gemlog

Release 1.7.2

Jake Anderson

Jul 12, 2023

CONTENTS

1 Contents **3**

1.1 API Reference 3

1.2 What's New 9

2 Indices and tables **15**

Index **17**

gemlog is a python package that provides functions for processing the log files from the Gem infrasound datalogger. The Gem infrasound logger is a low-cost, lightweight, low-power instrument for recording infrasound in field campaigns.

The source code for gemlog is maintained on github.

CHAPTER

ONE

CONTENTS

1.1 API Reference

1.1.1 gemlog

Functions for reading and processing data files from the Gem Infrasound Logger.

read_gem([path, nums, SN, units, bitweight, ...])	Read raw Gem files.
convert([rawpath, convertedpath, ...])	Read raw Gem files, interpolate them, and write output files in miniSEED or SAC format.
gem_cat(input_dir, output_dir[, ext, cat_all])	Merge raw data files so that all contain GPS data.
make_db(path[, pattern, savefile, verbose])	Create a database summarizing a set of converted data files.
calc_channel_stats(DB, t1, t2)	Calculate uptime and other statistics for all channels in a database.
get_gem_specs(SN)	Retrieve specs for a given Gem serial number.
read_gps(gps_dir_pattern, SN)	Read the most up-to-date GPS file for a given serial number.
summarize_gps(gps_dir_pattern[, ...])	Read up-to-date GPS data from all Gems in a project, estimate their locations using a robust trimmed-mean method.
make_gem_inventory(station_info, coords[, ...])	Create a station inventory for a Gem dataset.
rename_files(infile_pattern, station_info, ...)	Rename a set of converted data files, assigning the correct network, station, and location codes (instead of the original code with empty location/network and the station code being the serial number).

gemlog.read_gem

gemlog.**read_gem**(*path='raw'*, *nums=array([0, 1, 2, ..., 9997, 9998, 9999])*, *SN=''*, *units='Pa'*, *bitweight=nan*, *bitweight_V=nan*, *bitweight_Pa=nan*, *verbose=True*, *network=''*, *station=''*, *location=''*, *return_debug_output=False*, *require_gps=True*, *gps_strict_level=1*)

Read raw Gem files.

Parameters

- **path** (*str, default '.'*) – Path of folder containing raw Gem data files to read.
- **nums** (*list or np.array of integers*) – Numbers of raw Gem files to read. By default, it reads all files in 'path' for the specified serial number.

- **SN** (*str*) – One Gem serial number to read. Use a loop to read multiple Gems.
- **units** (*str, default 'Pa'*) – Desired output units. Options are 'Pa' (Pascals), 'V' (Volts), or 'counts'.
- **bitweight** (*float*) – The value of each count when converting between counts and other units (typically Pascals, possibly Volts). By default, it looks up the correct bitweight given the Gem's serial number and gain configuration. Leave this blank unless the Gem has been modified in a way that changes the bitweight.
- **bitweight_V** (*float*) – The value of each count when converting between counts and Volts. By default, it looks up the correct bitweight given the Gem's serial number and gain configuration. Leave blank unless the Gem has been modified in a way that changes the voltage bitweight.
- **bitweight_Pa** (*float*) – The value of each count when converting between counts and Pascals. By default, it looks up the correct bitweight given the Gem's serial number and gain configuration. Leave this blank unless the Gem has been modified in a way that changes the pressure bitweight.
- **verbose** (*boolean, default True*) – Whether to print verbose progress messages to the screen.
- **network** (*str*) – Two-character name of the sensor network. Leaving this blank is usually fine in subsequent data processing.
- **station** (*str*) – Name of the station (up to five characters) to assign to the data. If not provided, uses the Gem's serial number as the station ID.
- **location** (*str*) – Two-character location code for this Gem. Leaving this blank is usually fine in subsequent data processing.
- **return_debug_output** (*boolean, default False*) – If True, return extra output to understand the internal state of the function.
- **require_gps** (*boolean, default True*) – If True, read files whether or not they have GPS data. Sample times will not be precise enough for array processing.

Returns

dict with keys –

- data : obspy.Stream, infrasound data
- header : pandas.dataframe, information on raw Gem data files
- metadata : pandas.dataframe, state-of-health and other metadata time series
- gps : pandas.dataframe, GPS timing and location values

Note: All sample times involving the Gem (and most other passive seismic/acoustic data) are in UTC; time zones are not supported.

---

gemlog.convert

gemlog.**convert**(*rawpath='.'*, *convertedpath='converted'*, *metadatapath='metadata'*, *metadatafile=''*, *gpspath='gps'*, *gpsfile=''*, *t1=-inf*, *t2=inf*, *nums=nan*, *SN=''*, *bitweight=nan*, *units='Pa'*, *time_adjustment=0*, *blockdays=1*, *file_length_hour=24*, *station=''*, *network=''*, *location=''*, *output_format='MSEED'*)

Read raw Gem files, interpolate them, and write output files in miniSEED or SAC format.

Parameters

- **rawpath** (*str, default '.'*) – Path of folder containing raw Gem data files to read.
- **convertedpath** (*str, 'converted'*) – Path of folder where converted data files should be written. If this folder does not exist, convert will try to create it.
- **metadatapath** (*str, 'metadata'*) – Path of folder where metadata files should be written. If this folder does not exist, convert will try to create it.
- **metadatafile** (*str*) – File name for metadata file. Default is of the form 'XXXmetadata_NNN.txt', where 'XXX' is the Gem's serial number and 'NNN' is the file index (000 at first, incrementing by one for each subsequent calculation).
- **gpspath** (*str, 'gps'*) – Path of folder where gps files should be written. If this folder does not exist, convert will try to create it.
- **gpsfile** (*str*) – File name for gps file. Default is of the form 'XXXgps_NNN.txt', where 'XXX' is the Gem's serial number and 'NNN' is the file index (000 at first, incrementing by one for each subsequent calculation).
- **t1** (*float or obspy.UTCDateTime, default -np.inf*) – Start time for the conversion. If float, the number of seconds since the epoch (1970-01-01 00:00:00 UTC).
- **t2** (*float or obspy.UTCDateTime, default np.inf*) – Stop time for the conversion. If float, the number of seconds since the epoch (1970-01-01 00:00:00 UTC).
- **nums** (*list or np.array of integers*) – Numbers of raw Gem files to read. By default, it reads all files in rawpath for the specified serial number.
- **SN** (*str*) – One Gem serial number to read and convert. Use a loop to convert multiple Gems.
- **bitweight** (*float*) – The value of each count when converting between counts and other units (typically Pascals, possibly Volts). By default, it looks up the correct bitweight given the Gem's serial number and gain configuration. Leave this blank unless the Gem has been modified in a way that changes the bitweight.
- **units** (*str, default 'Pa'*) – Desired output units. Options are 'Pa' (Pascals), 'V' (Volts), or 'counts'.
- **time_adjustment** (*float, default 0*) – Amount to shift the sample times by in case of timing error, possibly due to leap second issues.
- **blockdays** (*float, default 1*) – Number of days worth of data to read in and convert at a time. If you have memory issues when converting data, try setting this to a smaller value.
- **file_length_hour** (*float, default 1*) – Length of the output files in hours.
- **station** (*str*) – Name of the station (up to five characters) to assign to the data. If not provided, uses the Gem's serial number as the station ID.
- **network** (*str*) – Two-character name of the sensor network. Leaving this blank is normally fine in subsequent data processing.

- **location** (*str*) – Two-character location code for this Gem. Leaving this blank is usually fine in subsequent data processing.
- **output_format** (*str, default 'MSEED'*) – Output file format. Currently, formats 'MSEED' and 'SAC' are supported; 'WAV' is partly supported.

Returns

None, writes output files only (converted, metadata, and gps)

Note: All sample times involving the Gem (and most other passive seismic/acoustic data) are in UTC; time zones are not supported.

---

gemlog.gem_cat

gemlog.**gem_cat**(*input_dir*, *output_dir*, *ext=''*, *cat_all=False*)

Merge raw data files so that all contain GPS data.

Translated from R code originally by Danny Bowman.

Parameters

- **input_dir** (*raw gem directory*) –
- **output_dir** (*path for renumbered and concatenated files*) –
- **ext** (*extension of raw files to convert (normally the serial number; sometimes TXT for old Gems)*) –

Returns

None; file output only

gemlog.make_db

gemlog.**make_db**(*path*, *pattern='*'*, *savefile=None*, *verbose=False*)

Create a database summarizing a set of converted data files.

Parameters

- **path** (*str*) – Path to folder containing converted data files to summarize.
- **pattern** (*str, default '*'*) – Glob-type pattern for selecting converted data files to summarize
- **savefile** (*str, default None*) – File name where database is written. Use 'savefile = None' to not save an output file.
- **verbose** (*bool, default False*) – If True, print progress updates.

Returns

pandas.DataFrame containing converted file database.

gemlog.calc_channel_stats

gemlog.**calc_channel_stats**(*DB*, *t1*, *t2*)

Calculate uptime and other statistics for all channels in a database.

Parameters

- **DB** (*pandas.DataFrame*) – Output of make_db().
- **t1** (*time-like*) – Start time for which statistics should be calculated.
- **t2** (*time-like*) – End time for which statistics should be calculated.

Returns

pandas.DataFrame containing the following columns –

- station : station name
- goodData : proportion of time window (t1-t2) that is not obviously bad (e.g., clipped)
- anyData : proportion of time window (t1-t2) for which data are available
- q1 : first quartile amplitude
- q3 : third quartile amplitude

gemlog.get_gem_specs

gemlog.**get_gem_specs**(*SN*)

Retrieve specs for a given Gem serial number.

Parameters

SN (*str or int*) –

Returns

dict with the following elements –

- version : Gem version number
- bitweight_V : voltage resolution for this Gem version [Volts per count]
- bitweight_Pa : pressure resolution for this Gem version [Pascals per count]

gemlog.read_gps

gemlog.**read_gps**(*gps_dir_pattern*, *SN*)

Read the most up-to-date GPS file for a given serial number.

Parameters

- **gps_dir_pattern** (*str*) – Path to the folder containing GPS data, or a glob-style pattern describing multiple folders.
- **SN** (*str*) – Serial number of the Gem being examined.

Returns

- *pandas.DataFrame containing columns year, date (day of year), lat, lon (all floats), and column*
- *t (obspy.UTCDateTime).*

gemlog.summarize_gps

gemlog.**summarize_gps**(*gps_dir_pattern*, *station_info=None*, *output_file=None*, *t1=None*, *t2=None*, *include_SN=None*, *exclude_SN=None*)

Read up-to-date GPS data from all Gems in a project, estimate their locations using a robust trimmed-mean method.

Parameters

- **gps_dir_pattern** (*str*) – Path to folder or glob-style pattern describing folders containing GPS data to review.
- **station_info** (*str*) – Path to text file containing table assigning serial numbers to network, station, and location codes.
- **output_file** (*str*) – Path to file where output should be written (optional)
- **t1** (*obspy.UTCDateTime or str*) – Start time; ignore gps data before t1 (e.g., '2022-12-31' or '2022-12-31_05:45:00')
- **t2** (*obspy.UTCDateTime or str*) – End time; ignore gps data after t2
- **include_SN** (*list*) – Serial numbers to process (default all)
- **exclude_SN** (*list*) – Serial numbers not to process (default none)

Returns

pandas.DataFrame containing the following columns –

- SN: serial number (str)
- lat: calculated average latitude (float)
- lon: calculated average longitude (float)
- lat_SE: standard error of average latitude (float)
- lon_SE: standard error of average longitude (float)
- elevation: elevation provided by user (-9999 if not provided) (float)
- starttime: time of first GPS data (obspy.UTCDateTime)
- endtime: time of last GPS data (obspy.UTCDateTime)
- num_samples: number of GPS strings recorded

If station_info is provided, then the following columns are also included:

- network: network code (str)
- station: station code (str)
- location: location code (str)
- elevation: elevation provided in station_info (str)

gemlog.make_gem_inventory

gemlog.**make_gem_inventory**(*station_info*, *coords*, *response='default'*)

Create a station inventory for a Gem dataset.

Parameters

- **station_info** (*str or pandas.DataFrame*) – file that contains a table with station definitions (columns for serial number, network, station, location, and channel)
- **coords** (*pandas.DataFrame*) – output of summarize_gps(), with 'elevation' column added
- **response** (*str*) – instrument response information (currently only 'default' is supported)

Returns

obspy.Inventory containing station metadata for the dataset

gemlog.rename_files

gemlog.**rename_files**(*infile_pattern*, *station_info*, *output_dir*, *output_format='mseed'*, *outfile_pattern='{year}-{mon}-{day}T{hour}_{min}_{sec}.{net}.{sta}.{loc}.{chan}.{fmt}'*)

Rename a set of converted data files, assigning the correct network, station, and location codes (instead of the original code with empty location/network and the station code being the serial number).

Parameters

- **infile_pattern** (*str*) – glob-type pattern defining the input files
- **station_info** (*str or pandas.DataFrame*) – file name for table assigning serial numbers to network, station, and location codes
- **output_dir** (*str*) – folder where output files should be written
- **output_format** (*str*) – default 'mseed'; 'sac' also works, as do other obspy-supported formats
- **outfile_pattern** (*str*) – format of output file names; note the abbreviations, and that 'jd' means 'day of year'

Returns

pandas.DataFrame containing the station_info table.

1.2 What's New

These are the changes in each version of gemlog.

1.2.1 v1.6.6 (November 10, 2022)

Breaking changes

Upgrading dependency version requirements due to security issue in older numpy

Deprecations

Enhancements

New tools to convert files with missing or inadequate GPS data (at user's risk!) Support for compact new raw format 1.10

Bug fixes

Testing

Documentation

- This sphinx documentation (GH#12)

Requirements

Contributors

1.2.2 v1.5.6 (October 5, 2021)

Breaking changes

Deprecations

Enhancements

Improve logs Add better user interface for working with Gem response and accessing noise specs Improve checks to identify unusable raw files

Bug fixes

Testing

Improvements to automated tests

Documentation

- This sphinx documentation (GH#12)

Requirements

Contributors

1.2.3 v1.4.3 (August 5, 2021)

Breaking changes

Deprecations

Enhancements

Added new fields to inventories to meet IRIS requirements (elevation, start/end times, azimuth, dip) Improvements to huddle_test Improved timing corrections (cubic interpolation) Remove overlapping samples in output miniSEED files for user convenience

Bug fixes

Testing

Documentation

- This sphinx documentation (GH#12)

Requirements

Contributors

1.2.4 v1.3.4 (December 28, 2020)

Breaking changes

Deprecations

Enhancements

Support for raw format 0.91 Enables writing day-long miniSEED files (instead of just hour-long) Switched to a cross-platform (windows-friendly) miniSEED file naming format that avoids colons

Bug fixes

Testing

Documentation

- This sphinx documentation (GH#12)

Requirements

Contributors

1.2.5 v1.3.1 (December 1, 2020)

Breaking changes

Deprecations

Enhancements

Support for old raw formats 0.8 and 0.85 Improvements to gemconvert Now hosted on PyPI

Bug fixes

Testing

Documentation

- This sphinx documentation (GH#12)

Requirements

Contributors

1.2.6 v1.2.2 (October 26, 2020)

Breaking changes

Deprecations

Enhancements

Bug fixes

No longer crashes when processing a mix of normal raw files and raw files that are empty or lack GPS data

Testing

Documentation

- This sphinx documentation (GH#12)

Requirements

Contributors

1.2.7 v1.1.2 (October 14, 2020)

Breaking changes

Deprecations

Enhancements

Added an option to gemconvert to allow output files of length other than 1 hour. Day-long miniSEED files are often more convenient than hour-long files.

Bug fixes

Fixed some issues with the Missing GPS Demo.

Testing

Documentation

- This sphinx documentation (GH#12)

Requirements

Contributors

1.2.8 v1.1.1 (October 10, 2020)

Breaking changes

Deprecations

Enhancements

Changed sample time interpolation from block-of-files level to single-file level, which prevents possible problems that can cause imprecise sample times.

Bug fixes

Testing

Documentation

- This sphinx documentation (GH#12)

Requirements

Contributors

1.2.9 v1.1.0 (October 7, 2020)

- Added new gem_cat feature for combining no-GPS files.

1.2.10 v1.0.2 (October 2, 2020)

- changed make_gem_inventory so that output stationXML passes the stationXML validator: https://github.com/iris-edu/StationXML-Validator/

1.2.11 v1.0.1 (September 29, 2020)

- changed 'master' branch to 'main'

1.2.12 v1.0.0 (September 17, 2020)

- function name changes (some with aliases) to follow python standards

1.2.13 v0.3.2 (September 9, 2020)

- added demo with inventory functions

1.2.14 v0.3.1 (September 4, 2020)

- more helpful logging output

1.2.15 v0.3.0 (September 4, 2020)

- 2020-09-04, cython added

1.2.16 v0.2.3 (September 1, 2020)

- improving testing, including empty/bad files

1.2.17 v0.2.2 (August 18, 2020)

- Automated github tests, setup.py improvements, and modest speed-up

1.2.18 v0.2.1

1.2.19 v0.0.5

- Added new functions to make network map from GPS data and rename mseeds from serial_number.channel to network.station.location.channel codes

CHAPTER

TWO

INDICES AND TABLES

- genindex
- search

INDEX

C

calc_channel_stats() (*in module gemlog*), 7
convert() (*in module gemlog*), 5

G

gem_cat() (*in module gemlog*), 6
get_gem_specs() (*in module gemlog*), 7

M

make_db() (*in module gemlog*), 6
make_gem_inventory() (*in module gemlog*), 9

R

read_gem() (*in module gemlog*), 3
read_gps() (*in module gemlog*), 7
rename_files() (*in module gemlog*), 9

S

summarize_gps() (*in module gemlog*), 8