
molecules Documentation

Release 0.0.3.2

Todd Young

Jan 18, 2019

Contents

1	Installation	1
1.1	Stable release	1
1.2	From sources	1
2	Usage	3
3	Contributing	5
3.1	Types of Contributions	5
3.2	Get Started!	6
3.3	Pull Request Guidelines	6
3.4	Tips	7
4	Credits	9
4.1	Development Lead	9
4.2	Contributors	9
5	History	11
5.1	0.0.1 (2018-10-3)	11
6	molecules	13
6.1	molecules package	13
7	Feedback	17
	Python Module Index	19

1.1 Stable release

To install molecules, run this command in your terminal:

```
pip install molecules
```

This is the preferred method to install molecules, 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.

1.2 From sources

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

You can either clone the public repository:

```
git clone git://github.com/yngtodd/molecules
```

Or download the [tarball](#):

```
curl -OL https://github.com/yngtodd/molecules/tarball/master
```

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

```
python setup.py install
```


CHAPTER 2

Usage

To use molecules in a project:

```
import molecules
```


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

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at <https://github.com/yngtodd/molecules/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.

3.1.2 Fix Bugs

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

3.1.3 Implement Features

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

3.1.4 Write Documentation

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

3.1.5 Submit Feedback

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

3.2 Get Started!

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

1. [Fork](#) the *molecules* repo on GitHub.
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/molecules.git
```

3. Create a branch for local development:

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

Now you can make your changes locally.

4. When you're done making changes, check that your changes pass style and unit tests, including testing other Python versions with tox:

```
tox
```

To get tox, just pip install it.

5. 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
```

6. Submit a pull request through the GitHub website.

3.3 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.rst.

3. The pull request should work for Python ≥ 3.6 and for PyPy. Check <https://travis-ci.org/yngtodd/molecules> under pull requests for active pull requests or run the `tox` command and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

```
py.test test/test_molecules.py
```


CHAPTER 4

Credits

4.1 Development Lead

- Todd Young GitHub: [yngtodd](#)

4.2 Contributors

None yet. Why not be the first?

5.1 0.0.1 (2018-10-3)

- First release on PyPI.

6.1 molecules package

6.1.1 Subpackages

molecules.data package

Subpackages

molecules.data.dataloaders package

Submodules

molecules.data.dataloaders.dataset module

class molecules.data.dataloaders.dataset.**ContactMapDataset** (*path, name=None*)

Bases: object

Data handling for Numpy stored contact maps

load_data (*shape=None*)

Load numpy array data.

Parameters **shape** (*tuple, optional*) – Shape of the data. Format: (H x W x C)

Returns

- **X_train** (*np.ndarray*) – Training set.
- **X_test** (*np.ndarray*) – Test set.

`molecules.data.dataloaders.fspeptide` module

`molecules.data.dataloaders.onefme` module

`molecules.data.dataloaders.utils` module

Module contents

Module contents

`molecules.ml` package

Subpackages

`molecules.ml.rl` package

Subpackages

`molecules.ml.rl.models` package

Submodules

`molecules.ml.rl.models.fc_policy` module

Module contents

Submodules

`molecules.ml.rl.reinforce` module

Module contents

`molecules.ml.supervised` package

Module contents

`molecules.ml.unsupervised` package

Subpackages

`molecules.ml.unsupervised.conv_vae` package

Submodules

`molecules.ml.unsupervised.conv_vae.decoder` module

`molecules.ml.unsupervised.conv_vae.encoder` module

Module contents

0.1. molecules package

`molecules.ml.unsupervised.embeddings` package

CHAPTER 7

Feedback

If you have any suggestions or questions about **molecules** feel free to email me at youngmt1@ornl.gov.

If you encounter any errors or problems with **molecules**, please let me know! Open an Issue at the GitHub <http://github.com/yngtodd/molecules> main repository.

m

- `molecules`, [15](#)
- `molecules.data.dataloaders`, [15](#)
- `molecules.data.dataloaders.dataset`, [13](#)
- `molecules.ml`, [15](#)
- `molecules.ml.rl`, [15](#)
- `molecules.ml.rl.models`, [15](#)
- `molecules.ml.supervised`, [15](#)

C

ContactMapDataset (class in
molecules.data.dataloaders.dataset), 13

L

load_data() (molecules.data.dataloaders.dataset.ContactMapDataset
method), 13

M

molecules (module), 15
molecules.data.dataloaders (module), 15
molecules.data.dataloaders.dataset (module), 13
molecules.ml (module), 15
molecules.ml.rl (module), 15
molecules.ml.rl.models (module), 15
molecules.ml.supervised (module), 15