
PyJibe Documentation

Release 0.6.6

Paul Müller

Jan 27, 2020

Contents

1	Getting started	3
1.1	Installation	3
1.2	Update	3
2	User Interface	5
2.1	Overview	5
2.2	Force-Distance Analysis	5
3	Quick Guides	13
3.1	Importing a pre-computed training set	13
4	Changelog	15
4.1	version 0.6.6	15
4.2	version 0.6.5	15
4.3	version 0.6.4	15
4.4	version 0.6.3	15
4.5	version 0.6.2	16
4.6	version 0.6.1	16
4.7	version 0.6.0	16
4.8	version 0.5.7	16
4.9	version 0.5.6	16
4.10	version 0.5.5	16
4.11	version 0.5.4	17
4.12	version 0.5.3	17
4.13	version 0.5.2	17
4.14	version 0.5.1	17
4.15	version 0.5.0	17
4.16	version 0.4.4	17
4.17	version 0.4.3	18
4.18	version 0.4.2	18
4.19	version 0.4.1	18
4.20	version 0.4.0	18
5	Bibliography	19
6	Indices and tables	21

PyJibe

PyJibe is a user interface for data analysis in atomic force microscopy (AFM) with an emphasis on biological specimens, such as tissue sections or single cells. This is the documentation of PyJibe version 0.6.6.

This documentation is also available as a [PDF](#).

1.1 Installation

PyJibe can be installed via multiple channels:

1. **Windows installer:** Download the latest version for your architecture (e.g. `PyJibe_X.Y.Z_win_64bit_setup.exe` for 64bit Windows) from the official [release page](#).
2. **macOS:** Download the latest version (`PyJibeApp_X.Y.Z.dmg`) from the official [release page](#).
3. **Python 3.6 with pip:** PyJibe can easily be installed with `pip`:

```
pip install pyjibe
```

To start PyJibe, simply run `python -m pyjibe` or just `pyjibe` in a command shell.

1.2 Update

If you install a newer (or older) version of PyJibe, the previously installed version will be automatically uninstalled.

If you installed `pyjibe` with `pip`, you may upgrade it with:

```
pip install --upgrade pyjibe
```

1.2.1 How to cite

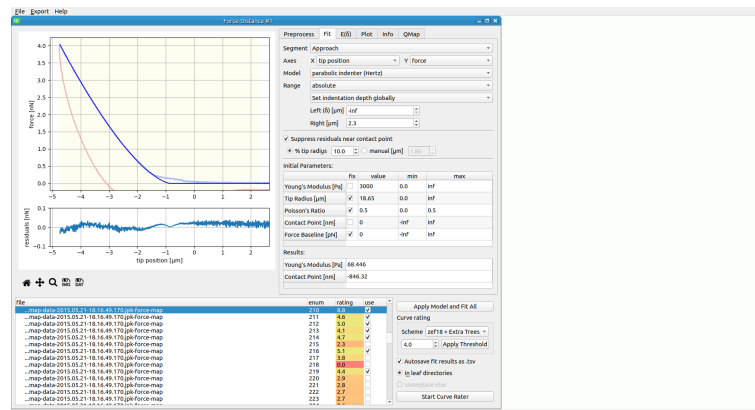
If you use PyJibe in a scientific publication, please consider citing Müller et al., *BMC Bioinformatics* (2019) [MAM+19].

2.1 Overview

When you first start PyJibe, you will be greeted with an empty user interface. To open experimental data, choose one of the corresponding options in the *File* menu.

2.2 Force-Distance Analysis

Force-distance (FD) measurements consist of an approach and a retract curve. The point of maximum indentation is usually defined by the acquisition software as a *setpoint* using an active feedback loop.

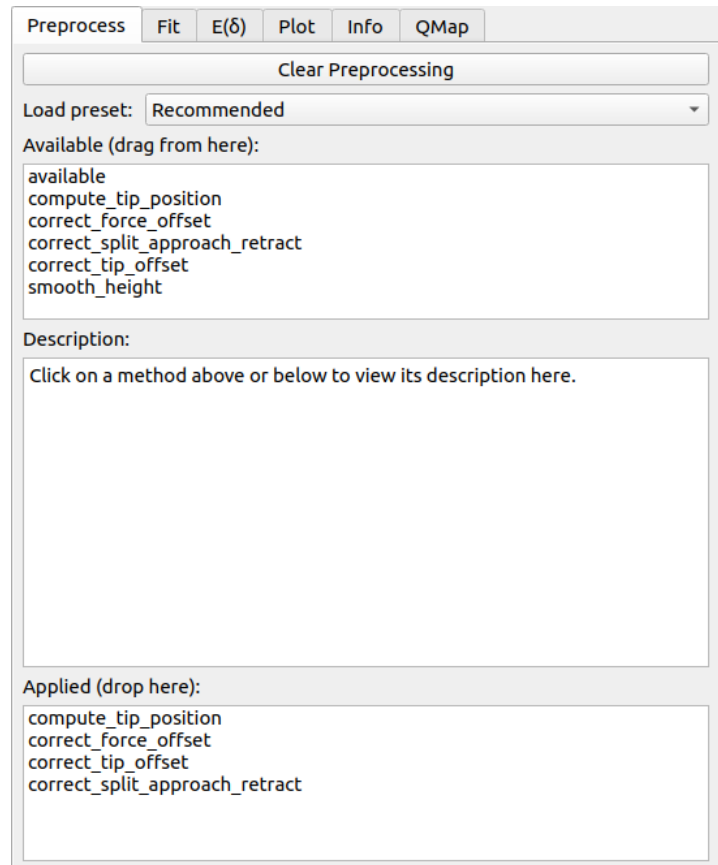


The screenshot shows a typical data analysis in PyJibe. A *.jpk-force-map* file has been loaded. The plot at the top shows the experimental FD curve (approach light blue, retract light red), the model fit (dark blue), and the fitting region (shaded yellow). Below are the fit residuals. On the right are the main controls (discussed in the sections below). At the bottom is the curve list, highlighting which curve is currently shown at the top. At the bottom right are the *curve list controls*.

Note: Notice how the fit residuals become small around the contact point? PyJibe comes with the feature *Suppress residuals near contact point* (in the fit tab on the right). This option suppresses the contributions of the FD curve near the contact point. This feature was designed for biological samples, where the physical tip-sample interactions near the “contact point” are not well-understood [MAM+19]. The contributions near the contact point are suppressed to give the remainder of the indentation curve more weight.

The dataset shown resembles a 2D scan of AFM force-distance curves from a live zebrafish spinal cord microtome section and is available on figshare [MMG20] (same data as in [MKH+20] fig. 1e and [MAM+19] fig. 3a,d,g).

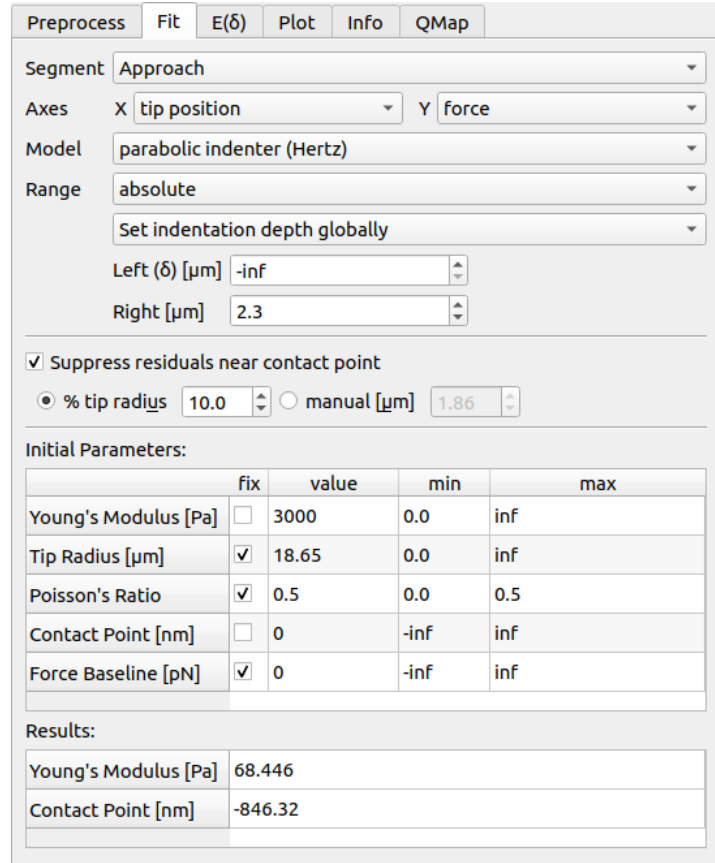
2.2.1 Tab: Preprocess



Before starting to fit FD data, it is important to perform the correct steps. Usually, this includes computing the tip position and correcting for a force and tip position offset. The offset corrections are only necessary for visualization purposes - Force offset and contact point are usually free fitting parameters.

To define a new preprocessing pipeline, click on *Clear Preprocessing* at the top of the panel. You may either load predefined presets or compile your own preprocessing pipeline via drag&drop from the *Available* to the *Applied* list controls. When selecting a preprocessing item in one of the list controls, its description is shown in the text control at the center.

2.2.2 Tab: Fit



Preprocess Fit $E(\delta)$ Plot Info QMap

Segment Approach

Axes X tip position Y force

Model parabolic indenter (Hertz)

Range absolute

Set indentation depth globally

Left (δ) [μm] -inf

Right [μm] 2.3

Suppress residuals near contact point

% tip radius 10.0 manual [μm] 1.86

Initial Parameters:

	fix	value	min	max
Young's Modulus [Pa]	<input type="checkbox"/>	3000	0.0	inf
Tip Radius [μm]	<input checked="" type="checkbox"/>	18.65	0.0	inf
Poisson's Ratio	<input checked="" type="checkbox"/>	0.5	0.0	0.5
Contact Point [nm]	<input type="checkbox"/>	0	-inf	inf
Force Baseline [pN]	<input checked="" type="checkbox"/>	0	-inf	inf

Results:

Young's Modulus [Pa]	68.446
Contact Point [nm]	-846.32

PyJibe offers a variety of options for fitting FD data. In the upper part, you may select the segment (approach or retract), the axes, the fit model, and the fitting range.

Range: The fitting range can be set to absolute (relative to the X axis after preprocessing or relative to the contact point. In the latter case, the fit is repeated four times, updating the new contact point at each iteration. You have three options for setting the indentation depth (left part of the fit interval):

1. globally: For each FD curve, the same indentation depth is used (after preprocessing has been applied).
2. individually: You select the indentation depth for each FD curve separately.
3. guess optimal indentation depth: The optimal indentation depth is guessed using the $E(\delta)$ -curve.

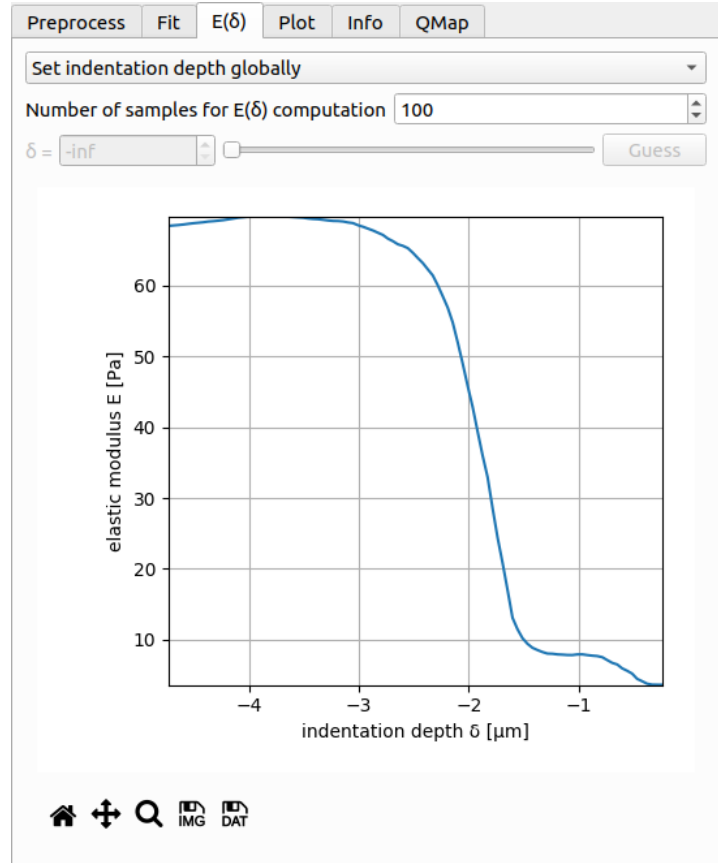
Suppress residuals near contact point: This option multiplies the residuals with a linear ramp, left and right of the contact point. The ramp has a value of 0 at the contact point and a value of 1 at the distance defined by the controls. All other residuals are not changed. This feature was designed for biological samples, where the physical tip-sample interactions near the “contact point” are not well-understood [MAM+19]. The contributions near the contact point are suppressed to give the remainder of the indentation curve more weight.

Ancillary Parameters: Ancillary parameters (not shown in the screenshot here), are parameters that are defined in the fitting model. They are computed prior to the fit and can be set as fitting parameters. Standard models usually do not have ancillary parameters.

Initial Parameters: These are the parameters set initially during fitting. In the table, you may choose which parameters should remain fixed during fitting, what the initial values should be, and in which interval this value may be varied.

Results: The fit results of the varied parameters are shown here.

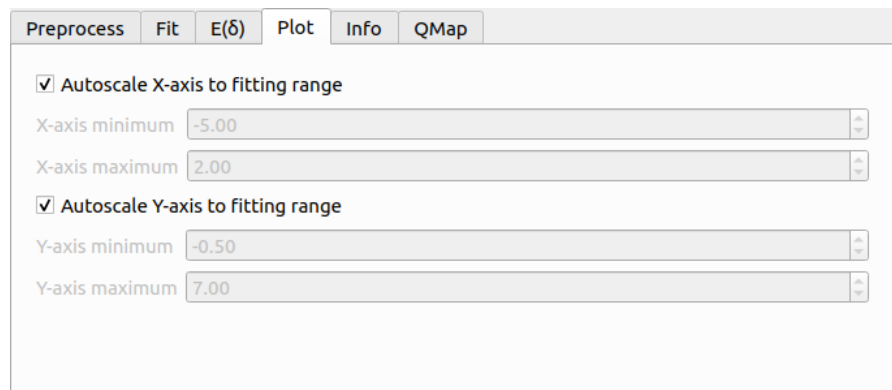
2.2.3 Tab: $E(\delta)$



The $E(\delta)$ curve is used to test whether the fitted Young's modulus E is dependent on the fitting interval. For a reliable fit, the $E(\delta)$ curve should exhibit a plateau. This is the case for the present example, where the fitted value of E does not vary much in the range from $\delta = -2\mu\text{m}$ to $\delta = -4.8\mu\text{m}$ (maximum indentation).

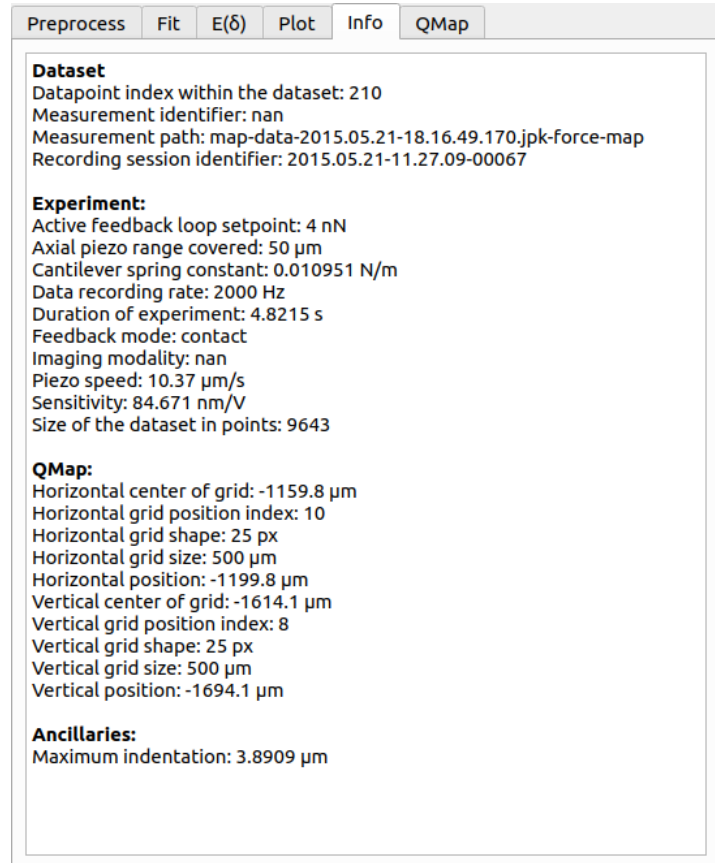
The control at the top is identical to the control in the *fitting tab*. You may choose the number of samples of the $E(\delta)$. The controls for setting the indentation depth manually are enabled when you select *set indentation depth individually* in the dropdown menu. Below the plot you have the options to export the $E(\delta)$ curve as an image or as a data file (.tsv). If you would like to export all $E(\delta)$ curves of the dataset, you can do so via the *Export* menu of the main window.

2.2.4 Tab: Plot



The plotting tab allows you to modify the plot settings. By default, the plot is adapted to the fitting interval. You may, however, modify the axes ranges manually by unchecking the check boxes.

2.2.5 Tab: Info



The info tab shows metadata related to the currently shown curve. Unknown values are indicated as *nan*.

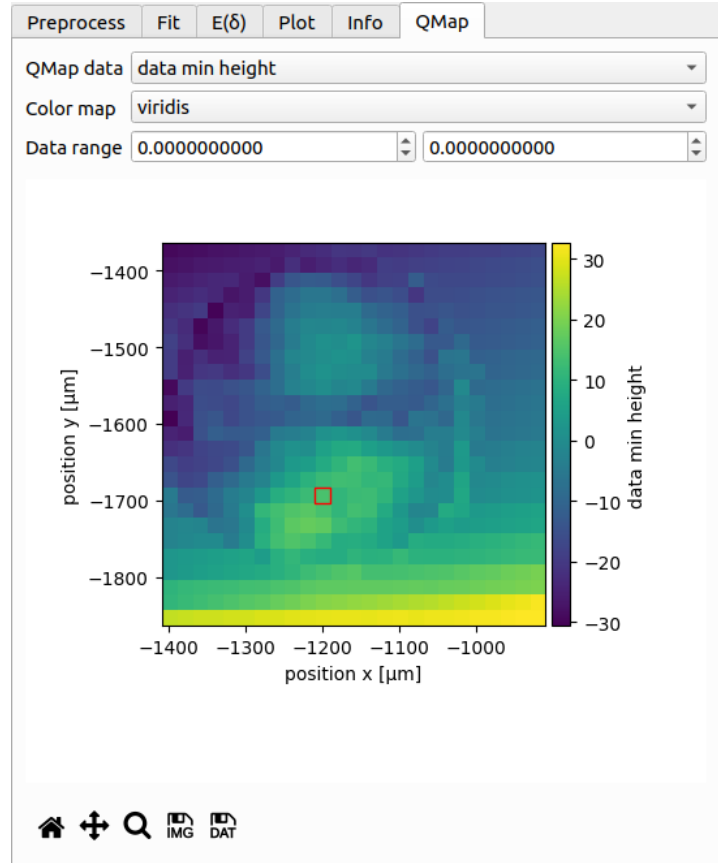
Dataset: All information related to the dataset on disk and of the curve relative to this dataset.

Experiment: All information on the actual experiment.

QMap: If the curve is part of a quantitative map (2D FD scan), then the scan grid properties and the curve position are listed.

Ancillaries: Ancillary parameters are computed for each model. Here, the *maximum indentation* is listed, which is the difference between the fitted contact point and the value of the tip position where the fitted curve has its maximum. Fit models may have their own specific ancillary parameters.

2.2.6 Tab: QMap

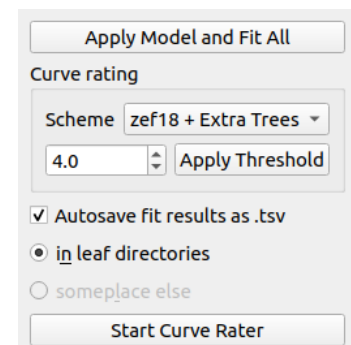


If the dataset consists of a quantitative map, then this map is shown here. It serves as an interactive overview of the dataset. You can choose which data to plot (e.g. piezo height, fitted parameters, curve rating, scan order), which colormap to use, and how the colormap should be scaled. The coordinate axes are identical to those shown in the info tab.

The red square in the plot indicates which curve is currently shown. You may click on the plot to select curves manually (as opposed to using the curve list at the bottom of the window).

Below the plot are controls for exporting the qmap as an image or as a text file.

2.2.7 Curve list controls



The curve list controls perform operations on the entire FD curve list. The *Apply Model and Fit All* button does exactly what it says.

Curve rating allows you to rate each indentation curve. This is useful e.g. when you would like to sort out bad curves automatically. In this example, all curves with a rating below a threshold of 4.5 are excluded from further analysis using the *zef18 + Extra Trees* rating scheme [MAM+19].

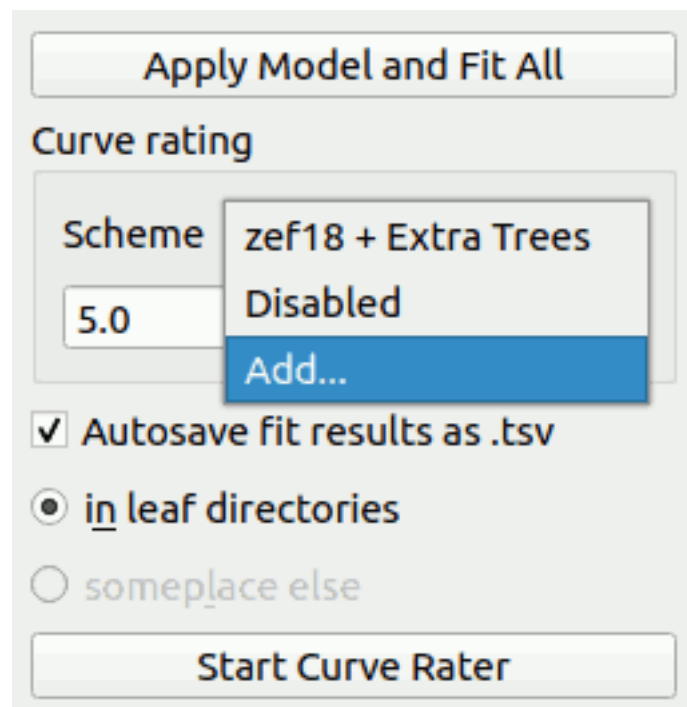
Below the curve rating options, you may enable or disable autosaving of the fit results. Only the results for curves that are selected as “use” in the curve list are exported.

For more information on rating, please have a look at the [nanite documentation](#). The button at the bottom starts the PyJibe curve rater which is compatible to the nanite rating workflow. If you would like to import your own training set, please read the quick guide *Importing a pre-computed training set*.

3.1 Importing a pre-computed training set

A training set is a set of text files containing rating features and manual user ratings. Training sets are the basis of the rating scheme employed in PyJibe (and nanite) [MAM+19]. By default, PyJibe comes with the *zef18* training set [MMG18]. For general information about how manual rating and generation of the training set works, please refer to the [nanite rating workflow](#).

In order to import a training set into PyJibe, the text files (`train_feat_*.txt` and `train_response.txt`) must be zipped and named according to the scheme `ts_NAME.zip`, where `NAME` is a descriptive name.



The import in PyJibe is carried out by selecting *Add...* in the *Scheme* dropdown menu, located in the *Curve rating* box (lower right of a Force-Distance analysis). A dialog will ask for the zipped training set (e.g. `ts_organoids19.zip`).

The import is persistent, i.e. the training set is extracted to the user's configuration directory and thus does not have to be imported again the next time PyJibe is run.

List of changes in-between PyJibe releases.

4.1 version 0.6.6

- fix: typo lead to error when using rating threshold

4.2 version 0.6.5

- build: macOS build failed due to PyInstaller issue 4626

4.3 version 0.6.4

- fix: info tab did not display QMap metadata
- fix: do not fit the entire dataset when applying a rating threshold but take the values from previous ratings
- fix: deselecting curves of a qmap resulted in ValueError if QMap tab was selected
- fix: set correct display range for residuals
- docs: UI section for basics FD analysis
- build: Windows build broken since 0.5.6

4.4 version 0.6.3

- fix: support JPK data recorded in the “force-modulation” feedback mode (bump afmformats from 0.5.0 to 0.5.1)

- fix: make sure ancillary parameters are computed from the initial parameters set in the user interface (not from the default model parameters)

4.5 version 0.6.2

- enh: allow to select which metadata is exported
- fix: do not apply and fit to all before exporting metadata (user may have performed individual fits)
- setup: bump nanite from 1.3.0 to 1.4.0

4.6 version 0.6.1

- build: workaround for Pyinstaller issue 4626

4.7 version 0.6.0

- feat: initial support for ancillary parameters
- fix: always display parameter units
- fix: other minor UI bugs
- setup: bump afmformats from 0.4.1 to 0.5.0
- setup: bump nanite from 1.2.4 to 1.3.0

4.8 version 0.5.7

- enh: allow to also set right part of fitting range individually
- fix: improve layout of FD fitting panel
- setup: bump nanite from 1.2.3 to 1.2.4 (improved default params)

4.9 version 0.5.6

- setup: bump nanite from 1.2.2 to 1.2.3 (fixes issue with fits not being redone when the user changes an initial parameter)
- fix: IndexError when editing the fitting range

4.10 version 0.5.5

- maintenance release

4.11 version 0.5.4

- enh: write maximum indentation to statistics file (#3)
- fix: unicode characters were not read correctly from statistics file by libre office (added UTF-8 BOM)

4.12 version 0.5.3

- enh: improved visualization of meta data parameters in the Info tab
- setup: bump afmformats from 0.3.0 to 0.4.1
- setup: bump nanite from 1.2.0 to 1.2.2

4.13 version 0.5.2

- ref: bump nanite to 1.2.0
- ref: migrate to afmformats (0.3.0)

4.14 version 0.5.1

- fix: missing title in FD window and missing FD number in export menu
- enh: add help menu with link to docs, about, and software used (#1)
- docs: add quick guide for importing a nanite training set
- docs: add black/white logo

4.15 version 0.5.0

- feat: allow to import training sets produced with nanite
- ref: major code refactoring of force-distance interface and cleanup
- setup: bump nanite dependency to $\geq 1.1.1$

4.16 version 0.4.4

- ci: pin 'joblib==0.11.0' on travis-CI (workaround for infinite loop in macOS build, <https://github.com/pyinstaller/pyinstaller/issues/4067>)
- docs: add installation instructions

4.17 version 0.4.3

- maintenance release (migrate to GitHub infrastructure)
- experimental macOS builds

4.18 version 0.4.2

- enh: merge training set and regressor selection into scheme selection (prevents confusion for users that are unsure what to choose)
- enh: user rating is saved during rating, not afterwards
- enh: rating containers for user rating can be reused explicitly
- enh: qmap plot now better distinguishes between unavailable and invalid data
- ref: migrate user rating output to nanite

4.19 version 0.4.1

- fix: autoscaling according to fitting range did not work
- fix: not possible to load .jpk-force-map files
- ref: migrate qmap generation to nanite

4.20 version 0.4.0

- migration to GitHub

CHAPTER 5

Bibliography

CHAPTER 6

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Bibliography

- [MKH+20] Stephanie Möllmert, Maria A. Kharlamova, Tobias Hoche, Anna V. Taubenberger, Shada Abuhattum, Veronika Kuscha, Thomas Kurth, Michael Brand, and Jochen Guck. Zebrafish spinal cord repair is accompanied by transient tissue stiffening. *Biophysical Journal*, 118(2):448–463, jan 2020. doi:10.1016/j.bpj.2019.10.044.
- [MMG20] Stephanie Möllmert, Paul Müller, and Jochen Guck. Two-dimensional force-distance afm map of a live zebrafish spinal cord microtome section. *Figshare*, Jan 2020. doi:10.6084/m9.figshare.11709720.v1.
- [MAM+19] Paul Müller, Shada Abuhattum, Stephanie Möllmert, Elke Ulbricht, Anna V. Taubenberger, and Jochen Guck. Nanite: using machine learning to assess the quality of atomic force microscopy-enabled nano-indentation data. *BMC Bioinformatics*, 20(1):1–9, sep 2019. doi:10.1186/s12859-019-3010-3.
- [MMG18] Paul Müller, Stephanie Möllmert, and Jochen Guck. Atomic force microscopy indentation data of zebrafish spinal cord sections. *Figshare*, 11 2018. doi:10.6084/m9.figshare.7297202.v1.