
worms Documentation

Release 0.1.26

Will Sheffler

Apr 25, 2018

Contents

1	worms	3
1.1	Features	3
1.2	Credits	3
2	Installation	5
2.1	Stable release	5
2.2	From sources	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	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.1.24 (2018-04-25)	15
6.2	0.1.23 (2018-04-24)	15
6.3	0.1.22 (2018-04-20)	15
6.4	0.1.21 (2018-04-19)	15
6.5	0.1.20 (2018-04-16)	15
6.6	0.1.19 (2018-04-6)	16
6.7	0.1.17 (2018-04-4)	16
6.8	0.1.16 (2018-03-15)	16
6.9	0.1.15 (2018-03-05)	16
6.10	0.1.14 (2018-02-28)	16
6.11	0.1.13 (2018-02-16)	16
6.12	0.1.12 (2018-02-16)	16
6.13	0.1.11 (2018-02-15)	16
6.14	0.1.10 (2018-02-15)	17
6.15	0.1.9 (2018-02-08)	17
6.16	0.1.8 (2018-02-07)	17

6.17	0.1.6 (2018-02-01)	17
6.18	0.1.6 (2018-02-01)	17
6.19	0.1.5 (2018-02-01)	17
6.20	0.1.4 (2018-02-01)	17
6.21	0.1.3 (2018-02-01)	17
6.22	0.1.2 (2018-01-23)	18
6.23	0.1.1 (2018-01-23)	18
7	Indices and tables	19

Contents:

CHAPTER 1

worms

Protein Origami via Genetic Fusions

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

1.1 Features

please send angry emails to willsheffler@gmail.com until there are docs here.

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install worms, run this command in your terminal:

```
$ pip install worms
```

This is the preferred method to install worms, 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 worms can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/willsheffler/worms
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/willsheffler/worms/tarball/master
```

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

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


CHAPTER 3

Usage

To use worms in a project:

```
import worms
```


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

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/willsheffler/worms/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

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

4.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/worms.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 worms
$ cd worms/
$ 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 flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 worms tests
$ python setup.py test or py.test
$ tox
```

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

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.

4.3 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/willsheffler/worms/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_worms
```


5.1 Development Lead

- Will Sheffler <willsheffler@gmail.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.1.24 (2018-04-25)

- change two-stage xindex threshold multiplier for 1st stage from 1.25 to 2.0

6.2 0.1.23 (2018-04-24)

- add initial draft of Una's criteria for P213 xtal

6.3 0.1.22 (2018-04-20)

- add initial draft of Una's criteria for P3, P4 and P6 layers

6.4 0.1.21 (2018-04-19)

- peace sign heterotrimer site compatibility bug fix
- xindex search now stores lists instead of one-per-bin

6.5 0.1.20 (2018-04-16)

- add more dihedral sym files

6.6 0.1.19 (2018-04-6)

- minor performance improvements in hash-index based search

6.7 0.1.17 (2018-04-4)

- hash-index based search for `Cyclic(..., origin_seg=...)`

6.8 0.1.16 (2018-03-15)

- I52 symmetry bug fix

6.9 0.1.15 (2018-03-05)

- add `NullCriteria` that always returns 0 err

6.10 0.1.14 (2018-02-28)

- fix provenance bug in 'cyclic entry' cases
- try to make serialization of `Segments` more efficient

6.11 0.1.13 (2018-02-16)

- raise exception if system too large

6.12 0.1.12 (2018-02-16)

- partial bignum fix

6.13 0.1.11 (2018-02-15)

- fix memory bug
- make distribution work better
- maybe fix pose bug, still some logic err, but maybe ok

6.14 0.1.10 (2018-02-15)

- add max_results option to grow
- fix C2 sym bug
- fix xform axis cen bug
- fix memory “bug” with batch parallel processing

6.15 0.1.9 (2018-02-08)

- add max_samples option to grow

6.16 0.1.8 (2018-02-07)

- origin_seg bug fix

6.17 0.1.6 (2018-02-01)

- middle-to-end cyclic fusions working
- add pretty logo of mid-to-end C3 fusion

6.18 0.1.6 (2018-02-01)

- bug fix in fullatom option

6.19 0.1.5 (2018-02-01)

- add fullatom option to Worms.sympose
- cyclic permutation working for simple beginning-to-end case

6.20 0.1.4 (2018-02-01)

- pypi deployment derp

6.21 0.1.3 (2018-02-01)

- pypi deployment derp

6.22 0.1.2 (2018-01-23)

- Add `__main__` for module to run tests
- move `worms.pdb` to `worms.data` because `pdb` is kinda reserved
- move utility stuff to `util.py`
- add some interactive visualization utils for debugging

6.23 0.1.1 (2018-01-23)

- First release on PyPI.

CHAPTER 7

Indices and tables

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