EDUQUAKES

seismo-live: An Educational Online Library of Jupyter Notebooks for Seismology

by Lion Krischer, Yongki Andita Aiman, Timothy Bartholomaus, Stefanie Donner, Martin van Driel, Kenneth Duru, Kristina Garina, Kilian Gessele, Tomy Gunawan, Sarah Hable, Céline Hadziioannou, Mathijs Koymans, John Leeman, Fabian Lindner, Angel Ling, Tobias Megies, Ceri Nunn, Ashim Rijal, Johannes Salvermoser, Sujania Talavera Soza, Carl Tape, Taufiq Taufiqurrahman, David Vargas, Joachim Wassermann, Florian Wölfl, Mitch Williams, Stephanie Wollherr, and Heiner Igel

ABSTRACT

Efficient computer programming is becoming a central requirement in quantitative Earth science education. This applies not only to the early career stage but—due to the rapid evolution of programming paradigms—also throughout professional life. At universities, workshops, or any software training events, efficient practical programming exercises are hampered by the heterogeneity of hardware and software setups of participants. Jupyter notebooks offer an attractive solution by providing a platform-independent concept and allowing the combination of text-editing, program execution, and plotting. Here, we document a growing library with dozens of Jupyter notebooks for training in seismology. The library is made "live" through a server that allows accessing and running the notebooks in the browser on any system (PC, laptop, tablet, smartphone), provided there is internet access. The library *seismo-live* contains notebooks on many aspects of seismology, including data processing, computational seismology, and earthquake physics, as well as reproducible papers and graphics. It is a community effort and is intended to benefit from continuous interaction with seismologists around the world.

INTRODUCTION AND MOTIVATION

Discoveries in seismology are largely driven by increasingly sophisticated computer programming methods and the application of these methods to large amounts of data. A consequence is that every practitioner in the earthquake sciences has to be capable of reading, writing, and understanding code. Advanced programming courses are unfortunately still not part of every university's curriculum, resulting in a training gap hindering scientific progress. Projects like Software Carpentry (Achterberg *et al.*, 2017; see Data and Resources) have been created to aid in closing it and *seismo-live*, the topic of this manuscript, aims to improve that situation specifically for seismology.

seismo-live is a growing, open-source, online collection of executable Jupyter notebooks (Shen, 2014; Kluyver *et al.*, 2016) covering a wide range of seismological topics with the explicit purpose of offering in-depth, hands-on tutorials, and exercises covering theory as well as practical implementations in Python. Jupyter notebooks combine documentation with editable and interactive code. Aiken *et al.* (2018) recently demonstrated the rational behind using these notebooks to teach seismology.

The assembly of Jupyter notebooks was initiated in 2014 with the preparation of programming practicals for the annual Munich Winter School (see Data and Resources) focusing on seismic data processing using ObsPy (Beyreuther *et al.*, 2010; Megies *et al.*, 2011; Krischer, Megies, *et al.*, 2015) and on the various numerical approaches in computational seismology (Igel, 2016). Our success with Jupyter notebook training sessions kicked off the preparation of a number of workshop programming sessions (e.g., for the TIme DEpendent Seismology [TIDES] project funded by the European Science Foundation). This was followed up by dedicated seminars at the Ludwig-Maximilians-University Munich at the M.Sc. level, during which students were involved in developing new notebooks for *seismo-live*, solving scientific or technological problems (e.g., earthquake location using probabilistic methods, simple earthquake rupture problems, parallel programming), and documenting them with the powerful combination of text, graphs, and executable codes.

While *seismo-live* could incorporate tutorials in many other open-source programming languages, all currently existing notebooks utilize the Python programming language (see Data and Resources; Van Rossum, 2011). Python is an interpreted, cross-platform, highly expressive language, and in recent years its large ecosystem of high-quality scientific third-party packages have been established as one of the most widely used languages across all sciences (Perkel, 2015). Available packages in seismology include the widely used ObsPy library, MSNoise (Lecocq *et al.*, 2014), and many others (e.g., MacCarthy and Rowe, 2014; Krischer, Fichtner, *et al.*, 2015;

doi: 10.1785/0220180167

van Driel *et al.*, 2015; Chen and Holland, 2016; Chamberlain *et al.*, 2017; Durand *et al.*, 2018).

The notebooks and libraries used are platform independent and can be run on any operating system. An important appeal of *seismo-live* is that all notebooks are hosted online in the cloud and can be run without installation. This removes any initial hurdle or barrier to getting started, as the sole prerequisite is a working internet connection and a modern web browser with javascript enabled.

seismo-live's long-term ambition is to be a community driven and organized library of high-quality interactive programming tutorials and exercises that can be run anywhere. Its scope is seismology-education focused notebooks that demonstrate theoretical concepts, practical applications, tutorials for libraries, and reproducible figures of published papers.

PHILOSOPHY AND CONCEPT

seismo-live's main purpose is to serve as an educational tool for new generations of seismologists. We believe that seismology is a golden opportunity for *seismo-live* in that our basic data (time series) are openly and easily accessible to all. An emphasis of each notebook is thus the educational aspect which is met in a number of ways: (1) clear and comprehensive explanations of theoretical as well as practical aspects including formulas, citations, and further references where appropriate, (2) code examples with extensive comments that aim to be as simple and understandable as possible, (3) embedded exercises where some pieces are left out on purpose, (4) notebooks with reference solutions for these exercises, and (5) successive notebooks that build on each other and introduce increasingly difficult subjects that together make up full courses.

Many of these notebooks have been designed for short or multiday workshops and tutorials. Nonetheless, care is taken that each notebook can be understood and used for self-study. We anticipate that this will increase in the future and plan to accommodate any well-designed workshop material. A few workshops have already been taught using the hosted solution on the *seismo-live* website. Even in cases where a local installation of workshop material on participants' computers is desired, the hosted version can still serve as a fallback in case of broken installations. Fixing those onsite would otherwise reduce teaching time.

Reproducible papers are another building block of *seismo-live*. We collect Jupyter notebooks that reproduce figures or results of papers. The need for and the growing concern regarding the current lack of reproducible results is generally recognized (Nature-Editorial, 2016) but changing this is a daunting task. Interactive notebooks that can recreate full papers or parts thereof, and in which interested readers can study the implementation and experiment with the parameters, are an important step toward that objective.

LIBRARY CONTENT

In the following, we summarize the current content of the library, which is also shown in Table 1. In the interest of space we cannot present an outline of every currently available notebook. Notebooks are grouped by topic and a hierarchical list of all available notebooks is shown to users upon entering *seismo-live* (see Fig. 1). Some exemplary illustrations of *seismo-live*'s graphical output are shown in Figure 2.

General

Even though Python is becoming increasingly popular in many fields, experience shows that in many cases prior knowledge of Python programming is very heterogeneous. Therefore, *seismo-live* offers a "Python Crash Course" that introduces some of the fundamental concepts of Python. It introduces the most important functionalities for seismic applications: numbers, strings, dictionaries, lists, and functions. This is complemented by a short course on "data visualization" using the *matplotlib* library (Hunter, 2007) with seismological examples: line and scatter plots, plot combinations, annotations, and high-quality plots for publications.

Seismic Data Processing

The analysis of seismic data is at the heart of most seismological research. The Python-based ObsPy library now provides stable solutions for the entire workflow from (potentially massive) data download to parallel data processing using the recent parallel Adaptable Seismic Data Format (ASDF; Krischer *et al.*, 2016). *seismo-live* contains an extensive introduction into the functionalities of ObsPy. This includes an introduction to the zoo of seismic data formats, access to a large number of data providers (FDSN compatible webservices, ArcLink, SEEDLink, Earthworm, NCEDC, etc.), exercises on metadata associated with seismic data, as well as basic processing tasks (instrument correction, filtering,...) using ObsPy modules.

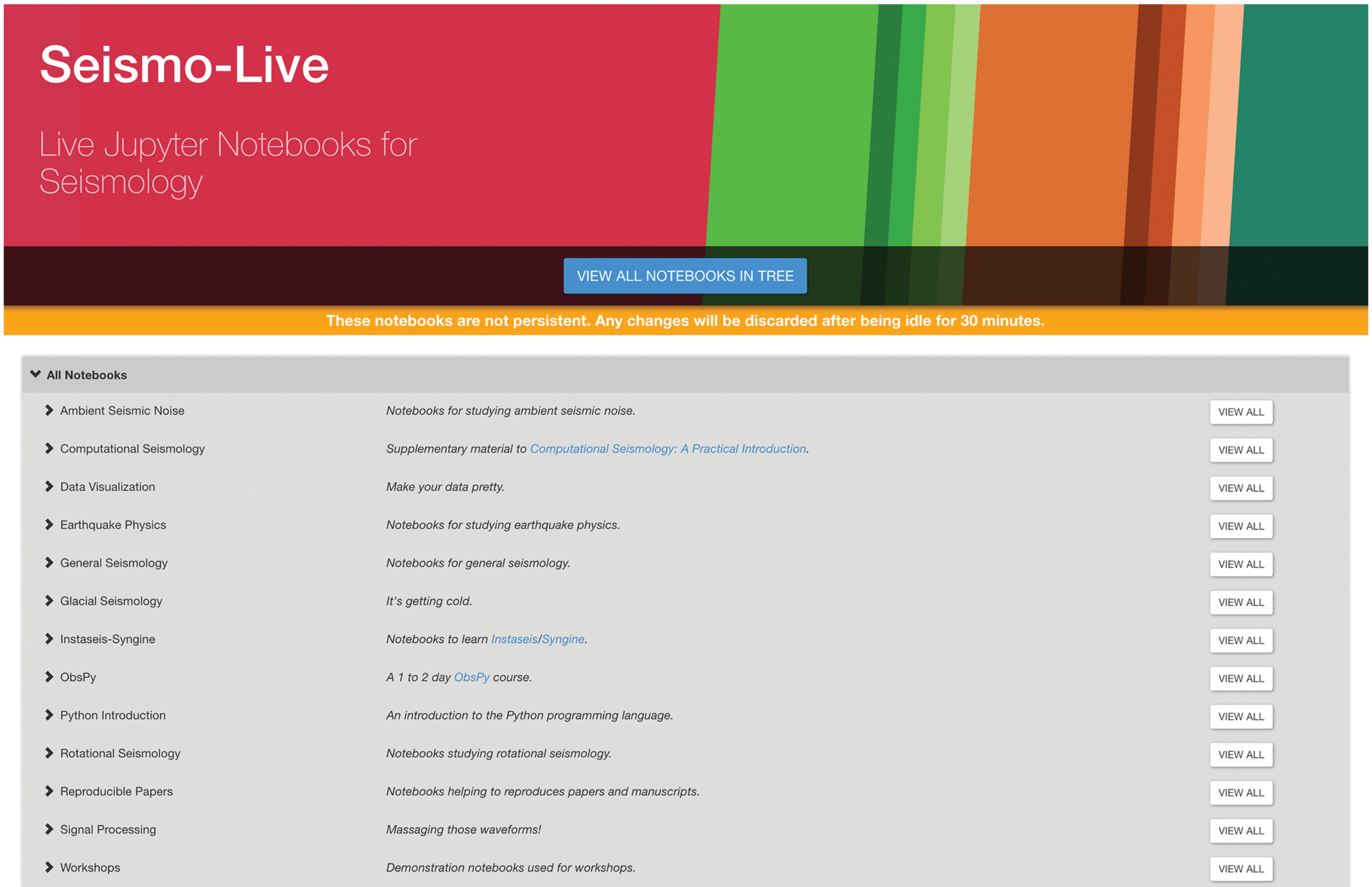
An essential ingredient in teaching seismic data processing is spectral analysis. *seismo-live* contains notebooks introducing the sampling theorem, windowing, tapering, the forward and inverse Fourier transforms, time–frequency analysis, and filtering, with specific applications to seismic data. These tools are used in the notebooks presenting ambient noise analysis, one of the most rapidly developing fields of seismology.

Rotational Seismology

Though still an exotic field of seismology, the analysis of joint translational and rotational motions, also called 6 degree-of-freedom (6 DoF), is gaining momentum (e.g., Igel *et al.*, 2015; Schmelzbach *et al.*, 2018). The 6 DoF concept implies that by combining translational and rotational motion observations at the same measurement point, one can obtain wavefield information (e.g., phase velocities, propagation directions) otherwise only accessible using seismic arrays. Also, a number of notebooks are demonstrating how to (additionally) download rotational ground motions (e.g., from ring lasers), and to make use of the powerful processing options deriving phase velocities, or back-azimuth information from point measurements. The processing steps trained with these notebooks are those automatically generating the output of the rotational event database (Salvermoser *et al.*, 2017).

Computational Seismology

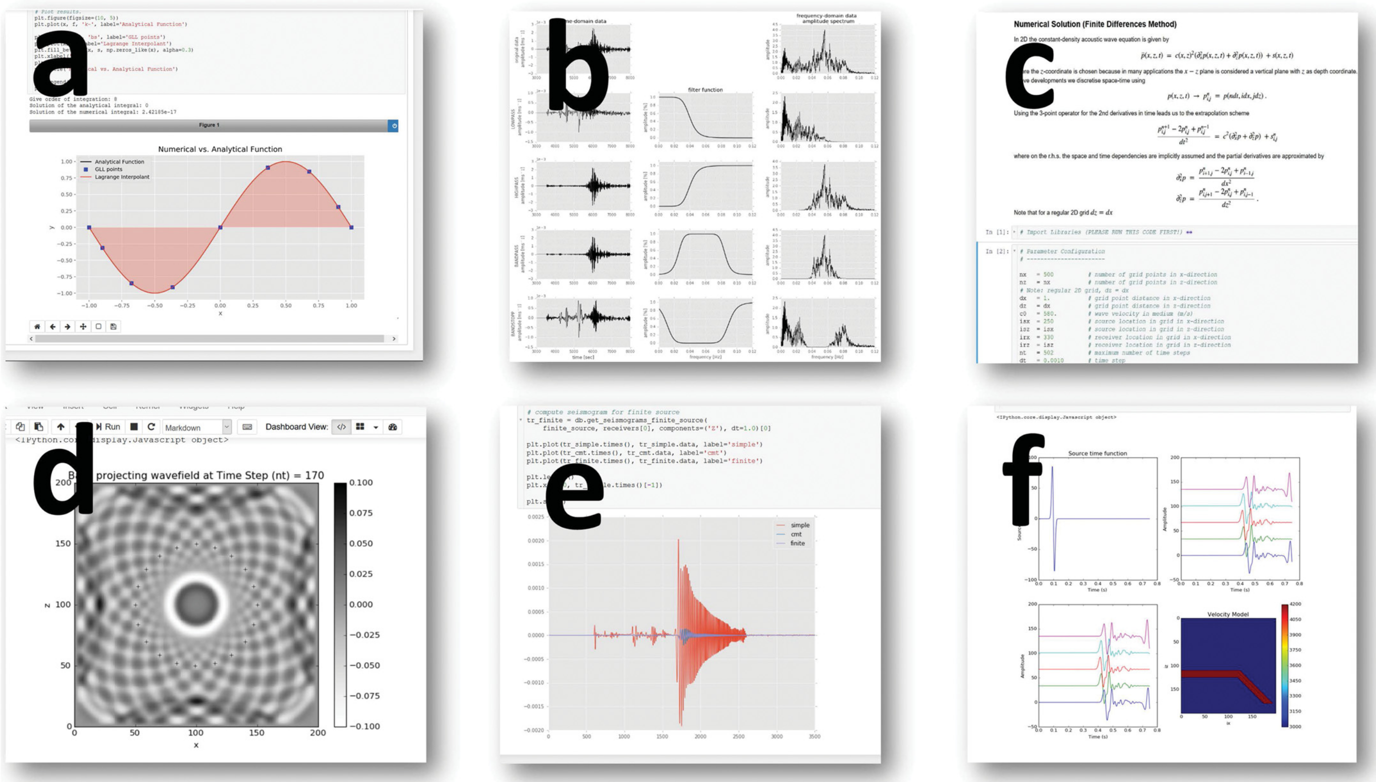
Many research projects in seismology use open-source software to calculate synthetic seismograms. This is often a sensible



▲ **Figure 1.** Screenshot of the hierarchical list of available notebooks shown to users upon entering *seismo-live*. Notebooks are organized by topic, and many notebooks exist in two variants: one with exercises, and one with the reference solution suitable for self-study.

Table 1
List of Notebooks Pof *seismo-live* at the Time of Writing

Topic	Summary	Notebook Count
Ambient Seismic Noise	Studying the ambient seismic noise field	1
Computational Seismology	Introduction to various techniques for numerical wave propagation	52
Data Visualization	How to plot date	1
Earthquake Physics	Rate-and-state friction	1
General Seismology	Cross-topic seismological application	1
Glacial Seismology	Seismology on glaciers	2
Instaseis-Syngine	Tutorials for Instaseis and Syngine	9
ObsPy	An in-depth multiday ObsPy workshop	17
Python Introduction	Python crash course	2
Rotational Seismology	Work with rotational data	3
Reproducible Papers	Notebooks to reproduce papers	7
Signal Processing	Filtering and other topics	6
Workshops	Notebook collections for workshops	19
		Total: 121



▲ **Figure 2.** Visual impressions of seismo-live Jupyter notebooks and output: (a) Gauss-Lobatto integration used in spectral-element methods; (b) illustration of various filter types on seismograms; (c) combination of equations and codes for numerical schema; (d) time reversal using the finite-difference method; (e) example of Instaseis seismograms; and (f) snapshots of seismograms for 2D acoustic wave propagation.

choice, because the software may handle challenging complexities (anisotropy, complex Earth geometries, …), and the software may have been carefully verified with analytical solutions based on simple Earth models. The drawback for users is that the theoretical concepts of these tools are often not well understood and therefore the potential traps are overlooked. To provide a simple introduction to computational methods in seismology, a substantial number of *seismo-live* notebooks present the simplest possible numerical solutions to seismic wave propagation problems (mostly in 1D), following the strategy presented in Igel (2016). This allows researchers to at least grasp the fundamental concepts and differences among the various available methods. These include the finite-difference and pseudospectral methods, including the generation of high-order operators, the concept of optimal operators, finite- and spectral-element methods, and the finite-volume and discontinuous Galerkin methods. Further examples are numerical solutions to the seismometer equation (damped harmonic oscillator), and the advection and diffusion–reaction equations.

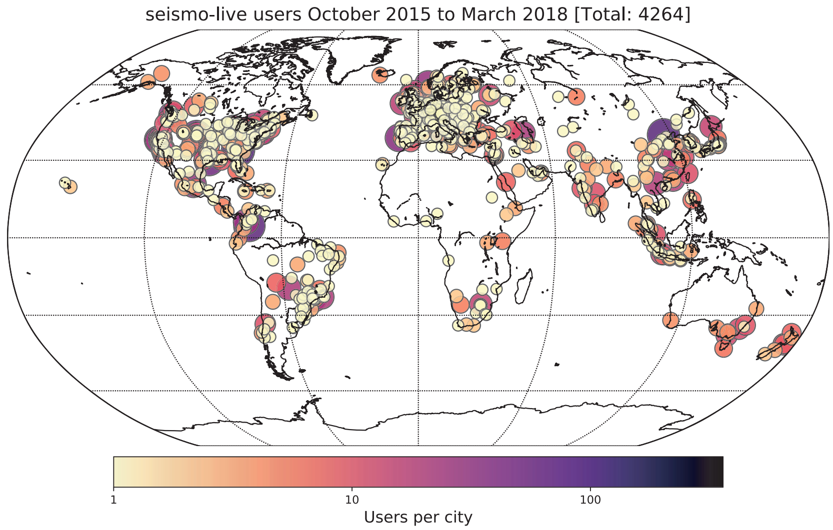
A number of analytical solutions are provided to allow proper verification and benchmarking of computational solutions. These include the acoustic wave equation (1D, 2D, and 3D), the full-space solution for a double-couple point source (Aki and Richards, 2002), and Lamb's problem (source in a half-space, Johnson, 1974). This is complemented by notebooks illustrating some fundamental properties of the seismic wave equations such as reciprocity (time reversal), and the illustration of numerical solutions as linear systems (i.e., pre/post filtering of numerical solutions based on the convolution theorem).

A tutorial is provided on the rapid (instantaneous) calculation of global synthetic seismograms in a spherically symmetric Earth model using AxiSEM/Instaseis/Syngine (Nissen-Meyer *et al.*, 2014; van Driel *et al.*, 2015; Krischer *et al.*, 2017). AxiSEM can generate spectral-element Green's function databases that can be used with Instaseis to very quickly generate synthetic seismograms at high frequencies without numerical calculations. Syngine is the corresponding webservice hosted by Incorporated Research Institutions for Seismology (IRIS; see Data and Resources). Examples are given in which real data are compared with Instaseis synthetic seismograms.

For the simulation of dynamic rupture, a rate-and-state friction toolbox is provided that enables modeling of frictional response to dynamic events such as velocity steps and time-dependent frictional healing.

Reproducible Research

There is a strong push from the science community to make scientific results and publications fully reproducible. Jupyter notebooks are excellent for this purpose. Therefore, we dedicate a section of the *seismo-live* project to reproducibility, and provide examples of notebooks that regenerate the results



▲ **Figure 3.** Unique users of *seismo-live* per city from October 2015 to March 2018, gathered with Google Analytics.

(graphics) of specific publications. This serves as an incentive to researchers to think "reproducible" during their projects and use Jupyter notebook (or similar) technology as a standard for preparing publications. Interested researchers are welcome to offer their research results as Jupyter notebooks.

Workshops

By now a number of workshops in seismology have been organized on a variety of topics with programming practicals based on Jupyter notebooks. In many cases, these workshops are focused on specific topics with accordingly adapted notebooks. To make these topical notebook collections available to the community we gather them under a specific *workshop* section. At the moment these include a course on specific webservices to access seismic data archives, as well as an ObsPy tutorial held at an IRIS co-organized workshop.

CONCLUSION AND PERSPECTIVE

The open-source, online, Python-based Jupyter notebook library *seismo-live* provides access to a substantial number of notebooks that can be used as (1) material for workshops and academic training, as well as (2) seeds for seismological research projects. The server-based online-executable opportunities removes the need (but does not exclude the option) to install the software locally on individual hardware. *seismo-live* is organized through an open-source GitHub project and is intended as a community effort that develops with the interaction and contributions of the seismological community. Early usage indicates global interest in the effort (Fig. 3). We suggest that Jupyter notebooks provide an excellent basis for collaborative and reproducible research projects.

Imminent extensions include notebooks on (1) probabilistic and linear(-ized) inverse problems with applications in seismology, (2) analysis of Earth's free oscillations, (3) processing tools for six-component processing for terrestrial and planetary seismology, and (4) computational solutions for earthquake physics.

A massive-open online course (MOOC) for the COURSERA platform (see Data and Resources) focusing on the teaching of numerical methods for the acoustic and elastic wave equation (called "Computers, Waves, Simulations") is in preparation and will be accessible in 2019.

DATA AND RESOURCES

All data and results reported in this article are accessible on the *seismo-live* website (www.seismo-live.org). At the time of writing, the *seismo-live* server is based on the *tmpnb* (https://github.com/jupyter/tmpnb) project, which internally uses light-weight Docker containers (https://docker.io). These technical details are prone to change over time as technologies evolve. Software Carpentry is available at https://software-carpentry.org; Munich Winter School website is availabe at www.geophysik.uni-muenchen.de/MESS; Python is available at https://python.org. Syngine is available at ds.iris.edu/ds/products/syngine/. A massive-open online course (MOOC) using Jupyter notebooks in the field of computational geophysics for the COURSERA platform is soon available at https://coursera.org. All websites were last accessed on September 2018.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the European Science Foundation for funding the TIme DEpendent Seismology (TIDES) project (coordinated by Andrea Morelli, University of Bologna). The idea for the *seismo-live* platform originated at the TIDES Meeting in Bertinoro 2015. The authors also gratefully acknowledge the support of the Leibniz Supercomputing Centre Munich (Anton Frank, Dieter Kranzlmüller) for their continuous support and the provision of computational resources. Additionally the authors are grateful for fruitful discussions and support in *seismo-live*'s early phase within the EU FP7 VERCE and the EU H2020 EPOS project. H. I. acknowledges support from the European Research Council (ERC-ADV Grant ROMY). The authors acknowledge the contribution of Lane Johnson providing his original code solving Lamb's problem.

REFERENCES

Achterberg, H., J. Adams, J. Adelman, J. Aranda, S. Bae, P. Barmby, E. Barr, D. Beitey, T. Bekolay, J. Berghold, *et al.* (2017). *Software Carpentry: Programming with Python*, available at https://zenodo.org/record/57492#.W5bJDPYyV08 (last accessed September 2018).

Aiken, J. M., C. Aiken, and F. Cotton (2018). A Python library for teaching computation to seismology students, *Seismol. Res. Lett.* **89**, no. 3, 1165–1171.

Aki, K., and P. Richards (2002). *Quantitative Seismology*, University Science Books, Sausalito, California.

Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann (2010). ObsPy: A Python toolbox for seismology, *Seismol. Res. Lett.* **81,** no. 3, 530.

Chamberlain, C. J., C. J. Hopp, C. M. Boese, E. Warren-Smith, D. Chambers, S. X. Chu, K. Michailos, and J. Townend (2017). Eqcorrscan: Repeating and near-repeating earthquake detection and analysis in python, *Seismol. Res. Lett.* **89,** no. 1, 173.

Chen, C., and A. A. Holland (2016). Phasepapy: A robust pure python package for automatic identification of seismic phases, *Seismol. Res. Lett.* **87,** no. 6, 1384.

Durand, S., R. Abreu, and C. Thomas (2018). Seistomopy: Fast visualization, comparison, and calculations in global tomographic models, *Seismol. Res. Lett.* **89,** no. 2A, 658.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment, *Comput. Sci. Eng.* **9,** no. 3, 90–95.

Igel, H. (2016). *Computational Seismology: A Practical Introduction*, Oxford University Press, Oxford, United Kingdom.

Igel, H., M. Bernauer, J. Wassermann, and K. U. Schreiber (2015). Rotational seismology: Theory, instrumentation, observations, applications, in *Encyclopedia of Complexity and Systems Science*, Springer-Verlag, New York, New York.

Johnson, L. R. (1974). Green's function for lamb's problem, *Geophys. J. Int.* **37,** no. 1, 99–131.

Kluyver, T., B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, and S. Corlay (2016). Jupyter notebooks —A publishing format for reproducible computational workflows, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt (Editors), IOS Press, Amsterdam, The Netherlands, 87–90.

Krischer, L., A. Fichtner, S. Zukauskaite, and H. Igel (2015). Large-scale seismic inversion framework, *Seismol. Res. Lett.* **86,** no. 4, 1198–1207.

Krischer, L., A. R. Hutko, M. van Driel, S. C. Stähler, M. Bahavar, C. Trabant, and T. Nissen-Meyer (2017). On-demand custom broadband synthetic seismograms, *Seismol. Res. Lett.* **88,** no. 4, 1127–1140.

Krischer, L., T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron, and J. Wassermann (2015). ObsPy: A bridge for seismology into the scientific Python ecosystem, *Comput. Sci. Discov.* **8,** no. 1, 14,003–14,020.

Krischer, L., J. Smith, W. Lei, M. Lefebvre, Y. Ruan, E. S. D. Andrade, N. Podhorszki, E. Bozda, and J. Tromp (2016). An adaptable seismic data format, *Geophys. J. Int.* **207,** no. 2, 1003–1011.

Lecocq, T., C. Caudron, and F. Brenguier (2014). MSNoise, a Python package for monitoring seismic velocity changes using ambient seismic noise, *Seismol. Res. Lett.* **85,** no. 3, 715–726.

MacCarthy, J. K., and C. A. Rowe (2014). Pisces: A practical seismological database library in Python, *Seismol. Res. Lett.* **85,** no. 4, 905–911.

Megies, T., M. Beyreuther, R. Barsch, L. Krischer, and J. Wassermann (2011). ObsPy—What can it do for data centers and observatories?, *Ann. Geophys.* **54,** no. 1, 47–58.

Nature-Editorial (2016). Reality check on reproducibility, *Nature* **533,** 437.

Nissen-Meyer, T., M van Driel, S. C. Stähler, K. Hosseini, S. Hempel, L. Auer, A. Colombi, and A. Fournier (2014). AxiSEM: Broadband 3-D seismic wavefields in axisymmetric media, *Solid Earth* **5,** no. 1, 425–445.

Perkel, J. M. (2015). Programming: Pick up Python, *Nature* **518,** no. 7537, 125–126.

Salvermoser, J., C. Hadziioannou, S. Hable, L. Krischer, B. Chow, J. Wassermann, U. Schreiber, A. Gebauer, and H. Igel (2017). An event database for rotational seismology, *Seismol. Res. Lett.* **88,** no. 3, 935–941.

Schmelzbach, C., S. Donner, H. Igel, D. Sollberger, T. Taufiqurrahman, F. Bernauer, M. Haeusler, C. V. Renterghem, J. Wassermann, and J. Robertsson (2018). Advances in 6-c seismology: Applications of combined translational 1 and rotational motion measurements in global and exploration seismology, *Geophysics* **83,** no. 3, WC53–WC69.

Shen, H. (2014). Interactive notebooks: Sharing the code, *Nature* **515,** no. 7525, 5–6.

van Driel, M., L. Krischer, S. C. Stähler, K. Hosseini, and T. Nissen-Meyer (2015). Instaseis: Instant global seismograms based on a broadband waveform database, *Solid Earth* **6,** no. 2, 701–717.

Van Rossum, G. (2011). *The Python Language Reference Manual*, Network Theory Ltd., Bristol, United Kingdom.

Lion Krischer[1]
Yongki Andita Aiman
Stefanie Donner[2]
Kenneth Duru
Kristina Garina
Kilian Gessele
Tomy Gunawan
Sarah Hable
Céline Hadziioannou[3]
Fabian Lindner[1]
Angel Ling
Tobias Megies
Ceri Nunn
Ashim Rijal
Johannes Salvermoser
Sujania Talavera Soza[4]
Taufiq Taufiqurrahman
David Vargas
Joachim Wassermann
Florian Wölfl
Mitch Williams
Stephanie Wollherr
Heiner Igel
Department of Earth and Environmental Sciences
Ludwig-Maximilians-University of Munich
Theresienstrasse 41
80333 Munich, Bavaria
Germany
lion.krischer@erdw.ethz.ch
yongki.aiman@gmail.com
stefanie.donner@bgr.de
kenneth.c.duru@gmail.com
kristinagarina@gmail.com
kilian@gessele.de
gunawantomy.pgr3@gmail.com
shable@geophysik.uni-muenchen.de
celine.hadziioannou@uni-hamburg.de
lindner@vaw.baug.ethz.ch
okling92@gmail.com
megies@geophysik.uni-muenchen.de
ceri.nunn@gmail.com
rijalashim@gmail.com
salv_johannes@gmx.de
sujaniaasereth@gmail.com
taufiqurrahman@hotmail.com
davofis123@gmail.com
joachim.wassermann@geophysik.uni-muenchen.de
flo.woelfl@web.de
mcbw00@gmail.com
wollherr@geophysik.uni-muenchen.de
heiner.igel@lmu.de

Timothy Bartholomaus
University of Idaho
tbartholomaus@uidaho.edu

Martin van Driel
Eidgenössische Technische Hoschschule Zürich
vandriel@erdw.ethz.ch

Mathijs Koymans
The Royal Netherlands Meteorological Institute (KNMI)
koymans@knmi.nl

John Leeman
Leeman Geophysical LLC
jleeman@ucar.edu

Carl Tape
University of Alaska, Fairbanks
ctape@alaska.edu

Published Online 26 September 2018

[1] Now at Eidgenössische Technische Hoschschule Zürich.
[2] Now at Bundesanstalten für Geowissenschaften und Rohstoffe, Hannover.
[3] Now at Center for Earth System Research and Sustainability (CEN), Universität Hamburg.
[4] Now at Utrecht University.