

# Time-Group Analysis: A Tutorial

Dafydd Gibbon

Summer School on Contemporary Phonetics and Phonology,  
Chongqing, Sichuan Foreign Studies University,  
2025-07-22 (online)

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

[https://uni-bielefeld.zoom-x.de/j/65248112276?  
pwd=INfKs4Rwe3tjnJicHt7s7aLTPxaMKw.1](https://uni-bielefeld.zoom-x.de/j/65248112276?pwd=INfKs4Rwe3tjnJicHt7s7aLTPxaMKw.1)

Meeting-ID: 652 4811 2276

Password: 932610



# Theoretical Background to Tools like TGA

## SOUNDS AND SYMBOLS

*PHONOLOGY*     *LINGUISTIC  
PHONETICS*

Domains:

- Auditory (mainly)
- Intuitions
- Close relation with writing

Methods:

- Self-observation
- Other-observation
- Transcription
- Intuitive understanding
- Hypothesis
- Formalisation
- Theory formation

## SOUNDS AND SIGNALS

*SIGNAL  
PHONETICS*

Domains:

- Articulatory
- Acoustic
- Auditory

Methods:

- Intuition
- Hypothesis
- Measurements
- Statistics
- Validation, test
- Formal theories

# Theoretical Background to Tools like TGA

## SOUNDS AND SYMBOLS

*PHONOLOGY*    *LINGUISTIC  
PHONETICS*

Domains:

- Auditory (mainly)
- Intuitions
- Close relation with writing

Methods:

- Self-observation
- Other-observation
- Transcription
- Intuitive understanding
- Hypothesis
- Formalisation
- Theory formation

## SOUNDS AND SIGNALS

*SIGNAL  
PHONETICS*

Domains:

- Articulatory
- Acoustic
- Auditory

Methods:

- Intuition
- Hypothesis
- Measurements
- Statistics
- Validation, test
- Formal theories

# Theoretical Background to Tools like TGA

## SOUNDS AND SYMBOLS

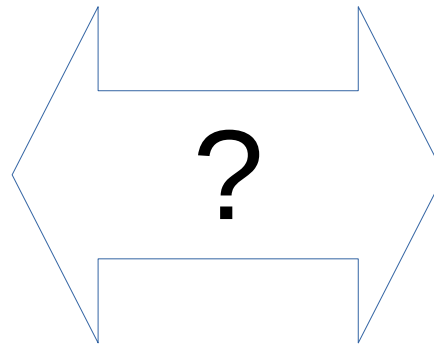
*PHONOLOGY*    *LINGUISTIC  
PHONETICS*

Domains:

- Auditory (mainly)
- Intuitions
- Close relation with writing

Methods:

- Self-observation
- Other-observation
- Transcription
- Intuitive understanding
- Hypothesis
- Formalisation
- Theory formation



## SOUNDS AND SIGNALS

*SIGNAL  
PHONETICS*

Domains:

- Articulatory
- Acoustic
- Auditory

Methods:

- Intuition
- Hypothesis
- Measurements
- Statistics
- Validation, test
- Formal theories

# Theoretical Background to Tools like TGA

## SOUNDS AND SYMBOLS

*PHONOLOGY*    *LINGUISTIC  
PHONETICS*

## SOUNDS AND SIGNALS

*SIGNAL  
PHONETICS*

Domains:

- Auditory (mainly)
- Intuitions
- Close relation with

Domains:

- Articulatory
- Acoustic
- Acoustic

**BRIDGES  
ACROSS  
THE SIGNAL-  
SYMBOL  
GAP?**

Methods:

- Self-observation
- Other-observation
- Transcription
- Intuitive understanding
- Hypothesis
- Formalisation
- Theory formation

Methods:

- Intuition
- Hypothesis
- Measurements
- Statistics
- Validation, test
- Formal theories

# Theoretical Background to Tools like TGA

## SOUNDS AND SYMBOLS

## SOUNDS AND SIGNALS

*PHONOLOGY*    *LINGUISTIC  
PHONETICS*

*SIGNAL  
PHONETICS*

### LabPhon

- Experiments

Domains:

- Auditory (mainly)
- Intuitions
- Close relation with writing

Domains:

- Articulatory
- Acoustic
- Auditory

Methods:

- Self-observation
- Other-observation
- Transcription
- Intuitive understanding
- Hypothesis
- Formalisation
- Theory formation

### TOOLS

- Praat
- Anno-Pro
- TGA
- 
- R, Python
- ChatGPT, DeepSeek

Methods:

- Intuition
- Hypothesis
- Measurements
- Statistics
- Validation, test
- Formal theories



# What is TGA?

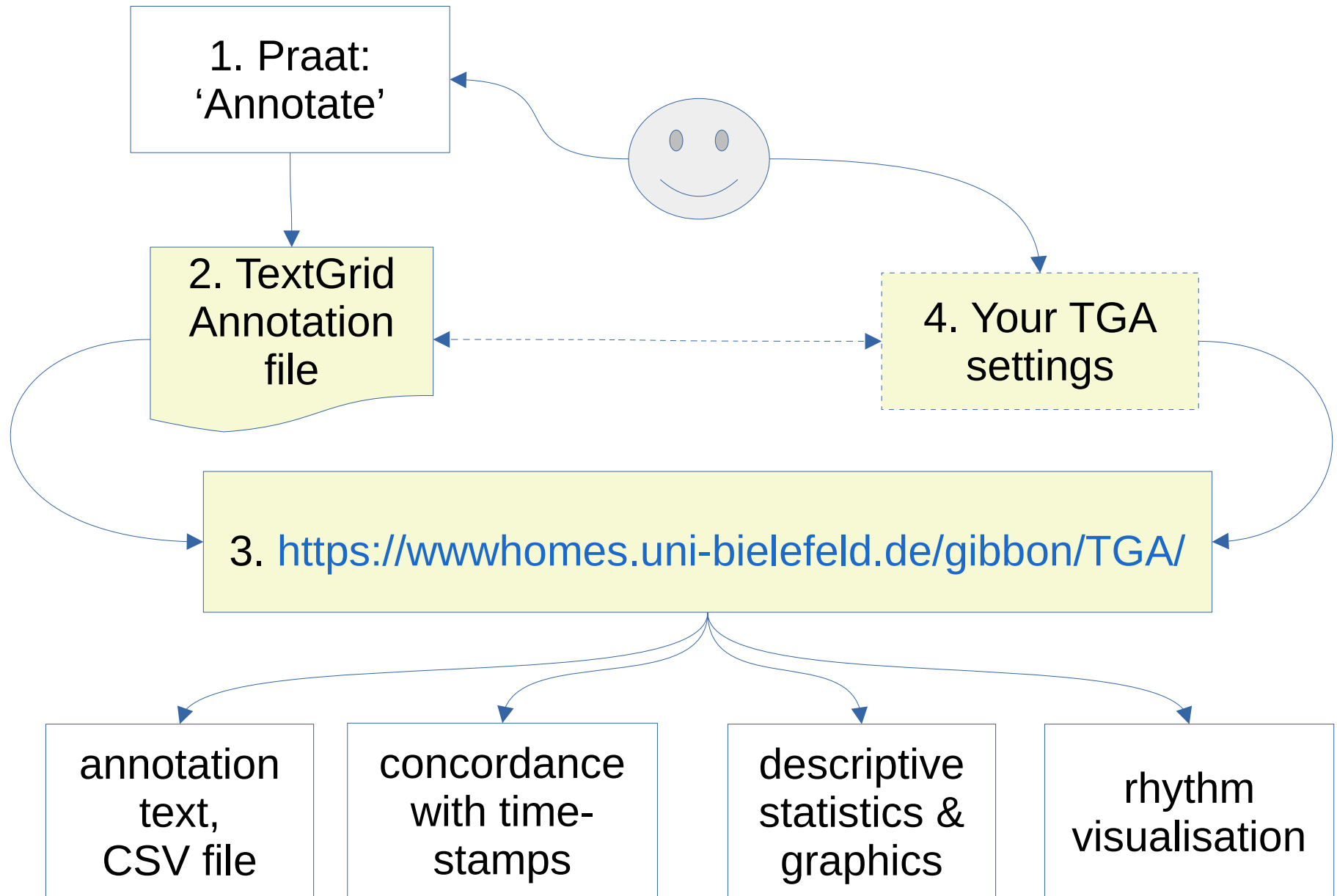
- “TGA” means “Time Group Analyser”.
- “Time Group” in this context: “Interpausal Unit”
- “Interpausal Unit”: the segment of an utterance between two pauses, for example:  
*he came <pause> he said hello <pause> and left*  
Interpausal units: “*he came*”, “*he said hello*”, “*and left*”
- “Pause” in this context: an unfilled pause (not a filled pause like *um*, *er*, ...), longer than ~200ms
  - Any symbol such as “\_”, “p”, “silence”, etc. can be used, but it must be used consistently.
  - And a pause can be whatever you want it to be!

# Speech annotation mining (e.g. rhythm analysis)

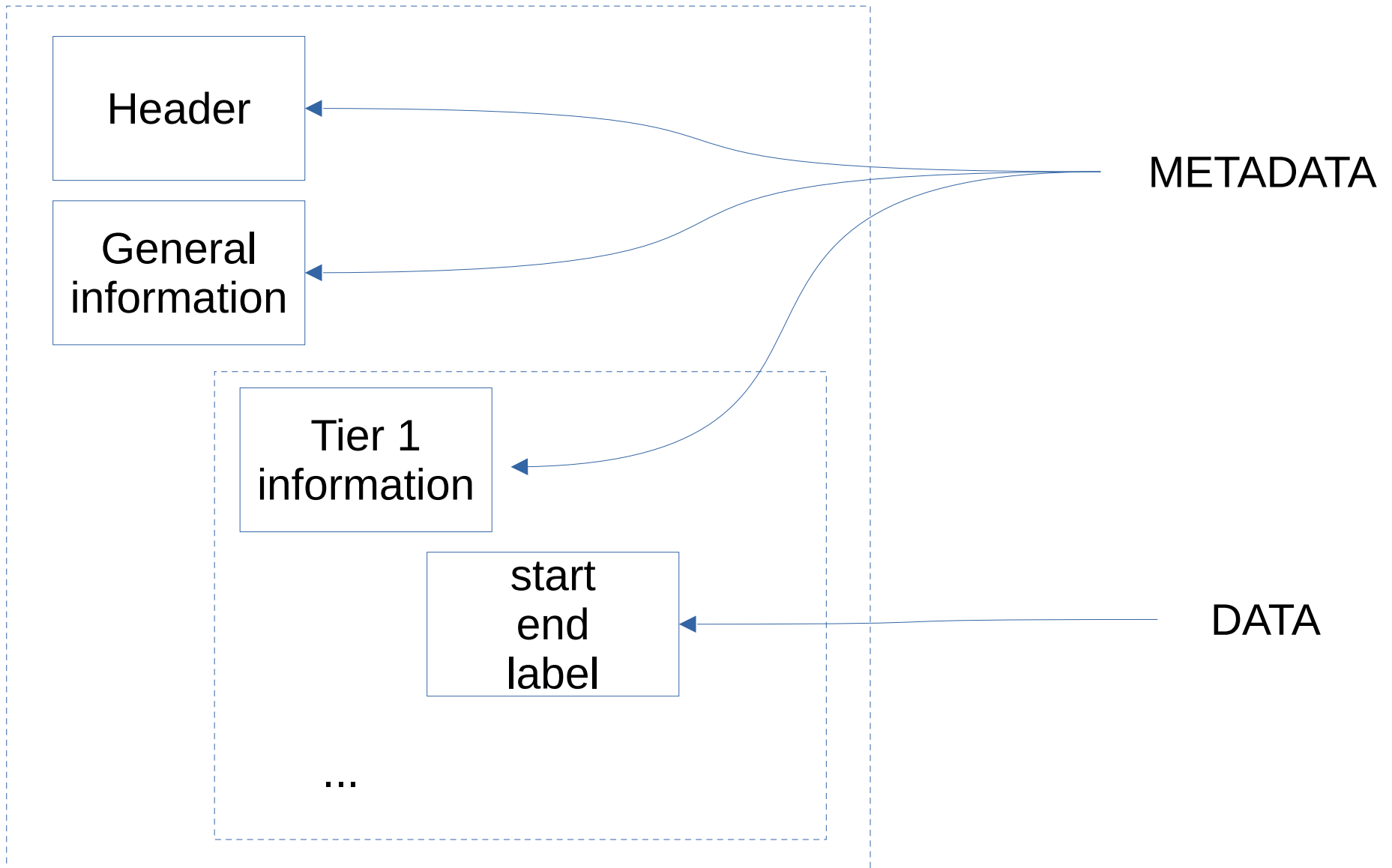
- “Speech”:  
spoken language
- “Annotation mining”:  
extraction of information from a speech annotation for the purpose of linguistic and statistical analysis
- “Information” is organised in tiers (levels):  
for example phones, tones, phonemes, morphemes, words, sentences, speech acts, intonations, ...
- “Speech annotation”:  
pair: <time-interval, label>  
in detail: <time-start, time-end, label>  
example: <0.453, 0.937, “chips”>

Note: Praat also allows the <**time-point, label**> annotation type, but TGA analyses only the <**time-interval, label**> annotation type.

# TGA workflow



# TextGrid file structure



# TextGrid Example

File type = "ooTextFile"

Object class = "TextGrid"

xmin = 0

xmax = 6.637428225932263

tiers? <exists>

size = 2

item []:

  item [1]:

    class = "IntervalTier"

    name = "Syllables"

    xmin = 0

    xmax = 6.637428225932263

    intervals: size = 33

    intervals [1]:

      xmin = 0

      xmax = 0.10080428396194088

      text = "\_"

# TextGrid Example

File type = "ooTextFile"  
Object class = "TextGrid"

xmin = 0  
xmax = 6.637428225932263  
tiers? <exists>  
size = 2  
item []:  
  item [1]:  
    class = "IntervalTier"  
    name = "Syllables"  
    xmin = 0  
    xmax = 6.637428225932263  
    intervals: size = 33  
    intervals [1]:  
      xmin = 0  
      xmax = 0.10080428396194088  
      text = "\_"

intervals [1]:  
  xmin = 0  
  xmax = 0.10080428396194088  
  text = " "  
intervals [2]:  
  xmin = 0.10080428396194088  
  xmax = 0.3043804625479265  
  text = "our"  
intervals [3]:  
  xmin = 0.3043804625479265  
  xmax = 0.5592462322036338  
  text = "foot"  
intervals [4]:  
  xmin = 0.5592462322036338  
  xmax = 0.8237319715738414  
  text = "ball"  
intervals [5]:  
  xmin = 0.8237319715738414  
  xmax = 1.117127054965459  
  text = "team"

# Using TGA

## TGA: Time Group Analyzer

An online tool for speech annotation mining

Dafydd Gibbon (Universität Bielefeld)

**The TGA tool is temporarily out of action due to central web service restructuring.**

Operation depends on central installation of the Python graphics module and will be resumed when this module has been installed.

[Temporary TGA site](#)

(Will be de-activated when the main TGA tool is fully available again.)

For specific purposes the following tools, which do not require graphics, are available:

TextGrid to CSV conversion: [TextGrid2CSV](#)

TGA-mini - TextGrid analyser: [TextGrid Analyser](#)

Descriptive statistics: [CalcuCopia](#)

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

# TGA: Time Group Analyzer

## An online tool for speech annotation mining

Dafydd Gibbon (Universität Bielefeld)

However, for specific purposes the following tools, which do not require graphics, are available:

TextGrid to CSV conversion: [TextGrid2CSV](#)

TGA-mini - TextGrid analyser: [TextGrid Analyser](#)

Descriptive statistics: [CalcuCopia](#)

**CITATION:** In publications which use the online TGA tool, please cite:

Gibbon, Dafydd and Jue Yu. 2016. "Time Group Analyzer: Methodology And Implementation." *The Phonetician* 111/112:9-34.

See [below](#) for a list of papers for which the TGA tool has been used.

[Demos](#) - [Graphics](#) - [Papers](#) - [Various notes](#)

V 1.00 2012-07-09

V 3.03 2013-03-30

V 3.04 2015-09-08

V 4.00 2016-01-01

V 5.00 2016-02-13



### **Note:**

1. To use the TGA online annotation mining tool, proceed to one of the demos, and replace the demo annotation with your own. Then adjust the parameters for the kind of analysis you are looking for.
2. The TGA application has now been re-designed as a multi-user system.
3. However, server space is limited, and therefore graphics files which are older than a certain time (initially set at 2 min) are removed when TGA is re-run, either by yourself or by another user.
4. Consequently, it can happen that someone else may unintentionally delete your graphics files. This will only affect you when you need to download the files.
5. If you have created graphics files and need to download them but the file has been removed because someone else has re-run the TGA, then simply reload with the browser.
6. If there are many users, this limit may cause problems. If you experience problems with this policy, please let me know and I will temporarily increase the wait time before cleanup.
7. Graphics files have the format: "TGA\_PID\_\*.png" (PID is the process ID).

# Sample data – start with one of these

1. To use the TGA online annotation mining tool, proceed to one of the demos, and replace the demo annotation with your own. Then adjust the parameters for the kind of analysis you are looking for.
2. The TGA application has now been re-designed as a multi-user system.
3. However, server space is limited, and therefore graphics files which are older than a certain time (initially set at 2 min) are removed when TGA is re-run, either by yourself or by another user.
4. Consequently, it can happen that someone else may unintentionally delete your graphics files. This will only affect you when you need to download the files.
5. If you have created graphics files and need to download them but the file has been removed because someone else has re-run the TGA, then simply reload with the browser.
6. If there are many users, this limit may cause problems. If you experience problems with this policy, please let me know and I will temporarily increase the wait time before cleanup.
7. Graphics files have the format: "TGA\_PID\_\*.png" (PID is the process ID).

[Demos](#) - [Graphics](#) - [Papers](#) - [Various notes](#)

## TGA demos

1. Farsi (SM: "The North Wind and the Sun")
  1. [Female](#)
  2. [Female-2.html](#)
  3. [Female-7.html](#)
  4. [Male-7.html](#)
  5. [Male-9.html](#)
2. Mandarin
  1. Syllables: [read-aloud Mandarin](#) from CASS corpus, [Praat long TextGrid format](#), default tier: [PY](#) (syllable).
  2. Syllabic lexical tone: [read-aloud Mandarin](#) from CASS corpus, [Praat long TextGrid format](#), default tier: [Tone](#) (includes box-and-whisker plot for different label types).
  3. Syllable rhymes: [read-aloud Mandarin](#) from Jue Yu corpus, [Praat long TextGrid format](#), default tier: [PY](#) (syllable).
3. Tem
  1. Syllables: [read-aloud Tem](#) from Tchagbale corpus for Tem<Gur<Niger-Congo (ISO 639-3 kdh, Togo), [Praat long TextGrid format](#), default tier: [Syllable](#).
  2. Syllables: [read-aloud Tem](#) from Tchagbale corpus for Tem<Gur<Niger-Congo (ISO 639-3 kdh, Togo), [tabular CSV format \(labelTABstartTABend\)](#), default tier: [Syllable](#).
  3. Syllabic lexical tone: [read-aloud Tem](#) from Tchagbale corpus for Tem<Gur<Niger-Congo (ISO 639-3 kdh, Togo), [Praat long TextGrid format](#), default tier: [Tone](#) (includes box-and-whisker plot for different label types).
4. English ([Lancaster SEC/MARSEC/Aix-MARSEC](#) corpus, Praat short TextGrid format, default tier [Syllables](#))
  1. [marsecA](#), Category A: Commentary:9066 words
  2. [marsecB](#), Category B: News Broadcasts: 5235 words
  3. [marsecC](#), Category C: Lecture Type 1: 4471 words
  4. [marsecD](#), Category D: Lecture Type 11: 7451 words
  5. [marsecE](#), Category E: Religious Broadcast: 1503 words
  6. [marsecF](#), Category F: Magazine-style reporting: 4710 words
  7. [marsecG](#), Category G: Fiction: 7299 words
  8. [marsecH](#), Category H: Poetry: 1292 words
  9. [marsecI](#), Category I: Dialogue: 6826 words

# TGA: Time Group Analyzer

Speech Annotation Data Mining

Dafydd Gibbon (Universität Bielefeld)

Demo of spoken Farsi, from SM corpus

Reset input TextGrid example

CALCULATE TGA OUTPUT

(Calculation can take time, depending on server load.)

## TGA INPUT, PROCESSING, OUTPUT PARAMETERS

### Parameters (1): TextGrid input control

Metadata:	<input type="text" value="You can put your metadata here."/>
Tier name:	<input type="text" value="Syllables"/> (change if necessary; max length 20; not needed for CSV formats) <b>The current version of the TGA online tool analyses sequential and hierarchical temporal relations on single tiers only, not temporal overlap relations between tiers.</b>
Boundary (e.g. pause) symbol:	<input type="text" value="_"/> (typical examples; max length 20; also needed for CSV formats) More than one pause symbol is permitted; separate symbols with spaces. Delete/change as necessary. If your pause symbol is not in the examples given, enter it. <b>Do NOT use spaces or empty labels as pause markers. Items with these are deleted in order to permit the analysis of sparse, opportunistic, 'agile' annotations.</b> <b>Note that the last label on the selected tier in your annotation <u>must</u> be a pause symbol.</b>

### Parameters (2): Time Group duration difference criteria:

TG type:	<input checked="" type="radio"/> <i>interpausal group</i> <input type="radio"/> <i>deceleration</i> (increasing duration) <input type="radio"/> <i>acceleration</i> (decreasing duration)
Local threshold:	<input type="text" value="10"/> ms (minimal duration distance recognized, e.g. 0 ... 300 ms for syllables) Used for local Duration Difference Token extraction and Time Tree parsing.
DDT symbols:	Longer: <input type="text" value="\"/> (1 char) Shorter: <input type="text" value="/"/> (1 char) Same: <input type="text" value="="/> (1 char) (Symbols for local threshold dependent difference Duration Tokens)
TT type:	<input type="radio"/> <i>(quasi-)iambic TTgt</i> <input type="radio"/> <i>(quasi-)trochaic TTlt</i> <input checked="" type="radio"/> <i>show all TT</i> <input type="radio"/> <i>(quasi-)iambic TTgte</i> <input type="radio"/> <i>(quasi-)trochaic TTlte</i> <input type="radio"/> <i>do not show TT</i>

Reset input TextGrid example

CALCULATE TGA OUTPUT

(Calculation can take time, depending on server load.)

## TGA INPUT FIELD

### Note:

1. Test with the preset TextGrid example, then delete, copy and paste your own TextGrid or CSV):
2. Input timestamp values in all formats are in seconds with '.' (dot) decimal point (following Praat TextGrid conventions), and neither milliseconds nor seconds with ',' (comma) decimal point. Output timestamp values are in milliseconds.

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 6.637428225932263
tiers? <exists>
size = 2
item []:
  item [1]:
    class = "IntervalTier"
    name = "Syllables"
    xmin = 0
    xmax = 6.637428225932263
    intervals: size = 33
    intervals [1]:
      xmin = 0
      xmax = 0.10080428396194088
      text = " "
    intervals [2]:
      xmin = 0.10080428396194088
      xmax = 0.3043804625479265
      text = "our"
    intervals [3]:
      xmin = 0.3043804625479265
      xmax = 0.5592462322036338
      text = "foot"
    intervals [4]:
      xmin = 0.5592462322036338
      xmax = 0.8237319715738414
      text = "ball"
    intervals [5]:
      xmin = 0.8237319715738414
      xmax = 0.8237319715738414
      text = " "
```

# Time Group Analyzer 5.00

Dafydd Gibbon, Bielefeld

Version 5.00: 2016-02-09  
 Generated on 2025-07-21 at 12:55:33 with tga-5.00-cgi.py

Metadata: You can put your metadata here.

Heuristic check of input data structure.  
 This is a pre-alpha version and is not foolproof. Remember: you put garbage in, you get garbage out!  
 After a rough check, assuming full TextGrid input format.

After a rough check, tier name seems ok. No guarantees.

Pauses look ok; if alphanumeric, check that there is no clash with a label if errors occur.  
 [('\_', 1869.9200073048842, 1983.2972461341162), (\_, 3999.273681056707, 4212.331637771596)]

## TGA "pausegroup" analysis of tier "Syllables"

[TG Summary](#) - [Label text](#) - [Graphs](#) - [Time Groups](#) - [DDT table](#) - [Overall stats](#) - [CSV output](#)

### TIME GROUP SUMMARY

Summary of general sequence properties (times in ms, initial and terminal pauses obligatory but ignored in calculation; see below for more details)

Duration properties (without pauses)				Duration properties (pauses)				Duration properties (all labels inc. pauses)			
Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values
n:	29	intercept:	246.167	n:	2	intercept:	113.377	n:	31	intercept:	237.913
min:	66.81071495517244	slope:	-2.724	min:	113.37723882923206	slope:	99.681	min:	66.81071495517244	slope:	-2.185
max:	323.26034811914246	std:	71.093	max:	213.05795671488931	std:	49.84	max:	323.26034811914246	std:	70.778
mean:	208.03	coeff var (%):	34.174	mean:	163.22	coeff var (%):	30.536	mean:	205.14	coeff var (%):	34.502
median:	203.58	nPVI:	39	median:	163.22	nPVI:	61	median:	203.58	nPVI:	38
mean rate:	4.81	rPVI:	76	mean rate:	6.13	rPVI:	100	mean rate:	4.87	rPVI:	75
median rate:	4.91	100*rPVI/med:	37	median rate:	6.13	100*rPVI/med:	61	median rate:	4.91	100*rPVI/med:	37
total:	6032.977471090645	nPVI*med/100:	79	total:	326.43519554412137	nPVI*med/100:	100	total:	6359.4126666347665	nPVI*med/100:	77
range:	256.44963316397			range:	99.68071788565726			range:	256.44963316397		

# Time Group Analyzer 5.00

Dafydd Gibbon, Bielefeld

Version 5.00: 2016-02-09

Generated on 2025-07-21 at 12:55:33 with tga-5.00-cgi.py

Metadata: You can put your metadata here.

### Heuristic check of input data structure.

This is a pre-alpha version and is not foolproof. Remember: you put garbage in, you get garbage out! After a rough check, assuming full TextGrid input format.

After a rough check, tier name seems ok. No guarantees.

Pauses look ok; if alphanumeric, check that there is no clash with a label if errors occur.  
[('\_', 1869.9200073048842, 1983.2972461341162), (\_, 3999.273681056707, 4212.331637771596)]

## TGA "pausegroup" analysis of tier "Syllables"

[TG Summary](#) - [Label text](#) - [Graphs](#) - [Time Groups](#) - [DDT table](#) - [Overall stats](#) - [CSV output](#)

### TIME GROUP SUMMARY

Summary of general sequence properties (times in ms, initial and terminal pauses obligatory but ignored in calculation; see below for more details)

Duration properties (without pauses)				Duration properties (pauses)				Duration properties (all labels inc. pauses)			
Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values
n:	29	intercept:	246.167	n:	2	intercept:	113.377	n:	31	intercept:	237.913
min:	66.81071495517244	slope:	-2.724	min:	113.37723882923206	slope:	99.681	min:	66.81071495517244	slope:	-2.185
max:	323.26034811914246	std:	71.093	max:	213.05795671488931	std:	49.84	max:	323.26034811914246	std:	70.778
mean:	208.03	coeff var (%):	34.174	mean:	163.22	coeff var (%):	30.536	mean:	205.14	coeff var (%):	34.502
median:	203.58	nPVI:	39	median:	163.22	nPVI:	61	median:	203.58	nPVI:	38
mean rate:	4.81	rPVI:	76	mean rate:	6.13	rPVI:	100	mean rate:	4.87	rPVI:	75
median rate:	4.91	100*rPVI/med:	37	median rate:	6.13	100*rPVI/med:	61	median rate:	4.91	100*rPVI/med:	37
total:	6032.977471090645	nPVI*med/100:	79	total:	326.43519554412137	nPVI*med/100:	100	total:	6359.4126666347665	nPVI*med/100:	77
range:	256.44963316397			range:	99.68071788565726			range:	256.44963316397		

**Time Group Analyzer 5.00**  
 Dafydd Gibbon, Bielefeld

Version 5.00: 2016-02-09  
 Generated on 2025-07-21 at 12:55:33 w

Metadata: You can put your metadata

Heuristic check of input data structure  
 This is a pre-alpha version and is not foo  
 After a rough check, assuming full TextG

$$nPVI = \frac{100}{m - 1} \sum_{k=1}^{m-1} \left| \frac{d_{k+1} - d_k}{\frac{d_{k+1} + d_k}{2}} \right|$$

After a rough check, tier name seems ok. No guarantees.

Pauses look ok; if alphanumeric, check that there is no clash with a label if errors occur.  
 [('\_', 1869.9200073048842, 1983.2972461341162), (\_, 3999.273681056707, 4212.331637771596)]

**TGA "pausegroup" analysis of tier "Syllables"**

[TG Summary](#) - [Label text](#) - [Graphs](#) - [Time Groups](#) - [DDT table](#) - [Overall stats](#) - [CSV output](#)

**TIME GROUP SUMMARY**

Summary of general sequence properties (times in ms, initial and terminal pauses obligatory but ignored in calculation; see below for more details)

Duration properties (without pauses)				Duration properties (pauses)				Duration properties (all labels inc. pauses)			
Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values
n:	29	intercept:	246.167	n:	2	intercept:	113.377	n:	31	intercept:	237.913
min:	66.81071495517244	slope:	-2.724	min:	113.37723882923206	slope:	99.681	min:	66.81071495517244	slope:	-2.185
max:	323.26034811914246	std:	71.093	max:	213.05795671488931	std:	49.84	max:	323.26034811914246	std:	70.778
mean:	208.03	coeff var (%):	34.174	mean:	163.22	coeff var (%):	30.536	mean:	205.14	coeff var (%):	34.502
median:	203.58	nPVI:	39	median:	163.22	nPVI:	61	median:	203.58	nPVI:	38
mean	4.81	rPVI:	76	mean	6.13	rPVI:	100	mean	4.87	rPVI:	75
rate:		100*rPVI/med:	37	rate:		100*rPVI/med:	61	rate:		100*rPVI/med:	37
median	4.91	nPVI*med/100:	79	median	6.13	nPVI*med/100:	100	median	4.91	nPVI*med/100:	77
rate:				rate:				rate:			
total:	6032.977471090645			total:	326.43519554412137			total:	6359.4126666347665		
range:	256.44963316397			range:	99.68071788565726			range:	256.44963316397		

**Time Group Analyzer 5.00**  
 Dafydd Gibbon, Bielefeld

Version 5.00: 2016-02-09  
 Generated on 2025-07-21 at 12:55:33 with

**Metadata:** You can put your metadata

**Heuristic check of input data structure**  
 This is a pre-alpha version and is not foolproof  
 After a rough check, assuming full TextGrid

$$nPVI = \frac{100}{m - 1} \sum_{k=1}^{m-1} \left| \frac{d_{k+1} - d_k}{\frac{d_{k+1} + d_k}{2}} \right|$$

After a rough check, tier name seems ok. No guarantees.

Pauses look ok; if alphanumeric, check the  
 [('\_', 1869.9200073048842, 1983.297246

**TGA "pausegroup" analysis**

$$nPVI = \frac{200}{n - 1} \sum_{k=0}^{n-2} \frac{|d_{k+1} - d_k|}{d_{k+1} + d_k}$$

- CSV output

**TIME GROUP SUMMARY**

Summary of general sequence properties (times in ms, initial and terminal pauses obligatory but ignored in calculation; see below for more details)

Duration properties (without pauses)				Duration properties (pauses)				Duration properties (all labels inc. pauses)			
Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values	Attributes	Values
n:	29	intercept:	246.167	n:	2	intercept:	113.377	n:	31	intercept:	237.913
min:	66.81071495517244	slope:	-2.724	min:	113.37723882923206	slope:	99.681	min:	66.81071495517244	slope:	-2.185
max:	323.26034811914246	std:	71.093	max:	213.05795671488931	std:	49.84	max:	323.26034811914246	std:	70.778
mean:	208.03	coeff var (%):	34.174	mean:	163.22	coeff var (%):	30.536	mean:	205.14	coeff var (%):	34.502
median:	203.58	nPVI:	39	median:	163.22	nPVI:	61	median:	203.58	nPVI:	38
mean	4.81	rPVI:	76	mean	6.13	rPVI:	100	mean	4.87	rPVI:	75
rate:		100*rPVI/med:	37	rate:		100*rPVI/med:	61	rate:		100*rPVI/med:	37
median	4.91	nPVI*med/100:	79	median	6.13	nPVI*med/100:	100	median	4.91	nPVI*med/100:	77
rate:				rate:				rate:			
total:	6032.977471090645			total:	326.43519554412137			total:	6359.4126666347665		
range:	256.44963316397			range:	99.68071788565726			range:	256.44963316397		

## Time Group Analyzer 5.00

Dafydd Gibbon, Bielefeld

Version 5.00: 2016-02-09

Generated on 2025-07-21 at 12:55:33 with

Metadata: You can put your metadata

Heuristic check of input data structure

This is a pre-alpha version and is not foolproof.  
After a rough check, assuming full TextG

$$\text{nPVI} = \frac{100}{m-1} \sum_{k=1}^{m-1} \left| \frac{d_{k+1} - d_k}{\frac{d_{k+1} + d_k}{2}} \right|$$

After a rough check, tier name seems ok. No guarantees.

Pauses look ok; if alphanumeric, check the  
[('\_', 1869.9200073048842, 1983.297246

TGA "pausegroup" analysis

$$\text{nPVI} = \frac{200}{n-1} \sum_{k=0}^{n-2} \frac{|d_{k+1} - d_k|}{d_{k+1} + d_k}$$

- CSV output

Let:

- $D_1 = (d_0, d_1, \dots, d_{n-1})$
- $D_2 = (d_1, d_2, \dots, d_n)$

$$\text{nPVI}(D) = \frac{200}{n} \sum_{k=0}^{n-1} \frac{|D_2[k] - D_1[k]|}{D_2[k] + D_1[k]}$$

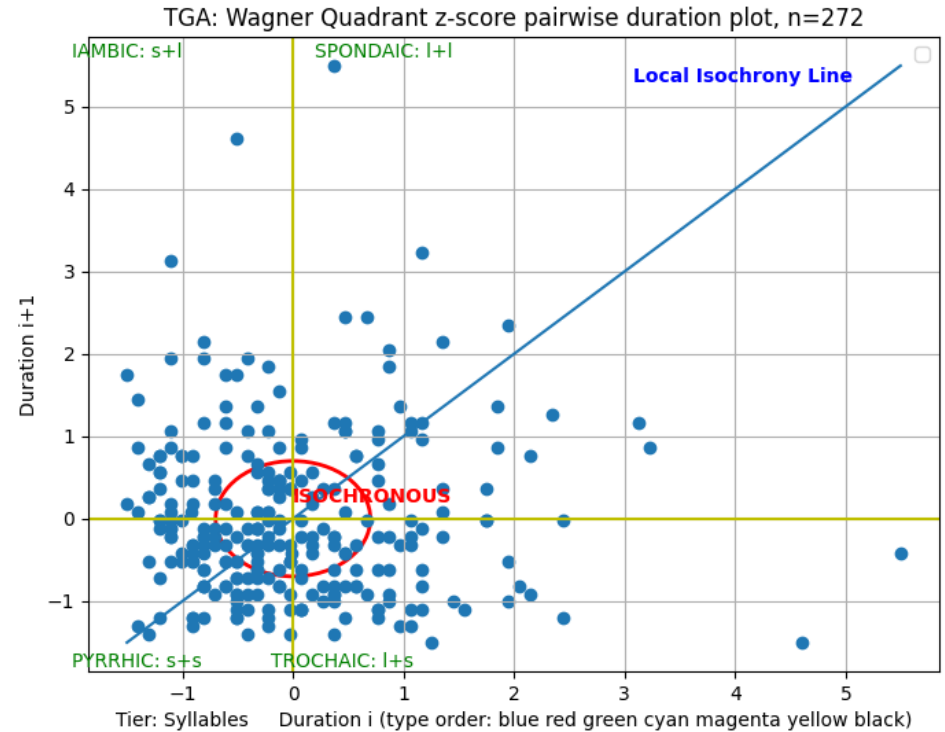
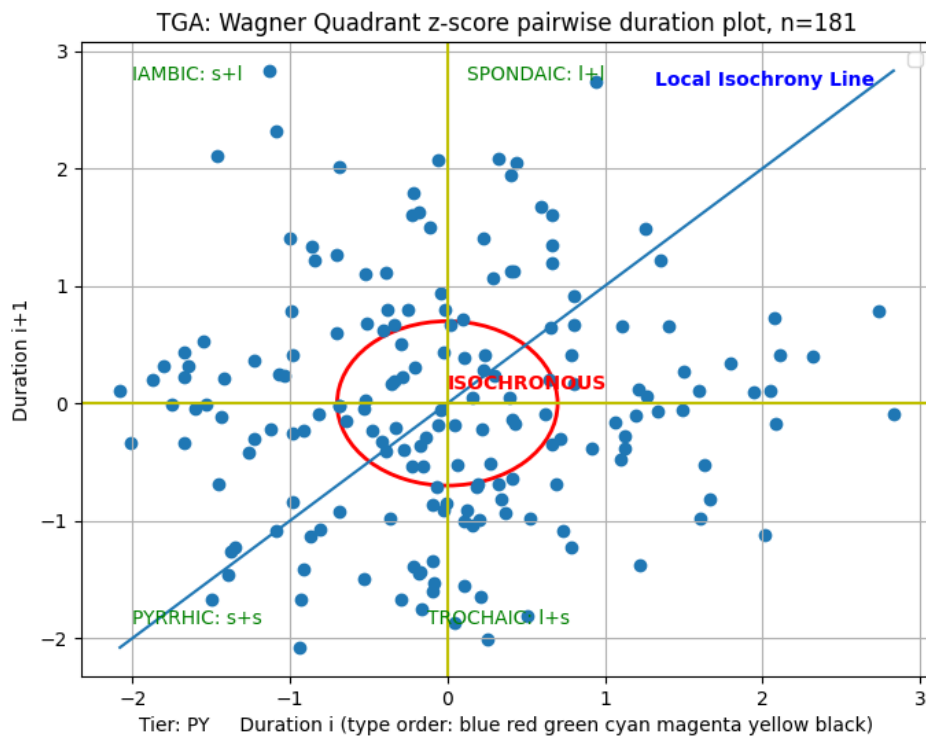
# Extraction of tier transcription

bei3 feng1 gen1 tai4 yang2 p  
you3 yi4 hui2 p  
bei3 feng1 gen1 tai4 yang2 zai4 nar4 zheng1 lun4  
shui2 de5 ben3 shi5 da4 p  
zheng1 lai2 zheng1 qu4 jiu4 shi4 fen1 bu4 chu1 gao1  
di1 lai2 p  
zhe4 shi2 hou5 lu4 shang5 lai2 le5 ge4 zou3 daor4  
de5 p  
ta1 shen1 shang5 chuan1 zhe5 jian4 hou4 da4 yi1 p  
ta1 men5 lia3 jiu4 shuo1 hao3 le5 p  
shui2 neng2 xian1 jiao4 zhe4 ge5 zou3 daor4 de5 tuo1  
xia4 ta1 de5 hou4 da4 yi1 p  
jiu4 suan4 shui2 de5 ben3 shi5 da4 p  
bei3 feng1 jiu4 shi3 jinr4 de5 gua1 qi3 lai2 le5 p  
bu2 guo4 p  
ta1 yue4 shi4 gua1 de5 li4 hai5 p  
na4 ge5 zou3 daor4 de5 p  
ba3 da4 yi1 guo3 de5 yue4 jin3 p  
hou4 lai2 bei3 feng1 mei2 far3 le5 p  
zhi3 hao3 jiu4 suan4 le5 p  
guo4 le5 yi2 huir4 p  
tai4 yang2 chu1 lai5 le5 p  
ta1 huo3 la4 la4 de5 yi2 shai4 p  
na4 ge5 zou3 daor4 de5 ma3 shang4 jiu4 ba3 na4  
jian4 hou4 da4 yi1 tuo1 xia4 lai2 le5 p  
zhe4 xiar4 bei3 feng1 zhi3 hao3 cheng2 ren4 p  
ta1 men5 lia3 dang1 zhong1 hai2 shi5 tai4 yang2 de5  
ben3 shi5 da4

Mandarin:

“Bei3 feng1 gen1 tai4 yang2”

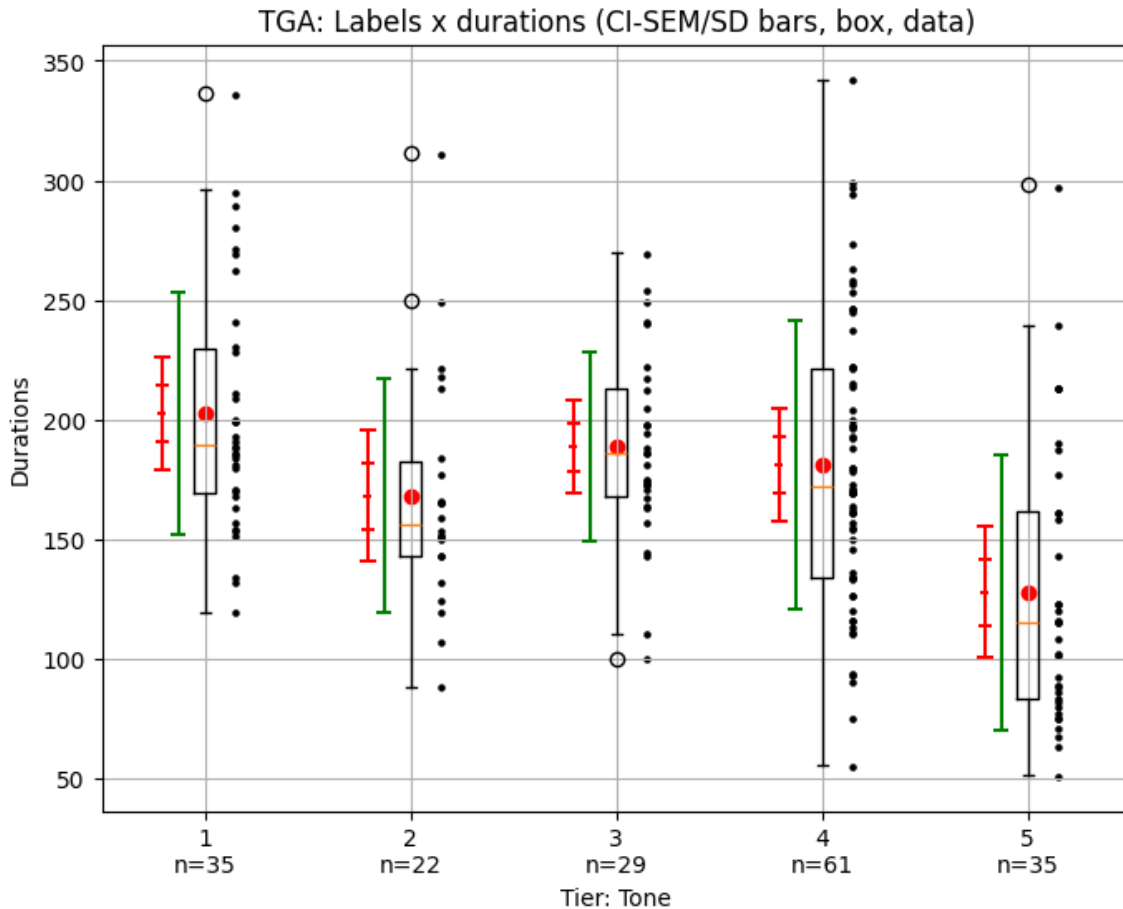
# Comparison of Mandarin and English timing



Story-telling: comparisons of translations of “The North Wind and the Sun” in Mandarin and English.

# Mandarin: Tone x Duration

# Tone distribution



3 1 1 4 2 p  
 3 4 2 p  
 3 1 1 4 2 4 4 1 4 2 5 3 5 4 p  
 1 2 1 4 4 4 1 4 1 1 1 2 p  
 4 2 5 4 5 2 5 4 3 4 5 p  
 1 1 5 1 5 4 4 4 1 p  
 1 5 3 4 1 3 5 p  
 2 2 1 4 4 5 3 4 5 1 4 1 5 4 4 1 p  
 4 4 2 5 3 5 4 p  
 3 1 4 3 4 5 1 3 2 5 p  
 2 4 p  
 1 4 4 1 5 4 5 p  
 4 5 3 4 5 p  
 3 4 1 3 5 4 3 p  
 4 2 3 1 2 3 5 p  
 3 3 4 4 5 p  
 4 5 2 4 p  
 4 2 1 5 5 p  
 1 3 4 4 5 2 4 p  
 4 5 3 4 5 3 4 4 3 4 4 4 4 1 1 4 2 5  
 p  
 4 4 3 1 3 3 2 4 p  
 1 5 3 1 1 2 5 4 2 5 3 5 4

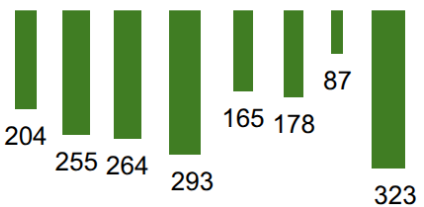
# Duration patterns and rhythm

#	n	dur	rate	mean	med	stdev	nPVI	mdPVI	PIM	PFD	inter	slope	pattern	s = text:dur; <i>italic</i> TGs: no pause)
1	8	1769	4.52	221.12	229.50	72.30	40	22	27	28	237.50	-4.68	<p>\ = \ / \ / \</p> <p>our foot ball team is ve ry good</p>	<p>our:203.57617858598564</p> <p>foot:254.86576965570725</p> <p>ball:264.4857393702076</p> <p>team:293.39508339161773</p> <p>is:164.5689044970179</p> <p>ve:177.60556500202233</p> <p>ry:87.35813472124232</p> <p>good:323.26034811914246</p> <p>_:113.37723882923206 #</p> <p><b>iambicTTgt:</b> (((our foot) (ball team)) ((is ve) (ry good)))</p> <p><b>iambicTTgte:</b> (our foot ball team is ve ry good _)</p> <p><b>trochaicTlit:</b> (((our (foot (ball (team is)))) (ve ry)) (good _))</p> <p><b>trochaicTlte:</b> (our foot ball team is ve ry good _)</p> <p><b>iambicTTgt:</b> (((our foot) (ball team)) ((is ve) (ry good)))</p> <p><b>iambicTTgte:</b> (our foot ball team is ve ry good _)</p> <p><b>trochaicTlit:</b> (((our (foot (ball (team is)))) (ve ry)) (good _))</p> <p><b>trochaicTlte:</b> (our foot ball team is ve ry good _)</p>
2	8	2016	3.97	252.00	262.50	47.75	25	20	14	15	205.42	13.31	<p>\ / \ / \ / \</p> <p>they won two cham pion ships this year</p>	<p>they:160.16080884084772</p> <p>won:263.0163741514839</p> <p>two:243.9146263080802</p> <p>cham:268.8938350263775</p> <p>pion:201.30303496510214</p> <p>ships:318.8596072970936</p> <p>this:261.5470089327605</p> <p>year:298.2811394008454</p> <p>_:213.05795671488931 #</p> <p><b>iambicTTgt:</b> ((they won) ((two cham) (pion ships)))</p> <p><b>iambicTTgte:</b> (they won two cham pion ships this year _)</p> <p><b>trochaicTlit:</b> (they ((won two) (cham pion)) ((ships this) (year _)))</p> <p><b>trochaicTlte:</b> (they won two cham pion ships this year _)</p> <p><b>iambicTTgt:</b> ((they won) ((two cham) (pion ships)))</p>

# Duration patterns and rhythm

#	n	dur	rate	mean	med	stdev	nPVI	ndPVI	PIM	PFD	inter	slope	pattern	s = text:dur; italic TGs: no pause)
1	8	1769	4.52	221.12	229.50	72.30	40	22	27	28	237.50	-4.68	<p>\ = \ / \ / \</p> <p>our foot ball team is ve ry good</p>	<p>our:203.57617858598564</p> <p>foot:254.86576965570725</p> <p>ball:264.4857393702076</p> <p>team:293.39508339161773</p> <p>is:164.5689044970179</p> <p>ve:177.60556500202233</p> <p>ry:87.35813472124232</p> <p>good:323.26034811914246</p> <p>_:113.37723882923206 #</p> <p><b>iambicTTgt:</b> (((our foot) (ball team)) ((is ve) (ry good)))</p> <p><b>iambicTTgte:</b> (our foot ball team is ve ry good _)</p> <p><b>trochaicTlit:</b> (((our (foot (ball (team is)))) (ve ry)) (good _))</p> <p><b>trochaicTlite:</b> (our foot ball team is ve ry good _)</p> <p><b>iambicTTgt:</b> (((our foot) (ball team)) ((is ve) (ry good)))</p> <p><b>iambicTTgte:</b> (our foot ball team is ve ry good _)</p> <p><b>trochaicTlit:</b> (((our (foot (ball (team is)))) (ve ry)) (good _))</p> <p><b>trochaicTlite:</b> (our foot ball team is ve ry good _)</p>
2	8	2016	3.97	252.00	262.50	47.75	25	20	14	15	205.42	13.31	<p>\ / \ / \ / \</p> <p>they won two cham pion ships this year</p>	<p>they:160.16080884084772</p> <p>won:263.0163741514839</p> <p>two:243.9146263080802</p> <p>cham:268.8938350263775</p> <p>pion:201.30303496510214</p> <p>ships:318.8596072970936</p> <p>this:261.5470089327605</p> <p>year:298.2811394008454</p> <p>_:213.05795671488931 #</p> <p><b>iambicTTgt:</b> ((they won) ((two cham) (pion ships)))</p> <p><b>iambicTTgte:</b> (they won two cham pion ships this year _)</p> <p><b>trochaicTlit:</b> (they ((won two) (cham pion)) ((ships this) (year _)))</p> <p><b>trochaicTlite:</b> (they won two cham pion ships this year _)</p> <p><b>iambicTTgt:</b> ((they won) ((two cham) (pion ships)))</p>

# Duration patterns and rhythm

#	n	dur	rate	mean	med	stdev	nPVI	ndPVI	PIM	PFD	inter	slope	pattern	s = text:dur; italic TGs: no pause)
1	8	1769	4.52	221.12	229.50	72.30	40	22	27	28	237.50	-4.68	\ = \ / \ / \ our foot ball team is ve ry good 	our:203.57617858598564 foot:254.86576965570725 ball:264.4857393702076 team:293.39508339161773 is:164.5689044970179 ve:177.60556500202233 ry:87.35813472124232 good:323.26034811914246 _:113.37723882923206 #

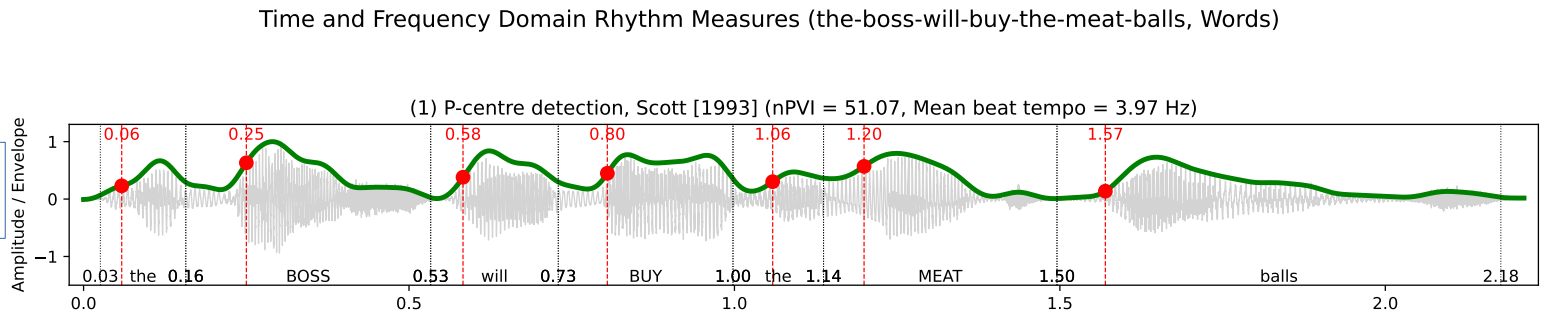
1) When we use Praat annotation, TGA, nPVI, etc., we are trying to cross the **symbol-signal gap** from the phonology and linguistic phonetic side, that is, from the **symbol-phonetic** side.

2) Other approaches try to cross the symbol-signal gap from the **signal phonetic** side (see the following slides).

3) In **AI** speech processing methods, both sides are analysed simultaneously with very big data sets.

# Time domain and frequency domain analysis

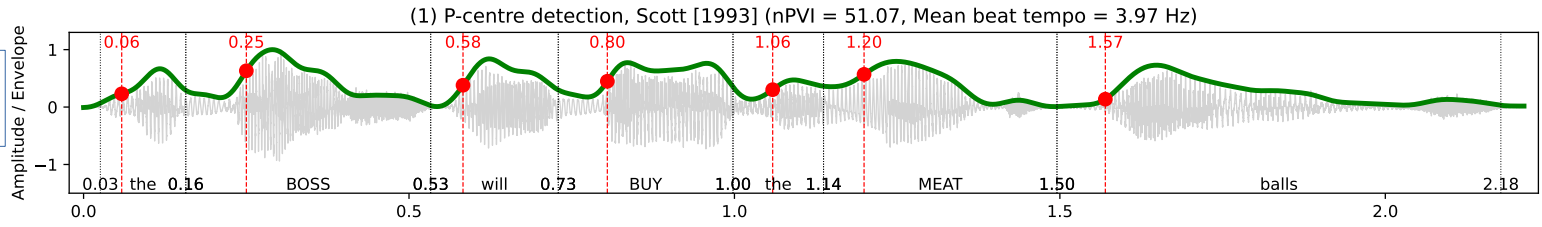
P-centres



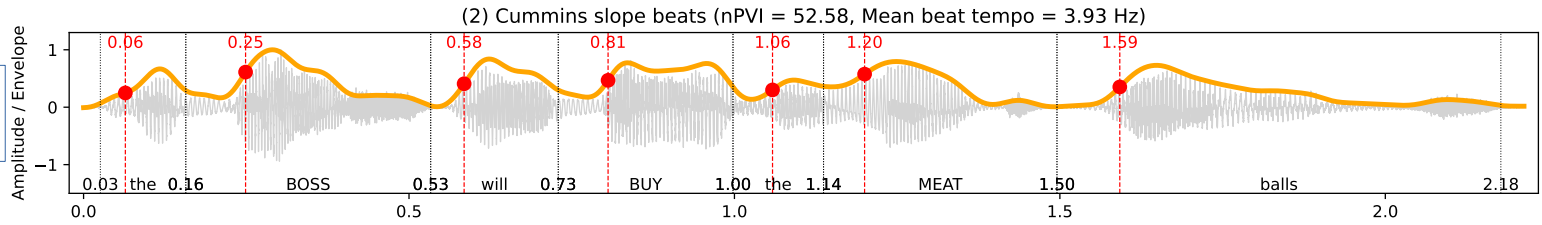
# Time domain and frequency domain analysis

Time and Frequency Domain Rhythm Measures (the-boss-will-buy-the-meat-balls, Words)

P-centres

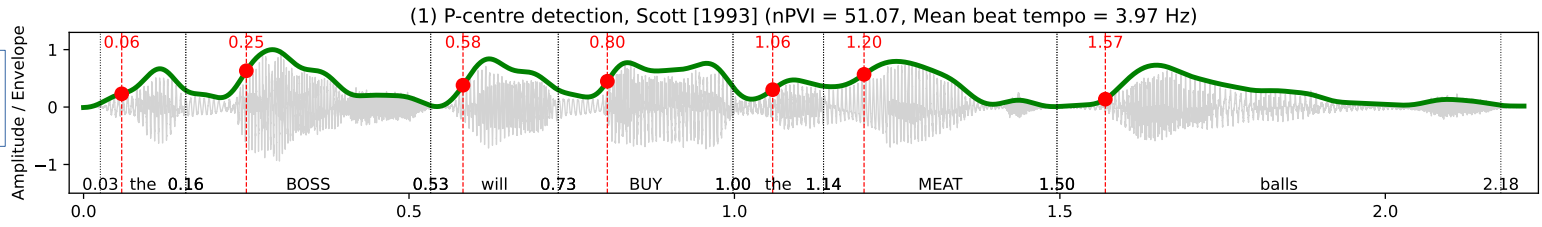


Beat slopes

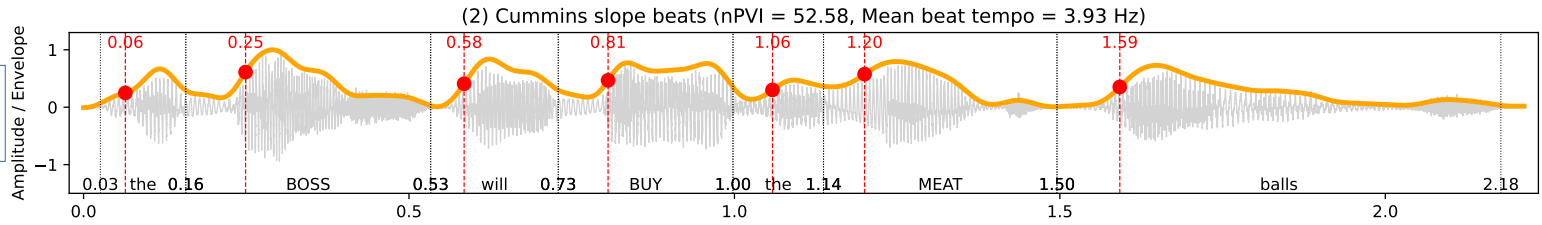


Time domain and frequency domain analysis

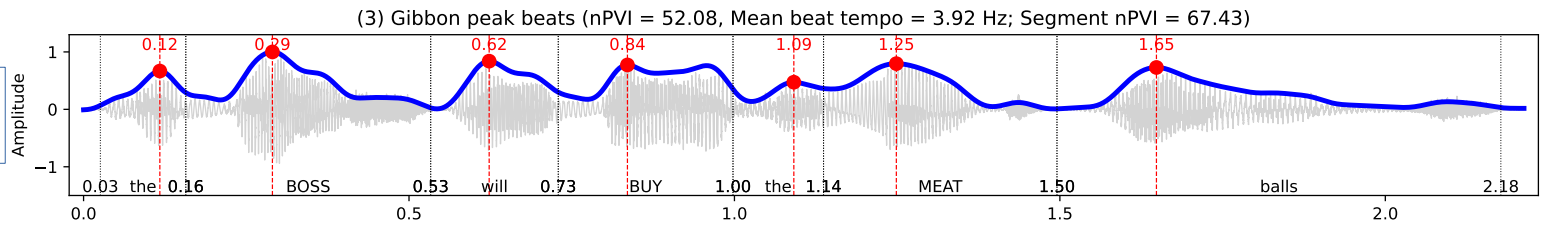
P-centres



Beat slopes

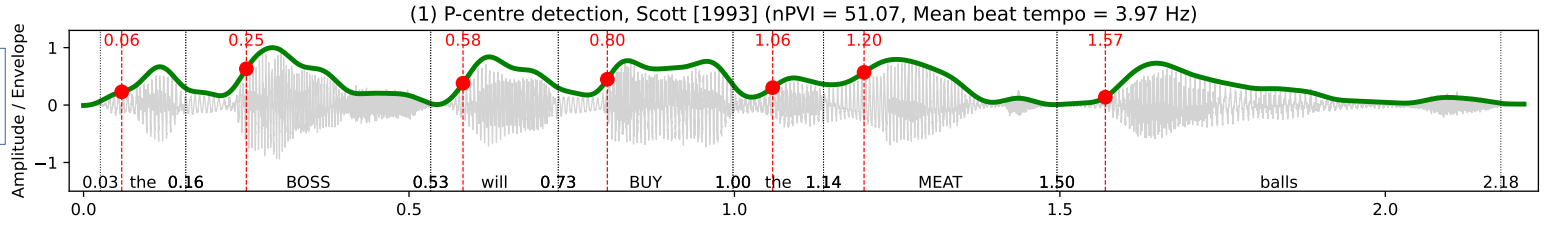


Beat peaks

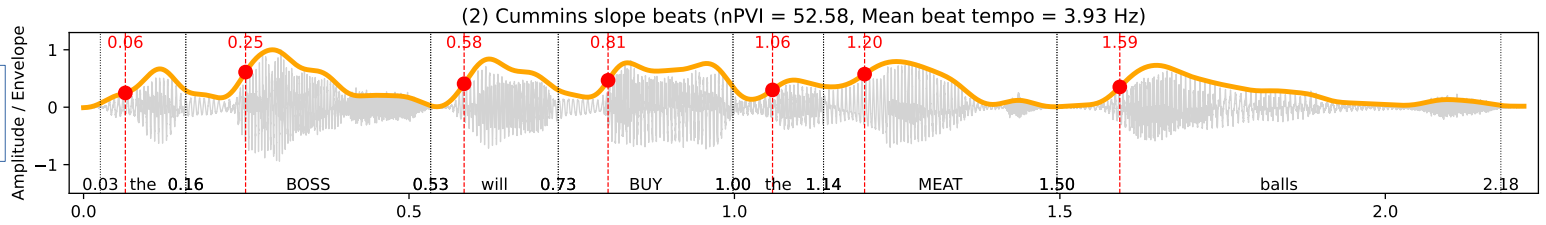


Time domain and frequency domain analysis

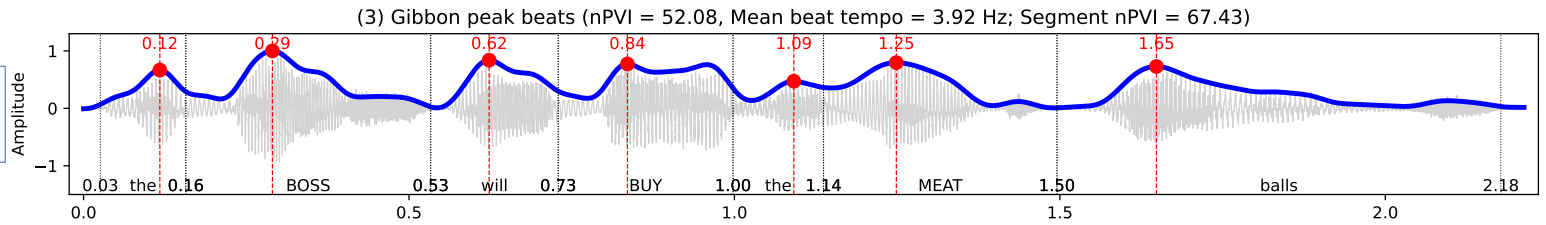
P-centres



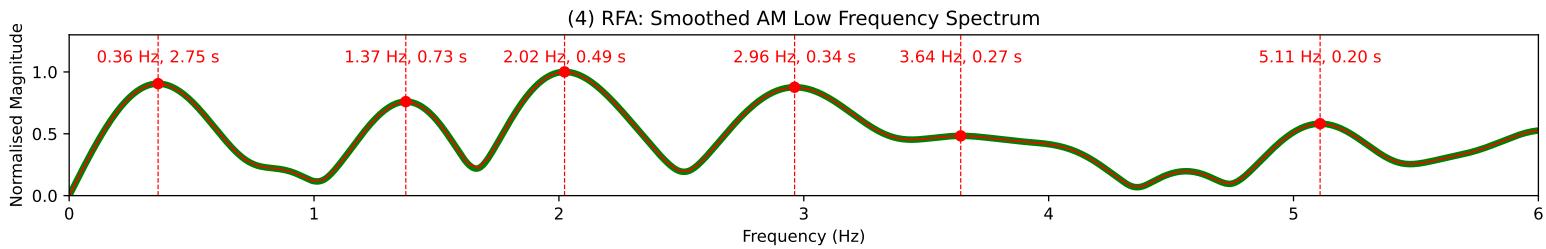
Beat slopes



Beat peaks

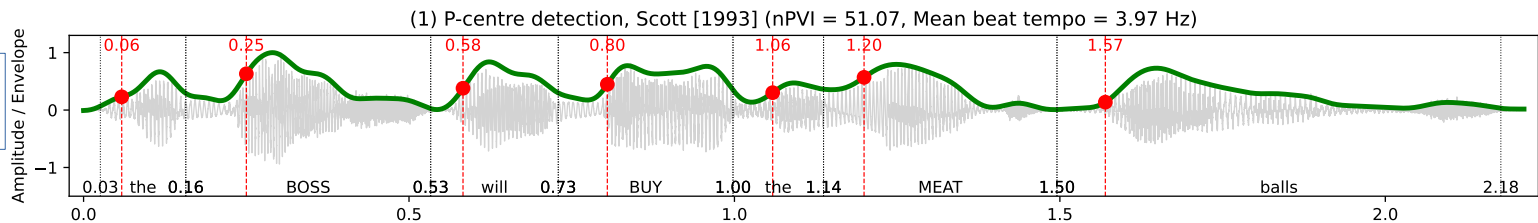


Rhythm frequencies

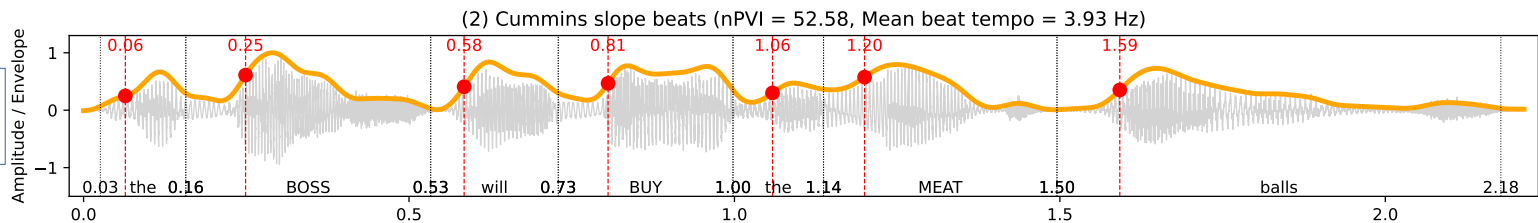


Time domain and frequency domain analysis

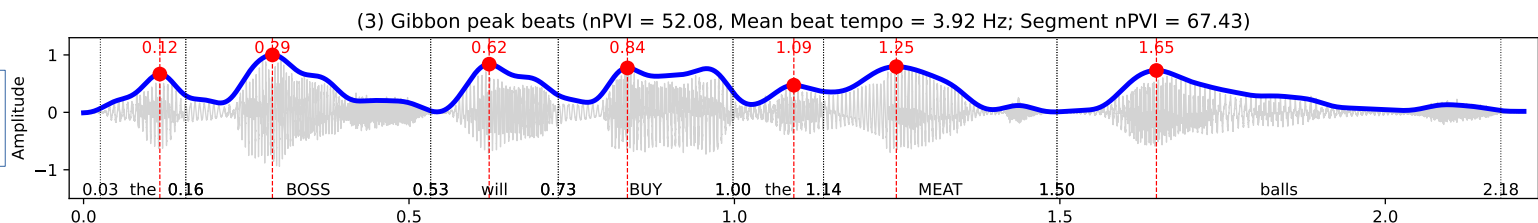
P-centres



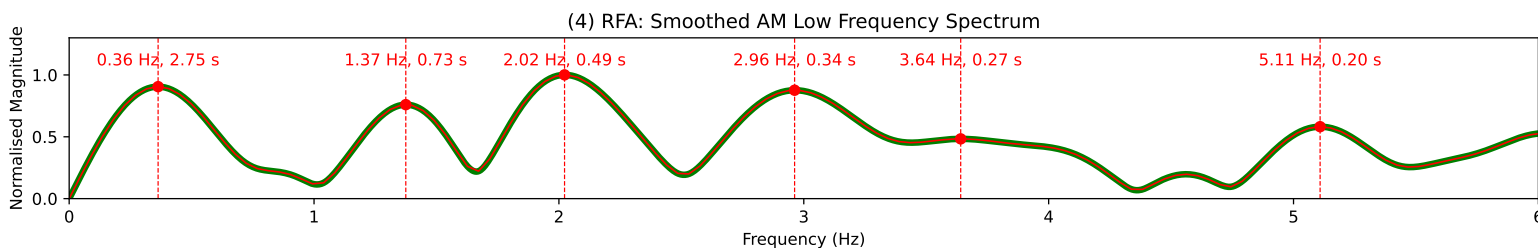
Beat slopes



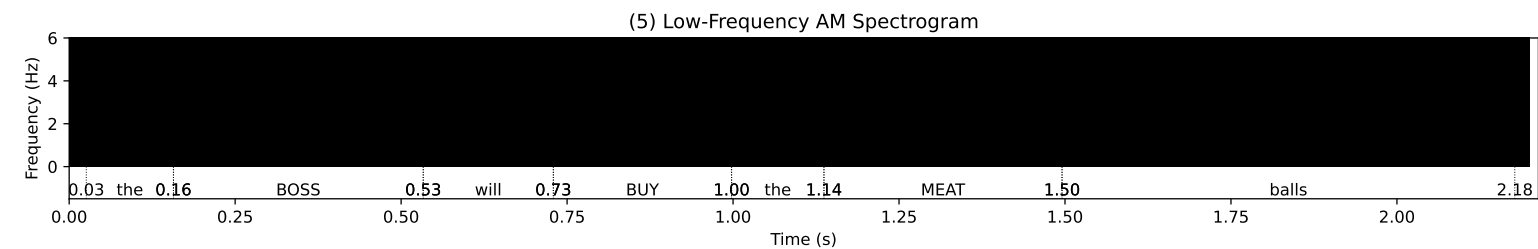
Beat peaks



Rhythm frequencies



Rhythm frequency changes



## CSV OUTPUT

CSV output (Python format): yes

```
["our",0.10080428396194088,0.3043804625479265,0.20357617858598565],  
["foot",0.3043804625479265,0.5592462322036338,0.25486576965570723],  
["ball",0.5592462322036338,0.8237319715738414,0.26448573937020764],  
["team",0.8237319715738414,1.117127054965459,0.202205082201617741],  
["is",1.117127054965459,1.281695959462477,1.4593015244644993],  
["ve",1.281695959462477,1.4593015244644993,1.546659659185741],  
["ry",1.4593015244644993,1.546659659185741,1.869920007304884],  
["good",1.546659659185741,1.869920007304884,1.983297246134116],  
["they",1.983297246134116,2.143458054974964,2.406474429126448],  
["won",2.143458054974964,2.406474429126448,2.650389055434528],  
["two",2.406474429126448,2.650389055434528,2.919282890460905],  
["cham",2.650389055434528,2.919282890460905,3.12058592542600],  
["pion",2.919282890460905,3.12058592542600,3.439445532723101],  
["ships",3.12058592542600,3.439445532723101,3.700992541655862],  
["this",3.439445532723101,3.700992541655862,3.999273681056707],  
["year",3.700992541655862,3.999273681056707,4.212331637771596],  
["_",3.999273681056707,4.212331637771596,4.383236691701131],  
["al",4.212331637771596,4.383236691701131,4.535501461673925],  
["so",4.383236691701131,4.535501461673925,4.76031434013860],  
["moved",4.535501461673925,4.76031434013860,5.052508986054096],  
["up",4.76031434013860,5.052508986054096,5.209940097067955],  
["in",5.052508986054096,5.209940097067955,5.349529792846676],  
["to",5.209940097067955,5.349529792846676,5.469061014774992],  
["the",5.349529792846676,5.469061014774992,5.5358717297301645],  
["se",5.469061014774992,5.5358717297301645,5.780566441144347],  
["cond",5.5358717297301645,5.780566441144347,0.24469471141418217],  
["di",5.780566441144347,5.961298363047323,0.18073192190297596],  
["vi",5.961298363047323,6.064153928357959,0.10285556531063594],  
["sion",6.064153928357959,6.193775279858824,0.12962135150086487],  
["",6.193775279858824,6.460216950596708,0.2664416707378841],
```

# CSV output

CSV database table format  
Character Separated Value  
Comma Separated Value

Copy and paste this into a text file  
to use for additional statistical  
analysis with a spreadsheet or  
statistical application  
(Excel, R, Python, ...)

# Let me tell you a secret ... ASK DEEPSEEK!

(or ChatGPT)

## My prompt:

Please write a Python script which translates a Praat TextGrid file into a CSV file with the following column types for the entries (this is for an interval tier):

tiername,text,start,end,diff

Note that  $\text{diff} = \text{end} - \text{start}$ . Also pay attention to an appropriate row format for point tiers.

**DeepSeek** responded with 19 pages of discussion, then Python code.

**ChatGPTo4** responded with 2 pages, with code and a short explanation.