetics, rhythm is generally seen as a longer-shorter relation between neighbouring label intervals in annotations, using irregularity indices based on descriptive statistics or neighbour-distance metrics such as nPVI, omitting frequency, tempo and other properties of rhythms.

In the present variant of the modulation-theoretic approach, the demodulated AM or FM signal is analysed by Fast Fourier Transform (FFT) and – if the utterance is indeed rhythmical, which is style-dependent – then peaks can be identified in the LF spectrum below 10 Hz and interpreted as RHYTHM FORMANTS. A broad-band input to spectral analysis is chosen, from which the LF spectra are extracted. The FM contribution to speech rhythm can be analysed in the same way.

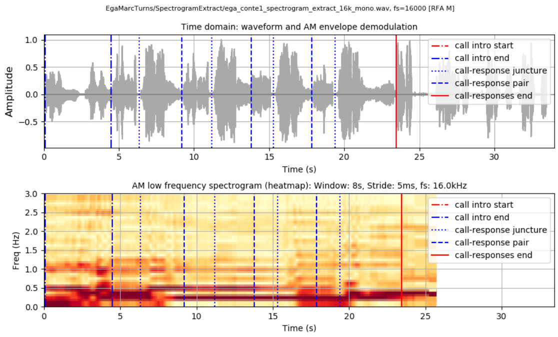

Figure 4: *Rhythm formants in Ega chanting: waveform (upper panel) and LF spectrogram with rhythm bars (lower panel).*

Many studies restrict analysis to the LF spectrum, but this is not adequate: the spectrum is atemporal and even a single beat can show in the LF spectrum. But rhythms have duration – one beat is not a rhythm, neither are two, but three beats provide two intervals for isochrony checking [37].

Consequently, a spectrogram is generated to show the time dimension of rhythm formants, shown by the parallel RHYTHM BARS in Figure 4 [1, 38]. In the figure, pairs of rhythm bars an octave apart show a rhythm hierarchy of interactive chants and their constituent turns. The AM (or FM) analysis comprises:

1. demodulation of the AM or FM envelope;
2. application of the FFT to a sequence of frames in the envelope, to generate spectral slices;
3. extraction of LF speech rhythm frequencies (e.g. 0-5 Hz) in each spectral slice to create the LF spectrogram;
4. visualisation of the LF spectrogram;
5. identification of bars in the LF spectrogram (Figure 4).

These rhythms may be much lower than 5 Hz in rhetorical or poetic rhythms. Figure 4 shows the LF spectrogram of a choral chant (7 s to 17 s), with two formants about an octave apart, at 0.256 Hz corresponding to a period of 3.91 s, and another at 0.469 Hz, corresponding to a period of 2.132 s. The reason for the near-octave relation is that the parallel rhythm bars relate to turns and turn dyads in the call-response structure of the choral chant.

5. Typology: AM and FM LF spectra

Hierarchical clustering is used to show rhythm differences between speech varieties, in this case between story-telling varieties in related West-African languages using the LF AM and FM spectra. The varieties are, in the order shown in Figure 5: Ibibio (Nigeria, ISO 639-3 *ibb*), reading of ‘North Wind and Sun’; Ivory Coast French, newspaper reading; stories by village narrators in Anyi (ISO 639-3 *any*) and Ega (ISO 639-3 *ega*), both Niger-Congo languages in Ivory Coast.

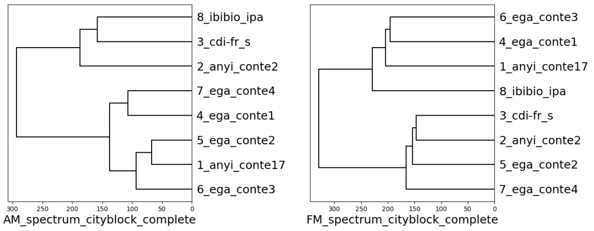

Figure 5: *AM (left) and FM (right) LF spectrum dendrograms.*

The AM (left) and FM (right) LF spectra are hierarchically clustered using the Manhattan Distance metric and farthest neighbour (complete) linkage. The main AM LF spectrum split is between formal styles (readings and a formal story) and informal interactive stories. On the same criteria, the FM dendrogram splits differently: items 2 and 3, both by Anyi native speakers, still cluster together, though with different neighbours. Item 8, Ibibio, clearly has different intonational variability; consequently further study is needed.

6. Discussion and Conclusion

Assessment of the value of the toolkit has a number of dimensions. A design criterion for evaluation is provided by the connected architecture of the toolkit, which provides conceptual coherence within an explicit and flexible theoretical framework. In addition to software assessment, blackbox validation by means of visualisations and comparison with annotation mining is used. Validity of the tools in application has been demonstrated in several studies, as already noted.

As a further method for functional and structural evaluation, the RMT code is available as open source software. The main point of open source code in this context is to provide a starting point for development of computational literacy among students of phonetics in the form of modifiable modules. The interface between the tools is CSV format so the results can be evaluated further with different software if preferred (R, MatLab, Stata, Praat scripts, spreadsheets).

Future work involves integration of other existing tools into the RMT concept. As indicated in the introduction, applications are anticipated in teaching computational phonetics, and also in research relating to modulation theory and to prosodic interface issues in linguistics. The algorithms are agnostic with regard to the data used and can be used in other acoustic scenarios such as music or with other sensor inputs. [1]

[1] Thanks to Bojia Wang, XISU, Xi’an, China, for collaboration on use cases for F0 models in L2 evaluation (cf. Figure 3, upper panel). Many thanks to Reviewers 1 and 2, respectively, for suggestions to relate RMT more clearly to the prosody tool ecosystem and to include worked examples in the code repository. Special thanks to Reviewer 3 for arguing that use of the AMDF algorithm was not sufficiently motivated; this section has been revised accordingly.

7. References

[1] D. Gibbon, "The rhythms of rhythm," *Journal of the International Phonetic Association, First View [online]*, pp. 1–33, 2021, [print] 53(1), 233-245, 2023.

[2] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9/10, pp. 341–345, 2001.

[3] Y. Xu, "ProsodyPro — a tool for large-scale systematic prosody analysis. tools and resources for the analysis of speech prosody," in *Proceedings of the Tools and Resources for the Analysis of Speech Prosody (TRASP) Conference*, Aix-en-Provence, 2013, pp. 7–10.

[4] P. Mertens, "The Prosogram model for pitch stylization and its applications in intonation transcription," in *Proceedings of the Second International Conference on Speech Prosody*, B. Bel and I. Marlien, Eds. Aix-en-Provence: SproSig, 2004.

[5] D. J. Hirst, "A Praat plugin for Momel and INTSINT with improved algorithms for modelling and coding intonation," in *Proceedings of the 16th International Congress of Phonetic Sciences (ICPhS)*, 2007, pp. 1233–1236.

[6] D. Hirst, "ProZed: A speech prosody analysis-by-synthesis tool for linguists," in *Proceedings of the International Conference on Speech Prosody*, 2012, pp. 15–18.

[7] P. Martin, "WinPitch: A multimodal tool for speech analysis of endangered languages," in *Proceedings of Interspeech*, 2011, pp. 3273–3276.

[8] H. Buschmeier and M. Włodarczak, "TextGridTools: A TextGrid processing and analysis toolkit for Python," in *Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, Bielefeld, 2013, pp. 152–157.

[9] D. Gibbon, "Speech rhythms: Learning to discriminate speech styles," in *Proceedings of the 11th International Conference on Speech Prosody*. SProSIG, 2022, pp. 302–306.

[10] H. Traunmüller, "Conventional, biological, and environmental factors in speech communication: a modulation theory," *Phonetica*, vol. 51, no. 1-3, pp. 170–183, 1994.

[11] F. Cummins and R. Port, "Rhythmic constraints on stress timing in English," *Journal of Phonetics*, vol. 26, no. 2, pp. 145–171, 1998.

[12] P. Barbosa, "Explaining cross-linguistic rhythmic variability via a coupled-oscillator model for rhythm production," in *Proceedings of the First International Conference on Speech Prosody*, B. Bel and I. Marlien, Eds. Aix-en-Provence: Laboratoire Parole et Langage, 2002, pp. 163–166.

[13] S. Tilsen and K. Johnson, "Low-frequency Fourier analysis of speech rhythm," *Journal of the Acoustical Society of America*, vol. 124, no. 2, pp. 34–39, 2008.

[14] B. Ludusan, A. Origlia, and F. Cutugno, "On the use of the rhythmogram for automatic syllabic prominence detection," in *Proceedings of Interspeech 2011*, 2011, pp. 2413–2416.

[15] S. Tilsen and A. Arvaniti, "Speech rhythm analysis with decomposition of the amplitude envelope: Characterizing rhythmic patterns within and across languages," *Journal of the Acoustical Society of America*, vol. 134, no. 1, pp. 628–639, 2013.

[16] K. M. Carbonell, R. A. Lester, B. H. Story, and A. J. Lotto, "Discriminating simulated vocal tremor source using amplitude modulation spectra," *Journal of Voice*, vol. 20, no. 2, pp. 140–147, 2015.

[17] L. He and V. Dellwo, "A Praat-based algorithm to extract the amplitude envelope and temporal fine structure using the Hilbert Transform," in *Proceedings of Interspeech 2016*, 2016.

[18] S. Frota, M. Vigário, M. Cruz, F. Hohl, and B. Braun, "Amplitude envelope modulations across languages reflect prosody," in *Proc. Speech Prosody 2022*, 2022, pp. 688–692.

[19] D. Gibbon, "CRAFT: A multifunction online platform for speech prosody visualisation," in *Proceedings of the 19th International Congress of Phonetic Sciences*, Melbourne, 2019, pp. 2956–2960.

[20] D. Gibbon and P. Li, "Quantifying and correlating rhythm formants in speech," in *Proceedings of Linguistic Patterns in Spontaneous Speech (LPSS)*. Taipei: Academia Sinica, 2019, pp. 1–6.

[21] D. Gibbon and X. Lin, "Rhythm zone theory: speech rhythms are physical after all," in *Approaches to the Study of Sound Structure and Speech*, M. Wrembel, A. Kiełkiewicz-Janowiak, and P. Gąsiorowski, Eds. London: Routledge, 2021, pp. 109–128.

[22] D. Gibbon, "New perspectives on Ibibio rhythm," in *Current Issues in Descriptive Linguistics and Digital Humanities A Festschrift in Honor of Professor Eno-Abasi Essien Urua*, M. E. Ekpenyong and I. I. Udoh, Eds. Singapore: Springer Nature, 2022.

[23] X. Lin and D. Gibbon, "Distant rhythms: computing fluency," in *Proceedings of the International Congress of Phonetic Sciences*. Prague: Charles University, 2023.

[24] B. Bigi, "SPPAS: a tool for the phonetic segmentation of speech," in *The eighth International Conference on Language Resources and Evaluation*, Istanbul, Turkey, 2012, pp. 1748–1755.

[25] E. Grabe and E. L. Low, "Durational variability in speech and the Rhythm Class Hypothesis," in *Laboratory Phonology 7*, C. Gussenhoven and N. Warner, Eds. Berlin, New York: De Gruyter Mouton, 2002, pp. 515–546.

[26] D. Gibbon, R. Moore, and R. Winski, Eds., *Handbook of Standards and Resources for spoken Language Systems*. Berlin: Mouton de Gruyter, 1997.

[27] D. Gibbon, "The future of prosody: It's about time," in *Proceedings of the 9th International Conference on Speech Prosody*. SProSIG, 2018, pp. 1–9.

[28] T. Kuczmarski, D. Duran, N. Kordek, and J. Bruni, "Second-degree polynomial model of Mandarin Chinese lexical tone F0 contours," in *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2013.*, P. Wagner, Ed. TUDpress, Dresden, 2013, pp. 218–222.

[29] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg, and H. J. Manley, "Average Magnitude Difference Function pitch extractor," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-22, no. 5, pp. 353–361, 1974.

[30] N. Prukkanon, K. Chamnongthai, Y. Miyanaga, and K. Higuchi, "VT-AMDF, a pitch detection algorithm," in *2009 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2009, pp. 453–456.

[31] Z. Han and X. Wang, "A signal period detection algorithm based on Morphological Self-Complementary Top-Hat Transform and AMDF," *Information*, vol. 10, no. 9, pp. 1–12, 2019.

[32] C. Muhammad, "Extended Average Magnitude Difference Function based pitch detection," *International Arab Journal of Information Technology*, vol. 8, no. 2, 2011.

[33] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," Washington DC, 2005.

[34] ——, "REAPER: Robust Epoch And Pitch EstimatoR," 2014. [Online]. Available: https://github.com/google/REAPER

[35] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, 2002.

[36] K. Wang, J. Liu, Y. Peng, and H. Huang, "Neural rapt: deep learning-based pitch tracking with prior algorithmic knowledge instillation," *International Journal of Speech Technology*, vol. 26, pp. 999–1015, 2023.

[37] S. Nakamura and Y. Sagisaka, "A requirement of texts for evaluation of rhythm in English speech by learners," in *17th International Congress of Phonetic Sciences*, I. P. Association, Ed. Hong Kong: International Phonetics Association, 2011, pp. 1438–1441.

[38] D. Gibbon, "Rhythm pattern discovery in Niger-Congo storytelling," in *Frontiers in Communication*, P. Barbosa, Ed., 2023, vol. 8, pp. 1–18, sec. Psychology of Language; Research Topic: Science, Technology, and Art in the Spoken Expression of Meaning.

Open Source RMT code: http://github/dafyddg/RMT/ (a manual and worked examples are also included).