# pygeo Documentation

***Release devel***

**Brendan Smithyman**

June 15, 2012

# CONTENTS

Contents:

CHAPTER

ONE

# PYGEO.SEGYREAD

The `pygeo.segyread` submodule is designed to allow interaction with geophysical (seismic) datafiles that use the SEG-Y format. The primary purpose of the package is to allow *read-only* access to the SEG-Y data format, though several provisions are made for creating or updating SEG-Y files.

## 1.1 SEGYFile

The `SEGYFile` class represents the SEG-Y or SU datafile efficiently, and initially loads only the metadata necessary to set certain parameters, viz: filesize, endian, data format. Several objects are created inside the namespace of the `SEGYFile` object, viz: **thead**, **bhead**, **trhead**, **endian**, **mendian**, **ns**, **ntr**, **filesize**, **ensembles**.

**class** `pygeo.segyread.`**`SEGYFile`**

Provides read access to a SEG-Y dataset (headers and data).

**Parameters**

- **filename** (*str*) – The system path of the SEG-Y file to open.
- **verbose** (*bool*) – Controls whether diagnostic information is printed. This includes status messages when endian and format conversions are made, and may be useful in diagnosing problems.
- **majorheadersonly** (*bool*) – Only read certain specific headers (legacy). No longer relevant, but may be expected by some old programs.
- **isSU** (*bool*) – Controls whether SEGYFile treats the datafile as a Seismic Unix variant SEG-Y file. This overrides assumptions for endianness and format, and presumes the absence of the 3200-byte text header and 400-byte binary header.
- **endian** (*str*) – Allows specification of file endianness [Foreign,Native,Little,Big]. By default this is auto-detected using a heuristic method, but it will fail for e.g., SEG-Y files that contain all zeros, or very noisy data.
- **usemmap** (*bool*) – Controls whether memory-mapped I/O is used. Default True. In most (all?) cases this should be more efficient, and will be disabled automatically if not supported.

**Returns** SEGYFile instance

**Variables**

- **thead** – *str* – contains an ASCII-encoded translation of the EBCDIC 3200-byte tape header.
- **bhead** – *dict* – contains key:value pairs describing the data in the 400-byte binary reel header.

- **trhead** – `SEGYTraceHeader` instance – acts like a list of all the trace headers. Individual items each return a dictionary that contains key:value pairs describing the data in the trace header.
- **endian** – *str* – describing the endian of the datafile.
- **mendian** – *str* – autodetected machine endian.
- **ns** – *int* – number of samples in each trace.
- **ntr** – *int* – number of traces in dataset.
- **filesize** – *int* – size of datafile in bytes.
- **ensembles** – *dict* – only exists if the experimental function `SEGYFile._calcEnsembles()` is called. Maps shot gather numbers to trace numbers. *Experimental*

**`__getitem__`**`()`

Returns traces from the open seismic dataset, with support for standard Python slice notation. Trace numbers are zero-based.

**Parameters** **index** – Slice object or trace number (using zero-based numbering).

**Returns** ndarray – 2D array containing (possibly non-adjacent) seismic traces

**`findTraces`**`()`

Finds traces whose header values fall within a particular range. Trace numbers are 1-based, i.e., for use with readTraces.

**Parameters**

- **key** (*str*) – Key value of trace header to scan (uses lower-case SU names; see TRHEADLIST.
- **kmin** (*int*) – Minimum key value (inclusive).
- **kmax** (*int*) – Maximum key value (inclusive).

**`readTraces`**`()`

Returns trace data as a list of numpy arrays (i.e. non-adjacent trace numbers are allowed). Requires that traces be fixed length.

**Parameters** **traces** (*list, None*) – List of traces to return, using 1-based trace numbering. Optional; if omitted, all traces are returned.

**Returns** ndarray – 2D array containing (possibly non-adjacent) seismic traces

Changed in version devel. This is now a legacy interface, and is superseded by the __getitem__ interface, which uses standard Python slice notation.

**`sNormalize`**`()`

Utility function that takes seismic traces and returns an amplitude normalized version.

**Parameters** **traces** (*ndarray, list*) – List or array of traces to normalize.

**`writeFlat`**`()`

Outputs seismic traces as a flat file in IEEE floating point and native endian.

**Parameters** **outfilename** (*str*) – Filename for new flat datafile.

*Experimental*

**`writeSEGY`**`()`

Outputs seismic traces in a new SEG-Y file, optionally using the headers from the existing dataset.

**Parameters**

- **outfilename** (*str*) – Filename for new SEG-Y datafile.
- **traces** (*ndarray, list*) – Array of seismic traces to output.
- **headers** (*list, None*) – List of three headers: [thead, bhead, trhead]. If omitted, the existing headers in the SEGYFile instance are used. *thead* is an ASCII-formatted 3200-byte text header. *bhead* is a list of binary header values similar to SEGYFile.bhead. *trhead* is a list or list-like object of trace header values.

**writeSU**()

Outputs seismic traces in a new CWP SU file, optionally using the headers from the existing dataset.

**Parameters**

- **outfilename** (*str*) – Filename for new SU datafile.
- **traces** (*ndarray, list*) – Array of seismic traces to output.
- **trhead** (*list, None*) – List or list-like object of trace header values. If omitted, the existing headers in the SEGYFile instance are used.

## 1.2 SEGYTraceHeader

The `SEGYTraceHeader` class efficiently indexes the trace headers of the parent `SEGYFile`. This makes it possible to access the headers of an individual trace, or a series of traces without prefetching them from the file on disk. It interfaces directly with the conventional or memory-mapped file object inside the `SEGYFile` object.

**class** pygeo.segyread.**SEGYTraceHeader**

Provides read access to trace headers from an existing `SEGYFile` instance.

**Parameters**

- **sf** – Parent class to attach to.
- **sf** – `SEGYFile`

**Returns** `SEGYTraceHeader` instance

**__getitem__**()

Returns dictionary (or list of dictionaries) that maps header information for each defined SEG-Y trace header. SU style names, see TRHEADLIST.

**Parameters** **index** – Slice object or trace number (using zero-based numbering).

**Returns** dict, list

CHAPTER

TWO

# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

# PYTHON MODULE INDEX

# INDEX