

# Implementing reproducibility in phonetic research: a computational workflow

Stefano Coretta

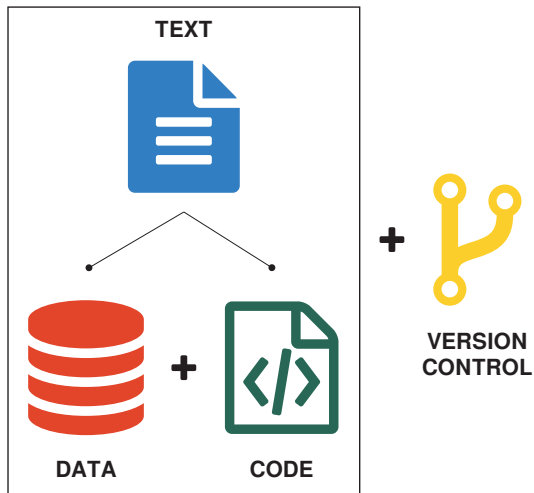
26/03/2017

# Reproducible research

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References



# Why should we care?

The **problem** (Sandve et al. 2013):

- difficulty of reproduction
- difficulty of replication
- retracted papers

The “Yokuts vowels” case (Weigel 2002):

- about **75%** of the data is contrived (Weigel 2005:149)
- some of the generalisations are **wrong** (Blevins 2004)

The **solution**:

- **Reproducible Research (RR)**

# Reproducible Research in linguistics

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References

- linked data (Bird & Simons 2003, Thieberger 2004)
- computational grammar (Maxwell & Amith 2005)
- RR in the Speech Sciences (Abari 2012)
  - lack of scientific culture
  - inefficiency of infrastructure

# The workflow of phonetic research

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References

- **Phase A:** scripting (Praat)
- **Phase B:** results and analysis
- **Phase C:** dissemination

# Phase A: source code and documentation

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References

## Praat scripting:

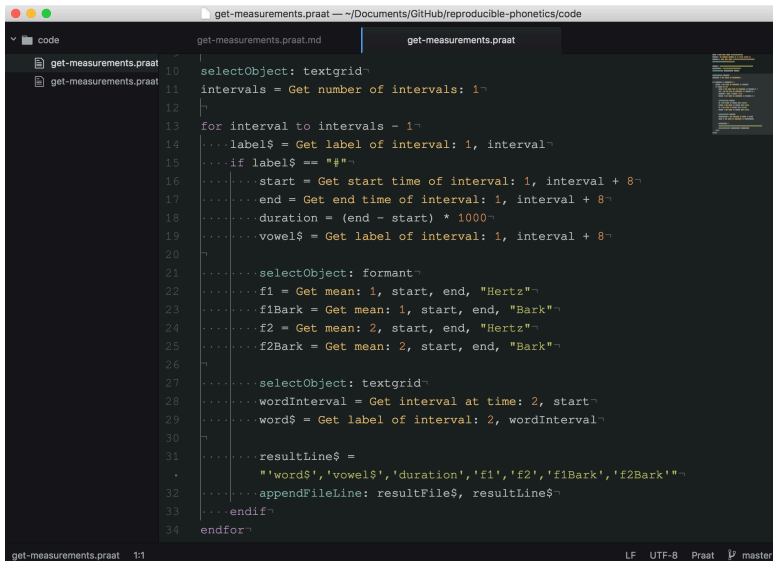
- Atom editor (for syntax highlighting and snippets)
- Literate Markdown
  - tangle: `lmt`
  - weaving: `pandoc`

# Atom

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

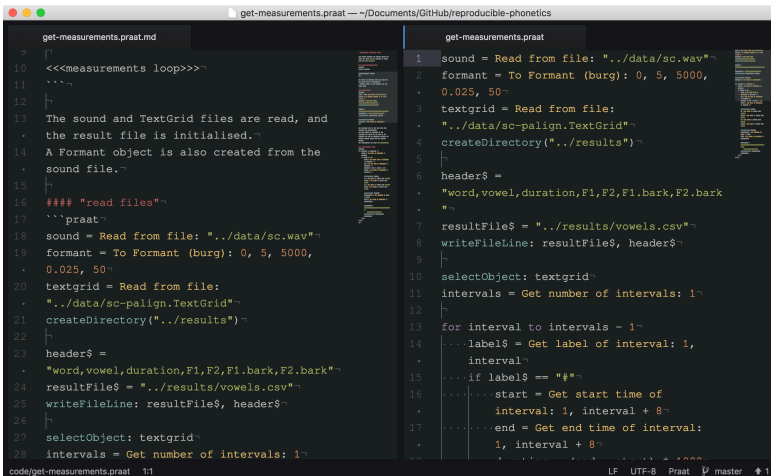
References



The screenshot shows the Atom text editor interface. The title bar indicates the file path: `get-measurements.praat — ~/Documents/GitHub/reproducible-phonetics/code`. The left sidebar shows a file explorer with a folder named `code` containing two files: `get-measurements.praat` and `get-measurements.praat.md`. The main editor area displays the contents of `get-measurements.praat`, which is a Praat script. The script is written in a dark theme with syntax highlighting. It defines a function `selectObject: textgrid` that iterates over intervals of a text grid, calculates the duration of each interval, and extracts formant data (F1 and F2) in both Hertz and Bark. The results are then written to a file. The script is 34 lines long.

```
10 selectObject: textgrid
11 intervals = Get number of intervals: 1
12
13 for interval to intervals - 1
14   ...label$ = Get label of interval: 1, interval
15   ...if label$ == "#"
16     ...start = Get start time of interval: 1, interval + 8
17     ...end = Get end time of interval: 1, interval + 8
18     ...duration = (end - start) * 1000
19     ...vowel$ = Get label of interval: 1, interval + 8
20   ...
21   ...selectObject: formant
22   ...f1 = Get mean: 1, start, end, "Hertz"
23   ...f1Bark = Get mean: 1, start, end, "Bark"
24   ...f2 = Get mean: 2, start, end, "Hertz"
25   ...f2Bark = Get mean: 2, start, end, "Bark"
26   ...
27   ...selectObject: textgrid
28   ...wordInterval = Get interval at time: 2, start
29   ...word$ = Get label of interval: 2, wordInterval
30   ...
31   ...resultLine$ =
32     "word$, 'vowel$', 'duration', 'f1', 'f2', 'f1Bark', 'f2Bark'"
33   ...appendFileLine: resultFile$, resultLine$
34   ...endif
35 endfor
```

The status bar at the bottom shows the file name `get-measurements.praat`, the line number `1:1`, the encoding `LF UTF-8`, the editor `Praat`, and the branch `master`.



```

get-measurements.praat.md
10 <<<measurements loop>>>~
11 ~~~~
12 |
13 The sound and TextGrid files are read, and
  · the result file is initialised.~
14 A Formant object is also created from the
  · sound file.~
15 |
16 ### "read files"~
17 ~~~praat~
18 sound = Read from file: "../data/sc.wav"~
19 formant = To Formant (burg): 0, 5, 5000,
  · 0.025, 50~
20 textgrid = Read from file:
  · "../data/sc-align.TextGrid"~
21 createDirectory("../results")~
22 |
23 header$ =
  · "word,vowel,duration,F1,F2,F1.bark,F2.bark"~
24 resultFile$ = "../results/vowels.csv"~
25 writeFileLine: resultFile$, header$~
26 |
27 selectObject: textgrid~
28 intervals = Get number of intervals: 1~

get-measurements.praat
1 sound = Read from file: "../data/sc.wav"~
2 formant = To Formant (burg): 0, 5, 5000,
  · 0.025, 50~
3 textgrid = Read from file:
  · "../data/sc-align.TextGrid"~
4 createDirectory("../results")~
5 |
6 header$ =
  · "word,vowel,duration,F1,F2,F1.bark,F2.bark"~
  · ""~
7 resultFile$ = "../results/vowels.csv"~
8 writeFileLine: resultFile$, header$~
9 |
10 selectObject: textgrid~
11 intervals = Get number of intervals: 1~
12 |
13 for interval to intervals - 1~
14 |...label$ = Get label of interval: 1,
  · interval~
15 |...if label$ == ""~
16 |...start = Get start time of
  · interval: 1, interval + 8~
17 |...end = Get end time of interval:
  · 1, interval + 8~
  
```

code/get-measurements.praat 1/1

LF UTF-8 Praat master ↕ 1

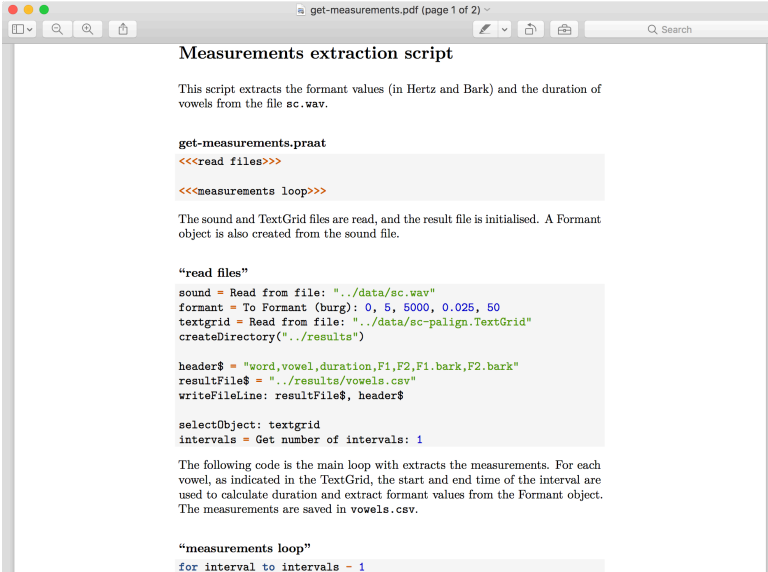


# pandoc

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References



The screenshot shows a PDF viewer window titled "get-measurements.pdf (page 1 of 2)". The document content is as follows:

## Measurements extraction script

This script extracts the formant values (in Hertz and Bark) and the duration of vowels from the file `sc.wav`.

```
get-measurements.praat
<<<read files>>>

<<<measurements loop>>>
```

The sound and TextGrid files are read, and the result file is initialised. A Formant object is also created from the sound file.

“read files”

```
sound = Read from file: "../data/sc.wav"
formant = To Formant (burg): 0, 5, 5000, 0.025, 50
textgrid = Read from file: "../data/sc-align.TextGrid"
createDirectory("../results")

header$ = "word,vowel,duration,F1,F2,F1.bark,F2.bark"
resultFile$ = "../results/vowels.csv"
writeFileLine: resultFile$, header$

selectObject: textgrid
intervals = Get number of intervals: 1
```

The following code is the main loop with extracts the measurements. For each vowel, as indicated in the TextGrid, the start and end time of the interval are used to calculate duration and extract formant values from the Formant object. The measurements are saved in `vowels.csv`.

“measurements loop”

```
for interval to intervals - 1
```

# Phase B: the speakr package

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References

speakr is an R package to aid Praat users:

- aim: tangle and run Praat scripts from within R
- two main functions
  - `lmt()`: tangle a Praat script
  - `praatRun()`: run a Praat script

# Phase B: the speakr package

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References

```
# Tangle a Praat script
```

```
lmt("code/get-measurements.praat.md")
```

```
# Run the script
```

```
praatRun("code/get-measurements.praat")
```

```
# Read the results of the script
```

```
vowels <- read_csv("results/vowels.csv") %>%
```

```
  mutate_if(is.character, as.factor) %>%
```

```
  mutate(vowel = factor(vowel, c("i", "e", "a",  
                                "o", "u")))
```

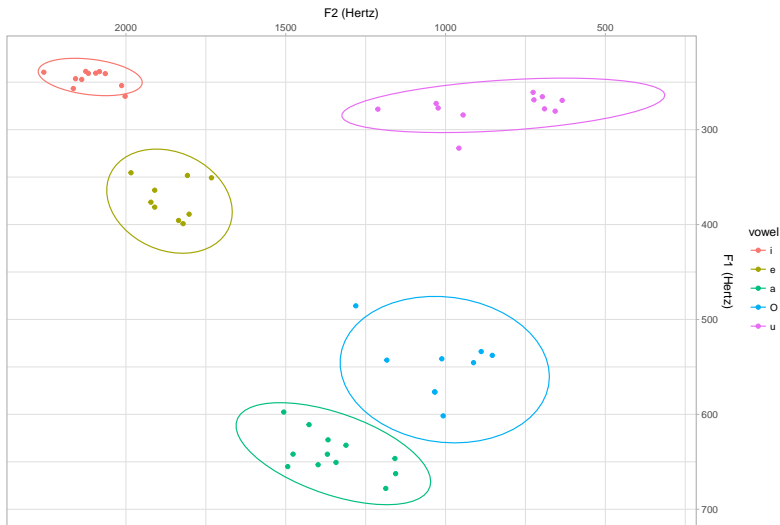
# Phase B: the speakr package

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References

Vowel plot of one speaker of Italian



# Phase C: dissemination

Implementing  
reproducibility  
in phonetic  
research: a  
computational  
workflow

Stefano  
Coretta

References

- GitHub
- Open Science Framework

- Abari, Kálmán. 2012. Reproducible research in speech sciences. *International Journal of Computer Science Issues* 9(6). 43–52.
- Bird, Steven & Gary Simons. 2003. Seven dimensions of portability for language documentation and description. *Language* 557–582.
- Blevins, Juliette. 2004. A reconsideration of Yokuts vowels. *International Journal of American Linguistics* 70(1). 33–51.
- Maxwell, Michael & Jonathan D. Amith. 2005. Language documentation: the Nahuatl grammar. In A. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing*, 474–485. Berlin Heidelberg: Springer-Verlag.

- Sandve, Geir Kjetil, Anton Nekrutenko, James Taylor & Eivind Hovig. 2013. Ten simple rules for reproducible computational research. *PLoS Computational Biology* 9(10). 1–4.
- Thieberger, Nicholas. 2004. Documentation in practice: Developing a linked media corpus of South Efate. In Peter K. Austin (ed.), *Language documenta and description*, vol. 2, Hans Rausing Endangered Languages Project, School of Oriental and African Studies, University of London.
- Weigel, William. 2005. *Yowlumne in the Twentieth century*: University of California, Berkley dissertation.
- Weigel, William F. 2002. The Yokuts canon: A case study in the interaction of theory and description. Paper presented at the annual meeting of the Linguistics Society of America, January 2002, San Francisco.