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*, *PFD*, *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*, *PFD* [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 <*label*, *start*, *end*>, 'annotation event' is used for pairs <*label*, *duration*>, 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 (Δdur) relative to thresholds: deceleration, rallentando, quasi-iambic (Δdur^{+}); acceleration, accelerando, quasi-trochaic (Δdur^{-}); 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 Δdur, 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 Δdur duration threshold range, e.g. 50...100 ms, 100 … 200 ms, etc.;
5. local duration Δdur threshold, for local structure determination;
6. local Δdur 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 Δdur 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 *Δt*;
2. *min*, *max*, *mean*, *median*, *range*: basic statistical properties;
3. *standard deviation*, *PIM*, *PFD*, *rPVI*, *nPVI*: ratio or difference measures of deviation *Δdur*, 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 *Δdur* 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.

$PIM(I_{1,\dots,n})$ =	$\sum_{i\neq j} \lvert\log\frac{I_i}{I_j}\rvert$
$PFD(foot_{1\dots n})$ =	$\frac{100\times\sum \lvert MFL-len(foot_i)\rvert}{len(foot_{1\dots n})}$ where MFL = $\frac{\sum_{i=1}^{n} len(foot_i)}{n}$
$rPVI(d_{1\dots m})$ =	$\sum_{k=1}^{m-1} \lvert d_k-d_{k+1}\rvert/(m-1)$
$nPVI(d_{1\dots m})$ =	$100\times\sum_{k=1}^{m-1} \lvert\frac{d_k-d_{k+1}}{(d_k+d_{k+1})/2}\rvert/(m-1)$

Table 1: Definitions of *PIM*, *PFD*, *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 *Δdur* pattern visualisations are defined:

1. tokenisation of duration differences *Δdur* 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 *Δt* 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 Δdur^{+} and Δdur^{-}, [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 Δdur^{+} and Δdur^{-} 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) INHERITS *duration*(TT_j)

Quasi-trochaic:

$TT_k \rightarrow TT_i\ TT_j$

duration(TT_i) > *duration*(TT_j)

duration(TT_k) 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, *Δdur* token patterns and *Time Trees*, (unlike *standard deviation*, *PIM*, *PFD*, *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 Δdur^{+} and Δdur^{-} deceleration-acceleration patterns,
   3. top-suspended column charts of durations,
   4. *Time Trees* built on the Δdur^{+} or Δdur^{-} 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 *Δdur* 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 required processing settings. The input is passed over the internet via the Common Gateway Interface (CGI) to the TGA server; processing is performed in Python, and the output returns to the user as an HTML page.

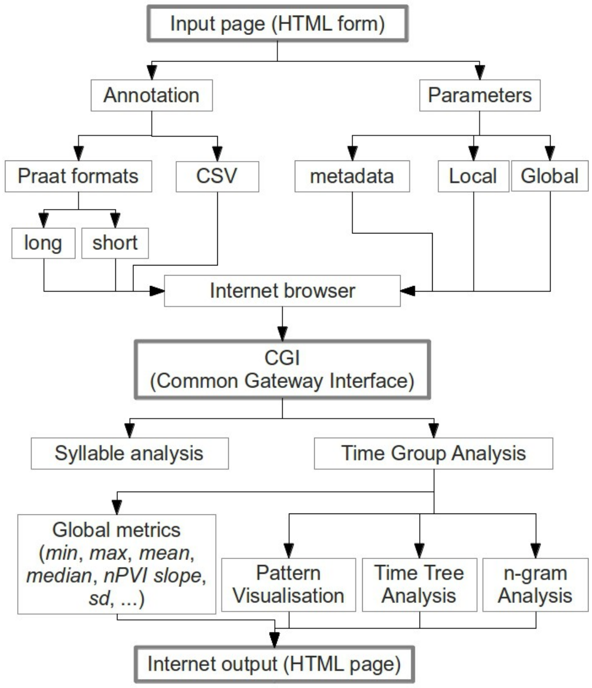


Figure 1: TGA architecture.

Currently accepted input formats are Praat short and long TextGrids, or CSV formats with various field separator options. The input screen layout is shown in Figure 2.

A number of output selection options are also provided on the input page: annotation text, global descriptive statistics, metrics for individual *Time Groups*, token patterns, *Time Trees* or selected output in CSV tables for further processing with spreadsheets, etc.

All the following illustrations are from TGA output for the syllable tier of Aix-MARSEC annotation A0101B.

The automatic label text output from the annotation elements in the *Time Groups* appears straightforwardly, as a list of *Time Group* text sequences:

'gUd 'mO:nIN _

'mO: 'nju:z @'baUt D@ 'revr@n 'sVn 'mjVN 'mu:n _

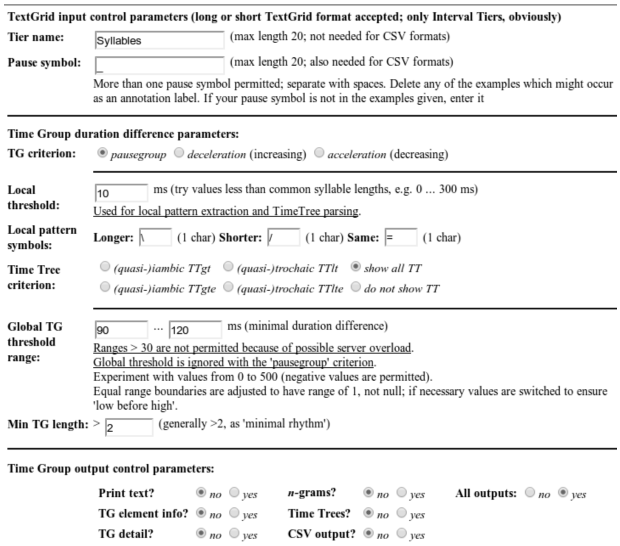
'faUnd@r @v D@ ,ju:nIfI'keISn 'tS3:tS _

'hu:z 'kVr@ntlI In 'dZeIl _

The quantitative output types display as tables, both for individual *Time Groups* and for generalisations over individual *Time Groups*, as in Figure 3.

Figure 4 shows two aligned novel visualisation types: duration tokens and duration bars. The top sequence of symbols represents tokenisations of *Δdur* between adjacent intervals, in this case showing a possibly rhythmical shorter-longer alternation (*Δdur* tokenisation is controlled by adjusting the local *Δdur* threshold setting).

The top-suspended column chart below the token sequence provides an iconic visualisation of durations, in width (to show time scaling) and in height. Top-suspension emphasizes the rallentando (deceleration, iambic, downwards) and accelerando (acceleration, trochaic, upwards) tendencies, providing immediate visual sources of hypotheses about rhythmicality for perceptual testing and linguistic analysis.



TextGrid input control parameters (long or short TextGrid format accepted; only Interval Tiers, obviously)

Tier name: Syllables (max length 20; not needed for CSV formats)

Pause symbol: _ (max length 20; also needed for CSV formats)

More than one pause symbol permitted; separate with spaces. Delete any of the examples which might occur as an annotation label. If your pause symbol is not in the examples given, enter it

Time Group duration difference parameters:

TG criterion: *pausegroup* *deceleration* (increasing) *acceleration* (decreasing)

Local threshold: 10 ms (try values less than common syllable lengths, e.g. 0 ... 300 ms)
Used for local pattern extraction and TimeTree parsing.

Local pattern symbols: **Longer:** \ (1 char) **Shorter:** / (1 char) **Same:** = (1 char)

Time Tree criterion: *(quasi-)iambic TTgt* *(quasi-)trochaic TTlt* *show all TT* *(quasi-)iambic TTgte* *(quasi-)trochaic TTlte* *do not show TT*

Global TG threshold range: 90 ··· 120 ms (minimal duration difference)
Ranges > 30 are not permitted because of possible server overload.
Global threshold is ignored with the 'pausegroup' criterion.
Experiment with values from 0 to 500 (negative values are permitted).
Equal range boundaries are adjusted to have range of 1, not null; if necessary values are switched to ensure 'low before high'.

Min TG length: > 2 (generally >2, as 'minimal rhythm')

Time Group output control parameters:

Print text? *no* *yes* ***n*-grams?** *no* *yes* **All outputs:** *no* *yes*
TG element info? *no* *yes* **Time Trees?** *no* *yes*
TG detail? *no* *yes* **CSV output?** *no* *yes*

Figure 2: Screenshot of parameter input options.

Summary table of global and accumulated TG duration functions (some do make sense...)
Time Group criterion: pausegroup, local threshold: 10, Min valid TG length: 2
Only inter-pause intervals measured; pauses not included

Overall duration:	48504	Overall raw longer, ms:	15401	Overall raw shorter, ms:	14521
Overall min:	20.00	Overall max:	990.00	Overall range:	970.00
Valid Time Groups:	34	Overall rate/sec:	5.67		
Components: global tendencies					
Overall mean:	176.38	Overall median:	150.00	Overall SD:	113.58
Overall npvi:	62.00	Overall intercept:	156.12	Overall slope:	0.15
Mean of means:	182.18	Median of means:	176.70	SD of means:	34.75
Mean of medians:	168.68	Median of medians:	160.00	SD of medians:	40.88
Mean of SDs:	90.02	Median of SDs:	86.16	SD of SDs:	39.87
Mean of nPVIs:	60.00	Median of mnPVIs:	51.00	SD of nPVIs:	17.91
Mean of intercepts:	143.59	Median of intercepts:	130.80	SD of intercepts:	71.16
Mean of slopes:	10.65	Median of slopes:	11.86	SD of slopes:	41.10
Components: correlations					
mean::TGdur:	-0.190	median::TGdur:	-0.427	SD::TGdur:	0.230
nPVI::TGdur:	0.097	slope::TGdur:	0.061	intercept::TGdur:	-0.178
nPVI::mean:	0.128	slope::mean:	0.028	intercept::mean:	0.503
nPVI::median:	0.026	slope::median:	0.005	intercept::median:	0.310
nPVI::SD:	0.383	slope::SD:	0.051	intercept::SD:	0.229

Figure 3: Screenshot of summary of collated *Time Group* properties and correlations.

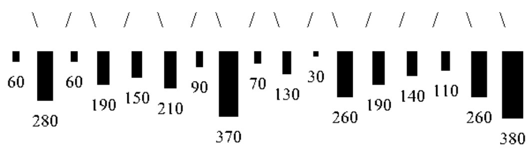


Figure 4: Duration difference token pattern (above) and top-suspended duration columns with duration represented by both width and length (below).

The *Δdur* token digram analysis provides the following output format, showing rank and frequency of token digrams (see Table 2 for token frequencies above 10%).

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

Table 2: *Δdur* 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: *Δdur* duration difference tokens; top-suspension column charts for *Δt* and *Δdur* visualisation, and *Δdur* 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”, Glot 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 Wlodarczak, 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.