Implementing reproducibility in phonetic research: a computational workflow

Stefano Coretta

University of Manchester

mFiL 2017
28 April 2017
A piece of research is **reproducible** when, along with its *results*, the *data* and the *computational environment* that produced those results are made available to other researchers (Fomel & Claerbout 2009).
Reproducible research

source

version control

text + code

data
Why should we care?

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

- difficulty of reproduction
- difficulty of replication
- retracted papers ([http://retractionwatch.com](http://retractionwatch.com))

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

- 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

- **Phase A**: scripting (Praat, Boersma & Weenink (2016))
- **Phase B**: results and analysis
- **Phase C**: dissemination
Phase A: source code and documentation

Praat scripting:

- Atom editor (https://atom.io)
  - syntax highlighting
  - autocompletion and snippets
- Literate Markdown
  - tangle: lmt (https://github.com/driusan/lmt)
  - weaving: pandoc (http://pandoc.org)
```plaintext
selectObject: textgrid
intervals = Get number of intervals: 1

for interval to intervals - 1
    label$ = Get label of interval: 1, interval
    if label$ == "#"
        start = Get start time of interval: 1, interval + 8
        end = Get end time of interval: 1, interval + 8
        duration = (end - start) * 1000
        vowel$ = Get label of interval: 1, interval + 8

        selectObject: formant
        f1 = Get mean: 1, start, end, "Hertz"
        f1Bark = Get mean: 1, start, end, "Bark"
        f2 = Get mean: 2, start, end, "Hertz"
        f2Bark = Get mean: 2, start, end, "Bark"

        selectObject: textgrid
        wordInterval = Get interval at time: 2, start
        word$ = Get label of interval: 2, wordInterval

        resultLine$ = "'word$', 'vowel$', 'duration', 'f1', 'f2', 'f1Bark', 'f2Bark'"
        appendFileLine: resultFile$, resultLine$
    endif
endfor
```
The sound and TextGrid files are read, and
the result file is initialised.
A Formant object is also created from the
sound file.

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

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

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

```plaintext
for interval to intervals - 1
    label$ = Get label of interval: 1,
    interval
    if label$ == ")"
    start = Get start time of
            interval: 1, interval + 8
    end = Get end time of interval:
           1, interval + 8
    ```
Measurements extraction script

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

```r
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”

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

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

```r
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”

```r
for interval to intervals - 1
```
Phase B: the `speakr` package

`speakr` is an R (R Core Team 2015) package to aid Praat users (under development):

- 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

```r
# 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
Phase C: dissemination

There is no investigation without dissemination.

Ricardo Bermúdez-Otero (p.c.)

- knitr (Xie 2014)
  - dynamic reports
  - reproducible documents
- GitHub (https://github.com)
  - versioning system (git)
  - online repository
- Open Science Framework (https://osf.io)
  - online repository (for data)
Summary

- share data, source file(s), versioning
- increasing awareness of RR in linguistics
- Atom, lmt, pandoc, speakr, knitr
- this presentation (along with source code and data) is available at https://github.com/stefanocoretta/reproducible-phonetics
THANK YOU!


R Core Team. 2015. R: A language and environment for statistical computing.


