

Introduction to Madagascar

Sergey Fomel

ABSTRACT

Madagascar is an open-source software package for geophysical data processing and reproducible numerical experiments. Its mission is to provide

1. a convenient and powerful environment
2. a convenient technology transfer tool

for researchers working with digital image and data processing. The package is available in an open-source form to allow effective collaboration of a wide community of developers. The technology developed using the Madagascar project management system is transferred in the form of recorded processing histories, which become “computational recipes” to be verified, exchanged, and modified by the users of the system.

WHAT IS MADAGASCAR?

Madagascar is an open-source software package for geophysical data processing and reproducible numerical experiments. Its most noticeable features are:

1. Madagascar is a relatively *new* package. It started in 2003, was developed entirely from scratch, and publicly released in 2006. Being a new package, it follows modern software engineering practices such as module encapsulation and test-driven development. A rapid development of a project of this scope (more than 300 main programs and more than 3,000 tests) would not be possible without standing on the shoulders of giants and learning from the 30 years of previous experience in open packages such as SEPlib and Seismic Unix. We have borrowed and reimplemented functionality and ideas from these other packages.
2. Madagascar is a *test-driven* package. Test-driven development is not only an agile software programming practice but also a way of bringing scientific foundation to geophysical research that involves numerical experiments. Bringing reproducibility and peer review, the backbone of any real science, to the field of computational geophysics is the main motivation for Madagascar development. The package consists of two levels: low-level main programs (typically developed in the C programming language and working as data filters) and high-level processing flows (described with the help of the Python programming language)

that combine main programs and completely document data processing histories for testing and reproducibility. Experience shows that high-level programming is easily mastered even by beginning students that have no previous programming experience.

3. Madagascar is an *open-source* package. It is distributed under the standard GPL open-source license, which places no restriction on the usage and modification of the code. Moreover, access to modifying the source repository is not controlled by one organization but shared equally among different developers. This enables an open collaboration among different groups spread all over the world, in the true spirit of the open source movement.
4. Madagascar uses a *simple, flexible, and universal* data format that can handle very large datasets but is not tied specifically to seismic data or data of any other particular kind. This “regularly sampled” format is borrowed from the traditional SEPlib. A universal data format allows us to share general-purpose data processing tools with scientists from other disciplines such as petroleum engineers working on large-scale reservoir simulations.

WHAT IS IN MADAGASCAR?

The package consists of

1. A collection of main programs. Most programs act as filters on input data and can be chained in a Unix pipeline. For example:

```
< data.rsfc sfwindow n1=100 | sfbandpass fhi=60 > data2.rsfc
```

This approach follows the Unix philosophy, as formulated by Doug McIlroy, the inventor of Unix pipes (Salus, 1994):

- (a) Write programs that do one thing and do it well.
- (b) Write programs to work together.
- (c) Write programs to handle text streams, because that is a universal interface.

Running a command (such as `sfwindow`) without parameters or the necessary input and output files shows a brief documentation, explaining the program purpose and parameters. Alternatively, brief documentation is provided by `sfdoc` program. Main program documentation in HTML format is available on the web at <http://www.reproducibility.org/RSF/>.

Madagascar uses *Regularly Sampled Format* (RSF) for data files, which is similar to the format used in the SEPlib library developed at the Stanford Exploration Project (SEP). The file format describes regularly sampled hypercubes. Up to

9 dimensions are supported. In accordance with the Unix philosophy, each RSF file (such as `data.rsf`) is a simple readable text. It contains a pointer (`in=` parameter) to the location of the binary data.

Madagascar provides programs for conversion to and from other formats such as SEG-Y and SU. Madagascar currently adopts Vplot file format, also developed at SEP, for generated graphics files.

2. An API (application programmer's interface) for programmers writing their own software to manipulate RSF files. The main software language of the Madagascar package is C. Interfaces to other languages (C++, Fortran-77, Fortran-90, Python, Matlab) are also provided.
3. A project management system. The system uses and extends SCons, an open-source software construction package, to document and maintain data processing flows. Documented projects become computational recipes that can be easily exchanged among Madagascar users.
4. A collection of reproducible documents, organized in living books. Each reproducible book contains a collection of Madagascar recipes (`SConstruct` files) used to generate book figures. The recipes cover a variety of data processing and imaging tasks described in the books. Figures and recipes serve dual purpose with respect to Madagascar maintenance. They provide demos for introducing new users to the functionality of the package and, at the same time, regression tests for assuring the system stability under change.

Follow the links at the end of this paper for additional documentation.

COPYRIGHT NOTICE

The Madagascar package is released in an open-source form under the standard GNU GPL license. In simple words, there are no restrictions on the use of the software (including copying, modifying, selling, etc.) However, there are restrictions on the software redistribution intended to prevent the package from losing its open-source status. Users are encouraged to submit their modifications back to the original distribution to the benefit of the whole user community.

ALTERNATIVES

In the present form, the Madagascar package, while being completely written from scratch, borrows ideas from the design of SEPLib, a publicly available software package, maintained by Bob Clapp at the Stanford Exploration Project. Generations of SEP students and researchers contributed to SEPLib. Most important contributions came from Rob Clayton, Jon Claerbout, Dave Hale, Stew Levin, Rick Ottolini, Joe

Dellinger, Steve Cole, Dave Nichols, Martin Karrenbach, Biondo Biondi, and Bob Clapp.

Madagascar also borrows ideas from Seismic Unix (SU), a package maintained by John Stockwell at the Center for Wave Phenomenon at the Colorado School of Mines (Stockwell, 1997, 1999). Main contributors to SU included Einar Kjartansson, Shuki Ronen, Jack Cohen, Chris Liner, Dave Hale, and John Stockwell. SU is open-source software (distributed with BSD-style license) starting with release 40 (April 10, 2007).

Another option for a seismic processing system is Free USP. USP is a processing package originally developed by Amoco and released by BP. Another package, DDS (Data Dictionary System) was also released by BP.

OTHER DOCUMENTS

- Installation instructions
- Self-documentation reference for Madagascar programs
- A guide to Madagascar programs
- A guide to RSF file format
- A guide to Madagascar programming interface
- A guide to programming with Madagascar
- A tour of Madagascar software
- A guide to SCons interface for reproducible computations

REFERENCES

- Salus, P. H., 1994, A quarter-century of Unix: Addison-Wesley.
- Stockwell, J. W., 1997, Free software in education: A case study of CWP/SU: Seismic Unix: The Leading Edge, **16**, 1045–1049.
- , 1999, The CWP/SU: Seismic Un*x package: Computers and Geosciences, **25**, 415–419.