Madagascar tutorial: Field data processing

Maurice the Aye-Aye

ABSTRACT

In this tutorial, you will learn about multiple attenuation using parabolic Radon transform (Hampson, 1986). You will first go through an example that explains the process step by step. You will be asked to change some parameters and add missing few lines. In the next part of the tutorial, you will be asked to apply the same workflow to another CMP gather. The CMP gathers used in the tutorial are from the Canterbury data set (Lu et al., 2003). By the end of this tutorial, you should have learned to:

1. apply NMO and inverse NMO for a CMP gather,
2. apply forward and inverse parabloic Radon transform,
3. design a mute function that preserves multiples in the Radon domain,
4. subtract multiples from the data,
5. create a semblance scan for a CMP gather.

PREREQUISITES

Completing this tutorial requires

- Madagascar software environment available from `http://www.ahay.org`
- LaTeX environment with `SEGTeX` available from `http://www.ahay.org/wiki/SEGTeX`

To do the assignment on your personal computer, you need to install the required environments. An Internet connection is required for access to the data repository.

The tutorial itself is available from the Madagascar repository by running

```
svn co https://rsf.svn.sourceforge.net/svnroot/rsf/trunk/book/rsf/school2012
```

INTRODUCTION

In this tutorial, you will be asked to run commands from the Unix shell (identified by `bash$`) and to edit files in a text editor. Different editors are available in a typical Unix environment (`vi`, `emacs`, `nedit`, etc.)

Your first assignment:

1. Open a Unix shell.
2. Change directory to the tutorial directory

   ```
   bash$ cd $RSFSRC/book/rsf/school2012
   ```

3. Open the `tutorial.tex` file in your favorite editor, for example by running

   ```
   bash$ nedit tutorial.tex &
   ```

4. Look at the first line in the file and change the author name from Maurice the Aye-Aye to your name (first things first).

DEMO

Part One

1. Change directory to `demo` directory

   ```
   bash$ cd demo
   ```

2. Run

   ```
   bash$ scons cmp.view
   ```

   in the Unix shell. A number of commands will appear in the shell followed by Figure 3(a) appearing on your screen.

3. To understand the commands, examine the script that generated them by opening the `SConstruct` file in a text editor. Notice that, instead of Shell commands, the script contains rules.
   - The first rule, `Fetch`, allows the script to download the input data file `cmp1.rsf` from the data server.
   - Other rules have the form `Flow(target,source,command)` for generating data files or `Plot` and `Result` for generating picture files.

- `Fetch`, `Flow`, `Plot`, and `Result` are defined in MADAGASCAR's `rsf.proj` package, which extends the functionality of SCons .

4. To better understand how rules translate into commands, run

   ```
   bash$ scons -c cmp.rsf
   ```

   The `-c` flag tells scons to remove the `cmp.rsf` file and all its dependencies.

5. Next, run

   ```
   bash$ scons -n cmp.rsf
   ```

   The `-n` flag tells scons not to run the command but simply to display it on the screen. Identify the lines in the `SConstruct` file that generate the output you see on the screen.

6. Run

   ```
   bash$ scons cmp.rsf
   ```

   Examine the file `cmp.rsf` both by opening it in a text editor and by running

   ```
   bash$ sfin cmp.rsf
   ```

   When you view cmp.rsf in the text editor, you see a history of all the programs used to create the file. The `sfin` program lists basic information about the file including data dimensions and extents of each axis.

Part Two

Figure 3(a) shows a CMP gather from Canterbury data set Line 12. The multiple energy appears at time around 2.25 s. Figure 1(b) shows the same gather after applying NMO correction with veloctiy equals to 1500 m/s. The multiple events starting at around 2.25 s and below are flatened while primary events , e.g at 2 s, are over corrected. The difference in move-out between the primaries and multiples, hence, can be used in Radon domain to attenuate multiple energy. Figure 2(a) is generated by forward parabolic Radon transform while Figure 1(d) is generated by inverse parabloic Radon transform. The purpose was to make sure that forward and inverse transforms do not cause any data loss.

Figure 2(a) shows the Radon transform of the CMP gather in Figure 3(a) while Figure 2(b) shows in the Radon domain the multiple energy only after mutting the primary energy. The protected multiples can be taken back to the time-offset domain and are subtracted from the data.

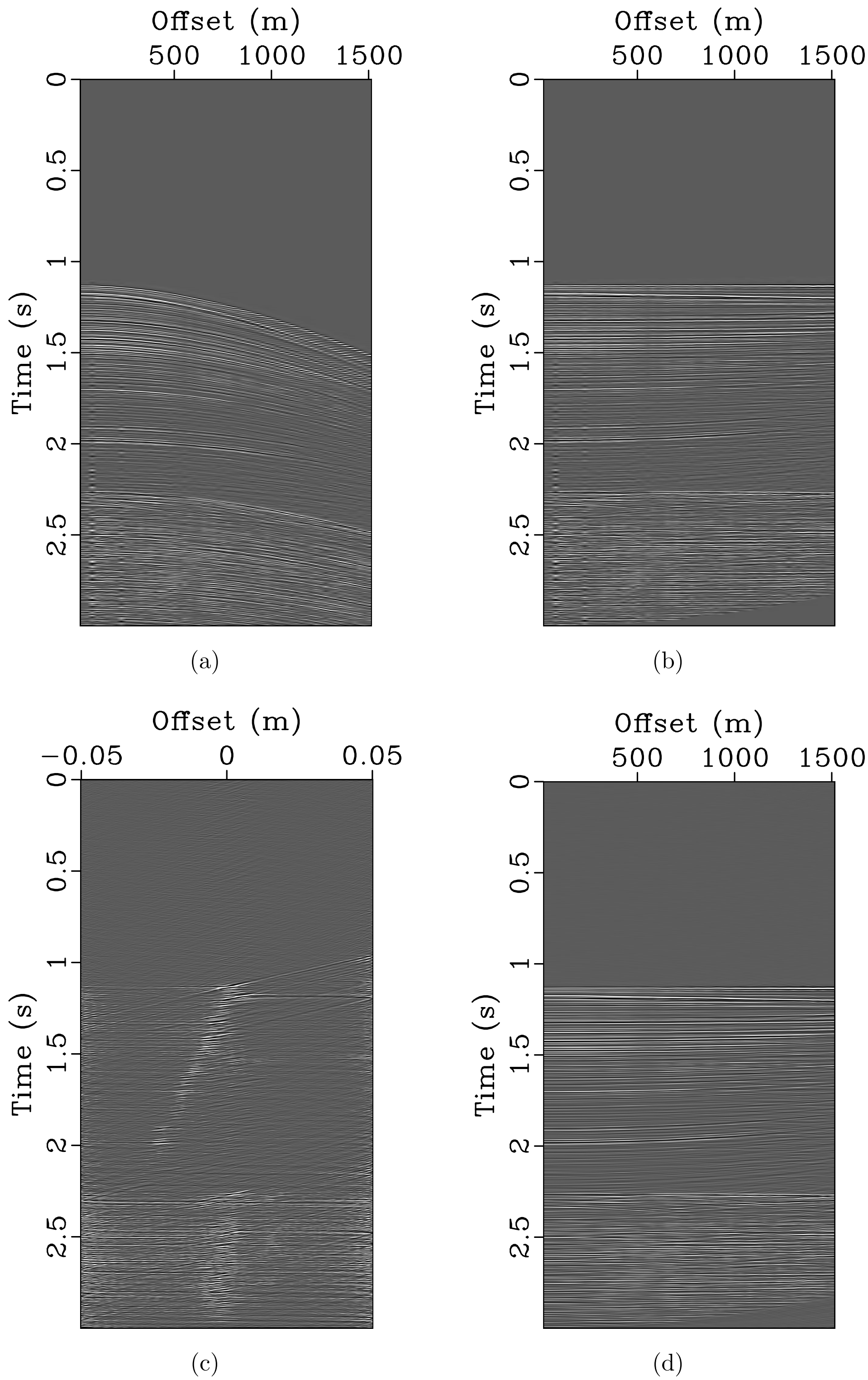
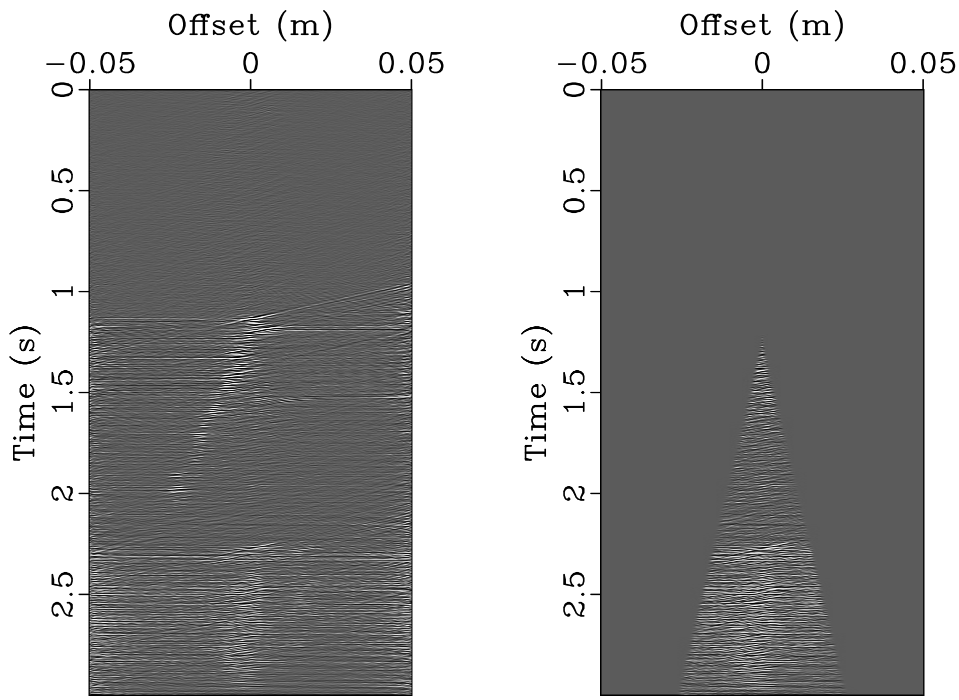


Figure 1: CMP gather from Canterbury dataset before applying NMO (a), after applying NMO (b), after Forward parabolic Radon transfrom (c), after applying inverse parabolic Radon transform (d). The forward and inverse parabolic Radon transforms are applied in sequence to examine the parameters of the process and to ensure that no events are lost during the process



(a) (b)

Figure 2: Forward Radon transform of the gather (a). Mute is applied to preserve multiples (b); so that multiples can be transformed to time-offset domain for subtraction from the CMP gather.

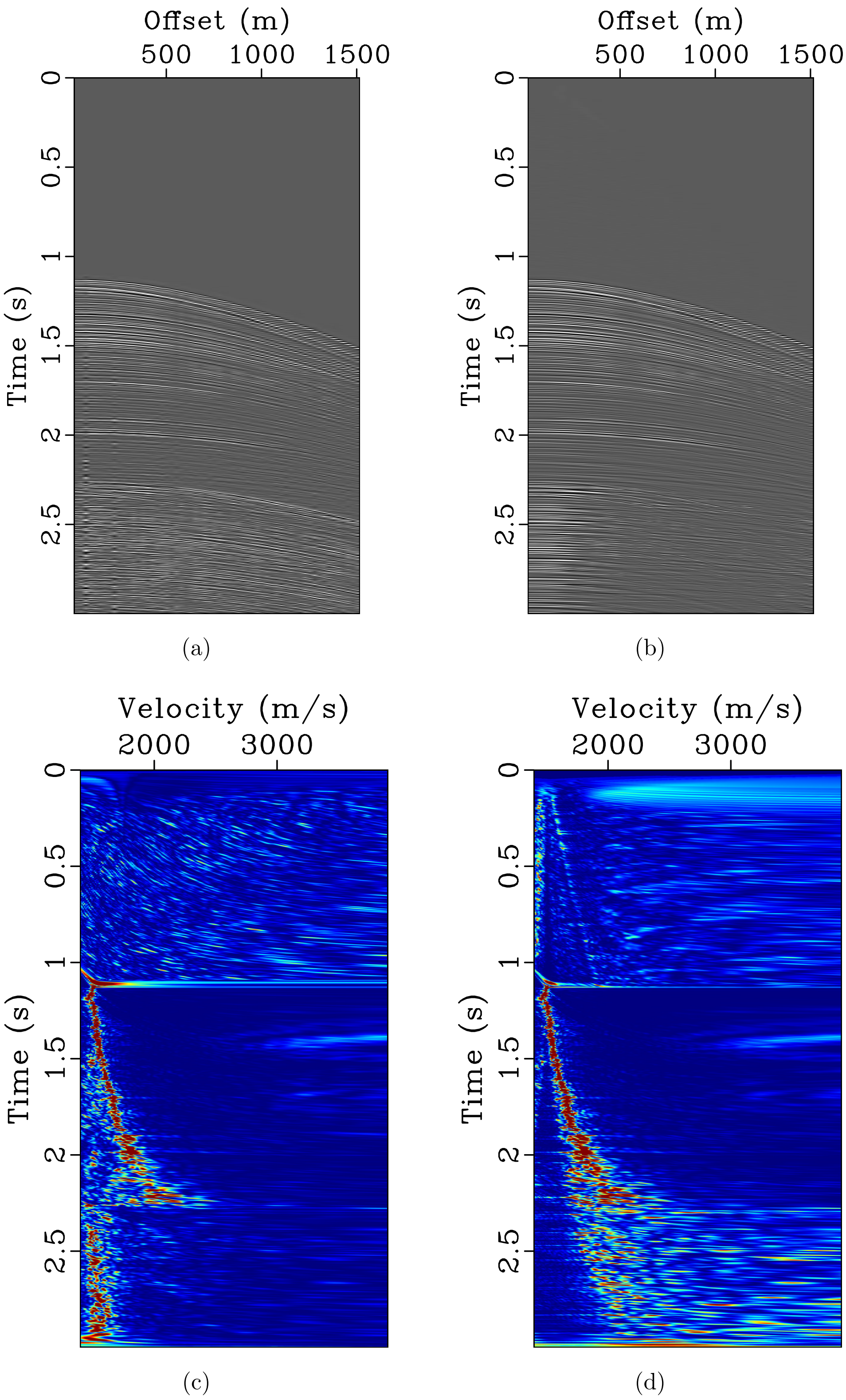


Figure 3: CMP gather before multiple attenuation (a). CMP gather after multiple attenuation (b). Gather in (a) is used to generated semblance scan in (c). Gather in (b) is used to generate semblance scan in (d).

CMP gather before multiple attenuation is shown in Figure 3(a) and the coresponding semblance scan is shown in Figure 3(c). The CMP gather after multiple attenuation is shown in Figure 3(b) and the coresponding semblance scan is shown in Figure 3(d). The semblance scans show how multiple energy is reduced for the CMP gather after multiple attenuation.

1. To examine the forward and inverse Radon transform, Run

   ```
   bash$ scons taup-qc.view
   ```

2. Edit the `SConstruct` file and find the line that says `CHANGE ME`, and modify the reference offset `x0` for `sfradon` program. To get more details about `sfradon` parameters, run

   ```
   bash$ sfradon
   ```

   Check your result by running

   ```
   scons taup-qc.view
   ```

3. Edit the `SConstruct` file and find the second `CHANGE ME`, and modify the starting time `t0` for `sfmutter`. To get more details about `sfmutter` parameters, run `sfmutter` in a Unix shell. Check your result by running

   ```
   scons taup-mult.view
   ```

4. Edit the `SConstruct` file and find the third `CHANGE ME`, and modify the parameter `v0` for `sfmutter`. Check your result by running

   ```
   scons taup-mult.view
   ```

5. Edit the `SConstruct` file and find the line that says ADD CODE to create `signal2.vpl`. To get more details about `sfgrey` parameters, run `sfgrey` in a Unix shell. Add your code and create the vpl file by running

   ```
   scons signal2.vpl
   ```

   Display the figure by running

   ```
   sfpen signal2.vpl
   ```

   `Hint`: the `SConstruct` file has similar code for creating the figure

6. Edit the `SConstruct` file and find the line that says ADD CODE to display `cmp.vpl` and `signal2.vpl`. Add your code and view the file by running

```
scons cmp-signal2.view
```

`Hint`: the `SConstruct` file has a similar example

7. Edit the `SConstruct` file and find the line that says ADD CODE to display `vscan-cmp.vpl` and `vscan-signal2.vpl`. Add your code and view the file by running

```
scons vcmp-signal2.view
```

`Hint`: the `SConstruct` file has a similar example

EXERCISE

In this part, your task is to apply the workflow explained above to a different CMP gather that requires different parameters. The same workflow should work here, but you need to observe that the CMP gather used for this exercise has shallow events. This means that, after applying NMO correction, amplitudes at far offstes of the shallow events get stretched. Therefore, an additional step is required for this CMP. We need to mute the distorted amplitudes. The mute is already applied in the `SConstruct`.

1. Change directory to `ex` directory

```
bash$ cd ../ex
```

2. Display the CMP gather after NMO with and without mute applied by running

```
scons nmo1-nmo.view
```

3. Your task is to add the necessary code to attenuate multiples for this CMP. The same work flow used in the `SConstruct` file under `demo` directory should work here with only changes to

   - `x0`
   - `t0`
   - `v0`
     where it says `CHANGE ME` in the comments

   `Hint`: You will need to copy part of `../demo/Sconstruct` to this `SConstruct` file.

WRITING A REPORT

1. Change directory to the parent directory

   ```
   bash$ cd ..
   ```

   This should be the directory that contains `tutorial.tex`.

2. Run

   ```
   bash$ sftour scons lock
   ```

   The `sftour` command visits all subdirectories and runs `scons lock`, which copies result files to a different location so that they do not get modified until further notice.

3. You can also run

   ```
   bash$ sftour scons -c
   ```

   to clean intermediate results.

4. Edit the file `paper.tex` to include your additional results. If you have not used LaTeX before, no worries. It is a descriptive language. Study the file, and it should become evident by example how to include figures.

5. Run

   ```
   bash$ scons tutorial.pdf
   ```

   and open `tutorial.pdf` with a PDF viewing program such as **Acrobat Reader**.

6. If you have LaTeX2HTML installed, you can also generate an HTML version of your paper by running

   ```
   bash$ scons tutorial.html
   ```

   and opening `tutorial_html/index.html` in a web browser.

```
from rsf.proj import *

# download cmp1.rsf from the server
Fetch('cmp1.rsf','cant12')

# convert to native format
Flow('cmp','cmp1','dd form=native')

```

```
# create cmp.vpl file
Plot('cmp','grey title=CMP ')

# water velocity 1500 m/s
wvel=1500

# NMO with water velocity
Flow('nmo','cmp','nmostretch half=n v0=%g'%wvel)

# create nmo.vpl
Plot('nmo','grey title=NMO')

# create cmp-nmo.vpl file under Fig directory
# cmp.vpl and nmo.vpl created earlier using Plot
# command will be ploted side by side
Result('cmp-nmo','cmp nmo','SideBySideAniso')

######################
# radon parameters
######################
ox=29.25
nx=60
dx=25
#-------------------------
x0=800   # CHANGE ME
#-------------------------
p0=-.05
dp=.0005
np=201

# forward Radon operator
radono='''
        radon  np=%d p0=%f dp=%f x0=%d parab=y
        ''' %(np,p0,dp,x0)

# inverse Radon operator
radonoinv='''
        radon  adj=n nx=%d ox=%g dx=%d x0=%d parab=y
        ''' %(nx,ox,dx,x0)

# Test radon parameters, apply forward and
# inverse Radon Transform, and QC results
##############################################
Flow('taup','nmo',radono)

```

```
# plot
Plot('taup','grey title=forward RT')

# Inverse
Flow('nmo2','taup',radonoinv)

# plot
Plot('nmo2','grey title=inverse RT')

# Display three figures to QC Radon parameters
# Check that forward and inverse Radon transforms
# do not change the data i.e events are preserved.

Result('taup-qc','nmo taup nmo2','SideBySideAniso')

#########################################
# design a mute function that protects
# multiples in the Radon domain
#########################################
#--------------------------------
t0=1.2   # CHANGE ME ; try 1.5
#--------------------------------
# vertical position of the triangle vertix

#---------------------------------
v0=.03   # CHANGE ME ; try .015
#---------------------------------
# slope of the triangle

Flow('taupmult','taup','mutter t0=%g v0=%g'%(t0,v0))
Plot('taupmult','grey title="multiples in Radon domain"')

# Display taup.vpl and taupmult.vpl
# This display allows a flip between
# the two figures
Result('taup-mult','taup taupmult',
        '''
        cat axis=3 ${SOURCES[1]}
        |grey
        ''')

# Transform mulitples from Radon domain to time-offset domain
Flow('multiple','taupmult',radonoinv)

# create multiple.vpl
```

```
Plot('multiple','grey title="multiples"')

# plot CMP and multiples side by side
Result('cmp-mult','nmo2 multiple','SideBySideAniso')

# Subtract multiples from the CMP
Flow('signal','multiple nmo2',
     '''
     add scale=-1,1 ${SOURCES[1]}
     ''')

# inverse NMO
Flow('signal2','signal',
     '''
     nmostretch inv=y half=n v0=%g
     | mutter v0=1900 x0=200
     '''%wvel)

#---------------------------------------
# ADD CODE to create signal2.vpl
#---------------------------------------


#-------------------------------------------------------
# ADD CODE to display cmp.vpl and signal2.vpl,
# make the figures flip back and forth so you
# can examine the the results of multiple
# attenuation. Let us call the output file
# cmp-signal2
#-------------------------------------------------------


######################
# Semblance Scan
######################
dv=10
nv=251
v0=1400
vscan='vscan v0=%d dv=%d nv=%d semblance=y half=n '%(v0,dv,nv)
pick='pick rect1=150 rect2=50 gate=20'

# semblance scan
Flow('vscan-cmp','cmp',vscan)

# semblance scan
```

```
Flow('vscan-signal2','signal2',vscan)

Plot('vscan-cmp',
        '''
        grey color=j allpos=y
        title="Velocity Scan - CMP"
        ''')

Plot('vscan-signal2',
        '''
        grey color=j allpos=y
        title="Velocity Scan - after demultiple"
        ''')

#----------------------------------------
# ADD CODE to display the two figures
# vscan-cmp.vpl and vscan-signal2.vpl
# side by side. Let us call the output
# file vcmp-signal2
#----------------------------------------



##########################################################
# This part is to create figures for tutorial.pdf
##########################################################
# define grey commands for figures to be included
# in tutorial.pdf
grey='''
      grey wanttitle=n labelfat=2 titlefat=2
      xll=2 yll=1.5 yur=9 xur=6
      '''

greyc='''
      grey wanttitle=n labelfat=2 titlefat=2
      xll=2 yll=1.5 yur=9 xur=6
      color=j allpos=y
      '''
# create plots
Result('cmp',grey)
Result('nmo',grey)
Result('taup',grey)
Result('nmo2',grey)
Result('taupmult',grey)
Result('signal2',grey)
```

```
Result('vscan-cmp',greyc)
Result('vscan-signal2',greyc)

End()
```

REFERENCES

Hampson, D., 1986, Inverse velocity stacking for multiple elimination: J. Can. Soc. Expl. Geophys, **22**, 44–55.

Lu, H., C. S. Fulthorpe, and P. Mann, 2003, Three-dimensional architecture of shelf-building sediment drifts in the offshore canterbury basin, new zealand: Marine Geology, **193**, 19 – 47.