A Tutorial for

SeismicUnixGui, a graphical user interface for Seismic*nix under Linux

Juan M. Lorenzo

Version: 0.87.1 (March, 2024)
Louisiana State University
Dept. Geology and Geophysics
Baton Rouge

1 General Information

1.1 Acknowledgements

This project is possible only because of the selfless work of others. I have borrowed notes extensively from the Colorado School of Mines website (Stockwell, 1999) for Seismic*nix. Over the years, many students have also contributed to these notes: Class of 2008: Erin Walden, Kody Kramer, Erin Elliott, Andrew Harrison, Andrew Sampson, Ana Felix, JohnD'Aquin, Russell Crouch, Michael Massengale, and David Smolkin; Chang Liu (2013), Nevra Bulut (2019), Daniel Locci-Lopez (2023).

I would greatly appreciate any and all questions you have regarding installation and running of any of the programs to help us continue developing SeismicUnixGui. Please send your questions to ***gllore@lsu.edu***. Please indicate what your operating system is and whether you have administrative privileges (preferred).

Thanks,

Juan Lorenzo, Socorro, NM, June 2023.

1.2 What is SeismicUnixGui?

SeismicUnixGui, a graphical user interface (GUI), serves to select and build sequences of Perl modules and their parameters. SeismicUnixGui generates two versions of these instructions in text files. These text files contain a shell and a Perl script version that can be modified and also executed independently of this GUI and from the command line.

Seismic*nix (Stockwell, 1999) is a widely distributed free software package for processing seismic reflection and signal processing. In Seismic*nix (SU), a sequence of independent programs receive modify and generate data files of streams of data that are displayed on the screen. The data file is read in and the generated output data are handled internally by stdin, stdout functions in C while the data exchanges between programs and the linux operating system are managed from the command line via pipes "|" and redirections "> or <" respectively. Traditionally, the instructions on the command line can be assembled and saved as re-usable bash scripts. SeismicUnixGui assembles these same scripts for the operating system to run with the help of modules written in Perl. SeismicUnixGui generates these scripts within the directory of the user and these scripts can be run independently of SeismicUnixGui running.

SeismicUnixGui is written using Perl/Tk which is mature, well-documented Perl module that allows its users to construct graphical user interfaces.

In a classroom environment, shell scripting of SU modules engages students and helps focus on the theoretical limitations and strengths of signal processing. However, complex interactive processing stages, e.g., selection of optimal stacking velocities, killing bad data traces, or spectral analysis requires advanced flows beyond the scope of introductory classes. In a research setting, special functionality from other free seismic processing software such as SioSeis (UCSD-NSF) can be incorporated readily via an object-oriented style to programming. Recently, we have incorporated a 1D raytracing package built by E. Vera of the University of Chile to display interactively forward model or arrival times over real data.

An object-oriented approach is a first step toward efficient extensible programming of multistep processes, and a simple GUI simplifies parameter selection and decision making. Currently, in SeismicUnixGui, Perl 5 packages wrap over 300 of the most common SU modules that are used in teaching undergraduate and first-year graduate student classes (e.g., filtering, display, velocity analysis and stacking). Perl packages (classes) can advantageously add new functionality around each module and clarify parameter names for easier usage. For example, through the use of methods, packages can isolate the user from repetitive control structures, as well as replace the names of abbreviated parameters with self-describing names. Moose, an extension of the Perl 5 object system, greatly facilitates an object-oriented style. Perl wrappers are self-documenting via Perl programming document markup language.

An automatic directory structure is created for the user in which data and programs are distributed according to a pre-defined hierarchy. All the directories and minimal files needed by SeismicUnixGui are created whenever a new 'Project' is created within the 'Project Selector' tool. The user can also create new projects within main GUI of SeismicUnixGui as well as selecting different projects. At all times the user can use linux commands to navigate freely through the directories. Sometimes the user may find it convenient to create new subdirectories within the existing file structure, which SeismicUnixGui would not be able to detect.

1.3 GUI Sections

1.3.1 Overview

The main GUI is divided into four sections (1) A top menu with an option of over 300 modules, (2) a menu on the left side for saving, opening, deleting and running a flow (sequence of assembled instructions). (3) For each geophysical module the parameter names and values are located in the central area of the image.

There is also a convenient cross symbol (**X**) , in the top-left corner of the window, that will kill all open windows, if the screen becomes to cluttered.

Before running each flow, the user must give the flow a name and save it to a file. The correct clicking sequence, is always “Save” or “Save As” followed by “Run”

Up to four, independent processing flows can be tested in the GUI (grey, pink, green and blue rectangles). The modules within each processing flow can be rearranged using the up (inverted **V**) and down (**V**) symbols or deleted (**-).** An entire flow (colored box) can be cleared by clicking on another cross (**X),** found on the right-hand-side of the window.**.**

Messages to the user will either appear in short-lived boxes or within the black box at the bottom of the GUI.

Data file names usually have a suffix that indicates the type of data contained, e.g., header.txt or 1.su contain data in ASCII and seismic-unix format respectively. Generally, the menus of the exclude the suffix and rely on the directory path to determine the type of file to be read. If you are reading a data file or writing a data file, the termination will determine the container directory. A common problem with not finding a file is that its suffix may be slightly different to the one that the GUI expects to find; e.g., “.sgy” instead of “.segy”, or “.text” instead of “.txt”

Example 1: If “su” Is entered then the GUI will assume the data is found in directory

~/Servilleta/seismics/data/loma_blanca/053018/H/1/su/**user**/

Example 2: If “bin” Is entered then the GUI will assume the data is found in directory

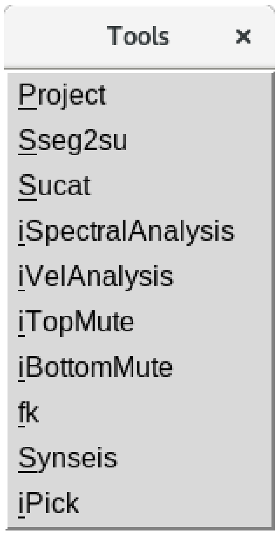
~/Servilleta/seismics/data/loma_blanca/053018/H/1/bin/**user**/

Example 3: If “segy” is entered then the GUI will assume the data is found in directory

~/Servilleta/seismics/data/loma_blanca/053018/H/1/segy/**user**/

1.3.2 Side Menus

1.3.2.1 Tools



Project: Defines the directory structure for data sets and programs in many languages, e.g. matlab, R, Perl etc.

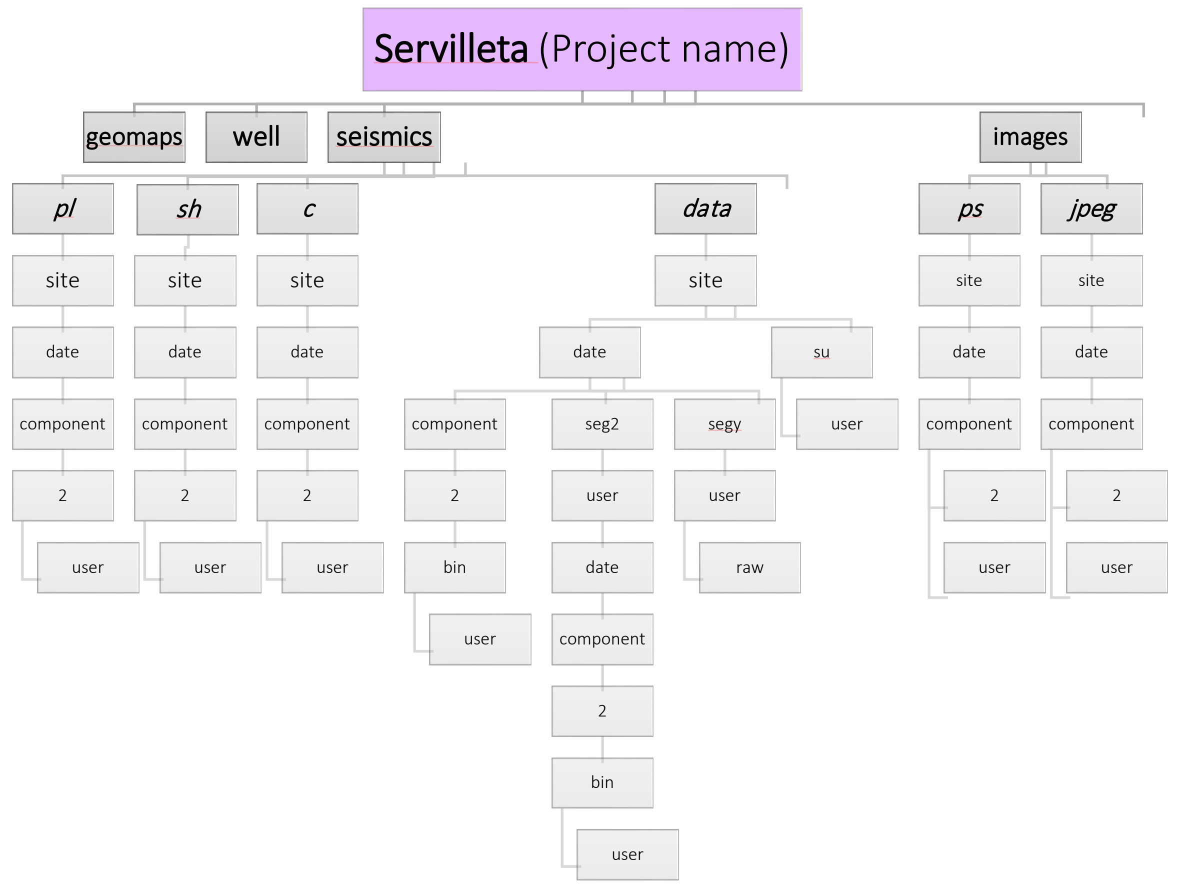
Sseg2su: Converts SEG-2 formatted data into the su format which is a simplified SEG-Y format (requires sioseis installation)

% Sseg2su

Sucat: Concatenates multiple files of any format into a single file. These files can have names related by a continuous sequence of integers, e.g., Seismic Unix data files: 1000.su, 1001.su, 1002.su. If not, a list of names can be specified. Output files from interactive muting or velocity analysis and that have specific “par” formats can be handled.

% Sucat

1.4 What is an example directory structure for a Project?



1.4.1 Copying data into the project directory structure from elsewhere in the system

If you want to copy seismic data that are already in SeismicUn*x format (SU), copy it into:

Generic: PROJECT_HOME/seismics/**data**/site/component/line/user

Real Example: /home/gllore/seismics/data/Servilleta/H/1/gllore

1.4.2 Where are my flows kept?

Generic: ls PROJECT_HOME/seismics/**pl**/site/component/line/user

1.5 Text conventions in this tutorial and their meaning

Left Mouse click is abbreviated to <MB1> Instruction

Right Mouse click is abbreviated to <MB3> Instruction

Variable names are shown in a large bold-style font.

% Command-line instructions are shown with pink background

1.6 Glossary

Term	Explanation and Example	Brief
HOME	Full linux directory path to the user's home directory, e.g. /home/xavier45	home directory path
PROJECT_HOME	Located inside **HOME** directory -- can be a soft link	project directory path
Projectname	e.g., Servilleta -- a National Wildlife Refuge in New Mexico, U.S.A.	name of the project
spare_dir	can be left empty	a bonus directory
date	053018	Of field work
component	Z stands for vertical and and H can be horizontal but any name is possible	Geophone particle displacement component
line	1	used to identify a profile
user	e.g., xavier45	login name
subUser	must be set to the user's login name, e.g.,also xavier45	Allows groups to share Project space
flow	Data_in, sugain, suximage	Sequence of programs to execute
geomaps	Directories will be created when working withmaps	Directories for third-party software (if installed and accessible)

sqlite	Databases	Directories for third-party software (if installed and accessible)
gmt	GMT	Directories for third-party software (if installed and accessible)
grass	GRASS GIS	Directories for third-party software (if installed and accessible)

Table 1: Definitions of terms used when creating working projects

2 Demonstration Projects

When either creating a new project or accessing a pre-existing project instances, always start by running the following instruction:
% SeismicUnixGui

2.1 A Quick start to preparing a demonstrations

2.1.1 Where are my data sets stored?

Before starting a new project you should understand the file structure in which programs and data sets are stored. The main directories are shown above for the example of Servilleta in Section 1.4.

2.1.2 Install example flows and data sets

Several example projects that contain data and examples flows can help you become acquainted with the Seismic Unix Tools. For example:

- ***Servilleta contains files from the 2018 IRIS internship orientation program***
- ***LSBB contains files from Pau University in France, courtesy of Dominique Dominique Rousset and Guy Sénéchal, both extensive contributors to the improvement of Seismic Unix.***
- ***Demos contains general demonstrations of tools not included in the previous tutorials***

The following is explained the SeismicUnixGui Installation manual (Section 1.3.6) but is repeated here for convenience of the user. Once you completely install SeismicUnixGui on your system, you can move or copy any of the accompanying demonstration folders to the home directory of the user, where /home/user is the complete path to the location of the user (=”user”).

```
% cp -R $installation_directory_for_demo_projects/Servilleta       /home/user/

% cp -R $installation_directory_for_demo_projects/LSBB             /home/user/

% cp -R $installation_directory_for_demo_projects/demos            /home/user/
```

2.1.3 Create a new project, e.g., Servilleta (IRIS demonstration data set)

The following instruction starts the program:

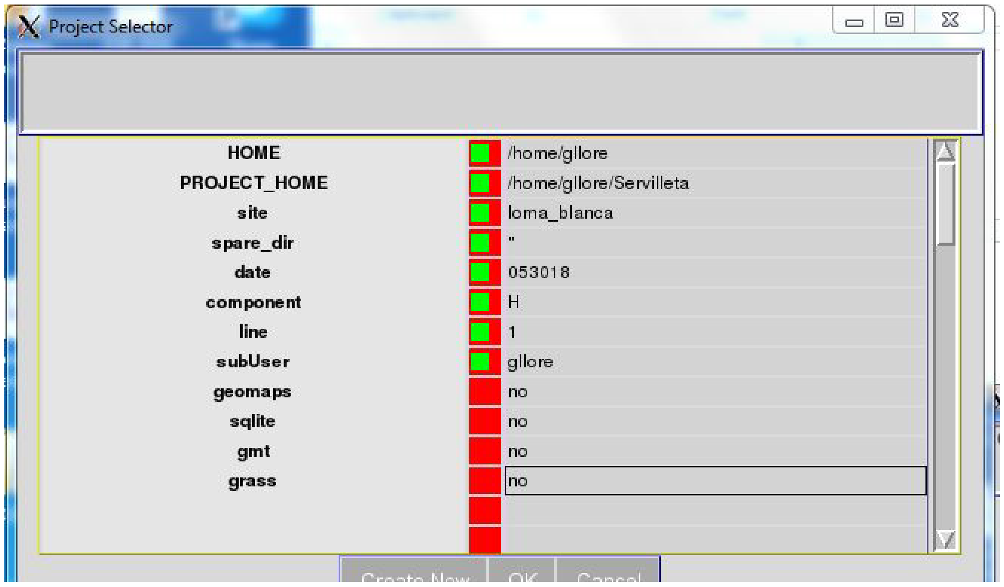
% SeismicUnixGui

If you do not have any projects created previously, then:

<MB1> Create New

Otherwise, go to next section 2.1.5: Open a pre-existing project.

After clicking on Create New, a default set of parameter names (e.g., **HOME**) and their values (e.g.**, /home/login_name**) appears:



HOME	🟩	/home/gllore
PROJECT_HOME	🟩	/home/gllore/Servilleta
site	🟩	loma_blanca
spare_dir	🟩	''
date	🟩	053018
component	🟩	H
line	🟩	1
subUser	🟩	gllore
geomaps	🟥	no
sqlite	🟥	no
gmt	🟥	no
grass	🟥	no

Figure 1: Screen capture of Project Selector Pane with parameters and their values

The Project Selector pane displays several default options that work with the test data set that is included for this tutorial. The old variables are defaulted from prior projects and serve as an example to guide your input. The home directory of the user is required to follow the standard linux file structure naming system.

These options should be <u>updated</u> with your <u>login</u> name, for example:

Parameter name	Default values
HOME	/home/**login_name**
PROJECT_HOME	/home/**login_name**/Servilleta

Site	Servilleta
spare_dir	“”
date	053018
component	Z
line	2
subUser	**login_name**
geomaps	no
sqlite*	no
gmt*	no
grass*	no

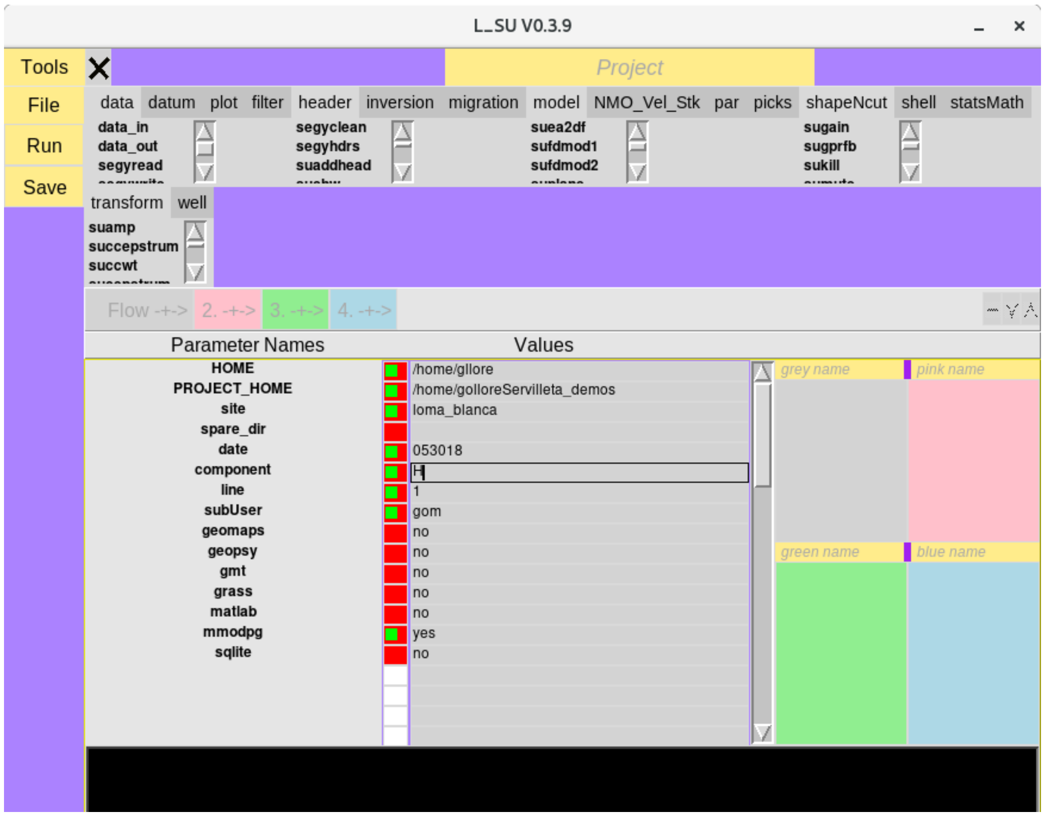
Table 1: Substitute login_name with your user name in three locations.

*** if set to ‘yes’ only the directories will come to be created although the accompany-ing programs are not yet available in this version (June, 2023)**

Finally, select: <MB1> OK

2.1.4 For the URISE data set, confirm you are working in a Project called “Servilleta”

In the top left menu, select <MB1> Tools->Project



Parameter Names		Values
HOME	🟩	/home/gllore
PROJECT_HOME	🟩	/home/golloreServilleta_demos
site	🟩	loma_blanca
spare_dir	🟥	
date	🟩	053018
component	🟩	H
line	🟩	1
subUser	🟩	gom
geomaps	🟥	no
geopsy	🟥	no
gmt	🟥	no
grass	🟥	no
matlab	🟥	no
mmodpg	🟩	yes
sqlite	🟥	no

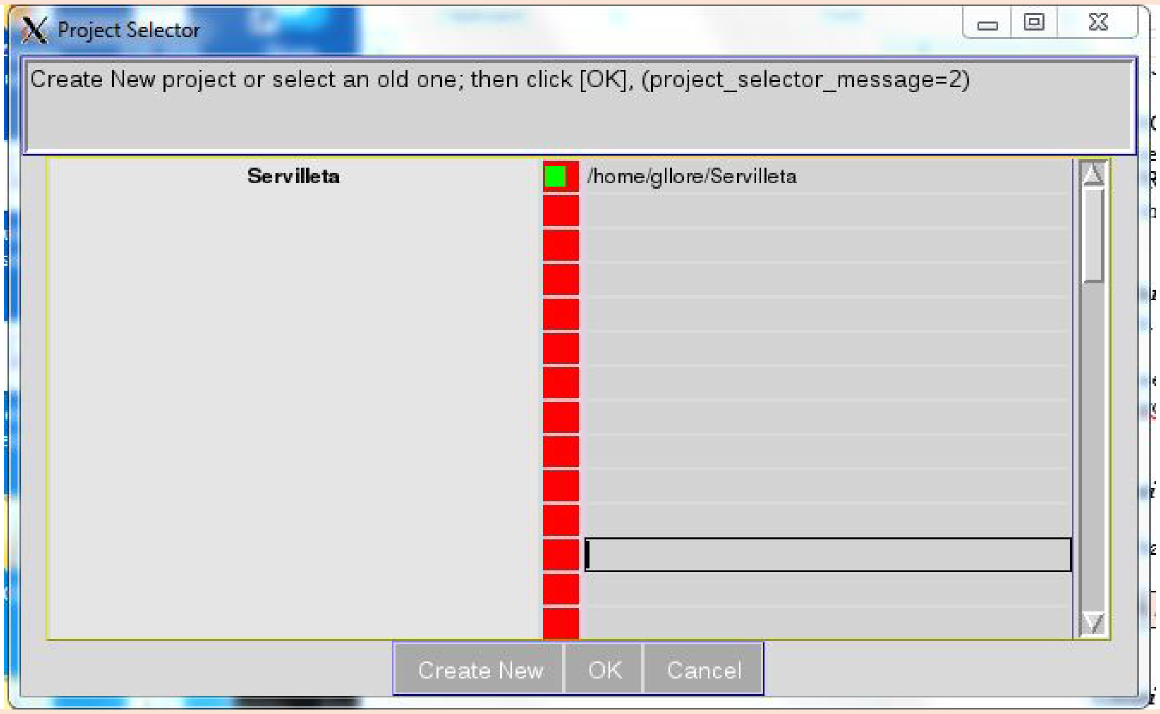
In the main window of the SeismicUnixGui you should see the previous changes you made to the same parameter values. It they are incorrect (e.g., the figure above shows an inconsistent use of the user name) you can introduce the correct names and, without exiting this window you can then select:

In the top left menu: <MB1> Save->Ru

2.2 Open a pre-existing project

2.2.1 The following instruction starts the program, and open the pane of the Project Selector window:

```
% SeismicUnixGui
```



Create New project or select an old one; then click [OK], (project_selector_message=2)

Servilleta	🟩	/home/gllore/Servilleta

If the project of interest (in this case Servilleta) is selected (button is green) :

Select: <MB1> on OK

2.3 Running your first flows

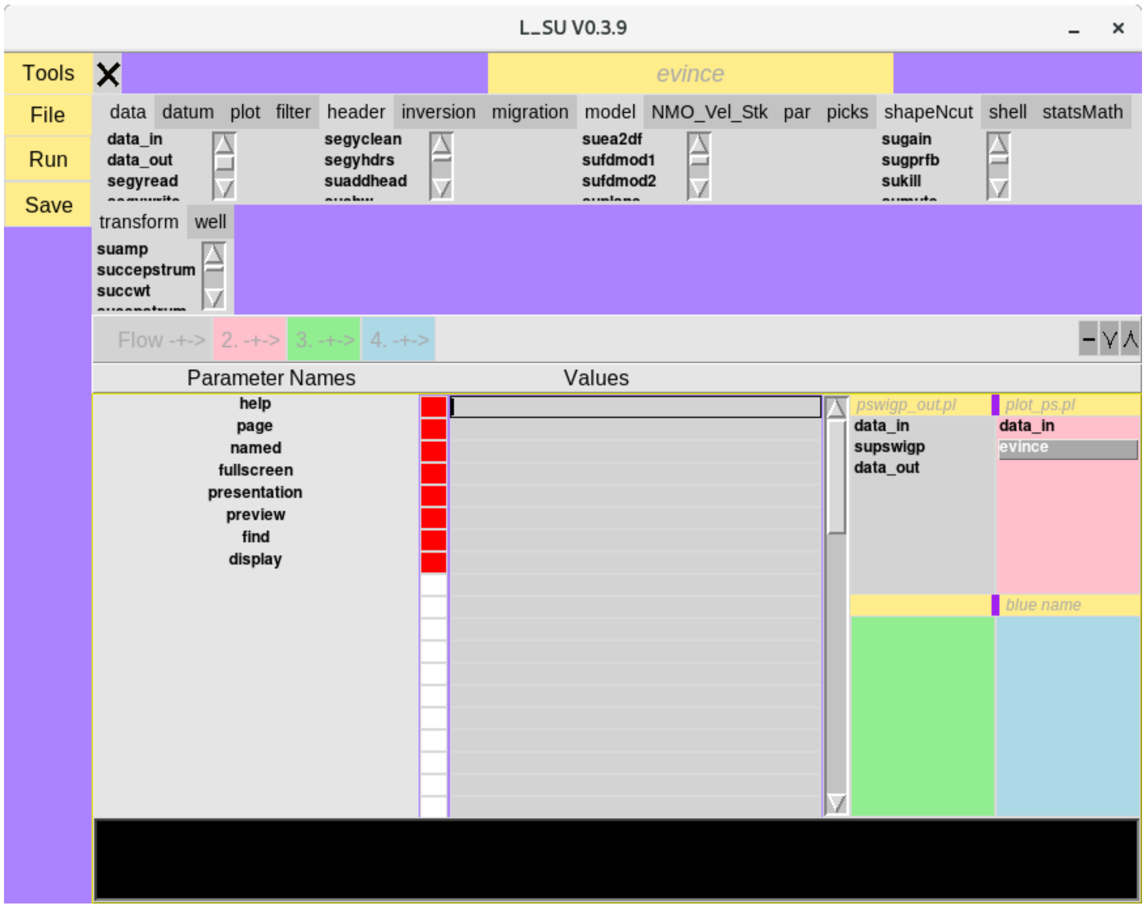
Assemble a sequence of modules to carry out a processing procedure. Choose one of four differently colored flow windows (grey, pink, green and blue) in which to place your sequence. The colored window appear on the right-hand side of the main window.

A module, with a specific functionality, is selected by clicking on its name from within the list on the left-hand side of the main window.

The module name must be transferred to the list on the right by clicking one of the four different colored flow arrows, just to the right of the word "Flow".

A final assembled flow must first be saved to a file before it is executed (**File->SaveAs**). Thereafter all executions require that the flow be first saved before running.

In a simple sequence of modules, data are usually read in first, the data are modified and the result is placed into another file or displayed using an imaging module (e.g., suximage, suxwigb)



1. Select the following named modules: ***data_in, supswig, and data_out***. Click on each names inside list on the left side of the window. When you do that, the words in the row immediately above will become activated. You will then be able to click on the words inside the grey box:

Flow-+->

You should be able to see the name of the program that you just selected move over to a colored box on the right-hand side of the window.

Select each of the three program names: ***data_in, supswigp***, and ***data_out***

2. You are required to select a **Value** for **base_file_name** (= "file name").

To do so, move your cursor into the corresponding row to the right of **base_file_name**.

A click of the right-mouse-button will automatically open a second window from which you can select a file, e.g. **“103.su”**.

Before you can run the program you have built, it must be saved:

For SeismicUnixGui GUI

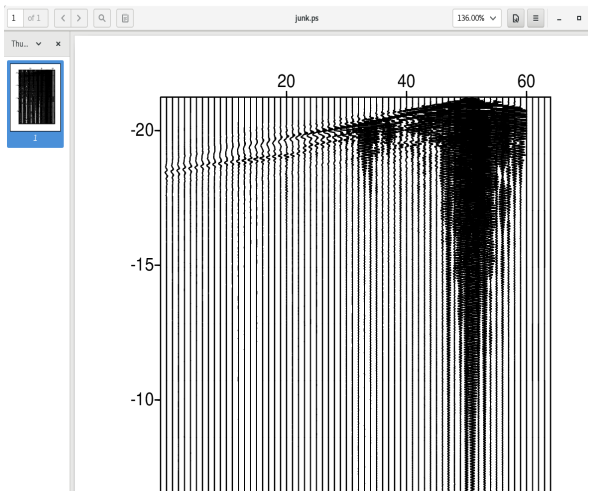
<MB 1> File/SaveAs

Save the resultant perl script file as, e.g.,

“pswigp_out.pl”

Then, click on

Tool: <MB 1> Run



Postscript plot viewed using the GUI

Tool: <MB1> Run

2.3.1 Perl and Shell script flows generated by SeismicUnixGui

GUI-generated perl script: plot_ps.pl

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl plot_ps.pl
```

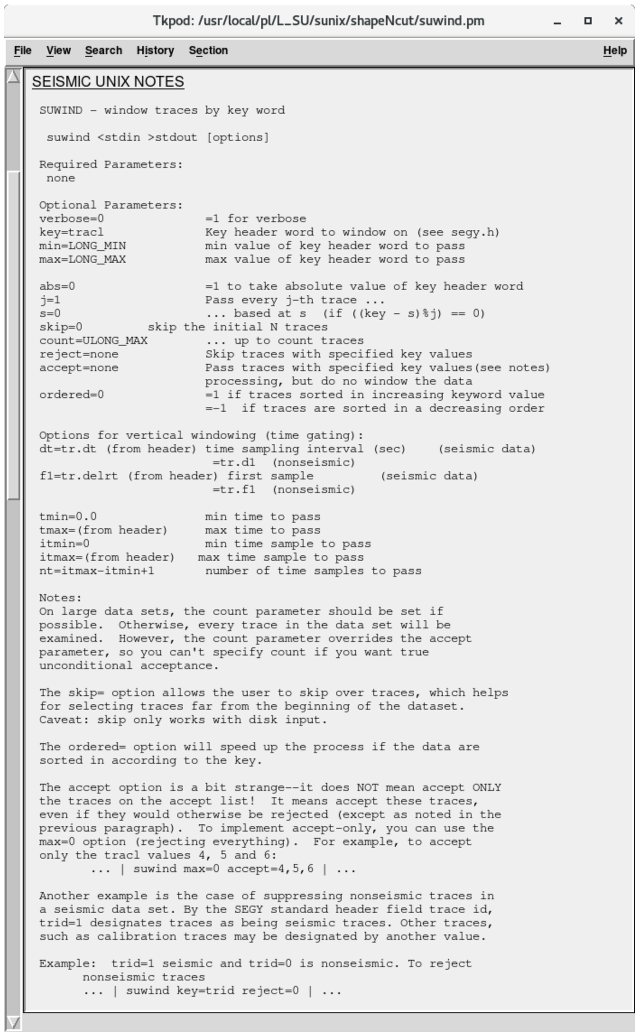
To run the bash script from the command line that is generated by plot_ps.pl:

```
% evince /home/user/Servilleta/seismics/images/ps/loma_blanca//053018/H/1/user/junk.ps &
```

(Note that **pswigp_out.pl** is run first and and **plot_ps.pl** second.

2.3.2 Access to Documentation

Select **<MB3>** over the name of the program:


Tkpod: /usr/local/pl/L_SU/sunix/shapeNcut/suwind.pm
File View Search History Section Help
SEISMIC UNIX NOTES

```
SUWIND - window traces by key word

 suwind <stdin >stdout [options]

Required Parameters:
 none

Optional Parameters:
verbose=0               =1 for verbose
key=tracl               Key header word to window on (see segy.h)
min=LONG_MIN            min value of key header word to pass
max=LONG_MAX            max value of key header word to pass

abs=0                   =1 to take absolute value of key header word
j=1                     Pass every j-th trace ...
s=0                     ... based at s  (if ((key - s)%j) == 0)
skip=0         skip the initial N traces
count=ULONG_MAX         ... up to count traces
reject=none             Skip traces with specified key values
accept=none             Pass traces with specified key values(see notes)
                        processing, but do no window the data
ordered=0               =1 if traces sorted in increasing keyword value
                        =-1  if traces are sorted in a decreasing order

Options for vertical windowing (time gating):
dt=tr.dt (from header) time sampling interval (sec)    (seismic data)
                         =tr.d1  (nonseismic)
f1=tr.delrt (from header) first sample         (seismic data)
                         =tr.f1  (nonseismic)

tmin=0.0                min time to pass
tmax=(from header)      max time to pass
itmin=0                 min time sample to pass
itmax=(from header)    max time sample to pass
nt=itmax-itmin+1        number of time samples to pass

Notes:
On large data sets, the count parameter should be set if
possible.  Otherwise, every trace in the data set will be
examined.  However, the count parameter overrides the accept
parameter, so you can't specify count if you want true
unconditional acceptance.

The skip= option allows the user to skip over traces, which helps
for selecting traces far from the beginning of the dataset.
Caveat: skip only works with disk input.

The ordered= option will speed up the process if the data are
sorted in according to the key.

The accept option is a bit strange--it does NOT mean accept ONLY
the traces on the accept list!  It means accept these traces,
even if they would otherwise be rejected (except as noted in the
previous paragraph).  To implement accept-only, you can use the
max=0 option (rejecting everything).  For example, to accept
only the tracl values 4, 5 and 6:
       ... | suwind max=0 accept=4,5,6 | ...

Another example is the case of suppressing nonseismic traces in
a seismic data set. By the SEGY standard header field trace id,
trid=1 designates traces as being seismic traces. Other traces,
such as calibration traces may be designated by another value.

Example:  trid=1 seismic and trid=0 is nonseismic. To reject
      nonseismic traces
      ... | suwind key=trid reject=0 | ...
```

Conventional Seismic Unix documentation for the modul: suwind

3 Simple Processing Flow: IRIS Data Set, Socorro New Mexico

Each year Incorporated Research Institutions for Seismology (IRIS) hold an orientation week for undergraduate research interns in the town of Socorro, New Mexico. As part of a week of training, the on May 30 of 2018, the students collected an active-source seismic data set, which we process using Seismic Unix.

3.1 Processing steps

The following outline is taken from a called notes.pl. This files exists in the perl flow directory (1.4.3) of the Servilelta_demos project. To get there change to the following directory:

```
% cd  /home/user/seismics/pl/site/component/line/user
```

To see the marked-up contents of the perl file:

```
%perldoc notes.pl
```

IRIS 2018 survey May 30 2018
on S bank of Rio Salado
along same line as pseudo-walkaway taken on 032618
shoot-through

Acquisition parameters

Date	**053018**
SI	**1000 S/s**
delrt	**-11 ms**
rec. length	**2 s**
num tracr	**64**
Live channels	**1-64**
Channel 1	**closest to recorder-- toward SE**
Channel 64	**farthest from recorder-- toward NW**
geophones: Geospace	**28 Hz L-4 3 component**
geophone spacing:	**1 m**
line orientation:	**NW-SE later shots more toward NW**
Number of Geophones	**60**
Shotpoint Spacing	**1 m**

GPS is available (etrex garmin 10 m)

	(sx-m)	**NOMINAL offset-m**	**ACTUAL (m)**
Raw SP 1	**0**	**1-60**	**0.5 - 59.5**
Raw SP 2	**1**	**0-59**	**-0.5 - 58.5**
Raw SP 3	**2**	**-1-58**	**-1.5 - 57.5**
Raw SP 4	**3**	**-2-57**	**-2.5 - 56.5**
Raw SP 60	**59**	**-58-1**	**-58.5 - 0.5**

Striker plate	**I-beam**
Hammer	**10 lb sledge**
No. blows	**3 per side**

Noise sources:	**5 - 10 mph from SE**
I-25	**to E**

Acquisition parameters taken from the file notes.pl

3.2 STEP 1. File format conversion

Tool: Seg2su (from GUI)

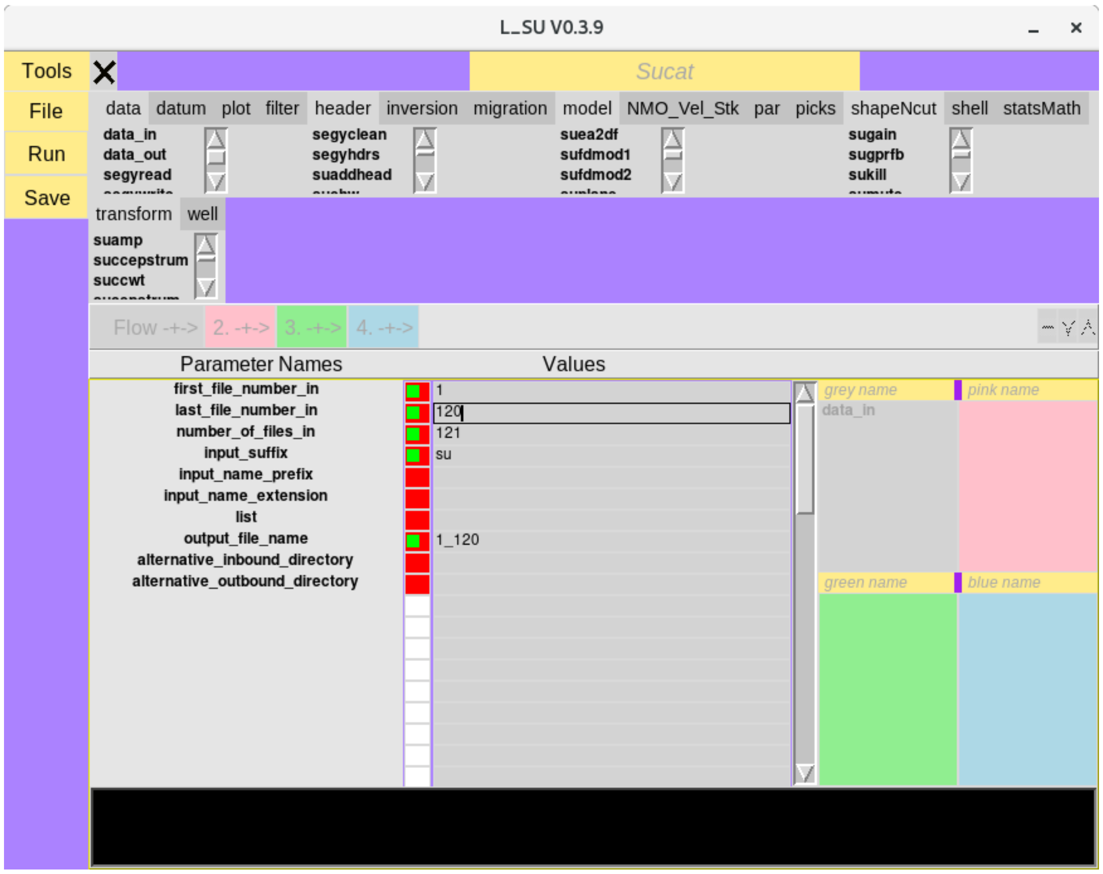
Purpose: Convert Seg2 to Seismic Unix format
Input: 1 to 120.dat

Output: 1 to 120.su

3.3 STEP 2. Concatenate files

Tool: Sucat (from GUI)
Purpose: cat all files
Input: 1.su to 120.su
Output: 1_120.su
Uses: /home/user/Servilleta/seismics/pl/loma_blanca/053018/**Sucat.config**

SeismicUnixGui Gui:



Parameter Names	Values
first_file_number_in	1
last_file_number_in	120
number_of_files_in	121
input_suffix	su
input_name_prefix	
input_name_extension	
list	
output_file_name	1_120
alternative_inbound_directory	
alternative_outbound_directory	

3.4 STEP 3. Clean headers

Flow name: **Suclean_geom.pl** (from GUI)

Purpose: Modify the geometry headers for shoot-through survey by wiping certain headers and populating new ones

Input: 1_120.su
Output: 1_120_clean_geom.su

3.5 STEP 4. Window the shotpoint gathers **(from GUI)**

Flow name**: Suwind.pl** (from GUI)

Purpose: Allow ONLY traces
traces 1-60
data: time 0 s to 1 s

Input: 1_120_clean_geom.su
Output: All.su

To view the data as an image: **view_All.pl** (Select and run Flow in GUI)

3.6 STEP 5. Separate shear-wave shots from alternate directions

Flow name: % perl Sudiff.pl (from command line, and in the "pl" directory)
Input: All.su
Output: L28HzHit_fromNE.su and L28HzHit_fromSW.su

Extract and group 'from-NE_shotgathers' from 'from-SW-gathers'

3.7 STEP 6. Negative stack

Flow name: **suop2.pl** (from GUI)
Input: L28HzHit_fromNE.su and L28HzHit_fromSW.su
Output: L28Hz_Ibeam.su

Subtract 'from-NE_shotgathers' from 'from-SW-gathers'

To view the data: **view_L28Hz_Ibeam.pl** (from GUI)

3.8 STEP 7. Add values for headers--gx,ep,sx

Flow name: **SuGeom2.pl** (from GUI)
Purpose: populate headers with meaningful values;
header names are: sx, gx, ep (explosion point), tracl, tracf, fldr

Input: L28Hz_Ibeam
Output: L28Hz_Ibeam_geom2

tracl now counts the sequential increase of traces for the whole line
tracf and fldr are now removed completely
ep signifies the shotpoint number
sx is the x location (m) of the shotpoint
gx is the x location (m) of the geophone

To verify new header parameters: **SuPlotHeader.pl** (from GUI)

To view new header parameter numerical values: **suxedit**
If you want to directly view the data change to the current data directory (2.4.1):

```
4        % cd /home/user/Servilleta/seimics/data/loma_blanca//053018/H/1/su/user
```

And then when you are in the correct data directory:

```
5        % suxedit L28Hz_Ibeam_geom2
```

(at trace number = 181):

```
>

tracl=181  tracr=1 ep=4 sx=3 gx=1 ns=1001 dt=1000
```

5.1 STEP 8. Modify Header files--offsets

make_offsets.pl (from GUI)

Purpose: Calculate offsets from the headers

Input: L28Hz_Ibeam_geom2
Output: L28Hz_Ibeam_geom3

Confirm the result by examing the numerical values for offset:

```
% cd /home/user/Servilleta/seimics/data/loma_blanca//053018/H/1/su/user
```

And then when you are in the correct data directory:

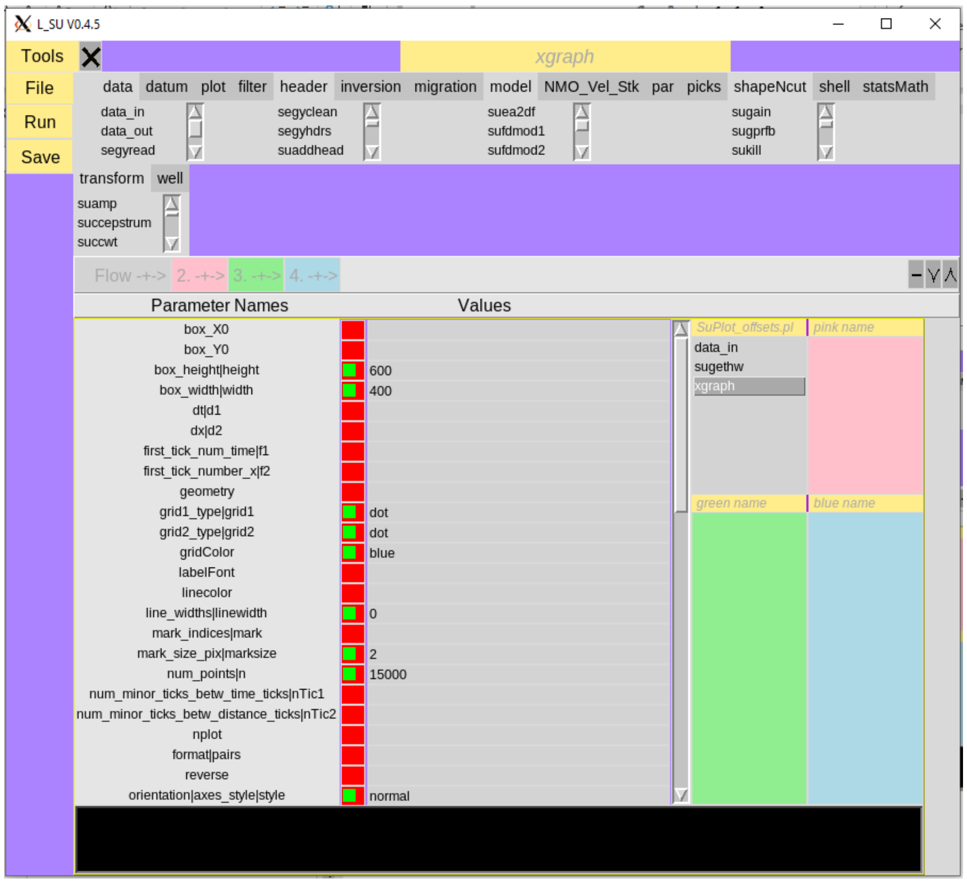
```
% **suxedit** L28Hz_Ibeam_geom3
```

(at trace number = 181):

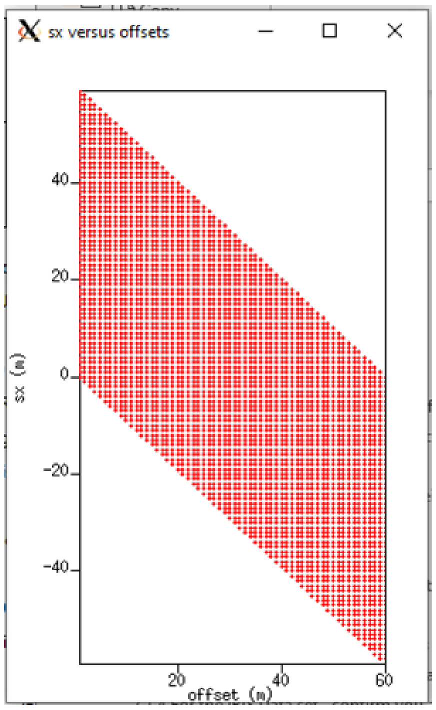
```
>

tracl=181  tracr=1 ep=4 offset=-2 sx=3 gx=1 ns=1001 dt=1000
```

Graphically, verify new header parameters using **SuPlot_offsets.pl** (from GUI)



Parameter Names		Values
box_X0	🟥	
box_Y0	🟥	
box_height|height	🟩	600
box_width|width	🟩	400
dt|d1	🟥	
dx|d2	🟥	
first_tick_num_time|f1	🟥	
first_tick_number_x|f2	🟥	
geometry	🟥	
grid1_type|grid1	🟩	dot
grid2_type|grid2	🟩	dot
gridColor	🟩	blue
labelFont	🟥	
linecolor	🟥	
line_widths|linewidth	🟩	0
mark_indices|mark	🟥	
mark_size_pix|marksize	🟩	2
num_points|n	🟩	15000
num_minor_ticks_betw_time_ticks|nTic1	🟥	
num_minor_ticks_betw_distance_ticks|nTic2	🟥	
nplot	🟥	
format|pairs	🟥	
reverse	🟥	
orientation|axes_style|style	🟩	normal



Plotted header values of sx versus offset display a regular geometric pattern that reflects the regular acquisition geometry of sources versus offest used in the experiment.

Verify the new header parameter values using **suxedit**

Convention: Positive offets are when geophones lie N of shot. Negative offsets are when shot lies N geophone

```
% cd /home/user/Servilleta/seimics/data/loma_blanca//053018/H/1/su/user
```

And then when you are in the correct data directory:

```
% suxedit L28Hz_Ibeam_geom3.su
```

5.2 STEP 8. Modify Header files--Make CMP's

make_cmp.pl (from GUI)

Purpose: Put cdp values in the ”cdp” headers

Input: L28Hz_Ibeam_geom3
Output: All_cmp

Numerically verify new header parameters using **suxedit**
Convention: The range of cdp varies systematically

```
% cd /home/user/Servilleta/seimics/data/loma_blanca//053018/H/1/su/user
```

And then when you are in the correct data directory:

```
% perl suxedit All_cmp.su
```

Records 59 through 61 are as follows:

```
59
```

```
 tracl=59 tracr=59 ep=1 cdp=29 offset=59 gx=59

 ns=1001 dt=1000

> 60

 tracl=60 tracr=60 ep=1 cdp=30 offset=60 gx=60

 ns=1001 dt=1000

> 61

 tracl=61 tracr=1 ep=2 cdp=1 sx=1 gx=1

 ns=1001 dt=1000
```

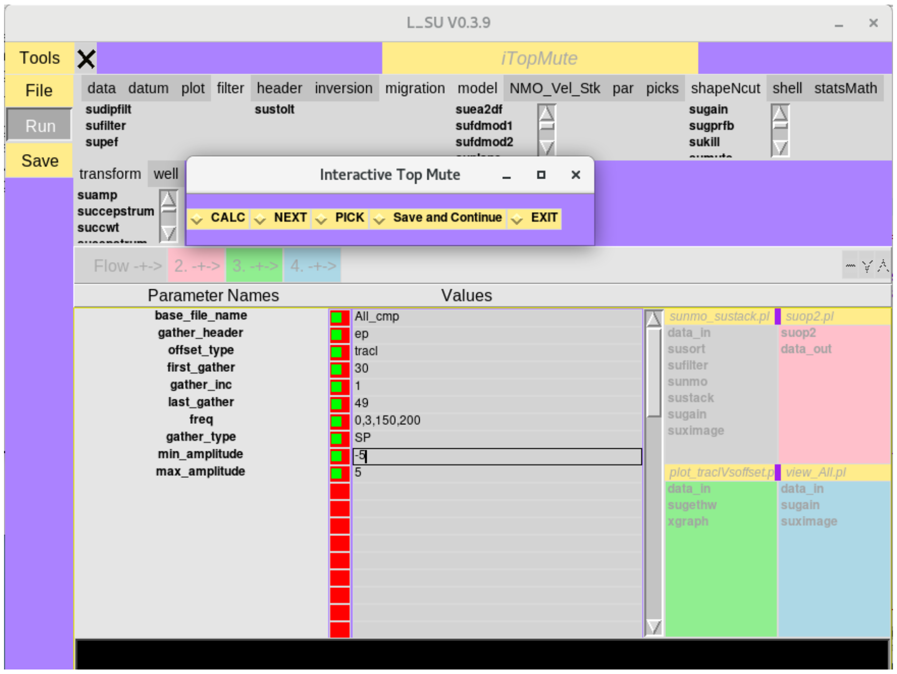
5.3

5.4 STEP 9. Dip filter

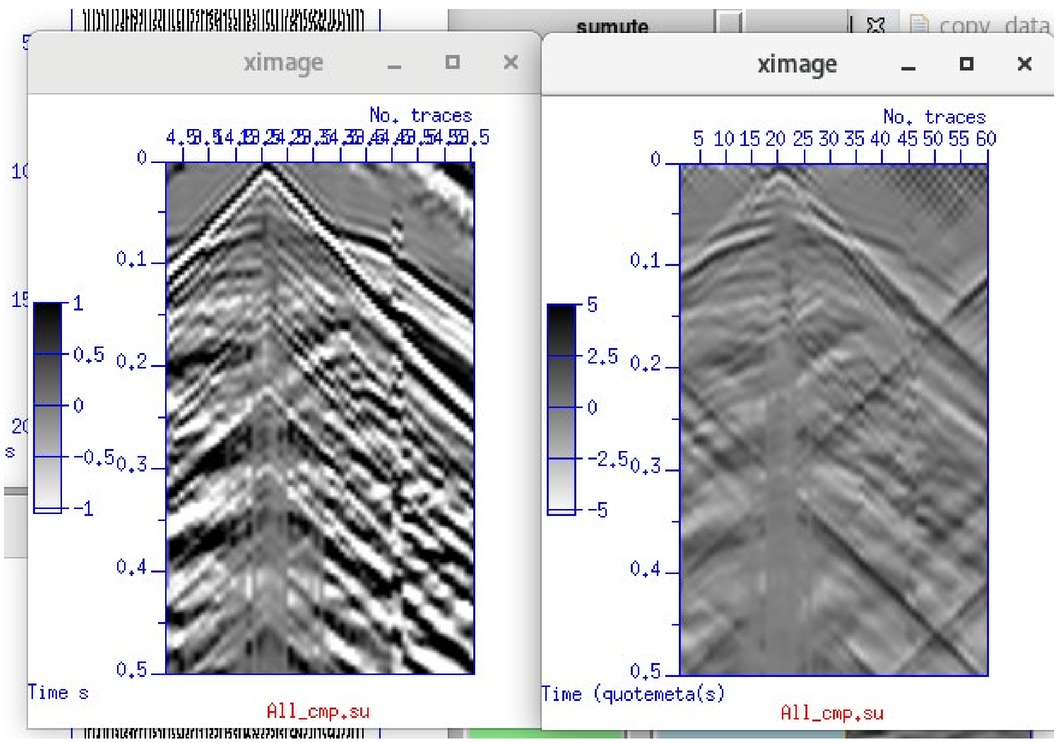
Tool: **ifk** (interactive velocity filtering)

Purpose: Useful for separation of reflections from surface waves
Uses: /home/user/Servilleta/seismics/**pl**/loma_blanca/053018/Sucat.config

SeismicUnixGui Gui



Parameter Names	Values
base_file_name	All_cmp
gather_header	ep
offset_type	tracl
first_gather	30
gather_inc	1
last_gather	49
freq	0,3,150,200
gather_type	SP
min_amplitude	-5
max_amplitude	5

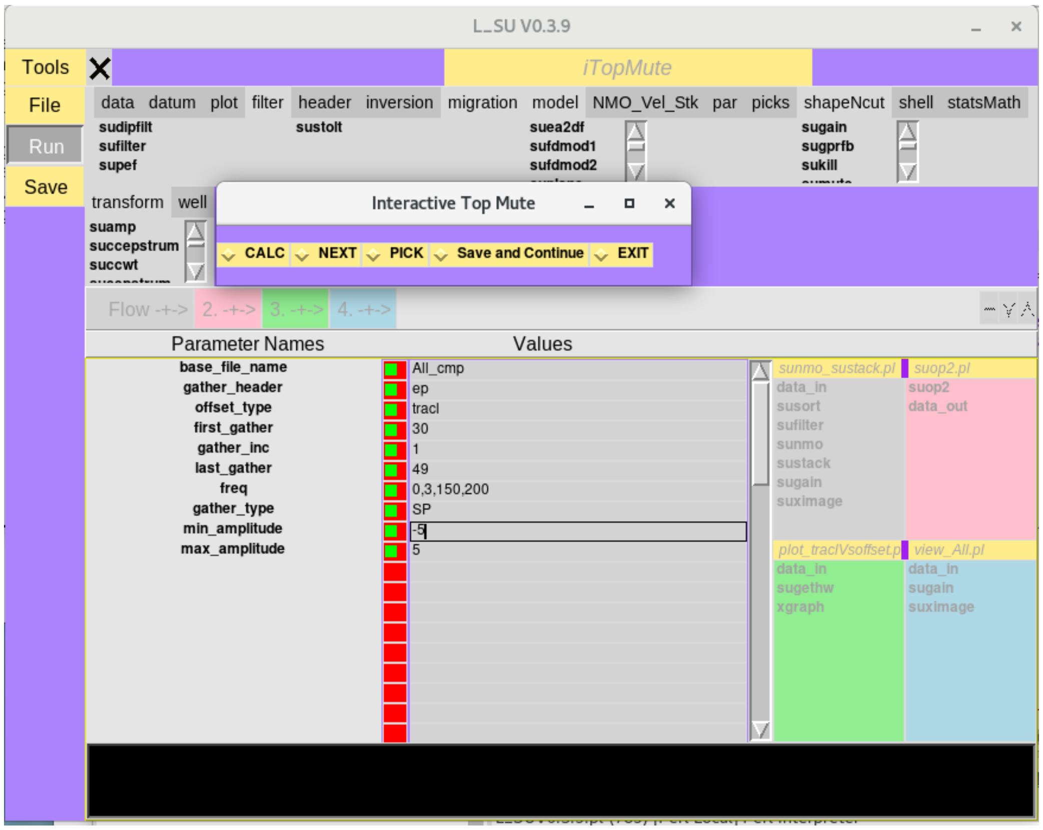


Before (left) and after (right) f-k filtering

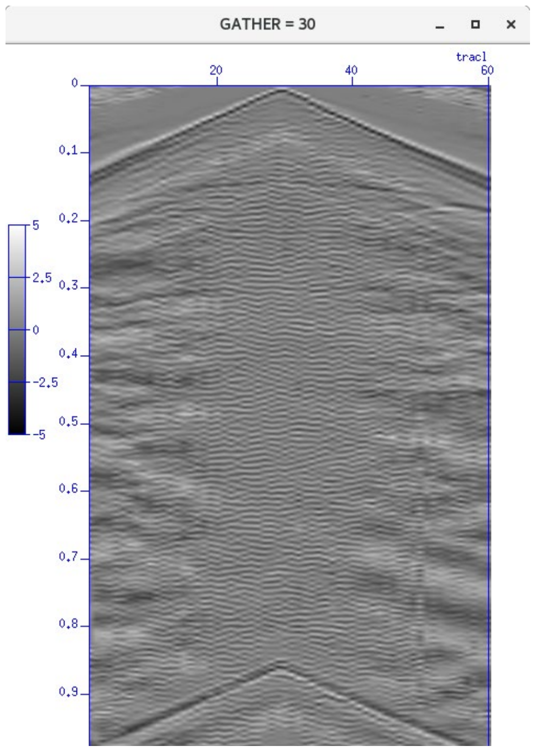
5.5 STEP 10. Test Muting of surface waves and refracted waves

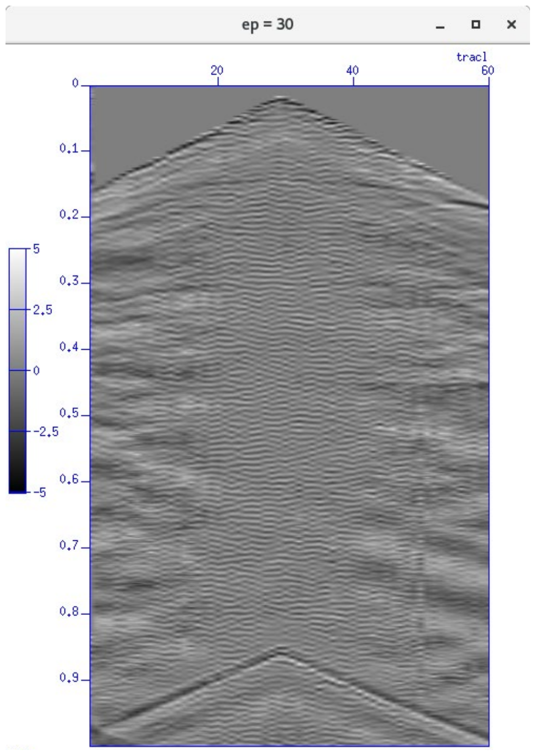
Tool: **iBottomMute** Interactive Top Bottom Mute,SP 1

Testing- not used in this flow
Input:
Output:
Uses: Uses: /home/user/Servilleta/seismics/pl/loma_blanca/053018/Sucat.config



Parameter Names	Values
base_file_name	All_cmp
gather_header	ep
offset_type	tracl
first_gather	30
gather_inc	1
last_gather	49
freq	0,3,150,200
gather_type	SP
min_amplitude	-5
max_amplitude	5



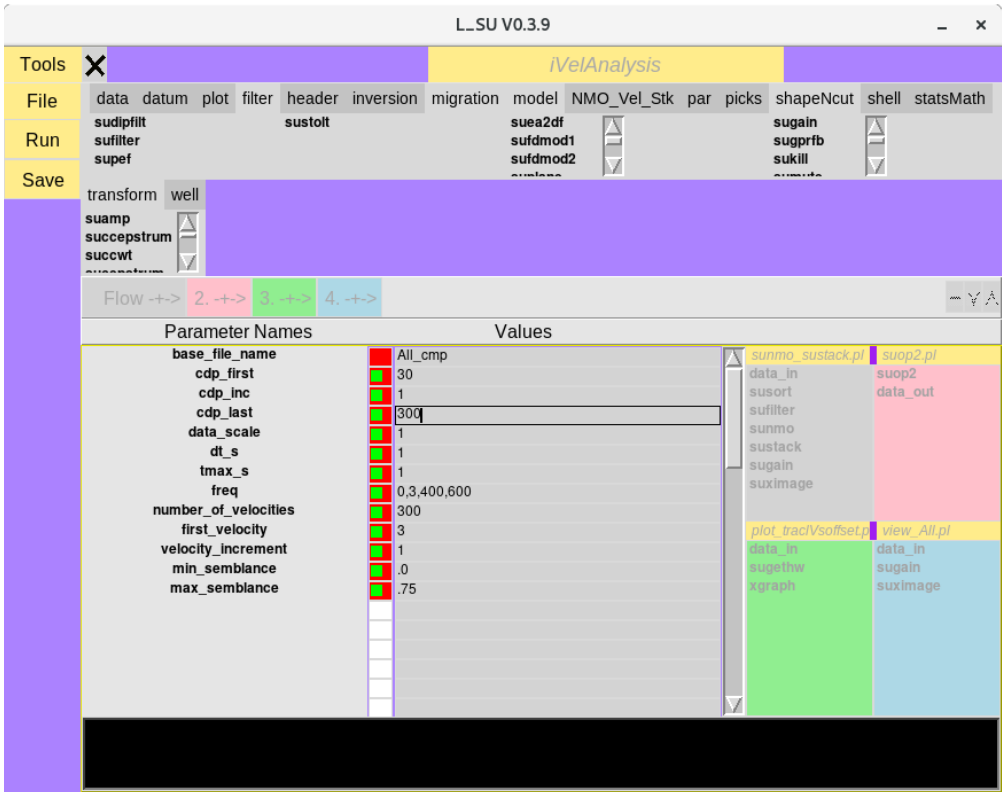


(Left) Before and after (Right) images of cdp=30 gather generated during application of interactive top-muting Tool.

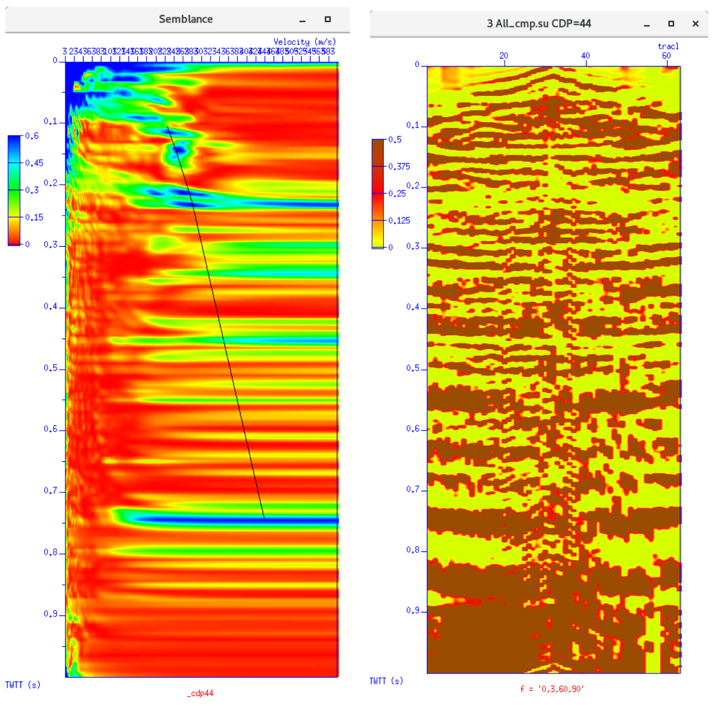
5.6 STEP 11. Test Semblance Analysis

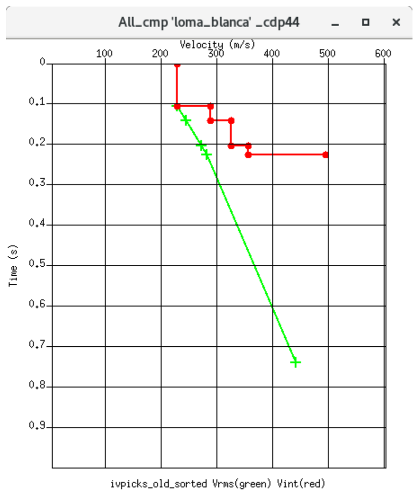
Tool **iVA:** Interactive velocity analysis
Uses: Uses: /home/user/Servilleta/seismics/pl/loma_blanca/053018/iVA.config

SeismicUnixGui Gui



Parameter Names	Values
base_file_name	All_cmp
cdp_first	30
cdp_inc	1
cdp_last	300
data_scale	1
dt_s	1
tmax_s	1
freq	0,3,400,600
number_of_velocities	300
first_velocity	3
velocity_increment	1
min_semblance	.0
max_semblance	.75



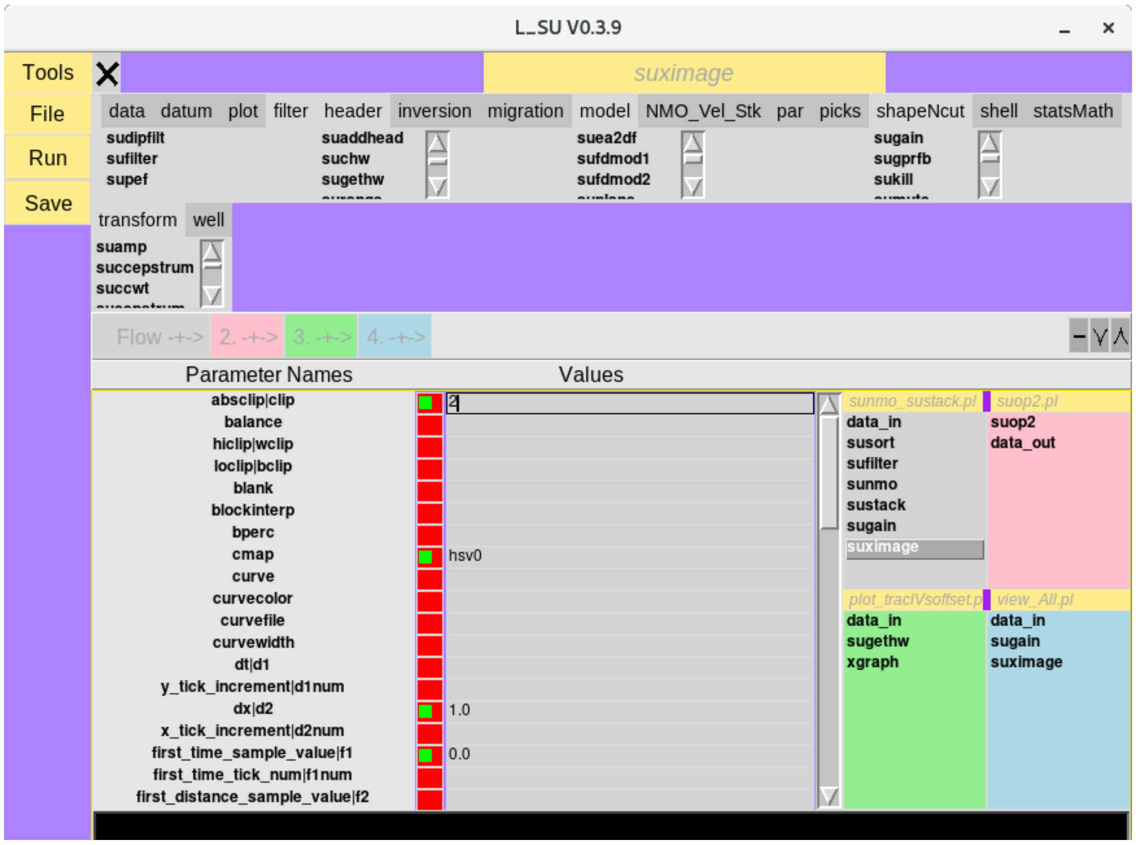


Velocity-time picks interpreted from the semblance plot (far left)

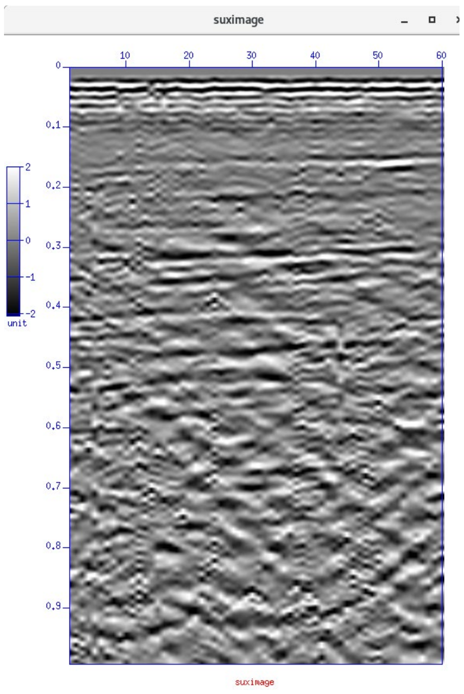
(Left) Velocity verusus time and semblance image (left) and two selected points connected by a line. (Right) CDP/CMP gather analyzed in the adjoining semblance image. Data are NMO-corrected with the two velocity-time values selected in the semblance image.

5.7 STEP 12. Normal Moveout and Stacking

Uses two velocity-time pairs from the iVA above.
STEP 12: SeismicUnixGui Gui:



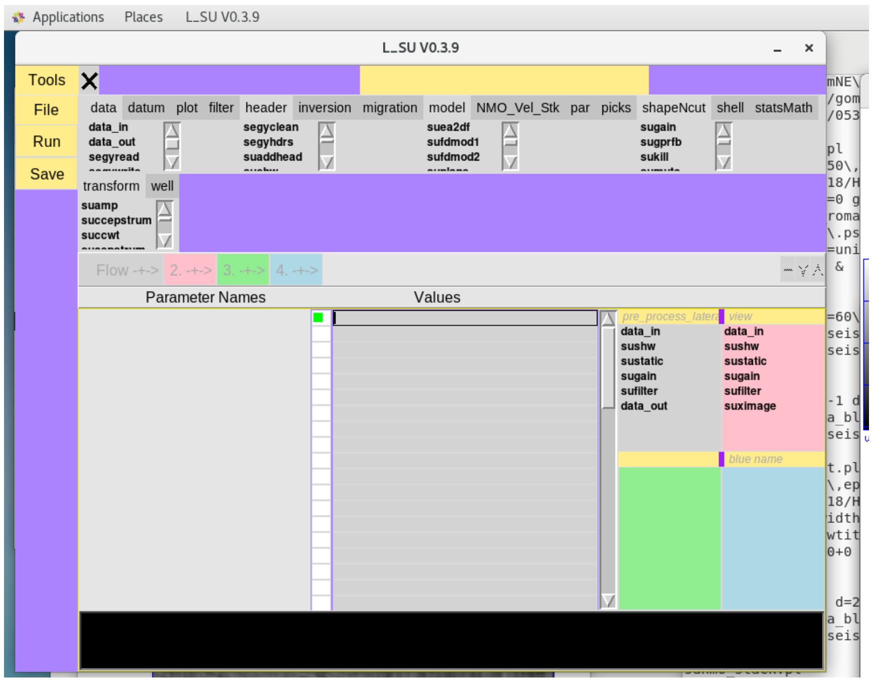
Parameter Names		Values
absclip|clip	🟩	2
balance	🟥	
hiclip|wclip	🟥	
loclip|bclip	🟥	
blank	🟥	
blockinterp	🟥	
bperc	🟥	
cmap	🟩	hsv0
curve	🟥	
curvecolor	🟥	
curvefile	🟥	
curvewidth	🟥	
dt|d1	🟥	
y_tick_increment|d1num	🟥	
dx|d2	🟩	1.0
x_tick_increment|d2num	🟥	
first_time_sample_value|f1	🟩	0.0
first_time_tick_num|f1num	🟥	
first_distance_sample_value|f2	🟥	

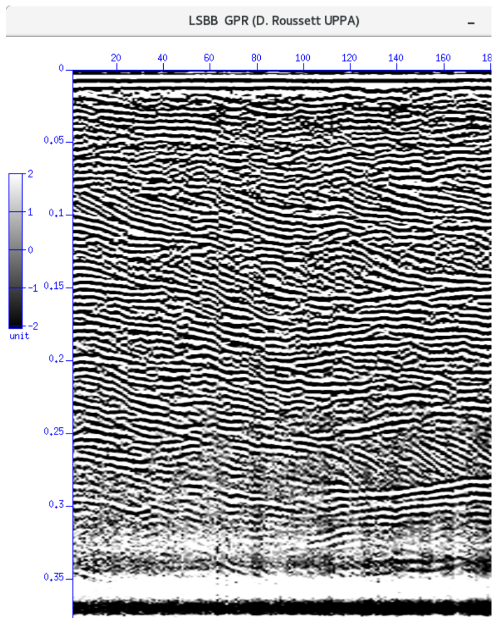


STEP 12: Output image of field data

6 Simple Processing Flow for GPR data

SeismicUnixGui GUI:





Output image of GPR data

7 Perl and Shell script flows generated by SeismicUnixGui

7.1 IRIS Data Set, Socorro, New Mexico

Project Name: Servilleta

STEP 2: GUI Tool Name: Sucat

Uses: /home/user/Servilleta/seismics/pl/loma_blanca/053018/Sucat.config

To run from the command line in the directory where the perl flows are kept (see 1.4.3)

```
% Sucat
```

STEP 5: GUI-generated perl script: suop2.pl

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl suop2.pl
```

To run the bash script from the command line that is generated by suop2.pl:

```
% suop2 \/home\/user\/Servilleta\/seis-
mics\/data\/loma_blanca\/\/053018\/H\/1\/su\/user\/L28HzHit_fromNE\.su
\/home\/user\/Servilleta\/seis-
mics\/data\/loma_blanca\/\/053018\/H\/1\/su\/user\/L28HzHit_fromSW\.su op=diff >
/home/user/Servilleta/seismics/data/loma_blanca//053018/H/1/su/user/L28Hz_Ibeam.su &
```

STEP 5: GUI-generated perl script: view_L28Hz_Ibeam.pl

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl view_L28Hz_Ibeam.pl
```

To run the bash script from the command line that is generated by view_L28Hz_Ibeam.pl:

```
% sufilter f=3\,6\,50\,80 verbose=0 < /home/user/Servilleta/seis-
mics/data/loma_blanca//053018/H/1/su/user/L28Hz_Ibeam.su | sugain agc=1 wagc=0\.1 |
suximage clip=1 cmap=hsv0 d2=1 f1=0 gridcolor=blue labelcolor=blue labelfont=Erg14 legend=1
legendfont=times_roman10 lwidth=16 lx=3 mpicks=\/dev\/tty n1tic=1 n2tic=1 perc=100 plot-
file=plotfile\.ps style=seismic title=suximage titlecolor=red titlefont=Rom22 tmpdir=\.\/
units=unit verbose=1 windowtitle=suximage wperc=100 xbox=500 ybox=500 wbox=550
hbox=550 &
```

STEP6: GUI-generated perl script: SuGeom2.pl

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl SuGeom2.pl
```

To run the bash script from the command line that is generated by SuGeom2.pl:

```
% sushw a=0\,1\,1 j=60\,60\,60 key=sx\,gx\,ep b=0\,1\,0 c=1\,0\,1 < /home/user/Servi-
lleta/seismics/data/loma_blanca//053018/H/1/su/user/L28Hz_Ibeam.su > /home/user/Servi-
lleta/seismics/data/loma_blanca//053018/H/1/su/user/L28Hz_Ibeam_geom2.su &
```

STEP 7: GUI-generated perl script: SuGeom3.pl

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl SuGeom3.pl
```

To run the bash script from the command line that is generated by SuGeom3.pl:

```
% suchw a=0 b=1 c=\-1 d=1 e=1 f=1 key1=offset key2=gx key3=sx < /home/user/Servilleta/seis-
mics/data/loma_blanca//053018/H/1/su/user/L28Hz_Ibeam_geom2.su > /home/user/Ser-
villeta/seismics/data/loma_blanca//053018/H/1/su/user/L28Hz_Ibeam_geom3.su &
```

STEP 7: GUI-generated perl script: plot_traclVsoffset.pl

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl plot_traclVsoffset.pl
```

To run the bash script from the command line that is generated by plot_traclVsoffset.pl:

```
% sugethw key=tracl\,ep output=binary < /home/user/Servilleta/seis-
mics/data/loma_blanca//053018/H/1/su/user/L28Hz_Ibeam_geom3.su | xgraph grid1=dot
grid2=dot gridColor=4 linewidth=0 marksize=1 n=15000 reverse=0 style=normal title=blue win-
dowtitle=windowtitle x1beg=0 x1end=120 x2beg=0 x2end=100 label2=ep label1=tracl -geome-
try 400x600+0+0 &
```

STEP 8: GUI-generated perl script: make_cmp.pl

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl make_cmp.pl
```

To run the bash script from the command line that is generated by plot_traclVsoffset.pl:

```
suchw a=0 b=1 c=1 d=2 e=1 f=1 key1=cdp key2=gx key3=sx < /home/user/Servilleta/seis-
mics/data/loma_blanca//053018/H/1/su/user/L28Hz_Ibeam_geom3.su > /home/user/Ser-
villeta/seismics/data/loma_blanca//053018/H/1/su/user/All_cmp.su &
```

STEP 12: GUI-generated perl script: sunmo_stack.pl

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl sunmo_stack.pl
```

To run the bash script from the command line that is generated by sunmo_stack.pl:

```
% cdp offset < /home/user/Servilleta/seis-
mics/data/loma_blanca//053018/H/1/su/user/All_cmp.su | sufilter f=10\,20\,70\,80 verbose=0
| sunmo invert=0 lmute=25 smute=1\.5 sscale=1 tnmo=0\,1 upward=0 vnmo=100\,600 | sus-
tack key=cdp normpow=0 nrepeat=1 repeat=0 verbose=0 | sugain agc=1 wagc=0\.2 tmp-
dir=\/tmp | suximage clip=2 cmap=hsv0 d2=1 f1=0 gridcolor=blue labelcolor=blue label-
font=Erg14 legend=1 legendfont=times_roman10 lwidth=16 lx=3 mpicks=\/dev\/tty n1tic=1
n2tic=1 perc=100 plotfile=plotfile\.ps style=seismic title=suximage titlecolor=red title-
font=Rom22 tmpdir=\.\/ units=unit verbose=1 windowtitle=suximage wperc=100 xbox=500
ybox=500 wbox=550 hbox=550 &
```

7.2 GPR data

Collected in Low-Noise Underground Gallery (LSBB) in southern France forming by Dominique Rousset of the Université de Pau et des Pays de l'Adour (UPPA) Institut Pluridisciplianire de Recherche Appliqué

Project Name: LSBB

To run the bash script from the command line that is generated by view_LSBB-1.pl:

To run from the command line in the directory where the perl flows are kept (see 1.4.3):

```
% perl view_LSBB-1.pl
```

In the case immediately above, the location of the perl flow is in the following directory: /home/user/LSBB/seismics/data/surface2tunnel/gpr/052011/shielded_antenna/250MHz/su/user/. In this example "user" is the name of the user and should be changed in your particular case.

To run the bash script from the command line that is generated by .pl:

```
% sushw a=9\,1 key=tstat\,cdp b=0\,1 < /home/user/LSBB/seismics/data/surface2tun-
nel/gpr/052011/shielded_antenna/250MHz/su/user/LSBB1\-1.su | sustatic hdrs=1 | sugain
mbal=1 tmpdir=\/tmp | sufilter f=0\,30\,400\,500 verbose=0 | suximage clip=2 cmap=hsv0
d2=1 f1=0 gridcolor=blue labelcolor=blue labelfont=Erg14 legend=1 legendfont=times_ro-
man10 lwidth=16 lx=3 mpicks=\/dev\/tty n1tic=1 n2tic=1 perc=100 plotfile=plotfile\.ps
style=seismic title=suximage titlecolor=red titlefont=Rom22 tmpdir=\.\/ units=unit verbose=1
windowtitle=LSBB\ \ GPR\ \(D\.\ Roussett\ UPPA\) wperc=100 xbox=500 ybox=500 wbox=550
hbox=550 &
```

7.3 General tools

7.3.1 How to mute a data set consisting of a range of multiple gathers

Data set: