
BMicro Documentation

Release 0.2.2

Paul Müller

Mar 22, 2022

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 ## 0.2.2 - 2022-03-22	11
4.2	version	11
4.3	version ### Changed	11
4.4	version - Code cleanup #145	11
4.5	version	11
4.6	version ### Fixed	11
4.7	version - Bring back multiprocessing support #149	11
4.8	version - Don't open multiple instances on Win when evaluating #147	11
4.9	version - Don't open multiple instances on MacOS with DMG package #140	11
4.10	version	11
4.11	version ## 0.2.1 - 2022-03-21	11
4.12	version	11
4.13	version ### Changed	11
4.14	version - Code cleanup #135 #142	11
4.15	version	11
4.16	version ### Fixed	11
4.17	version - Don't crash when extracted peaks are on a straight line #141	11
4.18	version - Don't crash when trying to evaluate #139	11
4.19	version	11
4.20	version ## 0.2.0 - 2022-03-15	11
4.21	version	11
4.22	version ### Added	11
4.23	version - Implement showing 3D data #128	11
4.24	version - Remember last used folder #132	11
4.25	version	11
4.26	version ### Fixed	11

4.27	version - Fix warning when file could not be opened #129	11
4.28	version - Fix build #131	11
4.29	version	11
4.30	version ## 0.1.6 - 2022-02-18	11
4.31	version	11
4.32	version ### Added	11
4.33	version - Add an about menu #119	11
4.34	version	11
4.35	version ### Changed	11
4.36	version - Update to PyQt6 to support M1 Macs #124	11
4.37	version	11
4.38	version ### Fixed	11
4.39	version - Request minimum required bmlab version #122	11
4.40	version - Pin Shapely to 1.8.0 #120 #121	11
4.41	version	11
4.42	version ## 0.1.5 - 2022-02-16	11
4.43	version	11
4.44	version ### Fixed	11
4.45	version - Adjust release pipeline to properly build and upload release artifacts #118 9347bce74f7bf1fb40ab823967f4be8490af47b9 eb5d7e7d2bcb4cf590e967a01a71d00aefced3b5	11
4.46	version	11
4.47	version ## 0.1.4 - 2022-02-16	11
4.48	version	11
4.49	version ### Added	11
4.50	version - Add keyboard shortcuts for file actions #117	11
4.51	version - Allow finding peaks in extraction view for all calibrations #111	11
4.52	version - Implement automatic peak finding in calibration view #110	11
4.53	version - Implement automatically finding peaks #99	11
4.54	version - Allow to browse through calibration spectra #89	11
4.55	version - Show selected extraction points in table, allow editing #85	11
4.56	version - Show colorbar with labels in evaluation view #70	11
4.57	version - Automatically load session if it exists #67	11
4.58	version	11
4.59	version ### Changed	11
4.60	version - Add action to exit app #107	11
4.61	version - Deploy with python-3.9 #105	11
4.62	version - Update pyinstaller version #104	11
4.63	version - Require python >= 3.7 #103	11
4.64	version - Add more package requirements #98	11
4.65	version - Suppress mean of empty slice warning #95	11
4.66	version - Suppress useless warning #88	11
4.67	version - Automatically set image shape on orientation change #84	11
4.68	version - Adjust to streamlined bmlab interface #83	11
4.69	version - Adjust to changes in bmlab #82	11
4.70	version - Refactor use of session #81	11
4.71	version - Only check bmlab main branch in workflows #72	11
4.72	version	11
4.73	version ### Fixed	11
4.74	version - Fix rotation labels and make rotations more generic #114	11
4.75	version - Fix matplotlib deprecation warnings #115	11
4.76	version - Fix build on appveyor and for MacOS #116	11
4.77	version - Correctly show evaluation plot after closing a file #102	11
4.78	version - Fix crash when clicking if no file is loaded #101	11
4.79	version - Try to fix slow evaluation plot update #97	11

4.80	version - Recalculate derived values when calibration changes #90	11
4.81	version - Correctly scale the data if the unit is GHz #77	11
4.82	version - Only update necessary tab when switching tabs #76	11
4.83	version - Show measurement of arbitrary dimensionality correctly #74	11
4.84	version - Actually run pytest in checks #73	11
4.85	version	11
4.86	version ## 0.1.3	11
4.87	version	11
4.88	version - setup: use “pytest” command instead of deprecated “setup.py test”	11
4.89	version - ui: add app icon (#9)	11
4.90	version - ui: data tab (#11)	11
4.91	version - build: add Windows and macOS build pipeline (#1)	11
4.92	version	11
4.93	version ## 0.1.2	11
4.94	version	11
4.95	version - CI automation	11
4.96	version	11
4.97	version ## 0.1.1	11
4.98	version	11
4.99	version - Test CI automation	11
4.100	version	11
4.101	version ## 0.1.0	11
4.102	version	11
4.103	version - dummy release	11
5	Bibliography	13
6	Indices and tables	15

This is BMicro, a graphical user interface for Brillouin microscopy data evaluation. This is the documentation of bmlab version 0.2.2.

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 (2022), BMicro version X.X.X: Python library for the post-measurement analysis of Brillouin microscopy 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 [bmlab](#) or [BMicro](#).

3.1 Development workflow

We use [GitHub projects](#) to manage the development workflow of [bmlab](#) and [BMicro](#). The main development project board is [Brillouin Evaluation in Python](#).

The development is split into “User Stories”, each of which is a collection of issues (identified via titles in the issues). The current work in progress (WIP) branch is named according to the currently active user story (e.g. *1-smoke-test*). Issues that are not part of the current user story should still be addressed in the current WIP branch.

Once you wish to address an issue, drag it from the “Open” or “Ready” column of the project board to the “In progress” column. Once you finished working on an issue, drag it to the “Done” column, but don’t close it yet (It should be discussed first in the dev meeting).

Notes:

- Please write test functions and keep code coverage above 90%.
- Please make sure to always edit the changelog for [BMicro](#) or [bmlab](#).
- 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. This also applies to [bmlab](#).

3.3 Tests

We try to adhere to test-driven development. Please always write test functions for your code. Make sure you have the required packages installed:

```
pip install -r tests/requirements.txt
```

You can run all tests via

```
python -m pytest tests
```

To check for code coverage, make sure the *coverage* Python package is installed and run

```
coverage run --source="bmicro" -m pytest tests
coverage report
```

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 GPG)
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 ## 0.2.2 - 2022-03-22

4.2 version

4.3 version ### Changed

4.4 version - Code cleanup #145

4.5 version

4.6 version ### Fixed

4.7 version - Bring back multiprocessing support #149

4.8 version - Don't open multiple instances on Win when evaluating #147

4.9 version - Don't open multiple instances on MacOS with DMG package #140

4.10 version

4.11 version ## 0.2.1 - 2022-03-21

4.12 version

4.13 version ### Changed

4.14 version - Code cleanup #135 #142

4.15 version

4.16 version ### Fixed

4.17 version - Don't crash when extracted peaks are on a straight line #141

4.18 version - Don't crash when trying to evaluate #139

4.19 version

4.1. version ## 0.2.2 - 2022-03-22

4.20 version ## 0.2.0 - 2022-03-15

BILBLIOGRAPHY

INDICES AND TABLES

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