

---

# **cymminmax Documentation**

***Release 0.1.0***

**John Kirkham**

**Nov 12, 2017**



---

## Contents

---

<b>1</b>	<b>cyminmax</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>API</b>	<b>9</b>
<b>5</b>	<b>Contributing</b>	<b>11</b>
<b>6</b>	<b>Credits</b>	<b>15</b>
<b>7</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



A minmax implementation in Cython for use with NumPy

- Free software: BSD 3-Clause
- Documentation: <https://cyminmax.readthedocs.io>.

## 1.1 Features

- TODO

## 1.2 Credits

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





### 2.1 Stable release

To install cyminmax, run this command in your terminal:

```
$ pip install cyminmax
```

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

You can either clone the public repository:

```
$ git clone git://github.com/jakirkham/cyminmax
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/jakirkham/cyminmax/tarball/master
```

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

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



## CHAPTER 3

---

### Usage

---

To use cyminmax in a project:

```
import cyminmax
```



## CHAPTER 4

---

API

---



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/jakirkham/cyminmax/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

cyminmax could always use more documentation, whether as part of the official cyminmax 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/jakirkham/cyminmax/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 *cyminmax* for local development.

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

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

3. Install your local copy into an environment. Assuming you have conda installed, this is how you set up your fork for local development (on Windows drop *source*). Replace “<some version>” with the Python version used for testing.:

```
$ conda create -n cyminmaxenv python="<some version>"
$ source activate cyminmaxenv
$ 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:

```
$ flake8 cyminmax tests
$ python setup.py test or py.test
```

To get flake8, just conda install it into your environment.

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.



## 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.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5, and 3.6. Check [https://travis-ci.org/jakirkham/cyminmax/pull\\_requests](https://travis-ci.org/jakirkham/cyminmax/pull_requests) and make sure that the tests pass for all supported Python versions.

## 5.4 Tips

To run a subset of tests:

```
$ py.test tests/test_cyminmax.py
```



## CHAPTER 6

---

### Credits

---

#### 6.1 Development Lead

- John Kirkham, Howard Hughes Medical Institute <kirkhamj@janelia.hhmi.org>

#### 6.2 Contributors

None yet. Why not be the first?



## CHAPTER 7

---

### Indices and tables

---

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