
DataSounds Documentation

Release 1.2.0

DataSounds

Sep 27, 2017

Contents

1	Readme	3
1.1	DataSounds	3
2	Installation	5
2.1	Dependencies	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
4.4	Tips	11
5	Credits	13
5.1	DataSounds development Lead	13
5.2	Contributors	13
6	History	15
6.1	1.2.0 (2013-12-22)	15
6.2	1.1.0 (2013-06-08)	15
6.3	0.1.0 (2013-03-15)	15
7	Indices and tables	17

Contents:

CHAPTER 1

Readme

DataSounds

Get sounds from temporal series, or another sequecial data. Visit us at www.datasounds.org.

At the command line:

```
$ git clone http://github.com/DataSounds/DataSounds.git
$ cd DataSounds
$ python setup.py install
```

Or using [pip](#) (program to easily install Python packages), which dynamically access the *Python Package Index* [PyPI](#).

```
$ pip install DataSounds
```

Or, with the controlled python ecosystem [virtualenvwrapper](#). After install [virtualenvwrapper](#), follow the instructions below:

```
$ mkvirtualenv DataSounds
$ workon DataSounds
(DataSounds) $ git clone http://github.com/DataSounds/DataSounds.git
(DataSounds) $ cd DataSounds
(DataSounds) $ python setup.py install
```

Dependencies

[Numpy](#) is a necessary packages to use DataSounds.

[Numpy](#) can be installed using [pip](#). If you use [virtualenvwrapper](#), this could be done inside your virtual environment. Normally, [Numpy](#) is installed as a dependency of DataSounds and should work if it was successfully compiled.

CHAPTER 3

Usage

```
from DataSounds.sounds import get_music, w2Midi
import numpy as np

data = np.random.rand(24)
music = get_music(data)
w2Midi('my_musica_data', music)
```

w2Midi writes a .midi file inside the current directory. In this case *my_music_data.midi* will be saved on disk. Enjoy it!

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

Types of Contributions

Report Bugs

Report bugs at <https://github.com/DataSounds/DataSounds/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.

Fix Bugs

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

Implement Features

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

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/DataSounds/DataSounds/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 :)

Get Started!

Ready to contribute? Here's how to set up *DataSounds* for local development.

1. Fork the *DataSounds* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/DataSounds.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 devDS
$ cd DataSounds/
$ python setup.py develop
```

4. Create a branch for local development:

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

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass through tests, including testing other Python versions with tox:

```
$ python setup.py test
$ tox
```

To get the necessary packages just install it as follows inside yours virtualenv:

```
$ pip install -r dev-requirements.txt
```

6. 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
```

7. Submit a pull request through the GitHub website.

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.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/DataSounds/DataSounds/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_sounds.py
```


DataSounds development Lead

- Arnaldo Russo <arnaldo@datasounds.org>
- Luiz Irber <luiz@datasounds.org>

Contributors

None yet. Why not be the first?

1.2.0 (2013-12-22)

1.1.0 (2013-06-08)

0.1.0 (2013-03-15)

- First release on PyPI.

CHAPTER 7

Indices and tables

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