
tremana Documentation

Release 0.0.1

Sebastian Weigand

Apr 23, 2022

USER DOCUMENTATION:

1	tremana	1
1.1	Features	1
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	Inner workings	7
4.1	tremana	7
5	Contributing	17
5.1	Types of Contributions	17
5.2	Get Started!	18
5.3	Pull Request Guidelines	19
5.4	Tips	19
5.5	Deploying	19
6	History	21
6.1	0.0.1 (2021-01-24)	21
7	Indices and tables	23
	Python Module Index	25
	Index	27

TREMANA

Medical tremor analysis package (e.g. for parkinsonian tremor)

- Free software: Apache Software License 2.0
- Documentation: <https://tremana.readthedocs.io>.

1.1 Features

- Library with functions for tremor analysis (WIP)
 - Accelerometry (WIP)
 - * Device raw data reader (WIP)
 - * FFT analysis (WIP)
- CLI for batch analysis (planned)
- Jupyter widgets for interactive analysis (planned)

INSTALLATION

2.1 Stable release

To install tremana, run this command in your terminal:

```
$ pip install tremana
```

This is the preferred method to install tremana, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for tremana can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/s-weigand/tremana
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/s-weigand/tremana/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER THREE

USAGE

To use tremana in a project:

```
import tremana
```


INNER WORKINGS

This is the detailed documentation of the inner workings of `tremana`.

<code>tremana</code>	Top-level package for tremana.
----------------------	--------------------------------

4.1 tremana

Top-level package for tremana.

Modules

<code>tremana.analysis</code>	Module containing functions to transform data and metrics.
<code>tremana.cli</code>	Console script for tremana.
<code>tremana.exceptions</code>	Custom exceptions for tremana.
<code>tremana.parsers</code>	Package containing parser for intermediate files and device raw data.
<code>tremana.tremana</code>	Main module.
<code>tremana.utils</code>	Utility functions which are used internally.
<code>tremana.warnings</code>	Custom Warnings for tremana.

4.1.1 analysis

Module containing functions to transform data and metrics.

Modules

<code>tremana.analysis.metrics</code>	Module containing metrics to be calculated on tremor accelerometry data or their FFT.
<code>tremana.analysis.transformations</code>	Transformations to be used on tremor accelerometry data (e.g.: FFT).

metrics

Module containing metrics to be calculated on tremor accelerometry data or their FFT.

Functions

Summary

<code>center_of_mass</code>	Calculate the center of mass of FFT spectra.
-----------------------------	--

center_of_mass

center_of_mass(*fft_spectra*: *pandas.core.frame.DataFrame*) → *pandas.core.frame.DataFrame*
 Calculate the center of mass of FFT spectra.

$$H_{cm} = \frac{1}{N-1} \frac{\sum_{i=1}^N ((i-1)X_i)}{\sum_{i=1}^N X_i}$$

Parameters **fft_spectra** (*pd.DataFrame*) – Dataframe with each column being a FFT spectrum.

Returns Dataframe with the center of mass in the with columns names same as the spectra.

Return type *pd.DataFrame*

transformations

Transformations to be used on tremor accelerometry data (e.g.: FFT).

Functions

Summary

<code>fft_spectra</code>	Calculate the FFT of accelerometry data.
<code>power_density_spectra</code>	Calculate the power density spectra of accelerometry data.

fft_spectra

fft_spectra(*input_dataframe: pd.DataFrame, columns: Iterable[str] | None = None, sampling_rate: int | float = 128, norm: bool = False*) → *pd.DataFrame*

Calculate the FFT of accelerometry data.

Parameters

- **input_dataframe** (*pd.DataFrame*) – Dataframe containing accelerometry data.
- **columns** (*Iterable[str], optional*) – Columns to calculate the FFT for, by default None which results in all columns to be used
- **sampling_rate** (*int*) – Number of sample per second, by default 128
- **norm** (*bool*) – Whether to normalize the data to 1 or not, by default False

Returns FFT spectra of the accelerometry data.

Return type *pd.DataFrame*

power_density_spectra

power_density_spectra(*input_dataframe: pd.DataFrame, columns: Iterable[str] | None = None, sampling_rate: int | float = 128, norm: bool = False*) → *pd.DataFrame*

Calculate the power density spectra of accelerometry data.

Compared to the FFT the resulting values are $\text{FFT}[-\text{freq}] * \text{FFT}[\text{freq}]$ with $\text{freq} \geq 0$.

Parameters

- **input_dataframe** (*pd.DataFrame*) – Dataframe containing accelerometry data.
- **columns** (*Iterable[str]*) – Columns to calculate the FFT for, by default None which results in all columns to be used
- **sampling_rate** (*int | float*) – Number of sample per second, by default 128
- **norm** (*bool*) – Whether to normalize the data to 1 or not, by default False

Returns Power density spectra accelerometry data.

Return type *pd.DataFrame*

4.1.2 cli

Console script for tremana.

4.1.3 exceptions

Custom exceptions for tremana.

Exceptions

Exception Summary

<i>TremanaException</i>	BaseException for expected possible errors with tremana.
<i>TremanaParsingException</i>	Baseclass for parsing Exceptions.
<i>TremanaParsingSampleRateException</i>	Error thrown when the samplerate can't be parsed to a numerical value.

TremanaException

exception `TremanaException(*args: object, msg: str)`

BaseException for expected possible errors with tremana.

Parameters `msg (str)` – Message to be printed.

TremanaParsingException

exception `TremanaParsingException(*args: object, msg: str, origin_file: str | os.PathLike[str] | None = None)`

Baseclass for parsing Exceptions.

Parameters

- `msg (str)` – Message to be printed.
- `origin_file (Union[str, os.PathLike])` – Path to the file causing the warning, by default None

TremanaParsingSampleRateException

exception `TremanaParsingSampleRateException(*args: object, sample_rate: str, origin_file: str | os.PathLike[str] | None = None)`

Error thrown when the samplerate can't be parsed to a numerical value.

Parameters

- `sample_rate (str)` – Value of the sample rate which should have been cast to float.
- `origin_file (Union[str, os.PathLike])` – Path to the file causing the warning, by default None

4.1.4 parsers

Package containing parser for intermediate files and device raw data.

Modules

<code>tremana.parsers.devices</code>	Package containing modules for raw data reading for specific devices.
--------------------------------------	---

devices

Package containing modules for raw data reading for specific devices.

Modules

<code>tremana.parsers.devices.somnowatch</code>	Module containing the somnowatch parser.
---	--

somnowatch

Module containing the somnowatch parser.

See: <https://somnomedics.de>

Classes

Summary

<code>SomnoWatchMetaData</code>	NamedTuple representing the somnowatch meta information.
---------------------------------	--

SomnoWatchMetaData

class **SomnoWatchMetaData**(*signal_type: str, start_date: str, sample_rate: float, length: int, unit: str*)

NamedTuple representing the somnowatch meta information.

Create new instance of SomnoWatchMetaData(signal_type, start_date, sample_rate, length, unit)

Attributes Summary

<code>length</code>	Alias for field number 3
<code>sample_rate</code>	Alias for field number 2
<code>signal_type</code>	Alias for field number 0
<code>start_date</code>	Alias for field number 1
<code>unit</code>	Alias for field number 4

Methods Summary

<code><i>count</i></code>	Return number of occurrences of value.
<code><i>index</i></code>	Return first index of value.

count

`SomnoWatchMetaData.count(value, /)`
Return number of occurrences of value.

index

`SomnoWatchMetaData.index(value, start=0, stop=9223372036854775807, /)`
Return first index of value.

Raises `ValueError` if the value is not present.

Methods Documentation

count(*value*, /)
Return number of occurrences of value.

index(*value*, *start*=0, *stop*=9223372036854775807, /)
Return first index of value.

Raises `ValueError` if the value is not present.

4.1.5 tremana

Main module.

4.1.6 utils

Utility functions which are used internally.

Modules

<code>tremana.utils.dataframe_helper</code>	Helper functions to extract information from DataFrames.
<code>tremana.utils.io</code>	Helper function for IO operations.

dataframe_helper

Helper functions to extract information from DataFrames.

Functions

Summary

<code>extract_position_by_value</code>	Extract index and column where the dataframe has the value <code>value</code> .
--	---

extract_position_by_value

extract_position_by_value(*df*: *pd.DataFrame*, *value*: *object*) → list[tuple[object, object]]

Extract index and column where the dataframe has the value `value`.

Parameters

- **df** (*pd.DataFrame*) – Dataframe the columns and indices should be extracted from.
- **value** (*object*) – Value to look for in `df`

Returns List of indices and column where `df` has the value `value`

Return type list[tuple[object, object]]

io

Helper function for IO operations.

Functions

Summary

<i>lazy_read_headers</i>	Lazy read headerlines of files.
--------------------------	---------------------------------

lazy_read_headers

lazy_read_headers(*file_paths*: *Iterable*[*str* | *os.PathLike*[*str*]], *, *lines_to_read*: *int* = 10) → list[list[*str*]]
 Lazy read headerlines of files.

Parameters

- **file_paths** (*Iterable*[*str* | *os.PathLike*[*str*]]) – Paths to the files which should be read.
- **lines_to_read** (*int*) – Number of lines to be read, by default 10

Returns Headerlines of the read files.

Return type list[list[*str*]]

4.1.7 warnings

Custom Warnings for tremana.

Functions

Summary

<i>filter_tremana_warnings</i>	Contextmanager to filter all warnings defined in <i>warning_types</i> .
--------------------------------	---

filter_tremana_warnings

filter_tremana_warnings(*warning_types*: *Iterable*[*type*[*TremanaBaseWarning*]], *message*: *str* = "") → *Generator*[*None*, *None*, *None*]
 Contextmanager to filter all warnings defined in *warning_types*.

Parameters

- **warning_types** (*Iterable*[*Type*[*TremanaBaseWarning*]]) – Warnings to be filtered
- **message** (*str*) – Regular expression to filter warnings by.

Exceptions

Exception Summary

<code>TremanaBaseWarning</code>	Basewarning for expected possible errors with tremana.
<code>TremanaNotSupportedWarning</code>	Basewarning for not yet supported features.
<code>TremanaParsingIgnoredSignalTypeWarning</code>	Warning for not supported signal types.
<code>TremanaParsingInconsistentMetadataWarning</code>	Warning for inconsistent metadata in a measurement.
<code>TremanaParsingIncorrectDateFormatWarning</code>	Warning when date format_str doesn't match date_str.
<code>TremanaSupressableWarning</code>	Basewarning for supressable warnings.

TremanaBaseWarning

exception `TremanaBaseWarning`(*args: object, msg: str, origin_file: str | os.PathLike[str] | None = None, append_msg: str | None = None)

Basewarning for expected possible errors with tremana.

Parameters

- **msg** (str) – Message to be printed.
- **origin_file** (Union[str, os.PathLike], optional) – Path to the file causing the warning, by default None
- **append_msg** (Optional[str], optional) – Message to be appended on msg, by default None

TremanaNotSupportedWarning

exception `TremanaNotSupportedWarning`(*args: object, msg: str, **kwargs: Any)

Basewarning for not yet supported features.

Parameters **msg** (str) – Message to be printed.

TremanaParsingIgnoredSignalTypeWarning

exception `TremanaParsingIgnoredSignalTypeWarning`(*args: object, signal_type: str, **kwargs: Any)

Warning for not supported signal types.

Parameters **signal_type** (str) – Value of the signal type.

TremanaParsingInconsistentMetadataWarning

```
exception TremanaParsingInconsistentMetadataWarning(*args: object, actual_value: Any,
                                                    expected_value: Any,
                                                    metadata_name: str, origin_file: str
                                                    | os.PathLike[str], **kwargs: Any)
```

Warning for inconsistent metadata in a measurement.

Parameters

- **actual_value** (*Any*) – Value of the metadata
- **expected_value** (*Any*) – Expected value of the metadata
- **metadata_name** (*str*) – Name of the metadata
- **origin_file** (*Union[str, os.PathLike]*) – Path to the file causing the warning

TremanaParsingIncorrectDateFormatWarning

```
exception TremanaParsingIncorrectDateFormatWarning(*args: object, date_str: str,
                                                    format_str: str, **kwargs: Any)
```

Warning when date format_str doesn't match date_str.

Parameters

- **date_str** (*str*) – Value of the string which should be parsed
- **format_str** (*str*) – Format string used to parse date_str.

TremanaSupressableWarning

```
exception TremanaSupressableWarning(*args: object, msg: str, append_msg: str | None = None,
                                     **kwargs: Any)
```

Basewarning for supressable warnings.

This baseclass can be used to conveniently suppress all none critical warnings.

See also:

[*filter_tremana_warnings*](#)

Parameters

- **msg** (*str*) – Message to be printed.
- **append_msg** (*Optional[str], optional*) – Message to be appended on msg, by default None

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/s-weigand/tremana/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

5.1.4 Write Documentation

tremana could always use more documentation, whether as part of the official tremana docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/s-weigand/tremana/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up `tremana` for local development.

1. Fork the `tremana` repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/tremana.git
```

3. Install your local copy into a virtualenv. Assuming you have `virtualenvwrapper` installed, this is how you set up your fork for local development:

```
$ mkvirtualenv tremana
$ cd tremana/
$ pip install -e .
```

4. install the `pre-commit` and `pre-push` hooks:

```
$ pre-commit install && pre-commit install -t pre-push
```

5. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

6. When you're done making changes, check that your changes pass `flake8` and the tests, including testing other Python versions with `tox`:

```
$ tox
```

To get `flake8` and `tox`, just `pip` install them into your virtualenv.

7. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

8. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.md.
3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check <https://github.com/s-weigand/tremana/actions> and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ pytest tests.test_tremana
```

5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch  
$ git push --follow-tags
```

Travis will then deploy to PyPI if tests pass.

HISTORY

6.1 0.0.1 (2021-01-24)

- First release on PyPI.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

t

- [tremana](#), 7
- [tremana.analysis](#), 7
- [tremana.analysis.metrics](#), 8
- [tremana.analysis.transformations](#), 8
- [tremana.cli](#), 9
- [tremana.exceptions](#), 9
- [tremana.parsers](#), 11
- [tremana.parsers.devices](#), 11
- [tremana.parsers.devices.somnowatch](#), 11
- [tremana.tremana](#), 12
- [tremana.utils](#), 13
- [tremana.utils.dataframe_helper](#), 13
- [tremana.utils.io](#), 14
- [tremana.warnings](#), 14

INDEX

C

`center_of_mass()` (in module *trem-ana.analysis.metrics*), 8
`count()` (*SomnoWatchMetaData* method), 12

E

`extract_position_by_value()` (in module *trem-ana.utils.dataframe_helper*), 13

F

`fft_spectra()` (in module *trem-ana.analysis.transformations*), 9
`filter_tremana_warnings()` (in module *trem-ana.warnings*), 14

I

`index()` (*SomnoWatchMetaData* method), 12

L

`lazy_read_headers()` (in module *tremana.utils.io*), 14

M

module
 tremana, 7
 tremana.analysis, 7
 tremana.analysis.metrics, 8
 tremana.analysis.transformations, 8
 tremana.cli, 9
 tremana.exceptions, 9
 tremana.parsers, 11
 tremana.parsers.devices, 11
 tremana.parsers.devices.somnowatch, 11
 tremana.tremana, 12
 tremana.utils, 13
 tremana.utils.dataframe_helper, 13
 tremana.utils.io, 14
 tremana.warnings, 14

P

`power_density_spectra()` (in module *trem-ana.analysis.transformations*), 9

S

SomnoWatchMetaData (class in *trem-ana.parsers.devices.somnowatch*), 11

T

tremana
 module, 7
tremana.analysis
 module, 7
tremana.analysis.metrics
 module, 8
tremana.analysis.transformations
 module, 8
tremana.cli
 module, 9
tremana.exceptions
 module, 9
tremana.parsers
 module, 11
tremana.parsers.devices
 module, 11
tremana.parsers.devices.somnowatch
 module, 11
tremana.tremana
 module, 12
tremana.utils
 module, 13
tremana.utils.dataframe_helper
 module, 13
tremana.utils.io
 module, 14
tremana.warnings
 module, 14
TremanaBaseWarning, 15
TremanaException, 10
TremanaNotSupportedWarning, 15
TremanaParsingException, 10
TremanaParsingIgnoredSignalTypeWarning, 15
TremanaParsingInconsistentMetadataWarning, 16
TremanaParsingIncorrectDateFormatWarning, 16
TremanaParsingSampleRateException, 10
TremanaSupressableWarning, 16