SEGY-SAK is a file format used in the field of geophysics, particularly in the acquisition and processing of seismic data. It stands for "SEG Y with Standard Annotation and Keys", and is an extension of the SEG Y format, which is a standard for digital storage and exchange of seismic data.

The main purpose of SEGY-SAK is to provide a standardized way of annotating and organizing seismic data, which can be very complex and difficult to interpret without proper documentation. The format includes a set of predefined headers and keys that describe important information about the seismic data, such as the acquisition parameters, processing steps, and interpretation results.

SEGY-SAK was developed by the Society of Exploration Geophysicists (SEG) as part of its efforts to promote standardization and interoperability in the field of geophysics. By using a standardized format like SEGY-SAK, different organizations and software tools can more easily exchange and analyze seismic data, which can lead to more accurate and reliable interpretations.

Overall, SEGY-SAK is an important tool for the management and analysis of seismic data, and is widely used in the oil and gas industry, as well as in academic research and other fields of geophysics.

The SEGY-SAK Python library provides a set of tools for reading, writing, and processing SEGY-SAK files in Python. Here are some of the things you can do with the library:

1. Read and write SEGY-SAK files: The library provides functions for reading and writing SEGY-SAK files in Python. This allows you to read in seismic data from SEGY-SAK files, and to save processed data back to SEGY-SAK files.
2. Extract data and headers: The library allows you to extract specific data and headers from SEGY-SAK files, such as the trace data, trace headers, and file headers. This can be useful for analyzing specific aspects of the seismic data.
3. Quality control: The library includes functions for checking the quality and integrity of SEGY-SAK files. For example, you can check that the file headers are consistent with the trace headers, and that the trace data is valid.
4. Processing and analysis: The library provides functions for processing and analyzing seismic data in SEGY-SAK files. For example, you can filter the data, perform amplitude scaling, and apply time shifts.
5. Visualization: The library includes functions for visualizing seismic data from SEGY-SAK files. For example, you can create seismic sections and plots of seismic data.

Overall, the SEGY-SAK Python library is a powerful tool for working with seismic data in SEGY-SAK format, and it can be used for a wide range of tasks in geophysics and related fields.

Check this link : https://segysak.readthedocs.io/en/latest/examples.html

```
#import libraries
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import pathlib
import segysak
```

```
# Define a Path object representing the file path "/content/volve10r12-full-twt-sub3d.sgy"
V3D_path = pathlib.Path("/content/volve10r12-full-twt-sub3d.sgy")

# Print the string "3D", the value of V3D_path, and whether the file or directory at that path exists
print("3D", V3D_path, V3D_path.exists())
```

```
3D /content/volve10r12-full-twt-sub3d.sgy True
```

SEGY (short for SEG-Y) is a file format used in the oil and gas industry for storing and exchanging seismic data. SEGY files consist of a sequence of binary records, each of which contains header information followed by a data trace. Here are the main components of a SEGY file:

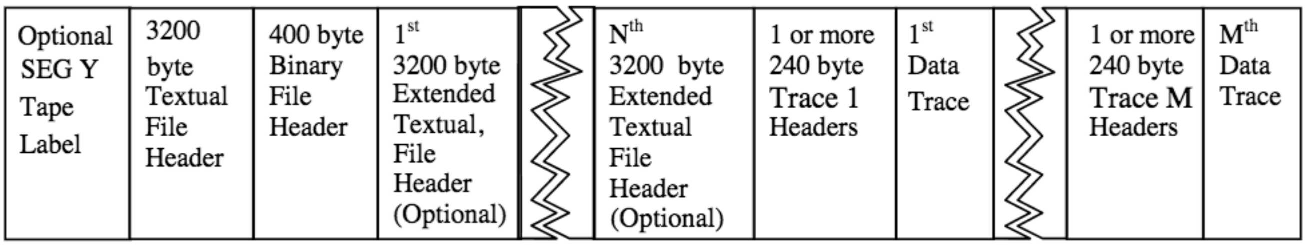
1. File Header: The file header is the first 3200 bytes of the SEGY file and contains general information about the data in the file, such as the **sample rate, number of traces, and data format**. The file header is divided into several sections, each of which contains specific information.
2. Trace Header: The trace header is a 240-byte section that precedes each trace in the file. It contains information about **the data trace, such as the trace number, offset, and gain**.
3. Data Trace: The data trace is the actual seismic data that is stored in the file. Each data trace consists of a sequence of samples, with each sample **representing the amplitude of the seismic wave at a specific point in time**.
4. End of File: The end of the file is a special record that marks the end of the SEGY file.

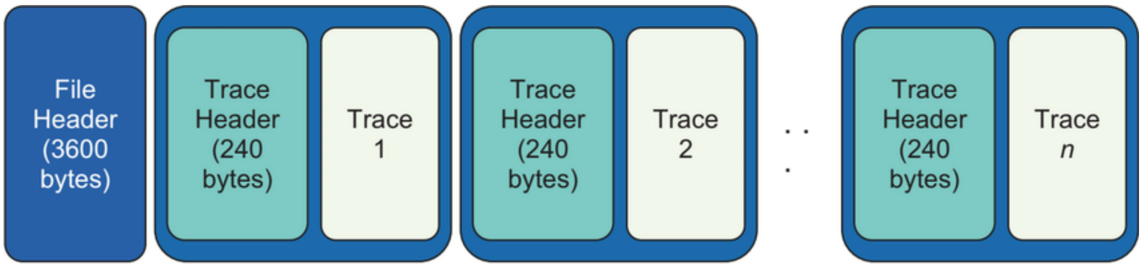
SEGY files can be stored in different data formats, such as IBM floating-point, IEEE floating-point, or two's complement integer. The data format is specified in the file header.

SEGY files can also contain additional optional information, such as text headers, binary headers, or extended text headers. These additional headers can be used to store metadata about the data, such as acquisition parameters or processing history.

SEGY files are typically large, with file sizes ranging from several megabytes to several gigabytes. They are commonly used

Double-click (or enter) to edit


Optional SEG Y Tape Label
3200 byte Textual File Header
400 byte Binary File Header
1st 3200 byte Extended Textual, File Header (Optional)
Nth 3200 byte Extended Textual File Header (Optional)
1 or more 240 byte Trace 1 Headers
1st Data Trace
1 or more 240 byte Trace M Headers
Mth Data Trace



Scan SEG-Y headers

A basic operation would be to check the text header included in the SEG-Y file. The get_segy_texthead function accounts for common encoding issues and returns the header as a text string.

```
from segysak.segy import get_segy_texthead

get_segy_texthead(V3D_path)
```

```
Text HeaderC 1 SEGY OUTPUT FROM Petrel 2017.2 Saturday, June 06 2020
10:15:00
C 2 Name: ST10010ZDC12-PZ-PSDM-KIRCH-FULL-T.MIG_FIN.POST_STACK.3D.JS-
017534
ÝCroC 3
C 4 First inline: 10090  Last inline: 10150
C 5 First xline:  2150   Last xline:  2351
C 6 CRS: ED50-UTM31 ("MENTOR:ED50-UTM31:European 1950 Based UTM, Zone 31
North,
C 7 X min: 433955.09 max: 436589.56 delta: 2634.47
C 8 Y min: 6477439.46 max: 6478790.23 delta: 1350.77
C 9 Time min: -3402.00 max: -2.00 delta: 3400.00
C10 Lat min: 58.25'52.8804"N max: 58.26'37.9493"N delta: 0.00'45.0689"
C11 Long min: 1.52'7.1906"E max: 1.54'50.9616"E delta: 0.02'43.7710"
C12 Trace min: -3400.00 max: -4.00 delta: 3396.00
C13 Seismic (template) min: -58.55 max: 54.55 delta: 113.10
C14 Amplitude (data) min: -58.55 max: 54.55 delta: 113.10
C15 Trace sample format: IEEE floating point
C16 Coordinate scale factor: 100.00000
C17
C18 Binary header locations:
C19 Sample interval             : bytes 17-18
C20 Number of samples per trace : bytes 21-22
C21 Trace date format           : bytes 25-26
C22
C23 Trace header locations:
C24 Inline number               : bytes 5-8
C25 Xline number                : bytes 21-24
C26 Coordinate scale factor     : bytes 71-72
C27 X coordinate                : bytes 73-76
C28 Y coordinate                : bytes 77-80
C29 Trace start time/depth      : bytes 109-110
C30 Number of samples per trace : bytes 115-116
C31 Sample interval             : bytes 117-118
C32
C33
```

```
#to investigate the trace header data more deeply, then segy_header_scan can be used to report basic statistics
from segysak.segy import segy_header_scan

scan = segy_header_scan(V3D_path)
scan
```

100% 1.00k/1.00k [00:00<00:00, 3.71k traces/s]

	byte_loc	count	mean	std	min
TRACE_SEQUENCE_LINE	1	1000.0	100.54	57.831072	1.0
TRACE_SEQUENCE_FILE	5	1000.0	10091.98	1.407687	10090.0
FieldRecord	9	1000.0	10091.98	1.407687	10090.0
TraceNumber	13	1000.0	100.54	57.831072	1.0
EnergySourcePoint	17	1000.0	0.00	0.000000	0.0
...	...	...	...	...	...
SourceEnergyDirectionMantissa	219	1000.0	0.00	0.000000	0.0
SourceEnergyDirectionExponent	223	1000.0	0.00	0.000000	0.0
SourceMeasurementMantissa	225	1000.0	0.00	0.000000	0.0
SourceMeasurementExponent	229	1000.0	0.00	0.000000	0.0

```
# To retreive the raw header content use segy_header_scrape. Setting partial_scan=None will return the full dataframe of trace header in
from segysak.segy import segy_header_scrape

scrape = segy_header_scrape(V3D_path, partial_scan=1000)
scrape
```

100% 1.00k/1.00k [00:00<00:00, 9.53k traces/s]

	TRACE_SEQUENCE_LINE	TRACE_SEQUENCE_FILE	FieldRecord	TraceNumber
0	1	10090	10090	1
1	2	10090	10090	2
2	3	10090	10090	3
3	4	10090	10090	4
4	5	10090	10090	5
...	...	...	...	...
995	188	10094	10094	188
996	189	10094	10094	189
997	190	10094	10094	190
998	191	10094	10094	191
999	192	10094	10094	192

1000 rows × 89 columns

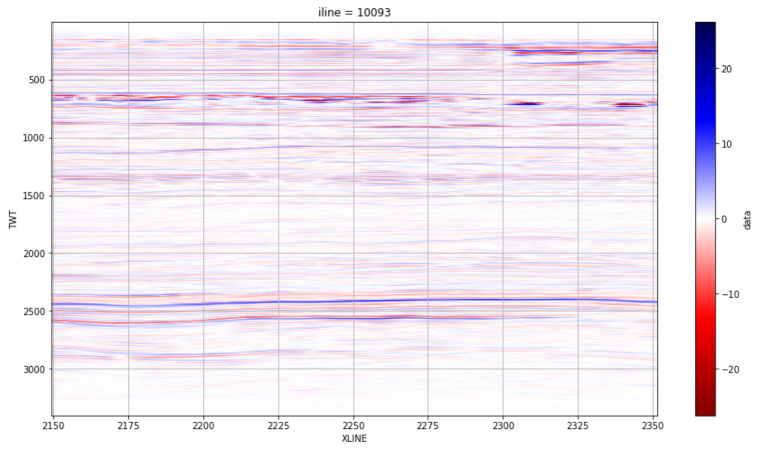
```
from segysak.segy import segy_loader, well_known_byte_locs

V3D = segy_loader(V3D_path, iline=189, xline=193, cdpx=73, cdpy=77, vert_domain="TWT")
V3D
```

```
100%                                   12.3k/12.3k [00:02<00:00, 4.86k
                                       traces/s]
Loading as 3D
Fast direction is INLINE_3D
Converting SEGY:                          12.3k/12.3k [00:04<00:00, 3.27k
100%                                      traces/s]
xarray.Dataset
▸ Dimensions:      (iline: 61, xline: 202, twt: 850)
▾ Coordinates:
    iline          (iline)              uint16   10090 10091 10092 ... 10149 1...
    xline          (xline)              uint16   2150 2151 2152 ... 2349 2350 ...
    twt            (twt)                float64  4.0 8.0 12.0 ... 3.396e+03 3.4e+...
    cdp_x          (iline, xline)       float32  4.364e+05 4.364e+05 ... 4.341...
    cdp_y          (iline, xline)       float32  6.477e+06 6.477e+06 ... 6.479...
▾ Data variables:
    data           (iline, xline, twt)  float32  0.02057 0.02204 0.01966 ... 0....
▾ Indexes:
    iline          PandasIndex
    xline          PandasIndex
    twt            PandasIndex
▾ Attributes:
    ns :           None
    sample_rate :  4.0
    text :         C 1 SEGY OUTPUT FROM Petrel 2017.2 Saturday, June 06 2020 1
                   0:15:00
                   C 2 Name: ST10010ZDC12-PZ-PSDM-KIRCH-FULL-T.MIG_FIN.PO
                   ST_STACK.3D.JS-017534
                   ÝCroC 3
                   C 4 First inline: 10090  Last inline: 10150
                   C 5 First xline:  2150   Last xline:  2351
                   C 6 CRS: ED50-UTM31 ("MENTOR:ED50-UTM31:European 1950 B
                   ased UTM, Zone 31 North,
                   C 7 X min: 433955.09 max: 436589.56 delta: 2634.47
                   C 8 Y min: 6477439.46 max: 6478790.23 delta: 1350.77
                   C 9 Time min: -3402.00 max: -2.00 delta: 3400.00
                   C10 Lat min: 58.25'52.8804"N max: 58.26'37.9493"N delta: 0.00'4
                   5.0689"
                   C11 Long min: 1.52'7.1906"E max: 1.54'50.9616"E delta: 0.02'43.
                   7710"
                   C12 Trace min: -3400.00 max: -4.00 delta: 3396.00
                   C13 Seismic (template) min: -58.55 max: 54.55 delta: 113.10
                   C14 Amplitude (data) min: -58.55 max: 54.55 delta: 113.10
                   C15 Trace sample format: IEEE floating point
                   C16 Coordinate scale factor: 100.00000
                   C17
                   C18 Binary header locations:
                   C19 Sample interval           : bytes 17-18
                   C20 Number of samples per trace : bytes 21-22
                   C21 Trace date format         : bytes 25-26
                   C22
                   C23 Trace header locations:
                   C24 Inline number             : bytes 5-8
                   C25 Xline number              : bytes 21-24
                   C26 Coordinate scale factor   : bytes 71-72
                   C27 X coordinate              : bytes 73-76
                   C28 Y coordinate              : bytes 77-80
                   C29 Trace start time/depth    : bytes 109-110
                   C30 Number of samples per trace : bytes 115-116
                   C31 Sample interval           : bytes 117-118
                   C32
                   C33
```

```
#Visualising data
fig, ax1 = plt.subplots(ncols=1, figsize=(15, 8))
iline_sel = 10093
V3D.data.transpose("twt", "iline", "xline", transpose_coords=True).sel(
    iline=iline_sel
).plot(yincrease=False, cmap="seismic_r")
plt.grid("grey")
plt.ylabel("TWT")
plt.xlabel("XLINE")
```

```
Text(0.5, 0, 'XLINE')
```



✓ 1s completed at 1:57 PM