# Reproducible Research in Speech Sciences

**Kálmán Abari**

**Institute of Psychology, University of Debrecen**
**Debrecen, H-4010, Hungary**

## Abstract

Reproducible research is the minimum standard of scientific claims in cases when independent replication proves to be difficult. With the special combination of available software tools, we provide a reproducibility recipe for the experimental research conducted in some fields of speech sciences. We have based our model on the triad of the R environment, the EMU-format speech database, and the executable publication. We present the use of three typesetting systems (LaTeX, Markdown, Org), with the help of a mini research.

***Keywords:*** *Reproducible Research, Speech Sciences, Literate Programming, R, Sweave, Knitr, Org-mode.*

## 1. Introduction

The concept of reproducible research is ambiguous, and can be misleading without appropriate context. In a wider sense, it is the basic standard of judging scientific claims, which promises the independent replication of the complete research. In theoretical sciences (e.g. mathematics), the comprehensive replication of results can be conducted by anybody with the help of the proof beside the theorem. In experimental sciences (e.g. physics, biology, or life sciences), the wording of the hypothesis and the exact description of the experiment make the evaluation of published findings possible. In this case, other researchers may check the validity of the hypothesis by way of replicating the experiment with the help of independently collected data. However, there is not always a possibility for the independent replication of the complete research, in lack of appropriate resources (e.g. data size, computing power), or due to other reasons (some of the research consume much time and money). In a narrower sense, reproducible research means the publication of the data used and of the computer codes, beyond the research results. Other researchers may check the validity of claims on the data published by way of running the code published, thus reproducible research has become a sort of minimum standard in judging scientific claims [16]. The concept of reproducible research in this sense comes from Jon Claerbout and his colleague [3]. In the summary of Buckeit and Donoho [2], Claerbout's idea is as follows: „An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures." In the present article, similarly to the above, we use the narrower sense of the notion of reproducible research, which definitely represents the direction towards the research methodology of the future, in our view.

Speech sciences comprise an exceedingly wide and multidisciplinary field. It has human speech communication in its centre, which has the aim of transferring ideas to other human beings in a voiced form. Speech is examined in the most comprehensive way by phonetics, whose researchers aim to discover and describe the process from cognitive speech planning, through articulation and the physical oscillation transmitted in the air, to speech perception. Phoneticians, who are in close communication with representatives of other fields of science (e.g. psycholinguistics, phoniatry, physiology, physical acoustics, otolaryngology, audiology), study (1) how we create speech from our thoughts with the help of our articulatory organs, (2) in what form the oscillation of the air contains the original thought or the corresponding linguistic denotation, and (3) how the original linguistic content or the speaker's original idea is restored from the acoustic signs. The process outlined above between the speaker and the listener(s) is called the speech chain.

Phonetics is basically an experimental field of science, so the computer at the middle of the last century launched revolutionary new changes in the several-century old history of speech research. It fundamentally transformed the measuring processes and research methods used until that time, and it made possible the most comfortable visualization of the speech signal and its manifold processing. The new discipline called digital speech processing, which, partly based on the results of digital signal processing, aims to analyse and model the components of the natural speech chain in a machine-based way.

However, due to the dynamic development of their performance, computers play an important role not only in basic research but also in the development of speech-based practical applications. By now, we have some potential for the machine-based application of certain elements of the natural speech chain beyond their computer-based analysis. The solutions mechanizing certain elements of the speech chain are already the products of a young field of science, namely speech technology. Speech synthesis substitutes human articulation and phonation, speech coding substitutes the medium of the acoustic signal, and automatic speech recognition replaces speech perception. These are the fundamental fields of speech technology, to which we may add, for example, the recognition of the speaker, speech enhancement, and the language recognition. While making use of the theoretical and applied results of speech research, speech technology primarily focuses on the development of the models and algorithms necessary for successful speech-based applications, and not on the description of the speech process. Thus, speech technology with its few decades of history has significantly widened the borders of speech sciences: it further enriches the scope of corresponding scientific fields with the disciplines of digital signal processing, mathematical statistics, pattern recognition and information technology.

Today, significant changes are present in the field of speech sciences. The use of large-scale speech corpora has come to the foreground in phonetic research and speech technology applications. From the point of view of corpus phonetics, it becomes possible to search variant and invariant characteristics in the speech signal, and to check classical problems of phonetics and phonology. Speech and speaker recognition programs as well as ones producing machine speech can obtain large-scale speech sample stored orderly for their algorithms, due to speech databases of this kind. The move towards large databases characterizes not only speech sciences but most branches of science. By today, most fields have computational versions, which are marked by large-scale computation tasks like sample search in large databases or the execution of large simulations. Through the above, a third scientific method appeared beyond the theoretical and empirical branches, namely scientific computation. Research results from these fields can be checked only in a limited way on the basis of scientific publications, as the re-run of the research is made impossible by the lack of the used program codes, their input parameters and the content of the databases. The likelihood of inadvertent errors increases during the course of complicated analyses, which may lead to misleading correspondences. In the case of publications presenting the development new algorithms or which are based on program outputs, readers demand familiarization with the computational code, its sharing, as it presents (and at the same time proves) results in a more detailed way compared to any human-language description. Thus, reproducible research means the solution for the limited credibility of the results from computational research [6] [16].

In this study, we examine tools that facilitate the publication of reproducible analyses in some fields of speech research. Firstly, we provide a brief overview on the history of reproducible research, and then we encounter tools whose appropriate combination allows for reproducible analysis in the field of speech sciences.

## 2. Reproducible Research

Literate programming [10] words ideas very close to the conception of reproducible research. The combination of the code and the text is already present here, whose aim, as put forward by Knuth, was the best possible documentation of programs. This facilitates software development and encourages writing better programs. `WEB`, developed by Knuth, makes possible the use of the instructions of Pascal language and the commands of TeX typesetting system in one single file. By way of the *weaving* process the TeX text of the documentation can be gained from this file, and the Pascal source code can be obtained with the help of the *tangling* process. The marked-up code blocks and the using of the identifiers to another code block make possible for us to organize the blocks differently during the *weaving* and *tangling* processes. Later on, `CWEB` [11] based on C language was developed, and it was amended by the possibility of C++ and Java documenting. The `noweb` system [9] allows for documentation independent of the target programming language, which significantly simplified the application of literate programming. However, literate programming in itself does not support reproducible research, as the execution of code blocks within the document is not possible. The execution of source codes is only possible in a single step, at the end of the *tangling* process with executing compiler or interpreter, but it has no effect at all on the woved text.

The expression of reproducible research first appeared in the work of Claerbout and his colleague [3]. They introduced a new process in geophysical research, by which, similarly to the Makefile system, they could reproduce the figures of the electronic documents with a single touch. This idea was carried forward by Fomel and his colleague [7] in the open-source `Madagascar` software package. They achieved simpler reproducibility accessible from various platforms with a tool similar to Makefile (`SCons`). Buckheit and Donoho [2] developed a new `Matlab` package (`WaveLab` toolbox) for the research of wavelets, which makes possible the publication of data, `Matlab` codes (tables and illustrations in the article to be generated) and documentations, beyond articles. `Madagascar` and `WaveLab` store the code separate from the text of the article. According to the principles of literate programming, the Sweave system [13] allows for program codes and texts (R and LaTeX) to be present in a single document. Sweave is one of the most popular tools of reproducible research, and several similar tools came up at its impact, which, instead of the outstandingly strong R/LaTeX pair, support other languages, like odfWeave [1] (R/ODF), R2HTML [2] (R/HTML), R2wd [3] (R/Word), Knitr [4] (R/Markdown), SASweave [5] (SAS/LaTex) és StatWeave [6] etc. Today, the greatest freedom is provided by the Org-mode of Emacs text editor, which fully supports literate programming and reproducible research as well [18]. The instructions of 39 different languages can be combined in a simple text form language due to the Babel extension of Emacs.

Reproducible research has gained ground in more and more fields with the gradual development of its tools. Beyond the fields of geophysics and harmonic analysis mentioned above, articles were published in the fields of economics, signal processing, statistics, biostatistics, econometrics, epidemiology, climatology, neurophysiology as well [5 p. 2]. More and more periodicals urge authors to publish not only the articles themselves but also the data and codes supporting their results. Scientific periodicals can be the drives for changing culture, as they expect reproducible research results according to a set protocol. Periodicals of this kind are, for example, Nature[7], The Insight Journal [8], Annals of Internal Medicine [9], Biostatistics[10], IEEE Sinal Processing Magazine[11]. *The Journal of Experimental Linguistics* was directly launched for the online publication of reproducible research results related to speech and language, in 2009. (Currently, articles are not accessible yet from the webpage[12].)

## 2.1 What is reproducible research?

According to reproducible research, the end product of research is the publication and all the data and source codes in its computational environment which are necessary for the reproduction of the research results[13]. The compendium of reproducible research [8] is gained by packing these elements. Its components are[14]:

- The publication of the research (in PDF or TeX/Word format, including references)
- Data (row data, cleaned data, codes on data purification, detailed data dictionary)
- Computing environment (a well-documented source code, the description of parameters, settings and the running platform)
- Results (figures, tables, numerical data and their description)
- Complementary material (e.g. video and audio files)

All of the above components mean the storing of one or more separate files. However, there are several software packages which, based on the principle of literate programming, can condense more of the above-mentioned components into a single "executable" file (cf. above e.g. Sweave, Org-mode). We can find the complete, ever-increasing list of these tools on the webpage of *Reproducible Research Planet!* [15]. The combination of text and code in a single file serves other purposes in reproducible research than in literate programming. During generating the document intended to be published (*weaving* procedure), program codes get executed, and their results (figures, tables, numerical data, etc.) immediately appear in the text.

The publication of reproducible research undoubtedly requires more effort from the author, but it has its advantages for both the author and the reader. Jon Claerbout worded one of its greatest advantages for authors: "One of the main tenets of reproducible research is that time turns each one of us into another person. By making an effort to communicate with strangers; we help ourselves to communicate with our future selves." [16] Therefore, it is a fundamental requirement to be able to reproduce our own research. Perhaps the most important advantage for the reader and for science is the transparency of the research. Reproducibility allows us to filter possible errors of results arising from the ever-greater computational potential of multidimensional databases.

It is important to note that we may benefit from the advantages of reproducible research not only in scientific publications, but in various pedagogical environments as

[1] http://cran.r-project.org/web/packages/odfWeave/index.html
[2] http://cran.r-project.org/web/packages/R2HTML/index.html
[3] http://cran.r-project.org/web/packages/R2wd/index.html
[4] http://cran.r-project.org/web/packages/knitr/index.html
[5] http://homepage.cs.uiowa.edu/~rlenth/SASweave/
[6] Support many languages (R, S-plus, SAS, Stata, Maple) and different word processing systems (LaTeX, ODT). Available: http://homepage.cs.uiowa.edu/~rlenth/StatWeave/
[7] http://www.nature.com/authors/policies/availability.html
[8] http://www.insight-journal.org/
[9] http://annals.org
[10] http://www.oxfordjournals.org/our_journals/biosts/for_authors/msprep_submission.html
[11] http://www.signalprocessingsociety.org/publications/periodicals/tsp/
[12] http://elanguage.net/journals/jel
[13] Understand Reproducible Research (http://www.rrplanet.com/reproducible-research/reproducible-research.shtml)
[14] Build Your Reproducible Research Compendium (http://www.rrplanet.com/reproducible-research/build-reproducible-research.shtml)
[15] http://www.rrplanet.com
[16] http://sepwww.stanford.edu/OLDWWW/research/redoc/IRIS.html

well. During in-class presentation or online tutorials, the textual description of algorithms and analyses is made understandable best by the combination of the executable code and the output. Reproducible documents also facilitate the application of the new pedagogical model, according to which "students learn through their own activities" [15]. The starting point for student projects can be a known research result or open problem in a reproducible form. As the results can easily be re-run and modified, there is a possibility for active learning, the revision of results, or even achieving new scientific results. The reproducible research form generally facilitates communication between lecturers and students, as a party can more easily enter the research environment constructed by the other party, with the help of available codes and data. Furthermore, some authors emphasise that questions related to reproducible research must be made part of the education [4], [6], [12].

## 3. Reproducible Research in Speech Sciences

Complete replication has great traditions in speech sciences. The phonetic phenomenon is examined parallel by several research groups, and this is true even if we limit the scope of examinations to one single language.

Beside complete replication, reproducible research also has its place in speech sciences. This is partly due to the use of large databases mentioned in the introduction, and that of the gradually more and more complex computations and algorithms. Reproducibility in this case helps avoiding inadvertent errors and the extensive documentation of the several parameters of the algorithms. On the other hand, the importance of reproducible research is emphasized by the fact that elements of mathematical statistics are very frequent in analyses. Publications often contain information on sampling, the transformation of data, on gaining features and derived data, and they give forth statistical indicators, tables, the figures designed diversely, and the results of hypothesis tests. In the above cases, beyond better understanding, reproducibility offers readers the possible use of their own databases and/or the modification of transformations leading to statistical tests. Following the modification of data and/or codes, they can check changes in hypothesis examinations, the tables and figures drawn. The reproducibility of a procedure comprising so many steps ensures safe implementation in the future for the author.

### 3.1 The Proposed Model

It would be too bold to offer a reproducibility recipe for publications in all fields of speech sciences. The general theory can be validated in every case, of course: beyond the text of the publication, the software environment, the code and data are to be attached. The present study aims to examine experimental research in the field of speech sciences which have the following data and code components:

a) (public) speech database (data)
b) speech signal processing on the database (code)
c) the query of the speech database and the statistical analysis of results achieved (code)
d) the visualisation of results from the analysis with the help of figures, tables and indicators (code).

In the centre of experimental speech research we find the speech database, which is the combination of voice samples with annotation and documentation. The speech database generally contains the voice samples of several speakers in a unified format. This format can be of many kinds, and it is always the research goal that determines the most appropriate form.

The speech signal part of the database can be diversely examined with the help of digital signal processing and speech technology algorithms. A part of the algorithm results can also be stored in the database, besides the speech signal. The scope of software tools and packages possible to be used in speech processing is getting wider. These are used under changing parameter settings, often in a new way, and with the combination of several tools. In the majority of publications, the aim is often the introduction of a new algorithm or method.

The spin-off information, the labelling data and the results from signal processing in annotated speech databases can be arrived at and used in research in many different ways. With the help of statistical tests, we can check our hypotheses on the appropriately filtered and pre-processed data.

Figures, tables, and indicators are indispensable parts of the publication. Their visualization and insertion in the text also need to be taken care of.

We propose the model in Figure 1 for the implementation of the above data and code components - typically appearing in basic phonetic research -, in reproducible environment. We build on the concept of executable article, according to which the text and the codes are present in one single file (Figure1: 4), and the ready publication effectively comes about by "executing" the article. The figures, tables and numerical indicators of the article can change dynamically during the "execution", on the basis of the content and program codes (Figure 1: 4, R code) of the database. The computer-based environment of the model is the R statistical program package [17], which provides a comprehensive frame for generating reproducible publications in Linux, Mac OS X, and Windows environment, too. R environment was basically created for the implementation of statistical analyses, but it also includes a general purpose, high-level programming language, which has been added an inexhaustible warehouse of functions. R is free and open software.

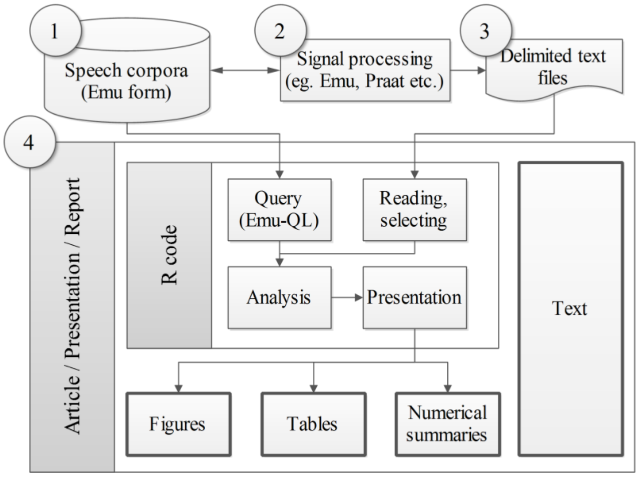


Fig. 1 The proposed model of reproducible research in speech research.

The reproducible environment contains four elements (according to the numbers of Figure 1):

1. EMU-format speech database
2. Software tools and packages for the analysis of the speech signal
3. The result of the analysis of the speech signal with tab-delimited text file. Results which cannot be inserted in the structure of the EMU database get in this category
4. The executable publication.

The EMU Speech Database System is a platform-independent (running on Linux, Mac OS X and Windows operational systems as well), integrated program package, which can be used for the generation, query, and analysis of annotated speech corpora [1]. Its services include the annotation of speech samples on the basis of waveform and spectogram. Furthermore, it has a digital signal processing module, too. It supports hierarchic and auto-segmented annotation, and the query of information stored this way, with its own query language (EMU-QL). The use of EMU is extremely facilitated by the fact that it has transparent interface from the start, for some statistical program package. Certain elements of the annotated speech database can be accessed first from the S, then *S-plus*, and today from the R statistical program package. Label files are freely convertible between Praat[17] and EMU, which significantly facilitates the generation of EMU-format speech databases.

Several procedures were developed in the last decades for the extraction of characteristics imbedded in speech. Parts of them were executed in EMU (e.g. F0 analysis, formant estimation, short-term autocorrelation function, short-term spectral analysis, Linear Prediction analysis, zero-crossing rates, etc.). The results of EMU analyses form an integral part of the EMU-format speech database, whose query is possible at a later stage from R, with the help of the EMU-QL language. This requires the `emu`[18] package of R. If we wish to run the script of another sign-processing program package (e.g. Praat, WaveSurfer/Snack[19]) or our personally developed algorithm, it is worth storing their results in a tab-delimited text file, because this format is easily readable for R. Let us note that R itself has speech processing functions, and several libraries aid sound analysis and signal processing. *Seewave* [20] offers an extremely rich set of functions for analysing, manipulating, displaying, and editing speech. It is excellent for visualizing oscillograms, 2D and 3D spectograms. *Sound*[21] and *tuneR*[22] packages offer basic functions for managing wav files, sound samples, and music. The *signal*[23] package provides Matlav/Octave-compatible signal processing functions. Furthermore, the *tcltk* package of R makes possible the use of *Tk* graphical user interface and issuing *Tcl* commands from within R. Speech processing libraries which have Tcl wrapper (e.g. WaveSurfer/Snack, ASSP) are directly accessible from R.

The executable article (Figure 1: 4) contains R codes and text. R codes ensure the query of data from the EMU speech database and from the tab-delimited text files. Furthermore, R codes are used for statistical analyses, and the visualization of images, tables and numerical indicators. The other part of the article is the text, whose format can vary. The R code can be combined with several document formatting languages. Through a simple example, we are presenting a system using three different formatting languages: LaTeX, Markdown and Org.

## 4. Example Application

We analyse formant frequency data[24] for presenting the practical application of reproducible research. In her study, Klára Magdics [14] published the average formant frequency values of Hungarian vowels, on the basis of manual measurements taken on voice spectograms. In a mini research, we compare the average values in unstressed position belonging to female speakers to the

[17] Praat is the most widely used free scientific software program for the analysis of speech: http://www.fon.hum.uva.nl/praat/

[18] http://cran.r-project.org/web/packages/emu/

[19] http://www.speech.kth.se/snack/

[20] http://cran.r-project.org/web/packages/seewave/index.html

[21] http://cran.r-project.org/web/packages/sound/index.html

[22] http://cran.r-project.org/web/packages/tuneR/index.html

[23] http://cran.r-project.org/web/packages/signal/index.html

[24] The frequency (Hz) of voice parts amplified during articulation can be demonstrated in the speech sign, and they change continuously.

formant data of our own database, which derive from one female speaker. We stored Magdics's average formant frequency values in a tab-delimited text file (Figure 1: 3), and our measurements in an EMU-format database (Figure 1: 1). One of the purposes of the example is to be able to compare the three form languages (LaTex, Markdown, Org-mode) responsible for generating the textual part of the executable publication (Figure 2: 4, Text). Therefore, the textual part of the research report includes a main title, a chapter title, a subtitle, and short paragraphs as well. Furthermore, the report includes mathematical formulae, which are frequent in texts, and as we will see, they do not cause problems in any of the systems. These are the static elements of the report. The dynamic part of the report is achieved through the execution of the R instructions during *weaving*. We present a numerical indicator (68) in the second paragraph of the report, as well as a table and a picture. These dynamic elements are carried out by almost the same R instructions in the three systems. We only need to pay attention to the difference between LaTeX or HTML in the code used for presenting the table. Thus, the other purpose of the example is to present how the dynamic R codes can be embedded in the text of the document. Let us now show in more detail how the three examination tools can be used in our mini research.

## 4.1 R/Sweave

`Sweave` allows us to embed R codes in a LaTeX document, according to the syntax of `noweb.` The `Rnw` extension file thus created is converted to a genuine LaTeX file by the `Sweave()` function of R (which can be further transformed to DVI and PDF files), and `Stangle()` saves codes in an R command file. Compared to the `noweb` system (and its predecessors), it is a major difference that R codes are already executed during the *weaving* procedure, and their results (picture, table and numerical data) appear in the final text extension document as well. If we change the data or the codes, we get the updated version of the document after the *weaving* procedure. Thus, an `Rnw` extension `Sweave` file can be regarded as an executable article (or to put it otherwise, a dynamic report), which, together with the data used ensure the reproducibility of our research results.

`Sweave` files contain document and code segments, which are called *chunk*s. A code chunk is the part between `<<>>=` and `@` in the file (see Appendix 2). There are various options in the opening part. `Label` names code chunks. There are 6 code chunks in the Sweave source code in Appendix 2. Their chosen names (`init`, `reading`, `query`, `analysis`, `table`, `figure`) reflect well the aim of the commands in them. (The names also harmonize with the labels of Figure 1, and they are used consistently in the other two systems as well.) We also use the `echo` option, which if `TRUE`, then commands echo in the document, whereas if it is `FALSE`, they do not. Here we completely avoid the visualization of commands used. The `results=hide` impedes the visualization of the different output of commands. The `fig=TRUE` option helps us to specify that a code block is responsible for generating and visualizing a figure. A table is generated in the chunk called `table`, with the help of the function `xtable()`, it converts R tables to LaTeX format. The `Sweave` system can make use of this comfortable possibility in the `results=tex` option. `\Sexpr{}` instruction makes it possible to place R commands in line. The complete `Rnw` file can be found in Appendix 2, and the generated PDF document in Appendix 1.

The generation of reproducible `Sweave` documents is greatly facilitated by the fact that the `Rnw` file can be converted into a PDF document with a single click from various graphical user interfaces. The most popular tool of this kind is RStudio[25].

## 4.2 R/Knitr

`Knitr` is an R package, which, on the one hand, complements the options offered by the `Sweave` system. The combination of LaTeX and R codes is made possible by a sort of syntax similar to that of `Sweave`, but it adds a few comfort functions. Beyond this, `Knitr` allows for the use of another document formatting system, namely Markdown, which is also supported by RStudio. The use of Markdown instead of LaTeX greatly simplifies the process of generating reproducible articles. The extension of documents generated by the joint use of R and Markdown is *Rmd*, and we have the possibility of generating an HTML file during `weaving.` The HTML file includes pictures embedded, not as a separate file, and it is capable of visualizing formulae with the help of MathJax[26]. Thus, we get the final form of our article in a single, easily portable HTML file. The author [27] recommends the use of Pandoc [28] document converting system for generating the desired target format starting from the HTML.

Appendix 3 shows the R/Markdown code of our mini research. We write R codes between `` ```r{} `` and `` ``` ``. The `weaving` procedure executes these code parts, and inserts the result in an HTML document. The syntax of LaTeX can be used to insert formulae, and in-line codes can be inserted between `` `r `` and `` ` ``. The generated HTML file can be found in Appendix 5.

[25] http://rstudio.org/

[26] http://www.mathjax.org/

[27] Yihui Xie (http://yihui.name/en/)

[28] http://johnmacfarlane.net/pandoc/

### 4.3 Org-mode/Babel

The `Org-mode` is a mode of Emacs text editor. Similar to Markdown, it is a plain text mark-up language for generating personal information, sketches and documentations. Source codes that can be executed with the help of the `Babel` extension can also be placed in the document between `#+begin_src` and `#+end_src`. The *org* source file of our research can be found in Appendix 4. The source language of code blocks can be optional, as `Org` currently supports 39 different programming languages, among them R, Python, Perl, C, C++, Java and Matlab languages. Codes of different languages can be combined in a single *org* file, and the full support of literate programming is also possible due to the naming and organization of code blocks. The execution of code blocks within the document is carried out here too, since it is possible not only at the end of the *tangling* procedure, thus it fully supports reproducible research as well. During exportation, several formats can be chosen. The PDF and HTML output hardly differ from the pictures in Appendix 1 and 5.

## 5. Limits

Today, we do not find a multitude of reproducible articles in science, even less so in the field of speech sciences. Peng [16] sees its reason in the lack of scientific culture, which should request the reproducibility of all scientific claims, and in the inefficiency of the infrastructure, which does not allow for the simple dissemination of reproducible research in a large scale. Vandewalle and his colleagues [20] mention that the publication of data is not always possible. If the databases used are protected by copyright, or they contain confidential information, their (on-line) publication is to be questioned. This especially concerns the field of speech technology, where targeted databases and algorithms are often created, whose publication is not easy. In such cases, reproducibility necessarily gets out of the spotlight.

Peng [16] defined an interval to check scientific claims, which has non-reproducible articles at one end and fully reproducible articles at the other end. In between the two, we find the different grades of reproducibility depending on how much the author makes available of his data and codes to others, and how simple it is to reconstruct the result on this basis. Some authors differentiate 3 stages [19], while others divide the reproducibility scale to 6 parts [20]. Thus, if data cannot be disseminated due to, for example, their confidentiality, or if they are too messy to be worth disseminating, the independent dissemination of codes is already a step towards reproducibility.

## 6. Conclusion

Reproducible research is a necessary but not sufficient condition for good research. Reproducibility in itself does not guarantee quality and the validity of results. It only ensures that the result is what we claim in the publication.

With the special combination of available tools, we have provided a reproducibility recipe for experimental research in some fields of speech sciences. We based our model on the R environment, the EMU-format speech database, and the executable publication. We allow the execution of optional signal processing applications for the versatile extraction of speech characteristics, but require proportioned text files as output, which is easily readable for R. We did not set the typesetting system of the executable publication, thus more tools can be used.

Table 1: Characteristics of tools mentioned in the article

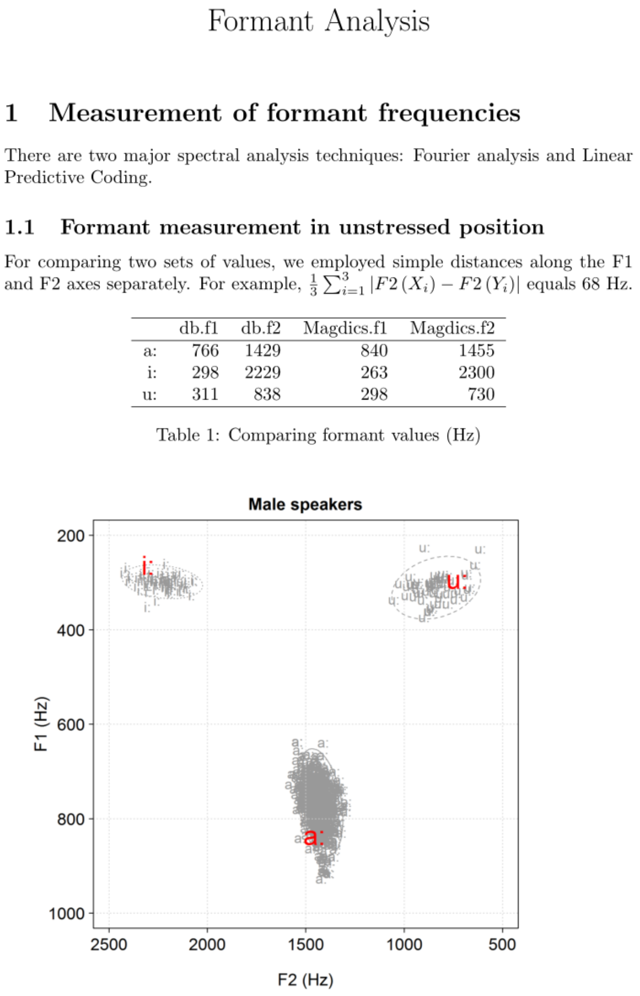
| ***Tool*** | ***Lang.*** | ***Typesetting*** | ***Export*** | ***LP*** | ***RR*** |
|---|---|---|---|---|---|
| `WEB` | Pascal | TeX | TeX | yes | no |
| `CWEB` | C/C++/ Java | TeX | TeX/HTML | yes | no |
| `noweb` | any | (La)TeX/ troff/HTML | (La)TeX/ troff/HTML | yes | no |
| `Sweave` | R | LaTeX | LaTeX/HTML | partial | yes |
| `Knitr` | R | LaTeX/ Markdown | LaTeX/HTML | partial | yes |
| `Org-mode` | any | Org | any | yes | yes |

We have summarized some of the reproducible tools detailed in the article in Table 1. According to the table, we can choose from `Sweave`, `Knitr` and `Org-mode` to execute reproducible research. If our aim is to keep contact with students or to carry out smaller research quickly, we strongly recommend the combination of `Knitr`/Markdown. The report exported in HTML can be immediately published accessibly from RStudio, too. Following this, our research report is available for anyone at the RPubs[29] site right away. We shared the result of our mini research with this method, thus it is available from *http://rpubs.com/abarik/901* URL. `Sweave` is the most appropriate system for larger reports, articles or books. In the case of multi-language environments, `Org-mode` seems to be the best choice.

The above systems can be expected to developed and simplified in the future. It is urgent to archive the complete compendium of reproducible research, as well as to ensure its long-term availability, that is, to establish the infrastructure of reproducibility in a wider range.

[29] *http://rpubs.com*

## Appendix 1. PDF form of Formant Analysis



# Formant Analysis

## 1 Measurement of formant frequencies

There are two major spectral analysis techniques: Fourier analysis and Linear Predictive Coding.

### 1.1 Formant measurement in unstressed position

For comparing two sets of values, we employed simple distances along the F1 and F2 axes separately. For example, $\frac{1}{3}\sum_{i=1}^{3}|F2(X_i) - F2(Y_i)|$ equals 68 Hz.

| | db.f1 | db.f2 | Magdics.f1 | Magdics.f2 |
|---|---|---|---|---|
| a: | 766 | 1429 | 840 | 1455 |
| i: | 298 | 2229 | 263 | 2300 |
| u: | 311 | 838 | 298 | 730 |

Table 1: Comparing formant values (Hz)

## Appendix 2. Sweave source code of Formant Analysis (formant.Rnw)

```
\documentclass[a4paper]{article}
\title{Formant Analysis}
\date{}
\begin{document}
\maketitle
<<label=init, echo=FALSE>>=
options(SweaveHooks=
    list(fig=function() source("graph_init.R")))
library(xtable); library(emu)
@
<<label=reading, echo=FALSE, results=hide>>=
# Reading
d <- read.table("magdics.txt", sep="\t",
                 header=T, dec=",")
# Selecting
d <- subset(d,subset=sex=="men" & stress=="yes",
     select=c("labels", "f1", "f2"))
names(d)[2:3] <- c("Magdics.f1", "Magdics.f2")
@
<<label=query, echo=FALSE, results=hide>>=
# EMU query
d.seg <- emu.query(template="rfa",pattern="f_*",
                   query="bphonetic=a:|i:|u:")
d.track <- emu.track(seglist=d.seg,
                     trackname="fm", cut=0.5)
names(d.track)[1:2] <- c("db.f1", "db.f2")
d.m <- cbind(d.seg, d.track[,1:2])
@
<<label=analysis, echo=FALSE, results=hide>>=
d.aggr<-aggregate(d.m[,c("db.f1","db.f2")],
        list(labels=d.m$labels),mean,na.rm=T)
d.ready <- merge(d.aggr, d, by="labels")
@

\section{Measurement of forman frequencies}
There are two major spectral analysis
techniques: Fourier analysis and Linear
Predictive Coding.

\subsection{Formant measurement in unstressed
position}
For comparing two sets of values, we employed
simple distances along the F1 and F2 axes
separately.For example,
$\frac{1}{3}\sum_{i=1}^{3}\left|F2\left({X}_{i}
\right)-F2\left({Y}_{i} \right) \right|$
equals \Sexpr{ round(sum(abs(d.ready$db.f2-
d.ready$Magdics.f2))/3, 0) } Hz.
<<label=table, echo=FALSE, results=tex>>=
rownames(d.ready) <- d.ready$labels
d.ready$labels <- NULL
xtable(d.ready, label="fm1", digits=0,
       caption="Comparing formant values (Hz)")
@
<<label=figure, echo=FALSE, fig=TRUE,
png=TRUE>>=
eplot(x=d.track[,1:2],labs=d.seg$labels,
      dopoints=T, doellipse=T, form=T,
      lty=1:3, xlab="F2 (Hz)", ylab="F1 (Hz)",
      col="gray60", main="Male speakers",
      xlim=c(500, 2500), ylim=c(200, 1000),
      panel.first = grid())
text(-d.ready$Magdics.f2,-d.ready$Magdics.f1,
     rownames(d.ready),col="red",cex=1.9)
@
\end{document}
```

## Appendix 3. R Markdown code of Formant Analysis (formant.Rmd)

~~~~
Formant Analysis
===================================================

```{r init, echo=FALSE, results='hide',
      message=FALSE}
library(xtable); library(emu)
```

```{r reading, echo=FALSE, results='hide'}
# Reading
d <- read.table("magdics.txt", sep="\t",
                header=T, dec=",")
# Selecting
d <- subset(d, subset=sex=="men" & stress=="yes",
            select=c("labels", "f1", "f2"))
names(d)[2:3] <- c("Magdics.f1", "Magdics.f2")
```

```{r query, echo=FALSE, results='hide'}
# EMU query
d.seg <- emu.query(template="rfa",pattern="f_*",
                   query="bphonetic=a:|i:|u:")
d.track <- emu.track(seglist=d.seg,
                     trackname="fm", cut=0.5)
names(d.track)[1:2] <- c("db.f1", "db.f2")
d.m <- cbind(d.seg, d.track[,1:2])
```

```{r analysis, echo=FALSE, results='hide'}
d.aggr<-aggregate(d.m[,c("db.f1","db.f2")],
         list(labels=d.m$labels),mean,na.rm=T)
d.ready <- merge(d.aggr, d, by="labels")
```

Measurement of forman frequencies
-----------------------------------------
There are two major spectral analysis techniques:
Fourier analysis and Linear Predictive Coding.

### Formant measurement in unstressed position

For comparing two sets of values, we employed
simple distances along the F1 and F2 axes
separately. For example,
$\frac{1}{3}\sum_{i=1}^{3}\left|F2\left({X}_{i}
\right)-F2\left({Y}_{i} \right) \right|$  equals
`r round(sum(abs(d.ready$db.f2-
d.ready$Magdics.f2))/3, 0)` Hz.

```{r table, echo=FALSE, results='asis'}
rownames(d.ready) <- d.ready$labels
d.ready$labels <- NULL
print(xtable(d.ready, label = "fm1", digits=0,
      caption = "Comparing formant values (Hz)",),
      type="html")
```

```{r figure, echo=FALSE}
eplot(x=d.track[,1:2],labs=d.seg$labels,
      dopoints=T, doellipse=T, form=T,
      lty=1:3, xlab="F2 (Hz)", ylab="F1 (Hz)",
      col="gray60", main="Male speakers",
      xlim=c(500, 2500), ylim=c(200, 1000),
      panel.first = grid())
text(-d.ready$Magdics.f2,-d.ready$Magdics.f1,
 rownames(d.ready), col="red", cex=1.9)
```
~~~~

## Appendix 4. Org-mode source code of Formant Analysis (formant.org)

~~~~
#+TITLE: Formant Analysis
#+PROPERTY: session *R*

#+name: init
#+begin_src R :exports none
library(xtable); library(emu)
#+end_src

#+name: reading
#+begin_src R :exports none
  # Reading
  d <- read.table("magdics.txt", sep="\t",
                  header=T, dec=",")
  # Selecting
  d <- subset(d,subset=sex=="men" &
   stress=="yes", select=c("labels","f1", "f2"))
  names(d)[2:3] <- c("Magdics.f1", "Magdics.f2")
#+end_src

#+name: query
#+begin_src R :exports none
  # EMU query
  d.seg <- emu.query(template="rfa",
                  pattern="f_*",
                  query="bphonetic=a:|i:|u:")
  d.track <- emu.track(seglist=d.seg,
                       trackname="fm", cut=0.5)
  names(d.track)[1:2] <- c("db.f1", "db.f2")
  d.m <- cbind(d.seg, d.track[,1:2])
#+end_src

#+name: analysis
#+begin_src R :exports none
  d.aggr<-aggregate(d.m[,c("db.f1","db.f2")],
        list(labels=d.m$labels),mean,na.rm=T)
  d.ready <- merge(d.aggr, d, by="labels")
#+end_src

* Measurement of forman frequencies
  There are two […] Linear Predictive Coding.

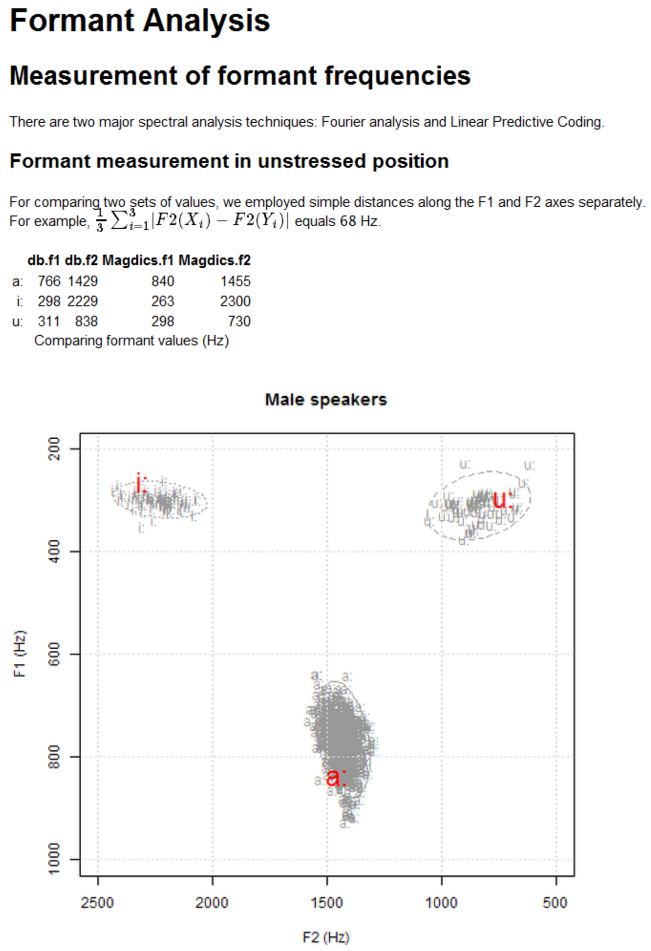
** Formant measurement in unstressed position
For comparing […] For example,
$\frac{1}{3}\sum_{i=1}^{3}\left|F2\left({X}_{i}
\right)-F2\left({Y}_{i} \right) \right|$ equals
 src_R{round(sum(abs(d.ready$db.f2-
d.ready$Magdics.f2))/3, 0)} Hz.

#+name: table
#+begin_src R :exports results :rownames yes
:colnames yes
  rownames(d.ready) <- d.ready$labels
  d.ready$labels <- NULL; d.ready
#+end_src

#+name figure
#+begin_src R :results graphics :exports none
:file plot.png
  eplot(x=d.track[,1:2],labs=d.seg$labels,
      dopoints=T, doellipse=T, form=T,
      lty=1:3,xlab="F2 (Hz)",ylab="F1 (Hz)",
      col="gray60",main="Male speakers",
      xlim=c(500, 2500), ylim=c(200, 1000),
      panel.first = grid())
text(-d.ready$Magdics.f2,-d.ready$Magdics.f1,
     rownames(d.ready),col="red", cex=1.9)
#+end_src

#+RESULTS: figure
[[file:plot.png]]
~~~~

## Appendix 5. HTML form of Formant Analysis



# Formant Analysis

## Measurement of formant frequencies

There are two major spectral analysis techniques: Fourier analysis and Linear Predictive Coding.

### Formant measurement in unstressed position

For comparing two sets of values, we employed simple distances along the F1 and F2 axes separately. For example, $\frac{1}{3}\sum_{i=1}^{3}|F2(X_i) - F2(Y_i)|$ equals 68 Hz.

| | db.f1 | db.f2 | Magdics.f1 | Magdics.f2 |
|---|---|---|---|---|
| a: | 766 | 1429 | 840 | 1455 |
| i: | 298 | 2229 | 263 | 2300 |
| u: | 311 | 838 | 298 | 730 |

Comparing formant values (Hz)

Male speakers

## References

[1] Lasse Bombien, Steve Cassidy, Jonathan Harrington, Tina John, and Sallyanne Palethorpe, "Recent Developments in the Emu Speech Database System", in Proceedings of the Australian Speech Science and Technology Conference, Auckland, New Zealand, 2006, pp. 313–316.

[2] Jonathan B. Buckheit and David L. Donoho, "WaveLab and reproducible research", Dept. of Statistics, Sanford Univ., Tech. Rep. 474. 1995.

[3] Jon Claerbout and Martin Karrenbach, "Electronic documents give reproducible research a new meaning", in Proc. 62nd Ann. Int. Meeting of the Soc. of Exploration Geophysics, 1992, pp. 601–604.

[4] Jennifer Crocker and M. Lynne Cooper, "Addressing scientific fraud", Science, Vol. 334, No. 6060, 2011, pp. 1182.

[5] Matthieu Delescluse, Romain Franconville, Sébastien Joucla, Tiffany Lieury, and Christophe Pouzat, "Making neurophysiological data analysis reproducible: Why and how?", Journal of Physiology (Paris), Vol. 106, No. 3-4, 2011, pp. 159–170.

[6] David Donoho, Arian Maleki, Inam Rahman, Morteza Shahram, and Victoria Stodden, "15 years of reproducible research in computational harmonic analysis", Technical report, 2008.

[7] Sergey Fomel and Gilles Hennenfent, "Reproducible computational experiments using SCons", in IEEE International Conference on Acoustics, Speech and Signal Processing, 2007, Vol. 4, pp. 1257–1260.

[8] Robert Gentleman and Duncan Temple Lang, "Statistical analyses and reproducible research", Technical report, Bioconductor Project, 2004.

[9] Andrew Johnson and Brad Johnson, "Literate programming using noweb", Linux Journal, Vol. 1997, No. 42, pp. 64–69.

[10] Donald E. Knuth "Literate programming", Computer Journal, Vol. 27, No. 2, 1984. pp. 97–111.

[11] Donald Ervin Knuth and Silvio Levy, The CWEB System of Structured Documentation: Version 3.0, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.

[12] Jelena Kovacevic, "How to encourage and publish reproducible research", in IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 4, 2007, pp. 1273–1276.

[13] Friedrich Leisch, "Sweave: Dynamic generation of statistical reports using literate data analysis", in Proceedings in Computational Statistics, Physica Verlag, Heidelberg, 2002, pp. 575–580.

[14] Klára Magdics, A magyar beszédhangok akusztikai szerkezete (The acoustic characteristics of Hungarian speech sounds), Budapest: Akadémiai Kiadó, 1965.

[15] David S. Moore, "New pedagogy and new content: The case of statistics", International Statistical Review, Vol. 65, 1997, pp. 123–165.

[16] Roger D. Peng, "Reproducible Research in Computational Science", Science, Vol. 334, No. 6060, 2011, pp. 1226–1227.

[17] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[18] Eric Schulte, Dan Davison, Thomas Dye, and Carsten Dominik, "A multi-language computing environment for literate programming and reproducible research", Journal of Statistical Software, Vol. 46, No. 3, 2012, pp.1–24.

[19] Matthias Schwab, Martin Karrenbach, and Jon Claerbout, "Making scientific computations reproducible", in Computing in Science & Engineering, Vol. 2, No. 6, 1997, pp. 61–67.

[20] Patrick Vandewalle, Jelena Kovacevic, and Martin Vetterli, "Reproducible research in signal processing - what, why, and how", IEEE Signal Processing Magazine, Vol. 26, No. 3, 2009. pp. 37–47.

**Kálmán Abari** received his MSc in Software Engineering from University of Debrecen, Hungary. He is currently pursuing his PhD degree in Computer Science from the same university. His research interests include Speech Research (acoustic characteristics of speech sounds), Statistical Analysis (functional data analysis) and Statistical Software (R). He has published 7 papers in refereed journals and international conference proceedings in the above areas. He works as teaching assistant at Institute of Psychology (Informatics, Statistics, Artificial Intelligence).