
BMicro Documentation

Release 0.1.1

Paul Müller

Jan 15, 2021

CONTENTS:

- 1 Getting started 3**
 - 1.1 Installation 3
 - 1.2 Citing BMicro 3
- 2 Code reference 5**
- 3 Development 7**
 - 3.1 Development workflow 7
 - 3.2 Documentation 7
 - 3.3 Tests 8
 - 3.4 Making a new release 8
 - 3.5 Continuous integration 8
- 4 Changelog 9**
 - 4.1 version 0.1.1 9
 - 4.2 version 0.1.0 9
- 5 Bibliography 11**
- 6 Indices and tables 13**

This is BMicro, a graphical user interface for Brillouin microscopy data evaluation. This is the documentation of bmlab version 0.1.1.

GETTING STARTED

1.1 Installation

To install BMicro, use one of the following methods:

- **from PyPI:** `pip install bmicro`
- **from sources:** `pip install .`

1.2 Citing BMicro

If you use BMicro in a scientific publication, please cite it with:

BMicro developers (2021), BMicro version X.X.X: Python library for the post-measurement analysis of real-time deformability cytometry data sets [Software]. Available at <https://github.com/BrillouinMicroscopy/BMicro>.

CODE REFERENCE

TODO

DEVELOPMENT

This section gives an overview about everything you need to know if you wish to contribute to BMicro.

3.1 Development workflow

The main branch for developing BMicro is `main`. If you want to make small changes like one-liners, documentation, or default values in the configuration, you may work on the `main` branch. If you want to change more, please (fork BMicro and) create a separate branch, e.g. `my_new_feature_dev`, and create a pull-request once you are done making your changes. Please make sure to edit the [Changelog](#).

Very important: Please try to always pull with rebase

```
git pull --rebase
```

instead of

```
git pull
```

to prevent confusions in the commit history.

3.2 Documentation

It is always helpful to have code examples and thorough descriptions in a documentation. We use sphinx-autodoc for the *code reference*, which means that the docstrings of your functions and classes are automatically rendered. Please make sure that this is working properly - go to the `docs` directory and execute:

```
pip install -r requirements.txt
sphinx-build . _build
```

This will create a file `_build/index.html` which you can open in your favorite browser.

3.3 Tests

We try to adhere to test-driven development. Please always write test functions for your code. You can run all tests via

```
python setup.py test
```

3.4 Making a new release

The release process of BMicro is completely automated. All you need to know is that you have to create an incremental tag:

```
git tag -a "0.1.3"
# or (if you have set up PGP)
git tag -s "0.1.3"
# and finally
git push --tags
```

For more information on how automatic deployment to PyPI works, please read on.

3.5 Continuous integration

The following things are automated:

- pytest and flake8 on Linux, macOS, and Windows via GitHub Actions: <https://github.com/BrillouinMicroscopy/BMicro/actions?query=workflow%3AChecks>

You should always check that all checks pass before you merge a pull request (A green state on your local machine does not mean a global green state).

- automatic deployment to PyPI on tag creation via GitHub Actions: <https://github.com/BrillouinMicroscopy/BMicro/actions?query=workflow%3A%22Release+to+PyPI%22>

Paul Müller created the **BMicro** package on PyPI and gave the user `ci_bm` permission to upload new releases. The password for this user is an [organization secret](#).

- documentation is built automatically (for all tags and for the latest commit to the main branch) on readthedocs: <https://readthedocs.org/projects/BMicro/builds/>
- coverage statistics are done with codecov: <https://codecov.io/gh/BrillouinMicroscopy/BMicro>

Please try stay above 90% coverage.

Badges for all of these CI tasks are in the main `README.rst` file.

CHANGELOG

List of changes in-between bmlab releases.

4.1 version 0.1.1

- CI automation

4.2 version 0.1.0

- dummy release

BILBLIOGRAPHY

INDICES AND TABLES

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