SYNESIS
A JOURNAL OF SCIENCE, TECHNOLOGY, ETHICS, AND POLICY

Interoperability of Free Software Packages to Analyze Functional Human Brain Data

Tilmann Sander-Thömmes [1*], Alois Schlögl [2]

1. Medical Physics and Metrological Information Technology, Physikalisch-Technische Bundesanstalt, Berlin, Germany, Email: tilmann.sander-thoemmes@ptb.de. 2. Scientific Computing and Core Infrastructure, Institute of Science and Technology, Klosterneuburg, Austria.

Abstract

Functional neuroimaging of the human brain using non-invasive measurement methods is a success story of both hard- and software development. Hardware is a domain of commercial companies for good reasons such as the need to maintain an expensive set of machines. Software does not require massive investments to start and therefore the development of analysis software follows different schemes. It can be commercial or non-commercial, free, open source or proprietary. A multitude of highly specialized analysis methods and complex processing chains have been documented in the scientific literature. It is highly inefficient if methods or algorithms are coded from scratch every time that they are needed for a research project. Therefore, large parts of the neuroscience research community have followed the "Linux" or free software paradigm in developing their own software tools in a cooperative way. Here typical properties of successful free software packages are described and it is argued, that a further boost is possible through interoperability of packages without sacrificing the specific target of each package. Interoperability can be achieved through a common data format for measured and processed data or through the incorporation of packages into other packages, i.e., delegating tasks to the most suitable package.

Keywords: free software, interoperability, brain imaging, electro/magnetoencephalography, functional magnetic resonance imaging

Introduction

In-vivo neuromaging relies on several measurement modalities, which either probe the vascular or the neuronal response. In most widespread use are functional magnetic resonance imaging (fMRI) to asseess the vascular responses, and electro- and magnetoencephalography (E/MEG) for neuronal responses (1). Other modalities (e.g., functional near-infrared spectroscopy, fNIRS) are also important, but are not discussed here for lack of space. The large datasets resulting from neuroimaging measurements require a massive amount of offline processing to obtain meaningful results. This is a consequence of both the size of the datasets, as well as the need to fit complicated models or apply a fixed processing chain (consisting of several algorithms) to the data (2). Furthermore, often several different models need to be tested and new models and pre-processing algorithms are continually added to the fund of capable computational tools. As a consequence, a majority of the neuroscientific community has adopted the free software model (3-5), and some software packages have been maintained for almost two decades.

Almost every software (exceptions here being some low level machine code) is based on some other software (e.g. compilers, libraries, interpreters). Depending whether the

prerequisites are freely available or not, we can distinguish between software ecosystems that are truly free and those that are proprietary. A collection of functions (a suite or toolbox) can be written for specialized mathematics software (e.g., the non-free MatlabTM [www.mathworks.com] or its free alternative Octave [www.gnu.org/software/octave/] and some functions suites are coded in a general purpose language (such as C or C++) using the provided specialized mathematics libraries. Another tool is the free Python (www.python.org) language and its Python(x, y) distribution for use in scientific applications (6).

It is important to understand the difference between the terms "free" software and proprietary software. Free software, as defined by the Free Software Foundation, provides "the users the freedom to run, copy, distribute, study, change, and improve the software" (7). In contrast, proprietary software has at least one of more of these rights withheld by some entity. Note, however, that the freedom to improve software implies that the software is open source. The reverse is not necessarily the case. It is interesting to note that the very nature of software facilitates the emergence of commercial monopolies, but at the same time major commercial support for free software exists. It is much more difficult to obtain a hardware monopoly, as hardware is assembled from parts from many different independent suppliers.

In the following essay we introduce the ecosystem of free software packages, and describe two typical interoperability mechanisms. Implications and added value from interoperability are discussed, and conclusions are drawn relevant to use, ethics and policy issues.

Software maintenance and acknowledgement

To develop successful software packages several conditions must be met:

i. An initial core development targeted at a fairly broad set of questions must reach maturity to share the code with others.
ii. Funding for software maintenance must be secured (this is not an easy task as the maintenance itself is not a scientific task and therefore difficult to justify in a grant application).
iii. An easy and quick response mechanism to any software bugs reported by the users must be in place.
iv. The software must successfully link with the scientific community (e.g., tutorials and introductory material should be offered). Code is sufficiently accessible for new users. An up-to-date website or help system is usually self-evident.
v. There is a person similar to a "Cerberus" separating useful additions to the package from unwanted clutter (if you are a "Cerberus", please do not be offended, your role is very important. Often it is very difficult to get past the Cerberus with code suggestions, but that is one of the secrets to achieving a successful package. After all the Cerberus has the responsibility for integrity and consistency of the code).

Points (i-v) are by no means exhaustive; but rather serve to illustrate the complexity of the task.

A largely unresolved issue is the proper acknowledgment of software package contributors, (i.e., the maintainer(s), code developers, and authors of published methods incorporated into the package). A processing chain can consist of several steps, each based on an algorithm published in a peer reviewed journal. Users of the software package can certainly not cite all original work; instead only the work relevant to the scientific question addressed in the study need be cited. Therefore, a mechanism must be developed to easily acknowledge contributors to free software packages and this acknowledgment should be regarded to be of similar value as a scientific paper. To include such a citation in the actual code is only a first step. (For example, the following citation from reference 9)):

function vol = ft_headmodel_singlesphere(geometry, varargin) % For MEG this implements Cuffin BN, Cohen D. "Magnetic fields of a dipole in % special volume conductor shapes" IEEE Trans Biomed Eng. 1977 Jul;24(4):372-81.

Future developments need to facilitate or automate bug reporting, so that first use disappointments are avoided. A two-level system of user contributions might be needed, in which first level consists of central hosting of untested user contributions; and second level consists of full integration of new code into the package.

Interoperability of software packages

Collecting a set of functions into a software package provides users with powerful tools to rapidly perform sophisticated analyses. A second degree of re-usability can be attained if several packages interact and complement each others' functionality. This can be an application programming interface that enables function calls across package boundaries. At minimum, a data exchange procedure between packages can be defined. These two types of interactions are illustrated in Figure 1 and 2.

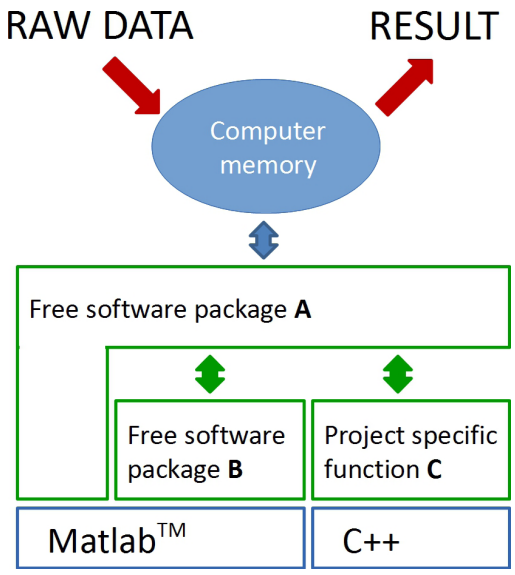

Figure 1. Multiple package data analysis pipeline. A top-level package A operates on data in memory and draws on the functionality from B and C. Packages B and C do not need to be based on the same programming language.

As shown Figure 1, measured data are loaded into memory, and a (top-level) package operates on these data. This top-level package uses functions from other packages for certain calculations. The final result is stored in memory and ready for visualization or storage. The packages need not be based on the same programming language, but the languages do need to provide programming interfaces. Examples for this type of interaction are SPM (8), which uses functions related to magnetoencephalography form FieldTrip (9). Another example is FieldTrip using functions from EEGLAB (10) to calculate independent component analysis. In the first case, FieldTrip is serving SPM (in the way that package B is called from A in Figure 1), in the second case FieldTrip is benefitting from EEGLAB.

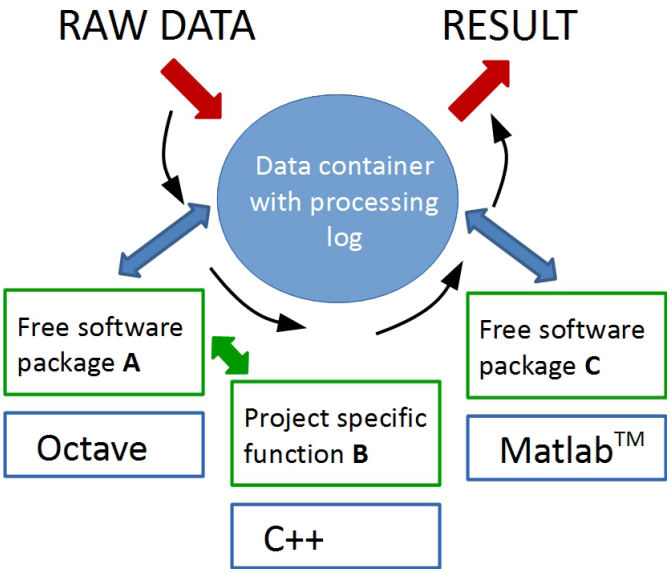

Figure 2. Data container based analysis pipeline. The data container model allows a pipeline of functions from different packages. A and C to modify its content and place log entries in the container. Still some packages might call functions from other packages as A is calling a function from B.

If a close integration of packages is not possible, a functionality similar to interoperability can be achieved by a data container with log capability as shown in Figure 2. Package A loads the data and processes them using function B, and stores the data back to the container. Package C then takes the data and modifies them into the final result. Each time the data in the container are changed, an entry must be added to the log. For this type of interoperability, packages like the BioSig (11) are very important as they provide input/output operations for many data formats.

Successful interoperability between different software systems depends on a number of different factors. State-of-the-art software structure requirements must be fulfilled so that two ore more systems work well together. Technical issues are sometimes also addressed by national or international standards and quasi-standards. If standards are not available, the free software development model provides an efficient way to establish quasi-standards. Another factor is legal requirements that establish if and how the systems can be connected together without infringement of rights. In software, these legal requirements mostly concern copyright and software licencing, yet sometimes, software technology patents need to be

considered. However, patents are considered a threat to interoperability and free software development often uses licensing regimes (e.g., GPL v3) with provisions against software patents. Obviously, patenting of software platforms, engagement of software intellectual property (IP) rights, and the leveraging of these rights can affect the ways that neuroimaging (and other types of bioengineering tools in/for neuroscience) are used and gain influence in international markets and socio-economic spheres. This raises a host of ethico-legal and even political issues, which while exceedingly relevant to science and technology policy, are outside the focus of this essay (for insight to these issues and problems, see recent work of Brindley and Giordano (12,13).

Discussion

If two separately developed packages share algorithms, their interoperability simplifies the verification of results associated with these algorithms. For non-compatible packages, a complete processing pipeline must be enabled in both packages. For interoperable packages, only the calls to the different routines implementing the same algorithm need to be repeated, and their output can be compared. Besides formal interoperability of packages through an application programming interface, it is optimal to check results provided by a given package function. Test data with known behavior need to be analyzed with the function. In case of multiple levels of interoperability, a certain basic trust in the developers ability is needed, as not all functionality can be verified.

An obvious question is why different packages have evolved instead of a single package covering all applicable methods and algorithms? In the main, this is a consequence of the interdisciplinary nature of the field and specialization in each contributory subfield. Research can be limited to one subfield, for example, for a pure fMRI study a single software package is often sufficient; Functionality will be tailored towards fMRI, allowing rapid data processing. Another project might involve MEG in a study involving anatomical MRI images. Consequently, an electrophysiological package is needed with some functionality from an MRI package. If in a secondary step the MEG data require more sophisticated analysis, then a dedicated package might be called from the primary electrophysiological package. As each package requires specialist knowledge, it is easier to maintain its integrity and verifiability if the package is limited in scope, but it should contain a programming interface. This represents an efficient use of the resources available to the research community.

To reiterate, the term free software package is not a contradiction to commercialization. There is quite a lively commercial ecosystem around free software, successful examples are Redhat (providing services to open source software and having 1.3 billion revenue in 2012 (14)), Google (an advertising company based on open source software (15)), etc. To clarify this, a possible future scenario is that eventually a commercial ecosystem around free neuroimaging software will develop, and it will be for the benefit of patients, and other shareholders in the healthcare system.

Depending upon the progress of neuroimaging as a tool to obtain diagnostic information for patients (e.g., with stroke, Parkinson´s, and Alzheimer diseases, etc.) the algorithms tested in free packages might be incorporated into a commercial device. The added value to the device manufacturer is then not based on (secretive) algorithms, but rather can be based on device reliability, ease of maintenance, data privacy, and ease of use. In this scenario it might even be required that the algorithm used for the extraction and/or identification of a diagnostic parameter is documented together with results (e.g, coupling strength between two brain areas might have significant diagnostic value). But a reported coupling strength dramatically depends on the algorithm chosen to calculate it and this needs to be documented. In laboratory medicine procedures, hardware design is proprietary, but the machine is calibrated using a standard sample. Use of a standard sample does not carry-over to the software environment, and instead, an open documentation of algorithms is needed to achieve reproducibility and the necessary control. This then necessitates guidelines and policies to direct and parameterize how such packages can, should, and cannot be employed.

Conclusions

Free software packages have become cornerstones of neuroimaging research. Their development provides impressive example of international cooperation. The conclusions drawn here, that interoperability affords an enormous additional capability is probably important both for other fields of science, as well as for the development of regulatory codes that enable large-scale information and technology sharing and transfer (on an international scale).

If a method or algorithm published in a peer reviewed journal is included in a free software package it can serve as a positive open-access review. In addition to shared software packages, other tools for sharing of raw data ("open data") have been developed . Certain safeguard benefits might outweigh disadvantages and misuse (16). Yet, the ethical, legal, and social implications of large-scale open data (i.e., "Big Data") remain in question. That advances in software development, use, and sharing will contribute to the growth and scope of such Big Data projects is undeniable. What this means for the economics, laws and probity of science in society is yet to be determined.

Acknowledgments

We are indebted to the following persons for helpful discussions and the exact type of stimulating collaboration discussed in the article: Y. Bao (China), A. de Cheveigné (France), V. Litvak (Great Britain), R. Oostenveld (The Netherlands), H. van Rijn (The Netherlands), U. Steinhoff (Germany), and B. Zhou (China). Furthermore the organizers of the 10th Sino-German Workshop (Hamburg, September 2013), Prof. E. Pöppel and Prof. S. Han, as well as the organizers of the Biomedizinische Technik Konferenz 2013 (Graz, September 2013) are acknowledged for their support and the possibility to present results related to this article. A final thank you note goes to Prof. J. Giordano of Ludwig Maximilians Universität (Germany) and Georgetown University Medical Center (US) for suggesting this article.

Competing interests

The authors declare that they have no competing interests.

References

1. Valdés-Sosa PA, Kötter R, Friston KJ. Introduction: multimodal neuroimaging of brain connectivity. Phil Trans R Soc B. 2005; 360:865-867.
2. Sander TH, Knösche TR, Schlögl A, Kohl F, Wolters CH, Haueisen J, Trahms L. Recent advances in modeling and analysis of bioelectric and biomagnetic sources. Biomed Tech. 2010; 55:65-76.
3. Baillet S, Friston K, Oostenveld R. Academic software applications for electromagnetic brain mapping using MEG and EEG. Computational Intelligence and Neuroscience [Internet]. 2011: ID 972050. Available from: http://www.hindawi.com/journals/cin/2011/972050/.
4. Gewaltig M-O, et al. Research topic "Python in neuroscience". Frontiers in Neuroinformatics [Internet]. 2009. Available from: http://www.frontiersin.org/Neuroinformatics/researchtopics/Python_in_neuroscience/8.
5. Hanke M, Halchenko YO. The 2011 survey of software usage in neuroscience research – Supplementary results. NeuroDebian [Internet]. 2011 [cited 2013 November]. Available from: http://neuro.debian.net/survey/2011/results.html.
6. The GNU operating system: What is free software? [Internet]. 1984-2013 [cited 2013 November]. Available from: https://www.gnu.org/philosophy/free-sw.html.
7. Gramfort A, Luessi M, Larson E., Engemann D, Strohmeier D, Brodbeck CL, Parkkonen L, Hämäläinen M. MNE software for processing MEG and EEG data, NeuroImage. 2013; in press.
8. SPM [homepage on the Internet]. Available from: http://www.fil.ion.ucl.ac.uk/spm/.
9. FieldTrip [homepage on the Internet]. Available from: http:/www.ru.nl/donders/fieldtrip.
10. EEGLAB [homepage on the Internet]. Available from: http://sccn.ucsd.edu/eeglab/index.html.
11. BioSig [homepage on the Internet]. Available from: http://biosig.sourceforge.net/.
12. Brindley T, Giordano J. Neuroethical, Legal, and Social (NELS) Implications of ownership of neurotechnology within the milieu of increasingly globalized intellectual property protection. Poster presented at International Neuroethics Society 2013 Annual Meeting, Nov. 8, 2013, San Diego, CA.
13. Brindley T, Giordano J, Neuroimaging: correlation, validity, value and admissibility: Daubert – and reliability – revisited. AJOB Neurosci. in press.
14. Kerner SM. Red Hat grows business to $1.3 billion as OpenStack Cloud opportunity looms large. IT Business Edge Network. 2013; March 28. Available from: http://www.datamation.com/cloud-computing/red-hat-grows-business-to-1.3-billion-as-openstack-cloud-opportunity-looms-large.html.
15. Google Open Source Programs Office [homepage on the Internet]. Available from: https://developers.google.com/open-source.
16. Cartlidge E. Opening data up to scrutiny. Physics World. 2013; October:18-19.