PyJibe Documentation

Release 0.9.1

Paul Müller

Jun 25, 2021

CONTENTS

1 Getting started **3**
- 1.1 Installation . . . 3
- 1.2 Update . . . 3
- 1.3 Supported file formats . . . 3
- 1.4 How to cite . . . 3

2 User Interface **5**
- 2.1 Overview . . . 5
- 2.2 Force-Distance Analysis . . . 5

3 Tutorials **13**
- 3.1 T1: Stiffness of two PAAm hydrogels . . . 13

4 Quick Guides **19**
- 4.1 Importing a pre-computed training set . . . 19

5 Changelog **21**
- 5.1 version 0.9.1 . . . 21
- 5.2 version 0.9.0 . . . 21
- 5.3 version 0.8.8 . . . 21
- 5.4 version 0.8.7 . . . 22
- 5.5 version 0.8.6 . . . 22
- 5.6 version 0.8.5 . . . 22
- 5.7 version 0.8.4 . . . 22
- 5.8 version 0.8.3 . . . 22
- 5.9 version 0.8.2 . . . 23
- 5.10 version 0.8.1 . . . 23
- 5.11 version 0.8.0 . . . 23
- 5.12 version 0.7.6 . . . 23
- 5.13 version 0.7.5 . . . 23
- 5.14 version 0.7.4 . . . 23
- 5.15 version 0.7.3 . . . 24
- 5.16 version 0.7.2 . . . 24
- 5.17 version 0.7.1 . . . 24
- 5.18 version 0.7.0 . . . 24
- 5.19 version 0.6.8 . . . 24
- 5.20 version 0.6.7 . . . 24
- 5.21 version 0.6.6 . . . 25
- 5.22 version 0.6.5 . . . 25
- 5.23 version 0.6.4 . . . 25

5.24 version 0.6.3 . . . 25
5.25 version 0.6.2 . . . 25
5.26 version 0.6.1 . . . 25
5.27 version 0.6.0 . . . 26
5.28 version 0.5.7 . . . 26
5.29 version 0.5.6 . . . 26
5.30 version 0.5.5 . . . 26
5.31 version 0.5.4 . . . 26
5.32 version 0.5.3 . . . 26
5.33 version 0.5.2 . . . 27
5.34 version 0.5.1 . . . 27
5.35 version 0.5.0 . . . 27
5.36 version 0.4.4 . . . 27
5.37 version 0.4.3 . . . 27
5.38 version 0.4.2 . . . 27
5.39 version 0.4.1 . . . 28
5.40 version 0.4.0 . . . 28

6 Bilbliography **29**

7 Indices and tables **31**

Bibliography **33**

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 0.9.1.

This documentation is also available as a PDF.

CHAPTER

ONE

GETTING STARTED

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
```

1.3 Supported file formats

PyJibe relies on the afmformats package. A list of supported file formats can be found here.

1.4 How to cite

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

> Paul Müller and others (2019), PyJibe version X.X.X: Atomic force microscopy data analysis [Software]. Available at https://github.com/AFM-analysis/PyJibe.

Please also consider citing Müller et al., *BMC Bioinformatics* (2019) [MAM+19] (for fitting and rating force-indentation data).

CHAPTER

TWO

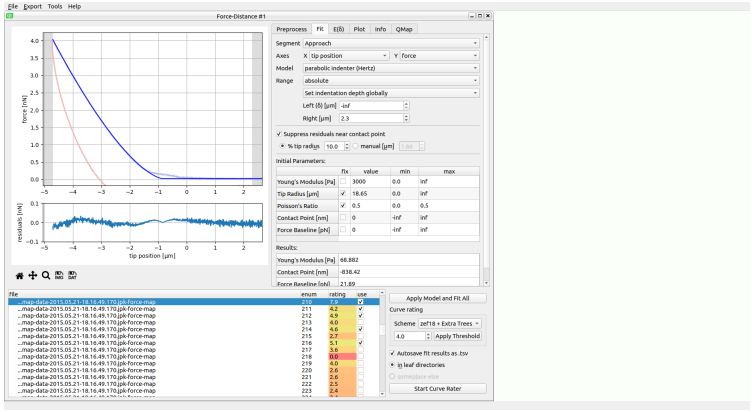
USER INTERFACE

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.

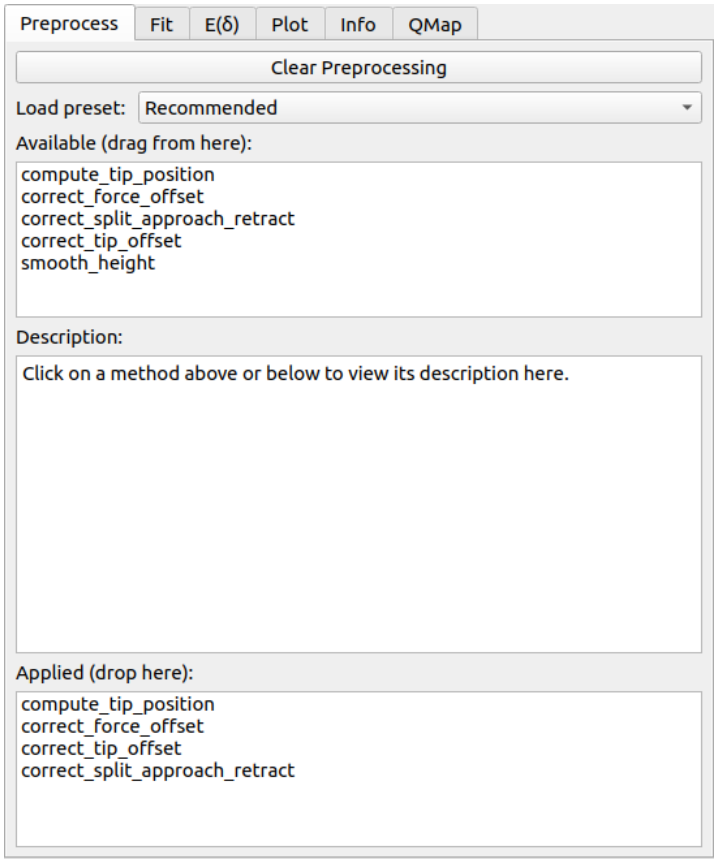


The screenshot shows a typical data analysis in PyJibe. A *.jpk-force-map* file has been loaded. The plot at the top shows the experimental FD curve (approach light blue, retract light red), the model fit (dark blue), and the fitting region (white region, bounded by gray background in the upper plot). 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).

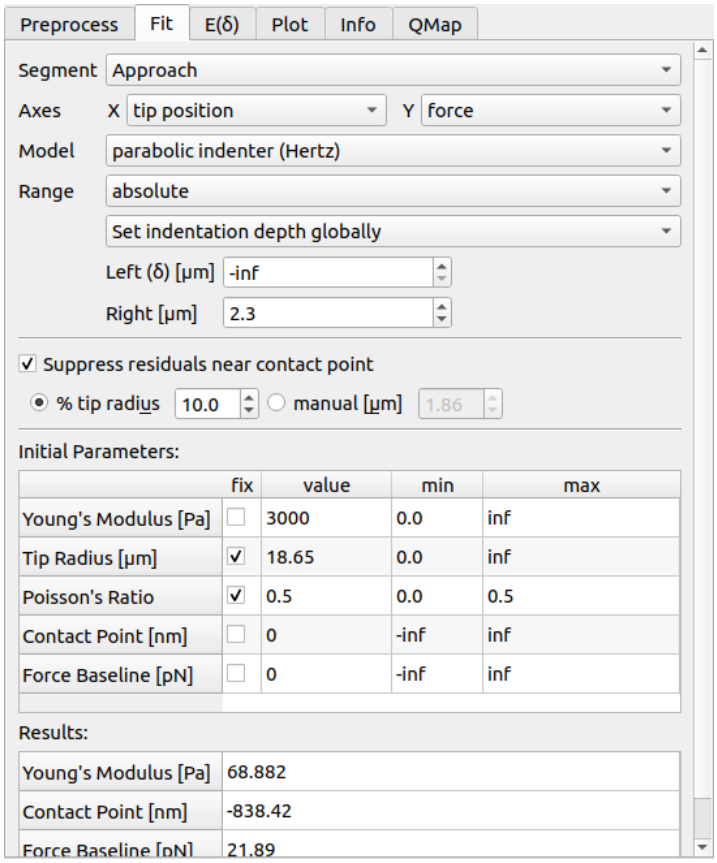
2.2.1 Tab: Preprocess



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 on of the list controls, its description is shown in the text control at the center.

2.2.2 Tab: Fit



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 option 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()-curve*.

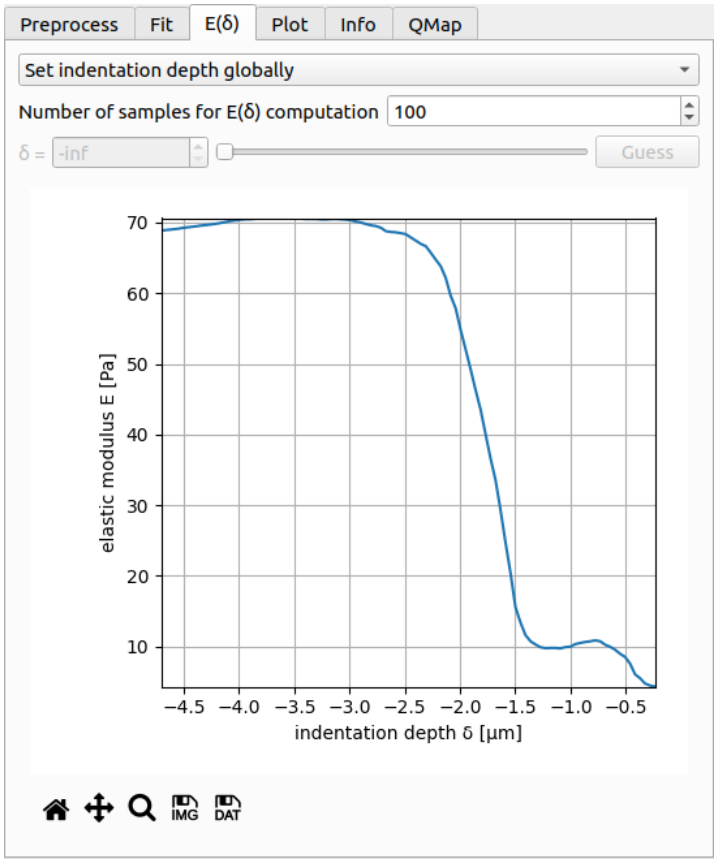
Suppress residuals near contact point: This option multiplies the residuals with a linear ramp, left and right of the contact point. The ramp has a value of 0 at the contact point and a value of 1 at the distance defined by the controls. All other residuals or not changed. This feature was designed for 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.

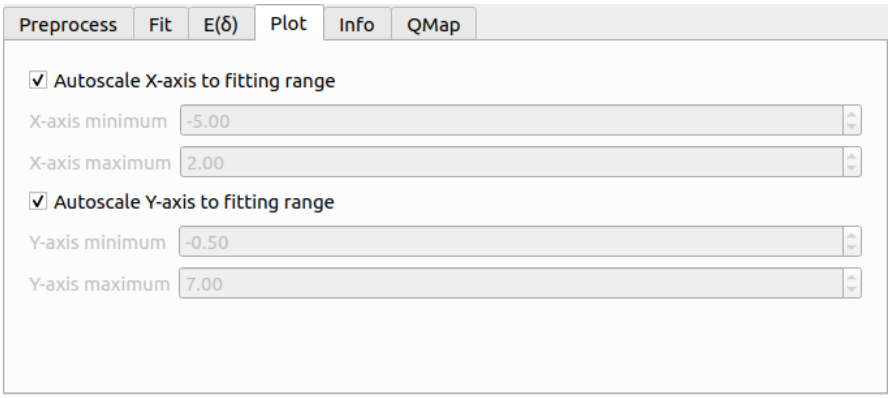
2.2.3 Tab: E()



The E() curve is used to test whether the fitted Young's modulus E is dependent on the fitting interval. For a reliable fit, the E() 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 =-2.5µm to =-4.8µm (maximum indentation).

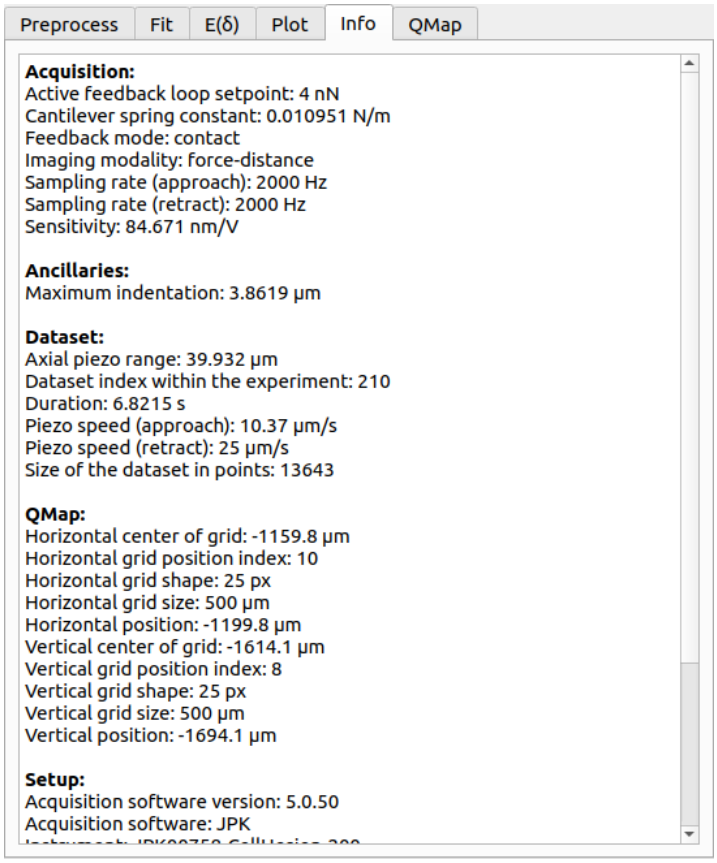
The control at the top is identical to the control in the *fitting tab*. You may choose the number of samples of the E(). The controls for setting the indentation depth manually are enabled when you select *set indentation depth individually* in the dropdown menu. Below the plot you have the options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

Acquisition: Settings of the acquisition software

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.

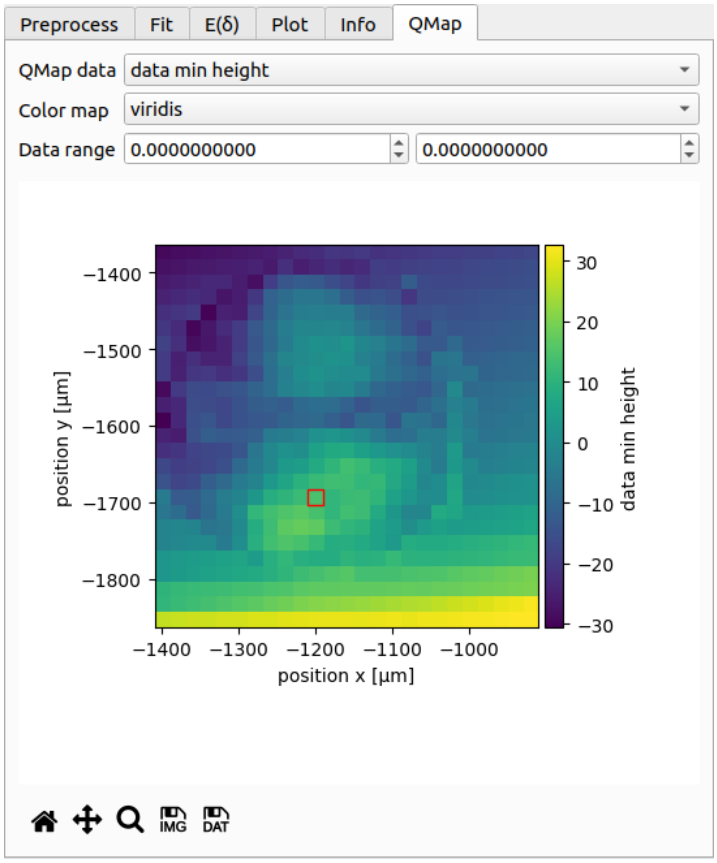
Dataset: Measurement parameters of the dataset

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

Setup: Information about the AFM setup used

Storage: Information about the data file, which are not particularly important for the experiment, but allow to identify a dataset or curve

2.2.6 Tab: QMap

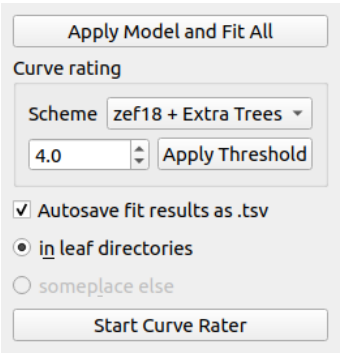


If the dataset consists of a quantitative map, then this map is shown here. It serves as an interactive overview of the dataset. You can choose which data to plot (e.g. piezo height, fitted parameters, curve rating, scan order), which colormap to use, and how the colormap should be scaled. The coordinate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

CHAPTER

THREE

TUTORIALS

3.1 T1: Stiffness of two PAAm hydrogels

3.1.1 Introduction

In this tutorial, you will quantify the stiffness of two polyacrylamide (PAAm) hydrogels. The gels were produced according to the protocol in [RLY+17] with an expected Young's modulus of 0.8 to 1.2 kPa (compliant gel) and 25 to 28 kPa (stiff gel).

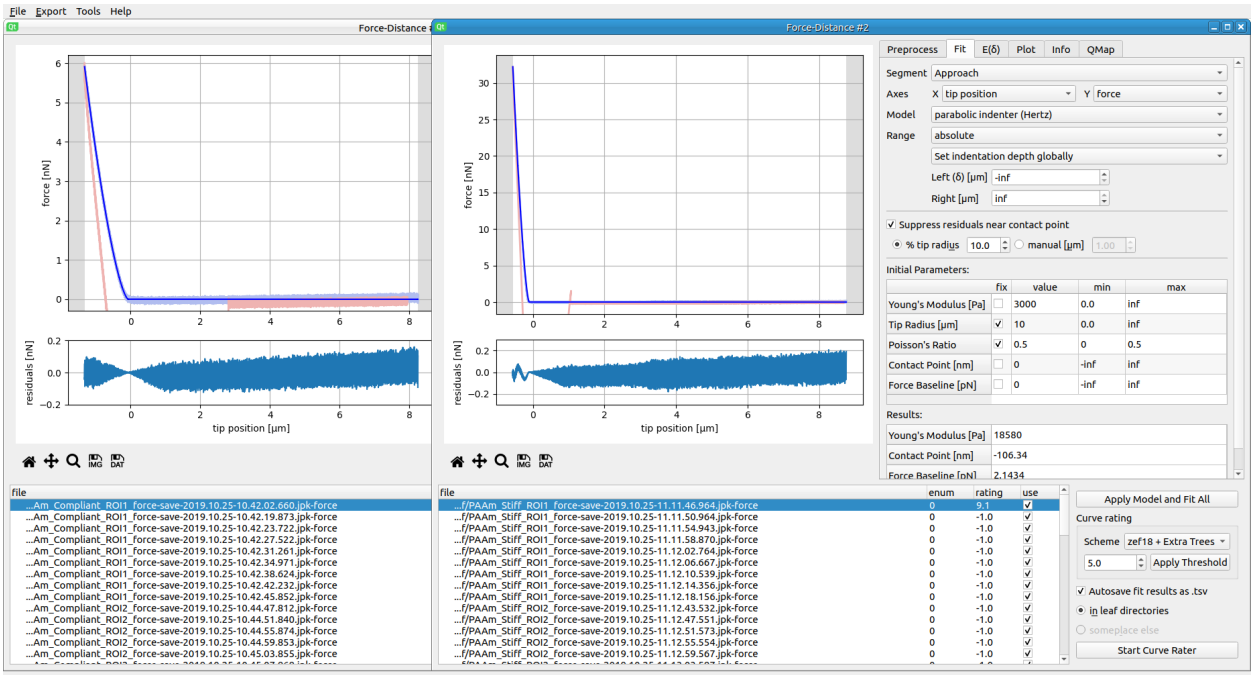
3.1.2 Prerequisites

- PyJibe version 0.6.7 or above
- Complete dataset available on figshare [RMG20]

3.1.3 Data analysis

To make data analysis easier, we first divide the dataset by copying the files of the compliant gel (*PAAm_Compliant_*.jpk-force*) and the stiff gel (*PAAm_Stiff_*.jpk-force*) into two separate folders "compliant" and "stiff".

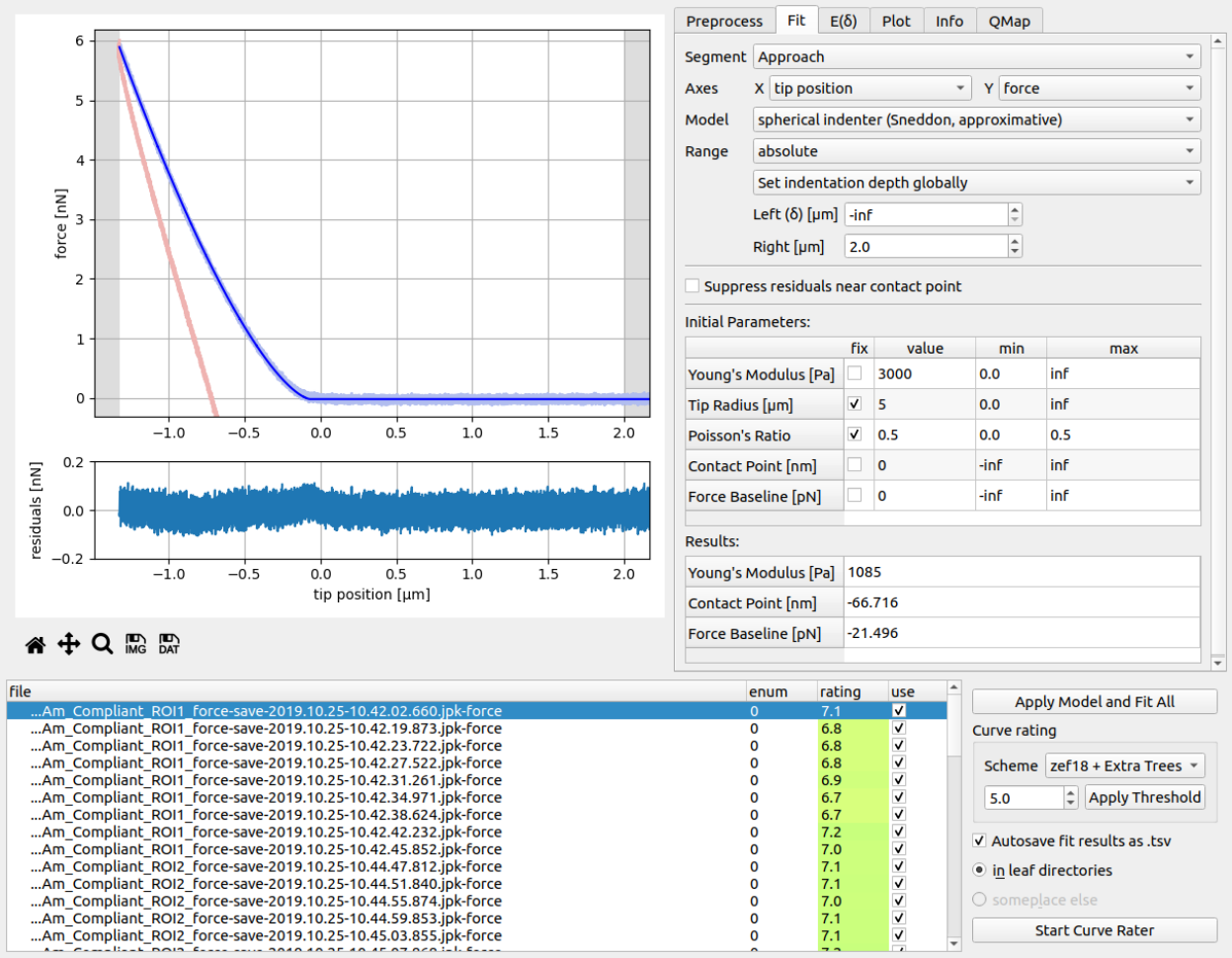
In PyJibe, we load each of the folders using *File | Open bulk data*. We now have two subwindows called *Force-Distance #1* and *Force-Distance #2* (screenshot below).



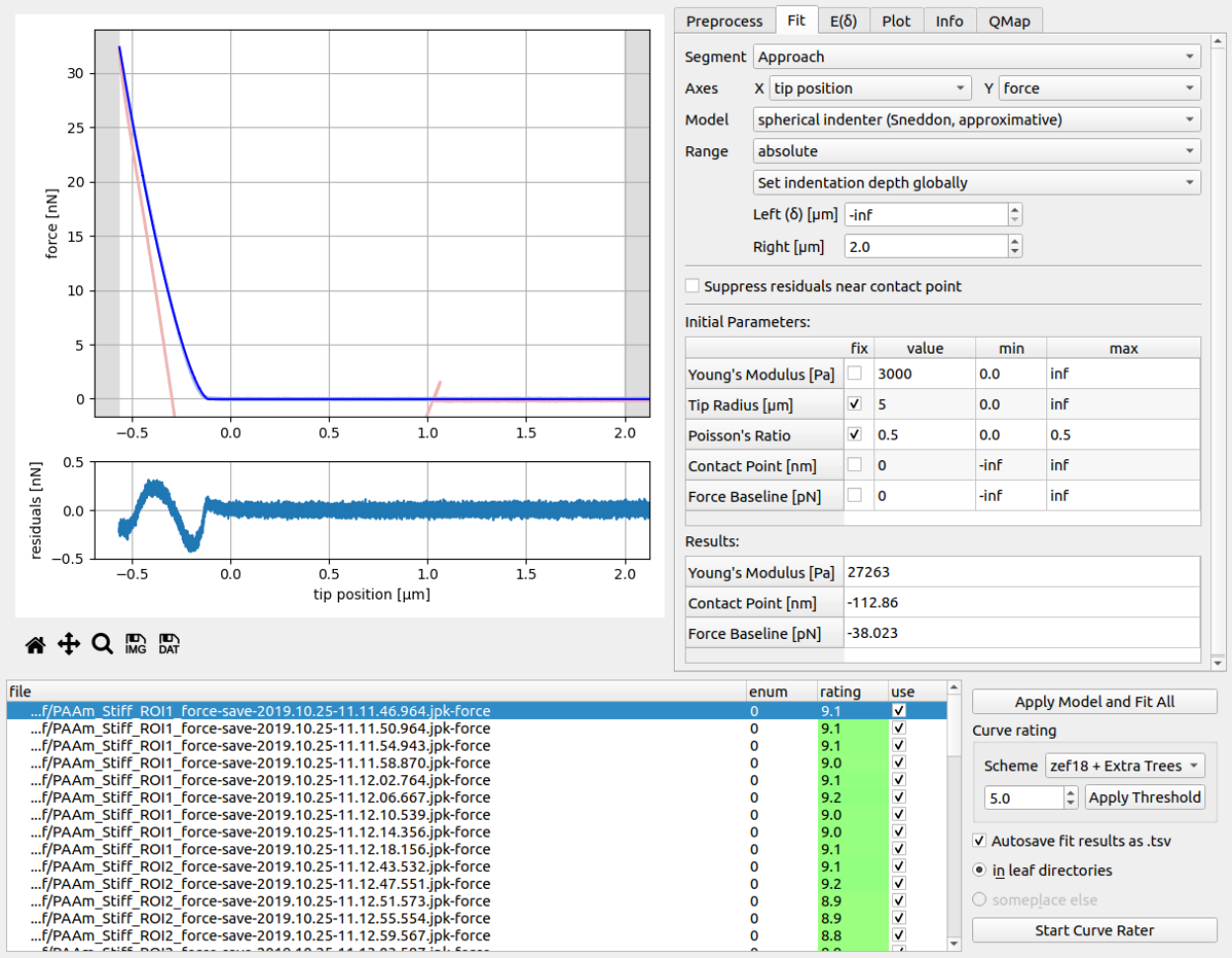
We change the analysis pipeline in each of the subwindows as follows:

- In the *Fit* tab, set model to *spherical indenter (Sneddon, approximative)*. The measurements were performed using a spherical indenter. The *approximative* description indicates, that this model is an analytical representation of the Sneddon model (as used by the JPK analysis software). See `nanite.model.model_sneddon_spherical_approximation` for more information.
- In the *Fit* tab, set the right range to 2 µm. This primarily helps with the visualization of the data.
- In the *Fit* tab, uncheck the *Suppress residuals near contact point* check box. This feature is not necessary for PAAm gels.
- In the *Fit* tab, set the initial parameter *Tip Radius* to 5 µm.

Now, click the button *Apply Model and Fit All*. The results should look like this, for the compliant gel…



. . . and for the stiff gel:



There are several things to note here:

- The fit of the compliant gel looks much better than the fit of the stiff gel: There seems to be a systematic (sine-like) deviation from the fit. One could only speculate about the reason for this deviation. However, compared to the maximal indentation force (setpoint at 32 nN, see *Info* tab), the deviation is comparatively small.
- Furthermore, the stiff gel measurements exhibit a "ringing", which is clearly visible in the approach part. This is a result of the *force-modulation* feedback mode. The gel is not probed in *contact* mode, but in a mixture between contact mode and intermittend mode (See the NanoWizard User Manual v. 4.2, sec. 5.7.). Maybe this ringing is not visible for the compliant gel, because of the setpoint at 6n N (The setpoint likely defines the amplitude of the force-modulation feedback mode and for the compliant gels, this amplitude might be below the noise level).
- The stiff gel gets a better rating than the compliant gel with the *zef18 + Extra Trees* rating scheme (please see the nanite rating workflow and [MAM+19] for how rating works). Of course, this observation is misleading - it nicely illustrates a limit of machine learning. The *zef18* training set was created using zebrafish spinal cord section measurements. In the context of hydrogels, it does not make much sense, although it appears to be consistent and gives both, compliant and stiff gels, a "good" rating.

3.1.4 Results

You might have realized that PyJibe creates the file *pyjibe_fit_results_leaf.tsv* in each of the measurement folders (if the *Autosave fit results as .tsv* check box is checked). These files contain (amongst other things) the fit results of each curve. With a simple Python script, we can visualize the Young's modulus of the two gels:

```
import matplotlib.pylab as plt
import searborn as sns

data_compl = pandas.read_table("./compliant/pyjibe_fit_results_leaf.tsv")
data_stiff = pandas.read_table("./stiff/pyjibe_fit_results_leaf.tsv")

sns.set_style("darkgrid")

plt.subplot(121, title="compliant hydrogel")
sns.boxplot("Young\'s Modulus [Pa]", data=data_compl, fliersize=0,
            color="#d6ff7d")
sns.swarmplot("Young\'s Modulus [Pa]", data=data_compl,
              size=5, color=".3", linewidth=0)

plt.subplot(122, title="stiff hydrogel")
sns.boxplot("Young\'s Modulus [Pa]", data=data_stiff, fliersize=0,
            color="#98ff80")
sns.swarmplot("Young\'s Modulus [Pa]", data=data_stiff,
              size=5, color=".3", linewidth=0)

plt.show()
```

The compliant hydrogel has a Young's modulus of 1090 $\pm$ 10 Pa (mean $\pm$ SD) and the stiff hydrogel has a Young's modulus of 27680 $\pm$ 270 Pa. These values agree well with the values we expected initially.

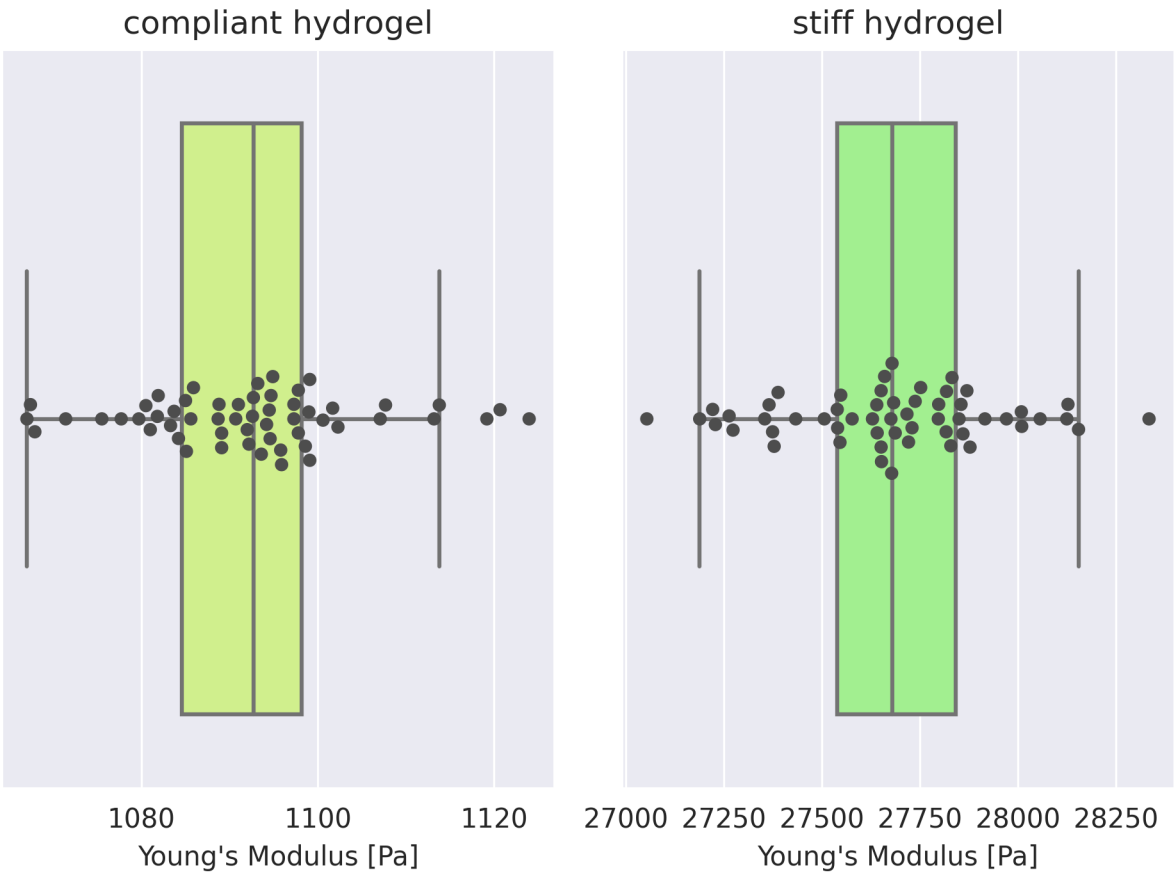


Fig. 3.1: Comparison of the hydrogels. Note that the X axes are scaled differently.

CHAPTER

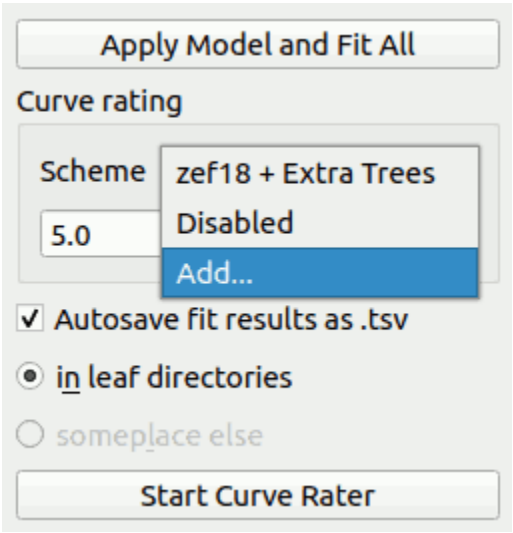
FOUR

QUICK GUIDES

4.1 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.

CHAPTER

FIVE

CHANGELOG

List of changes in-between PyJibe releases.

5.1 version 0.9.1

- docs: minor update
- fix: regression where left fitting range could not be set
- fix: improved checks for InfDoubleSpinBox which acted up when entering absolutely sane floating point values

5.2 version 0.9.0

- feat: allow users to enter missing metadata during loading of data files (#5)
- fix: automatically append .h5 suffix when rating datasets
- enh: add busy cursor where applicable
- ref: directory for importing training datasets for rating changed (but I think this currently affects nobody)
- setup: dump appdirs dependency (moved to QSettings)
- setup: bump afmformats from 0.12.5 to 0.13.3 (mostly #5 and speed)
- setup: bump nanite from 1.7.3 to 1.7.4 (mostly #5)
- tests: increase coverage

5.3 version 0.8.8

- fix: handle bad curves properly when running bulk analysis (#13)
- ref: make sure there are no duplicate files when loading data (possibly fixes #12)
- ref: cleanup - call overloaded processEvents
- ref: properly decorate pyqtSlots
- tests: add helper function for setting up datasets

5.4 version 0.8.7

- fix: add workaround for macOS where the non-native PyQt5 dialogs did not allow browsing to "/Volumes" (it is now added to the sidebar)
- build: bump afmformats from 0.12.4 to 0.12.5 (OSError: "Too many files open" with the JPK file format)
- build: bump nanite from 1.6.2 to 1.7.3
- build: pin PyQt5 for GitHub releases
- tests: don't used deprecated weight_cp function from nanite

5.5 version 0.8.6

- build: bump afmformats from 0.12.2 to 0.12.4 (AttributeError when reading HDF5 files)
- tests: deprecate setup.py test

5.6 version 0.8.5

- build: migrate from travisCI to GitHub Actions
- build: overhauled build pipeline (#9)

5.7 version 0.8.4

- build: add pyinstaller hooks for afmformats

5.8 version 0.8.3

- fix: main window not focused after startup
- fix: Windows installation location was confusing and did not coincide configuration file location
- fix: files are sorted before batch loading
- fix: change visualization of fitting region by gray wraparound instead of yellow highlight (not visible on all screens)
- setup: bump afmformats from 0.11.0 to 0.12.2 (improved JPK file format speed)

5.9 version 0.8.2

- setup: bump matplotlib to >=3 (NavigationToolbar2QT modifications)

5.10 version 0.8.1

- fix: correctly handle lmfit models with expressions
- setup: bump nanite from 1.6.0 to 1.6.2 (improve contact point retrieval for bad data, check models during registration)

5.11 version 0.8.0

- fix: exclude misplaced 'available' method from preprocessors
- setup: bump nanite from 1.5.5 to 1.6.0 (improved contact point estimation - this may introduce small changes in the fit result which should not be significant, but which justify a new minor version)

5.12 version 0.7.6

- maintenance release (fix macOS build)
- setup: bump afmformats from 0.10.2 to 0.11.0
- setup: bump nanite from 1.5.4 to 1.5.5

5.13 version 0.7.5

- maintenance release (fix build)

5.14 version 0.7.4

- setup: bump afmfomats from 0.10.0 to 0.10.2
- setup: bump nanite from 1.5.2 to 1.5.4
- build: add command-line option –version to just print the version and exit

5.15 version 0.7.3

- fix: update initial parameters when the user edits the left fitting interval boundary
- fix: conical indenter model did not work, because of a wrong (and unnecessary) entry in the *human_units* scheme.

5.16 version 0.7.2

- fix: setting bad fit range lead to ValueError during plotting

5.17 version 0.7.1

- fix: wrong SI units assigend to fit results parameters
- setup: bump afmformats from 0.9.0 to 0.10.0 (new formats: JPK (.jpk-qi-data), AFM workshop (.csv), NT-MDT (.txt exported by Nova))
- setup: bump nanite from 1.4.1 to 1.5.2

5.18 version 0.7.0

- feat: add data conversion tool
- enh: allow to select multiple files when loading data
- ref: changed order and clarified meaning of options in file menu
- setup: bump afmformats from 0.7.0 to 0.9.0 (fix HDF5 export, support new file formats)

5.19 version 0.6.8

- setup: bump afmformats from 0.5.1 to 0.7.0 (improved metadata view in info tab)
- setup: bump nanite from 1.4.1 to 1.5.1 (compatibility to afmformats)
- enh: fit results and parameter export dialog now supports new groups in afmformats 0.7.0
- enh: autosave now only stores fit results and rating parameters
- enh: remember scroll position in info tab

5.20 version 0.6.7

- setup: bump nanite from 1.4.0 to 1.4.1 (baseline is now a free parameter by default)
- docs: add first tutorial (PAAm gels)

5.21 version 0.6.6

- fix: typo lead to error when using rating threshold

5.22 version 0.6.5

- build: macOS build failed due to PyInstaller issue 4626

5.23 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

5.24 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)

5.25 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

5.26 version 0.6.1

- build: workaround for Pyinstaller issue 4626

5.27 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

5.28 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)

5.29 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

5.30 version 0.5.5

- maintenance release

5.31 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)

5.32 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

5.33 version 0.5.2

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

5.34 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

5.35 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 >=1.1.1

5.36 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

5.37 version 0.4.3

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

5.38 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

5.39 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

5.40 version 0.4.0

- migration to GitHub

CHAPTER

SIX

BILBLIOGRAPHY

CHAPTER

SEVEN

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 nanoindentation 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.

[RLY+17] Gonzalo Rosso, Ivan Liashkovich, Peter Young, Dominik Röhr, and Victor Shahin. Schwann cells and neurite outgrowth from embryonic dorsal root ganglions are highly mechanosensitive. *Nanomedicine: Nanotechnology, Biology and Medicine*, 13(2):493–501, feb 2017. doi:10.1016/j.nano.2016.06.011.

[RMG20] Gonzalo Rosso, Paul Müller, and Jochen Guck. Atomic force microscopy indentation data of stiff and compliant polyacrylamide hydrogels. *Figshare*, Jan 2020. doi:10.6084/m9.figshare.11637675.v3.