Making Neurophysiological Data Analysis Reproducible: An Example with the Open Source Software R

Christophe Pouzat

May 17 2010

Contents

1 Introduction

A Geophysicist's Viewpoint

> An article about computational neurophysiological data analysis or modeling 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 analysis/modeling results.

Jon Claerbout (with minor contextual adaptations).

Jon Claerbout seems to have been the first to advocate and *implement* reproducible research ideas, distributing compressed folder arborizations with data and `fortran` codes. The analysis was reproduced with the `make` command of Unix operating systems[1]. This approach has been extended by the `Madagascar`[2] project. Claerbout ideas have been enthusiastically adopted by another member of the same university, David Donoho from Stanford, who has developed a `Matlab` toolbox, `Wavelab`[3] implementing reproducible research ideas in a wavelet analysis context. See Donoho et al. [2008] for a summary of 15 years of practice with this methodology.

A Biostatistician's Viewpoint

> In concrete terms, the author selects the data she will use to defend a particular point of view or conclusion. She then transforms that data to produce figures, tables and to fit models. The output of these computations is assembled into the finished document and used to convince the readership that the point of view or conclusion is valid. In these terms one may view papers based on computation as advertisement. There is a leap of faith required by the readers; they must believe that the transformations and model fitting were done appropriately and without error.

Robert Gentleman and Duncan Temple Lang, "Statistical Analysis and Reproducible Research" (2004).

The cited technical report of Gentleman and Temple Lang [2004] has also been published as a "proper" journal article [Gentleman and Temple Lang, 2007].

The Reader/Referee Case

As a reader and/or a referee of neurophysiological papers you probably already wondered a couple of times about questions like:

- What would happen to the analysis (or simulation) results if a given parameter had another value?

[1] See: `http://sep.stanford.edu/doku.php?id=sep:research:reproducible`.

[2] `http://www.reproducibility.org/wiki/Main_Page`.

[3] `http://www-stat.stanford.edu/~wavelab/`.

- What would be the effect of applying my pre-treatment to the data instead of the one used by the authors?
- What would a given figure look like with a log scale ordinate instead of the linear scale use by the authors?

The Restart After a Break Case

As Neurophysiologists we all have to carry "complex" data analysis and data display. Often we work on a project for which we daily repeat a fairly long sequence of sometimes tricky analysis leading to beautiful graphics.We take a break and after 6 months or 1 year we get to do again very similar analysis for a related project...*Then the nightmare starts since we forgot*:

- The (numerical) filter settings we used,
- The detection threshold we used,
- The way to get initial guesses for our nonlinear fitting software to converge reliably.

2 The compendium concept

The Dynamic Document / Compendium Concept

One way to tackle these problems is to take seriously the *Dynamic Document / Compendium* concept of Gentleman and Temple Lang [2007]. Such "documents":

- encapsulate the actual work of the author, not just an abridged version, and allow different levels of detail to be displayed in different derived documents;
- are easy to re-run by any interested reader, potentially with different inputs;
- by providing explicit computational details they make it easier for others to adapt and extend the reported methodology;
- they enable the programmatic construction of plots and tables (in contrast with most of the current methods that are equivalent to cut and paste methodologies and have the associated problems);
- they allow the document to be treated as data or inputs to software and manipulated programmatically in numerous ways.

3 The Tools

What Do We Need?

- A general purpose analysis software like `Igor`, `Matlab`, `R`, and/or an easy way to call task specific software written in `C`, `fortran`, `Python`, etc;

- A tool to mix description and computation, that is a kind of "meta-programing" language;
- A way to put everything (data, meta-program, software developed specifically for the task) together.

Literate Programming

Literate programming is a phrase coined by Donald Knuth to describe the approach of developing computer programs from the perspective of a report or prose. The focus, then, is on description (and documentation) of the approach in human-readable form. This is in contrast to the normal approach of focusing on the code: A good programmer writes codes easily understood by compilers.

The Bad News

If you want to implement these ideas for classical `C`/`fortran` codes as well as for Dynamic Documents you are likely to have to learn a few new tools:

- LaTeX a set of macros developed by Leslie Lamport making it easy to use the TeX typesetting engine of Donald Knuth;
- `R`, an open source implementation of the `S` programming language developed by John Chambers.

The First Good News: LaTeX is Great!

Fair enough, the first week using LaTeX, coming from `Word`, can be a bit painful, but it worths it because:

- LaTeX allows you to produce beautiful prints and presentations (you do not need Power Point anymore, the LaTeX `beamer` class will do it for you);
- LaTeX allows you to focus on the logical structure of what you are writing, you do not mix up two different jobs: writer and typographer;
- LaTeX files are `ASCII` files and are therefore *small* and *easy to send* via Internet.

The Second Good News: `R` is Even Better!

- `R` is open source and runs on (almost) every computer (Unix, Windows, MacOS, etc);
- `R` has great graphical capabilities (better than `Igor`);
- an impressive range of user developed packages make R the tool of choice for computational data analysis;
- The user support through the R-help mailing list is astonishing.

More on R, For The Experts

- R can be easily extended by interfacing it with C/fortran libraries;
- It is easy to debug C/fortran libraries called from R;
- R implements typed variables, classes and methods (that is object oriented programing);
- Friedrich Leisch developed the *Sweave* package for R.

See Leisch [2002a,b, 2003a,b], Rossini [2001], Rossini and Leisch [2003] for details about Sweave.

Sweave is The Key Tool We Need

- Sweave processes meta-files containing both texts "chunks" written in LaTeX or HTML and R code chunks;
- It copies verbatim the text chunks in a new LaTeX file;
- It executes the commands in the R code and replaces the code chunks of the original meta-file by the results of their execution in the new LaTeX file (including figures).

If LaTeX scares you and you really do not want to use it, don't stop reading. You can also implement the reproducible analysis approach using OpenOffice instead of LaTeX. Max Kuhn, Steve Weaston and Nathan Coulter have developed odfWeave: Sweave processing of Open Document Format (ODF) files[4].

4 An Example

An Example

At that point I want to stop my "advertisement" for the reproducible research approach and show a simple example of its implementation in a spike sorting context.

In this example the role of the "finished" printed document is played by the file ReproducibleAnalysis.pdf and the "vignette" containing the mixture of description, typeset with LaTeX, and R code is contained in the file ReproducibleAnalysis.Rnw. Both of these files are included together with the analyzed data and analysis specific scripts in a compressed folder arborization that can be downloaded from http://sites.google.com/site/spiketrainanalysiswithr/ (at the bottom of the page).

[4] http://cran.at.r-project.org/web/packages/odfWeave/index.html.

References

David Donoho, Arian Maleki, Morteza Shahram, Victoria Stodden, and Inam Ur-Rahman. Fifteen Years of Reproducible Research in Computational Harmonic Analysis. Technical report, Stanford University, apr 2008. Available at: http://www-stat.stanford.edu/~donoho/Reports/2008/15YrsReproResch-20080426.pdf.

Robert Gentleman and Duncan Temple Lang. Statistical Analyses and Reproducible Research. Working Paper 2, Bioconductor Project Working Papers, 29 May 2004. URL http://www.bepress.com/bioconductor/paper2/. Available at: http://www.bepress.com/bioconductor/paper2/.

Robert Gentleman and Duncan Temple Lang. Statistical Analyses and Reproducible Research. *Journal of Computational and Graphical Statistics*, 16(1):1–23, 2007. doi: 10.1198/106186007X178663. URL http://pubs.amstat.org/doi/abs/10.1198/106186007X178663.

Friedrich Leisch. Sweave, Part I: Mixing R and LaTeX. *R News*, 2(3):28–31, December 2002a. URL http://CRAN.R-project.org/doc/Rnews/.

Friedrich Leisch. Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg, 2002b. URL http://www.stat.uni-muenchen.de/~leisch/Sweave. ISBN 3-7908-1517-9.

Friedrich Leisch. Sweave, Part II: Package Vignettes. *R News*, 3(2):21–24, October 2003a. URL http://CRAN.R-project.org/doc/Rnews/.

Friedrich Leisch. Sweave and Beyond: Computations on Text Documents. In Kurt Hornik, Friedrich Leisch, and Achim Zeileis, editors, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria*, 2003b. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/. ISSN 1609-395X.

A. J. Rossini. Literate Statistical Analysis. In Kurt Hornik and Friedrich Leisch, editors, *Proceedings of the 2nd International Workshop on Distributed Statistical Computing, Vienna, Austria*, 2001. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2001/Proceedings/. ISSN 1609-395X.

Anthony Rossini and Friedrich Leisch. Literate Statistical Practice. UW Biostatistics Working Paper Series 194, University of Washington, 2003.