

# TGA: a web tool for Time Group Analysis

Dafydd Gibbon

Fakultät für Linguistik und Literaturwissenschaft,  
Universität Bielefeld, Germany

gibbon@uni-bielefeld.de

## Abstract

Speech timing analysis in linguistic phonetics often relies on annotated data in *de facto* standard formats, such as Praat TextGrids, and much of the analysis is still done largely by hand, with spreadsheets, or with specialised scripting (e.g. Praat scripting), or relies on cooperation with programmers. The *TGA (Time Group Analyser)* tool provides efficient ubiquitous web-based computational support for those without such computational facilities. The input module extracts a specified tier (e.g. phone, syllable, foot) from inputs in common formats; user-defined settings permit selection of sub-sequences such as inter-pausal groups, and duration difference thresholds. Tabular outputs provide descriptive statistics (including modified deviation models like *PIM*, *PPD*, *nPVI*, *rPVI*), linear regression, and novel structural information about duration patterns, including difference *n-grams* and *Time Trees* (temporal parse trees).

**Index Terms:** web tools, speech timing, speech prosody, annotation processing, duration tokens, time trees

## 1. Background and requirements

Speech timing analysis in linguistic phonetics often relies on time-stamped annotated data in *de facto* standard formats, such as Praat TextGrids [1], Transcriber XML formats or tables with character separated fields (CSV tables). Typical applications are the analysis of speech rate, or measuring duration deviation and relative ‘fuzzy’ isochrony, either relative to the whole sequence, as with standard deviation, *Pairwise Irregularity Measure*, *PMI* [2], *Percentage Foot Deviation*, *PPD* [3], or relative to adjacent units (*raw* and *normalised Pairwise Variability Indices*, *rPVI*, *nPVI* [4]).

The literature reveals several methods for processing time-stamped data, in order of increasing sophistication:

1. copying into spreadsheets for semi-manual processing;
2. use of prefabricated Praat scripts for time-stamped annotations;
3. creation of Praat scripts for specific analysis tasks;
4. implementation of applications in general scripting languages such as *Perl*, *Tcl*, *Ruby* or *Python*, for *TextGrid*, *SAM*, *ESPS*, *WaveSurfer* etc., formats;
5. implementation in languages such as *C*, *C++* (mainly in speech technology applications), independently of time-stamping visualisation software.

The existence of many web applications and spreadsheet templates for manual calculation, sometimes with page user counts, documents the widespread use of (semi-)manual analysis methods. For those with programming abilities, libraries of analysis tools are available, e.g. those in the Aix-MARSEC repository [5], or parsing functions programmed in *Python*, such as the *Natural Language Took Kit*, *NLTK* [6], or

the *TextGrid tools* [7]. The web-based *Time Group Analyser (TGA)* tool, also implemented in *Python*, fills a gap between non-computational and computational users: a wide range of analyses is provided, with no need for *ad hoc* programming. *TextGrid* post-processing with the *TGA* is complementary to *TextGrid* generation with tools such as *SSPAS* [8].

The following account describes *TGA* input (Section 2), processing (Section 3), output (Section 4), and the *Python* implementation (Section 5). The term ‘annotation label’ is used for time-stamped triples  $\langle \text{label}, \text{start}, \text{end} \rangle$ , ‘annotation event’ is used for pairs  $\langle \text{label}, \text{duration} \rangle$ , and ‘Time Group’ refers to an event sequence with a well-defined boundary condition, such as an inter-pausal group or continuous deceleration or acceleration. The *TGA* functions analyse and visualise duration differences ( $\Delta dur$ ) relative to thresholds: deceleration, *rallentando*, quasi-iambic ( $\Delta dur^+$ ); acceleration, *accelerando*, quasi-trochaic ( $\Delta dur^-$ ); equality, threshold-relative ‘fuzzy isochrony’ ( $\Delta dur^=$ ).

## 2. Input and parameter setting

The *TGA* input module extracts a specified tier (e.g. phone, syllable, foot) from inputs in long or short *TextGrid* format or as *CSV* tables with any common separator. User-defined parameter settings currently include the following:

1. freely selected tier name, e.g. ‘Syllables’, and boundary symbol list (e.g. ‘\_’, ‘p’, ‘sil’, ‘\$p’ for pauses);
2. *Time Group* division criterion (by *pauses*; or based on  $\Delta dur$ , i.e. changes in speech rate: decrease (*deceleration*) or increase (*acceleration*));
3. minimal *Time Group* length in duration intervals (where rhythm is concerned, at least 2 interval events (linking 3 point/boundary events) are needed to define a rhythm [9]);
4. global  $\Delta dur$  duration threshold range, e.g. 50...100 ms, 100 ... 200 ms, etc.;
5. local duration  $\Delta dur$  threshold, for local structure determination;
6. local  $\Delta dur$  tokens for visualising duration patterns, e.g. ‘\’, ‘/’, ‘=’ for ‘longer’, ‘shorter’, ‘equal’;
7. *Time Tree* type specification (decelerating, *rallentando*, ‘quasi-iambic’ vs. accelerating, *accelerando*, ‘quasi-trochaic’).

## 3. TGA modules

Currently there are three main *TGA* modules besides I/O and format conversion: (1) text extraction; (2) global basic descriptive statistics for all elements of the specified tier; (3) segmentation of the tier into *Time Groups* with statistics for individual *Time Groups*, and with three new visualisation techniques for  $\Delta dur$  duration patterns: duration difference tokens, duration column charts, and *Time Trees*.

### 3.1. Text extraction

Labels are extracted from annotation elements as running text, separated into sequences by the boundary criteria, e.g. pause, specified in the input. When the annotation has been made without prior transcription there may be a need for text extraction, as documented by a number of web pages providing this functionality, for various purposes such as discourse analysis, natural language processing, archive search, re-use as prompts in new recordings. No further computational linguistic analysis of the text output is undertaken by the TGA at present.

### 3.2. Global descriptive statistics

For calculating global descriptive statistics, three versions of the data are prepared: (1) with all annotation elements on the tier, including boundary elements (e.g. pauses); (2) with only non-boundary elements; (3) with only boundary elements. The following information is provided for each data version:

1.  $n$ ,  $len$ : the number of elements in the input (for data versions with or without pauses, or only pauses), and the total duration  $\Delta t$ ;
2.  $min$ ,  $max$ ,  $mean$ ,  $median$ ,  $range$ : basic statistical properties;
3.  $standard\ deviation$ ,  $PIM$ ,  $PFD$ ,  $rPVI$ ,  $nPVI$ : ratio or difference measures of deviation  $\Delta dur$ , of an element from a reference value, e.g. mean or adjacent element;
4.  $linear\ regression\ (intercept,\ and\ slope)$ : slope indicates the average rate of duration change in the data (deceleration, acceleration, equality).

The  $PIM$ ,  $PFD$  and  $rPVI$  metrics are distinguished partly for their popularity, rather than for significant differences:  $PIM$  and  $PFD$  relate closely to standard deviation, though the  $PIM$  uses global ratios rather than differences. The  $nPVI$  on the other hand factors out drifting speech rates, and may thus diverge very widely from the mean. The measures are claimed to be rhythm metrics, though they define only necessary, not sufficient conditions for rhythm: unsigned  $\Delta dur$  values ignore alternation in duration patterns, a necessary condition for rhythm models (cf. Section 3.3). The formulae for  $PIM$ ,  $PFD$ ,  $rPVI$  and  $nPVI$  are shown in Table 1.

Statistical ‘rule of thumb’ quality scores, such as  $p$ -value or confidence intervals, are not included at this time.

$PIM(I_{1..n})$	$= \sum_{i \neq j} \left  \log \frac{I_i}{I_j} \right $
$PF D(foot_{1..n})$	$= \frac{100 \times \sum  MFL - len(foot_i) }{len(foot_{1..n})}$ where $MFL = \frac{\sum_{i=1}^n len(foot_i)}{n}$
$rPVI(d_{1..m})$	$= \sum_{k=1}^{m-1}  d_k - d_{k+1}  / (m-1)$
$nPVI(d_{1..m})$	$= 100 \times \sum_{k=1}^{m-1} \left  \frac{d_k - d_{k+1}}{(d_k + d_{k+1})/2} \right  / (m-1)$

Table 1: Definitions of  $PIM$ ,  $PF D$ ,  $nPVI$  measures.

### 3.3. Local Time Group statistics

Basic statistics and linear regression are calculated for each *Time Group* separately in the same way as for the global calculations. Minimal difference thresholds permit approximate (i.e. ‘fuzzy isochrony’) measures, rather than

strict time-stamp differences. Three novel structural  $\Delta dur$  pattern visualisations are defined:

1. tokenisation of duration differences  $\Delta dur$  into ‘longer’, ‘shorter’ and ‘equal’ duration difference tokens, represented by character symbols (cf. Figure 4 and Table 2), to support prediction of whether specific properties such as *rhythmic alternation* are likely to make sense;
2. top-suspended column chart illustrating the duration  $\Delta t$  of elements in the *Time Group* (Figure 4);
3. duration parse tree (*Time Tree*) for each *Time Group* (Figure 5), based on signed duration differences  $\Delta dur^+$  and  $\Delta dur^-$ , [10], [11], to facilitate study of correspondences between duration hierarchies and grammatical hierarchies.

The *Time-Tree* induction algorithm follows a standard deterministic context-free bottom-up left-right shift-reduce parser schedule. The grammars use  $\Delta dur^+$  and  $\Delta dur^-$  tests on annotation events in order to induce two types of *Time Tree*, with ‘quasi-iambic’ (decelerating, *rallentando*), and ‘quasi-trochaic’ (accelerating, *accelerando*) constituents:

*Quasi-iambic*:

$$TT_k \rightarrow TT_i TT_j$$

$$duration(TT_i) < duration(TT_j)$$

$$duration(TT_k) \text{ INHERITS } duration(TT_j)$$

*Quasi-trochaic*:

$$TT_k \rightarrow TT_i TT_j$$

$$duration(TT_i) > duration(TT_j)$$

$$duration(TT_k) \text{ INHERITS } duration(TT_i)$$

In each of these grammars, a right-hand side  $TT$  is a label-duration pair, and higher levels in the tree inherit durations recursively from the constituent annotation events.

Crucially,  $\Delta dur$  token patterns and *Time Trees*, (unlike *standard deviation*,  $PIM$ ,  $PF D$ ,  $rPVI$ ,  $nPVI$ ) use signed, not unsigned duration differences, and may therefore lay claim to representing true rhythm properties. In each case, the minimal local difference threshold setting applies.

## 4. Output

The output provides various list and table formats:

1. list of label text sequences within *Time Groups*, with any accompanying symbols for boundary events;
2. table of *Time Group* properties:
  1. statistical properties,
  2. tokenised  $\Delta dur^+$  and  $\Delta dur^-$  deceleration-acceleration patterns,
  3. top-suspended column charts of durations,
  4. *Time Trees* built on the  $\Delta dur^+$  or  $\Delta dur^-$  relations;
3. table with summary of basic statistics, linear regression, and correlations between different statistics, for the complete set of *Time Groups*;
4. list of  $\Delta dur$  duration difference token  $n$ -grams from all *Time Groups* (unigrams, digrams, trigrams, quadgrams and quingrams) to support analysis of rhythmically alternating patterns in the annotations;
5. various character-separated value tables of input and output for further analysis using other software.

## 5. Implementation

The architecture of the TGA tool implementation is visualised in Figure 1. The user inputs the annotation in a TextGrid or CSV format using an HTML form and selects the



Rank	Percent	Count	Token digram
1	22%	60	/\
2	20%	55	\/
3	11%	31	\ \

Table 2: *Adur* token rank and frequency analysis.

In this instance of ‘educated Southern British’ pronunciation, i.e. slightly modified Received Pronunciation (RP), alternations figure at the top two ranks, totalling 42% of the digrams, and therefore have potential for rhythm; deceleration patterns occupy rank 3.

Finally, perhaps the most interesting display format is the *Time Tree* visualisation, here shown as automatically generated nested parentheses. The example in Figure 5 illustrates this principle with the inter-pausal group ‘about Anglican ambivalence to the British Council of Churches’.

```
( ( (@ baUt)
  ( ( ({N gLI}
    (kn {m})
    (bI vl@ns)))
  ( ( ( (t@ D@)
    (brI tIS))
    ( ( kaUn
      ( sl
        (@v tS3:)))
      tSIz))
    PAUSE))
```

Figure 5: Automatic prettyprint of a quasi-iambic *Time Tree* in nested parenthesis notation.

The purpose of generating *Time Tree* output is to support study of the relation between temporal hierarchical structures and grammatical constituents in a systematic *a posteriori* manner, rather than postulating higher level units such as feet or other events types in an *a priori* prosodic hierarchy framework. This example shows a number of correspondences with grammatical units at different depths of embedding, e.g. ‘about’, ‘British’, ‘Anglican ambivalence’, ‘about Anglican ambivalence’, ‘Council of Churches’, ‘to the British Council of Churches’, including foot sequences of Jassem’s ‘Anacrusis + Narrow Rhythm Unit’ type [12].

## 6. Conclusion and outlook

The design and implementation of a web tool for support of linguistic phonetic analysis of speech timing, using time-stamped data, are described. Extensive basic statistical information is provided, including linear regression and correlations between different statistics. Three innovative visualisations are introduced: *Adur* duration difference tokens; top-suspension column charts for  $\Delta t$  and  $\Delta dur$  visualisation, and *Adur* based *Time Trees* automatically represented as nested parentheses. Informal evaluation by four trained phoneticians shows that the TGA tool reduces previous analysis times for time-stamped annotations by several orders of magnitude. Initial work on Mandarin Chinese is reported in [13] and [14]. An offline version of TGA for processing large annotation corpora rather than single files is undergoing testing.

The tokenisation and *Time Tree* techniques are very much research in progress. Ongoing work concerns extension of TGA functionality, particularly correspondences between *Time Groups* and grammatical, focus-based and rhetorical categories, coupled with the automatic discovery of inter-tier time relations based on temporal logics [15].

A recent version of the TGA, with the data illustrated in the present paper, can currently be accessed at the following URL:

<http://wwwhomes.uni-bielefeld.de/gibbon/TGA>

## 7. Acknowledgments

Special thanks are due to Jue Yu, Zhejiang University, Hangzhou, China, to Katarzyna Klessa and Jolanta Bachan, Adam Mickiewicz University, Poznań, Poland, for feedback on applications of the TGA, and to participants and reviewers of O-COCOSDA 2012, Macau, as well as reviewers of TRASP 2013 for very helpful critical discussion.

## 8. References

- [1] Boersma, P. “Praat, a system for doing phonetics by computer”, *Glott International* 5:9/10, 341-345, 2001.
- [2] Scott, D. R., Isard, S. D. and de Boysson-Bardies, B. “On the measurement of rhythmic irregularity: a reply to Benguerel”, *Journal of Phonetics* 14. 327–330, 1986.
- [3] Roach, P. “On the distinction between ‘stress-timed’ and ‘syllable-timed’ languages”. In *Linguistic Controversies: Essays in Linguistic Theory and Practice*, D. Crystal, Ed., London: Edward Arnold, 73–79, 1982.
- [4] Low, E. L., Grabe, E. and Nolan, F. “Quantitative characterisations of speech rhythm: Syllable-timing in Singapore English”. *Language and Speech* 43(4):377–401, 2000.
- [5] Hirst, D. Auran, C. and Bouzon, C. The Aix-MARSEC database. 2002-2004. Tech. Report, Equipe Prosodie et Représentation Formelle du Langage, Laboratoire CNRS UMR 6057 Parole et Langage, Université de Provence, Aix-en-Provence, 2009.
- [6] Bird, S., Klein, E. and Loper, E. *Natural Language Processing with Python*. Beijing, etc.: O’Reilly, 2009.
- [7] Buschmeier, H., and Włodarczak, M. “TextGridTools: A TextGrid Processing and Analysis Toolkit for Python”. *Tagungsband der 24. Konferenz zur Elektronischen Sprachsignalverarbeitung (ESSV 2013)*, Bielefeld, Germany, 152–15, 2013.
- [8] Bigi, B. SPPAS: a tool for the phonetic segmentations of speech, LREC 8, Istanbul, 2012.
- [9] Gibbon, D. “Computational modelling of rhythm as alternation, iteration and hierarchy,” in *Proceedings of ICPhS 15*, Barcelona, 2003.
- [10] Gibbon, D. “Corpus-based syntax-prosody tree matching”, in *Proceedings of Eurospeech 2003*, Geneva. 2003.
- [11] Gibbon, D. “Time Types and Time Trees: Prosodic Mining and Alignment of Temporally Annotated Data”. In: Sudhoff, S. et al. (2006). *Methods in Empirical Prosody Research*. Berlin: Walter de Gruyter, 281-209, 2006.
- [12] Gibbon, D., Hirst, D. and Campbell, N. 2012. *Rhythm, Melody and Harmony: Studies in Honour of Wiktor Jassem*. Poznań: Polish Phonetics Society (Speech & Language Technology 14/15).
- [13] Yu, J. and Gibbon, D. “Criteria for database and tool design for speech timing analysis with special reference to Mandarin”, *Proceedings of Oriental COCOSDA*, Macau, 2012.
- [14] Yu, J. 2013. *Timing analysis with the help of SPPAS and TGA tools*. TRASP 2013.
- [15] Carson-Berndsen, J. *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition*, Dordrecht: Kluwer Academic Publishers, 1998.