Madagascar - a powerful software package for multidimensional data analysis and reproducible computational experiments

Adrian D. Smith, Sergey Fomel*, Robert J. Ferguson

ABSTRACT

Reproducibility of published scientific findings is critical toward exposure of ideas and results to independent testing and replication by other scientists. Computational experiments are made readily reproducible in theory due to systematic characteristics of computer programs, but this proves more difficult in practice. Madagascar is a Unix-based open source software package that provides an environment for computational data analysis in geophysical and related fields. It incorporates functionality from pre-existing geophysical analysis libraries, and it allows the end user to completely package publications in a reproducible format using SCons and LaTeX. We present two simple computational examples illustrating the functionality of Madagascar. A local reconstruction of several figures from a published paper is given to highlight the power of Madagascar as a vehicle for generating reproducible research. Existing programs developed within CREWES can be incorporated into Madagascar's library. The installation of Madagascar on CREWES servers is highly recommended.

INTRODUCTION

The ability of peers to critically evaluate the findings of any investigation is an integral part of the scientific process. The scientific community strives to meet its basic responsibilities towards transparency, standardization, and data archiving (Hanson et al., 2011). The success and credibility of science are anchored in the willingness of scientists to expose their ideas and results to independent testing and replication by other scientists (Fomel and Claerbout, 2009). Replication is the ultimate standard by which scientific claims are measured (Peng, 2011). It allows independent researchers to address a scientific hypothesis and build evidence either for or against it. This traditional "culture of replication" has quickly weeded out spurious claims and enforced a disciplined approach to scientific discovery (Peng, 2011).

Science is driven by data, and new technology has vastly increased the ease of data collection and consequently the amount of and complexity of data collected (Hanson et al., 2011). Larger data sets have led to more computation as well as researchers in computationally oriented fields directly engaging in more science. Additionally, large available public databases have allowed for researchers to make scientific contributions without using the traditional tools of a given field (Peng, 2011). However, scientists are struggling with the huge amount, complexity, and variety of data (Hanson et al., 2011). The notion of replication is also made murkier by the advent of computational science (Peng, 2011).

*Bureau of Economic Geology, University of Texas at Austin

Reproducibility in Computational Science

The idea of a "replication by other scientists" in reference to computations is more commonly known as "reproducible research", coined by Jon Claerbout (Fomel and Claerbout, 2009). In the early 2000's, he and his students published a paper documenting their experience with creating and using a reproducible research environment (Schwab et al., 2000). According to Schwab et al. (2000), the need for a new environment stemmed from several issues experienced in the research lab. The primary issue was that researchers had issues reproducing their own computations without signficiant difficulty. Specifically, junior students building on the work of more advanced students frequently spent a considerable amount of time and effort just to reproduce their colleagues computational results (Schwab et al., 2000).

Minimum standards for assesing the value of scientific claims across the range of different disciplines associated with computational science have been called for by researchers (Yale Law School on Data and Code Sharing, 2012). The basic premise of a reproducibility standard is that every computational experiment has in theory a detailed log of every action taken by a computer (Peng, 2011). Generally, the standard of reproducibility calls for data and the computer codes used to analyze the data to be made available. However, this falls short of full replication becauase the same data are re-analyzed, rather than an analysis of indpendently collected data (Peng, 2011). This standard does allow though for limited exploration of the data and the analysis code, and aims to fill the gap in the scientific evidence-gathering process between full replication of a study and no replication (Figure 1), (Peng, 2011).

We now move on to a brief overview of "Madagascar", a software package designed to help meet a standard of reproducibility, specifically in the field of computational geophysics.

MADAGASCAR PACKAGE OVERVIEW

The Madagascar software package implements a computational environment that is designed both for conducting computational experiments in the area of large-scale geophysical analysis and for attaching links to software code and data in scientific publications in order to enable reproducible research (Fomel et al., 2013). At the time of writing, there are more than 120 scientific papers and book chapters complete with software codes necessary for verification and replication of computational results. The Madagascar project was started in 2003, version 1.0 being released in 2010 to the open community. Although the main applications have focused so far on exploration seismology in particular, the core package is suitable for other scientific fields requiring reproducible analysis of large-scale multidimensional data (Fomel et al., 2013).

The main Madagascar interface is the Unix shell command line, so a Unix/POSIX system or Unix emulator under Windows is required (Fomel and Hennenfent, 2007). The design of Madagascar follows the KISS Unix principles (Gancarz, 2003). Madagascar breaks the data analysis chain into multiple steps by writing short programs that implement individual steps. The programs act as filters by taking input from a disk file or from a

Unix pipe and writing either to disk or another pipe (Fomel et al., 2013). A universal data format called RSF (regularly sampled file) has been developed for use within Madagascar. The format is based on a text description that points to raw binary data stored in a separate file.[†] Although the majority of the programs currently in Madagascar focus on geophysical applications, users can use the API (application programmer's interface) for writing their own programs to manipulate RSF files. The primary language of Madagascar is C, but interfaces to other languages (C++, Fortran-77, Fortran-90, Python, MATLAB), are also available (Fomel and Hennenfent, 2007).

SCons and Reproducible Documents

The reproducible research system used by Madagascar is similar to that previously developed at the Stanford Exploration Project (SEP), which is based on "make" (Fomel and Hennenfent, 2007). In order to assemble data analysis workflows from individual programs, Madasgacar adopts SCons, a Python-based "make-like" utility (Knight, 2005). SCons configuration files (SConstruct) files are written in Python and specify the database of dependencies between input files, programs, and target files. Several advantages to using SCons include:

- SConstruct files are Python scripts, which are readable, simple, and powerful.
- SCons offers reliable, automatic, and extensible dependency analysis and creates a global view of all dependencies.
- SCons can detect changes not only in files, but also in commands used to build them.
- SCons is publicly released under a liberal open source license.

(Fomel and Hennenfent, 2007)

Within SCons, four specific commands are used to establish data-processing dependencies:

"Fetch" describes a rule for downloading data files from a remote data server or local data directory.

"Flow" describes a rule (command or Unix pipeline) for generating one or more target files from sources (none to many).

"Plot" is similar to "Flow", but the target file is a figure.

"Result" generates figures for inclusion in a publication.

(Fomel et al., 2013)

[†]A Guide to the RSF file format is available at www.ahay.org/wiki/Guide_to_RSF_file_format

The Madagascar environment can be thought of as existing on three different levels that correspond to different stages of research activities of a computational scientist (Figure 2), (Fomel et al., 2013). The uppermost level, level III, uses SCons to simplify creation of documents with results from workflows in level II. Customized SCons scripts create documents from LaTeX sources with output either in PDF or HTML format (Fomel and Hennenfent, 2007). An entire document can be packaged nicely into a single book or paper directory that contains all of the neccessary scripts needed to generate it (Figure 3).

MADAGASCAR EXAMPLES

In this section, two simple experiments are conducted to show different functions available at levels I and II of the Madagascar software architecture (Figure 2). Afterwards, SCons and Madagascar are use to reproduce figures from a published paper to demonstrate the abilities of Madagascar at the documentation level.

Image Processing

This first example is based upon a tutorial given by Fomel and Hennenfent (2007), using simple imaging processes to gain an understanding of how to navigate within Madagascar and generate a basic processing flow using SCons and an SConstruct file. A greyscale image is converted from JPEG format to RSF format on input, and random noise is added. An FFT is taken in the time domain (y-axis) on both the original and noisy image and the output FX spectra are plotted (Figure 4). While this is a simple example, it illustates the relative ease at which simple experiments can be conducted in Madagascar with SCons.

Velocity Model Building / Modelling

This example is based upon a tutorial made available by Kyle Shalek and Dr. Jeff Daniels at the Ohio State University. We generate synthetic velocity and density models manually in our SConstruct file (Figure 5 for the velocity model) and use them to run a 2D FD acoustic model (Figure 6). This example allows us insight into more advanced programs available in Madagascar, in particular seismic modelling utilities.

Full Geophysical Papers

This section demonstrates the full power of Madagascar and Scons as a method for packaging published research in a reproducible format. The particular examples shown here come from a paper published in *Geophysics* by Fomel et al. in 2007 entitled *"Post-stack velocity analysis by separation and imaging of seismic diffractions"* (Figures 7, 8, and 9). The files used to generate the paper are included in the Madagasar installation in the book/jsg/diffr directory within the source directory. Similar to the organizational structure described previously (Figures 2 and 3), the .tex, .bib, and top-level SConstruct file are located in the main folder. Three subfolders contain scripts that run the computations needed to generate figures for the paper. Provided permissions are set up correctly in order to be able to transfer the data from a remote server, one can re-create the entire paper by simply entering the command *scons* in the paper directory. This example highlights level

III (Figure 2) of the Madagascar package, specifically its documentation and publication features.

RECOMMENDATIONS

There are many benefits of using Madagascar to generate reproducible research. Those that have been presented here only scratch the surface of what is possible with the software. For CREWES in particular, there are several specific benefits to installation and use of the package:

1. The ability of colleagues within CREWES to be able to more efficiently follow and use workflows (avoiding the issues cited by (Schwab et al., 2000) in replicating previous work).
2. Complete, reproducible packaging of CREWES reports, conference abstracts, graduate theses, and other publications for internal use and for sponsors of CREWES. Additionally, previous work could be archived more efficiently.
3. The opportunity to collaborate with other research consortia using Madagascar. For example, Dr. Sergey Fomel and Dr. Paul Sava are two primary developers and drivers of Madagascar, and research produced at their institutions — the University of Texas at Austin and the Colorado School of Mines is available in reproducible format through Madagascar.
4. Existing software developed at CREWES (in particular the CREWES MATLAB package) can be incorporated within Madagascar as level I programs.

The source code for Madagascar is freely available at www.ahay.org. This website also serves as a primary source of information source for all things Madagascar. As the minimal dependency for installation is a C compiler and Python, it would be quite easy to install on CREWES servers. Other optional dependencies (such as the MATLAB API) are configured during the installation process using SCons (Fomel et al., 2013).

We recommend that CREWES install Madagascar on our Unix servers and it be used as a tool to further the reproducibility of research produced by CREWES.

ACKNOWLEDGEMENTS

We would like to thank CREWES and CREWES sponsors for supporting and funding this investigation.

REFERENCES

Fomel, S., and Claerbout, J. F., 2009, Reproducible research: Computing in Science & Engineering, **11**, No. 1, 5–7.

Fomel, S., and Hennenfent, G., 2007, Reproducible computational experiments using scons: 2007 International Conference on Acoustics, Speech and Signal Processing, **4**, 1257–1260.

Fomel, S., Landa, E., and Taner, M. T., 2007, Poststack velocity analysis by separation and imaging of seismic diffractions: Geophysics, **72**, No. 6, U89–U94.

Fomel, S., Sava, P., Vlad, I., Liu, Y., and Bashkardin, V., 2013, Madagascar: open-source software project for multidimensional data analysis and reproducible computational experiments: Journal of open research software, **1**, No. 1, 1–4.

Gancarz, M., 2003, Linux and the unix philosophy: Elsevier Science.

Hanson, B., Sugden, A., and Alberts, B., 2011, Making data maximally available: Science, **331**, No. 6018, 649.

Knight, S., 2005, Building software with scons: Computing in Science & Engineering, **7**, No. 1, 79–88.

Peng, R. D., 2011, Reproducible research in computational science: Science, **334**, No. 6060, 1226–1227.

Schwab, M., Karrenbach, M., and Claerbout, J., 2000, Making scientific computations reproducible: Computing in Science & Engineering, **2**, No. 6, 61–67.

Yale Law School on Data and Code Sharing, 2012, Addressing the need for data and code sharing in computational science: Computing in Science & Engineering, **12**, No. 5, 8–12.

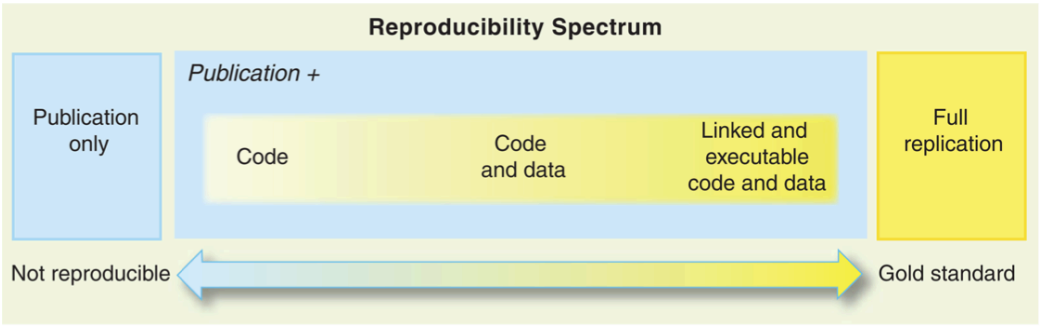

FIG. 1. Illustration of the concept of reproducibilty in computational research (Peng, 2011). The ultimate goal should be to make any research land as far to the right hand side of the spectrum as possible.

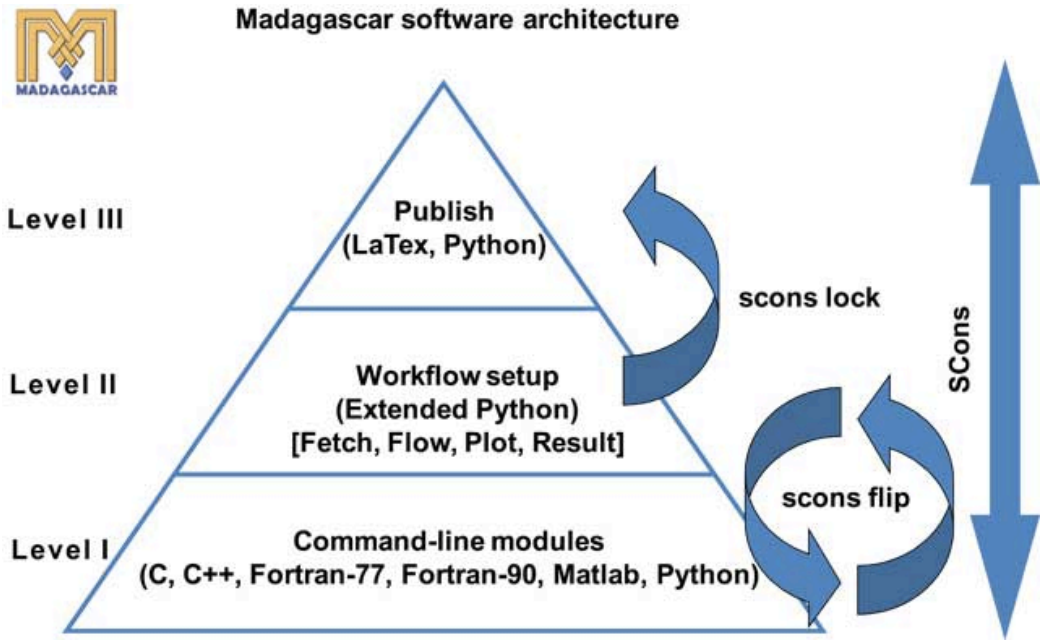

FIG. 2. Illustration of architecture of the Madagascar software package. The three levels are described as follows: (I) - Implementation of new computational algorithms for data analysis, involving writing low-level programs (II) - Testing of new algorithms or workflows on data by assembling workflows from existing command-line modules and tuning their parameters (III) - Documentation level. Results (figures) get referenced in the output publication (Fomel et al., 2013).

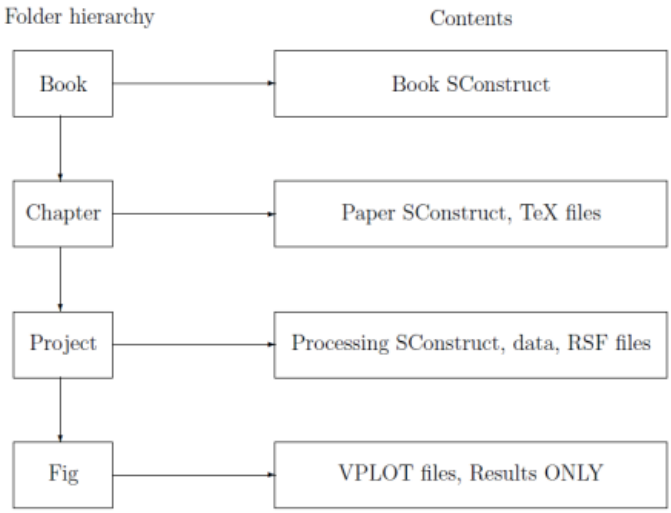

FIG. 3. Chart illustrating the organization of various file and folder locations used to generate a reproducible document. Image from: *www.ahay.org/wiki/Guide_to_RSF_file_format*

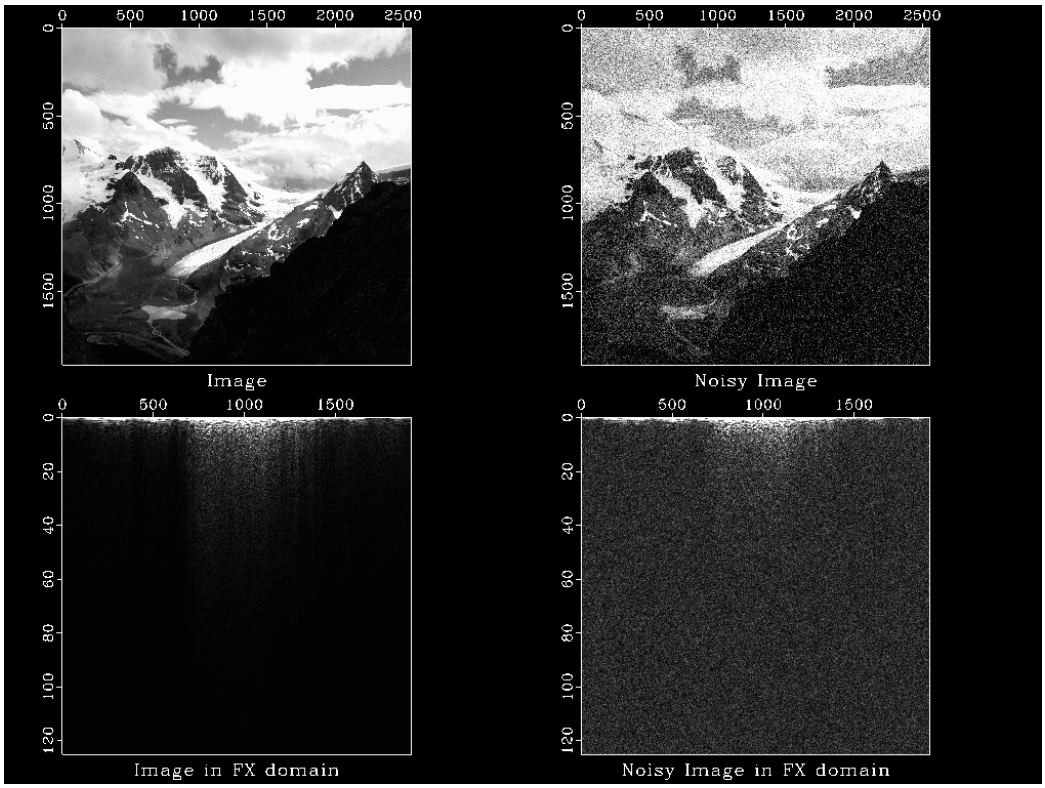

FIG. 4. Various images of the Athabasca glacier generated using a basic processing flow in an SConstruct file. The upper-left image is the original image, with its FX spectrum in the lower-left. The upper right is the original image with random noise added and the corresponding FX spectrum in the bottom-right image. The horizontal and vertical axes on the upper images represent pixel numbers. The horizontal axes on the lower images represent pixel numbers, with vertical axes of frequency in Hz.

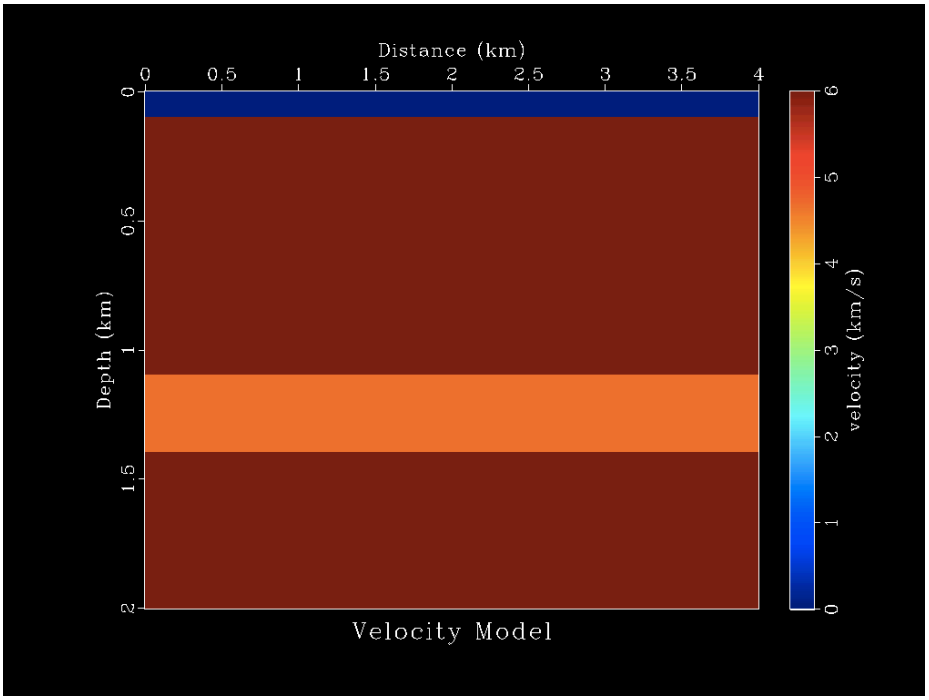

FIG. 5. Simple four layer velocity model generated in an SConstruct file. A low-velocity layer is located between 1.1 and 1.4 km depth.

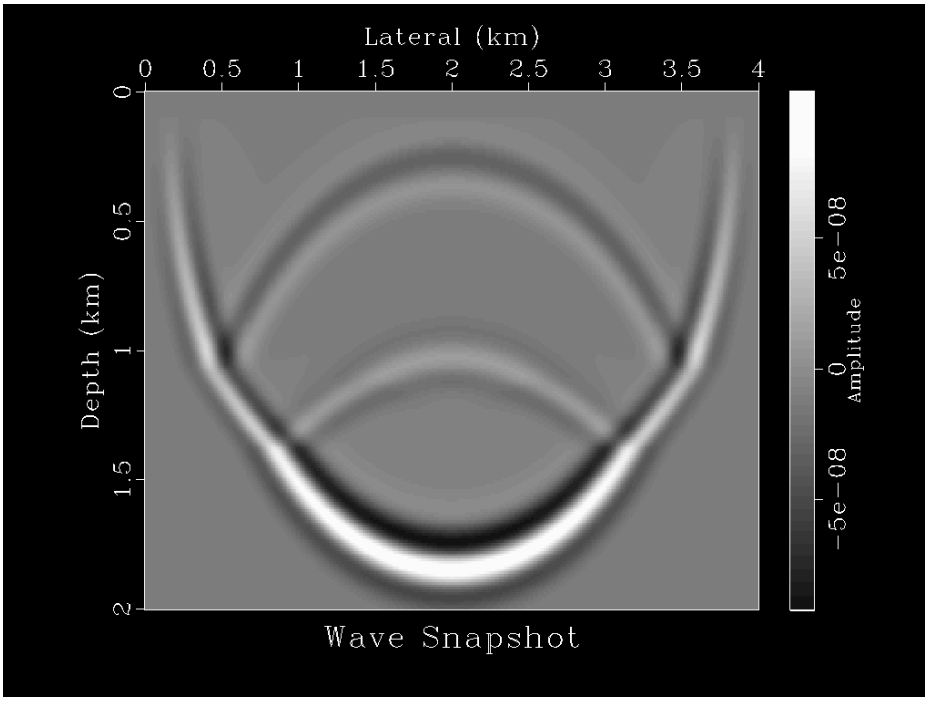

FIG. 6. Snapshot of a wavefield modelled using 2D FD acoustic forward modelling code available in Madagascar. The velocity model used is shown in Figure 5, with the source location at 2km lateral distance and 0km depth. The two reflections associated with the top and bottom of the low-velocity layer are quite visible.

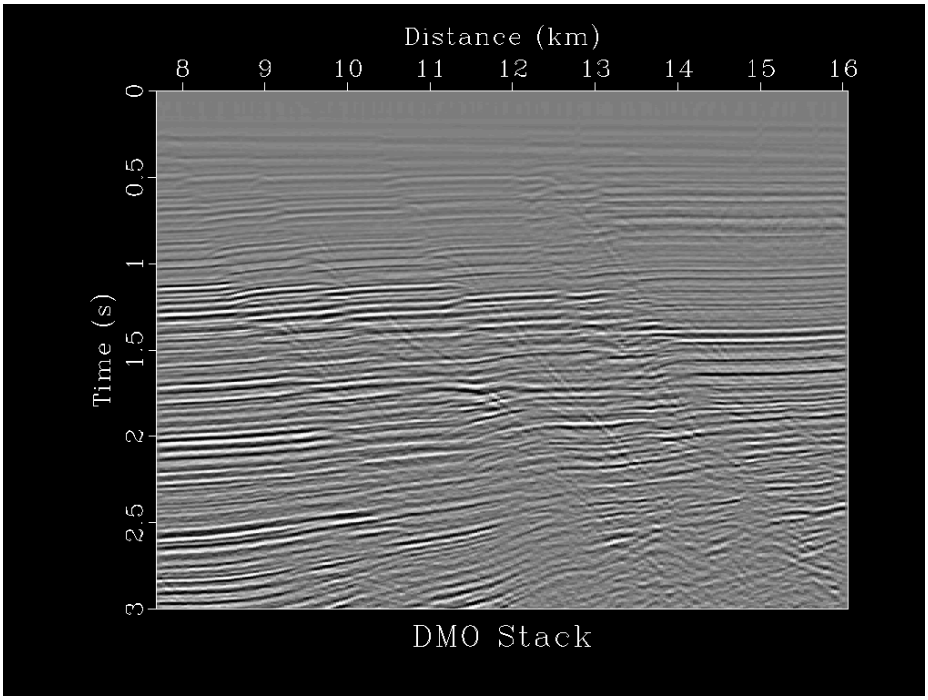

FIG. 7. Example of a reproducible figure from a published paper, generated on a local machine using codes in the Madagascar library. This particular figure can be found on page U-91 of Fomel et al. (2007).

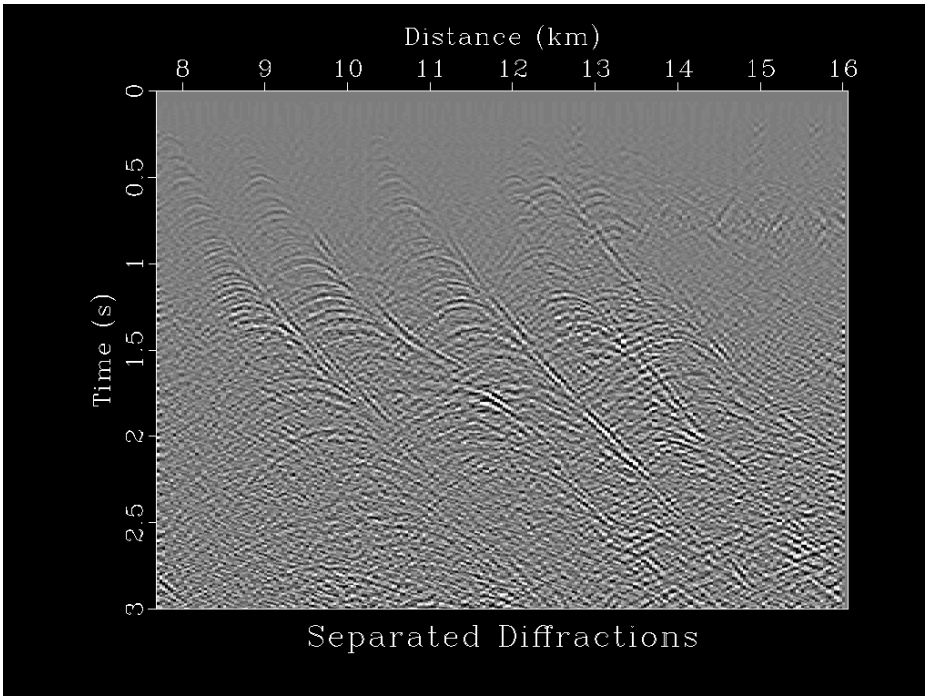

FIG. 8. Example of a reproducible figure from a published paper, generated on a local machine using codes in the Madagascar library. This particular figure can be found on page U-91 of Fomel et al. (2007).

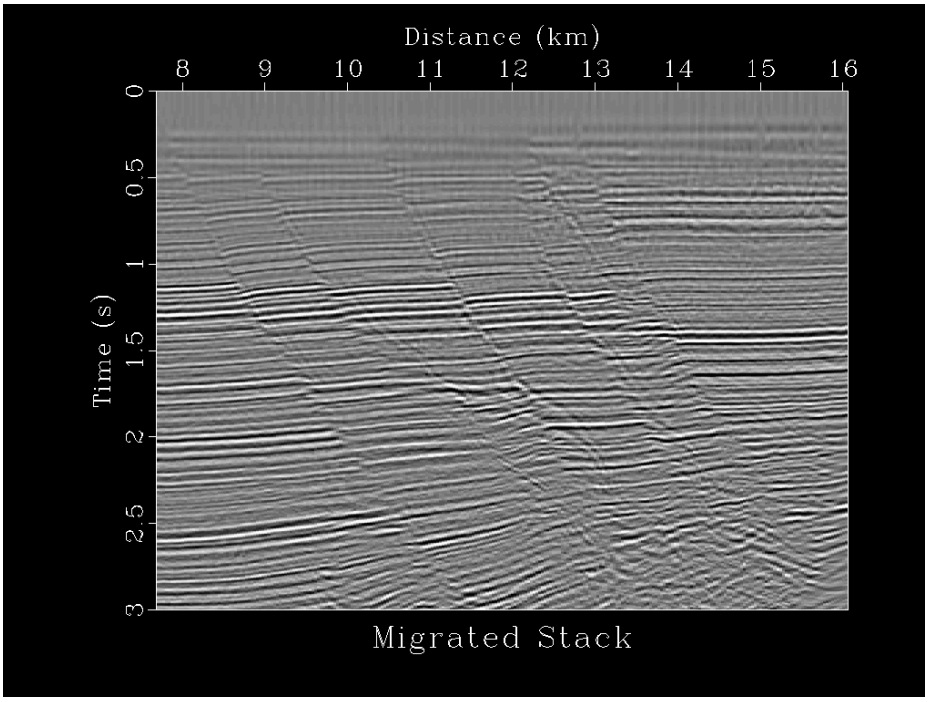

FIG. 9. Example of a reproducible figure from a published paper, generated on a local machine using codes in the Madagascar library. This particular figure can be found on page U-92 of Fomel et al. (2007).