

Package ‘gemlog’

October 6, 2017

Type Package

Title File Conversion for 'Gem Infrasound Logger'

Version 0.26

Date 2017-10-05

Author Jake Anderson

Maintainer Jake Anderson <ajakef@gmail.com>

Description Reads data files from the 'Gem infrasound logger' for analysis and converts to segy format (which is convenient for reading with traditional seismic analysis software). The Gem infrasound logger is an in-development low-cost, lightweight, low-power instrument for recording infrasound in field campaigns; email the maintainer for more information.

Depends signal

License GPL

LazyLoad yes

Repository CRAN

Date/Publication 2017-10-06 03:15:50 UTC

NeedsCompilation no

R topics documented:

| | |
|--------------------------|---|
| gemlog-package | 2 |
| Convert | 3 |
| gem_metadata | 5 |
| PlotMetadata | 6 |
| ReadGem | 7 |

| | |
|--------------|----------|
| Index | 9 |
|--------------|----------|

gemlog-package

*File Conversion for 'Gem Infrasound Logger'***Description**

Reads data files from the 'Gem infrasound logger' for analysis and converts to segy format (which is convenient for reading with traditional seismic analysis software). The Gem infrasound logger is an in-development low-cost, lightweight, low-power instrument for recording infrasound in field campaigns; email the maintainer for more information.

Details

The DESCRIPTION file:

```
Package:      gemlog
Type:        Package
Title:       File Conversion for 'Gem Infrasound Logger'
Version:     0.26
Date:       2017-10-05
Author:      Jake Anderson
Maintainer:  Jake Anderson <ajakef@gmail.com>
Description: Reads data files from the 'Gem infrasound logger' for analysis and converts to segy format (which is conv
Depends:     signal
License:     GPL
LazyLoad:   yes
Packaged:   2017-10-05 15:00:00 UTC
Repository:  CRAN
Date/Publication: 2017-10-05 15:00:00 UTC
```

Index of help topics:

```
Convert          Convert raw Gem data to segy
PlotMetadata    Read and plot Gem metadata
ReadGem         Read raw Gem data
gem_metadata    Example Metadata from Gem Logger
gemlog-package  File Conversion for 'Gem Infrasound Logger'
```

~~ An overview of how to use the package, including the most important ~~ functions ~~

Author(s)

Jake Anderson

Maintainer: Jake Anderson <ajakef@gmail.com>

References

~~ Literature or other references for background information ~~

Examples

```
## Not run:
# define bitweight for 0.5 inch sensor with Rg = 2.2k
sensitivity = 22.014e-6 # 22.014 uV/Pa
Rg = 2.2 # gain-setting resistor value in kilo-ohms
gain = 1 + 49.4/2.2 # amplifier gain
A2D = 0.256/2^15 # volts per count in analog-digital converter
bitweight = A2D / (gain * sensitivity) # conversion from counts to Pa (Pa/count)

# convert files from two Gems (SNs 000 and 001)
Convert('raw/000', bitweight = bitweight)
Convert('raw/001', bitweight = bitweight)

## End(Not run)

## Not run:
ReadGem(0:1, 'raw/000') # read files raw/000/FILE0000.TXT and raw/000/FILE0001.TXT

## End(Not run)
```

Convert

Convert raw Gem data to segy

Description

Convert takes a directory of raw Gem data files and converts them to PASSCAL segy files, including interpolating time with GPS strings and converting from counts to pressure units.

Usage

```
Convert(rawpath = ".", convertedpath = "converted", metadatapath = "metadata",
        metadatafile = NA, gpspath = "gps", gpsfile = NA, t1 = -Inf, t2 = Inf, nums = NaN,
        SN = character(), bitweight = 0.256/2^15/(4.6e-05 * 3.4/7)/23.455, time_adjustment = 0,
        yr = 2016, blockdays = 1)
```

Arguments

| | |
|---------------|--|
| rawpath | Directory containing raw data to be converted. |
| convertedpath | Directory (to be created, if necessary) where output segy files will be saved. |
| metadatapath | Directory (to be created, if necessary) where output metadata file will be saved. |
| metadatafile | Filename for output metadata. If set, overrides metadatapath. If unset, Convert creates the next logical filename in metadatapath. |

| | |
|-----------------|---|
| gpspath | Directory (to be created, if necessary) where output gps file will be saved. |
| gpsfile | Filename for output gps data. If set, overrides gpspath. If unset, Convert creates the next logical filename in gpspath. |
| t1 | Time at which conversion should start (class POSIXct). |
| t2 | Time at which conversion should end (class POSIXct). |
| nums | File numbers to convert. |
| SN | Serial number of Gem data files to convert (to be safe, when data from multiple Gems could be mixed). |
| bitweight | Conversion factor between counts and Pascals (Pa per count). |
| time_adjustment | Offset to add to output times (usually unnecessary, sometimes +/- 1 s). |
| yr | Year of data (unnecessary). |
| blockdays | Amount of data (measured in days) to process at a time. Normally 1, but for computers with little memory, might be necessary to make less than 1. |

Details

This is the usual function to use when converting data to segy files. To read data directly into R, use ReadGem.

Value

None; writes files only.

Note

A good directory structure might be something like

projectname

—raw

——010: Directory containing data from Gem SN 010 (e.g.)

——011: Directory containing data from Gem SN 011 (e.g.)

—converted

——seggy files

—gps

——010gps_000.txt: GPS file for Gem 010

——011gps_000.txt: GPS file for Gem 011

—metadata

——010metadata_000.txt: Metadata file for Gem 010

——011metadata_000.txt: Metadata file for Gem 011

—projectname_notes.txt

Author(s)

Jake Anderson

See Also

ReadGem

Examples

```
## Not run:
# define bitweight for 0.5 inch sensor with Rg = 2.2k
sensitivity = 22.014e-6 # 22.014 uV/Pa
Rg = 2.2 # gain-setting resistor value in kilo-ohms
gain = 1 + 49.4/2.2 # amplifier gain
A2D = 0.256/2^15 # volts per count in analog-digital converter
bitweight = A2D / (gain * sensitivity) # conversion from counts to Pa (Pa/count)

# convert files from two Gems (SNs 000 and 001)
Convert('raw/000', bitweight = bitweight)
Convert('raw/001', bitweight = bitweight)

## End(Not run)
```

gem_metadata

Example Metadata from Gem Logger

Description

Metadata (temperature, battery, etc.) from a Gem data logger. Provided as an example for running PlotMetadata.

Usage

```
gem_metadata
```

Format

List containing vectors (time series) of different metadata types.

Source

Recording made by author.

See Also

PlotMetadata

| | |
|--------------|-----------------------------------|
| PlotMetadata | <i>Read and plot Gem metadata</i> |
|--------------|-----------------------------------|

Description

ScanMetadata reads a Gem metadata file produced by Convert. PlotMetadata plots battery, temperature, GPS metadata.

Usage

```
ScanMetadata(fn, plot = TRUE)
PlotMetadata(M, xlim = range(M$t, na.rm = TRUE))
```

Arguments

| | |
|------|---|
| fn | Filename to read |
| plot | If true, plots metadata after reading the file |
| M | Metadata, such as output of ScanMetadata |
| xlim | Time limits to plot, in fractional days of year |

Value

ScanMetadata: list including metadata from file:

| | |
|-----------------|---|
| millis | millis count of metadata sample |
| batt | battery voltage |
| temp | temperature in (deg C) |
| maxWriteTime | maximum time required to write a sample |
| minFifoFree | minimum number of free samples in FIFO buffer |
| maxFifoUsed | maximum number of used samples in FIFO buffer |
| maxOverruns | maximum number of sample overruns |
| gpsOnFlag | 1 if gps is turned on, 0 otherwise |
| unusedStack1 | free memory in stack 1 |
| unusedStackIdle | free memory in idle stack |

PlotMetadata: None.

Author(s)

Jake Anderson

Examples

```
## Not run:
M = ScanMetadata('metadata/001metadata_000.txt') # scan the first metadata file from Gem SN 001

## End(Not run)

data(gem_metadata)
PlotMetadata(gem_metadata)
```

ReadGem

*Read raw Gem data***Description**

Reads raw Gem data into R. To write segy files, use Convert.

Usage

```
ReadGem(nums = 0:9999, path = "./", alloutput = FALSE,
        verbose = TRUE, requireGPS = FALSE, SN = character())
```

Arguments

| | |
|------------|--|
| nums | File numbers to convert. |
| path | Directory in which raw data files are contained. |
| alloutput | Include raw data in the output, in addition to the processed data. |
| verbose | Provide verbose output. |
| requireGPS | Require GPS strings to perform the conversion. |
| SN | If set, only read files of this serial number. |

Value

| | |
|---|------------------------|
| t | sample times (POSIXct) |
| p | samples (counts) |

- gps\$yyear of gps samples
- gps\$dategps sample time, as fractional day of year
- gps\$latlatitude
- gps\$lonlongitude
- metadata\$millis count of metadata sample
- metadata\$battbattery voltage
- metadata\$temptemperature in (deg C)

- metadata\$maxWriteTime maximum time required to write a sample
- metadata\$minFifoFree minimum number of free samples in FIFO buffer
- metadata\$maxFifoUsed maximum number of used samples in FIFO buffer
- metadata\$maxOverruns maximum number of sample overruns
- metadata\$gpsOnFlag 1 if gps is turned on, 0 otherwise
- metadata\$unusedStack1 free memory in stack 1
- metadata\$unusedStackIdle free memory in idle stack

- header\$file vector of raw file names
- header\$SN vector of Gem serial numbers
- header\$lat mean latitude
- header\$lon mean longitude
- header\$t1 start time
- header\$t2 end time
- header\$alloutput if alloutput == TRUE, list including raw data

Author(s)

Jake Anderson

See Also

Convert

Examples

```
## Not run:  
ReadGem(0:1, 'raw/000') # read files raw/000/FILE0000.TXT and raw/000/FILE0001.TXT  
  
## End(Not run)
```


Index

*Topic **IO**

Convert, [3](#)

PlotMetadata, [6](#)

ReadGem, [7](#)

*Topic **datasets**

gem_metadata, [5](#)

*Topic **package**

gemlog-package, [2](#)

*Topic **plot**

PlotMetadata, [6](#)

Convert, [3](#)

gem_metadata, [5](#)

gemlog (gemlog-package), [2](#)

gemlog-package, [2](#)

PlotMetadata, [6](#)

ReadGem, [7](#)

ScanMetadata (PlotMetadata), [6](#)