SoftwareX 6 (2017) 42–47

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Segy-change: The swiss army knife for the SEG-Y files

Giuseppe Stanghellini [a,*], Gabriela Carrara [b]

[a] ISMAR-CNR, Via Gobetti, 101, 40129 Bologna, Italy
[b] Independent Researcher, Monte San Pietro, Italy

ARTICLE INFO

Article history:
Received 19 January 2016
Received in revised form
20 October 2016
Accepted 13 January 2017

Keywords:
SEG-Y editing
SEG-Y recovery
Seismic processing

ABSTRACT

Data collected during active and passive seismic surveys can be stored in many different, more or less standard, formats. One of the most popular is the SEG-Y format, developed since 1975 to store single-line seismic digital data on tapes, and now evolved to store them into hard-disk and other media as well. Unfortunately, sometimes, files that are claimed to be recorded in the SEG-Y format cannot be processed using available free or industrial packages. Aiming to solve this *impasse* we present segy-change, a pre-processing software program to view, analyze, change and fix errors present in SEG-Y data files. It is written in C language and it can be used also as a software library and is compatible with most operating systems. Segy-change allows the user to display and optionally change the values inside all parts of a SEG-Y file: the file header, the trace headers and the data blocks. In addition, it allows to do a quality check on the data by plotting the traces. We provide instructions and examples on how to use the software.

Code metadata (mandatory)

Current code version	V1.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00019
Legal Code License	GPLv3
Code versioning system used	none
Software code languages, tools, and services used	C++, SDL2
Compilation requirements, operating environments & dependencies	C++, gnu make
If available Link to developer documentation/manual	
Support email for questions	g.stanghellini@ismar.cnr.it

1. Motivation and significance

Active seismic experiments collect data by generating an elastic wave at a given location and recording the reflected or refracted signals (amplitude vs. time) at one (single channel seismic data) or many different (multichannel) receivers. The seismic wave could be impulsive (an explosion) or non impulsive (an amplitude/frequency modulated vibration) generated at controlled position and time on land or at sea. Passive seismic experiments, record low frequency natural seismic waves in a continuous way where the time reference is set by accurate external clocks at the beginning and at the end of the acquisition. Generally seismic data are stored on magnetic tape or disk and are made by series of time–amplitude vector resulting from the digital sampling of received signal. To help data processing and interpretation, as much as possible information is stored into the file and trace headers, such as recording format, number of receivers, record length, sampling interval, time, 2-D or 3-D positions etc.

Among the many formats that were introduced through the years by the seismic industry, under the sponsorship of the Society of Exploration Geophysicists (SEG-Y, SEG-A, SEG-B, SEG-D, etc.) the SEG-Y format [1] has become very popular as an exchange format as well as for direct acquisition on the field.

The presence of a minimum set of information and its organization within a well known de-facto or standard format is required for proper processing data using seismic packages. The words "well-known" and "standard" in this case should be treated carefully, especially when dealing with the SEG-Y format, since deviations from standard are common due to various reasons, such as acquisition software implementation and/or lack of critical

* Corresponding author.
E-mail address: g.stanghellini@ismar.cnr.it (G. Stanghellini).

http://dx.doi.org/10.1016/j.softx.2017.01.003

information during data acquisition. As a consequence, sometimes files that are claimed to be recorded in the SEG-Y format cannot be processed using available packages. Existing free softwares such as Seisee [2] and Seisprho [3] are commonly used to quickly visualize (or image) seismic data files and while allowing some processing, they really do not change the data structure. Segy-change is a pre-processing tool. It allows to easily and efficiently fix the wrong sections within a seismic SEG-Y file, either at the header level or at the data level producing a new, corrected file, ready for being processed or exchanged. Segy-change hence is not an alternative to existing packages but is complementary to them.

2. Software description

`Segy-change` is a command line tool that can be executed inside a terminal window in Unix™, Linux and OsX or in a command tool in Windows©. The program reads an input file in SEG-Y format and after making changes on it can produce an output file in SEG-Y format, options and parameters are available on the command line to set the program behavior and operations. While the lack of a GUI could constitute a potential limit resulting in a steep learning curve for many users, the command line approach unleash many otherwise not achievable functions, like making shell scripts to automate repetitive tasks on big data sets. We believe that the improvement to the user-friendliness of a GUI is not worth of the functionalities that would be lost in our case.

2.1. Available command line switches

The program uses switches and options that can be divided into four main categories: (i) input/output switches, (ii) informative switches, (iii) processing switches and (iv) selecting switches and relatives options. The complete list of available switches are presented hereafter together with a brief description of their usage.

2.1.1. Input/output switches

Segy-change accepts as input a SEG-Y format file and after some processing produces an output le in SEG-Y format. Alternatively a postscript out-put file could be generated as here described.

- `-f` ” and ”`-o`
  set the input and output file names respectively. They can be both the character ‘-’ for stdin and stdout streams, useful for composite commands with the Unix pipe mechanism, widely used, for example, by the package “CWP/SU: Seismic Unix” [4]. An example of using segy-change to recover a SEG-Y file and subsequently piping it with “Seismic Unix” is included below.
- `-do_ps PAPER,N/cm,SCALE`
  generate a postscript plot of the selected portion of the input SEG-Y file. The use of this switch and the “-o” switch are mutually exclusive. The seismic traces are plotted vertically with the wiggle method (amplitude versus time as an oscillating line about a null point). The three required parameters are:
  - `PAPER`: Paper size of the plot (can be A0, A1, A2, A3, A4).
  - `N`: number of traces per cm to plot.
  - `SCALE`: multiplying factor for the trace values, this increase or decrease the amplitude of the waves, darkening or lightening the plot. It can be a decimal number.

2.1.2. Informative switches

Sometimes SEG-Y format files cannot be processed using available packages because some file sections are not correctly written In such cases segy-change informative switches allows to easily and efficiently individuate wrong sections within the file, either at the header level or at the data level, permitting to focus the problem. Informative switches are used to display all kind of informations from SEG-Y files, such as parameters stored inside the SEG-Y and traces headers as well as actual traces values. They are:

- `-info`
  Displays an informative window about the SEG-Y header.
- `-segy_info`
  Displays the full SEG-Y and trace header structure as published by SEG [1]
- `-dump`
  Print all traces values to stdout.
- `-dump_header_fields o0:t0,o1:t1,..`
  Dump the value contained into the specified header parameters. Each header parameter is specified by giving its offset from the header beginning, followed by the type of the parameter separated by the colon character. Available types are integer and short, respectively identified with the characters I and S.
- `-dump_header_fields p0_name,p1_name,..`
  Dump the value contained into the specified header parameters. Each header parameter is specified by giving its name. For this syntax the switch “`-use_names`” must be used.
- `-dump_xy SOURCE|RECEIVER`
  Print x and y location together with their unit of measure to stdout, one of the keywords `SOURCE` or `RECEIVER` must be given, according to the type of entry to print out.
- `-dump_trace_fields o0:t0,o1:t1,..`
  Dump the values of specified trace header parameters. Each header parameter is specified by giving its offset from the header beginning, followed by the type of the parameter. Available types are integer and short, respectively identified with the upper case characters I and S.
- `-dump_trace_fields p0_name,p1_name,..`
  Dump the values of specified trace header parameters. Each header parameter is specified by giving its name. For this syntax the switch “`-use_names`” must be used.
- `-scan`
  Scan the whole SEG-Y file and print useful info.
- `-v verbosity_level`
  Set the verbosity level of the program that is the level of additional informations printed out while running, the informative output is printed to stderr (standard error stream). Available values are 1, 2 or 3. Higher numbers means more verbosity.

2.1.3. Processing switches

These switches allow the user to specify the type of processing and/or modifications to be performed on the input SEG-Y file.

For this purpose a way to reference the parameters stored inside the SEG-Y header and the following trace headers must be given. Segy-change can set the parameter on which to work by two different ways: the first one by offset and the second one by name. Referencing a parameter by its name is trivial, one can specify a particular parameter with its name exactly as it is printed out by the use of the “`segy-change -use_names -segy_info`” command.

Sometimes acquisition software implements the storing of non-standard parameters into optional fields inside the SEG-Y and while this allows for great flexibility of the SEG-Y standard, it nevertheless deviates from it, making the use of “reference by name” useless. For the purpose of referencing such non standard parameters, it is then possible to use a pair: (offset:type), where offset is the address at which the parameter is stored inside the header and type is the type of the parameter. Actually only two types are present: ‘I’ and ‘S’ that is integer*4 and integer*2

respectively. For example, if we know that a non standard integer parameter is stored after 180 bytes from the beginning of the trace header, we can reference it by the pair: 180:I. It is obviously possible to use these offsets to refer to standard parameters too. To display the offsets of all standard parameters the command: "`segy-change -segy_info`" can be used.

2.1.4. SEG-Y and trace Header editing switches

- "`-EBCDIC file`"
  Replace the EBCDIC header section with the content of the specified ASCII file, converting it to EBCDIC before inserting it into the output SEG-Y file.
- "`-traces_per_record NUM`"
  Allows to override the value of this parameter. As stated before, traces are collected by the receivers after the generation of an elastic wave or at predefined starting time. When more than one receiver does the recording one group of many traces are collected at the same time and assembled as a "record". The number of such traces is stored inside the SEG-Y header into the "traces per record" parameter. We encountered many files with that parameter wrongly set to 0, while this can be acceptable for some seismic processing softwares that ask the user for its value, it would not be adequate for some other softwares that read its value from the SEG-Y header. The switch "`-traces_per_record NUM`": allows to override its current value. Used with the switch "`-change_header_fields 3212:S:NUM`" allows to permanently change it to "`NUM`".
- `-change_header_fields O0:T0:V0,O1,T1,V1,..`
  `-change_header_fields`
  `P0_NAME:V0,P1_NAME,V1,..`
  This switch allows to change the values stored inside the SEG-Y header parameters, it is available in two forms, in the first form three additional values are required for each parameter value to change, they are:
  - `OFFSET`: offset used to select which parameter has to be changed, to know what parameters:offsets:types are available use `-segy_info`.
  - `TYPE`: the type of the parameter to change, it can be 'I' or 'S', respectively for integer and short. **The user should pay great attention in using this parameter, if wrong type is given, the results will be unpredictable**.
  - `VAL`: the value to assign to the parameter.

  In the second form (the `-use_names` switch must also be given in the command line), two additional values are required for each field to change, they are:
  - `P_NAME`: the name of the parameter to be changed, to know what parameters are available use
    `-segy_info -use_names switch`.
  - `VAL`: the value to assign to the parameter.

  For example let us assume we want to change the value stored inside the parameter named "Job Identification Number".
  We run the command "`segy-change -segy_info`" to print all available parameters with their offsets to know that the wanted parameter is stored at offset 3200 of the SEG-Y header and is of integer type.
  In this case we can use the switch
  "`-change_header_fields 3200:I:1`" to change its value to 1.
  We can also use "`-change_header_fields`
  `JOB_IDENTIFICATION_NUMBER:1`" together with
  "`-use_names`" to do the same thing, using the parameter name instead of its offset.
- `-change_trace_fields`
  change trace header fields given a file with values. The file must be in the same format as the output obtained by the `-dump_trace_fields` switch, with or without the `-use_names switch`. See the section "Advanced feature" for a clarifying example.
- `-add_xy fname,SOURCE|RECEIVER`
  Allows to change the X, Y coordinates inside the traces headers. Some parameters of the trace header are there to store the coordinates of the sources and receivers of the acquired data. It sometimes happens, for different reasons, that the coordinates are not properly recorded during acquisition. In those cases they can be added later into the SEG-Y files by use of the switch "`-add_xy fname,SOURCE|RECEIVER`". While for this purpose it could be possible to use the switch "`-change_header_fields`", "`-add_xy`" makes things much easier. Only two parameters are needed: a file name containing the coordinates with a syntax that will be explained below and either the 'SOURCE' or 'RECEIVER' keyword according to the entry of the trace header to update.
  The file must contains for each trace the value of: ORIGINAL_FIELD_RECORD_NUMBER, TRACE_SEQUENCE_NUMBER_WITHIN_REEL, TRACE_NUMBER_WITHIN_FIELD_RECORD, x_coordinate, y_coordinate, UNIT_OF_MEASURE.
  It would be a good practice to use the switch "`-dump_xy`" to create an already formatted file that can then be edited and given back to the software using the "`-add_xy`" switch. As usual adding the switch "`-use_names`" to the command line is possible, giving the ability to perform the task using field names instead of offsets.
- `-do_op +|-|*|/:VAL`
  Applies the given arithmetic operation to trace data values, for example to multiply all trace values by 3.0 use:
  -do_op *:3.0
- `-no_header`
  Allows to not write the SEG-Y header to output file, this can be useful to concatenate many SEG-Y files of the same acquisition line into one single file. Incidentally it is useful also when operating with the Open Source seismic processing package "Seismic Unix" since it uses an internal format which is identical to the SEG-Y, with the only difference being the lack of the header.

2.1.5. Internal format conversion

- "`-flip_endianess`"
  Allows to change the endianess of the binary data representation. According to the SEG-Y standard the trace data are stored in binary format in the big-endian representation (the most significant byte comes first). However some commercial acquisition software does the recording in little-endian representation, probably for convenience as they run inside the Intel™ architecture. In situations where this happens, most of the available seismic software packages may not be able to read and process correctly the SEG-Y file. `segy-change` has the capability to convert the endianess from little-endian to big-endian, thanks to the `-flip_endianess` switch. An example showing how to recover a wrongly coded SEG-Y file is included into the EXAMPLES folder.
  Besides being organized as big-endian, the data inside the SEG-Y file may be stored with different binary codification, such as:
  - IBM floating point [5,6]
  - 32 bit signed integer
  - 16 bit signed integer
  - IEEE 754 floating point
- "`-convert FORMAT`"
  Allows the conversion of the trace values from the current format to any of the available formats, to specify the format to convert to the "`FORMAT`" parameter must be one of:
  - `F` for IBM floating point.
  - `I` for 32 bit signed integer.

– S for 16 bit signed integer.
– E for IEEE 754.

2.1.6. Trace and shot renumbering

- "`-irc NUM`" "`-itc NUM`"
  allows to perform respectively record and trace renumbering, starting with the given numbers. The record renumbering is most significant than the trace numbering so that when the trace number is incremented and reaches the "`traces_per_record`" value, it is reset to its initial value and the record number is incremented by one. The record number parameter affected by change is the "`ORIGINAL_FIELD_RECORD_NUMBER`".
  The "`traces_per_record`" value is contained inside the SEG-Y header and can be changed using the correct switches (see below). Great attention must be taken before changing it because a wrong value can compromise the readability of the SEG-Y file.

2.1.7. Selecting switches

Selecting switches are used to specify what part of the original input SEG-Y file should be taken into account. They can be used not only to select a subset of the traces and shots of the original SEG-Y file, but also a subset of the samples that they contains.

Traces can be selected not only specifying their shot and trace numbers, but also by matching their parameter values with a set of given values.

- `-trace first last`
  Sets the trace range to process, the parameter used to set the range is the TRACE_NUMBER_WITHIN_FIELD_RECORD.
- `-num_trace_offset num`
  Sets the offset at which, into the trace header, there is the field to use for the trace number.
  Useful when we do not want to use the TRACE_NUMBER_WITHIN_FIELD_RECORD field that is located at offset 12.
- `-record FIRST LAST`
  Sets the record numbers to process.
- `-all`
  Sets all traces and shots to process. This switch is implied if `-trace` and `-shot` switches are not given.
- `-skip_ntraces N`
  Allows to skip the first n-traces of the input files.
- `-only_ntraces N`
  Sets the number of traces to process.
- `-skip_nsamples N`
  Allows to skip the first n-samples of each trace.
- `-only_nsamples N`
  Sets the number of samples to process for each trace.
- `-x num`
  Allows to skip the first n bytes from SEG-Y file.
- `-only_traces_with O0:T0:V0,O1,T1,V1,...`
  `-only_traces_with P0_NAME:V0,P1_NAME,V1,...`
  Selects only traces with parameters equals to the given values, it is available, as usual, in two forms, in the first form three additional values are required for each parameter value to look for, they are:
  – `OFFSET`: offset used to point to the parameter, to know what parameters:offsets:types are present use `-segy_info`.
  – `TYPE`: the type of the parameter, it can be 'I' or 'S', respectively for integer and short.
  – `VAL`: the value to check for.
  In the second form, used in conjunction with the `-use_names` switch, two additional values are required for each parameter value to look for, they are:
  – `P_NAME`: the name of the parameter, to know what parameters are available use:
  `-segy_info -use_names switch`.
  – `VAL`: the value to check for.

3. Illustrative examples

3.1. Reading SEG-Y header

From hereafter a miscellanea of usage examples is included, starting from very basic functionalities to much more complex ones.

The simplest command one may use is:

```
segy-change -f in.sgy -info
```

In this case the program reads the SEG-Y file header, makes some checks on the consistency of the parameters, report errors if present, prints to the screen the values stored inside some of the most important SEG-Y header parameters and exits with the message "Normal completion". Without specifying the -o switch, the command does not produce any output files. This command is often used to start to work on a SEG-Y file because it prints out informations useful to identify the most important parameters of a SEG-Y file. An example of the output is given below:

```
================= SEGY HEADER =====================
                                      Line nr. : 1
                                      Reel nr. : 1
            Number of data traces per record : 48
  Sample interval for this reel (microseconds) : 1000
Number of samples per data trace for this reel : 4000
                      Data sample format code : 1
     Measurement system: 1 = feet, 2 = meters : 2
```

3.2. Converting the trace data

In cases where the acquired data inside the traces are encoded in a format that is not supported natively by the seismic software package in use, we may convert the data into one of the available format. For example if we want to convert the current format to IBM floating point format, we should use the command:

```
segy-change -f in.sgy -o out.sgy
            -convert F -all
```

where:

- "`-f in.sgy`" specifies the input file name.
- "`-o out.sgy`" specifies the output file name.
- "`-all`" specifies to process all the records and traces belonging to the SEG-Y file.
- "`-convert F`" specifies to convert traces data to IBM floating point format.

By executing it, segy-change updates also the SEG-Y header parameter "Data sample format code" according to the correct format.

Be aware that while the conversion is always possible, in some cases it is not straightforward; for example if we are in need of converting from IBM floating point to integer we must pay great attention to the loss of precision that this conversion implies. In those situations the conversion is however possible by scaling the data with the `-do_op` switch. To clarify that let us assume that we have a SEG-Y stored in IBM floating point and that we must convert it to integer codification, the first thing to do is to know the range of variability of the values stored inside the data blocks of the traces, for this purpose we can use:

```
segy-change -f in.segy -all -scan
```

For the most cases the printed min and max values will not scale well inside the range allowed by the integer representation, in those cases a scaling will be necessary. For example let us assume that trace data is in the $[-1, 1]$ range, obviously we cannot convert the data into integer codification without prior scaling it, otherwise we will end up with a sequence of 0's with some 1's. Keeping in mind that the range of variability of an integer is $[-32768, 32767]$, the necessary scaling can then be achieved with:

```
segy-change -f in.segy -do_op *:32767
            -convert I -o out.segy
```

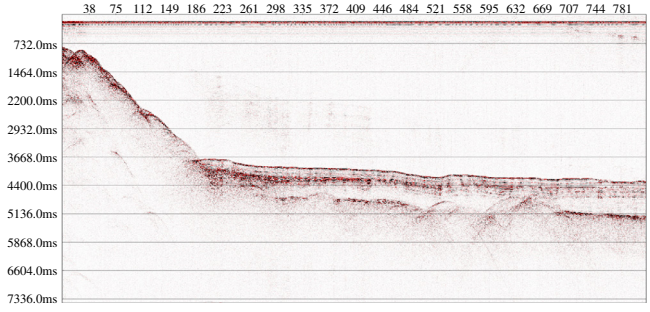

Fig. 1. Example of use of -view switch with EXAMPLE/6_channel.seg.

3.3. Displaying SEG-Y data

When operating with SEG-Y files, it can be very handful to display the data. This can be accomplished with segy-change itself or by using segy-change in conjunction with the open source seismic processing software CWP/SU [4].

To display interactively the SEG-Y data (only in X11) one can use `-view` switch (see Fig. 1). A window will open showing all SEG-Y data, the view can then be zoomed in and out using the SHIFT KEY + MOUSE DRAG, can be shifted using MOUSE DRAG and increased and decreased in contrast by using respectively * and / KEYS.

Other active keys are, 'p' to change palette, 'KEYPAD-*' to increase signal, 'KEYPAD-/' to lower signal.

While moving the mouse over the plot some informations such as record number, trace number, sample number and sample value of the data below the mouse pointer are shown on the terminal from which the software was executed.

To do a postscript plot we may use the `-do_ps` switch, that produces a postscript plot of the seismic data contained in the SEG-Y file. `-do_ps` requires three additional parameters separated by colons: the paper size, the width of the plot in number of traces, and a scale factor. For example to plot a SEG-Y with 10 traces per cm, the following command line may be used:

```
segy-change -f in.sgy
            -do_ps A4,10,0.000004
            > out.ps
```

The scaling factor is a multiplying factor applied at all traces values and has the effect to darken the plot (higher values means darker plot).

To do the plotting using CWP/SU we may use:

```
segy-change -f in.sgy
            -o -
            -only_n_traces 200 |
            segyread tape=- |
            supswigb n2=200
            va=5 nbpi=600
            > out.ps
```

Where segyread and supswigb are command line programs belonging to the CWPSU software package.

3.4. Advanced features

The program allows to view and edit any information contained into the SEG-Y file. This can be done using the "`-change_header_fields`", "`-change_trace_fields`" and "`-convert`" switches. We will focus here on the first two switches being the third one fully explained before, with them we can modify the values stored into the SEG-Y header as well as the trace headers.

The information within the SEG-Y header and trace headers are stored in binary format, each header contains a number of parameters. Each of them is stored into fields (memory locations) at increasing offsets from the beginning of file or trace. The full list of these parameters together with the offsets of their fields can be obtained from segy-change, by executing:

```
segy-change -segy_info
```

This includes all information for all SEG-Y header and trace header parameter names, parameter offsets and parameter types and can help when using the "`-change_header_fields`" switch.

For example, let us assume that we want to store the value "1" into the parameter named "Job Identification Number". To this purpose, the switch should given as:
"`-change_header_fields 3200:I:1`" where 3200 is the offset, I is the type (I=integer) and "1" is the value to store.

Multiple editing of more than one parameter can be accomplished by using a colon character to separate parameters specification, with the following syntax:
"`-change_header_fields 3200:I:1,3204:I:12`" In this case the value of the "Line Number" field is also changed to 12. We can specify in this way as many fields as desired, each separated by the comma character.

The "`-change_trace_fields`" works in a slightly different way, since in this case we have to specify the parameter to be changed and also the trace on which perform the changes, it would be almost impossible to do it on the command line. What we have to do is then to give to the program a file containing a list of traces and parameters to modify. To simplify the building of such file we may use the "`-dump_trace_fields`" switch that produces a list of the trace header parameters and their current values in the same format as required by the "`-change_trace_fields`" switch, the produced file can then be edited, modified and then given back to the program.

For example we may use:

```
segy-change -f sample.sgy
            -dump_trace_fields 28:S,30:S
            > tracefields.txt
```

where "28" and "30" are the offsets of
"`TRACE_IDENTIFICATION_CODE`" and
"`NUMBER_OF_VERTICALLY_SUMMED_TRACES`" parameters respectively.

By executing the aforementioned command the file named tracefields.txt is created, containing eventually:

```
Rec/Seq/Num = 9892/9892/1 : fields = 28,S,1; 30,S,0
Rec/Seq/Num = 9893/9893/1 : fields = 28,S,1; 30,S,0
Rec/Seq/Num = 9894/9894/1 : fields = 28,S,1; 30,S,0
Rec/Seq/Num = 9895/9895/1 : fields = 28,S,1; 30,S,0
Rec/Seq/Num = 9896/9896/1 : fields = 28,S,1; 30,S,0
Rec/Seq/Num = 9897/9897/1 : fields = 28,S,1; 30,S,0
..............
(a row for each trace is present hereafter
following the same format as the above ones)
```

Every line has a precise format: is divided into two parts separated by a colon. The first one contains the Rec/Seq/Num fields that are the values contained within ORIGINAL_FIELD_RECORD_NUMBER, the TRACE_SEQUENCE_NUMBER_WITHIN_REEL and the TRACE_NUMBER_WITHIN_FIELD_RECORD respectively. The second part contains a semicolon separated list of fields, each of them defined by three numbers separated by commas representing in order: parameter offset, parameter type and the new value to store.

We can then manually edit the file and use it back with the "`-change_trace_fields`" switch, or write a simple command script (usually in perl or some other scripting language) to automate the process of changing the file to our needs. To give it back to segy-change ww can use:

```
segy-change -f file.sgy
            -o modified_file.sgy
            -change_trace_fields tracefields.txt
            -all
```

The Rec/Seq/Num numbers in the text file are mandatory and must be in the same order as the SEG-Y file we want to process.

While the authors prefer to specify the parameters by their offsets as this is more powerful and allows to change also the non standard parameters that sometimes are present into the headers, for the sake of clearance and simplicity it is also possible to use parameter names.

In the previous case the sequence of commands that does exactly the same things are:

```
segy-change -f file.sgy
-dump_trace_fields
    TRACE_IDENTIFICATION_CODE,
    NUMBER_OF_VERTICALLY_SUMMED_TRACES
> tracefields.txt
```

giving as output a file containing:

```
Rec/Seq/Num = 9892/9892/1 : fields = TRACE_IDENTIFICATION_CODE,1;
                            NUMBER_OF_VERTICALLY_SUMMED_TRACES,0
Rec/Seq/Num = 9893/9893/1 : fields = TRACE_IDENTIFICATION_CODE,1;
                            NUMBER_OF_VERTICALLY_SUMMED_TRACES,0
Rec/Seq/Num = 9894/9894/1 : fields = TRACE_IDENTIFICATION_CODE,1;
                            NUMBER_OF_VERTICALLY_SUMMED_TRACES,0
Rec/Seq/Num = 9895/9895/1 : fields = TRACE_IDENTIFICATION_CODE,1;
                            NUMBER_OF_VERTICALLY_SUMMED_TRACES,0
Rec/Seq/Num = 9896/9896/1 : fields = TRACE_IDENTIFICATION_CODE,1;
                            NUMBER_OF_VERTICALLY_SUMMED_TRACES,0
..............
(a row for each trace is present hereafter
following the same format as the above ones)
```

That again, once edited to our needs, can be used to change the parameters values with the command:

```
segy-change -f file.sgy
-o modified_file.sgy
-change_trace_fields tracefields.txt
-use_names
-all
```

4. Impact

A SEG-Y data file usually contains large amount of data, often exceeding several gigabytes. In fact, the SEG-Y standard was developed to be able to contain a stream of data limited only by the size of the storing support, magnetic tape or disk, and by the size of files that can be handled on disk by a selected computer and Operating System. This can be a critical issue, since modern seismic acquisition systems can produce data at rates of many megabytes per second for a number of traces as high as 512 or even larger. The structure of the SEG-Y allows to write relatively simple programs to handle data stored using this format. When designing `segy-change` our aim was allowing the user to modify the file header leaving untouched the data blocks. This is particularly appreciated by the scientists who are always concerned about data integrity. In our experience missing information or wrongly written acquisition parameters (i.e. acquisition geometry) are common problems found in seismic data handling. In those cases, huge data sets composed often by hundred if not thousands of files, cannot be processed efficiently by hand, file after file but a more automated way must be implemented. When a way to fix them exists, segy-change can be used to prepare scripts that automatically correct that huge amount of data.

`Segy-change` is written in C language and can be compiled in most of the existing operating systems, the only prerequisite being the existence of a standard C compiler and some system libraries. To date, `segy-change` has been compiled and tested for Linux, Solaris, OsX and Windows workstations. The distributed source can be compiled to create an executable or included into a user developed program by appropriate compiler's directives. In this way it is very flexible and allows the user to develop his own extensions.

5. Conclusions

We have shown how `segy-change` can be used to recover seismic data stored inside badly or incomplete formatted SEG-Y files, how to use it to display the content of SEG-Y files and also how with little efforts it can be expanded in order to deal with a variety of problems.

We believe that this software might be useful to fix corrupted or badly formatted SEG-Y files, and could also be used in the field for a fast, inexpensive and reliable quality control of the data just after their acquisition.

In our research institution, we estimated that the last 30 years of acquisition at sea produced at least 10 000 kilometers of seismic lines that were stored in SEG-Y format. Such big amount of data is prone to all discussed errors. All over the world public research and private institutions have acquired and are still acquiring huge amount of SEG-Y seismic data. Assuming that even only a small part of the data could have acquisition problems we are quite confident that all laboratories in the world that manage seismic data will benefit from our software.

References

[1] Barry K, Cavers D, Kneale C. Recommended standards for digital tape formats. Geophysics 1975;40(2):344–52. http://dx.doi.org/10.1190/1.1440530.
[2] Pavlukhin S. Seisee user manual, 2004. http://www.dmng.ru/docs/Freeware/seisee/html/SeiSeeEng.htm.
[3] Gasperini L, Stanghellini G. Seisprho: An interactive computer program for processing and interpretation of high-resolution seismic reflection profiles. Comput Geosci 2009;35:1497–507.
[4] Stockwell JJW. The CWP/SU: Seismic Unix package. Comput Geosci 1999;25(4):415–9.
[5] IBM, IBM System/360 principles of Operations, Poughkeepsy, NY, USA; 1967.
[6] Fleming G. Data format and conversion information for heritage data at the National Space Science Data Center, 2002 http://nssdc.gsfc.nasa.gov/nssdc/formats/.