FishInspector

User Guide

FishInspector Version 1.03

A software to annotate phenotypic features of zebrafish embryos

TKS3 - Tobias Kießling, Scientific software solution
(www.tks3.de, tobiaskiessling@tks3.de)
Elisabet Teixido, Stefan Scholz - Helmholtz Centre for Environmental Research - UFZ
(www.ufz.de, stefan.scholz@ufz.de)

Table of Contents

1 Introduction 5

2 Getting Started 6

2.1 System Requirements 6

2.2 Key to using this guide 6

2.2.1 Icons in this guide 6

2.2.2 Typographical conventions 6

2.3 Installing FishInspector 7

3 FishInspector software overview 8

3.1 Main window 8

3.2 Main menu bar commands 8

3.2.1 File menu commands 8

3.2.2 Run menu commands 8

3.2.3 Export menu commands 8

3.2.4 Settings menu commands 9

3.2.5 Help menu commands 9

3.3 Features editor box 9

3.3.1 Toggle visibility 10

3.3.2 Edit feature 10

3.3.3 Enable or disable a feature 11

3.4 Set parameters 11

3.4.1 Modify and save parameters 12

3.5 Output data 13

4 Loading images 14

4.1 Load images from directory 14

4.2 Load image from image list file 14

4.3 Image filter 14

4.4 Zoom tool 15

5 Process image 15

5.1 Feature correction 16

6 Feature description 17

6.1 Manual 17
6.2 Capillary 18
6.2.1 Parameters 18
6.2.2 Display options 19
6.3 FishContour 19
6.3.1 Parameters 19
6.3.2 Display options 20
6.4 FishEye 20
6.4.1 Parameters 21
6.4.2 Display options 21
6.5 FishOrientation 21
6.5.1 Display options 22
6.5.2 Output 22
6.6 CentralDarkLine 23
6.7 Parameters 23
6.7.1 Display options 23
6.8 Bladder 23
6.8.1 Parameters 24
6.8.2 Display options 25
6.9 Notochord 25
6.9.1 Parameters 25
6.9.2 Display options 25
6.10 Otolith 26
6.10.1 Parameters 26
6.10.2 Display options 27
6.11 Yolk sac 27
6.11.1 Parameters 27
6.11.2 Display options 28
6.12 Pericard 28
6.12.1 Parameters 29
6.13 Pigmentation 29
6.13.1 Parameters 29

6.13.2 Display options 30

7 Rotate and Crop images workflow 31

7.1 Instructions 31

Last update: 08/02/2019

1 Introduction

Welcome to the FishInspector user guide. This software allows annotation of features in images of zebrafish embryos. The recent version requires images of a lateral position. It is important that the position is precise since deviation may confound with feature annotations. Images from any source can be used. However, depending on the image properties parameters may have to be adjusted. Furthermore, images obtained with normal microscope and not using an automated position system with embryos in glass capillaries require conversion using a KNIME workflow (available here). As a result of the analysis the software provides JSON files that contain the coordinates of the features. Coordinates are provided for eye, fish contour, notochord, otoliths, yolk sac, pericard and swim bladder. Furthermore, pigment cells in the notochord area are detected. Additional features can be manually annotated. It is the aim of the software to provide the coordinates, which are analysed subsequently to identify and quantify changes in the morphology of zebrafish embryos.

The software has been established by support of the German Federal Ministry of Education and Research to the project ZFminus1 (grant number 031A582) within the funding scheme "Alternatives to Animal Testing".

2 Getting Started

2.1 System Requirements

This program is supported and has been tested on Windows 7. We cannot guarantee full functionality if the software is installed on other Windows versions.

For other operating systems you can run the script from the source code (freely available on GitHub). This software has been written in Matlab R2015b and you will need a Matlab license if you want to modify the code and compile a new version. The software may run also on MacOS. However, it must be executed from the source code with a MacOS MATLAB installation or it should be compiled on MacOS MATLAB for being executed in a MacOS environment. Both things have not been tested yet. If you run the software successfully on MacOS please let us know.

📝 FishInspector software requires Matlab Runtime v 9.0. The executable windows file already includes its installation.

2.2 Key to using this guide

This guide includes information on:

- Installing the FishInspector software
- FishInspector overview
- FishInspector feature annotation description

2.2.1 Icons in this guide

Name	**Description**
Note	📝 This indicates supplementary explanations and useful tips. It is recommended that you read the text or follow the link.
Terms	📖 This indicates terms that are useful for understanding the explanations.

2.2.2 Typographical conventions

Convention	**Definition**
Forward Arrow →	The forward arrow symbol → instructs you to select a series of menu items in a specific order. For example, **Tools → Options** is equivalent to: From the **Tools** menu, select **Options**.

Boldface font	A **boldface** font indicates that the given word(s) are shown as toolbar button or menu selection. For example, if you are told to select a menu item in a particular order, such as **Tools → Options**, or to click **Save**.
Italic font	An *italic* font indicates the introduction of important terminology. Expect to find an explanation in the same paragraph or elsewhere in the guide.
`Code` and **`code bold`**	A `code` font indicates you are to make a keyboard entry. It also shows a programming code or code examples. The **`code bold`** font highlights the entry.

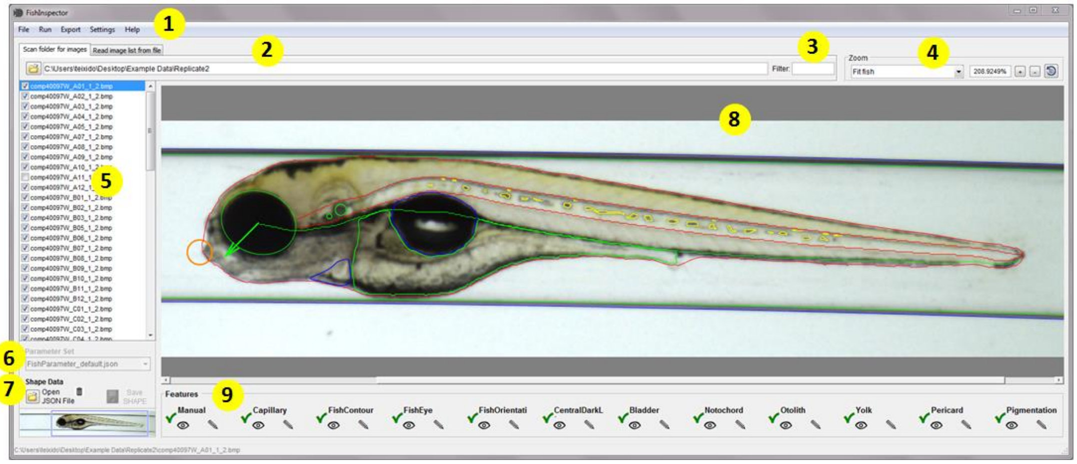
2.3 Installing FishInspector

- Download the executable file from the GitHub repository: https://github.com//sscholz-UFZ/FishInspector/releases
- Go to the folder where you downloaded the software: Double-click the **.exe** file to launch the **Setup** program.
- On the **FishInspector Installer** window click **Next**.
- On the **Installation options** window **choose installation folder** or leave the default one. Check the box **Add a Shortcut to Desktop** if required. Click **Next.**
- In the **required software** window, click to install the Matlab Runtime (FishInspector software requires Matlab Runtime v 9.0). If Matlab runtime is already installed, click **Next.**
- On the **confirmation** window click **Next** to start the FishInspector installation.
- Click **Finish** and start using FishInspector.

3 FishInspector software overview

3.1 Main window

Once the FishInspector starts up, the main window appears:



1. Main menu bar
2. Set input (scan folder or list images from file)
3. Filter images
4. Zoom and orientation box
5. List of loaded images
6. Parameter file selection
7. Open/save shape data
8. Image box display
9. Features editor box

3.2 Main menu bar commands

This section describes all menu commands in the **File**, **Run**, **Export**, **Settings** and **Help** menus.

3.2.1 File menu commands

Open directory – browse the file system to retrieve images from a directory.

Open image list – browse the file system to retrieve a text file (.txt) containing the full path to the image per line.

Quit – Close the FishInspector software.

3.2.2 Run menu commands

Process all images – automatically process all images for feature extraction.

Update feature in all images – update a specific feature for all loaded images.

Update depended feature – if ticked updates dependent features when images are analysed or a feature is updated.

Preserve manual selection – preserve a manual selection when images are analysed or a feature is updated.

3.2.3 Export menu commands

Export current image… – export current displayed image in png format with the annotated features.

Export image list – export the list of images loaded in .txt format.

Export all images with reduced background – export all loaded images with reduced background (currently only working for .tif images).

3.2.4 Settings menu commands

Always start editor with "Manual Selection" enabled – if ticked the image will be loaded with the "Manual Selection" mode enabled in the features editor box.

3.2.5 Help menu commands

Dos command – List of possible Dos commands to process images in batch mode with FishInspector software started from the command line in windows.

Licenses – display license information.

About… – display FishInspector software version, release date and link to license information.

3.3 Features editor box

This box displays the current set of features available for annotation:

- Manual selection
- Capillary
- Fish contour
- Fish eye
- Fish Orientation
- Central dark line
- Bladder
- Notochord
- Otolith
- Yolk sac
- Pericard
- Pigmentation

The detection of various features is organized hierarchically, that is, in order to locate a certain feature the locations of previously detected features are included. For example, detection of the contour of the embryo is guided by the capillary boundaries, since the software expects the embryo to be located inside the capillary (or least in a virtual capillary). Subsequently, other features are identified in a stepwise manner (See Figure 1). Hence, the detection of specific morphological features is dependent on the detection of other features and is facilitated by excluding regions that may interfere. The identification of the regions of interest is driven by visual observation and measurement of generic object properties. For example, once the contour of the fish is localized, the eye is detected by searching for a dark object either in the right or left half of the zebrafish.

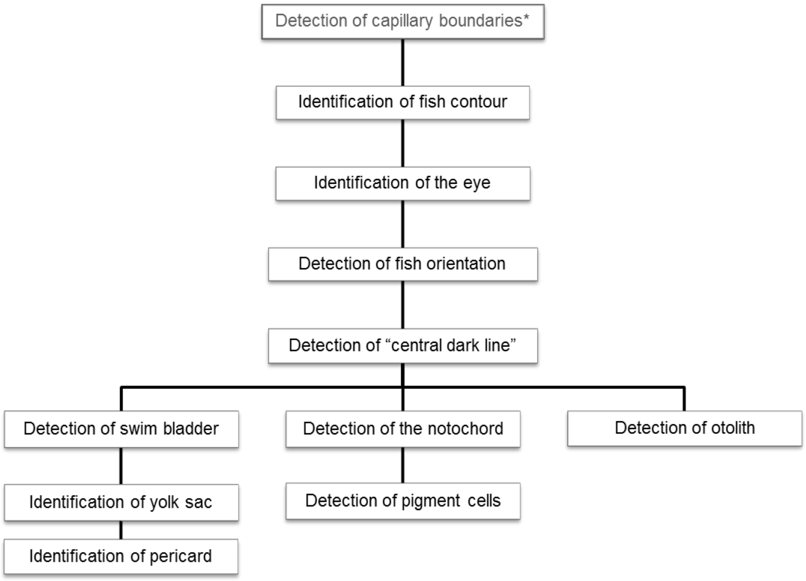
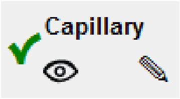
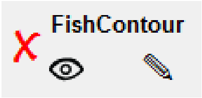

Detection of capillary boundaries*
Identification of fish contour
Identification of the eye
Detection of fish orientation
Detection of "central dark line"
Detection of swim bladder
Detection of the notochord
Detection of otolith
Identification of yolk sac
Detection of pigment cells
Identification of pericard

Figure 1. Stepwise feature recognition by the FishInspector software. The "central dark line", represents a structure of high contrast between the upper and bottom part of the fish, starting from the fish eye. This feature is only used to support the identification of other features and does not refer to a morphologically relevant entity. *Images without capillary need to be modified (i.e. insertion of a virtual capillary)by an automated workflow to be compliant with the FishInspector software (KNIME workflow [available here](#)).



Green tick next to the feature indicates that the feature has been detected.



Red cross next to the feature indicates that there was an error with the plugin and the feature can't be detected.

Red cross: This may happen for example when the capillary was not detected correctly (but it has a green tick), then the fish contour and any other feature that depends on the capillary fail and is marked by a red crosses.

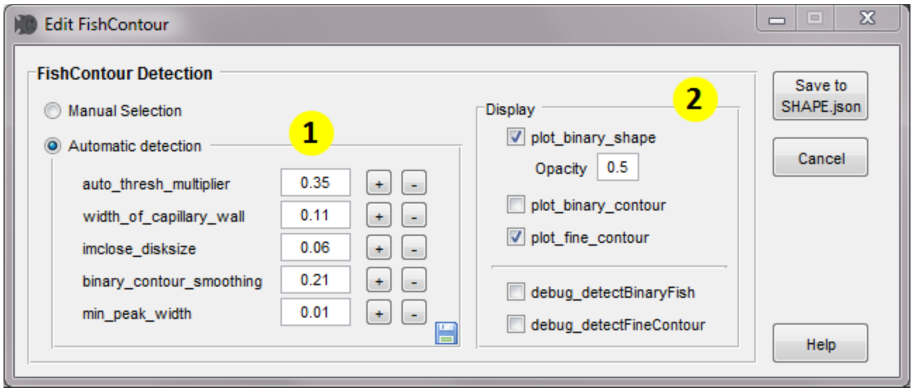
3.3.1 Toggle visibility

Press the 👁 icon to show or hide 👁 the feature in the image display.

3.3.2 Edit feature

Press the ✎ icon to access the editor window of a specific feature.

The editor window allows to choose between **manual selection** or **automatic detection** of the feature. When using the automatic detection, in most of the cases you will find a box containing the parameter that can be modified (1) (see section Modify and save parameters). The editor window also contains a display box (2) with the display options for that specific feature.



○ Manual Selection

● Automatic detection

auto_thresh_multiplier	0.35	+	-
width_of_capillary_wall	0.11	+	-
imclose_disksize	0.06	+	-
binary_contour_smoothing	0.21	+	-
min_peak_width	0.01	+	-

☑ plot_binary_shape

Opacity 0.5

☐ plot_binary_contour

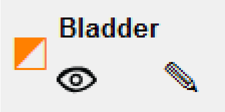
☑ plot_fine_contour

☐ debug_detectBinaryFish

☐ debug_detectFineContour

Not all features allow manual selection or have display options.

3.3.3 Enable or disable a feature

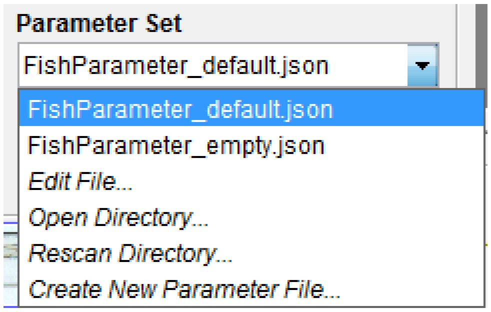


To enable or disable a feature, right click with the mouse on the feature and press (untick) **Use feature**. The disable feature will have a specific orange symbol next to it.

3.4 Set parameters

Given that establishment of a 100% correct automated feature detection would be very challenging and to allow improvement by the user, the software permits modification of the parameters used for the automated feature detection. This is very useful to adapt for specific image characteristics (contrast, intensity, RGB or grayscale, developmental stage). However, also for optimised parameters, the automated detection may not be successful. In this case it is possible to manually edit the annotation of the feature.

The parameters of all features are stored using a json file. The FishInspector software comes with a set of default parameters automatically loaded from the **parameter set** box in the left bottom part of the main window. Modified parameters can be saved in a specific file that can then be loaded in subsequent analyses with similar images.



This menu allows to:

Edit File – (Not active in recent version (1.03))

Open Directory – browse the file system to retrieve the folder where the parameters are saved.

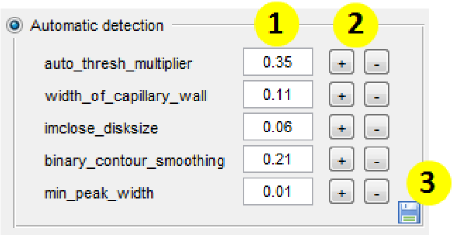
Rescan Directory – Refreshes the current directory that contains the parameters.

Create New Parameter File – Allow to create a new file containing new set of parameters.

Create a new parameter file if you have different types of images (from a different source) or different embryo stages that could differ on detection of the features.

3.4.1 Modify and save parameters

Features can have parameters to improve its automatic detection. These parameters can be modified in the editor window of each feature (see section Edit Feature). To modify a parameter either introduce directly a number value in the box next to the parameter (1) or click on the icon with a plus or minus sign (2). To save the parameters click on the save icon (3).



● Automatic detection

auto_thresh_multiplier	0.35
width_of_capillary_wall	0.11
imclose_disksize	0.06
binary_contour_smoothing	0.21
min_peak_width	0.01

To apply a change on a parameter/feature for all analysed images follow these steps:

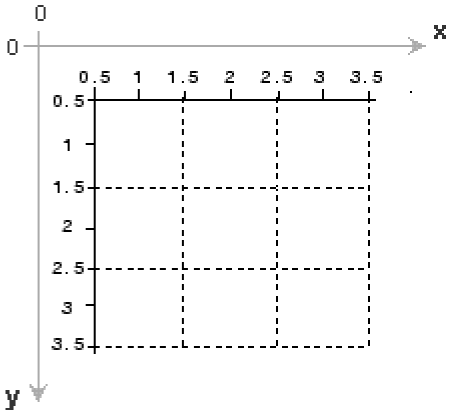
1. Disable the feature on the first image.
2. **Run** → **Update feature in all images** and select the specific feature to be updated.
3. Once the analysis is finished, enable the feature on the first image.
4. Correct the parameters and save the shape data by click on the **save SHAPE.json** button on the editor window of the feature.
5. **Run** → **Update feature in all images** and select the specific feature to update.

3.5 Output data

The resulting output of the FishInspector is a set of xy coordinates of the morphological feature detected.

FishInspector has been programmed in MATLAB®, which stores most images as two-dimensional arrays (i.e., matrices).Each element of the matrix corresponds to a single pixel in the displayed image. To access locations in images, the MATLAB Image Processing Toolbox™ uses several different image coordinate systems as conventions for representing images as arrays.

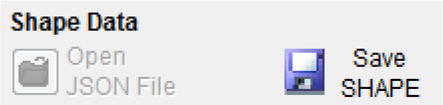
By default, the toolbox uses a spatial coordinate system for an image that corresponds to the image's pixel indices. It is called the intrinsic coordinate system and is illustrated in the following figure. The image is treated as a grid, ordered from top to bottom, and left to right. Y increases downward, while x increases to the right.



Coordinates are dependent on image resolution. Hence, for a comparative analysis it is recommended to use images with the same resolution. Otherwise subsequent analysis of JSON files may require adaptation for specific image resolution.

For each image analyzed, data are exported to a single JSON file, which is a language independent open-standard file format typically used for transmitting data between applications. The boundary coordinates of multiple features can then be stored in a structured text file. This allows the seamless integration of the FishInspector output into custom post-processing algorithms, which can be implemented in any programming language.

The output data file is generated in the same folder from which images have been loaded and takes the same name as the image file plus a suffix ending by __SHAPES. The output is generated by clicking either the **Save SHAPE** button situated on the left bottom part of the main window in the Shape Data box.



or by click on the **Save SHAPE.json** button on each feature editor window (see chapter Edit feature).

Open JSON File – opens the directory that contains the output Json file.

4 Loading images

Images can be loaded by open a directory or by reading image list from file.

4.1 Load images from directory

There are two options to proceed:

Select **File → Open directory** on the menu bar to open the file dialog, browse and select the folder where the images are stored.

Or select the **Scan folder for images** tab under the main menu bar and click 📂 to open the file dialog, browse and select the folder where the images are stored.

The names of the images loaded are displayed in the list box on the left of the main window. All images from parent and subfolders will be considered.

4.2 Load image from image list file

There are two options to proceed:

Select **File → Open image list** on the menu bar to open the file dialog, browse and select the text file (.txt) containing the full path to the images.

Or select the **Read image file from file** tab under the main menu bar and click 📂 to open the file dialog, browse and select the text file (.txt) containing the full path to the images.

The names of the images loaded are displayed in the list box on the left of the main window.

Most common image file formats are supported: .tif, .png, .bmp and .jpg. FishInspector supports RGB and grey images.

4.3 Image filter

The image filter box allows you to refine the image list to just load and process selected images. Specific words can be used to selectively analyse images (e.g.

“.bmp” will load only images with bmp format, or “_2” will load images which name contains these characters).

You can also deselect images by unchecking the tick boxes at the image file names (see Process image)

4.4 Zoom tool

The **Zoom** box is situated on the right top of the main window and allows you to adjust the image zoom to:

Fit fish – it resizes the image based on the contour of the fish.

Fit image – it resized the image to fit into the the size of the display window.

Fit image height – it resizes the image to fit the height of the image into the display window.

Fit image width – it resizes the image taking to fit the width of the image into the display window.

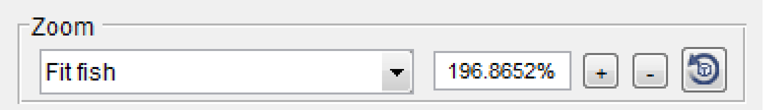
50% – it resizes the images to the 50% of its original size.

100% – it resizes the image to its original size.

200% – it resizes the images to 200% of its original size.

Custom – introduce a specific percentage to zoom out or zoom in.

To gradually enlarge or reduce the image click the icon with the plus or minus signs.


Zoom

Fit fish

196.8652%

Click the icon in the **Zoom** box to display the fish to its *standard orientation*, i.e. head to the left and tail to the right. In case the fish is not displayed in the *standard orientation* after activation of the icon, the orientation of the fish has not been identified correctly. In this case the fish orientation feature has to be corrected manually (see FishOrientation).

Standard orientation is defined as the fish with the head on the left, tail on the right and yolk sac on the bottom part.

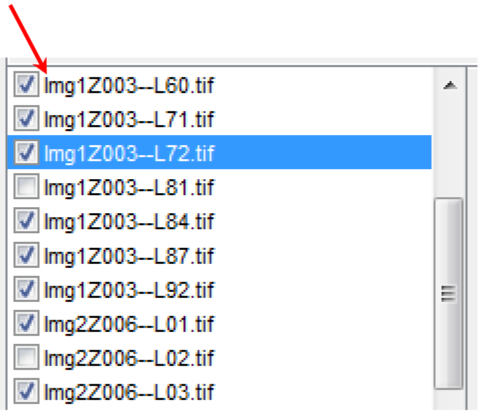
5 Process image

Once the images have been loaded you can start the analysis. Click on one image from the list box on the left (the name of the image will appear highlighted in blue). Then the detection of the features will automatically start from left to right. Loading of the first image may takes a bit longer. The time required for loading an image depends on the resolution and the computer processing power.

Click the icon in the **Zoom** box to display the fish to its *standard orientation* (more convenient for the visual analysis).

Before starting manual correction of features check if the parameter file (see section Set parameters) is the one you intend to use. Every time the FishInspector is started it loads the default parameter file. It is recommended that parameter files are optimised to reduce requirement for manual editing.

The tick mark on the left of the image name on the list box indicates that the analysis of the image is enabled, unticked images will not be analysed. Images may be deselected e.g. in case of insufficient positioning.



☑ Img1Z003--L60.tif
☑ Img1Z003--L71.tif
☑ Img1Z003--L72.tif
☐ Img1Z003--L81.tif
☑ Img1Z003--L84.tif
☑ Img1Z003--L87.tif
☑ Img1Z003--L92.tif
☑ Img2Z006--L01.tif
☐ Img2Z006--L02.tif
☑ Img2Z006--L03.tif

5.1 Feature correction

The software permits modification of the parameters used for the automatic feature detection or manual correction given that establishment of a 100% correct automated feature detection is very challenging.

Click on the icon to access the editor window of a specific feature. The best way to check and correct features is from left to right due to the dependency on detection of some features (see section Features editor box).

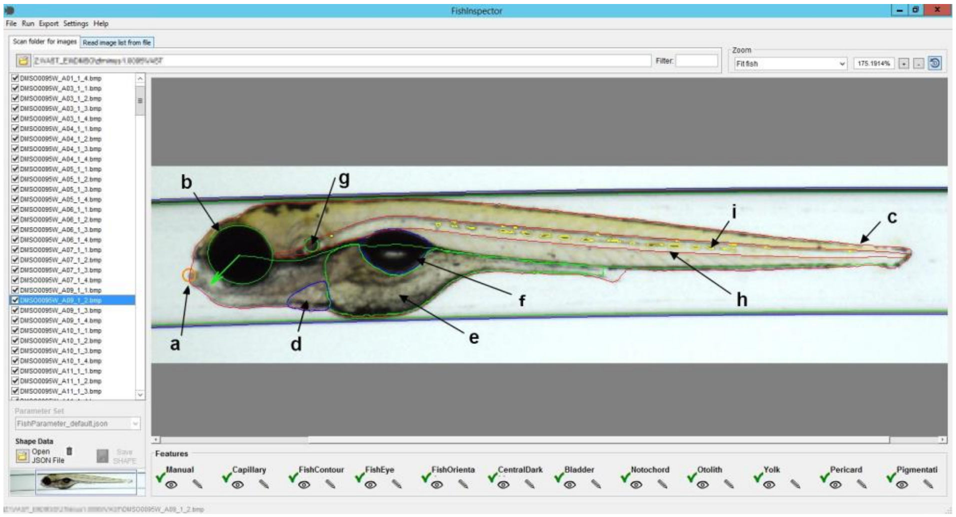
First try to correct the parameters of the feature to see if its detection can be improved, and save parameters if you want them to be used for subsequent image analysis (see Modify and save parameters). If no further improvement by changing parameters can be achieved, correct the feature manually (see section Edit Feature). Hold click and drag the mouse to correct the annotated lines or points.

Once the feature is corrected, click on **Save to SHAPE.json** to confirm and save changes. Any dependent features will be updated as well.

To disable the update of dependent features click on **Run** → **Update depended features**.

6 Feature description

This section contains the description of each feature and the functionality of the parameters of a given feature. The description is very detailed. However, to use the FishInspector it may not require that you understand the underlying processes in detail.



Zoom: Fit fish 175.1914%

Parameter Set: FishParameter_default.json

In the example shown above, the following features were detected : a, lower jaw tip using the **Manual** feature tool (orange), b, eye contour (green), c, fish contour (red), d, pericard (blue), e, yolk sac (green), f, swim bladder (blue), g, otolith (green), h, notochord (green), i, pigmentation (yellow).

6.1 Manual

The manual annotation allows to create a custom line or point to annotate feature for which not automatic detection tool is available.

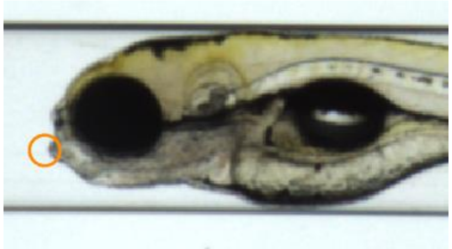
Press the ✎ icon of the **Manual** feature. Then use the context menu (by right click with the mouse on the image) to create a **new line** or **new point**.

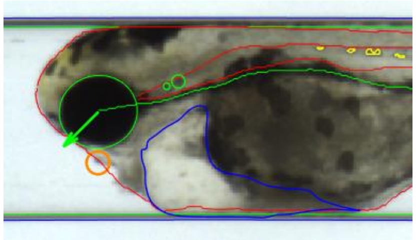
Hold click and drag the mouse to move a point to a different place or to modify the shape of a line. To enlarge or reduce the size of the point use the scroll wheel of the mouse.

To **delete** a point or line right click with the mouse on the line or point to erase and click **delete**.

Principally it is possible to apply multiple labels using the **Manual** feature tool. These will be stored in the order how the labels were applied. Hence, if you apply more than one label you need to process them in the same sequence since otherwise they would be confused in subsequent processing of JSON files. However, the order changes in case subsequent editing of a label. Hence, annotation of multiple features should be conducted with care.

Currently, jaw morphology analysis is at present only possible by using the manual selection tool and by labeling the lower jaw tip with a point mark. Jaw morphology analysis is usually conducted for embryos older than 72 hpf. When analysing jaw morphology, if you consider that an embryo has no jaw, insert a mark point below the eye on the fish contour. This is to prevent the failure of the subsequent analysis.





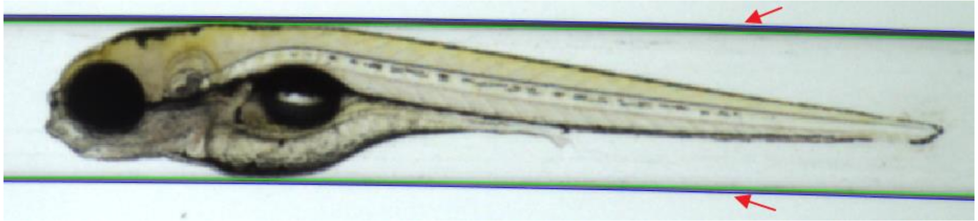
6.2 Capillary

The detection of the capillary is done in two steps.

First, a binary image is generated, using a dynamic threshold based on *Otsu's method* [1], multiplied by the **threshMultiplier** variable.

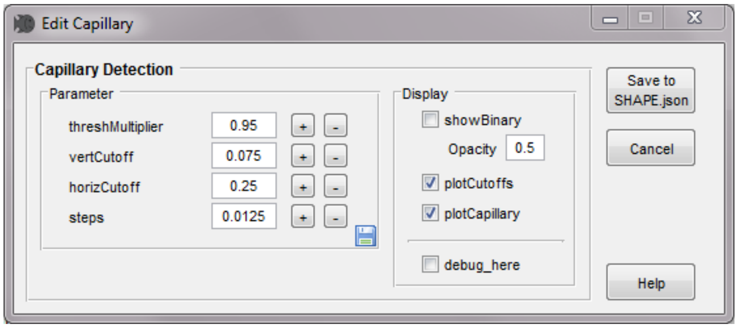
Second, based on the binary image, the position of the capillary wall is estimated. Regions at the border of the image with strong background variations and/or artefacts from stitching may affect detection of the capillary wall. Those regions can be excluded by adjusting the **vertCutoff** and **horizCutoff** variables, which represent fractions of the image-height and image-width respectively.

The image below indicates the detection of the capillary, green lines indicate the inner capillary wall and blue lines the outer capillary outline.



6.2.1 Parameters

[1] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.


Edit Capillary

Capillary Detection

Parameter

threshMultiplier	0.95
vertCutoff	0.075
horizCutoff	0.25
steps	0.0125

Display

□ showBinary

Opacity 0.5

☑ plotCutoffs

☑ plotCapillary

□ debug_here

threshMultiplier – modifies the threshold value obtained with the *Otsu's method* [1] using the graythresh function in MATLAB.

vertCutoff & horizCutoff – allow to exclude regions at the border of the image which may have strong background or artefacts and can affect the detection of the capillary wall.

steps – used to find the inner edges (lower and upper) of the capillary wall.

6.2.2 Display options

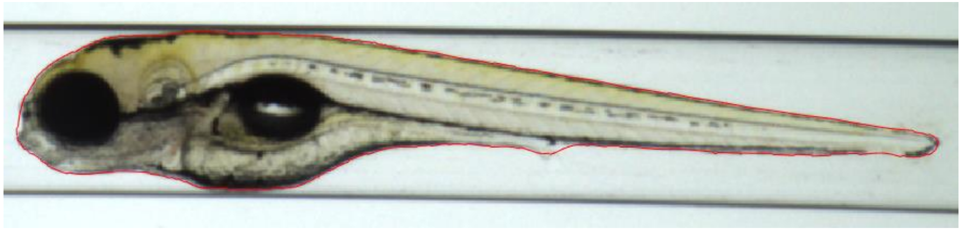
showBinary – overlays the binary image in the display window.

plotCutoffs – show or hide vertical and horizontal cut off.

plotCapillary – show or hide capillary annotation.

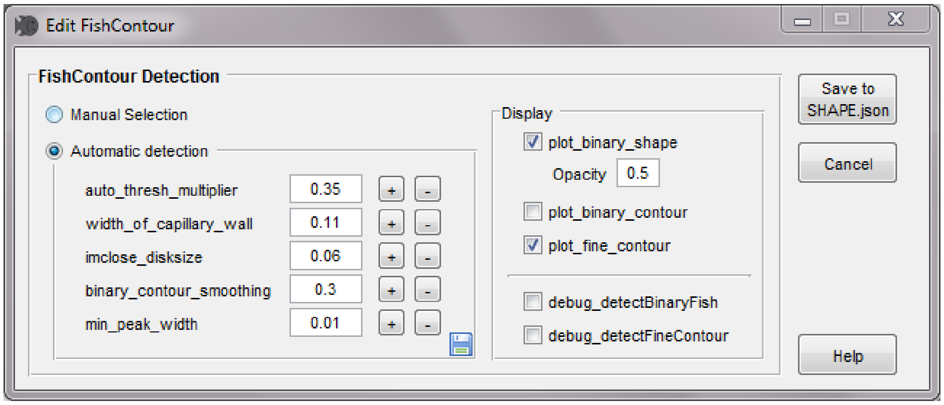
6.3 FishContour

The detection of the fish contour is based on a binary image generated using a dynamic threshold based on *Otsu's method* [1], multiplied by the **auto_thresh_multiplier** variable. The binary is generated from the inner image inside the capillary. Then the biggest region is selected (to discard artefacts) and holes are filled using the variable **imclose_dislike**. Boundary coordinates are obtained from this binary image and this is what is called *binary contour.* Finally, the binary contour is smoothed and an active contour is used to get the fine position and get the boundaries of a *fine contour* (in red on the image below).



All parameters used, except the auto_thresh_multiplier, are related to the average capillary width of each image. I.e. images with different resolutions do not require a different set of parameters since they are "normalised" by the capillary width.

6.3.1 Parameters



FishContour Detection

○ Manual Selection

● Automatic detection

auto_thresh_multiplier 0.35

width_of_capillary_wall 0.11

imclose_disksize 0.06

binary_contour_smoothing 0.3

min_peak_width 0.01

Display

☑ plot_binary_shape

Opacity 0.5

□ plot_binary_contour

☑ plot_fine_contour

□ debug_detectBinaryFish

□ debug_detectFineContour

auto_thresh_multiplier – modifies the threshold value obtained with the *Otsu's method* using the graythresh function in Matlab.

width_of_capillary_wall – all white border pixels with less than width capillary wall pixels are removed.

imclose_disksize – defines the radius of the disk-shaped structuring element used in the imclose function in Matlab to fill the holes of the binary image.

binary_contour_smoothing – defines the degree of smoothness of the fine contour goes from 0 (highly smooth contour) to infinite.

min_peak_width – is used in the *active contour* function to get the fine position of the fine contour. Takes values from 0 to 1, the greater the value the finest position takes.

Active contour models are used to find the boundaries of shapes in an image.

6.3.2 Display options

plot_binary_shape –overlays the binary shape in the display window

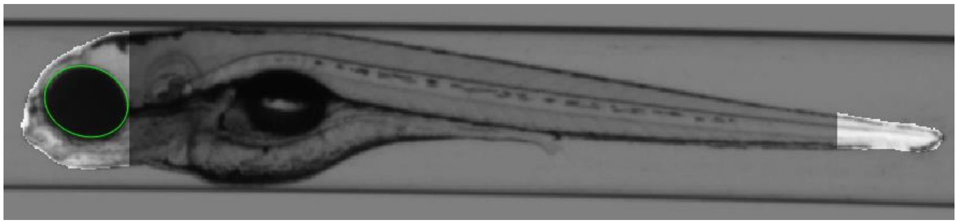
plot_binary_contour – show or hide binary contour.

plot_fine_contour – show or hide fine contour.

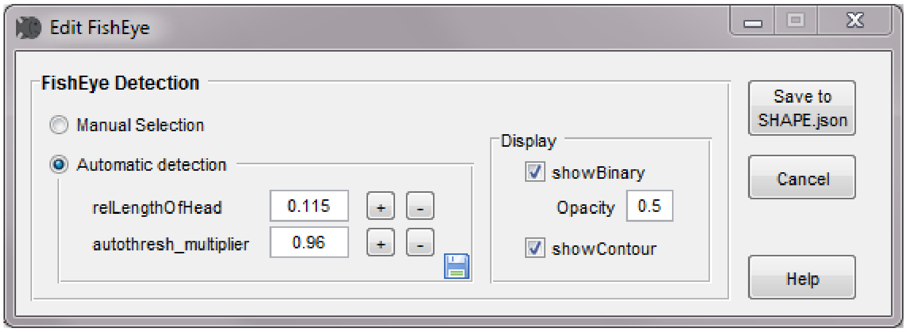
6.4 FishEye

The detection of the fish eye is based on the creation of a binary image only taking into account the region inside the contour of the fish. Moreover, the central part of the fish is excluded using the **relLengthOfHead** variable in order to find the darkest region in the two extreme parts of the fish (head or tail).

The image below displays the region excluded from the eye detection and the eye outlined in green.



6.4.1 Parameters


Edit FishEye

FishEye Detection

○ Manual Selection

● Automatic detection

relLengthOfHead 0.115

autothresh_multiplier 0.96

Display

☑ showBinary

Opacity 0.5

☑ showContour

relLengthOfHead – relative length of head to exclude central part of the fish.

autothresh_multiplier – modifies the threshold for the detection of the eye. Reduce it to detect a less pigmented eye.

6.4.2 Display options

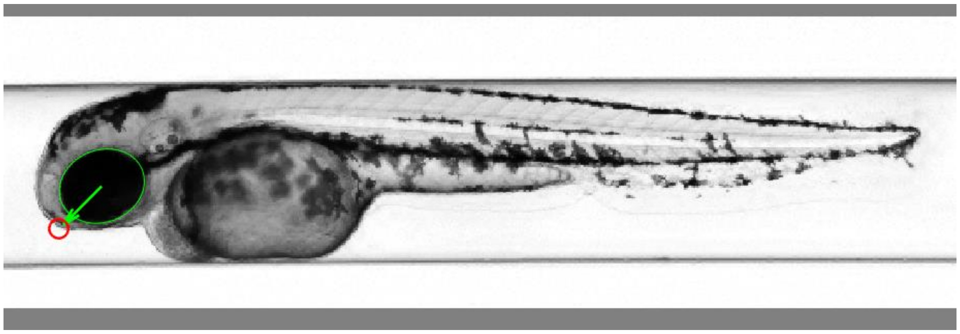
showBinary – show or hide the region that is excluded from the eye detection

showContour – show or hide the contour of the fish eye detected

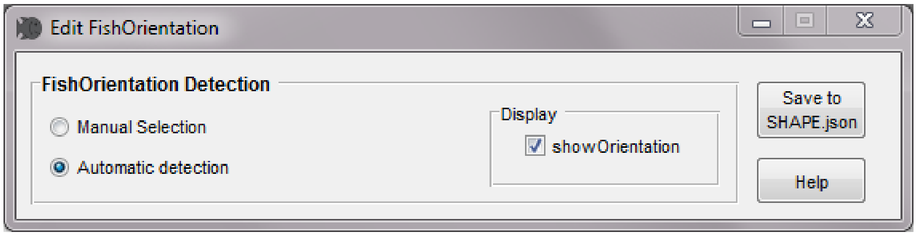
6.5 FishOrientation

This feature is used to determine the orientation of the fish. The horizontal orientation is detected by finding where the eye is (maximum sum of pixel values). The vertical orientation is determined in 3 steps: a) around the eye region the algorithm looks for the darkest spot inside the fishContour in each column. b) these spots are fitted to a quadratic function ax^2. c) The sign of a determines the vertical orientation.

The orientation is shown by an arrow, the standard orientation of the fish is with the head on the left and yolk sac on the bottom part as the picture shows. The arrow needs to point towards the ventral, rostral side of the fish embryo (as indicated in the image below).



To correct the orientation, select **manual selection** in the editor window of the feature and hold click on the red circle and drag the arrow to the correct position.



Edit FishOrientation

FishOrientation Detection

○ Manual Selection

● Automatic detection

Display

☑ showOrientation

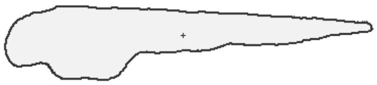
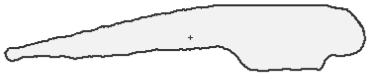
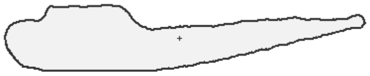
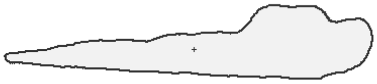
6.5.1 Display options

showOrientation – show or hide the green arrow (orientation).

6.5.2 Output

The orientation of the fish embryos in the original image is described in the output JSON files by two variables: horitzontally_flipped and vertically_flipped.

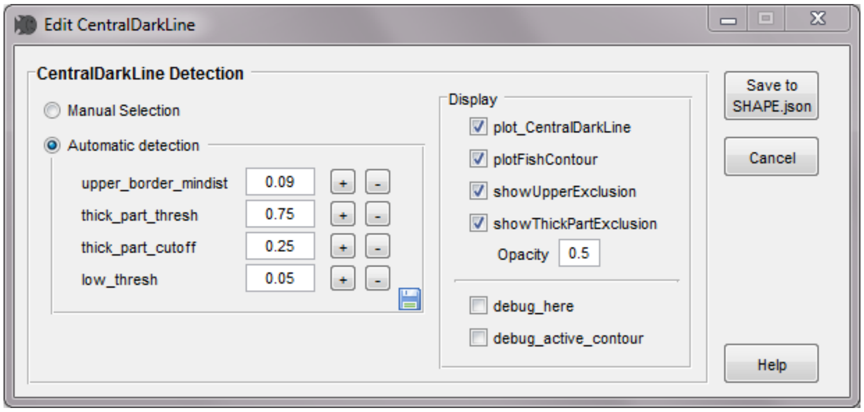
Remember that digital images are typically viewed with coordinates (indices) in which y increases downwards and x increases to the right (see section Output data). That should be taken into account when doing morphometric analysis.

	**Standard orientation** horitzontally_flipped:0 vertically_flipped: 0
	horitzontally_flipped:1 vertically_flipped: 0
	horitzontally_flipped:0 vertically_flipped: 1
	horitzontally_flipped:1 vertically_flipped: 1

6.6 CentralDarkLine

This feature is used only as reference for the detection of the subsequent features like yolk sac or notochord.

6.7 Parameters



Edit CentralDarkLine

CentralDarkLine Detection

○ Manual Selection

● Automatic detection

upper_border_mindist	0.09
thick_part_thresh	0.75
thick_part_cutoff	0.25
low_thresh	0.05

Display

☑ plot_CentralDarkLine

☑ plotFishContour

☑ showUpperExclusion

☑ showThickPartExclusion

Opacity 0.5

□ debug_here

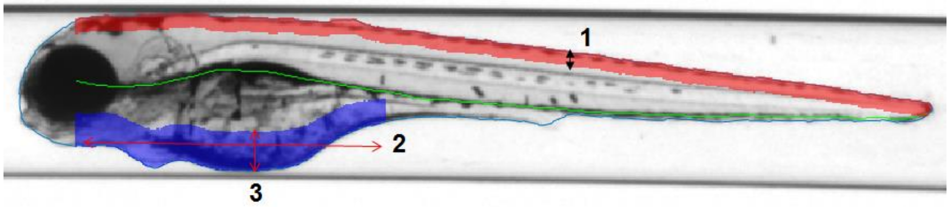
□ debug_active_contour

upper_border_mindist – allows to discard the upper part of the fish, to avoid dark pixels (1).

thick_part_thresh – modifies threshold of the lower excluded part (yolk sac) to expand or contract the area (2).

thick_part_cutoff – allows to discard lower part of the fish that can influence on the detection of the central dark line (3).

low_thresh – This value currently does nothing and could actually be removed (originally it was a lower threshold for finding the dark line)



6.7.1 Display options

plot_CentralDarkLine – show or hide the detected central dark line.

plotFishContour – show or hide the detected fish contour.

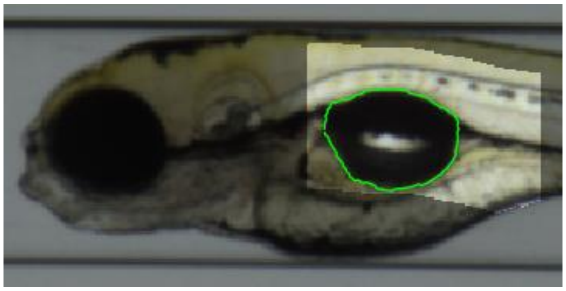
showUpperExclusion – show or hide upper part excluded.

showThickPartExclusion – show or hide lower part excluded.

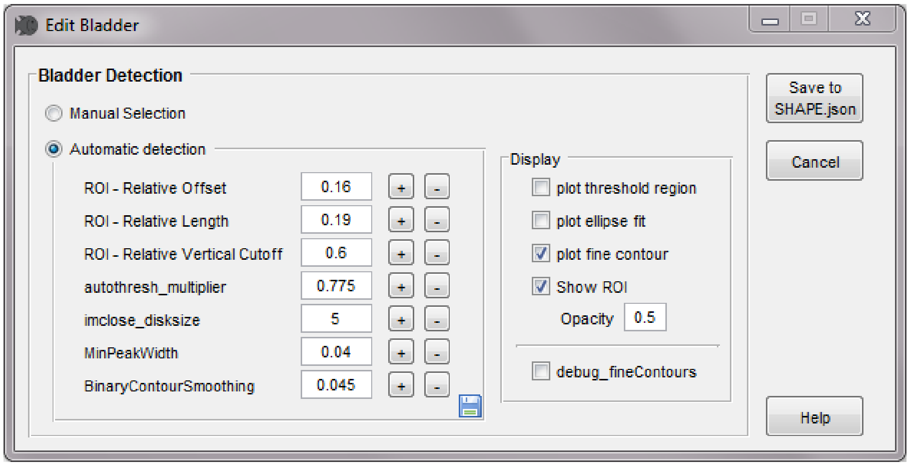
6.8 Bladder

Region of interest (ROI): subset of an image or a dataset identified for a

For the detection of the swim bladder a mask is generated to limit the principally possible location in the image. The mask is created along the central dark line taking into account the parameters that defines the *region of interest* (ROI, rectangle shown in the image below). After that a threshold is applied and the biggest dark blob is used to detect the swim bladder boundaries (in green).



6.8.1 Parameters



Edit Bladder

Bladder Detection

○ Manual Selection

● Automatic detection

ROI - Relative Offset	0.16
ROI - Relative Length	0.19
ROI - Relative Vertical Cutoff	0.6
autothresh_multiplier	0.775
imclose_disksize	5
MinPeakWidth	0.04
BinaryContourSmoothing	0.045

Display

□ plot threshold region

□ plot ellipse fit

☑ plot fine contour

☑ Show ROI

Opacity 0.5

□ debug_fineContours

ROI - Relative Offset – change the position along the central dark line of the region of interest.

ROI - Relative Length – increase or decrease the length of the region of interest from the right position.

ROI - Relative Vertical CutOff – wide or narrow the region of interest.

autothresh_multiplier – modifies the threshold for the detection of the swim bladder.

imclose_disksize –defines the radius of the disk-shaped structuring element used in the imdilate function in Matlab of the binary image.

MinPeakWidth – is used in the active contour function to get the fine position of the fine contour. Takes values from 0 to 1, the greater the value the finest position takes.

BinaryContourSmoothing – Takes values from 0 (very smoothed) to 1 (no smoothing at all). Used to refine contour of the bladder.

6.8.2 Display options

plot threshold region – show or hide the region after the threshold.

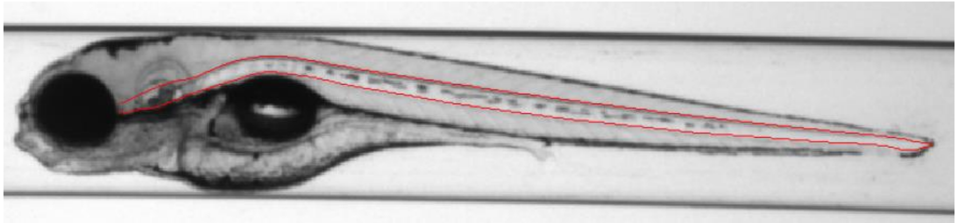
plot ellipse fit – show or hide the ellipse fit of the threshold area.

plot fine contour – show or hide the fine contour of the detected bladder boundaries.

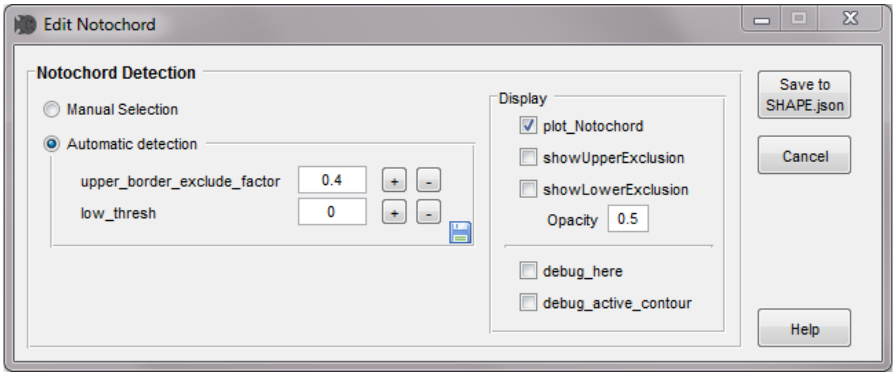
show ROI – show or hide the ROI for the bladder detection.

6.9 Notochord

To detect the notochord only the upper part of the central dark line is used by generating a mask using the fish contour, fish eye and central dark line. The display option **showLowerExclusion** shows or hides this mask. Also an exclusion region on the upper part of the fish is created to improve detection of the notochord (**showUpperExlcusion**).



6.9.1 Parameters



Edit Notochord

Notochord Detection

○ Manual Selection

● Automatic detection

upper_border_exclude_factor 0.4

low_thresh 0

Display

☑ plot_Notochord

□ showUpperExclusion

□ showLowerExclusion

Opacity 0.5

□ debug_here

□ debug_active_contour

upper_border_exlcude_factor – wide or narrow the excluded border region on the upper part of the fish.

low_thresh – this value determines which pixels are considered as a maximum (for the upper border of the notochord) and the minimum (for the lower border of the notochord)

6.9.2 Display options

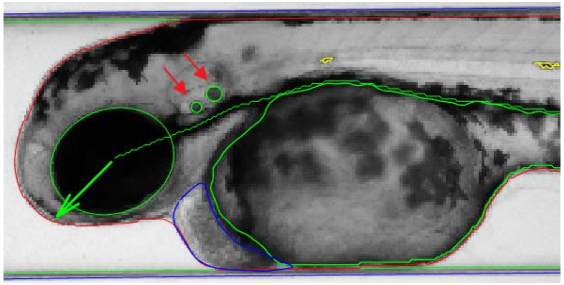
plot_Notochord – show or hide the notochord coordinates

showUpperExclusion – show or hide upper exclusion region of the fish.

showLowerExclusion – show or hide initial excluded region to detect the notochord.

6.10 Otolith

The otoliths are biomineralized ear stones that contribute to both hearing and vestibular function in fish. Embryos have two otoliths, the utricular (anterior) and saccular (posterior). The figure below display the location of the two otoliths of a zebrafish embryo (red arrows). Usually the posterior otolith is bigger than the anterior.



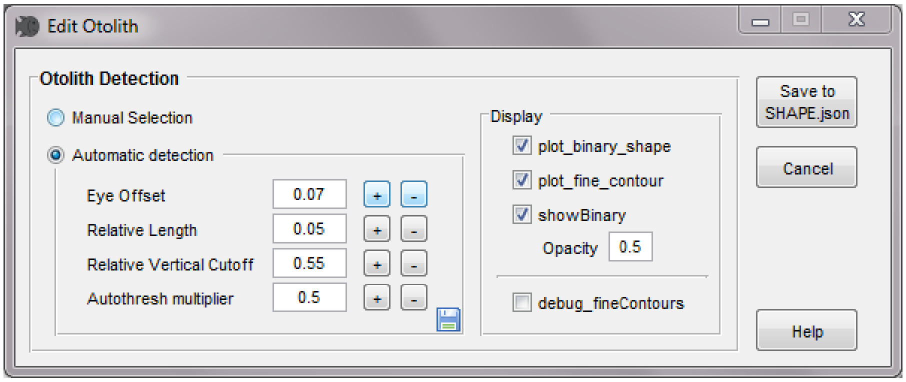
This feature is detected by first generating a mask to narrow down the region where to look for the otolith(s). For the mask some limits (cutoffs) are defined as variables: **Eye Offset**, **Relative Length** and **Relative Vertical CutOff**. Then a threshold is applied to find the darker dots that are the otoliths and finally the centroid (x,y) and radius of the detected object is saved as output.

The **manual selection** allows to erase or create new points and modify its size.

- Create **new point:** Use the context menu (by right click with the mouse on the image) and click **new point**. Hold click and drag the mouse to move a point to a different place. To enlarge or reduce the size of the point use the scroll wheel of the mouse.
- **Delete** a point: right click with the mouse on the point and click **delete.**

The otolith position is used to calculate the distance between the eye centroid and posterior otolith in subsequent analysis of JSON files. The distance depends on the developmental stage of the embryo. At present, the subsequent workflow requires the position of the larger, the posterior otolith. It is sufficient to check that the posterior otolith has received the bigger label. In the subsequent workflow only the position of the bigger label will be used to estimate the otolith-eye distance.

6.10.1 Parameters


Edit Otolith

Otolith Detection

○ Manual Selection

● Automatic detection

Eye Offset	0.07	+	-
Relative Length	0.05	+	-
Relative Vertical Cutoff	0.55	+	-
Autothresh multiplier	0.5	+	-

Display

☑ plot_binary_shape

☑ plot_fine_contour

☑ showBinary

Opacity 0.5

☐ debug_fineContours

Eye Offset – moves the mask along the central dark line (close or far from the eye).

Relative Length – Increases or decreases the width of the mask from the right part.

Relative Vertical CutOff – increases (by decreasing the value of the parameter) or decreases (by increasing the value of the parameter) the height of the mask from the top.

Autothresh multiplier – modifies the threshold value obtained with the *Otsu's method* used to detect the otoliths.

6.10.2 Display options

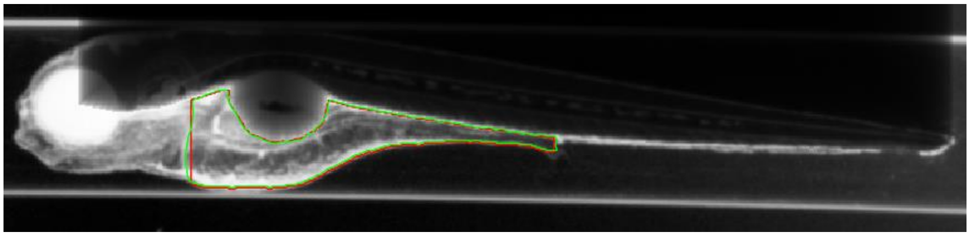
plot_binary_shape – show or hide binary shape of the detected otolith.

plot_fine_contour – show or hide otolith radius and position (ellipse).

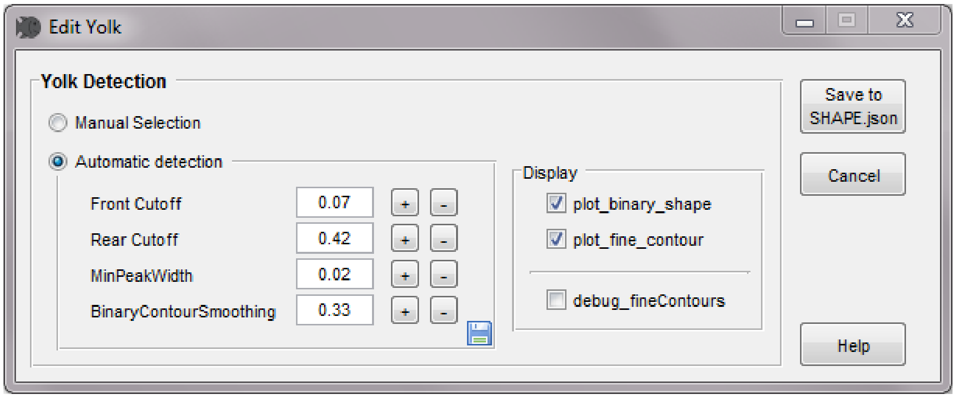
showBinary – show or hide mask created for the location of the otolith.

6.11 Yolk sac

The yolk sac is detected by excluding the upper part of the fish using the central dark line as reference and by using the inverted image. For the mask the bladder is excluded and also everything left to the right border of the eye. Some limits (cutoffs) are defined as variables: **Front CutOff** and **Rear CutOff.** The inverted image will be show during the edition of yolk sac boundaries.



6.11.1 Parameters



Yolk Detection

○ Manual Selection

● Automatic detection

Front Cutoff 0.07

Rear Cutoff 0.42

MinPeakWidth 0.02

BinaryContourSmoothing 0.33

Display

☑ plot_binary_shape

☑ plot_fine_contour

☐ debug_fineContours

Front Cutoff – increases or reduces the width on the left. To enlarge decrease the value.

Rear Cutoff – increases or reduces the width on the right.. To enlarge decrease the value.

MinPeakWidth – is used in the active contour function to get the fine position of the fine contour. Takes values from 0 to 1, the greater the value the finest position takes.

BinaryContourSmoothing – Takes values from 0 (very smoothed) to 1 (no smoothing at all). Used to refine contour of the yolk and create a fine contour.

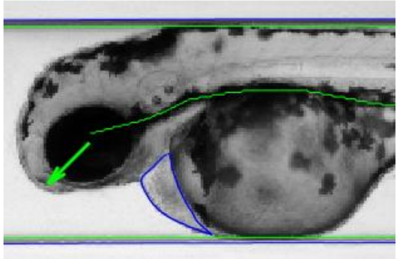
6.11.2 Display options

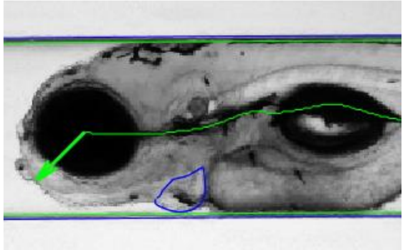
plot_binary_shape –show or hide binary shape of the detected yolk boundaries.

plot_fine_contour – show or hide fine contour of the yolk after smoothing.

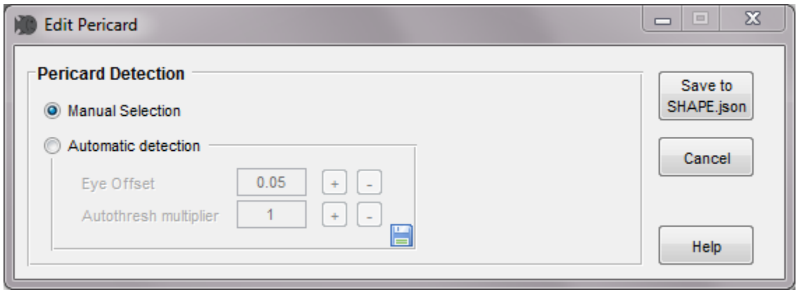
6.12 Pericard

At present, pericard always required manual correction and it is sometimes difficult to locate the boundaries precisely. To narrow its location a mask is created between the eye and yolk. A limit is set using the variable **Eye Offset**. For the detection and during edition it uses the inverted image. The images below show an example of the pericard region corrected (in blue) in 48hpf (top picture) and 96 hpf old embryos (bottom picture).





6.12.1 Parameters



Edit Pericard

Pericard Detection

● Manual Selection

○ Automatic detection

Eye Offset 0.05 + -

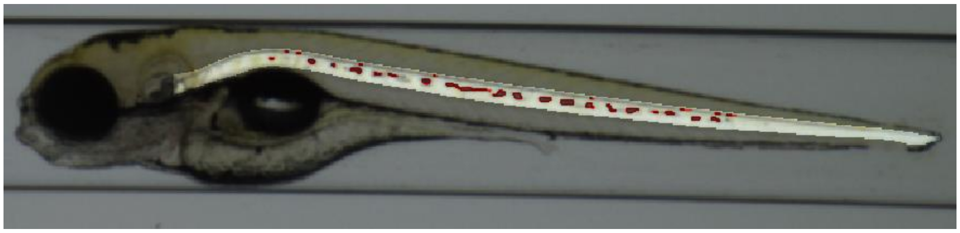
Autothresh multiplier 1 + -

Eye Offset – increases or reduces the width of the mask from the eye (larger values make the mask smaller).

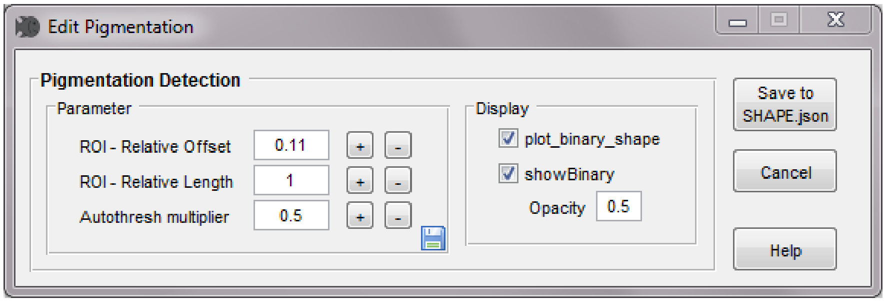
Autothresh multiplier – modifies the threshold value used to create the binary detected image of the pericard.

6.13 Pigmentation

The detection of the pigmentation is based on the detection of pigment cells inside the area delimited by the notochord (outlined in red in the image below). The pigmentation analysis was restricted to the notochord region to reduce variability. Many pigment cells are located dorsally. However, slight differences in position of the embryo may impact on the number of pigment cells visible in the lateral image. Hence, we included only regions where we anticipated that the number of pigment cells may not be affected by differences in positioning. A mask is created that uses the variables, **ROI - Relative Offset** and **ROI - Relative Length**. Pigment cells are detected applying a threshold (**Autothresh multiplier**).



6.13.1 Parameters



ROI - Relative Offset 0.11

ROI - Relative Length 1

Autothresh multiplier 0.5

☑ plot_binary_shape

☑ showBinary

Opacity 0.5

ROI - Relative Offset –increases or reduces the width of the mask from the head region. Adjust this Offset to avoid the detection of otoliths.

ROI - Relative Length – Increases or reduces the length of the mask from the tail position of the fish embryo.

Autothresh multiplier – modifies the threshold value to identify the pigment cells. Increasing values, increases the threshold to detect darker pigmented cells.

It is important to use the same **Autothresh multiplier** in all images if your objective is to detect changes in pigmentation area (sum area of pigment cells detected). If the threshold would be changed image to image the area of pigment cells would be affected and compromise a comparative analysis!

6.13.2 Display options

plot_binary_shape – show or hide boundaries of the pigment cells detected.

showBinary – show or hide the mask of the area for the location of the pigment cells.

7 ROTATE AND CROP IMAGES WORKFLOW

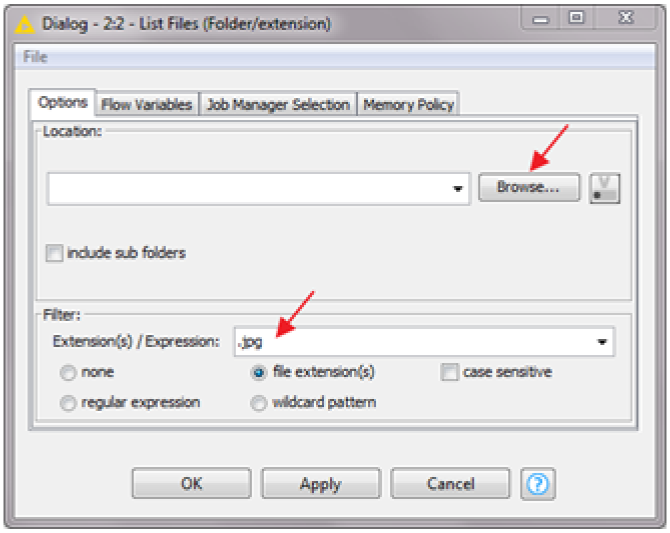
Images not obtained by automatic positioning in a glass capillary - and hence, not presenting capillary boundaries - can be used as well but require automatic conversion to an image with a virtual capillary. This can be done with a KNIME workflow and an embedded imageJ macro (download here). The workflow can handle multiple images simultaneously in a loop. Depending on the image quality certain parameters of the workflow may have to be adjusted. Images of embryos from a lateral orientation are required. The workflow is prepared for RGB images but it can be adapted for grayscale images.

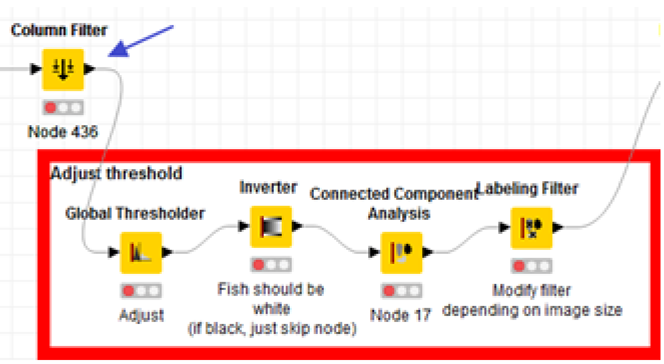
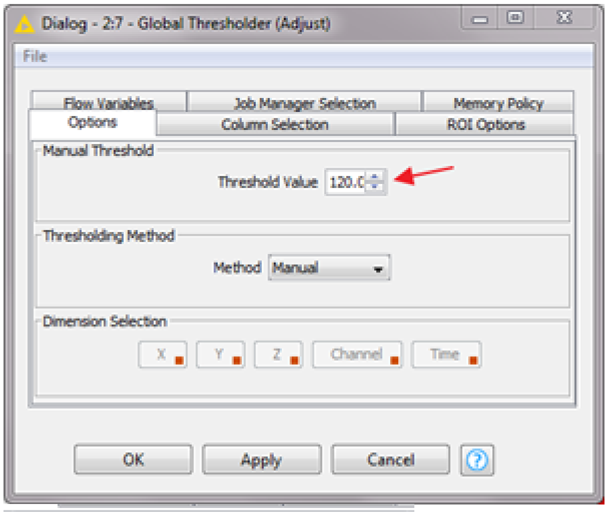
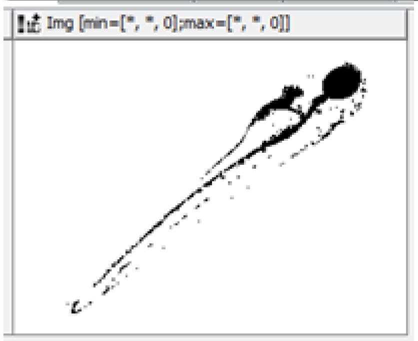
Knime extensions required:

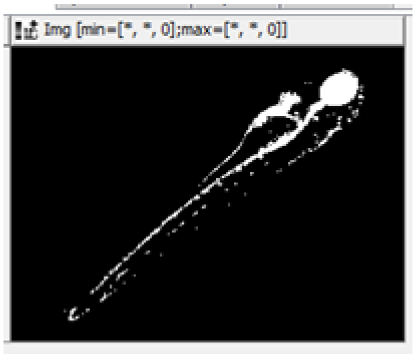
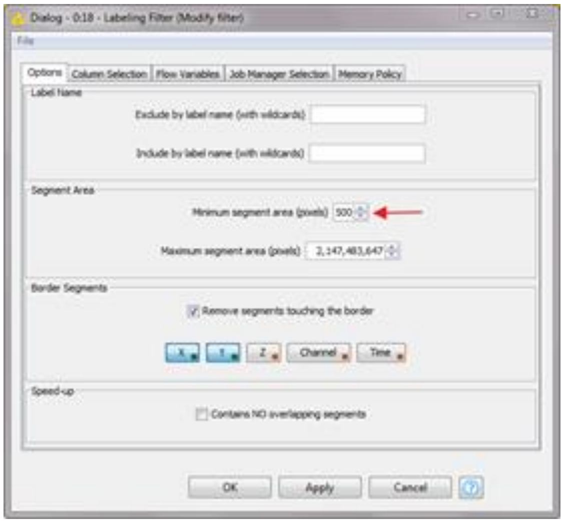
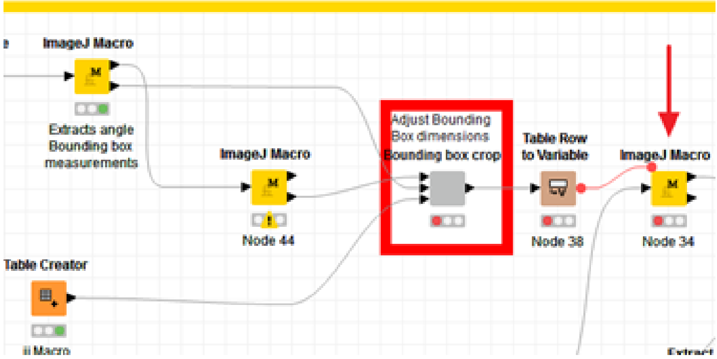
- Knime community contributions - Image Processing
- Image processing - ImageJ extension
- Knime Quick Forms

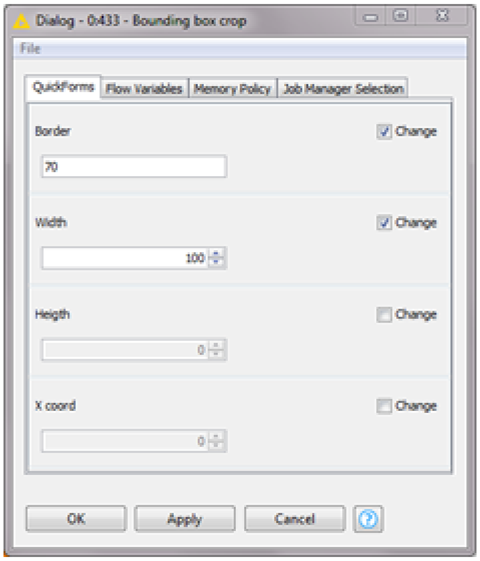
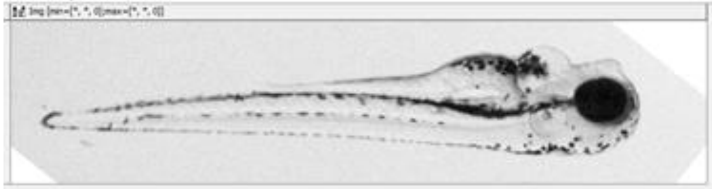
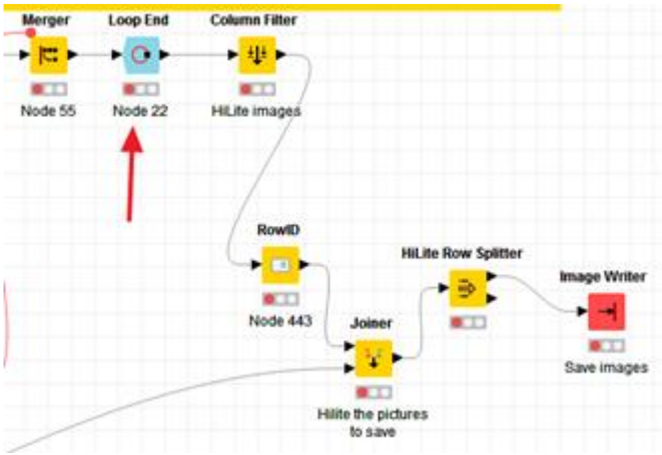
7.1 INSTRUCTIONS

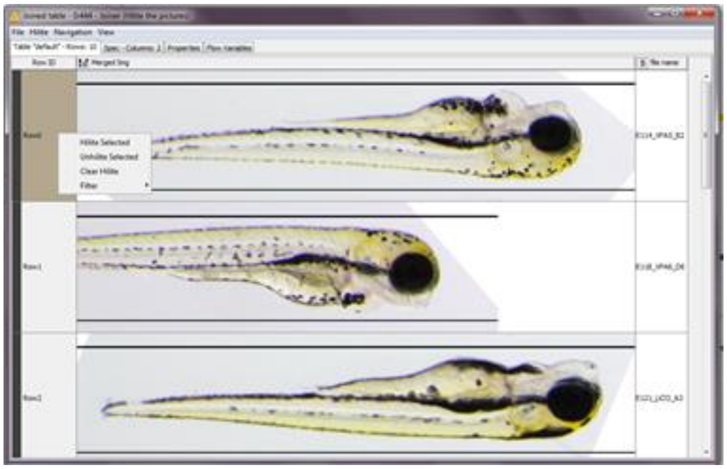
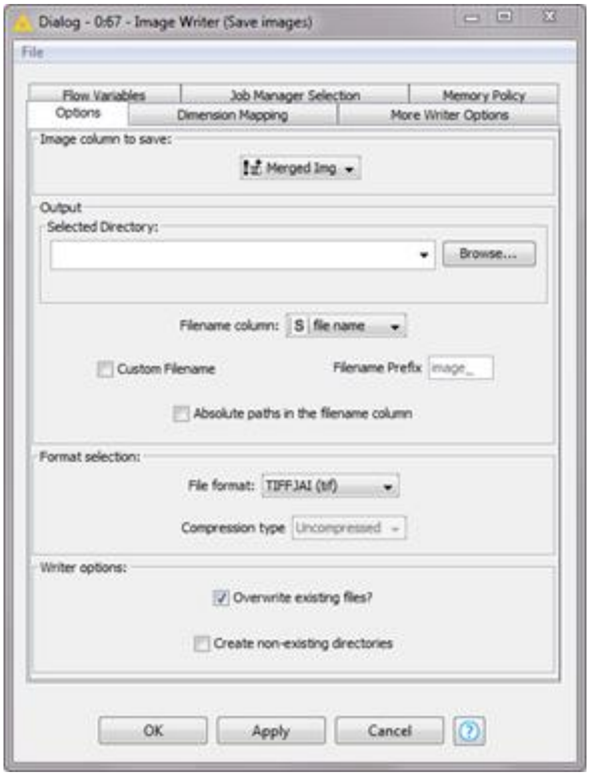
THE WORKFLOW STARTS WITH A LIST FILE NODE TO SELECT THE FOLDER. THIS AND A FEW OTHER NODES MAY REQUIRE SPECIFIC SETTINGS EXPLAINED IN THE TABLE BELOW. PERFORMANCE OF THE WORKFLOW CAN BE CHECKED AT EACH NODE BY A RIGHT MOUSE CLICK AND DISPLAY OF THE INTERMEDIATE RESULTS.

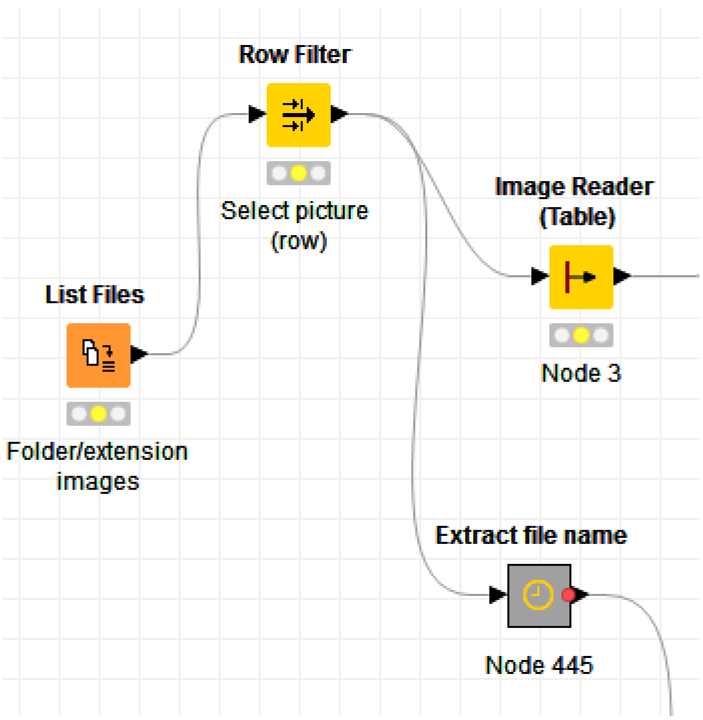
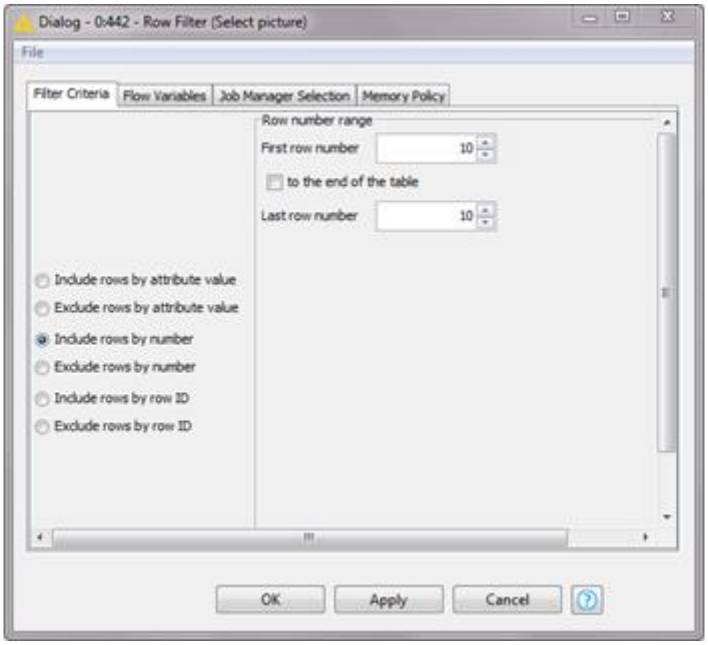
1. **List files node:** Browse to select the folder that contains the images and if necessary change extension. **Note: all pictures must have the same size/dimension**	 Dialog - 2:2 - List Files (Folder/extension) File Options Flow Variables Job Manager Selection Memory Policy Browse... OK Apply Cancel Location: ☐ include sub folders Filter: Extension(s) / Expression: .jpg ○ none ● file extension(s) ☐ case sensitive ○ regular expression ○ wildcard pattern

2. Just for the first picture: click on **Column Filter** node and execute	 Column Filter Node 436 Adjust threshold Global Thresholder Adjust Inverter Fish should be white (if black, just skip node) Connected Component Analysis Node 17 Labeling Filter Modify filter depending on image size
3. Adjust the threshold value on the **Global Thresholder** node. Execute the node and open view to check that the fish embryo is clearly visible in the binary image. Changes of the threshold may be required to adjust for optimal display with low background.	 Dialog - 2:7 - Global Thresholder (Adjust) File Flow Variables Job Manager Selection Memory Policy Options Column Selection ROI Options Manual Threshold Threshold Value 120.0 Thresholding Method Method Manual Dimension Selection X Y Z Channel Time OK Apply Cancel  Img [min=[*, *, 0];max=[*, *, 0]]

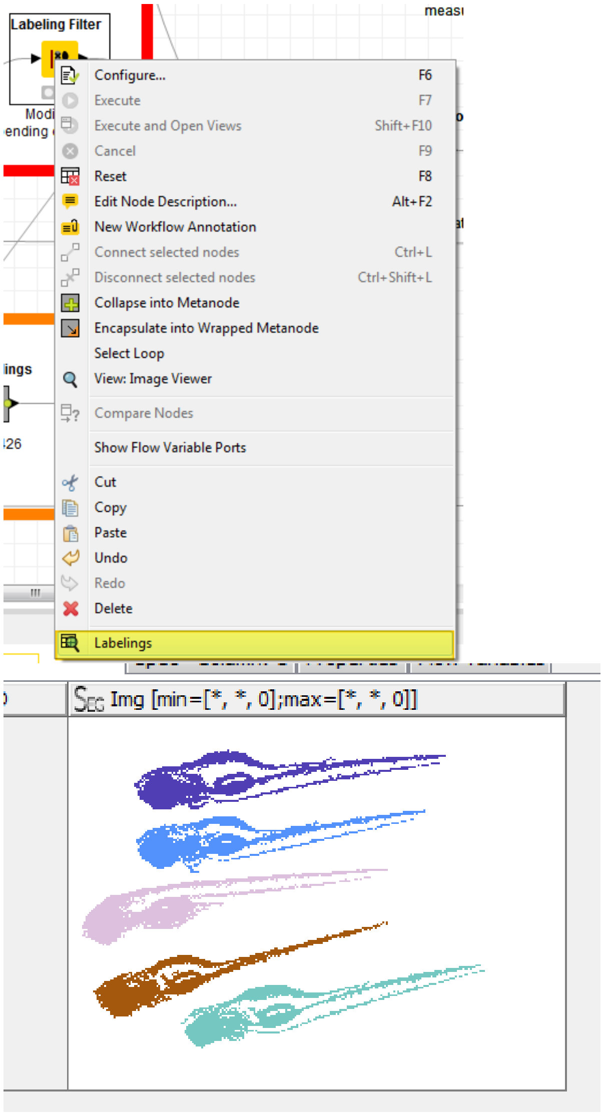
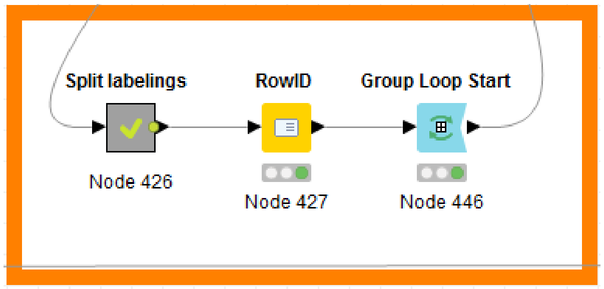
4. The next node inverts the image, if your image has a white background, skip the node. Image should be with black background like the image on the right.	
5. Adjust **labeling filter** node to remove small noise from the background. (settings also depend on image size). Minimum segment area (pixels) should be adjusted if necessary.	 Dialog - 0:18 - Labeling Filter (Modify filter) File Options | Column Selection | Flow Variables | Job Manager Selection | Memory Policy Label Name Exclude by label name (with wildcards) Include by label name (with wildcards) Segment Area Minimum segment area (pixels) 500 Maximum segment area (pixels) 2,147,483,647 Border Segments ☑ Remove segments touching the border X Y Z Channel Time Speed-up ☐ Contains NO overlapping segments OK Apply Cancel
6. Execute the **ImageJ Macro** node after the **Bounding box crop** wrapped metanode in order to check if the image has been cropped and the fish is completely visible and with the head on the right.	 ImageJ Macro Extracts angle Bounding box measurements ImageJ Macro Node 44 Adjust Bounding Box dimensions Bounding box crop Table Row to Variable Node 38 ImageJ Macro Node 34 Table Creator ij Macro Extract

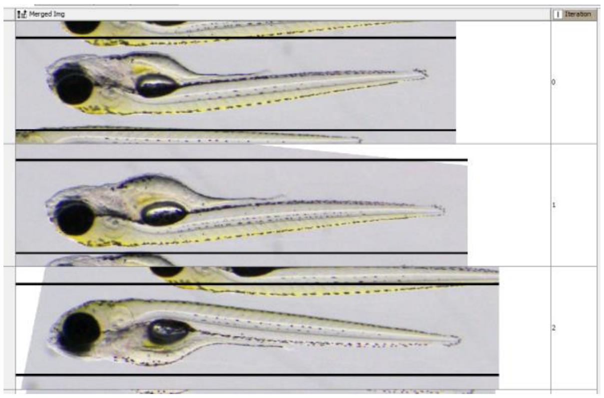
7. Adjust bounding box parameters (height, width, border or x coordinates) by double-click on the **Bounding box crop** wrapped metanode. · Execute and view **ImageJ macro** node to check again. The image should contain the complete fish embryo.	 Dialog - 0:433 - Bounding box crop File QuickForms | Flow Variables | Memory Policy | Job Manager Selection Border ☑ Change 70 Width ☑ Change 100 Heigth □ Change 0 X coord □ Change 0 OK | Apply | Cancel 
8. Execute the last **Loop End** node.	 Merger Node 55 Loop End Node 22 Column Filter HiLite images RowID Node 443 Joiner Hilite the pictures to save HiLite Row Splitter Image Writer Save images

9. Once finish, in the next **Joiner** node. · Right-click and click on **Joiner table.** · Select the images that were correctly processed: In the **Joiner table window**, click on the row ID of the image and then right click and click on **HiLite Selected**. · Write down the row number or name of the images that were not correctly processed.	 File Hilite Navigation View Row ID Merged Img file name Row0 Row1 Row2 Hilite Selected Unhilite Selected Clear Hilite Filter
10. Double-click on the **Image Writer** node and click on Browse to define the folder where you want to save the cropped images.	Dialog - 0:67 - Image Writer (Save images) Image column to save: Merged Img Output Selected Directory: Filename column: S file name □ Custom Filename Filename Prefix image_ □ Absolute paths in the filename column Format selection: File format: TIFFJAI (tif) Compression type Uncompressed Writer options: ☑ Overwrite existing files? □ Create non-existing directories File Flow Variables Job Manager Selection Memory Policy Options Dimension Mapping More Writer Options Browse... OK Apply Cancel

11. To process the images incorrectly cropped. Attached after the **List Files** node a **row filter** node. Filter by row the images that have previously failed to be processed correctly. Note: The row filter should be connected to the Image Reader and also to the Extract file name metanode.	 Row Filter Select picture (row) Image Reader (Table) Node 3 List Files Folder/extension images Extract file name Node 445  Dialog - 0:442 - Row Filter (Select picture) File Filter Criteria Flow Variables Job Manager Selection Memory Policy Row number range First row number 10 □ to the end of the table Last row number 10 ○ Include rows by attribute value ○ Exclude rows by attribute value ● Include rows by number ○ Exclude rows by number ○ Include rows by row ID ○ Exclude rows by row ID
12. Adjust **bounding box crop** parameters so that the entire fish embryo is displayed in the image window and execute the workflow again.	

Images with multiple embryos on it could also be used, the requirement is that embryos do not touch each other. The workflow is similar and I is named: Rotate and crop image_multiple.

1. Proceed in the same way as the previous workflow up to step 5.	
2. Check in the labeling filter node that embryos are identify with a unique label (color). - Right-click on the Labeling Filter node and select Labelings. - If you see multiple labels (colors) in the same fish, go to configure Global Thersholder node and increase threshold value. - If there are a lot of noise, increase the Minimum segment area (pixels) under the configuration of Labeling filter node.	 Labeling Filter Configure... F6 Execute F7 Execute and Open Views Shift+F10 Cancel F9 Reset F8 Edit Node Description... Alt+F2 New Workflow Annotation Connect selected nodes Ctrl+L Disconnect selected nodes Ctrl+Shift+L Collapse into Metanode Encapsulate into Wrapped Metanode Select Loop View: Image Viewer Compare Nodes Show Flow Variable Ports Cut Copy Paste Undo Redo Delete Labelings Img [min=[*, *, 0];max=[*, *, 0]]
3. The nodes inside de orange box, split individually the labels, so embryos can individually be cropped and rotated.	 Split labelings RowID Group Loop Start Node 426 Node 427 Node 446

4. Follow steps from 6 to 10 of the previous workflow.	 Merged Img Iteration 0 1 2