

Sergey Fomel, The University of Texas at Austin



What is Madagascar?

Madagascar is an open-source software package for multidimensional data analysis and [reproducible](#) computational experiments. Its mission is to provide

- a convenient and powerful environment
- a convenient technology transfer tool for researchers working with digital image and data processing in geophysics and related fields.

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 users of the system.

In a Nutshell, Madagascar...

...has had 7,660 commits made by 57 contributors representing 571,724 lines of code
...is mostly written in C with an average number of source code comments

- has a well established, mature codebase
- maintained by a large development team
- with increasing year-over-year commits

...took an estimated 154 years of effort to build (COCOMO model) starting with its first commit in May, 2003 ending with its most recent commit about 20 hours ago



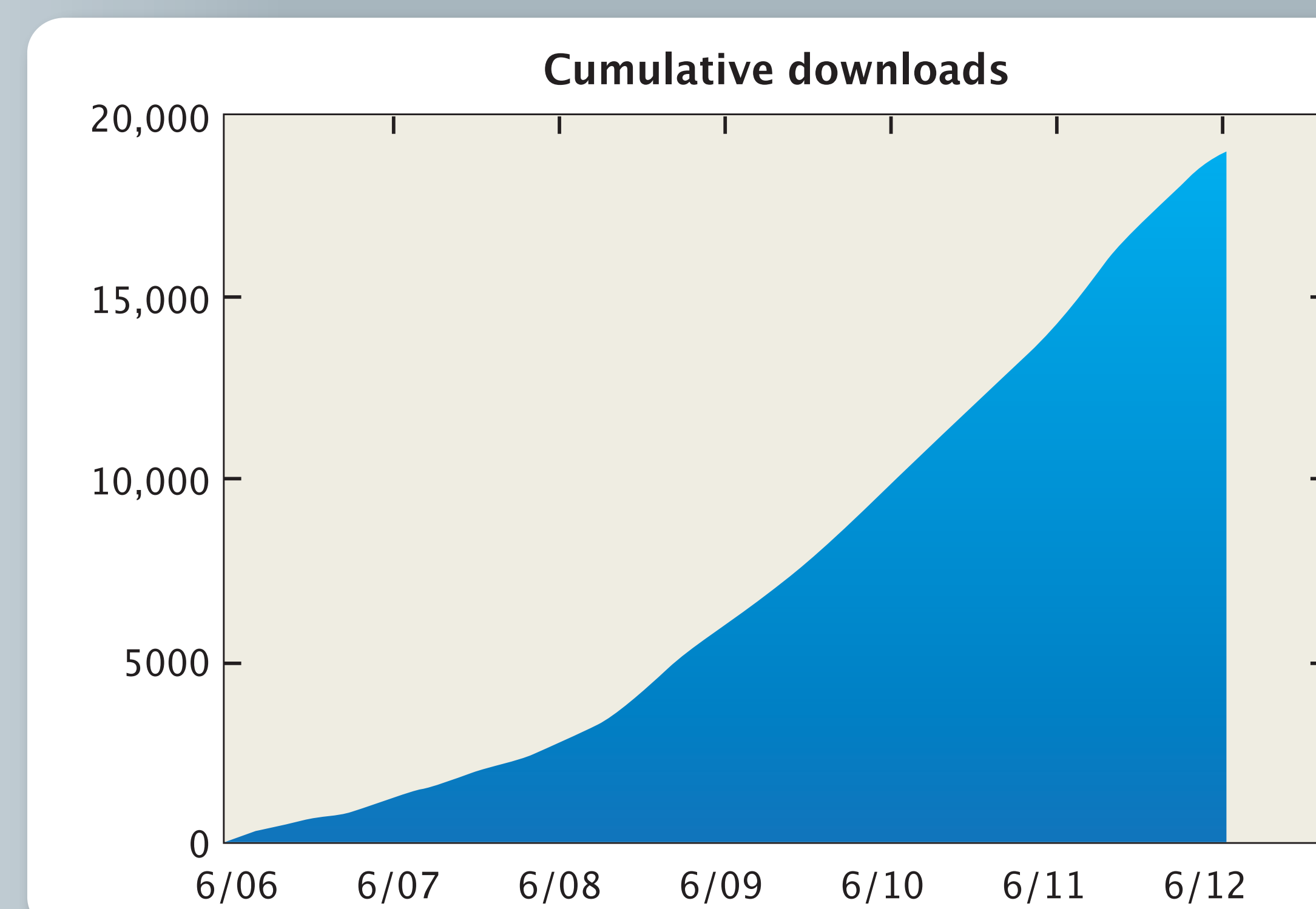
<http://www.ohloh.net/>

Who is Using Madagascar?

Since its first public release in 2006, Madagascar has been adopted by a number of users in industry and academia, mostly among researchers working with seismic imaging and geophysical data analysis. The Madagascar design is general and can be applied in principle to other applications that deal with analyzing large-scale multidimensional data.

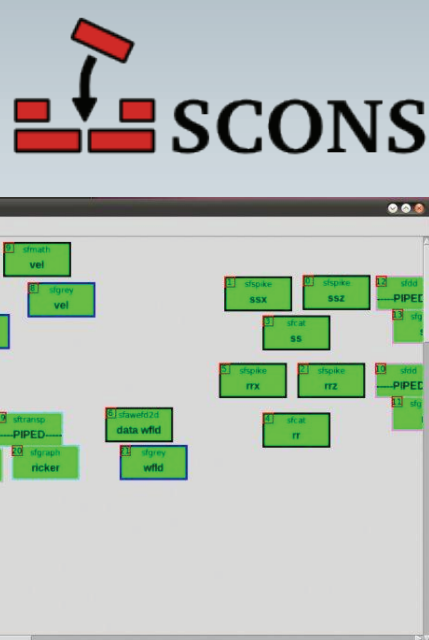
The current Madagascar distribution contains

- 932 filter programs (mostly written in C)
- 151 reproducible papers (LaTeX)
- 5,242 reproducible figures (Vplot)
- 503 computational recipes (Python/SCons)



How Does Madagascar use Python?

Madagascar uses Python mostly through SCons, a Python-based build system. Madagascar uses SCons for compiling software, managing data processing flows, and assembling [reproducible documents](#): scientific publications that link computational results with the data and software code that produce them. SCons configuration files (SConstruct scripts) serve as both computational recipes for users and regression tests for developers.

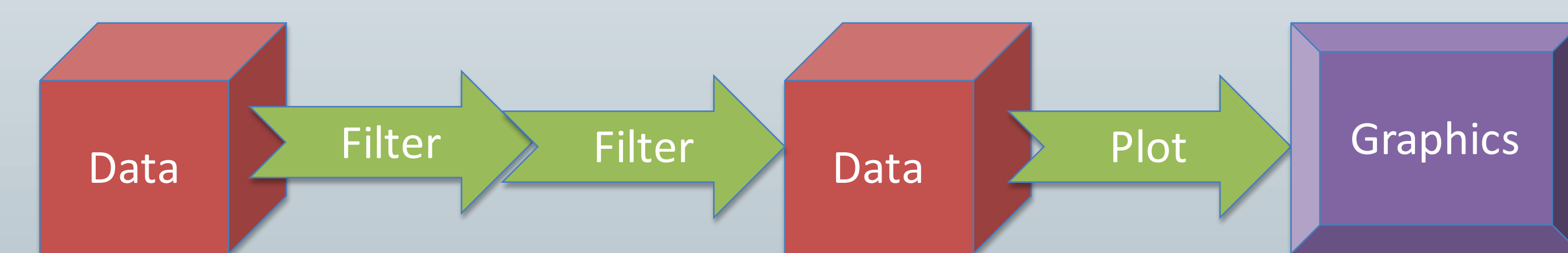


Language	Code Lines	Comment Lines	Comment ratio	Blank lines	Total lines	Total percentage
C	293,599	78,363	21.1%	71,645	443,607	54.2%
C++	121,193	21,787	15.2%	18,652	161,632	19.7%
Tex/LaTeX	111,957	10,162	8.3%	16,580	138,699	16.9%
Python	24,645	11,939	32.6%	6,948	43,532	5.3%
CUDA	8,902	3,057	25.6%	2,961	14,920	1.8%

tkMadagascar, a Graphical User Interface based on Tkinter provides a frontend for generating SConstruct files.

Python API

Madagascar data object is a multidimensional regularly sampled array stored on disk. The objects are manipulated by filter programs, which can be chained in Unix pipes. A particular kind of a filter is a plotting program which generates a graphics file instead of a data file. The goal of the Python API to Madagascar is to provide a high-level abstraction, which allows the user both to access filters and data files through a convenient interface and to write filter programs in Python.



Seismic Wedge Modeling Example

```
import m8r

# flat top of reservoir
top = m8r.math(n1=301,d1=1,o1=0,label1='Trace',
               output=0.06)[0]
# dipping bottom of reservoir
bot = m8r.math(output='input*(2-x/300)')[top]
# wedge model
wedge=m8r.cat(axis=2).unif2(n1=181,o1=0,d1=0.001,
                           unit1='s',label1='Time',
                           v00=[10,15,20])[top,bot]
# seismic convolution modeling
seis = m8r.ai2refl.ricker1(frequency=25)[wedge]
# plotting
seis.grey(title='Wedge Model',color='G')
wind = m8r.window(j2=10)[seis]
wind.wiggle(title='Wedge Model',
            poly=True,yreverse=True,transp=True)
```

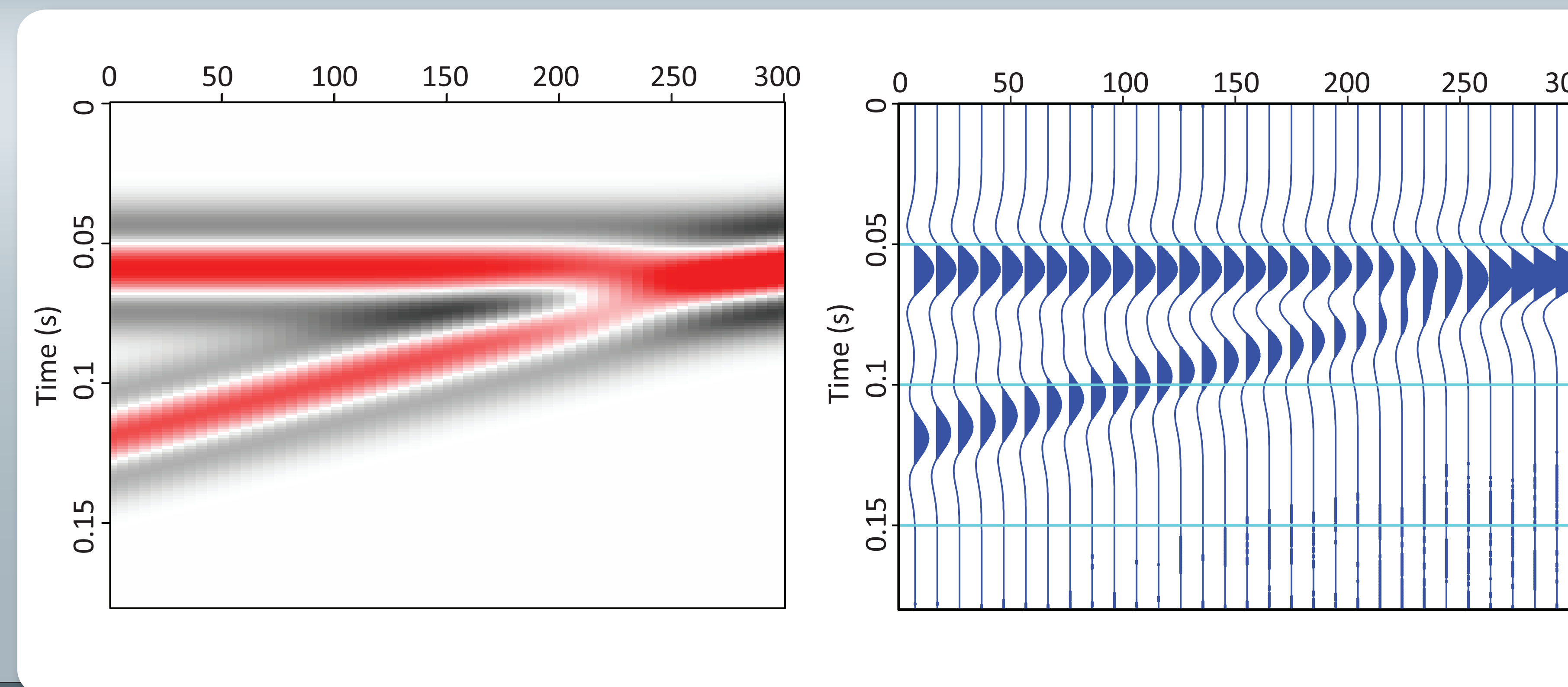


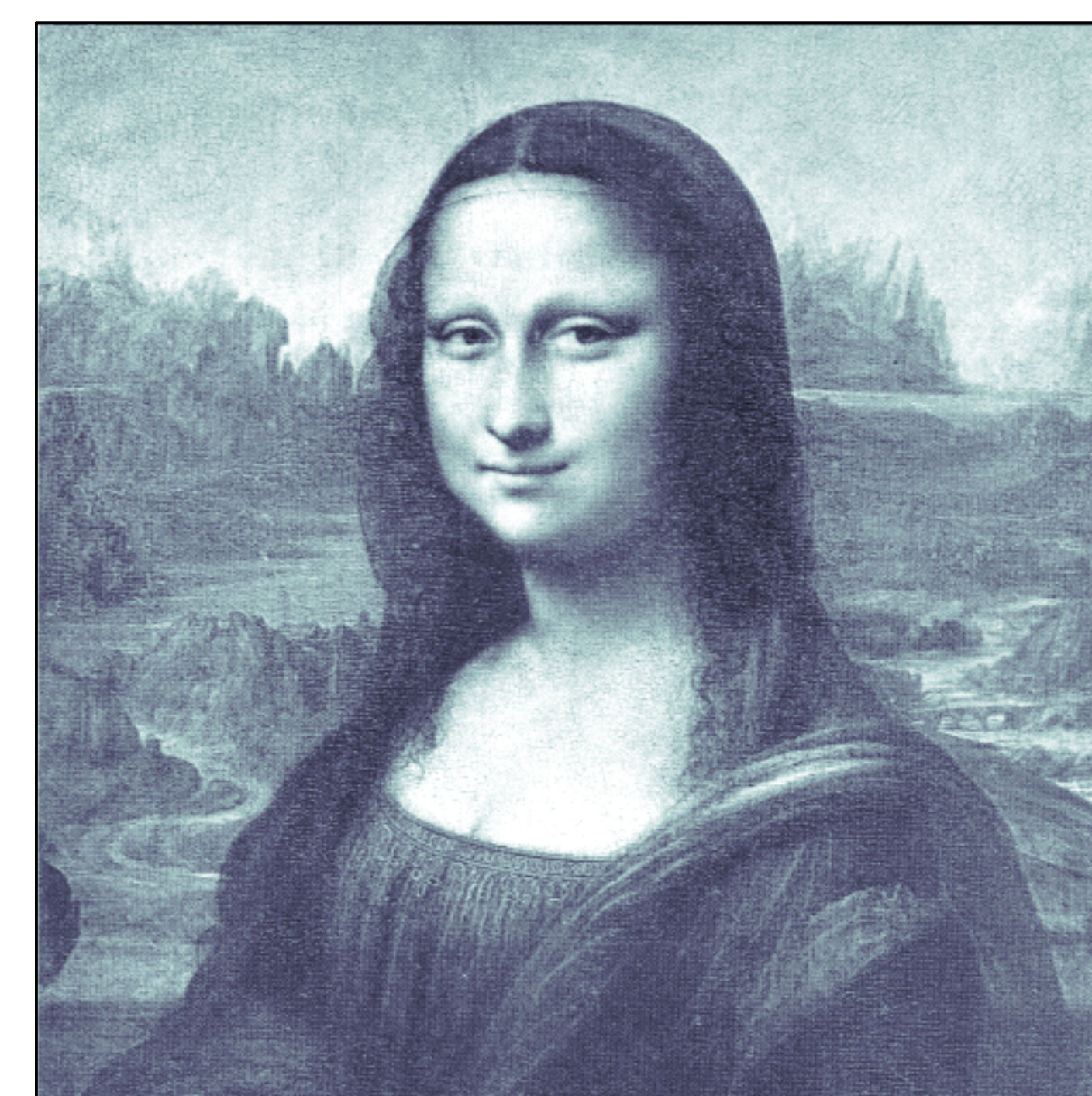
Image Analysis Example

```
import m8r

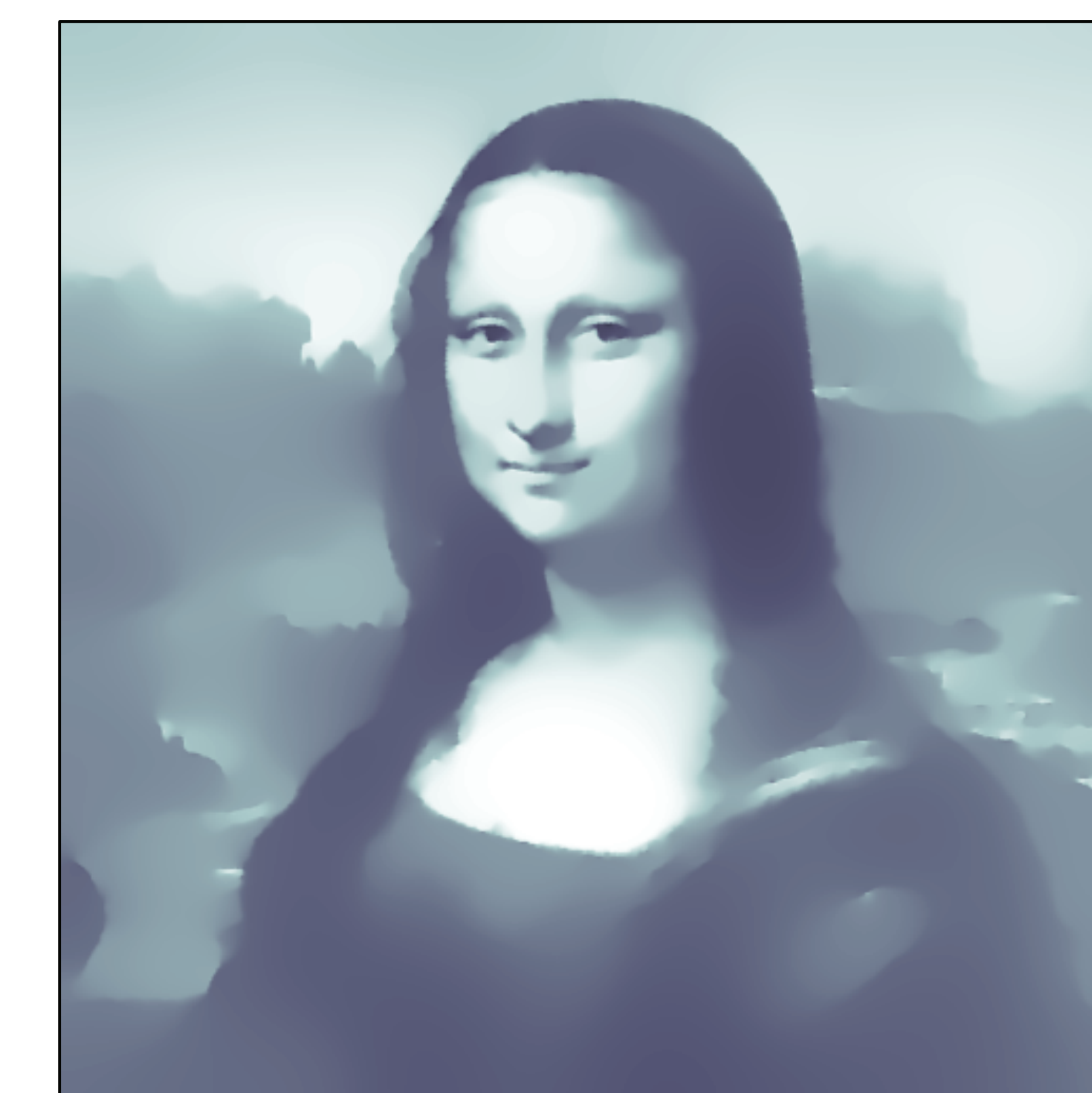
# Download data
m8r.Fetch('imgs','mona.img')
fmona=open('mona.txt','wt')
fmona.write("n1=512 n2=513 in=mona.img
           data_format=native_uchar\n")
fmona.close()

# Original image
mona = m8r.dd(type='float').transp['mona.txt']
mona.grey(title='Mona Lisa',
          allpos=1,color='b',screenratio=1,wantaxis=0)

# Edge preserving smoothing
mona2 = m8r.impl2(rect1=80,rect2=80,tau=1)[mona]
mona2.grey(title='Smoothed',
           allpos=1,color='b',screenratio=1,wantaxis=0)
```



Mona Lisa



Smoothed

Conjugate Gradients Algorithm

```
import m8r

random = m8r.noise(rep=1,type=0)

def dottest(oper,mod,dat):
    'Dot product test'
    mod = random[mod]
    dat = random[dat]
    print "L[m]*d=", oper(adj=0)[mod].dot(dat)
    print "L'[d]*m=", oper(adj=1)[dat].dot(mod)
```

```
def conjgrad(oper,dat,x0,niter):
    'Conjugate-gradient algorithm for min |Lx - d|^2'
    x = x0
    R = oper(adj=0)[x]-dat
    for iter in range(niter):
        g = oper(adj=1)[R]
        G = oper(adj=0)[g]
        gn = g.dot(g)
        print "iteration",iter+1,gn
        if 0==iter:
            s = g
            S = G
        else:
            alpha = gn/gnp
            s = g+s*alpha
            S = G+S*alpha
            gnp = gn
            beta = S.dot(S)
            alpha = -gn/beta
            x = x+s*alpha
            R = R+S*alpha
    return x
```

For More Information

To learn more about Madagascar, visit <http://www.ahay.org/> or attend the 2012 Madagascar School and Workshop in Austin, TX, on Friday-Saturday July 20-21.

For more information, see http://www.ahay.org/wiki/Austin_2012.



School and Workshop in Houston 2010



School and Workshop in Beijing 2011