

BioTracker User Guide

Introduction

This user guide aims to provide you with help to use the BioTracker software along with its currently available tracking and analysis modules. We will try to keep it up-to-date regarding new developments and changes. If you have any questions regarding BioTracker and its use, please contact us: [fubiotracker\[at\]gmail.com](mailto:fubiotracker[at]gmail.com)

Contents

1	Preface	3
2	Installation and prerequisites	3
2.1	Windows users	3
2.2	Developers and Linux users	3
2.3	System requirements:	3
2.4	Known issues	3
3	Using BioTracker	4
3.1	The main interface	4
3.2	Visualization options	10
3.3	Interacting with the tracking area	13
4	Tracking movement	14
4.1	Select input source	14
4.2	Images	15
4.3	Videos	15
4.4	Cameras	16
4.5	Annotations	17
4.6	How the arena is set and video pixels are calibrated to real-world measures	17
5	Selecting tracking modules	18
6	Background Subtraction Tracker	18
6.1	How detection of moving objects work	19
6.2	Defining detection parameters	22
7	Generated Output	22
7.1	CSV	23
7.2	JSON	24
7.3	Binary Serialization	24

8	Analyzing Data	24
8.1	File selection	25
8.2	Track Time Information	28
8.3	World Boundaries	29
8.4	Filtering and Smoothing	29
8.5	Inspect Data	29
8.6	Finalize	29
8.7	Results	30
9	Three simple recommendations for successful tracking	31
9.1	Rule 1	31
9.2	Rule 2	32
9.3	Rule 3	32
10	Developing BioTracker plugins	32

1 Preface

This user guide aims to provide you with help to use the BioTracker software along with its currently available tracking and analysis modules. We will try to keep it up-to-date regarding new developments and changes. If you have any questions regarding BioTracker and its use, please contact us: [fubiotracker\[AT\]gmail.com](mailto:fubiotracker[AT]gmail.com)

2 Installation and prerequisites

BioTracker was developed for Windows (7/8/10) and Linux.

2.1 Windows users

You can download the latest version of BioTracker at our website.

After downloading the Windows binaries, please unzip `BioTracker_V3.X.X.zip` to a destination of your choice. Open the folder and run “`BioTracker_core.exe`”. It runs out of the box without any installation process.

2.2 Developers and Linux users

You can build BioTracker from source. Please follow the instructions in the readme file.

2.3 System requirements:

- Hardware
- 500MB ~ 1024 MB of free RAM, depending on tracking algorithm and video resolution
- 200 MB of disk space
- Supported operating systems
 - Windows 10 (64 bit, binary)
 - Windows 7 (64 bit, binary)
 - Windows Vista (64 bit, untested, binary)
 - Windows 8 (64 bit, untested, binary)
 - Linux (32/64 bit, source only)

Note: CPU and RAM dimension will affect tracking performance, e.g., the speed of tracking (fps) depend heavily on your computer’s power.

2.4 Known issues

- Access rights can be an issue, particularly on Windows 10. Run BioTracker as administrator do avoid this or make sure it has suitable access rights for

everything it needs (files, cameras, etc).

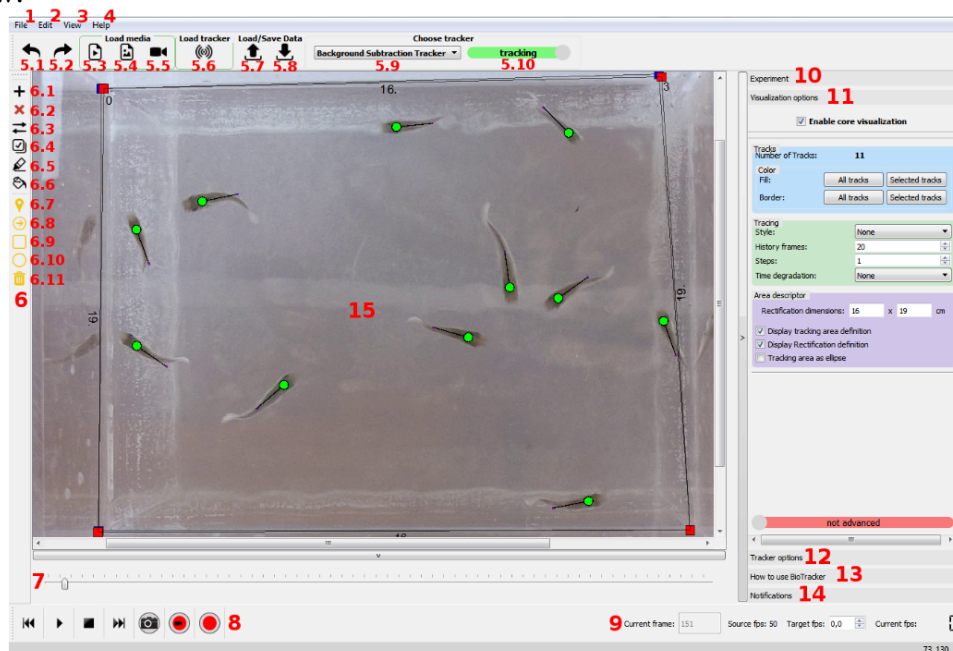
- On some Windows 10 Dell computers file dialogs crash frequently at random. This might be an issue of a shell extension or access rights.

For further issues and work in progress see our [issuetracker on github](#).

3 Using BioTracker

3.1 The main interface

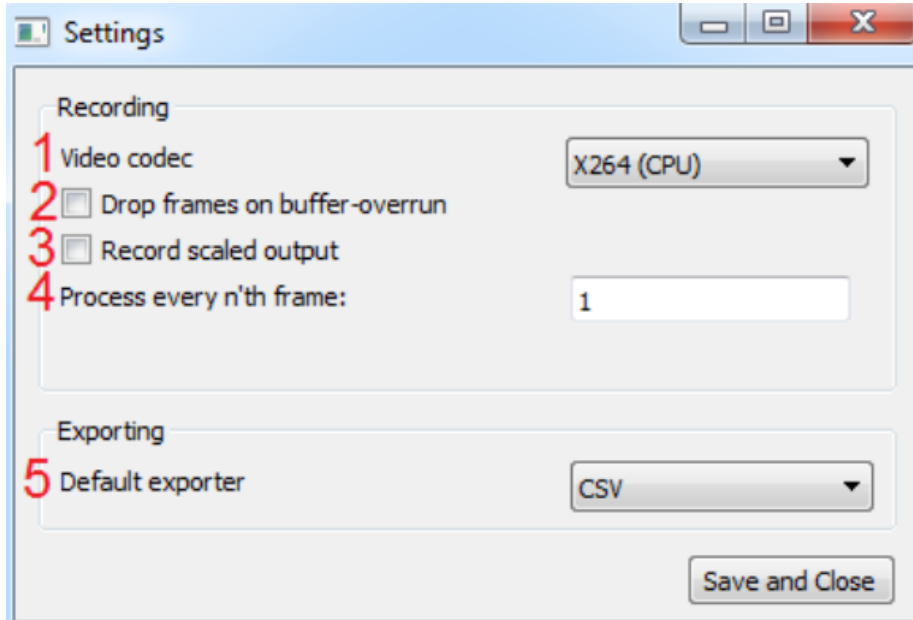
After you have opened BioTracker, you are presented the main interface window.



The main window of the BioTracker.

1. The “File” menu lets you select your input sources as well as tracking modules from a drop down menu. You may choose between a video file, a folder containing a set of pictures or a video stream. In the File menu you can also load additional trackers supplied as *.dll* (*Windows*) or *.so* (*Linux*) files. The binary of the biotracker comes with some trackers which are loaded automatically. These will appear in dropdown-box 5.9.
2. Via the “Edit” menu you can access the undo/redo functionality and the “settings” menu
3. The “View” menu offers toggles to show or hide the toolbars and the right and bottom panel.

4. In the “Extras” menu you can access the application settings, the application infos and the list of available shortcuts.

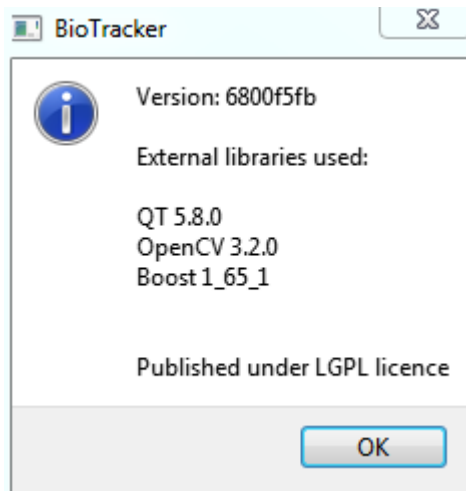


The Settings Window. Here you can apply configurations which are valid throughout the entire application. Changes will not take effect until restart of BioTracker.

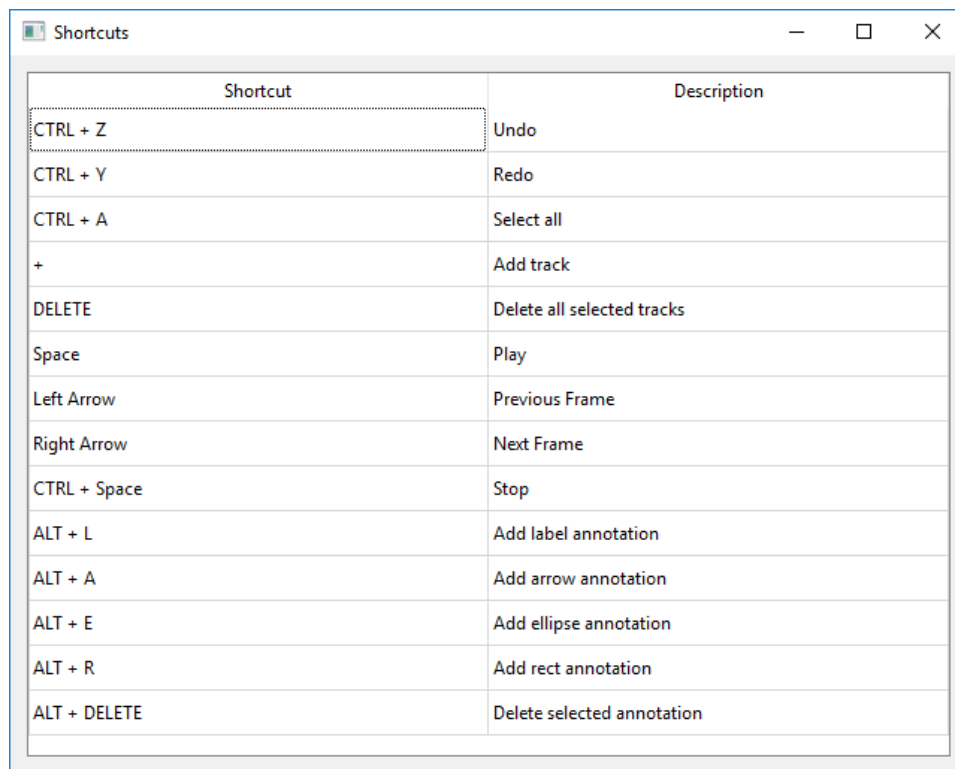
1. Video codec for both recording buttons in the main window. Individual codecs each have their own characteristics of speed, compression performance and usability. Do not confuse this with the videos container format - which will always be MKV (Matroska Video). Being able to compress the video in the BioTracker does not imply being able to playback or use it in other 3rd party applications. It is recommended to try out the toolset before relying on it.
2. When recording videos it might occur that frames come in faster than the BioTracker can turn them into videos. The default resolution is, that the BioTracker will wait until encoding is done. This will block the entire process, including tracking and acquisition of further frames, but assures that every frame which passed through the BioTracker is in the video. By clicking this checkbox this behaviour is changed into discarding any images not processed in time. This changes the tradeoff from integrity to performance (important for real time tracking applications).
3. This checkbox toggles what will be recorded in the Tracking Window with the last recording button.
4. The number N determines that only every N'th frame from the source video shall be processed. Hence the default N=1 means that every frame

will be processed. $N=5$ for instance means that only every 5th image in the source is processed, skipping its four predecessors.

5. The BioTracker writes the data provided by the tracking algorithm to a file. To do so it is using a generic file exporter which can be selected in this dropdown-box. Currently writing as CSV, JSON and binary files is possible. The CSV files are readable by every common version of Microsoft Excel and Open Office Calc. They will be stored in the “Tracks” subfolder of the BioTracker. Tracking output files are named according to the source video and the current timestamp.



The version information of the Biotracker, as well as other included libraries are displayed.



Shortcut	Description
CTRL + Z	Undo
CTRL + Y	Redo
CTRL + A	Select all
+	Add track
DELETE	Delete all selected tracks
Space	Play
Left Arrow	Previous Frame
Right Arrow	Next Frame
CTRL + Space	Stop
ALT + L	Add label annotation
ALT + A	Add arrow annotation
ALT + E	Add ellipse annotation
ALT + R	Add rect annotation
ALT + DELETE	Delete selected annotation

The list of all currently available shortcuts is displayed.

5. Controls

1. Undo action
2. Redo action
3. Load a video file
4. Load a set of pictures
5. Load a camera stream
6. Load a tracker
7. Load trajectory data from file. Needs to fit the currently loaded tracker.
8. Save the trajectory to a file
9. The dropdown-box lists all currently loaded tracking modules. Tracking modules located in the “Plugins” subdirectory of the BioTracker will be automatically loaded. You can select the tracking algorithm of your

choice here or load additional ones via the “File” menu. Doing so will add the tracking algorithm specific user interface in (12).

10. Start the tracking. If this is not activated, the tracking plugin will not be triggered every next frame. This switch automatically gets activated on starting an experiment.
6. Contains the tools to modify trajectories and entities and to add annotations to the current frame.
 1. Add a new trajectory to the data structure and show it on the medium
 2. Delete all currently selected trajectories
 3. Switch the ID of two selected trajectories
 4. Select all entities
 5. Change the border color of all selected entities
 6. Change the fill color of all selected entities
 7. Add a yellow label annotation to the current frame
 8. Add a yellow arrow annotation to the current frame
 9. Add a yellow rectangular annotation to the current frame
 10. Add a yellow elliptical annotation to the current frame
 11. Delete the currently selected annotation
7. This slider lets you seek in a video file or set of pictures once a source video is loaded. This is not available when using a camera (live stream).
8. This is the mediaplayer that grants you control over the source video. These buttons will be greyed out and dysfunctional when no source video is selected. The first button steps one frame in the source video back. The “play button” plays back the video as fast as possible, to the limit of the source fps. It will turn into a pause button while playing so you can pause playback at any time. The stop button stops the source video and resets it to frame 1. The next button skips to the next respective frame. The next three buttons will help you save the things you see in the tracking view to files. The camera button saves the currently displayed image to a file in the subfolder “screenshots” in

the biotracker directory. The record button with the camera symbol will record the source video without any overlays. I.e. if you are running a tracker which is currently displaying trackpoints on top of the video, these will not be saved in the video recorded by this button. The last button records a video of what you see, including the previously mentioned tracking overlay. By default, it records the video and overlay at original resolution. You may change this in the “Settings Window” to record the scaled output, I.e. you can set it to record exactly what you see, zoom included. Videos are saved in the “Videos” subdirectory of BioTracker.

9. Displays source fps and currently played fps. The source fps is the fps defined by the source video. This means that if you are using a camera stream which is configured to run at X fps, X will be the source fps. However, tracking can be slow depending on your hardware, source video and tracking algorithm. For this reason the BioTracker might not be able to keep up to the source fps and run slower. This is not harmful to integrity of tracking results. But if you are tracking directly from camera stream, some frames will be dropped if the current fps are less than source fps. There is also an input method to put an upper limit to the playback speed of the BioTracker.
10. Selecting a tracker using (5.9) will not start tracking procedure yet. To do so, tick the button (5.10) and start playing the video. Alternatively, just click “start experiment” in the experiment tab (10). For details on how to use the individual tracking module, see the sections on them. The “Finalize trial” button will finalize the current tracking session and trial. For CSV output this means the csv file is re-written in chronological order and replaces the current file. The file handle will be released so the operating system can move the file away. In any case the trajectory data structure will be cleared and all tracks removed. This effectively starts an entirely new tracking session on the source video.
11. In this tab you may adjust what and how the BioTracker shall display things to you. See visualization options.
12. Tracker: Every tracking module will have its specific user interface which is displayed here. For details the the section for the corresponding tracking plugin.
13. How to use BioTracker: Shows a short manual on how to use the BioTracker stepwise.

14. Shows notifications in a text browser. These can also be created by the tracker.
Colors:

- Messages (black)
- Warnings (orange)
- Errors (red)

3.2 Visualization options

Visualization options

1 **Enable core visualization**

2 **Tracks**
Number of Tracks: **2**
Color
Fill:
Border:

3 **Tracing**
Style:
History frames:
Steps:
Time degradation:

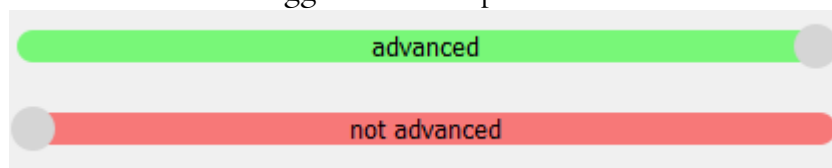
4 **Area descriptor**
Rectification dimensions: x cm
 Display tracking area definition
 Display Rectification definition
 Tracking area as ellipse

The visualization options tab

1. Check this box to have the tracking visualized by the core. Not displaying it at all may hinder your work with the tracker, but slightly improve performance.
2. This box provides access to configurations on how the individual entities are visualized over time and which shape. It also provides some general access functions
 - The “number of tracks” displays how many tracks are currently processed.

- (Color subbox) In the first row you can set a color for the inner area of a shape, either for the selected set or for all of them at once.
 - (Color subbox) In the second row you can set a color for the borders of a shape, either for the selected set or for all of them at once.
3. A trace is a visualization of a subjects position of the previous timesteps. In this view you may adjust this history and whether to see it at all.
- Tracing style denotes the type of marker. Available are points, arrowed lines and straight lines.
 - Number N of past trackpoints to be visualized for each track. E.g. setting this to 5 will cause it to visualize the past 5 trackpoints using the method from (a).
 - Here 1 means every timestep will be visualized in the trace. 2 means every second, etc.. The total number of trackpoints will be the number specified in (b) divided by the number specified in (c).
4. The “Area descriptor” box offers functionality to adjust rectification and tracking area.
- Define the size (width and height) of the tracking area.
 - Toggle visualization of rectification rectangle
 - Toggle visualization of tracking area shape
 - Toggle tracking area shape between rectangular and ellipsoid shape.

There is a switch to toggle advanced options.



The enabled and disabled switch to toggle advanced options in the “Visualization options” tab

Visualization options

Enable core visualization

Tracks
Number of Tracks: **2**

Dimensions
Set for:

Height:

Width:

Show orientation line
 Show track ID

Color
Fill:
Border:

Tracing
Style:

History frames:

Steps:

Time degradation:

Show frame number

Tracer dimensions
Tracer proportions:

Show orientation line

Area descriptor
Rectification dimensions: x cm

Display tracking area definition
 Display Rectification definition
 Tracking area as ellipse

Miscellaneous
 Enable antialiasing for entities
 Enable full antialiasing

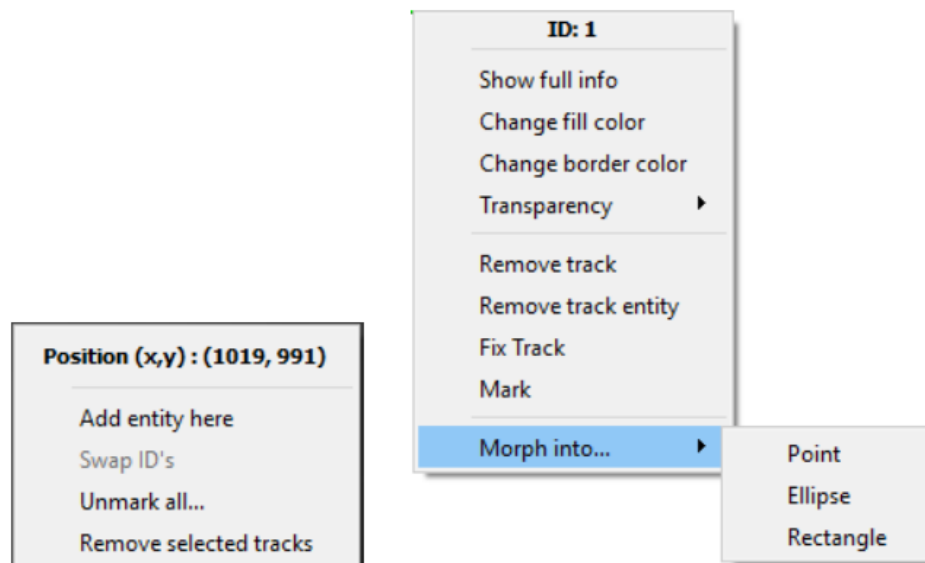
Advanced options of the options tab

I. (“Dimensions” groupbox)

- Dimension setter buttons for all or selected tracks
 - Button to revert the dimensions of all tracks back to its defaults
 - Checkbox to toggle entities orientation lines
 - Checkbox to toggle ID in the center of each entity
2. (“Tracer dimensions” groupbox)
- Spinbox to control the tracer size by ratio to entity (0,5 means half size)
 - Checkbox to toggle the orientation line of each tracer
3. (“Miscellaneous” groupbox)
- Checkbox to toggle antialiasing for all entities (at the cost of performance)
 - Checkbox to toggle antialiasing for the whole media view frame (at the cost of performance)

3.3 Interacting with the tracking area

You may select an entity by left clicking it. All selected entities will be highlighted by a solid outline. Multiple entities can be selected by left click into the view and drag the mouse to select a set of entities. Entities can always be added to the selection by holding the left “ctrl” key while clicking the entity.



Menu appearing when right-clicking the tracking area

Upon right click on free area you will presented the menu as seen on the left hand side.

- “Add entity here” will add a new entity at the currently clicked point.
- “Swap ID’s” will swap the ID’s of two selected entities. Only available if exactly two are selected.
- “Unmark all...” will clear the current selection.
- “Remove selected shapes” removes the currently selected shape or shapes.

Right click on an entity will yield a menu as seen on the right hand side.

- “Show full info” opens up a Window with information about the entity in this frame
- “Change fill color” changes the color of the entities area
- “Change border color” changes the color of the entities area border
- “Transparency” changes the transparency of the entity
- “Remove track” removes the whole track, no matter whether selected or not.
- “Remove track entity” only removes the right clicked entity, no matter whether selected or not.
- “Fix Track” fixes the data structure of the track, so the tracker does not change it (but you can)
- “Mark” and “Unmark” mark or unmark the right clicked entity, respectively.
- The “Morph into...” menu changes the type of the entity into compatible types. Current compatible types are point, rectangle and ellipse.

4 Tracking movement

4.1 Select input source

To start your tracking project, you need to select a source which can be a picture sequence, a previously recorded video file or a camera stream.

4.2 Images

A set of images may be tracked using the context menu “File → Open Pictures...”. In the file dialogue multiple pictures may be selected, which will be tracked sequentially according to their filename ascendingly.

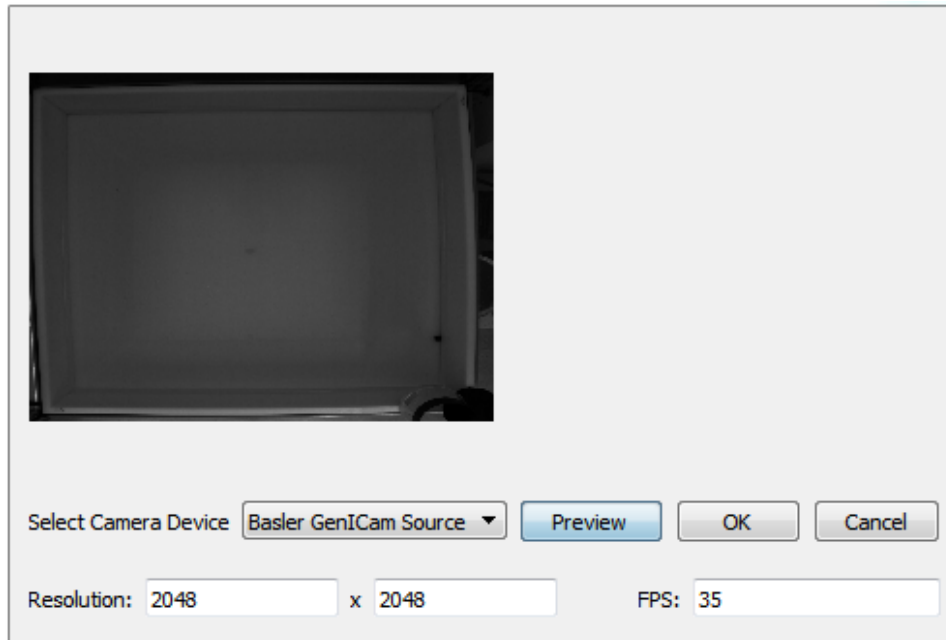
The software uses OpenCV 3.2 to decode videos and images. Hence the supported formats for images are as per documentation.

- Windows bitmaps - .bmp, .dib
- JPEG files - .jpeg, .jpg, .jpe
- JPEG 2000 files - .jp2
- Portable Network Graphics - .png
- Portable image format - .pbm, .pgm, .ppm
- Sun rasters - .sr, .ras
- TIFF files - .tiff, .tif

4.3 Videos

A video may be tracked using the context menu “File → Open Video...”. The framerate will be determined from the video file itself. The BioTracker only include free codecs for reading. However it supports most common container formats, such as mp4, avi or mkv. To read commonly used proprietary codecs, such as X264, you need to install these codecs from a third party. On Linux these can be acquired via repositories and are supplied via gstreamer. On Windows these can be acquired by downloading and installing codec packs and provided via directShow.

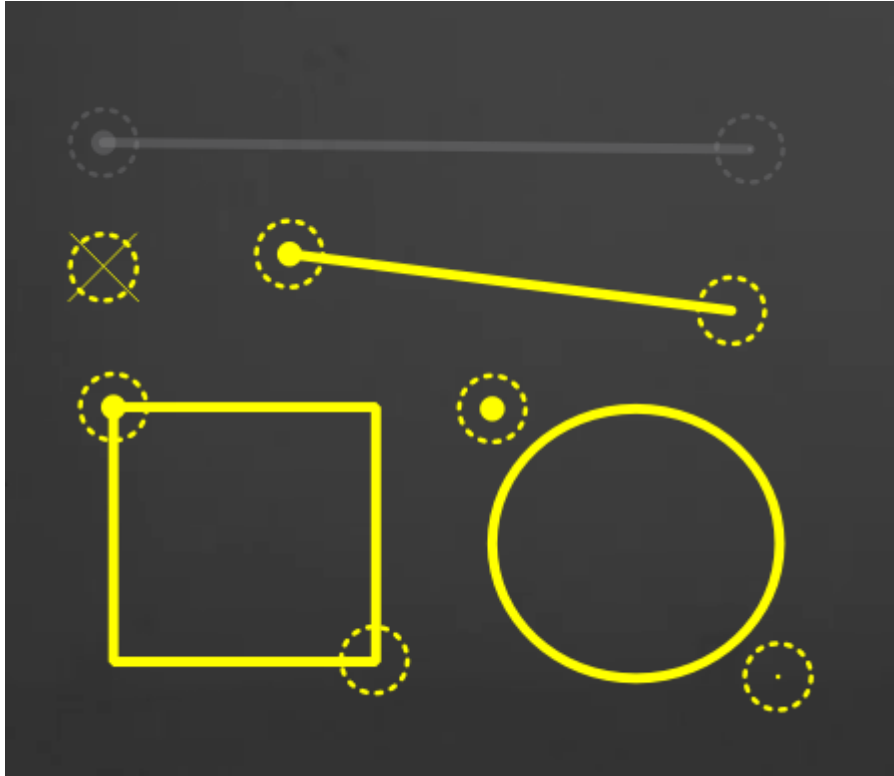
4.4 Cameras



Selecting “File → Open Camera...” in the context menu opens a form for selecting and configuring the camera input.

Any generic USB camera may be used for displaying and tracking. “Preview” will yield a preview image similar the example under given configuration, if the camera could be loaded properly. The preview image is stretched to a smaller resolution and might not retain the given aspect ratio. Resolution and FPS values “Default” mean the camera will not be reconfigured and remains at whatever configuration it comes up at loading. This enables the user to set these properties in an external tool and just leave them like that for use in the BioTracker. Any other values will be set to the camera. Faulty values will result in failing to open the camera for recording.

4.5 Annotations



Annotation examples.

The user can manually annotate videos and images with vectors or labels. The annotations will be saved to a .csv file that corresponds to the image or video filename with the added suffix “.annotations.csv”. Each annotation will be fixed to the frame it is created in. In other frames it will be greyed out. Annotations have circled handles which can be used to change the position of the annotation. To remove an annotation click on one of its handles, which will be marked red, and hit the ALT + delete keys or use the “Delete selected annotation” tool in the toolbar (6) To place a vector press ALT + A, then left click into the video to start the arrow. While keeping the mouse button pressed, drag the cursor to the end point of the vector and release the mouse button. To place a label press ALT + L, left click into the video and release the mouse button. Press ALT + R to create a rectangular and ALT + E to create an elliptical Annotation. There also exists a tool in the left toolbar (6) for each type of annotation.

4.6 How the arena is set and video pixels are calibrated to real-world measures

Upon loading a video file and enabling “display of rectification” and “tracking area” in the view tab, four blue squares will appear connected by fine black lines. This rectangle defines the rectification. Often the camera view is not perfectly

orthogonal to the recorded area. In fact the camera could be viewing the tracked area from an arbitrary angle and distance. This original image, which is perspectively distorted, is the input to the tracker. The tracker wants to estimate - according to its algorithm and limitations - an object's position in the source video. To do so it is useful for the tracker to use real world distances instead of pixels to get a realistic measure for distances. The user might be interested in an object's real world coordinates instead of pixels as well. To achieve this a rectification is necessary. In the BioTracker a rectification is defined via four points of known distance in the real world. These four points define a coordinate system in centimeters which can be used by the tracker and which are output for the user. Note that when recording at an angle, a rectangle in the real world might look distorted on the video. This can be corrected by selecting the corner points of the distorted rectangle for the rectification. The red corners define the tracking area. This definition is passed to the tracking plugin, which may use it to check whether a point is in- or outside the given area. This way points in irrelevant portions of the source video may be ignored to speed up tracking. Note that this behaviour is tracker dependent.

5 Selecting tracking modules

There are several tracking modules currently available and others will be available in future as they can be developed independent of the main BioTracker software. The main package of BioTracker already contains two tracking modules while other can be downloaded at [github releases page](#).

The following tracking modules are currently available:

- Background subtraction tracker
- Lucas-Kanade-tracker

6 Background Subtraction Tracker

The Background Subtraction Tracker (BST) is a tracker which relies on finding objects by telling them apart from the background. Here background is defined as all the things which do hardly or not at all move. Hence this algorithm is not able to track objects which remain still for a longer period of time. Also it fails to distinguish objects once they are in too close vicinity.

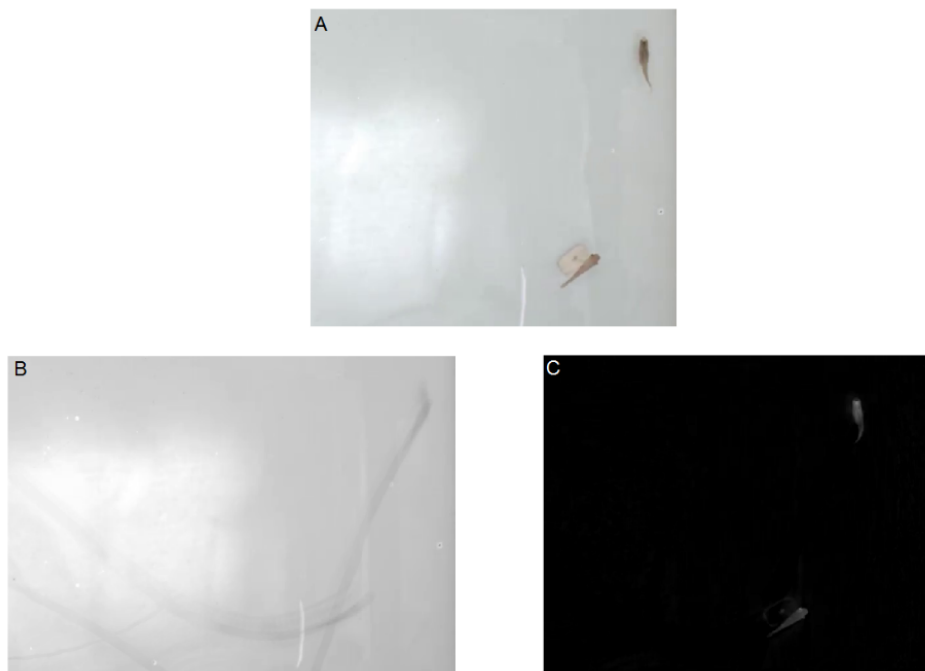


Sample of a tracking situation of two guppies

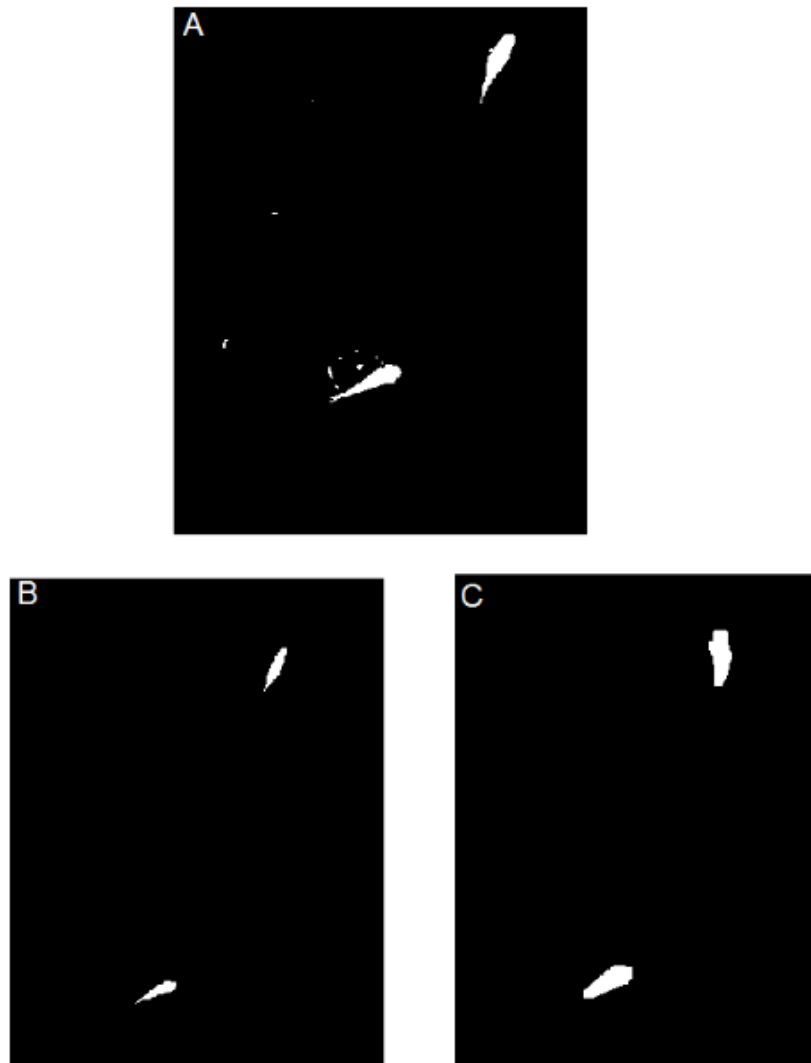
6.1 How detection of moving objects work

The BST detects moving objects in a step-wise manner which can be summarized as: Background Determination, Binarizing, De-noising, Detection

First, the background is determined. Original image A is separated into background B and foreground C.



The foreground is then binarized into absolute black and white as depicted in (A)



De-noising steps

Afterwards, de-noising steps Erosion and Dilation are operated. Erosion (B) reduces the size of every white portion of the image and thus reduces small noise by eliminating the dots entirely. Dilation widens the remaining white portions again which results in larger white objects and closes holes in them, see (C). The dilated image is then used to detect objects by fitting ellipses around the largest white blobs. Objects are recognized to be the same as in the previous image by euclidian distance.

6.2 Defining detection parameters

BinThreshold	15	▲▼	1
Size Erode	8	▲▼	2
Size Dilate	8	▲▼	3
Background Ratio	0,050000	▲▼	4
Min Blob Size	40	▲▼	5
Max Blob Size	10000	▲▼	6
Send Image:	Original ▼		7
	Reset Background		8

The user interface (Parameter View) of the Background Subtraction Tracker

1. Threshold for binarizing into black and white, where 0 is black and 255 is white. Here everything above 15 will be white.
2. Size of denoising in pixels. For low noise and low resolution low values are recommended, higher ones otherwise.
3. See (2). Usually chose the same as (2).
4. The rate at how fast the new images merge into the background. “New Background” = $(1 - \alpha) * \text{“Old Background”} + \alpha * \text{“Next Image”}$.
5. Minimum size for an object (in a connected area of pixels) to be acknowledged as an object.
6. Maximum size according to (5).
7. Select a certain picture to be displayed. Each step of detection described in this guide can be viewed live. Use this to check on your choice of parameters.
8. Resets the background to the current input image. This makes foreground equal to background.

7 Generated Output

The BioTracker 3 is designed to provide generic output methods for any tracker. The output method can be selected in the settings view. Currently CSV, Serialize (as binary) and JSON are available. All of them rely on the tracker defining what kind of data each node, i.e. each trackpoint stores. This is generic in nature, so the actual data found in the output depends on the tracking plugin being used.

Example: Data each trackpoints holds in the Background Subtraction tracker

- **FRAME:** Frame number in the source video.

- **MillisecsByFps:** Time in the source video since beginning of the video or camera stream. This is $((\text{"Frame No."} / \text{"Source FPS"}) * 1000)$. Note that if frames from a camera stream have been dropped (current fps < source fps) for any timestep, this may gain inaccuracy. Also note that for videos and pictures frames will never be dropped.
- **objectName:** The name assigned to an object. The Background Subtraction Tracker does not assign particular names.
- **valid:** The BioTracker itself does not delete tracks, but marks them invalid. If a track was made invalid, all its entities will be invalid. They will still be written for your reference and later loading.
- **id:** ID assigned to the track in the userinterface
- **coordinateUnit:** Units in which x and y are formatted
- **x:** x position of the entity. Whether this is px, cm or something else defined by the tracker. In the BST it is defined by coordinateUnit
- **y:** y position of the entity. Same as x.
- **rad:** Rotation of entity in radians.
- **deg:** Rotation of entity in degree.
- **xpx:** The X-Position in pixels
- **ypx:** The Y-Position in pixels
- **time:** Timestamp in unix time, ms since epoch (1.1.1970, 00:00)
- **timeString:** Same as time, but converted to a human-readable timestamp.

7.1 CSV

The exporter for tracking data generically outputs csv data from tracked trajectories. Details are always given in the csv header (which prints all the headers as listed in the previous example) and the documentation of the tracking module. Additionally before all the tracking output is written, a couple of lines of metadata are written. These are preceded by a # and contain information about the original video, the FPS of the video and the generation time.

7.2 JSON

JSON is just like CSV a text-based semi-human-readable format for writing data. This format is more suitable for automated processing and reading by specialized software.

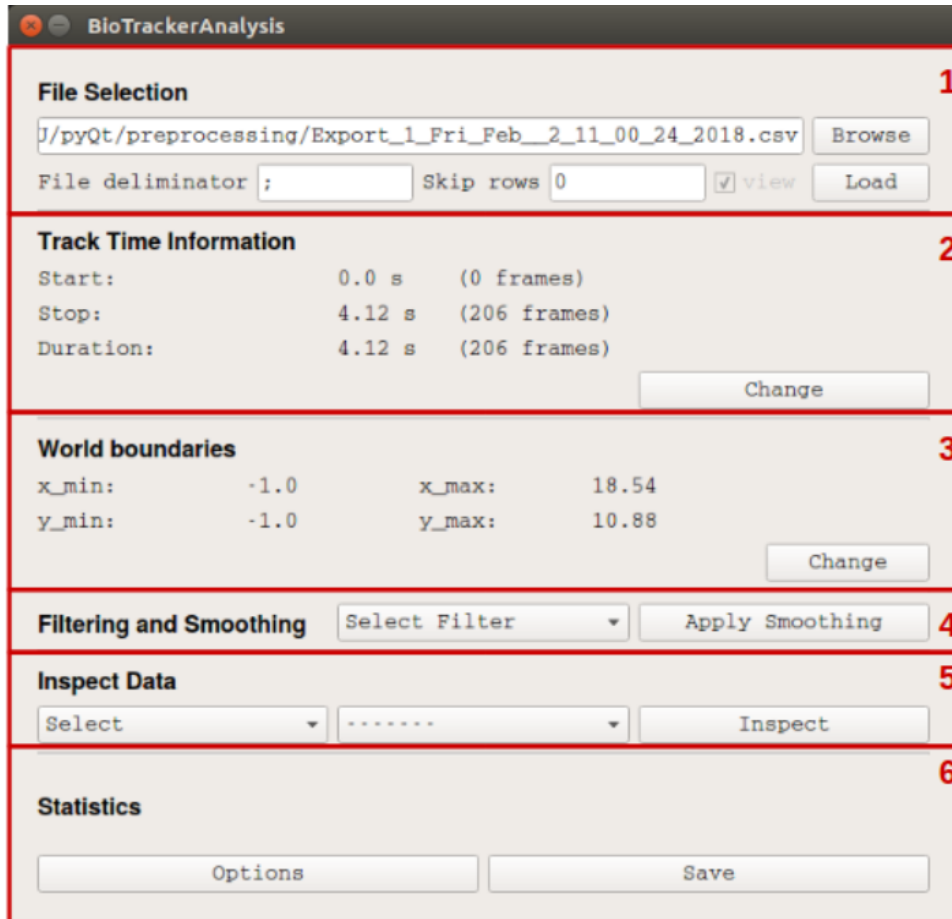
7.3 Binary Serialization

This format is non-human-readable. It is a binary serialization of the data in each trackpoint using the QT datastreams. A suitable format for use within the Biotracker but hardly interpretable by any other software.

8 Analyzing Data

“Along with the BioTracker comes a python tool named “Data Analysis Module” to evaluate CSV output of the trackers. The simplistic user interface and provides an easy way to calculate a number of metrics for arbitrary CSV files by using user annotated columns, which is done automatically for BioTracker conformate output. It can then calculate metrics like speed, inter-individual distance and transfer entropy for all individuals, pairwise respectively and write them back to new CSV files. There is also the option to filter arbitrary columns.

8.1 File selection



Loading a CSV file for analysis

- “Browse” will open a file selection dialogue which allows the user to choose a raw data file. Only accepted type is CSV.
- “File delimitator” column separator of the CSV file. By default this is a comma. Other accepted separator are semicolon or tab.
- With “Skip rows” the user can choose to skip the first n rows of the raw data file. By default this value is set to 0 (no skip). Lines beginning with # are interpreted as comments and will always be skipped. Also any column names or headers will be ignored as well as rows with missing values. Missing values means, if there is less values than indicated by the header. Empty, i.e. no text between two separators, is not considered missing.
- If “view” is unchecked the data display is skipped and data is loaded automatically using the latest defined column numbers/names. This option should only be used when analyzing csv files with the same column order because loading a csv file with the wrong column order may cause errors. It is generally recommended to have “view” checked and not skip the column selection.

- “Load” opens a display of the raw data if “view” is checked, where the user can assign names to the columns (see 1.1). If “view” is unchecked the data is loaded directly to the interface.

The screenshot shows the BioTrackerAnalysis window with a data table and configuration options. The table has 8 columns labeled 1 through 8. The data rows are as follows:

	1	2	3	4	5	6	7	8
1	0	0.0	0.0	True	3.0	cm	4.5526580...	7.
2	18	360.0	0.0	True	3.0	cm	-1.0	-1
3	19	380.0	0.0	True	3.0	cm	17.507194...	4.
4	20	400.0	0.0	True	3.0	cm	17.562034...	4.
5	21	420.0	0.0	True	3.0	cm	17.576349...	4.

Below the table, there are configuration options:

- frames: 1
- time (ms): 2
- fish_x: 7
- fish_y: 8
- fish_angle (deg): 9

At the bottom, there are three buttons: "Add Parameters", "Add/Remove Agents", and "OK".

Selecting the format of the CSV file.

1.1. Display section: shows the raw data, excluding comment lines, headers, and rows the user decided to skip. Only the first N lines will be shown to avoid delays.

1.2. Column selection area: allows to attribute columns of the original datafile to the relevant variables. In the current example this means that the frame numbers are in column 1, time-stamps in milliseconds in column 2, the x-position-component of the tracked animal in column 7 etc. To change these settings see 1.3.1 and 1.3.2

1.3. Change and Finalize area:

- “Add parameters” opens a menu where the time and angle format can be set (see 1.3.1)
- “Add/Remove Agents” opens a menu where agent can be added removed and renamed
- “OK” Loads the selected columns for analysis, saves settings and returns to the main window.

The screenshot shows a window titled "BioTrackerAnalysis" with the following settings:

- Time:**
 - frames datetime
 - milliseconds seconds
- Angle representation:**
 - deg rad
- Additional categories:**
 - Region
 - RoboMode

An "Apply" button is located at the bottom of the dialog.

Setting miscallenous options for the to be loaded CSV file

1.3.1

Check boxes allow to set the desired time format (datetime, milliseconds or seconds) and angle format (ddegree (deg) or radiant (rad)). Also additional categories can be selected but are currently not processed. Clicking “Apply” will save these settings and return to the table view.

The screenshot shows a window titled "BioTrackerAnalysis" with the following settings:

- Agents:** A text input field containing the number "2" and a "Change" button.
- Names:**
 - agent0: fish
 - agent1: agent1

An "OK" button is located at the bottom right of the dialog.

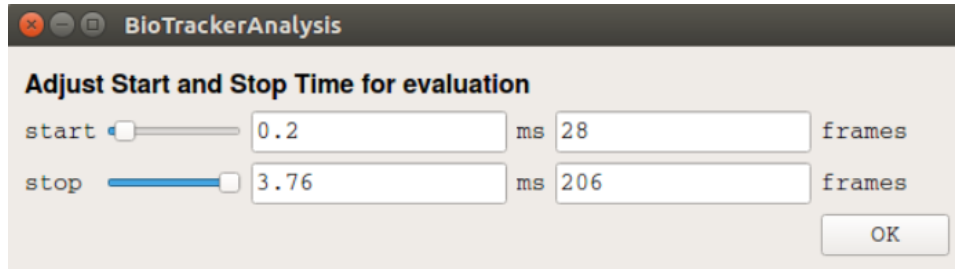
Selecting amount and names of agents.

1.3.2

The line called “Agents” allows to specify the numbers of agents to be analysed. Clicking the “Change” Button enables the user to give meaningful names to

these agents, otherwise they will have the default names `agent0`, `agent1`, etc. Clicking “OK” saves the changes and returns to table window.

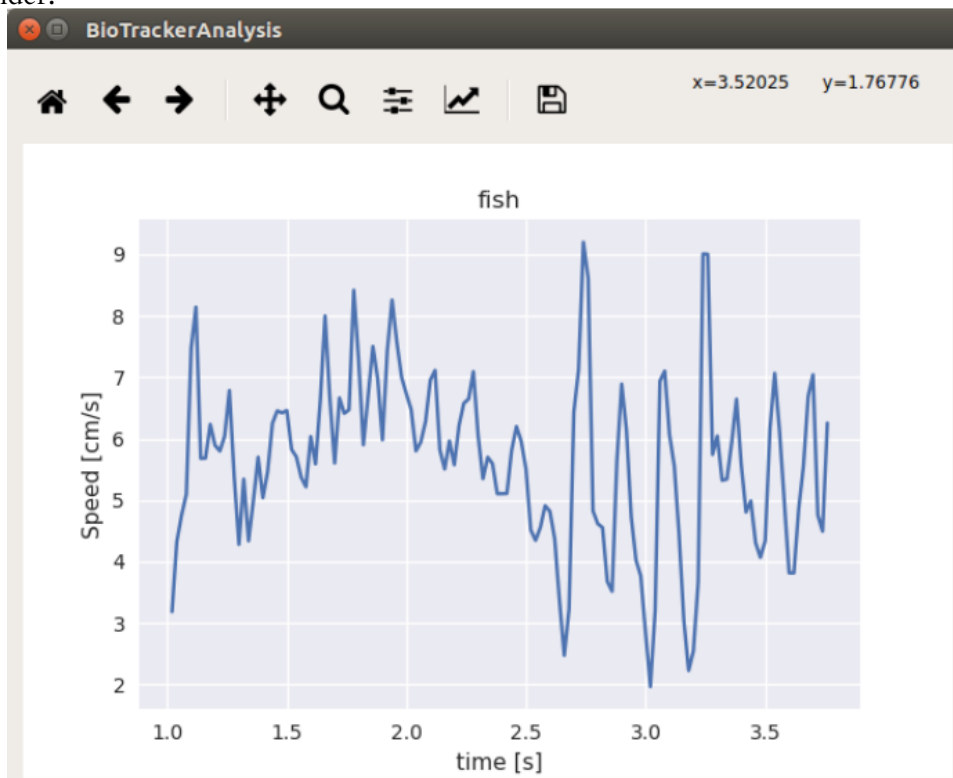
8.2 Track Time Information



View for changing start and stop time of observed data.

This window displays start time, stop time and duration of experiment (in second and frames) and allows to change the relevant time range. Independent of the selected time format the display here only used seconds (s).

“Change” Opens a window where start and stop time can be adjusted with a slider:



Example of plotting an agents speed over time

8.3 World Boundaries

Display the minimal and maximal x and y values of the raw data and allows to change the relevant region. All displayed values are interpreted as cm.

“Change” allows to edit the boundary values to e.g. perform analysis in a subregion.

“Add Subregions” opens a window where subregions of the main area can be defined. For those areas the same analysis is performed as for the main region, results appear as additional rows in info.csv (see Results section).

8.4 Filtering and Smoothing

“Select Filter” allows to select a filter (currently only median filter with $k=5$).

“Apply Smoothing” applied the selected filter component to the x and y component of each agent.

8.5 Inspect Data

Allows to generate plots of the data with respect to the currently selected spatial and temporal boundaries. The plots will show in a separate window and can be edited and saved from there.

1st dropdown menu allows to select the type of plot to be generated. Available options are: “Trajectory”, “Timeline”, “Histogramm” and “Boxplot”.

2nd dropdown menu depends on the selection in the first. For example Timelines are available for the parameters “speed”, “distance” and “angle”.

“Inspect” opens a display window with the desired plot.

8.6 Finalize

“Options” opens a window with three tabs: “Plots”, “Folders” and “Transfer Entropy”

- “Plots” allows the user to select plots that will be automatically saved with the results files. Options are all, none or individual selection - “Folders” provides an overview of currently used default folders e.g. for data or results - “Transfer Entropy” allows the user to select whether she wants to calculate Transfer Entropy for the currently selected data. Since this calculation is only applicable for two agents and can require considerable time and computing power, it must be explicitly selected by the user.

“Save” opens a folder selection dialogue allowing the user to specify a results directory. A folder called “BioTrackerAnalysis” + current date / number (e.g BioTrackerAnalysis_2018_02_22/008) will be created. This files in this folder are (1) timelines.csv (2) info.csv (3) plots. For further description see below.

8.7 Results

Info.csv contains basic data and parameters concerning the experiment setup, single agents and pairs of agents. In the following these results are described in more detail:

1.a. general information

Source	Name of the original datafile
x_min	Minimum value of all x-positions
x_max	Maximum value of all x-positions
y_min	Minimum value of all y-positions
y_max	Maximum value of all y-positions
start	Start time in seconds
stop	Stop time in seconds
filtered	“True” or “False” depending on whether filtering was performed

1.b. Agent specific information

trajectory_length	Total length of trajectory covered
speed_mean	Mean of agent’s speed
speed_var	Variance of agent’s speed
speed_min	Minimum value of agents speed
speed_25%	25 percentile of agent’s speed
speed_median	Median of agent’s speed (i.e. 50 percentile)
speed_75%	75 percentile of agent’s speed
speed_max	Maximum value of agents speed

1.c. Information about pairs of agents

Key	Interpretation
dist_mean	Mean of distance
dist_var	Variance of distance
dist_min	Minimum of distance
dist_25%	25 percentile of distance
dist_median	Median of distance (i.e. 50 percentile)
dist_75%	75 percentile of distance
dist_max	Maximum value of distance
closer_5cm_(s)	Time the agents spent closer to each other than a threshold distance (default: 5, 10, 15 and 20 cm)
closer_5cm_(%)	Percentage of the selected time range

Key	Interpretation
	that agents spent closer to each other than a threshold distance (same default values as above)
Correlation of speeds	Not yet implemented

2. Timelines.csv contains for each frame in the selected spatial and temporal range (as determined by x_min, x_max, y_min, y_max, start and stop in info.csv) the following values:

frame	Number of current frame
time	Timestamp of current frame in the original format
seconds	Timestamp of current frame in seconds
agent_x	X-Coordinate of agents position (using agents given name)
agent_y	Y-Coordinate of agents position (using agents given name)
agent_angle	Current angle of agent in the selected format i.e rad or deg
agent_vx	X-Component of agents velocity
agent_vy	Y-Component of agents velocity
agent_speed	Agents speed calculated as $\text{agent_vx}^2 + \text{agent_vy}^2$.
agent1/agent2_dist	Distance between two agents calculated via $\text{sqrt}((\text{agent1_x} - \text{agent2_x})^2) + \text{sqrt}((\text{agent1_y} - \text{agent2_y})^2)$

9 Three simple recommendations for successful tracking

In this part we would like to give some simple recommendations on how to plan and execute experiments that produce trackable videos.

9.1 Rule 1

First of all, if the subjects you want to track with BioTracker (or any other tracking software) is not sufficiently different in coloration from the background, a proper tracking will most likely fail. As an example, a white mouse recorded from above in a white-bottomed cage is a bad experimental choice. Most tracking algorithms use some kind of background subtraction, which basically requires that the subject's pixels are distinguishable from the background pixels. Also, tracking will be hard if the subjects were not moving. In this case, many algorithms that compare frame by frame which is the background (stable) and which is the subject (unstable) will fail to tell apart subject from background even if coloration differs. This also means that a moving, unstable background is bad for proper tracking outcomes. So, we recommend to design experiments in which subjects differ in coloration from the background subjects move (even slowly)

the background is stable (not moving, changing). In many cases, experimenters record their whole experiment, which is a good thing and recommended as good scientific practice. However, if the experimenter introduces the animal into the arena, hand or fingers (or dipnets or cages) are visible in the video and thus hamper a proper tracking. We thus recommend to either cut recorded videos to the essential parts - the phase in which the the animal should be tracked - or, depending which tracking module is loaded in BioTracker, to fast forward to those parts and only then initiate tracking.

9.2 Rule 2

Second, videos are a bunch of pixels (every digital picture consists of these small color squares) and tracking softwares just look for those pixels but not for cm or any other “real-world” measures. Thus, one has to tell the computer what size a pixel is in the real world. As this measure varies depending on your recording mode (camera-object distance, video resolution as well as recording quality), you have to have something in the video of known size you can use to calibrate pixels to real world measures. In the BST tracking module, a rectangle can be drawn upon points of known distance in the video and thus pixels are converted to real world measures.

We thus recommend to always put some kind of grid of known dimensions in the recording frame. This does not necessarily need to be inside the arena but well visible in the video.

9.3 Rule 3

Third, we just want to stress that recording orthogonal to the arena’s bottom makes life easier. If you record your subjects with an angle, a simple calibration of pixels to real-world measures is not possible that easily anymore.

10 Developing BioTracker plugins

For building a sample, see the Sampletracker repository. This is a full example doing some thresholding on an input image.

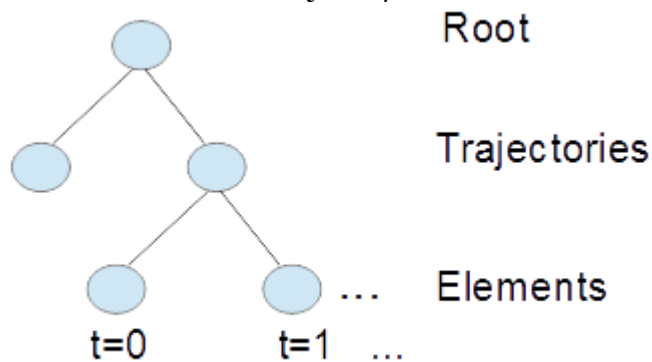
By copying the sample project you are given the MVC structure and can start coding right away. Nevertheless, here is a short introduction to the structure.

You can get started with the `BioTrackerTrackingAlgorithm::doTracking` function, which does the actual tracking, in the case of the sampletracker thresholding. The other classes are related to housekeeping tasks: Wrapping GUI input and structuring the MVC pattern.

The interface project provides you with more useful interface classes to build a MVC patterned tracker. The basic things a plugin must do are the following: Provide the core with two “IView” objects, one representing the Parameter and

Element view. Then it needs a Qt slot to receive frames and a Qt signal to send a frame back to the core.

After doing your tracking you need to send back a matrix for displaying by the core. For instance, you might have created a binarized version of the input and want it displayed. If you want to have the trackpoints rendered onto the original image, just send forth the matrix as you received it and store the trackpoints in the “Tracked Trajectory”. This is basically a tree structure storing all the tracks. Make sure your tracked elements are inserted at the corresponding point in time and that the trajectory is marked “valid”.



Schematic of the trajectory structure.

Once you are done with any tracking relevant task finalize it by emitting the “TrackingDone” signal. This tells the core it may now proceed to render the frame, load the next frame and in consequence call the tracking plugin again. Hence be careful with any task done after emitting the signal for concurrency reasons.