OpenSeaSeis

May 2016

John Stockwell
Research Associate
Center for Wave Phenomena
Colorado School of Mines
What is OpenSeaSeis?

- Originally “SeaSeis” by Bjorn Olofsson 2006
- Now owned by the Colorado School of Mines

“A small open source seismic processing package”

--Bjorn Olofsson
Contents of OpenSeaSeis

- SeaView – interactive seismic data viewer
- XseaSeis – graphical processing flow monitor that runs:
  - SeaSeis – main processing program
  - Modules – about 80 modules for processes
What OpenSeaSeis is Not

- Not a graphical user interface for SU
- Not complete set of processing tools
- SEG-Y Rev 0 and Rev 1, no Rev 2 support
- SEG-D support weak, no Sercel SEG-D support
- Largely 2D
What's good about OpenSeaSeis

• Well written and robust
• Modular: process_name.so (.so library functions)
• A simplified internal data format
• Built-in process and error logging
• Processing flows are relatively easy to set up
• Command line, shell script, or XseaSeis
• Runs on Linux, Mac, and Windows (some)
SeaSeis
Structure of SeaSeis

- The program 'seaseis' is framework
- Modules: process_name.so
- Example: gain.so does gaining
- Written largely in C++ with C subroutines
- Existing C programs from other packages can be modified to serve as modules
Geophysical processing modules

libmod_proc1.so

libmod_proc2.so

libmod_proc3.so

libmod_procN.so

Traces

Each trace object is read once, passed through each module

SeaSeis main controls the flow of data through each called module

Output traces1

Output traces2
SeaView – seismic viewer

• Robust and Fast
• Reads: SEGY, SU, (SEGD), RSF, ascii format data
• Image plots, wiggle traces, scrolling through data
• Multiple plots, plots of different types of data
• Header value plots
XseaSeis
Processing Flows and Xseaseis

• Flows are like a shell script that runs modules
• About 80 modules currently
• Gain, sort, mute, selecting traces, ...
• Semblance, NMO, stack, header arithmetic, …
• Input and output of SEGY, SU, RSF
• Modeling: 2D ray tracing
# Read data from SEGY and gain

## INPUT SEGY
Filename: original_data.segy

## SUBREAS
mode trace

## SGAIN
tgain 2.0

## OUTPUT SEGY
Filename: gained_seismic.segy
# Read Data from SEGY and gain

$INPUTSEG
  filename original_data.segy

$DEBIAS
  mode trace

$GAIN
  tgain 2.0

$OUTPUTSEG
  filename gained_seismic.segy
# Read Data from SEGY and Gain

```
# Read Data from SEGY and gain

$INPUT_SEGY
filename original_data.segy

$DEBIAS
mode trace

$_GAIN
tgain 2.0

$OUTPUT_SEGY
filename gained_seismic.segy
```
Pre-parse input file `/scratch1/john/cwpscratch/Data5/read_data_from_segy.flow`...

Pre-parser summary:

Parsed 16 lines,
  found 0 'define' statement(s),
  0 table(s), and
  4 module(s).

Input flow `/scratch1/john/cwpscratch/Data5/read_data_from_segy.flow`:

# Read Data from SEGY and gain

$INPUT_SEGY
  filename     original_data.segy

$DEBIAS
  mode trace
Run init phase...

Run INIT phase for module #1 INPUT_SEGY...
Input file # 0: original_data.segy

Segy EBCDIC header:

C 1 CLIENT
C 2 LINE
C 3 REEL NO
C 4 INSTRUMENT: MFG
C 5 DATA TRACES/RECORD
C 6 SAMPLE INTERNAL
C 7 RECORDING FORMAT
C 8 SAMPLE CODE: FLOATING PT
C 9 GAIN TYPE: FIXED
C10 FILTERS: ALIAS
C11 SOURCE: TYPE
C12 PATTERN:
C13 SWEEP: START
C14 TAPER: START LENGTH
C15 SPREAD: OFFSET
C16 GEOPHONES: PER GROUP
C17 PATTERN:
C18 TRACES SORTED BY: RECORD
C19 AMPLITUDE RECOVERY: NONF
Exec phase summary

<table>
<thead>
<tr>
<th></th>
<th>Module</th>
<th>Traces in</th>
<th>Traces out</th>
<th>CPU time</th>
<th>CPU time all</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INPUT_SERY</td>
<td>0</td>
<td>120120</td>
<td>9.157</td>
<td>9.157</td>
</tr>
<tr>
<td>2</td>
<td>DEBIAS</td>
<td>120120</td>
<td>120120</td>
<td>0.194</td>
<td>9.351</td>
</tr>
<tr>
<td>3</td>
<td>GAIN</td>
<td>120120</td>
<td>120120</td>
<td>0.100</td>
<td>9.451</td>
</tr>
<tr>
<td>4</td>
<td>OUTPUT_SERY</td>
<td>120120</td>
<td>120120</td>
<td>2.036</td>
<td>11.487</td>
</tr>
</tbody>
</table>

Total processing time: 12.948000 seconds

Date: Fri Apr 29 11:51:31 2016

Trace allocation summary:
Total number of used/allocated traces: 1/4
Total number of allocated (trace) memory: 0.00kb (= 0.00Mb)

End of log.
# Read Data from SEGY and gain

```
$INPUT_SEGY
  filename       original_data.segy

$DEBIAS
  mode  trace

$GAIN
  tgain 2.0

$OUTPUT_SEGY
  filename gained_seismic.segy
```
Sort data into CMP gathers
$INPUT_SEGY
filename    original_data.segy
header_select   cmp
select     1-2142
sort       increasing tree

$ENS_DEFINE
  header cmp

$GAIN
tgain       2.0

$SORT
  mode       ensemble
  header     offset increasing

$DEBIAS
  mode       trace

$output_segy
filename    gained_cmp.segy
Semblance analysis
$INPUTSEGY
filename multiple_suppressed_cmp.segy

$ENSDEFINE
header cmp

$SELECT
header cmp
select 1950

$OUTPUT
filename cmp_1950.cseis

$SEM BLANCE
vel_range 0 1450 3500
vel_inc 15
window 100

$OUTPUT
filename semblance_panel_1950.cseis
NMO - Stack
$INPUT_SEGY
  filename multiple_suppressed_comp.segy

$ENS_DEFINE
  header cmp

$NMO
  wave_mode pp_iso
  table nmo_vel_2.table
  table_col 2 3
  table_key cmp 1
  mode apply

$MUTE
  time 480

$OUTPUT_SEGY
  filename nmo_cmp.segy

$STACK
  mode ensemble
  header cmp

$OUTPUT_SEGY
  filename stack_nmo_cmp.segy
Teapot Dome
paging through a 3D data cube
Summary

- Promising package
- Viewer works well
- Additional modules needed
- Everybody is welcome to participate
OSS and SU are on Github!

- http://github.org/JohnWStockwellJr/SeisUnix
Acknowledgements

Thank you CWP Sponsors, and many thanks to Bjorn Olofsson.