



Reproducible research in practice

MADAGASCAR software package

Sergey Fomel

**Jackson School of Geosciences
The University of Texas at Austin**

July 1, 2010

SciPy  **2010**

Python for Scientific Computing Conference
June 28- July 3 • Austin, Texas



Outline

Reproducible Research

MADAGASCAR Project

What is Science?



What is Science?

Science is the systematic enterprise of gathering knowledge about the universe and organizing and condensing that knowledge into testable laws and theories. The success and credibility of science are anchored in the willingness of scientists to **independent testing and replication** by other scientists. This requires the **complete and open exchange of data, procedures and materials**.
American Physical Society, What is Science?

What is Reproducible Research?

- ▶ Attaching software code and data to publications
- ▶ Communicating computational results to a skeptic

An article about computational science 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 figures. [Jon Buckheit and David Donoho, WaveLab](#)

Reproducible Research Discussions

▶ <http://www.reproducibleresearch.net>

ICASSP 2007

Berlin-6 2008

CiSE 2009

- ▶ Donoho et al.
- ▶ LeVeque
- ▶ Ping & Eckel
- ▶ Stodden

IEEE Signal Processing Magazine 2009

- ▶ Vandewalle et al.

Yale Roundtable 2009

NSF Archive Workshop 2010



Personal Experience



1991–2001 Jon F. Claerbout

- ▶ Stanford Exploration Project
- ▶ Generations of Ph.D. students
- ▶ The principal beneficiary is the author

2003–Present **MADAGASCAR package**

- ▶ Software code requires continuous maintenance
- ▶ Maintenance requires an open community

Outline

Reproducible Research

MADAGASCAR Project



<http://www.ahay.org/>

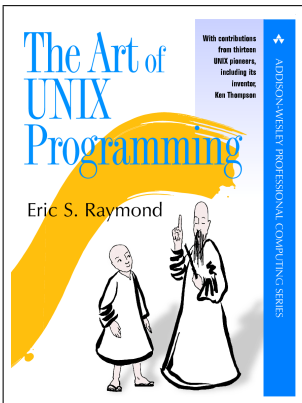
- ▶ Publicly released in 2006 (GPL)
- ▶ 1.0 release scheduled for July 2010
- ▶ School and Workshop in Houston on July 23-24, 2010
 - ▶ http://www.ahay.org/wiki/Houston_2010
- ▶ 25+ developers
- ▶ 250,000+ lines of code (20% Python)
- ▶ 10,000+ downloads from [SourceForge](#)
- ▶ 80 reproducible papers; 3,000 reproducible results
 - ▶ http://www.ahay.org/wiki/Reproducible_Documents



Thanks

- ▶ Vladimir Bashkardin, Jules Browaeys, William Burnett, Cody Brown, Maria Cameron, Lorenzo Casasanta, Joseph Dellinger, Jeff Godwin, Gilles Hennenfent, Trevor Irons, Jim Jennings, Long Jin, Roman Kazinnik, Siwei Li, Guochang Liu, Yang Liu, Doug McCowan, Henryk Modzelewski, Colin Russell, Paul Sava, Jeffrey Shragge, Xiaolei Song, Eduardo Filpo Silva, Ioan Vlad, Jia Yan, Lexing Ying

MADAGASCAR design



- ▶ Multidimensional arrays as file objects
- ▶ Simple universal file format
 - ▶ ASCII header file + data
- ▶ Filter programs to transfer files
 - ▶ C, C++, Fortran, Java, Matlab, Python
 - ▶ Combined with pipes and scripts
 - ▶ “Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.” Doug McIlroy

MADAGASCAR filter in Python

```
#!/usr/bin/env python

import numpy
import m8r

par = m8r.Par()
input  = m8r.Input()
output = m8r.Output()

n1 = input.int("n1") # trace length
n2 = input.size(1)  # number of traces

clip = par.float("clip")

trace = numpy.zeros(n1,'f')
for i2 in xrange(n2): # loop over traces
    input.read(trace)
    trace = numpy.clip(trace,-clip,clip)
    output.write(trace)
```

MADAGASCAR filter in C

```
#include <rsf.h>

int main(int argc, char* argv[])
{
    int n1, n2, i1, i2;
    float clip, *trace;
    sf_file in, out;

    sf_init(argc,argv);
    in = sf_input("in");
    out = sf_output("out");

    sf_histint(in,"n1",&n1); /* trace length */
    n2 = sf_leftsize(in,1); /* number of traces */
    if (!sf_getfloat("clip",&clip)) sf_error("Need clip=");

    trace = sf_floatalloc (n1);
    for (i2=0; i2 < n2; i2++) {
        sf_floatread(trace,n1,in);
        for (i1=0; i1 < n1; i1++) {
            if (trace[i1] > clip) trace[i1]= clip;
            else if (trace[i1] < -clip) trace[i1]=-clip;
        }
        sf_floatwrite(trace,n1,out);
    }

    exit(0);
}
```

MADAGASCAR script in Python

```
>>> import m8r
>>> spike = m8r.spike(n1=1000,n2=100)[0]
>>> spike
<m8r.File object at 0x4038b10>
>>> m8r.clip(clip=0.5)
<m8r.Filter object at 0x9976690>
>>> clipped = m8r.clip(clip=0.5)[spike]
>>> clipped2 = m8r.spike(n1=1000,n2=100).clip(clip=0.5)[0]
>>> import numpy
>>> clipped = numpy.clip(spike,-0.5,0.5)
```

```
bash$ sfspike n1=1000 n2=100 > spike.rsf
bash$ < spike.rsf sfclip clip=0.5 > clipped.rsf
bash$ sfspike n1=1000 n2=100 | sfclip clip=0.5 > clipped2.rsf
```

MADAGASCAR SConstruct script

```
from rsf.proj import Flow

Flow('spike', None, 'spike n1=1000 n2=100')
Flow('cliped', 'spike', 'clip clip=0.5')
```

```
bash$ scons
scons: Building targets ...
sfspike n1=1000 n2=100 > spike.rsf
< spike.rsf sfclip clip=0.5 > cliped.rsf
scons: Done building targets.
bash$ sed s/0.5/0.25/ < SConstruct > SConstruct2
bash$ mv SConstruct2 SConstruct
bash$ scons
scons: Building targets ...
< spike.rsf sfclip clip=0.25 > cliped.rsf
scons: Done building targets.
```

► <http://www.scons.org/>

Conclusions

- ▶ **Reproducible research**
 - ▶ **Attaching software and data to publications**
 - ▶ **Computational experiments communicated to a skeptic**
 - ▶ **Continuous maintenance requires an open community**
- ▶ **MADAGASCAR project**
 - ▶ **Practical implementation of reproducible research**
 - ▶ **Multidimensional arrays as file objects**
 - ▶ **Glued together by Python**

