DCscope Documentation

Release 2.28.1

Paul Müller and Christoph Herold

Jun 29, 2026

CONTENTS

DCscope (formerly Shape-Out) is a user interface for data analysis in deformability cytometry (DC). This is the documentation of DCscope version 2.28.1.

This documentation is also available as a PDF.

CHAPTER

ONE

GETTING STARTED

1.1 Installation

DCscope can be installed via multiple channels.

1. **Windows installer:** Download the latest version for your architecture (i.e. `DCscope_X.Y.Z_win_64bit_setup.exe`) from the official release page.
2. **Python 3.8 with pip:** DCscope can easily be installed with pip:

   ```
   python3 -m pip install dcscope
   ```

   To start DCscope, simply run `python3 -m dcscope` or `dcscope` in a command shell.

> **Note**
>
> There used to be macOS builds, but many users reported they were not working. Since the GitHub Actions build pipeline works (executable binary runs), this is either a CPU architecture issue or some kind of macOS safeguard (software signing?). At the moment, we don't have the resources to look into this, but we would greatly appreciate help in the form of a pull request.

1.2 Update

DCscope automatically searches for updates (you may opt-out via the Help menu) and notifies the user when a new version is available.

1. **Windows installer:** The older version of DCscope will be automatically uninstalled when installing a new version.
2. **macOS:** The older version of DCscope will be automatically uninstalled when installing a new version.
3. **Python 3.8 and pip:**

   ```
   pip install --upgrade dcscope
   ```

1.3 Supported data file formats

DCscope exclusively supports the .rtdc data file format. This file format is based on the HDF5 format which makes it portable, consistent, and efficient.

If you have **.tdms-based datasets** (created using antique versions of Shape-In), you can still use the data after you have converted them to .rtdc files using DCKit, which provides a convenient GUI for several other RT-DC data management tasks as well. The .rtdc file format is faster than the .tdms file format, occupies less disk space, and consists of only one file per measurement.

If you have **raw .rtdc files** that only contain the recorded images and no extracted events, you can perform segmentation and feature extraction using ChipStream.

1.4 How to cite

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

> Paul Müller and others (2019), DCscope (formerly Shape-Out) version 2.X.X: Graphical user interface for analysis and visualization of RT-DC data sets [Software]. Available at https://github.com/DC-analysis/DCscope.

If the journal does not accept `and others`, you can fill in the missing names in the authors section of the pyproject.toml file.

CHAPTER

TWO

RT-DC BASICS

This section conveys the basic understanding necessary for analyzing and interpreting RT-DC data. If you have the feeling that something is not covered here, please create an issue on GitHub.

2.1 Working Principle

In Real-time deformability cytometry (RT-DC), small objects, such as cells or beads, are flushed through a narrow channel at high speeds. The flow profile inside the channel causes hydrodynamic stresses that deform these objects [MOG+15], [MMM+17]. For each event (detected object), an image is recorded and analyzed in real-time [ORM+15]. The analysis consists of the extraction of several features from the event images, such as deformation, projected area, or average brightness. These features can then be used to identify and characterize sample populations; For instance, major blood cells and their pathological changes in disease conditions [THO+17].

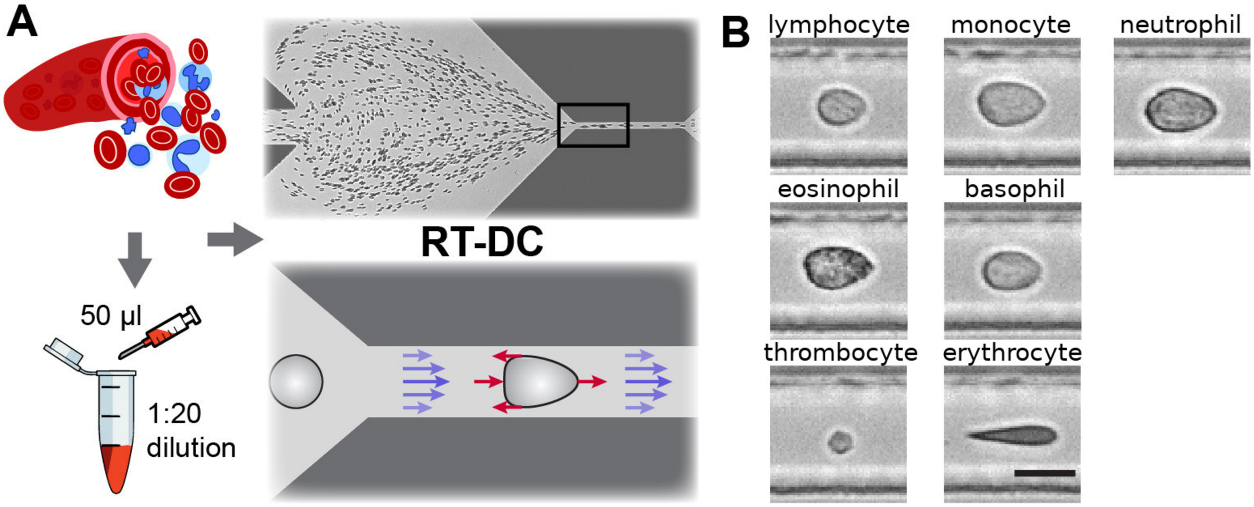


Fig. 2.1: Single-cell, morpho-rheological phenotyping of blood. (A) Analysis of whole, diluted blood. Hydrodynamic shear forces (red arrows) induce deformation of cells passing through a microfluidic channel (20x20 μm^2) at speeds of more than 30 cm/s (blue arrows). (B) Representative images of blood cell types acquired. Scale bar is 10 µm. Figure and caption adapted from Toepfner et al.[1].

---

[1] *Detection Of Human Disease Conditions By Single-Cell Morpho-Rheological Phenotyping Of Whole Blood* by Toepfner et al., licensed under CC BY 4.0 [THO+17].

2.2 Measured Features

A multitude of features can be extracted from the data recorded during an RT-DC measurement. These features are mostly computed live during data acquisition and are stored alongside the raw image data. Here, only the most important features are discussed. A full list of the features available in DCscope is maintained in the dclab documentation.

2.2.1 Area and porosity

The area is the projected object area which is determined via the contour of the binarized event image. DCscope differentiates between two types of area, area of the measured contour ("Measured area [px]") and area of the convex contour ("Convex area [px]" and "Area [µm^2]"). The convex contour is the convex hull of the measured contour and enables a quantification of porosity (convex to measured area ratio). The porosity is often used for filtering, e.g. to *remove high-porosity dirt particles in a preprocessing step*.

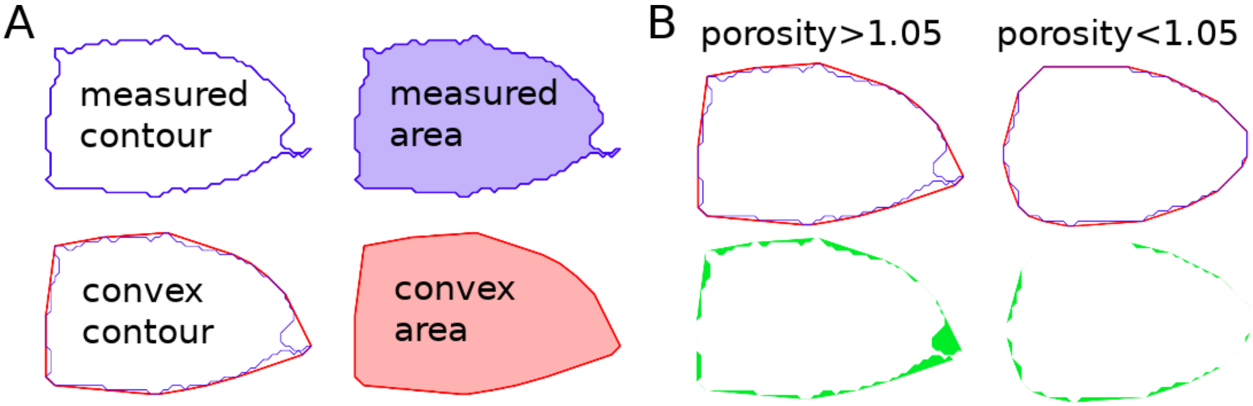


Fig. 2.2: Visualization of porosity. (A) The measured contour (blue line) defines the measured area (blue shade). The convex contour (red line) results in an area (red shade) that is usually larger than the measured area. (B) The porosity is the ratio between measured and convex contour. The difference (the "pores") between the measured and convex areas is indicated in green. Porosity is often used to remove events with non-physical contours, e.g. for cells all events with a porosity above 1.05.

A porosity of 1 means that the measured contour is convex. Note that the porosity can only assume values larger than 1. Also note that the convex contour/area is computed on the same pixel grid as the measured contour/area and is, as such, subject to pixelation artifacts.

2.2.2 Bounding box

The bounding box of an event image is the smallest rectangle (with its sides parallel to the x and y axes) that can hold the event contour. The aspect ratio of the bounding box is the rectangle's side length along x divided by the side length along y. The size of the bounding box along x and y as well as its aspect ratio are often used for filtering.

2.2.3 Brightness within contour

Quantifying the brightness values within the image contour yields information on object properties such as homogeneity or density. For instance, it has been shown that the quantities "mean brightness" and "convex area" are sufficient to identify (and count) all major blood cells in a drop of blood [THO+17].

In addition to the average brightness values, DCscope also has access to the standard deviation of the brightness in each image.

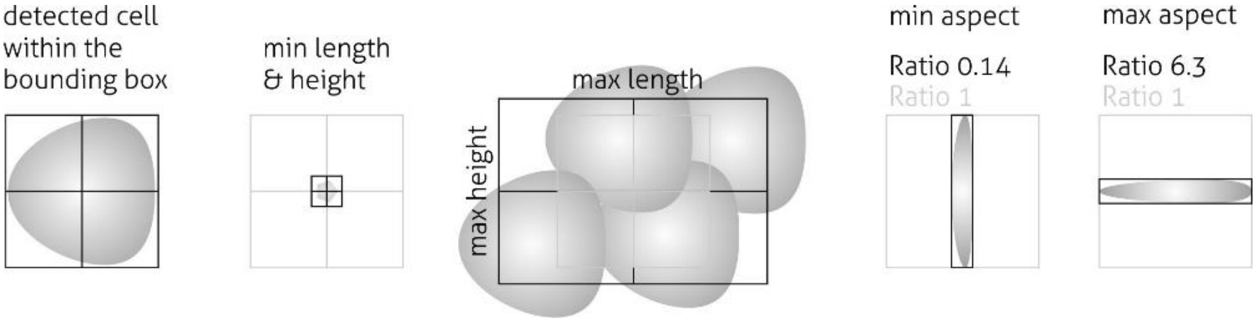


Fig. 2.3: Illustration of the event bounding box and its use cases. From left to right: definition of the bounding box, exclusion of small objects (e.g. debris) via the bounding box size, exclusion of clusters via the bounding box size, exclusion of objects elongated perpendicular to the channel axis, exclusion of objects elongated along the channel axis.

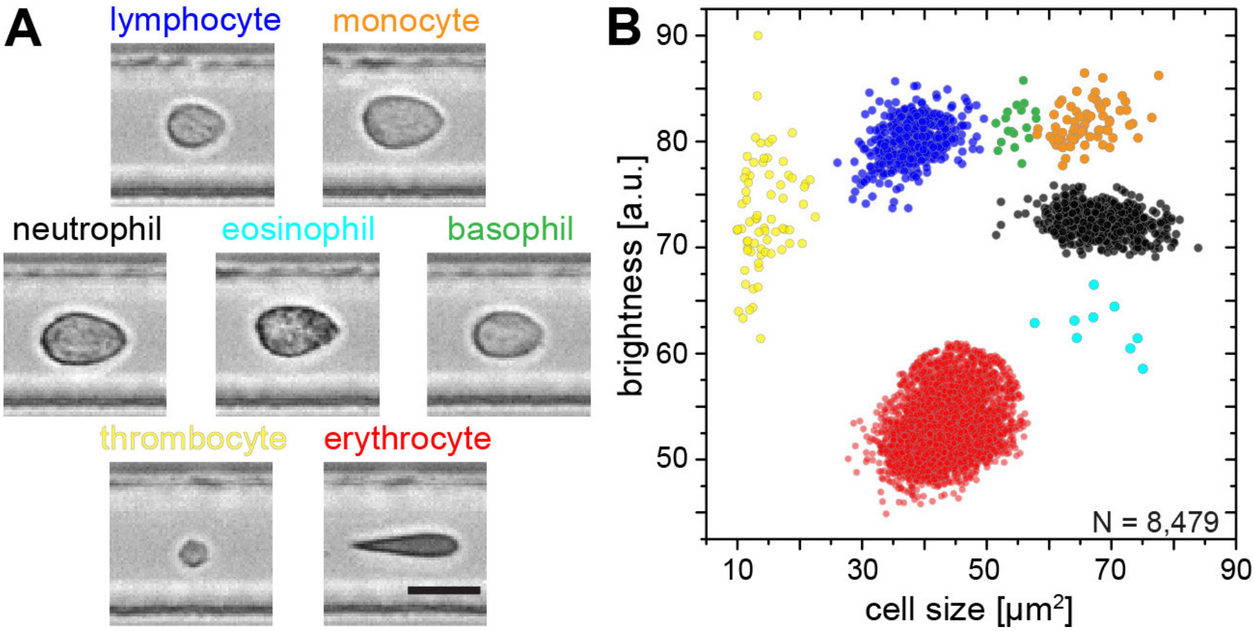


Fig. 2.4: Blood-classification with event brightness and cell size. (A) Representative images of blood cell types acquired. Scale bar is 10 µm. (B) Brightness versus cell size (area) scatter plot with cell types labeled according to the color scheme in (A). Figure and caption adapted from Toepfner et al.[Page 5, 1].

2.2.4 Deformation and elasticity

The deformation describes how much an event image deviates from a circular shape. It is defined via the circularity:

$$\begin{aligned} \text{deformation} &= 1 - \text{circularity} \\ &= 1 - 2\sqrt{\pi A}/l \end{aligned}$$

with the projected area A and the contour length of the convex hull of the event image l. The contour length is computed from the *convex* hull to prevent an overestimation due to irregular, non-convex event shapes. It has been shown that the knowledge of deformation and area allows to derive a value for elasticity in RT-DC [MOG+15] [MMM+17]. As a convenient measure for elasticity, isoelasticity lines are often employed to visualize stiffness.

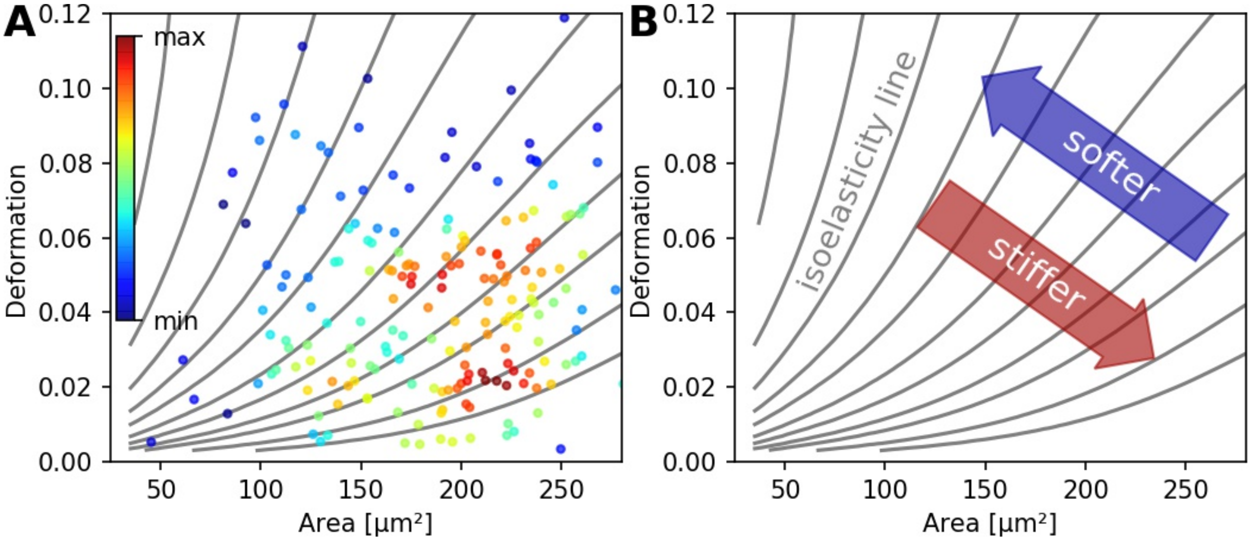


Fig. 2.5: (A) Typical deformation versus cell size scatter plot. The color scale indicates event density. (B) Isoelasticity lines derived from numerical simulations indicate trends in stiffness.

Note that it is also possible to directly *access the Young's modulus in DCscope*.

2.2.5 Fluorescence

Real-time fluorescence and deformability cytometry (RT-FDC) records, in addition to the event images, the fluorescence signal of each event [RPJ+18]. The raw fluorescence data consists of the one-dimensional fluorescence intensity trace from which features such as peak fluorescence or peak width can be computed. For more advanced applications, RT-FDC also supports multiple fluorescence channels.

2.2.6 Inertia ratio

The inertia ratio is the ratio of the second order central moments along x and y computed for the event contour. Thus, the inertia ratio is a measure of deformation. In contrast to deformation (which is computed from the contour), inertia ratio is based on the integral over the cell area and thus less prone to noise. In comparison to deformation, the inertia ratio has a low correlation to porosity. DCscope also allows to compute the principal inertia ratio which is the maximal inertia ratio that can be obtained by rotating the contour. Thus, the principal inertia ratio is rotation-invariant which makes it applicable to reservoir measurements where e.g. cells are not aligned with the channel. To quantify the alignment of the measured objects with the measurement channel, DCscope can additionally quantify the tilt of the contour relative to the channel axis.

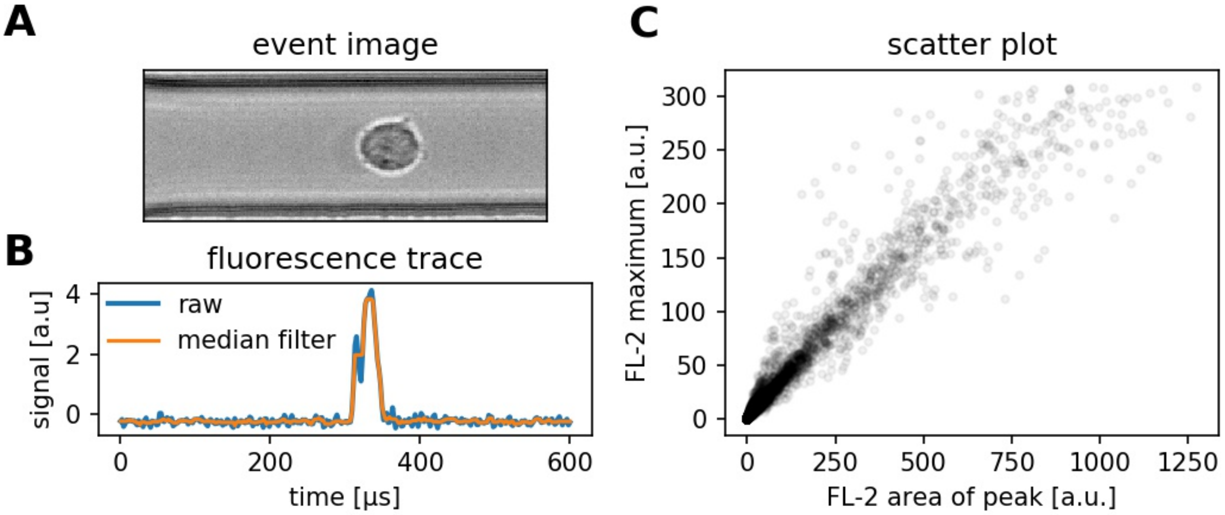


Fig. 2.6: (A) Event brightfield image. (B) Fluorescence trace of the event. The raw fluorescence data and the fluorescence data filtered with a rolling median filter (from which the parameters are computed) is shown. (C) Scatter plot of two parameters extracted from the fluorescence trace.

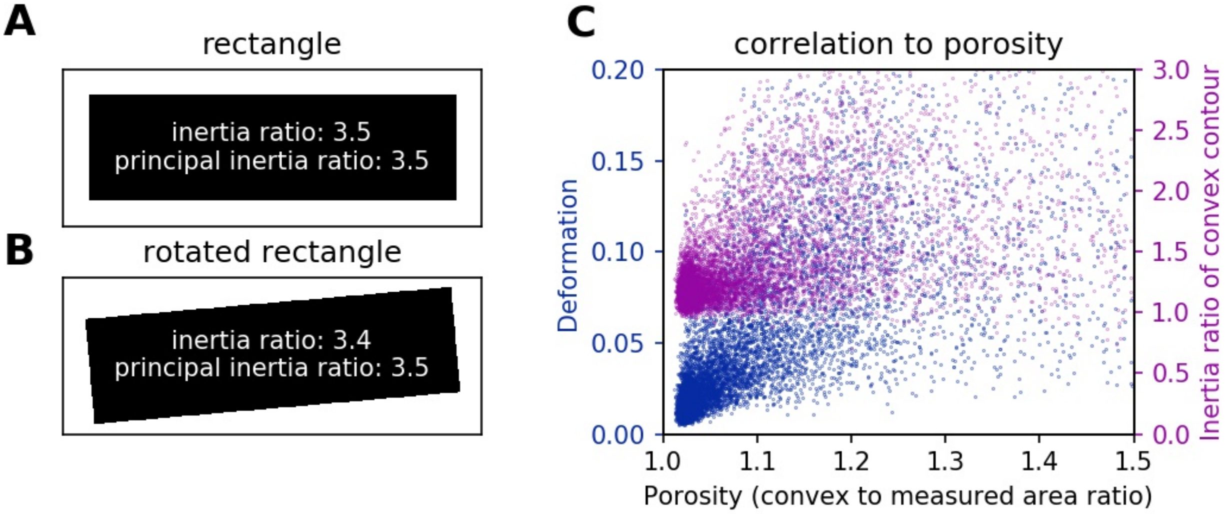


Fig. 2.7: (A) For a rectangle that is aligned with the coordinate axes, the inertia ratio and the principal inertia ratio are identical. (B) If the rectangle is rotated, the inertia ratio changes, but the principal inertia ratio does not. (C) Comparison between deformation and inertia ratio when plotted against porosity for an exemplary RT-DC experiment. Deformation exhibits a higher correlation to porisity.

2.2.7 Volume

DCscope can compute the volume from the event contour under the assumption of rotational symmetry, i.e. it is assumed that the projection of the cell volume onto the detector plane does not change when the cell is rotated, with a rotational axis parallel to the flow direction. The computation of the volume is based on a full rotation of the upper and the lower halves of the contour from which the average is then used [HWT02]. Volume has the advantage to be less correlated to deformation when compared to the projected area and it is therefore a better measure of cell size in the channel.

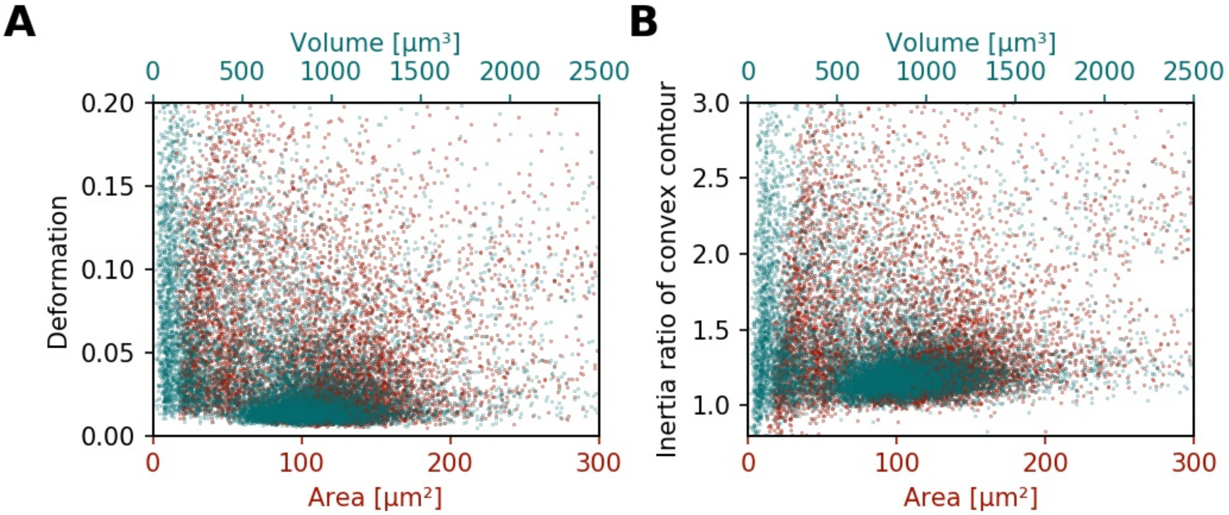


Fig. 2.8: (A) Deformation versus area (red) and volume (cyan) for an exemplary RT-DC dataset. There is a correlation between area and deformation, at least for small (<50µm^2) areas. (B) Inertia ratio versus area and volume. A correlation between inertia ratio and area is visible as well.

CHAPTER

THREE

USER INTERFACE

3.1 Terminology

DCscope introduces several terms in the user interface that also play a role in data analysis and are consequently used in the entire code base.

basin

A source of feature data stored in datasets other than the one opened by the user. For an overview on basins, please take a look at the dclab documentation. When exporting data from DCscope, the original dataset can be referenced via basins, allowing you to reduce data redundancy.

block matrix

The *Block Matrix* is a visualization of the analysis pipeline. It is divided into data matrix (filters) and plot matrix (plots).

dataslot or slot

A slot holds all information about a measurement: the path to the .rtdc file, fluorescence labels, display color, etc. (see the *Dataset* tab in the *Analysis View*).

filter

A filter is a set of filtering options (box filter, polygon filter, downsampling, etc.) that can be applied to a slot (see the *Filter* tab in the *Analysis View*). Filters can be exported and imported again in DCscope (.sof file format).

filter ray

A filter ray is a list of filters that can be applied to a slot. In DCscope, each row in the *Block Matrix* contains one filter ray.

pipeline

A pipeline consists of all filters, slots, plots, and the filter rays (the filter selection) applied to each slot. A DCscope session file (.so2) stores all information necessary to rebuild a pipeline.

plot

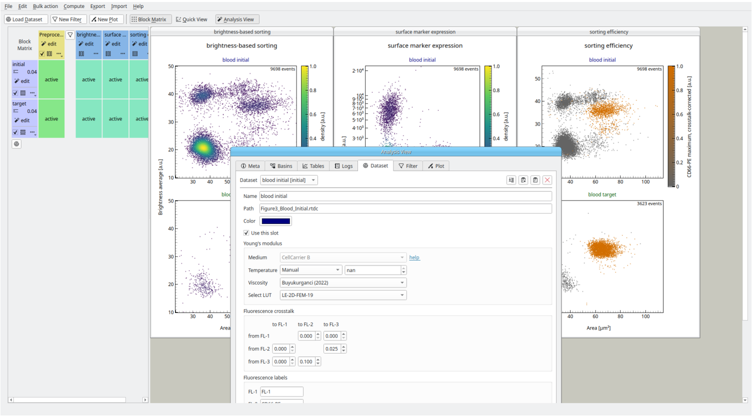
A plot is a user-defined visualization of a slot/filter ray combination.

3.2 Basic usage

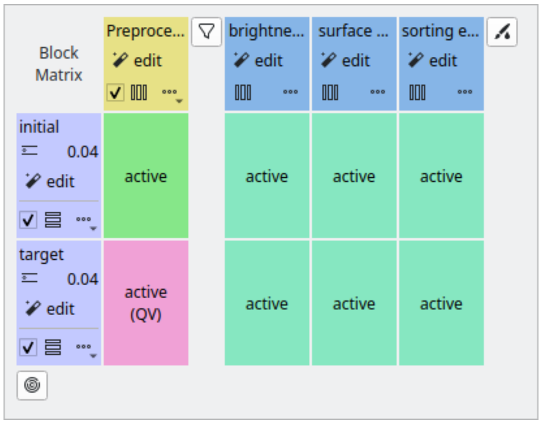
The user interface is split into several parts: the menu bar and the tool bar at the top, the Block Matrix on the left, and the Workspace on the right (example data taken from [NUH+20], [NUH+19]).

3.2.1 Menu bar and tool bar

The menu bar is used for session management (File menu) and makes additional functionalities available, such as batch processing, data statistics, data export, plot export, or filter import/export. The tool bar contains shortcuts for adding new datasets, filters, or plots (left part) and allows to hide/show the *Block Matrix* as well as the *Quick View* and *Analysis View* windows (right part).



3.2.2 Block Matrix



The Block Matrix gives an overview of the current analysis pipeline. Each row represents a dataset (purple). The columns represent either filters (yellow) or plots (blue) of your pipeline. You can change the order of datasets via the *Edit|Change dataset order* menu bar entry.

You can perform dataset operations in the purple rectangular area at the beginning of each row: To modify a dataset, click on the *edit* button. You can duplicate, insert anew (unmodified), or remove datasets using the dropdown menu. You can also exclude a dataset from an analysis via the check box.

Filters can also be modified, copied, removed and disabled. By default, all filters are disabled when they are created. To apply a filter to a dataset, click on the corresponding matrix element. The element changes its color from gray (incactive) to green (active). In DCscope, all filters that are applied to a dataset are called a **filter ray**. In the above example, the filter ray only consists of a single filter for each dataset. Filter rays may be different for each dataset.

By holding down the *Shift* key while clicking on a matrix element, you can activate the *Quick View* for the specific dataset (with filters applied up until the selected column). The block matrix element is then colored pink.

To add a plot, click on the *New Plot* button in the tool bar. This adds a plot column with a blue header to the Block Matrix and creates an empty plot window. You can add datasets to your plot by clicking on the corresponding matrix elements. In the above example, both datasets are being used in all three plots.

The modification of datasets, filters, and plots is discussed below.

3.2.3 Workspace

The Workspace is designed as an infinite scrollable area and contains all plot windows as well as the *Quick View* and Analysis View windows.

3.2.4 Analysis View

The analysis view consists of seven tabs that allow you to inspect the datasets loaded and to perform filtering and plotting actions.

Meta

The **Meta** tab displays all metadata of the selected dataset that are stored in the original .rtdc file. Here you can check and compare measurement and postprocessing parameters.

Basins

The **Basins** tab yields insight into the basins that are loaded for a dataset. Find out more about basins in the dclab documentation.

Tables

The **Tables** tab allows you to visualize additional telemetry recorded during the measurement. You can use it for quality control and to identify reasons for temporal trends within a dataset.

Logs

The **Logs** tab gives access to all logs stored in a dataset.

Dataset

The **Dataset** tab allows to specify additional metadata, such as unique colors used for plotting and additional metadata for computing the Young's modulus or correcting for fluorescence cross-talk. It also allows to specify fluorescence channel labels that will then be used for labeling the axes of plots.

Filter

The **Filter** tab is used to modify the filters of the pipeline. New box filters can be added by selecting *Choose box filters...*. Polygon filters are created in the *Quick View* window.

Plot

The **Plot** tab allows to specify all plotting parameters. Please take special note of the *Division* option in the *Layout* section (defines the arrangement of the subplots) and the *Marker hue* option in the *Scatter plot* section (allows you the specify whether the scatter data points are colored according to a kernel density estimate (KDE), another feature dimension, or the dataset color specified in the *Dataset* tab). In this example, contour plots are not used.

3.2.5 Quick View

The Quick View is meant for dataset exploration. It consists of a scatter plot on the left (left click for panning and right-click for zooming) and a set of tool panels that are accessible via the corresponding tool buttons on the right.

Use the **Plot** panel to define all plot parameters. It also displays common statistics of the two features plotted. The drop down menus for the X and Y axes list the available features for the current dataset. The background color for each of the features is an indicator for the feature availability:

- *green*: The feature data are present in the current dataset or are already computed.

Dataset: example data for docs

Experiment

Sample name: example data for docs
Index of measurement run: 9
Number of recorded events: 47
Date of measurement : 2024-12-16
Start time of measurement : 11:07:15
Unique measurement identifier: 2024-rcs-f48f-4453-b2da-2adc08db444d_dcn-1a01d20-c0az
Start of measurement in unix time: 1734343635.00 s

Pipeline

Generation ID: 11
Data ID: hdf:p=0.2668^i=0
Background ID: sparsemed:k=200^s=1^t=0^f=0.8^o=1
Segmenter ID: legacy:t=-6^bl=0^bi=5^d=1:cle=1^f=1^clo=0
Feature extractor ID: legacy:b=1^h=1^v=1
Gating ID: norm:o=0^s=10
Hash: 1d01d203e2d80e9c04060d67fcbdadbc
Event mapping from original dataset: 0
Event yield: 1929466

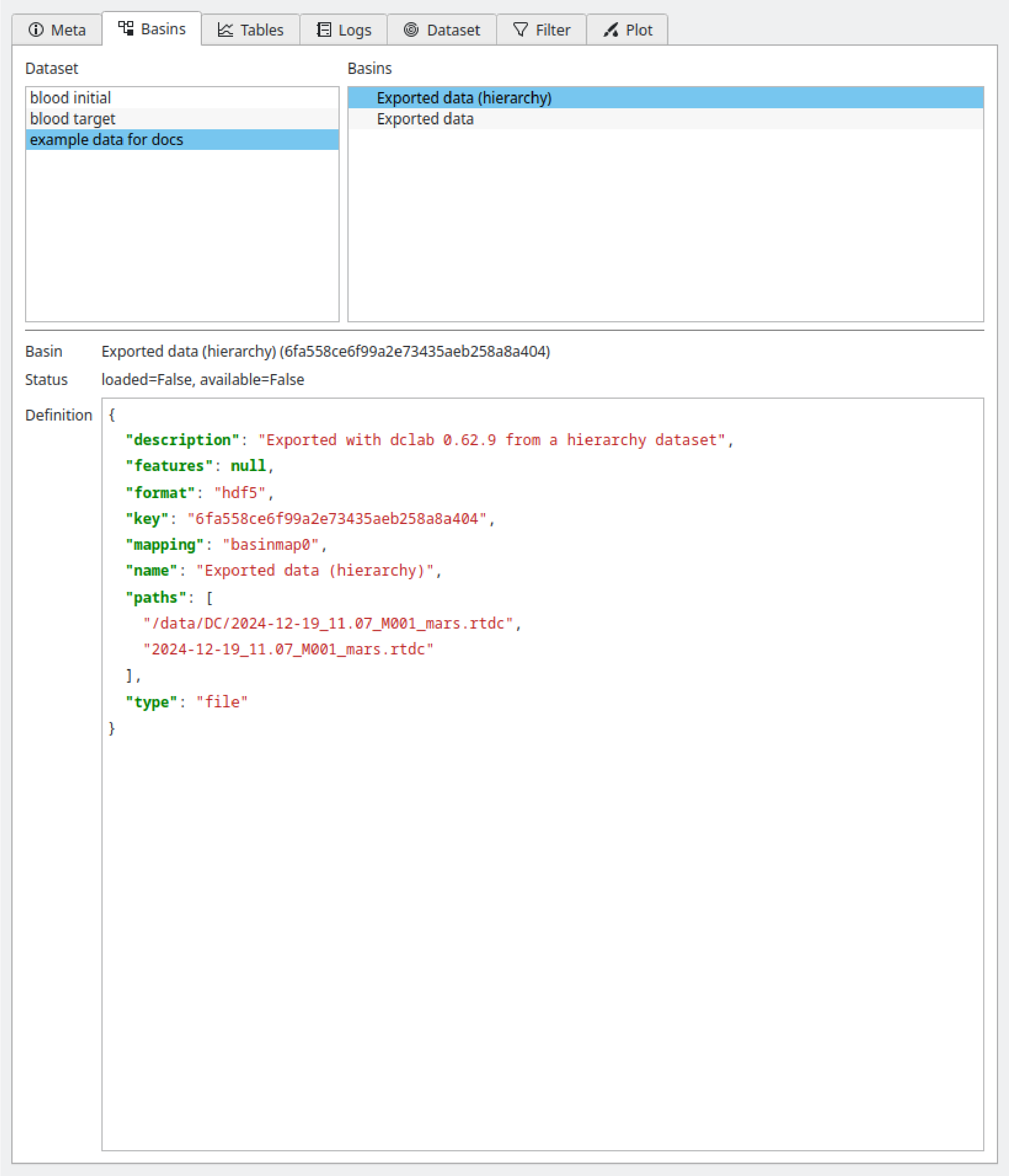
User

dcevent ppid background: sparsemed:k=200^s=1^t=0^f=0.8^o=1
dcevent ppid data: hdf:p=0.2668^i=0
dcevent ppid feature: legacy:b=1^h=1^v=1
dcevent ppid gate: norm:o=0^s=10
dcevent ppid generation: 11C
dcevent ppid hash: 4825a5c3afde8e37911fee45f75d7ea7
dcevent ppid segmenter: legacy:t=-6^bl=0^bi=5^d=1:cle=1^f=1^clo=0

Setup

Medium used: 0.59% MC-PBS
Width of microfluidic channel: 20.00 µm
Imaged chip region: channel
Flow rate in channel: 0.0600 µL/s
Sample flow rate: 0.0150 µL/s
Sheath flow rate: 0.0450 µL/s
Unique setup identifier: RC-bdcc-e5ce-0256-a74d-win-8537

Fig. 3.1: Meta tab in the Analysis View.



Basin: Exported data (hierarchy) (6fa558ce6f99a2e73435aeb258a8a404)

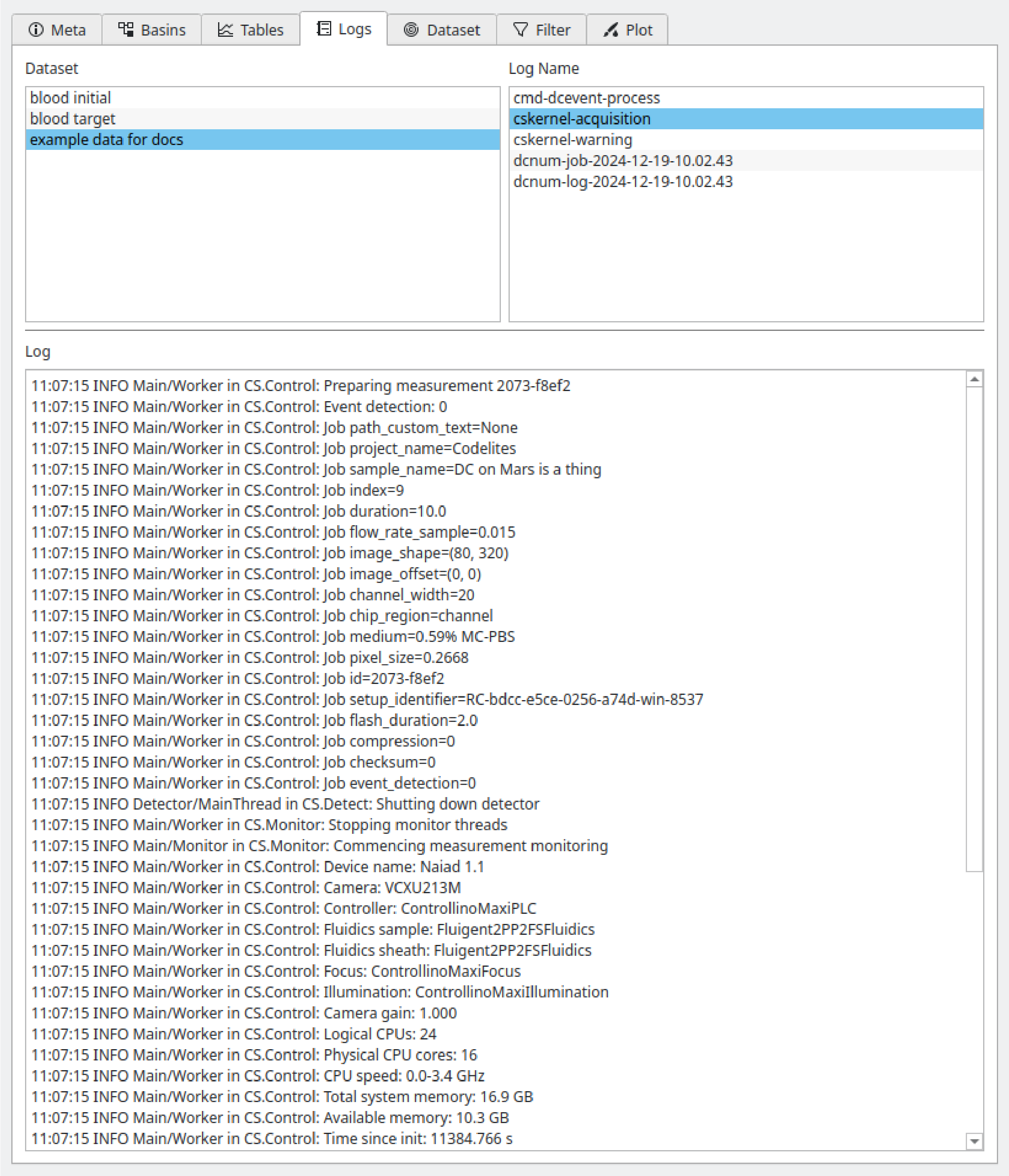
Status: loaded=False, available=False

Definition:

```
{
  "description": "Exported with dclab 0.62.9 from a hierarchy dataset",
  "features": null,
  "format": "hdf5",
  "key": "6fa558ce6f99a2e73435aeb258a8a404",
  "mapping": "basinmap0",
  "name": "Exported data (hierarchy)",
  "paths": [
    "/data/DC/2024-12-19_11.07_M001_mars.rtdc",
    "2024-12-19_11.07_M001_mars.rtdc"
  ],
  "type": "file"
}
```

Fig. 3.2: Basins tab in the Analysis View.

Fig. 3.3: Tables tab in the Analysis View.



Dataset

- blood initial
- blood target
- example data for docs

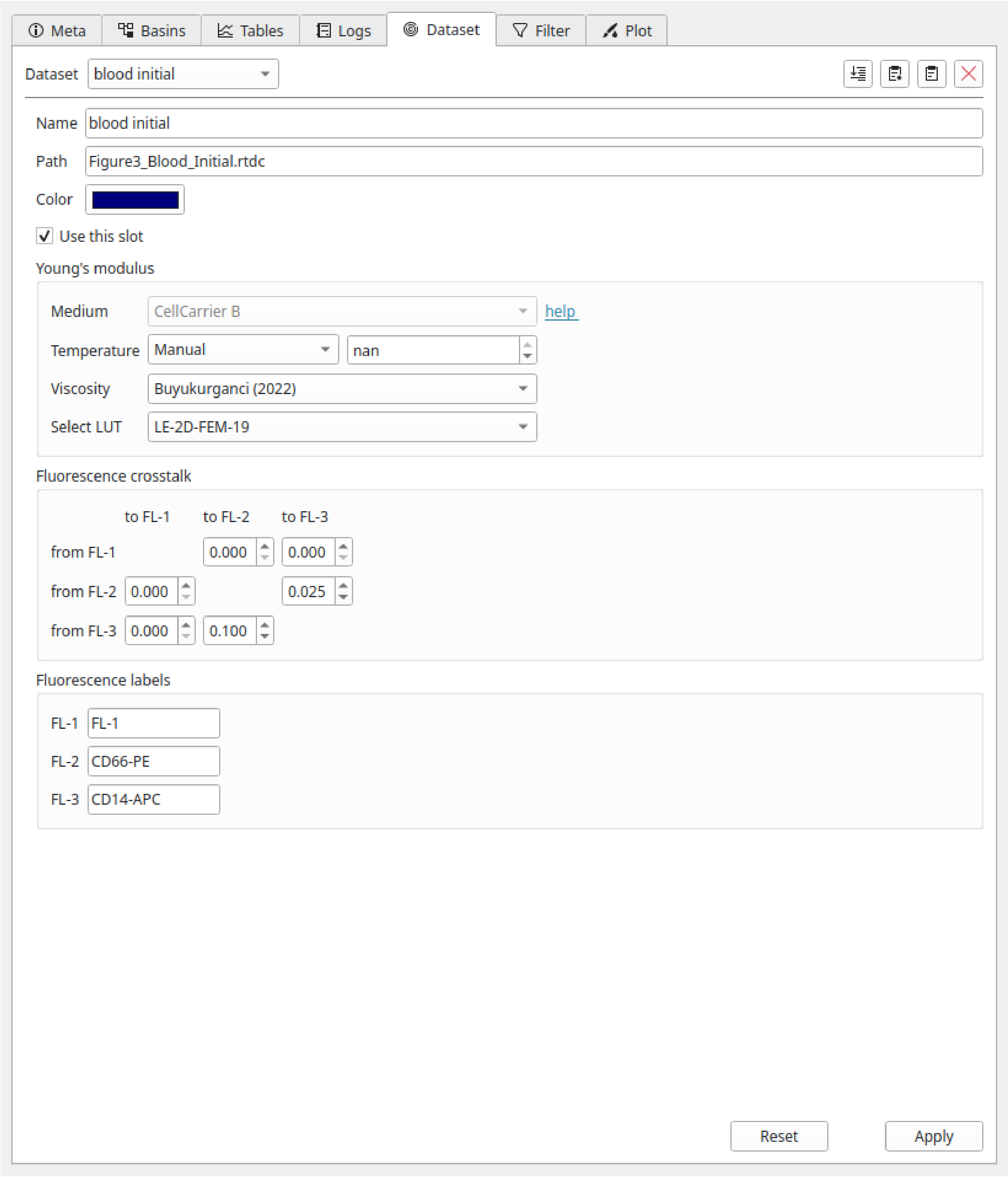
Log Name

- cmd-dcevent-process
- cskernel-acquisition
- cskernel-warning
- dcnum-job-2024-12-19-10.02.43
- dcnum-log-2024-12-19-10.02.43

Log

```
11:07:15 INFO Main/Worker in CS.Control: Preparing measurement 2073-f8ef2
11:07:15 INFO Main/Worker in CS.Control: Event detection: 0
11:07:15 INFO Main/Worker in CS.Control: Job path_custom_text=None
11:07:15 INFO Main/Worker in CS.Control: Job project_name=Codelites
11:07:15 INFO Main/Worker in CS.Control: Job sample_name=DC on Mars is a thing
11:07:15 INFO Main/Worker in CS.Control: Job index=9
11:07:15 INFO Main/Worker in CS.Control: Job duration=10.0
11:07:15 INFO Main/Worker in CS.Control: Job flow_rate_sample=0.015
11:07:15 INFO Main/Worker in CS.Control: Job image_shape=(80, 320)
11:07:15 INFO Main/Worker in CS.Control: Job image_offset=(0, 0)
11:07:15 INFO Main/Worker in CS.Control: Job channel_width=20
11:07:15 INFO Main/Worker in CS.Control: Job chip_region=channel
11:07:15 INFO Main/Worker in CS.Control: Job medium=0.59% MC-PBS
11:07:15 INFO Main/Worker in CS.Control: Job pixel_size=0.2668
11:07:15 INFO Main/Worker in CS.Control: Job id=2073-f8ef2
11:07:15 INFO Main/Worker in CS.Control: Job setup_identifier=RC-bdcc-e5ce-0256-a74d-win-8537
11:07:15 INFO Main/Worker in CS.Control: Job flash_duration=2.0
11:07:15 INFO Main/Worker in CS.Control: Job compression=0
11:07:15 INFO Main/Worker in CS.Control: Job checksum=0
11:07:15 INFO Main/Worker in CS.Control: Job event_detection=0
11:07:15 INFO Detector/MainThread in CS.Detect: Shutting down detector
11:07:15 INFO Main/Worker in CS.Monitor: Stopping monitor threads
11:07:15 INFO Main/Monitor in CS.Monitor: Commencing measurement monitoring
11:07:15 INFO Main/Worker in CS.Control: Device name: Naiad 1.1
11:07:15 INFO Main/Worker in CS.Control: Camera: VCXU213M
11:07:15 INFO Main/Worker in CS.Control: Controller: ControllinoMaxiPLC
11:07:15 INFO Main/Worker in CS.Control: Fluidics sample: Fluigent2PP2FSFluidics
11:07:15 INFO Main/Worker in CS.Control: Fluidics sheath: Fluigent2PP2FSFluidics
11:07:15 INFO Main/Worker in CS.Control: Focus: ControllinoMaxiFocus
11:07:15 INFO Main/Worker in CS.Control: Illumination: ControllinoMaxiIllumination
11:07:15 INFO Main/Worker in CS.Control: Camera gain: 1.000
11:07:15 INFO Main/Worker in CS.Control: Logical CPUs: 24
11:07:15 INFO Main/Worker in CS.Control: Physical CPU cores: 16
11:07:15 INFO Main/Worker in CS.Control: CPU speed: 0.0-3.4 GHz
11:07:15 INFO Main/Worker in CS.Control: Total system memory: 16.9 GB
11:07:15 INFO Main/Worker in CS.Control: Available memory: 10.3 GB
11:07:15 INFO Main/Worker in CS.Control: Time since init: 11384.766 s
```

Fig. 3.4: Logs tab in the Analysis View.



Dataset: blood initial

Name: blood initial

Path: Figure3_Blood_Initial.rtdc

Color

☑ Use this slot

Young's modulus

Medium: CellCarrier B help

Temperature: Manual nan

Viscosity: Buyukurganci (2022)

Select LUT: LE-2D-FEM-19

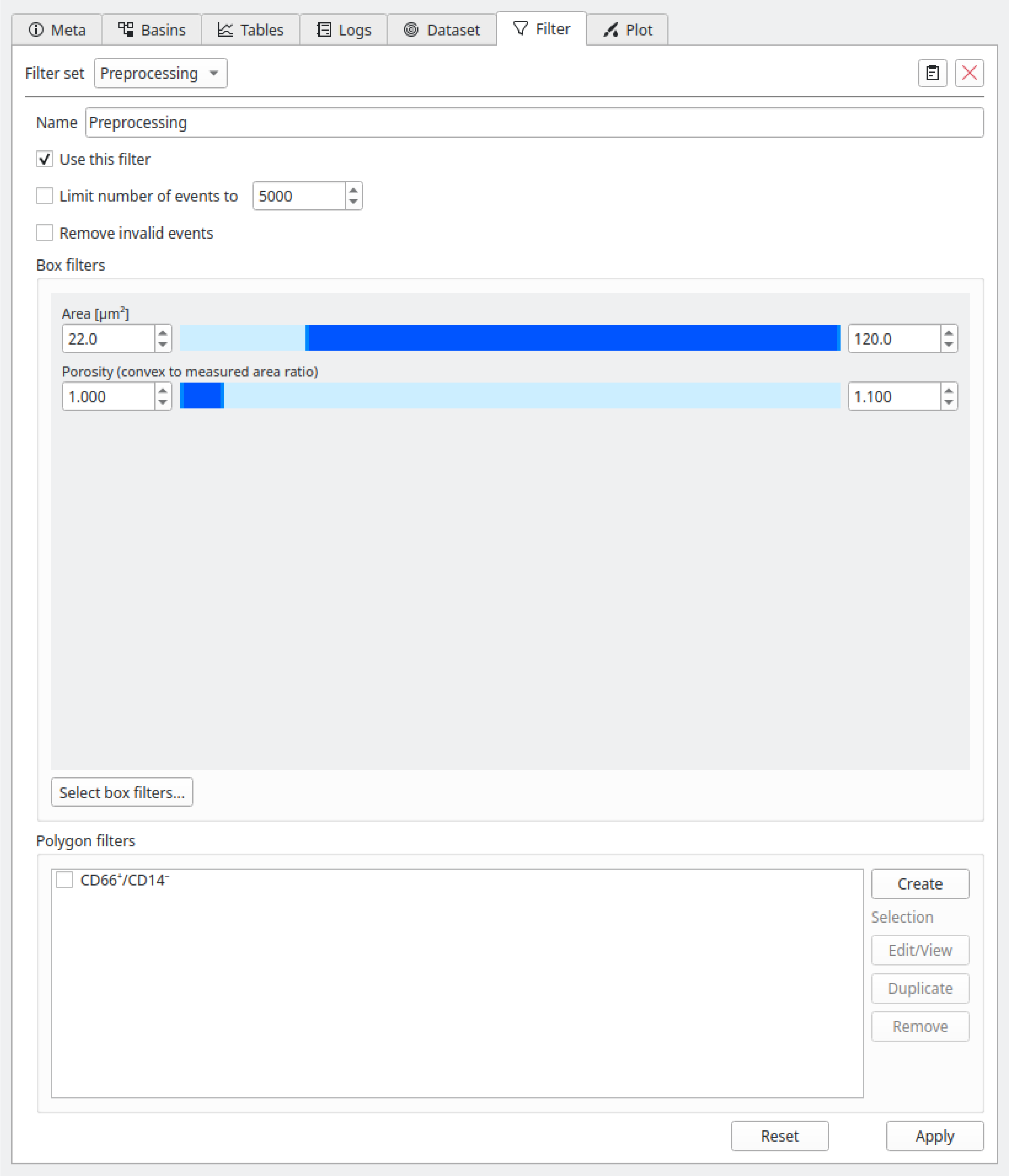
Fluorescence crosstalk

	to FL-1	to FL-2	to FL-3
from FL-1		0.000	0.000
from FL-2	0.000		0.025
from FL-3	0.000	0.100	

Fluorescence labels

FL-1	FL-1
FL-2	CD66-PE
FL-3	CD14-APC

Fig. 3.5: Dataset tab in the Analysis View.



Filter set: Preprocessing

Name: Preprocessing

☑ Use this filter

☐ Limit number of events to 5000

☐ Remove invalid events

Box filters

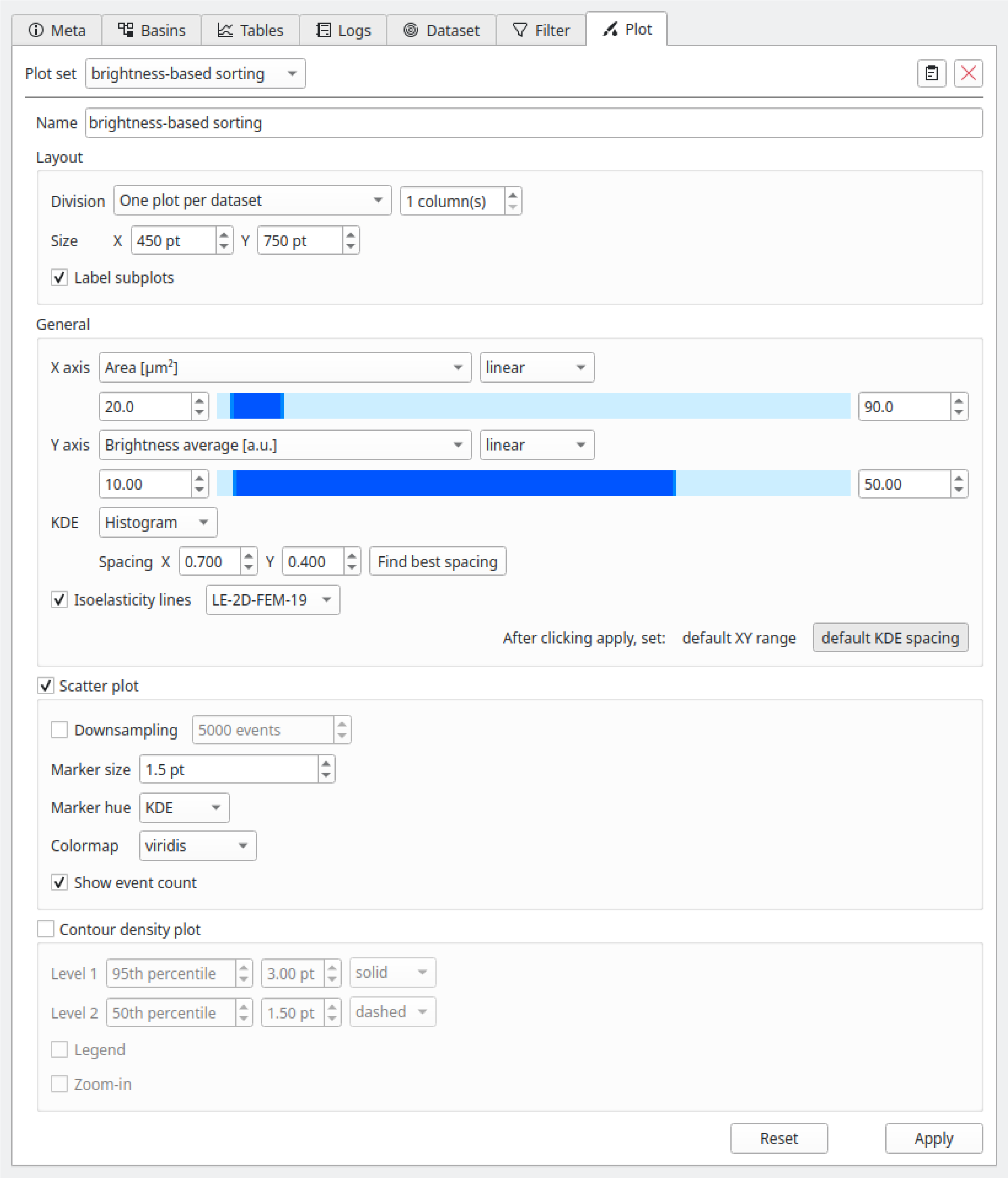
Area [µm^2]: 22.0 – 120.0

Porosity (convex to measured area ratio): 1.000 – 1.100

Polygon filters

☐ CD66$^+$/CD14$^-$

Fig. 3.6: Filter tab in the Analysis View.



Plot set brightness-based sorting

Name brightness-based sorting

Layout

Division One plot per dataset 1 column(s)

Size X 450 pt Y 750 pt

☑ Label subplots

General

X axis Area [μm^2] linear

20.0 90.0

Y axis Brightness average [a.u.] linear

10.00 50.00

KDE Histogram

Spacing X 0.700 Y 0.400

☑ Isoelasticity lines LE-2D-FEM-19

After clicking apply, set: default XY range default KDE spacing

☑ Scatter plot

☐ Downsampling 5000 events

Marker size 1.5 pt

Marker hue KDE

Colormap viridis

☑ Show event count

☐ Contour density plot

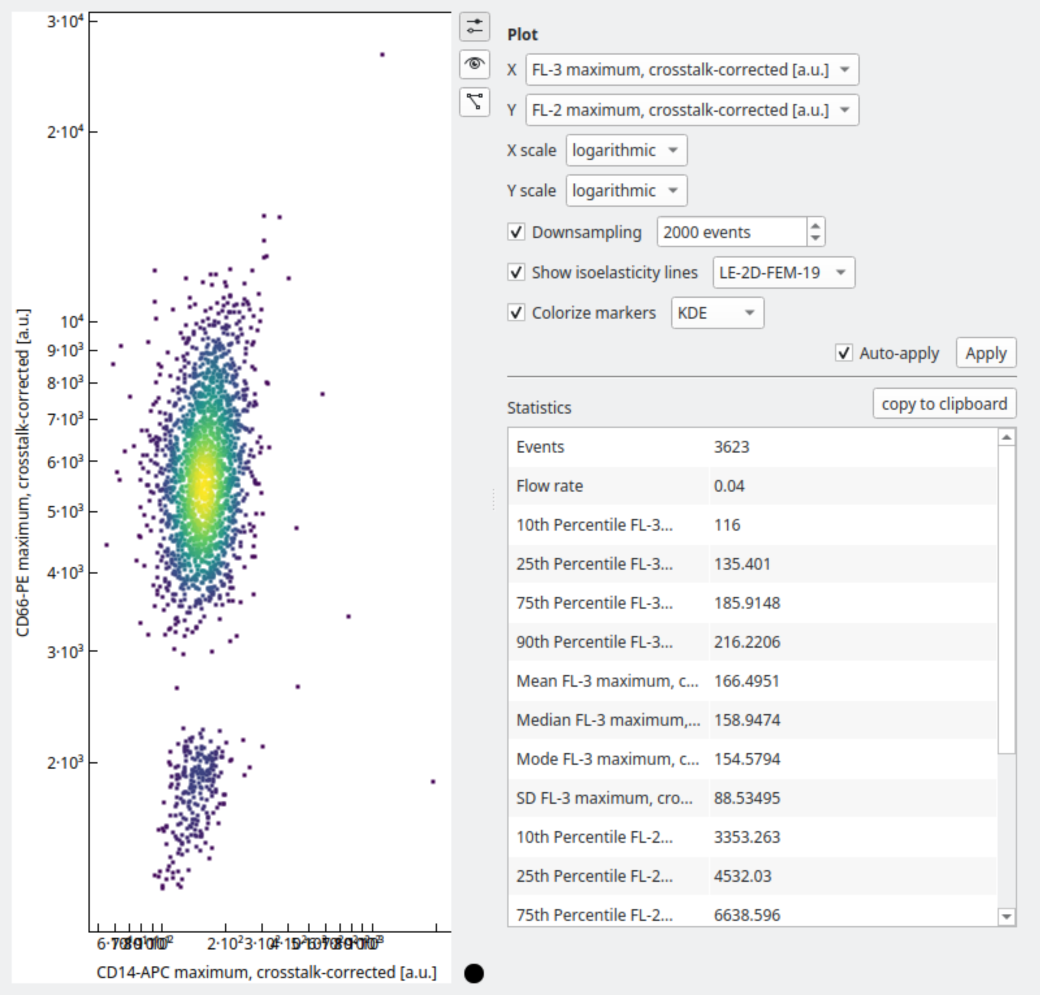
Level 1 95th percentile 3.00 pt solid

Level 2 50th percentile 1.50 pt dashed

☐ Legend

☐ Zoom-in

- *blue*: The feature data are part of another dataset, a basin, which may at a remote location (e.g. on DCOR) or on the local file system.
- *orange*: The feature data must be computed before it can be displayed. The feature is an ancillary feature.



Plot

X FL-3 maximum, crosstalk-corrected [a.u.]

Y FL-2 maximum, crosstalk-corrected [a.u.]

X scale logarithmic

Y scale logarithmic

☑ Downsampling 2000 events

☑ Show isoelasticity lines LE-2D-FEM-19

☑ Colorize markers KDE

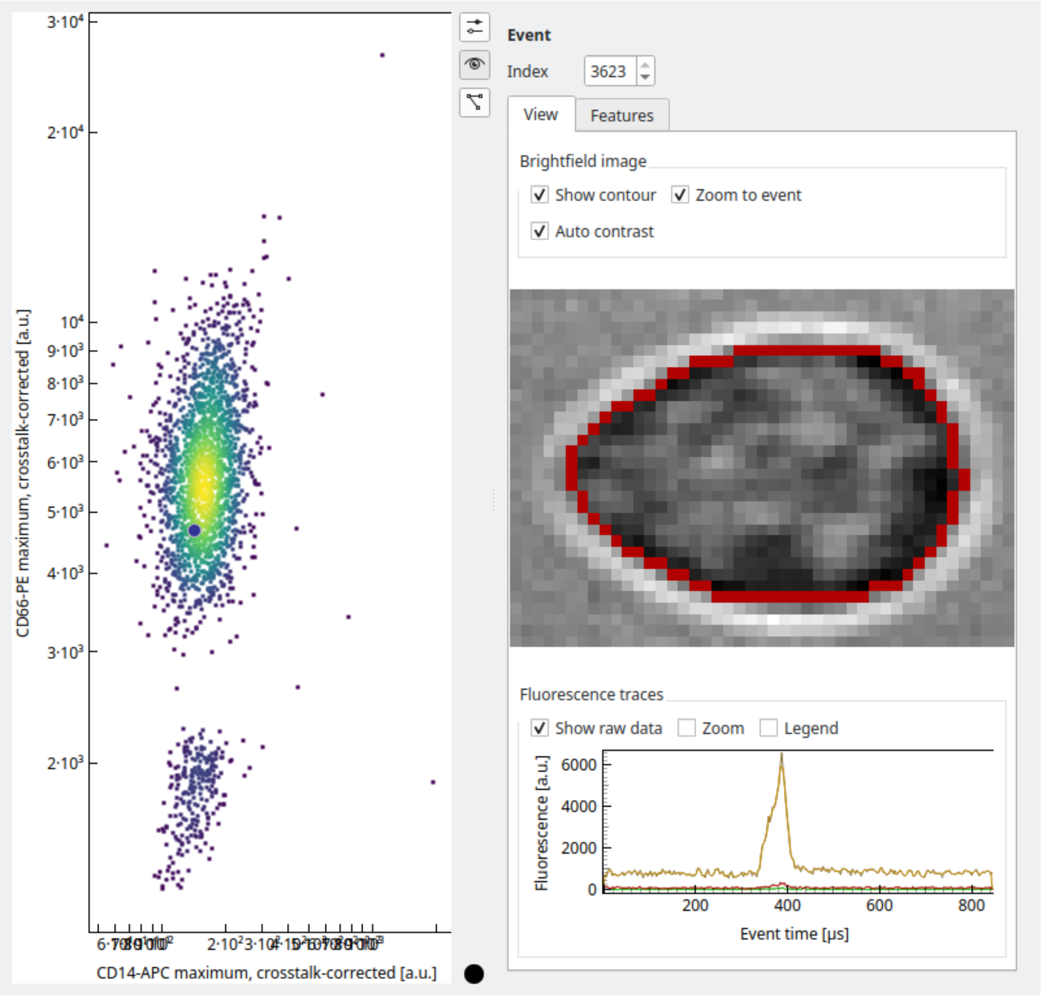
☑ Auto-apply Apply

Statistics

Events	3623
Flow rate	0.04
10th Percentile FL-3...	116
25th Percentile FL-3...	135.401
75th Percentile FL-3...	185.9148
90th Percentile FL-3...	216.2206
Mean FL-3 maximum, c...	166.4951
Median FL-3 maximum,...	158.9474
Mode FL-3 maximum, c...	154.5794
SD FL-3 maximum, cro...	88.53495
10th Percentile FL-2...	3353.263
25th Percentile FL-2...	4532.03
75th Percentile FL-2...	6638.596

The **Event** panel displays all parameters of an individual event. You can select single events by clicking in the scatter plot or by scrolling through the *Index* spin control. If available, the event image is shown alongside the fluorescence trace of the event. All features of the event are listed in a separate tab.

The **Polygon Filter** panel allows you to create and modify polygon filters. When the panel is active you can move the mouse pointer across the scatter plot and the image of the event closest to the mouse pointer is displayed.



Event

Index 3623

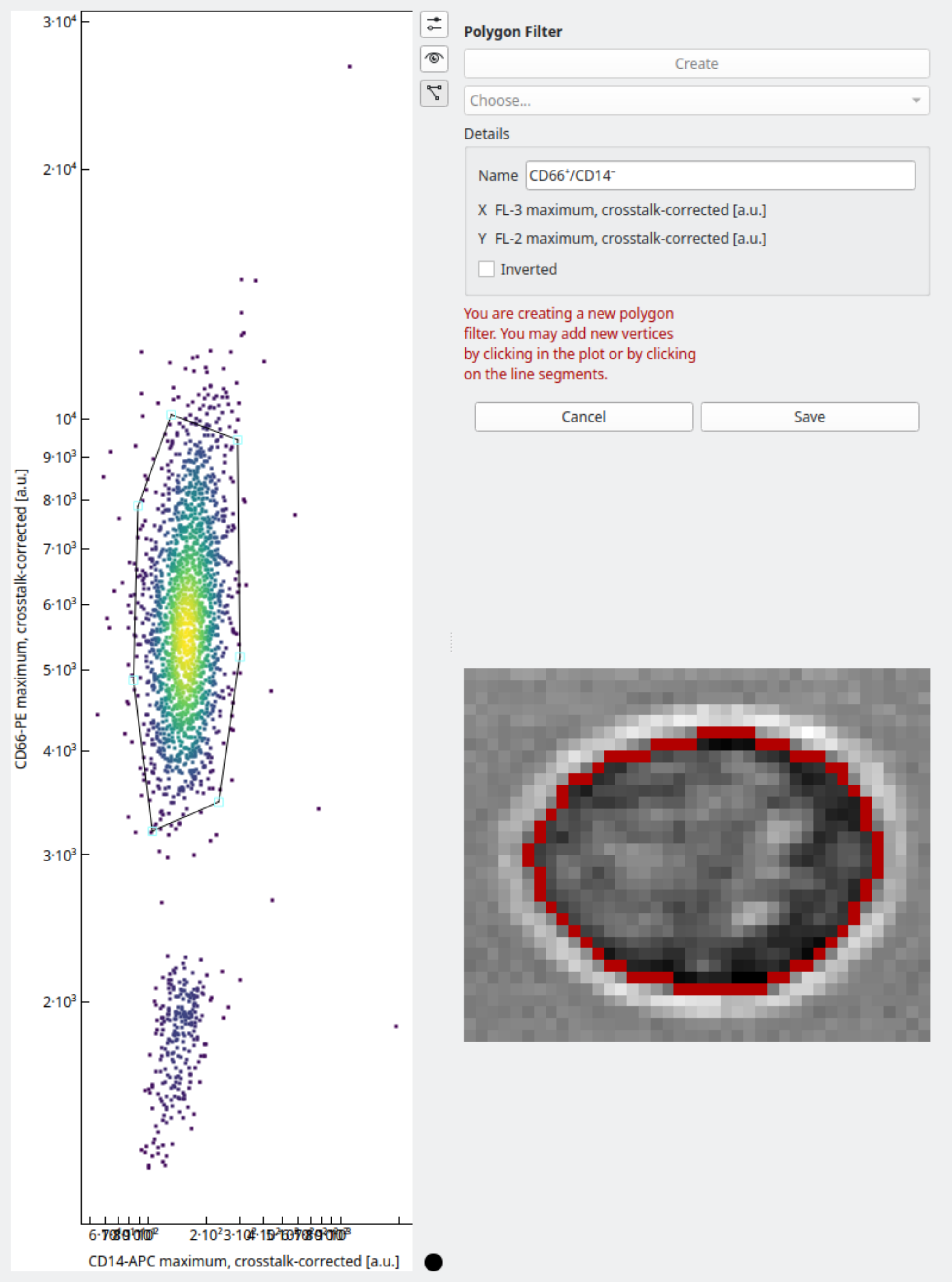
Brightfield image

☑ Show contour ☑ Zoom to event

☑ Auto contrast

Fluorescence traces

☑ Show raw data ☐ Zoom ☐ Legend



Name CD66$^+$/CD14$^-$

X FL-3 maximum, crosstalk-corrected [a.u.]

Y FL-2 maximum, crosstalk-corrected [a.u.]

☐ Inverted

You are creating a new polygon filter. You may add new vertices by clicking in the plot or by clicking on the line segments.

CHAPTER

FOUR

QUICK GUIDES

4.1 Checking experimental data for consistency

- **Trends over time:**

  To verify that the sample "behaves" stationary during the measurement, it is helpful to plot a few selected features versus time. No systematic deviations should be observable, as in the figure below.

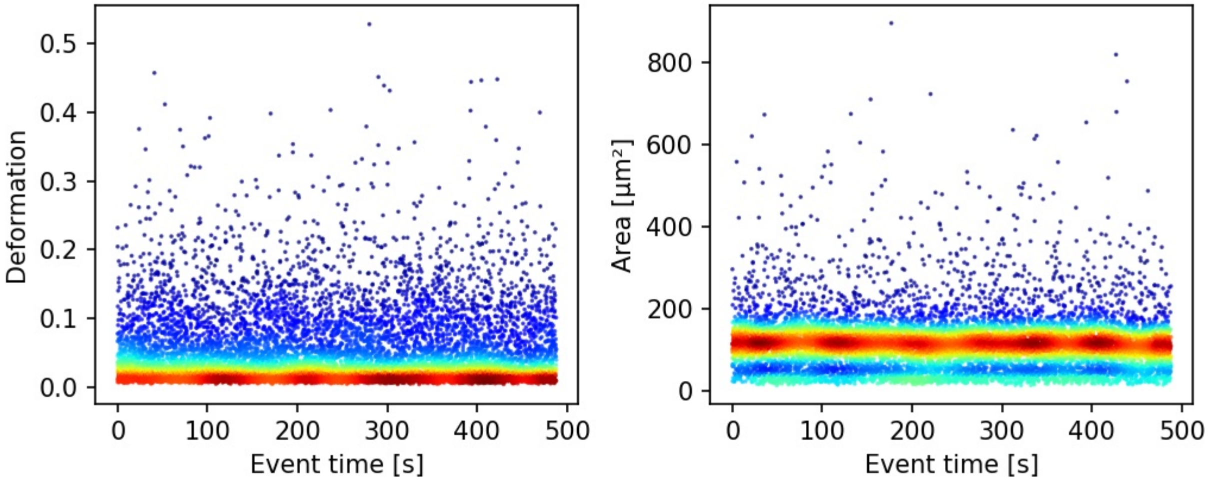


Fig. 4.1: Deformation and event area versus time. There is no systematic trend over time, indicating stable measurement conditions.

- **Countrate:**

  Trends over time do not directly visualize the count rate (measured events per second). The count rate is the slope of the event index versus event time plot and should remain constant.

4.2 Excluding invalid events

Filters can be used to exclude invalid events, such as cell debris, cell doublets or larger aggregates, or cells that are not intact, from an analysis.

- **Area:**

  The event area (the area defined by the event contour) can be used to filter cell aggregates (larger area), dead or degenerate cells, as well as cell debris (small area). Use the deformation versus area plot to identify exclusion candidates by clicking on an event and visualizing it in the event area.

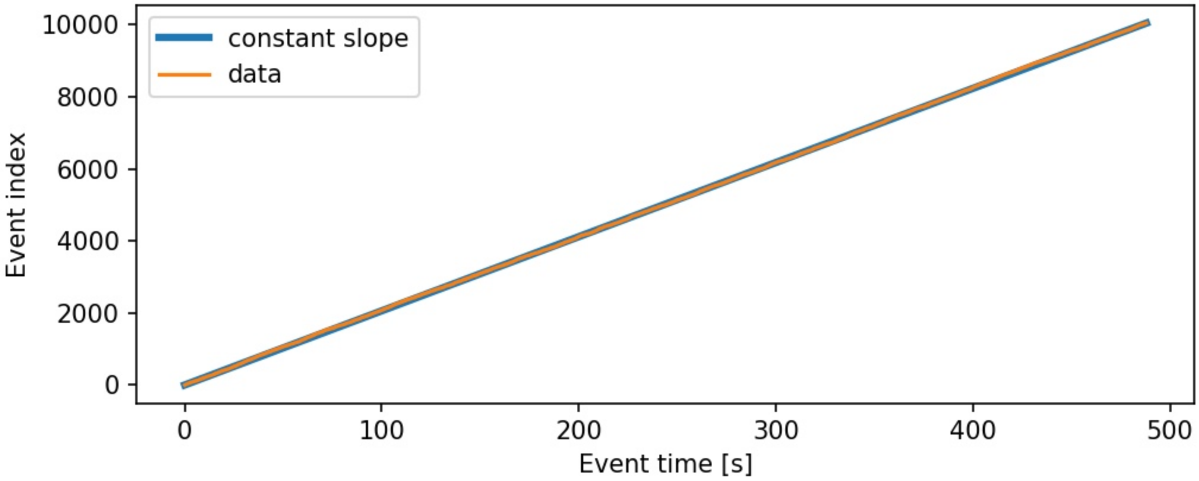


Fig. 4.2: The event index versus event time plot shows a constant slope, an indicator for stable measurement conditions.

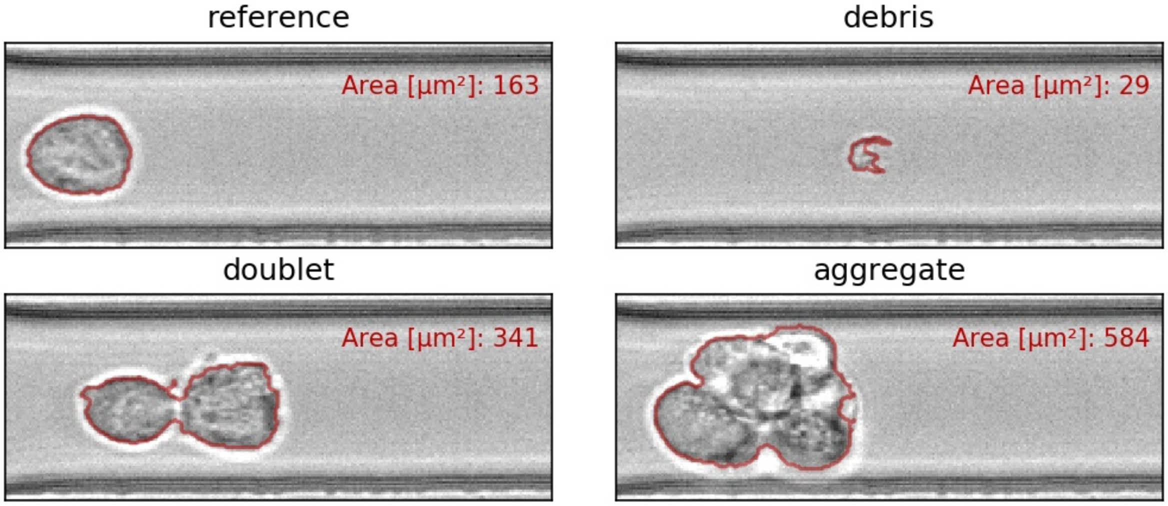


Fig. 4.3: Examples of events with small or large areas.

- **Aspect and inertia ratio:**

  The aspect ratio of the bounding box and the inertia ratio of the contour can be used to filter cell aggregates and otherwise invalid events. For instance, an aspect ratio below 1 (elongation perpendicular to the channel axis) is most-likely debris and can be excluded from the analysis. An inertia ratio below 1 also indicates invalid events.

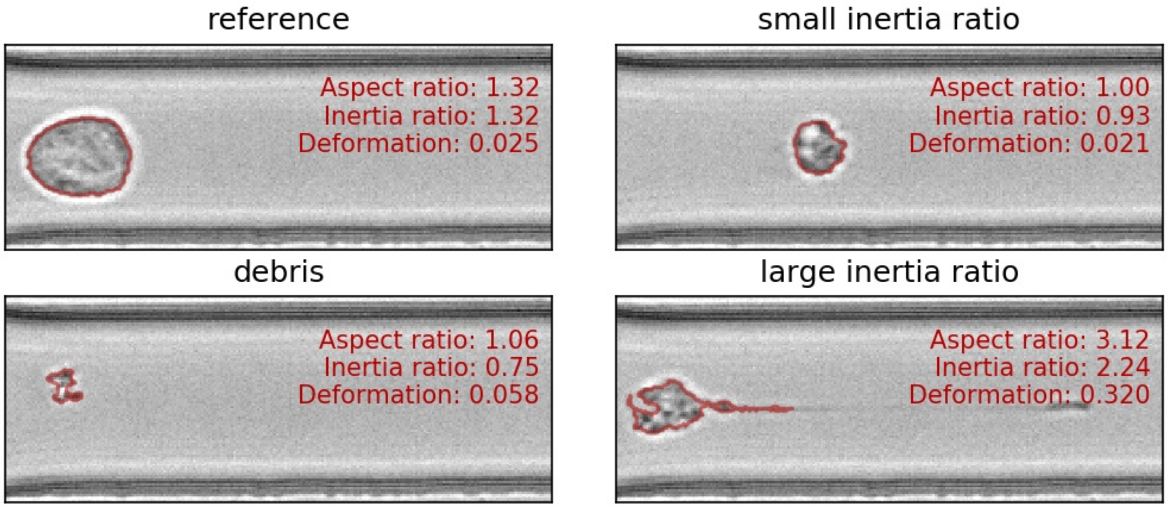


Fig. 4.4: Examples of events with various aspect and inertia ratios.

- **Porosity:**

  The porosity is the ratio between measured contour and the convex contour. Porosity is commonly used to remove events with non-physical contours, e.g. for cells, all events with a porosity above 1.05.

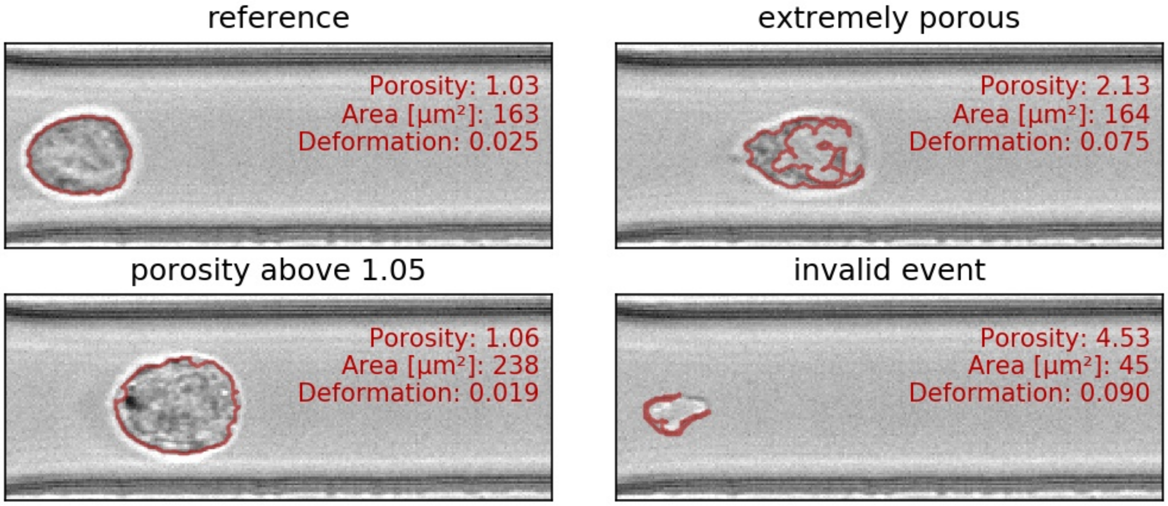
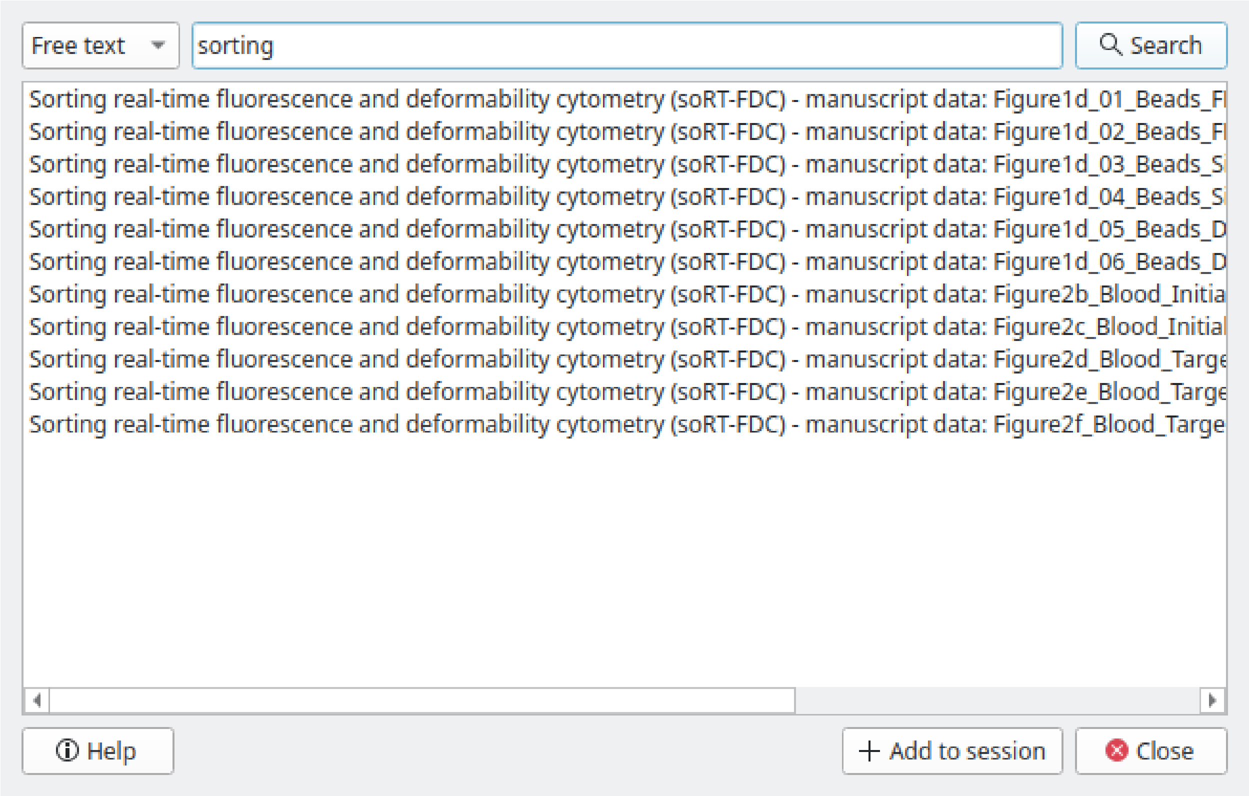


Fig. 4.5: Examples of events with various porosities.

4.3 Accessing DCOR data

The deformability cytometry open repository (DCOR) allows you to upload and access RT-DC datasets online (internet connection required). In DCscope, you can access parts of the dataset (e.g. just two features) without downloading the entire data file (which includes image, contour, and traces information). This saves disk space and time when accessing large datasets.



Free text

sorting

Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure1d_01_Beads_Fl
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure1d_02_Beads_Fl
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure1d_03_Beads_S
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure1d_04_Beads_S
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure1d_05_Beads_D
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure1d_06_Beads_D
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure2b_Blood_Initia
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure2c_Blood_Initial
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure2d_Blood_Targe
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure2e_Blood_Targe
Sorting real-time fluorescence and deformability cytometry (soRT-FDC) - manuscript data: Figure2f_Blood_Targe

Fig. 4.6: Dialog for loading DCOR data.

In the *File* menu, choose *Load DCOR data*. A dialog (Fig. 4.6) lets you search data in the DCOR repository.

Note

If you need to access private data, you have to specify your personal API key in the preferences dialog (via the *Edit* menu). There, you can also specify which server you wish to connect to. You can find your personal API Key in your user pofile page after you logged in (e.g. at https://dcor.mpl.mpg.de).

You can choose between a free text search (dataset and resource names, tags, etc.) and an identifier search (dataset and resource identifiers). After clicking the *Search* button, you may select the resources you wish to load into your sesssion in the list view below. Then, click *Add to session* to load the data.

4.4 Young's Modulus

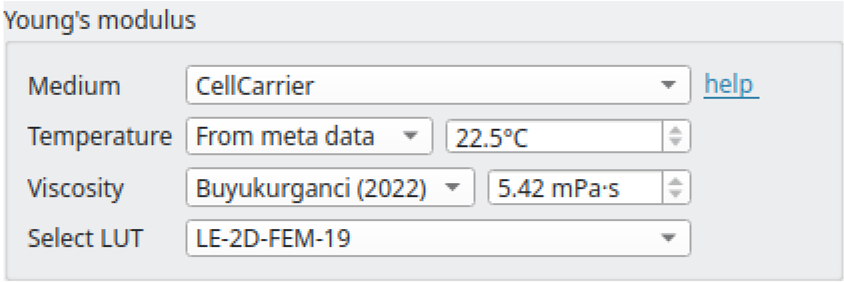
With DCscope, it is possible to translate the measured area and deformation to the Young's modulus based on numerical simulation for fully elastic spheres according to Mokbel et al. [MMM+17].

The theoretical background, including the computation of the viscosity for the MC-PBS media used in RT-DC, is discussed in more detail in the dclab docs.

Note

For reservoir data, the Young's modulus cannot be computed. The corresponding options are hidden.

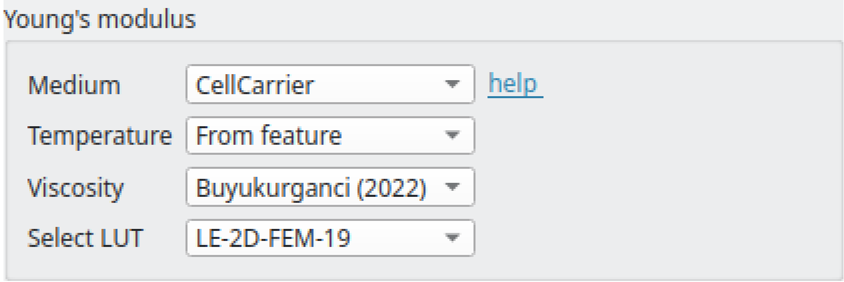
4.4.1 Parameter settings



Young's modulus

Medium	CellCarrier	
Temperature	From meta data	22.5°C
Viscosity	Buyukurganci (2022)	5.42 mPa·s
Select LUT	LE-2D-FEM-19	

Fig. 4.7: Temperature taken from the metadata.

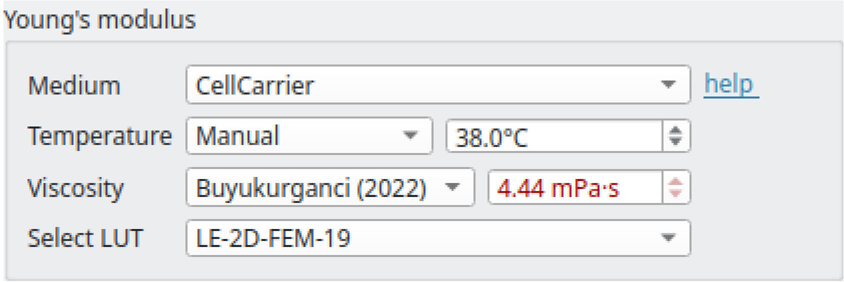


Young's modulus

Medium	CellCarrier
Temperature	From feature
Viscosity	Buyukurganci (2022)
Select LUT	LE-2D-FEM-19

Fig. 4.8: Use the temperature (temp) feature.

The parameters for computing the Young's modulus can be set in the *Dataset* tab of the *Analysis View*. The Young's modulus is computed using a precomputed look-up table and additionally depends on channel width, flow rate, pixel size (pixelation correction), and viscosity. For known media, such as CellCarrier, the viscosity can be derived from channel width, flow rate, and temperature. In some RT-DC setups, the chip temperature is recorded during the measurement. For instance, in Fig. 4.7, the average chip temperature of 22.5°C from the [setup] metadata section is used. The value of the resulting viscosity is shown below. If the chip temperature is recorded for each event, then the user may select the *From feature* option (Fig. 4.8). In this case, the Young's modulus is computed from the individual viscosities for each event.

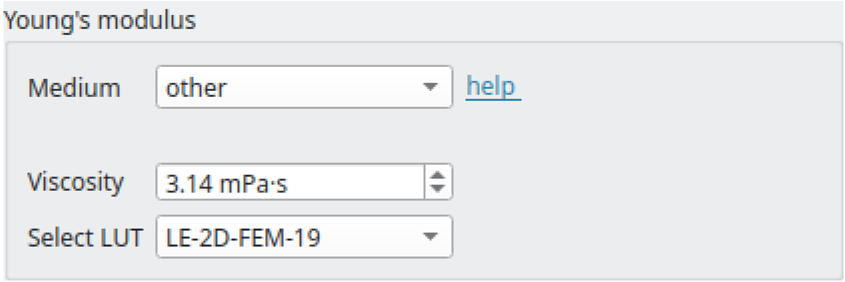


Young's modulus

Medium	CellCarrier	
Temperature	Manual	38.0°C
Viscosity	Buyukurganci (2022)	4.44 mPa·s
Select LUT	LE-2D-FEM-19	

Fig. 4.9: Temperature outside of known range.

If the temperature is not given as a feature or as metadata, then you may select the temperature manually. This case is visualized in Fig. 4.9. Here, the temperature is purposely set outside of the known range defined in [BBN+23], which is highlighted by coloring the viscosity red.


Young's modulus

Medium: other

Viscosity: 3.14 mPa·s

Select LUT: LE-2D-FEM-19

Fig. 4.10: Manually set the viscosity.

You may also set the viscosity manually by selecting *other* as a medium (Fig. 4.10). In this case, the values for temperature are irrelevant. Please only use this option if you know what you are doing (e.g. you have considered shear-thinning and temperature-dependence [BBN+23, Her17]).

Click *Apply* for any changes to take effect. The Young's modulus is then available for the selected dataset.

> **Note**
>
> In order to prevent users from doing a wrong analysis, DCscope does not allow users to change the buffer in the *Dataset* tab if the buffer medium (e.g. 0.59% MC-PBS) was selected in Shape-In or CytoShot. Therefore, the analysis of the Young's modulus cannot be re-done for a different buffer.

4.4.2 Bulk actions

The *Bulk action* menu has an entry for Young's modulus computation. In the associated dialog you can set the corresponding parameters for all datasets of the current session. Note that the options in this dialog do not necessarily reflect the options available for the individual datasets. Only valid options are adopted. For instance, you will not be able to change the medium for a dataset if a medium is already given in its metadata. To verify the options set, you can always check the current setting via the *Analysis View* (see above).

4.4.3 Validity

The computation of the Young's modulus is valid only for objects that initially have a spherical shape. In addition, the deformation and size values must be in a "valid region". Events outside this region will have a *nan*-valued (not a number) Young's modulus. Note that as a result, these events will be excluded from all plots when *remove invalid events* is checked in the *Filter* configuration tab.

The invalid regions (white in the figure above) include objects that are very small or objects with very low deformation values. The reason for that is a very steep increase of the Young's modulus with little decrease in deformation that could potentially result in very large simulation errors. In addition, regions with high deformation are invalid as well, because the simulations do not converge (objects simulated with lower Young's moduli become more and more elongated until they rupture). In practice, this means that the channel size has to be selected carefully to match the object sizes. Larger object sizes require wider channels if meaningful values for the Young's modulus are to be computed.

In the following, additional visualizations for other commonly used channel sizes and flow rates are shown:

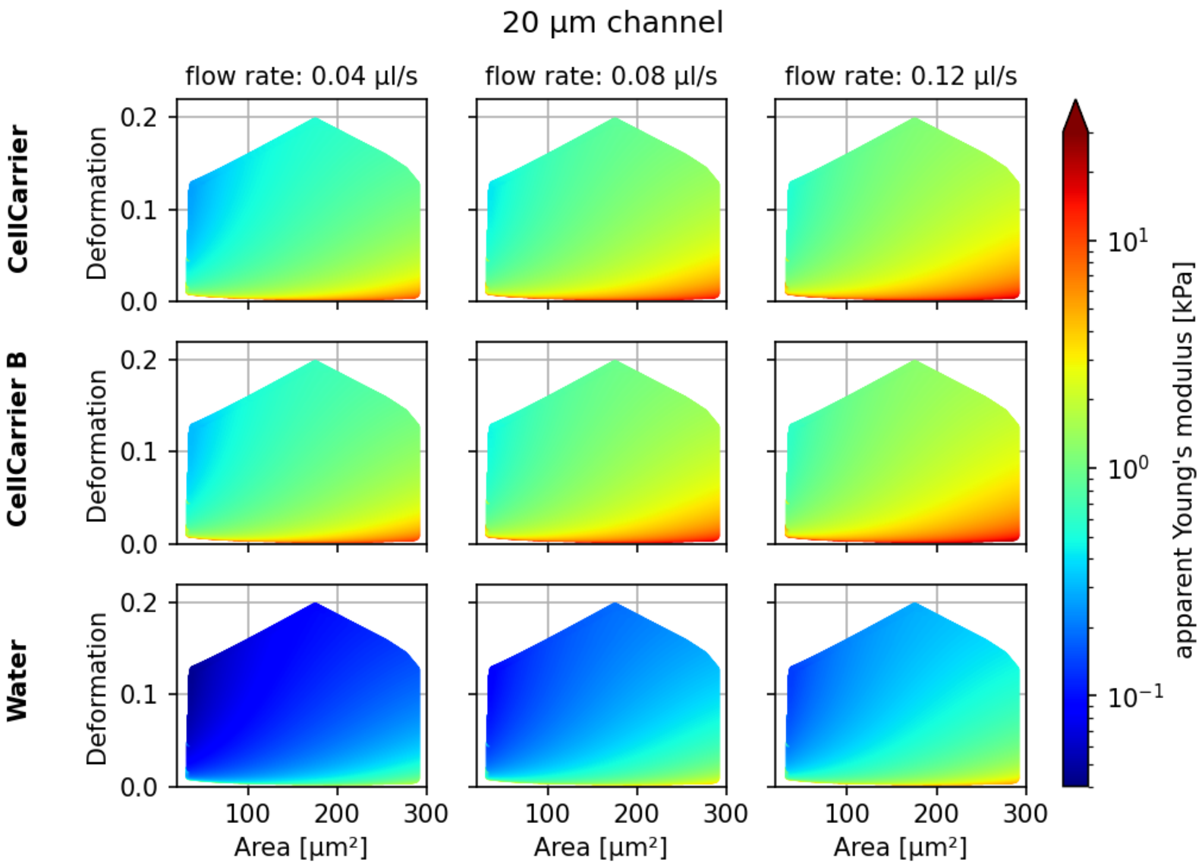
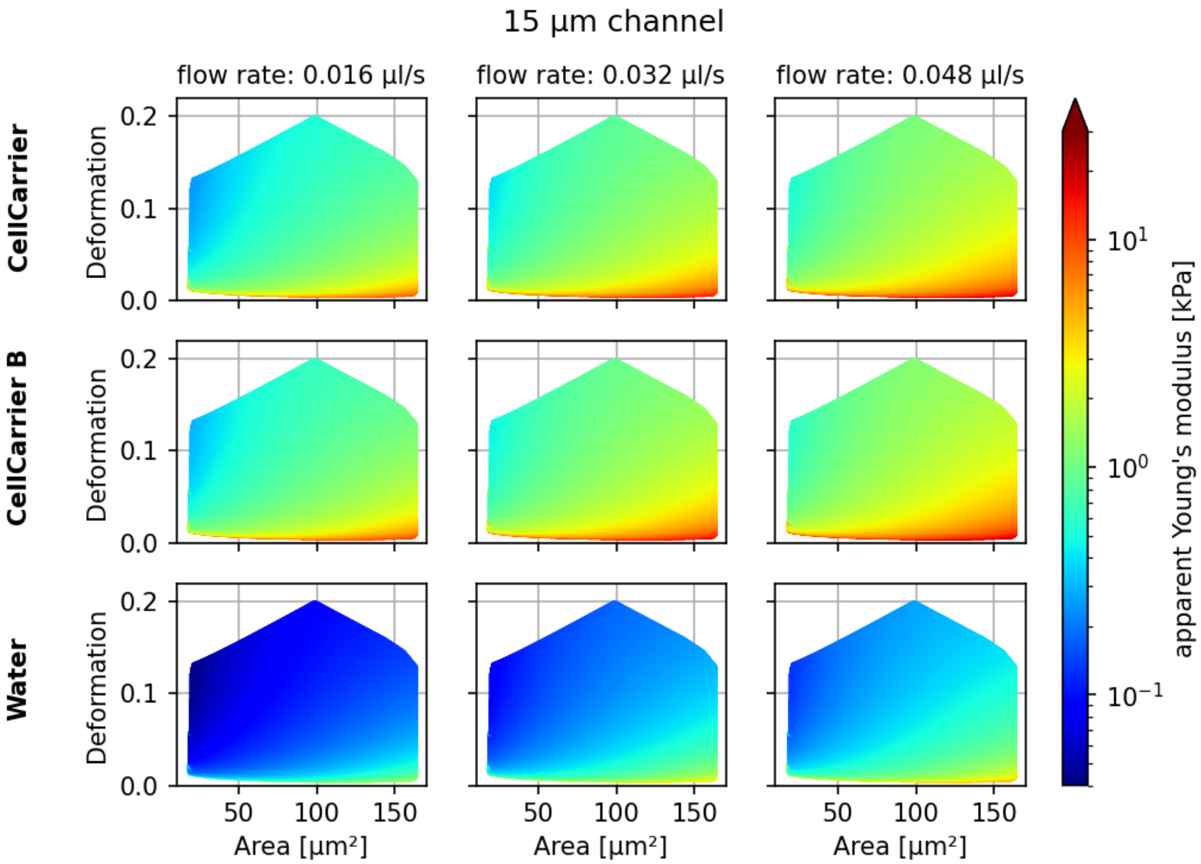
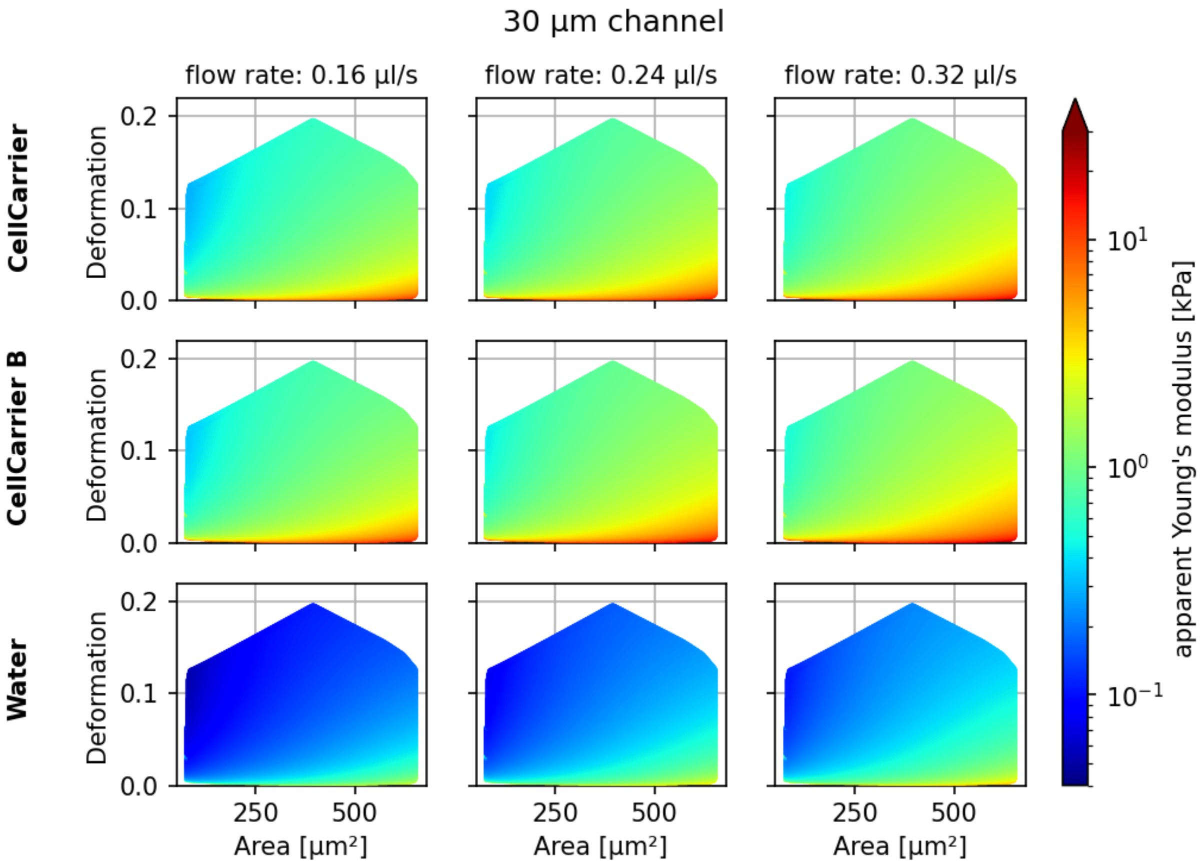
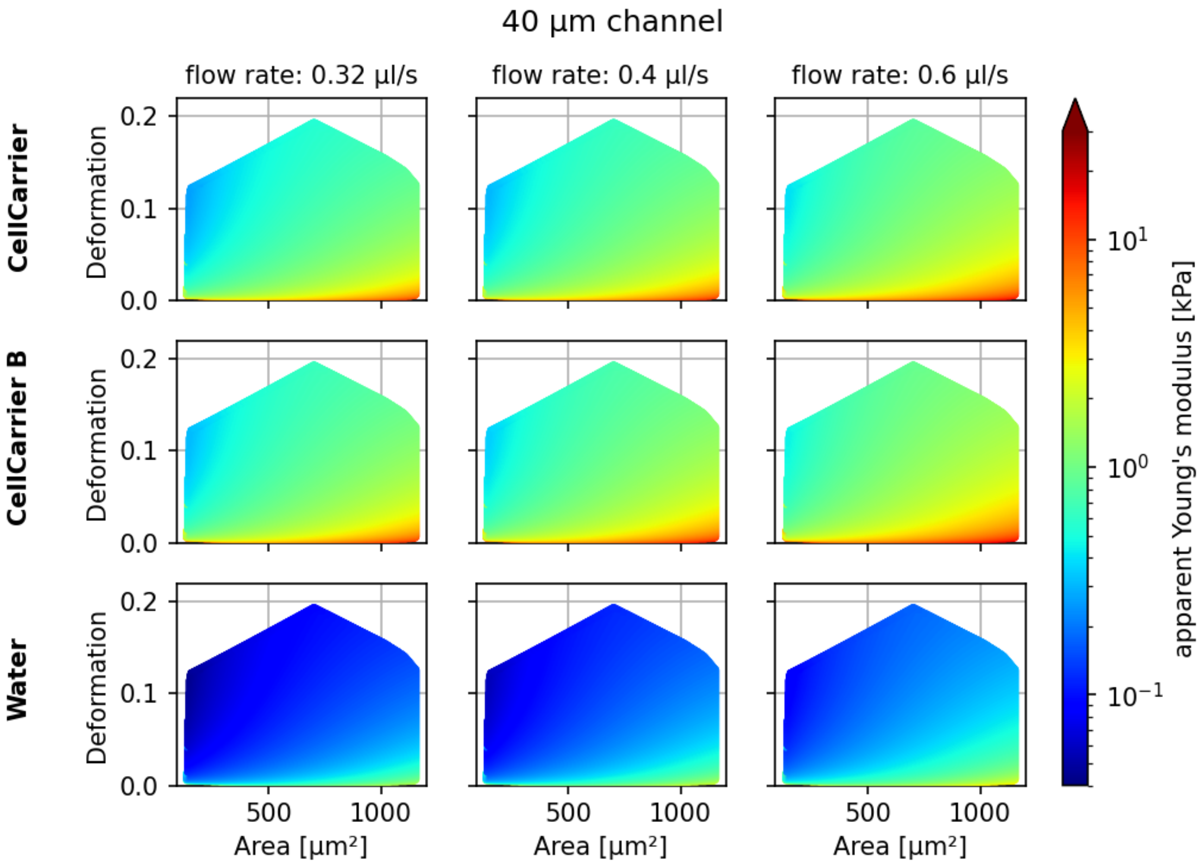


Fig. 4.11: Visualizations of the support and the values of the look-up table (LUT) used for determining the Young's modulus from deformation and cell area. The values of the Young's moduli in the regions shown depend on the channel size, the flow rate, the temperature, and the viscosity of the medium [MOG+15]. Here, they are computed for a 20 µm wide channel at 23°C with an effective pixel size of 0.34 µm. The viscosity of the MC-PBS buffers is computed according to [BBN+23] and the data are corrected for pixelation effects according to [Her17].


15 µm channel
flow rate: 0.016 µl/s
flow rate: 0.032 µl/s
flow rate: 0.048 µl/s
CellCarrier
CellCarrier B
Water
Deformation
0.0
0.1
0.2
50
100
150
Area [µm²]
apparent Young's modulus [kPa]
10^{1}
10^{0}
10^{-1}


30 µm channel
flow rate: 0.16 µl/s
flow rate: 0.24 µl/s
flow rate: 0.32 µl/s
CellCarrier
CellCarrier B
Water
Deformation
0.0
0.1
0.2
250
500
Area [µm²]
apparent Young's modulus [kPa]
10^{-1}
10^{0}
10^{1}


40 µm channel
flow rate: 0.32 µl/s
flow rate: 0.4 µl/s
flow rate: 0.6 µl/s
CellCarrier
CellCarrier B
Water
Deformation
0.0
0.1
0.2
500
1000
Area [µm²]
apparent Young's modulus [kPa]
10^{1}
10^{0}
10^{-1}

4.4.4 Implementation

As described above, the Young's modulus can be derived in multiple ways, for known media and global or event-based temperature values. The underlying implementation is described in the dclab docs.

4.5 Statistical significance testing (R and lme4)

Since version 2.5.0, DCscope allows you to quantify the statistical significance (p-value) when comparing two measurement groups. For this to work, you need to have R installed (tested with R 4.1.2). The analysis uses linear mixed effects models (LMM) as described in [HMMO18] and yields results identical to those obtained with DCscope 1.

The LMM analysis is performed using the lme4 R package and is based on the implementation in dclab (please read for more information on the options available) which implements a wrapper calling `Rscript`. Here, we make use of the dataset [XRM+20] to illustrate this functionality.

4.5.1 Basic linear mixed models

We would like to quantify the difference between human skeletal stem cells (SSC) and the human osteosarcoma cell line MG-63 (which is often used as a model system for SSCs) using a likelihood ratio test based on LMM.

Please load `this session` into DCscope (via *File | Open Session*). The experimental data are accessed from DCOR, so you will need a working internet connection. The session contains five datasets:

- SSC_16uls_rep1_20150611.rtdc (SSC repetition 1)
- SSC_16uls_rep2_20150611.rtdc (SSC repetition 2)
- MG63_pure_16uls_rep1_20150421.rtdc (MG-63 repetition 1)
- MG63_pure_16uls_rep2_20150422.rtdc (MG-63 repetition 2)
- MG63_pure_16uls_rep3_20150422.rtdc (MG-63 repetition 3)

Note

The data sets are ordered according to run index. In an ideal case, the run index would resemble the repetition of an experiment. Here, however, the run index is only an internal lab book reference.

In this example, we treat SSC as our "treatment" and MG-63 as our "control". These are just group names that remind us that we are comparing one type of sample against another type.

Note that the settings for Young's modulus computation have already been defined and that the datasets are already filtered with these box filters:

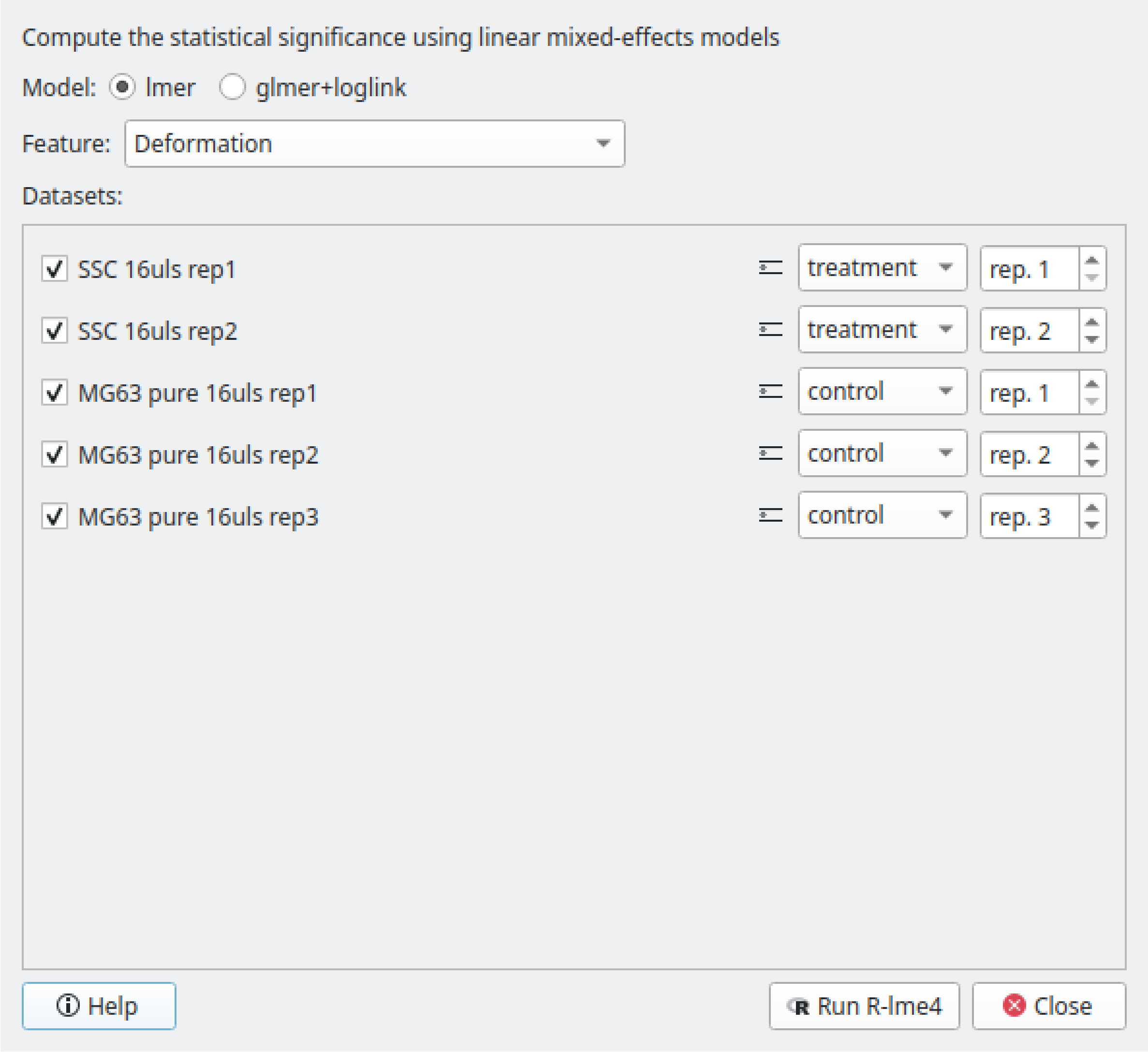
- Porosity (*area_ratio*): 0 to 1.05
- Area [µm^2] (*area_um*): 120 to 550
- Deformation (*deform*): 0 to 0.1

To perform an LMM analysis, go to *Compute | Statistical significance* and set the groups (SSC: treatment; MG63: control) and repetitions (enumeration by date-time).

Make sure that the feature "Deformation" is selected and hit *Run R-lme4*. The results of the LMM analysis are shown in a new dialog.

The most important numbers are extracted from the report and are shown at the top:

- **Fixed effect intercept:** The mean of the parameter chosen for all controls.



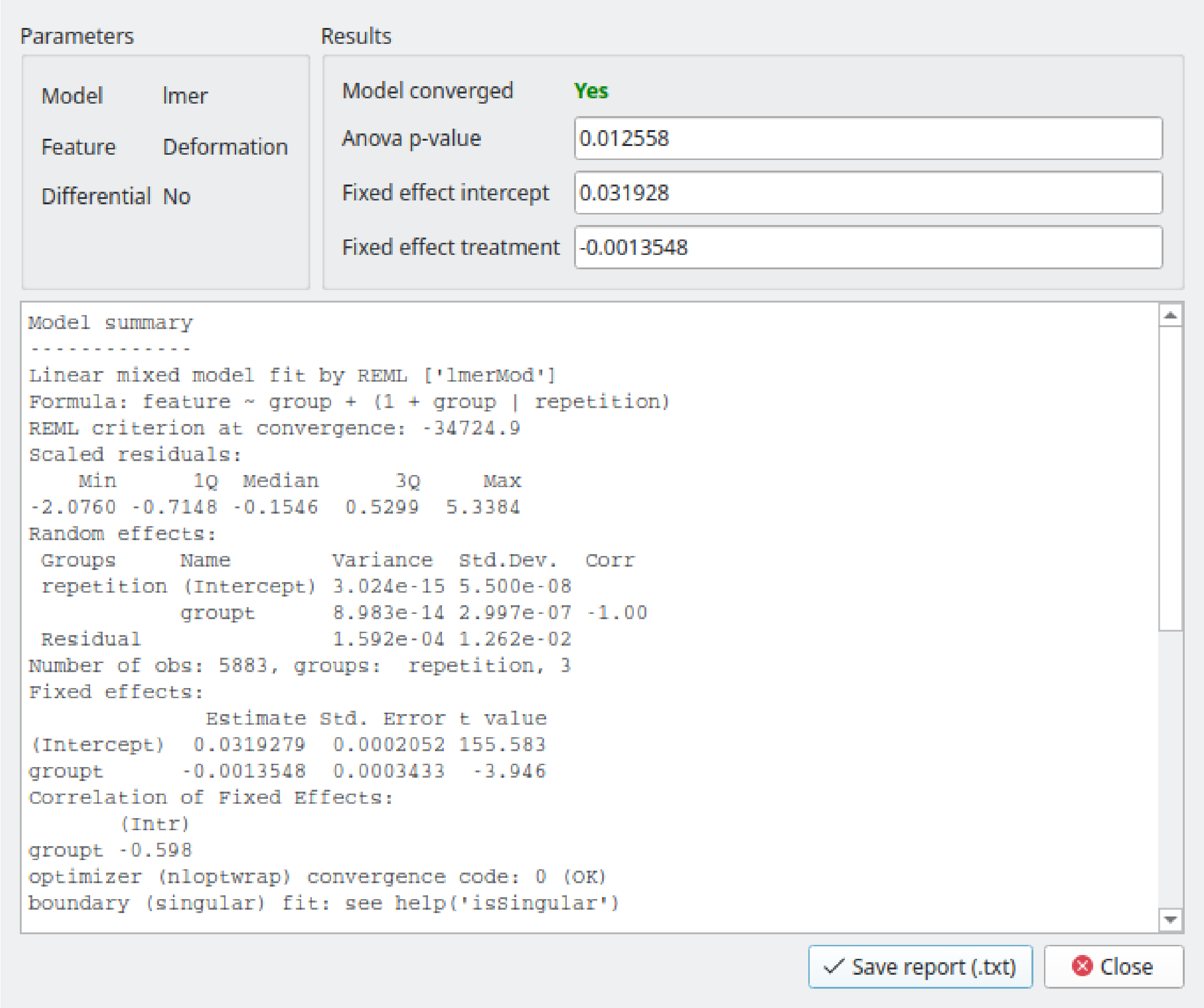
Compute the statistical significance using linear mixed-effects models

Model: ● lmer ○ glmer+loglink

Feature: Deformation

Datasets:

☑ SSC 16uls rep1	treatment	rep. 1
☑ SSC 16uls rep2	treatment	rep. 2
☑ MG63 pure 16uls rep1	control	rep. 1
☑ MG63 pure 16uls rep2	control	rep. 2
☑ MG63 pure 16uls rep3	control	rep. 3



Parameters

Model lmer

Feature Deformation

Differential No

Results

Model converged **Yes**

Anova p-value 0.012558

Fixed effect intercept 0.031928

Fixed effect treatment -0.0013548

```
Model summary
-------------
Linear mixed model fit by REML ['lmerMod']
Formula: feature ~ group + (1 + group | repetition)
REML criterion at convergence: -34724.9
Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.0760 -0.7148 -0.1546  0.5299  5.3384
Random effects:
 Groups     Name        Variance  Std.Dev.  Corr
 repetition (Intercept) 3.024e-15 5.500e-08
            groupt      8.983e-14 2.997e-07 -1.00
 Residual               1.592e-04 1.262e-02
Number of obs: 5883, groups:  repetition, 3
Fixed effects:
              Estimate Std. Error t value
(Intercept)  0.0319279  0.0002052 155.583
groupt      -0.0013548  0.0003433  -3.946
Correlation of Fixed Effects:
       (Intr)
groupt -0.598
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')
```

- **Fixed effect treatment:** The effect size of the parameter chosen between the mean of all controls and the mean of all treatments.
- **Anova p-value:** The p-value of the ANOVA test.

In the report, you will also find the **Full coefficient table**, which shows the effect size of the parameter chosen between control and treatment for every single experiment.

We are interested in the p-value, which is 0.01256 for Deformation. We repeat the analysis with Area (0.0002183) and Young's modulus (0.0002777). The p-values indicate that MG-63 (mean elastic modulus 1.26 kPa) cells are softer than SSCs (mean elastic modulus 1.54 kPa) [HMMO18].

4.5.2 LMM analysis of differential deformation

To check whether the deformation we see is not just an inherent deformation (visible in the reservoir data), we would like to perform a differential feature analysis.

Please load `this session` into DCscope (via *File* | *Open Session*). The experimental data are accessed from DCOR, so you will need a working internet connection. The session contains ten datasets (reservoir measurements in addition to the measurements above).

As in the example above, go to *Compute* | *Statistical significance* and set the groups and repetitions.

The results of the LMM analysis for differential deformation are:

The p-value for the differential deformation is magnitudes lower than the p-value for the (non-differential) deformation in the previous example. This indicates that there is a non-negligible initial deformation of the cells in the reservoir.

4.6 Loading Extensions (plugin features)

Since version 2.9.0, DCscope allows loading dclab plugin features. You may need those if you need to quantify anything that is not covered by the default features.

Note

If you installed DCscope via installer (not via *pip*), then many extensions might not work due to software dependencies that those extensions might have. If this happens, please create an issue in the DCscope repository so we can find a solution.

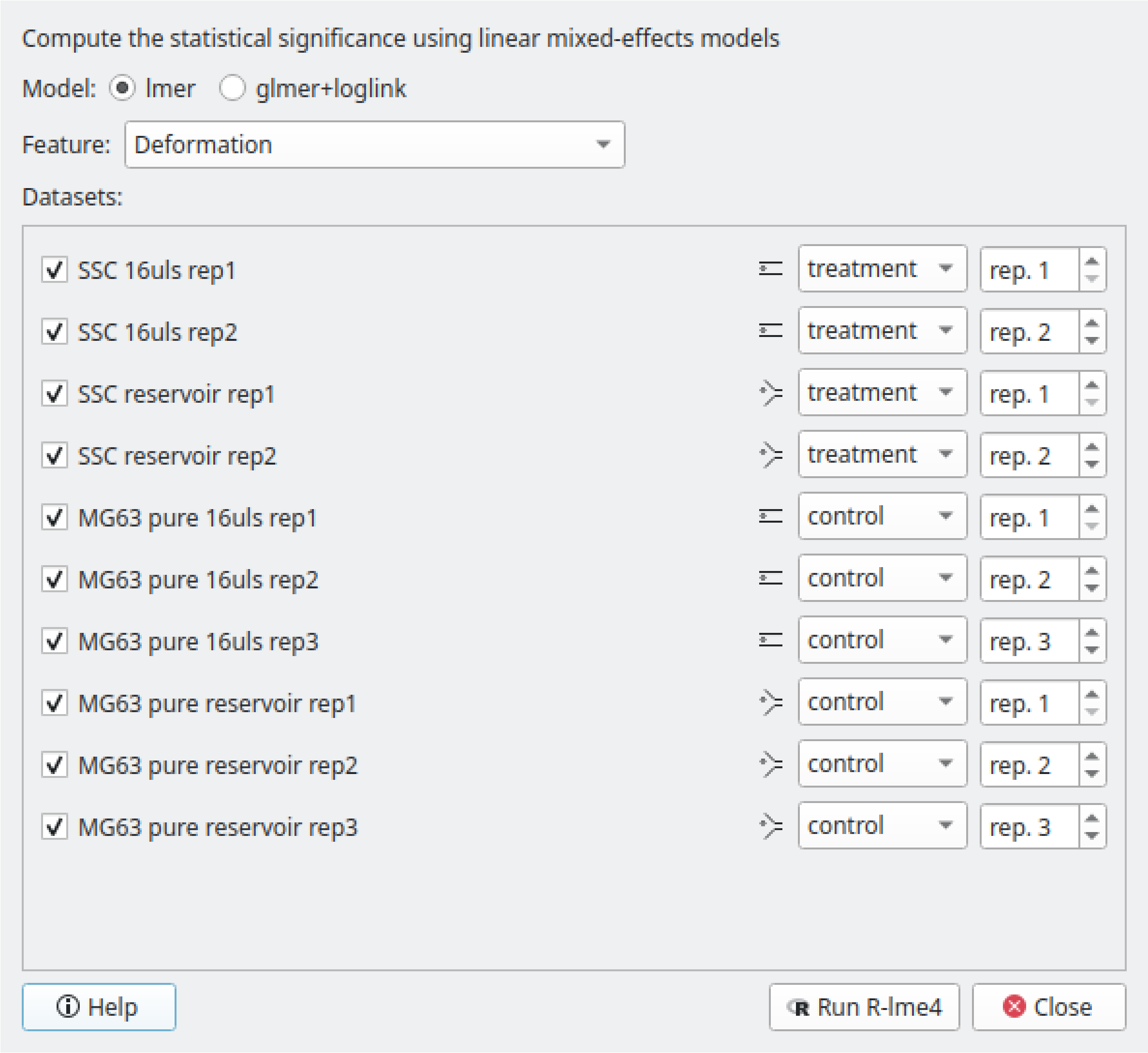
Warning

Extensions can be harmful. Please only load extensions that you received first-hand from people you trust.

You can load and manage extensions via the *Edit* | *Preferences* dialog in the *Extensions* tab.

4.6.1 Example: Fluorescence density

Download the `extension_fl1_density.py` extension and add it to DCscope. You will see a new scalar feature named "FL-1 density [a.u.]" that quantifies the collected fluorescence signal per object volume for the fluorescence channel 1. Using this extension as a template, you could create the density features for the other fluorescence channels as well.



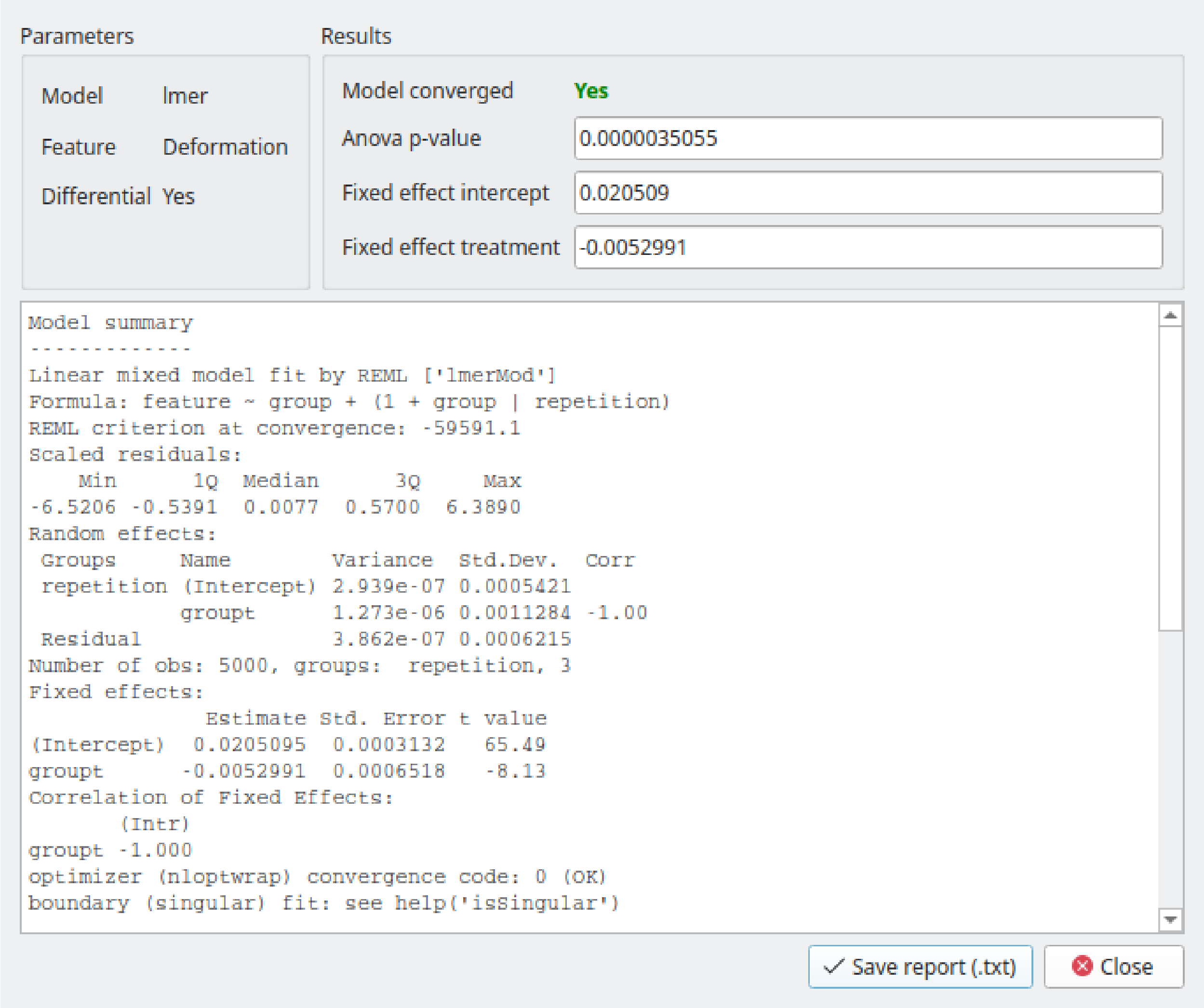
Compute the statistical significance using linear mixed-effects models

Model: ● lmer ○ glmer+loglink

Feature: Deformation

Datasets:

☑	SSC 16uls rep1	treatment	rep. 1
☑	SSC 16uls rep2	treatment	rep. 2
☑	SSC reservoir rep1	treatment	rep. 1
☑	SSC reservoir rep2	treatment	rep. 2
☑	MG63 pure 16uls rep1	control	rep. 1
☑	MG63 pure 16uls rep2	control	rep. 2
☑	MG63 pure 16uls rep3	control	rep. 3
☑	MG63 pure reservoir rep1	control	rep. 1
☑	MG63 pure reservoir rep2	control	rep. 2
☑	MG63 pure reservoir rep3	control	rep. 3



Model lmer

Feature Deformation

Differential Yes

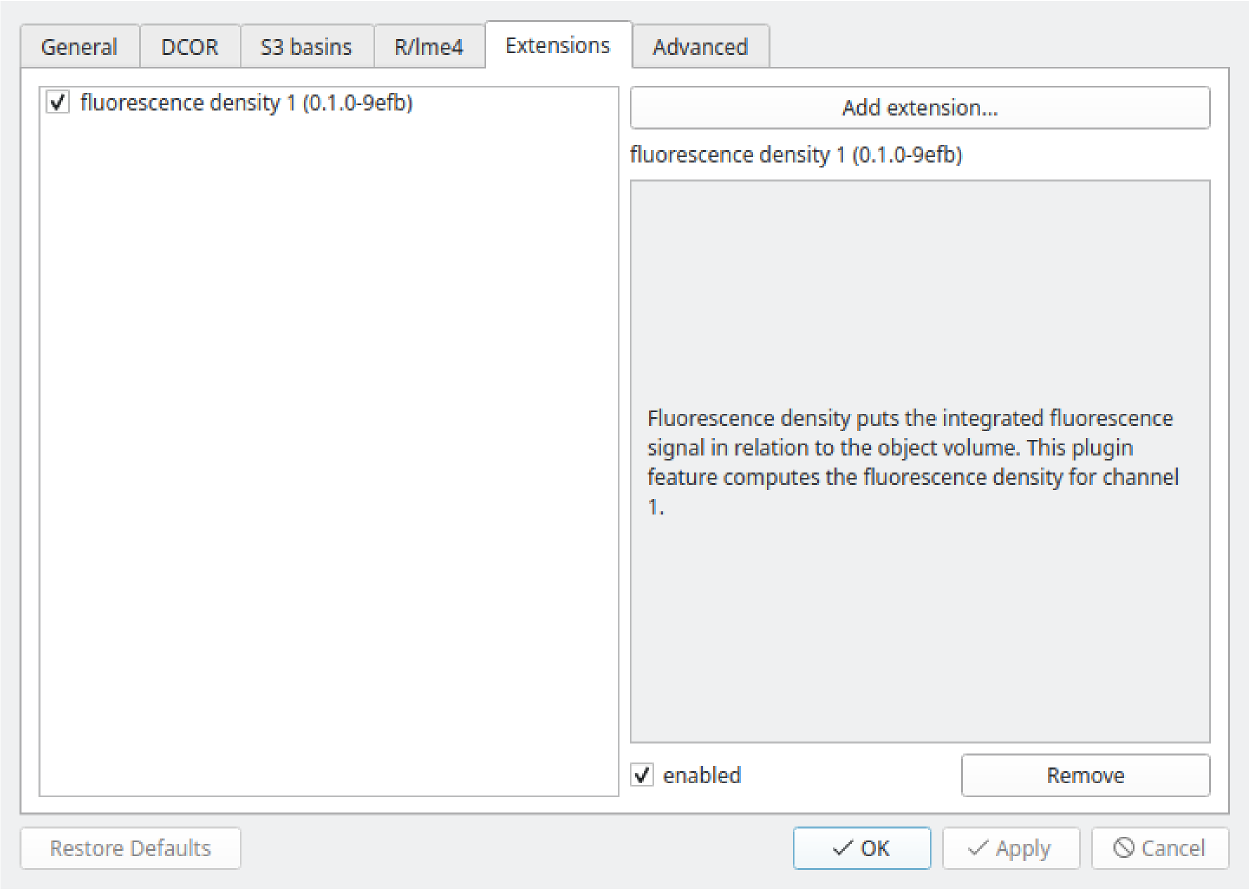
Model converged **Yes**

Anova p-value 0.0000035055

Fixed effect intercept 0.020509

Fixed effect treatment -0.0052991

```
Model summary
-------------
Linear mixed model fit by REML ['lmerMod']
Formula: feature ~ group + (1 + group | repetition)
REML criterion at convergence: -59591.1
Scaled residuals:
    Min      1Q  Median      3Q     Max
-6.5206 -0.5391  0.0077  0.5700  6.3890
Random effects:
 Groups     Name        Variance  Std.Dev.  Corr
 repetition (Intercept) 2.939e-07 0.0005421
            groupt      1.273e-06 0.0011284 -1.00
 Residual               3.862e-07 0.0006215
Number of obs: 5000, groups:  repetition, 3
Fixed effects:
              Estimate Std. Error t value
(Intercept)  0.0205095  0.0003132   65.49
groupt      -0.0052991  0.0006518   -8.13
Correlation of Fixed Effects:
       (Intr)
groupt -1.000
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')
```



☑ fluorescence density 1 (0.1.0-9efb)

fluorescence density 1 (0.1.0-9efb)

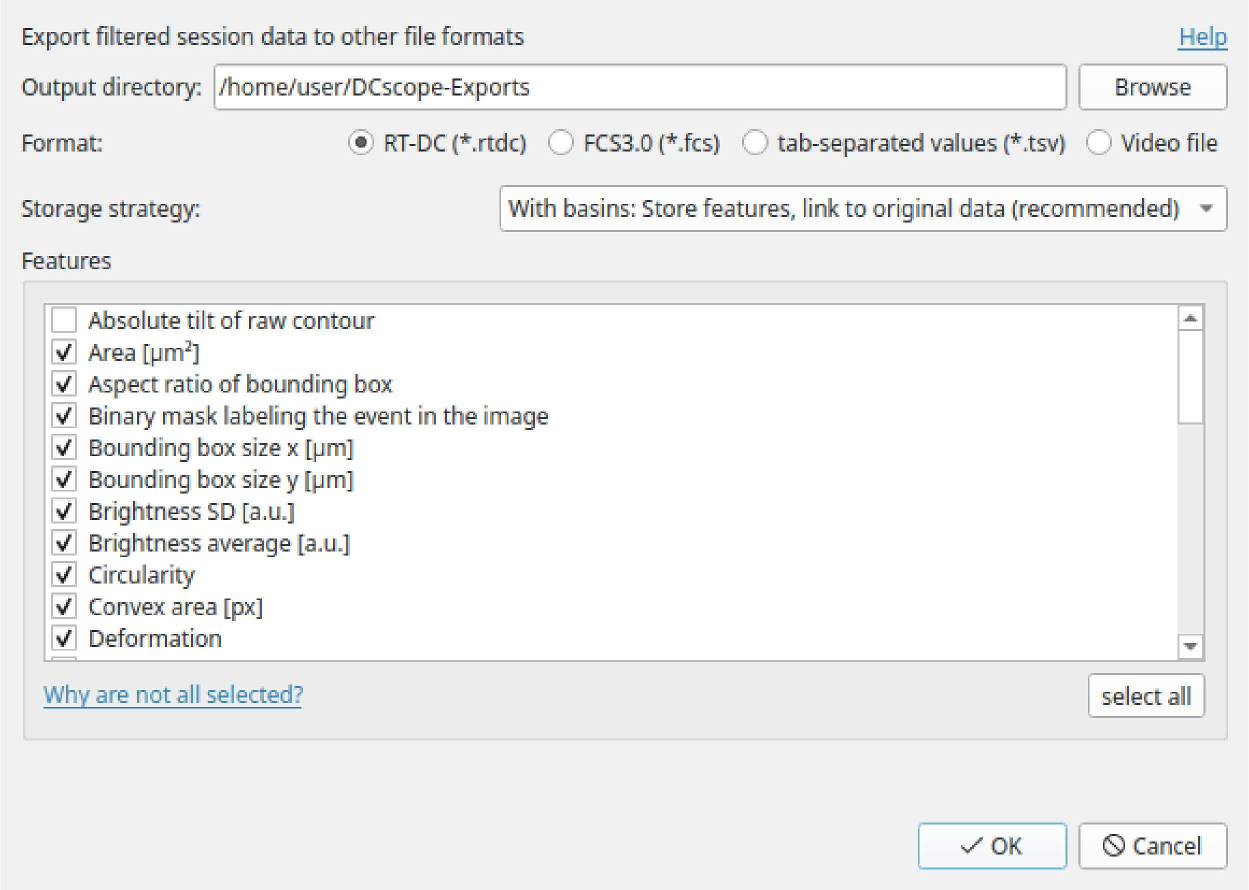
Fluorescence density puts the integrated fluorescence signal in relation to the object volume. This plugin feature computes the fluorescence density for channel 1.

☑ enabled

4.7 Export DC data

4.7.1 Use cases

Sometimes you would like to thin out the number of events or features in a dataset and export the result to a file. For instance, DCscope can export your DC data to the flow cytometry standard (.fcs) file format or to tab-separated values (.tsv) files for data analysis in other software. If you export to the .rtdc file format, you will stay in the DC ecosystem, allowing you to make use of dclab and DCscope or other software that can interface with HDF5 data. Since DCscope.22.0, it is also possible to export event images as a video file.



Export filtered session data to other file formats

Output directory: /home/user/DCscope-Exports

Format: ● RT-DC (*.rtdc) ○ FCS3.0 (*.fcs) ○ tab-separated values (*.tsv) ○ Video file

Storage strategy: With basins: Store features, link to original data (recommended)

Features

☐ Absolute tilt of raw contour
☑ Area [µm^2]
☑ Aspect ratio of bounding box
☑ Binary mask labeling the event in the image
☑ Bounding box size x [µm]
☑ Bounding box size y [µm]
☑ Brightness SD [a.u.]
☑ Brightness average [a.u.]
☑ Circularity
☑ Convex area [px]
☑ Deformation

Why are not all selected?

Fig. 4.12: Data export for current session

4.7.2 Export options

If you export to the .rtdc file format, you have several options that allow you to fine-tune what should be stored in the output file:

Features

The .rtdc file format supports non-scalar feature data such as *image* and *trace*. Consequently, these features are only available in the list when you select the .rtdc file format. Note that the features that are selected by default are those that do not require any additional computation time before the export can take place. If you select ancillary features such as *volume*, the time required to perform the export will be longer.

Storage strategy

Since DCscope.16 and with the introduction of basins (which tell dclab/DCscope where to find features belonging to the same dataset in other locations) in dclab 0.58.0, you can select a storage strategy. The storage strategy defines whether features and/or basins are stored in the output file.

- No basins: This is the behavior in earlier versions of DCscope. Only the selected features are written to the .rtdc file. Use this option if you are certain that you do not want basins.
- With basins: This is the recommended choice for normal usages. The selected features are written to the output file and all available basins (including the original file) are stored in the output file. If you open such an exported file, all features from the original dataset are still available as long as the original file or the basins defined therein are reachable.
- Only basins: This option is useful if you have a very confined analysis pipeline and would like to avoid data redundancy. In the output file, there will be no feature data. The output file will only contain the basin information. Naturally, this will massively speed-up data export (less data written to disk, no ancillary features are computed, no basin features are retrieved from other sources). The downside is that if none of the basins are available, the exported files are useless. Make sure that the source files with which you are working are always available. You can achieve that by working with data on DCOR, by making sure that the source files are always available on the same path or by exporting the data to the same directory as the source file.

4.7.3 Frequently asked questions

Why are not all features selected automatically? And why does exporting data take very long when I click on *select all* before?

By default, DCscope only selects those features that are readily available. Some features, such as volume, have to be computed by DCscope before they can be exported. The computation of features always happens for the **entire** dataset (not only for the features that you are exporting). Thus, even if you only export ~500 events, data export can take very long if the input file is large. This is also the reason why data export only takes long the first time you export from a dataset. Note that **you are not losing any feature data when you are exporting with the default settings**, unless you are exporting to .fcs or .tsv files, because DCscope can also compute missing features after opening the exported files.

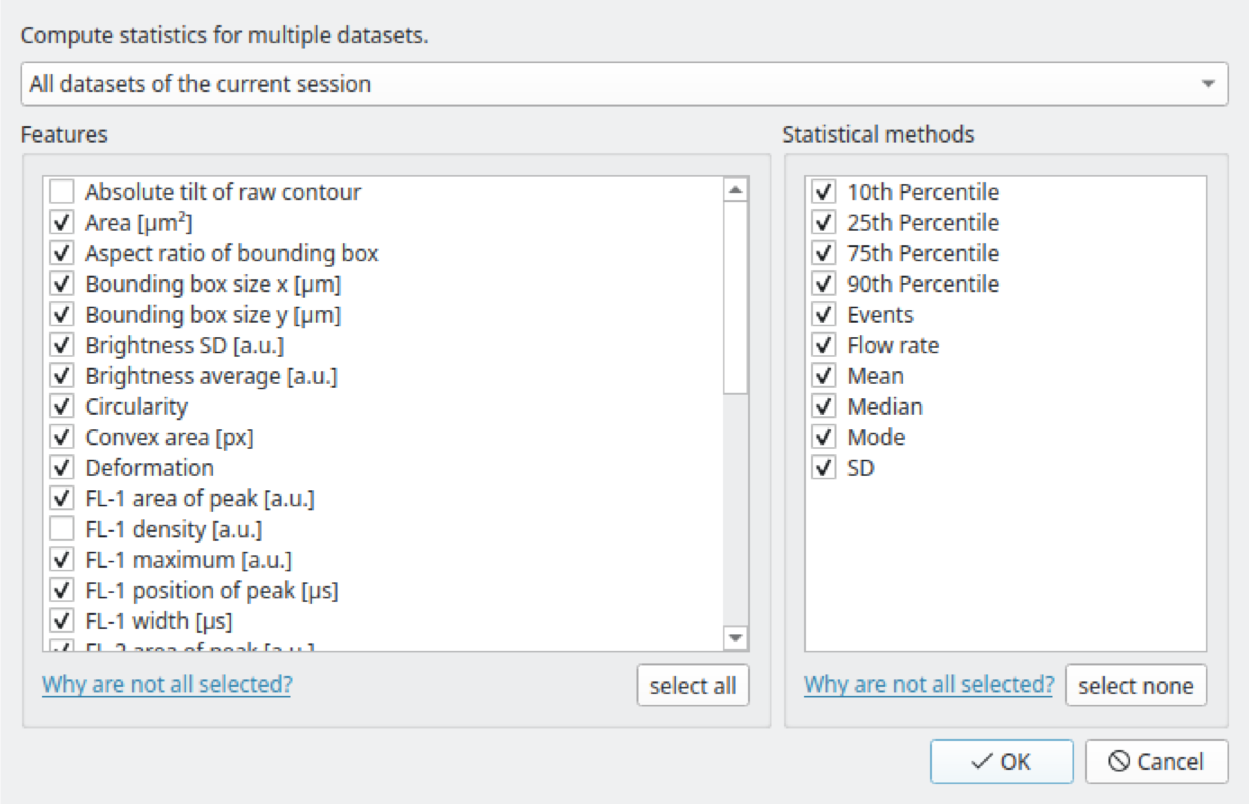
4.8 Export statistical data

The *Quick View window* quantifies the statistics for the features shown there. If you need access to statistics for more features of a dataset or if you need to quantify the statistics for multiple datasets on disk, you can use the *Statistics* tool available via the *Compute* menu.

There are two modes of operation. The first mode exports statistical data from the current session (Fig. 4.13). Statistics are always computed after the filter ray has been applied. You may select the features in the table on the left and the desired statistics in the table on the right.

The second mode computes statistics for .rtdc data on disk (Fig. 4.14). Select the *All datasets in a directory* option and (optionally) a filter ray from the current session that you would like to apply before computing the statistics.

Note that statistics may not be available for all selected features in all datasets. In such a case, statistics are exported as *nan* values in the resulting .tsv file.



Compute statistics for multiple datasets.

All datasets of the current session

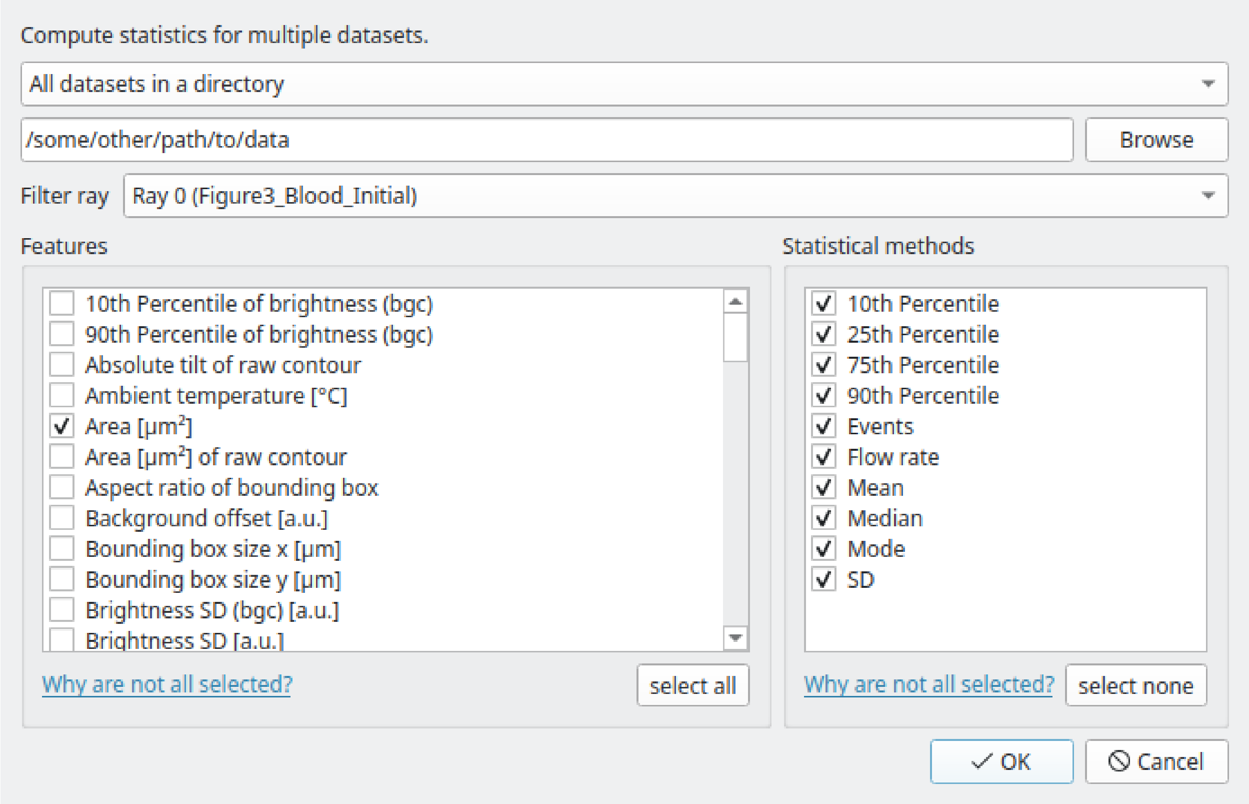
Features

☐ Absolute tilt of raw contour
☑ Area [µm²]
☑ Aspect ratio of bounding box
☑ Bounding box size x [µm]
☑ Bounding box size y [µm]
☑ Brightness SD [a.u.]
☑ Brightness average [a.u.]
☑ Circularity
☑ Convex area [px]
☑ Deformation
☑ FL-1 area of peak [a.u.]
☐ FL-1 density [a.u.]
☑ FL-1 maximum [a.u.]
☑ FL-1 position of peak [µs]
☑ FL-1 width [µs]

Statistical methods

☑ 10th Percentile
☑ 25th Percentile
☑ 75th Percentile
☑ 90th Percentile
☑ Events
☑ Flow rate
☑ Mean
☑ Median
☑ Mode
☑ SD

Fig. 4.13: Statistics export of current session



Compute statistics for multiple datasets.

All datasets in a directory

/some/other/path/to/data

Filter ray: Ray 0 (Figure3_Blood_Initial)

Features

- ☐ 10th Percentile of brightness (bgc)
- ☐ 90th Percentile of brightness (bgc)
- ☐ Absolute tilt of raw contour
- ☐ Ambient temperature [°C]
- ☑ Area [µm²]
- ☐ Area [µm²] of raw contour
- ☐ Aspect ratio of bounding box
- ☐ Background offset [a.u.]
- ☐ Bounding box size x [µm]
- ☐ Bounding box size y [µm]
- ☐ Brightness SD (bgc) [a.u.]
- ☐ Brightness SD [a.u.]

Statistical methods

- ☑ 10th Percentile
- ☑ 25th Percentile
- ☑ 75th Percentile
- ☑ 90th Percentile
- ☑ Events
- ☑ Flow rate
- ☑ Mean
- ☑ Median
- ☑ Mode
- ☑ SD

Fig. 4.14: Compute statistics for other data

CHAPTER

FIVE

DEVELOPER'S GUIDE

If you are here, then you would like to contribute to DCscope. Some parts of this document might need improvement and corrections.

5.1 Versioning

We try to adhere to semantic versioning DCscope uses setuptools-scm to automatically determine the version from the latest Git tag.

Setuptools-scm might cause problems for you if you are working on your own fork or branch and after installing DCscope in your local environment you get a version like *0.0.0post1421_cdqut*. The solution is to pull the tags from the main repository and after that install DCscope again.

If you have figured out how to do this, please update this document with a PR :).

5.2 Changelog

The *CHANGELOG* in the root of the repository consists of the corresponding git tags and the changes that were made in-between. We try to adhere to conventional commits.

Ideally, each entry in the changelog corresponds to one Git commit, but of course, this is not always feasible. You might want to read through the document to get an idea.

Note that there is only one changelog which might cause merge conflicts when you are making a PR. We might want to switch to something else in the future, but for now we keep this single-file system for the sake of simplicity.

Also note that if you are adding a new feature/enhancement/bugfix, make sure to properly increment the (possibly existing) future tag in the changelog. E.g. if the current version is "2.1.0" and somebody else made a bugfix, adding "2.1.1" to the changelog, then if you implement a new feature, you have to replace "2.1.1" with "2.2.0" in the changelog (because a new feature bumps the minor version).

5.3 CI/CD

We use GitHub Actions for testing and building artifacts for releases. Check the *.github/workflows* directory.

5.4 Documentation

The documentation is built using sphinx. To build the documentation, simply run this in the root of the repository:

```
# install the project, if not already done
pip install -e .
```

(continues on next page)

(continued from previous page)

```
# install requirements for docs, if not already done
pip install -r docs/requirements.txt
# go to the docs directory
cd docs
# build the docs
sphinx-build . _build
```

You can then open *_build/index.html* in your browser. The documentation is automatically built and deployed by readthedocs.io.

5.5 Pull Requests

The ideal route of contribution is via pull requests into the *main* branch. Make sure to

- name your forked branch after problem you are addressing
- update the changelog correctly
- all tests and linting (*flake8*) pass in CI

5.6 GUI development

DCscope is based on the Qt6 framework. This means you can use QtCreator/QtDesigner to graphically design the *.ui* files that are then loaded when starting DCscope. We use *PyQt6* (not *PySide6*) to interface with Qt.

Note that for some reason, new versions of QtDesigner produce .ui files that are incompatible with *pyuic6* (which converts the .ui files into Python code). As a result, we have to use QtCreator 8 in order for things to work (If you have a better solution, please let us know.): https://github.com/qt-creator/qt-creator/releases/tag/v8.0.2

The .ui files are loaded directly into DCscope and are not (as it is shown in many tutorials) converted to .py files and added to git versioning. If this annoys you, because it makes your development experience (e.g. code completion) unbearable: Feel free to scratch that itch and migrate DCscope to the .py-style scheme. The only requirements for that to be merged are:

- put all .ui files in a directory called *ui* in the root of the repository
- make sure the .ui files are included in the .tar.gz Python source distributable
- write a script that automatically converts all .ui files to .py files in the correct locations
- exclude the automatically generated .py files from linting
- add a GitHub Action that makes sure that changes in the .ui files are reflected in the .py files (to prevent a developer from forgetting to build the .py files)

CHAPTER

SIX

CHANGELOG

List of changes in-between DCscope releases.

6.1 version 2.28.1

- reg: limits of box filters incorrectly shown when QuickView was active
- enh: *Pipeline.get_min_max_coarse* handles inf data
- enh: align-right box filter selection button in Analysis View
- enh: disable "Apply" button when changing box filters in Analysis View
- build: include latest version of pyqtgraph to fix SVG export issue (#228)

6.2 version 2.28.0

- feat: introduce *TaskManager* and *TaskWorker* for more progress bars
- fix: plot resized when applying plot settings in AnalysisView
- fix: memory leak in plotting (plot elements were not removed)
- fix: KDE contour spacing test only done on largest percentile contour
- fix: deleting last filter in Analysis View caused *IndexError*
- fix: deleting last plot in Analysis View caused *IndexError*
- reg: subplot export via context menu did not work
- reg: dot-menu for plot matrix element removed in refactoring process
- enh: show number of tasks running in status bar
- enh: migrate data file export to *Taskmanager*
- enh: show progress bar when opening a session (uses *Taskmanager*)
- enh: enable rubberband-resize for all subwindows
- enh: maximize main window before loading session
- enh: streamline session loading (remove duplicate plotting calls)
- enh: move plot data checking to background thread
- enh: dark mode icons (#226)
- setup: bump dclab to 0.71.9 (thread-safety for filtering)

- docs: update screenshots (and recipes)
- ref: scatter plot data generation done in background thread
- ref: contour plot data generation done in background thread
- ref: use *TaskManager* for computing plot data in QuickView
- ref: cleanup main.py, session.py with the help of pyright
- ref: move custom colorbar definitions to *colorbaritem*
- ref: put heavy-duty plotting code into *pipeline_plot_compute*
- ref: separate PipelinePlotItem into its own submodule
- ref: compute brush color in background thread alongside KDE
- ref: *zoomin_contours* calls *compute_contours_from_state* unneccesarily
- tests: update manual tests session with correct contour spacing

6.3 version 2.27.0

- fix: delay on selecting "Plot" tab when "Marker hue" was set to "Feature"
- fix: switching tabs "View/Features" in QuickView did not update event data
- fix: set NaN-valued KDE of scatter plot points to zero
- fix: properly close DCOR search background runner in dialog
- fix: properly stop *EventGetterThread* on exit
- fix: due to race condition, sometimes incorrect range was shown in Plot tab
- enh: give all separate *QThread* object a name
- enh: remove wait cursor from some methods in QuickView
- enh: add "default KDE spacing" checkable button in AnalysisView Plot ("auto spacing" in pipeline *Plot* class)
- enh: change "auto range" checkbox to "default XY range" checkable button in AnalysisView Plot
- ref: migrate from loading .ui files to pyuic6-pregenerated Python files
- ref: migrate check for range from *PipelinePlot.update_content* to *Pipeline.check_auto_range_and_spacing*
- setup: bump dclab to 0.71.7 (ValueError when encountering empty pipeline)
- ci: bump build pipeline to Python 3.13

6.4 version 2.26.3

- enh: fetch QuickView image and trace data in *EventGetterThread*
- enh: busy indicator in QuickView when event data is fetched in the background
- ref: new *qv_event_getter* and *qv_image_vis* submodules
- setup: bump dclab to 0.71.6 (disable file locking, improved DCOR support, IndexError during plotting)

6.5 version 2.26.2

- reg: filters not applied in final plots since 2.25.1
- fix: prevent *pipeline.lock* deadlocks via Qt signal cluttering
- fix: update axes ranges when “Auto XY-range” is selected in plotting
- enh: remember plot size when manually resizing
- enh: increase default disk store size from 2GB to 9GB
- enh: reduce delay after selecting axes in the Plot view
- ref: incorporate *PipelinePlot* into sender-receiver signal pyramid
- setup: bump dclab to 0.71.2 (disk-caching of DCOR/HTTP/S3 data)

6.6 version 2.26.1

- fix: possibly fix *KeyError* on hover in QuickView by doing a *GraphicsScene.hoverItems.clear()* whenever something changes (#205)
- fix: disable Block Matrix QuickView selection when QuickView is closed
- fix: don’t update hidden QuickView panel when the pipeline changes
- fix: don’t replot in QuickView when the plot parameters are identical
- fix: always show event view in QuickView when initialized
- enh: allow to modify caching settings in the “Advanced” preferences tab
- enh: joint KDE map for scatter and contour visualization (#111)
- enh: setup basic logging
- enh: inform user about unavailable contour data with dclab 0.71.0 (#207)
- ref: move contour spacing plot settings to “General” section
- ref: migrate find smooth contour spacing code to dclab
- ref: remove redundant *plotting* submodule
- setup: bump dclab to 0.71.1 (smooth contours, joint KDE maps)

6.7 version 2.26.0

- feat: edit, duplicate and remove polygon filters in the “Filter” view
- fix: override-check in filter export did not work
- enh: enable disk caching for kernel density estimate data
- enh: bring windows to front when requesting controls via Block Matrix
- docs: update screenshots
- ref: disable caching for plots (fully done in dclab)
- setup: bump dclab to 0.69.2 (KDE caching, polygon filter stability)

6.8 version 2.25.4

- setup: bump dclab to 0.68.0 (open SWMR files in read mode)

6.9 version 2.25.3

- fix: *Pipeline.deduce_reduced_sample_names* failed for duplicate DCOR data
- fix: handle datasets without any events in the plot window (#223)
- setup: bump dclab to 0.67.5 (handle NaN-valued tables on DCOR)

6.10 version 2.25.2

- fix: check directory existence when loading files (#227)

6.11 version 2.25.1

- fix: removing a box filter did not clear the filter in some cases
- ref: do not allow specifying filters in *FilterRay.get_dataset()*
- ref: unify hashing caching of datasets in *FilterRay.segments*
- ref: remove ambiguous *FilterRay.root_child* workaround
- ref: simplify *FilterRay.get_final_child*
- tests: improve *FilterRay* test coverage
- setup: bump dclab to 0.67.4 (correct *reset_filter* method)

6.12 version 2.25.0

- feat: increase width of dataset tile in block matrix via settings (#225)
- fix: IndexError when trying to view the logs of a log-less dataset
- fix: prevent unchecking toolButtons for tool selection in QuickView
- fix: boolean settings value not read from settings file correctly
- fix: remove image flickering in QuickView when switching between slots
- fix: Auto-Range in analysis plot view sometimes unchecked
- fix: busy cursor when in load data file dialog
- enh: reduce sample names in block matrix (strip common prefix and suffix)
- enh: instantaneous tool tips in block matrix
- enh: show the created or quickviewed filter in analysis tab (#147)
- enh: show the created plot in analysis tab
- enh: improved presentation of multiple datasets with identical sample names
- ref: tentative *set_pipeline*, so widgets have direct access to the pipeline
- ref: raise ValueError in unreachable case in DataSlot creation

- ref: restructure pipeline management with sender-receiver signal pyramid
- ref: move slot, plot, and filter duplication to *Pipeline* class
- ref: refer power to Slot/Plot/Filter creation to *Pipeline*
- setup: bump pyqtgraph to 0.14.0
- tests: fix manual tests (session file broken)
- build: disable non-functional macOS builds

6.13 version 2.24.0

- feat: add contour-only plotting mode (#210)
- setup: bump dclab to 0.67.3 (data export, downsampling with NaNs)

6.14 version 2.23.1

- fix: QuickView plotting failed for all-NaN coordinate data (#223)
- build: remove pin scipy==1.11.4

6.15 version 2.23.0

- feat: add zoom-in functionality for contour plots (#211)
- fix: cache selected event per dataset in QuickView (#196)
- enh: add new logo (scope view and block matrix)

6.16 version 2.22.2

- fix: error message dialog limited to traceback length of three
- fix: copy error text to clipboard did not work in error message dialog
- enh: decorate QEvent-accepting slots in *main.py*
- enh: do not export "contour" to HDF5 files ("mask" is sufficient and faster)
- docs: add missing image for data export
- ref: only use closeEvent when exiting DCscope GUI
- setup: bump dclab to 0.67.0

6.17 version 2.22.1

- feat: notify user when loaded data have different pipeline hashes (#217)
- fix: prevent accidental polygon filter creation (#148)
- fix: disable QuickView controls when there is no data to display
- fix: IndexError when trying to show a deleted slot in QuickView (#214)
- setup: bump dclab to 0.64.0 (refresh DCOR basins)
- ref: simplify contour data retrieval and caching in plot_cache.py

6.18 version 2.22.0

- feat: export image data as video (#202)
- fix: persist feature selection in QuickView when switching data slots (#206)
- fix: show correct progress during data export (#138)
- docs: properly employ license GPL-3.0-or-later
- ref: fix deprecation warnings
- setup: bump dclab to 0.63.0

6.19 version 2.21.5

- fix: disable table graphs widget when an image-type table is shown
- fix: accessing table data from DCOR
- enh: do not keep dataset references in QuickView, only remember IDs
- enh: use bulk emodulus computation as viscosity calculator
- setup: bump dclab to 0.62.13

6.20 version 2.21.4

- fix: incorrect path list check when loading broken dataset
- fix: duplicate filters via block matrix results in AttributeError (#184)
- fix: plot sizes chaotic due to improper use of setFixedSize (#176)
- fix: plot size not set correctly when loading session
- fix: base automated plot size adjustment on minimal plot dimensions (#180)
- docs: update screenshot scripts and refresh screenshots
- enh: allow to change plot window size via edge handles (#51)
- setup: migrate pyqtgraph from 0.13.3 to 0.13.6 (#109)

6.21 version 2.21.3

- enh: support displaying unnamed table data
- setup: bump dclab to 0.62.11

6.22 version 2.21.2

- fix: signal race condition when editing filters in AnalysisView (#148)
- fix: signal race condition when editing plots in AnalysisView (#172)
- enh: use QuickView selection to navigate AnalysisView slots and filters
- enh: when batch-loading datasets, allow individual files to fail
- enh: auto-select logs in AnalysisView

- enh: improve table graph selection (remember graph, clear if unavailable)

6.23 version 2.21.1

- fix: maximize image size in QuickView
- fix: colormap for QPI data only updating for one image view in QuickView
- ref: QuickView code needed tidying up

6.24 version 2.21.0

- feat: implement showing dataset basins in the analysis view (#182)
- feat: implement showing dataset logs in the analysis view (#140)
- feat: implement showing dataset tables in the analysis view (#141)
- enh: more Quick View default choices than "deform" and "area_um"
- enh: update icon theme
- docs: document new features in Analysis View
- setup: pygments (for json formatting) is now an explicit dependency
- setup: bump dclab to 0.62.9
- ref: simplify logic in Analysis View
- ref: do not override *QtWidgets*' *__setstate__* and *__getstate__*; use *write_pipeline_state* and *read_pipeline_state* instead

6.25 version 2.20.1

- fix: regression TypeError when trying to plot contour data

6.26 version 2.20.0

- feat: display and handle quantitative phase imaging data
- fix: filter range control not updating limits, displaying value "100" (#183)
- fix: filter range control not updating handles when resized
- enh: improve plotting performance in QuickView polygon selection mode

6.27 version 2.19.1

- fix: "buyukurganci-2022" was not actually the default viscosity model
- ref: remove distutils dependency
- docs: add developer's section

6.28 version 2.19.0

- feat: support accessing private S3 basins (#174)
- enh: enable lut selection in analysis view plots (#71)
- fix: only allow user to choose “R.exe” in R binary manual search (#173)
- enh: correctly show system R binary in settings
- ref: migrate from pyqt5 to pyqt6 (#165)
- ref: migrate from pkg_resources to importlib.resources (#155)
- setup: support Python 3.12
- setup: new dependency “boto3” via “dclab[s3]””

6.29 version 2.18.2

- enh: enable lut selection to plot isoelasticity lines (#71)
- enh: replace findText with findData of comboBox_visc_model
- fix: allow to select R binary when not found automatically
- enh: enable lut selection for emodulus computation in bulk actions (#71)
- enh: enable lut selection for emodulus computation (#71)
- setup: bump dclab to 0.62.5 (properly resolve path to R executable)

6.30 version 2.18.1

- setup: bump dclab to 0.61.3 (replace rpy2 with wrapper) (#73 #161)

6.31 version 2.18.0

- BREAKING CHANGE: set default viscosity model for Young’s modulus computation to “buyukurganci-2022”, because it is more accurate and also because it is defined for the “0.83% MC-PBS” medium
- BREAKING CHANGE: remove support for machine learning extensions (.modc files), because they are not used
- fix: remove leading/trailing slashes for DCOR servers
- setup: bump dclab to 0.61.1 (fix basin-based array access, export from datasets with internal basins, other minor fixes)

6.32 version 2.17.0

- setup: bump dclab to 0.60.0 (internal basin support, export logs)

6.33 version 2.16.5

- setup: bump dclab to 0.59.1 (mapped-basin now fully supported)

6.34 version 2.16.4

- setup: bump dclab to 0.59.0 (improved support for basin features)

6.35 version 2.16.3

- maintenance release

6.36 version 2.16.2

- fix: data export feature selection must default to features_loaded
- enh: only show local features in QuickView Event Feature tab
- enh: show dcnum pipeline identifiers in Analysis view
- enh: add section:key tool tips for analysis view metadata
- enh: added links to docs explaining default export selection
- docs: remove Shape-Out 1 reference
- setup: remove rpy2 from dependencies until #161 is fixed
- setup: migrate from setup.py to pyproject.toml
- setup: bump dclab to 0.58.7 (performance)

6.37 version 2.16.1

- setup: bump dclab to 0.58.4 (reduced memory usage when computing contour-based features such as volume, export dataset configuration as metadata in .tsv export, support both posix and nt relative basin paths)

6.38 version 2.16.0

- feat: enable storage strategy (basin control) in export dialog (#159)
- fix: ask for .tsv output path before computing statistics
- enh: color feature data in QuickView combobox according to availability
- enh: hide basinmap features in scatter plot selections (#159)
- docs: remove reference to Shape-Out 1 in introduction
- docs: mention basins and link to dclab (#159)
- docs: add quick guide for data export
- ref: reduce usage of *dclab.dfn.scalar_feature_names* (#108)
- setup: bump dclab to 0.58.2

6.39 version 2.15.5

- fix: right-click menu not working for plot widgets (#160)
- ref: introduce *FeatureComboBox* class for listing features

6.40 version 2.15.4

- fix: very small numbers not parsed when showing metadata (#158)
- fix: do not allow question mark "?" in exported file names (#156)
- fix: do not override files when exporting data (#153)
- fix: remove the "so2exp" prefix when exporting data (#157)

6.41 version 2.15.3

- build: PyInstaller to rebuild its bootloader (avoid false positive malware detection)

6.42 version 2.15.2

- fix: handle zero-valued contour spacing
- setup: bump dclab to 0.57.4 (handle zero-valued spacing for KDE contour)

6.43 version 2.15.1

- fix: statistics not updating when changing plot axes (#154)
- setup: bump dclab from 0.57.0 to 0.57.1

6.44 version 2.15.0

- enh: avoid creating hierarchy when no filters are applied (speed)
- enh: cache computed statistics
- ref: avoid division-by-zero warning
- setup: bump dclab from 0.56.3 to 0.57.0 (speed)

6.45 version 2.14.3

- fix: intermediate plotting of first event whenever a slot is added
- fix: catch zero-division warning when plotting empty image
- enh: remember selected events in QuickView
- enh: avoid showing first dataset event when user clicks in QuickView
- setup: bump dclab from 0.55.7 to 0.56.3 (enhanced basin support)

6.46 version 2.14.2

- setup: bump dclab from 0.55.6 to 0.55.7

6.47 version 2.14.1

- fix: perform sanity check for contour spacing before plotting
- fix: use pyqtgraph.SpinBox to edit box filters (#149)
- enh: minor optimization of min/max feature algorithm
- enh: only preselect innate features when computing statistics (#150)
- setup: bump dclab to 0.55.6 (new scalar features, dcor access)
- setup: bump pyqtgraph from 0.12.4 to 0.13.3

6.48 version 2.14.0

- ref: don't set RTDCBase.identifier to slot identifier
- setup: drop support for Python 3.8
- setup: bump dclab from 0.54.2 to 0.55.4 (better DCOR support)
- build: use Python 3.10 for building executables

6.49 version 2.13.7

- ci: fix rtd build pipeline
- setup: bump dclab from 0.52.0 to 0.54.2 (improved basin support)

6.50 version 2.13.6

- ci: fix Windows build pipeline

6.51 version 2.13.5

- setup: bump requests to 2.31.0 (CVE-2023-32681)
- setup: bump scipy to 1.10.0 (CVE-2023-25399)
- setup: bump dclab from 0.50.4 to 0.52.0

6.52 version 2.13.4

- tests: fix fake dataset creation

6.53 version 2.13.3

- fix: casting error when using integer feature for plot color
- fix: also export logs and tables to .rtdc with prefix "so2exp_"
- enh: minor minimization in waiting times
- ref: unify json dumps when saving sessions
- setup: bump dclab from 0.48.4 to 0.50.4

6.54 version 2.13.2

- docs: don't build PDF on rtd (failed)
- setup: bump dclab from 0.48.2 to 0.48.4 (don't load invalid inertia ratio data from existing HDF5 files)

6.55 version 2.13.1

- setup: bump dclab from 0.48.1 to 0.48.2 (improved speed, fixed bug with inertia ratio in long channels)

6.56 version 2.13.0

- feat: support new viscosity model buyukurganci-2022 (#136)
- enh: increase speed of QuickView event visualization
- enh: turn on auto-apply in QuickView by default
- setup: bump dclab from 0.47.2 to 0.48.1

6.57 version 2.12.0

- feat: allow to disable KDE or use feature data as marker hue in QuickView
- fix: do not perform auto-apply when changing downsampling value and downsampling is disabled in QuickView
- fix: catch IndexError when trying to view an event that is not there anymore in polygon-selection-mode in QuickView
- enh: make auto-apply in QuickView plot settings optional
- docs: update screenshot scripts

6.58 version 2.11.11

- enh: strip http(s):// from DCOR server string in settings
- enh: improve GUI of QuickView (#133, #43)
- setup: bump dclab from 0.47.0 to 0.47.2

6.59 version 2.11.10

- fix: detect chaged files on disk when reloading metadata
- setup: bump dclab from 0.46.4 to 0.47.0 (new bg_med feature and other convenience methods)

6.60 version 2.11.9

- fix: include logs when exporting data (#126)
- setup: bump dclab from 0.46.2 to 0.46.4 (new features)

6.61 version 2.11.8

- setup: bump dclab from 0.45.0 to 0.46.2 (#132)
- build: pin pyinstaller version to 5.3
- ref: use *hashobj* from upstream dclab (#132)

6.62 version 2.11.7

- build: fix build pipelines

6.63 version 2.11.6

- setup: bump dclab from 0.43.0 to 0.45.0 (scalar feature support)

6.64 version 2.11.5

- build: use InnoSetup "autopf" feature for installation directory

6.65 version 2.11.4

- enh: preselect only innate features in data export to reduce computation time
- docs: minor improvements
- setup: bump dclab from 0.42.2 to 0.43.0 (improved compression)
- setup: include zstandard library in distribution installer

6.66 version 2.11.3

- setup: bump dclab from 0.42.1 to 0.42.2 (lazy-load .rtdc data which increases initial opening times)
- setup: bump pyqtgraph from 0.12.3 to 0.12.4

6.67 version 2.11.2

- setup: bump dclab from 0.42.0 to 0.42.1 (support background image access from DCOR)

6.68 version 2.11.1

- fix: allow to specify 0.5 increments when filtering integer features (#123)
- setup: bump dclab from 0.41.0 to 0.42.0

6.69 version 2.11.0

- feat: allow to load encrypted dcor access tokens (#122)
- setup: bump dclab from 0.39.17 to 0.41.0
- setup: bump scipy to 1.8.0

- build: bump Python version to 3.9

6.70 version 2.10.0

- feat: export filter ray for each dataset via shortcut (#119)
- setup: bump dclab from 0.39.15 to 0.39.17 (#118)

6.71 version 2.9.6

- fix: extensions not loaded during startup on Windows (#117)

6.72 version 2.9.5

- fix: polygon translation via drag-n-drop in polygon edit mode was not taken into account (#115)
- enh: add (de-)select all button for lists in ComputeStatistics (#116)
- enh: convert all selectable lists in dialogs to new BulkList class
- setup: bump dclab from 0.39.13 to 0.39.15

6.73 version 2.9.4

- enh: display session file name in window title
- setup: drop support for older numpy versions
- setup: bump dclab from 0.39.9 to 0.39.13 (fix visualization of the "ml_class" score when the ml_score features contain nans)

6.74 version 2.9.3

- fix: implement closeEvent in main and ask user before exiting
- fix: fix possible failure during update check
- fix: show a message box when the R version is incompatible (#114)
- ref: only reload R/lme4 information in preferences when the corresponding tab is selected to avoid preferences lock-up (#114)
- tests: use hidden settings variable to skip asking the user to clear the session
- setup: bump dclab from 0.39.5 to 0.39.9

6.75 version 2.9.2

- setup: bump dclab from 0.39.3 to 0.39.5 (fix opening files with some online filter settings)

6.76 version 2.9.1

- enh: add button "Find best spacing" in Plot tab in Analysis View that iteratively finds sane spacing values (#110)
- enh: notify user in plot if contour spacing is too large (#110)

- enh: add action to remove all datasets from a session (#97)
- fix: mysterious behavior of subclassed pg.ColorBarItem (#109)
- setup: bump dclab from 0.39.1 to 0.39.3 (increased verbosity, renamed feature labels, refactored DCOR access)

6.77 version 2.9.0

- feat: add support for extensions in the form of loading dclab plugin and machine-learning features (#85)
- setup: drop support for Python 3.7
- setup: bump dclab from 0.37.2 to 0.39.1

6.78 version 2.8.1

- fix: wrong scale for viscosity in bulk emodulus computation (#104)
- fix: save unused polygon filters in sessions (#101)
- fix: allow to delete polygon filters (#100)
- setup: bump dclab from 0.37.0 to 0.37.2 (enhanced speed for operations involving contour computation)

6.79 version 2.8.0

- BREAKING CHANGE: The volume feature in all previous versions was overestimated by about 2µm^3. Please re-run your analysis pipeline.
- setup: bump pyqtgraph from 0.12.2 to 0.12.3
- setup: bump dclab from 0.36.1 to 0.37.0 (faulty volume computation)

6.80 version 2.7.4

- setup: bump dclab from 0.35.6 to 0.36.1 (improved export speed)

6.81 version 2.7.3

- build: remove console window (#102)

6.82 version 2.7.2

- setup: bump dclab from 0.35.4 to 0.35.6 (Young's modulus could not be computed in some cases)

6.83 version 2.7.1

- enh: display online polygon filters in metadata tab (#94)
- enh: show user configuration section in metadata tab (#86)
- setup: bump dclab from 0.34.3 to 0.35.4
- tests: change test data to avoid warnings in dclab 0.35.0

6.84 version 2.7.0

- feat: accept DCOR drag events from DCOR-Aid data browser

6.85 version 2.6.9

- fix: don’t reset color and name of dataset during duplicate matrix action (#96)

6.86 version 2.6.8

- fix: adapt color scheme to work in dark mode (#92)

6.87 version 2.6.7

- fix: correctly handle missing features on data export (#80)
- ref: cleanup (default arguments and PyQt5 object references)
- setup: bump dclab from 0.34.2 to 0.34.3 (set R_HOME)
- build: bump rpy2 from 3.4.2 to 3.4.5
- build: bump pyqtgraph from 0.12.1 to 0.12.2 and remove colorbar export workarounds

6.88 version 2.6.6

- fix: false alarms about wrong pyqtgraph version on startup (removed the responsible code, because there are now more regular releases of pyqtgraph)
- enh: allow passing session files, datasets, and filters via the command line (#88)
- enh: associate the .so2 file format with Shape-Out on Windows (#88)
- build: migrate to Inno Setup 6
- ref: avoid pyqtgraph DeprecationWarning due to *translate*

6.89 version 2.6.5

- fix: migrate to pg.ColorBarItem (with some manual patches from upstream) to resolve issues with SVG export (#87)
- fix: units of viscosity is mPa*s (#91)
- setup: bump pyqtgraph from 0.12.0 to 0.12.1
- setup: bump dclab from 0.33.3 to 0.34.2
- ci: fix build issues for macOS

6.90 version 2.6.4

- fix: HTML-escape plot titles and axis labels (#82)

6.91 version 2.6.3

- build: pin rpy2 to 3.4.2 on Windows (#84)

6.92 version 2.6.2

- setup: migrate to pyqtgraph 0.12.0
- setup: bump dclab from 0.33.1 to 0.33.3
- docs: fix rtd build

6.93 version 2.6.1

- setup: remove deprecated setup.py test
- setup: bump dclab to 0.33.1 (filters were ignored when exporting trace data to hdf5, see https://github.com/DC-analysis/DCKit/issues/17)

6.94 version 2.6.0

- feat: add background correction in quickview (#54)
- fix: workaround for different plot widths due to title label (#33)

6.95 version 2.5.2

- build: add pyinstaller hook for pyqtgraph

6.96 version 2.5.1

- fix: preferences dialog did not open in macOS App bundle (#76)
- enh: disable R settings in preferences dialog if rpy2 is missing
- build: bump dclab from 0.29.1 to 0.32.4 (this should have happened in the 2.5.0 release which introduced R-lme4 analysis)
- build: bump pyqtgraph to commit c1900 (post 0.11.1) which mainly improves plot export (#59)
- tests: introduce manual tests that require a human in "tests-manual"
- ref: key [calculation]: "emodulus model" is deprecated in dclab
- ref: use "lut_identifier" instead of deprecated "method" keyword when retrieving isoelasticity lines

6.97 version 2.5.0

- feat: implement linear-mixed effects models with R/lme4 (#55)
- ci: overhauled build process

6.98 version 2.4.15

- ci: fix rtd build
- fix: typo in infobox in analysis view (#69)
- fix: macOS builds did not use our pyqtgraph fork (reported in #70)

6.99 version 2.4.14

- maintenance release

6.100 version 2.4.13

- maintenance release

6.101 version 2.4.12

- maintenance release

6.102 version 2.4.11

- maintenance release

6.103 version 2.4.10

- maintenance release

6.104 version 2.4.9

- maintenance release

6.105 version 2.4.8

- maintenance release

6.106 version 2.4.7

- maintenance release

6.107 version 2.4.6

- maintenance release

6.108 version 2.4.5

- maintenance release

6.109 version 2.4.4

- fix: box filters selections bled to other filters (#67)
- build: do not use development version of dclab
- ci: move to GitHub Actions

6.110 version 2.4.3

- enh: improve UI (tooltips, icons)
- build: Windows build won't start due to breaking change in Windows ucrtbase.dll (#64)
- build: Migrate Windows build to PyInstaller 4.1

6.111 version 2.4.2

- fix: a regression introduced during refactoring in 2.4.0 made it impossible to select the temperature for known media
- fix: allow to reorder the dataset list (#46)
- enh: iconify buttons at top of Analysis View

6.112 version 2.4.1

- fix: strip white-space characters from line edits in the preferences to avoid line breaks introduced in the API key via copy-paste
- fix: not possible to create plots for datasets without the bright_avg feature; added a check for feature availability before updating the interface (#62)

6.113 version 2.4.0

- feat: allow to set Young's modulus parameters for all open datasets via the "Bulk action" menu (#60)
- fix: force temperature for Young's modulus computation to the value stored in the dataset for the "config" scenario
- enh: update menu keyboard shortcuts

6.114 version 2.3.0

- enh: added preferences dialog
- enh: improved user experience for searching DCOR data
- ref: migrate settings to QSettings
- setup: remove appdirs dependency, because we let QSettings determine where to store settings

6.115 version 2.2.6

- fix: support user-defined media introduced in dclab 0.29.1 (#53)
- fix: porosity box filter precision at lower limit (#58)

- docs: update bibliography
- bump dclab from 0.29.0 to 0.29.1

6.116 version 2.2.5

- setup: correct installation directory on Windows
- setup: bump dclab from 0.27.11 to 0.29.0
- fix: main window not focused after startup (#52)

6.117 version 2.2.4

- fix: UI slot issues (probably a racing condition) in the analysis view; editing the viscosity for "other" medium not possible on frozen Windows and macOS application (#49)

6.118 version 2.2.3

- fix: update Quick View plot when values for the Young's modulus computation are changed (#44)
- fix: use fixed point size in Block Matrix
- enh: improve plot matrix column/row visualization and show flow rate (#45)
- docs: update visualization of Young's modulus look-up tables using new figure script from dclab
- build: bump Python to 3.8 for Windows builds
- setup: bump dclab from 0.27.5 to 0.27.11 (improvement of hierarchy child handling)

6.119 version 2.2.2

- fix: correct box filter ranges when feature contains inf values
- enh: add pretty tool icons from KDE breeze theme

6.120 version 2.2.1

- fix: zero-division error when setting "one plot per dataset" for an empty plot (#41)
- build: pin PyQt5 to 5.14.2 on Linux/macOS

6.121 version 2.2.0

- setup: bump dclab from 0.26.2 to 0.27.5 (support for machine learning features)
- fix: modify the GUI to enable box filtering for ml_score_??? features
- ref: pipeline.get_features now only returns features that are available (previously all features were returned when no datasets were in the pipeline)

6.122 version 2.1.6

- fix: export file names were allowed to have invalid characters
- fix: Quick View showed filtered events that were not plotted (statistics were computed correctly though)
- fix: Quick View was not very informative when all events were filtered away - a message is now shown
- setup: bump dclab from 0.26.1 to 0.26.2 (fix bin num computation using Doane's formula)

6.123 version 2.1.5

- fix: marker hue feature data min/max was always computed even if not selected; for datasets without bright_avg (standard hue) this resulted in long waiting times
- fix: avoid pyqtgraph error messages in pipeline plots by disabling hovering events (not used anyway)
- fix: plot export did not work for single plots
- fix: exported event images sometimes had wrong contrast

6.124 version 2.1.4

- enh: add busy cursors wherever it seems fitting
- setup: bump dclab from 0.24.7 to 0.26.1 (volume-deformation isoelasticity lines, improved DCOR caching)

6.125 version 2.1.3

- fix: opening sessions did not work due to missing slot decorator
- fix: in some cases the temperature/viscosity setting was not correctly written to the UI for individual slots when switching between slots with fundamentally different settings (e.g. channel and reservoir)
- fix: KeyError when switching between slots with different temperature options
- setup: bump dclab from 0.23.0 to 0.24.7 (equidistant isoelastics, faster polygon filtering, crosstalk correction failed due to improper configuration update since Shape-Out 2.1.1)

6.126 version 2.1.2

- fix: ValueError when trying to compute contour spacing, because no datasets are selected for a plot
- fix: coloring scatter plot by feature did not work when the feature contains nan values
- fix: search path in session should not be filename but its directory
- setup: bump dclab from 0.22.5 to 0.23.0 (possible HDF5 export issue)

6.127 version 2.1.1

- fix: include private datasets when searching with DCOR API Key
- fix: remove UI option to compute Young's modulus for reservoir data (new "emodulus enabled" key in pipeline slot)
- fix: *Pipeline* object not returned by *session.open_session*

- setup: bump dclab from 0.22.1 to 0.22.5 (improve contour retrieval for mask images with artefacts, disable computation of Young's modulus for reservoir measurements)

6.128 version 2.1.0

- feat: basic DCOR support
- fix: improved session support (older pre-release sessions)
- docs: add DCOR guick guide
- enh: export event image via context menu in Quick View (#35)
- setup: bump dclab from 0.21.1 to 0.22.1 (DCOR support)

6.129 version 2.0.3

- setup: bump dclab from 0.20.3 to 0.21.1
- docs: add terminology section (#12)

6.130 version 2.0.2

- fix: IndexError when removing a plot (#36)
- fix: ValueError when filtering all events in Quick View (#37)
- fix: TypeError when removing datasets and opening Quick View (#38)
- fix: RuntimeError when removing datasets and opening Quick View (#38)
- fix: ValueError when removing activated datasets and opening Quick View

6.131 version 2.0.1

- fix: correctly distinguish prereleases when checking for new versions
- enh: allow loading data via drag&drop

6.132 version 2.0.0

- initial release

CHAPTER

SEVEN

BILBLIOGRAPHY

CHAPTER

EIGHT

IMPRINT/IMPRESSUM

8.1 Privacy policy

This documentation is hosted on https://readthedocs.org/ whose privacy policy applies.

CHAPTER

NINE

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

[BBN+23] Beyza Büyükurgancı, Santanu Kumar Basu, Markus Neuner, Jochen Guck, Andreas Wierschem, and Felix Reichel. Shear rheology of methyl cellulose based solutions for cell mechanical measurements at high shear rates. *Soft Matter*, pages –, 2023. doi:10.1039/D2SM01515C.

[HWT02] David Halpern, Howard B. Wilson, and Louis H. Turcotte. Gauss integration with geometric property applications. In *Advanced Mathematics and Mechanics Applications Using MATLAB, Third Edition*. Chapman & Hall, sep 2002. doi:10.1201/9781420035445.ch5.

[HMMO18] M. Herbig, A. Mietke, P. Müller, and O. Otto. Statistics for real-time deformability cytometry: Clustering, dimensionality reduction, and significance testing. *Biomicrofluidics*, 12(4):042214, 2018. doi:10.1063/1.5027197.

[Her17] Christoph Herold. Mapping of Deformation to Apparent Young's Modulus in Real-Time Deformability Cytometry. *ArXiv e-prints 1704.00572 [cond-mat.soft]*, 2017. arXiv:1704.00572v1.

[MOG+15] Alexander Mietke, Oliver Otto, Salvatore Girardo, Philipp Rosendahl, Anna Taubenberger, Stefan Golfier, Elke Ulbricht, Sebastian Aland, Jochen Guck, and Elisabeth Fischer-Friedrich. Extracting Cell Stiffness from Real-Time Deformability Cytometry: Theory and Experiment. *Biophysical Journal*, 109(10):2023–2036, nov 2015. doi:10.1016/j.bpj.2015.09.006.

[MMM+17] M. Mokbel, D. Mokbel, A. Mietke, N. Träber, S. Girardo, O. Otto, J. Guck, and S. Aland. Numerical Simulation of Real-Time Deformability Cytometry To Extract Cell Mechanical Properties. *ACS Biomaterials Science & Engineering*, 3(11):2962–2973, jan 2017. doi:10.1021/acsbiomaterials.6b00558.

[NUH+20] Ahmad Ahsan Nawaz, Marta Urbanska, Maik Herbig, Martin Nötzel, Martin Kräter, Philipp Rosendahl, Christoph Herold, Nicole Toepfner, Marketa Kubankova, Ruchi Goswami, Shada Abuhattum, Felix Reichel, Paul Müller, Anna Taubenberger, Salvatore Girardo, Angela Jacobi, and Jochen Guck. Intelligent image-based deformation-assisted cell sorting with molecular specificity. *Nature Methods*, 2020. doi:10.1101/862227.

[NUH+19] Ahmad Ahsan Nawaz, Marta Urbanska, Maik Herbig, Martin Nötzel, Martin Kräter, Philipp Rosendahl, Christoph Herold, Nicole Töpfner, Marketa Kubankova, Ruchi Goswami, and et al. Sorting real-time fluorescence and deformability cytometry (sort-fdc) - manuscript data. Dec 2019. doi:10.6084/m9.figshare.11302595.v1.

[ORM+15] Oliver Otto, Philipp Rosendahl, Alexander Mietke, Stefan Golfier, Christoph Herold, Daniel Klaue, Salvatore Girardo, Stefano Pagliara, Andrew Ekpenyong, Angela Jacobi, Manja Wobus, Nicole Töpfner, Ulrich F Keyser, Jörg Mansfeld, Elisabeth Fischer-Friedrich, and Jochen Guck. Real-time deformability cytometry: on-the-fly cell mechanical phenotyping. *Nature Methods*, 12(3):199–202, feb 2015. doi:10.1038/nmeth.3281.

[RPJ+18] Philipp Rosendahl, Katarzyna Plak, Angela Jacobi, Martin Kraeter, Nicole Toepfner, Oliver Otto, Christoph Herold, Maria Winzi, Maik Herbig, Yan Ge, Salvatore Girardo, Katrin Wagner, Buzz Baum, and Jochen

Guck. Real-time fluorescence and deformability cytometry. *Nature Methods*, 15(5):355–358, apr 2018. doi:10.1038/nmeth.4639.

[THO+17] Nicole Toepfner, Christoph Herold, Oliver Otto, Philipp Rosendahl, Angela Jacobi, Martin Krater, Julia Stachele, Leonhard Menschner, Maik Herbig, Laura Ciuffreda, Lisa Ranford-Cartwright, Michal Grzybek, Unal Coskun, Elisabeth Reithuber, Genevieve Garriss, Peter Mellroth, Birgitta Henriques-Normark, Nicola Tregay, Meinolf Suttorp, Martin Bornhauser, Edwin R. Chilvers, Reinhard Berner, and Jochen Guck. Detection of human disease conditions by single-cell morpho-rheological phenotyping of whole blood. *eLife*, 7:e29213, jun 2017. doi:10.1101/145078.

[XRM+20] Miguel Xavier, Philipp Rosendahl, Paul Müller, Maik Herbig, and Jochen Guck. Real-time deformability cytometry data of primary human skeletal stem cells and the human osteosarcoma cell line mg-63. Feb 2020. doi:10.6084/m9.figshare.11662773.v2.